1. 前言 2

2. 特色简介 2

3. 内部功能和芯片封装 2

3.1 内部功能方块图 2
3.2 特殊功能寄存器（SFR）一览表 4
3.3 片脚功能图和芯片封装片脚图 5
3.4 片脚定义一览表 6
3.5 片脚定义一览表 7

4. 资源特点及使用编程要点 8

4.1 16 位计数/定时器 T2 8
4.1.1 定时器 2 捕捉模式 9
4.1.2 定时器 2 自动重装模式 10
4.1.3 定时器 2 波特率模式 11
4.1.4 可编程时钟输出模式 12
4.2 可编程计数器阵列（PCA） 12
4.2.1 PCA 捕捉模式 14
4.2.2 PCA 16 位软定时器模式 15
4.2.3 PCA 高速输出模式 15
4.2.4 PCA 的 PWM（脉宽调制）模式 16
4.2.5 PCA 模块 4 看家狗定时器模式 17
4.3 全双工增强型 UART（异步收发器）串行口 18
4.4 7 源 4 优先级嵌套中断系统 22
4.5 数据存储器 22
4.5.1 内部数据存储器 22
4.5.2 外部数据存储器 24
4.6 程序存储器 24
4.6.1 P89C51Rx+上电复位代码的执行 24
4.6.2 P89C51Rx+ 在系统中编程（ISP） 26
4.6.3 P89C51Rx+在运行中编程（IAP） 29
4.7 芯片的特殊工作模式 33
4.7.1 另钟频模式 33
4.7.2 休眠工作模式 33
4.7.3 下电工作模式 33
4.7.4 ONCE 仿真模式 34

5. 订货须知 34

1. 前言

P89C51RC+和 P89C51RD+两种单片机都是 8 位 80C51 单片机的派生产品。它们在完全保留 80C51 指令系统和硬件结构的大框架外，发生了多方面的加强、扩展、翻新、和创新。在最大限度地利用原有的结构的方方面面可以说做到了淋漓尽致，P89C51RD+将原有的对外部数据和程序存储器的 16 位寻址机制加以利用，把片上的 RAM 扩展到 1k 字节、片上的 FLASH EPROM 扩展到 64k 字节，满足当今用嵌入式高级语言对片上大存储器容量的需要。因为 FLASH 存储器的采用，使在 ISP（在系统中编程）乃至 IAP（在运行中编程）等先进技术有了实现的可能，芯片上免费提供 BOOT ROM 固件，并且巧妙地解决了固件和 FLASH 的地址覆盖问题和一些具体实现细节问题，使它们的实现变得简单而现成。

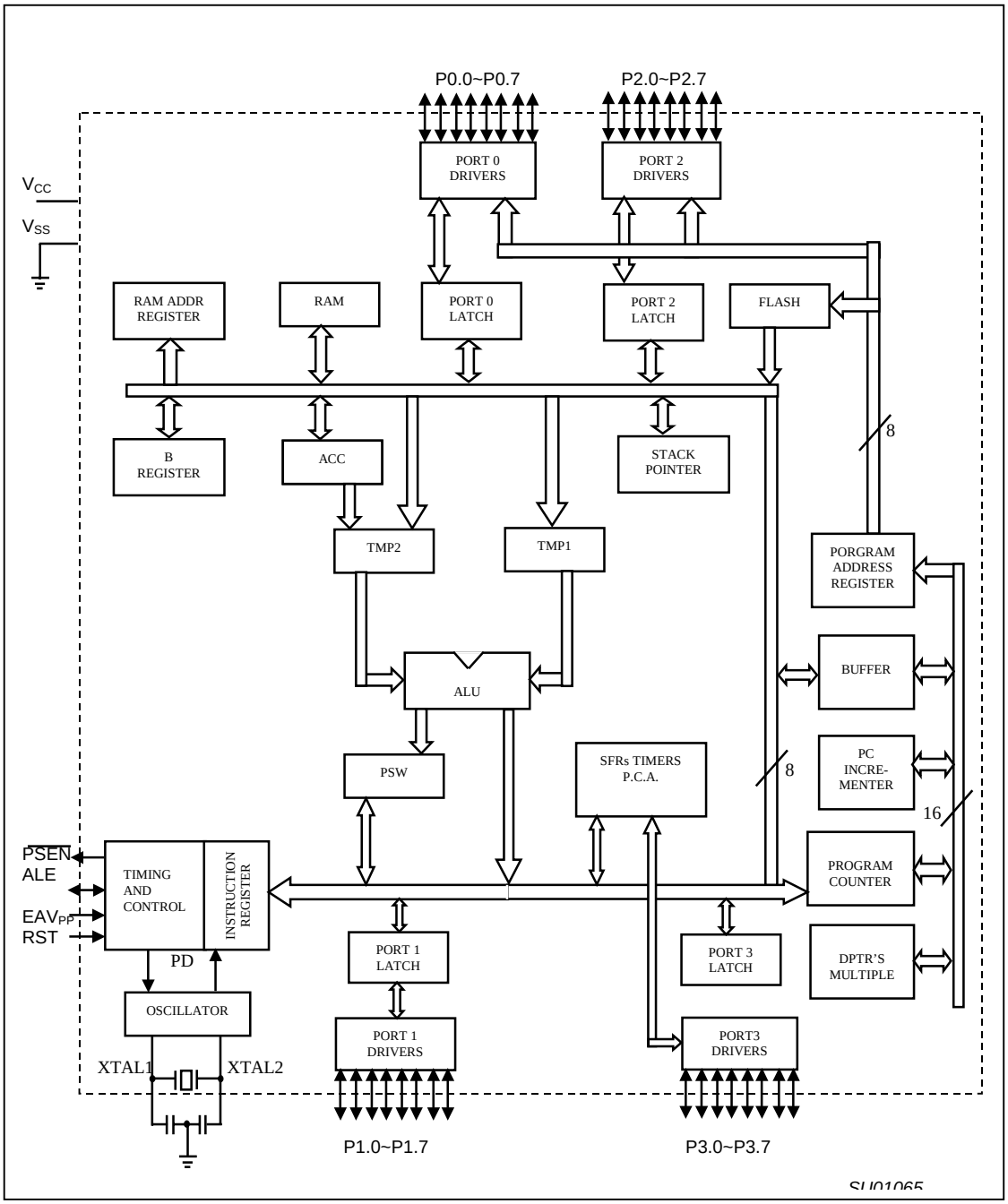
P89C51RC+与 P89C51RD+之间，实际上，仅有存储器容量大小的不同：片上 RAM 前者为 512 字节，后者 1024；片上 FLASH/EPROM 前者为 32K 字节，后者 64K。为了叙述上的简化，下面将二者泛称为 P89C51R**x**+（x=C、D）。又彷此，也将 P89C51RC2 和 P89C51RD2 二者合并称做 P89C51R**x**2（x=C、D）。至于 P89C51Rx+与 P89C51Rx**2** 之间的区别仅在编程电压的不同，前者需要专供 12 伏的编程电压，而后者由于使用 5 伏编程电压，故不需要专门编程电压。为此，下面就专门介绍 P89C51Rx+；除非专门说明，否则暗含它们四者在所讨论的方面是一样的。

2. 特色简介

- 时钟频率 0~33MHz，全静态操作。另有一种芯片钟频为 0~20MHz，但 1 机器周期改为 6 时钟周期（原来为 1 机器周期=12 时钟周期）。
- 双 DPTR 指针。
- 新增加片内 16 位寻址 ERAM（扩展 RAM）。
- 新增加片内程序存储器空间采用非易失 FLASH。同时提供 BOOT ROM 固件，实现 ISP（在系统中编程）和 IAP（在运行中编程）。
- 新增强后四个 I/O 口均有复用功能。
- 增强定时器 2 具有扑捉、自动+1/-1 重装计数器、波特率发生器、编程时钟输出模式。
- 5 模块 PCA。支持上下沿扑捉 PWM 输出、软件定时器、看门狗定时器等模式。
- 的全双工 UART 串行口，新增硬件帧检错和地址自动识别电路。
- 嵌套中断系统增强为 7 源 4 优先级。
- 为降低 EMI 无必要不开放 ALE 脚上的信号。
- 支持 0 钟频模式、空闲模式、下电模式、和 ONCE 仿真模式等。

3. 内部功能和芯片封装

3.1 内部功能方块图


P0.0~P0.7
P2.0~P2.7
PORT 0 DRIVERS
PORT 2 DRIVERS
VCC
VSS
RAM ADDR REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
8
TMP2
TMP1
PORGRAM ADDRESS REGISTER
ALU
BUFFER
PSW
SFRs TIMERS P.C.A.
8
PC INCRE-MENTER
16
PSEN
ALE
EAVPP
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
PROGRAM COUNTER
PD
PORT 1 LATCH
PORT 3 LATCH
DPTR'S MULTIPLE
OSCILLATOR
PORT 1 DRIVERS
PORT3 DRIVERS
XTAL1
XTAL2
P1.0~P1.7
P3.0~P3.7
SU01065

3.2 特殊功能寄存器（SFR）一览表

表 3.2.1 特殊功能寄存器（SFR）一览表

符 号	说 明	直接地址	位地址，符号，或 P 口的功能 MSB							LSB	复位值
ACC*	Accumlator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	—	—	—	—	—	—	EXTRAM	A0	XXXXXX00B
AUXR#[2]	Auxiliary 1	A2H	—	—	ENBOOT	—	GF2	0	—	DPS	XX0X00X0B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH									XXXXXXXXB
CCAP1H#	Module 1 Capture High	FBH									XXXXXXXXB
CCAP2H#	Module 2 Capture High	FCH									XXXXXXXXB
CCAP3H#	Module 3 Capture High	FDH									XXXXXXXXB
CCAP4H#	Module 4 Capture High	FEH									XXXXXXXXB
CCAP0L#	Module 0 Capture Low	EAH									XXXXXXXXB
CCAP1L#	Module 1 Capture Low	EBH									XXXXXXXXB
CCAP2L#	Module 2 Capture Low	ECH									XXXXXXXXB
CCAP3L#	Module 3 Capture Low	EDH									XXXXXXXXB
CCAP4L#	Module 4 Capture Low	EEH									XXXXXXXXB
CCAPM0#	Module 0 Mode	DAH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	X00000000B
CCAPM1#	Module 1 Mode	DBH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	X00000000B
CCAPM2#	Module 2 Mode	DCH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	X00000000B
CCAPM3#	Module 3 Mode	DDH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	X00000000B
CCAPM4#	Module 4 Mode	DEH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	X00000000B
			DF	DE	DD	DC	DB	DA	D9	D8	
CCON*#	PCA Counter Control	D8H	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0	00X00000B
CH#	PCA Counter High	F9H									00H
CL#	PCA Counter Low	E9H									00H
COMD#	PCA Conter Mode	D9H	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF	00XXX000B
DPTR:	Data Pointer(2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt Enable	A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt Priority	B8H	—	PPC	PT2	PS	PT1	PX1	PT0	PX0	X0000000B
			B7	B6	B5	B4	B3	B2	B1	B0	
IPH#	Interrupt Priority High	B7H	—	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	X0000000B
			87	86	85	84	83	82	81	80	
P0*	Port0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port1	90H	CEX4	CET3	CET2	CET1	CET0	EC1	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD8	AD9	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port3	B0H	RD	WR	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TXD	RXD	FFH
PCON#[1]	Power Control	87H	SMOD1	SMOD2	—	POF	GF1	GF0	PD	IDL	00XXX000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status Word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	000000X0B
RACAP2H#	Timer 2 Capture High	CBH									00H
RACAP2L#	Timer 2 Capture Low	CAH									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									XXXXXXXXB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	T1	R1	00H
SP	Stack Pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*		88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
T2MOD#	Timer 2 Mode Control	C9H	—	—	—	—	—	—	T2OE	DCEN	XXXXXX00B
TH1	Timer High 0	8CH									00H

TH1	Timer High 1	8DH									00H
TH2#	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2#	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDTRST	Watchdog Timer Reset	A6H									

* 可位寻址的 SFRs

对比 80C51 的 SFR 有被改动或添加内容

— 保留位

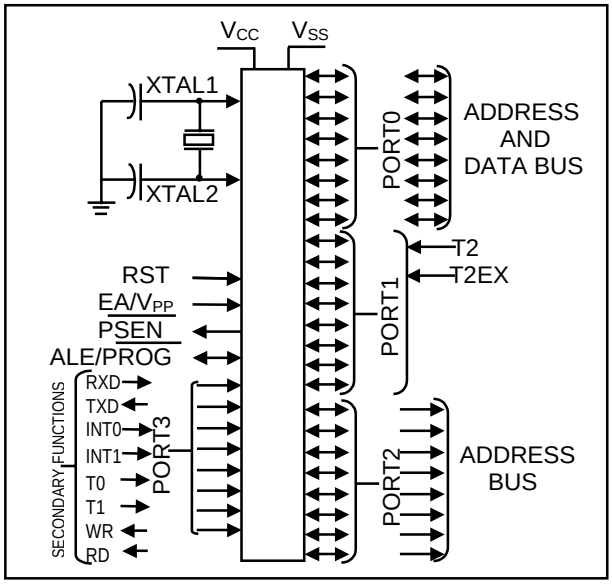
1. 位值依赖于复位源。（拐上电复位或热复位等）

2. ENBOOT 位的状态依赖于 FLASH 寄存器 STATUS BYTE 的内容和 RST 脚上信号转入无效时间的 PSEN 电平。

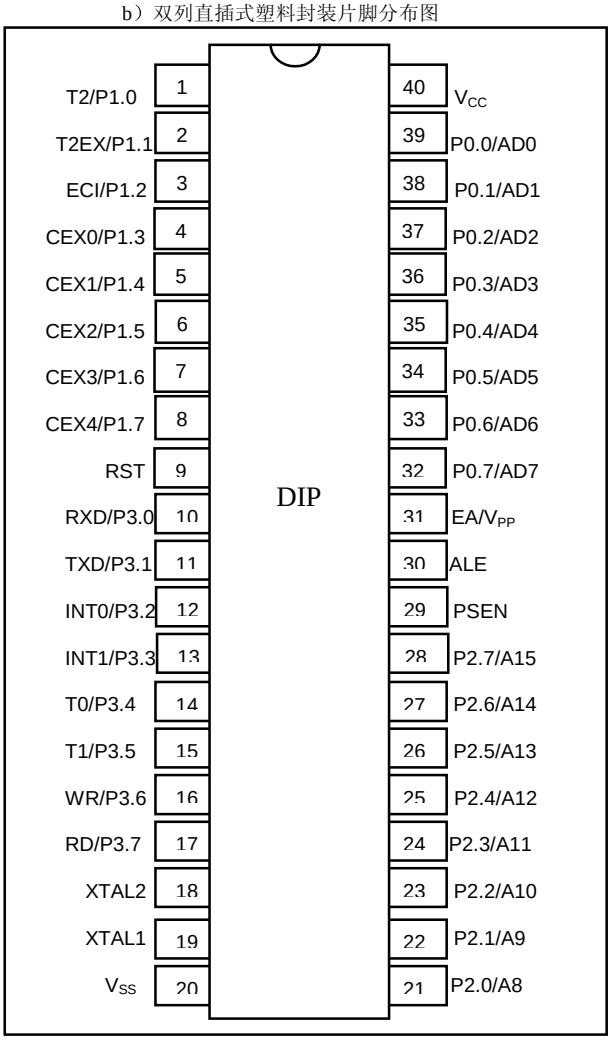
3.3 片脚功能图和芯片封装片脚图

（图 3.3.1 见下页）

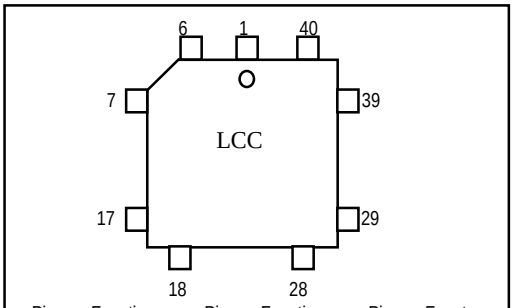
a）片脚逻辑图



b）双列直插式塑料封装片脚分布图



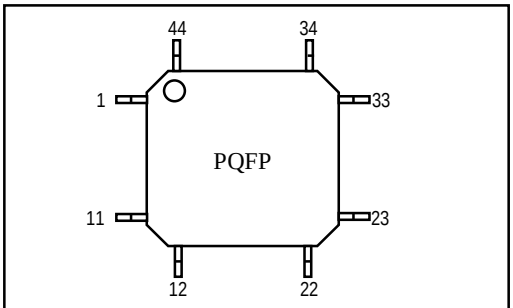
c）双列直插式塑料封装片脚分布图



Pin	Function	Pin	Function	Pin	Functon
1	NIC*	16	P3.4/T0	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1	32	$\overline{PSEN}$
3	P1.1/T2EX	18	P3.6/$\overline{WR}$	33	ALE
4	P1.2/ECI	19	P3.7/$\overline{RD}$	34	NIC*
5	P1.3/CEX0	20	XTAL2	35	$\overline{EA}$/VPP
6	P1.4/CEX1	21	XTAL1	36	P0.7/AD7
7	P1.5/CEX2	22	VSS	37	P0.6/AD6
8	P1.6/CEX3	23	NIC*	38	P0.5/AD5
9	P1.7/CEX4	24	P2.0/A8	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/$\overline{TXD}$	28	P2.4/A12	43	P0.0/AD0
14	P3.2/INT0	29	P2.5/A13	44	VCC
15	P3.3/INT1	30	P2.6/A14		

*内部无连接 *SU00831*

d）扁平式塑料封装片脚分布图



Pin	Function	Pin	Function	Pin	Functon
1	P1.5/CEX2	16	VSS	31	P0.6/AD6
2	P1.6/CEX3	17	NIC*	32	P0.5/AD5
3	P1.7/CEX4	18	P2.0/A8	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/INT0	23	P2.5/A13	38	VCC
9	P3.3/INT1	24	P2.6/A14	39	NIC*
10	P3.4/T0	25	P2.7/A15	40	P1.0/T2
11	P3.5/T1	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/VPP	44	P1.4/CEX1
15	XTAL1	30	P0.7/AD7		

*内部无连接 *SU00831*

3.5 片脚定义一览表

表 3.4.1 管脚定义一览表

助记符	片腿号			类型	名字与功能
	PDIP	PLCC	PQFP		
Vss	20	22	16	I	地：0 伏，参考零。
Vcc	40	44	36	I	电源：正常、休闲、下电等模式的电源电压。
P0.0-0.7	39-32	43-36	37-30	I/O	口 0：口 0 是漏极开路双向 I/O 口。向口 0 的腿写 0，腿即浮空，可用做高阻抗输入腿。在向片外程序和数据存储器读写时，口 0 是低字节地址总线和数据总线复用口。这时，若向外输出 1 态使用的是内部强上拉。
P1.0-1.7	1-8	2-9	40-44,1-3	I/O	口 1：口 1 是 8 位双向 I/O 口，除 P1.6 和 P1.7 腿为漏极开路外，各腿均有内部上拉。向有内部上拉的各腿写 1 后，可用做输入腿。做输入腿用时，遇口 1 腿被外部拉向低电平，将经由内部上拉向外供电流，即 DC 电气特性表中的 IIL。 对于单片机 89C51RB+/RC+/RD+和 89C51RB2/RC2/RD2，口 1 腿还有如下的复用功能：
	1	2	40	I	T2(P1.0): T2 计数器模式的外部计数输入腿或 T2 可编程时钟模式的输出腿。
	2	3	41	I	T2EX(P1.1): 用于 T2 的自动重装/扑捉/计数器方向控制。
	3	4	42	I/O	ECI(P1.2): PCA 的外部时钟输入腿。
	4	5	43	I/O	CEX0(P1.3): PCA 模块 0 扑捉/比较模式的外部 I/O 腿。
	5	6	44	I/O	CEX1(P1.4): PCA 模块 1 扑捉/比较模式的外部 I/O 腿。
	6	7	1	I/O	CEX2(P1.5): PCA 模块 2 扑捉/比较模式的外部 I/O 腿。
	7	8	2	I/O	CEX3(P1.6): PCA 模块 3 扑捉/比较模式的外部 I/O 腿。
	8	9	3	I/O	CEX4(P1.7): PCA 模块 4 扑捉/比较模式的外部 I/O 腿。
P2.0-2.7	21-28	24-31	18-25	I/O	口 2：口 2 是 8 位双向 I/O 口，各腿均有内部上拉。向有内部上拉的各腿写 1 后，可用做输入腿。做输入腿用时，遇口 2 腿被外部拉向低电平，将经由内部上拉向外供电流，即 DC 电气特性表中的 IIL。在向片外程序取指令和使用 MOVX @DPTR 指令进行外部数据存储器 16 位寻址时，口 2 输出地址总线的高字节。这时，若向外输出 1 态使用的是内部强上拉。当使用 MOV @Ri 指令进行外部数据存储器 8 位寻址时，口 2 输出特殊功能寄存器 P2 的内容。
P3.0-3.7	10-17	11,13-19	5,7-13	I/O	口 3：口 3 是 8 位双向 I/O 口，各腿均有内部上拉。向有内部上拉的各腿写 1 后，可用做输入腿。做输入腿用时，遇口 1 腿被外部拉向低电平，将经由内部上拉向外供电流，即 DC 电气特性表中的 IIL。 对于单片机 89C51RB+/RC+/RD+和 89C51RB2/RC2/RD2，口 3 腿还有如下的复用功能：
	10	11	5	I	RXD(P3.0): 串行口输入腿。
	11	13	7	O	TXD(P3.1): 串行口输出腿。
	12	14	8	I	*INT0(P3.2): 外部中断 0 输入腿。
	13	15	9	I	*INT1(P3.3): 外部中断 1 输入腿。
	14	16	10	I	T0(P3.4): 定时器 T0 的外部输入腿。
	15	17	11	I	T1(P3.5): 定时器 T1 的外部输入腿。
	16	18	12	O	*WR(P3.6): 外部数据存储器的写选通信号输出腿。
	17	19	13	O	*RD(P3.7): 外部数据存储器的读选通信号输出腿。
RST	9	10	4	I	复位腿：在系统振荡器稳定振荡的前提下，加于本腿的高电平持续 2 个机器周期，本单片机进行内部复位。内部已有接向 Vss 的电阻，仅在本腿与 Vcc 之间加一只电容，就可以实现上电复位。
ALE	30	33	27	O	地址锁存使能腿：向片外程序和数据存储器读写时，本腿发出锁存脉冲，将地址的低字节进行锁存。正常时，每机器周期本腿发出 2 次 ALE 脉冲，它可以对外用做定时或时钟信号。应该指出，对于外部读写操作，每机器周期只发出 1 次 ALE 脉冲，另一次跳过而不发出。又 ALE 脉冲还可以用程序禁止其发出。将 SFR 的 AUXR 的 0 位置位，一般地说禁止 ALE 的发出，仅在执行指令 MOVX 和 MOVC 时发一个 ALE 脉冲。AUXR 的 0 位被清零，禁止解除，每机器周期发出 2 次 ALE。
*PSEN	29	32	26	I	使能程序存储器腿：本腿发出程序存储器读选通信号。当去外

					部程序存储器中取代码时，*PSEN 每机器周期发出 2 次。但是，当去外部数据存储器读写时，2 次*PSEN 都会跳过不发。去内部程序存储器中取代码时，*PSEN 也不会发生。
*EA/Vpp	31	35	29	I	外部存储使能 / 编程电源腿：本腿被外部电路拉到低电平时，本应从内部程序存储器中取指令被迫改到从外部程序存储器取指令。本腿保持高电平，恢复正常。本腿上的电平状态在 RST 腿复位完成时被锁存，并且以后保持不变。本腿在编程时，用于输入编程电压。
XTAL1	19	21	15	I	晶体 1 腿：振荡反向放大器输入端，通向内部时钟电路。
XTAL2	18	20	14	O	晶体 2 腿：振荡反向放大器输出端。

注 1，表中“*”号表示负有效信号。

2，为避免上电时发生“锁存”现象，每一条腿上的电压不得高于 Vcc+0.5 伏低于 Vss-0.5 伏。

4. 资源特点及使用编程要点

4.1 16 位计数/定时器 T2

P89C51Rx+具有与标准 80C51 一样的定时器 0 和定时器 1，此外，还有定时器 2。关于定时器 0 和定时器 1 请见标准 80C51 的技术资料。现在介绍定时器 2。与定时器 2 有关的 SFR 有两个：T2CON 和 T2MOD（见图 4.1.2）。定时器 2 有四种模式：捕捉、自动重装、波特率、和可编程时钟输出模式。下面分别介绍。

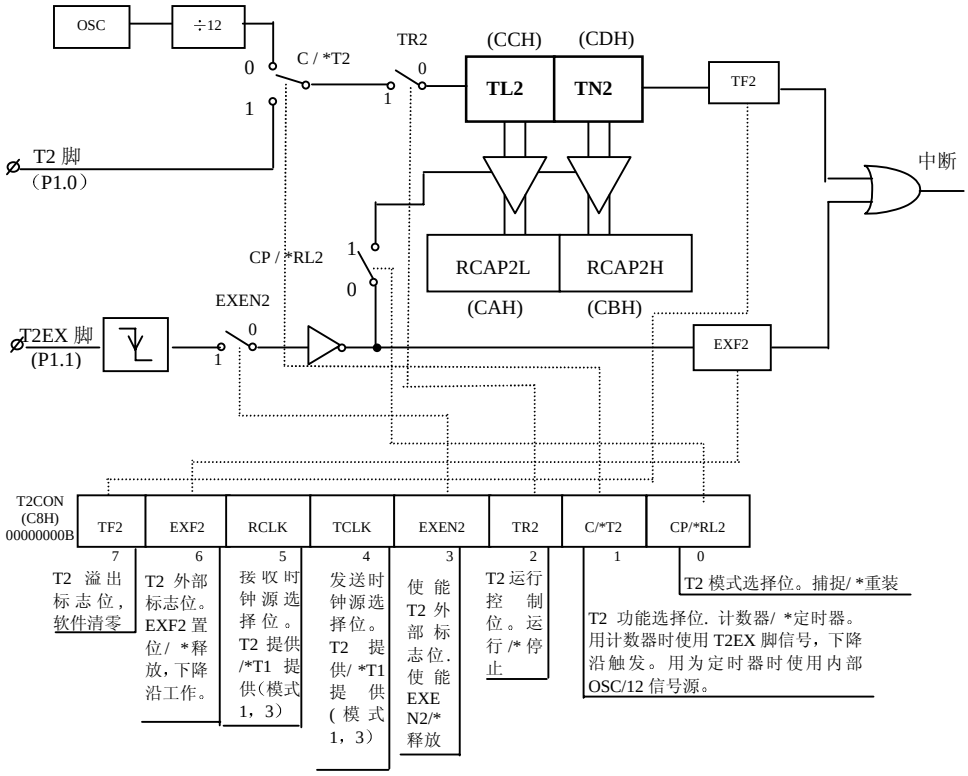


图 4.1.1 定时器 2 捕捉模式

4.1.1 定时器 2 捕捉模式

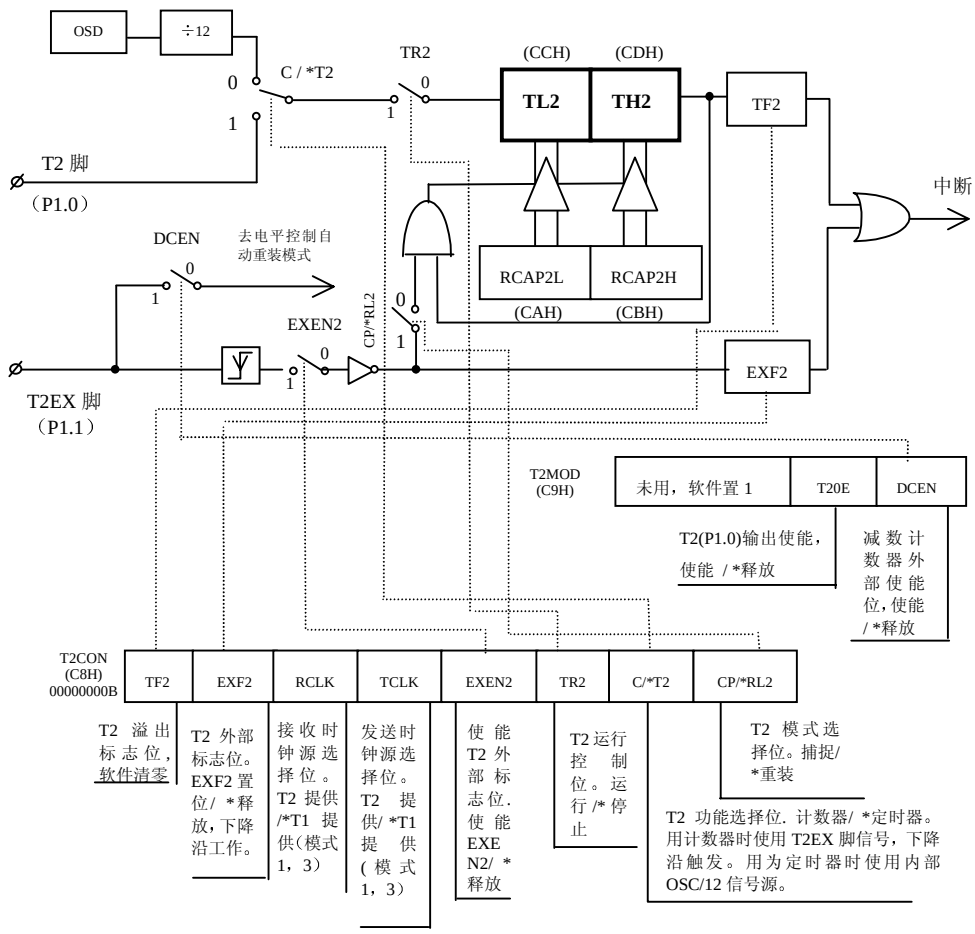


图 4.1.2 定时器 2 自动重装模式之一（外部下降沿控制）

分两步来理解捕捉模式，先看定时器 2 工作于一般的计数器/定时器模式，再看捕捉模式。请见图 4.1.1 的上半部，从 T2 腿上外加计数脉冲，如果软件开关 C/*T2 和 TR2 均处在 1 态，则 T2 工作于计数器模式。在计数器 TL2、TH2 计数值达到 FFFFH 时发生溢出，将中断标志 TF2 置位，并提出中断。事先向 TL2、TH2 置入初值，等到中断发生时便知其间 T2 腿上输入的数脉冲数为（0FFFFH--初值）。这是计数器的用法。定时器模式，是用内部振荡器作为计数脉冲，由于内部振荡器的周期（T=1/ fosc）已知，由计数值可求出定时值：（0FFFFH--初值）/ fosc。这是定时器的用法。现在看 T2 的捕捉模式。捕捉模式的关键是应从片腿 EXT2 上加一个脉冲信号，此信号的下降沿使 T2 与 RCAP2 之间的控制门导通，将 T2 当时的计数值记录到 RCAP2 中（即将 TL2、TH2 的计数值分别记录于 RCAP2L 和 RCAP2H 之中），这就是所谓的。与捕捉的同时，还将中断标志 EXF2 置位（不是 TF2！）并可提出中断申请。捕捉模式，可以灵活扩展出很多用途。配合 T2 的定时器模式，可以测量 EXT2 腿上脉冲信号的宽度；配合 T2 的计数器模式，可以比较 EXT2 和 T2 腿上两路脉冲信号的频率等等。

图中同时给出与本模式有关的 SFR 和各位的定义，同时给出了有关的示意电路图。将电路图与相关的 SFR 展现与一张图上，互相印证，应用目标明确，由于软硬件的结合，很容易一看自明，易于理解。这张图的另一个重要价值，在于辅助你轻松的编写片上资源的初始化程序。对于当前讨

论的具体内容，它几乎是全息的，无须再另外查找资料。而且，图示思路明确，据此编写的初始化程序，既不会遗漏又不致多写，避免遗漏和错误的产生。

4.1.2 定时器 2 自动重装模式

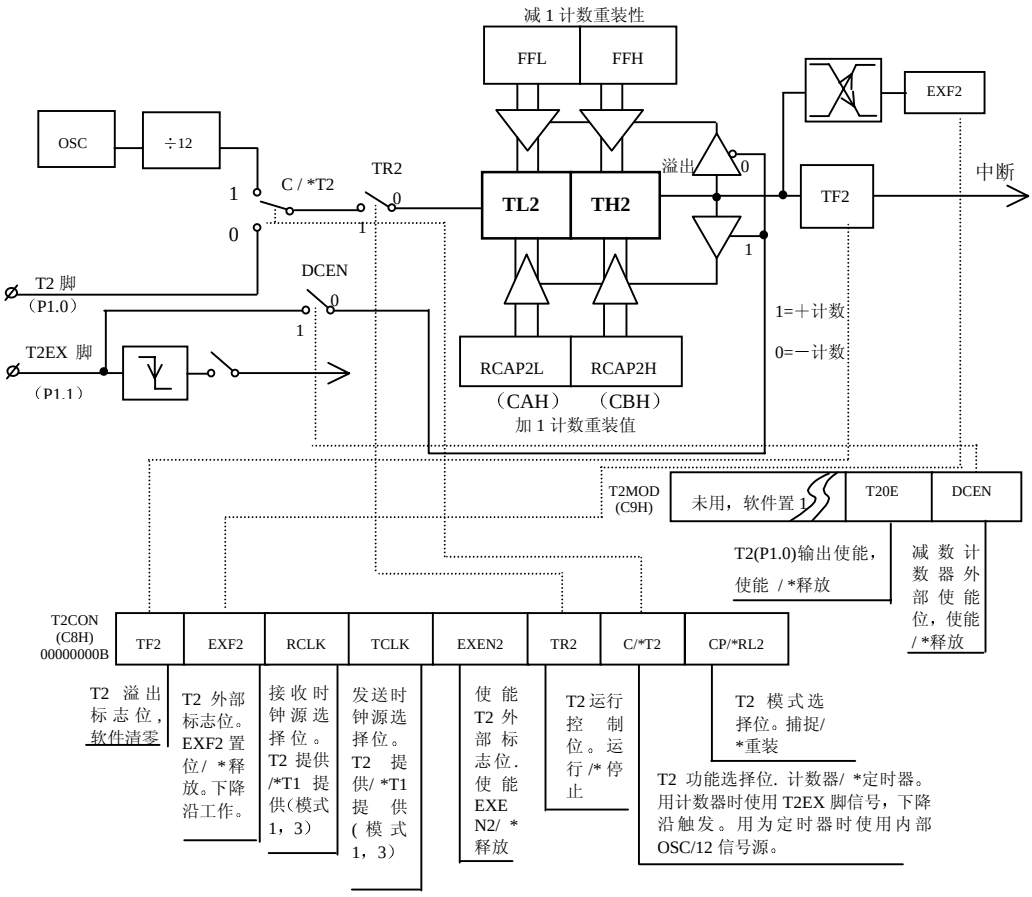


图 4.1.3 定时器 2 自动重装模式之二（外部电平控制加减计数）

应当清楚，自动重装模式的基础还是计数器或定时器模式。所谓自动重装，就是预先将 T2 的初值装入 RCAP2 中，当某些条件出现时，可自动将 RCAP2 中的初值重新装到 T2 中去。共有三种条件出现时需要自动重装：1，T2 溢出；2，EXT2 腿的信号出现下降沿；3，EXT2 腿上的信号出现高电平或低电平时。之所以提供这么多种可能重装的条件，是为了满足用户灵活使用的方便。图 4.1.2 是 T2 溢出或 EXT2 腿信号出现下降沿，经或门实现的重装。标志位 TF2 和 EXF2 可用于区分引起重装的原因。图 4.1.3 是最后一种的电平触发重装的模式。芯片还巧妙地利用了高、低电平分别去触发+1 计数器和-1 计数器。图 4.1.2 和图 4.1.3，从编程上来说，最重要的区别是软开关 DCEN 的状态。 图 4.1.2 的 DCEN 置为 0 态，使用 EXT2 腿引入信号的下降沿触发；图 4.1.3 的 DCEN 置为 1 态，使用 EXT2 腿引入信号的电平触发。T2 工作在+1 计数器时，在计数值达到 0ffffH 发生上溢出时，用 RCAP2 的值进行重装；T2 工作在-1 计数器时，重装值固定用 0ffffH，当减到与 RCAP2 的予装值相等时，实行所谓的下溢出。溢出时都将标志 TF2 置位，并提出中断申请。值得一提的是，外部标志位 EXF2 用法有点特别，它接在溢出线上，随着溢出的出现，它的电平在 0、1 之间翻动,但由于这时的 EXF2 是孤立的，不会去申请中断。

4.1.3 定时器 2 波特率模式

波特率发生器主要是用于串行口的收、发信。大家知道在标准 80C51 中，已有定时器 1 可用于产生波特率供给串行口的收、发信。在 P89C51Rx+中又多提供了一个波特率发生源，其目的是使用户对于收信和发信具有分别采用不同波特率的灵活性，见图 4.1.4 的左上部。波特率源的选择是通过软件向特殊功能寄存器 T2CON 中的 RCLK、TCLK 写 1 或 0 来进行的。

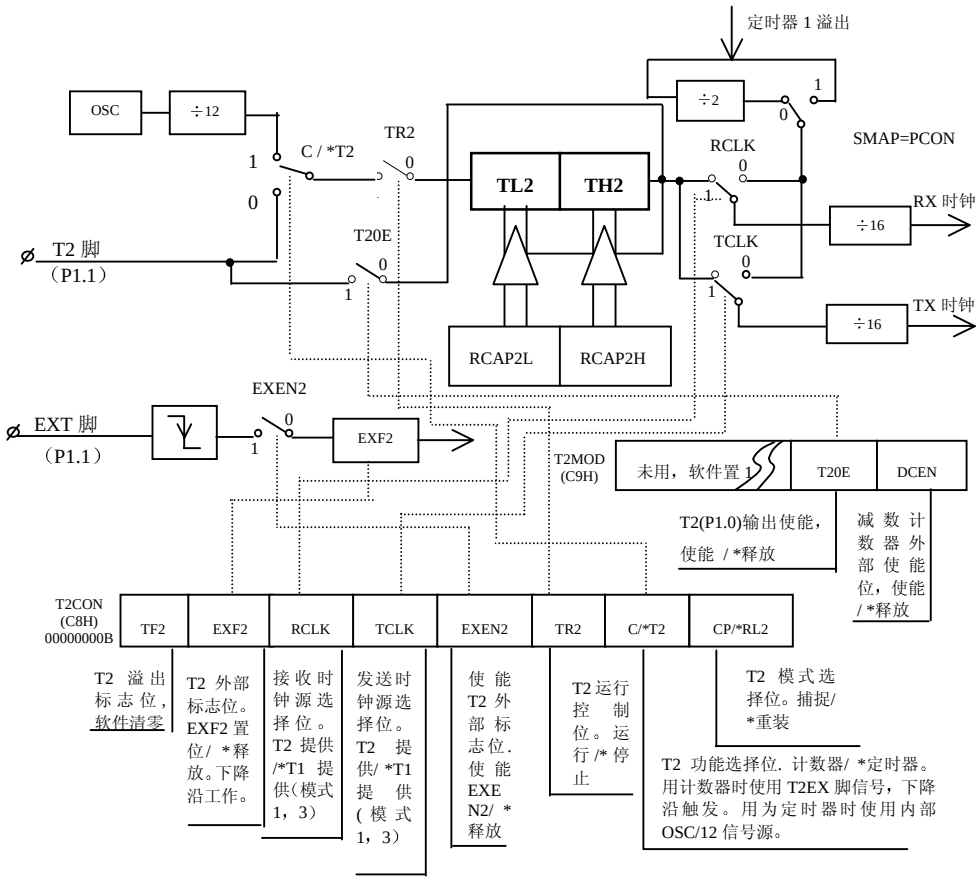


图 4.1.4　定时器 2 波特率发生器模式及时钟输出模式

定时器 2 的波特率模式本质上仍是自动重装模式，只不过输出仅供串行口使用，而且不触发中断。工作时应先通过 RCLK、TCLK 确定本波特率发生器是用于收信还是发信，然后将波特率常数用软件写入 RCP2L 和 RCP2H，随后用 C/*T2 选择外部或者内部脉冲源，最后，将 TR2 置 1，启动定时器 2 开始工作。

波特率常数的求算公式如下：

波特率＝定时器 2 溢出率 / 16

因采用内部或者外部脉冲源而有下列进一步的关系：

内部脉冲源

波特率＝ $fosc/[32x[65566-(RCAP2H,RCAP2L)]]$

$(RCAP2H, RCAP2L) = 65536 - (\frac{fosc}{32x波特率})$

其中：*fosc* 为振荡器频率

外部脉冲源

波特率＝外部脉冲源重复率(EXT2 腿) / 16[65536-(RCAP2H,RCAP2L)]

(RCAP2H,RCAP2L)＝65536－外部脉冲源重复率 / 16x 波特率

切记，定时器 2 用做波特率模式时，不要再对 TH2 和 TL2 进行读写，因为其脉冲源无论用的是外部的或者内部的，其计数频率与机器的读写钟频是不同步的，读写的结果不会准确。对于 RCAP2H 和 RCAP2L 也不要读写，以免干扰自动重装，造成错误。若一定要读写，只有先把 TR2 清另，将定时器 2 退出运行，再行读写。

4.1.4 可编程时钟输出模式

可编程时钟输出模式是由定时器 2 波特率模式派生而来，它把波特率的输出端另引一路到 T2 腿(P1.0)，向片外输出占空比为 50%时钟信号；这一路时钟输出是受控于 T2MOD 的 T2OE 位，仍见图 4.1.4。输出时钟的频率和计数值按下面的公式计算：

时钟频率＝ $fosc/[4x[65566-(RCAP2H,RCAP2L)]]$

$(RCAP2H, RCAP2L) = 65536 - (\frac{fosc}{4x波特率})$

对于 16Mhzde 晶振，可输出时钟的频率为 61Hz~4MHz.。由图 4.1.4 也可以看出 T2 的时钟输出模式是不会引发中断的。应该指出，T2 的时钟输出模式与波特率模式是可以同台共现的，但条件是波特率和时钟频率只能取相同的值。

4.2 可编程计数器阵列（PCA）

P89C51Rx+的可编程计数器阵列是由五个一样的、以计数器为主的模块组成，每个模块除为主的计数器外，还辅之以比较器 / 沿捕捉器。每个模块都可以经编程实现下列多种模式：1，捕捉模式，2，软定时器模式，3，高速输出模式，4，PWM（脉宽调制）模式；另外，模块 4（即第五个模块）还单独多有一个看家狗模式。先介绍各模块的计数器部分、各计数器溢出后的中断部分、以及各个模块公用的计数源部分，然后再介绍模块的各种工作模式。

图 4.2.1 的中部是各模块的计数器和比较器部分、右侧是各计数器溢出后的中断部分和各比较器产生的中断部分。各模块的计数器溢出共用一个标志 CF 和一个中断使能位 ECF，见图 4.2.1；各模块的比较器有自己的输出标志 CF0~CF4 和自己的中断使能位 ECCF0~ECCF4。图 4.2.1 左侧是公用的

计数源部分，共有四个可能的：1，fosc/16（振荡器 16 分频）；2, fosc/4（振荡器 4 分频）；3, 定时器 0 溢出；4，由 ECI 腿（P1.2）输入的外部计数源；它们为每个模块的定时器所共用。

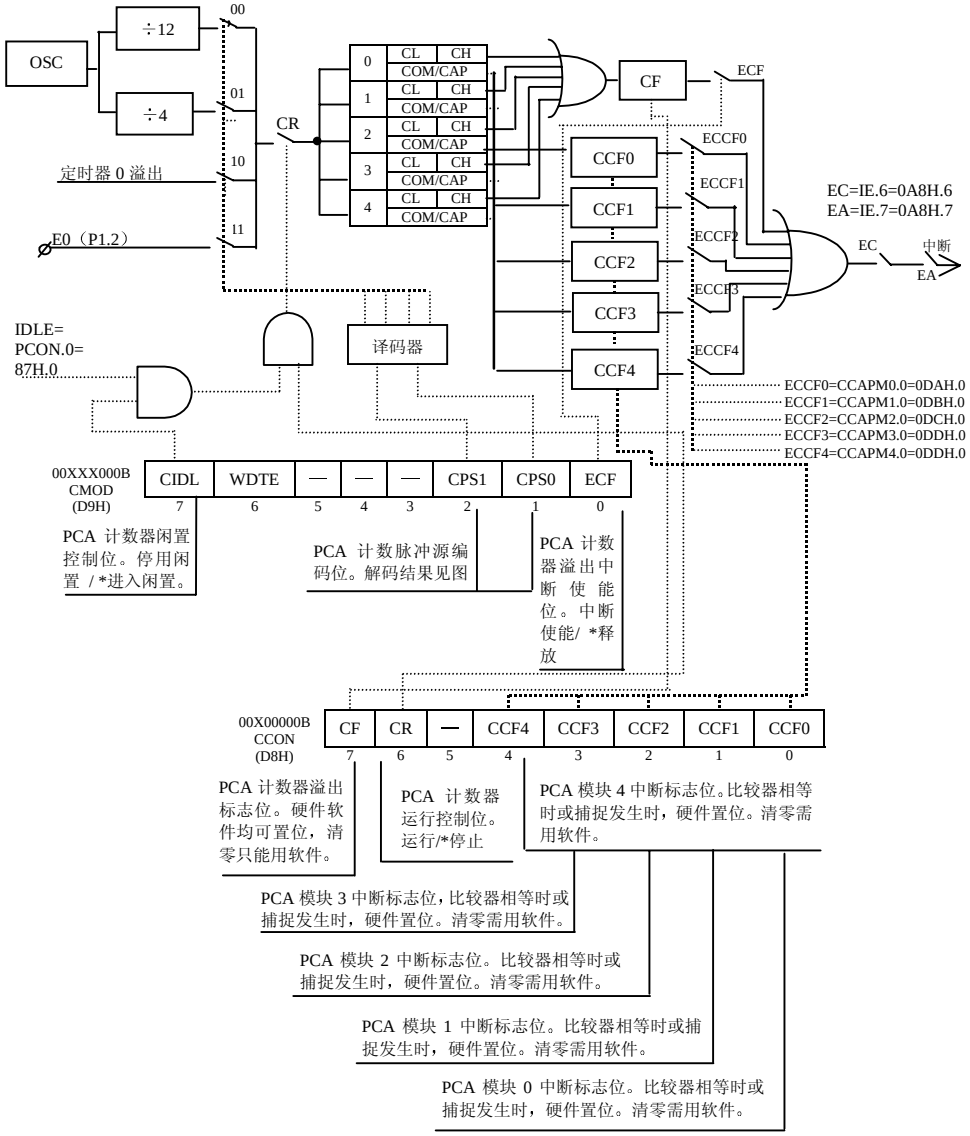


图 4.2.1　计数器/定时器和中断部分

与上述三部分有关的 SFR 有两个：CMOD 和 CCON，它们的说明见图 4.2.1 的下部。上面提到的 CF 和 ECF 就是 CMOD 中的两位。上面提到的四个可能的定时器源也是由 CMOD 的 CPS1、CPS0 两位经编码选定的，各模块定时器的总体启动由 CMOD 的 CR 控制，但应注意，CR 还受芯片 IDLE 模式的牵制，若芯片工作于 IDLE 模式时希望 PCA 模块均停止工作，则应将 CMOD 的 CIDL 置 1。各模块的定时器溢出都经或门打出共用的标志，CCON 的 CF。用 CF 去申请中断（定时器溢出中断），应将 CMOD 中的 ECF 位使能。各个模块附属的比较器/捕捉器，在发生输出时，分别打出 CCON 中的标志 CCF0~CCF4；而与各标志相应的中断使能位，ECCF0~ECCF4，则分散在对应模块的模式寄存器 CCAPM0~CCAPM4 中（见图 4.2.3）。

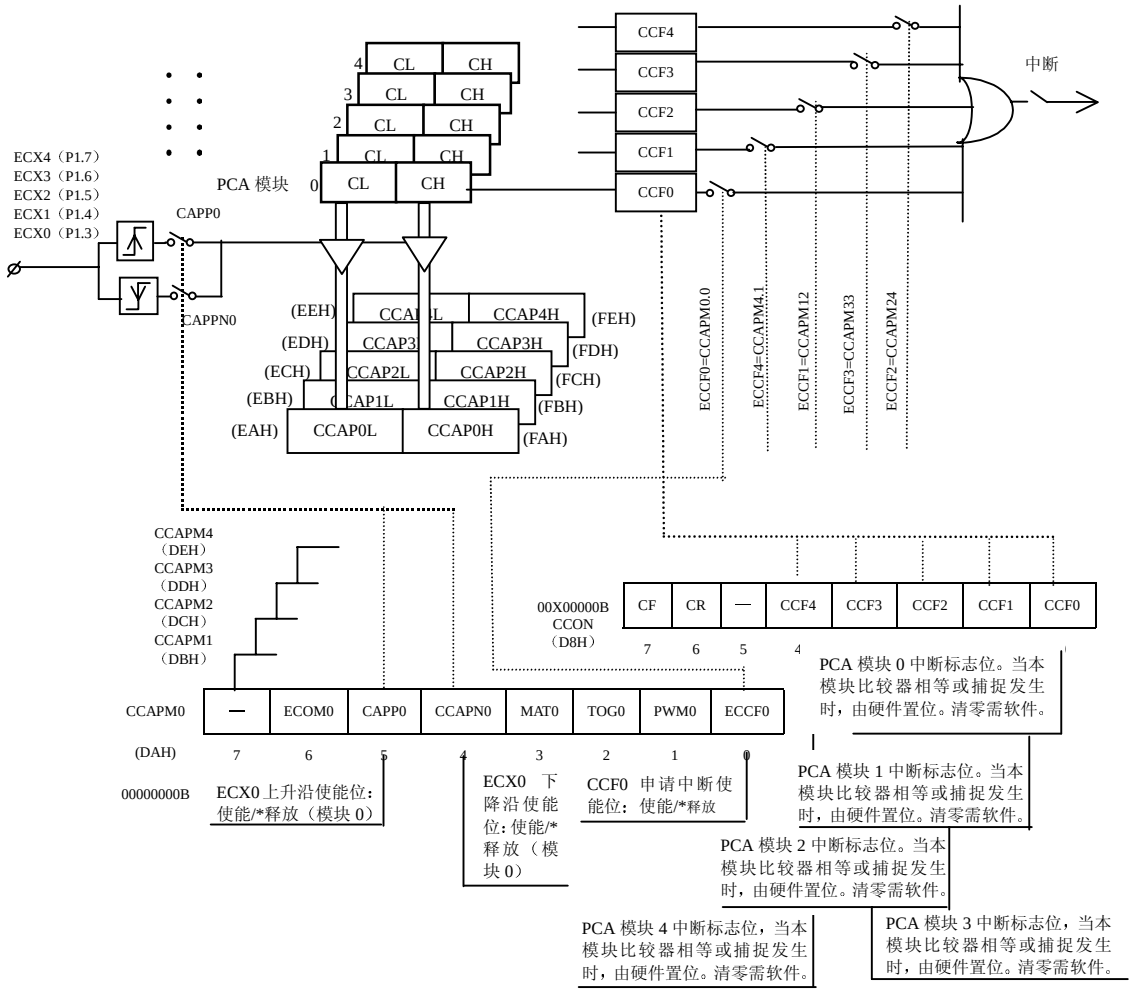


图 4.2.2　PCA 捕捉模式

4.2.1　PCA 捕捉模式

图 4.2.2 主要显示模块 0 的捕捉模式，但同时也标出了其它模块的相应内容。这一部分涉及到的 SFR，除去 CCON 之外，还有 15 个 SFR，即模块 0~4 的模式寄存器（CCAPM0~CCAPM4）、模块 0~4 的捕捉寄存器高字节（CCAP0H~CCAP4H）、模块 0~4 的捕捉寄存器低字节（CCAP0L~CCAP4L）。 这一部分涉及到 5 个输入腿，CEX0~CEX4，分属模块 0~4。这 5 个输入腿是口 P1.3~P1.7 的复用功能。对于模块 0，从腿 CEX0 输入的信号当出现上升或下降沿时，打开控制门将 CL 和 CH 的内容打入 CCAP0L 和 CCAP0H 记录下来，与此同时置 CCF0 标志，如果 ECCF0 已经使能则去申请中断。至于上升还是下降沿捕捉，则通过对 0 模式寄存器 CCAPM0 的 CAPP0 和 CCAPN0 位置 1 或置 0 来控制；如 CAPP0 置 1，同时 CCAPN0 置 0，为上升沿捕捉。

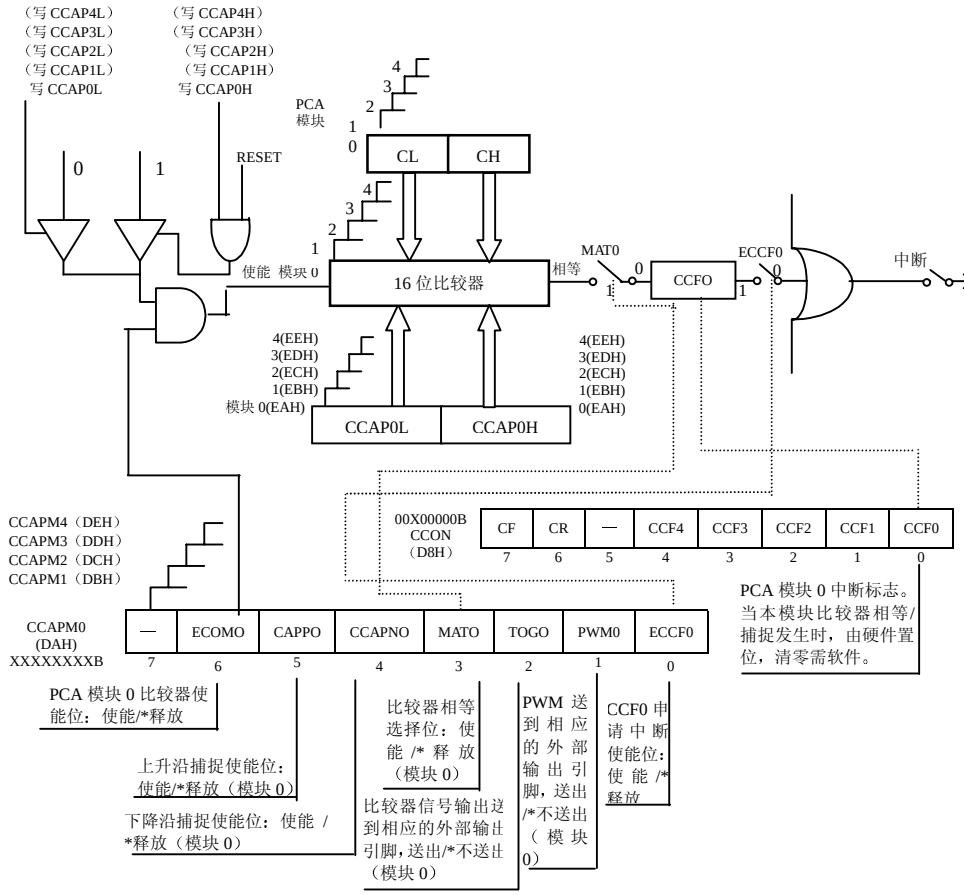


图 4.2.3　PCA 定时器模式

4.2.2　PCA　16 位软定时器模式

通过编程预先设定一个时间，到达设定时间则产生标志或中断，这就叫做软定时器。它不象硬件定时器一定要等到定时器溢出才产生标志或中断，而是软件设定的时间一到就产生标志或中断。操作相对简单而快捷，但需要比较器的帮助。图 4.2.3 是选定模块 0 的软定时器模式。预先设定的时间先装入 CCAP0L 和 CCAP0H，再将比较器的多种输出条件中选定为比值相等（MAT0=1），然后使能比较器（ECOM0=1），比较开始，当计数器 CH、CL 的值达到与 CCAP0H、CCAP0L 中的值时产生标志 CCF0，并在 CCAPM0 的 ECCF0 位=1 时去中断。

4.2.3　PCA 高速输出模式

图 4.2.4 是模块 0 的高速输出模式。它与软定时器相似，唯一的不同是比较器输出端另有一路经 T 触发器（Toggle Flip-flop，乒乓触发器）接向片腿 CEX0，将软定时器产生的高速数字信号输出。

每次定时到，在中断服务程序中要为下一段输出信号进行编程。编程时，先将所需的定时装入捕捉

寄存器（如 CCAP0H、CCAP0L），再将比较器的输出条件设定为比值相等（如 MAT0=1），再将 Toggle 使能位置 1（如 TOG0=1），然后使能比较器（如 ECOM0=1），比较开始，当计数器 CH、CL 的值达到与 CCAP0L 和 CCAP0H 中的值相等时则产生标志 CCF0，在 ECCF0=1 时去中断；与

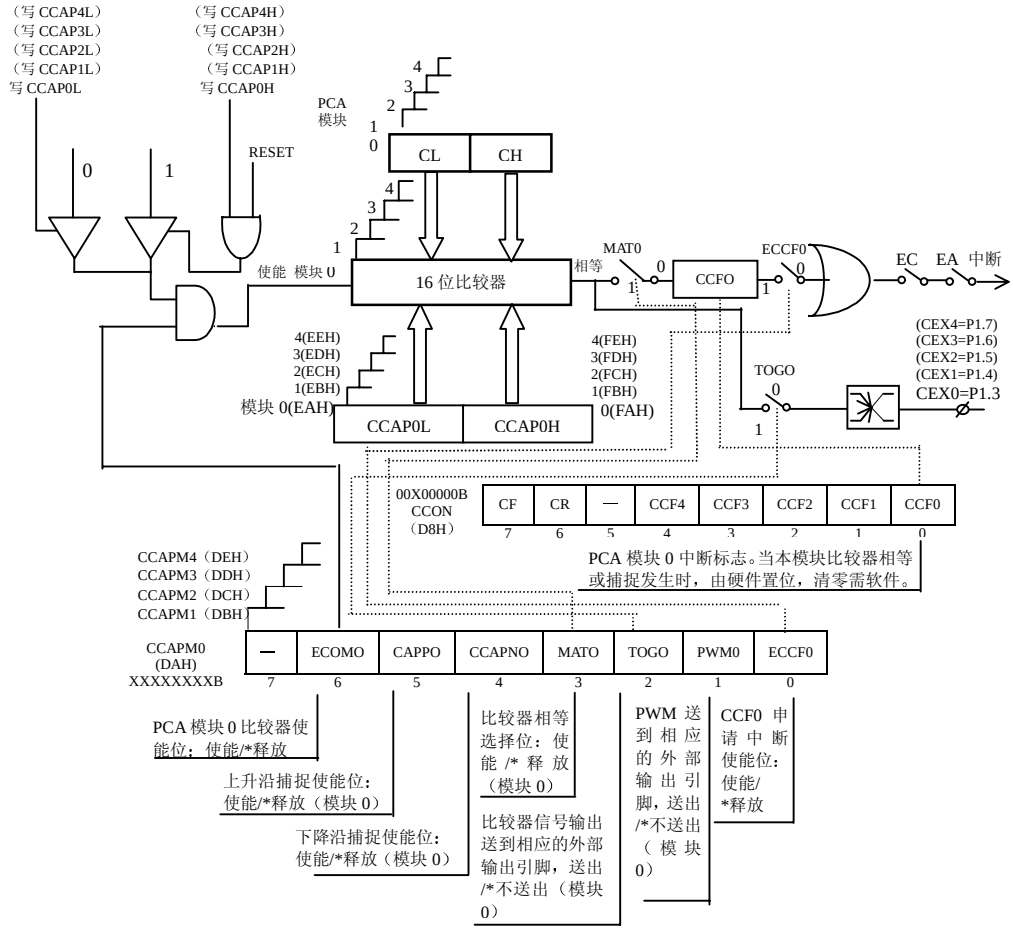


图 4.2.4　PCA 高速传输出模式

此同时，另一路输向片腿 CEX0，在 TOG0=1 的情况下将原电平翻转，准备好下一段信号的输出。

4.2.4　PCA 的 PWM（脉宽调制）模式

脉宽调制就是将具有一定重复频率的脉冲宽度（可用高电平宽度或低电平宽度），受某值大小的调制，此宽度占脉冲周期的百分比叫做占空比。所以受某值大小调制后的脉冲，其波形占空比也随着改变。图 4.2.5 是模块 0 的脉宽调制模式。图中的 8 位比较器将予置于 8 位捕捉寄存器低字节（如 CCAP0L）中的某值，与改变着的 8 位计数器值相比较，比较结果：CL<CCAPL 的某值输出低电平到片腿（如 CEX0），当 CL>=CCAP0L 时输出高电平到片腿。直到 CL 溢出，随后将捕捉寄存器高字节（如 CCAP0H）中的重装值再度装入 CCAP0L；不断重复上述周期。8 位计数器计数计数到 255 就溢出，随后重装。所以，片腿输出的 PWM 波形的周期是：256*CL 计数源的周期；而占空比是：CCAP0L 的予装值 / 256。应当指出，PCA 各模块输出 PWM 的重复频率是一样的，因为它们共用同一计数源；但是，各模块输出的占空比可以不同。

编程时，先将欲调制的某值装入捕捉寄存器的低字节和高字节（如 CCAP0L 和 CCAP0H），再将 PCA 模式寄存器（如 CCAPM0）中的 PWM 控制位置 1（如 PWM0=1），再将其模式寄存器

的比较器使能位置 1（如 ECOM0=1），便开始比较，当计数器 CL 的值达到与 CCAP0L 值时，片腿的电平由低转高，直到 CL 溢出，随后的过程就是自动重复的了。

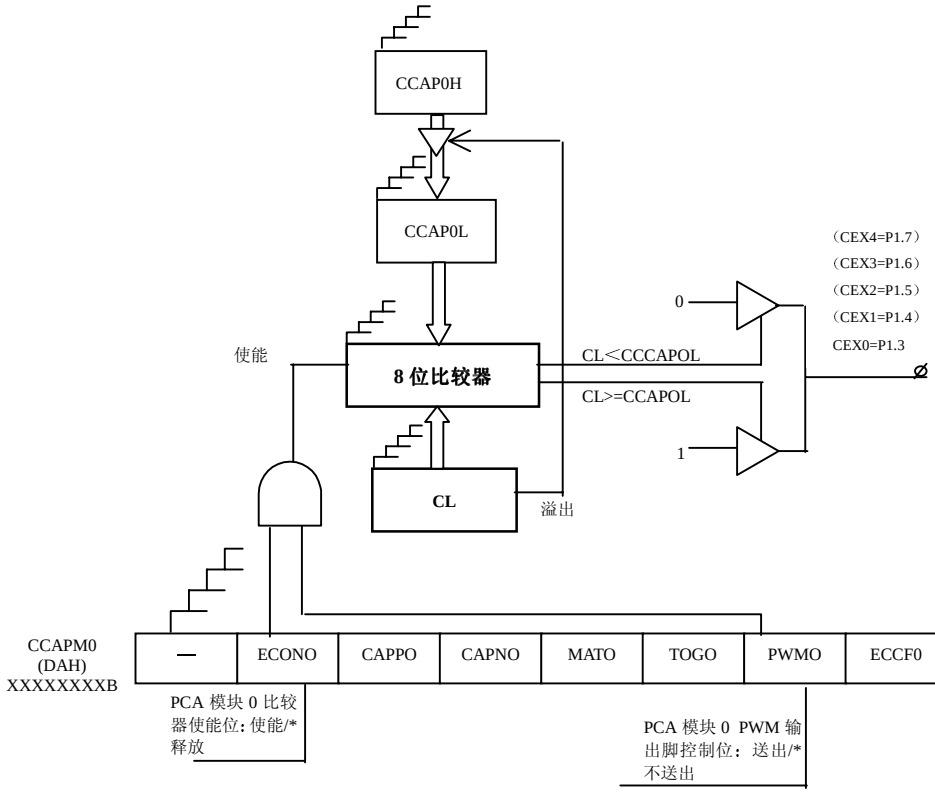


图 4.2.5　PCA PMW 模式

4.2.5　PCA 模块 4 看家狗定时器模式

在遭受大噪音、电源毛刺或静电放电等的干扰，造成芯片软件执行出轨，看家狗定时器有助于恢复软件的正常执行，对提高系统的可靠性有好处。PCA 的模块 4 提供了看家狗定时器模式。图 4.2.6 是模块 4 的看家狗定时器模式。图中的 16 位比较器将予置于 16 位捕捉寄存器 CCAP4L 和 CCAP4L 中的某值与改变着的 16 位 PCA 计数器值相比较，当两值达到相等，同时 WDTE 已置 1 的情况下产生内部复位信号。看家狗定时器的工作方式是应在比较器尚未走到相等之前又再度置数，使程序正常执行时看家狗定时器永远走不到相等；只在程序执行出轨，由于未能按照预定安排重新置数，而致比较器走到了相等，产生了内部复位信号，使软件的执行从头开始。值得指出的是，虽然 CH、CL 和再置 CCAP4H、CCAP4L 都能达到比较器永远走不到相等的目的，但是 PCA 的 5 个计数器共用计数源，如果要给模块 4 的 CH、CL 重新置数，势必要总体释放 PCA 计数器的启动位，因为其它 PCA 模块的正常工作，这是不允许的。所以，只能使用对 CCAP4L、CCAP4L 再行置数。编程时，先将 CCAP4L、CCAP4L 置数；再将 PCA 模式寄存器 CMOD 的看家狗的使能位 WDTE 置 1；最后将模块 4 模式寄存器 CCAPM4 的位 MAT4 置 1 和位 ECOM0 置 1，看家狗开始工作。下面给出看家狗的初始化程序：

```
INIT-WATCHDOG:
      MOV  CCAP4,     #0FFH       ；置最长数 0FFFFH，先写低字节
      MOV  CCAP4H,    #OFFH       ；  需在走完前于程序的适当地点调用装狗子程
      ORL   CMOD,     #40H        ；  向 CMOD 的位 WDTE 置 1，使能看家狗中断
      MOV  CCAPM4,    #48H        ；  向 CCAPM4 的位 MAT4 和位 ECOM4 置 1
```

此后，需要即时地不断重复地给 CCAP4L、CCAP4L 置数，使程序正常执行时不会走到比较器相等。将给 CCAP4H、CCAP4L 置数的程序段写成子程序形式，在应用程序的每隔一段处调用一次此子程序。下面给出向 CCAP4L、CCAP4L 置数的装狗子程序：

```
WATCHDOG：
    CLR  EA                  ；总体关中断
    MOV  CCAP4L，  #00       ；此数应按实际改动，此处的 100H 为例。
    MOV  CCAP4H，  #1H       ；
    SETB EA                  ；总体开中断
```

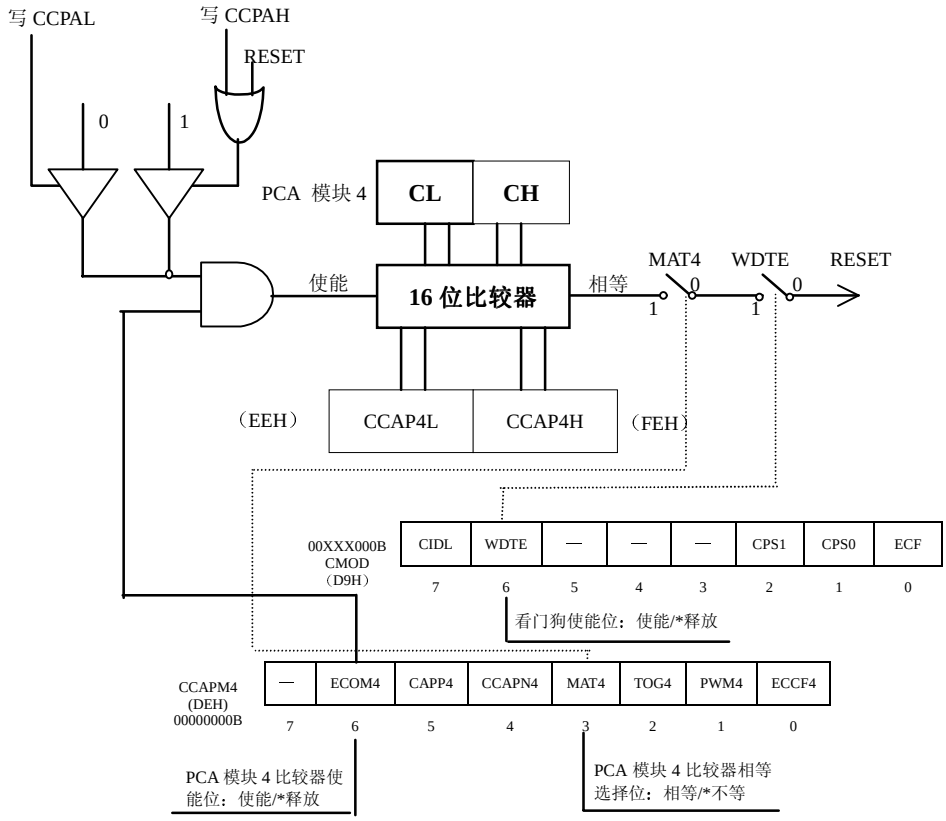


图 4.2.6　PCA 模式 4 看门狗定时器模式

4.3　全双工增强型 UART（异步收发器）串行口

原始的标准 8051 串行口就是全双工双缓冲器的 UART，有四种工作模式：

模式 0　移位寄存器模式　　波特率固定，fosc/12
模式 1　8 数据位 UART　　波特率可变，(1/16 或 1/32)*T1 溢出率
模式 2　9 数据位 UART　　波特率固定，(1/32 或 1/64)*fosc
模式 3　9 数据位 UART　　波特率可变，(1/16 或 1/32)*T1 溢出率

所谓增强型 UART 是增加了两个方面的硬件功能：帧错误检测和地址自动识别。帧错误检测只对模式 2~3 有效，地址自动选择则对模式 1~3 有效。帧错误检测　用检测停止位的方法来反映帧中发生的错误。当发现任何形式的位丢失都会将 SFR 的 SCON 第七位，帧错误标志位 FE 置 1。从图 4.3.1 可以看出，FE 是与 SM0

（模式选择的高位）复用。它们由 SFR 中 PCON 的第六位 SMOD0 进行选择：SMOD0 置 0 选中 FE；置 1 选中 SM0(模式选择高位)。正因选中 FE 时影响了模式选择的高位，而致帧错误检测只对模式 2~3 有效。但是应当说明的是，对于模式 1 实际上内部还是有停止位的检测的，检测正确才启动 RI 将此帧收下来，不正确便不启动 RI 而将此帧直接放弃，仅仅是未留下检测结果而已。所以，对于模式 1，凡收的帧便都是没有位丢失的数据；而模式 2~3 反而要查 FE(帧错误标志位)的状态。

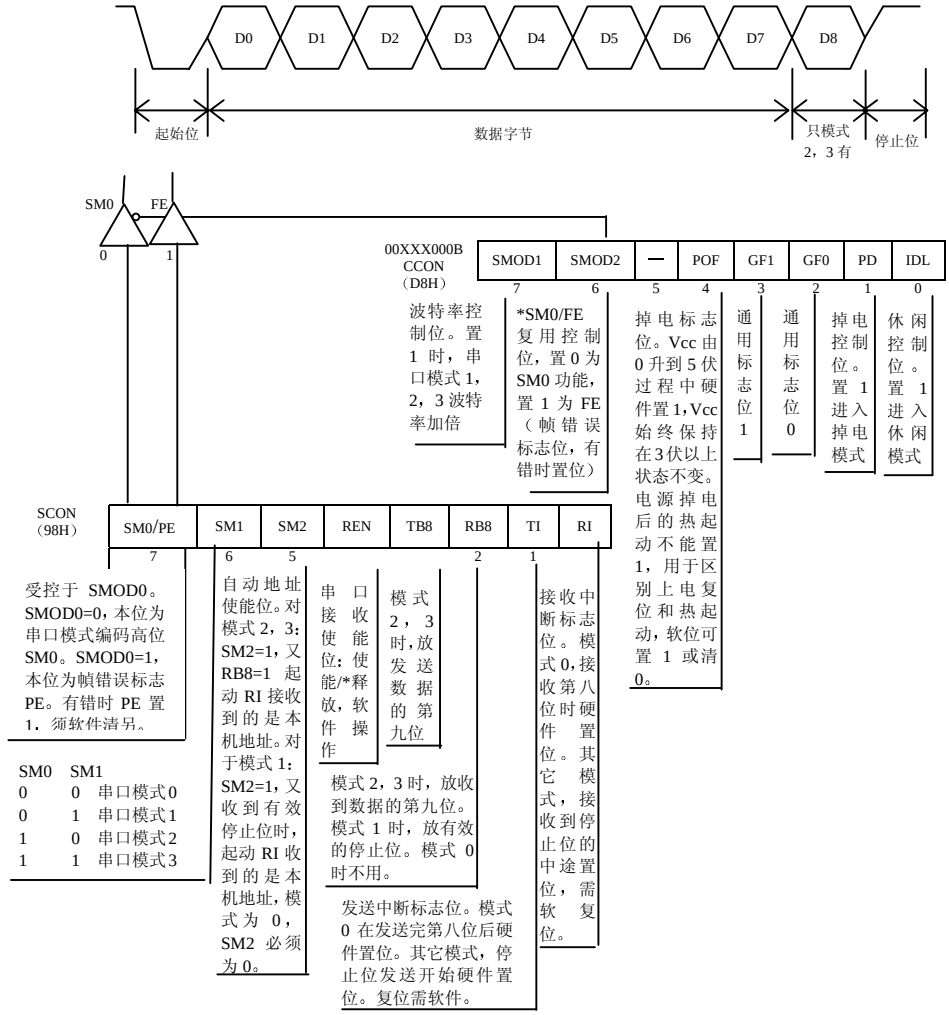


图 4.3.1　帧错误检测

地址自动选择　标准 8051 串行口的特殊功能寄存器 SCON 有一位 SM2，专为多机通信时区分对地址帧和数据帧之用。将 SM2 置 1 时：对模式 2~3 来说，收到第九位（即放入 RB8 的内容）为 1，表示此帧的数据域的前 8 位为地址，并且启动 RI 将此帧收下来，如果收到第九位（即放入 RB8 的内容）为 0，则不启动 RI 而将此帧放弃；对模式 1 来说，放入 RB8 的是有效的停止位 1，并且启动 RI 将此帧收下来，若不是有效的停止位，则不启动 RI 而将此帧直接放弃。如果将 SM2 置 0，便是放弃对地址的辨识，一律做数据接受。上面说的是标准 8051 的情况，而 P89C51Rx+则是将上述过程自动化了。对于 P89C51Rx+，只要将 SM2 置 1，收到的第一帧必是地址，而且是你的地址，所以不要再用软件去读和判，硬件自动化省的正是这些时间的开销，应毫不等待地将 SM2 清 0，等着接受随后到来的数据。数据结束后又应及时将 SM2 置 1，等着再次呼叫你的地址。

为了实现地址的自动识别，有一些定义地址的算法和初始化的工作要做。下面讲这方面的内容。

地址自动识别，需要用到两个特殊功能寄存器，SADDR(从机地址寄存器)和 SADEN(从机地址掩码寄存器)。从机的编码地址放在 SADDR 中，每个从机有唯一的地址掩码，放在 SADEN 中。允许主机与一个或多个从机同时通信，所以需要有一种算法，在主机只发某个综合地址码，便能令主机心目中要通信的一个或多个从机各自看到的都仅只是自己的地址。现在用例子来说明可能算法中的一种。为了假定每台从机的编码地址是一样的，但各自的地址掩码是不一样的，且都是唯一的。下面看一主三从的系统：

	从机 0	从机 1	从机 2
SAADR	1010 1000	1010 1000	1010 1000
SADEN	1111 1001	1111 1010	1111 1100
相与	1010 1xx0	1010 1x0x	1010 10xx
综合地址码(呼从机 0)	1010 1110(中)	1010 1110(不中)	1010 1110(不中)
综合地址码(呼从机 1)	1010 1101(不中)	1010 1101(中)	1010 1101(不中)
综合地址码(呼从机 2)	1010 1011(不中)	1010 1011(不中)	1010 1011(中)
综合地址码(呼从机 0+1)	1010 1100(中)	1010 1100(中)	1010 1100(不中)
综合地址码(呼从机 1+2)	1010 1001(不中)	1010 1001(中)	1010 1001(中)
综合地址码(呼从机 2+0)	1010 1010(中)	1010 1010(不中)	1010 1010(中)
综合地址码(呼从机 0+1+3)	1010 1000(中)	1010 1000(中)	1010 1000(中)

各从机的 SAADR 与 SADEN 相“与”，所得数中的 x 位为可任取 1 或 0，取后的综合码再和自己的 SADEN 相“与”所得结果与原来一样。若将 xx 位均取 1 所得的综合地址码由主机发出，则只有一台　从机选中。若将 xx 中的一个选 0 一个选 1，所得的综合地址码由主机发出，则会有两台从机选中。若 xx 都选 0，则三台从机都选中。本系统的三台从机都选中，又叫广播方式。对于广播方式下面有更简单的算法。

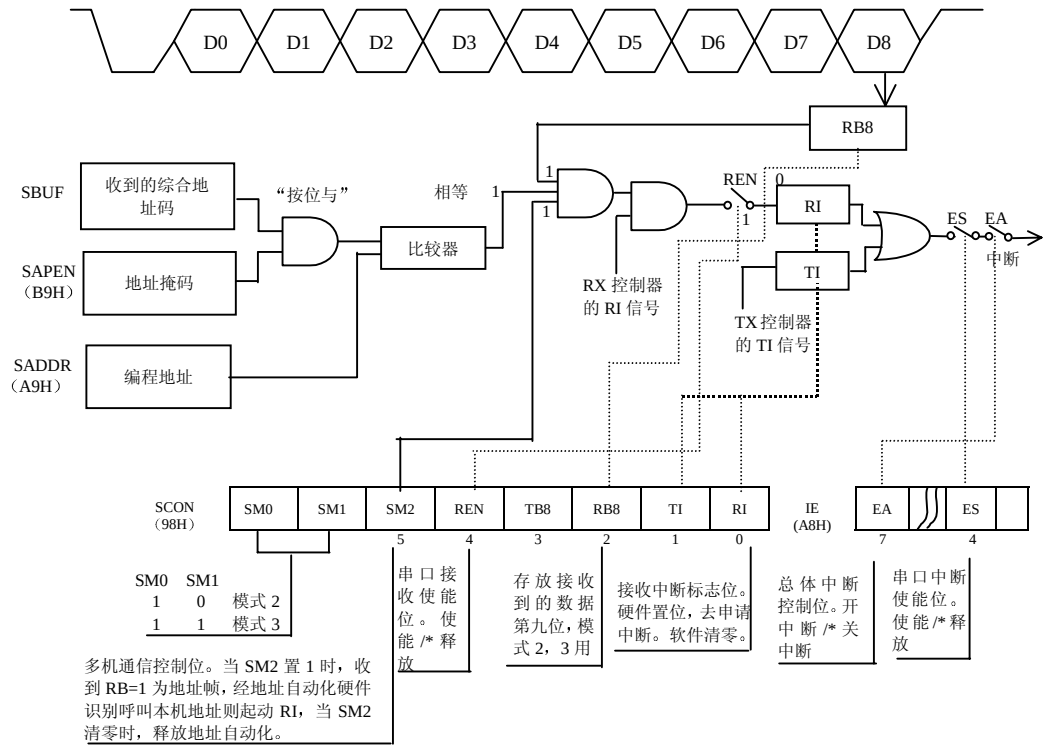


图 4.3.2　自动地址识别

广播方式使用综合地址码和从机 SADEN 相“或”的算法。主机固定发 0FFH 综合地址码和任何从机 SADEN 相“或”，为 0FFH（这是必然的）则判为广播方式，启动 RI 将此帧收下来，立刻

用软件将 SM2 清 0，等待接受数据帧。

中断值	四级顺位级别	向量地址
X0	1	03H
T0	2	08H
X1	3	13H
T1	4	1BH
PCA	5	33H
SP	6	23H
T2	7（低）	2BH

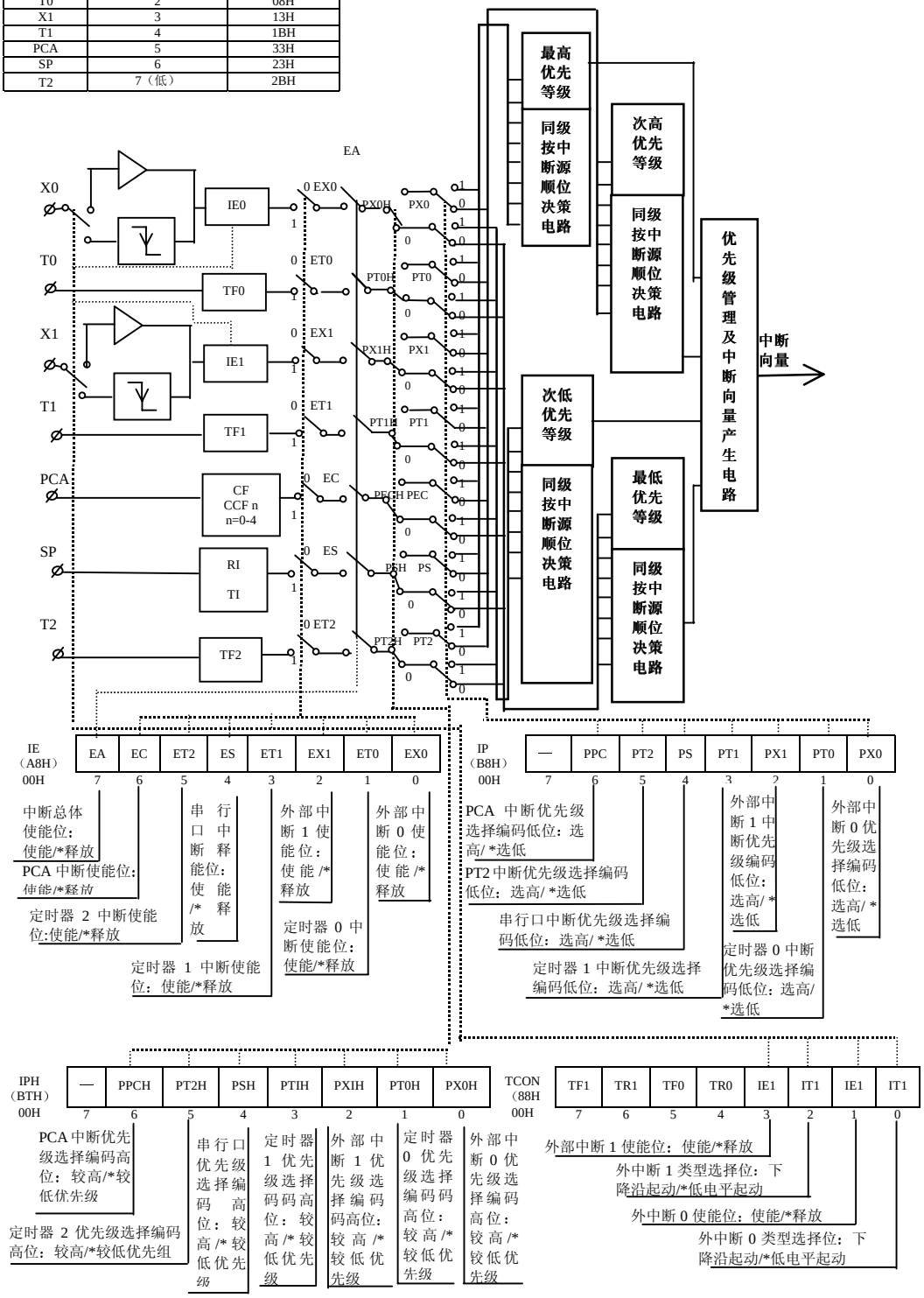


图 4.4.1　中断系统

自动地址识别的编程如下：1，将编码地址和本机的掩码地址分别写入特殊功能寄存器 SADDR 和 SADEN；2，选串口模式 2 或 3（见图 4.3.2）；3，向 SM2 写 1，进入自动地址识别模式。

芯片复位，SADDR 和 SADEN 被清 0，自动地址识别模式被终止，串口进入标准 8051 的 UART 方式。

4.4 7 源 4 优先级嵌套中断系统

标准 8051 是 5 源 2 优先级嵌套中断系统。P89C51Rx+多了定时器 2 和 PCA 两个中断源，与它们有关的中断使能位和中断优先级控制位分别补充在 IE 和 IP 特殊功能寄存器的 5 位、6 位（见图 4.4.1）；P89C51Rx+嵌套的优先级由原来的 2 级增加到 4 级，为此，增加了一个高位的优先级控制寄存器 IPH（见图 4.4.1），为每个中断源的优先级选择添了一位编码的高位，从而将每个中断源的优先级升为 4 级（见图 4.4.1 左上角部分）。两位编码的优先级定义如下：

IPH.x	IP.x	中断源优先级
0	0	0 级（优先级最低）
0	1	1 级
1	0	2 级
1	1	3 级（优先级最高）中断源

至于每一级中同时发出的不同中断申请，则按下述固定顺位进行判定：

同优先级中中断源查询顺位及向量地址

顺位	1(最高)	2	3	4	5	6	7(最低)
中断源	X0	T0	X1	T1	PCA	SP	T2
向量地址	03H	0BH	13H	1BH	33H	23H	2BH

各中断源的向量地址在上表中也已给出。图 4.4.1 总结了 P89C51Rx+中断系统有关的全部内容，对于中断系统的初始化和管理十分有用。

4.5 数据存储器

标准 8051 是哈佛结构的计算机，即将数据和程序截然分开，分别存放在不同的物理存储器中。数据存储器存放变量或运行中间产生的数据；程序存储器存放程序和常量。本节专讲 P89C51Rx+的数据存储器的结构：

4.5.1 内部数据存储器

1、 低 128 字节 RAM（00H~7FH），可直接寻址和间接寻址。如：

直接寻址

```
MOV   7FH,   #DATA
```

间接寻址

```
MOV   R0,      #7FH      ；用于地址<100H。R0 也可改用 R1。
MOV   @R0,     #DATA
```

2、 高 128 字节 RAM（80H~FFH），只能间接寻址。如：

```
MOV   R0,      #0A0H          ;用于地址<100H。R0 也可改用 R1。
MOV   @R0,     #DATA）
```

3、 特殊功能寄存器 SFR（80H~FFH），只能直接寻址。如：

```
MOV   0A0H,    #DATA
```

以上与标准的 8051 内部数据存储器结构是完全相同的。

下面是 P89C51Rx+新增的内部数据存储器：

4、 内部扩展 RAM（ERAM）：100H(256) / 300H(768)字节（89C51RC+ 00H~FFH / 89C51RD+ 00H~2FFH ）。

内部 ERAM 必须使用间接寻址。为区分去内部 ERAM 还是外部数据存储器寻址，须先对特殊功能寄存器 AUXR 的位 EXTRAM 进行选择（见图 4.5.1）：

EXTRAM=1　　寻址外部数据存储器

EXTRAM=0　　寻址内部扩展 RAM（ERAM）

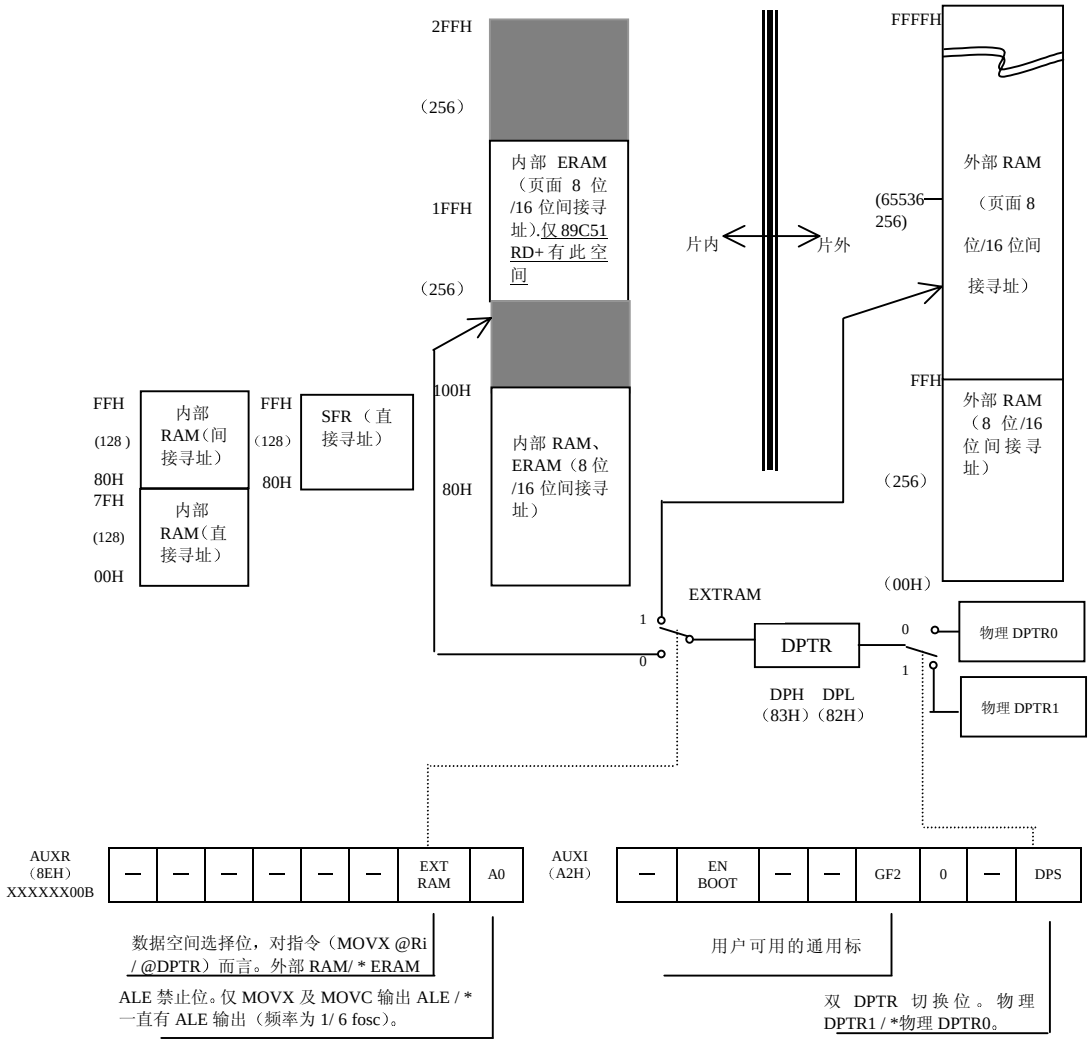


图 4.5.1　片内、片外数据存储器分布及片内 ERAM 寻址图

例 1　寻址 ERAM 地址 0A0H：

```
ANL     AUXR,    #0FDH     ;  EXTRAM=0
MOV     R0,      #0A0H     ;  地址装入 8 位间址寄存器
MOVX    @R0,     #DATA     ;  寻址<100H。R0 也可改用 R1。
```

寻址>=100H 地址，必需使用 16 位间址寄存器 DPTR。

例 2 寻址 ERAM 地址 1A0H：

```
ANL     AUXR,    #0FDH     ;  EXTRAM=0
MOV DPTR,        #1A0H     ;  地址装入 16 位 DPTR
MOVX    @DPTR,   #DATA     ;  寻址>=100H
```

4.5.2 外部数据存储器

可用地址空间为 0000H~0FFFFH，具体使用的空间依物理存储器的多少而定。对外部数据存储器必须使用间接寻址。为区分去内部 ERAM 还是外部数据存储器寻址，须先对特殊功能寄存器 AUXR 的位 EXTRAM 进行选择（见图 4.5.1）：

EXTRAM=1　　寻址外部数据存储器

EXTRAM=0　　寻址内部扩展 RAM（ERAM）

例 1　址外部数据存储器地址 0A0H：

```
ORL     AUXR，   #02H      ；  EXTRAM=1
MOV     R0,      #0A0H     ；  地址装入 8 位间址寄存器
MOVX    @R0,     #DATA     ；寻址<100H。R0 也可改用 R1。
```

寻址>=100H 地址，必需使用 16 位间址寄存器 DPTR。

例 2　寻址外部数据存储器地址 1A0H：

```
ORL     AUXR，   #02H      ；  EXTRAM=1
MOV     DPTR,    #1A0H     ；  地址装入 16 位 DPTR
MOV     A，      #DATA     ；用 DPTR 间址必须经累加器传送
MOVX    @DPTR,   A         ；寻址>=100H
```

4.6　程序存储器

先回顾标准 8051 的程序存储器结构：程序存储器分片内部分和片外部分，片内、片外统一编址，片内部分放在低端，由 0000H 排起，标准 8051 片内部分为 4K 字节，终止于 0FFFH；片外部分接续编址，从 1000H 排起，最多排到 0FFFFH，所以 8051 系列的 程序存储器片内、片外加起来最多只能有 64K。又片内的程序存储器空间可以转移到片外，只要把片腿*EA 事先拉到低点平，然后起动单片机。P89C51RC+有片内程序存储器 32K，而 P89C51RC+有 64K；并且分成 8K 和 16K 的存储块，见图 4.6.1。关于片内程序存储器的类型，P89C51Rx+采用了最先进的 FLASH（快闪）EPROM 存储器，并没有使用 EPROM 和 EEPROM 等 FLASH（快闪）EPROM 存储器比 EPROM 存储器的改进之处是不仅可以读，而且可以用软件快速地擦除和写入，从而引出了在系统中编程（ISP）和在运行中编程（IAP）等新技术。P89C51Rx+可以和其它具有片上 EPROM 单片机一样地用商用编程器进行并行地编程（当然该编程器应支持对 P89C51Rx+编程）。此外，菲利浦公司为 P89C51Rx+在片内提供了一个名叫引导 ROM（BOOT ROM）的 1K 字节的固件。固件上有引导装载程序，可以接收主机经串口传来的命令和数据（如经 PC 机的 RS232 口）；还有对 FLASH 进行串行擦除和写入等多种子程序。这个固件是放在 64K 程序存储器的最高端的，与片内 FLASH 地址 0FC00H~0FFFFH 相覆盖，需要用特殊功能寄存器 AUXR1 的 ENBOOT 位进行固件和 FLASH 之间的切换（见图 4.6.1）：

ENBOOT=1　　地址在 0FC00H～0FFFFH 范围，寻址到固件

ENBOOT=0　　地址在 0FC00H～0FFFFH 范围，寻址到 FLASH

由于 BOOT ROM 固件的存在，使对 P89C51Rx+进行擦除和写入有了三种方法：1，商用编程器进行并行编程；2，用 BOOT ROM 固件由主机进行串行的“在系统中编程（ISP）”；3，在应用程序中调用 BOOT ROM 中的擦除、写入等子程序，进行串行的“在运行中编程（IAP）”。关于 ISP 和 IAP 后面有专门的论述。

4.6.1 P89C51Rx+上电复位代码的执行

P89C51Rx+有两个特殊的 FLASH 寄存器：STATUS BYTE 和 BOOT VECTOR（注意：　它们不在 SFR 中，在 FLASH 空间里）。上电复位，RST 腿由有效正电平转入下降沿时，P89C51Rx+自动检查 STATUS BYTE 的内容，如果是 0，则转去 0000H 地址开始执行程序，这是正常运行方式；如果不是 0，则去 BOOT VECTOR 寄存器取其内容，做为程序计数器的高字节，低字节固定为 00H

（参见图 4.6.1）。芯片出厂时给 BOOT VECTOR 寄存器预置为缺省值 0FCH，相当于 0CF00H 地址，这个地址被固件空间和 FLASH 空间所覆盖，需要用特殊功能寄存器 AUXR1 的 ENBOOT 位进行选择：

ENBOOT=1　　切入固件空间

ENBOOT=0　　切入 FLASH 空间

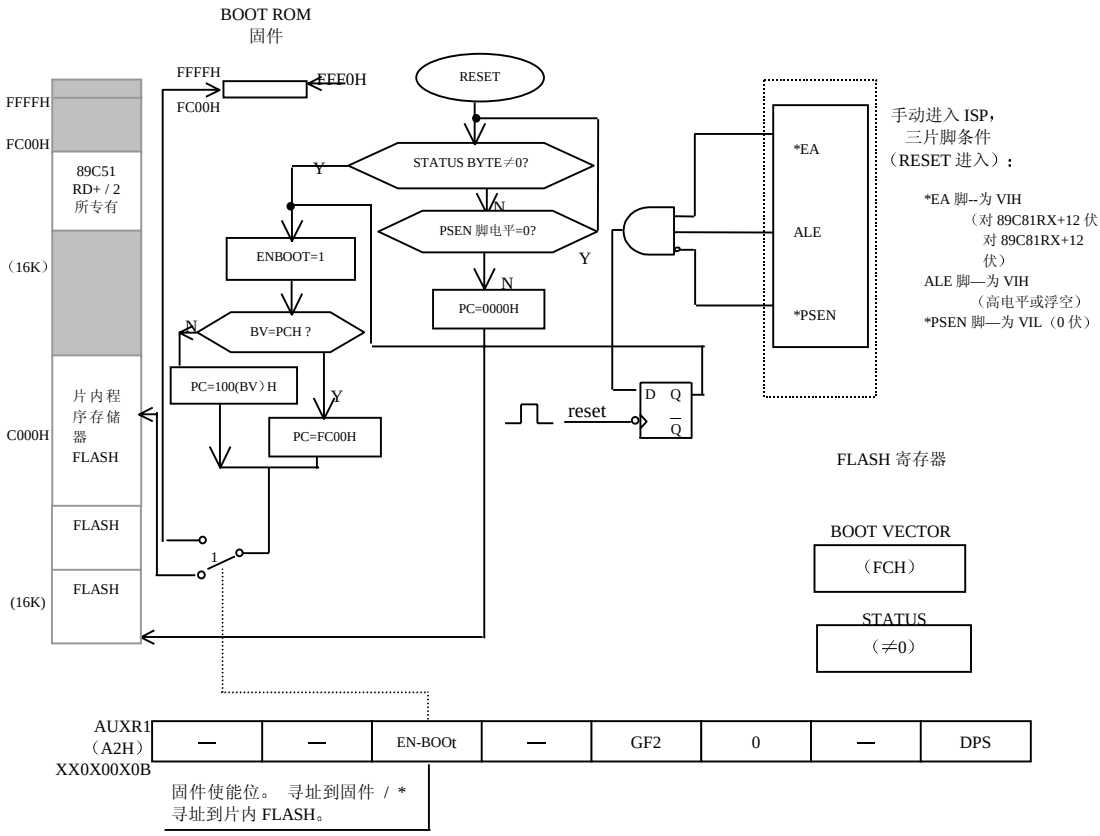


图 4.6.1　片内程序存储器（FLASH 及 ROM 固件）分布图和上电流程序图

ENBOOT 位可由软件或和硬件设置。有两种情况，硬件会自动地将 ENBOOT 置 1：一个是在复位时，因 STATUS BYTE 出厂时预置了非 0 缺省值，导致 ENBOOT 位被硬件自动置成 1；另一个是在 RST 转入下降沿时，下述三条片腿受控如下而致 ENBOOT 位被自动置 1：

- *PSEN 腿被拉到低电平；
- ALE 腿浮空；
- EA 腿的电压>高电平（Vih）。

由于 ENBOOT 位的被置 1，0FC00H 地址被映射到固件的入口，从而被引导进入 ISP 运行状态（参见图 4.6.1）。进入 ISP 运行状态后，主要的工作是：从主机将用户的最终目标代码经串口写入 FLASH；写完之后再完成如下配置任务，即先将 STATUS BYTE 和 BOOT VECTOR 擦除，随后将 BOOT VECTOR 编程为缺省值 0FCH，及 STATUS BYTE 编程为 0。关闭电源，再从新上电起动，芯片便从地址 0000H 开始执行应用程序的目标代码，系统进入正常运行方式。

进入正常运行方式后，如果再想转入 ISP 方式，只有采用上述三腿加适当电平的手动控制进入。

用户也可不用 BOOT ROM 中提供的 ISP 编程程序，而使用自编的编程程序。这时，应将自己的编程程序放在页界上，并把页界的高字节写入 BOOT VECTOR 中。BOOT VECTOR 中的地址，如果既非 0FCH 又非自己编程程序的页界高地址则上电复位的话，程序必将飞溢。这时，唯一解决问题的办法只能使用商用编程器对 BOOT VECTOR 的内容进行并行地改写。

还有一点要注意，当对 FLASH 进行擦除时，FLASH 寄存器 STATUS BYTE 和 BOOT VECTOR 也同时被擦除，所以必需对它们从新编程。还有 BOOT VECTOR 未被擦除前 STATUS BYTE 无法被擦除。

4.6.2 P89C51Rx+ 在系统中编程（ISP）

ISP 编程需要使用芯片的 5 条腿：TxD、RxD、Vcc、Vss、和 Vpp。为了和主机进行 RS232 串口通信，还需要少量的器件和 DB9 型小插座，见图 4.6.2.1。

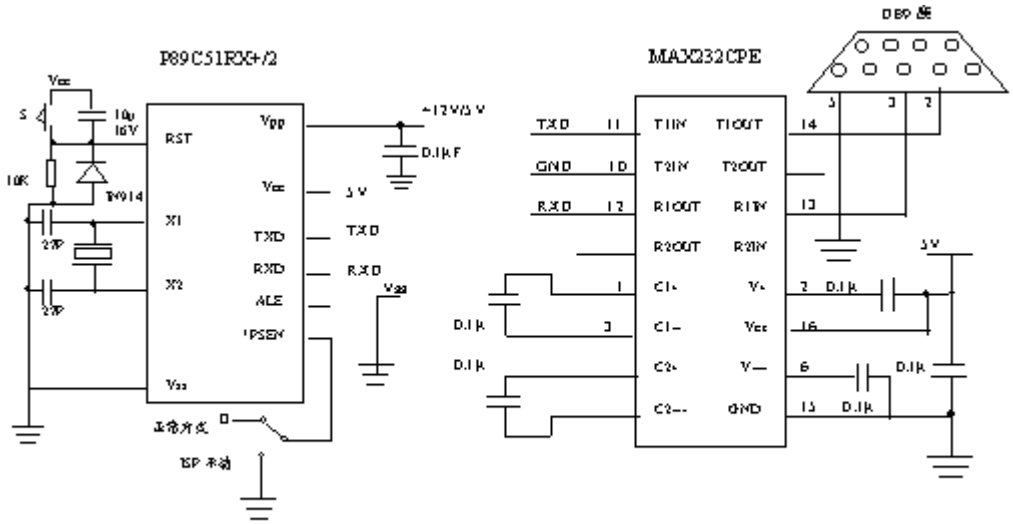


图 4.6.2.1　89C51RX+及 89C51RX2 的 ISP 方式和与主机连接的参考电路

ISP 的通信采用 Intel-Hex 记录格式。Intel-Hex 记录使用 16 进制数表达的字节数据（用 ASCII 字符传送），即一个字节两个 ASCII 符。下面是这种记录的格式：

: NNAAAARRDD…DDCC<cr lf>

其中：:号—是 Intel-Hex 记录的头标

NN—是记录中的字节数，规定不超过 16（10H）字节

AAAA—是记录中数据域首字节的存放地址，如果，记录中数据域字节为 0，则将 AAAA 置为 0000H。

RR—是记录的类型号，见表 4.6.2.1

DD—是数据域，数据域字节可为 0，最多为 10H

CC—是校验和

编程时所用的记录类型列于表 4.6.1。

表 4.6.2.1　　ISP 所用 Intel-Hex 记录的格式类型一览表

记 录 类型	功 能	命令或数据
00	传编程数据并完成编程	格式　:nnaaaa00dd…ddcc 其中　nn　=记录中所传数据字节数，16 进制表示 aaaa　=编程数据首字节地址 dd…dd　=ASCII 码的编程字节数据 cc　=校验和 例　:10000800AF5F67F0602703E0322CFA92007780C3FD
O1	传文件结束符(EOF)	格式　:xxxxxx01cc 其中　xxxxxx　=域空间占满，值无所谓 cc　=校验和 例　:00000001FF
02	指定振荡器频率	格式　:01xxxx02ddcc 其中　xxxx　=域空间占满，值无所谓 dd　=振荡器频率，取 MHz 的整数值（做下舍弃） cc　=校验和 例　:0100000210ED　dd=10H(频率在 16.0~16.9 间均取此值)
03	杂项编程功能（指对 FLASH 寄存器、FLASH 加密位的先擦除再编程）同时含各 FLASH 块的擦除	格式　:nnxxxx03ffssddcc 其中　nn　=记录中所传数据字节数，16 进制表示 xxxx　=域空间占满，值无所谓 03　=杂项编程功能及块擦除 ff　=子功能码 ss　=选择码 dd　=写入数据（需要时才有，不需要没有） cc　=校验和 **1、擦除 FLASH 块** 子功能码 ff =01 选择码　ss= 00H，20H，40H，80H，或 C0H 00H 选 FLASH 块 0　对应地址 0000H~1FFFH 20H 选 FLASH 块 1　对应地址 2000H~3FFFH 40H 选 FLASH 块 2　对应地址 4000H~7FFFH 80H 选 FLASH 块 3　对应地址 8000H~BFFFH C0H 选 FLASH 块 4　对应地址 C000H~FFFFH 例　:0200000301C03A　擦除 FLASH 块 4 **2、擦除 BOOT VECTOR 和 STATUS BYTE** 子功能码　ff =04 选择码　ss= 值无所谓，域空间占满 例　:020000030400F7　擦除两个 FLASH 寄存器 **3、编程加密位** 子功能码　ff =05 选择　ss= 00，01，或 02 00　对加密位 1 编程，禁止写 FLASH 01　对加密位 2 编程，禁止校验 FLASH 02　对加密位 3 编程，禁止外部执行 FLASH 代码 例　:020000030501F5　对加密位 2 编程 **4、编程 BOOT VECTOR 和 STATUS BYTE** 子功能码　ff =06 选择码　ss= 00，或 01 00　对 STATUS BYTE 编程 01　对 BOOT VECTOR 编程 例　:010000030601FCF7　BOOT VECTOR 编程为 0FCH **5、擦除全芯片 FLASH(包括各 FLASH 块、加密位、FLASH 寄存器)及设置 STATUS BYTE 和对 BOOT VECTOR 置缺省值（仅** P89C51Rx2 有此项功能，而 P89C51Rx+无） 子功能码　ff =07 选择码　ss= 无 例　:0100000307F5　FLASH 全擦并置 FLASH 寄存器为缺省值
04	显示指定地址段 FLASH 的数据或做空白检查（显示由收到主机任意字符开始，将数据经串口送出，显示	格式　:05xxxx04sssseeeeffcc 其中　05　=记录中所传数据字节数，16 进制表示 xxxxx　=域空间占满，值无所谓 04　=显示指定地址段 FLASH 的数据或做空白检查 ssss　=起始地址 eeee　=结束地址 ff　=子功能码: 00，或 01

	格式为地址后跟 10H 字节数据，中途收到主机任意字符停止送数据。加密位 2 被编程 FLASH 数据不外送	00=显示 FLASH 数据 01=空白检查 cc =校验和 例 :0500000440004FFF0069 显示 4000~4FFF FLASH 段
05	各种读功能	格式 :02xxxx05ffsscc 其中 02 =记录中所传数据字节数，16 进制表示 xxxx =域空间占满，值无所谓 05 =各种读功能 ffss =子功能和选择码： 0000 =读签名字节—生产厂 id(15H) 0001 =读签名字节—芯片 id #1(c2H) 0000 =读签名字节—芯片 id #2 0700=读加密位 0701=读 STATUS BYTE 0702=读 BOOT VECTOR cc =校验和 例 :020000050001F8 读签名字节—设备 id #1
06	直接装载波特率（仅 P89C51Rx2 有此项功能，而 P89C51Rx+无）	格式 :02xxxx06hhllcc 其中 02 =记录中所传数据字节数，16 进制表示 xxxx =域空间占满，值无所谓 06 =直接装载波特率功能码 hh =定时器 2 计数值高字节 ll =定时器 2 计数值低字节 cc =校验和 例 :02000006F500F3 定时器 2 计数值 F500

ISP 编程的具体步骤建议如下：

1、 进入 ISP 运行模式，用前述出厂缺省方法或手动方法。
2、 由主机向芯片发送一个大写“U”字符，供自动确定波特率使用。
3、 由主机向芯片发送一个记录指定振荡器频率。
4、 由主机向芯片发送一个记录擦除指定存储块。
5、 由主机向芯片发送 FLASH 编程记录，重复到目标程序送完。
6、 擦除 BOOT VECTOR 和 STATUS BYTE 的内容。
7、 若欲继续使用固件引导进行 ISP 编程的话，再将 BOOT VECTOR 编程为缺省值 0FCH。
8、 将 STATUS BYTE 编程为 00H，使下次上电复位时芯片由 0000H 开始执行，进入正常运行模式，。

ISP 可以在宽广的范围内适应用户基于振荡器频率的或不依赖于振荡器频率的波特率。它通过首次接收到的单个字符实现对位时间的测量，并用这个信息给基于振荡器频率的计数器进行波特率设定。为此，主机初始须向芯片发送一个大写“U”字符，以备芯片对波特率进行设定。

完成了波特率的设定，ISP 固件即可接受 Intel-Hex 格式的记录。P89C51Rx+收到记录后先计算记录的校验和，如与校验字节值不相符，则向主机发出“X“回应，表示接收错；如相符，则向主机发出“.“回应，表示妥收，随后按照类型号完成指定的操作，见表 4.6.1。对于数据编程的记录稍有不同，除收到记录，经计算与校验和不符向主机发出“X“回应外，如相符，不是立即向主机发“.”回应，而是立即对 FLASH 按所收到的记录开始编程，若发生字节编程错误，向主机发出“R”，仅在全部字节编程正确，才向主机发出“.”回应，表示执行也成功。主机收到回应“R”，知道编程的定时有问题，应发 02 号记录，为芯片规定振荡器频率。

ISP 设计时是不要求指定振荡器频率，因自身能够确定可用的波特率和编程脉冲的定时。但当需要为产生正确的定时而辅助指定振荡器频率时，可用 02 号记录实现。

P89C51Rx+Intel-Hex 格式的记录，根据记录类型号知道应完成的任务内容，调用 BOOT ROM 固件

中相应的子程序自动完成必要的操作。所以，在微控制器一侧，只要用户把 RS232 的联线接好，并保证起机能进入 ISP 运行模式之外，几乎没有工作要做。然而，在主机一侧需要按表 4.6.2.1 排出各种有用的记录码和编写传送记录的通信程序。如果主机使用 PC 且操作系统为 Win95 以上，可以编写 Windows 图形界面的通用 ISP 程序，从而使主机侧的工作变成只需用鼠标点击就可轻松地完成各种记录的发送。事实上，如果使用 PC 机，你连程序都不用编，因为可以从 Internet 网上免费下载菲利普公司提供的 WinISP 实用软件。下载网址为：www.semiconductors.philips.com。下面简述 ISP 主机侧实用软件 WinISP 的使用。

将 WinISP 装入 PC，用鼠标指定所用微控制器的型号（P89C51RC+、P89C51RD+、P89C51RC2、P89C51RD2），为 ISP 选定 PC 机侧所用的串口（COM1~COM4），然后进入 WinISP 的命令选择：
点击 LOAD FILE 按钮，往对话框输入欲装入文件名
点击 ERASE BLOCKS 按钮，用鼠标选中欲擦除 FLASH 块，再点击 ERASE 按钮完成擦除
同理，点击 BLANK CHECK 按钮进行空白检查
点击 PROGRAM PART 按钮进入编程
点击 READ PART 按钮进入读数据
点击 VERIFY PART 按钮进入校验
在 RANGE 栏中输入起始地址和终止地址，点击 FILL BUFFER 按钮，再往对话　框输入欲装入的数据码，完成数据装入等

4.6.3　P89C51Rx+在运行中编程（IAP）

在运行中编程（IAP）是指在用户的应用程序中对 FLASH 块、FLASH 寄存器、加密位等进行擦除和编程等操作。事实上，BOOT　ROM 固件中已经固化有上述擦除和编程等子程序，只要应用程序来调用即可。为了调用的方便给各种功能子程序一个共用的函数调用入口，名为 PGM-MTP，地址为 0FFF0H；而必要的输入参数则通过寄存器带入，至于返回参数如果有需要则放在 ACC 中。擦除和编程所用的振荡器频率，经下舍弃为整数值后，应作为输入参数规定放于寄存器 R0，为区分各功能子程序而用的功能码则放于寄存器 R1，其余参数见表 4.6.3.1 。表 4.6.3.1 是 IAP 函数调用的功能一览表，应仔细阅读，使用时加以套用。

P89C51Rx+的加密位有互相独立的三位：

加密位 1	置 1	禁止写 FLASH
加密位 2	置 1	禁止校验 FLASH
加密位 3	置 1	禁止外部执行 FLASH 代码
加密位 1,2,3	全清 0	执行外部程序存储器 MOVC 取不到内部存储器代码
各加密位	任何设置	对芯片 FLASH 的擦除没有制约作用

表 4.6.3.1　　IAP 函数调用的功能一览表

序号	调用功能	输入 / 返回 参数
01	擦除 FLASH 块	输入参数：R0　=振荡器频率（下舍弃为整数） R1　=01H　；擦除 FLASH 块功能码 DPTR =0000H，2000H，4000H，8000H，C000H 0000H 擦除 FLASH 块 0　对应 0000H~1FFFH 2000H 擦除 FLASH 块 1　对应 2000H~3FFFH 4000H 擦除 FLASH 块 2　对应 4000H~7FFFH 8000H 擦除 FLASH 块 3　对应 8000H~BFFFH C000H 擦除 FLASH 块 4　对应 C000H~FFFFH ACC　=欲编程字节内容 返回参数：无 函数调用例： ；函数入口（各功能共用）：PGM-MTP（0FFF0H） ；　功能名：ERSBLK ERSBLK: MOV　AUXR1，#20H　；AUXR1 的位 ENBOOT 置 1

		MOV R0，#11 ；假定 fosc=11.0592，取 11 MOV R1，#01H ;字节编程功能码 MOV DPTR，#blk-addr ; 欲编程地址 CALL PGM-MTP ；调用 RET
02	字节编程	输入参数：R0 =振荡器频率（下舍弃为整数） R1 =02H ；字节编程功能码 DPTR =欲编程字节地址 ACC =欲编程字节内容 返回参数：ACC =0（调用成功），非 0（调用失败） 函数调用例： ； 函数入口（各功能共用）：PGM-MTP ； 功能名：WRDATA WRDATA： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#02H ;字节编程功能 MOV DPTR，#address ；欲编程地址 MOV ACC，#mydata ；欲编程内容 CALL PGM-MTP ；调用 RET
03	读 FLASH 字节	输入参数： R0 =振荡器频率（下舍弃为整数） R1 =03H ；读字节功能码 DPTR =欲读字节地址 返回参数：ACC =读出字节内容 函数调用例： ； 函数入口（各功能共用）：PGM-MTP ； 功能名：RDDATA RDDATA： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#03H ；读字节功能 MOV DPTR，#address ；欲读字节地址 CALL PGM-MTP ；调用 RET
04	擦除 BOOT VECTOR (BV) 和 STATUS BYTE (SB)	输入参数：R0 =振荡器频率（下舍弃为整数） R1 =04H ；擦除 BV 和 SB 功能码 DPH =00H DPL =无关紧要 返回参数：无 函数调用例： ； 函数入口（各功能共用）：PGM-MTP ； 功能名：ERBVSB ERBVSB： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#04H ;字节编程功能 MOV DPH，#00H ；欲编程地址 CALL PGM-MTP ；调用 RET
05	加密位编程	输入参数： R0 =振荡器频率（下舍弃为整数） R1 =05H ；加密位编程功能码 DPH =00H DPL =00，01，或 02 00 对加密位 1 编程，禁止写 FLASH 01 对加密位 2 编程，禁止校验 FLASH 02 对加密位 3 编程，禁止外部执行 FLASH 代码 返回参数：无 函数调用例： ； 函数入口（各功能共用）：PGM-MTP ； 功能名：WRSB1（将加密位 1 置位） ； WRSB2（将加密位 2 置位） ； WRSB3（将加密位 3 置位） WRSB1： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1

		MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#05H ；加密位编程功能 MOV DPTR，#0000H ；欲编程加密位 1 CALL PGM-MTP ；调用 RET WRSB2： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#05H ；加密位编程功能 MOV DPTR，#0001H ；欲编程加密位 2 CALL PGM-MTP ；调用 RET WRSB3： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#05H ；加密位编程功能 MOV DPTR，#0002H ；欲编程加密位 3 CALL PGM-MTP ；调用 RET
06	STATUS BYTE 和 BOOT VECTOR 编程	输入参数： R0 =振荡器频率（下舍弃为整数） R1 =06H ; SB 和 BV 编程功能码 DPH =00H DPL =00，01 00 对 STATUS BYTE 编程 01 对 BOOT VECTOR 编程 ACC =欲编程内容 返回参数： ACC =已编程内容 函数调用例： ；函数入口（各功能共用）：PGM-MTP ； 功能名：WRSB（对 STATUS BYTE 编程） ； WRBV（对 BOOT VECTOR 编程） WRSB： MOV AUXR1，#20H ;AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ;假定 fosc =11。0592，取 11 MOV R1，#06H ; SB 和 BV 编程功能 MOV DPTR，#0000H ；欲编程 SB MOV A，#new-sb ； 欲编程值 CALL PGM-MTP ；调用 RET WRBV： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#06H ; SB 和 BV 编程功能 MOV DPTR，#0001H ；欲编程 BV MOV A，#new-bv ； 欲编程值 CALL PGM-MTP ；调用 RET
07	读各加密位、SB、BV 内容	输入参数： R0 =振荡器频率（下舍弃为整数） R1 =07H ；读加密位、SB、BV 功能码 DPH =00H DPL =00，01，02 00 读各加密位内容 01 读 STATUS BYTE 内容 02 读 BOOT VECTOER 内容 返回参数：ACC =读出内容（读出各加密位内容放 2~0 位） 函数调用例： ；函数入口（各功能共用）：PGM-MTP ； 功能名： RDSBITS（读各加密位内容） ； RDSB （读 STATUS BYTE 内容） ； RDBV （读 BOOT VECTOER 内容） RDSBITS： MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#07H ；读加密位、SB、BV 功能码 MOV DPTR，#0000H ；读各加密位内容

		CALL PGM-MTP ；调用 RET RDSB: MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#07H ；读加密位、SB、BV 功能码 MOV DPTR，#0001H ；读 SB 内容 CALL PGM-MTP ；调用 RET RDBV: MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#07H ；读加密位、SB、BV 功能码 MOV DPTR，#0002H ；读 BV 内容 CALL PGM-MTP ；调用 RET
08	擦除全芯片 FLASH(包括擦除各 FLASH 块、FLASH 寄存器、加密位)并随后将 SB 和 BV 置为缺省值。（仅 P89C51Rx2 有此项功能，而 P89C51Rx+无）	输入参数： R0 =振荡器频率（下舍弃为整数） R1 =08H ；擦除全芯片 FLASH 及置缺省值 DPH =无用 DPL =无用 返回参数：无 函数调用例: ；函数入口（各功能共用）：PGM-MTP ； 功能名： FCE（全芯片擦除，含置必要缺省值） FCE: MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#08H ；擦除全芯片 FLASH CALL PGM-MTP ；调用 RET
09	读 ID	输入参数： R0 =振荡器频率（下舍弃为整数） R1 =00H ；读 ID 功能码 DPH =00H DPL =00H，01H，02H 00H （生产厂 ID） 01H （芯片 ID #1） 02H （芯片 ID #2） 返回参数：ACC =读出内容 函数调用例: ；函数入口（各功能共用）：PGM-MTP ； 功能名： RDMID （读生产厂 ID 内容） ； RDDID1 （读芯片 ID #1 内容） ； RDDID2 （读芯片 ID #2 内容） RDMID: MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#00H ；读 ID 功能码 MOV DPTR，#0000H ；读生产厂 ID 内容 CALL PGM-MTP ；调用 RET RDDID1: MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#00H ；读 ID 功能码 MOV DPTR，#0001H ；读芯片 ID #1 内容 CALL PGM-MTP ；调用 RET RDDID2: MOV AUXR1，#20H ；AUXR1 的位 ENBOOT 置 1 MOV R0，#11 ；假定 fosc =11。0592，取 11 MOV R1，#00H ；读 ID 功能码 MOV DPTR，#0002H ；读芯片 ID #2 内容 CALL PGM-MTP ；调用 RET

4.6.4 芯片振荡器特性和复位

片上提供振荡器电路，但晶体需要经由 XTAL1 和 XTAL2 两腿外加，XTAL1 和 XTAL2 是内部反向放大的输入端和输出端（见图 3.1.1）。如果使用外部时钟源，应由 XTAL1 接入，而 XTAL2 空着不用。对外部时钟源的占空比没有要求，因为内部使用了二分频电路；但，外部时钟源的高低电平的上下限必须满足数据规范表中的要求。

所谓复位，是指在振荡器工作稳定后（一般不超过 10 毫秒），芯片的复位腿 RST 保持正的有效电平至少两个机器周期（24 振荡器周期），以便内部的复位操作起作用。如果是上电复位，Vcc 电压与 RST 腿电平应同时起动。RST 腿的电压达到 Vih1 的低限时，口 1、2、3 进入它们的复位状态 0FFH。*EA 腿的状态，在 RST 腿电平转为无效低电平时被锁存不变。其余各 SFR 复位后的状态见特殊功能寄存器表的复位值一栏。

4.7 芯片的特殊工作模式

本书所介绍的芯片有四种特殊工作模式如下。

4.7.1 另钟频模式

芯片的全静态工艺设计是指钟频可以逐级降低，以节约电源的能耗，降低到极限就是另钟频，此时虽然不能工作，但片内 RAM 和特殊功能寄存器的内容会保持原工作值不变，等待时机恢复工作。后面的下电工作模式就是既能恢复工作又最省电的一种模式。

4.7.2 休眠工作模式

休眠工作模式是指:芯片的 CPU 部分因钟频被切断而停止工作，但片上的外部设备如定时器、中断系统、串并口等仍处于工作状态，而控制信号 ALE 和*PSEN 均为 1。关于 CPU 工作所需的的程序指针、堆栈指针等，还有片内 RAM 和特殊功能寄存器的内容都保持着休眠前的工作值不变，等待时机恢复工作。

启动休眠工作模式，需要执行将特殊功能寄存器 PCON 的位 IDL（PCON.0）置 1 的指令（如 ORL PCON，#01H），待此指令执行结束，即进入休眠模式。结束休眠模式有两种方法：1，任何已开放的中断被触发，引起硬件对 IDL 位的清 0，从而结束休眠模式。当中断服务程序结束，因其 RETI 指令的执行，而将控制返回到 IDL 置 1 指令的下条指令继续执行。为区分结束休眠产生的中断和正常中断可使用通用标志 GF0（PCON.2）或 GF1（PCON.3）。例如，启动 IDL 指令前将通用标志之一置 1，并在中断服务程序中检查此标志。2，复位腿加有效复位电平，结束休眠模式。腿上的有效复位电平应保持两个机器周期，才能使内部 RESET 机制起作用，引起从 0000 地址开始执行应用程序（注意，并非接续执行）。有一种情况应当注意，因 RST 腿上的有效复位电平至少要保持两个机器周期才能使内部 RESET 机制起作用，故在这两个机器周期时间内，一旦 CPU 的时钟被接通，有可能偷着执行了 IDL 指令的下条指令。这下一条指令若是一般的指令倒也无所谓，因为随后就是复位；唯独这下一条指令不能是操作 P 口的指令，因为有可能向接在口上的外部电路发出错误的控制，造成不良后果。为了防备万一，所以，IDL 指令的下一条不应放操作 P 口的指令。

4.7.3 下电工作模式

下电工作模式是指主动将电源切除迫使芯片进入 0 钟频状态。CPU 的内容（如程序指针、堆

栈指针等）、片内 RAM 和 SFR 的内容，在电压下降到 2 伏时被保持起来；片上的外部设备，如定时器、中断系统、串并口也停止工作；控制信号 ALE 和*PSEN 转为 0 态。

启动下电工作模式，需要执行将特殊功能寄存器 PCON 的位 IDL（PCON.1）置 1 的指令（如 ORL PCON，#02H），待此指令执行结束，即进入下电模式。为结束下电模式，应先给电源 Vcc，使振荡器起振并稳定（不超过 10ms）；为结束下电模式用下面两种方法之一：1，已开放的**外部中断**被触发，引起硬件对 PD（PCON.1）的清 0，结束下电状态（注意，外部中断应事先置为电平触发方式）。也就是将外部中断腿由高电平瞬时拉低即退出下电状态。当此中断被服务结束，由其最后的 RETI 指令返回到原启动 PD 指令的下条指令，程序继续执行。为区分结束下电模式产生的中断和正常中断也可使用通用标志 GF0（PCON.2）或 GF1（PCON.3）。2，用复位腿加有效复位电平结束下电模式。振荡器起振并稳定情况下，有效复位电平持续两个机器周期即能成功地引起复位，从新从 0000 地址开始执行。复位操作会将 SFR 内容全部复位，但片内 RAM 内容未改动。

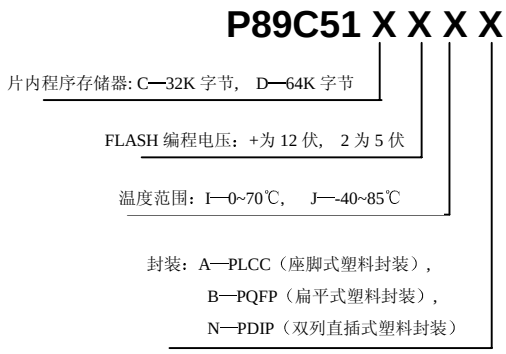
应该指出，芯片上还提供了一个断电标志位 POF（POWER OFF FLAG，位于 PCON.4）。在 Vcc 由 0 伏升到 5 伏的过程中 POF 被置位，只要 Vcc 保持在 3 伏以上，标志会一直维持为 1。软件可以对它置位和清 0。利用 POF 可以区别上电复位和热复位。

4.7.4 ONCE 仿真模式

ONCE（On-Circuit Emulation）仿真模式方便仿真的进行，因为不用把芯片从电路版上拔下，再插上仿真头。只要芯片进入 ONCE 仿真模式，仿真器或仿真 CPU 就可以通过目标板上微控制器芯片的 P 口，控制芯片的运行，实现测验和调试。进入 ONCE 仿真模式的方法是：将 ALE 拉低、*PSEN 拉高时进行复位，且复位解除后 ALE 依旧被拉低。

解除 ONCE 仿真模式很简单，将 ALE 和*PSEN 腿恢复到正常电平，进行正常复位，即回到正常模式。

4.8 订货须知



例：P89CRD2JA

P89C51RD+JN 芯片：选 D--64K 字节片内 FLASH，选 2—5 伏编程电源，选 J--温度范围 40~85℃，选 A--座脚式塑料封装的,。