# W77E58 规格书

Winbond Electronics Corp.

## 8位微控制器

目 录:

## 1. 概述

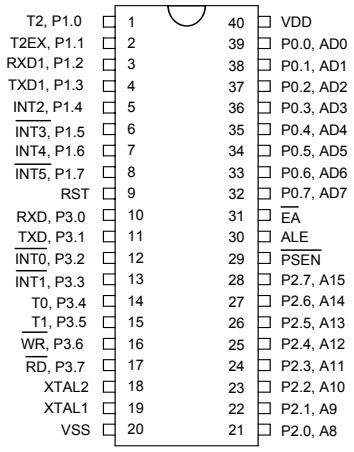
W77E58是一个快速8051 兼容微控制器；它的内核经过重新设计，提高了时钟速度和存储器访问周期速度。经过这种改进以后，在相同的时钟频率下，它的指令执行速度比标准8051 要快许多。一般来说，按照指令的类型，W77E58的指令执行速度是标准8051的1.5-3倍。整体来看，W77E58的速度比标准的8051快2.5倍。在相同的吞吐量及低频时钟情况下，电源消耗也降低。由于采用全静态CMOS设计，W77E58能够在低时钟频率下运行。W77E58内含32KB Flash EPROM，工作电压为4.5v-5.5v，具有 1KB片上外部数据存储器，当用户应用时使用片上SRAM代替外部SRAM，可节省更多I/O口。

## 2. 特性

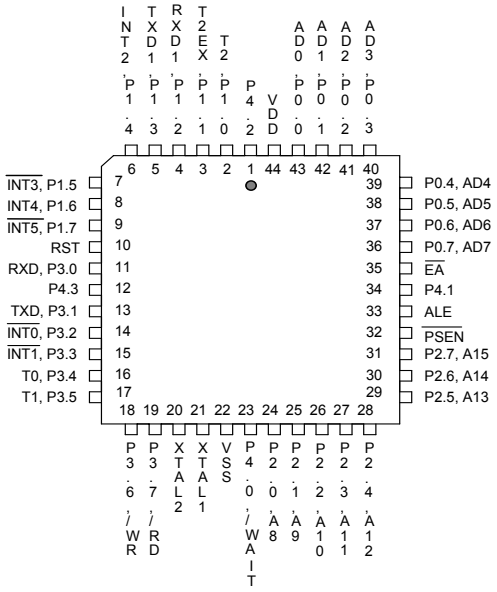
- 8位CMOS微控制器
- 每4个时钟周期为一个机器周期的高速结构，最大外部时钟频率为40MHZ
- 与标准80C52管脚兼容
- 指令与MCS-51兼容
- 4个8位I/O口
- 一个附加的4位I/O口和等待状态控制信号 (仅限 44-脚PLCC/QFP 封装)
- 3个16位定时/计数器
- 12个中断源，2级中断能力
- 片上振荡器及时钟电路
- 二个增强型全双工串行口
- 32KB，Flash EPROM
- 256字节片内暂存RAM
- 片内1KB外部数据存储器（用MOVX指令访问）
- 可编程看门狗定时器
- 软件复位
- 2个16位数据指针
- 对外部RAM及外设的访问周期可以进行软件编程
- 封装:
  - DIP 40: W77E58-40
  - PLCC 44: W77E58P-40
  - QFP 44: W77E58F-40
  - 无铅封装 DIP 40:  W77E058A40DL
  - 无铅封装 PLCC 44: W77E058A40PL
  - 无铅封装 PQFP 44: W77E058A40FL

## 3. 管脚配置

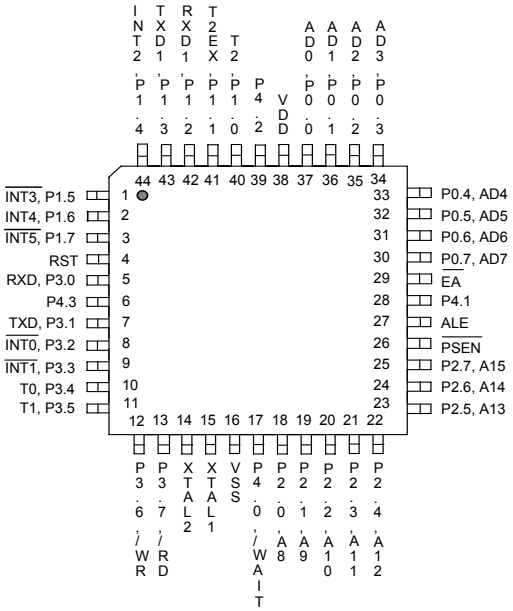
40-Pin DIP (W77E58)




44-Pin PLCC (W77E58P)




44-Pin QFP (W77E58F)



## 4. 管脚描述

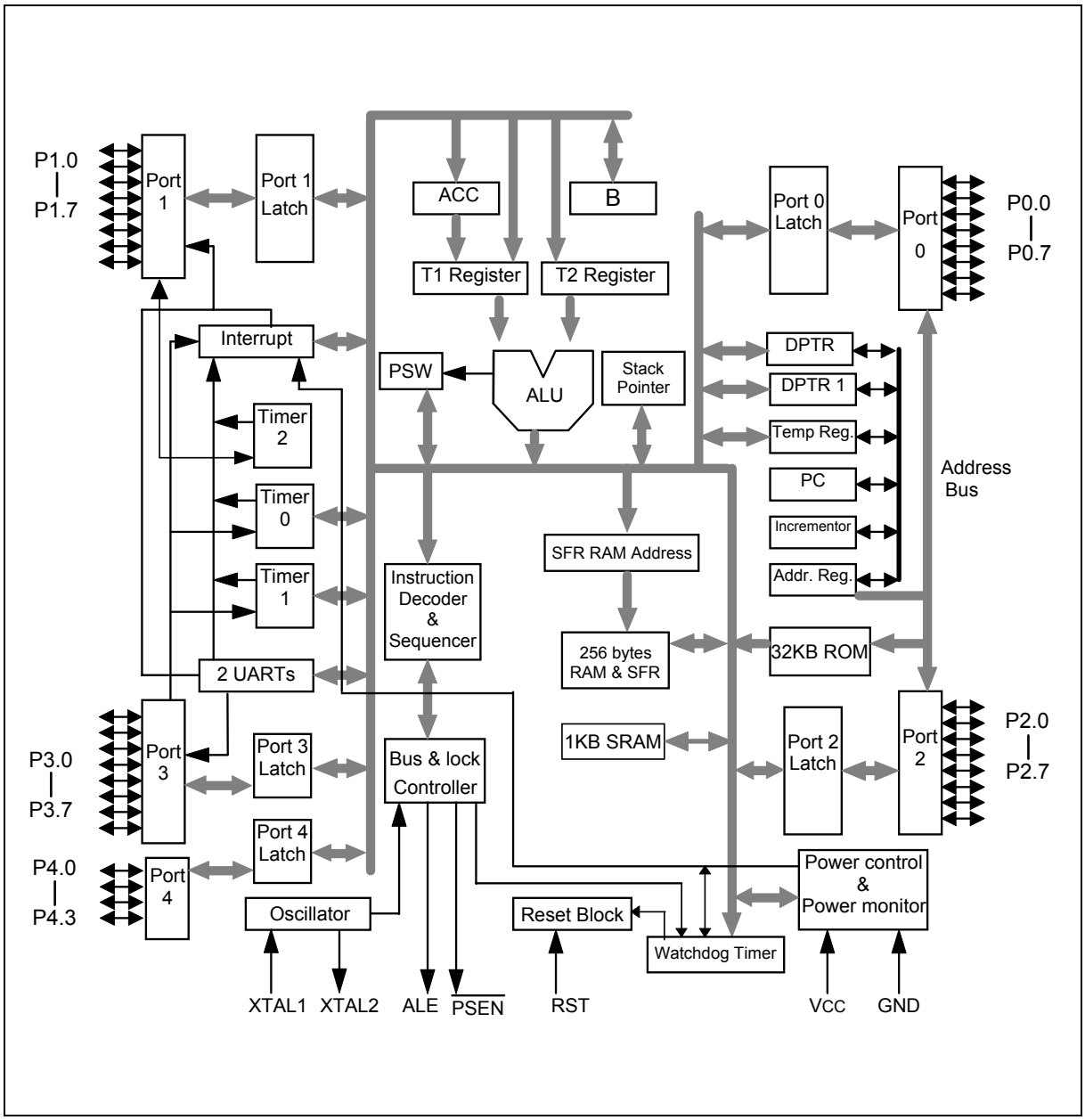
| 符号 | 类型 | 描述 |
|---|---|---|
| $\overline{EA}$ | I | 外部访问使能：此管脚使处理器访问外部ROM。当$\overline{EA}$ 保持高电平时，处理器访问内部ROM。如果$\overline{EA}$ 管脚为高电平且程序计数器指向片内ROM空间，ROM的地址和数据就不会出现在总线上。 |
| $\overline{PSEN}$ | O | 程序存储使能：在执行取指令（fetch）和MOVC的操作时，此管脚允许外部ROM数据出现在P0口的地址/数据总线上。当访问内部ROM时，此管脚上不输出$\overline{PSEN}$ 的选通信号。 |
| ALE | O | 地址锁存使能：ALE用于将P0口地址锁存，使其和数据分离。 |
| RST | I | 复位：振荡器运行时，此管脚上出现两个机器周期的高电平将使器件复位。 |
| XTAL1 | I | 石英晶体1：晶体振荡器的输入。此管脚可由一个外部时钟驱动。 |
| XTAL2 | O | 石英晶体2：晶体振荡器的输出。XTAL2是XTAL1的反相端。 |
| $V_{SS}$ | I | 地：地电位 |
| $V_{DD}$ | I | 电源：电源工作电压 |
| P0.0–P0.7 | I/O | 端口0：端口0是一个双向I/O口，在访问外部存储器时，端口0可用作低位地址/数据总线。端口0 是一个开漏极端口，在进行编程时需要连接一个外部上拉电路。 |
| P1.0–P1.7 | I/O | 端口1：端口1是一个具有内部上拉电路的双向I/O 口。有复用功能位，如下：<br>T2（P1.0）： 定时/计数器2的外部计数输入<br>T2EX（P1.1）:定时/计数器2的重装载/捕获控制<br>RXD1(P1.2): 串行口2 RXD<br>TXD1(P1.3): 串行口2 TXD<br>INT2(P1.4): 外部中断 2<br>$\overline{INT3}$ (P1.5): 外部中断 3<br>INT4(P1.6): 外部中断 4<br>$\overline{INT5}$ (P1.7): 外部中断 5 |
| P2.0–P2.7 | I/O | 端口2：端口2是一个具有内部上拉电路的双向I/O 口。此端口提供访问外部存储器的高位地址。 |

管脚描述，续

| 符号 | 类型 | 描述 |
|---|---|---|
| P3.0–P3.7 | I/O | 端口3：端口3是一个具有内部上拉电路的双向I/O 口。所有位都有复用功能，如下：<br>RXD（P3.0）：串行口接收器输入<br>TXD（P3.1）：串行口发送器输出<br>$\overline{\text{INT0}}$（P3.2）：外部中断0<br>$\overline{\text{INT1}}$（P3.3）：外部中断1<br>T0（P3.4）：定时器0外部输入<br>T1（P3.5）：定时器1外部输入<br>$\overline{\text{WR}}$（P3.6）：外部数据存储器写选通<br>$\overline{\text{RD}}$（P3.7）：外部数据存储器读选通 |
| P4.0–P4.3 | I/O | 端口4：可位寻址的双向I/O 口P4，P4.0也提供 $\overline{\text{WAIT}}$ 功能提供等待控制信号。P4.3 也提供$\overline{\text{REBOOT}}$ 的功能，该功能用来从LD flash 中重启。 |

*注释：类型 I：输入，O：输出，I/O：双向口.

## 5. 方块图


P1.0
P1.7
Port 1
Port 1 Latch
ACC
B
T1 Register
T2 Register
Port 0 Latch
Port 0
P0.0
P0.7
Interrupt
PSW
ALU
Stack Pointer
DPTR
DPTR 1
Temp Reg.
PC
Incrementor
Addr. Reg.
Address Bus
Timer 2
Timer 0
Timer 1
2 UARTs
Instruction Decoder & Sequencer
SFR RAM Address
256 bytes RAM & SFR
32KB ROM
1KB SRAM
P3.0
P3.7
Port 3
Port 3 Latch
Bus & lock Controller
Port 2 Latch
Port 2
P2.0
P2.7
P4.0
P4.3
Port 4
Port 4 Latch
Power control & Power monitor
Oscillator
Reset Block
Watchdog Timer
XTAL1
XTAL2
ALE
PSEN
RST
VCC
GND

## 6. 功能描述

W77E58与8052在管脚及指令集上兼容。它具有8052的资源如：4个双向8位I/O口，3个16位定时器/计数器，全双工串行和若干中断源。

W77E58中建有一个更加快速，性能更好的8位CPU，它的内核经过重新设计，提高了时钟速度和存储器访问周期速度。性能的提高不仅仅在于使用高频的振荡器，还在于W77E58将多数标准的8052指令的机器周期从12个时钟减少至4个时钟。这样性能就提高了1.5-3倍。另外W77E58还可调整MOVX指令的周期，范围为2个机器周期-9个机器周期。这种设计使得W77E58能够更有效的访问慢速或快速外部RAM及外设。W77E58内含1KB用MOVX 指令访问的数据存储器，地址范围为0000H-03FFH。它只能用MOVX指令来访问，可由软件来选择是否使用这个片上SRAM。

W77E58是与8052兼容的，因此具有8052的特性；相比8052它的速度提高，耗电量减少。他的指令集基本与8051相同；多了一条DEC DPTR (操作码 A5H, DPTR减 1)指令。8051每12个时钟周期为一个机器周期，而W77E58每4个时钟周期为一个机器周期。这样提高了W77E58的指令执行速度。因此与8052相比即使在时钟频率相同的情况下W77E58也可以以更高速度运行。由于采用全静态CMOS设计，W77E58能够在低时钟频率下运行，在相同指令吞吐量的情况下，电源消耗也降低。

机器周期缩短至4个时钟周期，是W77E58速度提高的主要原因。W77E58具有所有8052的特性，同时也具有一些新的外设及特性。

**I/O 口**

W77E58有4个8位I/O口，及一个附加的4位I/O口。当处理器用MOVC或MOVX指令执行外部程序、访问外部设备/存储器时，P0口可用作地址/数据总线。此时它内部有强上拉或下拉功能，无须再使用外部上拉。否则它是带有开漏输出的通用I/O口。P2口主要提供16位地址的高8位。当用作地址线时它同样具有强上拉或下拉功能。P1、P3口是I/O口同时具有不同的功能 。P4口（限PLCC/QFP封装）是和P1、P3相同的通用I/O口。P4.0有CP/$\overline{RL2}$ 的复用功能是等待状态中的控制信号。当等待状态控制信号使能后，P4.0是输入口。

**串行口**

W77E58有2个增强型串行口，功能与标准8052串行口相似。W77E58的串行口能以不同的方式运行，以获得时序相似。**注意串行口0可以用定时器1或2做波特率发生器，但串行口1只能用定时器1做波特率发生器**。串行口有自动地址识别和贞错误检测的增强功能。

**定时器**

W77E58有3个16位定时器，其功能与8052体系中的定时器类似。当作为定时器使用时，可将它们设置为每4个时钟周期进行一次计数，或者每12个时钟周期进行一次计数。这位用户提供了模拟8052时钟运行的一种方式。W77E58具有特殊的功能，看门狗定时器。该定时器可用作系统监控器，或超长周期定时器。

**中断**

W77E58的中断系统与标准8052之中断系统有细微的差别。由于存在新增功能和外设，中断源的数量和中断向量都相应得增加。W77E58提供12个中断源2级中断能力，包括6个外部中断，定时器中断及串行I/O口中断。

### 数据指针

在标准8052中只有一个16位数据指针（DPL，DPH）。在W77E58中还有一个16位数据指针（DPL1，DPH1）。这个数据指针位于标准8052中未定义的SFR地址中。W77E58中还有一条DEC DPTR指令（操作码 A5H），用以提高程序的灵活性。

### 电源管理

类似于标准80C52，W77E58有空闲和掉电2种节电方式。另外W77E58还提供一个新的称为**经济**模式的节电方式，它允许用户将时钟频率进行4、64或1024的分频。在空闲模式下，CPU核停止工作，而定时器、串行口、中断时钟继续运行。在掉电模式下，所有时钟停止工作，芯片运行完全停止，是最省电的运行模式。

### 片上数据SRAM

W77E58有1K字节的数据SRAM空间，它是可读写的并且是存储器映射的。这些片上MOVX SRAM用MOVX指令来访问。这片区域不用于存放可执行代码。对于片内256字节暂存RAM和这些1K字节数据SRAM来说，不存在数据的冲突和重叠，因为他们有不同的寻址方式和单独的访问指令。PMR寄存器中的DME0位来使能片上MOVX SRAM，在复位后DME0位为0，因此MOVX SRAM是被关闭的，所有对0000H-FFFFH地址空间的访问均为对外部SRAM的访问。

## 7. 存储器组织

W77E58将存储器分为2个独立的区域：程序存储器区和数据存储器区。程序存储器区用来存放程序代码，数据存储器区用来存放数据及存储器映射的设备需要用到的数据。

### 7.1 程序存储器

W77E58提供32KB大小的程序存储器，这些ROM区与8052的ROM区功能类似，所有指令都从这些区域中取出执行。MOVC指令同样也访问这些区域，超过片上ROM最大地址范围后，系统将访问外部存储器。

### 7.2 数据存储器

W77E58最多可以访问64KB的外部数据存储器。这个存储器区域用MOVX指令来访问。不同于其他8051的衍生产品，W77E58还内建一个1KB字节的MOVX SRAM数据存储器。这1KB的数据存储器的地址范围为0000H-03FFH。对该数据存储器的访问是受软件控制的。当软件允许访问该区域时，访问地址范围为0000H-03FFH的MOVX指令将读写MOVX SRAM数据存储器的内容。当地址范围超过03FFH后，系统将自动访问外部数据存储器。当软件禁止访问该区域时，该区域将被映射为外部数据存储器。任何访问地址为0000H-FFFFH的MOVX指令都将访问到外部数据存储器。这是W77E58默认的运行环境。另外W77E58还有标准的256字节暂存数据存储器。这片区域可以间接或直接访问。由于这片区域只有256字节，因此仅适用于数据量较小的场合。当数据量较多时，可以考虑同时使用2个数据存储器。片上MOVX SRAM，同外部RAM一样只可由MOVX指令来访问，但是片上MOVX SRAM拥有最快的访问速度。

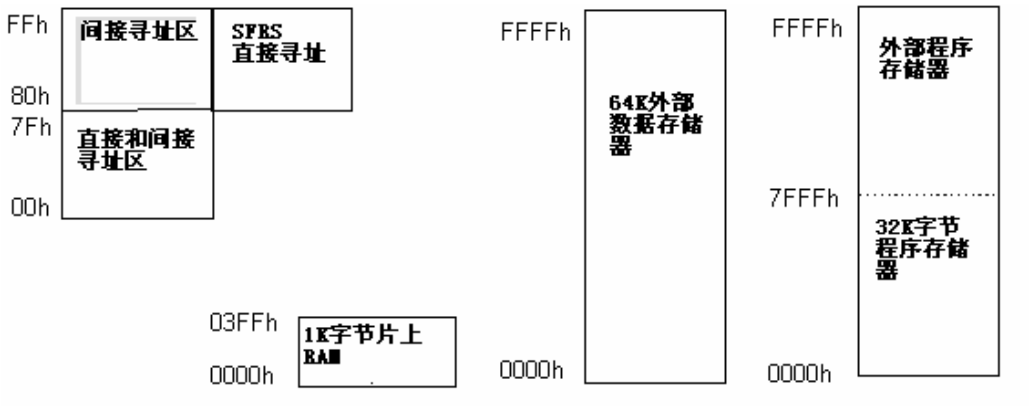



图1;存储器映射

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| FFh<br>80h | Indirect RAM | | | | | | | |
| 7Fh<br>30h | Direct RAM | | | | | | | |
| 2Fh | 7F | 7E | 7D | 7C | 7B | 7A | 79 | 78 |
| 2Eh | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| 2Dh | 6F | 6E | 6D | 6C | 6B | 6A | 69 | 68 |
| 2Ch | 67 | 66 | 65 | 64 | 63 | 62 | 61 | 60 |
| 2Bh | 5F | 5E | 5D | 5C | 5B | 5A | 59 | 58 |
| 2Ah | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| 29h | 4F | 4E | 4D | 4C | 4B | 4A | 49 | 48 |
| 28h | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| 27h | 3F | 3E | 3D | 3C | 3B | 3A | 39 | 38 |
| 26h | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
| 25h | 2F | 2E | 2D | 2C | 2B | 2A | 29 | 28 |
| 24h | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 23h | 1F | 1E | 1D | 1C | 1B | 1A | 19 | 18 |
| 22h | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| 21h | 0F | 0E | 0D | 0C | 0B | 0A | 09 | 08 |
| 20h | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| 1Fh<br>18h | Bank 3 | | | | | | | |
| 17h<br>10h | Bank 2 | | | | | | | |
| 0Fh<br>08h | Bank 1 | | | | | | | |
| 07h<br>00h | Bank 0 | | | | | | | |

Bit Addressable
20H - 2FH

图2. 暂存 RAM/寄存器寻址

## 8. 特殊功能寄存器

W77E58用特殊功能寄存器(SFRs)来控制监测系统运行和系统的模式。

特殊功能寄存器位于80H-FFH的地址空间内，只能用直接寻址的方式来访问。一些特殊功能寄存器是可位寻址的，这个功能特别适用于只想修改寄存器中的某一位而不影响其他位的场合。可位寻址的特殊功能寄存器，其地址编号是以0或8结尾。W77E58中含有标准8052中所有的特殊功能寄存器，同时也加入了一些新的特殊功能寄存器。在一些应用场合，8052中未被定义的位被赋予了新的功能。下表列出了W77E58中的特殊功能寄存器，每行分了9列。空白项表示该地址空间没有SFR存在，对这些空间的访问将会得到全1的结果。

**表1：特殊功能寄存器列表**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F8 | EIP | | | | | | | |
| F0 | B | | | | | | | |
| E8 | EIE | | | | | | | |
| E0 | ACC | | | | | | | |
| D8 | WDCON | | | | | | | |
| D0 | PSW | | | | | | | |
| C8 | T2CON | T2MOD | RCAP2L | RCAP2H | TL2 | TH2 | NVMCON | NVMDAT |
| C0 | SCON1 | SBUF1 | WSCON | | PMR | STATUS | NVMSEL | TA |
| B8 | IP | SADEN | SADEN1 | | | | | |
| B0 | P3 | | | | | | | |
| A8 | IE | SADDR | SADDR1 | ROMCON | SFRAL | SFRAH | SFDFD | SFRCN |
| A0 | P2 | | P4CSIN | | | P4 | | |
| 98 | SCON0 | SBUF | P42AL | P42AH | P43AL | P43AH | | CHPCON |
| 90 | P1 | EXIF | P4CONA | P4CONB | P40AL | P40AH | P41AL | P41AH |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | |
| 80 | P0 | SP | DPL | DPH | DPL1 | DPH1 | DPS | PCON |

注释：有加粗边框的那一列为可位寻址的特殊功能寄存器

特殊功能寄存器简介：

**端口0**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |

助记符：P0　　地址： 80h

端口0是一个开漏双向I/O口，在访问外部存储器时，它会分时输出16位地址的低位字节和8位数据。

**堆栈指针**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SP.7 | SP.6 | SP.5 | SP.4 | SP.3 | SP.2 | SP.1 | SP.0 |

助记符：SP　　地址： 81h

堆栈指针存储暂存RAM中堆栈的起始地址，就是说他总指向栈顶。

**数据指针低字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPL.7 | DPL.6 | DPL.5 | DPL.4 | DPL.3 | DPL.2 | DPL.1 | DPL.0 |

助记符：DPL　　地址： 82h

标准8052中16位数据指针的低字节。

**数据指针高字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPH.7 | DPH.6 | DPH.5 | DPH.4 | DPH.3 | DPH.2 | DPH.1 | DPH.0 |

助记符：DPH　　地址： 83h

标准8052中16位数据指针的高字节。

**数据指针1低字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPL1.7 | DPL1.6 | DPL1.5 | DPL1.4 | DPL1.3 | DPL1.2 | DPL1.1 | DPL1.0 |

助记符：DPL1　　地址： 84h

W77E58种另一个16位数据指针的低字节，用户可以通过设置DPS在DPL, DPH 和 DPL1, DPH1之间切换。DPS=1时，DPTR指令将访问DPL1和DPH1。如果不需要使用数据指针，那么用户也可将它们用作一般的数据寄存器。

**数据指针1高字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPH1.7 | DPH1.6 | DPH1.5 | DPH1.4 | DPH1.3 | DPH1.2 | DPH1.1 | DPH1.0 |

助记符：DPH1　　地址： 85h

W77E58种另一个16位数据指针的高字节，用户可以通过设置DPS在DPL, DPH 和 DPL1, DPH1之间切换。DPS=1时，DPTR指令将访问DPL1和DPH1。如果不需要使用数据指针，那么用户也可将它们用作一般的数据寄存器。

**数据指针选择寄存器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | DPS.0 |

助记符：DPS　　地址： 86h

DPS.0: 该位用来选择是用DPL，DPH还是使用DPL1，DPLH1。DPS=1 时使用DPL1，DPLH1，否则使用DPL，DPH。

DPS.1-7:保留位，读出后为0

**电源控制**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0D | SMOD0 | - | - | GF1 | GF0 | PD | IDL |

助记符：PCON　　地址： 87h

SMOD: 该位置1时，会使串行口在模式1，2，3下的波特率加倍

SMOD0: 贞错误检测使能：该位置1时，SCON.7表示一个贞错误它是FE（贞错误）标志。当该位0，SCON.7的功能与标准8052中SCON.7相同。

GF1-0: 这2位是通用的标志位。

PD: 将该位置1，系统进入掉电模式；该模式下，所有时钟停止工作，程序也不再执行。

IDL: 将该位置1，系统进入空闲模式；该模式下，CPU的时钟停止工作，程序停止运行；但串口、定时器、中断的时钟没有停止，这些功能模块仍正常运行。

**定时器控制**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |

助记符：TCON　　地址： 88h

TF1： 定时器1溢出标志；在定时器1溢出时该位置1。当程序响应定时器1中断执行相应的中断服务程序时，该位自动清0。软件也可对该位置位或复位。

TR1： 定时器1启动控制：该位由软件来置位或清零来启动或关闭定时器。

TF0： 定时器0溢出标志；在定时器1溢出时该位置1。当程序响应定时器0中断执行相应的中断服务程序时，该位自动清0。软件也可对该位置位或复位。

TR0： 定时器0启动控制：该位由软件来置位或清零来启动或关闭定时器。

IE1： 外部中断1标志；当$\overline{INT1}$上出现电平跳变时由硬件置1；若被设置为下沿触发并导致中断转跳的话，会自动清除为0，否则完全根据外部中断。

IT1: 外部中断1触发方式控制；1：低电平边沿触发； 0：低电平触发

IE0： 外部中断0标志；当$\overline{INT0}$上出现电平跳变时由硬件置1；若被设置为下沿触发并导致中断转跳的话，会自动清除为0，否则完全根据外部中断。

IT0: 外部中断0触发方式控制；1：低电平边沿触发； 0：低电平触发

**定时器模式控制**

位：

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| GATE | C / $\overline{T}$ | M1 | M0 | GATE | C / $\overline{T}$ | M1 | M0 |
| TIMER1 | | | | TIMER0 | | | |

助记符：TMOD 地址： 89h

GATE: 门控位为1时，定时器/计数器的运行除受TRx控制外还受$\overline{INTx}$ 控制，当TRx和$\overline{INTx}$ 均为1时定时器/计数器开始运行。该位为0时，定时器的运行只受TRx的控制。

C / $\overline{T}$: 定时器/计数器工作方式选择：为0时以定时器的方式运行；为1时对TX脚上的高到低电平变化进行计数。

**M1, M0: 模式选择位：**

| M1 | M0 | 模式 |
|---|---|---|
| 0 | 0 | 模式 0: 13位定时器 |
| 0 | 1 | 模式 1: 16位定时器 |
| 1 | 0 | 模式 2: 8位自动重装地定时器，重装值位于THx中 |
| 1 | 1 | 模式3: （仅适用于T0 ）TL0是受定时器0 控制的8位定时器/计数器。TH0 是受定时器1控制的8位定时器/计数器。定时器1在此方式下不工作。 |

**定时器0 LSB**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL0.7 | TL0.6 | TL0.5 | TL0.4 | TL0.3 | TL0.2 | TL0.1 | TL0.0 |

助记符：TL0　　地址： 8Ah

TL0.7–0: Timer 0 LSB

**定时器1 LSB**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL1.7 | TL1.6 | TL1.5 | TL1.4 | TL1.3 | TL1.2 | TL1.1 | TL1.0 |

助记符：TL1　　地址： 8Bh

TL1.7–0: Timer 1 LSB

**定时器0 MSB**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH0.7 | TH0.6 | TH0.5 | TH0.4 | TH0.3 | TH0.2 | TH0.1 | TH0.0 |

助记符：TH0　　地址： 8Ch

TH0.7–0: Timer 0 MSB

**定时器1 MSB**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH1.7 | TH1.6 | TH1.5 | TH1.4 | TH1.3 | TH1.2 | TH1.1 | TH1.0 |

助记符：TH1　　地址： 8Dh

TH1.7–0: Timer 1 MSB

**时钟控制**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WD1 | WD0 | T2M | T1M | T0M | MD2 | MD1 | MD0 |

助记符：CKCON　　地址： 8Eh

WD1-0: 看门狗定时器模式选择位: 这些位决定看门狗定时器的溢出时间。对4个溢出时间选项来说，系统复位时间是看门狗定时器溢出后+512个时钟。

| WD1 | WD0 | Interrupt time-out | Reset time-out |
|---|---|---|---|
| 0 | 0 | $2^{17}$ | $2^{17}$ + 512 |
| 0 | 1 | $2^{20}$ | $2^{20}$ + 512 |
| 1 | 0 | $2^{23}$ | $2^{23}$ + 512 |
| 1 | 1 | $2^{26}$ | $2^{26}$ + 512 |

T2M: 定时器2时钟选择：为1 时定时器2 的时钟源是系统时钟源的4分频，为0时定时器2的时钟源是系统时钟源的12分频。

T1M: 定时器1时钟选择：为1 时定时器1的时钟源是系统时钟源的4分频，为0时定时器2的时钟源是系统时钟源的12分频。

T0M: 定时器0时钟选择：为1时定时器0的时钟源是系统时钟源的4分频，为0时定时器0的时钟源是系统时钟源的12分频。

MD2–0: MOVX指令周期选择：这3位用来选择MOVX指令的周期；MOVX指令周期可变使得用户无需增加额外电路就可访问慢速外部存储器或设备。$\overline{RD}$ 和 $\overline{WR}$ 信号周期也会有相应的变化。当访问片上SRAM时，MOVX的指令周期总是2个机器周期而不管MID2-0如何设置。MID2-0 的默认值是1，如果用户希望提高访问速度那么可以将MID2-0设为0。

| MD2 | MD1 | MD0 | Stretch 值 | MOVX 周期 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 2 机器周期 |
| 0 | 0 | 1 | 1 | 3 机器周期 *(默认)* |
| 0 | 1 | 0 | 2 | 4 机器周期 |
| 0 | 1 | 1 | 3 | 5 机器周期 |
| 1 | 0 | 0 | 4 | 6 机器周期 |
| 1 | 0 | 1 | 5 | 7 机器周期 |
| 1 | 1 | 0 | 6 | 8 机器周期 |
| 1 | 1 | 1 | 7 | 9 机器周期 |

**端口1**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |

助记符：P1　　地址： 90h

P1.7–0: 通用　I/O端口。在端口读访问时多数指令会对这个端口进行读操作。对于读-修改-写指令，对这个端口的操作时读。这个端口的一些管脚具有复用功能。

P1.0 : T2　　定时器2的外部I/O口
P1.1 : T2EX　　T2EX定时器2捕捉/重装触发
P1.2 : RXD1　　串行口1接收
P1.3 : TXD1　　串行口1发送
P1.4 : INT2　　外部中断 2
P1.5 : $\overline{\text{INT3}}$　　外部中断 3
P1.6 : INT4　　外部中断 4
P1.7 : $\overline{\text{INT5}}$　　外部中断 5

**外部中断 标志**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IE5 | IE4 | IE3 | IE2 | XT/RG | RGMD | RGSL- | - |

助记符：EXIF　　地址： 91h

IE5: 外部中断 5 标志. 在$\overline{\text{INT5}}$ 上检测到负跳变后由硬件置位.

IE4: 外部中断 4 标志. 在INT4上检测到负跳变后由硬件置位.

IE3: 外部中断 3 标志. 在$\overline{\text{INT3}}$ 上检测到负跳变后由硬件置位.

IE2: 外部中断 2 标志. 在INT2上检测到负跳变后由硬件置位.

XT/RG:　晶体/RC振荡器选择。该位置位后系统选择晶振或外部时钟来作为系统时钟。将该位清0选择片内RC振荡器为时钟源。XTUP(STATUS.4)和XTOFF（PMR.3）必须在该位置1前清零。不按此规则来做的操作将被视为无效操作。该位在上电复位后会被置1，它不会被其他形式的复位改变。

RGMD:　RC模式寄存器。该位指明当前微控制器的时钟来源。为0时CPU的时钟来源是外部晶振或振荡器。为1时时钟来源是片内RC电路。在上电复位后该位为0，且不受其他形式复位的影响。

RGSL:　RC振荡器选择。该位选择从掉电模式恢复后，系统的时钟来源。为1则系统从掉电模式恢复后，RC振荡器用作时钟来源；为0 则系统在等待晶体电起振后，将外部晶振作为系统时钟来源。

**串行口控制寄存器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0/FE | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |

助记符：SCON　　地址： 98h

SM0/FE: 串行口0，模式0控制位或贞错误标志位。PCON特殊功能寄存器中的SMOD0位决定该位的功能。下面会描述SM0的运行功能。当用作贞错误标志时，该位的置位表示一个无效的停止位。该位必须由软件来清除。

SM1: 串行口模式位1:

| SM0 | SM1 | 模式 | 说明 | 数据长度 | 波特率 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 同步 | 8 | 时钟的4或12分之一 |
| 0 | 1 | 1 | 异步 | 10 | 可变 |
| 1 | 0 | 2 | 异步 | 11 | 时钟的64或32分之一 |
| 1 | 1 | 3 | 异步 | 11 | 可变 |

SM2: 多机通信控制。将该位置1，则使能模式2及模式3下的多机通信功能。在模式2或3下，如果SM2置1，那么收到的第九位数据RB8是0的话，RI将不会置位。在模式1下如果SM2置1，那么在没有收到有效的停止位前RI是不会置位的。在模式0下，SM2位控制着串行口的时钟。如果清0，那么串行口的时钟是系统时钟的12分频。这样系统就与标准8052兼容。如果该位置1，那么串行口的时钟是系统时钟的4分频，这样就加快了同步通信的速度。

REN: 接收使能，置1时打开串行口接收功能，否则关闭该功能。

TB8: 模式2和3中要被发送的第九位数据。软件可以根据需求将该位置1或清0。

RB8: 模式2和3中接收到的第九位数据。模式1下，若SM2=0则RB8是接收到的停止位。模式0下该位无意义。

TI: 发送中断标志：模式0下该标志由硬件在发送完8位数据后置位，而在其他模式下在串行发送到停止位的开始时置位。该位必须由软件来清除.

RI: 接收中断标志：模式0下该标志由硬件在接收到8位数据后置位，而在其他模式下在串行接收到停止位的中间时置位。该位必须由软件来清除。

**串行数据缓冲寄存器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SBUF.7 | SBUF.6 | SBUF.5 | SBUF.4 | SBUF.3 | SBUF.2 | SBUF.1 | SBUF.0 |

助记符：SBUF 地址： 99h

SBUF.7-0: 串行口0接收或发送的数据都放在这个寄存器中。实际上该地址上有2个独立的8位寄存器。一个用于接收数据，一个用于发送数据。对它进行读操作将会接收串行数据，对它进行写操作则发送串行数据。

**端口 2**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |

助记符：P2 地址： A0h

P2.7-0: 端口2是内部有上拉的双向I/O口，在访问外部存储器时输出高8位地址。

**端口4**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | P4.3 | P4.2 | P4.1 | P4.0 |

助记符：P4　　地址： A5h

P4.3-0: 端口4 是内部有上拉的双向I/O口。端口4 不可以位寻址（不能对其使用SETB或CLR指令）

**中断使能**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EA | ES1 | ET2 | ES | ET1 | EX1 | ET0 | EX0 |

助记符：IE　　地址： A8h

EA:　中断总控制位。使能/关闭所有中断。
ES1:　使能串口1中断
ET2:　使能定时器2 中断.
ES:　使能串口0 中断t.
ET1:　使能定时器1中断
EX1:　使能外部中断1
ET0:　使能定时器0中断
EX0:　使能外部中断0

**从机地址**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

助记符：SADDR　　地址： A9h

SADDR:  SADDR 中应当写入串行口0进行多机通信时的广播地址或是从机的地址。

**从机地址 1**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

助记符：SADDR1　　地址： AAh

SADDR1: SADDR1中应当写入串行口1进行多机通信时的广播地址或是从机的地址。

**端口3**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |

助记符：P3　　地址： B0h

P3.7-0: 通用I/O口，每个管脚也有其相应的复用功能。复用功能描述如下：

| | | |
|---|---|---|
| P3.7 | $\overline{RD}$ | 访问外部RAM时的读信号 |
| P3.6 | $\overline{WR}$ | 访问外部RAM时的写信号 |
| P3.5 | T1 | 定时/计数器1外部输入信号 |
| P3.4 | T0 | 定时/计数器0外部输入信号 |
| P3.3 | $\overline{INT1}$ | 外部中断输入1 |
| P3.2 | $\overline{INT0}$ | 外部中断输入0 |
| P3.1 | TxD | 串行口0输出 |
| P3.0 | RxD | 串行口0输入 |

**中断优先级**

位：

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | PS1 | PT2 | PS | PT1 | PX1 | PT0 | PX0 |

助记符：IP　　地址： B8h

IP.7: 该位无意义对该位的访问将读到高电平

PT2: PT2 = 1 将定时器2中断设为高优先级

PS: PS = 1将串行口0设为高优先级

PT1: PT1 = 1将定时器1中断设为高优先级

PX1: PX1 = 1将外部中断1设为高优先级

PT0: PT0 = 1将定时器0中断设为高优先级

PX0: PX0 = 1将外部中断1设为高优先级

**从机地址屏蔽使能**

位：

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

助记符：SADEN　　地址： B9h

SADEN: 该寄存器使能串口0的自动地址识别功能，当SADEN中的某位被置为1，那么SADDR寄存器中的相应位会同接收到到的数据进行比较。如果SADEN.n被设为0，那么系统会忽略对该位的比较。如果SADEN为全0，那么对于所有的地址贞系统都会产生中断。

**从机地址屏蔽使能1**

位：

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

助记符：SADEN1　　地址： BAh

SADEN1: 该寄存器使能串口0的自动地址识别功能，当SADEN1中的某位被置为1，那么SADDR1寄存器中的相应位会同接收到到的数据进行比较。如果SADEN.n被设为0，那么系统会忽略对该位的比较。如果SADEN为全0，那么对于所有的地址贞系统都会产生中断。

**串行口控制1**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0_1/FE_1 | SM1_1 | SM2_1 | REN_1 | TB8_1 | RB8_1 | TI_1 | RI_1 |

助记符：SCON1　　地址： C0h

SM0_1/FE_1: 串行口1，模式0控制位或贞错误标志位。PCON特殊功能寄存器中的SMOD0位决定该位的功能。下面会描述SM0_1的运行功能。当用作贞错误标志时，该位的置位表示一个无效的停止位。该位必须由软件来清除。

SM1_1: 串行口模式位1:

| SM0_1 | SM1_1 | 模式 | 说明 | 数据长度 | 波特率 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Synchronous | 8 | 4/12 Tclk |
| 0 | 1 | 1 | Asynchronous | 10 | variable |
| 1 | 0 | 2 | Asynchronous | 11 | 64/32 Tclk |
| 1 | 1 | 3 | Asynchronous | 11 | variable |

SM2_1: 多机通信控制。将该位置1，则使能模式2及模式3下的多机通信功能。在模式2或3下，如果SM2_1置1，那么收到的第九位数据RB8_1是0的话，RI将不会置位。在模式1下如果SM2_1置1，那么在没有收到有效的停止位前RI是不会置位的。在模式0下，SM2_1位控制着串行口的时钟。如果清0，那么串行口的时钟是系统时钟的12分频。这样系统就与标准8052兼容。如果该位置1，那么串行口的时钟是系统时钟的4分频，这样就加快了同步通信的速度。

REN_1: 接收使能，置1时打开串行口接收功能，否则关闭该功能。

TB8_1: 模式2和3中要被发送的第九位数据。软件可以根据需求将该位置1或清0。

RB8_1: 模式2和3中接收到的第九位数据。模式1下，若SM2_1=0则RB8是接收到的停止位。模式0下该位无意义。

TI_1: 发送中断标志：模式0下该标志由硬件在发送完8位数据后置位，而在其他模式下在串行发送到停止位的开始时置位。该位必须由软件来清除.

RI_1: 接收中断标志：模式0下该标志由硬件在接收到8位数据后置位，而在其他模式下在串行接收到停止位的中间时置位。该位必须由软件来清除。

**串行数据接收缓冲 1**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SBUF1.7 | SBUF1.6 | SBUF1.5 | SBUF1.4 | SBUF1.3 | SBUF1.2 | SBUF1.1 | SBUF1.0 |

助记符：SBUF1　　地址： C1h

SBUF1.7-0: 串行口1接收或发送的数据都放在这个寄存器中。实际上该地址上有2个独立的8位寄存器。一个用于接收数据，一个用于发送数据。对它进行读操作将会接收串行数据，对它进行写操作则发送串行数据。

**ROMMAP**

| 位 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WS | 1 | - | - | - | 1 | 1 | 0 |

助记符：ROMMAP　　地址： C2h

WS: 等待状态信号使能，将该位置位将使能P4.0上的 $\overline{\text{WAIT}}$ 信号。系统会在MOVX指令执行期间通过P4.0对 $\overline{\text{WAIT}}$ 信号进行采样该位是受时控保护的。

```
TA      REG   C7H
ROMMAP  REG   C2H
CKCON   REG   8EH
ANL     CKCON,#11111000B   ; 设stretch 值为0
MOV     TA,#AAH
MOV     TA,#55H
ORL     ROMMAP,#10000000B ; 设WS位stretch值为0打开等待信号
```

**电源管理寄存器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CD1 | CD0 | SWB | - | - | ALE-OFF | - | DME0 |

助记符：PMR　　地址： C4h

CD1, CD0: 时钟分频选择位。这些位用来选择产生机器周期的时钟数。有3种时钟数供选择4，64，或1024。在模式切换期间，首先要回到4时钟数模式。例如要在64时钟和1024时钟数间切换，那么必须先回到4时钟数模式，在从4时钟数模式进入1024时钟数模式。

| CD1, | CD0 | 时钟数/机器周期 |
|---|---|---|
| 0 | 0 | 保留 |
| 0 | 1 | 4 |
| 1 | 0 | 64 |
| 1 | 1 | 1024 |

SWB: 切换使能。该位置位后会使系统在外部中断或串行口信号产生后使系统的机器周期变回4个时钟周期。系统在外部中断打开且确实发生一次中断，系统进入中断向量或是串行口上出现起始位有效负跳变后，将系统机器周期切换回4个时钟周期。.

ALE0FF: 该位置1 后，当系统不访问外部程序和数据存储器时系统不会发出ALE信号。当访问外部存储器时，系统会自动产生ALE信号而不管此时ALEOFF是否置位。

0 = ALE 信号不被关闭;1 = ALE 信号被关闭

DME0: 该位决定用户是否可以访问片上“MOVX SRAM”该位置一后系统便可以访问片上SRAM。

SWB: 切换使能。该位置位后会使系统在外部中断或串行口信号产生后使系统的机器周期变回4个时钟周期。系统在外部中断打开且确实发生一次中断，系统进入中断向量或是串行口上出现起始位有效负跳变后，将系统机器周期切换回4个时钟周期。.

ALE0FF: 该位置1 后，当系统不访问外部程序和数据存储器时系统不会发出ALE信号。当访问外部存储器时，系统会自动产生ALE信号而不管此时ALEOFF是否置位。

0 = ALE 信号不被关闭;1 = ALE 信号被关闭

DME0: 该位决定用户是否可以访问片上“MOVX SRAM”该位置一后系统便可以访问片上SRAM。

**状态寄存器：**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | HIP | LIP | XTUP | SPTA1 | SPRA1 | SPTA0 | SPRA0 |

助记符：STATUS 地址： C5h

HIP: 高优先级中断状态。置位时表示软件正在执行一个高优先级中断服务，当遇到相应的RTI指令后，该位会被清零。

LIP: 低优先级中断状态。置位时表示软件正在执行一个低优先级中断服务，当遇到相应的RTI指令后，该位会被清零。

XTUP: 晶体振荡器起振标志。该位置位后表示系统已经探测到稳定的时钟信号。系统每次从掉电模式退出后，晶体振荡器都会重新启动，硬件会将该位清零。在上电复位后该位置1。

SPTA1: 串行口1发送进行时。当串行口1发送数据时该位置位。当硬件将TI置位后该位清零。当SWB=1且该位置一时对CD1，CD0的改变将会被忽略。

SPRA1: 串行口1接收进行时。当串行口1接收数据时该位置位。当硬件将RI置位后该位清零。当SWB=1且该位置一时对CD1，CD0的改变将会被忽略。

SPTA0: 串行口0发送进行时。当串行口0发送数据时该位置位。当硬件将TI置位后该位清零。当SWB=1且该位置一时对CD1，CD0的改变将会被忽略。

SPRA0: 串行口0接收进行时。当串行口0接收数据时该位置位。当硬件将RI置位后该位清零。当SWB=1且该位置一时对CD1，CD0的改变将会被忽略。

**时控访问**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TA.7 | TA.6 | TA.5 | TA.4 | TA.3 | TA.2 | TA.1 | TfA.0 |

助记符：TA 地址： C7h

TA: 时控访问寄存器用于控制对保护位的访问。要访问被保护的位，用户首先要向TA寄存器写入AAH，然后立即再写入55H，之后系统将提供3个机器周期的时间以供用户访问被保护的位。该寄存器用于保护一些会影响系统正常运行的关键寄存器，防止代码误写这些寄存器。详细说明请看第14章。

**定时器2控制**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C / $\overline{T2}$ | CP / $\overline{RL2}$ |

助记符：T2CON　　地址： C8h

TF2: 定时器2溢出标志：该位置位表示定时器2溢出。在向下计数方式中，如果计数值与捕捉寄存器的数值相等TF2也会置位。而且该位仅在RCLK和TCLK都为0的情况下被置位。该位只能由软件来清0，软件同样也可以对该位置1或清0。

EXF2: 定时器2外部事件标志：依照 CP / $\overline{RL2}$，EXEN2及DCEN的设置，在T2EX管脚（P1.1）上出现低电平跳变，或定时器2溢出时该位置位。如果是电平负跳变使该位置位，那么必须由软件来清0。如果打开相应的中断，那么当软件将该位置位或是检测到一个电平负跳变时，会引发一个定时器中断。

RCLK: 接收时钟标志：该位决定串行口0在模式1和3下接收数据时的时基。如果该位置0，那么用定时器1的溢出做波特率发生器，否则将会用定时器2 的溢出做波特率发生器。将该位置位将迫使定时器2 用作波特率发生器。

TCLK: 发送时钟标志：该位决定串行口0在模式1和3下发送数据时的时基。如果该位置0，那么用定时器1的溢出做波特率发生器，否则将会用定时器2 的溢出做波特率发生器。将该位置位将迫使定时器2 用作波特率发生器。

EXEN2: 定时器2外部事件使能。如果定时器2不用做波特率发生器时 ，该位将控制定时器2的捕捉/重装功能的开启与关闭。如果该位置0，那么T2EX管脚上的电平变化将被忽略，否则T2EX上的电平变化将会引发捕捉或重装。

TR2: 定时器2运行控制该位用于打开/关闭定时器2，该位清零时定时器2停止运行并且TH2和TL2 中的内容被保留。

C / $\overline{T2}$: 计数器/定时器选择位，该位决定定时器2是用作定时器还是计数器。如果定时器2用作波特率发生器（每个tick2个时钟），那么该位的设置对定时器2没有影响。为0则定时器2 是一个以按T2M设置的速率进行工作的定时器。为1它会对T2 脚上的负跳变进行计数。

CP / $\overline{RL2}$: 捕捉/重装选择：该位决定定时器2 是工作在捕捉模式还是重装模式。如果RCLK或TCLK置位，那么该位会被忽略定时器2会在每次溢出后自动重装。如果该位为0那么在每次定时器2溢出或是当EXEN2=1且在T2EX上检测到下降电平时，定时器2 会自动重装。如果该位为1当EXEN2=1且在在T2EX上检测到下降电平时，定时器2 会进行一次捕捉。

**定时器2模式控制**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HC5 | HC4 | HC3 | HC2 | T2CR | - | T2OE | DCEN |

助记符：T2MOD　　地址： C9h

HC5: 硬件清除 $\overline{INT5}$ 标志使能。将该位置位，当外部中断5发生且系统进入中断服务程序后，硬件将自动清除外部中断5标志。

HC4: 硬件清除INT4标志使能。将该位置位，当外部中断4发生且系统进入中断服务程序后，硬件将自动清除外部中断4标志。

HC3: 硬件清除 $\overline{\text{INT3}}$ 标志使能。将该位置位，当外部中断3发生且系统进入中断服务程序后，硬件将自动清除外部中断3标志。

HC2: 硬件清除INT2标志使能。将该位置位，当外部中断2发生且系统进入中断服务程序后，硬件将自动清除外部中断2标志。

T2CR: 定时器2捕捉复位。在定时器2 捕捉模式下该位控制当TL2和TH2的计数值传送到捕捉寄存器后，系统是否自动复位定时器2。

T2OE: 定时器2输出使能。该位用于打开/关闭定时器2时钟输出功能。

DCEN: 向下计数使能：该位与T2EX管脚相结合，控制定时器2 16位自动重装模式下的计数方向。

**定时器2 捕捉寄存器低字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP2L.7 | RCAP2L.6 | RCAP2L.5 | RCAP2L.4 | RCAP2L.3 | RCAP2L.2 | RCAP2L.1 | RCAP2L.0 |

助记符：RCAP2L　　地址： CAh

RCAP2L: 当定时器2 工作于捕捉模式时，该寄存器用于保存TL2的计数值。当定时器2工作于16位自动重装模式时，RCAP2L也用于保存16位自动重装值的低8位数值。

**定时器2 捕捉寄存器高字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP2H.7 | RCAP2H.6 | RCAP2H.5 | RCAP2H.4 | RCAP2H.3 | RCAP2H.2 | RCAP2H.1 | RCAP2H.0 |

助记符：RCAP2H　　地址： CBh

RCAP2H: T当定时器2 工作于捕捉模式时，该寄存器用于保存TH2的计数值。当定时器2工作于16位自动重装模式时，RCAP2H也用于保存16位自动重装值的高8位数值。

**定时器2 低字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL2.7 | TL2.6 | TL2.5 | TL2.4 | TL2.3 | TL2.2 | TL2.1 | TL2.0 |

助记符：TL2　　地址： CCh

TL2: Timer 2 LSB

**定时器2 高字节**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH2.7 | TH2.6 | TH2.5 | TH2.4 | TH2.3 | TH2.2 | TH2.1 | TH2.0 |

助记符：TH2　　地址： CDh

TH2: Timer 2 MSB

**程序状态字**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |

助记符：PSW　　地址： D0h

CY: 进位标志：当ALU进行算术运算产生进位或借位时置位。

AC: 辅助进位标志：高半字节运算产生进位或借位时置位。

F0: 用户标志0 ：用户可以使用的通用标志位。

RS.1-0: 寄存器区选择位：

| RS1 | RS0 | 寄存器区 | 地址 |
|---|---|---|---|
| 0 | 0 | 0 | 00-07h |
| 0 | 1 | 1 | 08-0Fh |
| 1 | 0 | 2 | 10-17h |
| 1 | 1 | 3 | 18-1Fh |

OV: 溢出标志：作为一个预先操作，当第七位而不是第八位产生进位时该标志被设置。

F1: 用户标志1： 用户可以使用的通用标志位。

P: 奇、偶标志位。由硬件控制其置位与复位。用于表示累加器中“1”的数目奇数还是偶数。

**看门狗定时器控制器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SMOD_1 | POR | - | - | WDIF | WTRF | EWT | RWT |

助记符：WDCON　　地址： D8h

SMOD_1: 将该位置位后，系统会将串行口模式 1，2，3中的波特率加倍。

POR: 上电复位标志: 在上电后硬件会将该位置1，该位可由软件读写，将该位清零的唯一方法是向该位写入一个数据。

WDIF: 看门狗定时器中断标志。如果看门狗中断使能，硬件会将该位置1 表示看门狗定时器中断产生。如果看门狗定时器中断关闭，那么该位的置位表示看门狗定时器已经超时。该位必须由软件来清零。

WTRF: 看门狗定时器复位标志。当看门狗定时器产生中断后，硬件会将该位置位。软件可以读取该位的状态，但必须由软件来将该位清除。一个掉电复位也会尽清除该位。软件可以用该位来判断复位的原因。如果EWT=0，看门狗定时器对该位不会有影响。

EWT: 看门狗定时器复位使能位。将该位置1会使能看门狗定时器复位功能。

RWT: 复位看门狗定时器；该位将看门狗定时器置为一个已知的状态；它同样可以用来在看门狗定时器溢出前将其复位；当EWDI（EIE.4）=1且没有在看门狗定时器超时前对EWT置位将会引起一个中断，当EWT=1 那么在该中断产生后经过512个时钟，将会产生看门狗定时器复位。该位被硬件清除。

外部复位产生后，WDCON的值为0x0x0xx0b。当看门狗定时器复位产生后，WTRF 被置为1 而在上电复位时该位被置位0。当外部复位产生时WTRF保持原有数值不变。上电复位时POR为被置为1，EWT在上电复位时为0，且不会受其它复位方式的影响。

对这个寄存器中数据的读取没有任何限制。但对POR, EWT, WDIF和 RWT 位的写需要进行时控访问才可以进行。对剩下位的写访问没有任何限制。请参考时控寄存器的说明。

**累加器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 |

助记符：ACC　　地址： E0h

ACC.7-0: A (或ACC)寄存器是标准8052的累加器。

**扩展中断使能**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | EWDI | EX5 | EX4 | EX3 | EX2 |

助记符：EIE　　地址： E8h

EIE.7-5: 保留位，对他们的访问将读到高电平

EWDI: 使能看门狗定时器中断

EX5: 外部中断 5 使能.

EX4: 外部中断 4 使能.

EX3: 外部中断 3 使能.

EX2: 外部中断 2 使能.

**B寄存器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |

助记符：B　　地址： F0h

B.7-0: B寄存器是标准8052中的辅助累加器

**扩展中断优先级寄存器**

| 位： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | PWDI | PX5 | PX4 | PX3 | PX2 |

助记符：EIP　　地址： F8h

EIP.7-5: 保留位

PWDI: 看门狗定时器优先级设定

PX5: 外部中断 5 优先级. 0 =低优先级, 1 =高优先级.

PX4: 外部中断 4 优先级 0 =低优先级, 1 = 高优先级

PX3: 外部中断 3优先级. 0 =低优先级, 1 = 高优先级

PX2: 外部中断 2优先级. 0 =低优先级, 1 = 高优先级.

## 9. 指令

W77E58 执行8032 体系微处理器中的所有的指令。指令的功能，对标志位及状态位的影响完全与标准8032 处理器的指令相同。但是指令的时序存在差别；主要是有2个原因，第一W77E58每4个时钟周期为一个机器周期，而标准8032每12个时钟周期为一个极其周期。另外W77E58 每个机器周期只有一个取动作，而标准8032 每个机器周期有2个取动作。

W77E58 的优势在于由于每个机器周期只有一个取动作，因此对大多数指令来说其机器周期数和它的操作数数目相同。而对于跳转和调用指令，会增加一个指令周期用以计算新的程序地址。从整体上来说，W77E58 减少了空取和等待的周期，因而提高了系统的效率。

**表2 指令对标志位的影响**

| 指令 | 进位位 | 溢出位 | 辅助进位位 | 指令 | 进位位 | 溢出位 | 辅助进位位 |
|---|---|---|---|---|---|---|---|
| ADD | X | X | X | CLR C | 0 | | |
| ADDC | X | X | X | CPL C | X | | |
| SUBB | X | X | X | ANL C, bit | X | | |
| MUL | 0 | X | | ANL C, bit | X | | |
| DIV | 0 | X | | ORL C, bit | X | | |
| DA A | X | | | ORL C, bit | X | | |
| RRC A | X | | | MOV C, bit | X | | |
| RLC A | X | | | CJNE | X | | |
| SETB C | 1 | | | | | | |

"X" 表示是每条指令执行后都会对它有影响.

**表 3. W77E58指令的时序**

| 指令 | 16进制操作数 | 字节 | W77E58 机器周期 | W77E58 时钟周期 | 8032 时钟周期 | W77E58 与. 8032速度对比 |
|---|---|---|---|---|---|---|
| NOP | 00 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R0 | 28 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R1 | 29 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R2 | 2A | 1 | 1 | 4 | 12 | 3 |
| ADD A, R3 | 2B | 1 | 1 | 4 | 12 | 3 |
| ADD A, R4 | 2C | 1 | 1 | 4 | 12 | 3 |
| ADD A, R5 | 2D | 1 | 1 | 4 | 12 | 3 |
| ADD A, R6 | 2E | 1 | 1 | 4 | 12 | 3 |
| ADD A, R7 | 2F | 1 | 1 | 4 | 12 | 3 |
| ADD A, @R0 | 26 | 1 | 1 | 4 | 12 | 3 |
| ADD A, @R1 | 27 | 1 | 1 | 4 | 12 | 3 |
| ADD A, direct | 25 | 2 | 2 | 8 | 12 | 1.5 |
| ADD A, #data | 24 | 2 | 2 | 8 | 12 | 1.5 |
| ADDC A, R0 | 38 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R1 | 39 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R2 | 3A | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R3 | 3B | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R4 | 3C | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R5 | 3D | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R6 | 3E | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R7 | 3F | 1 | 1 | 4 | 12 | 3 |

W77E58指令的时序(续)

| 指令 | 16进制操作数 | 字节 | W77E58 机器周期 | W77E58 时钟周期 | 8032 时钟周期 | W77E58 与. 8032速度对比 |
|---|---|---|---|---|---|---|
| ADDC A, @R0 | 36 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, @R1 | 37 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, direct | 35 | 2 | 2 | 8 | 12 | 1.5 |
| ADDC A, #data | 34 | 2 | 2 | 8 | 12 | 1.5 |
| ACALL addr11 | 71, 91, B1, 11, 31, 51, D1, F1 | 2 | 3 | 12 | 24 | 2 |
| AJMP ADDR11 | 01, 21, 41, 61, 81, A1, C1, E1 | 2 | 3 | 12 | 24 | 2 |
| ANL A, R0 | 58 | 1 | 1 | 4 | 12 | 3 |
| ANL A, R1 | 59 | 1 | 1 | 4 | 12 | 3 |
| ANL A, R2 | 5A | 1 | 1 | 4 | 12 | 3 |
| ANL A, R3 | 5B | 1 | 1 | 4 | 12 | 3 |
| ANL A, R4 | 5C | 1 | 1 | 4 | 12 | 3 |
| ANL A, R5 | 5D | 1 | 1 | 4 | 12 | 3 |
| ANL A, R6 | 5E | 1 | 1 | 4 | 12 | 3 |
| ANL A, R7 | 5F | 1 | 1 | 4 | 12 | 3 |
| ANL A, @R0 | 56 | 1 | 1 | 4 | 12 | 3 |
| ANL A, @R1 | 57 | 1 | 1 | 4 | 12 | 3 |
| ANL A, direct | 55 | 2 | 2 | 8 | 12 | 1.5 |
| ANL A, #data | 54 | 2 | 2 | 8 | 12 | 1.5 |
| ANL direct, A | 52 | 2 | 2 | 8 | 12 | 1.5 |
| ANL direct, #data | 53 | 3 | 3 | 12 | 24 | 2 |
| ANL C, bit | 82 | 2 | 2 | 8 | 24 | 3 |
| ANL C, /bit | B0 | 2 | 2 | 8 | 24 | 3 |
| CJNE A, direct, rel | B5 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE A, #data, rel | B4 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE @R0, #data, rel | B6 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE @R1, #data, rel | B7 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R0, #data, rel | B8 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R1, #data, rel | B9 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R2, #data, rel | BA | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R3, #data, rel | BB | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R4, #data, rel | BC | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R5, #data, rel | BD | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R6, #data, rel | BE | 3 | 4 | 16 | 24 | 1.5 |

| 指令 | 16进制操作数 | 字节 | W77E58<br>机器周期 | W77E58<br>时钟周期 | 8032<br>时钟周期 | W77E58 与.<br>8032速度对比 |
|---|---|---|---|---|---|---|
| CLR A | E4 | 1 | 1 | 4 | 12 | 3 |
| CPL A | F4 | 1 | 1 | 4 | 12 | 3 |
| CLR C | C3 | 1 | 1 | 4 | 12 | 3 |
| CLR bit | C2 | 2 | 2 | 8 | 12 | 1.5 |
| CPL C | B3 | 1 | 1 | 4 | 12 | 3 |
| CPL bit | B2 | 2 | 2 | 8 | 12 | 1.5 |
| DEC A | 14 | 1 | 1 | 4 | 12 | 3 |
| DEC R0 | 18 | 1 | 1 | 4 | 12 | 3 |
| DEC R1 | 19 | 1 | 1 | 4 | 12 | 3 |
| DEC R2 | 1A | 1 | 1 | 4 | 12 | 3 |
| DEC R3 | 1B | 1 | 1 | 4 | 12 | 3 |
| DEC R4 | 1C | 1 | 1 | 4 | 12 | 3 |
| DEC R5 | 1D | 1 | 1 | 4 | 12 | 3 |
| DEC R6 | 1E | 1 | 1 | 4 | 12 | 3 |
| DEC R7 | 1F | 1 | 1 | 4 | 12 | 3 |
| DEC @R0 | 16 | 1 | 1 | 4 | 12 | 3 |
| DEC @R1 | 17 | 1 | 1 | 4 | 12 | 3 |
| DEC direct | 15 | 2 | 2 | 8 | 12 | 1.5 |
| DEC DPTR | A5 | 1 | 2 | 8 | - | - |
| DIV AB | 84 | 1 | 5 | 20 | 48 | 2.4 |
| DA A | D4 | 1 | 1 | 4 | 12 | 3 |
| DJNZ R0, rel | D8 | 2 | 3 | 12 | 24 | 2 |
| DJNZ R1, rel | D9 | 2 | 3 | 12 | 24 | 2 |
| DJNZ R5, rel | DD | 2 | 3 | 12 | 24 | 2 |
| DJNZ R2, rel | DA | 2 | 3 | 12 | 24 | 2 |
| DJNZ R3, rel | DB | 2 | 3 | 12 | 24 | 2 |
| DJNZ R4, rel | DC | 2 | 3 | 12 | 24 | 2 |
| DJNZ R6, rel | DE | 2 | 3 | 12 | 24 | 2 |
| DJNZ R7, rel | DF | 2 | 3 | 12 | 24 | 2 |
| DJNZ direct, rel | D5 | 3 | 4 | 16 | 24 | 1.5 |
| INC A | 04 | 1 | 1 | 4 | 12 | 3 |
| INC R0 | 08 | 1 | 1 | 4 | 12 | 3 |
| INC R1 | 09 | 1 | 1 | 4 | 12 | 3 |
| INC R2 | 0A | 1 | 1 | 4 | 12 | 3 |
| INC R3 | 0B | 1 | 1 | 4 | 12 | 3 |
| INC R4 | 0C | 1 | 1 | 4 | 12 | 3 |

| 指令 | 16进制操作数 | 字节 | W77E58<br>机器周期 | W77E58<br>时钟周期 | 8032<br>时钟周期 | W77E58 与.<br>8032速度对比 |
|---|---|---|---|---|---|---|
| INC R6 | 0E | 1 | 1 | 4 | 12 | 3 |
| INC R7 | 0F | 1 | 1 | 4 | 12 | 3 |
| INC @R0 | 06 | 1 | 1 | 4 | 12 | 3 |
| INC @R1 | 07 | 1 | 1 | 4 | 12 | 3 |
| INC direct | 05 | 2 | 2 | 8 | 12 | 1.5 |
| INC DPTR | A3 | 1 | 2 | 8 | 24 | 3 |
| JMP @A+DPTR | 73 | 1 | 2 | 8 | 24 | 3 |
| JZ rel | 60 | 2 | 3 | 12 | 24 | 2 |
| JNZ rel | 70 | 2 | 3 | 12 | 24 | 2 |
| JC rel | 40 | 2 | 3 | 12 | 24 | 2 |
| JNC rel | 50 | 2 | 3 | 12 | 24 | 2 |
| JB bit, rel | 20 | 3 | 4 | 16 | 24 | 1.5 |
| JNB bit, rel | 30 | 3 | 4 | 16 | 24 | 1.5 |
| JBC bit, rel | 10 | 3 | 4 | 16 | 24 | 1.5 |
| LCALL addr16 | 12 | 3 | 4 | 16 | 24 | 1.5 |
| LJMP addr16 | 02 | 3 | 4 | 16 | 24 | 1.5 |
| MUL AB | A4 | 1 | 5 | 20 | 48 | 2.4 |
| MOV A, R0 | E8 | 1 | 1 | 4 | 12 | 3 |
| MOV A, R1 | E9 | 1 | 1 | 4 | 12 | 3 |
| MOV A, R2 | EA | 1 | 1 | 4 | 12 | 3 |
| MOV A, R3 | EB | 1 | 1 | 4 | 12 | 3 |
| MOV A, R4 | EC | 1 | 1 | 4 | 12 | 3 |
| MOV A, R5 | ED | 1 | 1 | 4 | 12 | 3 |
| MOV A, R6 | EE | 1 | 1 | 4 | 12 | 3 |
| MOV A, R7 | EF | 1 | 1 | 4 | 12 | 3 |
| MOV A, @R0 | E6 | 1 | 1 | 4 | 12 | 3 |
| MOV A, @R1 | E7 | 1 | 1 | 4 | 12 | 3 |
| MOV A, direct | E5 | 2 | 2 | 8 | 12 | 1.5 |
| MOV A, #data | 74 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R0, A | F8 | 1 | 1 | 4 | 12 | 3 |
| MOV R1, A | F9 | 1 | 1 | 4 | 12 | 3 |
| MOV R2, A | FA | 1 | 1 | 4 | 12 | 3 |
| MOV R3, A | FB | 1 | 1 | 4 | 12 | 3 |
| MOV R4, A | FC | 1 | 1 | 4 | 12 | 3 |
| MOV R5, A | FD | 1 | 1 | 4 | 12 | 3 |
| MOV R6, A | FE | 1 | 1 | 4 | 12 | 3 |
| MOV R7, A | FF | 1 | 1 | 4 | 12 | 3 |

| 指令 | 16进制操作数 | 字节 | W77E58<br>机器周期 | W77E58<br>时钟周期 | 8032<br>时钟周期 | W77E58 与.<br>8032速度对比 |
|---|---|---|---|---|---|---|
| MOV R1, direct | A9 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R2, direct | AA | 2 | 2 | 8 | 12 | 1.5 |
| MOV R3, direct | AB | 2 | 2 | 8 | 12 | 1.5 |
| MOV R4, direct | AC | 2 | 2 | 8 | 12 | 1.5 |
| MOV R5, direct | AD | 2 | 2 | 8 | 12 | 1.5 |
| MOV R6, direct | AE | 2 | 2 | 8 | 12 | 1.5 |
| MOV R7, direct | AF | 2 | 2 | 8 | 12 | 1.5 |
| MOV R0, #data | 78 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R1, #data | 79 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R2, #data | 7A | 2 | 2 | 8 | 12 | 1.5 |
| MOV R3, #data | 7B | 2 | 2 | 8 | 12 | 1.5 |
| MOV R4, #data | 7C | 2 | 2 | 8 | 12 | 1.5 |
| MOV R5, #data | 7D | 2 | 2 | 8 | 12 | 1.5 |
| MOV R6, #data | 7E | 2 | 2 | 8 | 12 | 1.5 |
| MOV R7, #data | 7F | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R0, A | F6 | 1 | 1 | 4 | 12 | 3 |
| MOV @R1, A | F7 | 1 | 1 | 4 | 12 | 3 |
| MOV @R0, direct | A6 | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R1, direct | A7 | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R0, #data | 76 | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R1, #data | 77 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, A | F5 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R0 | 88 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R1 | 89 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R2 | 8A | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R3 | 8B | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R4 | 8C | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R5 | 8D | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R6 | 8E | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R7 | 8F | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, @R0 | 86 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, @R1 | 87 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, direct | 85 | 3 | 3 | 12 | 24 | 2 |
| MOV direct, #data | 75 | 3 | 3 | 12 | 24 | 2 |
| MOV DPTR, #data 16 | 90 | 3 | 3 | 12 | 24 | 2 |
| MOVC A, @A+DPTR | 93 | 1 | 2 | 8 | 24 | 3 |

| 指令 | 16进制操作数 | 字节 | W77E58 机器周期 | W77E58 时钟周期 | 8032 时钟周期 | W77E58 与. 8032速度对比 |
|---|---|---|---|---|---|---|
| MOVX A, @R0 | E2 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX A, @R1 | E3 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX A, @DPTR | E0 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @R0, A | F2 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @R1, A | F3 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @DPTR, A | F0 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOV C, bit | A2 | 2 | 2 | 8 | 12 | 1.5 |
| MOV bit, C | 92 | 2 | 2 | 8 | 24 | 3 |
| ORL A, R0 | 48 | 1 | 1 | 4 | 12 | 3 |
| ORL A, R1 | 49 | 1 | 1 | 4 | 12 | 3 |
| ORL A, R2 | 4A | 1 | 1 | 4 | 12 | 3 |
| ORL A, R3 | 4B | 1 | 1 | 4 | 12 | 3 |
| ORL A, R4 | 4C | 1 | 1 | 4 | 12 | 3 |
| ORL A, R5 | 4D | 1 | 1 | 4 | 12 | 3 |
| ORL A, R6 | 4E | 1 | 1 | 4 | 12 | 3 |
| ORL A, R7 | 4F | 1 | 1 | 4 | 12 | 3 |
| ORL A, @R0 | 46 | 1 | 1 | 4 | 12 | 3 |
| ORL A, @R1 | 47 | 1 | 1 | 4 | 12 | 3 |
| ORL A, direct | 45 | 2 | 2 | 8 | 12 | 1.5 |
| ORL A, #data | 44 | 2 | 2 | 8 | 12 | 1.5 |
| ORL direct, A | 42 | 2 | 2 | 8 | 12 | 1.5 |
| ORL direct, #data | 43 | 3 | 3 | 12 | 24 | 2 |
| ORL C, bit | 72 | 2 | 2 | 8 | 24 | 3 |
| ORL C, /bit | A0 | 2 | 2 | 6 | 24 | 3 |
| PUSH direct | C0 | 2 | 2 | 8 | 24 | 3 |
| POP direct | D0 | 2 | 2 | 8 | 24 | 3 |
| RET | 22 | 1 | 2 | 8 | 24 | 3 |
| RETI | 32 | 1 | 2 | 8 | 24 | 3 |
| RL A | 23 | 1 | 1 | 4 | 12 | 3 |
| RLC A | 33 | 1 | 1 | 4 | 12 | 3 |
| RR A | 03 | 1 | 1 | 4 | 12 | 3 |
| RRC A | 13 | 1 | 1 | 4 | 12 | 3 |
| SETB C | D3 | 1 | 1 | 4 | 12 | 3 |
| SETB bit | D2 | 2 | 2 | 8 | 12 | 1.5 |
| SWAP A | C4 | 1 | 1 | 4 | 12 | 3 |
| SJMP rel | 80 | 2 | 3 | 12 | 24 | 2 |
| SUBB A, R0 | 98 | 1 | 1 | 4 | 12 | 3 |

| 指令 | 16进制操作数 | 字节 | W77E58<br>机器周期 | W77E58<br>时钟周期 | 8032<br>时钟周期 | W77E58 与.<br>8032速度对比 |
|---|---|---|---|---|---|---|
| SUBB A, R2 | 9A | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R3 | 9B | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R4 | 9C | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R5 | 9D | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R6 | 9E | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R7 | 9F | 1 | 1 | 4 | 12 | 3 |
| SUBB A, @R0 | 96 | 1 | 1 | 4 | 12 | 3 |
| SUBB A, @R1 | 97 | 1 | 1 | 4 | 12 | 3 |
| SUBB A, direct | 95 | 2 | 2 | 8 | 12 | 1.5 |
| SUBB A, #data | 94 | 2 | 2 | 8 | 12 | 1.5 |
| XCH A, R0 | C8 | 1 | 1 | 4 | 12 | 3 |
| XCH A, R1 | C9 | 1 | 1 | 4 | 12 | 3 |
| XCH A, R2 | CA | 1 | 1 | 4 | 12 | 3 |
| XCH A, R3 | CB | 1 | 1 | 4 | 12 | 3 |
| XCH A, R4 | CC | 1 | 1 | 4 | 12 | 3 |
| XCH A, R5 | CD | 1 | 1 | 4 | 12 | 3 |
| XCH A, R6 | CE | 1 | 1 | 4 | 12 | 3 |
| XCH A, R7 | CF | 1 | 1 | 4 | 12 | 3 |
| XCH A, @R0 | C6 | 1 | 1 | 4 | 12 | 3 |
| XCH A, @R1 | C7 | 1 | 1 | 4 | 12 | 3 |
| XCHD A, @R0 | D6 | 1 | 1 | 4 | 12 | 3 |
| XCHD A, @R1 | D7 | 1 | 1 | 4 | 12 | 3 |
| XCH A, direct | C5 | 2 | 2 | 8 | 12 | 1.5 |
| XRL A, R0 | 68 | 1 | 1 | 4 | 12 | 3 |
| XRL A, R1 | 69 | 1 | 1 | 4 | 12 | 3 |
| XRL A, R2 | 6A | 1 | 1 | 4 | 12 | 3 |
| XRL A, R3 | 6B | 1 | 1 | 4 | 12 | 3 |
| XRL A, R4 | 6C | 1 | 1 | 4 | 12 | 3 |
| XRL A, R5 | 6D | 1 | 1 | 4 | 12 | 3 |
| XRL A, R6 | 6E | 1 | 1 | 4 | 12 | 3 |
| XRL A, R7 | 6F | 1 | 1 | 4 | 12 | 3 |
| XRL A, @R0 | 66 | 1 | 1 | 4 | 12 | 3 |
| XRL A, @R1 | 67 | 1 | 1 | 4 | 12 | 3 |
| XRL A, direct | 65 | 2 | 2 | 8 | 12 | 1.5 |
| XRL A, #data | 64 | 2 | 2 | 8 | 12 | 1.5 |
| XRL direct, A | 62 | 2 | 2 | 8 | 12 | 1.5 |
| XRL direct, #data | 63 | 3 | 3 | 12 | 24 | 2 |

## 9.1 指令时序

指令时许对W77E58来说是一个很重要的特性，对于用软件的方式来产生定时的用户更为重要。它也向用户说明W77E58与标准8032在时序上的差别。在W77E58中每个机器周期是4个时钟周期，每个时钟周期都是一个确定的状态。因此一个机器周期由4个确定的状态C1、C2、C3、C4组成。由于每条指令的执行速度都加快，所以时钟的2个跳变边沿都用于内部时序。因此时钟的占空比接近于50%，以避免时间上发生冲突。前面已经说到W77E58每一个机器周期进行一次代码读取操作，因此对大多数指令来说，执行指令的机器周期与操作码中的字节数相同。系统总共有256个操作码，其中有128个是单周期指令。因此在W77E58中有一半的指令会在4个时钟周期内执行完毕。对多数双字节指令来说，指令的执行周期是2个机器周期。但也有指令为一个字节但周期是2个时钟周期的情况；一个需要特别注意的指令是MOVX指令，在标准8032中他的指令周期固定为2个机器周期. 但在W77E58中他的指令周期可变为2-9个机器周期。$\overline{RD}$ 和 $\overline{WR}$ 信号也有相应的变化。这为用户访问快速或慢速设备就带来了方便，不需使用额外的外围电路，也减少了软件负担。剩下的指令的机器周期数目可以是3个，4个，5个。注意在W77E58中基于指令字节数目的不同，共有5种类型的指令，而标准8032中只有3种指令类型。但是W77E58中每4个时钟周期为一个机器周期，而不是标准8032中每12个时钟周期为一个机器周期。因此尽管指令种类增多， W77E58中的指令执行速度要比标准8032快1.5-3倍。（以时钟周期计算）

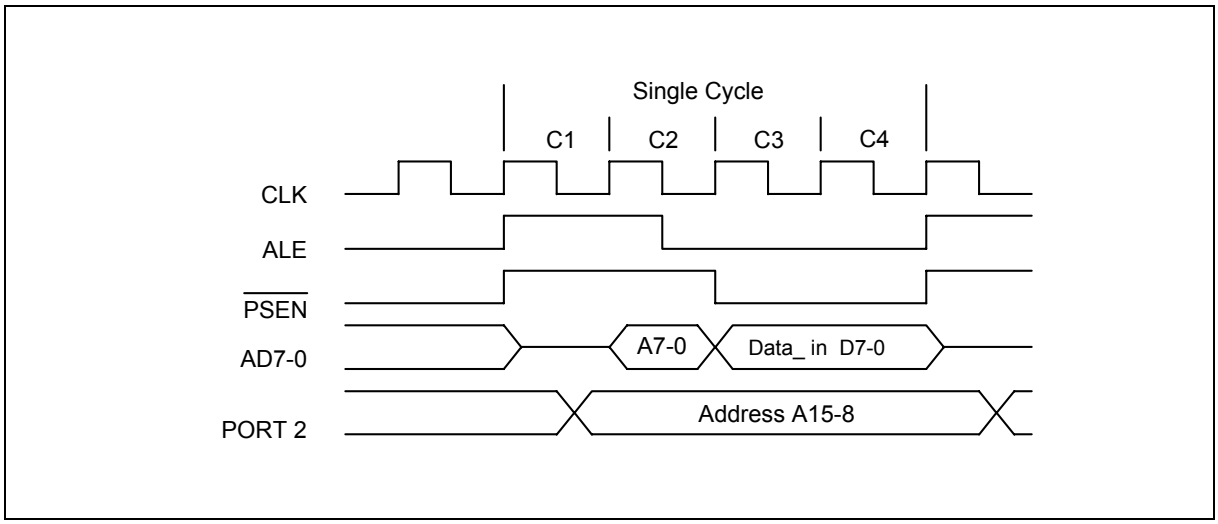



图3：单周期指令时序

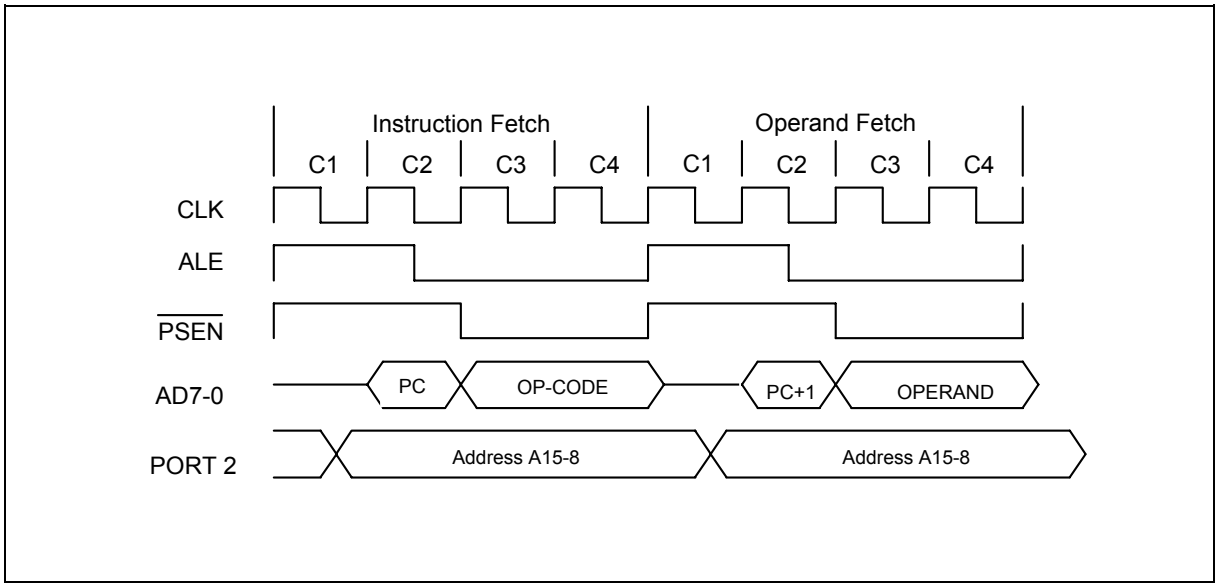



图4：双周期指令时序

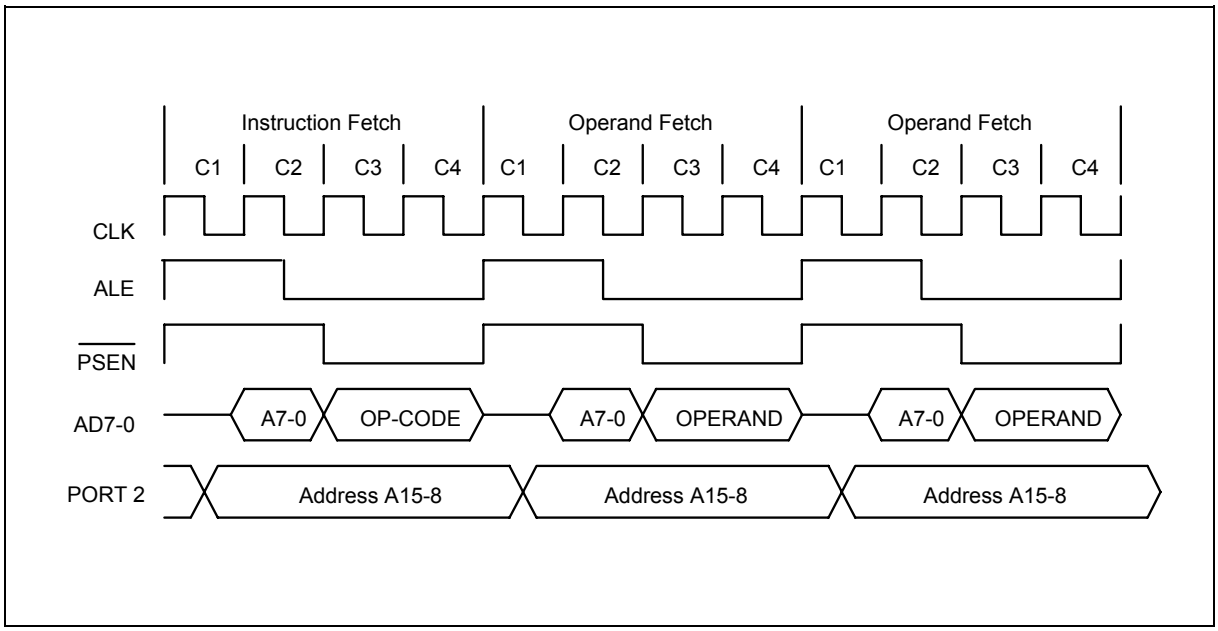



图5：3周期指令时序

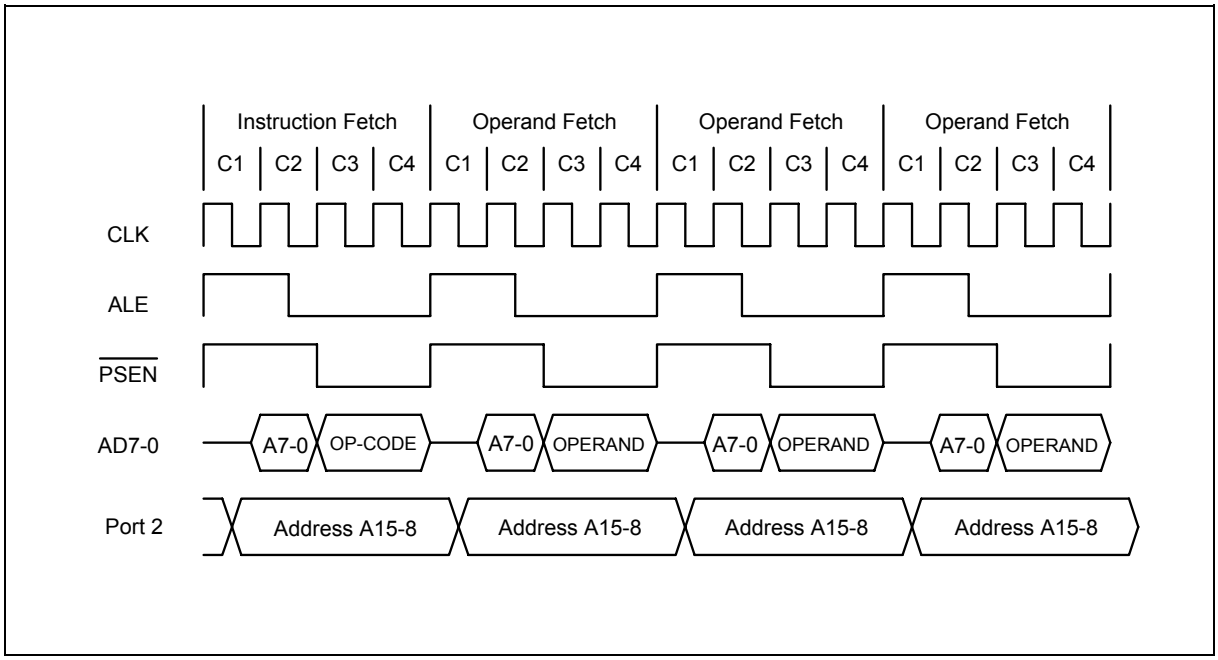



图6：四周期指令时序

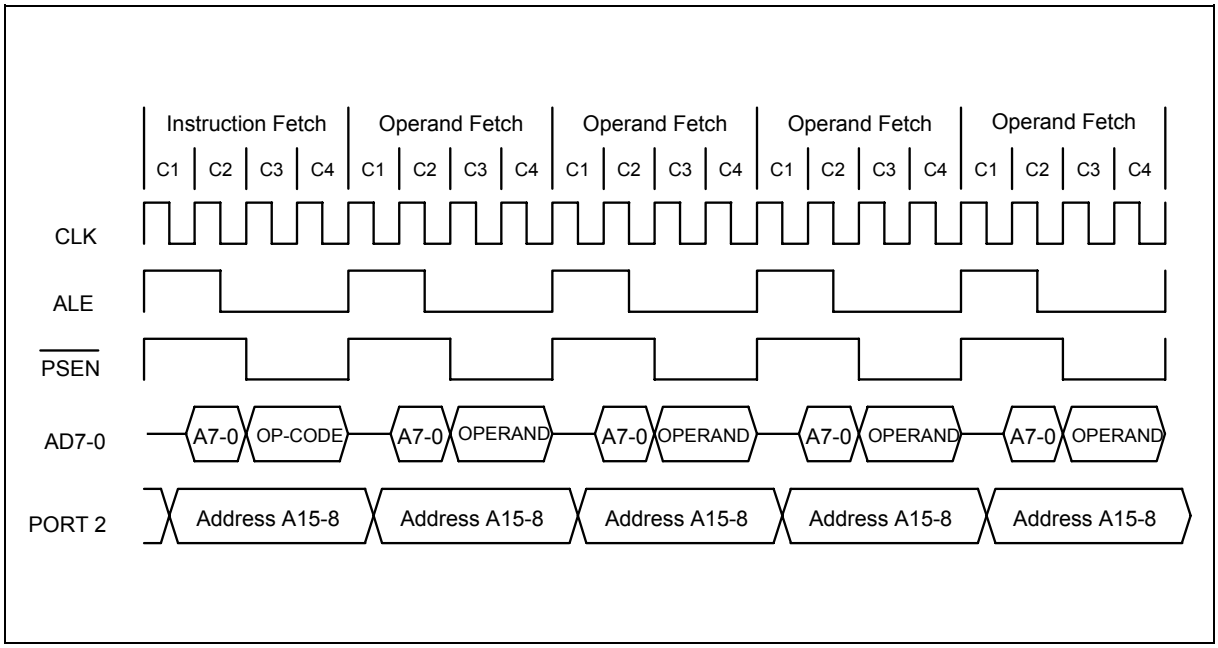



图7：5周期指令时序

## 9.2 MOVX 指令

同标准8032一样，W77E58使用MOVX指令来访问外部数据存储器。外部数据存储器的地址空间还包含了存储器映射的外设。MOVX指令执行的结果同标准8032是一致的，但选通信号的时序和运行方式不同，这样的修改给了用户极大的灵活性。

MOVX指令有2种类型。MOVX @Ri 和MOVX @DPTR。MOVX @Ri指令中外部地址由2部分组成，外部地址的低8位由选定寄存器区中的Ri提供，高8位由P2口提供。在MOVX @DPTR指令中16位地址全由DPTR指针提供。

由于 W77E58 有2个数据指针DPTR和DPTR1，用户可以通过设置/清除DPS位来选择DPTR或DPTR1。DPS位是DPS特殊功能寄存器（地址为86H）中的最低位。DPS=0，选用DPTR，DPS=1选用DPTR1。用户可以用该位在DPTR和DPTR1中切换。INC指令是实现这种切换方式的最快的方法。有2个数据指针的优势在于进行数据块搬移的操作中，下面的代码说明双数据指针如何提高代码的效率。

只使用一个数据指针的数据块搬移:

; SH和SL是源地址的高低字节
; DH和DL是目标地址的高低字节
; CNT记录已经搬移了的字节数

| | | | W77E58的机器周期 # |
|---|---|---|---|
| MOV | R2, #CNT | ; 为R2装载新的计数值 | 2 |
| MOV | R3, #SL | ; 在R3中保存源地址低字节 | 2 |
| MOV | R4, #SH | ; 在R4中保存源地址的高字节 | 2 |
| MOV | R5, #DL | ; 在R5中保存目标地址的低字节 | 2 |
| MOV | R6, #DH | ; 在R6中保存目标地址的高字节 | 2 |
| LOOP: | | | |
| MOV | DPL, R3 | ; 在DPL中保存源地址低字节 | 2 |
| MOV | DPH, R4 | ; 在DPH中保存源地址高字节 | 2 |
| MOVX | A, @DPTR | ; 从源地址向累加器中保存数据 | 2 |
| INC | DPTR | ; 增加源地址 | 2 |
| MOV | R3, DPL | ; 将源地址低字节存入R3 | 2 |
| MOV | R4, DPH | ; 将源地址高字节存入R4 | 2 |
| MOV | DPL, R5 | ; 目标地址低字节存入DPL | 2 |
| MOV | DPH, R6 | ; 目标地址高字节存入DPH | 2 |
| MOVX | @DPTR, A | ; 向目标写入数据 | 2 |
| INC | DPTR | ; 增加目标地址 | 2 |
| MOV | DPL, R5 | ; 在R5中保存新目标地址的低字节 | 2 |
| MOV | DPH, R6 | ; 在R6中保存新目标地址的高字节 | 2 |
| DJNZ | R2, LOOP | ; 将计数器减一，如果不为0再执行次 | 2 |

标准8032机器周期 = 10 + (26 * CNT)
W77E58中机器周期 = 10 + (26 * CNT)
如果 CNT = 50
标准8032中的时钟数= ((10 + (26 *50)) * 12 = (10 + 1300) * 12 = 15720

W77E58中的时钟数= ((10 + (26 * 50)) * 4 = (10 + 1300) * 4 = 5240

用双数据指针来移动数据块：

```
; SH和SL是源地址的高低字节
; DH和DL是目标地址的高低字节
; CNT记录已经搬移了的字节数
```

| | | | W77E58的机器周期 # |
|---|---|---|---|
| MOV | R2, #CNT | ; R2中装载计数值 | 2 |
| MOV | DPS, #00h | ; 选定 DPTR | 2 |
| MOV | DPTR, #DHDL | ; 在DPTR 中装入目标地址 | 3 |
| INC | DPS | ; 选定DPTR1 | 2 |
| MOV | DPTR, #SHSL | ; 在DPTR1中装入源地址 | 3 |
| LOOP: | | | |
| MOVX | A, @DPTR | ; 从源地址处拿出数据 | 2 |
| INC | DPTR | ; 源地址加一 | 2 |
| DEC | DPS | ; 选定DPTR | 2 |
| MOVX | @DPTR, A | ; 向目标写入数据 | 2 |
| INC | DPTR | ; 增加源地址 | 2 |
| INC | DPS | ; 选定DPTR1 | 2 |
| DJNZ | R2, LOOP | ; 检查是否已经全部移完 | 3 |

W77E58中机器周期 = 12 + (15 * CNT)

如果CNT = 50

W77E58中的时钟数 = (12 + (15 * 50)) * 4 = (12 + 750) * 4 = 3048

可以看出在第一个程序中，标准8032用15720个时钟完成任务，W77E58仅用5240个时钟就完成任务。在第二个程序中时钟数又减少到3048个，如果数据块增大那么系统节省的时间将更多。

## 9.3 外部数据存储器访问时序

MOVX 指令的时序是W77E58另一大特性。在标准的8032中，MOVX指令的周期固定，为2个机器周期。但在W77E58中MOVX指令的周期可以按照用户的需求改变。

指令以正常的4个时钟周期开始，在下一个时钟周期，W77E58输出要访问的外部数据存储器的地址，此刻才进行真正的访问。用户可以通过设置STRETCH的数值来改变这个周期时间的长短。用CKCON寄存器中的3个位来设置STRETCH的值。这3位是M2-0（CKCON中的2-0位），这3位给出8种不同的访问时间选项。STRETCH的取值范围为0-7，这样MOVX指令周期的变化范围就是2-9个机器周期。 注意这样的设置仅对MOVX指令有效；默认状态下STRETCH值为1，MOVX的指令周期为3个时钟周期。如果需要用户可以将STRETCH值设为0，使MOVX的指令周期为2个时钟周期，以获得最快的访问速度。

表4：数据存储器STRETCH值

| M2 | M1 | M0 | 机器周期 | $\overline{RD}$ 或 $\overline{WR}$ 信号时钟数 | 25 MHz 下 $\overline{RD}$ 或 $\overline{WR}$ 信号宽度 | 40 MHz 下 $\overline{RD}$ 或 $\overline{WR}$ 信号宽度 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 2 | 2 | 80 nS | 50 nS |
| 0 | 0 | 1 | 3 (default) | 4 | 160 nS | 100 nS |
| 0 | 1 | 0 | 4 | 8 | 320 nS | 200 nS |
| 0 | 1 | 1 | 5 | 12 | 480 nS | 300 nS |
| 1 | 0 | 0 | 6 | 16 | 640 nS | 400 nS |
| 1 | 0 | 1 | 7 | 20 | 800 nS | 500 nS |
| 1 | 1 | 0 | 8 | 24 | 960 nS | 600 nS |
| 1 | 1 | 1 | 9 | 28 | 1120 nS | 700 nS |

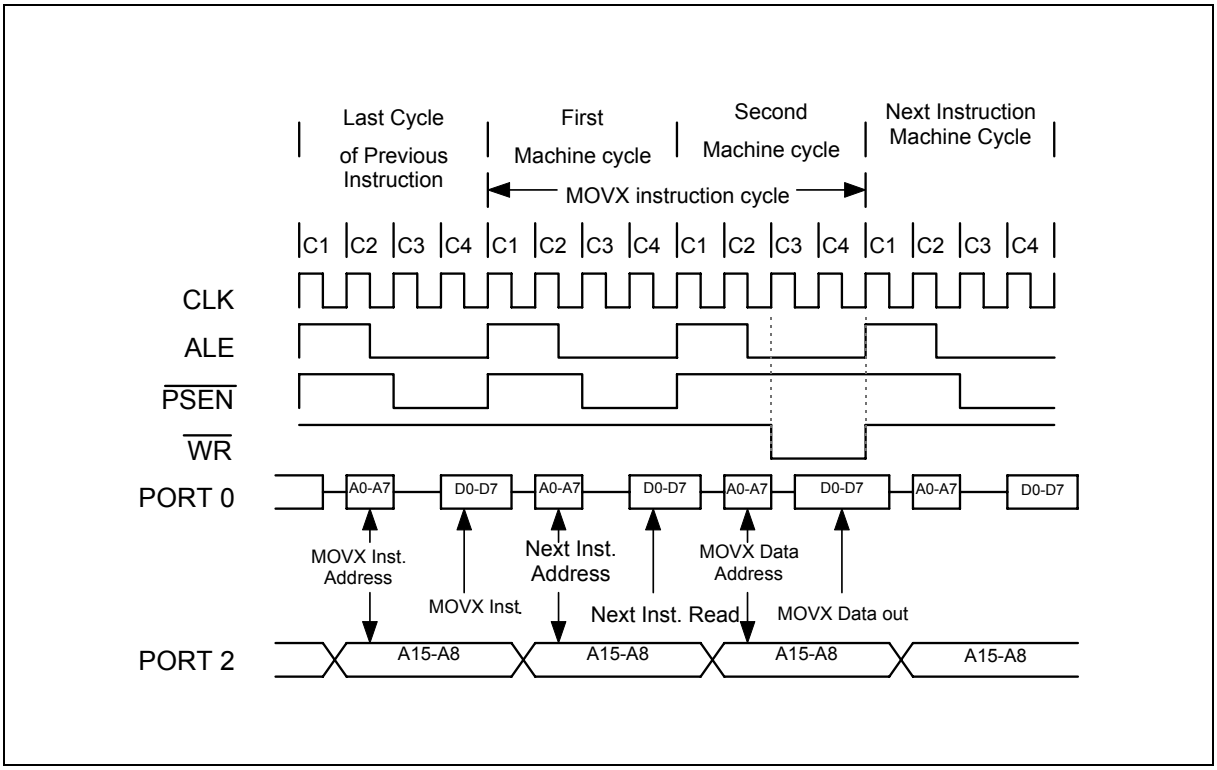



图8: STRETCH=0时的外部数据存储器访问时序

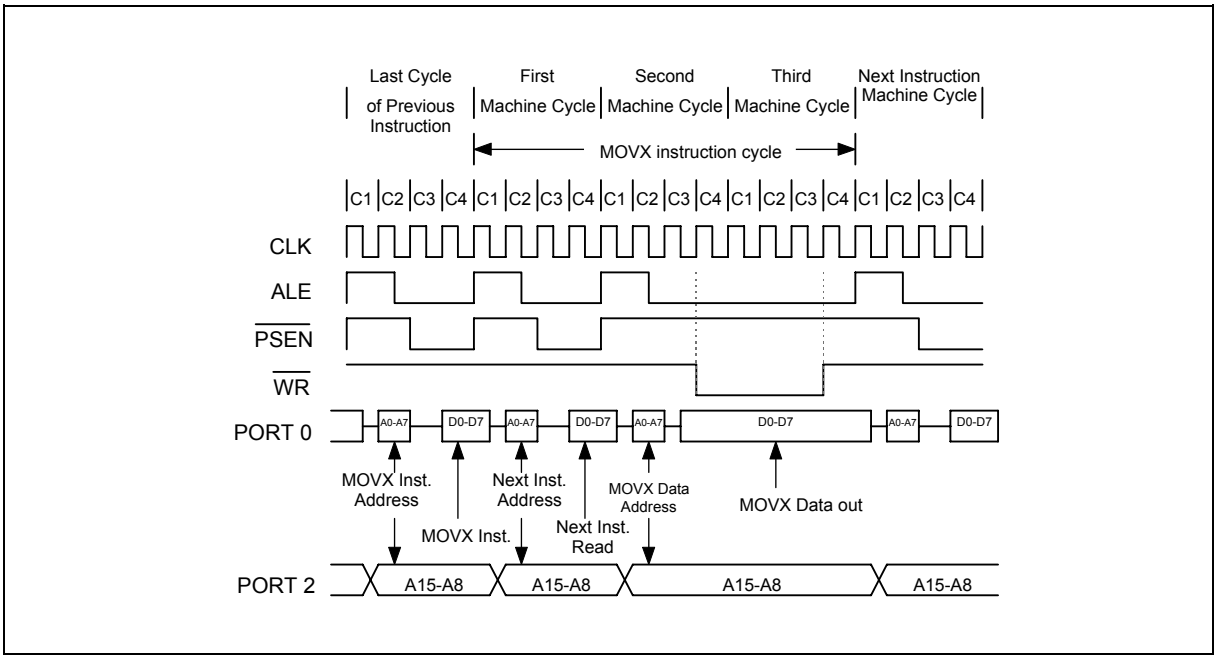



图9：STRETCH=1时的外部数据存储器访问时序

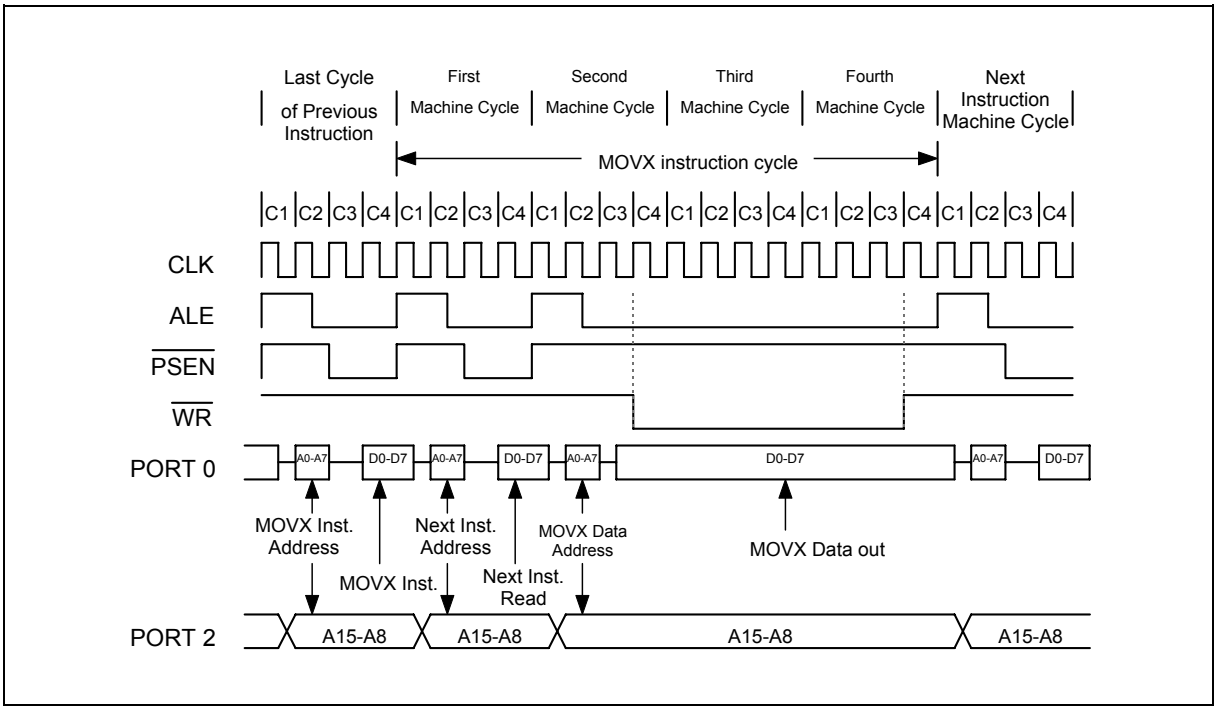



图10：STRETCH=2时的外部数据存储器访问时序

## 9.4 等待状态控制信号

无论软件是否使用stretch值来改变MOVX指令的周期，W77E58还提供一种 $\overline{\text{WAIT}}$ 硬件信号，来实现对外部数据访问时序的加宽。等待状态控制信号是P4.0脚的一个复用功能，仅在PLCC/QFP44脚封装的芯片中有此功能。通过设置WS（WSCON.7）位可以打开这个功能。在打开该功能后，stretch值的设定决定MOVX指令的最小机器周期。但这时系统会在C3态对 $\overline{\text{WAIT}}$ 信号进行采样，如果采样到该信号，那么系统会在下个机器周期内多插入一个机器周期（等待周期）。插入的等待周期是没有限制的，因此MOVX指令将在等待信号结束后才会结束。使用等待信号，就实现了对外部设备访问的动态时序。WS位是受时控保护的。

WS=1且 stretch = 0 会使能等待信号。

# 10. 电源管理

W77E58有若干节电选项来帮助用户减少电源消耗。W77E58的节电模式为掉电模式、经济模式以及空闲模式。

## 10.1 空闲模式

用户通过将1 写入PCON.0，使系统进入空闲模式。把系统放入空闲模式的指令是系统在进入空闲模式前执行的最后一条指令。在空闲模式下，提供给CPU的时钟被切断，但是中断、定时器、串行口的时钟照常工作。这样CPU就进入冻结状态；程序计数器、堆栈指针、程序状态字、累加器及其他一些寄存器的内容保持不变。ALE和 $\overline{\text{PSEN}}$ 在空闲模式下处于高电平状态。各个端口维持进入空闲模式前的逻辑状态。有2种方式可以让系统从空闲模式中退出。由于中断控制器依旧在工作，因此任何使能的中断都可以让系统退出空闲模式。当这样的中断发生时，系统将自动清除空闲位，退出空闲模式并转向相应的中断服务程序。在中断服务程序完成后，系统将在使系统进入空闲模式的那条指令之后继续程序的运行。

复位同样可以使系统退出空闲模式。实现复位的方式有在RST脚上输入高电平，上电复位以及看门狗定时器复位。外部复位时，高电平至少要维持2个机器周期（8个时钟周期），以便系统识别外部复位信号。复位后程序指针数值为0000H，所有SFR都回到初始状态。由于时钟并没有停止工作因此程序会被立即执行。在空闲模式下，看门狗定时器依旧工作，因此如果看门狗定时器中断打开，看门狗定时器溢出后会产生中断使系统退出空闲模式。软件必须复位看门狗定时器，以便在看门狗定时器溢出并经过512个时钟周期后将系统复位。当W77E58以复位的方式从空闲模式中退出后，系统将从头开始执行指令。

## 10.2 经济模式

微控制器的电源消耗与它的运行频率有关。W77E58提供一种经济模式来动态的减少内部时钟而无须外部器件。默认状态下一个机器周期是4个时钟周期。在经济模式下，一个机器周期可以是4，64或1024个时钟周期。它使系统工作在一个可以接受的速度范围内，并且减少了电源消耗。在空闲模式下，CPU核的时钟被停止，但其它外设的时钟如看门狗定时器的时钟依旧是4个时钟周期。在经济模式下，所有外设的时钟的速度与核的时钟速度相同。因此，经济模式下的空闲模式更加省电。

| CD1 | CD0 | 时钟数/机器周期 |
|---|---|---|
| 0 | 0 | 保留 |
| 0 | 1 | 4(默认) |
| 1 | 0 | 64 |
| 1 | 1 | 1024 |

这些设置在延迟了一个指令周期后有效。在64分频和1024间分频，需要先回到4分频模式。这就是说软件不能直接在64分频和1024分频间切换。CPU必须先回到4分频模式，然后再进入64 分频或1024分频模式。

在经济模式下，由于波特率改变了串行口就不能正确发送/接收了。在一些系统中，一些外部中断需要最快速的处理。为了解决这些问题，W77E58提供了一种机制，在该机制下，当有串行口操作，及外部中断产生时，CPU会切换回4分频模式。切换功能由SWB（PMR.5）位来控制，SWB=1则打开切换模式。当串行口进行接收/发送操作或有外部中断产生，且该中断未被阻止时，系统会进行一次切换以保证操作的正确性。对串行口来说，进行接收时在起始位的下降沿会触发一次切换。在发送时，对SBUF的写操作会引发一次切换以保证发送的正确性。切换功能不受串行口中断标志的影响。再一个切换产生后，可以手动用软件把CPU切换回经济模式。当打开切换功能，且在进行串行接收/发送时，对CD0和CD1的改变会被忽略。看门狗定时器复位，上电/掉电复位，或外部复位都会使CPU退出切换模式。

## 10.3 掉电模式

用户通过将1 写入PCON.1，使系统进入掉电模式。把系统放入掉电模式的指令是系统在进入掉电模式前最后执行的一条指令。在掉电模式下，系统所有的时钟都停止工作设备进入停止状态。系统所有的工作都停止，这样电源的消耗就降至最低。在这种情况下，ALE及$\overline{\text{PSEN}}$ 管脚输出低电平。端口上输出其相应SFR寄存器内的值。

复位以及电平跳变出发的中断可以使系统退出掉电模式。外部复位可让系统退出中断，RST脚上的高电平将终止掉电模式，然后重新开启时钟。程序将从0000H处开始执行，由于在掉电模式中时钟停止工作，因此看门狗定时器不能提供复位功能让系统退出掉电模式。

如果EA=1，外部中断被设置为电平触发方式而且相应的外部中断开放，那么外部中断输入脚上的低电平将迫使系统退出掉电模式。如果上面所述的条件满足，当外部中断输入脚上有低电平信号时，该信号将重新启动时钟。设备转向相应的中断服务程序，在ISR服务完成后，系统将从使系统进入掉电模式的那条指令之后继续程序的运行。

表5：空闲和掉电模式下外部脚的状态

| 模式 | 程序存储器 | ALE | $\overline{\text{PSEN}}$ | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| 空闲 | 内部 | 1 | 1 | 数据 | 数据 | 数据 | 数据 |
| 空闲 | 外部I | 1 | 1 | 浮空 | 数据 | 地址 | 数据 |
| 掉电 | 内部 | 0 | 0 | 数据 | 数据 | 数据 | 数据 |
| 掉电 | 外部 | 0 | 0 | 浮空 | 数据 | 数据 | 数据 |

## 11. 复位条件

用户有很多与硬件相关的选项来将W77E58复位。一般来说许多寄存器在复位后都将回到其初始值，而不管复位的类型如何。但有些标志位的状态取决于复位的类型。用户可以根据这些标志位来判断复位的类型。有2种方法可以将系统复位：1.外部复位信号；2.看门狗定时器复位。

### 11.1 外部复位

系统在每个机器周期的C4态对RST管脚进行连续的采样。因此RST管脚上的电平至少要维持2个机器周期，以保证系统检测到有效的RST高电平。然后复位电路将同步发出复位信号，因此复位是一个同步的动作，要求时钟在此期间一直运行来实现外部复位。

系统进入复位状态以后，只要RST脚上电平一直为高，那么系统就一直处于复位状态中。在RST信号撤除后，系统仍将会在2个机器周期内保持复位状态，然后才从0000H处开始执行程序。对外部复位来说，没有与之配套的标志位。但是由于另外的2种复位模式都有相应的标志位存在，那么当其他2个标志位为零时，可以将外部复位认为是默认的复位情况。

软件在读取POR位以后必须将其清除，否则将会影响到将来对复位状态的判断。如果发生掉电的情况（VDD低于Vrst）,那么系统将会回到复位状态。当电源恢复正常，系统会再进行一次上电复位延迟并设置POR标志位。

### 11.2 看门狗定时器复位

看门狗定时器是一个带可编程溢出时间的自由运行的定时器。用户可以在任何时候清除看门狗定时器，使它重新开始计数。当看门狗定时器溢出后，将会产生一个中断（如果该中断打开）如果用户允许看门狗定时器产生复位信号，那么在其溢出（未被清零）且经过512个时钟后看门狗定时器会产生一个复位信号。这样会使系统进入复位状态。这个状态由硬件维持2个机器周期。一旦退出复位状态，系统将从0000H处执行代码。

### 11.3 复位状态

大多数SFR在复位后回到其初始状态。程序计数器被设为0000H，而且只要复位状态一直保持，它也将维持0000H的数值不变。但是复位不影响片上RAM的状态。RAM中的数据在复位期间维持不变。但是堆栈指针变为07H，因此堆栈的数据会丢失。如果VDD低于2V（维持RAM中数据所需的最小电压），那么RAM中的数据就会丢失。因此第一次上电复位后RAM中的数据不确定，而当电源电压跌至2V以下后，RAM中数据丢失。

复位后大多数SFR被清除，中断和定时器被关闭。如果复位源是上电复位，那么看门狗定时器也被关闭。端口特殊寄存器中的值是FF，所以端口上将输出全高电平。由于没有片内上拉，P0口的状态是浮空的。

表6 SFR复位值

| SFR名称 | 复位值 | SFR名称 | 复位值 |
|---|---|---|---|
| P0 | 11111111b | IE | 00000000b |
| SP | 00000111b | SADDR | 00000000b |
| DPL | 00000000b | P3 | 11111111b |
| DPH | 00000000b | IP | x0000000b |
| DPL1 | 00000000b | SADEN | 00000000b |
| DPH1 | 00000000b | T2CON | 00000000b |
| DPS | 00000000b | T2MOD | 00000x00b |
| PCON | 00xx0000b | RCAP2L | 00000000b |
| TCON | 00000000b | RCAP2H | 00000000b |
| TMOD | 00000000b | TL2 | 00000000b |
| TL0 | 00000000b | TH2 | 00000000b |
| TL1 | 00000000b | TA | 11111111b |
| TH0 | 00000000b | PSW | 00000000b |
| TH1 | 00000000b | WDCON | 0x0x0xx0b |
| CKCON | 00000001b | ACC | 00000000b |
| P1 | 11111111b | EIE | xxx00000b |

SFR复位值续

| SFR名称 | 复位值 | SFR名称 | 复位值 |
|---|---|---|---|
| SCON | 00000000b | B | 00000000b |
| SBUF | xxxxxxxxb | EIP | xxx00000b |
| P2 | 11111111b | PC | 00000000b |
| SADDR1 | 00000000b | SADEN1 | 00000000b |
| SCON1 | 00000000b | SBUF1 | xxxxxxxxb |
| ROMMAP | 01XXX110b | PMR | 010xx0x0b |
| EXIF | 0000xxx0b | STATUS | 000x0000b |
| P4 | xxxx1111b | | |

WDCON 中的位按照不同的复位类型进行置位/清0

| | 外部复位 | 看门狗定时器复位 | 上电复位 |
|---|---|---|---|
| WDCON | 0x0x0xx0b | 0x0x01x0b | 01000000b |

POR（WDCON.6 ）在上电复位后置位。 WTRF（WDCON.2）在看门狗定时器复位后置位。 上电复位后会将该位清除。EWT （WDCON.1）也在上电复位时清除，这样就将看门狗定时器复位关闭。看门狗定时器或外部复位不会影响EWT位。

## 12. 中断

W77E58 的中断分2个优先级12个中断源。每个中断源都有相应的优先级设置位 ，标志位中断向量及使能位。另外系统可以关闭或打开所有中断。

### 12.1 中断源

外部中断 $\overline{INT0}$ 和 $\overline{INT1}$ 按照IT0和IT1的设置可以是边沿触发或是电平触发。TCON中的IE0和IE1 位是外部中断的标志位，检测这2位的状况可以知道是否产生了外部中断。在边沿触发模式中，系统在每个机器周期都要采样INTx脚。如果在一个周期里采样到高电平在下一个周期里采样到低电平，那么系统就检测到了一个高电平到低电平的跳变，此时相应的IEx位置位，同时向系统申请中断服务。由于系统在每个机器周期都要对外部中断进行采样，因此外部中断输入脚上的高电平或低电平至少要维持一个机器周期。当系统响应中断执行中断服务程序时，IEx位被自动清除。如果选择电平触发方式，那么中断请求源的低电平信号必须保持到系统响应该中断。在进入中断服务程序时，IEx位不会被硬件清零。如果外部中断输入脚上的电平在中断服务程序完成后依然保持，系统会立即识别该中断再次进入同样的中断服务程序，注意外部中断INT2到 $\overline{INT5}$ 只能是边沿触发。默认条件下，外部中断INT2到 $\overline{INT5}$ 对应的标志位必须由软件来清除。也可以将它们设置为由硬件清除，方法是将T2MOD中的相应位置位。例如如果HC2置位那么硬件会在系统进入外部中断2中断服务程序以后自动清除外部中断2发生标志。

当TF0、TF1 标志位置位时会产生定时器0 和定时器1 中断。当定时器溢出时这些标志位会置位。当执行定时器中断服务程序时，这些标志位会被硬件自动清零。定时器2 中断的产生取决与TF2 和EXF2的逻辑或。当定时器2 溢出或是遇到捕捉/重装事件时这些标志位会置位。当系统执行定时器2 中断服务程序时，这些标志位不会被硬件清除。软件应当解析定时器2中断的类型并清除相应的标志位。

看门狗定时器可以用作系统监控器或是一个简单的定时器。无论以何种方式工作，当定时器超时后。看门狗定时器中断标志WDIF（WDCON.3）会置位，如果EIE.4=1，那么这时会产生一个中断。

所有中断产生标志均可由硬件置位/复位，同样若软件将这些位置位也可以引发中断。各个中断可以由IE寄存器中的相应位来打开或关闭。IE中有一个中断总控制位，可以打开或关闭所有的中断。

串行口会在接收和发送数据时产生中断，串行口有2个中断源分别是SCON中的RI和TI，以及SCON1中的RI_1和TI_1。这些位不会被硬件自动清零，用户必须用软件来将这些位清零。

所有产生中断的位可以由软件来置位和清零，因此可以由软件来引发相应的中断。各个中断可由IE中的相应位来打开或关闭，IE中还有EA位来控制除PFI外所有中断的打开或关闭。

### 12.2 优先级结构

对中断来说，系统为其提供3种优先级：最高、高和低。对中断可将其设置为高优先级或低优先级。但是系统中存在一个预定义的中断处理顺序结构，用于处理同时产生且优先级又相同的中断。结构的具体方式见下表。各个中断按照其中断优先级顺序编号。

表7：中断优先级结构

| 中断源 | 标志 | 优先级 |
|---|---|---|
| 外部中断 0 | IE0 | 1(highest) |
| 定时器0溢出 | TF0 | 2 |
| 外部中断 1 | IE1 | 3 |

| | | |
|---|---|---|
| 定时器1溢出 | TF1 | 4 |
| 串行口 | RI + TI | 5 |
| 定时器2溢出 | TF2 + EXF2 | 6 |
| 串行口1 | RI_1 + TI_1 | 7 |
| 外部中断 2 | IE2 | 8 |
| 外部中断 3 | IE3 | 9 |
| 外部中断 4 | IE4 | 10 |
| 外部中断 5 | IE5 | 11 |
| 看门狗定时器器 | WDIF | 12 (lowest) |

在每个机器周期中系统都会对中断标志进行采样，对已采样的中断标志的轮询和优先级解析也在同一个机器周期内完成。如果中断产生的条件满足，硬件会执行一个内部LCALL指令，使程序进入相应得中断向量处。产生中断的条件如下：

1. 系统没有执行相同优先级或优先级更高的中断服务程序。
2. 当前的轮询周期是当前指令执行时的最后一个机器周期。
3. 当前指令的操作不涉及IP，IE，EIP或EIE寄存器，也不是一条RETI指令。

如果上述条件有一条不满足，LCALL指令就不会执行。轮询周期会在每个机器周期内重复，中断源的采样也在同一个机器周期内完成。如果在一个周期内中断标志置位但系统没有对它响应，或是在上述条件满足时中断源又无效，那么中断服务就不会被执行。就是说中断不会被记录，每个轮询周期都是新的。

处理器响应中断后，执行一个LCALL指令，使系统进入相应得中断向量。这是系统会自动或不会自动清除中断标志位。当发生定时器中断时，当系统进入定时器中断向量后，系统会自动清除相应的标志位。在发生外部中断时，对于INT0和INT1只有当他们是边沿触发时，硬件才会将他们清零。对于串行口中断，其中断标志不会被硬件清除。对于定时器2中断，硬件也不会自动把他的中断标志位清除。看门狗定时器中断标志WDIF需要用软件来清除。硬件LCALL指令和软件LCALL指令的功能完全一样。这条指令将程序计数器存入堆栈，但不保存程序状态字。PC中装入，相应中断向量的入口地址。相应的入口地址如下：

表8 各个中断源的向量地址

| 中断源 | 向量地址 | 中断源 | 向量地址 |
|---|---|---|---|
| 定时器0溢出 | 000Bh | 外部中断 0 | 0003h |
| 定时器1溢出 | 001Bh | 外部中断 1 | 0013h |
| 定时器2溢出 | 002Bh | 串行口 | 0023h |
| 外部中断 2 | 0043h | 串行口1 | 003Bh |
| 外部中断 4 | 0053h | 外部中断 3 | 004Bh |
| 看门狗定时器 | 0063h | 外部中断 5 | 005Bh |

向量地址并没有进行一致分配，这是为了能够满足将来系统扩展的需要。

程序会在向量入口处执行，直到遇到RETI指令。在执行RETI指令时，处理器将堆栈的内容弹出，并为PC装入堆栈顶上的数据。用户必须注意，堆栈回到了硬件LCALL指令后的状态。如果堆栈的内容被修改，处理器是不会知道的，将依旧从被中断后的地方继续执行。注意RET指令的执行效果与RETI指令一样，但它不会通知中断系统，中断处理已经完成，处理器会认为系统仍在进行中断服务程序。

### 12.3 中断响应时间

对中断的响应时间取决于多个因素，例如中断的类型和当前正在执行的指令。对于外部中断 $\overline{INT0}$ 到 $\overline{INT5}$ 系统会在每个机器周期的C3态被采样，然后相应的IEx标志位将被置位或复位。当定时器0，1 溢出后，其溢出标志会在机器周期的C3态置位。这些标志位会在下一个机器周期内被轮询。如果产生一个中断请求，且中断响应的条件满足，硬件会执行一个LCALL指令。LCALL指令需要4个机器周期来完成。应次在中断标志位被置位，到进入相应的中断向量至少需要5个机器周期。

如果上述条件不满足，那么中断响应时间将会延长。如果系统正在执行一个更高优先级或优先级相同的中断服务程序，那么中断响应的时间就取决于中断的类型了。如果轮询周期不是当前正在执行的指令的最后一个机器周期，那么响应时间会被延迟。（如果没有进行其它的中断服务程序）则最大的中断响应时间为W77E58在执行IE，IP，EIE，EIP寄存器写操作或是执行乘法，除法指令的时候。从中断源产生到中断响应最长的延迟时间是12个机器周期，其中包括用于检测中断的1个机器周期，用来完成对IE，IP，EIE，EIP寄存器操作的2个机器周期，用来完成乘除指令的5个机器周期，及用以完成硬件LCALL指令的4个机器周期。

因此在一个单中断系统中，中断响应时间最少5个机器周期最长12个机器周期。最长的12个机器周期的响应时间为48个时钟周期。标准8051的最大延迟为8个机器周期（96个时钟周期）。这样W77E58就最大把中断响应时间减小了50%。

## 13. 编程定时器/计数器

W77E58有3个16位可编程定时器/计数器，一个可编程看门狗定时器。看门狗定时器的运行方式不同于其它3个定时器。

### 13.1 定时器/计数器0 & 1

W77E58有2个16位定时器/计数器，这些定时器中都有2个8位寄存器以构成16位的计数寄存器。对于定时器0它们是TH0（高8位的计数寄存器）和TL0（低8位的计数寄存器）。定时器1 也有类似的计数寄存器TH1和TL1。可以将它们设置为定时器（对机器周期进行计数）和外部事件计数器。

将它们设置为定时器后，定时器将对时钟周期计数。时钟源可以是系统时钟的12分频或是系统时钟的4分频。在计数器模式下，每当检测到外部计数输入脚上的负电平跳变（T0针对定时器0，T1针对定时器1），计数寄存器的内容就会加一。T0和T1上的电平在每个机器周期的C4态被采样，如果在一个机器周期采样到高电平，在下一个机器周期采样到低电平，那么就会确认一个电平由高到低的跳变，计数器寄存器指针加一。由于需要2个机器周期来确认管脚上的电平负跳变，因此外部输入信号的最大频率是主频的24分之一。无论是定时器还是计数器，计数寄存器都在机器周期的C3态加一。因此在定时器模式下，在T0 和T1 脚上检测到的电平负跳变会在紧跟着检测到该电平跳变后的那个机器周期中使计数器加1。

由TMOD寄存器中的C / $\overline{T}$ 位来确定定时器/计数器以何种方式工作。每个定时器/计数器都有它自己的模式选择位；TMOD中用第2位选择定时器/计数器0 的功能、第6位来选择定时器/计数器1的功能。此外每

个定时器/计数器都可以选定4种运行方式中的一种来运行。由TMOD中的M0 和M1位来选择定时器的工作模式。

## 13.2 时钟源选择

W77E58为定时器提供2种时钟源，一种是标准8051时钟源，即系统工作频率的1/12为计数时钟源。这种运行方式保证了时间循环与标准的8051一致，这也是W77E58默认的定时器时钟来源。用户也可以选择让时钟以加速的方式来运行，这时的计数时钟源是系统工作频率的1/4，这样就将计数速度加快了3倍。由CKCON中的T0M和T1M 位来选择加速计数模式。复位后这些为变为0 ，定时器工作在标准8051模式下。如果用户要将计数器设为加速模式，应该把这2位置1。

**模式0**

模式0下，是13位的定时器/计数器，由8位的THx和TLx的低5位组成。TLx会在时钟源的负跳变处加一，当TLx的第五位由1变0后，THx开始计数。当THx的数值由FF变为00以后，TCON中的溢出标志位TFx会置位。当TRx置位且GATE为0或$\overline{INTx}$ 为1时，计数输入才有效。 C / $\overline{T}$ =0时，定时器/计数器对时钟周期进行计数， C / $\overline{T}$ =1时对P3.4(T0)

以及P3.5（T1）上的1到0跳变进行计数。当13位的定时器计数值变为1FFFH后，下一次计数会使其变为0000H。此时相关的溢出标志位置位如果中断打开，此时还会产生一个定时器中断。注意如果将其用作定时器那么时钟源可以是系统时钟周期的1/12或1/4。

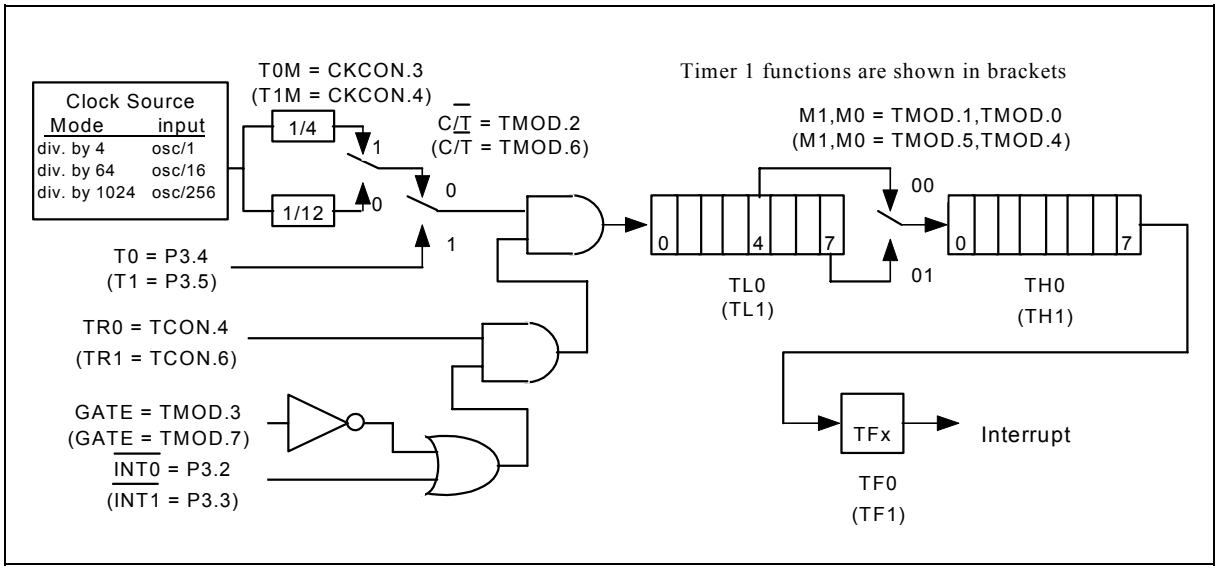



图11： 模式0和模式1下的定时器/计数器

**模式1：**

模式1与模式0 非常相似，只是模式1下定时器/计数器为16位的，而非13位。就是说是用THx和TLx的全部16位来计数。当计数值由FFFFH向0000H翻转后，相应的溢出标志置1，并产生中断。对时钟源的选择与模式0 下的方式一致，门控方式也同模式0 相同。

**模式2:**

模式2下定时器/计数器为自动重装模式。此模式下TLx是一个8位的计数器，THx保存重装计数值。当TLx由FFH向00H溢出后，TCON中的TFx标志置位THx中内容重装至TLx，继续计数过程。重装过程中

THx内的值保持不变。当TRx置位且GATE为0或$\overline{INTx}$为1时，计数器才真正开始工作。同其它2种方式一样，模式2的时钟源可以是系统时钟周期的1/12或1/4。也可对Tn脚上的脉冲输入计数。

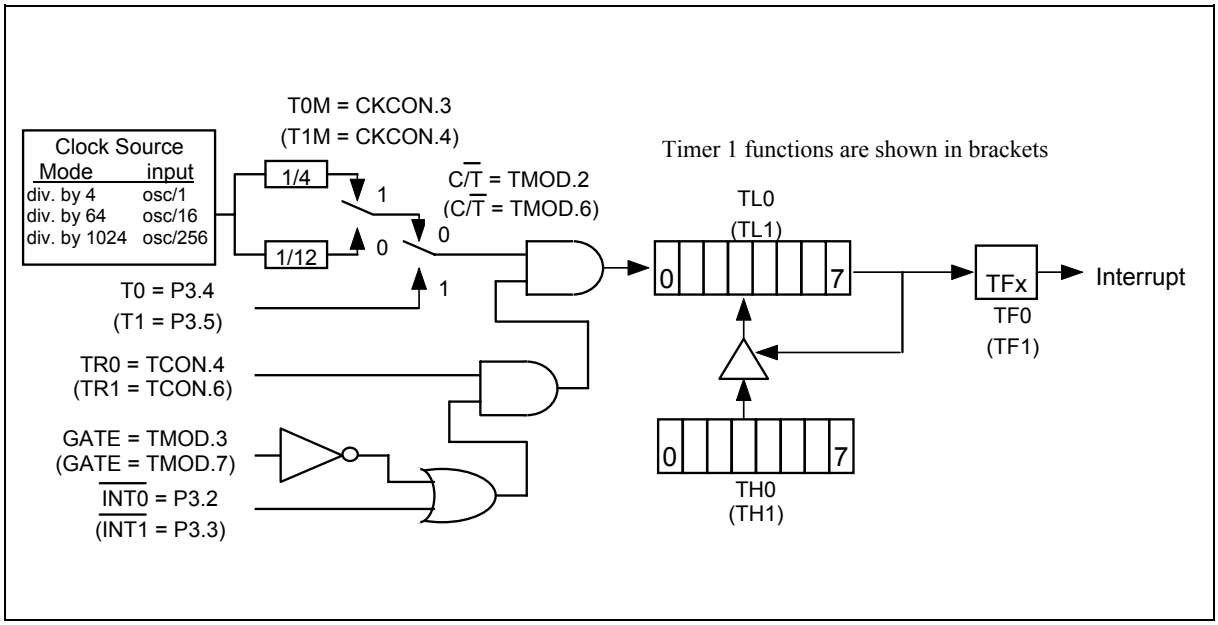



图12：模式2下的定时器/计数器.

**模式3：**

对2个定时器/计数器来说，他们的模式3 有着不同的工作方式。对定时器/计数器1 来说模式3 会将其停止；对定时器/计数器0来说模式3下TL0 和TH0是 2个独立的8位计数寄存器。下图表示这种模式下的逻辑关系。模式3下TL0用定时器0的控制位：如C / T, GATE, TR0, INT0 和TF0。TL0可以用来对时钟周期来计数（时钟源的1/12或1/4）以及对T0 脚上的1到0 跳变计数。TH0 只能对内部时钟源计数，并使用定时器/计数器1的控制位（TR1和TF1）。当需要额外的8位定时器时可以使用模式3 。 当定时器0处于模式3时，定时器1依然可以工作在模式0、1、2下，但它的灵活性受到限制。虽然基本功能得以维持，但已不能对TF1和TR1进行控制。此时定时器1依然可以使用GATE及INT1脚。另外可以通过将其放入或离开模式3的方式来打开或关闭它。它同样可以用作串行口的波特率发生器。

## 13.3 定时器/计数器2

定时器/计数器2 是由T2MOD进行配置并受T2CON寄存器进行控制的向上/向下定时器/计数器。定时器/计数器2有捕捉和重装功能。同定时器0、1一样定时器2 有灵活的设置方式和对时钟源的选择。定时器/计数器2 的时钟源可以是外部输入时钟（T2 脚），也可是被12或者4分频的震荡体时钟。TR2=1时该时钟运行，TR2=0时该时钟停止。

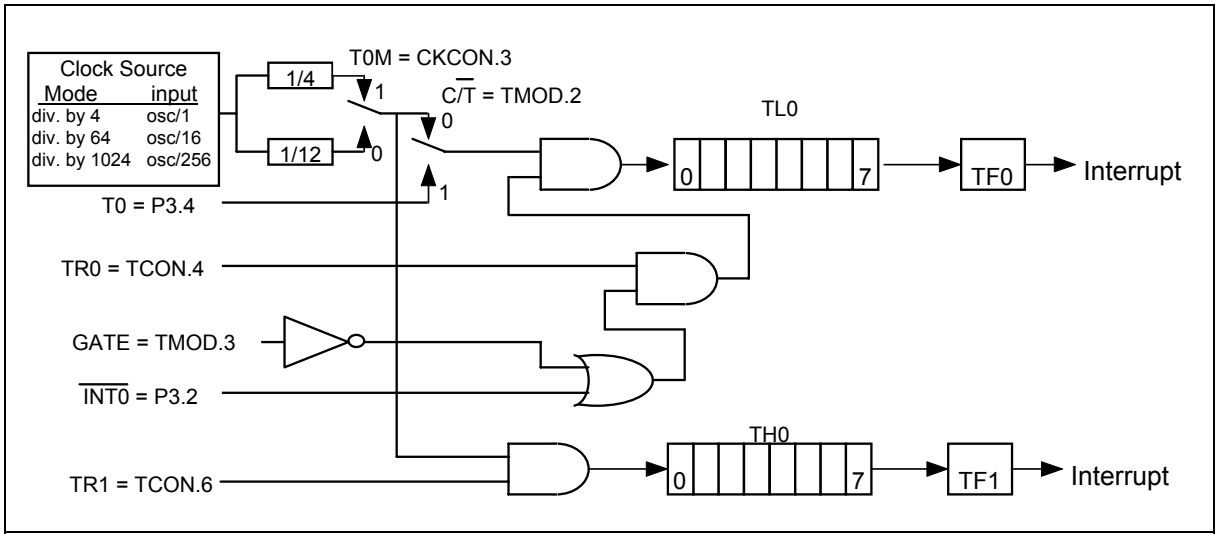



图3：模式0下的定时器0

### 捕捉模式

捕捉模式由T2CON中的CP / RL2 位来设置，置1后定时器/计数器2进入捕捉模式。在捕捉模式下定时器/计数器2 为一个16位向上计数器。当计数值由FFFFH变为0000H后TF2置位并且产生一个中断。如果EXEN2=1，那么T2EX 脚上的负跳变会使TL2和TH2中的数值装入RCAP2L和RCAP2H寄存器中。此时T2CON中的EXF2 位会置位，并产生一个中断。将T2CR位置位，W77E58会在TL2和TH2中的值被捕捉后自动将定时器2复位。

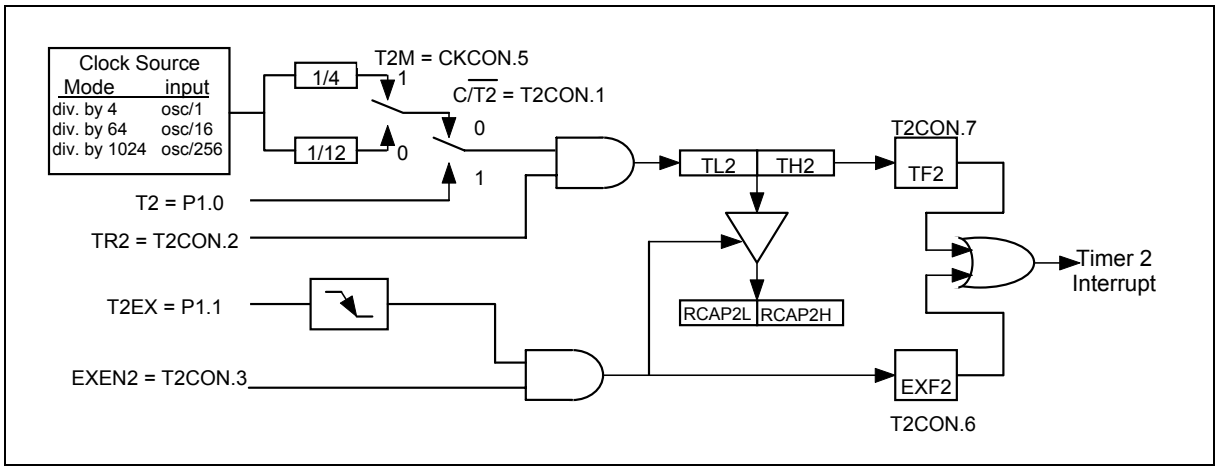



图14：16位捕捉模式

### 向上计数，自动重装方式

当T2CON中 CP / $\overline{RL2}$ =0且T2MOD中DCEN=0时定时器2进入向上计数，自动重装方式。此模式下定时器2是16位的向上计数器，当计数值由FFFFH向0000H翻转时，RCAP2L和RCAP2H中的内容被自动重装至TL0和TH0。重装时TF2置位。如果EXEN2=1，那么T2EX脚上的负跳变也会引起一次重装动作，这时T2CON中的EXF2位置位。

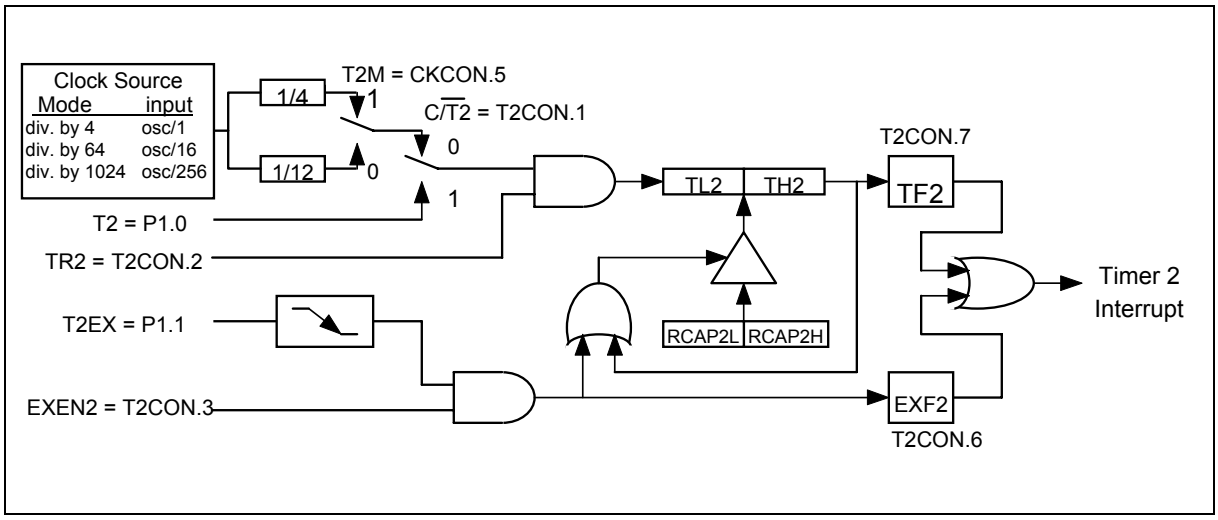



图15：16位自动重装向上计数器

### 向上/向下自动重装计数器

当T2CON中 CP / $\overline{RL2}$ =0且T2MOD中DCEN=1时定时器2进入向上/向下计数，自动重装方式。此模式下定时器2是计数方向受T2EX控制的计数器。当T2EX脚上的电平为1，计数器就向上计数。当向上计数溢出后，捕捉寄存器中的数值被自动重装至计数器中。当计数器的数值计数到与捕捉寄存器中的数值相同时，TL2和TH2中会自动装入FFFFH并开始向下计数。2种情况下重装时都会使TF2、EXF2置位，但在这种模式下EXF2 的置位不会引发中断。

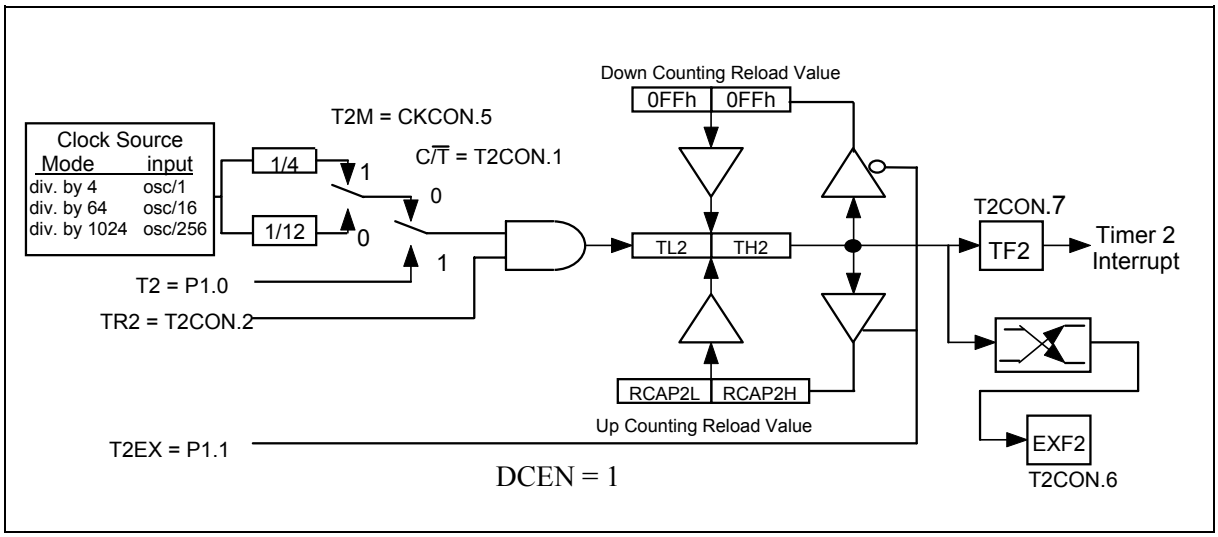



图16：16位自动重装向上/向下计数器

### 波特率发生器

当T2CON中的RCLK=1且TCLK=1时，定时器2 进入波特率发生器模式。在此模式下，定时器2是一个16位的自动重装计数器，当计数值从FFFFH向0000H翻转后TL2 和TH2 会自动重装。这时TF2 不会置位，如果EXEN2=1，那么T2EX脚上的负跳变会使T2CON中的EXF2置位，产生一个中断。

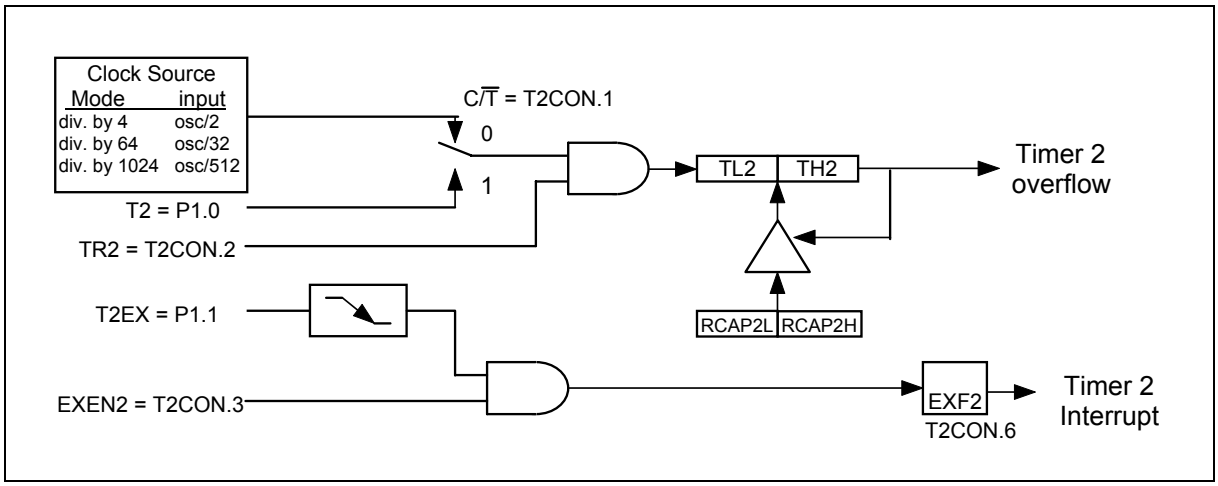



图17：波特率发生器模式

### 可编程时钟输出

定时器2有一个新的时钟输出功能，可以在P1.0上输出占空比为50%的波形。它可以用作可编程时钟发生器。要使它输出时钟信号，软件必须使T2OE = 1，C/T2 = 0，CP/RL = 0。将TR2置1将打开定时器。该模式与波特率发生器模式类似。但在定时器2溢出后，系统不产生中断。因此可以把定时器2同时用作彼特率发生器和时钟发生器。时钟输出频率由下面的等式决定。

**时钟输出频率 = 晶振频率 / [4 X (65536-RCAP2H, RCAP2L) ]**

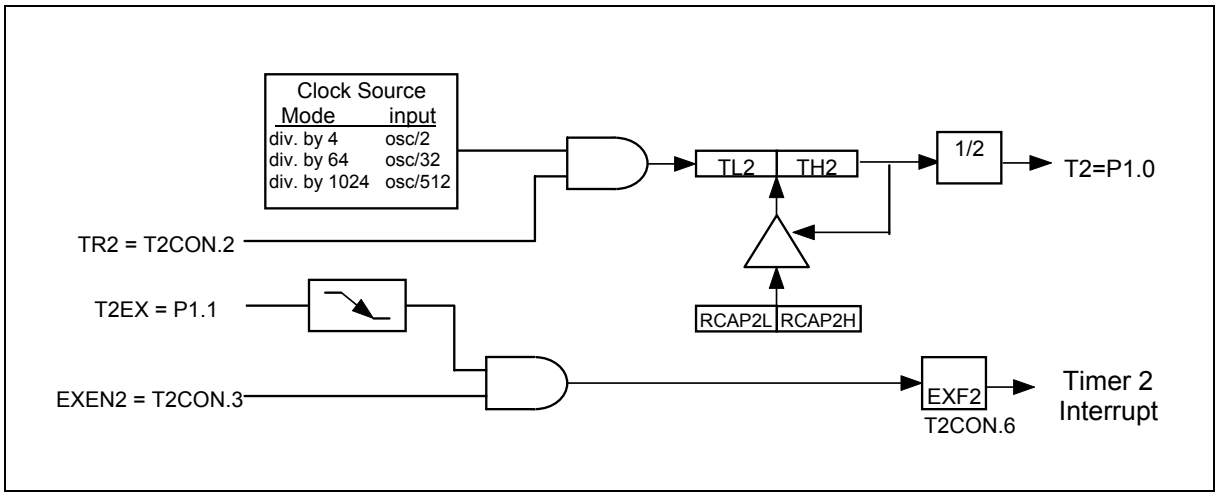



图18：可编程时钟输出模式

## 14. 看门狗定时器

看门狗定时器是一个自行运行定时器，用户可通过编程将其设置为系统监控器，时基发生器或事件定时器。该定时器基于一组分频器，对系统时钟频率进行分割。分频器输出可选，并决定溢出时间。溢出时，如果看门狗有效（且看门狗定时器复位打开），将引起系统复位。看门狗溢出中断以及看门狗复位功能可由软件设置，将2者的功能合并或分离（即看门狗定时器溢出并使系统复位以及看门狗定时器仅溢出而不引发系统复位）。

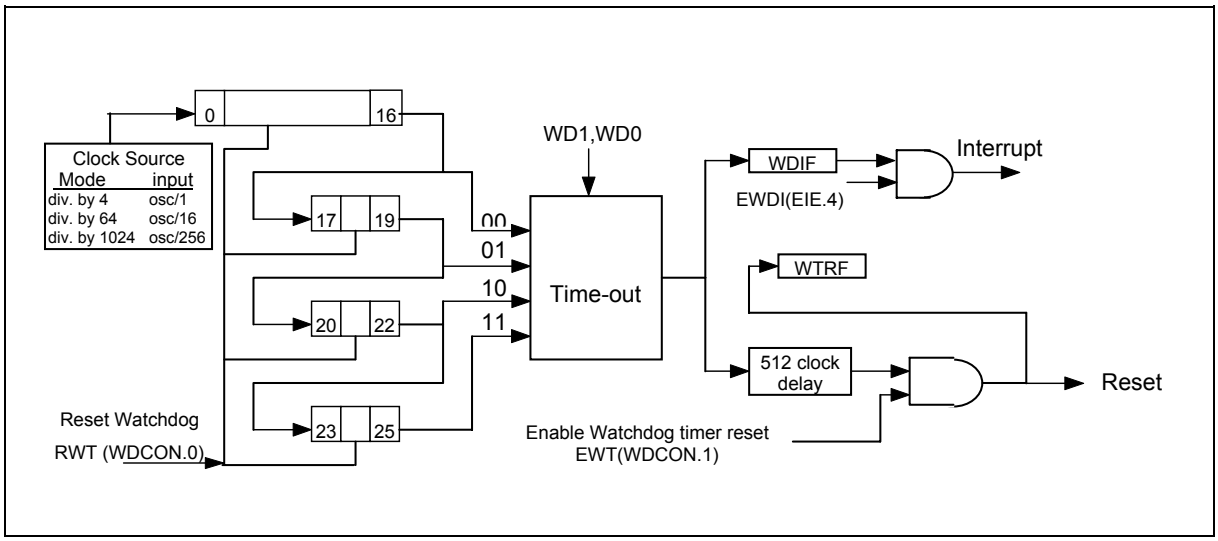



图19：看门狗定时器

看门狗定时器应先用RWT来重新启动，这保证看门狗定时器从一个确定状态开始运行。RWT位用来复位看门狗定时器。该位会自动清0，就是说在软件向该位写入1后，系统会自动把该位清为0。将RWT位设为1后，看门狗定时器会对时钟周期进行计数。超时时间由WD1和WD0位来决定（CKCON.7和CKCON.6）。超时时间到以后，WDIF位置位；之后看门狗定时器将等待512个时钟周期，如果EWT=1且在等待期间没有对RWT进行操作，那么512个时钟周期以后会产生看门狗定时器复位。这个复位会持续2个机器周期同时WTRF标志位置位，软件可以用此位来判别是否是看门狗定时器复位。

看门狗定时器可以用作一个简单的定时器，此时中断和复位功能被关闭。每次超时时间到以后WDIF位会置位。可以对WDIF位进行轮询来检测看门狗定时器的溢出与否，并用RWT位来复位看门狗定时器。看门狗定时器也可用作一个能超长计时的定时器，在这种模式下看门狗定时器中断有效，每次溢出后并在EA=1 时会产生看门狗定时器中断。

看门狗定时器主要用作一个系统监控器，在实时控制的应用中尤为重要。如果出现电源脉冲干扰或电磁干扰，处理器将会运行不确定的代码。如果不及时检查，整个系统可能会崩溃。用户可以在软件中使用看门狗定时器来防止程序运行的错误；用户在软件中适当的地方安排看门狗定时器复位程序，每当运行到看门狗定时器复位程序时就将看门狗定时器复位防止看门狗定时器复位的产生。如果系统受到干扰，程序运行发生异常，系统就可能不会运行看门狗定时器的复位代码，此时系统就会被看门狗定时器复位。

对于不同的时钟速率，看门狗定时器将会产生不同的溢出时间。当使能看门狗定时器复位后，这个复位会在其溢出并经过512个时钟周期后结束。

表9：看门狗定时器溢出值

| WD1 | WD0 | Watchdog Interval | 时钟数目 | 1.8432 MHz下的时间 | 10 MHz下的时间 | 25 MHz下的时间 |
|---|---|---|---|---|---|---|
| 0 | 0 | $2^{17}$ | 131072 | 71.11 mS | 13.11 mS | 5.24 mS |
| 0 | 1 | $2^{20}$ | 1048576 | 568.89 mS | 104.86 mS | 41.94 mS |
| 1 | 0 | $2^{23}$ | 8388608 | 4551.11 mS | 838.86 mS | 335.54 mS |
| 1 | 1 | $2^{26}$ | 67108864 | 36408.88 mS | 6710.89 mS | 2684.35 mS |

看门狗定时器在上电或掉电复位后无效，看门狗定时器复位不会关闭看门狗定时器，但会将它重新启动，软件应重新启动看门狗定时器把它放入一个确定的状态。

### 14.1 门狗定时器控制

WDIF: WDCON.3－看门狗定时器中断标志。在看门狗定时器溢出后该位置位。如果看门狗定时器中断使能，并且系统总中断打开，那么会产生看门狗定时器中断。软件或任何形式的复位都可以将该位清除。

WTRF: WDCON.2－看门狗定时器复位标志。当看门狗定时器复位后置位。该位可用来判别复位的类型。软件可以读取该位，但必须手动清除。掉电复位会将此位清除。如果EWT＝0，该位不会受看门狗定时器的影响。

EWT: WDCON.1－看门狗定时器复位使能位。为1时使能看门狗定时器复位功能为0 关闭该功能，此时看门狗定时器自由运行。

RWT: WDCON.0 – 将看门狗定时器复位。该位用于清除看门狗定时器并将它复位。该位会自动清零，在软件向该位写入1后，系统会自动将它置0。如果看门狗定时器复位使能，那么软件必须在看门狗定时器溢出后512个时钟周期内将看门狗定时器清零，否则将会产生一个看门狗定时器复位。

### 14.1 时钟控制

WD1, WD0: CKCON.7, CKCON.6 – 看门狗定时器模式选择位。这2位用来选择看门狗定时器的溢出时间。复位在定时器溢出并经过512个时钟周期后发生。

默认的看门狗溢出时间是$2^{17}$ 个时钟，是最短的溢出时间。EWT,WDIF和RWT是受时控访问限制的位。这种机制可以防止软件意外读写这些寄存器位。更为重要的是，它将防止无关代码关闭，启动看门狗定时器。

## 15. 串行口

W77E58有一个全双工串行口。该串行口还为用户提供帧错误检测、自动地址识别等附加功能。该串行口提供同步及异步通信方式。在同步模式下串行口产生时钟并以半双工的方式工作。在异步模式下，能以全双工的方式工作，即可以同时收发数据。发送，接收寄存器均用SBUF来访问。对SBUF的写是发送数据，从SBUF读是读取数据。串行口能以4种不同的方式工作。

## 15.1 模式 0

该模式提供与外部设备进行同步通信的方式。在该模式下，串行数据由RXD脚进行收发，而TXD脚用于产生移位时钟。在发送或接收时TXD上的时钟由W77E58提供。这种方式下是以半双工的形式进行通信，每帧接收或发送8位数据。数据的最低位被最先发送或接收，波特率固定为振荡源频率的1/12或1/4。波特率由SM2（SCON.5）位来决定，当SM2=0时波特率为时钟平率的1/12，当SM2=1时波特率为时钟频率的1/4。模式0中的可编程波特率功能是标准8051和W77E58的唯一区别。

下图是模式0的功能方块图。数据由RXD线进行收发。TXD线用来输出移位时钟，移位时钟用来给W77E58和其他设备串行接收/发送数据。对SBUF的写将会发送数据，此时移位时钟启动数据从RXD脚串行移出，直至送完8位数据。如果SM2=1，在TXD脚上的移位时钟下跳变之前RXD上的数据会维持1个时钟周期。，之后TXD脚上的电平变低并维持2个时钟周期，之后TXD脚上电平变高。如果SM2=0，RXD上的数据在TXD变低前会维持3个时钟周期，之后TXD上电平会变低6个时钟周期，之后再变高。这样就保证了在接收端数据可以在TXD的上升沿处同步，在TXD的下降沿处被接收。

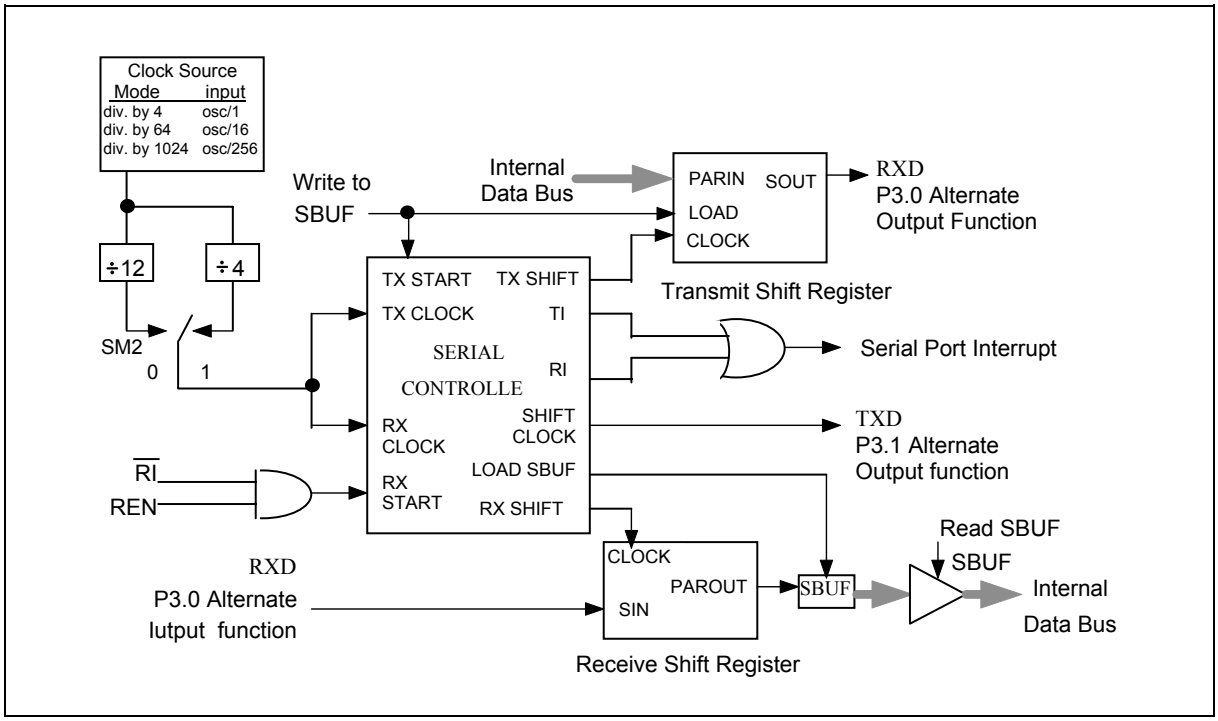



图20：模式0下的串口

TI标志位在发送完最后一位数据后的C1态置1，当REN=1 且RI=0时串行口接收数据。移位时钟被激活，串行口会在移位时钟的上升沿锁定数据。外部设备要在移位时钟的下降沿处送出数据。这个过程持续到8位数据全部发送完毕。RI会在TXD的最后一个下降沿处置1，这时接收动作结束，RI要由软件清零。

## 15.2 模式 1

在模式1下，串行口以全双工的方式工作。串行通信的数据贞由10位数据组成，在RXD和TXD脚上进行收发。10位数据组成如下：起始位（位0），8位数据（最低位在前），终止位（1）。在接收端，停止

位进入SCON的RB8位。在该模式下波特率可变，波特率可以是定时器1溢出率的1/16或1/32。由于定时器1 的溢出率可以按需要设定，因此波特率的选择范围很宽。

向SBUF写入数据后将启动一次发送动作，串行数据的第一位在一个16分频计数器的第一次翻转后的C1态，被送到TXD脚，下一位数据在下一次16分频计数器翻转后的C1态送至TXD脚。因此数据的传送与这个16分频的计数器同步，而不是直接写入接收端的SBUF。在发送完9位数据后，会发送停止位。在停止位输出到TXD脚以后，TI会在C1态置位。这发生在向SBUF写入数据后16分频计数器的第11次翻转以后。当REN=1 时系统进行接收操作，接收器以所选波特率的16倍速度采样RXD脚状态。

当RXD脚上接收到1-0跳变就启动接收器接收。接收的值是3次采样中至少2次相同的值，以保证接收准确。在起始位，如果接收到的值不为0 ，则起始位无效，复位接收电路，当再次接收到一个由1-0 的跳变时重新启动接收器。如果接收值为0 起始位有效，接收器开始接收本页的其余信息。

在接收了8位数据以后，还将接收一个停止位，进入RB8，之后RI置位。但这种情况是在RI=0，且SM2=0或接收到的停止位为1时才有效。

如果上述条件满足，则停止位进入RB8，8位数据进入SBUF，RI置位，否则丢弃接收到的贞数据。在停止位的中间，接收器重启，开始新的一次接收。

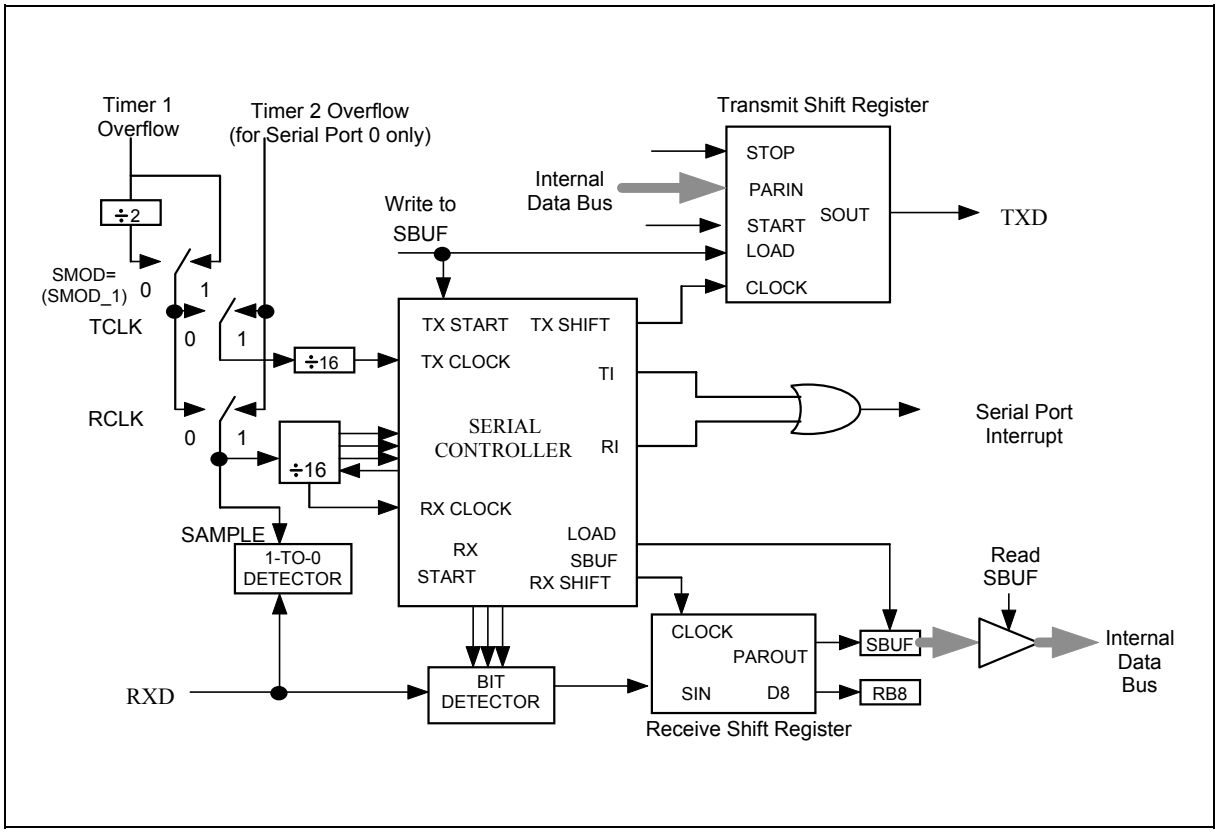



图21：模式1下的串口

## 15.3 模式 2

该模式用11位数据来进行全双工异步通信。下图是对他的功能描述。数据由起始位位（0），8位数据（最低位在前），可编成的第9位数据（TB8）和停止位组成。第9位数据接收至RB8。波特率是时钟频率的1/32 或1/64，由PCON中的SMOD位来选择。向SBUF中写入数据启动一次发送， 串行数据的第一位在一个16分频计数器的第一次翻转后的C1态，被送到TXD脚，下一位数据在下一次16分频计数器翻转后的C1态送至TXD脚。因此数据的传送与这个16分频的计数器同步，而不是直接写入接收端的SBUF。在发送完9位数据后，会发送停止位。在停止位输出到TXD脚以后，TI会在C1态置位，这发生在向SBUF写入数据后16分频计数器的第11次翻转以后。

当REN=1 时系统进行接收操作，接受器以所选波特率的16倍速度采样RXD脚状态。当RXD脚上接收到1-0跳变就启动接收器接收。接收的值是3次采样中至少2次相同的值，以保证接收准确。在起始位，如果接收到的值不为0 ，则起始位无效，复位接收电路，当再次接收到一个由1-0 的跳变时重新启动接收器。如果接收值为0 起始位有效，接收器开始接收本贞的其余信息。

在接收了9位数据以后，还将接收一个停止位，进入RB8，之后RI置位。但这种情况是在RI=0，且SM2=0或接收到的停止位为1时才有效。

如果上述条件满足，则停止位进入RB8，8位数据进入SBUF，RI置位，否则丢弃接收到的贞数据。在停止位的中间，接收器重启，开始新的一次接收。

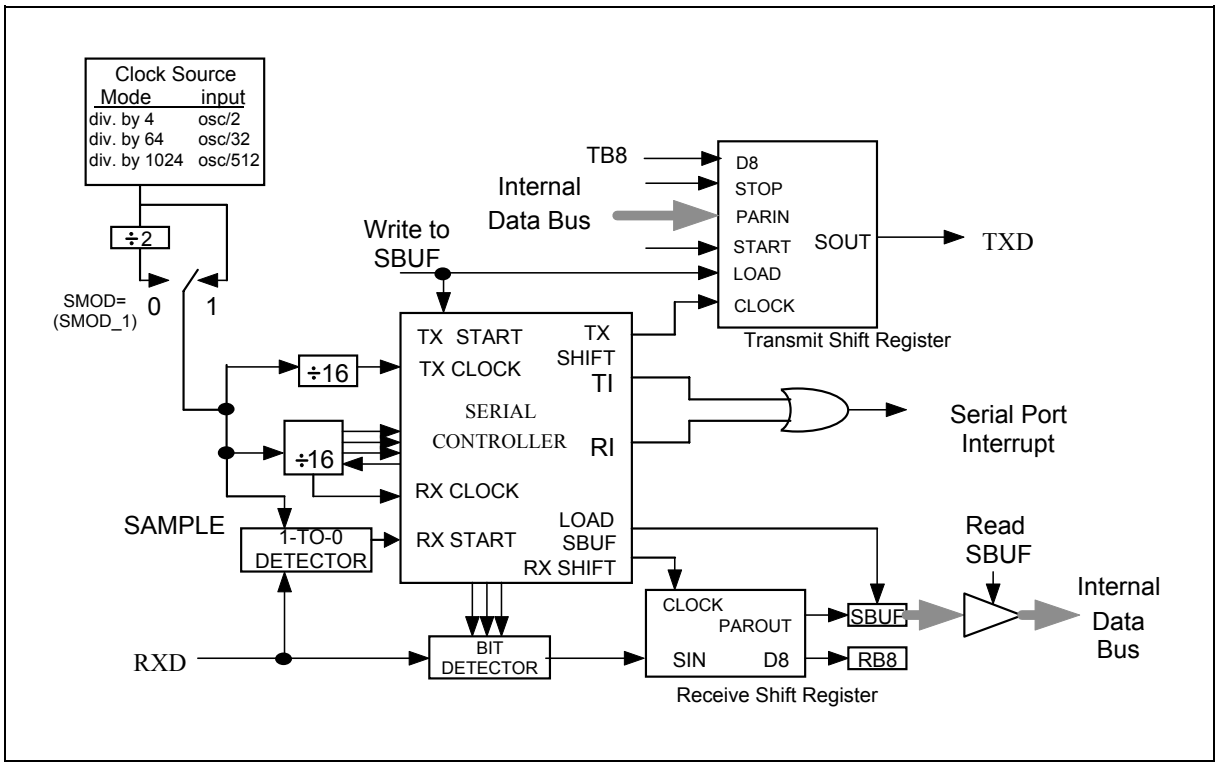



图22：模式2下的串口

## 15.4 模式3

模式3中除了波特率可编程外，其他方面都与模式2相同。用户必须在进行串行通信前初始化SFR寄存器。初始化动作包括模式和波特率的选择。如果是用模式1 或模式3，那么定时器1也要被初始化。在所有的模式中向SBUF写入数据将启动一次发送。在模式0中当RI=0和REN=1时启动一次接收。这时TXD脚上会出现同步时钟，并在RXD脚上传送8位数据。在其他模式下，接收动作在REN=1且接收到数据后就启动。外部设备以发送起始位的方式来开始串行通信。

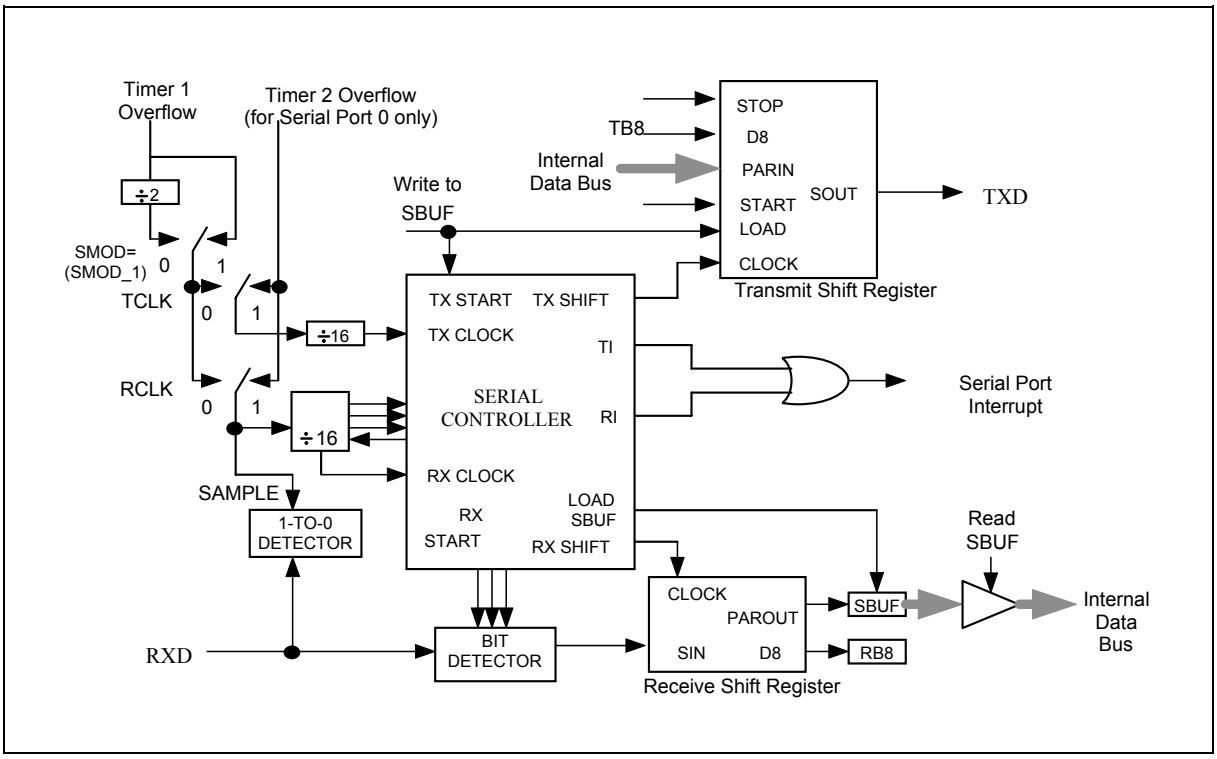



图23：模式3下的串行口

**表10串行口的模式**

| SM1 | SM0 | 模式 | 类型 | 波特率时钟 | 贞大小 | 起始位 | 终止位 | 第9位 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 同步 | 4 or 12 TCLKS | 8 位 | 无 | 无 | 无 |
| 0 | 1 | 1 | 异步 | Timer 1 or 2 | 10 位 | 1 | 1 | 无 |
| 1 | 0 | 2 | 异步 | 32 or 64 TCLKS | 11 位 | 1 | 1 | 0, 1 |
| 1 | 1 | 3 | 异步 | Timer 1 or 2 | 11位 | 1 | 1 | 0, 1 |

## 15.5 贞错误检测

当没有检测到一个有效的停止位时，可能就出现了一个贞错误。这表示一个无效的串行数据接收。通常错误是由串行通信线上的干扰造成的。W77E58可以检测这种错误，并将标志位置位，以供软件进行检测。.

SCON.7.是FE标志（贞错误标志）（FE_1）。在标准8051种该位是SM0 ，但在W77E58中它有附加功能称为SM0/FE。他们其实是相互独立的标志位。一个是SM0 ，一个是FE。具体访问哪一个位是由SMOD0（PCON.6）决定的。当SMOD0=1时访问FE标志位，当SMOD0=0时访问SM0位。

FE标志由硬件置位且必须由软件清0。注意在对FE标志位进行读写时，SMOD0必须为1。如果FE置位，那么下次接收到的正确数据贞不会将其清除。对该位的清除必须由软件来完成。

## 15.6 多机通信

多机通信利用模式2和模式3下的第九位数据，RI仅在接收的数据贞的地址符合本机地址或系统进行广播通信时置位。硬件所具有的特性，免除了要软件进行地址识别的麻烦。

在多机通信模式下，当第9位置1 时，发送的数据是地址贞。当主机想对从机发送数据块，它首先发送从机的地址贞，当从机在接收地址贞时，他们的SM2 位必须为高。这保证他们能在接收到地址贞时产生中断。自动地址识别功能保证只有在接收到的地址和本机地址符合时才产生中断。地址比较由硬件来完成。

被寻址的从设备将SM2位清零，然后准备开始接收数据。SM2=0后，每当接收到一个有效数据贞从机就会产生一个中断。未被寻址的从设备不会受到影响，因为他们在等待自身地址的到来。在模式1 中，第九位是停止位，1是有效的停止数据。如果SM2=1那么只有在接收到有效数据且自身被寻址后RI才会置1。

主机可以用从机地址来选择性的访问从机。可以用广播的方式来寻址所有的从机。从机的地址由SADDR和SADEN寄存器来定义，从机地址是由SADDR设定的8位数据，如果SADEN中相应的位置0则SADDR中对应的位就无效。只有当SADEN中的相应位为1，SADDR中的数据才有效。

下面的代码说明如何定义从机地址，以及寻址不同的从机

Slave 1:

SADDR 1010 0100

SADEN 1111 1010

Given 1010 0x0x

Slave 2:

SADDR 1010 0111

SADEN 1111 1001

Given 1010 0xx1

从机1和2 的地址在最低位处不同， 在从机1中该位被忽略，而在从机2中该位有效。因此要与从机2通信的话，那么他地址数据的位1应该为1。如果主机要与所有从机通信，那么地址数据的位0=1且位1=0。位3被忽略。这样就形成了广播地址。

主机能用广播的方式来和所有从机通信，地址是SADDR和SADEN中数据的逻辑与。相应得位如果为0，那么该位就被忽略。在大多数应用场和，广播地址是FFH，而在上面的例子中从机1的广播地址是（1111111X），从机2的广播地址是（11111111）。

SADDR 和SADEN的地址分别是 A9h和B9h。复位后，2个寄存器的值均为0；这样广播地址和给定的地址都无效，这样多机通信功能就被关闭。

## 16. 时控访问保护

W77E58有许多新的功能，如看门狗定时器，片上ROM 大小调整，等待状态控制信号，上电/掉电复位标志，这些对系统的正常运行来说非常的重要。如果不加以保护，无关代码可能会改写看门狗定时器的相应位，而使系统工作不正常或失控。为了保护这些位，W77E58提供了一种保护机制，来控制对这些位的写操作。这种保护是通过时控访问来实现的。

在这种方式下，对被保护的位的访问是受时间限制的。要对他进行写操作，那么时控窗口必须打开，否则写操作无效。当条件满足时，时控窗口开放3个机器周期。在3个机器周期过后，时控窗口自动关闭。要打开时控窗口，必须先向TA寄存器写入AAH，再写入55H。TA寄存器的地址是C7H，下面列出对时控寄存器进行访问的推荐代码：

```
TA      REG    0C7h          ;定义位于C7H处的新寄存器TA
MOV     TA, #0AAh
MOV     TA, #055h
```

当软件向TA写入AAh后，计数器开始计数，计数器会等待3个机器周期来接受55h;如果在3个机器周期内接收到了55h,那么时控窗口被打开。时控窗口开放3个机器周期，期间用户可以对被保护的位进行读写。一旦时控窗口关闭，那么要重复上述过程来访问被保护的位。

**时控访问的例子：**

例1：有效访问

```
MOV     TA, #0AAh          3 M/C          注: M/C =机器周期
MOV     TA, #055h          3 M/C
MOV     WDCON, #00h        3 M/C
```

例2：有效访问

```
MOV     TA, #0AAh          3 M/C
MOV     TA, #055h          3 M/C
NOP                        1M/C
SETB    EWT                2 M/C
```

例3：有效访问

```
MOV     TA, #0Aah          3 M/C
MOV     TA, #055h          3 M/C
ORL     WDCON, #00000010B  3M/C
```

例4：有效访问

```
MOV     TA, #0AAh          3 M/C
MOV     TA, #055h          3 M/C
NOP                        1 M/C
NOP                        1 M/C
CLR     POR                2 M/C
```

例5：无效访问

```
MOV     TA, #0AAh          3 M/C
```

```
NOP                     1 M/C
MOV   TA, #055h         3 M/C
SETB  EWT               2 M/C
```

在前2个例子中，对被保护位的写是在3个机器周期以内完成的。例3中对保护位的写操作是在时控窗口关闭后进行的，此时不会对被保护的位产生效果。 例4中是在第4个机器周期对被保护位进行写操作，因此写操作根本无效。

## 17. 片上Flash EPROM特性

W77E58有多种方式来对片上Flash EPROM编程。所有的这些操作都通过RST, ALE, $\overline{PSEN}$, A9CTRL(P3.0), A13CTRL(P3.1), A14CTRL(P3.2), OECTRL(P3.3), $\overline{CE}$(P3.6), $\overline{OE}$(P3.7), A0(P1.0) 及VPP($\overline{EA}$)脚来完成。另外A15–A0(P2.7–P2.0, P1.7–P1.0)和D7–D0(P0.7–P0.0)在这些操作中分别用作地址和数据总线。

### 17.1 读操作

这个操作可以让用户读出他们的代码和安全位。如果锁止位为低，那么读出的数据无效。

#### 输出关闭方式

当$\overline{OE}$ 变高， D7..D0上不会有数据出现。

#### 编程操作

该操作用来将程序和安全位写入Flash EPROM中。当Vpp达到Vcp（12.5v）， $\overline{CE}$ 为低$\overline{OE}$为高时系统进入编程操作。

#### 编程效验操作

要对编程操作后的数据进行效验。这个操作在写完每个字节后进行一次。it will ensure a substantial program margin。

#### 擦除操作

擦除造作是唯一可以让程序数据由0变1的方法。这个操作会将Flash EPROM 中的数据和安全位从0变1。当Vpp达到Vep且$\overline{CE}$为低， $\overline{OE}$ 为高时系统进入擦除操作方式。

#### 擦除效验操作Erase Verify Operation

在擦除操作完成后，要对芯片进行检查看是否成功完成擦除操作。这个操作在VPP = VEP(14.5V), $\overline{CE}$ 为高且$\overline{OE}$ 为低时进行。

#### 编程/擦除禁止操作 /Erase Inhibit Operation

这个操作方式允许对多个芯片以不同的数据并行编程。当P3.6($\overline{CE}$) = VIH, P3.7($\overline{OE}$) = VIH时芯片的编程擦除操作被禁止。

| 操作 | P3.0 (A9 CTRL) | P3.1 (A13 CTRL) | P3.2 (A14 CTRL) | P3.3 (OE CTRL) | P3.6 ($\overline{CE}$) | P3.7 ($\overline{OE}$) | $\overline{EA}$ ($V_{PP}$) | P2,P1 (A15..A0) | P0 (D7..D0) | 注释 |
|---|---|---|---|---|---|---|---|---|---|---|
| 读 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Address | Data Out | |
| 输出禁止 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | X | Hi-Z | |
| 编程 | 0 | 0 | 0 | 0 | 0 | 1 | $V_{CP}$ | Address | Data In | |
| 编程效验 | 0 | 0 | 0 | 0 | 1 | 0 | $V_{CP}$ | Address | Data Out | @3 |
| 擦除 | 1 | 0 | 0 | 0 | 0 | 1 | $V_{EP}$ | A0:0, others: X | Data In 0FFH | @4 |
| 擦除效验 | 1 | 0 | 0 | 0 | 1 | 0 | $V_{EP}$ | Address | Data Out | @5 |
| 编程/擦除禁止 | X | 0 | 0 | 0 | 1 | 1 | $V_{CP}$/ $V_{EP}$ | X | X | |

注释：

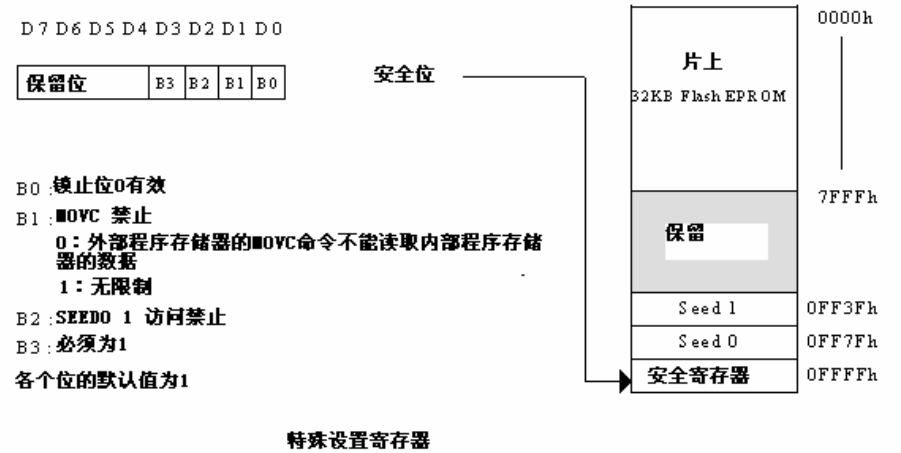
1. 所有操作的运行条件是 RST = $V_{IH}$, ALE = $V_{IL}$ and $\overline{PSEN}$ = $V_{IH}$.
2. $V_{CP}$ = 12V, $V_{EP}$ = 14V, $V_{IH}$ = $V_{DD}$, $V_{IL}$ = $V_{SS}$.
3. 编程效验在一个编程动作完成后进行.
4. 擦除操作可以擦除全部RON单元和安全位
5. 擦除效验在一个擦除动作完成后进行.

## 18. 安全位

在片上 Flash EPROM模式中，Flash EPROM中的数据可以被反复擦写验证，直到用户对Flash EPROM中的代码确定为止，此时系统代码就被保护起来。对Flash EPROM 中内容进行保护的操作描述如下。

W77E58中有可以被编程器访问的特殊设定寄存器。该寄存器在普通模式下不能访问，只能在编程期间被访问。这些位再由1变0后就无法再改变。只有擦除整个芯片中的内容操作，才能将它们复位。

安全位寻址为Flash EPROM操作模式，地址为#0FFFFh。




特殊设置寄存器

**B0：锁止位**

此位是用来保护用户在W77LE532 中的程序代码。在完成编程和校验操作后，设置此位。一旦该位设置为0，就无法再对Flash EPROM的数据和特殊设置寄存器进行访问。

**B1：MOVC 禁止**

此位用来限制MOVC指令的可访问区域。它可防止外部程序存储器的MOVC指令读取内部程序代码。当此位被设置为0，外部程序存储器的MOVC 指令只可以访问外部存储器代码，而不能访问内部存储器。内部程序存储器的MOVC 指令可以访问内部和外部存储器中的ROM数据。如果此位设置为1，则对MOVC指令没有限制。

**B2：加密特性**

该位用来打开/关闭代码保护加密逻辑。一旦加密逻辑打开，那么P0口上的数据是经过加密的。只有在擦除全片时才能关闭该功能。

## 19. 电气特性

| 参数 | 符号 | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| 直流电源电压 | $V_{DD}$–$V_{SS}$ | -0.3 | +7.0 | V |
| 输入电压 | $V_{IN}$ | $V_{SS}$-0.3 | $V_{DD}$+0.3 | V |
| 工作温度 | $T_A$ | 0 | +70 | °C |
| 贮存温度 | Tst | -55 | +150 | °C |

注释：超出最大绝对额定值表所列的情况使用，会对器件的可靠性和寿命造成严重损害。

## 19.1 直流特性

| 参数 | 符号 | 说明 | | | 测试条件 |
|---|---|---|---|---|---|
| | | 最小值 | 最大值 | 单位 | |
| 工作电压 | $V_{DD}$ | 4.5 | 5.5 | V | |
| 工作电流 | $I_{DD}$ | - | 50 | mA | No load<br>$V_{DD}$ = RST = 5.5V |
| 空闲电流 | $I_{IDLE}$ | - | 20 | mA | Idle mode<br>$V_{DD}$ = 5.5V |
| 掉电电流 | $I_{PWDN}$ | - | 50 | μA | Power-down mode<br>$V_{DD}$ = 5.5V |
| 输入电流P1, P2, P3 | $I_{IN1}$ | -50 | +10 | μA | $V_{DD}$ = 5.5V<br>$V_{IN}$ = 0V or $V_{DD}$ |
| 输入电流RST（*1） | $I_{IN2}$ | -10 | +300 | μA | $V_{DD}$ = 5.5V<br>0<$V_{IN}$<$V_{DD}$ |
| P0, EA输入漏电流 | $I_{LK}$ | -10 | +10 | μA | $V_{DD}$ = 5.5V<br>0V<$V_{IN}$<$V_{DD}$ |

DC Characteristics, continued

| 参数 | 符号 | 说明 | | | 测试条件 |
|---|---|---|---|---|---|
| | | 最小值 | 最大值 | 单位 | |
| 逻辑1到0的转换电流<br>P1, P2, P3 | $I_{TL}$[*4] | -500 | -200 | μA | $V_{DD}$ = 5.5V<br>$V_{IN}$ = 2.0V |
| P0, P1, P2, P3，EA<br>输入低电压 | $V_{IL1}$ | 0 | 0.8 | V | $V_{DD}$ = 4.5V |
| RST输入低电压（*1） | $V_{IL2}$ | 0 | 0.8 | V | $V_{DD}$ = 4.5V |
| 输入低电压XTAL1 (*3) | $V_{IL3}$ | 0 | 0.8 | V | $V_{DD}$ = 4.5V |
| P0，P1，P2，P3，EA<br>输入高电压 | $V_{IH1}$ | 2.4 | $V_{DD}$ +0.2 | V | $V_{DD}$ = 5.5V |
| RST输入高电压 | $V_{IH2}$ | 3.5 | $V_{DD}$ +0.2 | V | $V_{DD}$ = 5.5V |
| XTAL输入高电压（*3） | $V_{IH3}$ | 3.5 | $V_{DD}$ +0.2 | V | $V_{DD}$ = 5.5V |
| 输出低电压P1, P2, P3 | $V_{OL1}$ | - | 0.45 | V | $V_{DD}$ = 4.5V<br>$I_{OL}$ = +2 mA |
| 输出低电压<br>P0, ALE, $\overline{PSEN}$[*2] | $V_{OL2}$ | - | 0.45 | V | $V_{DD}$ = 4.5V<br>$I_{OL}$ = +4 mA |
| 输出高电压<br>P1, P2, P3 | $V_{OH1}$ | 2.4 | - | V | $V_{DD}$ = 4.5V<br>$I_{OH}$ = -100 μA |
| 输出高电压<br>P0, ALE, $\overline{PSEN}$[*2] | $V_{OH2}$ | 2.4 | - | V | $V_{DD}$ = 4.5V<br>$I_{OH}$ = -400 μA |

注：

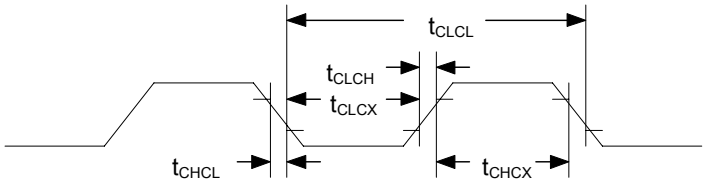
*1. RST脚为施密特触发

*2. P0, ALE and $\overline{PSEN}$ 在外部访问模式中测试

*3. XTAL1是CMOS输入

*4. 当P1 P2 P3 上的管脚被外部拉高或拉低时，他们会产生变迁电流。当VIN为2V时，变迁电流达到最大值

## 19.2 交流特性




Note: Duty cycle is 50%.

外部时钟特性

| 参数 | 符号 | 最小值. | 典型值. | 最大值. | 单位 | 注释 |
|---|---|---|---|---|---|---|
| 时钟高时间 | $t_{CHCX}$ | 12 | - | - | nS | |
| 时钟低时间 | $t_{CLCX}$ | 12 | - | - | nS | |
| 时钟上升时间 | $t_{CLCH}$ | - | - | 10 | nS | |
| 时钟下降时间 | $t_{CHCL}$ | - | - | 10 | nS | |

## 18.1 交流特性说明

| 参数 | 符号 | 时钟最小值 | 时钟最大值 | 单位 |
|---|---|---|---|---|
| 振荡器频率 | $1/t_{CLCL}$ | 0 | 40 | MHz |
| ALE脉冲宽度 | $t_{LHLL}$ | $1.5t_{CLCL}$ - 5 | | nS |
| 地址有效到ALE低 | $t_{AVLL}$ | $0.5t_{CLCL}$ - 5 | | nS |
| ALE低后地址保持 | $t_{LLAX1}$ | $0.5t_{CLCL}$ - 5 | | nS |
| ALE为MOVX指令低后地址保持 | $t_{LLAX2}$ | $0.5t_{CLCL}$ - 5 | | nS |
| ALE低后到指令读入有效 | $t_{LLIV}$ | | $2.5t_{CLCL}$ - 20 | nS |
| ALE 低到PSEN低 | $t_{LLPL}$ | $0.5t_{CLCL}$ - 5 | | nS |
| $\overline{\text{PSEN}}$ 脉冲宽度 | $t_{PLPH}$ | $2.0t_{CLCL}$ - 5 | | nS |
| $\overline{\text{PSEN}}$ 低到指令读入有效 | $t_{PLIV}$ | | $2.0t_{CLCL}$ - 20 | nS |
| PSEN后输入指令保持 | $t_{PXIX}$ | 0 | | nS |
| PSEN后输入指令浮空 | $t_{PXIZ}$ | | $t_{CLCL}$ - 5 | nS |
| Port 0地址保持到指令有效 | $t_{AVIV1}$ | | $3.0t_{CLCL}$ - 20 | nS |
| Port 2地址保持到指令有效 | $t_{AVIV2}$ | | $3.5t_{CLCL}$ - 20 | nS |
| $\overline{\text{PSEN}}$ 低到地址浮空 | $t_{PLAZ}$ | 0 | | nS |
| 读后数据保持 | $t_{RHDX}$ | 0 | | nS |
| 读后数据浮空 | $t_{RHDZ}$ | | $t_{CLCL}$ - 5 | nS |
| 地址浮空变低 | $t_{RLAZ}$ | | $0.5t_{CLCL}$ - 5 | nS |

### MOVX 在使用STRETCH时的特性

| 参数 | 符号 | 时钟最小值 | 时钟最大值 | 单位 | 参数 |
|---|---|---|---|---|---|
| 数据总线ALE脉冲宽度 | $t_{LLHL2}$ | $1.5t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| ALE为MOVX指令低后地址保持 | $t_{LLAX2}$ | $0.5t_{CLCL} - 5$ | | nS | |
| $\overline{RD}$ 信号宽度 | $t_{RLRH}$ | $2.0t_{CLCL} - 5$<br>$t_{MCS} - 10$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| $\overline{WR}$ 信号宽度 | $t_{WLWH}$ | $2.0t_{CLCL} - 5$<br>$t_{MCS} - 10$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| $\overline{RD}$ 低到有效数据输入 | $t_{RLDV}$ | | $2.0t_{CLCL} - 20$<br>$t_{MCS} - 20$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| 读后数据保持 | $t_{RHDX}$ | 0 | | nS | |
| 读后数据浮空 | $t_{RHDZ}$ | | $t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| ALE低到数据有效 | $t_{LLDV}$ | | $2.5t_{CLCL} - 5$<br>$t_{MCS} + 2t_{CLCL} - 40$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Port 0地址保持到指令有效 | $t_{AVDV1}$ | | $3.0t_{CLCL} - 20$<br>$2.0t_{CLCL} - 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| ALE 低到 $\overline{RD}$ 或 $\overline{WR}$ 低 | $t_{LLWL}$ | $0.5t_{CLCL} - 5$<br>$1.5t_{CLCL} - 5$ | $0.5t_{CLCL} + 5$<br>$1.5t_{CLCL} + 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Port 0 地址到$\overline{RD}$ 或 $\overline{WR}$ 低 | $t_{AVWL}$ | $t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Port 02地址到$\overline{RD}$ 或 $\overline{WR}$ 低 | $t_{AVWL2}$ | $1.5t_{CLCL} - 5$<br>$2.5t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| 数据有效到 WR 发送 | $t_{QVWX}$ | -5<br>$1.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| 写后数据保持 | $t_{WHQX}$ | $t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| $\overline{RD}$ 低到地址浮空 | $t_{RLAZ}$ | | $0.5t_{CLCL} - 5$ | nS | |
| $\overline{RD}$ 或 $\overline{WR}$ 高到ALE高 | $t_{WHLH}$ | 0<br>$1.0t_{CLCL} - 5$ | 10<br>$1.0t_{CLCL} + 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |

注: $t_{MCS}$是与 Stretch memory 周期选择有关的参数.下表列出$t_{MCS}$ 周期.

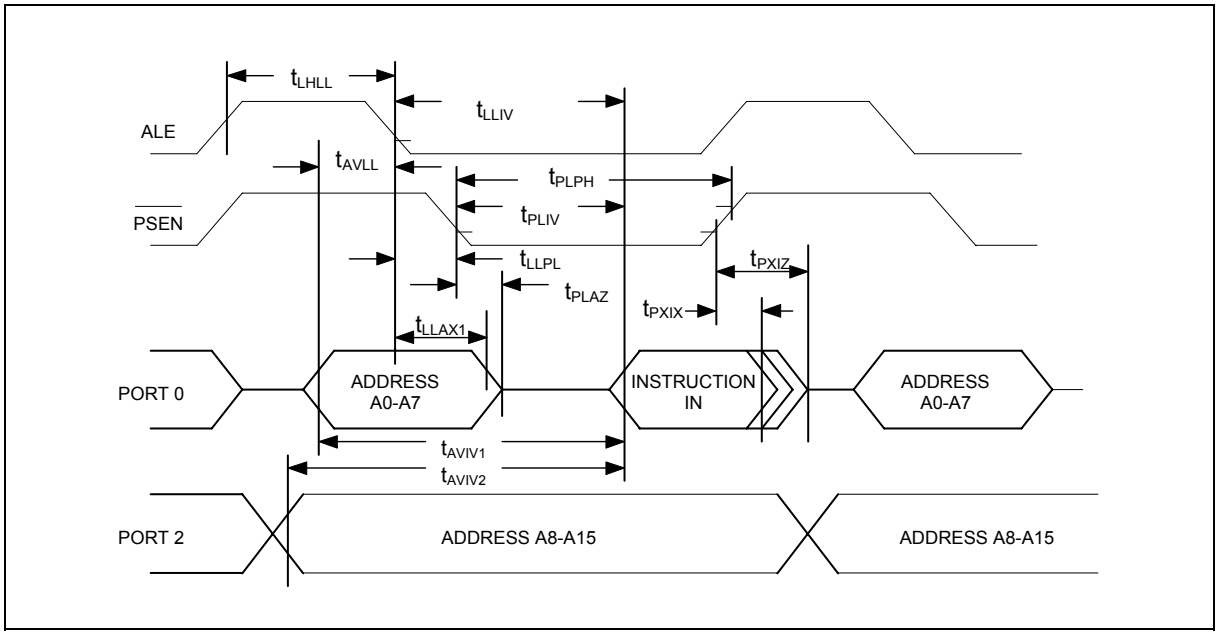
| M2 | M1 | M0 | MOVX 周期 | $t_{MCS}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 2机器周期 | 0 |
| 0 | 0 | 1 | 3机器周期 | 4 $t_{CLCL}$ |
| 0 | 1 | 0 | 4机器周期s | 8 $t_{CLCL}$ |
| 0 | 1 | 1 | 5机器周期 | 12 $t_{CLCL}$ |
| 1 | 0 | 0 | 6机器周期 | 16 $t_{CLCL}$ |
| 1 | 0 | 1 | 7机器周期s | 20 $t_{CLCL}$ |
| 1 | 1 | 0 | 8机器周期s | 24 $t_{CLCL}$ |
| 1 | 1 | 1 | 9机器周期 | 28 $t_{CLCL}$ |

**对逻辑符号的解释：**

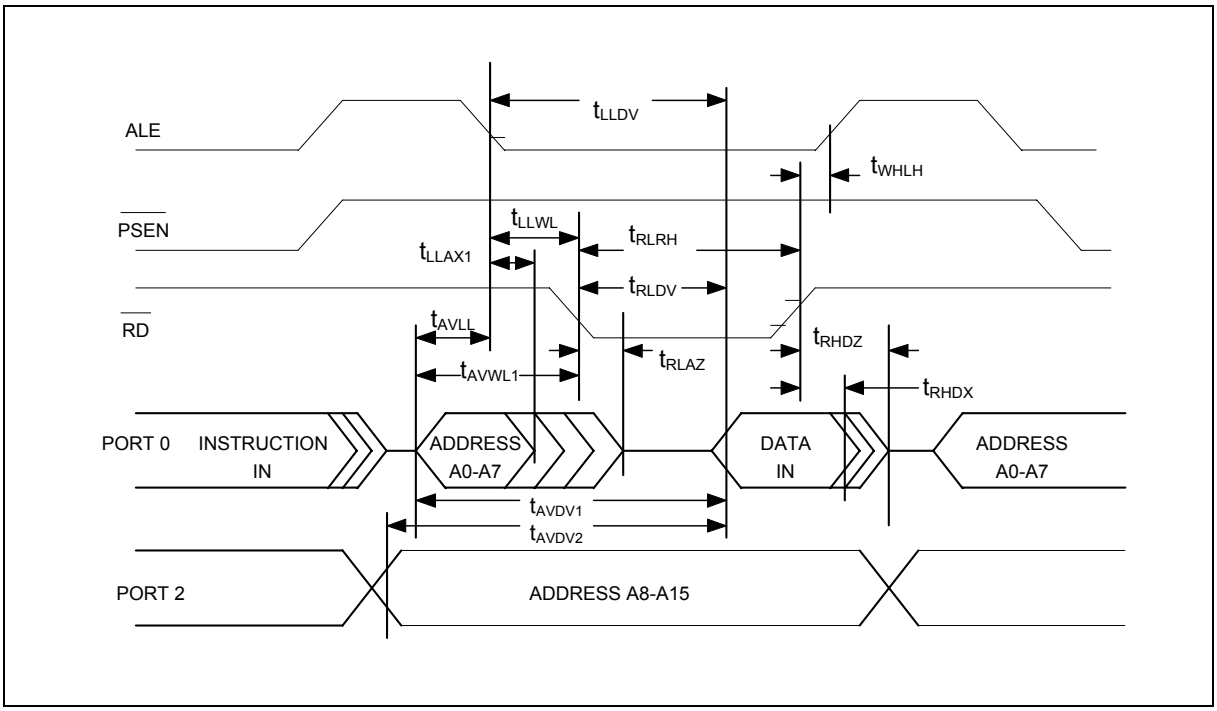
为了保持与8051 体系的兼容性，W77LE532使用与8051 相同的逻辑符号；解释如下：

| | | | |
|---|---|---|---|
| t | 时间 | A | 地址 |
| C | 时钟 | D | 输入数据 |
| H | 逻辑高 | L | 逻辑低 |
| I | 指令 | P | $\overline{PSEN}$ |
| Q | 输出数据 | R | $\overline{RD}$ 信号 |
| V | 有效 | W | $\overline{WR}$ 信号 |
| X | 非有效状态 | Z | 三态 |

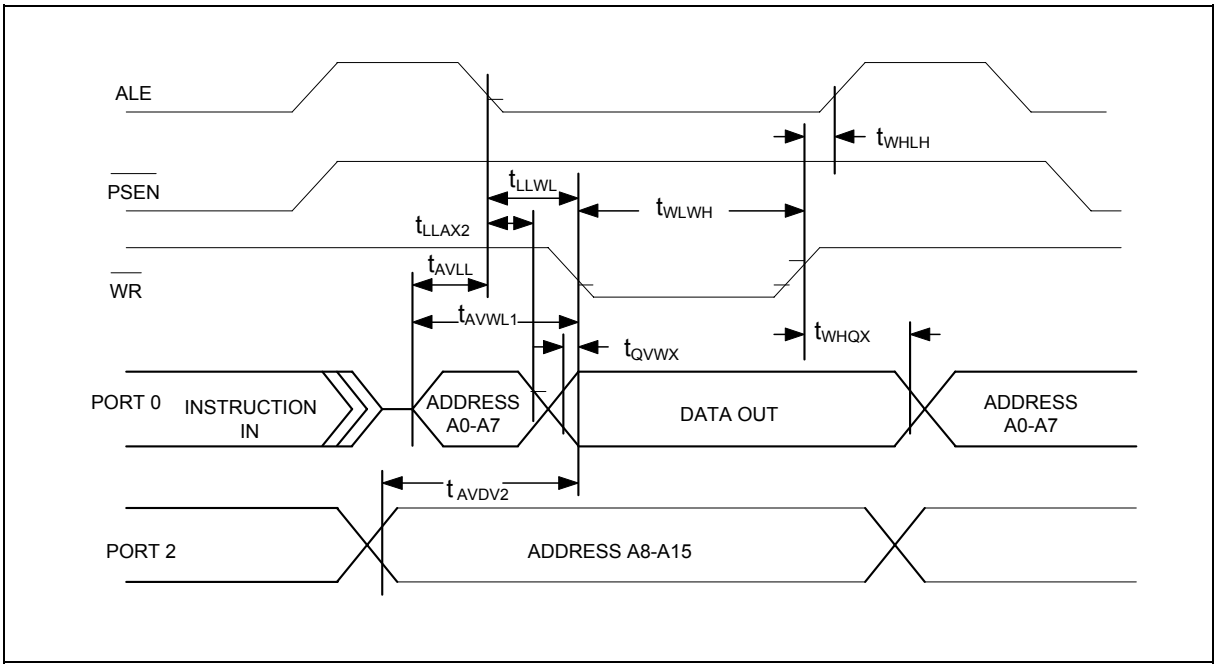
**程序读周期**



## 数据读周期


ALE
PSEN
RD
PORT 0
PORT 2
INSTRUCTION IN
ADDRESS A0-A7
DATA IN
ADDRESS A0-A7
ADDRESS A8-A15
tLLDV
tWHLH
tLLWL
tRLRH
tLLAX1
tRLDV
tAVLL
tRLAZ
tRHDZ
tRHDX
tAVWL1
tAVDV1
tAVDV2

## 数据写周期


ALE
PSEN
WR
PORT 0
PORT 2
INSTRUCTION IN
ADDRESS A0-A7
DATA OUT
ADDRESS A0-A7
ADDRESS A8-A15
tWHLH
tLLWL
tWLWH
tLLAX2
tAVLL
tAVWL1
tQVWX
tWHQX
tAVDV2

## 20. 典型应用电路

### 20.1 扩展外部程序存储器和晶振电路

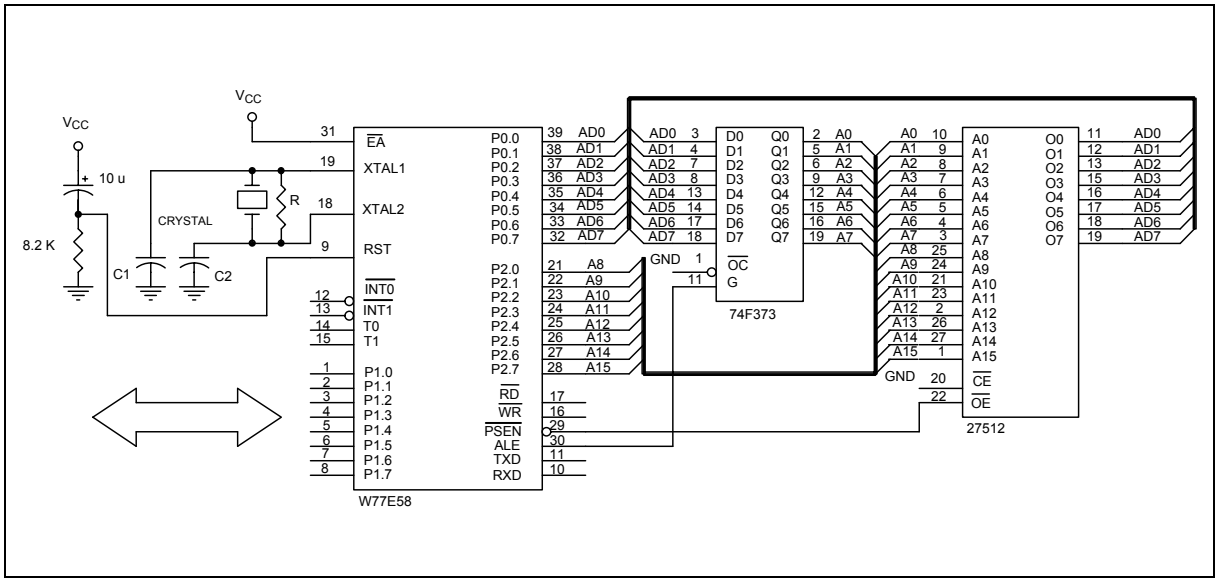



Figure A

| 晶振 | C1 | C2 | R |
|---|---|---|---|
| 16 MHz | 30P | 30P | - |
| 24 MHz | 15P | 15P | - |
| 33 MHz | 10P | 10P | 1.8K |
| 40 MHz | 1P | 1P | 2.4K |

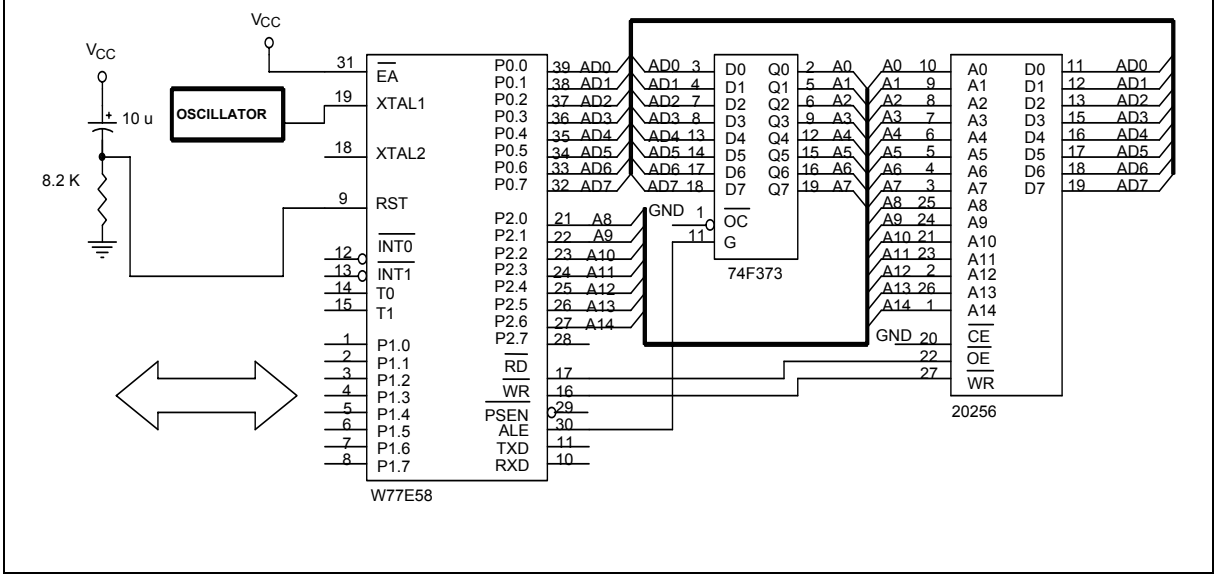
上表列出C1，C2，R的参考值

注：C1, C2, R请参见图A.

.

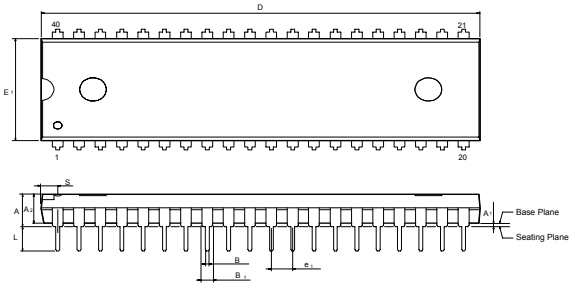
典型应用电路 (续)

## 20.2 外扩数据存储器和晶振




# 21. 封装尺寸

## 21.1 40-pin DIP

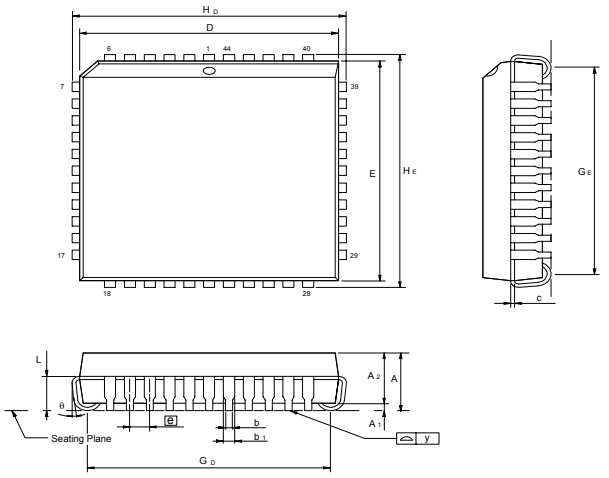



| Symbol | Dimension in inches | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min. | Nom. | Max. | Min. | Nom. | Max. |
| A | — | — | 0.210 | — | — | 5.334 |
| $A_1$ | 0.010 | — | — | 0.254 | — | — |
| $A_2$ | 0.150 | 0.155 | 0.160 | 3.81 | 3.937 | 4.064 |
| B | 0.016 | 0.018 | 0.022 | 0.406 | 0.457 | 0.559 |
| $B_1$ | 0.048 | 0.050 | 0.054 | 1.219 | 1.27 | 1.372 |
| c | 0.008 | 0.010 | 0.014 | 0.203 | 0.254 | 0.356 |
| D | — | 2.055 | 2.070 | — | 52.20 | 52.58 |
| E | 0.590 | 0.600 | 0.610 | 14.986 | 15.24 | 15.494 |
| $E_1$ | 0.540 | 0.545 | 0.550 | 13.72 | 13.84 | 13.97 |
| $e_1$ | 0.090 | 0.100 | 0.110 | 2.286 | 2.54 | 2.794 |
| L | 0.120 | 0.130 | 0.140 | 3.048 | 3.302 | 3.556 |
| a | 0 | — | 15 | 0 | — | 15 |
| $e_A$ | 0.630 | 0.650 | 0.670 | 16.00 | 16.51 | 17.01 |
| S | — | — | 0.090 | — | — | 2.286 |

Notes:
1. Dimension D Max. & S include mold flash or tie bar burrs.
2. Dimension E1 does not include interlead flash.
3. Dimension D & E1 include mold mismatch and are determined at the mold parting line.
4. Dimension B1 does not include dambar protrusion/intrusion.
5. Controlling dimension: Inches.
6. General appearance spec. should be based on final visual inspection spec.

封装尺寸 (续)

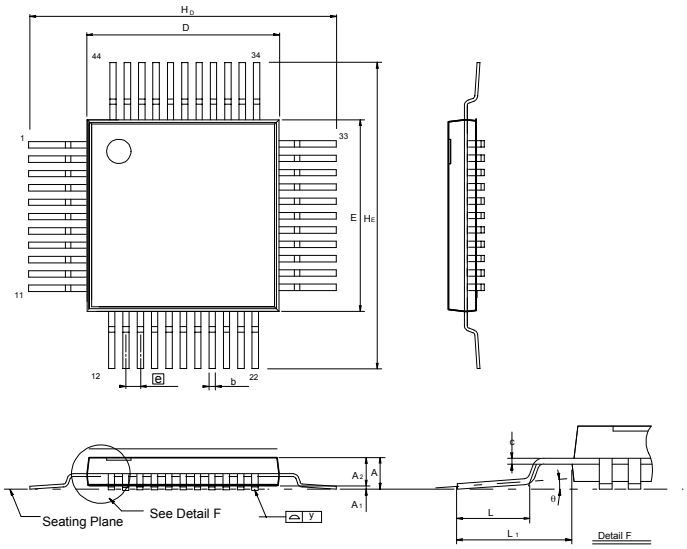
## 21.2 44-pin PLCC




| Symbol | Dimension in inches | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min. | Nom. | Max. | Min. | Nom. | Max. |
| A | — | — | 0.185 | — | — | 4.699 |
| $A_1$ | 0.020 | — | — | 0.508 | — | — |
| $A_2$ | 0.145 | 0.150 | 0.155 | 3.683 | 3.81 | 3.937 |
| $b_1$ | 0.026 | 0.028 | 0.032 | 0.66 | 0.711 | 0.813 |
| b | 0.016 | 0.018 | 0.022 | 0.406 | 0.457 | 0.559 |
| c | 0.008 | 0.010 | 0.014 | 0.203 | 0.254 | 0.356 |
| D | 0.648 | 0.653 | 0.658 | 16.46 | 16.59 | 16.71 |
| E | 0.648 | 0.653 | 0.658 | 16.46 | 16.59 | 16.71 |
| e | 0.050 BSC | | | 1.27 BSC | | |
| $G_D$ | 0.590 | 0.610 | 0.630 | 14.99 | 15.49 | 16.00 |
| $G_E$ | 0.590 | 0.610 | 0.630 | 14.99 | 15.49 | 16.00 |
| $H_D$ | 0.680 | 0.690 | 0.700 | 17.27 | 17.53 | 17.78 |
| $H_E$ | 0.680 | 0.690 | 0.700 | 17.27 | 17.53 | 17.78 |
| L | 0.090 | 0.100 | 0.110 | 2.296 | 2.54 | 2.794 |
| y | — | — | 0.004 | — | — | 0.10 |

Notes:
1. Dimension D & E do not include interlead flash.
2. Dimension b1 does not include dambar protrusion/intrusion.
3. Controlling dimension: Inches
4. General appearance spec. should be based on final visual inspection spec.

## 21.3 44-pin QFP




| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min. | Nom. | Max. | Min. | Nom. | Max. |
| A | --- | --- | --- | --- | --- | --- |
| $A_1$ | 0.002 | 0.01 | 0.02 | 0.05 | 0.25 | 0.5 |
| $A_2$ | 0.075 | 0.081 | 0.087 | 1.90 | 2.05 | 2.20 |
| b | 0.01 | 0.014 | 0.018 | 0.25 | 0.35 | 0.45 |
| c | 0.004 | 0.006 | 0.010 | 0.101 | 0.152 | 0.254 |
| D | 0.390 | 0.394 | 0.398 | 9.9 | 10.00 | 10.1 |
| E | 0.390 | 0.394 | 0.398 | 9.9 | 10.00 | 10.1 |
| e | 0.025 | 0.031 | 0.036 | 0.635 | 0.80 | 0.952 |
| $H_D$ | 0.510 | 0.520 | 0.530 | 12.95 | 13.2 | 13.45 |
| $H_E$ | 0.510 | 0.520 | 0.530 | 12.95 | 13.2 | 13.45 |
| L | 0.025 | 0.031 | 0.037 | 0.65 | 0.8 | 0.95 |
| $L_1$ | 0.051 | 0.063 | 0.075 | 1.295 | 1.6 | 1.905 |
| y | — | — | 0.003 | — | — | 0.08 |
| θ | 0° | — | 7° | 0° | — | 7° |

Notes:
1. Dimension D & E do not include interlead flash.
2. Dimension b does not include dambar protrusion/intrusion.
3. Controlling dimension: Millimeter
4. General appearance spec. should be based on final visual inspection spec.

## 22. 文件版本描述

| 版本 | 日期 | 页 | 描 述 |
|---|---|---|---|
| SC1 | 02/18/2005 | - | 初次发行 |

**Headquarters**
No. 4, Creation Rd. III,
Science-Based Industrial Park,
Hsinchu, Taiwan
TEL: 886-3-5770066
FAX: 886-3-5665577
http://www.winbond.com.tw/

**Taipei Office**
9F, No.480, Rueiguang Rd.,
Neihu District, Taipei, 114,
Taiwan, R.O.C.
TEL: 886-2-8177-7168
FAX: 886-2-8751-3579

**Winbond Electronics Corporation America**
2727 North First Street, San Jose,
CA 95134, U.S.A.
TEL: 1-408-9436666
FAX: 1-408-5441798

**Winbond Electronics Corporation Japan**
7F Daini-ueno BLDG, 3-7-18
Shinyokohama Kohoku-ku,
Yokohama, 222-0033
TEL: 81-45-4781881
FAX: 81-45-4781800

**Winbond Electronics (Shanghai) Ltd.**
27F, 2299 Yan An W. Rd. Shanghai,
200336 China
TEL: 86-21-62365999
FAX: 86-21-62365998

**Winbond Electronics (H.K.) Ltd.**
Unit 9-15, 22F, Millennium City,
No. 378 Kwun Tong Rd.,
Kowloon, Hong Kong
TEL: 852-27513100
FAX: 852-27552064