

P89C51RA2/RB2/RC2/RD2xx 中文器件手册

概述

P89C51RA2/RB2/RC2/RD2xx 具有 8K/16K/32K/64K 并行可编程的非易失性 FLASH 程序存储器，并可实现对器件串行在系统编程 (ISP) 和在应用中编程(IAP)。在系统编程 (ISP: In-System Programming): 当 MCU 安装在用户板上时，允许用户下载新的代码。在应用中编程 (IAP: In-Application Programming): MCU 可以在系统中获取新代码并对自己重新编程。这种方法允许通过调制解调器连接进行远程编程。片内 ROM 中固化的默认的串行加载程序 (Boot Loader) 允许 ISP 通过 UART 将程序代码装入 Flash 存储器，而 Flash 代码中则不需要加载程序。对于 IAP，用户程序通过使用片内 ROM 中的标准程序对 Flash 存储器进行擦除和重新编程。

该器件可通过并行编程或在系统编程的方法对一个 Flash 位进行编程，从而选择 6 时钟或 12 时钟模式。此外，也可通过时钟控制寄存器 CKCON 中的 X2 位选择 6 时钟或 12 时钟模式。另外，当处于 6 时钟模式时，外围功能可以选择一个机器周期 6 时钟或是 12 时钟。这是通过 CKCON 寄存器进行选择的。

该系列微控制器是 80C51 微控制器的派生器件，是采用先进 CMOS 工艺制造的 8 位微控制器，指令系统与 80C51 完全相同。该器件有 4 组 8 位 I/O 口、3 个 16 位定时/计数器、多中断源-4 中断优先级-嵌套的中断结构、1 个增强型 UART、片内振荡器及时序电路。

新增的特性使得 P89C51RA2/RB2/RC2/RD2 成为功能更强大的微控制器，更好地支持应用于脉宽调制，高速 I/O，递增/递减计数能力（如电机控制）等场合。

选型表

型号	存储器				定时器				串行接口				ADC bits/ch.	I/O 口	中断 (外部/电平/触发)	程序加密	默认时钟速率	可选项速率	复位有效 低/高	6/12 时钟 时的最大 频率 (MHz)	3V 时的 频率范围 (MHz)	5V 时的 频率范围 (MHz)
	RAM	ROM	OTP	Flash	定时/计数 器个数	PWM	PCA	WD	UART	I ² C	CAN	SPI										
P89C51RD2xx	1K	-	-	64K	4	√	√	√	√	-	-	-	-	32	7(2)/4	√	12 时钟	6 时钟	高	20/33	-	0-20/33
P89C51RC2xx	512B	-	-	32K	4	√	√	√	√	-	-	-	-	32	7(2)/4	√	12 时钟	6 时钟	高	20/33	-	0-20/33
P89C51RB2xx	512B	-	-	16K	4	√	√	√	√	-	-	-	-	32	7(2)/4	√	12 时钟	6 时钟	高	20/33	-	0-20/33
P89C51RA2xx	512B	-	-	8K	4	√	√	√	√	-	-	-	-	32	7(2)/4	√	12 时钟	6 时钟	高	20/33	-	0-20/33

特性

- 80C51 中央处理单元
- 具有 ISP 和 IAP 功能的片内 FLASH 程序存储器
- 片内 Boot ROM 包含底层 FLASH 编程子程序，以实现通过 UART 下载程序
- 可实现最终用户应用的编程 (IAP)
- 与 87C51 兼容的并行编程硬件接口
- 每个机器周期为 6 个时钟周期 (标准)
- 可通过并行编程器选择 6 时钟/12 时钟模式 (芯片擦除后默认的时钟模式为 12 时钟)
- 可通过 ISP 对选择 6 时钟/12 时钟模式的 Flash 位进行擦除和编程
- 可通过 SFR 位 “在运行中改变” 6 时钟/12 时钟模式
- 当 CPU 为 6 时钟模式时，外围功能 (PCA、定时器、UART) 可选择使用 6 时钟/12 时钟模式
- 采用 6 时钟周期时频率可高达 20MHz (相当于 40MHz)，采用 12 时钟周期时频率可达 33MHz
- 全静态操作
- RAM 可外部扩展到 64K 字节
- 4 个中断优先级
- 7 个中断源

- 4 个 8 位 I/O 口
- 全双工增强型 UART
 - 帧错误检测
 - 自动地址识别
- 电源控制模式
 - 时钟可停止和恢复
 - 空闲模式
 - 掉电模式
- 可编程时钟输出
- 异步端口复位
- 双 DPTR 寄存器
- 低 EMI (禁止 ALE)
- 可编程计数器阵列 (PCA)
 - PWM
 - 捕获/比较

器件比较表

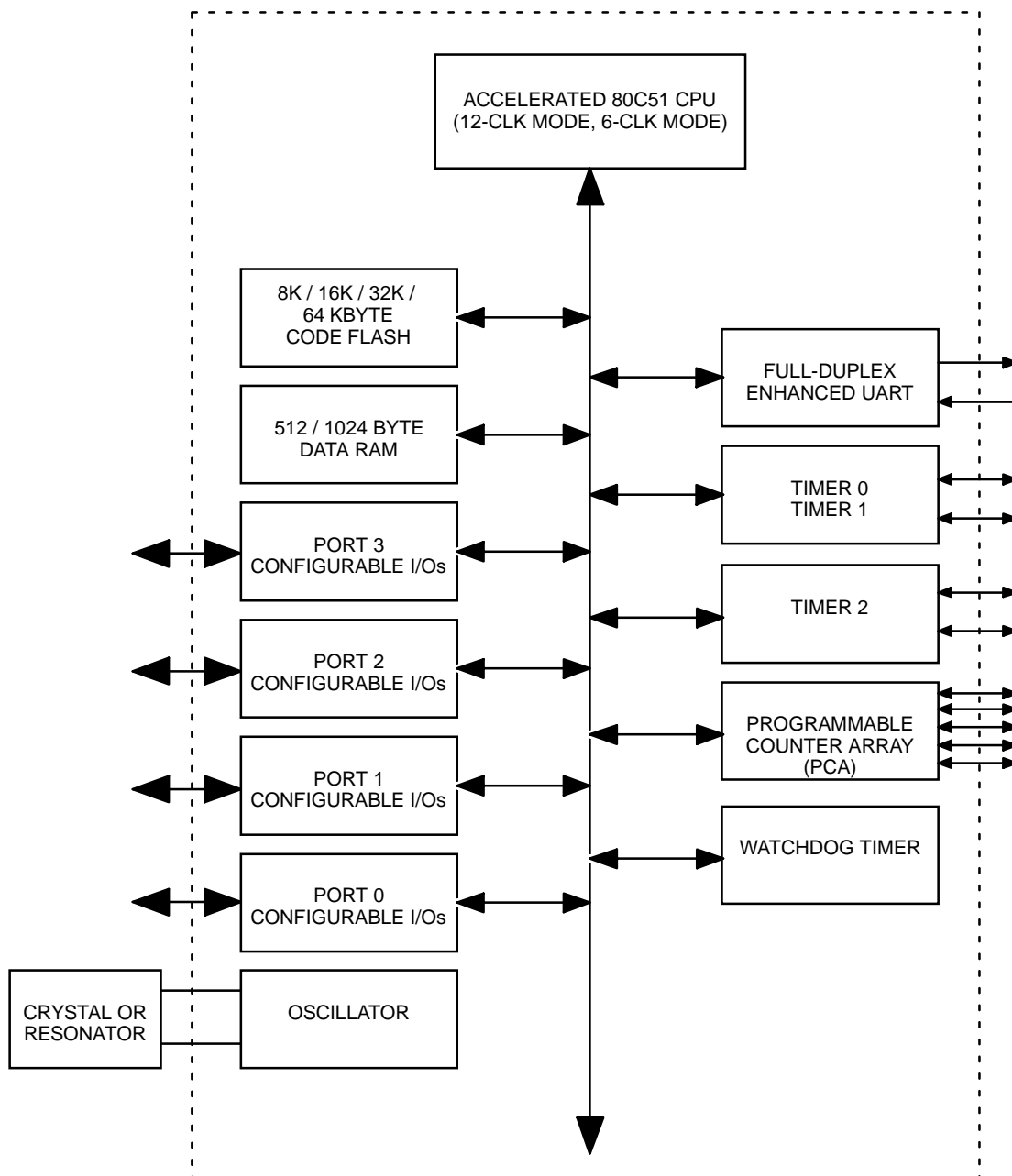
项目	第一代器件	第二代器件	区别
类型描述	P89C51Rx2Hxx(x)	P89C51Rx2xx(x)	字母 H 被取消
编程算法	当使用并行编程器时, 请确定选择 P89C51Rx2Hxx(x) 器件	当使用并行编程器时, 请确定选择 P89C51Rxx(x) (无字母 H)	由于制造工艺的改变, 编程算法和以前不同
时钟模式(1)	默认为 6 时钟, 可使用并行编程器通过 OTP 配置位设置为 12 时钟模式, 但设置后不能再编程为 6 时钟模式。	默认为 12 时钟, 使用并行编程器或 ISP 方式通过 Flash 配置位设置为 6 时钟模式 (可重新编程)	为用户提供更多的灵活性, 与老版本的 P89C51Rx+ 更加兼容。
时钟模式(2)	N/A	可通过 SFR 位 X2 (CKCON.0) 在运行中改变时钟模式。	可通过软件改变时钟模式
外围功能的时钟模式	N/A	CPU 以 6 时钟模式运行时, 外围功能可以 12 时钟模式运行	更多的灵活性, 更低的功耗
FLASH 块结构	两个 8K 字节块 1~3 个 16K 字节块	2~16 个 4K 字节块	更多灵活性

订购信息

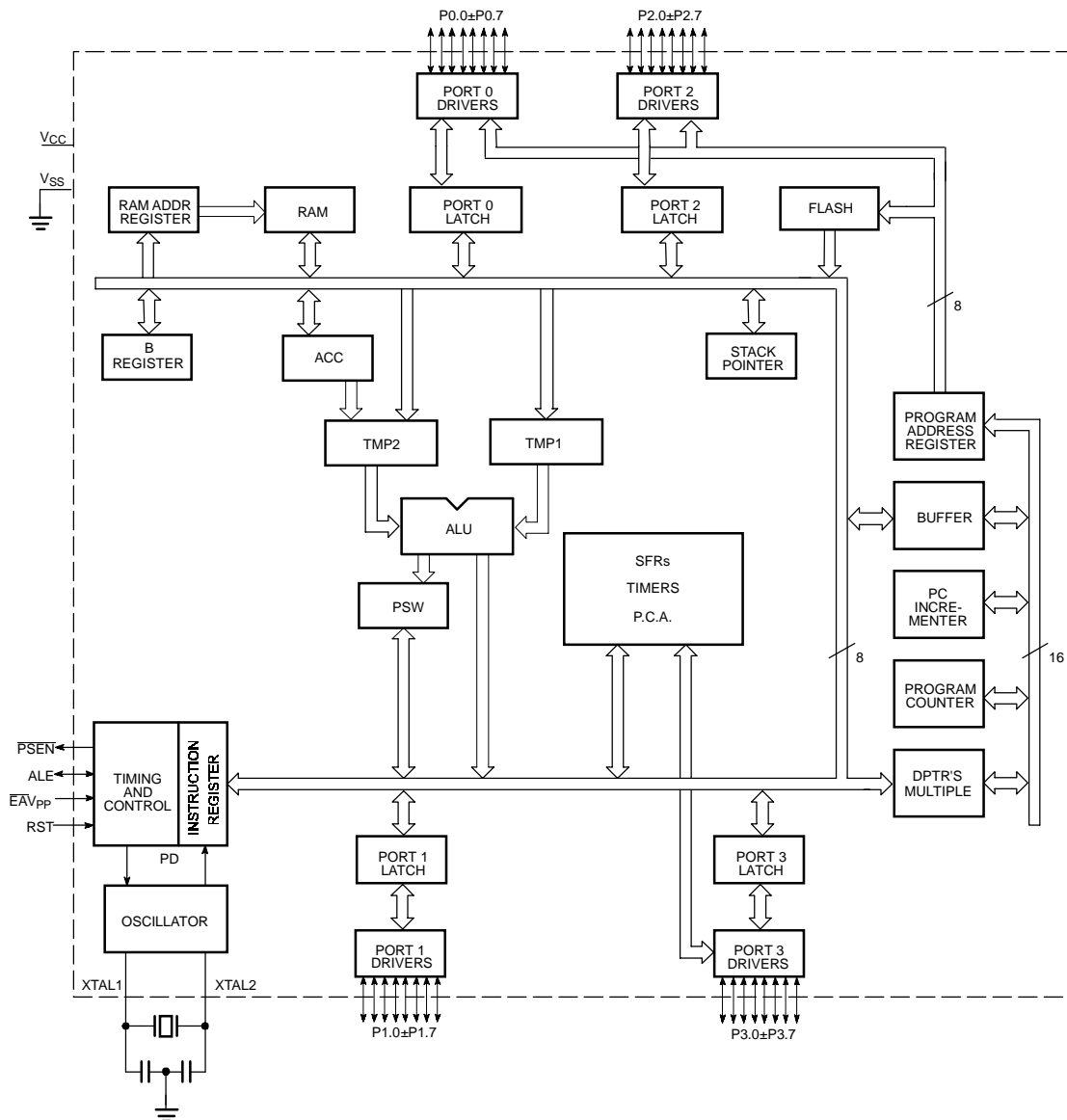
	货品号	存储器		温度范围和封装	电压范围	频率 (MHz)	
		Flash	RAM			6 时钟	12 时钟
1	P89C51RA2BA	8KB	512B	0~+70, PLCC	4.5-5.5V	0-20	0-33
2	P89C51RA2BBD	8KB	512B	0~+70, LQFP	4.5-5.5V	0-20	0-33
3	P89C51RB2BA	16KB	512B	0~+70, PLCC	4.5-5.5V	0-20	0-33
4	P89C51RB2BBD	16KB	512B	0~+70, LQFP	4.5-5.5V	0-20	0-33
5	P89C51RC2BP	32KB	512B	0~+70, PDIP	4.5-5.5V	0-20	0-33
6	P89C51RC2BA	32KB	512B	0~+70, PLCC	4.5-5.5V	0-20	0-33

7	P89C51RC2FA	32KB	512B	-40~+85,PLCC	4.5-5.5V	0-20	0-33
8	P89C51RC2BBD	32KB	512B	0~+70,LQFP	4.5-5.5V	0-20	0-33
9	P89C51RC2FBD	32KB	512B	-40~+85,LQFP	4.5-5.5V	0-20	0-33
10	P89C51RD2BP	64KB	1 KB	0~+70,PDIP	4.5-5.5V	0-20	0-33
11	P89C51RD2BA	64KB	1 KB	0~+70,PLCC	4.5-5.5V	0-20	0-33
12	P89C51RD2BBD	64KB	1 KB	0~+70,LQFP	4.5-5.5V	0-20	0-33
13	P89C51RD2FA	64KB	1024B	-40~+85,PLCC	4.5-5.5V	0-20	0-33

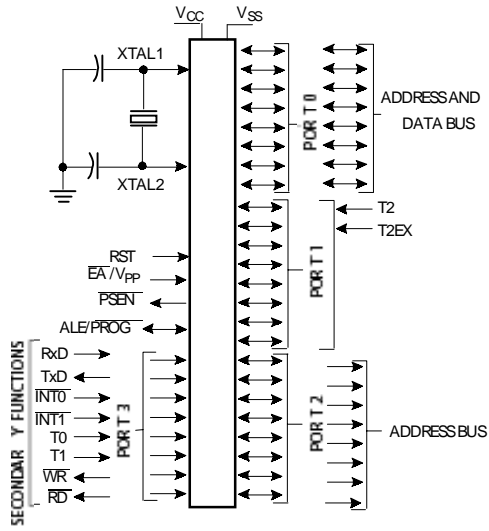
框图 1



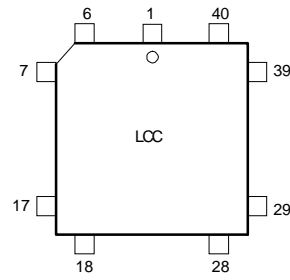
框图 2 (CPU)



逻辑符号



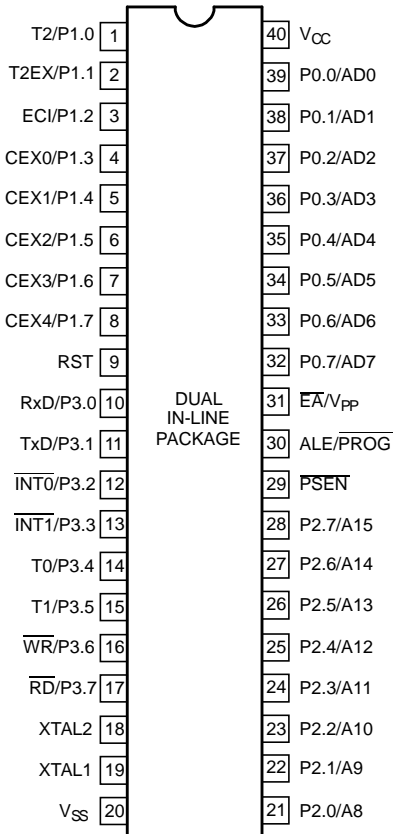
PLCC 管脚功能



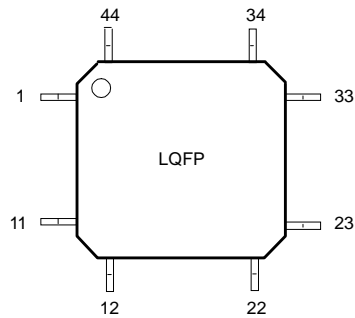
Pin	Function	Pin	Function	Pin	Function
1	NIC*	16	P3.4/T0	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1	32	PSEN
3	P1.1/T2EX	18	P3.6/WR	33	ALE/PROG
4	P1.2/ECI	19	P3.7/RD	34	NIC*
5	P1.3/CEX0	20	XTAL2	35	EA/Vpp
6	P1.4/CEX1	21	XTAL1	36	P0.7/AD7
7	P1.5/CEX2	22	Vss	37	P0.6/AD6
8	P1.6/CEX3	23	NIC*	38	P0.5/AD5
9	P1.7/CEX4	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/TxD	28	P2.4/A12	43	P0.0/AD0
14	P3.2/INT0	29	P2.5/A13	44	Vcc
15	P3.3/INTT	30	P2.6/A14		

* NO INTERNAL CONNECTION

DIP 管脚功能



LQFP 管脚功能



Pin	Function	Pin	Function	Pin	Function
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/INTT	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/PROG	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		

* NO INTERNAL CONNECTION

管脚描述

名称	管脚号			类型	名称和功能
	DIP	LCC	QFP		
V _{ss}	20	22	16	I	地: 0V 参考点
V _{cc}	40	44	38	I	电源: 提供掉电、空闲、正常工作电压
P0.0-0.7	39-32	43-36	37-30	I/O	P0 口: P0 口是开漏双向口, 可向其写入 1 使其状态为悬浮, 用作高阻输入。P0 也可以在访问外部程序存储器时作地址的低字节, 在访问外部数据存储器时作数据总线, 此时通过内部强上拉传送 1。
P1.0-1.7	1-8	2-9	40-44 1-3	I/O	P1 口: P1 口是带内部上拉的双向 I/O 口, 向 P1 口写入 1 时, P1 口被内部上拉为高电平, 可用作输入口。当作为输入脚时, 被外部拉低的 P1 口会因为内部上拉而输出电流(见 DC 电气特性)。P1 口第 2 功能:
	1	2	40	I/O	T2(P1.0): 定时/计数器 2 的外部计数输入/时钟输出
	2	3	41	I	T2EX(P1.1): 定时/计数器 2 重载/捕捉/方向控制
	3	4	42	I	ECI(P1.2): PCA 的外部时钟输入
	4	5	43	I/O	CEX0(P1.3): PCA 模块 0 捕获/比较模式的外部 I/O 管脚
	5	6	44	I/O	CEX1(P1.4): PCA 模块 1 捕获/比较模式的外部 I/O 管脚
	6	7	1	I/O	CEX2(P1.5): PCA 模块 2 捕获/比较模式的外部 I/O 管脚
	7	8	2	I/O	CEX3(P1.6): PCA 模块 3 捕获/比较模式的外部 I/O 管脚
	8	9	3	I/O	CEX4(P1.7): PCA 模块 4 捕获/比较模式的外部 I/O 管脚
P2.0-2.7	21-28	24-31	18-25	I/O	P2 口: P2 口是带内部上拉的双向 I/O 口, 向 P2 口写入 1 时, P2 口被内部上拉为高电平, 可用作输入口。当作为输入脚时, 被外部拉低的 P2 口会因为内部上拉而输出电流(见 DC 电气特性)。在访问外部程序存储器和外部数据时分别作为地址高位字节和 16 位地址(MOVX @DPTR), 此时通过内部强上拉传送 1。当使用 8 位寻址方式(MOV @Ri)访问外部数据存储器时, P2 口发送 P2 特殊功能寄存器的内容。P2.7 在编程/擦除时必须为“1”
P3.0-3.7	10-17	11, 13-19	5, 7-13	I/O	P3 口: P3 口是带内部上拉的双向 I/O 口, 向 P3 口写入 1 时, P3 口被内部上拉为高电平, 可用作输入口。当作为输入脚时, 被外部拉低的 P3 口会因为内部上拉而输出电流(见 DC 电气特性)。P89C51RX2 的 P3 口脚具有以下特殊功能:
	10	11	5	I	RxD(P3.0): 串行输入口
	11	13	7	O	TxD(P3.1): 串行输出口
	12	14	8	I	$\overline{\text{INT0}}$ (P3.2): 外部中断 0
	13	15	9	I	$\overline{\text{INT1}}$ (P3.3): 外部中断
	14	16	10	I	T0(P3.4): 定时器 0 外部输入
	15	17	11	I	T1(P3.5): 定时器 1 外部输入
	16	18	12	I	$\overline{\text{WR}}$ (P3.6): 外部数据存储器写信号
	17	19	13	I	$\overline{\text{RD}}$ (P3.7): 外部数据存储器读信号
RST	9	10	4	I	复位: 当晶振在运行中, 只要复位管脚出现 2 个机器周期高电平即可复位, 内部有扩散电阻连接到 V _{ss} , 仅需要外接一个电容到 V _{cc} 即可实现上电复位。

ALE	30	33	27	O	地址锁存使能：在访问外部存储器时，输出脉冲锁存地址的低字节，在正常情况下，ALE 输出信号恒定为 1/6 振荡频率。并可用作外部时钟或定时，注意每次访问外部数据时一个 ALE 脉冲将被忽略。ALE 可以通过置位 SFR auxiliary.0 禁止，置位后 ALE 只能在执行 MOVX 指令时被激活。
PSEN	29	32	27	O	程序存储使能：读外部程序存储。当从外部读取程序时，PSEN 每个机器周期被激活两次，在访问外部数据存储器 PSEN 无效，访问内部程序存储器时 PSEN 无效。
EA/Vpp	31	35	29	I	外部寻址使能/编程电压：在访问整个外部程序存储器时，EA 必须外部置低。如果 EA 为高时，将执行内部程序。当 RST 释放后 EA 脚的值被锁存，任何时序的改变都将无效。该引脚在对 FLASH 编程时用于输入编程电压(Vpp)。
XTAL1	19	21	15	I	晶体 1: 振荡反向放大器输入端和内部时钟发生电路输入端
XTAL2	18	20	14	O	晶体 2: 振荡反向放大器输出端

注：为了避免上电时的“latch-up”效应，任意管脚（Vpp 除外）上的电压最大不能高于 Vcc+0.5,最低不能低于 Vss- 0.5。

表 1 P89C51RX2 特殊功能寄存器

名称	定义	地址	位功能和位地址								复位值
			E7	E6	E5	E4	E3	E2	E1	E0	
ACC*	累加器	E0H									00H
AUXR#	辅助功能寄存器	8EH	-	-	-	-	-	-	EXTRAM	AO	xxxxxxx0B
AUXR1#	辅助功能寄存器 1	A2H	-	-	ENBOOT	-	GF2	0	-	DPS	02H ¹
B*	B 寄存器	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	模块 0 捕获高字节	FAH									xxxxxxxB
CCAP1H#	模块 1 捕获高字节	FBH									xxxxxxxB
CCAP2H#	模块 2 捕获高字节	FCH									xxxxxxxB
CCAP3H#	模块 3 捕获高字节	FDH									xxxxxxxB
CCAP4H#	模块 4 捕获高字节	FEH									xxxxxxxB
CCAP0L#	模块 0 捕获低字节	EAH									xxxxxxxB
CCAP1L#	模块 1 捕获低字节	EBH									xxxxxxxB
CCAP2L#	模块 2 捕获低字节	ECH									xxxxxxxB
CCAP3L#	模块 3 捕获低字节	EDH									xxxxxxxB
CCAP4L#	模块 4 捕获低字节	EEH									xxxxxxxB
CCAPM0#	模块 0 模式	DAH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x000000B
CCAPM1#	模块 1 模式	DBH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x000000B
CCAPM2#	模块 2 模式	DCH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x000000B
CCAPM3#	模块 3 模式	DDG	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x000000B
CCAPM4#	模块 4 模式	DEH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x000000B
			DF	DE	DD	DC	DB	DA	D9	D8	
CCON*#	PCA 计数器控制	D8H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00x00000B
CH#	PCA 计数器高字节	F9H									00H
CKCON#	时钟控制	8FH	-	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2	x000000B

CL#	PCA 计数器低字节	E9H									00H
CMOD#	PCA 计数器模式	D9H	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF	00xxx000B
DPTR:											
DPH	指针高字节	83H									00H
DPL	指针低字节	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	中断使能	A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
			BF	BE	BD	BC	BBB	BA	B9	B8	
IP*	中断优先级	B8H	—	PPC	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000B
			B7	B6	B5	B4	B3	B2	B1	B0	
IPH#	中断优先级高字节	B7H	—	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000B
			87	86	85	84	83	82	81	80	
P0*	P0 口	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	P1 口	90H	CEX4	CEX3	CEX2	CEX1	CEX0	ECI	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	P2 口	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	P3 口	B0H	\overline{RD}	\overline{WR}	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	RxD	FFH
PCON# ¹	电源控制寄存器	87H	SMOD1	SMOD0	—	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
RACAP2H#	定时器 2 捕获高字节	CBH									00H
RACAP2L#	定时器 2 捕获低字节	CAH									00H
SADDR#	从地址	A9H									00H
SADEN#	从地址屏蔽	B9H									00H
SBUF	串口数据缓冲区	99H									xxxxxxx0B
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	串行口控制	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	堆栈指针	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	定时器控制	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			8F	8E	8D	8C	8B	8A	89	88	
T2CON*	定时器 2 控制	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$\overline{CT2}$	$\overline{CP/RL2}$	00H
T2MOD#	定时器 2 模式控制	C9H	—	—	—	—	—	—	T2OE	DCEN	xxxxxx00B
TH0	定时器 0 高字节	8CH									00H
TH1	定时器 1 高字节	8DH									00H
TH2#	定时器 2 高字节	CDH									00H
TL0	定时器 0 低字节	8AH									00H
TL1	定时器 1 低字节	8BH									00H
TL2#	定时器 2 低字节	CCH									00H
TMOD	定时器模式	89H	GATE	\overline{CT}	M1	M0	GATE	\overline{CT}	M1	M0	00H

注：带“*”号的 SFR 可位寻址。

带“#”号的 SFR 表示从 80C51 的 SFR 修改而来或新增加的。

“—”表示保留位

1. 复位值取决于复位源

振荡器特性

XTAL1 和 XTAL2 分别作为一个反相放大器的输入和输出。此管脚可配置为使用内部振荡器。要使用外部时钟源驱动器件时，XTAL2 可以不连接而由 XTAL1 驱动。外部时钟信号无占空比的要求。但高低电平的最长和最短时间必须符合手册的规定。

该器件在出厂时配置为每机器周期 12 个时钟。参见“12 时钟模式”。可通过商用的 Flash 编程器或 ISP 或软件配置为每机器周期 6 个时钟，参见“6 时钟模式”。

时钟控制寄存器 (CKCON)

该器件支持通过 SFR 位 (X2) 或 Flash 配置位 (FX2, 位于保密 Flash 块中) 对 6 时钟/12 时钟模式进行配置。当 FX2 编程为 6 时钟模式时，X2 (CKCON.0) 将不起作用。CKCON 还支持单独对外围功能的时钟速率进行配置。当运行于 6 时钟模式时，外围功能可单独由 $f_{osc}/6$ 或 $f_{osc}/12$ 驱动。运行于 12 时钟模式时，则只能使用 $f_{osc}/12$ 。

CKCON 地址=8FH		复位值=x000 0000B					
不可位寻址							
7	6	5	4	3	2	1	0
—	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2
符号	功能						
—	保留位						
WDX2	看门狗时钟；0=6 时钟，1=12 时钟						
PCAX2	PCA 时钟；0=6 时钟，1=12 时钟						
SIX2	UART 时钟；0=6 时钟，1=12 时钟						
T2X2	定时器 2 时钟；0=6 时钟，1=12 时钟						
T1X2	定时器 1 时钟；0=6 时钟，1=12 时钟						
T0X2	定时器 0 时钟；0=6 时钟，1=12 时钟						
X2	CPU 时钟；1=6 时钟，0=12 时钟						

下面是外围功能时钟源的真值表：

FX2 时钟模式位	X2	外围功能时钟模式位 (例如 T0X2)	CPU 模式	外围功能时钟速率
擦除	0	x	12 时钟 (默认)	12 时钟 (默认)
擦除	1	0	6 时钟	6 时钟
擦除	1	1	6 时钟	12 时钟
编程	x	0	6 时钟	6 时钟
编程	x	1	6 时钟	12 时钟

复位

在振荡器工作时，将 RST 脚保持至少两个机器周期高电平 (6 时钟模式时为 12 个振荡器周期，12 时钟模式时为 24 个振荡器周期) 可实现复位。为了保证上电复位的可靠，RST 必须保持足够长时间的高电平该时间至少为振荡器的稳定时间 (通常为几个毫秒) 加上两个机器周期。上电时 V_{CC} 和 RST 必须同时上升

以实现正确的启动。当复位电压大于 $V_{IH}(min.)$ 时, I/O 口 1、2 和 3 异步复位。当 RST 撤除时, EA 的值被锁存。

低功耗模式

时钟停止模式

静态设计使时钟频率可以降至 0MHz(停止)。当振荡器停振时, RAM 和 SFR 的值保持不变。该模式允许逐步应用并可将时钟频率降至任意值以实现系统功耗的降低。如要实现最低功耗则建议使用掉电模式。

空闲模式

空闲模式(见表 2)中, CPU 进入睡眠状态, 但片内的外围电路仍然保持工作状态。正常操作模式的最后一条指令执行进入空闲模式。空闲模式下, CPU 内容、片内 RAM 和所有 SFR 保持原来的值。任何被使能的中断(此时, 程序从中断服务程序处恢复并继续执行)或硬件复位(与上电复位使用相同的方式启动处理器)均可终止空闲模式。

掉电模式

为了进一步降低功耗, 通过软件可实现掉电模式(见表 2)。该模式中, 振荡器停振并且在最后一条指令执行进入掉电模式。降到 2.0V 时, 片内 RAM 和 SFR 保持原值, 在退出掉电模式之前 V_{CC} 必须升至规定的最低操作电压。

硬件复位或外部中断均可结束掉电模式。硬件复位使 SFR 值重新设置, 但不改变片内 RAM 的值。外部中断允许 SFR 和片内 RAM 都保持原值。

要正确退出掉电模式, 在 V_{CC} 恢复到正常操作电压范围之后, 复位或外部中断开始执行并且要保持足够长的时间(通常小于 10ms)以使振荡器重新启动并稳定下来。

使用外部中断时, INT0 和 INT1 必须使能且配置为电平触发。将管脚电平拉低使振荡器重新启动, 退出掉电模式后将管脚恢复为高电平。一旦中断被响应, RETI 之后所执行的是进入掉电模式指令的后一条指令。

上电标志

当 V_{CC} 电压从 0V 上升到 5V 时, 片内电路将上电标志位(POF)置位。上电标志位可通过软件置位或清零, 用户可确定复位是由上电引起的还是由掉电唤醒引起的。当 V_{CC} 电压在 3V 以上时, POF 不受影响。

表 2 空闲模式和掉电模式时外部管脚的状态

模式	程序存储器	ALE	PSEN	口 0	口 1	口 2	口 3
空闲	内部	1	1	数据	数据	数据	数据
空闲	外部	1	1	悬浮	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	悬浮	数据	数据	数据

设计中的注意事项

当空闲模式被硬件复位所中止时, 器件在内部复位之前从停止处恢复程序正常运行, 时间为 2 个机器周期。这段时间内片内硬件禁止对内部 RAM 的访问, 但对 I/O 口的访问未被禁止。当 Idle 模式被复位所中止时, 为了消除可能产生的误写操作, 应用 Idle 模式指令后的指令不应执行写 I/O 口或写外部存储器操作。

ONCE™ 模式

ONCE(在线仿真)模式实现了对系统的测试和调试而不需要将器件从电路中移去。进入 ONCE 模式的条件:

1. 当器件复位且 \overline{PSEN} 为高电平时, 将 ALE 置低电平;
2. 在 RST 撤除时, ALE 保持低电平。

当器件处于 ONCE 模式时, P0 口处于悬浮状态, 其它 I/O 口、ALE 和 \overline{PSEN} 为弱上拉。振荡电路保

持工作状态，器件处于该模式时，可用仿真器或测试 CPU 驱动电路。执行正常复位时恢复正常操作。

可编程时钟输出

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

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

要将定时/计数器 2 配置为时钟发生器， $\overline{C/T2}$ (T2CON.1) 必须清零，T2MOD 中的 T2OE 位必须置位。启动定时器 2 必须将 TR2(T2CON.2) 置位。

时钟输出频率由振荡器频率和定时器 2 捕获寄存器的重新装入值确定，公式如下：

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

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

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

定时器 0 和 1 的操作

定时器 0 和 1

定时和计数功能由特殊功能寄存器 TMOD 的控制位 C/T 进行选择。这两个定时/计数器有 4 种操作模式，通过 TMOD 的 M1 和 M0 选择。两个定时/计数器的模式 0、1 和 2 都相同，模式 3 不同。如下所述：

模式 0

将定时器设置成模式 0 时类似 8048 定时器，即 8 位计数器带 32 分频的预分频器。图 2 所示为模式 0 工作方式。

此模式下，定时器寄存器配置为 13 位寄存器。当计数从全为“1”翻转为全为“0”时，定时器中断标志位 TF_n 置位。当 TR_n=1 同时 GATE=0 或 $\overline{\text{INTn}}=1$ 时定时器计数。置位 GATE 时允许由外部输入 $\overline{\text{INTn}}$ 控制定时器，这样可实现脉宽测量。TR_n 为 TCON 寄存器内的控制位（图 3）。

该 13 位寄存器包含 TH_n 全部 8 个位及 TL_n 的低 5 位。TL_n 的高 3 位不定，可将其忽略。置位运行标志 (TR_n) 不能清零此寄存器。

模式 0 的操作对于定时器 0 及定时器 1 都是相同的。两个不同的 GATE 位 (TMOD.7 和 TMOD.3) 分别分配给定时器 0 及定时器 1。

模式 1

模式 1 除了使用了 TH_n 及 TL_n 全部 16 位外，其它与模式 0 相同。

模式 2

此模式下定时器寄存器作为可自动重装的 8 位计数器 (TL_n)，如图 4 所示。TL_n 的溢出不仅置位 TF_n，而且将 TH_n 内容重新装入 TL_n，TH_n 内容由软件预置。重装时 TH_n 内容不变。模式 2 的操作对于定时器 0 及定时器 1 是相同的。

模式 3

在模式 3 中，定时器 1 停止计数，效果与将 TR1 设置为 0 相同。

此模式下定时器 0 的 TL0 及 TH0 作为两个独立的 8 位计数器。图 5 为模式 3 时的定时器 0 逻辑。TL0

占用定时器 0 的控制位：C/T, GATE, TR0, INT0 及 TF0。TH0 限定为定时器功能（计数器周期），占用定时器 1 的 TR1 及 TF1。此时 TH0 控制“定时器 1”中断。

模式 3 可用于需要一个额外的 8 位定时器的场合。定时器 0 工作于模式 3 时，80C51 看似有 3 个定时器/计数器，当定时器 0 工作于模式 3 时，定时器 1 可通过开关进入/退出模式 3,它仍可作为串行端口的波特率发生器，或者应用于任何不要求中断的场合。

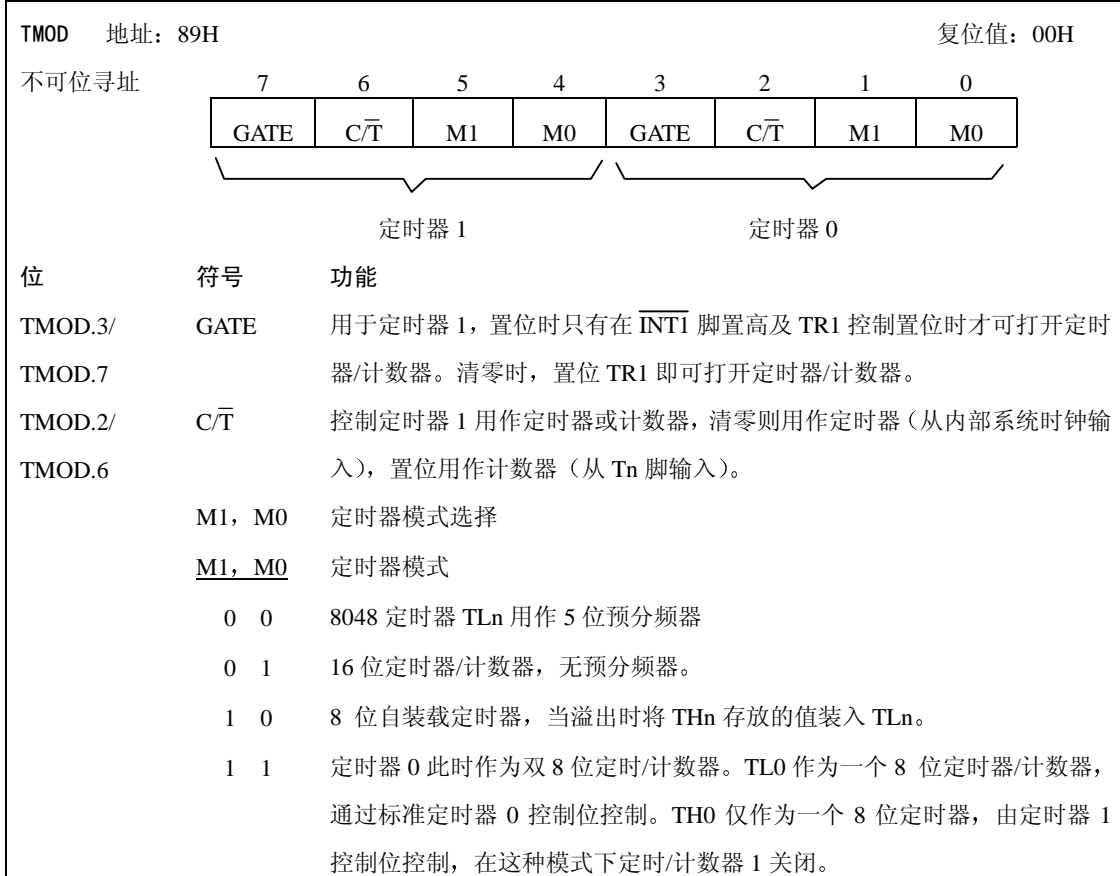


图 1 定时/计数器 0/1 模式控制寄存器 (TMOD)

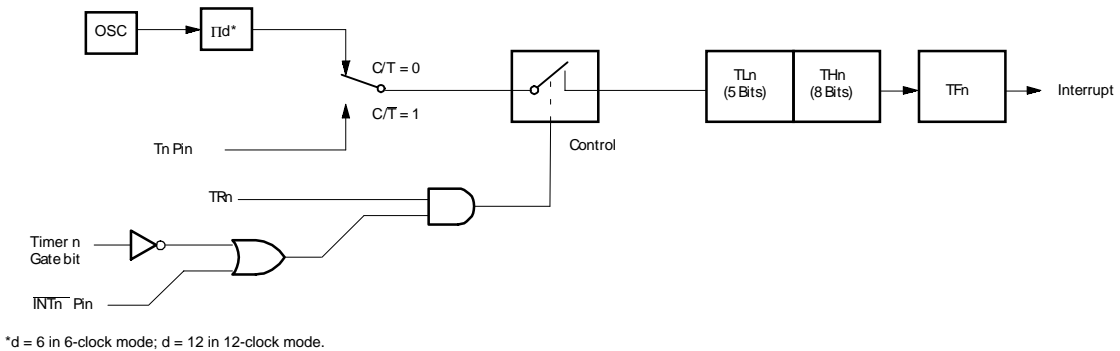
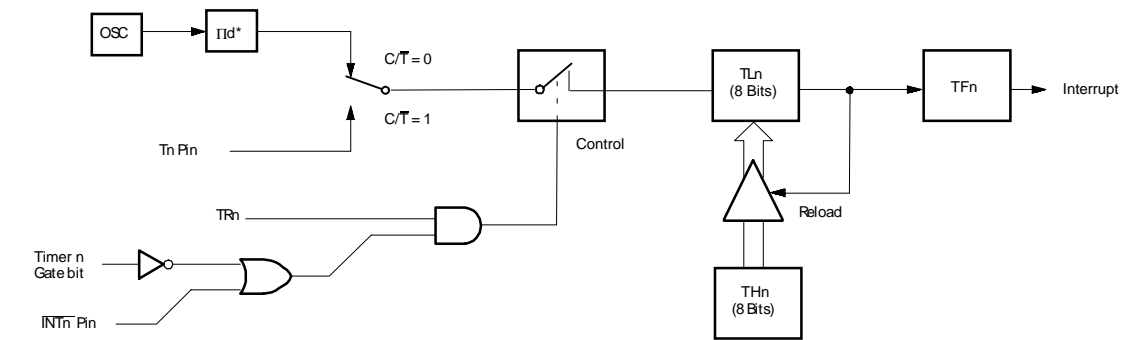


图 2 定时/计数器 0/1 的模式 0: 13 位定时/计数器

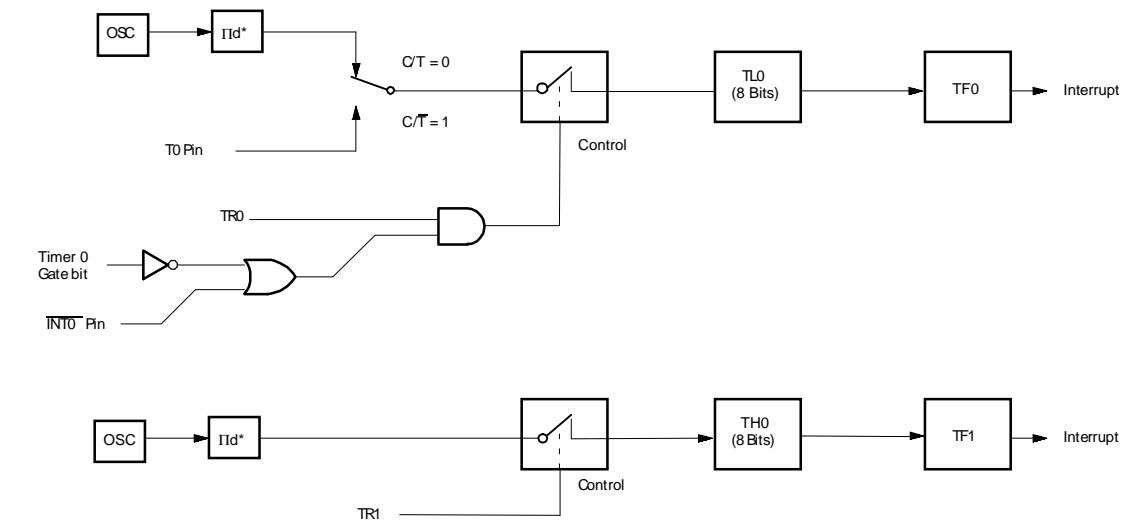
TCON 地址: 88H	
可位寻址	7 6 5 4 3 2 1 0
复位值: 00H	TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0
位	符号 功能
TCON.7	TF1 定时器 1 溢出标志。定时/计数器溢出时由硬件置位。中断处理时由硬件清除。或用软件清除。
TCON.6	TR1 定时器 1 运行控制位。由软件置位/清零将定时/计数器打开/关闭。
TCON.5	TF0 定时器 0 溢出标志。定时/计数器溢出时由硬件置位。中断处理时由硬件清除。或用软件清除。
TCON.4	TR0 定时器 0 运行控制位。由软件置位/清零将定时/计数器打开/关闭。
TCON.3	IE1 中断 1 边沿触发标志。当检测到外部中断 1 边沿时由硬件置位。中断处理时清零。
TCON.2	IT1 中断 1 触发类型控制位, 由软件置位/清零以选择外部中断以下降沿/低电平方式触发。
TCON.1	IE0 中断 0 边沿触发标志。当检测到外部中断 0 边沿时由硬件置位。中断处理时清零。
TCON.0	IT0 中断 0 触发类型控制位, 由软件置位/清零以选择外部中断以下降沿/低电平方式触发。

图 3 定时器/计数器控制寄存器 (TCON)



*d = 6 in 6-clock mode; d = 12 in 12-clock mode.

图 4 定时/计数器 0/1 的模式 2: 8 位自动重装



*d = 6 in 6-clock mode; d = 12 in 12-clock mode.

图 5 定时/计数器 0 的模式 3: 双 8 位计数器

定时器 2 的操作

定时器 2

定时器 2 是一个 16 位定时/计数器。通过设置特殊功能寄存器 T2CON 中的 $C/\overline{TR2}$ 位, 可将其作为定时器或计数器 (见图 6)。定时器 2 有三种操作模式: 捕获、自动重新装载 (递增或递减计数) 和波特率发生器, 这三种模式由 T2CON 中的位进行选择 (见表 3)。

捕获模式

在捕获模式中, 通过 T2CON 中的 EXEN2 设置两个选项。如果 EXEN2=0, 定时器 2 作为一个 16 位定时器或计数器 (由 T2CON 中 $C/\overline{TR2}$ 位选择), 溢出时置位 TF2 (定时器 2 溢出标志位)。该位可用于产生中断 (通过使能 IE 寄存器中的定时器 2 中断使能位)。如果 EXEN2=1, 与以上描述相同, 但增加了一个特性, 即外部输入 T2EX 由 1 变 0 时将定时器 2 中 TL2 和 TH2 的当前值各自捕获到 RCAP2L 和 RCAP2H。另外, T2EX 的负跳变使 T2CON 中的 EXF2 置位, EXF2 也象 TF2 一样能够产生中断 (其向量与定时器 2 溢出中断地址相同, 定时器 2 中断服务程序通过查询 TF2 和 EXF2 来确定引起中断的事件)。捕获模式如图 2 所示。在该模式中, TL2 和 TH2 无重新装载值。甚至当 T2EX 产生捕获事件时, 计数器仍以 T2EX 的负跳变或振荡频率的 1/12 (12 时钟模式) 或 1/6 (6 时钟模式) 计数。

自动重装模式 (递增/递减计数器)

16 位自动重装模式中, 定时器 2 可通过 $C/\overline{TR2}$ 配置为定时器/计数器, 编程控制递增/递减计数。计数的方向是由 DCEN (递减计数使能位) 确定的, DCEN 位于 T2MOD 寄存器 (见图 8) 中。当 DCEN=0 时, 定时器 2 默认为向上计数; 当 DCEN=1 时, 定时器 2 可通过 T2EX 确定递增或递减计数。

图 9 显示了当 DCEN=0 时, 定时器 2 自动递增计数。在该模式中通过设置 EXEN2 位进行选择。如果 EXEN2=0, 定时器 2 递增计数到 0FFFFH 并在溢出后将 TF2 置位, 然后将 RCAP2L 和 RCAP2H 中的 16 位值作为重新装载值装入定时器 2。RCAP2L 和 RCAP2H 的值是通过软件预设的。

如果 EXEN2=1, 16 位重新装载可通过溢出或 T2EX 从 1→0 的负跳变实现。此负跳变同时将 EXF2 置位。如果定时器 2 中断被使能, 则当 TF2 或 EXF2 置 1 时产生中断。

在图 10 中 DCEN=1 时, 定时器 2 可递增或递减计数。此模式允许 T2EX 控制计数的方向。当 T2EX 置 1 时, 定时器 2 递增计数, 计数到 0FFFFH 后溢出并置位 TF2。还将产生中断 (如果中断被使能), 定时器 2 的溢出将使 RCAP2L 和 RCAP2H 中的 16 位值作为重新装载值放入 TL2 和 TH2。

当 T2EX 置零时, 将使定时器 2 递减计数。当 TL2 和 TH2 计数到等于 RCAP2L 和 RCAP2H 时, 定时器产生溢出。定时器 2 溢出置位 TF2, 并将 0FFFFH 重新装入 TL2 和 TH2。

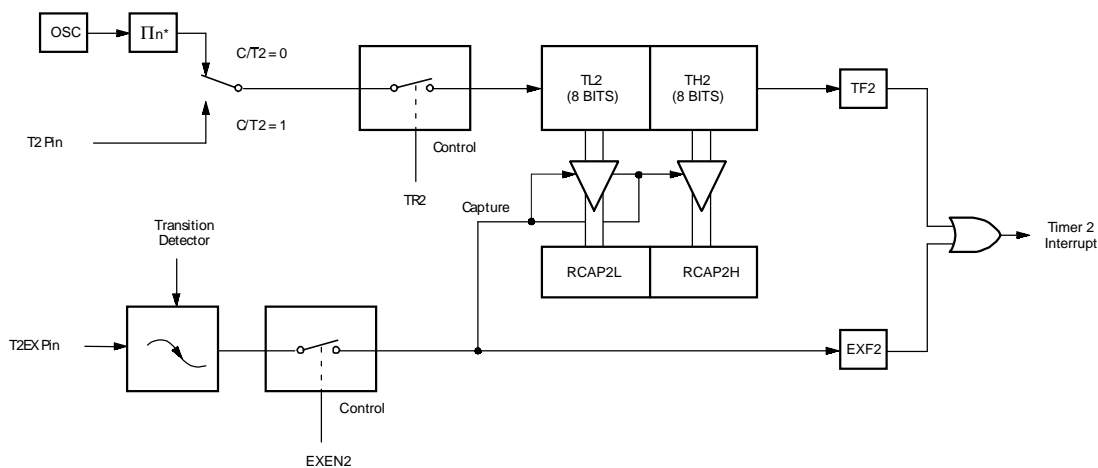
当定时器 2 递增/递减产生溢出时, 外部标志位 EXF2 翻转。如果需要, 可将 EXF2 位作为第 17 位。在此模式中, EXF2 标志不会产生中断。

表 4 定时器 2 工作方式

RCLK+TCLK	CP/ $\overline{RL2}$	TR2	模式
0	0	1	16 位自动重装
0	1	1	16 位捕获
1	X	1	波特率发生器
X	X	0	(关闭)

T2CON 地址=0C8H 可位寻址		复位值=00H						
	7	6	5	4	3	2	1	0
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
符号	位	名称和意义						
TF2	T2CON.7	定时器 2 溢出标志。定时器 2 溢出时置位，必须由软件清除。当 RCLK 或 TCLK=1 时，TF2 将不会置位。						
EXF2	T2CON.6	定时器 2 外部标志。当 EXEN2=1 且 T2EX 的负跳变产生捕获或重装时，EXF2 置位。定时器 2 中断使能时，EXF2=1 将使 CPU 从中断向量处执行定时器 2 中断子程序。EXF2 位必须用软件清零。在递增/递减计数器模式 (DCEN=1) 中，EXF2 不会引起中断。						
RCLK	T2CON.5	接收时钟标志。RCLK 置位时，定时器 2 的溢出脉冲作为串行口模式 1 和模式 3 的接收时钟。RCLK=0 时，将定时器 1 的溢出脉冲作为接收时钟。						
TCLK	T2CON.4	发送时钟标志。TCLK 置位时，定时器 2 的溢出脉冲作为串行口模式 1 和模式 3 的发送时钟。TCLK=0 时，将定时器 1 的溢出脉冲作为发送时钟。						
EXEN2	T2CON.3	定时器 2 外部使能标志。当其置位且定时器 2 未作为串行口时钟时，允许 T2EX 的负跳变产生捕获或重装。EXEN2=0 时，T2EX 的跳变对定时器 2 无效。						
TR2	T2CON.2	定时器 2 启动/停止控制位。置 1 时启动定时器。						
C/T2	T2CON.1	定时器/计数器选择。(定时器 2) 0=内部定时器 (OSC/12 或 OSC/6) 1=外部事件计数器 (下降沿触发)						
CP/RL2	T2CON.0	捕获/重装标志。置位：EXEN2=1 时 T2EX 的负跳变产生捕获。清零：EXEN2=1 时定时器 2 溢出或 T2EX 的负跳变都可使定时器自动重装。当 RCLK=1 或 TCLK=1 时，该位无效且定时器强制为溢出时自动重装。						

图 6 定时器/计数器 2 (T2CON) 控制寄存器



* n = 6 in 6-clock mode, or 12 in 12-clock mode.

图 7 定时器 2 捕获模式

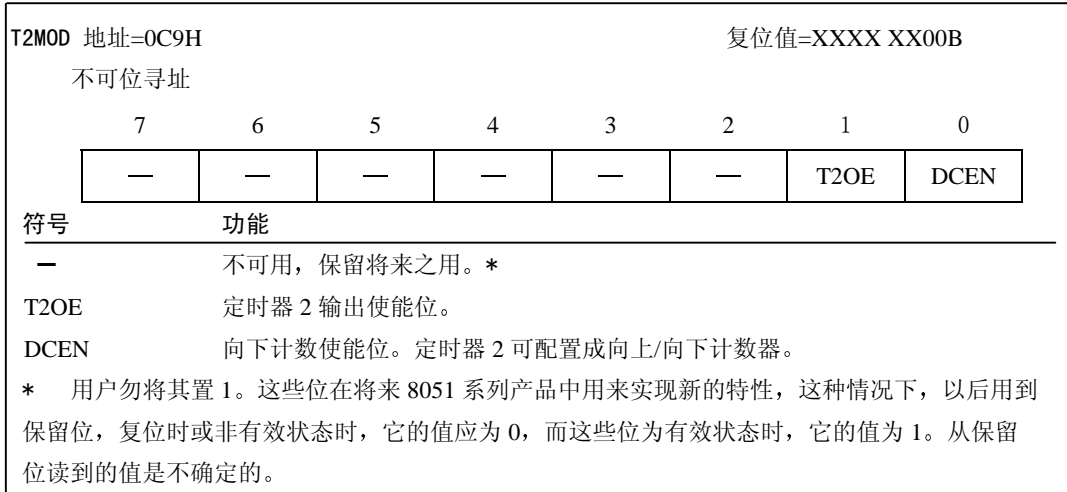
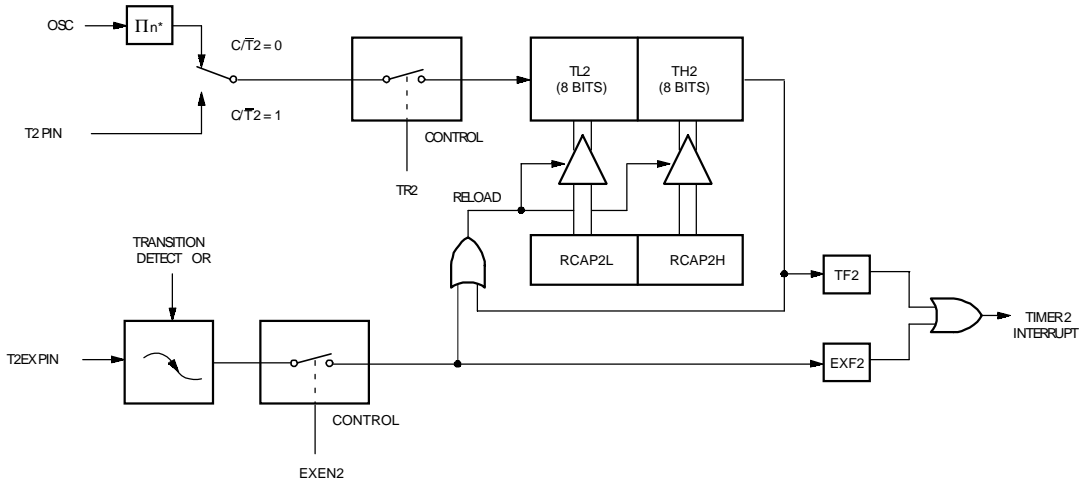
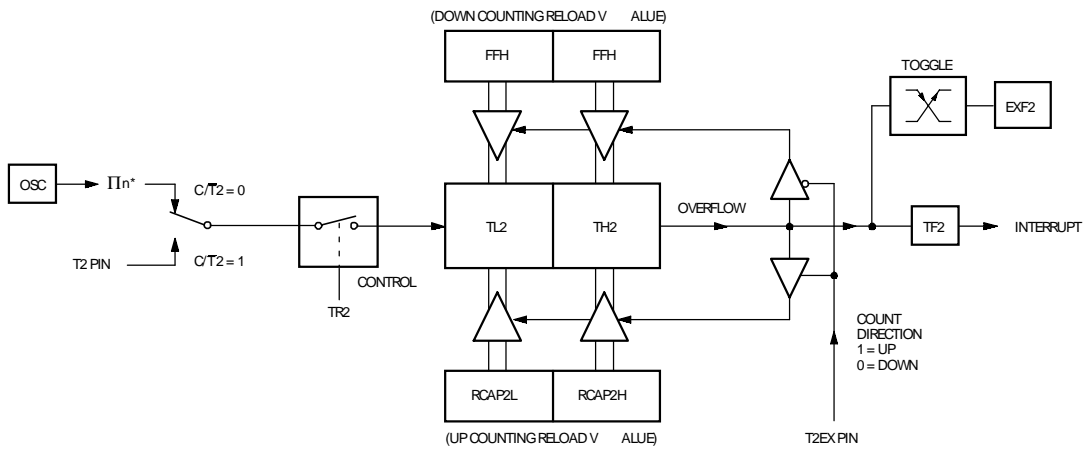


图 8 定时器 2 模式 (T2MOD) 控制寄存器



* n = 6 in 6-clock mode, or 12 in 12-clock mode.

图 9 定时器 2 自动重装模式 (DCEN=0)



* n = 6 in 6-clock mode, or 12 in 12-clock mode.

图 10 定时器 2 自动重装模式 (DCEN=1)

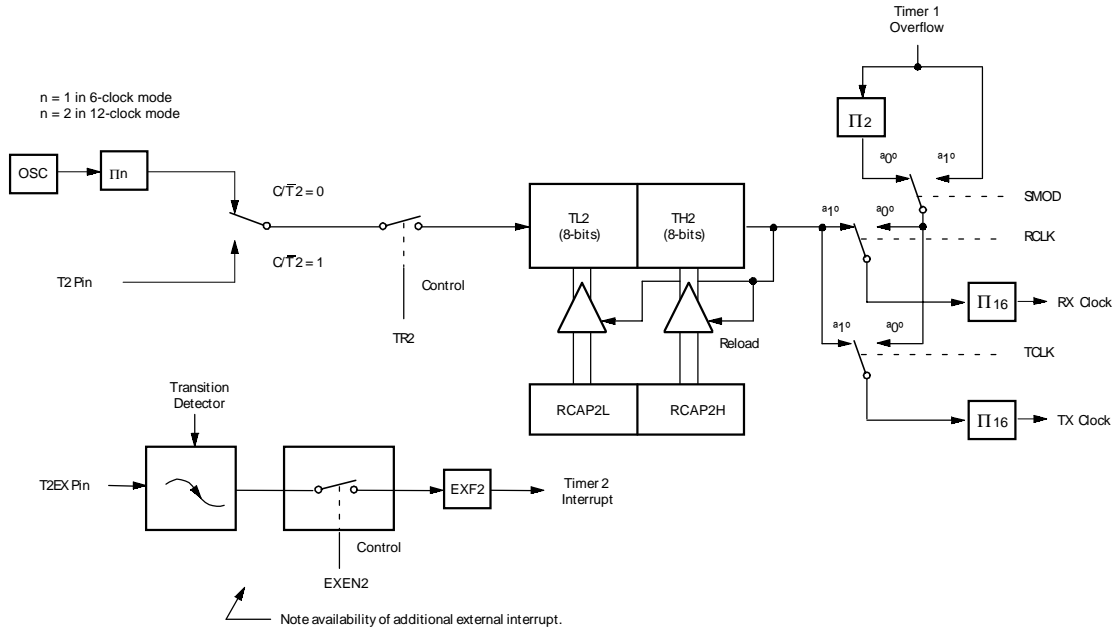


图 11 定时器 2 波特率发生器模式

波特率发生器模式

寄存器 T2CON 的位 TCLK 和（或）RCLK（见表 3）允许从定时器 1 或定时器 2 获得串行口发送和接收的波特率。当 TCLK=0 时，定时器 1 作为串行口发送波特率发生器；当 TCLK=1 时，定时器 2 作为串行口发送波特率发生器。RCLK 对串行口接收波特率有同样的作用。通过这两位，串行口能得到不同的接收和发送波特率——一个通过定时器 1 产生，另一个通过定时器 2 产生。

图 11 所示为定时器 2 工作在波特率发生器模式。与自动重装模式相似，当 TH2 溢出时，波特率发生器模式使定时器 2 寄存器重新装载来自寄存器 RCAP2H 和 RCAP2L 的 16 位的值，寄存器 RCAP2H 和 RCAP2LR 的值由软件预置。

当工作于模式 1 和模式 3 时，波特率由下面给出的定时器 2 溢出率所决定：

$$\text{模式 1 和模式 3 的波特率} = \frac{\text{定时器 2 溢出速率}}{16}$$

定时器可配置成“定时”或“计数”方式，在许多应用上，定时器被设置在“定时”方式（C/T2*=0）。当定时器 2 作为定时器时，它的操作不同于波特率发生器。

通常，定时器 2 作为定时器，它会在每个机器周期递增（1/6 或 1/12 振荡频率）。当定时器 2 作为波特率发生器时，它在 6 时钟模式下，以振荡器频率递增（12 时钟模式时为 1/2 振荡频率）。这样，波特率公式如下：

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

此处：n = 16（6 时钟模式）或 32（12 时钟模式）

RCAP2H,RCAP2L)=RCAP2H 和 RCAP2L 的内容，为 16 位无符号整数。

如图 11 所示，定时器 2 作为波特率发生器，仅当寄存器 T2CON 中的 RCLK 和（或）TCLK=1 时，定时器 2 作为波特率发生器才有效。注意 TH2 溢出并不置位 TF2，也不产生中断。这样，当定时器 2 作为波特率发生器时，定时器 2 中断不必被禁止。如果 EXEN2（T2 外部使能标志）被置位，在 T2EX 中，由 1 到 0 的转换会置位 EXF2（T2 外部标志位），但并不导致（TH2，TL2）重装（RCAP2H，RCAP2L）。因此，当定时器 2 用作波特率发生器时，如果需要，T2EX 可用作附加的外部中断。

当定时器工作在波特率发生器模式下，则不要对 TH2 和 TL2 进行读写，每隔一个状态时间（0sc/2）或由 T2 进入的异步信号，定时器 2 将加 1；在此情况下对 TH2 和 TH1 进行读写是不准确的。可对 RCAP2 寄存器进行读，但不要进行写，否则将导致自动重装错误。当对定时器 2 或寄存器 RCAP 进行访问时，应关闭定

时器（清零 TR2）。

表 5 列出了常用的波特率和如何用定时器 2 得到这些波特率。

表 5 由定时器 2 产生的常用波特率

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	9.6K	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

波特率公式汇总

定时器 2 工作在波特率发生器模式，外部时钟信号由 T2 脚进入，波特率为：

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

如果定时器 2 采用内部时钟信号，则波特率为：

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

此处：n = 32（12 时钟模式）或 16（6 时钟模式）

f_{osc} = 振荡器频率

自动重装值可由下式得到：

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

定时器/计数器 2 的设置

除了波特率发生器模式，T2CON 不包括 TR2 位的设置，TR2 位需单独设置来启动定时器。表 6，表 7 给出了 T2 作为定时器和计数器的设置。

表 6 T2 作为定时器

模式	T2CON	
	内部控制（注 1）	外部控制（注 2）
16 位重装	00H	08H
16 位捕获	01H	09H
波特率发生器接收和发送相同波特率	34H	36H
只接收	24H	26H
只发送	14H	16H

表 7 T2 作为计数器

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

注：1. 仅当定时器溢出时进行捕获和重装

2. 当定时/计数器溢出并且 T2EX(P1.1)发生电平负跳变时产生捕获和重装（定时器 2 用于波特率发生器模式时除外）。

全双工增强型 UART

标准 UART 操作

串口为全双工结构，表示可以同时发送和接收。它还具有接收缓冲，在第一个字节从寄存器读出之前，可以开始接收第二个字节。（但是如果第二个字节接收完毕时第一个字节仍未读出，其中一个字节将会丢失。串口的发送和接收寄存器都是通过 SFR SBUF 进行访问的。写入 SBUF 的数据装入发送寄存器，对 SBUF 的读操作是对物理上分开的接收寄存器进行访问。

串口有 4 种操作模式：

模式 0

串行数据通过 RxD 进出。TxD 输出时钟。每次发送或接收以 LSB（最低位）作首位，每次 8 位。波特率固定为 MCU 时钟频率的 1/12(12 时钟模式)或 1/6(6 时钟模式)。

模式 1

TxD 脚发送，RxD 脚接收，每次数据为 10 位，一个起始位（0），8 个数据位（LSB 在前）及一个停止位（1）。当接收数据时，停止位存于 SCON 的 RB8 内，波特率可变，由定时器 1 溢出速率决定。

模式 2

TxD 脚发送，RxD 脚接收，每次数据为 11 位，一个起始位（0），8 个数据位（LSB 在前），一个可编程第 9 位数据及一个停止位（1）。

发送时，第 9 个数据位（SCON 内 TB8 位）可置为 0 或 1。例如将奇偶位（PSW 内 P 位）移至 TB8。接收时，第 9 位数据存入 SCON 的 RB8 位，停止位忽略。波特率可编程为 MCU 时钟频率的 1/32 或 1/64（12 时钟模式）（6 时钟模式下为 1/16 或 1/32）。

模式 3

TxD 脚发送，RxD 脚接收，每次数据为 11 位，一个起始位（0），8 个数据位（LSB 为首位），一可编程的第 9 位数据及一个停止位（1）。事实上模式 3 除了波特率外均与模式 2 相同。其波特率可变并由定时器 1 溢出率决定。

在上述 4 种模式中，发送过程是以任意一条以写 SBUF 作为目标寄存器的指令开始的，模式 0 时接收通过设置 R1=0 及 REN=1 初始化，其它模式下如若 REN=1 则通过起始位初始化。

多机通信

UART 模式 2 及模式 3 有一个专门的应用领域即多机通信。在这些模式时，接收为 9 位数据。第 9 位存入 RB8。接下来为停止位。UART 可编程为：接收到停止位时，仅当 RB8=1 时串口中断才有效。可通过置位 SCON 内 SM2 位来选择这一特性。下述为多机系统利用这一特性的一种方法。

当主机需要发送一数据块给数台从机之一时，首先发送出一个地址字节对目标从机进行识别。地址与数据字节通过第 9 位数据区别，其中地址字节的第 9 位为 1，而数据字节为 0。SM2=1 时，数据字节不会使各从机产生中断，而地址字节则令所有从机中断，这样各从机可以检查接收到的数据判断是否被寻址。被寻址的从机即可清除 SM2 位以准备接收随后数据内容。未被寻址的从机的 SM2 位仍为 1 则不理睬随后数据继续各自工作。

模式 0 时 SM2 无效，模式 1 时 SM2 用于检验停止位是否有效。在模式 1 时，如果 SM2=1，那么只有接收到有效的结束位才可产生接收中断。

串行端口控制寄存器（SCON）

串行端口控制及状态寄存器即 SCON，如图 12 所示，其中包括模式选择位，以及发送和接收的第 9

位数据 (TB8 及 RB8), 以及串行端口中断位 (TI 及 RI)。

SCON		地址: 98H							
可位寻址		7	6	5	4	3	2	1	0
复位值:	00H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
位	符号	功能							
SCON.7	FE	帧错误位。当检测到一个无效停止位时, 通过 UART 接收器设置该位, 但它必须由软件清零。要使该位有效, PCON 寄存器中的 SMOD0 位必须置 1。							
SCON.7	SM0	和 SM1 定义串口操作模式。要使该位有效, PCON 寄存器中的 SMOD0 必须置 0。							
SCON.6	SM1	和 SM0 定义串行口操作模式 (见下表)							
		<u>SM0</u>	<u>SM1</u>	<u>UART 模式</u>	<u>波特率</u>				
		0	0	0: 同步移位寄存器	fosc/12 (12 时钟模式) 或 fosc/6 (6 时钟模式)				
		0	1	1: 8 位 UART	可变				
		1	0	2: 9 位 UART	fosc /64 或 fosc /32(12 时钟); fosc /32 或 fosc /16(6 时钟模式)				
		1	1	3: 9 位 UART	可变				
SCON.5	SM2	在模式 2 和 3 中多处理机通信使能位。在模式 2 或 3 中, 若 SM2=1, 且接收到的第 9 位数据 (RB8) 是 0, 则 RI (接收中断标志) 不会被激活。在模式 1 中, 若 SM2=1 且没有接收到有效的停止位, 则 RI 不会被激活。在模式 0 中, SM2 必须是 0。							
SCON.4	REN	允许接收位。由软件置位或清除。REN=1 时, 允许接收, REN=0 时, 禁止接收。							
SCON.3	TB8	模式 2 和 3 中发送的第 9 位数据, 可以按需要由软件置位或清除。							
SCON.2	RB8	模式 2 和 3 中已接收的第 9 位数据, 在模式 1 中, 或 sm2=0, RB8 是已接收的停止位。在模式 0 中, RB8 未用。							
SCON.1	TI	发送中断标志。模式 0 中。在发送完第 8 位数据时, 由硬件置位。其它模式中, 在发送停止位之初, 由硬件置位。在任何模式中, 都必须由软件来清除 TI。							
SCON.0	RI	接收中断标志, 模式 0 中, 接收第 8 位结束时由硬件置位。其它模式中, 在接收停止位的中间时刻, 由硬件置位。在任何模式(SM2 所述情况除外)必须由软件清除 RI。							

图 12 串行控制寄存器 (SCON)

波特率

操作模式 0 的波特率是固定的, 为 fosc/12 (12 时钟模式) 或 fosc/6 (6 时钟模式)。模式 2 的波特率取决于 PCON 寄存器中的 SMOD1 位的值。在 12 时钟模式下, 若 SMOD1=0 (复位后的值), 波特率为振荡器频率/64, 若 SMOD1=1, 波特率为振荡器频率/32; 在 6 时钟模式下则分别为振荡器频率/32 或振荡器频率/16)。

$$\text{模式2波特率} = \frac{2^{\text{SMOD}}}{n} \times (\text{振荡器频率})$$

此处: n=64 (12 时钟模式) 或 32 (6 时钟模式)

在 80C51 中, 模式 1 和模式 3 的波特率由定时器 1 的溢出速率决定。

使用定时器 1 作波特率发生器

当定时器 1 用作波特率发生器, 模式 1 和 3 中波特率由定时器 1 的溢出速率和 SMOD1 的值决定。

$$\text{模式1,3波特率} = \frac{2^{\text{SMOD}}}{n} \times (\text{定时器1溢出速率})$$

此处: n=32 (12 时钟模式) 或 16 (6 时钟模式)

在此应用中定时器 1 不能用作中断, 定时器 1 可以工作在定时或计数方式和 3 种工作模式中任何一个。在最典型应用中, 它用作定时器方式工作自动重装载模式 (TMOD 的高半字节为 0010B), 它的波特率值由下式给出:

$$\text{模式1,3波特率} = \frac{2^{\text{SMOD}}}{n} \times \frac{\text{振荡器频率}}{12 \times [256 - (\text{TH1})]}$$

此处: n=32 (12 时钟模式) 或 16 (6 时钟模式)

可以定时器 1 的中断实现非常低的波特率。将定时器配置为 16 位定时器 (TMOD 的高半字节为 0001B), 并使用中断进行 16 位软件重装。图 13 列出了几个常用的波特率以及如何从定时器 1 获得。

模式	波特率		f _{osc}	SMOD	定时器1		
	12-clock 模式	6-clock 模式			C/T	模式	重装载值
模式 0 Max	1.67 MHz	3.34 MHz	20 MHz	X	X	X	X
模式 2 Max	625 k	1250 k	20 MHz	1	X	X	X
模式 1,3 Max	104.2 k	208.4 k	20 MHz	1	0	2	FFH
模式 1,3	19.2 k	38.4 k	11.059 MHz	1	0	2	FDH
	9.6 k	19.2 k	11.059 MHz	0	0	2	FDH
	4.8 k	9.6 k	11.059 MHz	0	0	2	FAH
	2.4 k	4.8 k	11.059 MHz	0	0	2	F4H
	1.2 k	2.4 k	11.059 MHz	0	0	2	E8H
	137.5	275	11.986 MHz	0	0	2	1DH
	110	220	6 MHz	0	0	2	72H
	110	220	12 MHz	0	0	1	FE8H

图 13 由定时器 1 产生的通用波特率

UART 模式 0

串行数据由 RxD 端出入。TxD 输出同步移位时钟, 发送或接收的是 8 位数据, 低位在先, 其波特率固定为 MCU 时钟的 1/12(12 时钟模式)或 1/6(6 时钟模式), 图 14 是串行口模式 0 的功能方框简图及相关的时序图。

执行任何一条把 SBUF 作为目的寄存器的指令时, 就开始发送。S6P2 时刻的“写 SBUF”信号将 1 装入发送移位寄存器的第 9 位, 并通知发送控制部分开始发送。写 SBUF 信号有效后一个完整的机器周期后 SEND 端有效。

SEND 使能 RxD (P3.0) 端送出数据, TxD (P3.1) 输出移位时钟。每个机器周期的 S3、S4 及 S5 状态内移位时钟为低电平, 而 S6、S1 及 S2 状态内为高。在 SEND 有效时, 每一机器周期的 S6P2 时刻发送移位寄存器的内容右移一位。

数据位向右移时, 左边添加零。当数据字节最高位 (MSB) 移到移位寄存器的输出端时, 其左边是装入“1”的第 9 位, 再左的内容均为 0, 此时通知 Tx 控制模块进行最后一位移位处理后禁止 SEND 并置位 TI, 所有这些步骤均在“写入 SBUF”后第 10 个机器周期的 S1P1 时进行的。接收初始化条件是 REN=1 及 RI=0。下一机器周期的 S6P2 时, RX 控制单元向接收移位寄存器写入 1111 1110 并在下一个时钟使 RECEIVE 端有效。

RECEIVE 使能移位时钟转换 P3.1 功能, 移位时钟在每个机器周期的 S3P1 及 S6P1 跳变。在 RECEIVE 有效时每一机器周期的 S6P2 时刻, 接收移位寄存器内容向左移一位。从右移位进来的值是该机器周期 S5P2 时从 P3.0 脚上采样得来的。

数据从右边移入时, 左边移出为“1”。当初始时置入最右端的“0”移至最左端时, 通知 RX 控制时钟作最后一次移位后装入 SBUF。在写入 SCON 清除 RI 后第 10 个机器周期, RECEIVE 端被清除且置位 RI。

UART 模式 1

串行口工作于模式 1 时, 传输的是 10 位: 1 位起始位 (0), 8 位数据 (低位在先) 及一位停止位 (1)。由 RxD 接收, TxD 发送。接收时, 停止位存入 SCON 内 RB8。80C51 波特率取决于定时器 1 的溢出速率。图 15 所示为串行口模式 1 的功能简图及相应的发送/接收时序。

发送过程是由执行一条以 SBUF 为目的寄存器的指令启动的。“写 SBUF”信号还把 1 (TB8) 装入发送移位寄存器的第 9 位, 同时通知发送控制器进行发送。实际上发送过程开始于 16 分频计数器下次翻转后的那个机器周期的 S1P1 时刻。每位的发送时序与 16 分频计数器同步, 而并不与“写 SBUF”信号同步。

发送以激活 SEND 端开始, 向 TxD 发送一起始位。一位 (时间) 以后 DATA 端有效, 使输出移位寄存器中数据得以送至 TxD。再过一位, 产生第一个移位脉冲。

数据向右移出, 左边不断填以 0, 当数据字节的最高位移到移位寄存器的输出位置时, 其左边是装入“1”的第 9 位, 再左的内容均为 0。此时通知 TX 控制器作最后一次移位, 然后禁止 SEND 端并置位 TI。这都发生于“写 SBUF”后 16 分频计时器的第 10 次翻转时。

接收在 RxD 端检测到负跳变时启动, 为此 MCU 对 RxD 不断采样, 采速率为波特率的 16 倍。当检测到负跳变时, 16 分频计数器立即复位, 同时将 1FFH 写入输入移位寄存器。复位 16 分频计时器确保计时器翻转时位与输入数据位时间同步。

计数器的 16 个状态将每个位时间分为 16 份。在第 7、8、9 状态时, 位检测器对 RxD 端的值采样。取值为三个采样值中取多数 (至少 2 个) 作为读入值, 这样可以抑制噪声。如果所接收的第一位不为 0, 说明它不是一帧数据的起始位, 该位被摒弃, 接收电路被复位, 等待另一个负跳变的到来。这用来防止错误的起始位。如果起始位有效, 则被移入输入移位寄存器, 并开始接收这一帧中的其它位。

当数据位逐一由右边移入时, “1”从左边被移出。当起始位 0 移到最左边时 (模式 1 为 9 位寄存器), 通知接收控制器进行最后一次移位, 将移位寄存器内容 (9) 位分别装入 SBUF 及 RB8, 并置 RI=1。仅当最后一位移位脉冲产生时同时满足下述 2 个条件: ①RI=0, ②SM2=0 或接收到的停止位=1, 才会装载 SBUF 和 RB8, 并且置位 RI。

上述两个条件任一不满足, 所接收到的数据帧就会丢失, 不再恢复。两者都满足时, 停止位就进入 RB8, 8 位数据进入 SBUF, RI=1。这时, 无论上述条件满足与否, 接收控制单元都会重新等待 RxD 的负跳变。

模式 2 和模式 3

模式 2 和 3 中, 发送 (通过 TxD) 和接收 (通过 RxD) 都是 11 位, 包括 1 位起始位 (0), 8 位数据位 (LSB 在先), 1 位可编程数据位 (第 9 位) 及一位停止位 (1)。发送时, 第 9 位数据位 (TB8) 可置为 0 或 1。接收时, 第 9 位存入 SCON 的 RB8。模式 2 时波特率可编程选为 MCU 时钟频率的 1/32 或 1/64 (12 时钟模式) 或 1/16 或 1/32 (6 时钟模式)。模式 3 时可由定时器 1 获取可变的波特率。

图 16 及 17 所示为模式 2、3 时串行口的功能简图。接收部分与模式 1 相同。发送部分仅发送移位寄存器内第 9 位和模式 1 有所不同。

发送过程是由执行一条以 SBUF 为目的寄存器的指令启动的。“写 SBUF”同时将 TB8 装入发送移位寄存器的第 9 位位置上。并通知发送控制器进行一次发送。发送过程由于 16 分频计数器下一次翻转后机器周期的 S1P1 时刻开始。

发送过程由使能 SEND 有效开始, 将一个起始位送到 TxD 端。一位时间后, DATA 有效, 数据由移位寄存器送入 TxD 端。再过一位后产生第一个移位脉冲。第一个移位时钟将“1” (停止位) 送入移位寄存器的第 9 位, 此后每次移位只把 0 送入第 9 位, 所以当数据位向右移出时, “0”从左边移入。当 TB8 移至输出位置上时, 它左边就是停止位, 其余位均为零。此时将通知发送控制器作最后一次移位, 然后使 SEND 无效并置位 TI。这些均发生在“写 SBUF”后第 11 次计数器翻转时, MCU 以 16 倍波特率对 RxD 脚进行采样, 一旦检测到负跳变, 16 分频计数器立即复位同时将 1FFH 写入输出移位寄存器。

在每一位的第 7、8、9 状态时, 位检测器对 RxD 端值进行采样。对三个采样值取多数 (至少 2 次) 为确定值以抑制噪声。如若所接收的第一位不为 0, 接收电路复位, 单元等待下一个负跳变的出现。如果起始位有效, 则被移入输入移位寄存器, 并开始接收这一帧中的其它位。

数据位从右边移入, “1”从左边移出。当起始位移至寄存器 (模式 2~3 时为 9 位寄存器) 的最左端时, 通知接收控制器进行最后一次移位, 并装入 SBUF 及 RB8 并置位 RI。仅当产生最后一位移位脉冲时同时满足下列 2 个条件: ①RI=0, ②SM2=0 或接收到的第 9 位数据为 1 时, 才装载 SBUF 和 RB8 并置位。

上述两个条件任一不满足, 所接收到的数据帧就会丢失不再恢复, RI 仍为 0。当两者都满足时, 第 9

位数据位就装入 RB8，前 8 位数据则装入 SBUF，一个位时间后，无论上述条件满足与否，单元都会重新等待 RxD 端的负跳变。

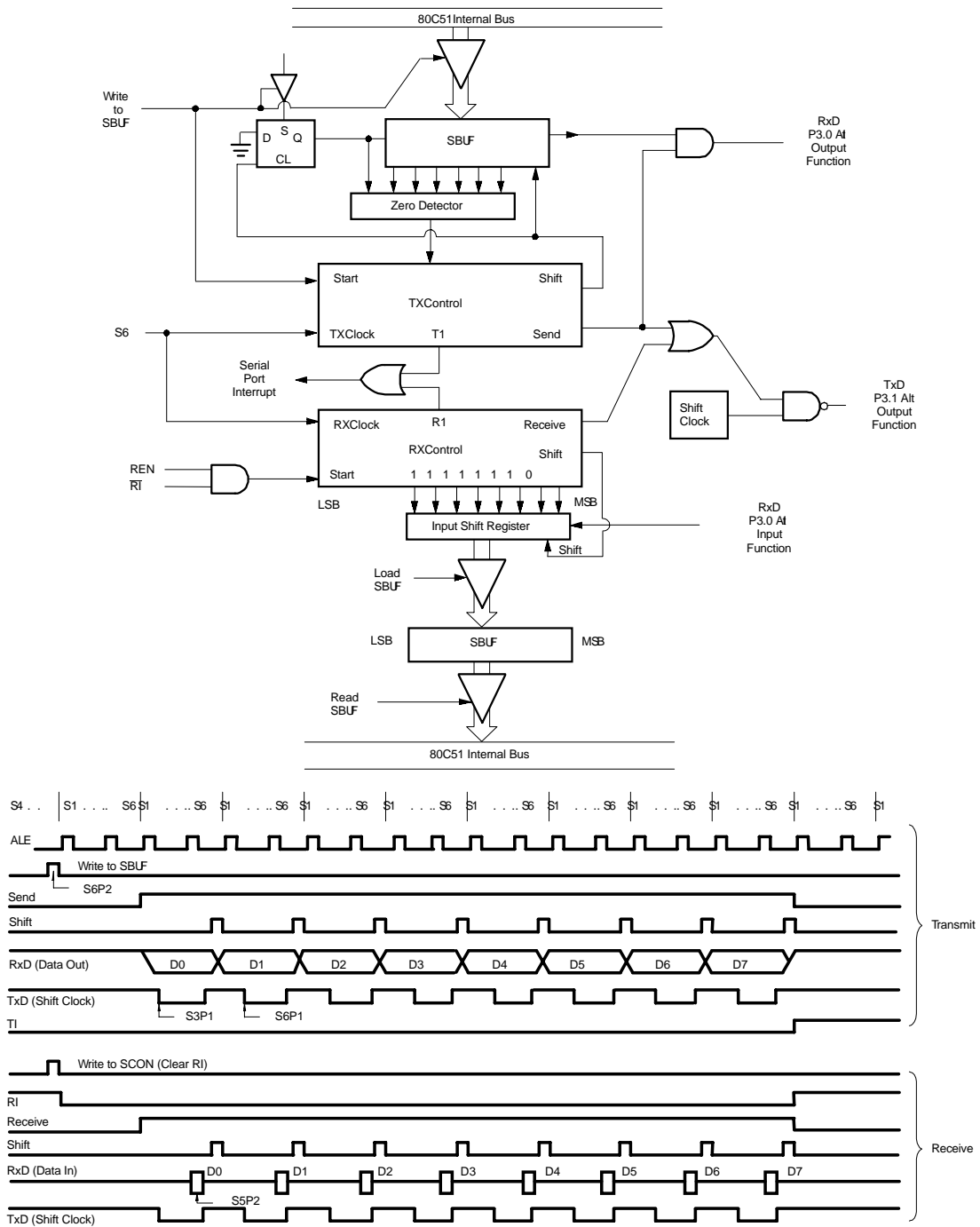


图 14 串口模式 0

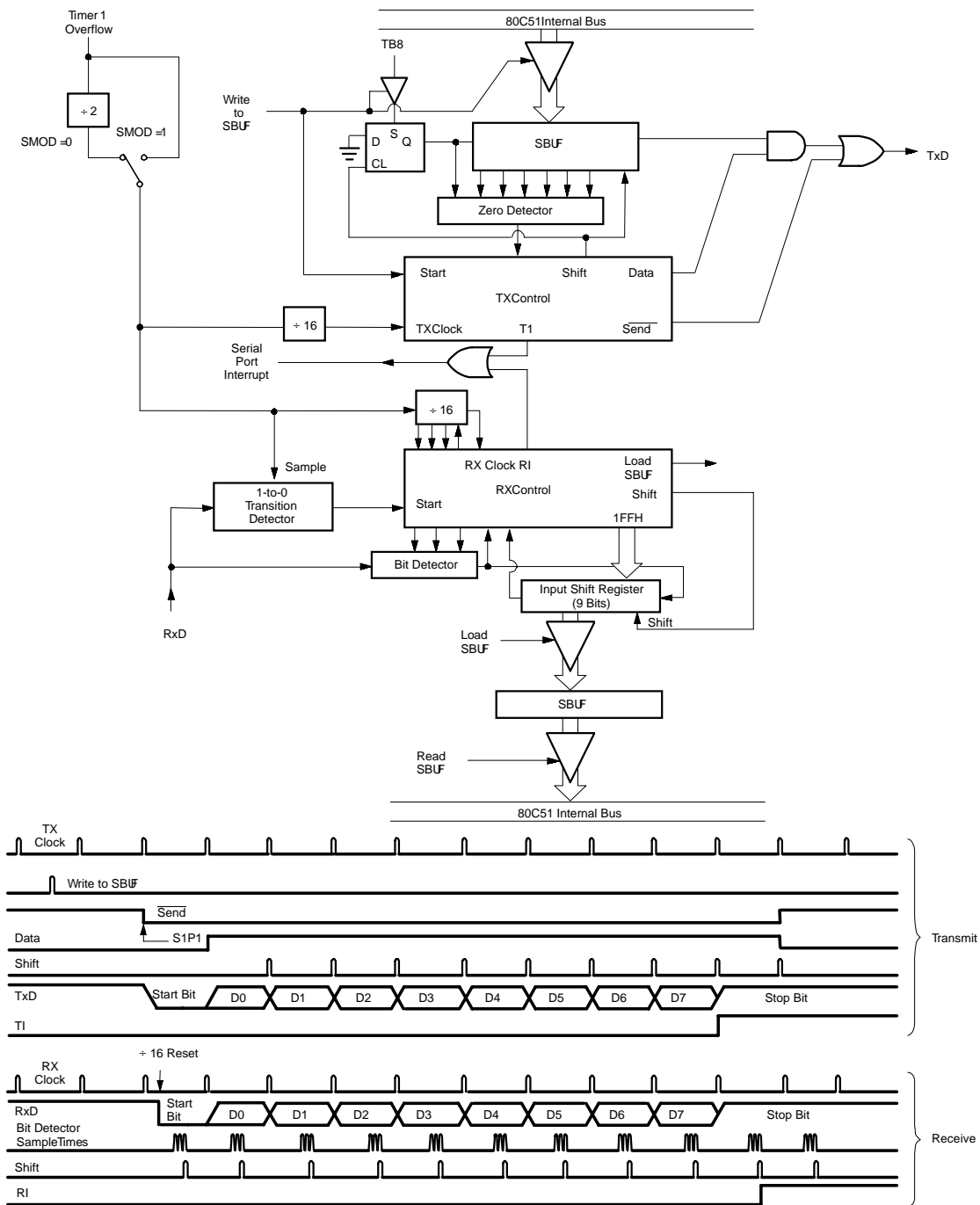


图 15 串口模式 1

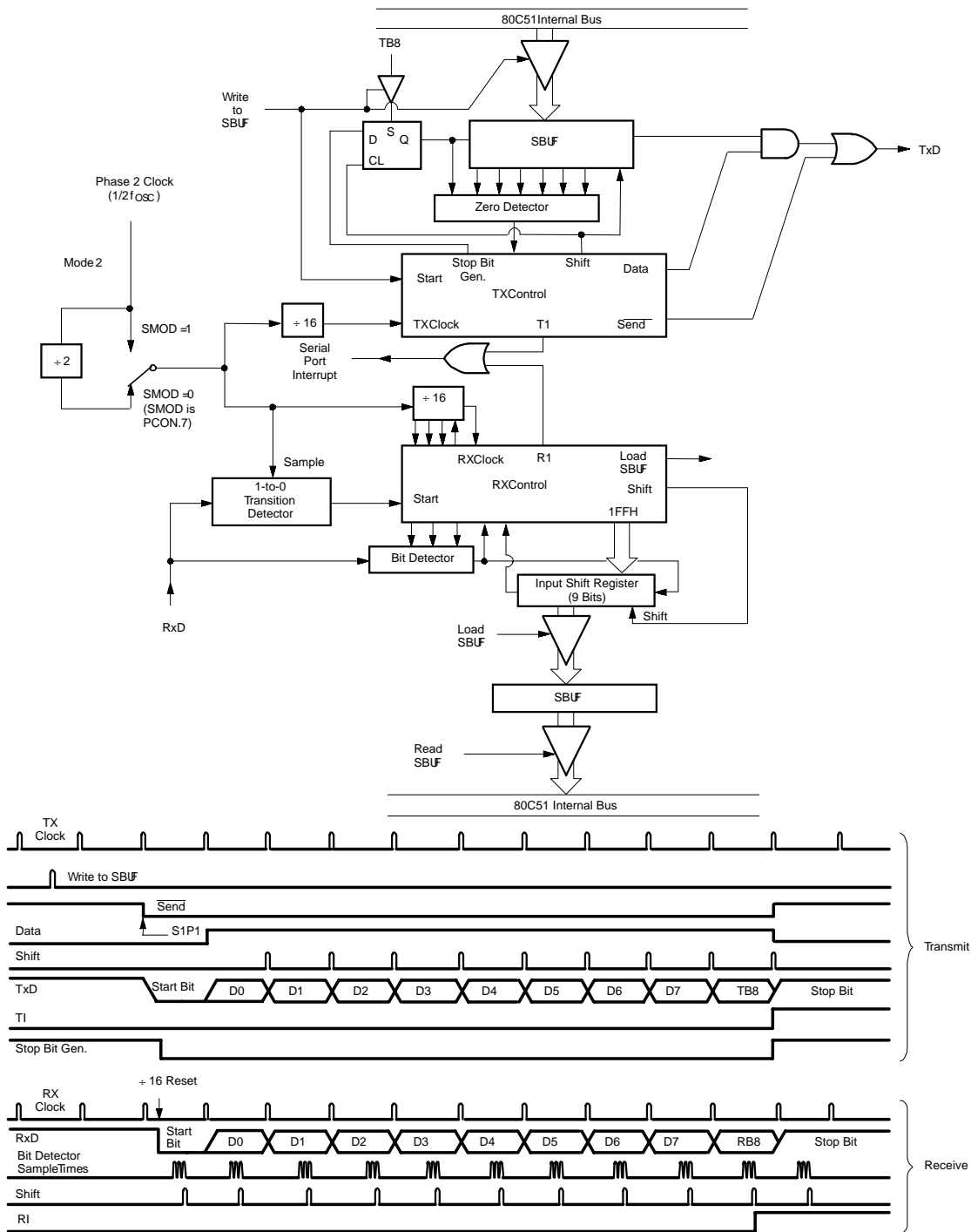


图 16 串口模式 2

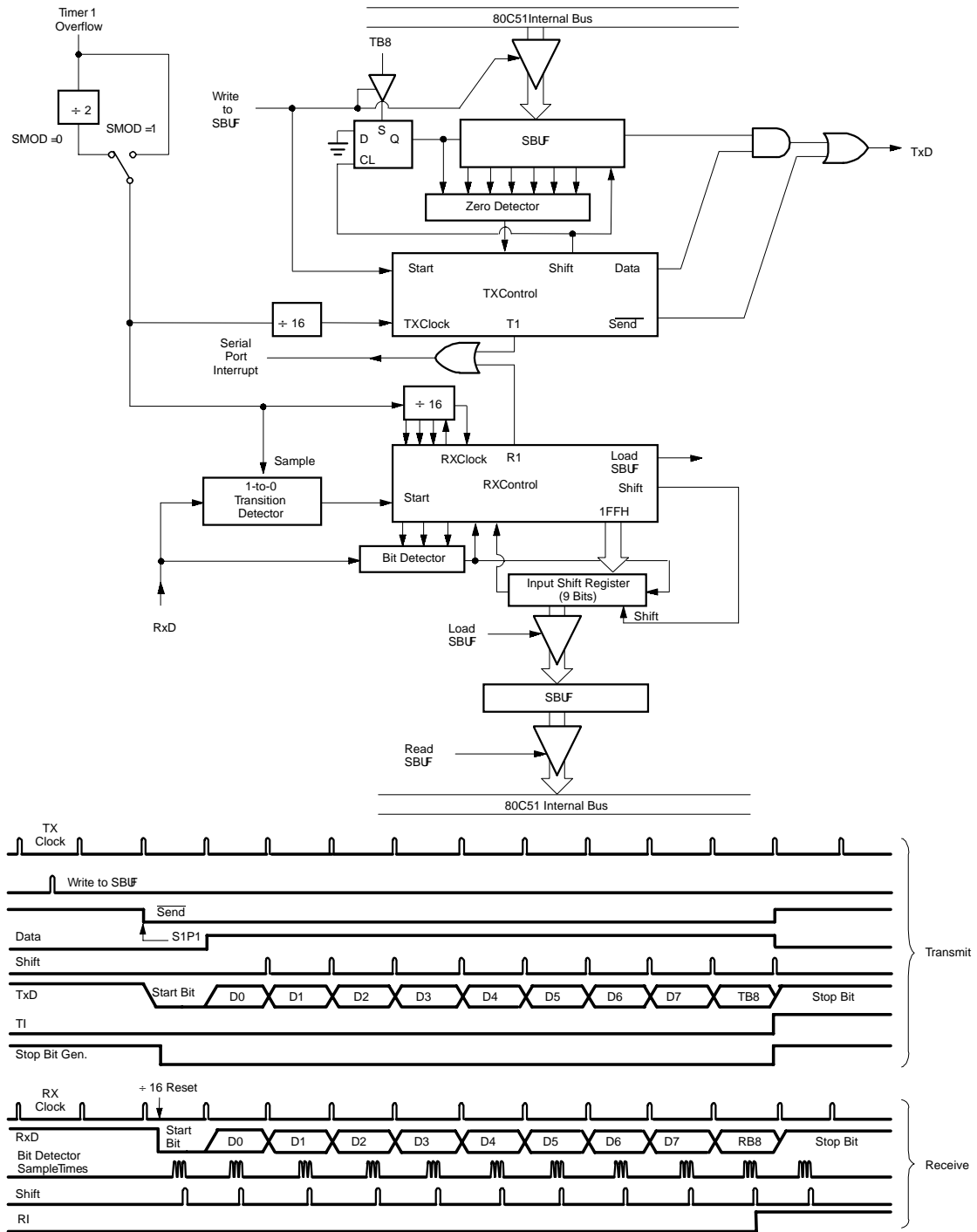


图 17 串口模式 3

增强型 UART 操作

除了标准操作模式外，UART 可实现自动地址识别和通过查询丢失的停止位进行帧错误检测。UART 还支持多机通信。

当使用帧错误检测时，丢失的位将会置位 SCON 中的 FE 位。FE 与 SM0 共用 SCON.7，通过 PCON.6 (SMOD0) 选择。如果 SMOD0 置位，SCON.7 作为 FE，SMOD0 为 0 时，SCON.7 作为 SM0。作为 FE 时，SCON.7 只能由软件清零（见图 18）。

自动地址识别

自动地址识别是这样一种特性，它使 UART 可以通过硬件比较从串行数据流中识别出特定的地址。这样就不必花费大量软件资源去检查每一个从串口输入的串行地址。将 SCON 内 SM2 置位可启用该特性。在 9 位 UART 模式（模式 2 和模式 3）下，如果接收的字节中包含“给定”地址或“广播”地址，接收中断标志（RI）将自动置位。在 9 位模式下要求第 9 个信息位为 1 以表明该信息内容是地址而非数据。

使用自动地址识别特性时，主机通过调用特定从机地址选择与一个（或多个）从机通信。使用广播地址时，所有从机都被联系。在此使用了两个特殊功能寄存器：SADDR 表示从机地址，SADEN 表示地址屏蔽。SADEN 用于定义 SADDR 内哪几位需使用而哪几位不予考虑。SADEN 可以与 SADDR 逻辑“与”得出给定的地址，用于对每一从机进行寻址。示例如下：

```
从机 0          SADDR=1100 0000
                SADEN=1111 1101
                特定地址=1100 00X0
```

```
从机 1          SADDR=1100 0000
                SADEN=1111 1110
                特定地址=1100 000X
```

上例中 SADDR 相同，而 SADEN 不同以区分两个从机。从机 0 要求 0 位为 0 而忽略 1 位。从机 1 则要求 1 位为 0 而忽略 0 位。由于从机 1 的 1 位必须为 0，从机 0 只能取独有的地址 1100 0010 以区别。由于从机 0 的 0 位必须为 1，从机 1 只能取独有的地址 1100 0001 以区别。而取地址 1100 0000 时两从机都可被寻址。

下例所示为选择从机 1、2 而不选从机 0：

```
从机 0          SADDR=1100 0000
                SADEN=1111 1001
                特定地址=1100 0XX0
```

```
从机 1          SADDR=1110 0000
                SADEN=1111 1010
                特定地址=1100 0X0X
```

```
从机 2          SADDR=1110 0000
                SADEN=1111 1100
                特定地址=1110 00XX
```

上述三个从地址只有低 3 位不同。从机 0 要求位 0=0，它可通过 1110 0110 单独寻址；从机 1 要求位 1=0，可通过 1110 0101 单独寻址；从机 2 要求位 2 为 0，可通过 1110 0011 单独寻址。由于必须使地址字节的第 2 位为“1”以屏蔽从机 2，因此使用地址 1110 0100 可选通从机 0 和 1 同时屏蔽从机 2。将 SADDR 和 SADEN 相“或”后产生每个从机的“广播”地址，结果为零的位视为无关位。大多数情况下，无关位被认为是 1，这样，“广播”地址为 FFH。复位时 SADDR 和 SADEN 均为 00H，此时产生了一个所有位都是无关位的给定地址，也即“广播”地址。这样有效地禁止了自动寻址模式，并允许微控制器使用不带有上述特性的标准 UART 驱动器。

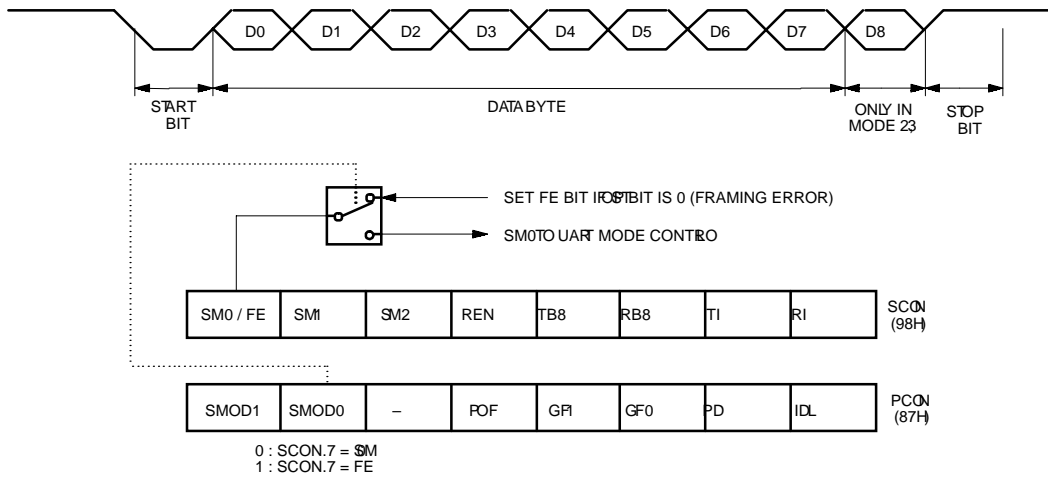


图 18 UART 帧错误检测

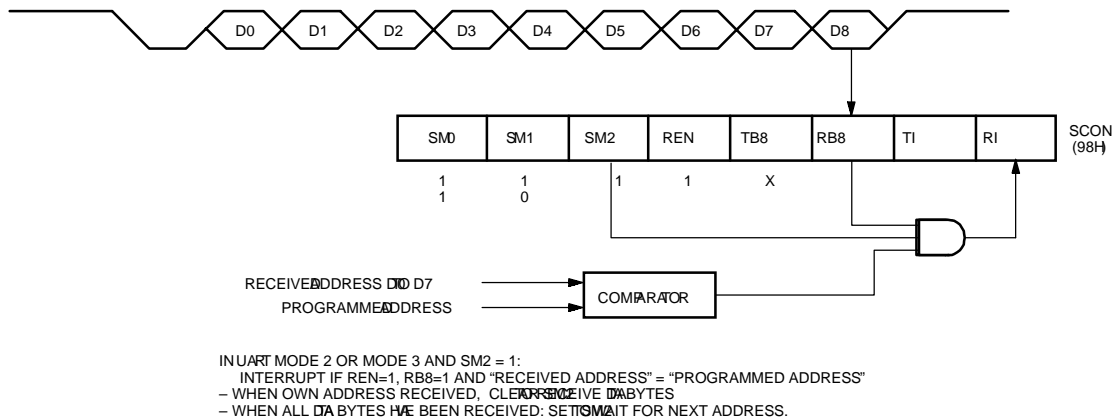


图 19 UART 多机通信,自动地址识别

中断优先级结构

P89C51RA2/RB2/RC2/RD2xx 有 7 个中断源, 4 个中断优先级。

中断优先级和 3 个特殊功能寄存器相关, 他们分别是 IE, IP 和 IPH(见图 20、21、22)。

寄存器 IPH (中断优先级高) 组成 4 级中断结构, IPH 的地址位于 SFR 中的 B7H。IPH 寄存器及其位的描述见图 22。

IPH 寄存器的功能很简单, IPH 和 IP 组合使用决定每一个中断的优先级。如下表所示:

优先级位		中断优先级
IPH. x	IP. x	
0	0	0 级(最低级)
0	1	1 级
1	0	2 级
1	1	3 级(最高级)

表 7 中断表

中断源	优先顺序	请求位	硬件清除	入口地址
X0	1	IE0	N(L) ¹ Y(T) ²	03H
T0	2	TP0	Y	0BH
X1	3	IE1	N(L) Y(T)	13H
T1	4	TF1	Y	1BH
PCA	5	CF,CCFn(n=0-4)	N	33H
SP	6	R1,T1	N	23H
T2	7	TF2,EXF2	N	2BH

注:

1. L= 电平激活
2. T= 翻转激活

IE(0A8H)		7	6	5	4	3	2	1	0
		EA	EC	ET2	ES	ET1	EX1	ET0	EX0
		使能位=1, 使能中断							
		使能位=0, 禁止中断							
位	标号	功能							
IE.7	EA	全局禁止位。如果 EA=0, 禁止所有中断, 如果 EA=1, 通过置位或清除使能位, 对应的每个中断被使能或禁止。							
IE.6	EC	PCA 中断使能位							
IE.5	ET2	定时器 2 中断使能位							
IE.4	ES	串行口中断使能位							
IE.3	ET1	定时器 1 中断使能位							
IE.2	EX1	外部中断 1 使能位							
IE.1	ET0	定时器 0 中断使能位							
IE.0	EX0	外部中断 0 使能位							

图 20 IE 特殊功能寄存器

IP(0B8H)		7	6	5	4	3	2	1	0
		—	—	PT2	PS	PT1	PX1	PT0	PX0
		中断优先级控制位=1, 定义为高优先级中断							
		中断优先级控制位=0, 定义为低优先级中断							
位	标号	功能							
IP.7	—	—							
IP.6	PPC	PCA 中断优先级控制位							
IP.5	PT2	定时器 2 中断优先级控制位							
IP.4	PS	串行口中断优先级控制位							
IP.3	PT1	定时器 1 中断优先级控制位							
IP.2	PX1	外部中断 1 中断优先级控制位							
IP.1	PT0	定时器 0 中断优先级控制位							
IP.0	PX0	外部中断 0 中断优先级控制位							

图 21 IP 寄存器

IPH(B7H)							
7	6	5	4	3	2	1	0
—	—	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
中断优先级控制位=1, 定义为高优先级中断							
中断优先级控制位=0, 定义为低优先级中断							
位	标号	功能					
IPH.7	—	—					
IPH.6	PPCH	PCA 中断优先级控制位高					
IPH.5	PT2H	定时器 2 中断优先级控制位高					
IPH.4	PSH	串行口中断优先级控制位高					
IPH.3	PT1H	定时器 1 中断优先级控制位高					
IPH.2	PX1H	外部中断 1 中断优先级控制位高					
IPH.1	PT0H	定时器 0 中断优先级控制位高					
IPH.0	PX0H	外部中断 0 中断优先级控制位高					

图 22 IPH 寄存器

4 个中断级比 80C51 多 2 个。在没有产生同级的中断和更高级的中断情况下，中断将被执行，如果同级的中断或更高级的中断正在执行，新的中断只有等到正在执行的中断结束才能被执行。在更低级的中断正在执行情况下产生新的中断时，低级的中断停止，转而执行新的中断，直到新中断完成才可以执行被停止的中断。

降低 EMI

所有口控制输出时都有转换率。这限制了由于输出信号的快速切换所产生的 EMI。该转换率由工厂设定，大约为 10ns 的上升和下降时间。

降低 EMI 模式

当位 AO (AUXR.0) 置位时，禁止 ALE 输出。

AUXR (8EH)

7	6	5	4	3	2	1	0
—	—	—	—	—	—	EXTRAM	AO

AUXR.1 EXTRAM

AUXR.0 AO

详见图 37。

双 DPTR 结构

双 DPTR 结构 (如图 23) 提供了一种用于寻址外部数据存储器的方法。有两个 16 位 DPTR 寄存器可以寻址外部存储器。通过对 AUXR1 的 DPS 位编程可实现两个 DPTR 寄存器的切换。

- 新寄存器名: AUXR1#
- SFR 的地址: A2H
- 复位值: xxx000x0B

AUXR1 (A2H)

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

此处: DPS 为 AUXR1 的位 0, 用于切换指针 DPTR0 和 DPTR1

选择寄存器	DPS
DPTR0	0
DPTR1	1

当切换 DPTR0 和 DPTR1 时, 应当通过软件来保存 DPS。

GF2 是一个由用户定义的标志位。注意 AURX 的位 2 不能写, 而读出值为 0。通过执行 INC DPTR 指令, 能对 DPS 快速切换, 且不会影响 GF2 位。

ENBOOT 位决定 BOOTROM 是否使能: 如果复位时状态字节非零或 PSEN 被拉低, ALE 悬浮为高且在复位下降沿时 EA>VIH, ENBOOT 位自动置位。

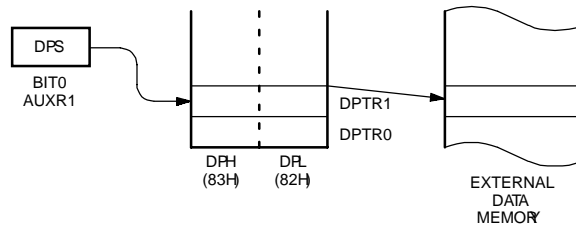


图 23

DPTR 指令

DPTR 指令根据当前 AUXR1 位 0 的值, 可作为数据指针。下面是使用 DPTR 的 6 个指令:

INC	DPTR	数据指针加 1
MOV	DPTR,#data16	DPTR 装载 16 位常数
MOV	A,@A+DPTR	将与 DPTR 相关的代码字节送入 ACC
MOVX	A,@DPTR	外部 RAM (16 位地址) 的内容装入 ACC
MOVX	@DPTR,A	把 ACC 的内容送到外部 RAM (16 位地址)
JMP	@A+DPTR	间接跳转到与 DPTR 相关的地址

可以通过寻址 SFR 的低字节或高字节来寻址数据指针。更详细的内容可参见应用指南 AN458。

可编程计数器阵列 (PCA)

P89C51RA2/RB2/RC2/RD2xx 的可编程计数器阵列是由五个 16 位捕获/比较模块组成的特殊定时器。每个模块都可以经编程实现下列多种模式: 捕获模式, 软件定时器模式, 高速输出模式和 PWM 脉宽调制模式。每个模块都有一个 P1 口的管脚与之对应。例如, 模块 0 连接到 P1.3 (CEX0), 模块 1 连接到 P1.4 (CEX1)。PCA 的基本配置见图 24。

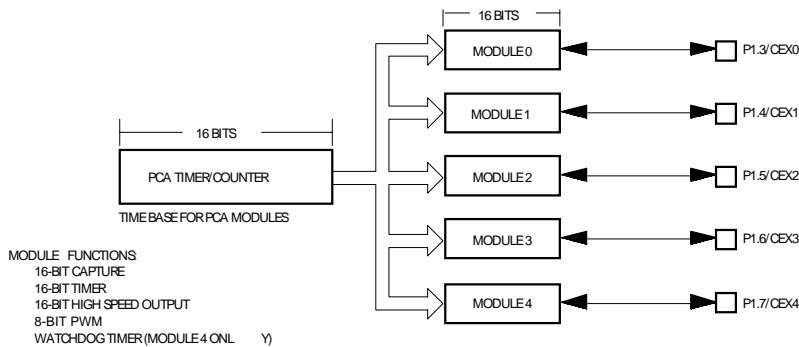


图 24 可编程计数器阵列 (PCA)

PCA 定时器所有 5 个模块有一个公共的时基, 并可编程为: 1/6 振荡器频率, 1/2 振荡器频率, 定时器 0 溢出或者 ECI 脚 (P1.2) 的输入。定时器的计数源由 CMOD 中的 CPS1 和 CPS0 决定 (见图 27)。

CPS1	CPS0	PCA定时器计数源
0	0	1/6振荡器频率（6时钟模式） 1/12振荡器频率（12时钟模式）
0	1	1/2振荡器频率（6时钟模式） 1/4振荡器频率（12时钟模式）
1	0	定时器0溢出
1	1	ECI脚的外部输入

在寄存器CMOD中有3个附加的位与PCA相关。其中CIDL位允许PCA在空闲模式时停止，WDTE使能/禁止模块4的看门狗功能，ECF可以控制产生中断，而PCA溢出标志CF（位于CCON中）在PCA定时器溢出时置位。这些功能如图25所示。看门狗定时器功能由模块4实现（见图35）。

寄存器CCON包含了PCA的运行控制位CR，溢出标志CF以及每个模块的中断标志（如图28所示）。要运行PCA，CR位必须由软件置位，清零该位则关闭PCA。CF位(CCON.7)在PCA计数器溢出时置位，此时若ECF位（位于CMOD中）为1则产生中断。CF位只能由软件清零。CCON寄存器中的位0~4为各个模块的中断标志位。当发生捕获/匹配时由硬件置位，只能由软件清零。PCA中断系统如图26所示。

PCA中的每个模块都有一个SFR与之对应，例如，CCAPM0用于模块0，CCAPM1用于模块1（见图29）。这些寄存器包含了控制各模块工作方式的位。当相关模块发生匹配或者比较时，ECCF位（CCAPMn.0，n=0,1,2,3或4）使能CCON中的CCF标志以产生中断。PWM位（CCAPMn.1）使能脉宽调制模式。TOG位（CCAPMn.2）置位且PCA计数器与模块的捕获/比较寄存器匹配时，CEX输出发生翻转。MAT位（CCAPMn.3）置位且PCA计数器与模块的捕获/比较寄存器匹配时，CCON中的CCFn位置位。

CAPN(CCAPMn.4)和CAPP(CCAPMn.5)决定捕获输入被激活的边沿。CAPN使能下降沿，CAPP使能上升沿。如果两个位都置位则在上升和下降沿均可使能产生捕获。ECOM(CCAPMn.6)置位可使能比较器功能。如图30所示为CCAPMn设置实现不同的PCA功能。

有两个额外的寄存器与每个PCA模块相关。分别为CCAPnH和CCAPnL，它们保存用于产生捕获或者比较的16位计数值。当模块用于PWM模式时，这两个寄存器用于控制输出信号的占空比。

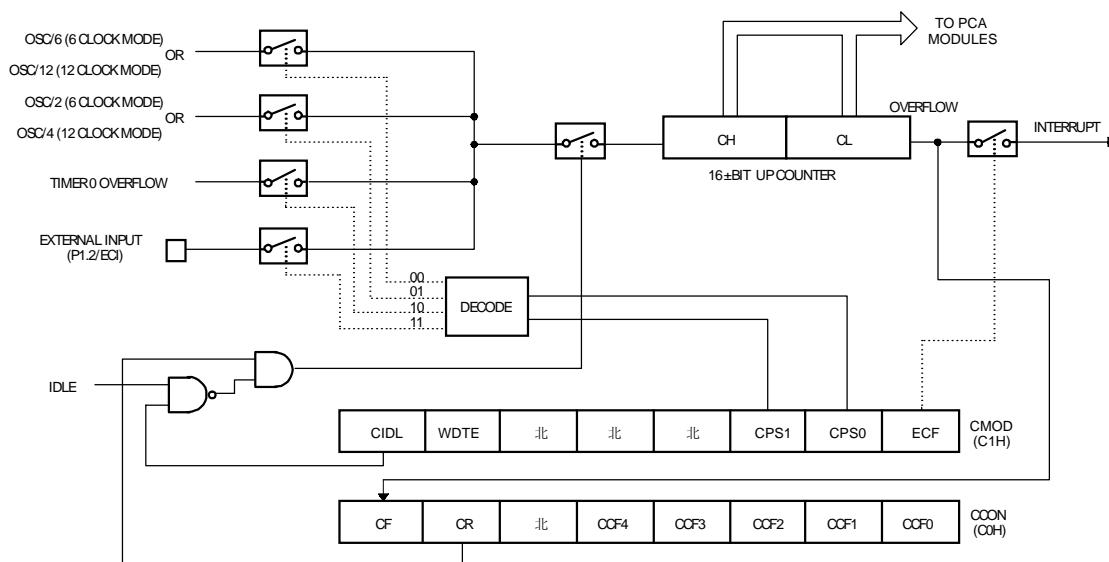


图25 PCA定时/计数器

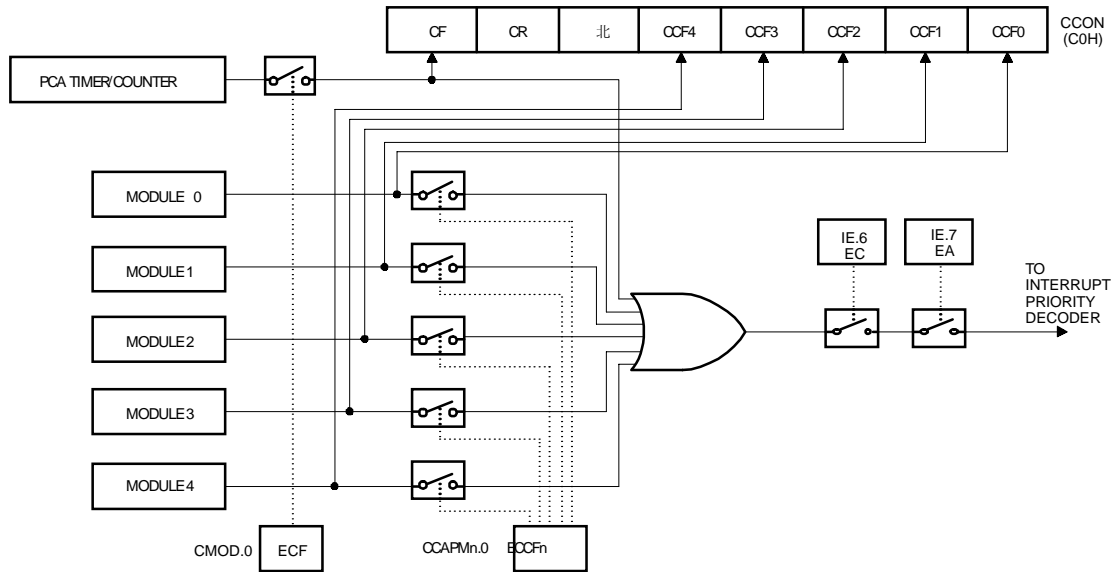


图26 PCA中断系统

CMOD 地址=D9H								复位值=00XX X000B
CIDL	WDTE	-	-	-	CPS1	CPS0	ECF	
7	6	5	4	3	2	1	0	
符号	功能							
CIDL	计数器空闲控制:CIDL=0 时, PCA 在空闲模式下继续工作。CIDL=1 时, PCA 在空闲模式下关闭。							
WDTE	看门狗定时器使能: WDTE=0 时禁止 PCA 模块 4 的看门狗功能。WDTE=1 时使能。							
-	保留位, 编程时勿将其置位。							
CPS1	PCA 计数脉宽选择位 1							
CPS0	PCA 计数脉宽选择位 0							
	CPS1	CPS0	选择 PCA 输入*					
	0	0	0 内部时钟, 6 时钟模式下为 $f_{osc}/6$ (12 时钟模式下为 $f_{osc}/12$)					
	0	1	1 内部时钟, 6 时钟模式下为 $f_{osc}/2$ (12 时钟模式下为 $f_{osc}/4$)					
	1	0	2 定时器 0 溢出					
	1	1	3 ECI/P1.2 脚输入的外部时钟(6 时钟模式最大为 $f_{osc}/4$, 12 时钟模式下为 $f_{osc}/8$)					
ECF	PCA 使能计数器溢出中断: ECF=1 使能 CCON 中的 CF 位产生中断; ECF=0 时禁止 CF 的功能。							
注: * f_{osc} =振荡器频率								

图27 CMOD: PCA计数器模式寄存器

CCON 地址=D8H				复位值=00X0 0000B			
可位寻址							
CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0
7	6	5	4	3	2	1	0
符号	功能						
CF	PCA 计数器溢出标志。计数器溢出时由硬件置位。CMOD 中 ECF 置位时，CR 为中断标志位。CF 可由硬件或软件置位，但只能由软件清零。						
CR	PCA 计数器运行控制位。由软件置位将 PCA 计数器打开；要关闭 PCA 计数器必须由软件清零。						
—	保留位，编程时勿将其置位。						
CCF4	PCA 模块 4 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。						
CCF3	PCA 模块 3 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。						
CCF2	PCA 模块 2 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。						
CCF1	PCA 模块 1 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。						
CCF0	PCA 模块 0 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。						

图28 CCON:PCA计数器控制寄存器

CCAPMn 地址: CCAPM0 0DAH				复位值=X000 0000B			
CCAPM1 0DBH							
CCAPM2 0DCH							
CCAPM3 0DDH							
不可位寻址 CCAMP4 0DEH							
—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
7	6	5	4	3	2	1	0
符号	功能						
—	保留位，编程时勿将其置位。						
ECOMn	比较器使能位。ECOMn=1 使能比较器功能。						
CAPPn	上升沿捕获，CAPPn=1 使能上升沿捕获。						
CAPNn	下降沿捕获，CAPNn=1 使能下降沿捕获。						
MATn	匹配。MATn=1 时，PCA 计数器值与该模块比较/捕获寄存器值的匹配使 CCON 中的 CCFn 位置位，标志中断。						
TOGn	触发。TOGn=1 时，PCA 计数器值与该模块比较/捕获寄存器值的匹配使 CEXn 脚的电平发生翻转。						
PWMn	脉宽调制模式。PWMn=1 使能 CEXn 脚作为脉宽调制信号的输出。						
ECCFn	使能 CCF 中断。使能 CCON 中的比较/捕获标志 CCFn 以产生中断。						

图29 PCA模块比较/捕获寄存器

-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCn	模块功能
X	0	0	0	0	0	0	0	无
X	X	1	0	0	0	0	X	16位捕获(CEXn脚上升沿触发)
X	X	0	1	0	0	0	X	16位捕获(CEXn脚下沿触发)
X	X	1	1	0	0	0	X	16位捕获(CEXn脚双边沿触发)
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30 PCA模块模式 (CCAPMn寄存器)

PCA捕获模式

要使用PCA模块中的捕获模式，必须将CCAPMn中的位CAPN和/或CAPP置位。CEXn输入的信号出现上升或下降沿时，PCA硬件将PCA计数器寄存器(CH和CL)的值装入模块捕获寄存器(CCAPnL和CCAPnH)。与此同时置位CCFn 标志。如果CCAPMn中的ECCFn位已经置位，则会产生中断。如图31所示。

16位软件定时器

要使用PCA模块中的软件定时器模式，必须将CCAPMn中的位ECOM和MAT置位。PCA定时器和模块的捕获寄存器进行比较。当两者的值相同且CCAPMn中的ECCFn位已经置位时，将会产生中断。如图32所示。

高速输出模式

在该模式下，每次当PCA定时器和模块的捕获寄存器的值相匹配时，模块对应的输出脚的电平将会发生翻转。要启用该模式，必须将CCAPMn中的TOG、MAT和ECOM都置位。见图33。

脉宽调制模式

所有的PCA模块都可用作PWM输出。其功能如图24所示。输出的频率由PCA定时器决定。由于所有的模块共用PCA定时器，因此它们具有相同的输出频率。每个模块的占空比可通过各自的捕获寄存器CCAPL单独设置。当PCA CL寄存器的值小于CCAPLn的值，输出为低；当CL的值等于或大于CCAPLn的值时，输出为高。当CL从FF溢出到00时，CCAPLn将CCAPHn的值重新装入。这就使PWM不会出现误操作。要启用PWM模式，必须将CCAPMn中的位PWM和ECOM都置位。

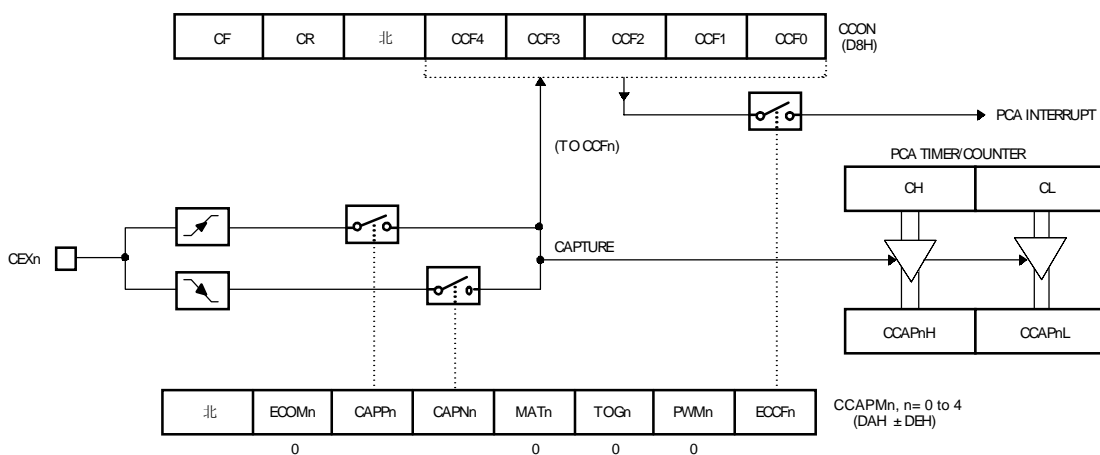


图31 PCA捕获模式

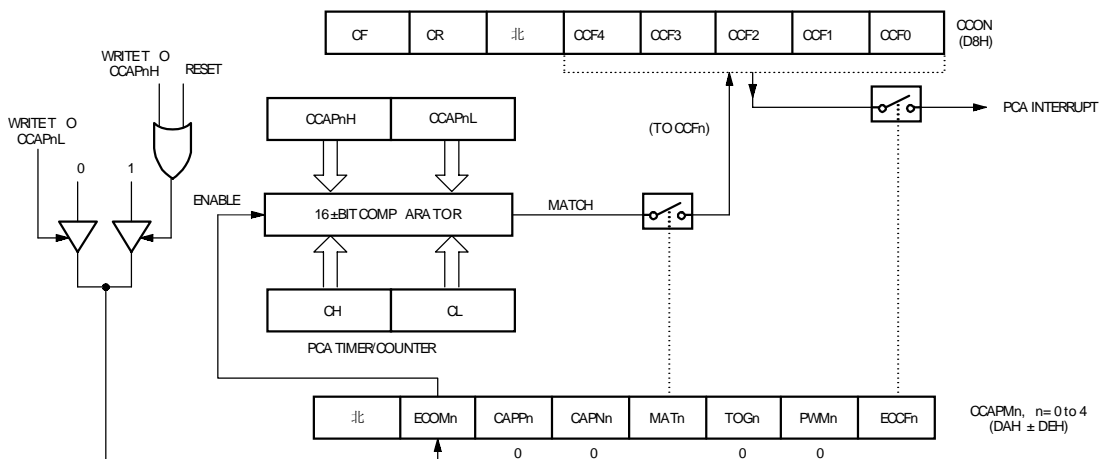


图32 PCA比较模式

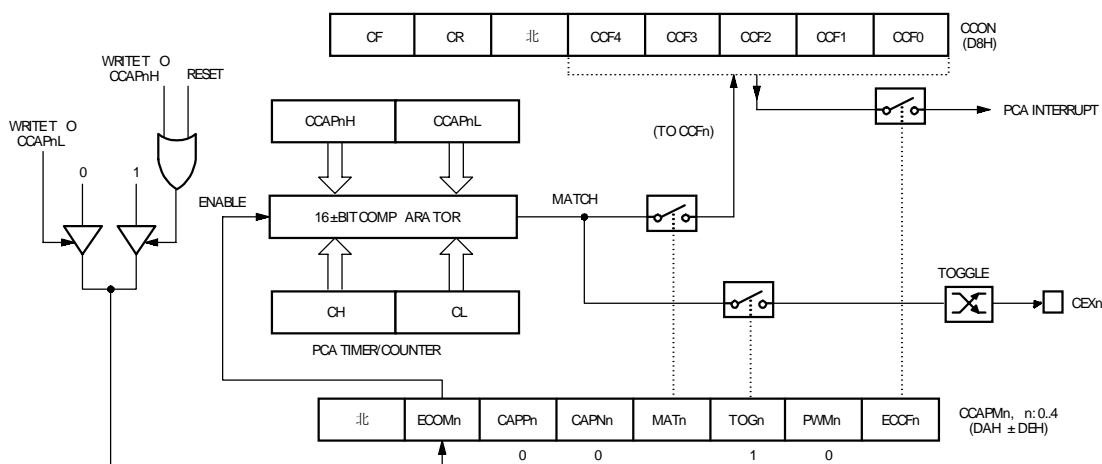


图33 PCA高速输出模式

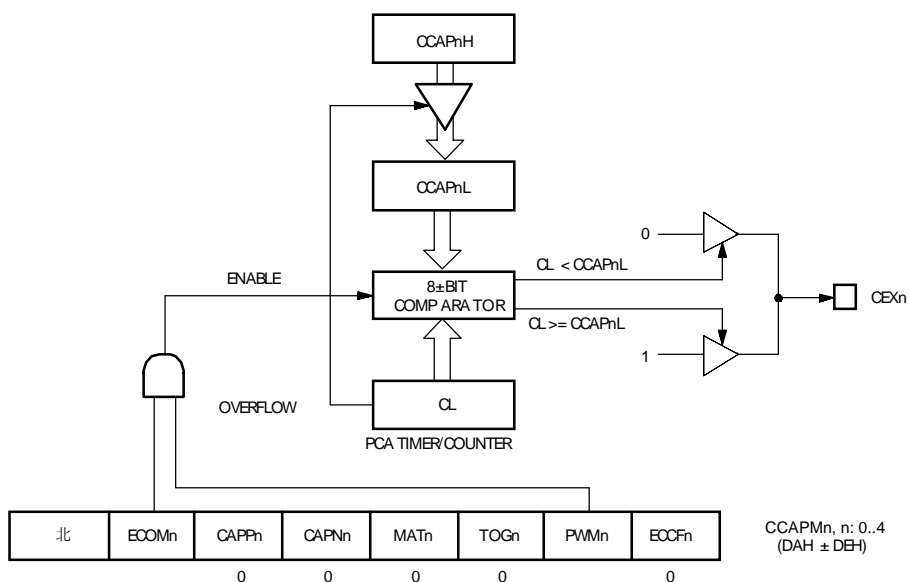


图34 PCA PWM模式

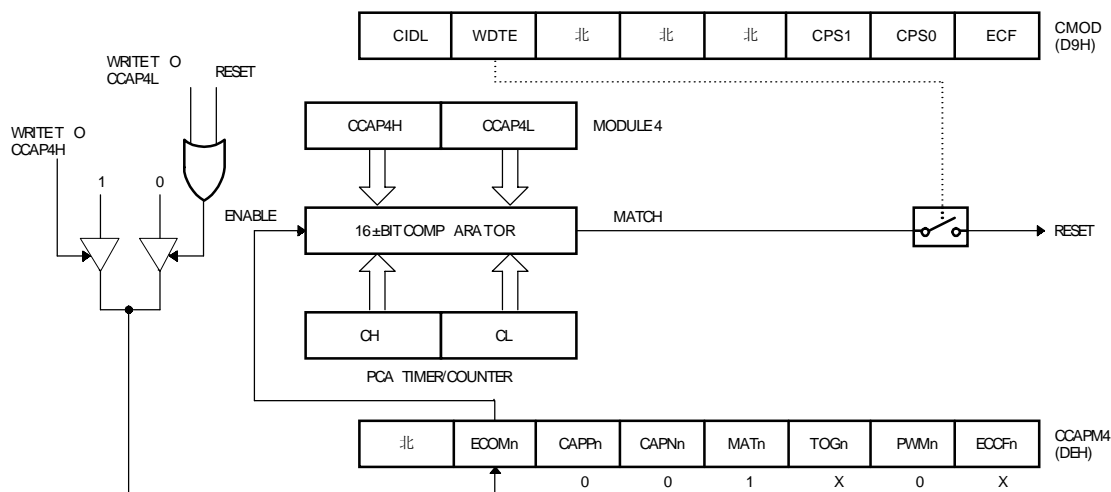


图35 PCA看门狗定时器(模块4)

PCA看门狗定时器

由PCA实现的片内看门狗定时器可改善系统的稳定性而不需要增加芯片的成本。看门狗定时器对于那些易受噪声、电源杂波或者静电放电影响的系统很有用处。PCA模块中只有模块4可编程为看门狗。如果不需要看门狗，该模块还可以用作其它模式。如图35所示为看门狗工作的示意图。用户将一个16位数值预先装入比较寄存器。就像其它比较模式一样，该16位值与PCA定时器值相比较。如果相同则产生一个内部复位，但不会将复位管脚RST拉高。

为了防止发生复位，用户有下列3个选择：

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

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

图36所示为看门狗定时器的初始化代码。模块4配置为比较模式，同时CMOD中的WDTE位必须置位。用户软件必须周期性改变（CCAP4H,CCAP4L）防止发生复位。该程序不能作为一个中断服务程序，因为如果程序指针跑飞并进入一个死循环，仍然执行中断将看门狗复位，这样将起不到使用看门狗的目的。应当在主程序小于PCA定时器 2^{16} 个计数的间隔内调用该程序。

```

INIT_WATCHDOG:
    MOV    CCAPM4,#4CH    ;模块4为比较模式
    MOV    CCAP4L,#0FFH  ;先写低字节
    MOV    CCAP4H,#0FFH  ;在PCA定时器计数到FFFF之前必须改变该比较值
    ORL   CMOD,#40H      ;置位WDTE位以能使看门狗定时器,不改变CMOD的其它
    位
;-----
;主程序,周期性调用子程序WATCHDOG
;-----
WATCHDOG:
    CLR   EA              ;关中断
    MOV   CCAP4L,#00      ;下一个比较值与当前PCA定时器值相差不超过255个计数
    MOV   CCAP4H,CH
    SETB  EA
    
```

图36 PCA看门狗定时器初始化代码

扩展数据RAM寻址

P89C51RA2/RB2/RC2/RD2xx内部数据存储器分为四部分:RAM低128字节、高128字节、128字节特殊功能寄存器(SFR)和256字节扩展RAM(ERAM)(RD2为768字节)。

1. RAM低128字节(地址00H到7FH)可直接或间接寻址
2. RAM高128字节(地址80H到FFH)只能间接寻址
3. 特殊功能寄存器(地址80H到FFH)只能直接寻址
4. 256/768字节扩展RAM(地址00H到1FFH/2FFH)通过将EXTRAM位清零实现MOVX间接寻址,见图37。

AUXR 地址 = 0DAH							复位值 = xxxx xx00B	
不可位寻址								
	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EXTRAM	AO
符号	功能							
AO	禁止/使能 ALE							
	AO 操作模式							
	0 以恒定的 1/3 振荡器频率发出 ALE 信号 (12 时钟模式下为 1/6 fosc)							
	1 ALE 仅当执行 MOVX 或 MOVC 指令时才有效							
EXTRAM	使用 MOVX @Ri/@DPTR 访问内部/外部 RAM							
	EXTRAM 操作模式							
	0 使用 MOVX @Ri/@DPTR 访问内部 RAM							
	1 访问外部数据存储器							
—	保留位, 编程时勿将其置位。							

图37 AUXR: 辅助寄存器

RAM低128字节可直接或间接寻址。高128字节只能间接寻址,高128字节与SFR占用相同的地址空间。不过它们在物理上是分开的。当指令访问高于7FH的内部地址时,CPU通过使用的寻址方式确定是对高128字节RAM还是对SFR进行访问。指令使用直接寻址方式访问SFR,例如:

MOV 0A0H, #data

使用间接寻址方式访问高128字节RAM，例如：

MOV @R0, #data

ERAM可通过将EXTRAM位清零用MOVX实现间接寻址。这部分ERAM在物理上位于片内，在逻辑上占用了外部数据存储器开始的7936字节。EXTRAM=0时，ERAM通过MOVX和R0、R1的组合使用实现间接寻址。对ERAM的访问不会影响P0口、P3.6(WR#)和P3.7(RD#)。在对外部寻址时，P2 SFR为输出。例如，EXTRAM=0时

MOVX @R0, #data

此处R0=0A0H，是对地址为0A0H的ERAM寻址而不是对外部存储器寻址。对高于ERAM地址的外部数据地址进行访问的方式与标准80C51相同。P0和P2作为数据/地址总线，P3.6和P3.7作为读/写时序信号。见图28。

EXTRAM=1时，MOVX @Ri和MOVX @DPTR指令与标准80C51相似。MOVX @Ri提供一个8位地址与数据复用P0口，任意的输出口都可用作输出高位地址。这提供了外部翻页功能。MOVX @DPTR产生一个16位地址，P2口输出高8位地址(DPH)，低8位地址(DPL)与数据复用P0口。MOVX @Ri和MOVX @DPTR在P3.6(WR#)和P3.7(RD#)上产生读或者写信号。堆栈指针可位于内部256字节RAM中的任意位置。堆栈不能位于ERAM中。

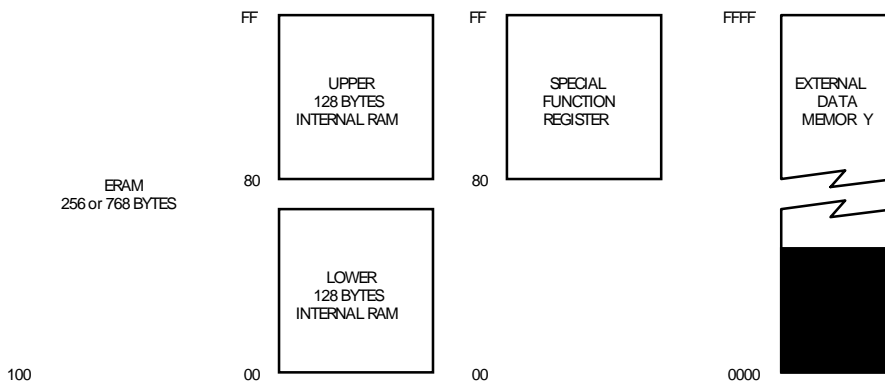


图38 内部和外部数据存储器地址空间(EXTRAM=0)

硬件看门狗定时器(WDT)

WDT在CPU落入软件运行失控的情况下可作为一种恢复的方法。WDT包含一个14位计数器和看门狗定时器复位SFR(WDTRST)。WDT在上电时被禁止。若想使能WDT，用户必须将01EH和0E1H依次写入WDTRST(地址0A6H)。当WDT使能后，在振荡器运行的情况下每过一个机器周期加1。此后无法禁止WDT，除非通过复位(硬件复位或者WDT溢出复位)。当WDT溢出时，它会在RST脚产生一个复位脉冲。

使用WDT

要使能WDT，用户必须将01EH和0E1H依次写入WDTRST。当WDT使能后，用户需要定期将01EH和0E1H依次写入WDTRST以避免WDT溢出。14位计数器在到达16383(3FFFH)后溢出并将芯片复位。当WDT使能后，在振荡器运行的情况下每过一个机器周期加1。这意味着用户必须在16383个机器周期内至少复位WDT一次。要想复位WDT，用户必须将01EH和0E1H依次写入WDTRST。WDTRST是一个只写寄存器。复位脉冲持续 $98 \times T_{osc}$ (6时钟模式)(12时钟模式下为196)，此处 $T_{osc}=1/f_{osc}$ 。

FLASH EPROM 存储器

概述

P89C51RA2/RB2/RC2/RD2xx Flash 存储器增加了 EPROM 所没有的电可擦除和编程特性。对芯片的擦除操作将整个程序存储区都擦除。块擦除功能可实现对任意 Flash 块进行擦除。在系统编程和标准的并行编

程都是可行的。片内产生的擦除和写入时序为用户提供了良好的编程接口。

P89C51RA2/RB2/RC2/RD2xx Flash 存储器甚至在经过 10,000 次擦除和编程之后仍然能可靠地保存存储器的内容。存储单元的设计使擦除和编程结构最优化。此外, 先进的沟道氧化工艺和低内部电场的结合使擦除和编程操作更加可靠。P89C51RA2/RB2/RC2/RD2xx 使用+5V Vpp 即可实现编程和擦除算法。

在系统编程 (ISP) 和在应用中编程 (IAP) 特性

- Flash EPROM 带块擦除的内部程序存储器
- 内部固化了 1K 字节的 Boot ROM, 包含了一个低级的在系统编程子程序和一个默认的串行装载程序。用户可调用这些程序来实现在应用中编程(IAP)。可将 Boot ROM 关闭以提供对整个 64k Flash 存储器的访问。
- Boot 向量允许用户将 Flash 装载代码放入 Flash 存储器内的任何位置。这种配置为用户提供了应用的灵活性。
- Boot ROM 中的默认装载子程序允许通过串口进行编程, 不需要用户提供装载程序。
- 如果内部程序存储区被禁止(EA=0), 外部程序空间最多可达 64k 字节。
- 编程和擦除电压为+5V(可承受+12V)。
- 使用 ISP/IAP 进行读/编程/擦除:
 - 字节编程(8μs)
 - 典型的快速擦除时间:
 - 块擦除(4k 字节)小于 3 秒
 - 整片擦除:
 - RD2xx(64K)小于 11 秒
 - RC2(32K)小于 7 秒
 - RB2(16K)小于 5 秒
 - RA2(8K)小于 4 秒
- 并行编程的硬件接口与 87C51 兼容
- 在系统编程
- 可对 Flash 程序代码进行编程加密
- 每个字节至少 10,000 次擦除/编程周期
- 数据至少可保存 10 年

Flash 的编程和擦除

有三种方法可实现对Flash的编程或擦除。第一, 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)。第二, 用Boot ROM固件由主机进行串行的在系统中编程ISP。第三, 使用商用编程器进行并行编程。对该器件进行并行编程的方法与87C51相似, 但不完全相同。商用编程器必须支持该器件。

Flash存储器空间

Flash用户代码存储器结构

P89C51RA2/RB2/RC2/RD2xx包含16KB/32KB/64KB的Flash程序存储器。它们以4K字节程序块为单位ISP和IAP BootROM子程序将通过附加的块数目分配支持新的4K字节块的规格, 并同时保持对以前8K字节和16K字节块分配的兼容。

状态字节/Boot向量块

该器件有一个4K字节的块用来保存状态字节和Boot向量(状态字节块)。状态字节通过ISP、IAP和并行模式进行编程。擦除状态字节和Boot向量中的任何一个都会擦除整个块的内容, 因此状态字节和Boot向量总是一起被擦除, 但编程是各自独立的。

保密/用户配置块

该器件有一个4K字节的块用来保存保密位、6时钟/12时钟模式选择位FX2和4095个用户可编程字节。可通过ISP、IAP和并行方式进行编程。保密位可防止通过并行编程器对代码进行读或写，但对ISP和IAP的禁止必须通过软件进行控制。该Flash块只有在执行ISP、IAP和并行模式下的整片擦除功能时才被擦除。该保密特性可保护软件不被盗版并防止Flash中的内容被读出。保密位位于Flash中。有3个可编程保密位可提供对片内代码和数据不同级别的保护（见表11）。4095个用户可编程字节不属于用户代码存储器的一部分，它可通过ISP、IAP或并行编程器读出。

6时钟/12时钟模式位FX2在上电时被锁存。该位可通过ISP或IAP改变并在下次复位时生效。这避免了在ISP操作中改变波特率。

Boot ROM

当微控制器对自身的Flash存储器进行编程时，所有底层操作的细节都由固化在1k字节Boot ROM中的代码进行处理。Boot ROM与Flash存储器是各自独立的。用户程序简单调用Boot ROM中带适当参数的公共入口即可实现所需要的操作。Boot ROM的操作包括以下内容：擦除程序块，编程字节，校验字节，编程保密位等等。当Boot ROM打开时，它与从0FC00到0FFFFH的程序存储器的地址相重叠。可以将Boot ROM关闭，这样就可对这一部分的程序存储区进行访问。

Flash 的结构

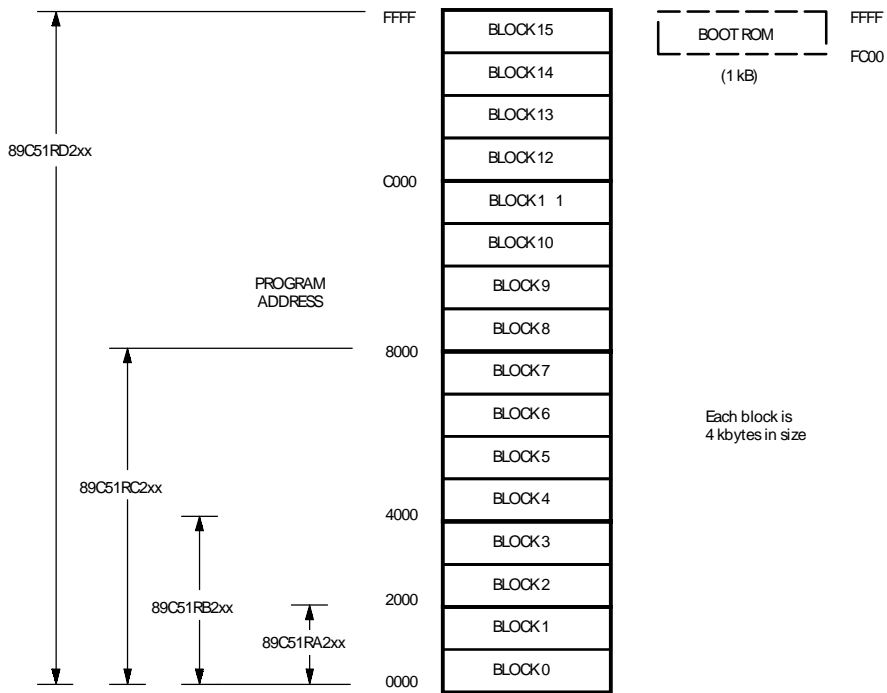


图 39 Flash 存储器结构

上电复位代码的执行

P89C51RA2/RB2/RC2/RD2xx包含两个特殊的Flash寄存器：BOOT向量和状态字节（注意：它们不在SFR中而在FLASH空间里）。在复位信号的下降沿，P89C51RA2/RB2/RC2/RD2xx检查状态字节中的内容。如果为0，则转去0000H地址开始执行程序。这是用户应用代码的正常起始地址。如果状态字节不为0，则将BOOT向量的值作为程序计数器的高字节，低字节固定为00H。工厂的默认设定为0FCH，对应Boot ROM的地址0FC00H。

注：当对FLASH进行擦除时，FLASH寄存器状态字节和BOOT向量也同时被擦除。所以必须对它们重新编程。

Boot装载程序的硬件激活

Boot装载程序也可通过拉低PSEN，ALE悬浮为高且在复位下降沿时EA>VIH启动执行。这和状态字节非零的效果是一样的。这样就可以在正常时执行用户代码，但又可以手动强制进入ISP操作。如果改变BOOT向量出厂时默认的设定1FH，将不再指向工厂预先编程的ISP BOOT装载程序。如果发生这种情况，改变BOOT向量的唯一方法就是使用并行编程。并行编程提供用户应用程序所不包含的定制的装载程序，以实现BOOT向量和状态字节的擦除和编程。在对Flash进行编程后，状态字节可编程为0，以允许用户应用程序从地址0000H开始执行。

在系统编程 (ISP)

ISP编程不需要将微控制器从系统中移出。在系统编程特性包含了一系列内部的硬件资源，与内部固件相结合可实现通过串口对P89C51RA2/RB2/RC2/RD2xx的远程编程。固件由Pilips提供并嵌入到每一个芯片当中。

Pilips的在系统编程特性使嵌入式应用中的在电路编程成为可行，并最大限度减小了额外的元件开销和电路板面积。ISP功能使用芯片的5个管脚TxD、RxD、Vcc、Vss和Vpp（见图40）。只需要一个小的连接器就可实现通过外部电路使用该特性的应用接口。Vpp电压必须充分的去耦，而且不能超过手册规定的最大值。

可从嵌入式系统学会得到免费的ISP软件：“FlashMagic”

1. 直接用浏览器打开下面的网页：

<http://www.esacademy.com/software/flashmagic/>

2. 下载 Flashmagic

3. 执行 “flashmagic.exe” 安装该软件

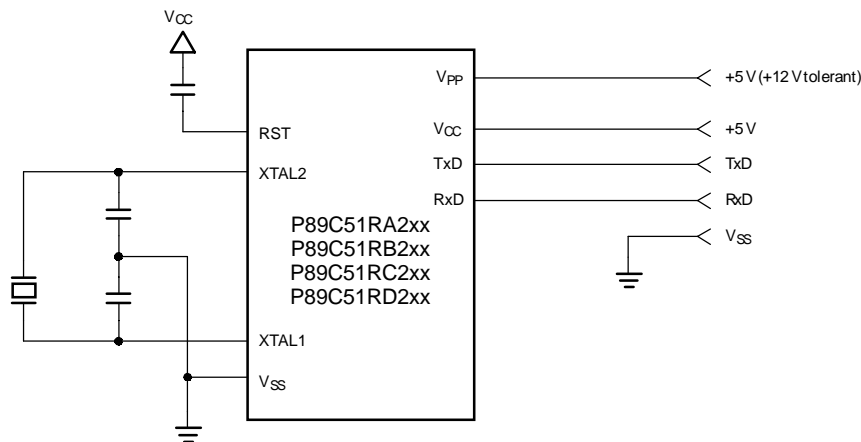


图40 在系统编程的管脚配置

使用在系统编程 (ISP)

ISP特性可以适应用户应用中较宽的范围的波特率而不依赖于振荡器频率。它还可以适应较宽范围的振荡器频率。这是通过对一个接收到的字符中单个位进行位时间测量来实现的。并用这个信息根据基于振荡器频率的定时器值对波特率进行编程。为此ISP特性要求主机必须向芯片发送一个初始化字符（大写的U）以使芯片对波特率进行设定。ISP固件提供对接收字节的自动回送。

一旦波特率的设定完成之后，ISP固件将只接受Intel-Hex格式的记录。Intel Hex格式的记录包含了用于表示十六进制数的ASCII字符。汇总如下：

: NNAAAARRDD..DDCC<crLf>

其中‘NN’表示记录中数据字节的个数。P89C51RA2/RB2/RC2/RD2xx最多可接受16个数据字节。

‘AAAA’字符串表示记录中首字节的地址。如果记录中的字节数为0，通常该区域设置为‘0000’。‘RR’表示记录的类型。‘00’表示数据类型，‘01’表示文件结束标识。数据字节的最大数目限制为16。ISP命

令汇总表8。

收到记录后，P89C51RA2/RB2/RC2/RD2xx将数据保存并计算校验和。在整个记录接收完毕后，执行由记录类型所指示的操作。如果校验和发生错误，芯片从串口发出‘X’表示校验和错误。如果校验结果与记录相符，在大多数情况下，从串口发出‘.’表示正确接收。随后按照类型号完成指定的操作。

对于数据记录(记录类型00)会有一个额外的校验。在校验和相符且对记录中所有字节都成功编程后，才发送字符‘.’。对于数据记录，‘X’表示校验和不相符，‘R’表示其中一个字节编程错误。需要在对数据编程之前发送类型02记录(指定振荡器频率)。ISP的特性使得它不要求指定振荡器频率用以产生波特率和编程脉冲的时序。用户只需要向芯片提供要求产生正确时序的信息。记录类型02可实现这一目的。

从Pilips Semiconductor网站可以得到通过PC实现ISP编程的软件WinISP。此外，网站上还提供第三方的商用串行和并行编程器的清单。

表 8 ISP 所用 Intel-Hex 记录

记录类型	命令/数据 功能
00	编程数据 :nnaaaa00dd... ddcc 此处: nn = 记录的字节数(hex) aaaa = 记录中首字节的地址 dd... dd = 编程数据字节 cc = 校验和 例:10008000AF5F67F0602703E0322CFA92007780C3FD
01	文件结束(EOF), 无操作 :xxxxxx01cc 此处: xxxxxx = 无关位 cc = 校验和 例:00000001FF
03	混合写功能: nnxxxx03ffssddcc 此处: nn = 记录的字节数(hex) xxxx= 无关值 03 = 写功能 ff = 子功能代码 ss = 选择的代码 dd = 数据输入(如果需要) cc = 校验和 子功能代码=01(擦除 8K/16K Flash 块) ff = 01 ss = 块代码 块 0, 0k 到 8k, 00H 块 1, 8k 到 16k, 20H(RB2, RC2, RD2) 块 2, 16k 到 32k, 40H(RC2, RD2) 块 3, 32k 到 48k, 80H(RD2) 块 4, 48k 到 64k, C0H(RD2) 例: 0200000301C03A 擦除块 4 子功能代码=04(擦除 Boot 向量和状态字节)

	<p>ff = 04 ss = 无关</p> <p>例: 020000030400F7 擦除Boot向量和状态字节 子功能代码=05(编程保密位)</p> <p>ff = 05 ss = 00 对保密位1编程(禁止写入FLASH) 01 对保密位2编程(禁止校验FLASH) 02 对保密位3编程(禁止执行外部存储器代码)</p> <p>例: 020000030501F5 对保密位2编程 子功能码=06(编程Boot向量或状态字节)</p> <p>ff = 06 ss = 00 对状态字节编程 01 对Boot向量编程 02 编程FX2位(dd = 80)</p> <p>dd = 数据</p> <p>例 1: 030000030601FCF7 Boot 向量编程为 0FCH 例 2: 0300000306028072 编程 FX2 位 (选择 12 时钟模式)</p> <p>子功能码=07 (整片擦除) 擦除所有FLASH块、保密位以及将Boot向量和状态字节设置为缺省值</p> <p>ff = 07 ss = 无关 dd = 无关</p> <p>例: 0100000307F5 整片擦除 子功能代码=0C(擦除 4K Flash 块)</p> <p>ff = 0C ss = 块代码如下:</p> <p style="padding-left: 40px;">块 0, 0k 到 4k, 00H 块 1, 4k 到 8k, 10H 块 2, 8k 到 12k, 20H(RB2, RC2, RD2) 块 3, 12k 到 16k, 30H(RB2, RC2, RD2) 块 4, 16k 到 20k, 40H(RC2, RD2) 块 5, 20k 到 24k, 50H(RC2, RD2) 块 6, 24k 到 28k, 60H(RC2, RD2) 块 7, 28k 到 32k, 70H(RC2, RD2) 块 8, 32k 到 36k, 80H(RD2) 块 9, 36k 到 40k, 90H(RD2) 块 10, 40k 到 44k, A0H(RD2) 块 11, 44k 到 48k, B0H(RD2) 块 12, 48k 到 52k, C0H(RD2) 块 13, 52k 到 56k, D0H(RD2) 块 14, 56k 到 60k, E0H(RD2) 块 15, 60k 到 62k, F0H(RD2)</p> <p>例: 020000030C20CF 擦除块 2</p>
--	--

04	<p>显示器件数据或查空—整个Flash阵列的内容以一定的格式从串口送出。包括一个地址以及其后的16个字节数据。如果保密位2已编程，将不会显示器件的内容。向串口发送数据由接收到的任意字符启动并由接收到的任意字符终止。</p> <p>格式: 05xxxx04sssseeeffcc</p> <p>其中 05 = 记录中数据字节数 xxxxx = 所要求的字段, 但其值为“无关” 04 = “显示器件数据或查空”功能代码 ssss = 起始地址 eeee = 结束地址 ff = 子功能代码: 00 或01 00 = 显示数据 01 = 查空 cc = 校验和</p> <p>例1: 0500000440004FFF0069 显示地址4000~4FFF FLASH段内容 例2: 0500000400000FFF0069 显示数据块中的数据(地址0000处的数据无效)</p>
05	<p>混合读功能</p> <p>格式: 02xxxx05ffsscc</p> <p>其中 02 = 记录中数据字节数 xxxx = 所要求的字段, 但其值为“无关” 05 = “杂项读”功能代码 ffss = 子功能和选择码 0000 = 读标识字节—厂商ID(15H) 0001 = 读标识字节—器件ID #1 (C2H) 0002 = 读标识字节—器件ID #2 0003 = 读FX2位 0080 = 读ROM代码版本 0700 = 读保密位 0701 = 读状态字节 0702 = 读Boot向量 cc = 校验和</p> <p>例 1: 020000050001F8 读标识字节—器件 ID #1 例 2: 020000050003F6 读 FX2 位 (Bit7=0 表示 12 时钟模式, Bit7=1 表示 6 时钟模式) 例 3: 02000005008079 读 ROM 代码版本 (0A: 版本 A, 0B: 版本 B)</p>
06	<p>直接装载波特率</p> <p>格式: 02xxxx06hhllcc</p> <p>其中 02 = 记录中数据字节数 xxxx = 所要求的字段, 但其值为“无关” 06 = “直接装载波特率”功能代码 hh = 定时器2高字节 ll = 定时器2低字节 cc = 校验和</p> <p>例 :02000006F500F3 定时器 2 值 F500</p>
07	<p>编程数据块中的数据</p> <p>格式: nnaaaa07dd... ddcc</p>

	<p>其中 nn = 记录中数据字节数</p> <p>aaaa = 第一个字节的存储器地址 (有效地址: 0001~0FFFH)</p> <p>dd...dd = 数据字节</p> <p>cc = 校验和</p> <p>例: 10008007AF5F67F0602703E0322CFA92007780C3F6</p>
--	---

在应用中编程方法

通过一个应用程序对几个IAP子程序的调用, 可实现有选择地对FLASH块进行擦除和编程。所有的调用都是通过一个公共的接口PGM_MTP来实现的。在调用位于地址FFF0H的PGM_MTP之前, 通过设置微控制器的寄存器可选择编程的功能。振荡器频率以兆赫为单位向下舍入到最近的整数。例如, 晶振为11.0592MHz时, R0设定为11。结果返回到寄存器中。IAP调用见表9。

表9 IAP 调用

IAP 调用	参数
编程字节	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 06h 或 R1 = 82h(清零WDT)</p> <p>DPTR= 编程字节地址</p> <p>ACC = 编程的字节</p> <p>返回参数</p> <p>ACC = 00 表示通过, !=00 表示失败</p>
擦除 4K 代码块 (新功能)	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 0(快速擦除) 或 R1 = 8Ch(清零WDT)</p> <p>DPH = 4K代码块地址:</p> <p style="padding-left: 40px;">块 0, 0k 到 4k, 00H</p> <p style="padding-left: 40px;">块 1, 4k 到 8k, 10H</p> <p style="padding-left: 40px;">块 2, 8k 到 12k, 20H</p> <p style="padding-left: 40px;">块 3, 12k 到 16k, 30H</p> <p style="padding-left: 40px;">块 4, 16k 到 20k, 40H</p> <p style="padding-left: 40px;">块 5, 20k 到 24k, 50H</p> <p style="padding-left: 40px;">块 6, 24k 到 28k, 60H</p> <p style="padding-left: 40px;">块 7, 28k 到 32k, 70H</p> <p style="padding-left: 40px;">块 8, 32k 到 36k, 80H</p> <p style="padding-left: 40px;">块 9, 36k 到 40k, 90H</p> <p style="padding-left: 40px;">块 10, 40k 到 44k, A0H</p> <p style="padding-left: 40px;">块 11, 44k 到 48k, B0H</p> <p style="padding-left: 40px;">块 12, 48k 到 52k, C0H</p> <p style="padding-left: 40px;">块 13, 52k 到 56k, D0H</p> <p style="padding-left: 40px;">块 14, 56k 到 60k, E0H</p> <p style="padding-left: 40px;">块 15, 60k到64k, F0H</p> <p>DPL = 00H</p> <p>返回参数</p> <p>ACC = 00 表示通过, !=00 表示失败</p>

<p>擦除 8K/16K 代码块</p>	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 0(快速擦除) 或 R1 = 81h(清零WDT)</p> <p>DPH = 块代码如下:</p> <p> 块0, 0k~8k, 00H</p> <p> 块1, 8k~16k, 20H</p> <p> 块2, 16k~32k, 40H</p> <p> 块3, 32k~48k, 80H</p> <p> 块4, 48k~64k, C0H</p> <p>DPL = 00H</p> <p>返回参数</p> <p>ACC = 00 表示通过, !=00 表示失败</p>
<p>擦除状态字节 /Boot 向量</p>	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 04h 或 R1 = 84h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 无关</p> <p>返回参数</p> <p>ACC = 00 表示通过, !=00 表示失败</p>
<p>编程保密位</p>	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 05h 或 R1 = 85h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 00h — 保密位#1(禁止写入Flash)</p> <p> 01h — 保密位#2(禁止对Flash进行校验)</p> <p> 02h — 保密位#3(禁止外部程序)</p> <p>返回参数</p> <p>ACC = 00 表示通过, !=00 表示失败</p>
<p>编程状态字节</p>	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 06h 或 R1 = 86h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 01H — 编程状态字节</p> <p>ACC = 状态字节</p> <p>返回参数</p> <p>ACC = 00 表示通过, !=00 表示失败</p>
<p>编程 Boot 向量</p>	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 06h 或 R1 = 86h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 01H — 编程Boot向量</p> <p>ACC = Boot向量</p> <p>返回参数</p> <p>ACC = 00 表示通过, !=00 表示失败</p>

读器件数据	<p>输入参数:</p> <p>R1 = 06h 或 R1 = 86h(清零WDT)</p> <p>DPTR= 读出字节的地址</p> <p>返回参数</p> <p>ACC = 读出字节的值</p>
读制造厂商 ID	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 00h 或 R1 = 80h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 00H(制造厂商ID)</p> <p>返回参数</p> <p>ACC = 读出字节的值</p>
读器件 ID #1	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 00h 或 R1 = 80h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 01H(器件ID #1)</p> <p>返回参数</p> <p>ACC = 读出字节的值</p>
读器件 ID #2	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 00h 或 R1 = 80h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 02H(器件ID #2)</p> <p>返回参数</p> <p>ACC = 读出字节的值</p>
读保密位	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 07h 或 R1 = 87h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 00H(保密位)</p> <p>返回参数</p> <p>ACC = 读出字节的值</p>
读状态字节	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 07h 或 R1 = 87h(清零WDT)</p> <p>DPH = 00h</p> <p>DPL = 01H(状态字节)</p> <p>返回参数</p> <p>ACC = 读出字节的值</p>
读 Boot 向量	<p>输入参数:</p> <p>R0 = 振荡器频率(整数)</p> <p>R1 = 07h 或 R1 = 87h(清零WDT)</p> <p>DPH = 00h</p>

	DPL = 02H(Boot向量) 返回参数 ACC = 读出字节的值
--	---

保密性

保密特性是为了保护软件的版权，防止 FLASH 的内容被读出。保密位位于 FLASH 中。P89C51RX2 有 3 个可编程保密位，可为片内的代码和数据提供不同级别的保护（见表 8）。与 ROM 和 OTP 版本不同的是，这 3 个保密位是独立的，LB3 包含了 LB1 的保密功能。

表 10

级别	保密位			保护描述
	LB1	LB2	LB3	
1	0	0	0	禁止外部程序存储器中的指令 MOV _C 读取内部存储器的代码字节
2	1	0	0	禁止执行块擦除，禁止擦除或编程状态字节或 Boot 向量
3	1	1	0	禁止对程序进行校验
4	1	1	1	禁止执行外部程序

- 注： 1. 保密位是各自独立的，不管保密位状态如何，都可执行整片擦除。
2. 未定义保密位的其它组合。
3. LB_x 的设置不能防止对未编程字节的编程。

极限参数

参数	额定值	单位
操作温度	0~+70 或 -40~+85	°C
贮存温度范围	-65~+150	°C
\overline{EA}/V_{pp} 脚相对于 V _{SS} 的电压	0~+13.0	V
其它任何脚相对于 V _{SS} 的电压	-0.5~+6.5	V
每个 I/O 脚的最大 I _{OL}	15	mA
功率损耗（指器件表面的发热，而非器件的功耗）	1.5	W

DC 电气特性

T_{amb}=0°C~+70°C 或 -40°C~+85°C; 5V±10%; V_{SS}=0V

标号	参数	测试条件	极限			单位
			最小	典型 ¹	最大	
V _{IL}	输入低电压	4.5V<V _{CC} <5.5V	-0.5		0.2V _{CC} -0.1	V
V _{IH}	输入高电压(P0,1,2,3, \overline{EA})		0.2V _{CC} +0.9		V _{CC} +0.5	V
V _{IHI}	输入高电压,XTAL1,RST		0.7V _{CC}		V _{CC} +0.5	V
V _{OL}	输出低电压,P1,2,3 ⁸	V _{CC} =4.5V I _{OL} =1.6mA ²			0.4	V
V _{OL1}	输出低电压,P0 口,ALE, \overline{PSEN} ^{7,8}	V _{CC} =4.5V I _{OL} =3.2mA ²			0.45	V
V _{OH}	输出高电压,P1,2,3 ³	V _{CC} =4.5V I _{OH} =-30uA	V _{CC} -0.7			V
V _{OHI}	输出高电压 ALE ⁹ , \overline{PSEN} ³	V _{CC} =4.5V I _{OH} =-3.2mA	V _{CC} -0.7			V

I_{IL}	逻辑 0 输入电流,P1,2,3	$V_{IN}=0.4V$	-1		-75	V
I_{TL}	逻辑 1 到 0 的转变电流,P1,2,3 ⁶	$V_{IN}=2.0V$ 参见注释 4			-650	V
I_{LI}	输入漏电流,P0	$0.45 < V_{IN} < V_{CC} - 0.3$			± 10	μA
I_{CC}	电源电流(参见图 36): 激活模式 空闲模式 掉电模式或时钟停止(参见图 42 的条件) 编程和擦除模式	参见注 5 $T_{amb}=0^{\circ}C \sim +70^{\circ}C$ $T_{amb}=-40^{\circ}C \sim +85^{\circ}C$ $f_{osc}=20MHz$			<30 <40 60	100 125 mA
R_{RST}	内部复位下拉电阻		40		225	k Ω
C_{IO}	管脚电容 ¹⁰ (\overline{EA} 脚除外)				15	pF

注:

- 1) 不能保证典型值,因为这些值是在室温、5V 下测得。
- 2) P0 口和 P2 口上的容性负载会产生噪声叠加到 1 口、3 口和 ALE 的低电平上。噪声产生的原因是在总线操作期间,0 口和 2 口从 1 到 0 的跳变会使外部总线电容对 0 口和 2 管脚放电,在最恶劣的情况下(容性负载>100pF),ALE 管脚上的噪声脉冲可超过 0.8V。在这种境况下,可以通过施密特触发器或者带有施密特触发 STROBE 输入的地址锁存器来校正 ALE。 I_{OL} 会超过测试条件下的电流。
- 3) 容性负载加到 0 口和 2 口会导致 ALE 和 PSEN 管脚瞬时低于 $V_{CC}-0.7V$,当地址位稳定下来。
- 4) 当口 1、2、3 被外部电路拉低时,口上从 1 到 0 的跳变将产生跳变电流,当输入电压大约在 2v 时,跳变电流达到最大。
- 5) 图 51 到 54 所示为 I_{CC} 的测试条件,图 48 为 I_{CC} 和频率的关系。
 激活模式: $I_{CC(MAX)}=(0.9 \times \text{FREQ.}+10.5) \text{ mA}$
 空闲模式: $I_{CC(MAX)}=(0.33 \times \text{FREQ.}+2.5) \text{ mA}$
- 6) 应用温度 $T=0^{\circ}C \sim +70^{\circ}C$ 。
- 7) 口 0、ALE 和 \overline{PSEN} 脚的负载电容为 100pF,其他输出口为 80 pF。
- 8) 在稳定的状态条件下, I_o 低被外部限制如下:
 - i. 每个管脚的最大 I_{OL} 15mA (注: 85 $^{\circ}C$ 规格)
 - ii. 每个 8 位口的最大 I_{OL} 26 mA
 - iii. I_{OL} 输出最大总和 71mA
 - iv. 如果 I_{OL} 超过测试条件, V_{OL} 可能会超过相应规格。不能保证超过测试电流。
- 9) ALE 的测试是 ALE 关断情况下,测出 ALE 的高电位值。
- 10) 管脚电容特性并不由测试得出,而是由其特性保证。管脚电容小于 25 pF。陶瓷电容小于 15pF (\overline{EA} 是 25pF)。

AC 电气特性(12 时钟模式)

符号	图号	参数	变化的时钟		33MHz 时钟		单位
			最小	最大	最小	最大	
$1/t_{CLCL}$	41	振荡器频率	0	33			MHz
t_{LHLL}	41	ALE 脉宽	$2t_{CLCL}-40$		21		ns
t_{AVLL}	41	地址有效到 ALE 低	$t_{CLCL}-25$		5		ns
t_{LLAX}	41	ALE 为低后的地址保持	$t_{CLCL}-25$		5		ns
t_{LLIV}	41	ALE 低到有效指令输入		$4t_{CLCL}-65$		55	ns
t_{LLPL}	41	ALE 低到 PSEN 低	$t_{CLCL}-25$		5		ns
t_{PLPH}	41	PSEN 脉宽	$3 t_{CLCL}-40$		45		ns
t_{PLIV}	41	PSEN 低到有效指令输入		$3 t_{CLCL}-60$		30	ns
t_{PXIX}	41	PSEN 之后输入指令保持	0		0		ns
t_{PXIZ}	41	PSEN 之后输入指令悬浮		$t_{CLCL}-25$		5	ns
t_{AVIV}	41	地址到有效指令输入		$5 t_{CLCL}-80$		70	ns
t_{PLAZ}	41	PSEN 低到地址悬浮		10		10	ns
数据存储器							
t_{RLRH}	42,43	RD 脉宽	$6 t_{CLCL}-100$		82		ns
t_{WLWH}	42,43	WR 脉宽	$6 t_{CLCL}-100$		82		ns
t_{RLDV}	42,43	RD 低到有效数据输入		$5 t_{CLCL}-90$		60	ns
t_{RHDX}	42,43	RD 后的数据保持	0		0		ns
t_{RHDZ}	42,43	RD 后的数据悬浮		$2t_{CLCL}-28$		32	ns
t_{LLDV}	42,43	ALE 低到有效数据输入		$8 t_{CLCL}-150$		90	ns
t_{AVDV}	42,43	地址到有效数据输入		$9 t_{CLCL}-165$		105	ns
t_{LLWL}	42,43	ALE 低到 RD 或 WD 低	$3 t_{CLCL}-50$	$3 t_{CLCL}+50$	40	140	ns
t_{AVWL}	42,43	地址有效到 WD 低或 RD 低	$4 t_{CLCL}-75$		45		ns
t_{QVWX}	42,43	数据有效到 WR 翻转	$t_{CLCL}-30$		0		ns
t_{WHQX}	42,43	WR 之后的数据保持	$t_{CLCL}-25$		5		ns
t_{QVWH}	43	数据有效到 WR 高	$7 t_{CLCL}-130$		80		ns
t_{RLAZ}	42,43	RD 低到地址悬浮		0		0	ns
t_{WHLH}	42,43	RD 或 WD 高到 ALE 高	$t_{CLCL}-25$	$t_{CLCL}+25$	5	55	ns
外部时钟							
t_{CHCX}	45	高电平时间	17	$t_{CLCL}-t_{CLCX}$			ns
t_{CLCX}	45	低电平时间	17	$t_{CLCL}-t_{CHCX}$			ns
t_{CLCH}	45	上升时间		5			ns
t_{CHCL}	45	下降时间		5			ns
移位寄存器							
t_{XLXL}	44	串口时钟周期	$12 t_{CLCL}$		360		ns
t_{QVXH}	44	输出数据建立到时钟上升沿	$10 t_{CLCL}-133$		167		ns
t_{XHQX}	44	时钟上升沿后输出数据的保持时间	$2t_{CLCL}-80$		50		ns
t_{XHDX}	44	时钟上升沿后输入数据的保持时间	0		0		ns
t_{XHDV}	44	时钟上升沿到输入数据有效		$10 t_{CLCL}-133$		167	ns

1. 这些参数都在工作温度范围内有效，除非有特别的说明。
2. 口 0、ALE 和 \overline{PSEN} 脚的负载电容为 100pF，其他输出口为 80 pF。

AC 电气特性(6 时钟模式)

符号	图号	参数	变化的时钟		33MHz 时钟		单位
			最小	最大	最小	最大	
$1/t_{CLCL}$	41	振荡器频率	0	20			MHz
t_{LHLL}	41	ALE 脉宽	$t_{CLCL}-40$		10		ns
t_{AVLL}	41	地址有效到 ALE 低	$0.5 t_{CLCL}-20$		5		ns
t_{LLAX}	41	ALE 为低后的地址保持	$0.5 t_{CLCL}-20$		5		ns
t_{LLIV}	41	ALE 低到有效指令输入		$2t_{CLCL}-65$		35	ns
t_{LLPL}	41	ALE 低到 PSEN 低	$0.5 t_{CLCL}-20$		5		ns
t_{PLPH}	41	PSEN 脉宽	$1.5 t_{CLCL}-20$		30		ns
t_{PLIV}	41	PSEN 低到有效指令输入		$1.5 t_{CLCL}-60$		15	ns
t_{PXIX}	41	PSEN 之后输入指令保持	0		0		ns
t_{PXIZ}	41	PSEN 之后输入指令悬浮		$0.5 t_{CLCL}-20$		5	ns
t_{AVIV}	41	地址到有效指令输入		$2.5 t_{CLCL}-80$		45	ns
t_{PLAZ}	41	PSEN 低到地址悬浮		10		10	ns
数据存储							
t_{RLRH}	42,43	RD 脉宽	$3 t_{CLCL}-100$		50		ns
t_{WLWH}	42,43	WR 脉宽	$3 t_{CLCL}-100$		50		ns
t_{RLDV}	42,43	RD 低到有效数据输入		$2.5 t_{CLCL}-90$		35	ns
t_{RHDX}	42,43	RD 后的数据保持	0		0		ns
t_{RHDZ}	42,43	RD 后的数据悬浮		$t_{CLCL}-20$		5	ns
t_{LLDV}	42,43	ALE 低到有效数据输入		$4 t_{CLCL}-150$		50	ns
t_{AVDV}	42,43	地址到有效数据输入		$4.5 t_{CLCL}-165$		60	ns
t_{LLWL}	42,43	ALE 低到 RD 或 WD 低	$1.5 t_{CLCL}-50$	$1.5 t_{CLCL}+50$	25	125	ns
t_{AVWL}	42,43	地址有效到 WD 低或 RD 低	$2 t_{CLCL}-75$		25		ns
t_{QVWX}	42,43	数据有效到 WR 翻转	$0.5 t_{CLCL}-25$		0		ns
t_{WHQX}	42,43	WR 之后的数据保持	$0.5 t_{CLCL}-20$		5		ns
t_{QVWH}	43	数据有效到 WR 高	$3.5 t_{CLCL}-130$		45		ns
t_{RLAZ}	42,43	RD 低到地址悬浮		0		0	ns
t_{WHLH}	42,43	RD 或 WD 高到 ALE 高	$0.5 t_{CLCL}-20$	$1.5 t_{CLCL}-50$	5	45	ns
外部时钟							
t_{CHCX}	45	高电平时间	20	$t_{CLCL}-t_{CLCX}$			ns
t_{CLCX}	45	低电平时间	20	$t_{CLCL}-t_{CHCX}$			ns
t_{CLCH}	45	上升时间		5			ns
t_{CHCL}	45	下降时间		5			ns
移位寄存器							
t_{XLXL}	44	串口时钟周期	$6 t_{CLCL}$		300		ns
t_{QVXH}	44	输出数据建立到时钟上升沿	$5 t_{CLCL}-133$		117		ns
t_{XHQX}	44	时钟上升沿后输出数据的保持时间	$t_{CLCL}-30$		20		ns
t_{XHDX}	44	时钟上升沿后输入数据的保持时间	0		0		ns
t_{XHDV}	44	时钟上升沿到输入数据有效		$5 t_{CLCL}-133$		117	ns

1. 这些参数都在工作温度范围内有效，除非有特别的说明。
2. 口 0、ALE 和 \overline{PSEN} 脚的负载电容为 100pF，其他输出口为 80 pF。

AC 特性符号说明

每一时序符号有 5 个字符。首先是“t”(=时间)，其字符基于他们的位置、名称和逻辑状态说明如下：

- | | |
|--------------|------------------------------|
| A—地址 | P— $\overline{\text{PSEN}}$ |
| C—时钟 | Q—数据输出 |
| D—输入数据 | R— $\overline{\text{RD}}$ 信号 |
| H—逻辑高电平 | t—时间 |
| I—指令 | W— $\overline{\text{WR}}$ 信号 |
| L—逻辑低电平或 ALE | X—不再是有效逻辑电平 |
| Z—悬浮 | V—有效 |

例如： t_{AVLL} = 从地址有效到 ALE 为低的时间。

t_{lp1} = 从 ALE 为低到 PSEN 为低的时间。

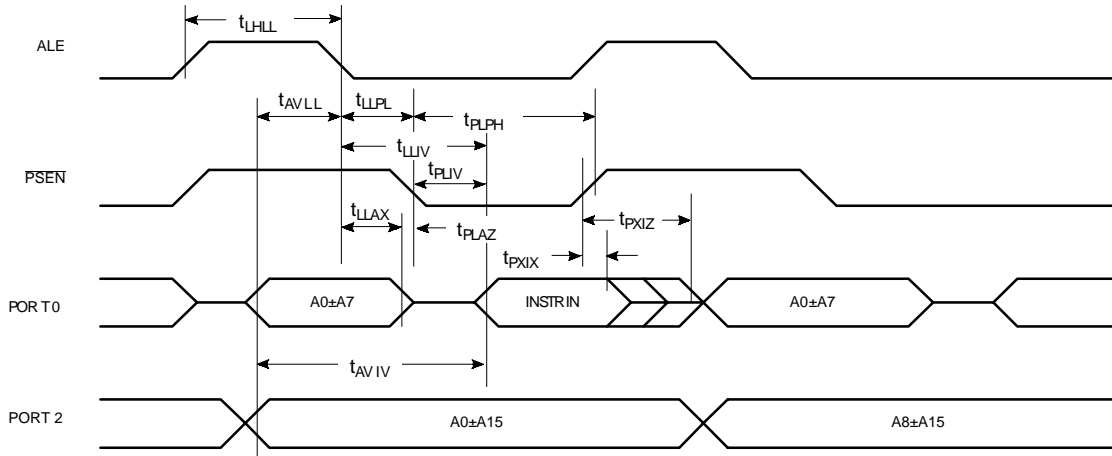


图 41 外部程序存储器的读周期

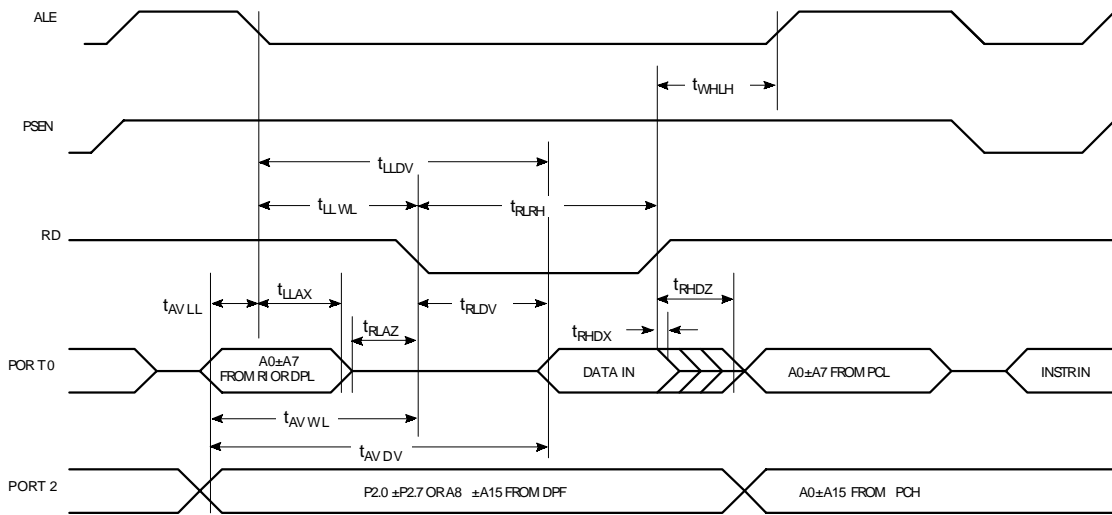


图 42 外部数据存储器的读周期

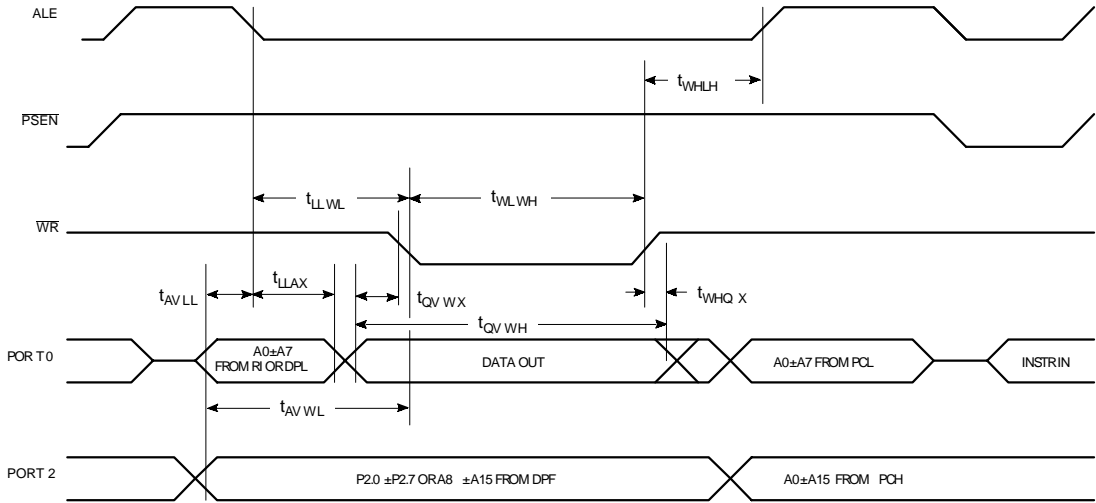


图 43 外部数据存储器的写周期

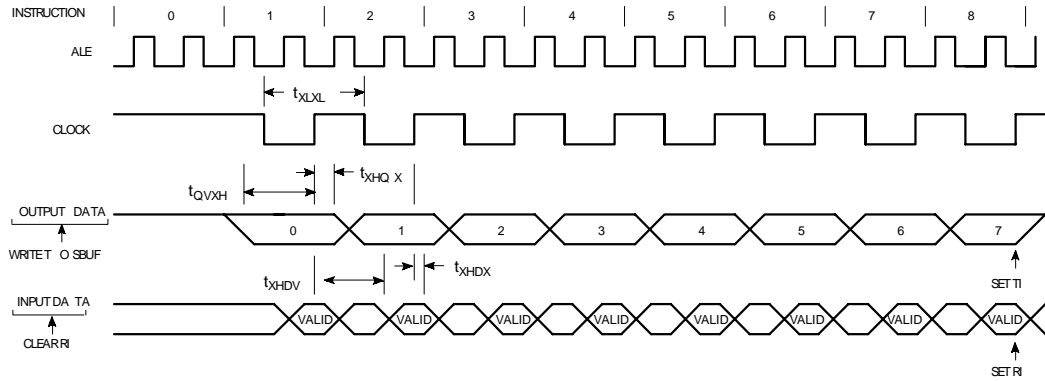


图 44 移位寄存器模式时序

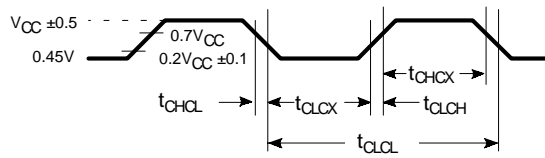
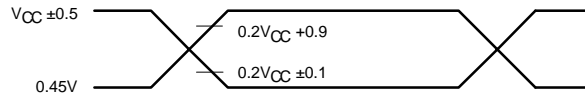


图 45 外部时钟驱动



NOTE:
AC inputs during testing are driven at $V_{CC} \pm 0.5$ for a logic '1' and 0.45V for a logic '0'.
Timing measurements are made at V_{IH} min for a logic '1' and V_{IL} max for a logic '0'.

图 46 AC 输入/输出测试



NOTE:
For timing purposes, a port is no longer floating when a 100mV change from load voltage occurs, and begins to float when a 100mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OH}/I_{OL} \leq +20mA$.

图 47 悬浮波形

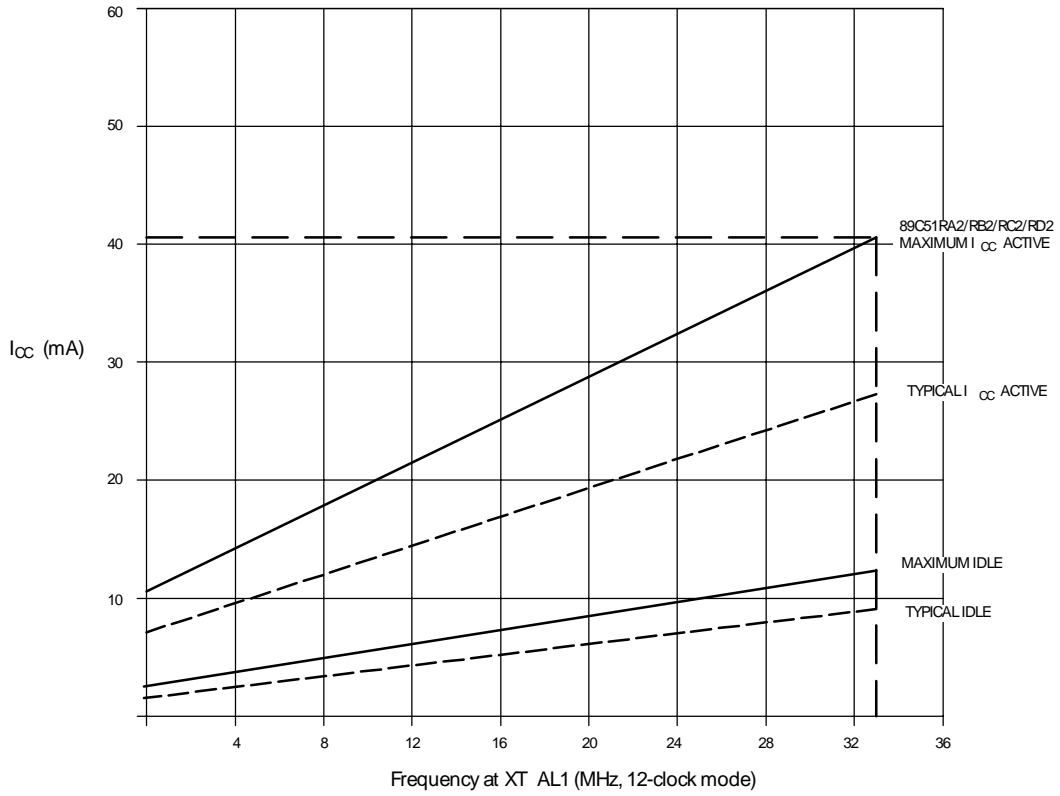


图 48 I_{CC} 与频率的关系(频率范围内测得的有效数据)

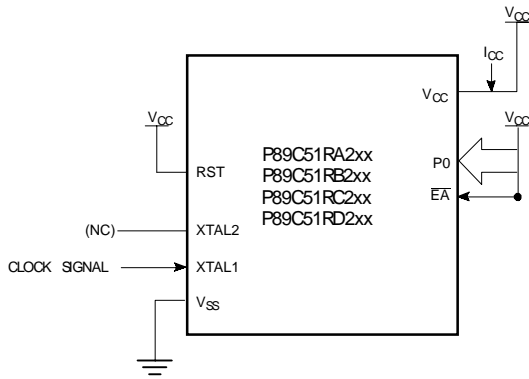


图 49 I_{CC} 测试条件(激活模式;未连接其他管脚)

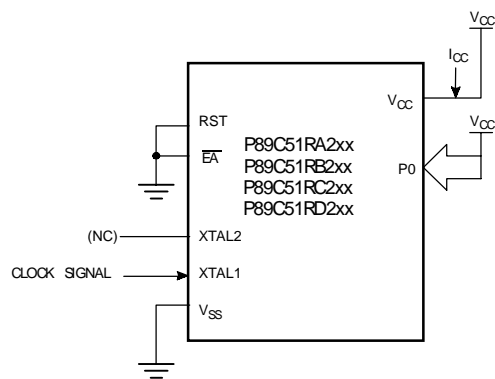


图 50 I_{CC} 测试条件(空闲模式;未连接其他管脚)

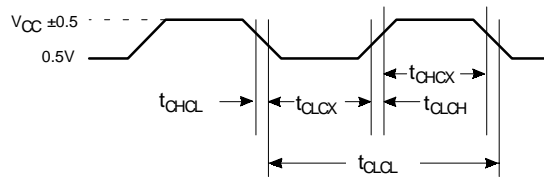


图 51 用于 I_{CC} 测试的时钟信号波形(激活和空闲模式下)

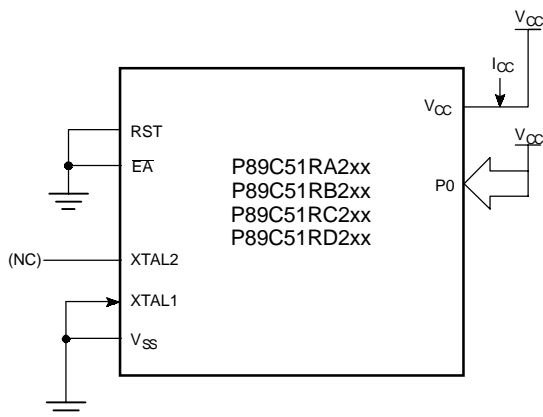
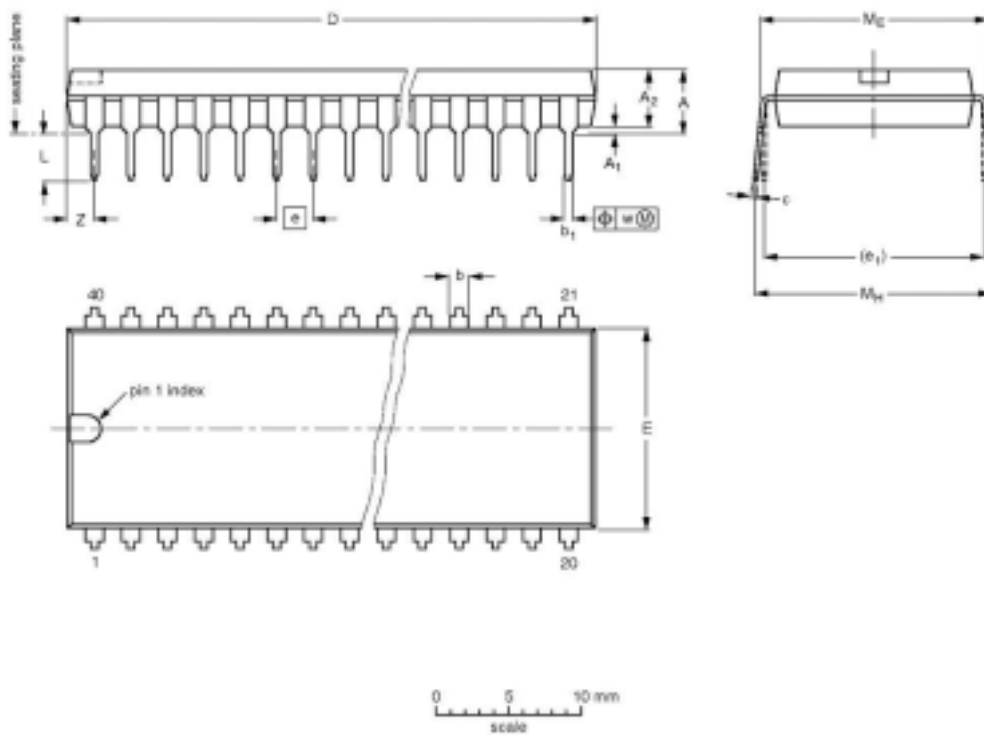


图 52 Icc 测试条件(掉电模式; Vcc=2V~5.5V)

DIP 封装:40 脚



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

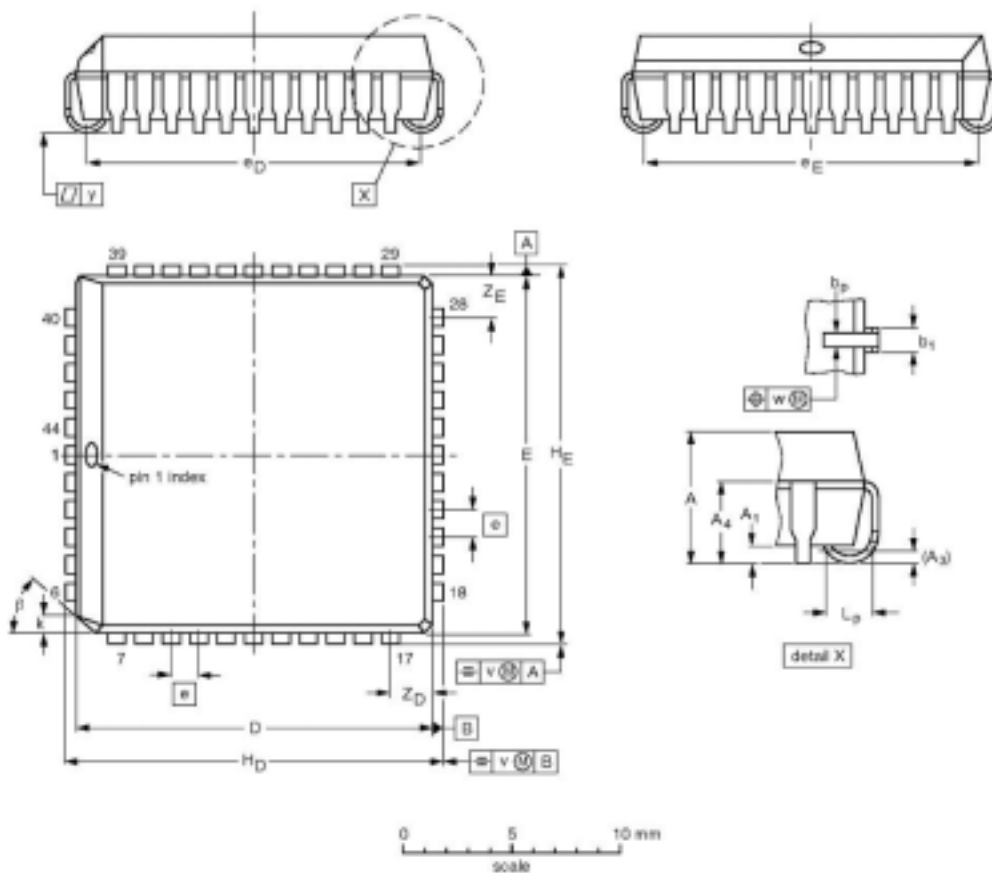
UNIT	A max.	A ₁ min.	A ₂ max.	b	b ₁	e	D ⁽¹⁾	E ⁽¹⁾	e	e ₁	L	M _G	M _H	w	Z ⁽¹⁾ max.
mm	4.7	0.51	4.0	1.70 1.14	0.53 0.28	0.36 0.23	52.90 51.50	14.1 13.7	2.54	15.24	3.60 3.05	15.60 15.24	17.42 15.90	0.254	2.25
inches	0.19	0.020	0.16	0.067 0.045	0.021 0.015	0.014 0.009	2.087 2.028	0.56 0.54	0.10	0.60	0.14 0.12	0.62 0.60	0.69 0.63	0.01	0.089

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT129-1	051G08	MO-015	SC-511-40			95-01-14 95-12-27

PLCC 封装:44 脚



DIMENSIONS (mm dimensions are derived from the original inch dimensions)

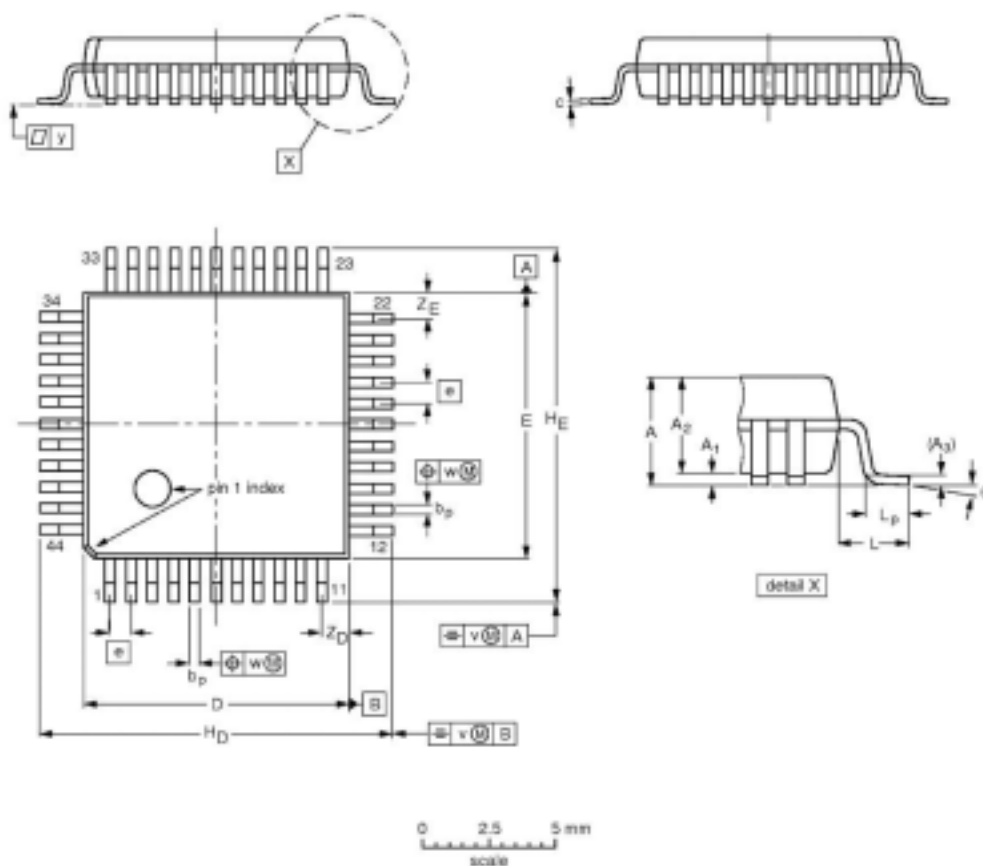
UNIT	A	A ₁ min.	A ₃	A ₄ max.	b _p	b ₁	D ⁽¹⁾	E ⁽¹⁾	e	e _D	e _E	H _D	H _E	k	L _p	v	w	y	Z _D ⁽¹⁾ max.	Z _E ⁽¹⁾ max.	β
mm	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.00 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	1.44 1.02	0.18	0.18	0.1	2.16	2.16	45°
inches	0.180 0.165	0.02	0.01	0.12	0.021 0.013	0.032 0.026	0.650 0.650	0.656 0.650	0.05	0.63 0.59	0.63 0.59	0.695 0.685	0.695 0.685	0.048 0.042	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

Note

1. Plastic or metal protrusions of 0.25 mm (0.01 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA		
SOT187-2	112E10	MS-018	EDR-7319		99-12-27 01-11-14

QFP 封装: 44 脚



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A ₁	A ₂	A ₃	b _p	e	D ⁽¹⁾	E ⁽¹⁾	e	H _D	H _E	L	L _p	v	w	y	Z _D ⁽¹⁾	Z _E ⁽¹⁾	θ
mm	1.60	0.15 0.05	1.45 1.25	0.25	0.45 0.30	0.20 0.12	10.10 9.90	10.10 9.90	0.80	12.15 11.85	12.15 11.85	1.0	0.75 0.45	0.20	0.20	0.10	1.14 0.85	1.14 0.85	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT389-1	138E08	MS-026			99-12-17 00-01-19