描述

P89C660/662/664 包括一个非易失性的 16KB/32KB/64KB 两个可平行编程并且是串行的系统内应用上的可编程的 Flash 存储器。系统内编程（ISP）允许用户下载新的代码而控制器处于应用中。应用上编程（IAP）即指控制器去取新程序代码，并对自身再次编程，当处于系统中。这允许远程编程通过一个 moden 连接。ROM 里的一个默认的串行 loader（boot loader）编程允许 Flash 存储器系统内编程经过 UART，而没有对 loader 的需要在 Flash 代码里。对于应用上编程，通过对 ROM 里包括的标准子程的使用用户程序对 Flash 存储器擦写和再编程。

这个器件在 6 个时钟周期内执行一条指令，是传统的 80C51 的两倍。一个 OTP 结构位让用户选择传统的 12 个时钟周期。

这个器件是个 8 位微型控制器制造领先于 CMOS 工艺，衍生于 80C51 控制器家族。这个指令集和 80C51 的指令集是 100％的执行和时序兼容。

这个器件有四个 8 位 I/O 端，三个 16 位定时器/计数器，一个复用源，四个优选级，嵌套中断结构，一个强大的 UART 和片内振荡器以及时序电路。

P89C660/662/664 增加的特征是使其成为一个强大的控制器，为应用要求一个 PWM，高速的 I/O 和加/减计数能力，像汽车控制。

特征

- 80C51 中央处理单元
- 带有 ISP 和 IAP 的片内 Flash 存储器
- Boot ROM 包括低级别的 Flash 编程子程用于经过 UART 下载
- 可由 IAP 编程
- 每个机器周期 6 个时钟周期操作（标准）
- 每个机器周期 12 个时钟周期操作（属性）
- 在每个机器周期 6 个时钟周期下，速度高达 20MHz（相当于 40MHz 性能）；在每个机器周期 12 个时钟周期下，速度高达 33MHz
- 完全静态操作
- RAM 可扩展到外部 64K 字节
- 4 个优先级中断
- 8 个中断源
- 4 个 8 位 I/O 端
- 全双工强大的 UART
  - 一桢错误检测
  - 一自动地址识别
- 电压控制模式
  - 一时钟可被中止和继续
  - 一空闲模式
  - 一掉电模式
- 可编程的时钟输出
- 两个 DPTR 寄存器
- 一个同步端口复位
- 低 EMI（禁止 ALE）
- I^2C 串行接口
- 可编程的计数器阵列（PCA）

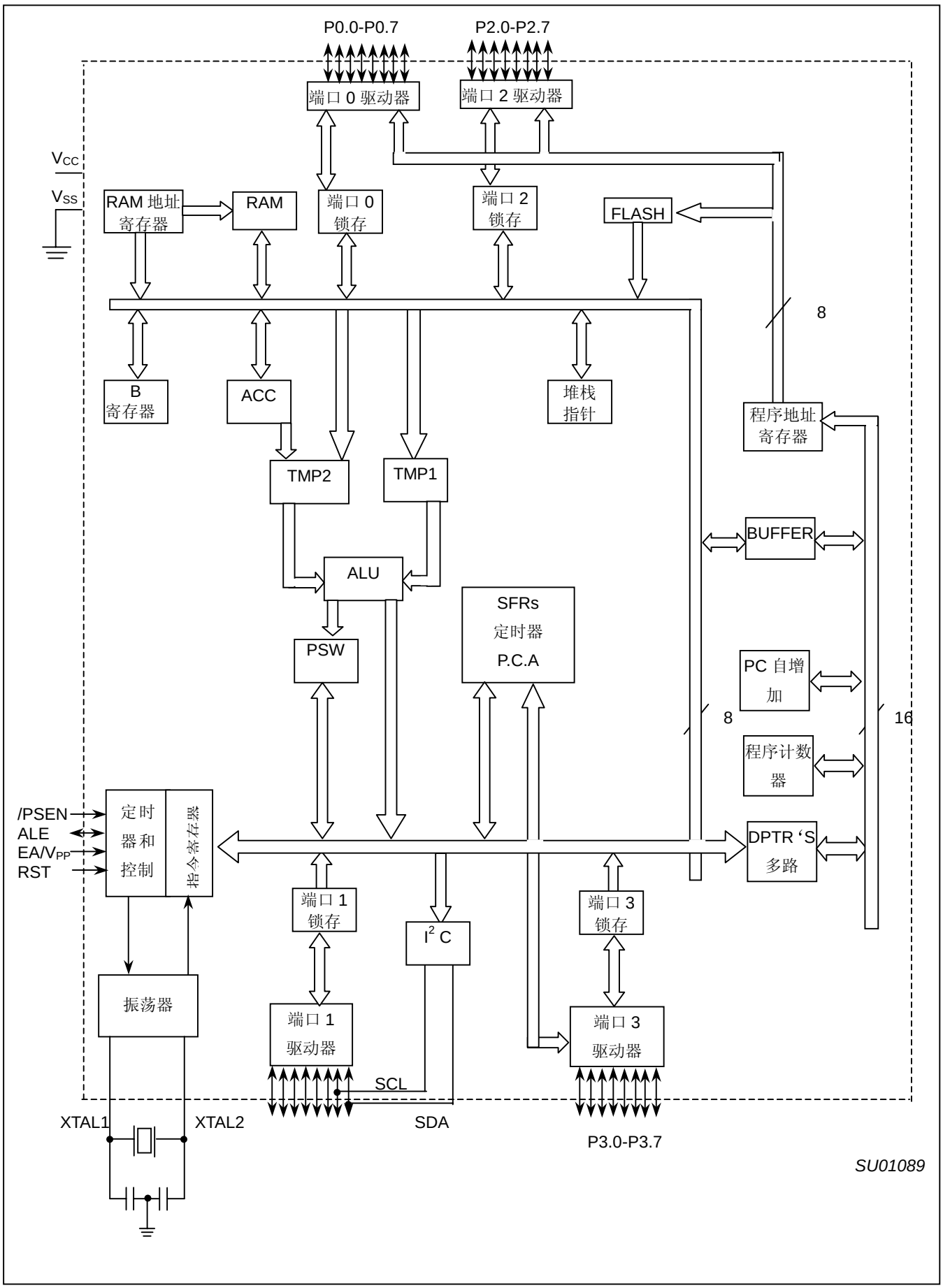
—PWM

—捕捉/比较

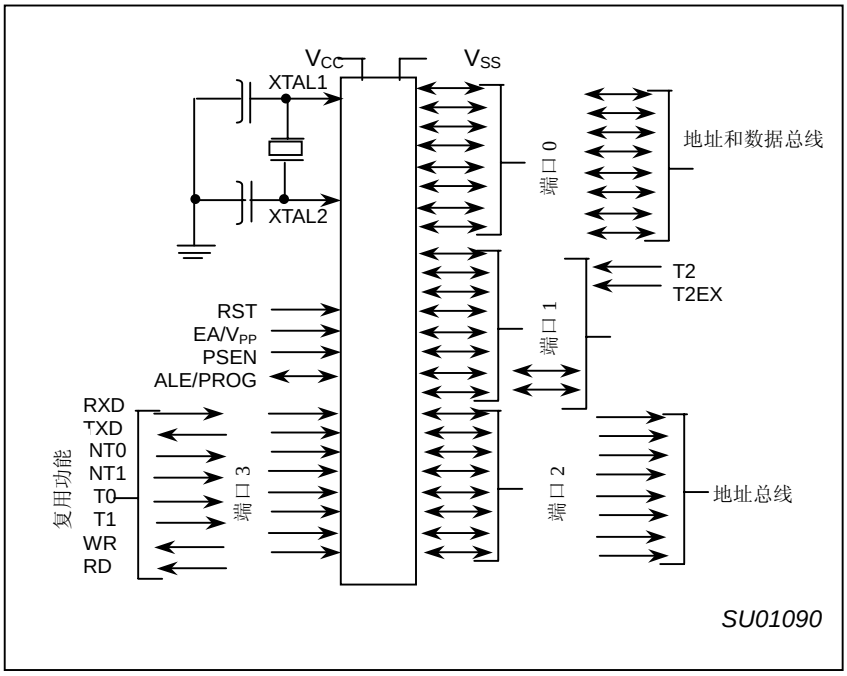
• 恰当地适合于 IPMI 应用

订购信息

器件	存储		温度范围和封装	电压范围	频率（MHz）		DWG #
	Flash	RAM			6 时钟周期模式	12 时钟周期模式	
P89C660HBA	16KB	512B	0-70,PLCC	4.5-5.5V	0-20 MHz	0-33 MHz	SOT187-2
P89C660HFA	16KB	512B	-40-+85,PLCC	4.75-5.25V	0-20 MHz	0-33 MHz	SOT187-2
P89C660HBBD	16KB	512B	0-70,LQFP	4.5-5.5V	0-20 MHz	0-33 MHz	SOT389-1
P89C662HBA	32KB	1KB	0-70,PLCC	4.5-5.5V	0-20 MHz	0-33 MHz	SOT187-2
P89C662HFA	32KB	1KB	-40-+85,PLCC	4.75-5.25V	0-20 MHz	0-33 MHz	SOT187-2
P89C662HBBD	32KB	1KB	0-70,LQFP	4.5-5.5V	0-20 MHz	0-33 MHz	SOT389-1
P89C662HFBD	32KB	1KB	-40-+85, LQFP	4.75-5.25V	0-20 MHz	0-33 MHz	SOT389-1
P89C664HBA	64KB	2KB	0-70,PLCC	4.5-5.5V	0-20 MHz	0-33 MHz	SOT187-2
P89C664HFA	64KB	2KB	-40-+85,PLCC	4.75-5.25V	0-20 MHz	0-33 MHz	SOT187-2
P89C664HBBD	64KB	2KB	0-70,LQFP	4.5-5.5V	0-20 MHz	0-33 MHz	SOT389-1
P89C664HFD	64KB	2KB	-40-+85, LQFP	4.75-5.25V	0-20 MHz	0-33 MHz	SOT389-1

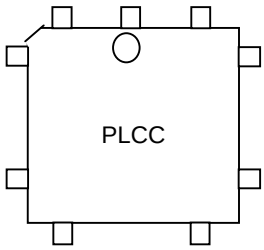

P0.0-P0.7
P2.0-P2.7
端口 0 驱动器
端口 2 驱动器
VCC
VSS
RAM 地址寄存器
RAM
端口 0 锁存
端口 2 锁存
FLASH
8
B 寄存器
ACC
堆栈指针
程序地址寄存器
TMP2
TMP1
BUFFER
ALU
SFRs 定时器 P.C.A
PSW
PC 自增加
8
16
程序计数器
/PSEN
ALE
EA/VPP
RST
定时器和控制
指令寄存器
DPTR ‘S 多路
端口 1 锁存
I2C
端口 3 锁存
振荡器
端口 1 驱动器
端口 3 驱动器
SCL
SDA
XTAL1
XTAL2
P3.0-P3.7
SU01089

逻辑标号



管脚

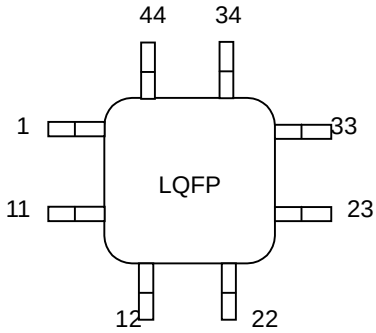
PLCC 封装



管脚	功能	管脚	功能	管脚	功能
1	NIC*	16	P3.4/T0/CEX3	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1/CEX4	32	PSEN
3	P1.1/T2EX	18	P3.6/WR	33	ALE
4	P1.2/ECI	19	P3.7/RD	34	NIC*
5	P1.3/CEX0	20	XTAL2	35	EA/ V_{PP}
6	P1.4/CEX1	21	XTAL1	36	P0.7/AD7
7	P1.5/CEX2	22	V_{SS}	37	P0.6/AD6
8	P1.6/SCL	23	NIC*	38	P0.5/AD5
9	P1.7/SDA	24	P2.0/AB	39	P0.4/AD4
10	RST	25	P2.1/A9	40	P0.3/AD3
11	P3.0/RxD	26	P2.2/A10	41	P0.2/AD2
12	NIC*	27	P2.3/A11	42	P0.1/AD1
13	P3.1/TxD	28	P2.4/A12	43	P0.0/AD0
14	P3.2/$\overline{INT0}$	29	P2.5/A13	44	V_{CC}
15	P3.3/$\overline{INT1}$	30	P2.6/A14		

*无内部连接

PQFP 封装



管脚	功能	管脚	功能	管脚	功能
1	P1.5/CEX2	16	V_{SS}	31	P0.6/AD6
2	P1.6/SCL	17	NIC*	32	P0.5/AD5
3	P1.7/SDA	18	P2.0/AB	33	P0.4/AD4
4	RST	19	P2.1/A9	34	P0.3/AD3
5	P3.0/RxD	20	P2.2/A10	35	P0.2/AD2
6	NIC*	21	P2.3/A11	36	P0.1/AD1
7	P3.1/TxD	22	P2.4/A12	37	P0.0/AD0
8	P3.2/$\overline{INT0}$	23	P2.5/A13	38	V_{CC}
9	P3.3/$\overline{INT1}$	24	P2.6/A14	39	NIC*
10	P3.4/T0/CEX3	25	P2.7/A15	40	P1.0/T2
11	P3.5/T1/CEX4	26	PSEN	41	P1.1/T2EX
12	P3.6/WR	27	ALE	42	P1.2/ECI
13	P3.7/RD	28	NIC*	43	P1.3/CEX0
14	XTAL2	29	EA/ V_{PP}	44	P1.4/CEX1
15	XTAL1	30	P0.7/AD7		

*无内部连接

管脚描述

助记符	管脚号		类型	名称和功能
	PLCC	PQFP		
V_{SS}	22	16	I	**接地**：0V 参考
V_{CC}	44	38	I	**提供电压**：为正常，空闲，和掉电模式提供电压。
P0.0-P0.7	43-46	37-30	I/O	**端口 0**：端口 0 是个开漏，准双向 I/O 端，端口 0 管脚对它们悬空有 1s 的写入时间和能被用作高阻抗输入。端口 0 也是复用的低字节地址和数据总线在对外部程序存储器和数据存储器访问期间。在这个应用里，它使用强上拉当 emitting 1s。
P1.0-P1.7	2—9	40—44 1—3	I/O	**端口 1**：端口 1 是个 8 位准双向在所有管脚带有内部上拉除了 P1.6 和 P1.7 是开漏的。端口 1 管脚有 1s 写入时间，被内部上拉为高，能用作输入。作为输入，被外部拉为低端口 1 管脚，will source current 因为内部的上拉。(见 DC 电气特性：I_{IL}) P89C660/662/664 端口 1 复用功能包括：
	2	40	I/O	**T2（P1.0）**：定时器/计数器 2 外部计数输入/时钟输出（见可编程的时钟输出）
	3	41	I	**T2EX（P1.1）**：定时器/计数器 2 重载/捕捉/方向控制。
	4	42	I	**ECI（P1.2）**：对 PCA 的外部时钟输入
	5	43	I/O	**CEX0（P1.3）**：捕捉/比较外部 I/O 用于 PCA 的模式 0
	6	44	I/O	**CEX1（P1.4）**：捕捉/比较外部 I/O 用于 PCA 的模式 1
	7	1	I/O	**CEX2（P1.5）**：捕捉/比较外部 I/O 用于 PCA 的模式 2
	8	2	I/O	**SCL（P1.6）**：I^2C 总线时钟线（开漏）
	9	3	I/O	**SDA（P1.7）**：I^2C 总线数据线（开漏）
P2.0-P2.7	24-31	18-25	I/O	**端口 2**：端口 2 是个 8 位准双向带有内部上拉的 I/O 端。端口 2 管脚有 1s 被写入时间，被内部上拉为高，能用作输入。作为输入，端口 2 管脚被外部拉为低 will source current 因为内部上拉。(见 DC 电气特性：I_{IL}) 端口 2 放射出高字节地址在从外部程序存储器取回代码期间以及对外部数据存储器访问期间，使用 16 位地址（MOVX @DPTR)。在这个应用里，它使用内部强上拉当 emitting 1s。在对外部数据存储器访问期间，使用 8 位地址（MOVX Ri)。端口 2 放射出 P2 特殊功能寄存器的内容。
	11, 13-19	5，7-13	I/O	**端口 3**：端口 3 是个 8 位准双向带有内部上拉的 I/O 端。端口 2 管脚有 1s 被写入时间被内部上拉为高，能用作输入。作为输入，端口 2 管脚被外部拉为低 will source current 因为内部上拉。(见 DC 电气特性：I_{IL})端口 3 也服务于 P89C660/662/664 的特殊特征，如下列表所示：
	11	5	I	**RxD（P3.0）**：串行输入端
	13	7	O	**TxD（P3.1）**：串行输出端
	14	8	I	**/INT0（P3.2）**：外部中断
	15	9	I	**/INT1（P3.3）**：外部中断
	16	10	I	**CEX3/T0（P3.4）**：定时器 0 外部输入；捕捉/比较外部 I/O，用于 PCA 模式 3
	17	11	I	**CEX4/T1（P3.5）**：定时器 1 外部输入；捕捉/比较外部 I/O，用于 PCA 模式 4
	18	12	O	**/WR（P3.6）**：外部数据存储器写选通
	19	13	O	**/RD（P3.7）**：外部数据存储器读选通
RST	10	4	I	**Reset**：在这个管脚上为高持续为两个机器周期，当振荡器正在运行，使这个器件复位。接到 V_{SS} 一个内部电阻允许上电复位，仅使用一个外部电容接 V_{CC}
ALE	33	27	O	**地址锁存使能**：输出脉冲用于锁定地址的低字节在对外部存储器访问期间。在正常操作下，每个机器周期 ALE 被发射两次，可用于外部定时或计时。注意一个 ALE 脉冲被忽略在对外部数据存储器访问期间。ALE 能被无效，通过对 SFR auxiliary.0. 置位。因为这个位的置位，ALE 仅在 MOVX 指令中有效。
/PSEN	32	26	O	**程序存储选通**：对外部程序存储吕的读选通。当从外部程序程序器执行代码，/PSEN 是每个机器周期有效两次，除了两个 PSEN 有效被忽略在对外部数据存储器访问期间。在从内部程序存储器取指期间，/PSEN 无效。
(/EA) /V_{PP}	35	29	I	**外部访问使能/编程电压提供**：/EA 必须被保持为低，使器件从外部程序存储器取指令有效。如果/EA 被保持为高，器件从内部程序存储器执行。/EA 管脚上值被锁定当 RST 被释放，任何变化都没有作用。这个管脚也接收编程提供电压（V_{PP}）在 Flash 编程期间。
XTAL1	21	15	I	**晶振 1**：振荡器的输入，内部时钟产生器的输入
XTAL2	20	14	O	**晶振 2**：振荡器的输出

注：

为了避免“锁定”作用在上电时，任何管脚上电压（除了 V_{PP}）不能高于 V_{CC}+0.5V 或低于 V_{CC}—0.5V。

表 1 特殊功能寄存器

标记	描述	直接地址	位地址，标号，或复用端功能 MSB							LSB	复位值
ACC*	累加器	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	辅助	8EH	—	—	—	—	—	—	EXTRAM	A0	xxxxxx10B
AUXR1#	辅助 1	A2H	—	—	ENBOOT	—	GF2	0	—	DPS	xxxxx0x0B
B*	B 寄存器	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	模为 0 捕捉高	FAH									xxxxxxxxB
CCAP1H#	模为 1 捕捉高	FBH									xxxxxxxxB
CCAP2H#	模为 2 捕捉高	FCH									xxxxxxxxB
CCAP3H#	模为 3 捕捉高	FDH									xxxxxxxxB
CCAP4H#	模为 4 捕捉高	FEH									xxxxxxxxB
CCAP0L#	模为 0 捕捉低	EAH									xxxxxxxxB
CCAP1L#	模为 1 捕捉低	EBH									xxxxxxxxB
CCAP2L#	模为 2 捕捉低	ECH									xxxxxxxxB
CCAP3L#	模为 3 捕捉低	EDH									xxxxxxxxB
CCAP4L#	模为 4 捕捉低	EEH									xxxxxxxxB
CCAPM0#	模为 0 模式	C2H	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	模为 1 模式	C3H	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	模为 2 模式	C4H	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	模为 3 模式	C5H	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	模为 4 模式	C6H	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
			C7	C6	C5	C4	C3	C2	C1	C0	
CCON* #	PCA 计数器控制	C0H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00x0000B
CF#	PCA 计数器高	F9H									00H
CL#	PCA 计数器低	E9H									00H
CMOD#	PCA 计数器模式	C1H	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF	00xxx000B
DPTR:	数据指针（2 字节）										
DPH	数据指针高	83H									00H
DPL	数据指针低	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IEN0*	中断使能 0	A8H	EA	EC	ES1	ES0	ET1	EX1	ET0	EX0	00H
IEN1*	中断使能 1	E8	—	—	—	—	—	—	—	ET2	xxxxxxx0B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	中断优先级	B8H	PT2	PPC	PS1	PS0	PT1	PX1	PT0	PX0	x0000000B
IPH#	中断优先级高		B7	B6	B5	B4	B3	B2	B1	B0	
IPH#	中断优先	B7H	PT2H	PPCH	PS1H	PS0H	PT1H	PX1H	PT0H	PX0H	x0000000B

	级高									
			87	86	85	84	83	82	81	80
P0-*	端口 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90
P1*	端口 1	90H	SDA	SCL	CEX2	CEX1	CEX0	ECI	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0
P2*	端口 2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0
P3*	端口 3	B0H	/RD	/WR	T1/CEX4	T0/CEX3	/INT1	/INT0	TxD	RxD	FFH
PCON#[1]	电压控制	87H	SM0D1	SM0D0	—	POF	GF1	GF0	PD	IDL	00xxx000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	程序状态字	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00000000B
RCAP2H#	定时器 2 捕捉高	CBH									00H
RCAP2L#	定时器 2 捕捉低	CAH									00H
SADDR#	从机地址	A9H									00H
SADEN#	从机地址屏蔽	B9H									00H
S0BUF	串行数据缓冲器	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
S0CON*	串行控制	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	堆栈指针	81H									07H
S1DAT#	串行 1 数据	DAH									00H
S1IST	串行 1 内部状态	DCH									xxxxxxxx
		DB H	从机地址							GC	00H
S1STA#	串行 1 状态	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
			DF	DE	DD	DC	DB	DA	D9	D8	
S1CON*#	串行 1 控制	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00000000B
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	定时器控制	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*	定时器 2 控制	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
T2MOD#	定时器 2 模式控制	C9H	—	—	—	—	—	—	T2OE	DCEN	Xxxxxx00B
TH0	定时器 0 高	8CH									00H
TH1	定时器 1 高	8DH									00H
TH2#	定时器 2 高	CDH									00H
TL0	定时器 0 低	8AH									00H
TL1	定时器 1 低	8BH									00H
TL2#	定时器低 2	CCH									00H
TMOD	定时器模式	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDTRST	看门狗定时器复位	A6H									

*SFRs 是可寻址。＃SFRs 被修改过的或被加到 80C51 SFRs。—保留位。

振荡器特征

XTAL1 和 XTAL2 分别是反相放大器的输入和输出。这个管脚可被作为片内振荡使用。

为了从外部时钟源驱动这个器件，XTAL1 应被驱动而 XTAL2 不被连接。应注意到最小和最大的高和低时序在数据表中被规格化。

这个器件在每个机器周期 6 个时钟周期下运行，在数据表中可被查到作为“6 个时钟模式”（这个性能和标准 80C51 系列器件的两倍频率相同）。在可达到的商业级 EPROM 编程上它可被有选择地构造运行，和每个机器周期 12 个时钟周期相同，在数据表中可被查到作为“12 个时钟模式”。一旦 12 个时钟模式被构成，则它不能被改回到 6 个时钟模式。

复位

通过将 RST 管脚保持为高至少两个机器周期（在 6 个时钟模式里是 12 个振荡周期，或 24 个振荡周期在 12 个时钟模式里），一个复位被完成，当振荡器正在运行时。为了确保一个很好的上电复位，RST 管脚为高必须足够长以允许振荡器启动（正常地几微秒）加两个机器周期。在上电时，V_{CC} 和 RST 上的电压必须同时增加以便一个适当的启动。端口 1，2，3 必须被异步地驱动为它们的复位条件当电压高于 V_{IH1}（最小）被应用在 RESET。EA 管脚上值被锁定当 RST 被 deasserted 没有进一步的作用。

低电压模式

停止时钟模式

静态设计使时钟速度可被降低到 0MHz（被停止）。当振荡器被停止，RAM 和特殊功能寄存器保留它们的值。这个模式允许一步一步地利用，允许降低的系统电压消耗，通过降低时钟频率到任何值。最低的电压消耗掉电模式被建议采纳。

空闲模式

在空闲模式里（见表 2），CPU 处于休眠状态而片内所有的周边器件保持活跃。调入空闲模式的指令是最后被执行的指令在正常操作模式下在空闲模式活跃以前。CPU 内容，片内 RAM，所有特殊功能寄存器保持不变在这个模式期间。这个空闲模式可被终止通过一个有效的中断（在中断服务子程时，过程被开始并继续），或启动处理器在同样的方式下作为上电复位的一个硬件复位。

掉电模式

为了节约更多的能量，一个掉电模式可被软件调入。在这个模式里，持振荡器被终止，调入的掉电指令是最后被执行的指令。片内 RAM 和特殊功能寄存器保持它们的值，V_{CC} 必须回到最小的规格化的电压在掉电模式被中止前。

一个硬件中断或外部复位可用于从掉电模式中退出。复位再次定义了 SFR，但没有改变片内 RAM。一个外部中断允许 SFR 和片内 RAM 保留它们的值。

为了很好地中止掉电，复位或外部中断不应被执行在 V_{CC} 被恢复到它的正常操作电压前，必须被保持活跃足够长，以便振荡器能够重新启动和稳定（正常地低于 10ms）。

带有外部中断，INT0 和 INT1 必须被使能和构成为电平敏感型。保持这个管脚为低，振荡器能够重新启动，但返回为高时将完成退出。一旦中断被服务，下一条被执行指令在 RETI 后将是紧接那条使器件处于掉电模式的指令。

掉电标记

掉电标记（POF）由片内电路置位当 P89C660/662/664 上 V_{CC} 电压从 0 到 5V。POF

位由软件置位或清除，允放用户决定复位是上电的结果还是掉电后热启动的结果。V_{CC} 电压必须保持为高于 3V，使 POF 保持不受 V_{CC} 电压的影响。

设计考虑

当空闲模式被硬件复位中止，器件正常恢复到程序执行，从它离开的位置，高达两个机器周期在内部复位算法控制之前。在这个事件里，片内硬件禁止访问内部 RAM，但可以对端口引脚访问。为了消除预料之外的写的可能性当空闲被复位中止时，紧接着调入空闲指令的那条指令不应是对端口引脚的写入或外部存储器写入的那条指令。

在线仿真模式（ONCE™）

在线仿真模式（“on circuit Emulation）促进系统的测试和调试，器件不需要从电路中移走。这个 ONCE 模式由下面调入：

1．将 ALE 置为低，当器件处于复位及/PSEN 为高。

2．ALE 保持为低，当 RST 不活跃时。

当器件处于 ONCE 模式，端口 0 管脚进入悬浮状态，其他管脚的引脚和 ALE 和/PSEN 被弱上拉。振荡器电路保持有效。当器件处于这个模式下，一个仿真器或测试 CPU 可用于驱动这个电路。正常操作被恢复当一个正常复位被应用。

可编程的时钟输出

一个 50％周期循环时钟可被编程输出到 P1.0。这个管脚除了作为一个固定的 I/O 外，还有两个作复用功能，它能被编程为：

1．为定时器/计数器 2 输入外部时钟或

2．输出一个 50％周期循环时钟范围从 122Hz 到 8MHz 在一个 16 MHz 操作频率下（在 12 个时钟模式里，是 61Hz 到 4MHz）。

为了形成定时器/计数器 2 作为一个时钟产生器，位 C/T2（在 T2CON）必须被清除，T2MOD 里的位 T20E 必须被置位。位 TR2（T2CON.2）也必须被置位去重新启动这个定时器。

时钟输出频率取决于振荡器频率和定时器 2 捕捉寄存器的重载值（RCAP2H，RCAP2L），如下列方程所示：

$$n=\frac{\text{振荡器频率}}{n\times(65536-\text{RCAP2H，RCAP2L})}$$

n= 2 在 6 个时钟模式里

4 在 12 个时钟模式里

（RCAP2H，RCAP2L）＝RCAP2H 和 RCAP2L 的内容，被看作为 16 位不带符号的整数。

在时钟输出模式里，定时器 2roll over 将不会产生一个中断。这和用作波特率产生器类似。将定时器 T2 用作波特率产生器和同时作为时钟产生器是可行的。注意，但是波特率和时钟输出频率是一样的。

表 2 外部管脚状态在空闲和掉电模式期间

模式	程序存储器	ALE	/PSEN	端口 0	端口 1	端口 2	端口 3
空闲	内部	1	1	数据	数据	数据	数据
空闲	外部	1	1	悬空	数据	地址	数据
掉电	内部	0	0	数据	数据	数据	数据
掉电	外部	0	0	悬空	数据	数据	数据

I^2C 串行通信－SIO1

I^2C 串行端口和 8XC554，8XC654，8XC652 器件上的 I^2C 串行端口一样。子系统的操作在 80C554、83C654、87C554 数据表中有详细地描述。

注意在P89C660/662/664和8XC552的I^2C管脚有复用功能对端口管脚P1.6和P1.7。因为这个，在这些部份上的 P1.6 和 P1.7 没有上拉结构，因此 P1.6 和 P1.7 在 P89C660/662/664 上有开漏输出。

串行控制寄存器（S1CON）－见表 3

S1CON（D8H）

CR2	ENS1	STA	STO	SI	AA	CR1	CR0

位 CR1，CR0 和 CR2 决定串行时钟频率，即就是由主机操作模式中产生的。

表 3 串行时钟频率

6 时钟模式								
			位频率（kHz）在 f_{osc}					
CR2	**CR1**	**CR0**	**3MHz**	**6MHz**	**8MHz**	**12 MHz[2]**	**15MHz[2]**	**f_{osc} 分频数**
0	0	0	23	47	62.5	94	117[1]	128
0	0	1	27	54	71	107[1]	134[1]	112
0	1	0	31	63	83.3	125[1]	156[1]	96
0	1	1	37	75	100	150[1]	188[131]	80
1	0	0	6.25	12.5	17	25	31	480
1	0	1	50	100	133[1]	200[1]	250[1]	60
1	1	0	100	200	267[1]	400[1]	500[1]	30
1	1	1	0.24<62.5	0.49<62.5	0.65<55.6	0.98<50.0	1.22<52.1	48×（256－（定时器 1 重载值））
			0<255	0<254	0<253	0<251	0<250	定时器 1 重载值在模式 2
12 时钟模式								
			位频率（kHz）在 f_{osc}					
CR2	**CR1**	**CR0**	**3MHz**	**6MHz**	**8MHz**	**12 MHz[2]**	**15MHz[2]**	**f_{osc} 分频数**
0	0	0	23	47	62.5	94	117[1]	256
0	0	1	27	54	71	107[1]	134[1]	224
0	1	0	31	63	83.3	125[1]	156[1]	192
0	1	1	37	75	100	150[1]	188[131]	160
1	0	0	6.25	12.5	17	25	31	960
1	0	1	50	100	133[1]	200[1]	250[1]	120
1	1	0	100	200	267[1]	400[1]	500[1]	60
1	1	1	0.24<62.5	0.49<62.5	0.65<55.6	0.98<50.0	1.22<52.1	96×（256－（定时器 1 重载值））
			0<255	0<254	0<253	0<251	0<250	定时器 1 重载值在模式 2

注：

1．这些频率超过 I^2C 总线规格的 100KHz 的上限，不能在 I^2C 总线应用中使用。

2．f_{osc}＝12MHz/15MHz，最大的 100KHz 的 I^2C 总线频率不能被识别，由于固定的分频率。

3．f_{osc}＝24MHz/30MHz，最大的 100KHz 的 I^2C 总线频率不能被识别，由于固定的分频率。

定时器 2 操作

定时器 2

定时器 2 是个 16 位定时器/计数器，可以既作为定时器，又作为计数器使用，在特殊功能寄存器 T2CON 的 C/T2 位选择（见图 1）。定时器有三个操作模式：捕捉，自动重载（加或减计数），波特率产生器，由 T2CON 的位选择，如表 4 所示。

捕捉模式

在捕捉模式，有两个属性由 T2CON 里的 EXEN2 位选择。如果 EXEN2＝0，则定时器 2 是个 16 位定时器和计数器（由 T2CON 的 C/T2 位选择），一旦溢出，TF2 位被置位，定

时器 2 溢出位。这位可用作产生一个中断（通过使 IE 寄存器的的定时器 2 中断位使能）。如果 EXEN2＝1，定时器 2 作为上面描述的操作，但增加了一个在外部输入 T2EX 上 1－0 的瞬变，导致在定时器寄存器 2 里，TH2，TL2 分别被捕捉进寄存器 RCAP2L 和 RCAP2H。另外，T2EX 上的瞬变导致 T2CON 里的 EXF2 位被置位，以及 EXF2 位像 TF2 位能产生一个中断（它的矢量和定时器 2 溢出中断位置一样定时器 2 中断服务子程能够查询 TF2 和 EXF2 去决定何种事情导致的中断）。捕捉模式如图 2 所示（在这个模式里，TL2 和 TH2 没有重载值，甚至当一个捕捉事件发生在 T2EX，计数器保持对 T2EX 管脚上瞬变计数或 osc/6 脉冲（在 12 时钟模式里，是 osc/12））。

自动重载模式（加或减计数器）

在 16 位自动重载模式里，定时器 2 可被构造作为定时器或计数器，然后可被编程加或减。计数方向由 DCEN 位（减计数器使能）决定，该位位于 T2MOD 寄存器里（见图 3）。当复位被应用于 DCEN＝0，即指定时器 2 将默认为加。如果 DCEN 被置位，定时器 2 可加或减取决于 T2EX 管脚上的值。

图 4 显示了定时器 2 自动加，因为 DCEN＝0。在这个模式里，有两个属性由 T2CON 寄存器的 EXEN 位选择。如果 EXEN＝0，定时器 2 加至 0FFFFH，对 TF2（溢出标志）置位，一旦溢出时。这导致定时器 2 寄存器被 RCAPL2 和 RCAP2H 的 16 值重载。RCAPL2 和 RCAP2H 的值由软件方法预设置。

如果 EXEN2＝1，那么一个 16 位重载值可由一个溢出或在输入 T2EX 上 1－0 瞬变上触发。这个瞬变也对 EXF2 位置位。定时器 2 中断，如果使能的话，能被产生当 TF2＝1 或 EXF2＝1。

在图 5，DCEN＝1，使定时器 2 使能加或减。这个模式允许 T2EX 管脚控制计数方向。当逻辑 1 应用在管脚 T2EX，定时器 2 将加。在 0FFFFH，定时器 2 将溢出，并对 TF2 标志置位，该位能产生中断，如果中断被使能。定时器溢出也导致 RCAPL2 和 RCAP2H 的 16 值被重载进 TL2 和 TH2。

当逻辑 0 应用在管脚 T2EX，定时器 2 将减。定时器将下溢，当 TL2 和 TH2 的值和 RCAPL2 和 RCAP2H 的贮存的值相等。定时器 2 下溢对 TF2 标志置位，使 0FFFFH 被再次载入定时器寄存器的 TL2 和 TH2。

外部标志 EXF2 切换当定时器 2 下溢或上溢。EXF2 位可被用作精度的第 17 位如果需要的话。EXF2 标志不产生一个中断在这种模式里。

表 4 定时器 2 操作模式

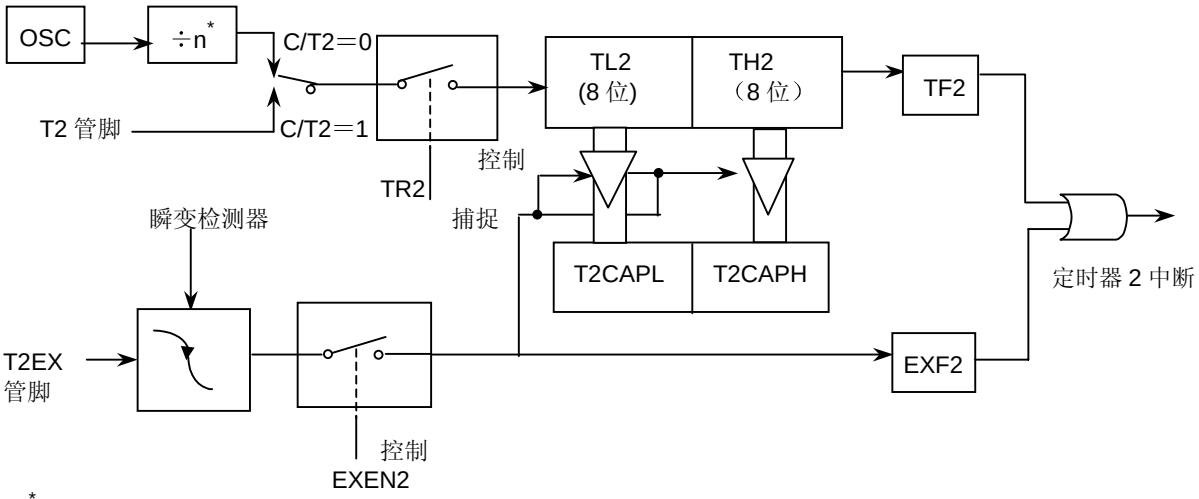
RCLK＋TCLK	CP/（/RL2）	TR2	MODE
0	0	1	16 位自动重载
0	1	1	16 位捕捉
X	X	1	波特率产生器
X	X	0	（off）

(MSB)							(LSB)
T0F2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

标记	位置	名称和意义
TF2	T2CON.7	定时器 2 溢出标志由定时器 2 溢出设置，必须由软件清除。TF2 不会被置位当 TCLK 或 RCLK＝1。
EXF2	T2CON.6	定时器 2 外部标志置位，当一个捕捉或一个重载由 T2EX 上的负瞬变产生且 EXEN2＝1。当定时器 2 中断被使能，EXF2＝1 将使 CPU 将指向定时器 2 中断子程。EXF2 必须由软件清除。EXF2 不产生一个中断在加/减计数器模式（DCEN＝1）。
RCLK	T2CON.5	接收时钟标志。当置位时，使串行口使用定时器 2 脉冲作为它的接收时钟在模式 1，3。RCLK＝0 使定时器 1 溢出作为接收时钟。
TCLK	T2CON.4	发送时钟标志，当置位时，使串行口使用定时器 2 脉冲作为它的发送时钟在模式 1，3。TCLK＝0 使定时器 1 溢出作为发送时钟。
EXEN2	T2CON.3	定时器 2 外部使能标志。当置位时，允许一个捕捉或重载发生作 T2EX 上的负瞬变的结果如果定时器 2 没有被用作时钟串行端。EXEN＝0 导致定时器 2 忽略 T2EX 上的事件。
TR2	T2CON.2	定时器 2 启始/停止控制。逻辑 1 启动这个定时器。
C/T2	T2CON.1	定时器或计数器选择（定时器 2） 0＝内部定时器（OSC/6 在 6 个时钟模式或 OSC/12 在 12 个时钟模式） 1＝外部事件计数器（下降边触发）
CP/RL2	T2CON.0	捕捉/重载标志。当置位，捕捉将发生在 T2EX 上的负瞬变如果 EXEN2＝1。当被清除时，定时器 2 溢出或负瞬变发生在 T2EX 上，自动重载将发生当 EXEN2＝1。当 RCLK＝1 或 TCLK＝1，这位被忽略，定时器被强制自动重载，当定时器 2 溢出时。

SU01251

图 1 定时器/计数器 2（T2CON）控制寄存器



n* =6 在 6 个时钟模式里，或 12 在 12 个时钟模式里。

SU01252

图 2 定时器 2 在捕捉模式

T2MOD　地址＝0C9H

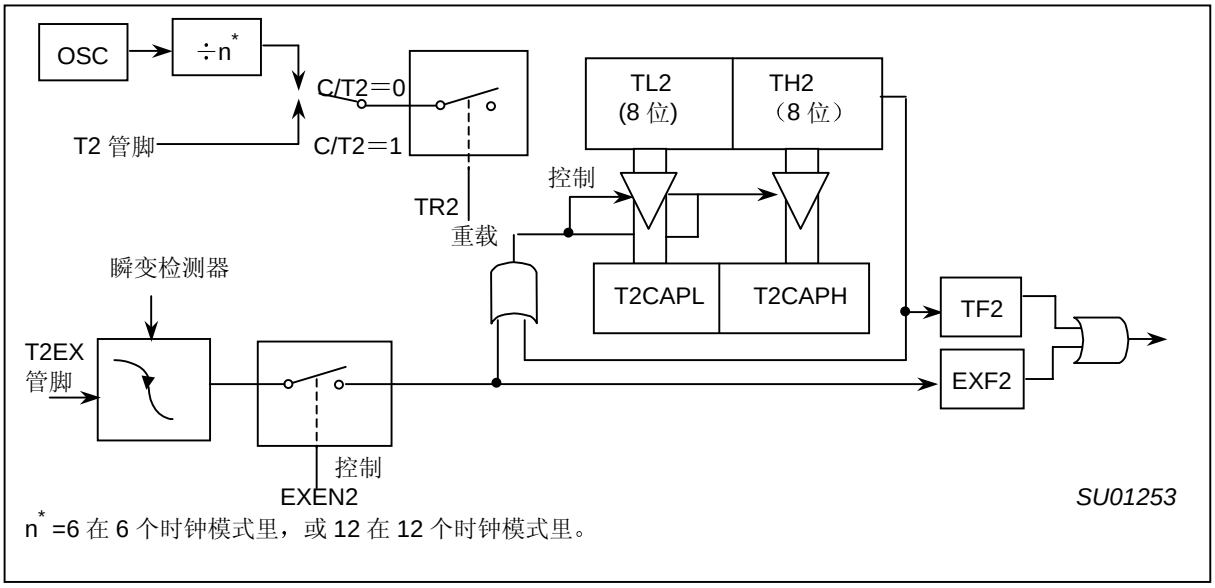
不可位寻址

—	—	—	—	—	—	T2OE	DCEN
位	7	6	5	4	3	2	1	0

标号	功能
—	没有作用，留作以后使用
T2OE	定时器 2 输出使能位
DCEN	减计数使能位。当被置位时，允许定时器 2 被构成为加/减计数器。

*用户软件不应对保留位写，这些位在 80C51 系列产品用作新特征。在这种情况下，新位的复位或非活跃值将为零，活跃值将为 1。从保留位读出的值是不确定的。

图 3 定时器 2 模式（T2MOD）控制寄存器



n* =6 在 6 个时钟模式里，或 12 在 12 个时钟模式里。

图 4 定时器 2 在自动重载模式（DCEN＝0）

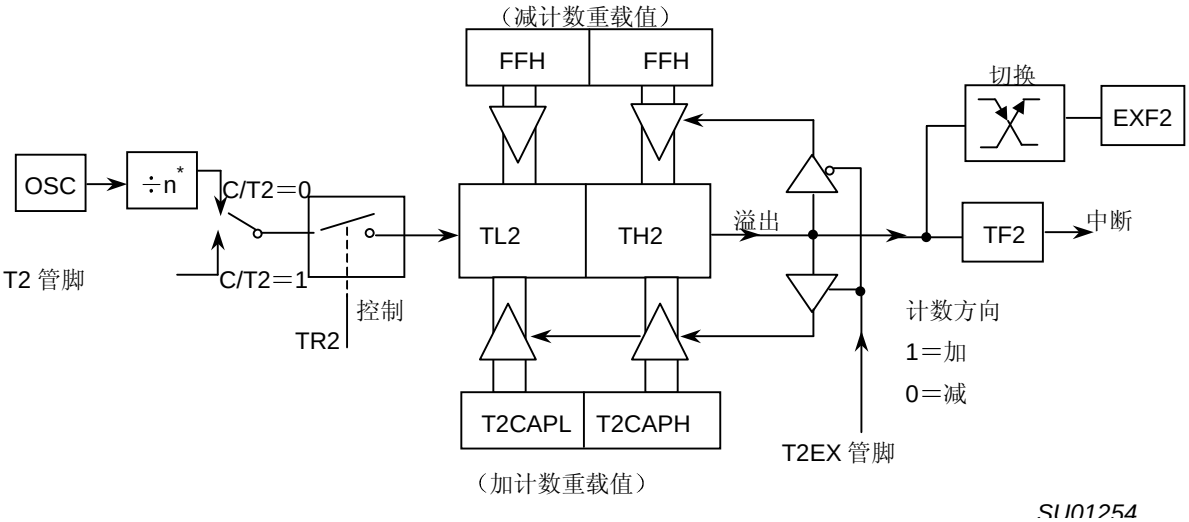


图 5 定时器 2 自动重载模式（DCEN＝1）

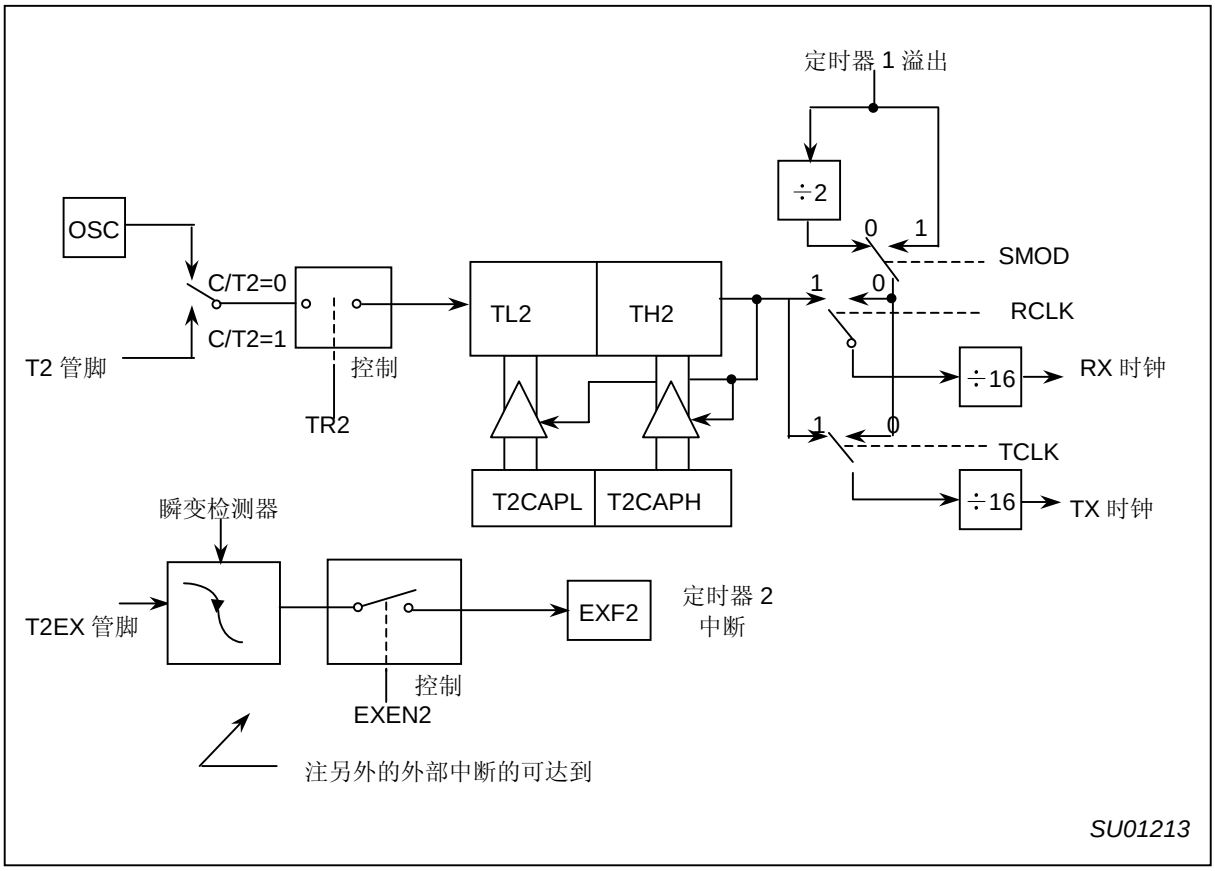


图 6 定时器 2 在波特率产生器模式

表 5 定时器 2 产生的普通的波特率

波特率		振荡器频率	定时器 2	
12 时钟模式	6 时钟模式		RCAP2H	RCAP2L
375K	750K	12MHz	FF	FF
9.6K	19.2K	12MHz	FF	D9
2.8K	5.6K	12MHz	FF	B2
2.4K	4.8K	12MHz	FF	64
1.2K	2.4K	12MHz	FE	C8
300	600	12MHz	FB	1E
110	220	12MHz	F2	AF
300	600	6MHz	FD	8F
110	220	6MHz	F9	57

波特率产生器模式

T2CON 里的 TCLK and/or RCLK 允许串行端发送和接收波特率被定时器 1 或定时器 2 驱动。当 TCLK＝0，定时器 1 被用作串行端发送波特率产生器。当 TCLK＝1，定时器 2 被用作串行端发送波特率产生器。RCLK 对于串行端接收波特率有同样的效果。有这两位，串行端有不同接收和发送波特率－一个定时器 1 产生，另一个由定时器 2 产生。

图 6 显示了定时器 2 在波特率产生模式。波特率产生模式和自动重载模式一样，；因为 TH2 里的 rollover 使定时器 2 寄存器被 RCAP2H 和 RCAP2L 的 16 位值重载，这些值由软件预先设置。

模式 1 和 3 的波特率由定时器 2 的溢出率决定：

$$\text{模式 1 和 3 波特率} = \frac{\text{定时器 2 溢出率}}{}$$

定时器可被构成为既作为定时器使用或计数器使用。在许多应用中，它被构成为定时器（C/T2*=0）。定时器操作是不同于定时器 2 当它被用作波特率产生器。

通常，作为定时器，它将自加 1 每个机器周期（也就是说，1/6 振荡器频率在 6 个时钟模式，1/12 振荡器频率在 12 个时钟模式）。作为波特率产生器，它在 6 个时钟模式里以振荡器频率增加（OSC/2 在 12 个时钟模式）。因此波特率公式如下：

$$\text{模式 1 和 3 波特率} = \frac{\text{振荡器频率}}{[\,n^{*} \times [65536-(\text{RCAP2H,RCAP2L})]]}$$

*n=16 在 6 个时钟模式
　　32 在 12 个时钟模式

(RCAP2H,RCAP2L)＝RCAP2H,RCAP2L 的内容被作为未带符号的整数。

定时器 2 作为波特率产生器模式见图 6 所示，仅当在 T2CON 寄存器里 RCLK and/or TCLK＝1 时，才是合法的。注意在 TH2 里的 rollover 对 TF2 不置位，也不产生一个中断。因此，定时器 2 不必无效在其处于波特率产生器模式。如果 EXEN2（T2 外部使能标志）被置位，T2EX 上的 1－0 瞬变（定时器/计数器 2 触发器输入）将对 EXF2（T2 外部标志）置位但不导致从 RCAP2H，RCAP2L 到 TH2，TL2 的重载。因此，当定时器 2 作为波特率产生器使用，T2EX 能作为额外的外部中断使用，如果需要的话。

当定时器 2 处于波特率产生器模式，不应对 TH2、TL2 尝试读或写。作为波特率产生器，定时器 2 被加 1 每 state time（osc/2）或异步于管脚 T2；在这些条件下，TH2 或 TL2 的读或写也许不会准确。RCAP2 寄存器可被读，但不应被写入，因为一个写可覆盖一个重载和导致写 and/or 重载错误。定时器应被关掉（清除 TR2）在访问定时器 2 或 RCAP2 寄存器之前。

波特率方程的总结

定时器 2 处于波特率产生模式。如果定时器 2 是通过管脚 T2（P1.0）计时，波特率是：

$$\text{波特率} = \frac{\text{定时器 2 溢出率}}{16}$$

如果定时器 2 是内部计时，波特率是：

$$\text{波特率} = \frac{f_{osc}}{[\,n^{*} \times [65536-(\text{RCAP2H,RCAP2L})]]}$$

*n=16 在 6 个时钟模式
　　32 在 12 个时钟模式

f_{osc}＝振荡器频率

为了获取重载值，上面的方程应被再次重写

$$\text{RCAP2H,RCAP2L} = 65536 - \left(\frac{f_{osc}}{n^{*} \times \text{波特率}}\right)$$

定时器/计数器 2 建立

除了波特率产生器模式，T2CON 给的值不包括 TR2 位的设置。因此，位 TR2 必须被单独设定以打开定时器。见表 6 定时器 2 作为定时器的设定。也见表 7 作为计数器的设定。

表 6 作为定时器的定时器 2

模式	T2CON	
	内部控制（注 1）	外部控制（注 2）
16 位重载值	00H	08H
16 位捕捉	01H	09H
波特率产生器接收和发送同样波特率	34H	36H
仅接收	24H	26H
仅发送	14H	16H

表 7 作为计数器的定时器

模式	TMOD	
	内部控制（注 1）	外部控制（注 2）
16 位	02H	0AH
自动重载	03H	0BH

注：

1．捕捉/重载仅发生在定时器/计数器溢出时。

2．捕捉/重载发生在定时器/计数器溢出时和 T2EX 上（P1.1）1—0 瞬变除了当定时器 2 波特率产生器模式。

加强型 UART

UART 在所有通常模式里操作在以 80C51 8 位为基础的微型控制器数据手册 IC20 的第一部份有描述。另外 UART 能完成桢错误检测通过寻找丢失位，以及能完成自动地址识别。UART 也完全支持多机通信和标准 80C51UART 一样。

当用于桢错误检测，UART 在通迅中搜寻丢失的停止位。一个丢失位将对 S0CON 里的 FE 位置位。这个 FE 位和 SM0 共占用 S0CON.7 位，S0CON.7 的功能由 PCON.6（SMOD0）（见图 7）决定。如果 SMOD0 被置位，则 S0CON.7 的功能作为 FE。S0CON.7 的功能作为 SM0 当 SMOD0 被清除。当用作 FE ，S0CON.7 仅能由软件清除，有关见图 8。

自动地址识别

自动地址识别是个允许 UART 识别在串行位流中某个地址通过使用硬件作比较。这个特征节约了大量的软件覆盖，通过取消软件检查每个经过串行端的串行流。这个特征通过对 S0CON 里的 SM2 位置位，能够使能。在 9 位 UART 模式，模式 2，模式 3，接收中断标志（RI）将被自动置位当接收的字节包含指定的地址或广播地址。9 位模式要求第九个信息位是个 1 以表明接收的信息是个地址而不是数据。自动地址识别见图 9。

8 位模式被称为模式 1。在这个模式里，RI 标志将被置位如果 SM2 被使能并且信息接收位有个紧跟 8 位后的有效的停止位，且信息是指定的地址或广播地址。

模式 0 是个循环寄存器模式，SM2 被忽略。

使用自动地址识别特征允许一个主机有选择性的和一个或更多的从机通迅通过调入指定的从机地址或地址。所有的从机可被连接通过使用广播地址。两个特殊功能寄存器用于确定从机的地址，SADDR，和地址屏蔽，SADEN。SADEN 确定 SADDR 里的那些位被使用，哪些位不用。SADEN 可和 SADDR 逻辑与，以产生一个主机将定位的从机地址。指定地址的使用允许多个从机被使用而排除其他。下面这个例子将有助于说明这个方案的通用性。

从机 0	SADDR＝	1100 0000
	SADEN＝	1111 1101
	指定的 ＝	1100 00X0
从机 1	SADDR＝	1100 0000
	SADEN＝	1111 1110
	指定的 ＝	1100 000X

在上面的例子中，SADDR 和 SADEN 数据一样是用于区别两个从机。从机 0 要求一个 0 在位 0 并忽略位 1。从机 1 要求一个 0 在位 1 并忽略位 0。一个唯一地址对于从机 0 是 1100 0010。因为从机 1 要求在位 1 是个 0。一个唯一地址对于从机 1 是 1100 0001，因为在位 0 是个 1 将排除从机 0。两个从机在同时可被选择，地址在位 0 是 0（对于从机 0），在位 1 是 0（对于从机 1），因此两个可用 1100 0000 选址。

在一个更复杂的系统中，下面用于选择从机 1，从机 2，而排除从机 0：

从机 0	SADDR＝	1100 0000
	SADEN＝	1111 1001
	指定的 ＝	1100 0XX0
从机 1	SADDR＝	1110 0000
	SADEN＝	1111 1010
	指定的 ＝	1110 0X0X
从机 2	SADDR＝	1110 0000
	SADEN＝	1111 1100
	指定的 ＝	1110 00XX

在上面的例子中，区别三个从机是低三位。从机 0 要求位 0 是个 0，它能被单独地定位为 1110 0110。从机 1 要求位 1＝0，它能被单独地定位为 1110 和 0101。从机 2 要求位 2 为 0，唯一地址是 1110 0011。选择从机 0 和 1 而排除从机 2 用地址 1110 0100，因为使位 2＝1 是必要的去排除从机 2。

每个从机的广播地址由 SADDR 和 SADEN 逻辑或得到。在结果里 0 被看作没有用。在许多情况下，这个广播地址为 FF 十六进制。

一旦复位，SADDR 和 SADEN 被载入 0。这个产生一个无用的指定地址以及无用的广播地址。这将使自动地址识别模式无效，允许微型控制器使用标准的 UART 驱动器，不用这个特征。

S0CON　地址＝98H
可位寻址

位	SM0/FE	SM1	SM0	REN	TB8	RB8	TI	RI
	7	6	5	4	3	2	1	0

标记	功能
FE	桢错误检测位。这个位被接收器设置当一个非合法停止位被检测到。FE 位不被合法桢清除但应被软件清除。SMOD0 位必须被置位，使对 FE 位访问有效。
SM0	串行端模式位 0，（SMOD0 必须＝0 去访问位 SM0）
SM1	串行端模式位 1

SM0	SM1	模式	描述	波特率**
0	0	0	循环寄存器	f_{osc}/6（6 时钟模式）或 f_{osc}/12（12 时钟模式）
0	1	1	8 位 UART	可变
1	0	2	9 位 UART	f_{osc}/32 或 f_{osc}/16（6 时钟模式）或 f_{osc}/64（12 时钟模式）或 f_{osc}/32（12 时钟模式）
1	1	3	9 位 UART	可变

SM2	使自动地址识别特征有效在模式 2 或 3。如果 SM2＝1，RI 将不会被置位除非接收的第九个数据位（RB8）是 1，表明是个地址，并且接收的字节是指定的地址或广播地址。在模式 1，如果 SM2＝1，RI 将不会被激活除非一个合法的停止位被接收，接收的字节是指定的地址或广播地址。在模式 0，SM0 位应为零。
REN	串行接收使能。由软件置位使接收使能。由软件清除使接收无效。
TB8	在模式 2 和 3，第九个数据位被发送。由软件设置或清除。
RB8	在模式 2 和 3，被接收的第九个数据位。在模式 1，如果 SM2＝0，RB8 是被接收的停止位。在模式 0，RB8 没有作用。
TI	发送中断标志。在模式 0 里，第 8 位时序的末端由硬件置位或在其他模式里，在停止位的开始置位，在任何串行发送。必须由软件清除。
RI	接收中断标志。在模式 0 里，第 8 位时序的末端由硬件置位，或中途通过停止位在其他模式里，

注
*SMOD0 位于 PCON.6。
**f_{osc}=振荡器频率

SU01451

图 7 S0SCON：串行端控制寄存器

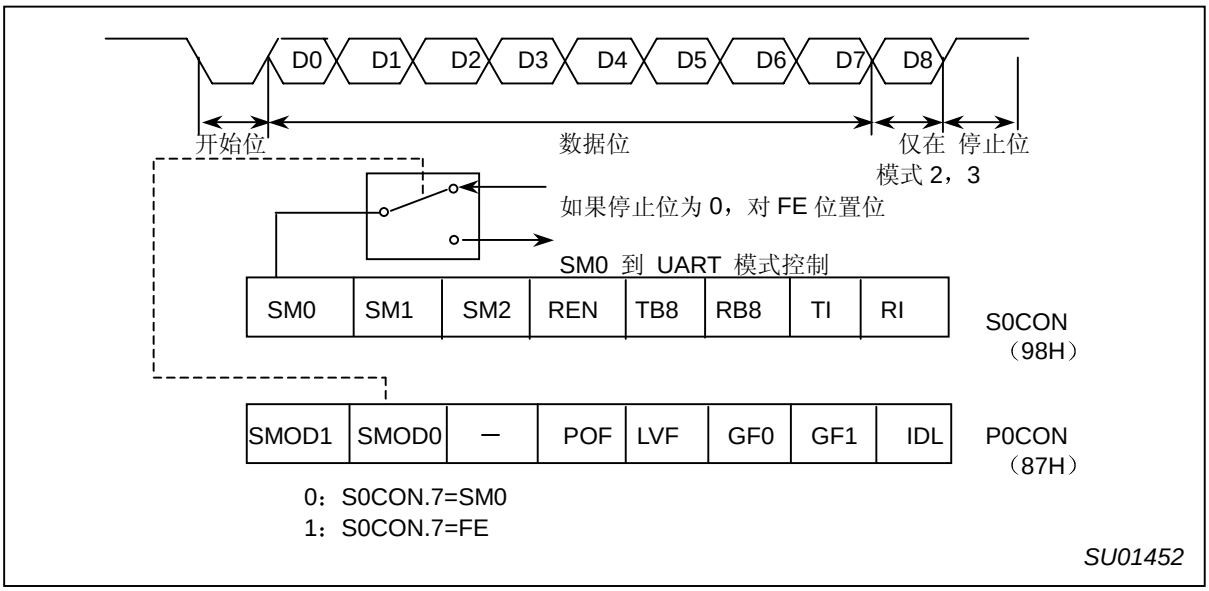


图 8 UART 桢错误检测

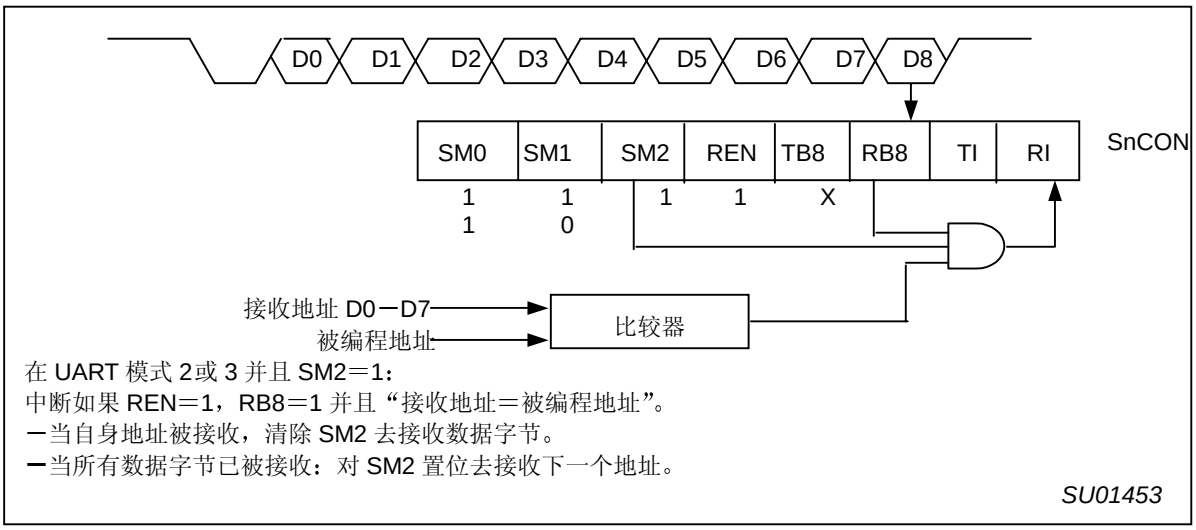


图 9 UART 多机通信，自动地址识别

中断优先级结构

P89C660/662/664 有 8 个源四级中断结构（见表 8）。

有四个 SFRs 和四级中断连接起来，是 IE，IP，IEN1，IPH。（见图 10，11，12，13）IPH（中断优先级高）使四级中断结构可能。IPH 位于 SFR 地址 B7H。IPH 寄存器的结构和它的位描述见图 12。

IPH SFR 的功能当和 IP SFR 连接时，决定每个中断的优先级。每个中断的优先级被决定如下面的表所示：

优先级位		中断优先级
IPH.x	IP.x	
0	0	级别 0（最低优先级）
0	1	级别 1
1	0	级别 2
1	1	级别 3（最高优先级）

表 8 中断表

源	竞选级	请求位	硬件清除	矢量地址
X0	1	IE0	N(L)[1] Y(T)[2]	03H
SI01(I²C)	2	—	N	2BH
T0	3	TP0	Y	0BH
X1	4	IE1	N(L) Y(T)	13H
T1	5	TF1	Y	1BH
SP	6	RI，TI	N	23H
T2	7	TF2，EXF2	N	3BH
PCA	8	CF，CCFn n=0-4	N	33H

注：

1．L＝级别激活

2．T＝瞬变激活

	7	6	5	4	3	2	1	0
IEN0（0A8H）	EA	EC	ES1	ES0	ET1	EX1	ET0	EX0

使能位＝1 中断有效
使能位＝0 无效位

位	标记	功能
IEN0.7	EA	总无效位。如果 EA＝0，所有中断无效。如果 EA＝1，每个中断可单独有效或无效通过对它的使能位置位或清除。
IEN0.6	EC	PCA 中断使能位。
IEN0.5	ES1	I^2C 中断使能位
IEN0.4	ES0	串行端中断使能位
IEN0.3	ET1	定时器 1 中断使能位
IEN0.2	EX1	外部中断 1 使能位
IEN0.1	ET0	定时器 0 中断使能位
IEN0.0	EX0	外部中断 0 使能位

SU01454

图 10 IE 寄存器

	7	6	5	4	3	2	1	0
IP（0B8H）	PT2	PPC	PS1	PS0	PT1	PX1	PT0	EX0

优先级位＝1 分配高优先级
优先级位＝0 分配低优先级

位	标记	功能
IP.7	PT2	定时器 2 中断优先级位
IP.6	PPC	PCA 中断优先级位
IP.5	PS1	串行 I/O（I^2C）中断优先级位
IP4	PS0	串行端中断优先级位
IP.3	PT1	定时器 1 中断优先级位
IP.2	PX1	外部中断 1 优先级位
IP.1	PT0	定时器 0 中断优先级位
IP.0	PX0	外部中断 0 优先级位

SU01455

	7	6	5	4	3	2	1	0
IPH（B7H）	PT2H	PPCH	PS1H	PS0H	PT1H	PX1H	PT0H	PX0

优先级位＝1 分配高优先级
优先级位＝0 分配低优先级

位	标记	功能
IPH.7	PT2H	定时器 2 中断优先级位高
IPH.6	PPCH	PCA 中断优先级位高
IPH.5	PS1H	串行 I/O（I^2C）中断优先级位高
IPH.4	PS0H	串行端中断优先级位高
IPH.3	PT1H	定时器 1 中断优先级位高
IPH.2	PX1H	外部中断 1 优先级位高
IPH.1	PT0H	定时器 0 中断优先级位高
IPH.0	PX0H	外部中断 0 优先级位高

SU01456

图 12 IPH 寄存器

	7	6	5	4	3	2	1	0
IPH（B7H）	—	—	—	—	—	—	—	ET2

优先级位＝1 分配高优先级
优先级位＝0 分配低优先级

位	标记	功能
IEN1.7	—	
IEN1.6	—	
IEN1.5	—	
IEN1.4	—	
IEN1.3	—	
IEN1.2	—	
IEN1.1	—	
IEN1.0	ET2	定时器 2 中断使能位

SU01095

降低 EMI 模式

AUXR 寄存器的 AO 位（AUXR.0）当置位时，ALE 输出无效。

降低 EMI 模式

AUXR（8EH）

7	6	5	4	3	2	1	0
—	—	—	—	—	—	EXTRM	AO

AUXR.1　EXTRM
AUXR.0　AO　　关掉 ALE 输出

双 DPTR

双 DPTR 结构（见图 14）是确定外部数据存储器的位置的方法。有两个 16 位 DPTR 寄存器，可以定位外部存储器，以及一个叫做 DPS＝AUXR1/位 0 允许程序代码在这之间转换。

- 新寄存器名称：AUXR1＃
- SFR 地址：A2H
- 复位值：xxxxxx0x0B

AUXR1(A2H)

7	6	5	4	3	2	1	0
—	—	ENBOOT	—	GF2	0	—	DPS

DPS＝AUXR1/位 0＝在 DPTR0 和 DPTR1 之间切换

Select　Reg	DPS
DPTR0	0
DPTR1	1

DPS 状态位应被软件保存当在 DPTR0 和 DPTR1 之间切换。

GF2 位是通用的用户定义标志。注意位 2 是不可写的，总是读作零。这允许 DPS 位 0 被很快地触发，通过执行一个 INC AUXR1 指令，而不影响 GF2 位。

ENB00T 位决定 BOOTROM 是否使能或无效。这个位被自动设置如果状态字节没有零在复位期间或/PSEN 被拉为低，ALE 悬空为高，EA>V_{IH} 在复位下降边的电压。否则这个位被清除在复位期间。

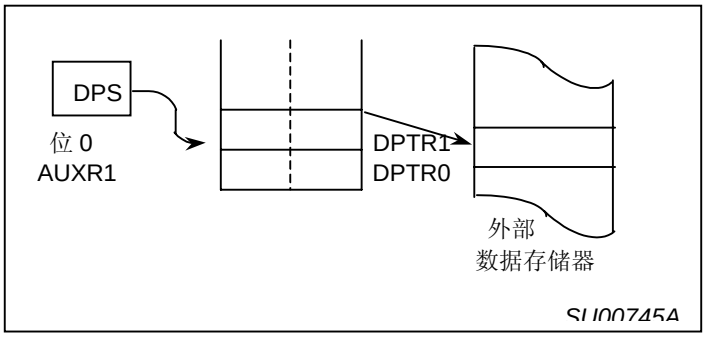


图 14

DPTR 指令

有关数据指针的 DPTR 指令由 AUXR1/位 0 选择。使用 DPTR 六条指令如下：

INC DPTR	数据指针加 1
MOV DPTR，＃data16	用 16 位常数载入 DPTR
MOV A，　@A+DPTR	有关 DPTR 的代码字节送到 ACC
MOVX A ,@DPTR	将外部 RAM 送到 ACC
MOVX @DPTR，A	将 ACC 送到外部 RAM（16 位地址）
JMP @A+DPTR	跳转到和 DPTR 间接有关的地址

数据指针可被访问在一个字节一个字节基础上，通过对 SFR 访问的指令的低和高字节说明。见 AN458 应用可获取更详细的信息。

可编程计数器阵列（PCA）

在 89C66x 上的可编程计数器阵列是个特殊的 16 位定时器，有五个 16 位捕捉/比较模块和定时器连接在一起。每个模块可被编程在下列四种模下操作：上升 and/or 下降边捕捉，软件定时器，高速输出或脉冲调制模块。每个模块在端口 1 有个管脚和它连接。模块零被接至 P1.3（CEX0）。模块 1 接至 P1.4（CEX1），等等。PCA 结构见图 15 所示。

CPS1	CPS0	PCA 定时器计数器源
0	0	1/6 振荡器频率（6 个时钟模式）
		1/12 振荡器频率（12 个时钟模式）
0	1	1/2 振荡器频率（6 个时钟模式）
		1/4 振荡器频率（12 个时钟模式）
1	0	定时器 0 溢出
1	1	在 ECI 管脚上外部输出

在 CMOD SFR 里，是三个额外的位和 PCA 接在一起。它们是允许 PCA 停止在空闲模式的 CIDL 位，是在模块 4 使看门狗有效或无效的 WDTE 位，以及被置位的 ECF 导致一个中断和 PCA 溢出标志 CF（在 CCON SFR 里）被置位当 PCA 定时器溢出。这些功能见图 16 所示。

看门狗定时器功能在模块 4 有效（见图 25）。

CCON SFR 包括 PCA 的运行控制位和 PCA 定时器的标志（CF）和每个模块（有关见图 19）。为了运行 PCA，CR 位（CCON. 6）必须由软件置位。PCA 被关掉通过清除这位。CF 位被置位当 PCA 计数器溢出和一个中断将产生如果 CMOD 寄存器的 ECF 位被置位，CF 位可被软件清除。CCON 寄存器的位 0 到位 4 是模块的标志位（位 0 用于模块 0，位 1 用于模块 2，等等），由硬件置位当匹配或捕捉发生时。这些标志也由软件清除。PCA 中断系统见图 17。

PCA 里的每个模块有个特殊功能寄存器和它连接，这些寄存器是：CCAPM0 用于模块 0，CCAPM1 用于模块 1，等等。（见图 20）这些寄存器包括控制每个模块要操作在何种模式的位。ECCF 位（CCAPMn.0，n=1，2，3，4 取决于模块）使 CCON 的 SFR 的 CCF 标志有效产生一个中断当一个匹配或比较发生，在相关的模块里。PWM（CCAPMn.1）使脉冲宽度调制使能。TOG 位（CCAPMn.2）当置位时导致和模块有关的 CEX 输出切换当 PCA 计数器和模块的捕捉/比较器寄存器的匹配发生时。匹配位 MAT（CCAPMn.3）当置位时导致 CCON 寄存器里的 CCFn 位置位当 PCA 计数器和模块的捕捉/比较器寄存器的匹配发生时。

下面两个位 CAPN（CCAPMn.4）和 CAPP（CCAPMn.5）决定捕捉输入将有效的那个边。CAPN 使负极边有效，CAPP 使正极边有效。如果两位都被设置，两个边都有效，一个捕捉将发生在任何一个瞬变上。在 ECOM 寄存器的（CCAPMn.6）最后一位当置位时将使比较器功能有效。图 21 显示了 CCAPMn 设置用于不同的 PCA 功能。

有两个另外的和每个 PCA 模块有关的寄存器，是 CCAPnH 和 CCAPnL，用于存储 16 位计数的存储器，当一个捕捉或一个比较应该发生。当一个模块用于 PWM 模式时，这些寄存器用于控制输出的循环周期。

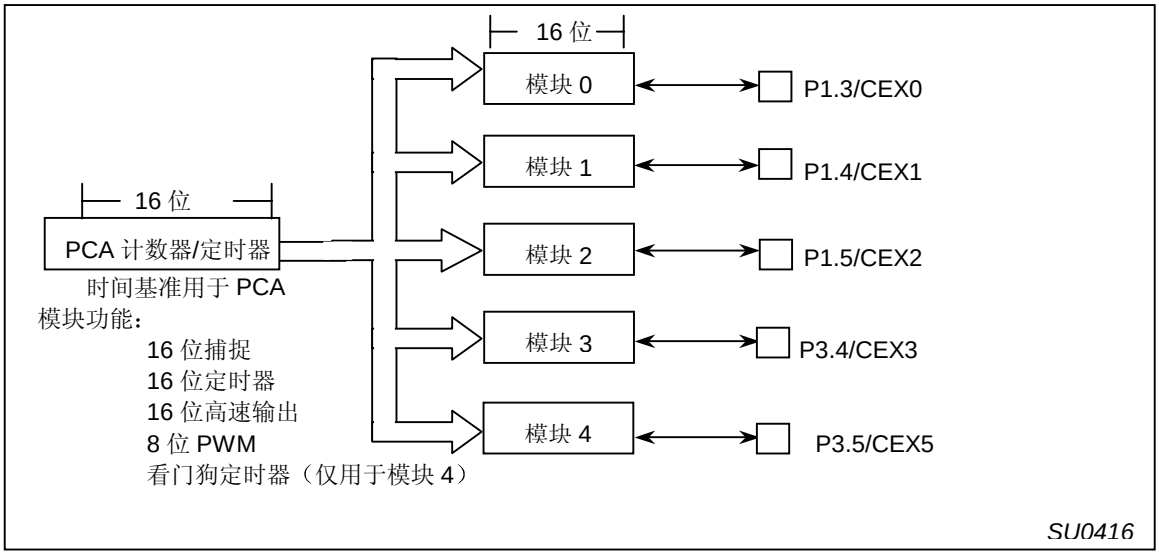


图 16 可编程计数器（PCA）

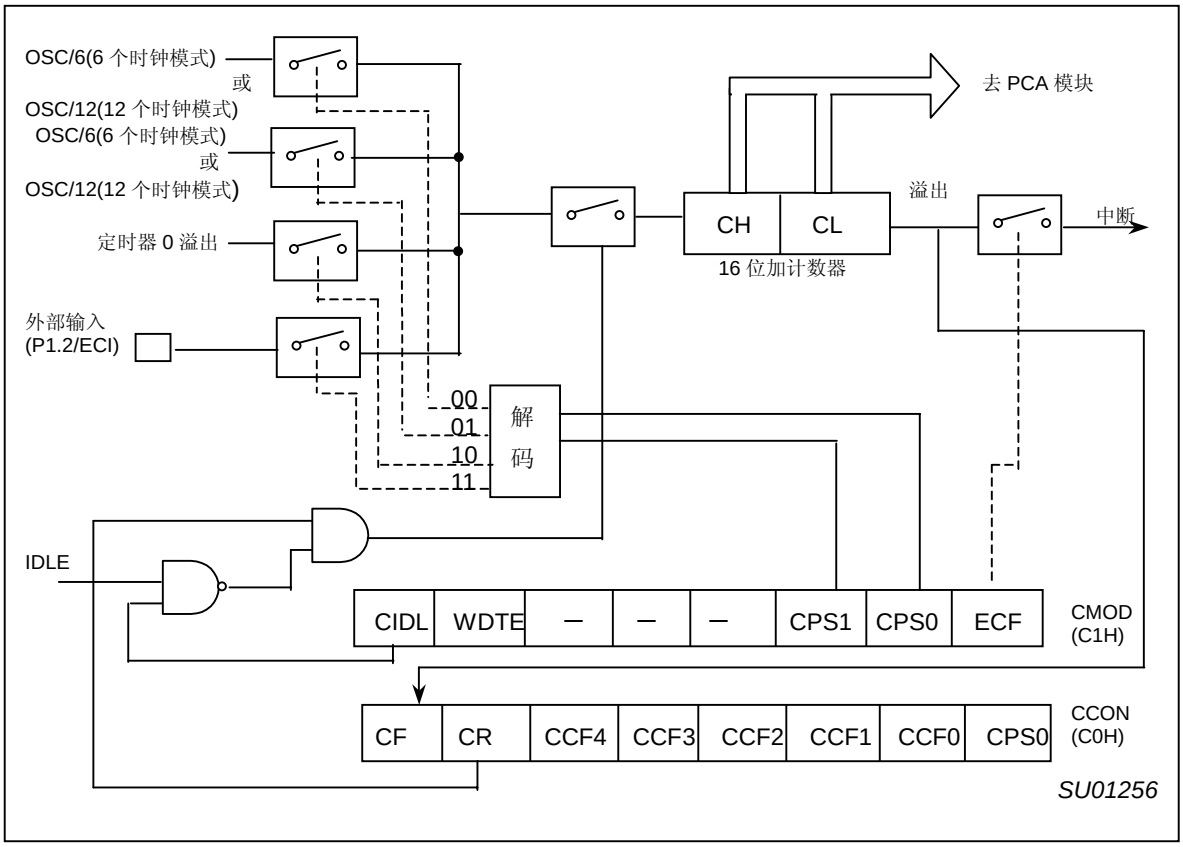


图 16 PCA 定时器/计数器

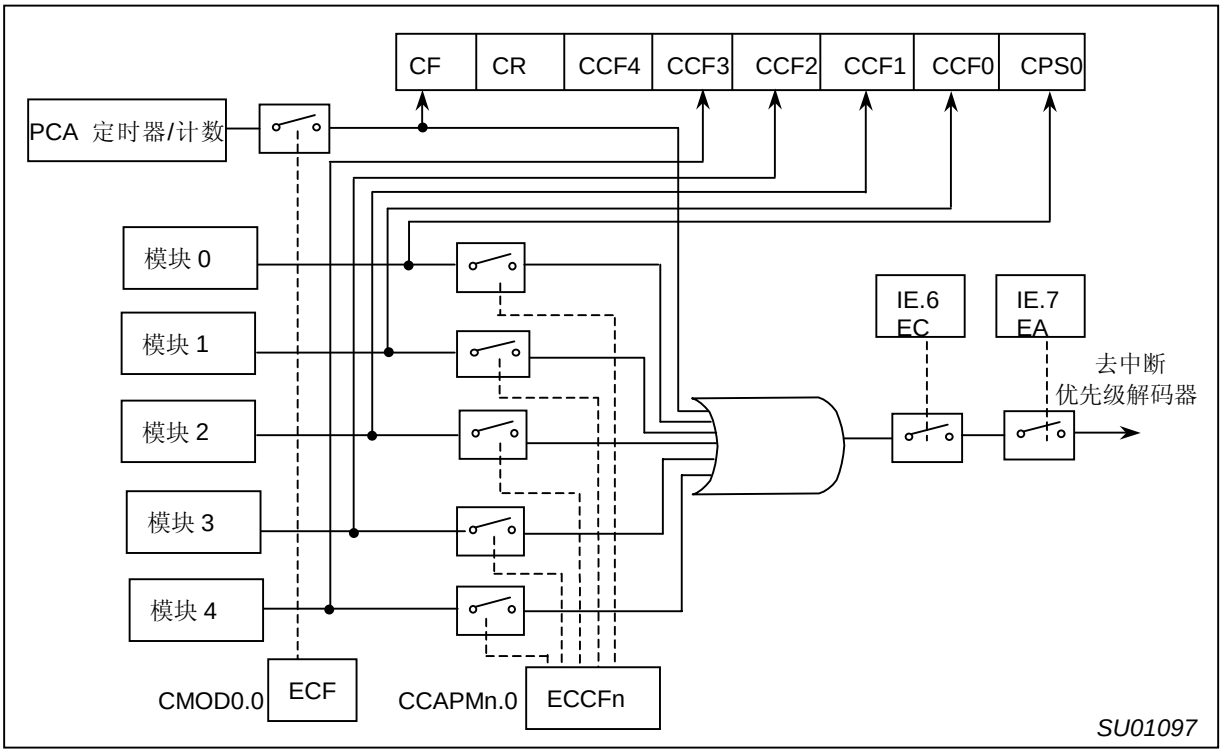


图 17 PCA 中断系统

CMOD 地址＝C1H　　复位值＝00XX X000B

CIDL	WDTE	—	—	—	CPS1	CPS0	ECF
位： 7	6	5	4	3	2	1	0

标记	功能
CIDL	计数器空闲控制：CIDL＝0 对 PCA 计数器编程以继续它的功能，在空闲模式期间。CIDL＝1 对它编程以关掉空闲模式。
WDTE	看门狗定时器使能：WDTE＝0 使看门狗定时器功能在 PCA 模块 4 上失效。WDTE＝1，使它有效。
—	没有用，用作以后使用。
CPS1	PCA 计数脉冲选择位 1。
CPS0	PCA 计数脉冲选择位 0。

CPS1	**CPS0**	**选择的 PCA 输入****	
0	0	0	内部时钟，f_{osc}/6 在 6 时钟模式（f_{osc}/12 在 12 时钟模式）
0	1	1	内部时钟，f_{osc}/2 在 6 时钟模式（f_{osc}/4 在 12 时钟模式）
1	0	2	定时器 0 溢出
1	1	3	在 ECI/P1.2 管脚上外部时钟（最大频率＝f_{osc}4 在 6 时钟模式，f_{osc}/8 在 6 时钟模式）

ECF	PCA 使计数器溢出中断使能：ECF＝1 使 CCON 里的 CF 位有效去产生一个中断。ECF＝0，使 CF 位无效。

注：
*用户软件不应对保留位写入。这些位用于 80C51 以后增加的新特征。在这种情况下，新位的复位值或非激活值为零，激活值为 1。从保留位读出的值是不确定的。
** f_{osc}=振荡器频率

SU01257

图 18 CMOD：PCA 计数器模式寄存器

CCON 地址＝C1H　　复位值＝00X0 0000B

CF	CR	CCF4	CCF3	CCF2	CCF1	CCF0	CPS0
7	6	5	4	3	2	1	0

标记	功能
CF	PCA 计数器溢出标志。由硬件置位当计数器 rolls over。CF 标志一个中断如果位 ECF 在 CMOD 被置位。CF 可被硬件置位或软件，但必须由软件清除。
CR	PCA 计数运行控制位。由软件置位去启动 PCA 计数器。必须由软件清除去关掉 PCA 计数器。
—	没有用，用作以后使用。
CCF4	PCA 模块 4 中断标志。由硬件置位当一个匹配或捕捉发生。必须由软件清除。
CCF3	PCA 模块 3 中断标志。由硬件置位当一个匹配或捕捉发生。必须由软件清除。
CCF2	PCA 模块 2 中断标志。由硬件置位当一个匹配或捕捉发生。必须由软件清除。
CCF1	PCA 模块 1 中断标志。由硬件置位当一个匹配或捕捉发生。必须由软件清除。
CCF0	PCA 模块 0 中断标志。由硬件置位当一个匹配或捕捉发生。必须由软件清除

注：
*用户软件不应对保留位写入。这些位用于 80C51 以后增加的新特征。在这种情况下，新位的复位值或非激活值为零，激活值为 1。从保留位读出的值是不确定的。
** f_{osc}=振荡器频率

SU01099

图 19 CCON：PCA 计数器控制寄存器

CCAPMn 地址 CCAPM0　0C2H　　复位值＝X000 000B
CCAPM1　0C3H
CCAPM2　0C4H
CCAPM3　0C5H
CCAPM4　0C6H

不可位寻址：

—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
7	6	5	4	3	2	1	0

标记	功能
—	没有用，留作以后使用。
ECOMn	比较器使能。ECOMn＝1 使比较器功能使能。
CAPPn	比较器正极，CAPPn＝1 使正极边捕捉使能。
CAPNn	比较器负极，CAPNn＝1 使负极边捕捉使能。
MATn	匹配。当 MATn＝1，PCA 计数器和模块的比较/捕捉寄存器的匹配将导致 CCON 里的 CCFn 位被置位，标明一个中断。
TOGn	切换。当 TOGn＝1，PCA 计数器和模块的比较/捕捉寄存器的匹配将导致 CEXn 管脚切换。
PWMn	脉冲宽度调制模式。PWMn＝1 使 CEXn 管脚用作脉冲宽度调制输出。
ECCFn	使 CCF 中断使能。使 CCON 里的寄存器的比较/捕捉标志 CCFn 产生一个中断。

注：
*用户软件不应对保留位写入。这些位用于 80C51 以后增加的新特征。在这种情况下，新位的复位值或非激活值为零，激活值为 1。从保留位读出的值是不确定的。
** f_{osc}=振荡器频率

SU01100

图 21 CCAPMn：PCA 模块模式（CCAPMn 寄存器）

—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	模块功能
X	0	0	0	0	0	0	0	空操作
X	X	1	0	0	0	0	X	16 位捕捉由 CEXn 正极边触发
X	X	0	1	0	0	0	X	16 位捕捉由 CEXn 负极边触发
X	X	1	1	0	0	0	X	16 位捕捉由 CEXn 上瞬变
X	1	0	0	1	0	0	X	16 位软件定时器
X	1	0	0	1	1	0	X	16 位高速输出
X	1	0	0	0	0	1	0	8 位 PWM
X	1	0	0	1	X	0	X	看门狗定时器

图 21 PCA 模块模式（CCAPMn 寄存器）

PCA 捕捉模式

为了在捕捉模式里使用某个 PCA 模块，CCAPM 一位或两位 CAPN 和 CAPP 必须被置位。用于这个模块的外部 CEX 输入被采样。当一个合法的瞬变发生，PCA 硬件将 PCA 计数器的寄存器的值载入模块的捕捉寄存器（CCAPnL 和 CCAPnH）。如果在 CCON SFRR 的 CCFn 位和 CCAPMn SFR 的 ECCFn 位被置位，则一个中断将被产生。有关见图 22。

16 位软件定时器模式

PCA 模块可被用作软件定时器通过对 CCAPMn 寄存器的 ECOM 和 MAT 位置位。PCA 定时器将被和模块的捕捉寄存器相比较，当一个匹配发生，一个中断发生。如果 CCFn（CCON SFR）和 ECCFn（CCAPMn SFR）位都被置位（见图 23）。

高速输出模式

在这个模式里，和 PCA 模块接在一起的 CEX 输出（在端口 1），每个匹配发生时，将在 PCA 计数器和模块的捕捉寄存器之间切换。为了激活这个模式，在 CCAPMn SFR 里的 TOG，MAT 和 ECOM 位必须被置位（见图 24）。

脉冲宽度调制模式

所有的 PCA 模块可用作 PWM 输出。图 25 显示了 PWM 功能。输出的频率取决用于 PCA 定时器的源。所有的模块有相同的输出频率；因为它们共用 PCA 定时器。每个模块的周期循环是单独的可变的，使用模块的捕捉寄存器 CCAPLn。当 PCA CL SFR 的值低于模块的 CCAPLn SFR，输出将为低，当它等于或高于，输出将为高。当 CL 从 FF 溢出到 00，CCAPLn 被重载入 CCAPHn 值。模块的 CCAPMn 寄存器的的 PWM 和 ECOM 位必须被置位，以使 PWM 模式使能。

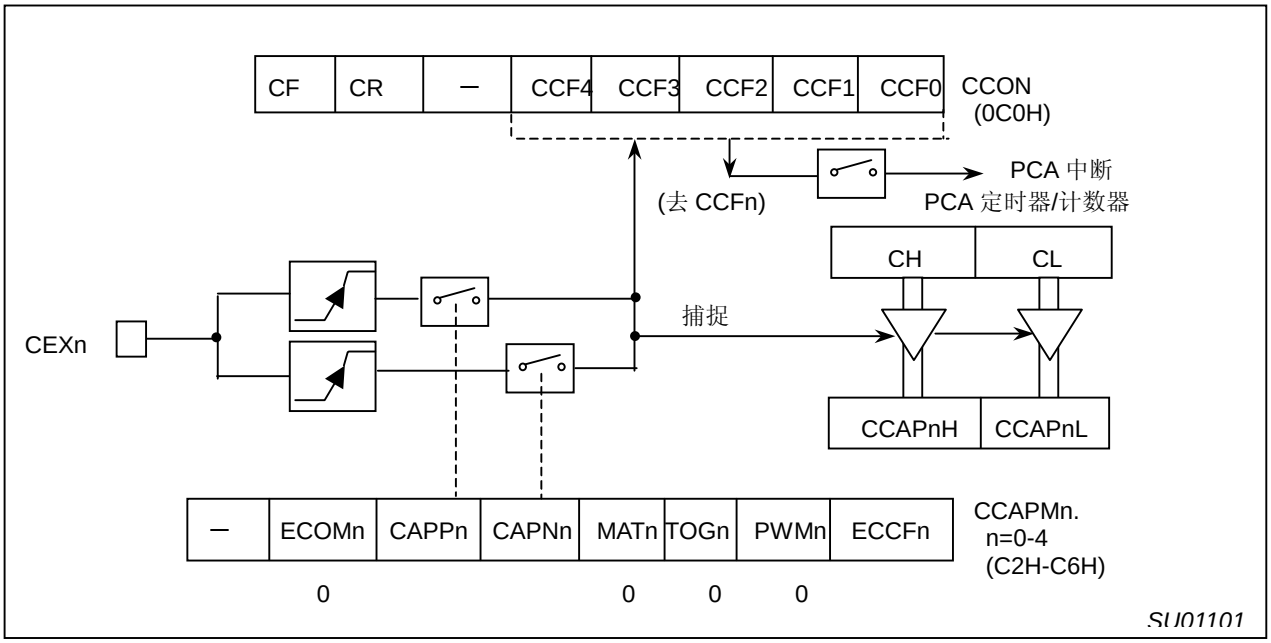


图 22 PCA 捕捉模式

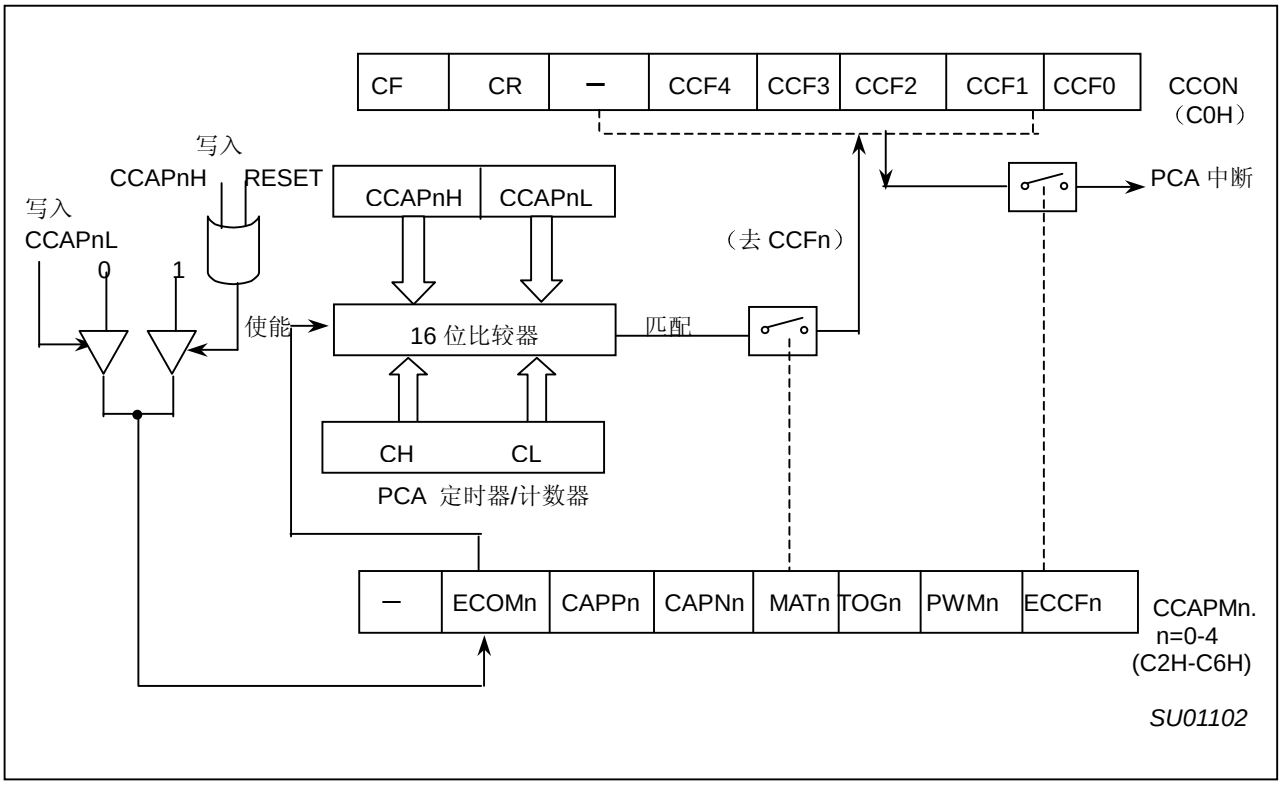


图 23 PCA 比较模式

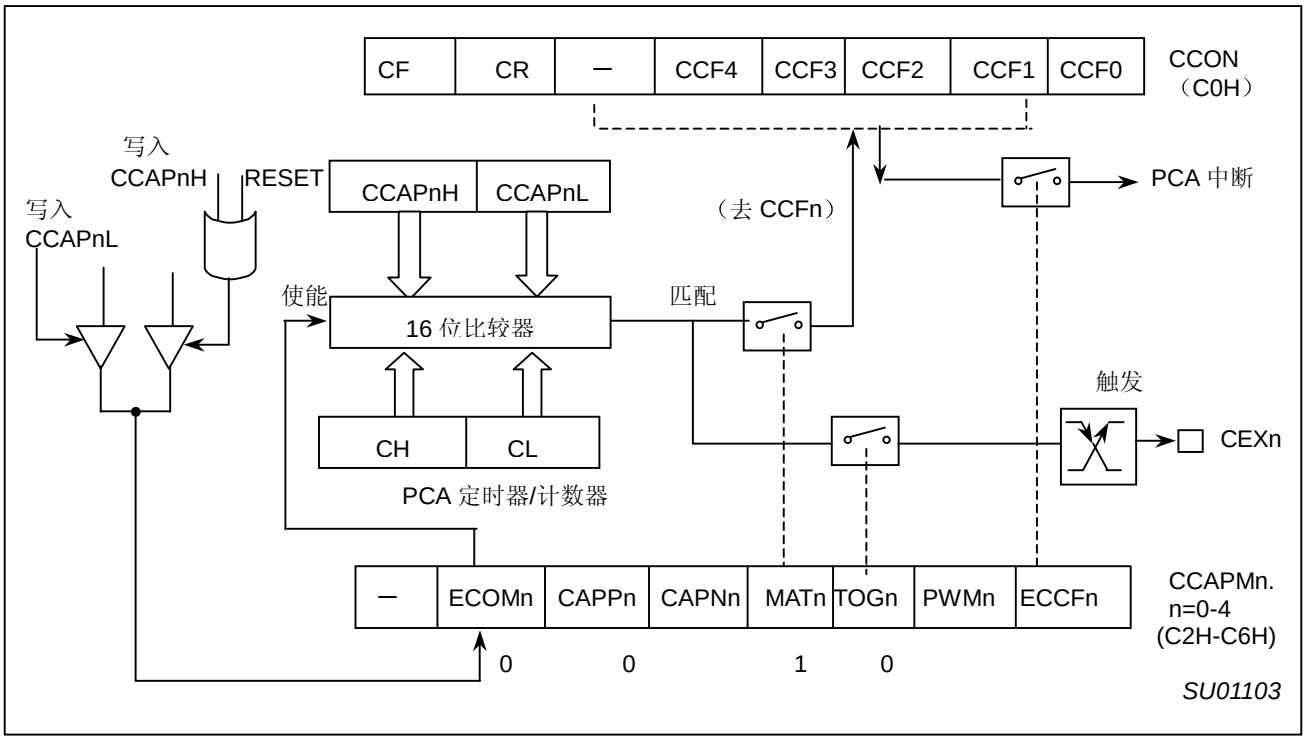


图 24 PCA 高速输出模式

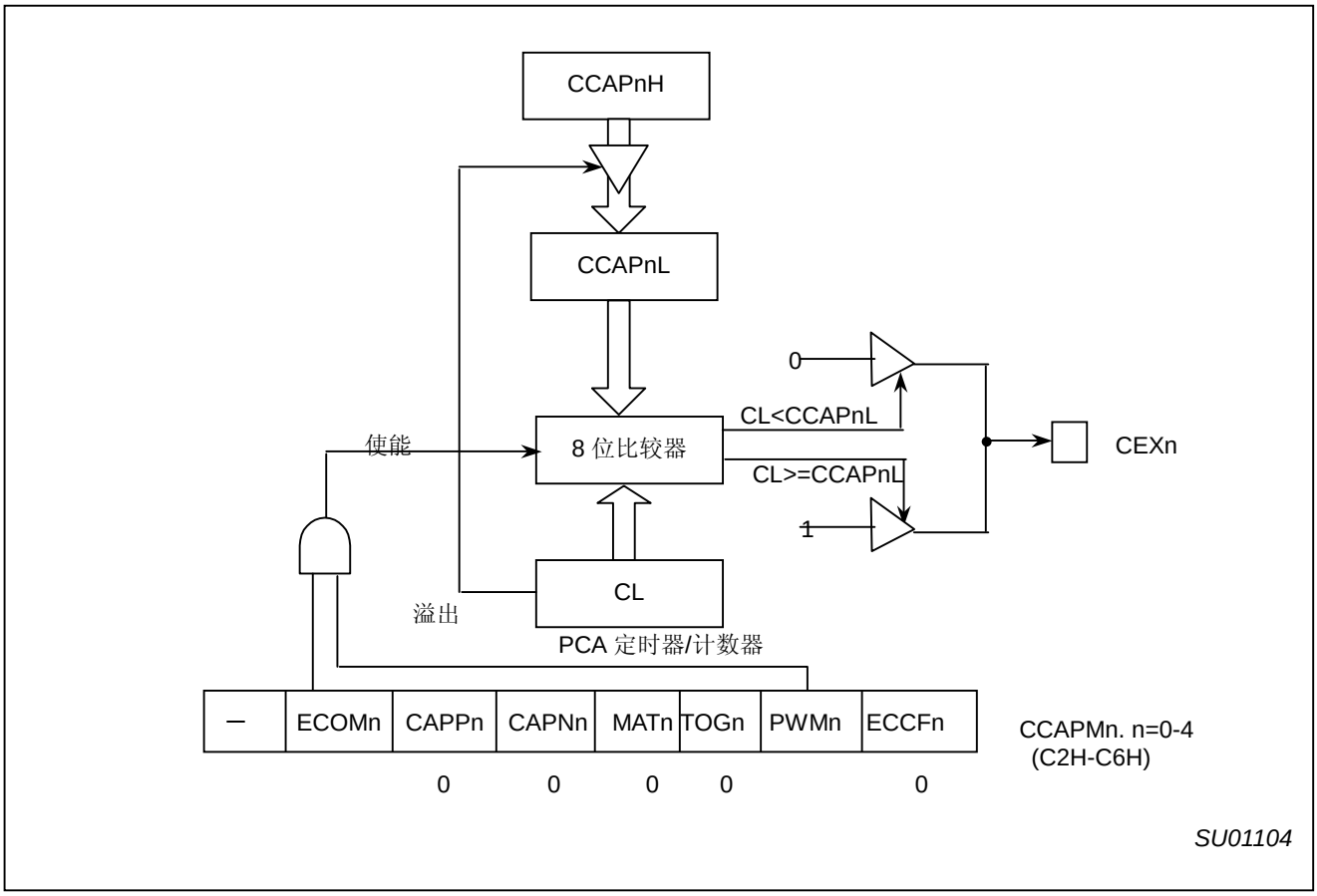


图 25 PCA PWM 模式

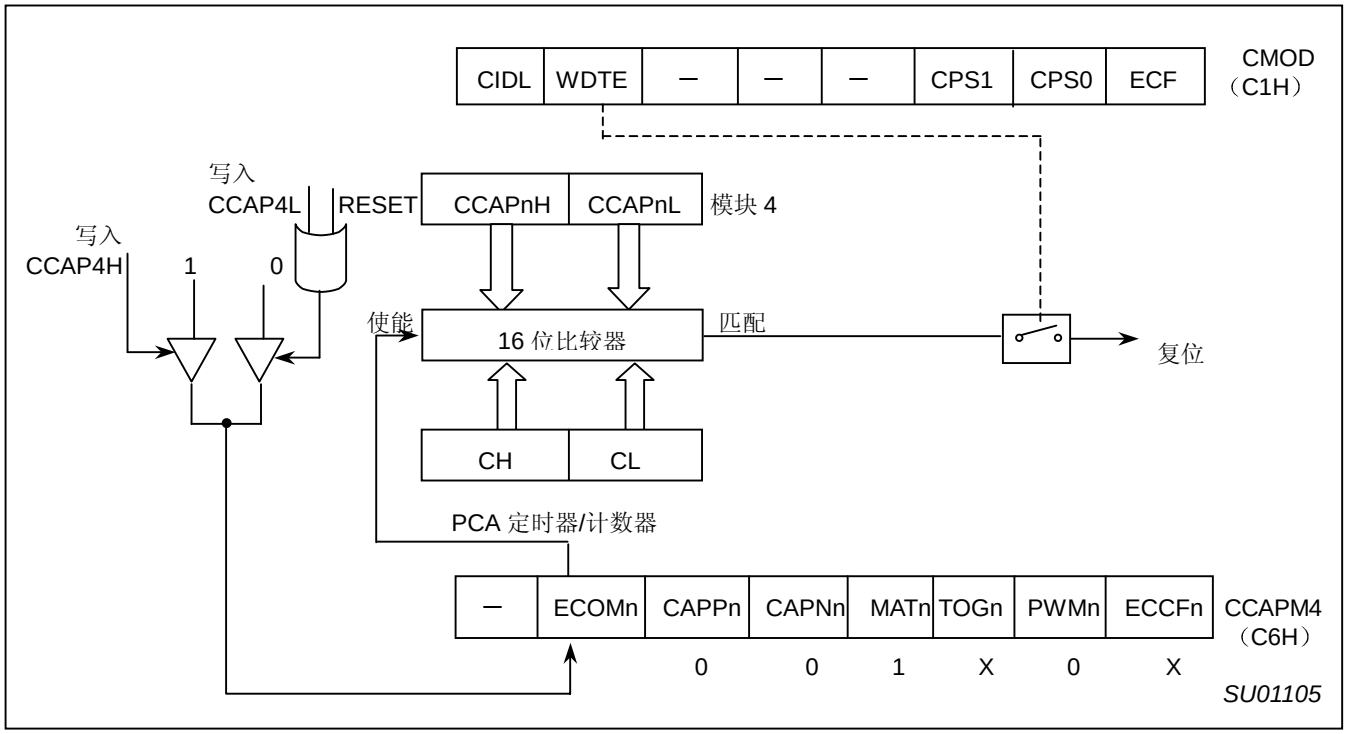


图 26 PCA 看门狗定时器 m (仅仅模式 4)

PCA 看门狗定时器

在 PCA 里有看门狗定时器用于提高系统可靠性不用增加片内数目。看门狗定时器对于易有噪声的、功率假信号、静电 discharge 的系统是有用的。模式 4 是 PCA 模块可被比编程作为一个看门狗。但是这个模式仍然能用于其他模式，如果不需要看门狗。

图 26 显示了看门狗如何工作的图。在比较寄存器里，用户预先载入 16 位值。和其他比较器模式一样，这个 16 位值被和 PCA 定时器值相比较。如果一个匹配被允许发生，一个内部复位将被产生。这不会导致 RST 管脚被驱动为高。

为了延迟复位，用户有三个选项：

1．间期地改变比较器的值，因此它不会和 PCA 定时器匹配。

2．间期地改变 PCA 定时器的值，因此它不会和比较器匹配，或

3．使看门狗无效，通过清除 WDTE 位在匹配发生前，然后使其使能。

前两个选项要更可靠点，因为看门狗定时器从来不会失效在选择 3#。如果程序计数器一旦进入 astray，一个匹配将发生，并导致内部复位。第二个选项不被建议如果其他 PCA 模块正被使用。切记，PCA 定时器是所有模块的时间基准；改变时间基准对于其他模块不是个好主意。；因此在许多应用中，第一个方法是最佳选择。

图 27 显示了初始化看门狗定时器的代码。模块 4 能被构造在比较模式里，CMOD 里的 WDTE 位必须也被置位。用户软件必须定期改变（CCAP4H，CCAPR4L）去阻止和 PCA 定时器（CH，CL）的匹配发生。这个代码在看门狗子程里被给予在图 27。

这个子程不是一个中断服务子程的一部份。因为如果程序计数器误入迷途并陷进一个无穷循环，中断将被服务以及看门狗将保持复位。因此，看门狗的目的将失效。相反，在 PCA 定时器的 2^{16} 里，从主程里调用这个子程。

```
初始化看门狗：
       MOV   CCAPM4，  #4CH      ；  模块 4 处于比较模式
       MOV   CCAP4L，  #0FFH     ；首选写入低字节
       MOV   CCAP4H，  #0FFH     ；在比较器计数高达 FFFF HEX，这些比较值必须被改变
       ORL   CMOD ，  #40H       ；对 WDTE 进行置位，使看门狗定时器使能不改变 CMOD 的其他位。


；*****************************************************************************
；
；主程 goes here，定期地调用看门狗
；
；  *****************************************************************************
；
看门狗：
       CLR     EA                        ；清除中断
       MOV     CCAP4L，#00               ；下一个比较值
       MOV     CCAP4H，CH                ；当前 PCA 的 255 数
       SETB    EA                        ；定时器值
       RET
```

图 27 PCA 看门狗定时器初始化代码

可扩展的数据 RAM 寻址

P89C660/662/664 有内部数据存储器，被分成四个段：较低的 RAM 128 字节，较高的 RAM128 字节，128 字节特殊功能寄存器（SFR），和可扩展的 256 字节 RAM（ERAM）（768 字节用于‘662；1792 用于’664）。

这四个段是：

1．较低的 RAM 128 字节（地址 00H 到 7FH）是可以直接和间接寻址。

2．较高的 RAM128 字节（地址 80H 到 FFH）仅是可以间接寻址。

3．特殊功能寄存器，SFRs，（地址 80H 到 FFH）仅是可以直接寻址。

4．256/768/1792 字节扩展的 RAM（ERAM，00H－XFFH/2FFH/6FFH）可以间接访问通过传送外部指令，MOVX，EXTRAM 位被清除，见图 28。

较低的 128 个字节可被间接或直接寻址访问。较高的 128 字节仅可被间接寻址访问。较高的 128 字节和 SFR 占用同样的地址空间，也即它们有同样的地址，但物理上和 SFR 空间分开。

当一个指令访问一个内部位置在地址 7FH 上，CPU 知道是对较高的 128 字节的 RAM 访问还是对 SFR 空间访问通过指令里用的寻址模式。使用直接寻址指令访问 SFR 空间。例如：

```
MOV   0A0H，#data
```

在位置 0A0H 访问 SFR。使用间接寻址指令访问较高的数据 RAM 的 128 字节。例如：

```
MOV   @R0，#data
```

R0 包括 0A0H，在地址 0A0H 访问数据字节，而不是 P2（地址是 0A0H）。

ERAM 可被间接寻址访问，在 EXTRAM 清除，并使用 MOVX 指令。存储器的这部份物理上位于片内，逻辑上占用外部数据存储器的头 7936 字节。

在 EXTRAM＝0，ERAM 被间接寻址，使用 MOVX 指令和选择的存储组寄存器 R0，R1 或 DPTR 联系。对 ERAM 的访问不会影响 P0 端，P3.6(WR#)和 P3.7(RD#)。P2 SFR 是输出在外部寻址期间，例如：在 EXTRAM＝0，

```
MOVX   @R0，#data
```

R0 包括 0A0H，在地址 0A0H 访问 ERAM，而不是外部存储器。对外部数据存储访问位置高于 ERAM，访问将被完成在 MOVX DPTR 指令里，和标准 80C51 同样的方法，因此，P0 和 P2 作为数据/地址总线，P3.6 和 P3.7 作为读和写的时序信号。有关见图 29。

在 EXTRAM＝1 时，MOVX @Ri 和 MOVX @DPTR 和标准 80C51 类似。MOVX @Ri 将提供一个 8 位地址和数据复用在端口 0 和任何一个输出端管脚可用作输出较高的地址位。这是提供外部翻页能力。MOVX @DPTR 将产生一个 16 位地址。端口 2 输出高八位地址（DPH 的内容），而端口 0 复用低八位带有数据的地址位（DPL）。MOVX @Ri 和 MOVX @DPTR 在 P3.6（/WR）和 P3.7（/RD）将产生读或写信号。

堆栈指针（SP）可位于内部数据存储器的 256 字节 RAM（较低和较高的 RAM）的任何位置，堆栈不可位于 ERAM。

AUXR　地址＝8EH　　复位值＝xxxx xx10B

不可位寻址

–	–	–	–	–	–	EXTRAM	AO	
位：	7	6	5	4	3	2	1	0

标记	功能
AO	无效/有效 ALE **AO　操作模式** 0　ALE 被以一个常量振荡器频率的 1/3（6 时钟模式；1/6f_{OSC} 在 12 个时钟模式）。 1　ALE 是有效仅当 MOVX @Ri/ @DPTR 时。
EXTRAM	**EXTRM　操作模式** 0　内部 ERAM 访问使用 MOVX @Ri/ @DPTR 1　外部数据存储器访问
–	没有用，留作以后使用*.

注：
*用户软件不应对保留位写入。这些位用于 80C51 以后增加的新特征。在这种情况下，新位的复位值或非激活值为零，激活值为 1。从保留位读出的值是不确定的。

SU01417

图 28 AUXR：附属寄存器

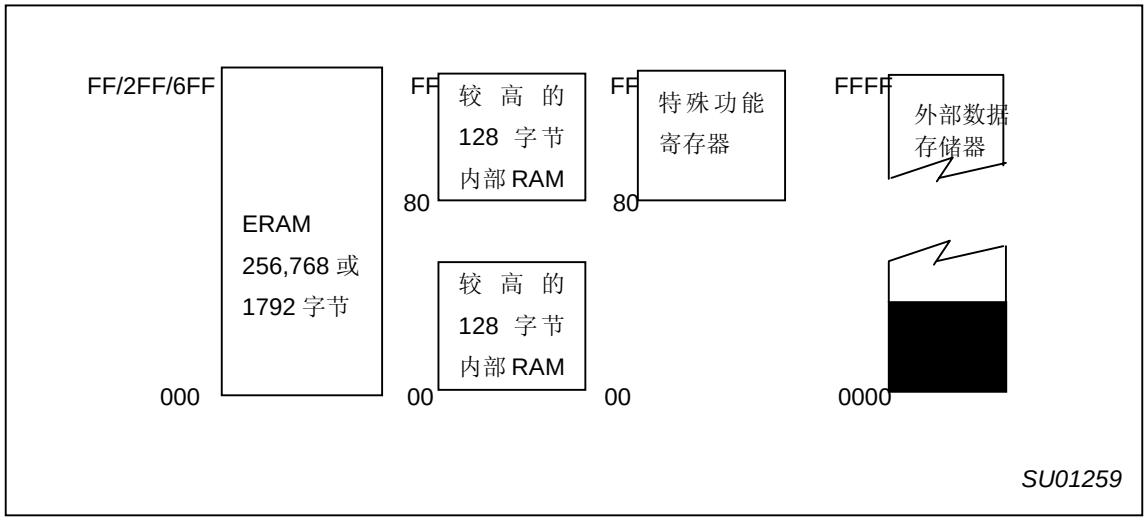


图 29 内部和外部数据存储器地址空间在 EXTRM＝0 时

硬件看门狗定时器（一次使能带有复位输出用于 P89C660/662/664）

WDT 被用作一种恢复方法在 CPU 也许会受到软件失常的情况下。WDT 由一个 16 位计数器和看门狗定时器复位（WDTRST）SFR 组成。WDT 在复位时是无效的。为了使 WDT 使能，用户必须对 WDTRST，SFR 位置 0A6H 顺序写入 01EH，和 0E1H。当 WDT 被使能，每个机器周期将它将加 1，同时振荡器正在运行，没有别的办法使 WDT 失效除非复位（要不硬件复位或 WDT 溢出复位）。当 WDT 溢出，它将驱动一个输出复位高脉冲在 RST 管脚（见下面的 note）。

使用 WDT

为了使 WDT 使能，用户必须对 WDTRST，SFR 位置 0A6H 顺序写入 01EH，和 0E1H。当 WDT 被使能，用户需要运转它通过对 WDTRST 写入 01EH、0E1H 以避免 WDT 溢出。14 位计数器溢出当它达到 16383（3FFFH），然后将复位这个器件。当 WDT 被使能，每个机器周期将它将加 1，同时振荡器正在运行。这意味着用户必须对 WDT 复位至少每个 16383 机器周期。为了复位 WDT，用户必须对 WDTRST 写入 01EH，和 0E1H。WDTRST 仅由寄存器写入。WDT 计数器不能被读或写。当 WDT 溢出，它将产生一个输出 RESET 脉冲在复位管脚（见下面的 note）。RESET 脉冲持续的时间是 $98\times T_{OSC}$（6 个时钟模式；196 在 12 个时钟模式），$T_{OSC}=1/f_{OSC}$。为了充分利用 WDT，它在下面代码中应被服务，即定期地被执行在要求的时间内以阻止一个 WDT 复位。

绝对最大率 [1, 2, 3]

参数	rating	单位
基准下的工作温度	0-+70 或－40to +85	℃
存储温度范围	-65to +150	℃
（/EA）/V_{PP} 管脚上电压到 V_{SS}	0to+13.0	V
在任何其他管脚上电压到 V_{SS}	-0.5to +6.5	V
每个 I/O 管脚上最大 I_{OL}	15	mA
功率消耗（以封装热传输为基础，不是器件功率消耗）	1.5	W

注：

1. 对这些列表的在绝对最大率下加压可导致对器件的永久损害。这仅是个器件的压力范围和功能操作在这些或任何一个条件而不是在此规格的 AC 和 DC 电气特性部份没有被暗示。
2. 这个产品包括规格化的电路设计用于内部器件的保护以免过度的静电损害。但是，仍然应小心以避免运用超过额定的最大值。
3. 参数是有效的是操作温度范围内除非其他特殊说明。所有电压以 V_{SS} 为基准，除非特殊说明。

DC 电气特性

T_{anb}=0℃-70℃,5V±10%或-40℃-+85℃；5V±5%；V_{SS}＝0V

标记	参数	测试条件	限度			单位
			最小	典型	最大	
V_{IL}	输入低电压	4.5V<V_{CC}<5.5V	-0.5		0.2V_{CC}-0.1	V
V_{IL2}	输入低电压到 p1.6/SCL, p1.7/SDA[11]		-0.5		0.3V_{DD}	V
V_{IH}	输入高电压（端口 0、1、2、3、/EA）		0.2V_{CC}+0.9		V_{CC}+0.5	V
V_{IH1}	输入高电压，XTAL1，RST		0.7V_{CC}		V_{CC}+0.5	V
V_{IH2}	输入高电压， p1.6/SCL, p1.7/SDA[11]		0.7V_{DD}		6.0	V
V_{OL}	输出低电压，端口 1、2、3[8]	V_{CC}=4.5V I_{OL}=1.6mA[2]			0.4	V
V_{OL1}	输出低电压，端口 0，ALE，/PSEN[7, 8]	V_{CC}=4.5V I_{OL}=3.2mA[2]			0.45	V
V_{OL2}	输出低电压， p1.6/SCL, p1.7/SDA	I_{OL}=3.0mA			0.4	V
V_{OH}	输出高电压，端口 1、2、3[3]	V_{CC}=4.5V I_{Oh}=-30μA	V_{CC}-0.7			V
V_{OH1}	输出高电压，(端口 0 在外部总线模式)，ALE[9]，/PSEN[3]	V_{CC}=4.5V I_{Oh}=-3.2mA	V_{CC}-0.7			V
I_{IL}	逻辑 0 输入电流，端口 1、2、3	V_{IN}=0.4V	-1		-75	μA
I_{TL}	逻辑 1—0 瞬变电流，端口 1、2、3[6]	V_{IN}=2.0V 见注 4			-650	μA
I_{LI}	输入漏电流，端口 0	0.45<V_{IN}<V_{CC}-0.3			±10	μA
I_{L2}	输入漏电流， p1.6/SCL, p1.7/SDA	0V<VI<6V 0V< V_{DD}<5.5V			10	μA
I_{CC}	电压提供电流（见图 37）:	见注 5				
	活跃模式（见注 5）					
	空闲模式（见注 5）					
	掉电模式或时钟终止（见图 44）	T_{anb}=0℃-70℃		20	100	μA
		T_{anb}=-40℃-+85℃			125	μA
	编程和可擦除模式	f_{OSC}=20MHz		60		mA
R_{RST}	内部复位下拉电阻		40		225	kΩ
C_{IO}	管脚电容[10]（除了/EA）				15	pF

注：

1. 典型的范围没有被保证，上面列出的值处于室温，5V。
2. 端口 0 和 2 的负载电容可导致寄生噪声在 ALE 的 V_{OL} 上和端口 1 和 3 被重叠。这噪声由于外部总线电容放电进端口 0 和端口 2 管脚当这些管脚在 1—0 瞬变时在总线操作期间。在最坏的情况下（负载电容>100PF），在 ALE 管脚上的噪声脉冲可超过 0.8V。在种情况下，可用一个施密特触发器去限制 ALE 或用一个施密特触发器 STROBE 输入锁定地址。I_{OL} 能超过这些条件如果没有单独的输出吸入电流超过 5mA 以及没有两个输出超过测试条件。
3. 端口 0 和 2 的负载电容可导致 ALE 和/PSEN 上的 V_{OH} 暂时性地下降低于 V_{CC}-0.7 规格当地址位正稳定时。
4. 端口 1、2、3 和源瞬变电流当它们被从外部 1—0 驱动。瞬变电流达到它的最大值当 V_{IN} 大约是 2V。
5. 见图 41 到 44，I_{CC} 测试条件，图 37 I_{CC} 对应的频率。
6. 这个值应用于 T_{anb}=0℃-＋70℃。
7. 端口 0、ALE、和/PSEN 负载电容＝100PF，所有其他输出的负载电容＝80PF。

8．在稳定状态（无瞬变）条件下，I_{OL} 必须被外部限制如下：

每个端口引脚最大 I_{OL}：15mA(这是 85℃的规格)

每个八位端口最大 I_{OL}：26mA

对于所有输出总的 I_{OL}：71 mA

9．ALE 被测试到 V_{OH1}，除了当 ALE 是 off，V_{OH} 是电压规格。

10．管脚电容被描述，但没有被测试。管脚电容低于 25PF。陶瓷管脚电容低于 15PF（除了/EA 是 25PF）。

11．P1.6 和 P1.7 的输入门槛电压满足 I^2C 规格，因此一个电压低于 1.5V，将被识别为逻辑 0，一个输入电压高于 3.0V，将被识别为逻辑 1。

AC 电气特性（6 个时钟模式）

T_{anb}=0℃-70℃,5V±10%或-40℃-+85℃；5V±5%；V_{SS}＝0V[1, 2, 3]

标记	数字	参数	可变时钟		20MHz 时钟[4]		单位
			最小	最大	最小	最大	
$1/t_{CLCL}$	30	振荡器频率	0	20	0	20	MHz
t_{LHLL}	30	ALE 脉冲宽度	t_{CLCL}-40		10		ns
t_{AVLL}	30	地址有效到 ALE 为低	$0.5t_{CLCL}$-20		5		ns
t_{LLAX}	30	ALE 为低后地址维持	$0.5t_{CLCL}$-20		5		ns
t_{LLIV}	30	ALE 低到有效指令输入		$2t_{CLCL}$-65		35	ns
t_{LLPL}	30	ALE 低到/PSEN 为低	$0.5t_{CLCL}$-20		5		ns
t_{PLPH}	30	/PSEN 脉冲宽度	$1.5t_{CLCL}$-45		30		ns
t_{PLIV}	30	/PSEN 低到有效指令输入		$1.5t_{CLCL}$-60		15	ns
t_{PXIX}	30	/PSEN 后输入指令维持	0		0		ns
t_{PXIZ}	30	/PSEN 后输入指令悬空		$0.5t_{CLCL}$-20		5	ns
t_{AVIV}	30	地址到有效指令输入		$2.5t_{CLCL}$-80		45	ns
t_{PLAZ}	30	/PSEN 低到地址悬空		10		10	ns
数据存储器							
t_{RLRH}	31，32	/RD 脉冲宽度	$3t_{CLCL}$-100		50		ns
t_{WLWH}	31，32	/WR 脉冲宽度	$3t_{CLCL}$-100		50		ns
t_{RLDV}	31，32	/RD 低到有效数据输入		$2.5t_{CLCL}$-90		35	ns
t_{RHDX}	31，32	/RD 后数据维持时间	0		0		ns
t_{RHDZ}	31，32	/RD 后数据悬空时间		t_{CLCL}-20		5	ns
t_{LLDV}	31，32	ALE 低到有效数据输入		$4t_{CLCL}$-150		50	ns
t_{AVDV}	31，32	地址到有效数据输入		$4.5t_{CLCL}$-165		60	ns
t_{LLWL}	31，32	ALE 低到/RD 低或/WR 低	$1.5t_{CLCL}$-50	$1.5t_{CLCL}$+50	25	125	ns
t_{AVWL}	31，32	地址有效到/RD 低或/WR 低	$2t_{CLCL}$-75		25		ns
t_{QVWX}	31，32	数据有效到/WR 瞬变	$0.5t_{CLCL}$-25		0		ns
t_{WHQX}	31，32	/WR 后数据维持时间	$0.5t_{CLCL}$-20		5		ns
t_{QVWH}	32	数据有效到/WR 为高	$3.5t_{CLCL}$-130		45		ns
t_{RLAZ}	31，32	/RD 低到地址地址悬空		0		0	ns
t_{WHLH}	31，32	/RD 或/WR 高到 ALE 为高	$0.5t_{CLCL}$-20	$0.5t_{CLCL}$+20	5	45	ns
外部时钟							
t_{CHCX}	34	为高	20	t_{CLCL}- t_{CLCX}			ns
t_{CLCX}	34	为低	20	t_{CLCL}- t_{CHCX}			ns
t_{CLCH}	34	上升		5			ns
t_{CHCL}	34	下降		5			ns
循环寄存器							
t_{XLXL}	33	串行端时钟循环时间	$6t_{CLCL}$		300		ns
t_{QVXH}	33	输出数据设定到时钟上升边	$5t_{CLCL}$-133		117		ns
t_{XHQX}	33	时钟上升边后输出数据设定	t_{CLCL}-30		20		ns
t_{XHDX}	33	时钟上升边后输入数据维持	0		0		ns
t_{XHDV}	33	时钟上升边到输入数据有效		$5t_{CLCL}$-133		117	ns

注：

1．参数是有效的在操作温度范围之上，除非特殊说明。

2．端口 0、ALE、和/PSEN 负载电容＝100PF，所有其他输出的负载电容＝80PF。

3．微型控制器到器件的连接带有悬空时间高达 45ns 是允许的，限定的总线竞争将不会导致对端口 0 驱动器的损害。

4．部份被测试到 2MHz，但被保证在低于 0Hz 操作。

AC 电气特性（6 个时钟模式）（继续）

T_{anb}=0℃-70℃,5V±10%或-40℃-+85℃；5V±5%；V_{SS}＝0V[1, 2,]

标记	参数	输入	输出
I^2C 接口			
$t_{HD,\ STA}$	开始条件维持时间	≧$7t_{CLCL}$	>4.0μs[4]
t_{LOW}	SCL 低时间	≧$8t_{CLCL}$	>4.7μs[4]
t_{HIGH}	SCL 高时间	≧$7t_{CLCL}$	>4.0μs[4]
t_{RC}	SCL 上升时间	≦1μs	-[5]
t_{FC}	SCL 下降时间	≦0.3μs	<0.3μs[6]
$t_{SU,\ DAT1}$	数据设定时间	≧250ns	>10 t_{CLCL}- t_{RD}
$t_{SU,\ DAT2}$	SDA 设定时间（在 rep，开始条件前）	≧250ns	>1μs[4]
$t_{SU,\ DAT3}$	SDA 设定时间（在停止条件前）	≧250ns	>4t_{CLCL}
$t_{HD,\ DAT}$	数据维持时间	≧0ns	>4t_{CLCL}- t_{FC}
$t_{SU,\ STA}$	重复的开始设定时间	≧$7t_{CLCL}$[4]	>4.7μs[4]
$t_{SU,\ STO}$	停止条件设定时间	≧$7t_{CLCL}$[4]	>4.0μs[4]
t_{BUF}	总线空闲时间	≧$7t_{CLCL}$[4]	>4.7μs[4]
t_{RD}	SDA 上升时间	≦1μs[7]	-[5]
t_{FD}	SDA 下降时间	≦300ns[7]	<0.3μs[6]

注：

1．参数是有效的在操作温度范围之上，除非特殊说明。

2．端口 0、ALE、和/PSEN 负载电容＝100PF，所有其他输出的负载电容＝80PF。

3．这些值被刻划过，但没有被 100％产品测试过。

4．以 100Kbit/s，其他位速度这个值和 100Kbit/s 成反比。

5．由外部总线电容和外部拉电阻决定，这必须<1μs。

6．在 SDA 和 SCL 线上带有低于 3 t_{CLCL} 持续的尖峰将被滤掉，在 SDA 和 SCL 线上最大电容＝400PF。

7．t_{CLCL}＝1/f_{OSC}=一个振荡器时钟周期在管脚 XTAL1。

AC 电气特性（12 个时钟模式）

T_{anb}=0℃-70℃,5V±10%或-40℃-+85℃；5V±5%；V_{SS}＝0V[1, 2, 3]

标记	数字	参数	可变时钟		20MHz 时钟[4]		单位
			最小	最大	最小	最大	
1/t_{CLCL}	30	振荡器频率	0	20	0	33	MHz
t_{LHLL}	30	ALE 脉冲宽度	2t_{CLCL}-40		21		ns
t_{AVLL}	30	地址有效到 ALE 为低	t_{CLCL}-25		5		ns
t_{LLAX}	30	ALE 为低后地址维持	t_{CLCL}-25		5		ns
t_{LLIV}	30	ALE 低到有效指令输入		4t_{CLCL}-65		55	ns
t_{LLPL}	30	ALE 低到/PSEN 为低	t_{CLCL}-25		5		ns
t_{PLPH}	30	/PSEN 脉冲宽度	3t_{CLCL}-45		45		ns
t_{PLIV}	30	/PSEN 低到有效指令输入		3t_{CLCL}-60		30	ns
t_{PXIX}	30	/PSEN 后输入指令维持	0		0		ns
t_{PXIZ}	30	/PSEN 后输入指令悬空		t_{CLCL}-25		5	ns
t_{AVIV}	30	地址到有效指令输入		5t_{CLCL}-80		70	ns
t_{PLAZ}	30	/PSEN 低到地址悬空		10		10	ns

数据存储器							
t_{RLRH}	31，32	/RD 脉冲宽度	$6t_{CLCL}$-100		82		ns
t_{WLWH}	31，32	/WR 脉冲宽度	$6t_{CLCL}$-100		82		ns
t_{RLDV}	31，32	/RD 低到有效数据输入		$5t_{CLCL}$-90		60	ns
t_{RHDX}	31，32	/RD 后数据维持时间	0		0		ns
t_{RHDZ}	31，32	/RD 后数据悬空时间		t_{CLCL}-20		32	ns
t_{LLDV}	31，32	ALE 低到有效数据输入		$4t_{CLCL}$-150		90	ns
t_{AVDV}	31，32	地址到有效数据输入		$4.5t_{CLCL}$-165		105	ns
t_{LLWL}	31，32	ALE 低到/RD 低或/WR 低	$1.5t_{CLCL}$-50	$1.5t_{CLCL}$+50	40	140	ns
t_{AVWL}	31，32	地址有效到/RD 低或/WR 低	$2t_{CLCL}$-75		45		ns
t_{QVWX}	31，32	数据有效到/WR 瞬变	t_{CLCL}-30		0		ns
t_{WHQX}	31，32	/WR 后数据维持时间	t_{CLCL}-25		5		ns
t_{QVWH}	32	数据有效到/WR 为高	$7t_{CLCL}$-130		80		ns
t_{RLAZ}	31，32	/RD 低到地址地址悬空		0		0	ns
t_{WHLH}	31，32	/RD 或/WR 高到 ALE 为高	t_{CLCL}-25	t_{CLCL}+25	5	55	ns
外部时钟							
t_{CHCX}	34	为高	17	t_{CLCL}- t_{CLCX}			ns
t_{CLCX}	34	为低	17	t_{CLCL}- t_{CHCX}			ns
t_{CLCH}	34	上升		5			ns
t_{CHCL}	34	下降		5			ns
循环寄存器							
t_{XLXL}	33	串行端时钟循环时间	$12t_{CLCL}$		300		ns
t_{QVXH}	33	输出数据设定到时钟上升边	$10t_{CLCL}$-133		167		ns
t_{XHQX}	33	时钟上升边后输出数据设定	$2t_{CLCL}$-80		50		ns
t_{XHDX}	33	时钟上升边后输入数据维持	0		0		ns
t_{XHDV}	33	时钟上升边到输入数据有效		$10t_{CLCL}$-133		167	ns

注：

1．参数是有效的在操作温度范围之上，除非特殊说明。

2．端口 0、ALE、和/PSEN 负载电容＝100PF，所有其他输出的负载电容＝80PF。

3．微型控制器到器件的连接带有悬空时间高达 45ns 是允许的，限定的总线竞争将不会导致对端口 0 驱动器的损害。

4．部份被测试到 3.5MHz，但被保证在低于 0Hz 操作。

AC 电气特性（12 个时钟模式）（继续）

T_{anb}=0℃-70℃,5V±10%或-40℃-+85℃；5V±5%；V_{SS}＝0V[1, 2,]

标记	参数	输入	输出
I^2C 接口			
$t_{HD,\ STA}$	开始条件维持时间	≧$14t_{CLCL}$	>4.0μs[4]
t_{LOW}	SCL 低时间	≧$16t_{CLCL}$	>4.7μs[4]
t_{HIGH}	SCL 高时间	≧$14t_{CLCL}$	>4.0μs[4]
t_{RC}	SCL 上升时间	≦1μs	-[5]
t_{FC}	SCL 下降时间	≦0.3μs	<0.3μs[6]
$t_{SU,\ DAT1}$	数据设定时间	≧250ns	>20 t_{CLCL}- t_{RD}
$t_{SU,\ DAT2}$	SDA 设定时间（在 rep，开始条件前）	≧250ns	>1μs[4]
$t_{SU,\ DAT3}$	SDA 设定时间（在停止条件前）	≧250ns	>8t_{CLCL}
$t_{HD,\ DAT}$	数据维持时间	≧0ns	>8t_{CLCL}- t_{FC}
$t_{SU,\ STA}$	重复的开始设定时间	≧$14t_{CLCL}$[4]	>4.7μs[4]
$t_{SU,\ STO}$	停止条件设定时间	≧$14t_{CLCL}$[4]	>4.0μs[4]
t_{BUF}	总线空闲时间	≧$14t_{CLCL}$[4]	>4.7μs[4]
t_{RD}	SDA 上升时间	≦1μs[7]	-[5]
t_{FD}	SDA 下降时间	≦300ns[7]	<0.3μs[6]

注：

1. 参数是有效的在操作温度范围之上，除非特殊说明。
2. 端口 0、ALE、和/PSEN 负载电容＝100PF，所有其他输出的负载电容＝80PF。
3. 这些值被刻划过，但没有被 100％产品测试过。
4. 以 100Kbit/s，其他位速度这个值和 100Kbit/s 成反比。
5. 由外部总线电容和外部拉电阻决定，这必须<1μs。
6. 在 SDA 和 SCL 线上带有低于 3 t_{CLCL} 持续的尖峰将被滤掉，在 SDA 和 SCL 线上最大电容＝400PF。
7. t_{CLCL}＝1/f_{OSC}＝一个振荡器时钟周期在管脚 XTAL1。
   对于 63ns< t_{CLCL}<285ns(16MHz>f_{OSC}>3.5MHz)，I^2C 接口满足 I^2C 总线规格对于速率高达 100Kbit/s。

AC 标记的解释

每个时序标记有五个特征，第一个特征总是“t”（＝time）。其他特征，取决于它们的位置，表明一个信号的名字或此信号的逻辑状态。这些是：

A－	地址	Q－	输出数据
C－	时钟	R－	/RD 信号
D－	输入数据	t－	时间
H－	逻辑高电平	V－	有效
I－	指令（程序存储器内容）	W－	/WR 信号
L－	逻辑高电平，或 ALE	X－	不再有一个有效的逻辑电平
P－	/PSEN	Z－	悬空

例如：

t_{AVLL}＝地址有效到 ALE 为低时间。

t_{LLPL}＝ALE 低到/PSEN 为低时间。

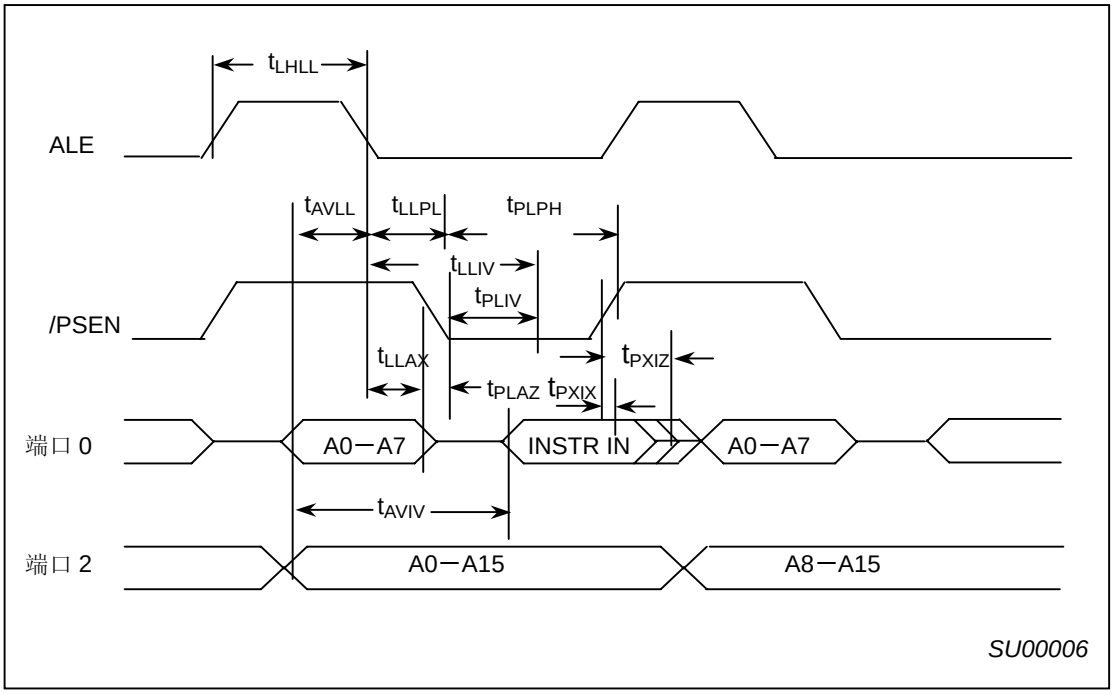


图 30 外部程序存储器读循环

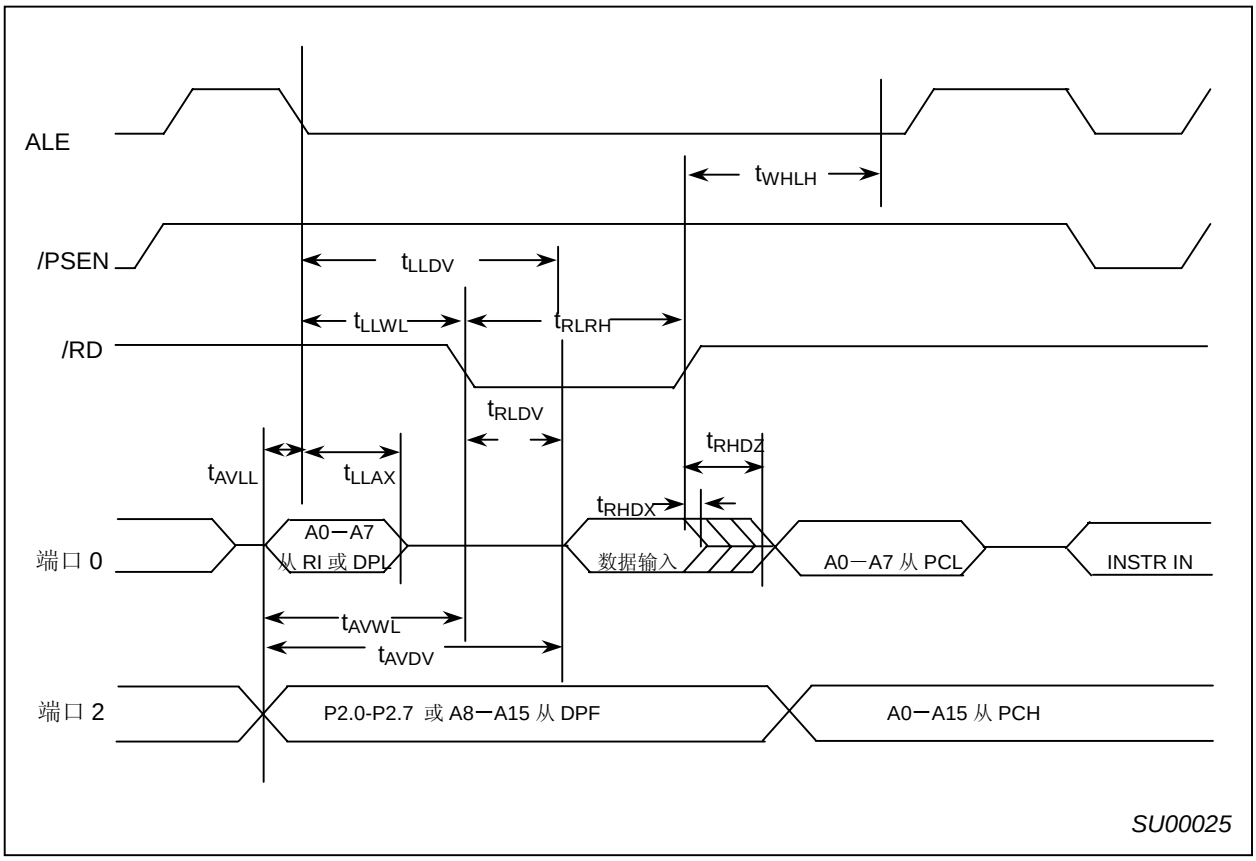


图 31 外部数据存储器读周期

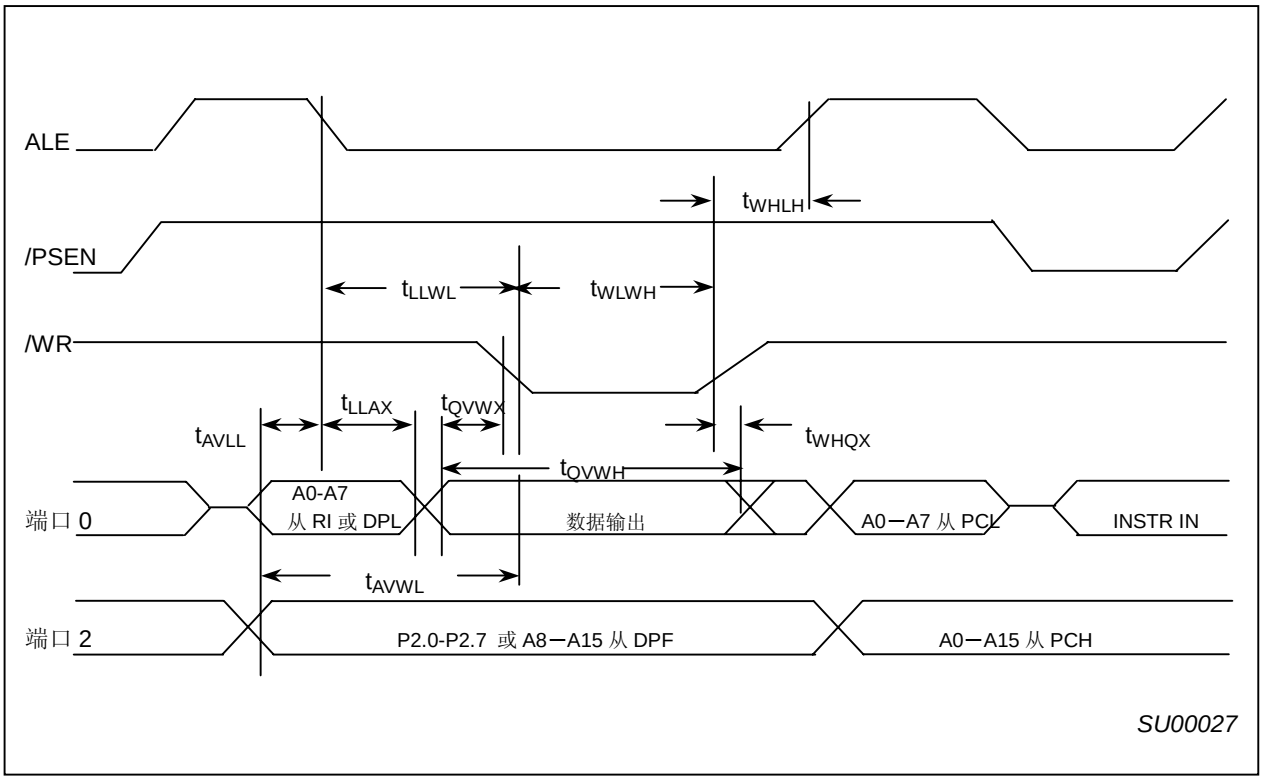


图 32 外部数据存储器写周期

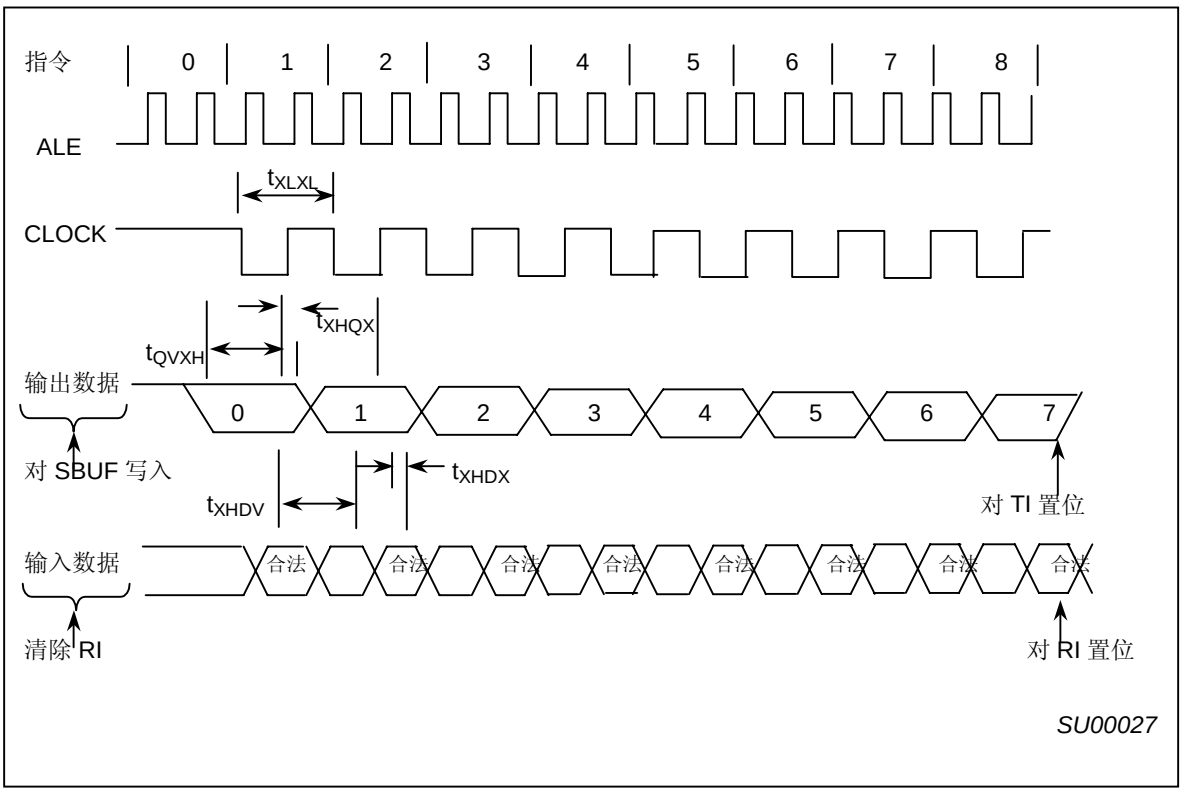


图 33 循环寄存器模式定时

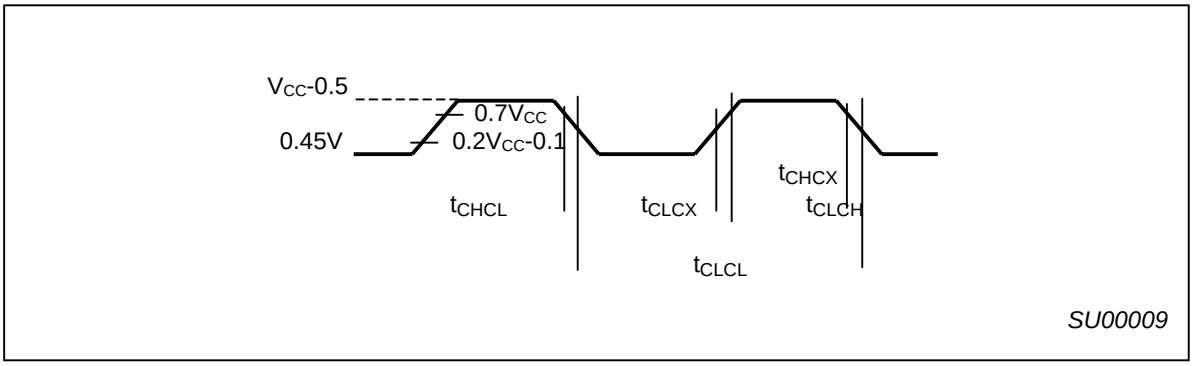
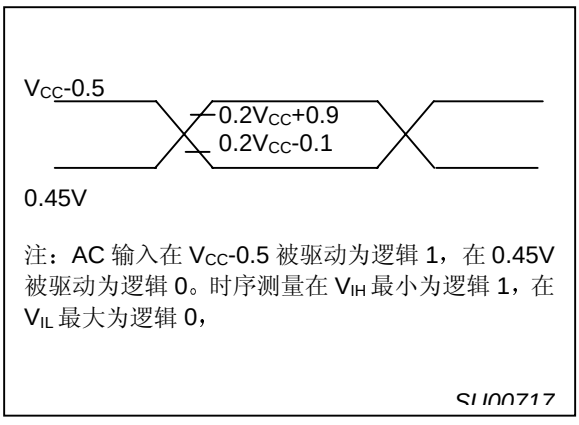


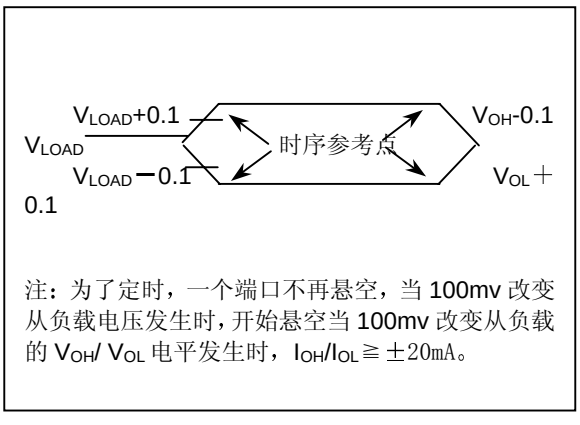
图 34 外部时钟驱动



注：AC 输入在 V_{CC}-0.5 被驱动为逻辑 1，在 0.45V 被驱动为逻辑 0。时序测量在 V_{IH} 最小为逻辑 1，在 V_{IL} 最大为逻辑 0，

SU00717

图 35 AC 测试输入/输出



注：为了定时，一个端口不再悬空，当 100mv 改变从负载电压发生时，开始悬空当 100mv 改变从负载的 V_{OH}/ V_{OL} 电平发生时，I_{OH}/I_{OL}≅±20mA。

图 36 悬浮波形

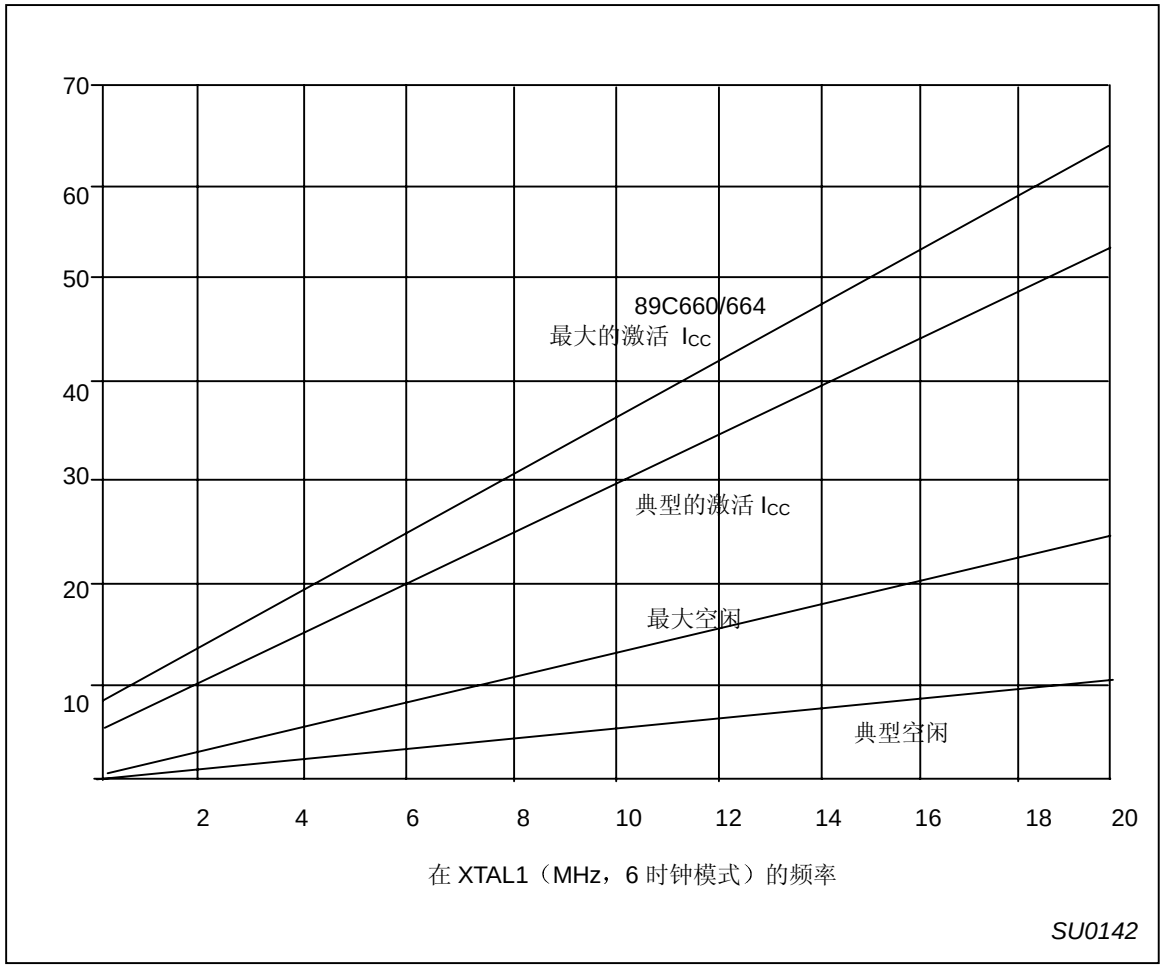


图 37 I_{CC}VS 频率仅当器件的频率规格在测试下是合法的

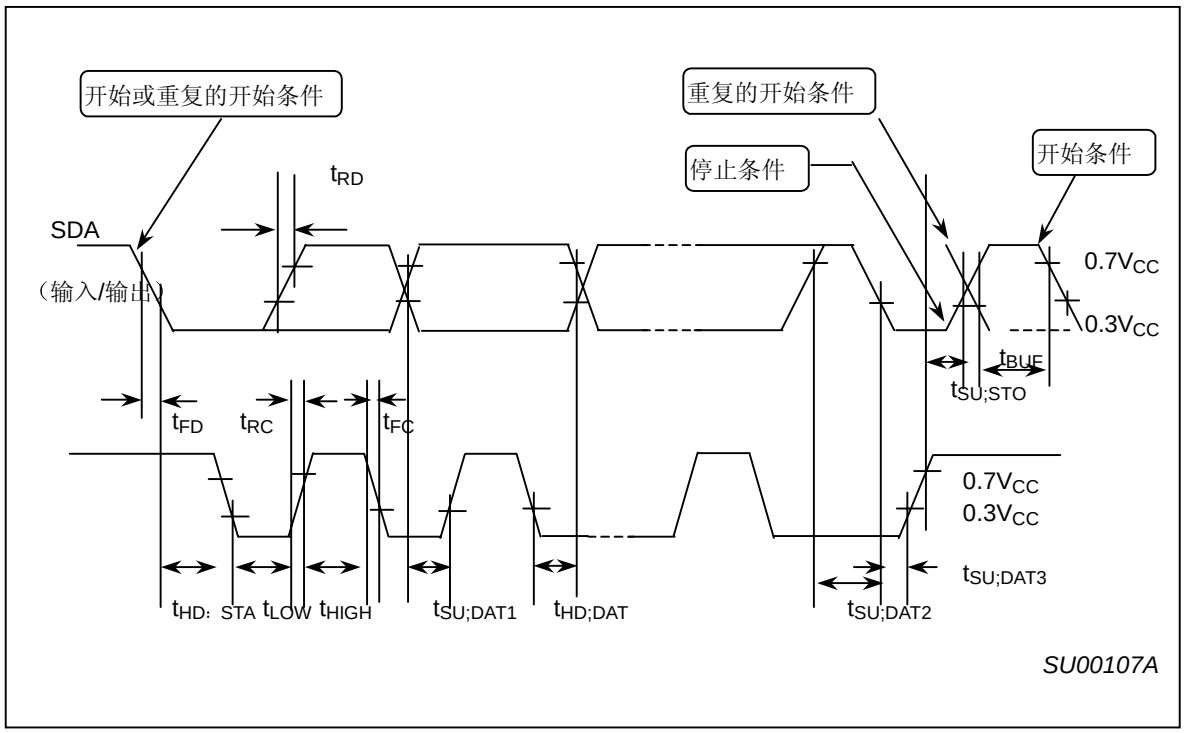
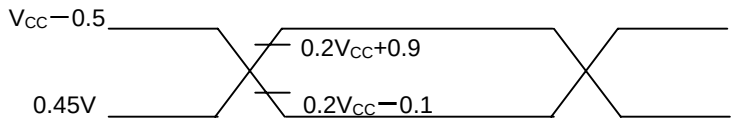


图 38 定时 SI01（I^2C）接口



注：
在测试期间 AC 输入在 V_{CC}－0.5 时为逻辑 1，在 0.45V 时为逻辑 0。
时序测量在 V_{IH} 最小为逻辑 1，在 V_{IL} 最大为逻辑 0，

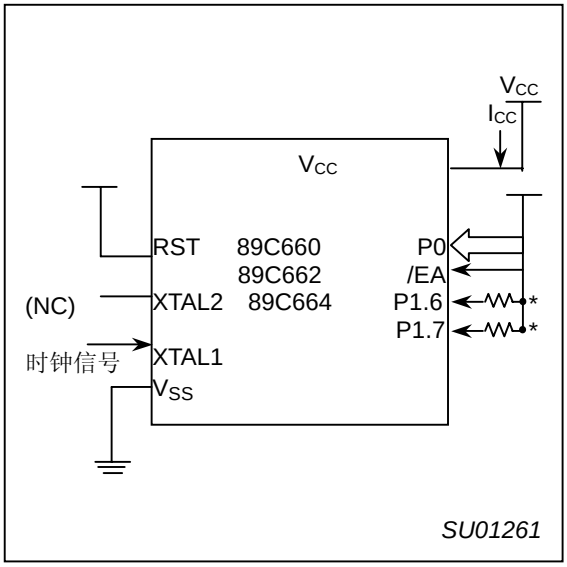
SU00010

图 39 AC 测试输入/输出

注：为了定时，一个端口不再悬空，当 100mv 改变从负载电压发生时，开始悬空当 100mv 改变从负载的 V_{OH}/ V_{OL} 电平发生时，I_{OH}/I_{OL}≧±20mA。

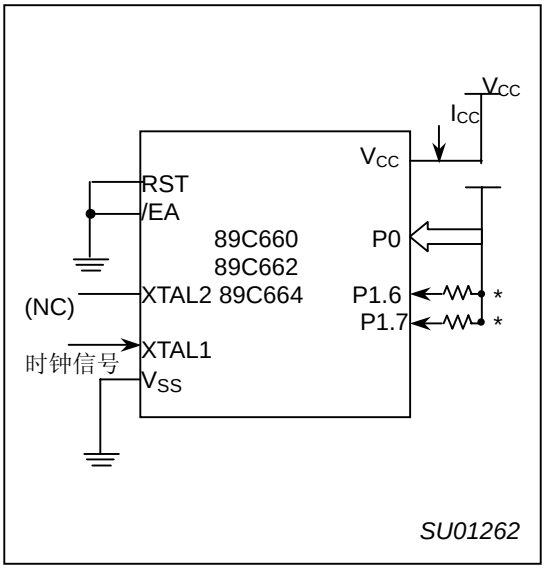
SU00011

图 40 悬浮波形



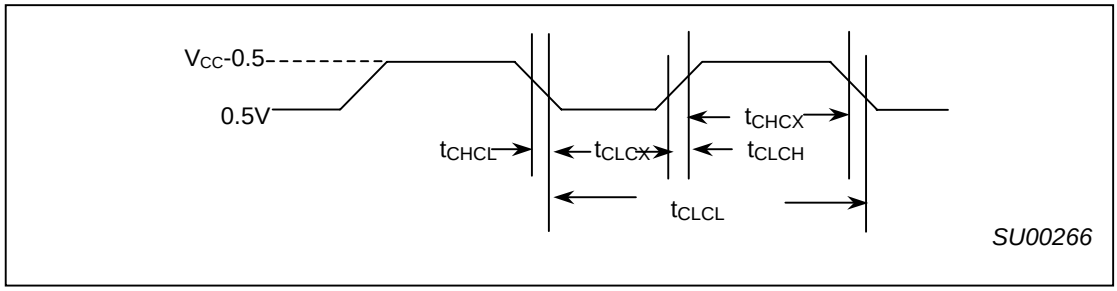
SU01261

图 41 I_{CC} 测试条件，激活模式
所有其他管脚没有被连接



SU01262

图 42 I_{CC} 测试条件，空闲模式
所有其他管脚没有被连接



SU00266

43 时钟信号波形用 I_{CC} 测试在激活模式和空闲模式

t_{Clcl}= t_{CHCL}=10ns

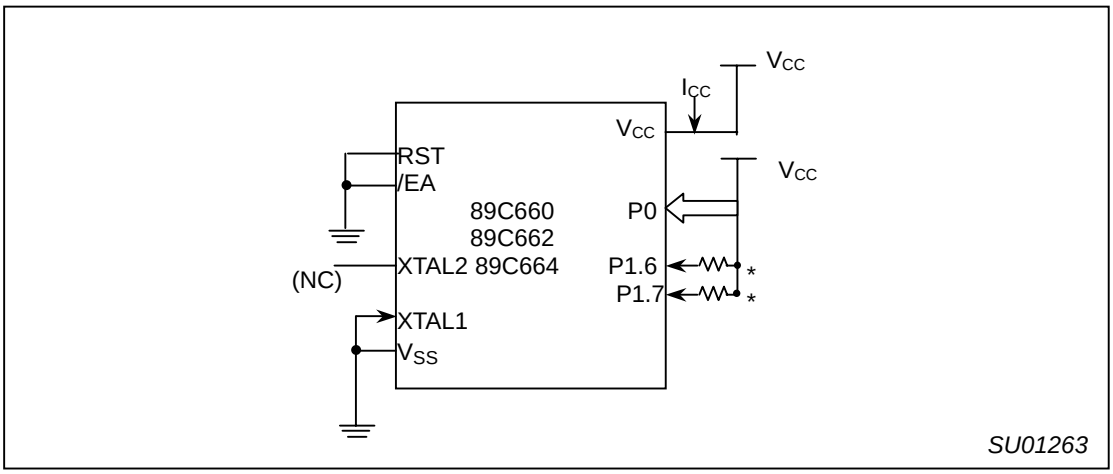


图 44 I_{CC} 测试条件，掉电模式所有其他管脚没有被连接；V_{CC}＝2Vto 2.5V

FLASH EPROM 存储

总述

P89C660/662/664 FLASH EPROM 存储器增加了线内电擦除和编程功能。FLASH 可被像字节一样读和写。片内擦除操作可以擦掉整个程序存储器。这个擦除功能能够擦掉任何一个 FLASH 字节块。系统内编程和平行编程都可以达到。片内擦除和写时序操作归于用户编程接口。

P89C660/662/664 FLASH 可靠地保存存储器内容甚至在 10，000 擦除和编程周期后。单元被设计成优化擦除和编程结构。另外，先进的 tunnel oxide processing 和低 electric fields 用于擦除和编程操作，产生可靠循环。P89C660/662/664 使用＋5V V_{PP}，以完成编程/擦除算法。

特征：

• 带有块擦除的 FLASH EPROM 内部程序存储器

• 内部有固定的 1K 字 boot ROM，包括低电平系统内编程子程和默认的串行负载。用户程序能够调用这些子程去完成应用内程序编程（IAP）。Boot rom 可被关掉以完成对所有的 64K 字节 FLASH 存储器的访问。

• 引导矢量允许用户的 Flash 代码可以驻留在 Flash 存储空间的任何地方。对于用户来说，这个结构提供了灵活性。

• 在引导 ROM 里默认的代码允许经过串行端编程而不需要用户的代码。

• 外部有高达 64K 字节存储器，如果内部程序存储器失效的话（/EA＝0）。

• 编程和擦除电压＋5V（＋12Vtolerant）。

• 读/编程/擦除：

－字节 wise 读（100ns 访问时间）

－字节编程（10μs）

－典型的擦除次数：

在 2 秒内块擦除（8K 或 16K 字节）

在 2 秒内完全擦除（64K 字节）

• 带有 87C51 可兼容硬件接口的平行编程

• 系统内编程

• Flash 里有可编程的安全位。

• 每个字节的 10，000 最小擦除/编程
• 10 年最小数据保持

飞利浦 89C51 以 FLASH 为基础的控制器特征

Flash 组织

P89C660/662/664 包括 Flash 程序存储器的 16KB/32KB/64K 字节。这个存储器被组织成 5 个分开的块。头两块是 8K 字节，存储空间从 0－3FFF hex。最后三块是 16K 字节，地址从 4000 到 FFFF hex。

图 45 显示了 Flash 存储器结构。

Flash 编程和擦除

Flash 存储器有三种擦除或编程的方法可被使用。首先，Flash 可被编程或擦除在 end-user 应用里，通过调用低电压程序通过引导 ROM 里的通用入口点。end-user 应用必须是执行的代码从不同的块而不是正被擦除或正编程的块。第二，片内 ISP 引导代码可被调入。ISP 引导代码将轮流调入低电压程序通过相同的被 end-user 应用所使用的引导 ROM 里的通用入口点。第三，Flash 可被编程或擦除使用平行的方法，通过使用一个商业级的编程器。这些器件所使用的平行编程方法和由 EPROM87C51 使用的方法一样，但是它不是完全相同，商业级编程器需要这些器件的支持。

引导 ROM

当控制器对它自己的 Flash 存储器编程时，所有低级细节由永久性地固定在引导 ROM 里的 1K 字节的代码处理，独立于 Flash 存储器。一个用户程序带有相应参数调用通用入口点在引导 ROM 里，以完成预期的操作。引导 ROM 操作包括：块擦除，程序字节，确认字节，程序安全锁定位等等。引导 ROM 覆盖了程序存储器空间在地址空间的顶部从 FC00－FFFF hex，当它被使能时。引导 ROM 可被关掉，因此 Flash 存储器上层的 1K 字节是可访问的。

上电复位代码执行

P89C660/662/664 包括两个特殊功能寄存器：引导矢量和状态字节。在复位的下降边，P89C660/662/664 检查状态字节的内容。如果状态字节被设置为零，上电执行在位置 0000H 开始，这是用户程序代码的正常开始位置。当状态字节被设置为其他值，经引导矢量的内容被用作为执行地址的高字节，低字节为 00H。工厂默认值是 0FCH，相应于地址 0FC00H，用于工厂掩膜－ROMISP 引导代码。一个用户代码可被写入，A custom boot loader can be written with the boot vector set to the custom boot loader。

注：当擦除引导矢量或状态字节，两个字节同时被擦除。在对状态字节擦除和更新后，必须对引导矢量再编程。

引导代码的硬件激活

引导代码可被执行，通过将 PSEN 维持为低，P2.7 为高，/EA 大于 V_{IH}（像＋5V），ALE 为高（或不连接）在 RESET 的下降边。这是同样的效果当没有零的状态字节。这允许一个程序被建立将正常地执行用户代码但能被人为地强制进入 ISP 操作。

如果引导矢量的工厂默认值被改变，它将不再指向 ISP 掩膜引导代码。如果发生这样情况，仅有的方法是改变引导矢量的内容，通过平行编程方法，如果用户程序不包括一个定制的用于提供引导矢量和状态字节擦除和再编程的代码。

对 Flash 编程后，状态字节应被编程为零，为了允许用户程序代码的扫执行在地址 0000H 开始。

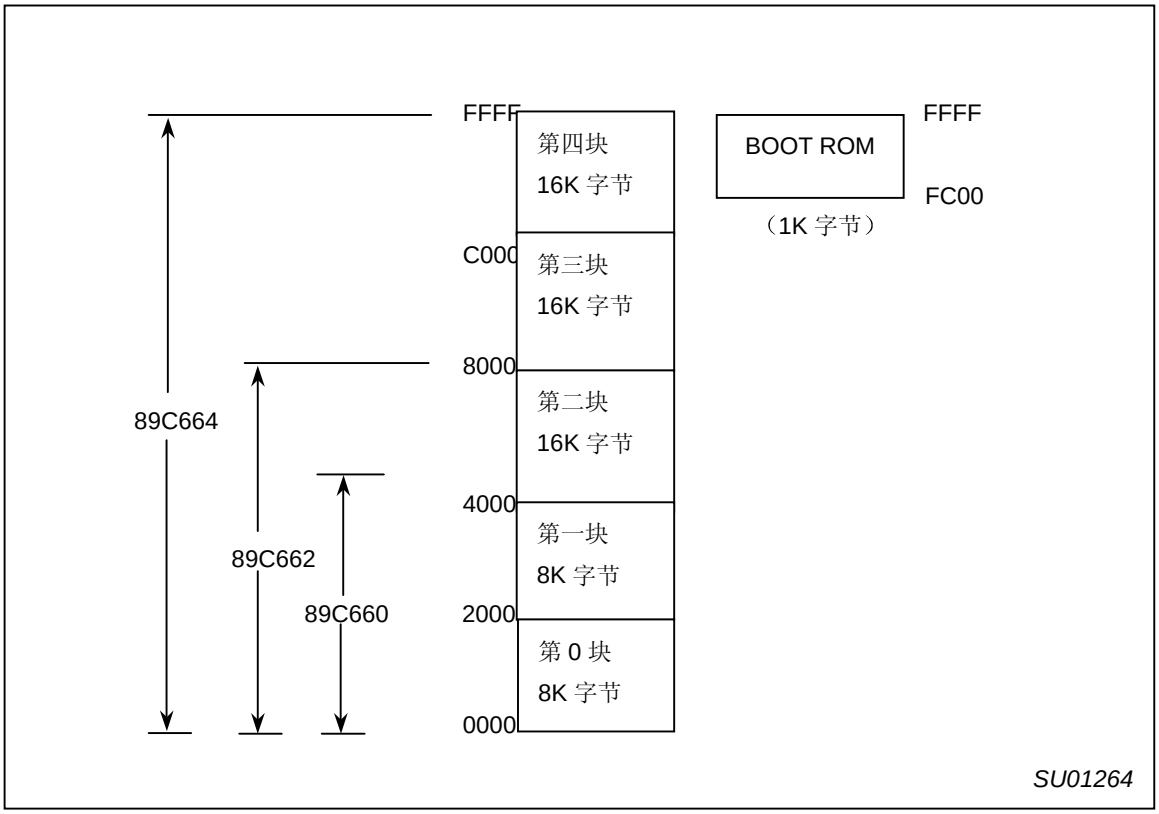


图 45 Flash 存储器构造

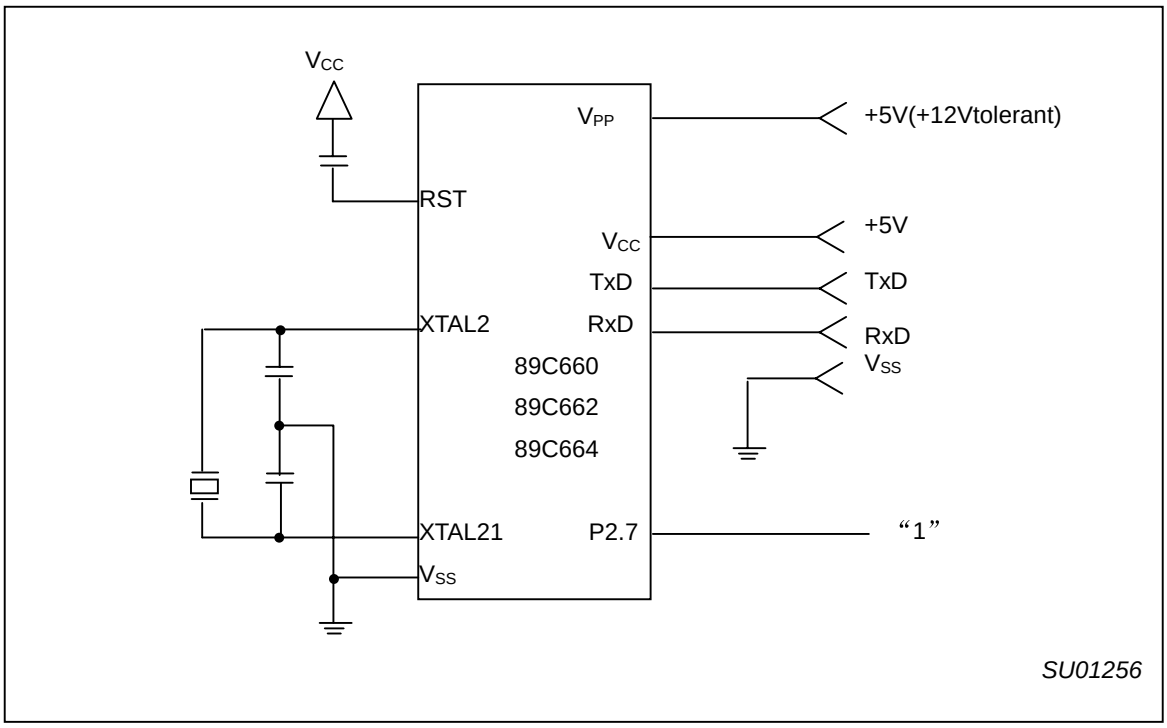


图 46 带有最少管脚的系统内编程

系统内编程（ISP）

系统内编程（ISP）被完成，而不需要将控制器从系统中移走。系统内编程（ISP）工具由一系列内部硬件资源和相应的内部固件组成以促进 P89C660/662/664 的遥控编程通过串行端。这个固件由飞利浦提供，并被嵌进每个 P89C660/662/664。

飞利浦系统内编程（ISP）工具已做在电路内，用最少额外的代价在嵌入式应用中编程可达到。

ISP 功能使用五个管脚：TxD,RxD,V_{SS}, V_{CC} 和 V_{PP}（见图 46）。仅需要一个小连接器将用户程序接到外部电路以便使用这个特征。V_{PP} 应被尽量地去耦，并且不允许超过数据表的限度。

使用系统内编程（ISP）

ISP 特征允许一个广泛的波特率范围在用户程序中被使用，独立于振荡器频率。它也是可调节到振荡器频率广泛的范围。这通过在接收的字节里对一个单个位的位时测量来完成。这个信息然后被用来对波特率进行编程根据定时器数目，以振荡器频率为基准。ISP 特征要求一个初始字节（一个大写字母 U）被送至 P89C660/662/664 以确定波特率。ISP 固件提供接收字节的自动回应。

波特率初始化一旦被完成，ISP 固件将接收内部 HEX 型 record。内部 HEX 型 record 由 ASCII 字节组成用于代表 hex 二进制值，总结如下：

：NNAAAARRDD..DDCC<crlf>

在内部 hex record，“NN”代表数据字节的数目在 record 里。P89C660/662/664 将接收高达 16 个(10H)数据字节。“AAAA”串代表 record 里的第一个字节的地址。如果在 record 里是零，这个字段总被设为 0000。“RR”串表明 record 类型。一种“00” record 类型是个数据 record。一种“01” record 类型表明文件结束的标记。在这个应用里，额外的 record 类型将被加入表明命令或数据。在 record 里数据字节的最大数目被限定为 16 字节(二进制)。ISP 命令被总结在表 9。

作为一个 record，被 P89C660/662/664 接收，record 里的信息被存储起来和 checksum 计算被完成。被 record 类型指明的操作直到完整的 record 被接收到才能完成。若一个错误发生在 checksum，P89C660/662/664 将发送一个“X”表明一个 checksum 错误。如果 checksum 计算被发现匹配 record 里的 checksum，则命令将被执行。在许多情况下，record 的成功接收将被表明，通过发送一个“a”字节出去（显示内部程序存储器的内容是个异常）。

在一个数据 record 情况下（record 类型 00），一个另外的检查被写成“A”将不会被发送除非 record 里的 checksum 匹配计算过的 checksum，record 里的所有字节被成功地编程。对于一个数据 record，一个“X”表明 checksum 匹配失败，一个“R”字节表明一个数据字节没有正确地编程。必须发送一个 02record（指定的振荡器频率）到 P89C660/662/664 在编程数据之前。

ISP 设备被设计用于特殊的晶振不被要求，为了产生波特率或时序编程脉冲，用户需要对 P89C660/662/664 提供按要求产生正确的时序的信息。record 类型 02 专用于此类。

WinISP，一种软件工具使带有 PC 的 ISP 编程有效，在飞利浦网站上可找到。另外，在网站上列出了第三批商业级串行和并行的编程器。

表 9 由系统内编程的内部－HEX record

RECORD 类型	命令/数据功能
00	编程数据 ：nnaaaa00dd....ddcc Nn　＝record 里的字节数目 Aaaa　＝ record 里的第一个字节的存储地址 dd....dd　＝数据字节 cc　＝checksum 例如：10008000AF5F67F0602703E0322CFA92007780C3FD
01	文件结束（EOF），空操作 ：xxxxxx01cc xxxxxx　＝所需的字段，但值为任意 cc　＝checksum 例如：00000001F
02	指定的振荡器频率 ：01xxxx02ddcc xxxxxx　＝所需的字段，但值为任意 dd　＝整型的振荡器频率下舍到最近的 MHz 例如：　0100000210ED　（dd=10h=16 用于 16.0-16.9 MHz）
03	多方面的写功能 ：nnxxxx03ffssddcc nn　＝record 里的(hex)字节数目 xxxx　＝所需的字段，但值为任意 03　＝写功能 ff　＝子功能代码 ss　＝选择代码 dd　＝数据输入（作为需要） cc　＝checksum 子功能代码＝01（可擦除块） ff　＝01 ss　＝块码如下所示： 块 0，　0K　to　8K，　00H 块 1，　8K　to　16K，　20H 块 2，　16K　to　32K，　40H 块 3，　32K　to　48K，　80H 块 4，　48K　to　64K，　C0H 例如：0200000301C03C　擦除块 4 子功能代码＝04（可擦除引导矢量和状态字节） ff　＝04 ss　＝为任意值 例如：020000030400F7　擦除引导矢量和状态字节 子功能代码＝05（程序安全位） ff　＝05 ss　＝00 程序安全位 1（禁止对 Flash 写入） 01 程序安全位 2（禁止 Flash 确认） 02 程序安全位 3（外部存储器无效） 例如： ：020000030501F5　程序安全位 2

	子功能代码＝06（程序状态字节或引导矢量） ff　＝06 ss　＝00 程序状态字节 　　01 程序引导矢量 例如：0100000307F5　全擦除
04	器件数据或空格检查－Record 类型 04 的显示导致整个 Flash 阵列的内容被发送出端口在一个显示的格式里。这个显示由一个地址和带有启始地址的 16K 字节的内容组成。如果安全位 2 已被编程，没有器件内容的显示发生。到串行端的数据被任一个接收的字节初始化，以及被任何一个字节的接收所中止。 通用的功能 04 的格式 　：05xxxx04sssseeeeffcc 05　＝record 里的(hex)字节数目 xxxx　＝所需的字段，但值为任意 04　＝“显示器件数据或空格检查”功能代码 ssss　＝启始地址 eeee　＝结束地址 ff　＝子功能 　　00＝显示数据 　　01＝空格检查 cc　＝checksum 例如：　0500000440004FFF069　显示 4000－4FFF
05	多方面的读功能 通用的功能 05 的格式 　：02xxxx05ffsscc 02　＝ record 里的(hex)字节数目 xxxx　＝所需的字段，但值为任意 05　＝“多方面读”功能代码 ffss　＝子功能和选择代码 cc　＝checksum 例如：020000050001FB　读器件识别标志字节－器件 id #1
06	波特率直接载入 通用的功能 06 的格式 　：02xxxx06hhllcc 02　＝record 里的(hex)字节数目 xxxx　＝所需的字段，但值为任意 06　＝“波特率直接载入”；功能代码 hh　＝定时器 2 的高字节 ll　＝定时器 2 的低字节 cc　＝checksum 例如：02000006F500F3

应用内编程方法

几个应用内编程调用（IAP）是可达到的，通过一个应用程序允许 Flash 选择性的擦除和编程。所有的调用被完成通过一个通用接口，PGM_MTP。编程功能由控制器的寄存器的设置所决定在表明对 PGM_MTP 调用在 FFF0H 之前。振荡器的频率是个整型数，下舍到最近的 MHz。例如，R0 设为 11，即是 11.0592 MHz。结果被返回到寄存器里。IAP 调用如表 10 所示。

看门狗的使用（WDT）

89C66X 支持 WDT 的使用在 IAP 里。用户确定看门狗被喂养通过对功能参数的最重要的位置位，看门狗在 R1 里被通过先于调用 PGM_MTP。WDT 功能仅被支持用于块擦除当使用快速块擦除。快速块擦除被一个带有寄存器 R0＝0 的块擦除完成。在 IAP 期间要求 WDT 喂养应被在使用 WDT 应用中完成，因为喂养 WDT 将启动 WDT，如果 WDT 不是正在工作。

表 10 IAP 调用

IAP 调用	参数
编程数据字节	参数输入： R0 ＝振荡器频率（整型） R1 ＝02h R1 ＝82h(WDT 喂养,RX2 和 66X) DPTR＝到程序的字节地址 ACC ＝到程序的字节 参数的返回 ACC ＝00，如果通过，！00 如果失败 例子： ；*****程序器件数据〔Ddata)***** ；*****ACC 保持数据到写 ；***** DPTR 保持字节的地址到写***** ；*****返回带有 ACC ＝00h 如果成功，除非 ACC NEQ 00h WRData： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃02H　；程序数据功能 MOV　A，Mydata　；要写的数据 MOV　DPTR，Address　；确认要读的字节 CALL　PGM_MTP　；执行这个功能 RET
擦除块	参数的输入： R0 ＝振荡器频率（整型） R0 ＝0（快速擦除，RX2 和 66X） R1 ＝01h R1 ＝81h(WDT 喂养,RX2 和 66X；只能快速擦除使用) DPH ＝块码如下所示： 块 0，　0K　to　8K，　00H 块 1，　8K　to　16K，　20H 块 2，　16K　to　32K，　40H 块 3，　32K　to　48K，　80H 块 4，　48K　to　64K，　C0H DPL＝00h 参数返回 空 例子： ；*****擦除代码存储块***** ；*****DPH（7：5）表明 5 块中哪块被擦除 ；*****用于块的 DPTR 值是： ；　0000h＝块 0

	；　2000h＝块 1 ；　4000h＝块 2 ；　8000h＝块 3 ；　C000h＝块 4 块擦除： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃02H　；程序数据功能 MOV　A，Mydata　；要写的数据 MOV　DPTR，Address　；确认要读的字节 CALL　PGM_MTP　；执行这个功能 RET
擦除引导矢量和状态字节	参数输入： R0 ＝振荡器频率（整型） R1 ＝04h R1 ＝84h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝任意值 参数返回 空 例子： ；*****擦除引导矢量（BV）和状态字节（SB）***** ；*****注：这个命令将 BV 和 SB 都擦除 ERSBBV; MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃02H　；程序数据功能 MOV　A，Mydata　；要写的数据 MOV　DPTR，Address　；确认要读的字节 CALL　PGM_MTP　；执行这个功能 RET
程序安全位	参数输入 R0 ＝振荡器频率（整型） R1 ＝05h R1 ＝85h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝00h－安全位＃1（禁止对 FLASH 写入） 01h－安全位＃2（禁止 FLASH 确认） 02h－安全位＃3（外部存储器无效） 参数返回 空 例子： ；*****程序安全位 1***** ；*****DPTR 表明被编程的安全位***** WRSB1： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃05H　；程序安全位功能 MOV　DPTR，＃0000h　；确认安全位 1

	CALL　PGM_MTP　；执行这个功能 RET ；*****程序安全位 2***** ；*****DPTR 表明被编程的安全位***** WRSB2： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃05H　；程序安全位功能 MOV　DPTR，＃0000h　；确认安全位 1 CALL　PGM_MTP　；执行这个功能 RET ；*****程序安全位 3***** ；*****DPTR 表明被编程的安全位***** WRSB3： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃05H　；程序安全位功能 MOV　DPTR，＃0000h　；确认安全位 1 CALL　PGM_MTP　；执行这个功能 RET
程序状态字节	参数输入： R0 ＝振荡器频率（整型） R1 ＝06h R1 ＝86h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝00h－程序状态字节 ACC ＝程序状态字节 参数返回 ACC＝00 如果通过；非 00 如果失败 例子： ；*****程序状态字节***** ；*****DPTR 表明程序引导矢量***** ；*****ACC 保持引导矢量的新值被编程***** WRBV： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃06H　；程序状态字节或引导矢量 MOV　DPTR，＃0001h　；确认引导矢量 MOV　A，NEW_SB　；引导矢量的新值 CALL　PGM_MTP　；执行这个功能 RET
读器件数据	参数输入 R1＝03h R1=83h(WDT 喂 RX2 和 66X) DPTR＝要读的字节地址 参数返回 ACC＝要读的字节的值 例子： ；*****读器件数据（DData）***** ；***** DData 返回在 ACC 里***** ；*****DPTR 保持要读的字节地址*****

	RDData： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃03H　；读数据功能 MOV　DPTR，地址　；确定要读的字节地址 CALL　PGM_MTP　；执行这个功能 RET
读生产商 ID	参数输入： R0 ＝振荡器频率（整型） R1 ＝00h R1 ＝80h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝00h（生产商 ID） 参数返回 ACC＝要读的字节的值 例子： ；*****读生产商 ID（MID）***** ；***** MID 返回在 ACC 里（应为 15h） RDMID： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃00H　；读 MISC 功能 MOV　DPTR，＃0001H　；确认 MID CALL　PGM_MTP　；执行这个功能 RET
读器件 ID＃1	参数输入： R0 ＝振荡器频率（整型） R1 ＝00h R1 ＝80h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝01h（器件 ID＃1） 参数返回 ACC＝要读的字节的值 例子： ；*****读器件 ID1（DID1）***** ；***** DID1 返回在 ACC 里 RD DID1： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃00H　；读 MISC 功能 MOV　DPTR，＃0001H　；确认器件 ID1 CALL　PGM_MTP　；执行这个功能 RET
读器件 ID＃2	参数输入： R0 ＝振荡器频率（整型） R1 ＝00h R1 ＝80h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝02h（器件 ID＃2） 参数返回 ACC＝要读的字节的值

	例子： ；*****读器件 ID2（DID2）***** ；***** DID2 返回在 ACC 里 RDDID2： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃00H　；读 MISC 功能 MOV　DPTR，＃0002H　；确认器件 ID2 CALL　PGM_MTP　；执行这个功能 RET
读安全位	参数输入： R0 ＝振荡器频率（整型） R1 ＝07h R1 ＝87h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝00h（安全位） 参数返回 ACC＝要读的字节的值 例子： ；*****读安全位（SBits）***** ；***** SBits 返回在 ACC 里 RD SBits： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃07H　；读 MISC 功能 MOV　DPTR，＃0000H　；确认安全位 CALL　PGM_MTP　；执行这个功能 RET
读状态字节	参数输入： R0 ＝振荡器频率（整型） R1 ＝00h R1 ＝80h(WDT 喂养,RX2 和 66X) DPH ＝00h DPL ＝01h（状态字节） 参数返回 ACC＝要读的字节的值 例子： ；*****读状态字节（SB）***** ；***** SB 返回在 ACC 里 RDSB： MOV　AUXR1 ，＃20H　；对 ENBOOT 位置位 MOV　R0，＃11　；FOSC MOV　R1，＃07H　；读 MISC 功能 MOV　DPTR，＃0001H　；确认状态字节 CALL　PGM_MTP　；执行这个功能 RET
读引导矢量	参数输入： R0 ＝振荡器频率（整型） R1 ＝07h R1 ＝87h(WDT 喂养,RX2 和 66X) DPH ＝00h

	DPL ＝02h（引导矢量） 参数返回 ACC＝要读的字节的值 例子： ；*****读引导矢量（BV）***** ；***** BV 返回在 ACC 里 RDBV： MOV　AUXR1 ，＃20H　　；对 ENBOOT 位置位 MOV　R0，＃11　　；FOSC MOV　R1，＃07H　　；读 MISC 功能 MOV　DPTR，＃0002H　　；确认引导矢量 CALL　PGM_MTP　　；执行这个功能 RET

安全位

安全特征保护软件盗版，防止 FLASH 的内容被读出。安全锁定位位于 FLASH。P89C660/662/664 有三个可编程安全锁定位，为片内代码和数据提供不同的保护级别（见表 11）

表 11

安全锁定位				保护描述
级别	LB1	LB2	LB3	
1	0	0	0	MOVC 指令从外部存储器执行是无效的，从内部存储器取代码
2	1	0	0	和级别 1 相同，块擦除是无效的，状态字节或引导矢量的擦除和编程是无效的
3	1	1	0	和级别 2 相同，加上代码存储器的确认是无效的
4	1	1	1	和级别 3 相同，加上外部执行是无效的

注：

1．安全位互相独立。整片内的擦除可被完成，不论安全位的状态如何。

2．任何其他的锁定位连接未定义。

3．LBx 的置位不妨碍未编程位的编程。