

第8章 89C51单片机扩展存储器的设计

8.1 系统扩展结构

89C51系统扩展结构如图8-1所示。

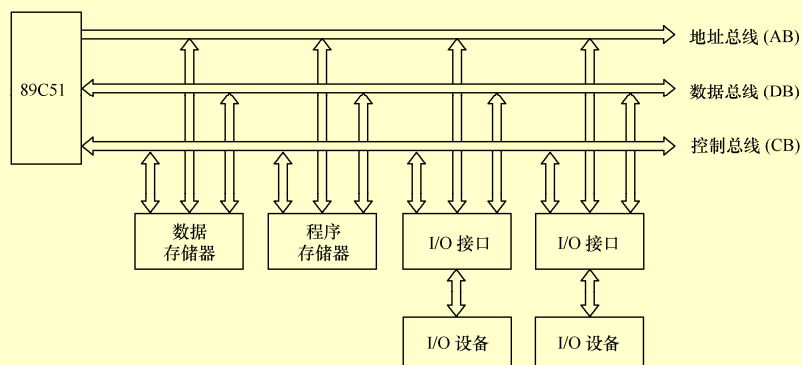


图8-1

8.2 系统总线及总线构造

按功能通常把系统总线分为三组，如图8-1所示。

1. 地址总线（Address Bus, AB）

地址总线用于传送单片机发出的地址信号，以便进行存储单元和I/O接口芯片中的寄存器选择。地址总线是单向传输的。

2. 数据总线（Data Bus, DB）

数据总线用于在单片机与存储器之间或与I/O端口之间传送数据。数据总线是双向的，可以进行两个方向的传送。

3. 控制总线（Control Bus, CB）

控制总线实际上就是单片机发出的各种控制信号线。

8.2.1 构造系统总线

系统扩展的首要问题：构造系统总线。

系统总线上“挂”存储器芯片或I/O接口芯片，“挂”存储器芯片就是存储器扩展，“挂”I/O接口芯片就是I/O扩展。

1. 以P0口作为低8位地址/数据总线

89C51由于受引脚数目的限制，数据线和低8位地址线复用。

为了将它们分离出来，需要外加地址锁存器，从而构成与一般CPU相类似的片外三总线，见图8-2。

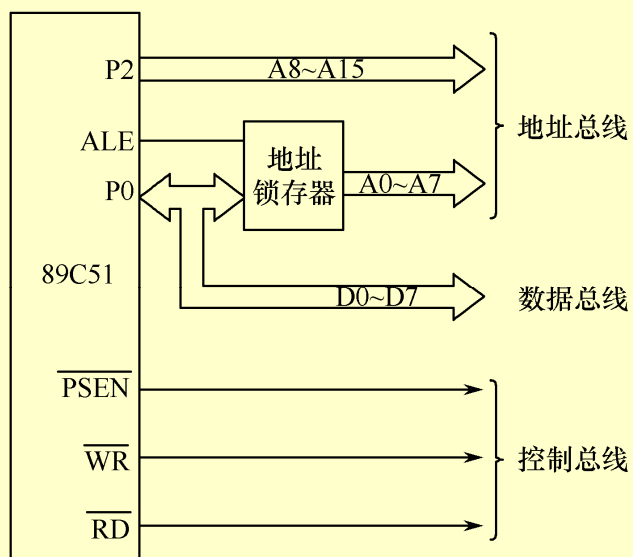
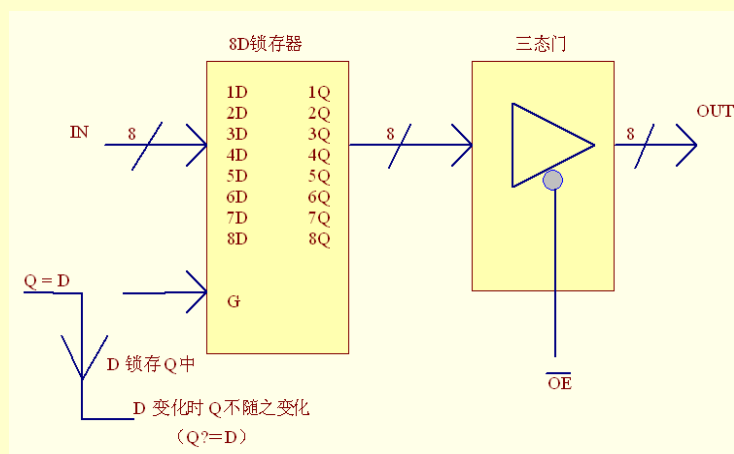
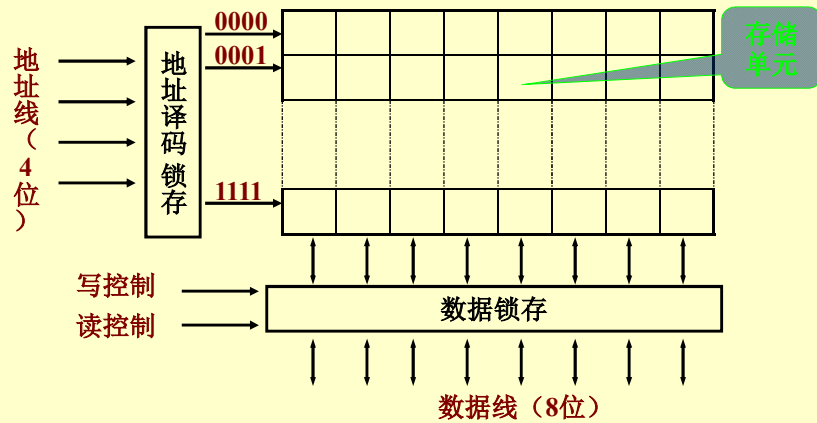


图8-2

外部存储器及其地址范围

- 存储器示意：其地址范围是0000B~1111B（0H~FH）



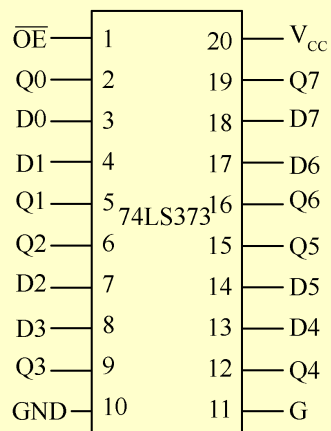


图8-7

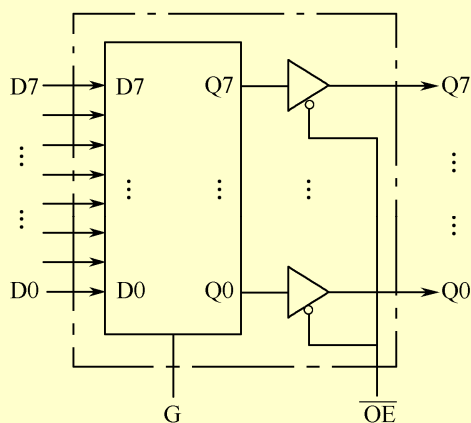


图8-8

引脚说明如下：

D7~D0：8位数据输入线。

Q7~Q0：8位数据输出线。

G：数据输入锁存选通信号，

OE*：数据输出允许信号

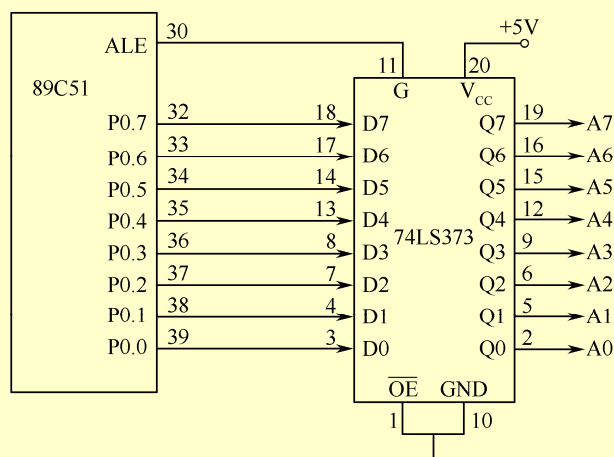


图8-9

74LS373功能如表8-3所示。

表8-3 74LS373功能表

OE*	G	D	Q
0	1	1	1
0	1	0	0
0	0	×	不变
1	×	×	高阻态

8.3 程序存储器EPROM的扩展

采用只读存储器，非易失性。

(1) 掩膜ROM

在制造过程中编程。成本较高，因此只适合于大批量生产。

(2) 可编程ROM (PROM)

用独立的编程器写入。但PROM只能写入一次，且不能再修改。

(3) EPROM

电信号编程，紫外线擦除的只读存储器芯片。

(4) E²PROM (EEPROM)

电信号编程，电信号擦除的ROM芯片。读写操作与RAM几乎没有什么差别，只是写入的速度慢一些。但断电后能够保存信息。

(5) Flash ROM

又称闪烁存储器，简称闪存。大有取代E²PROM的趋势。

目前许多公司生产的以8051为内核的单片机，在芯片内部大多集成了数量不等的Flash ROM。

例如，美国ATMEL公司生产的与51系列单片机兼容的产品89C2051/89C51/89C52/89C55，片内分别有2KB/4KB/8KB/20KB的Flash ROM，来作为EPROM使用。

对于这类单片机，在片内的Flash ROM满足要求的情况下，扩展外部程序存储器的工作就可省去。

8.3.1 常用EPROM芯片介绍

典型芯片是27系列产品，例如，2764（8KB×8）、27128（16KB×8）、27256（32KB×8）、27512（64KB×8）。

“27”后面的数字表示其位存储容量。

随着大规模集成电路技术的发展，大容量存储器芯片的产量剧增，售价不断下降，其性价比明显增高，而且由于有些厂家已停止生产小容量的芯片，使市场上某些小容量芯片的价格反而比大容量芯片还贵。

所以，在扩展程序存储器设计时，应尽量采用大容量芯片。

1. 常用的EPROM芯片

27系列EPROM芯片的引脚如图8-11所示，参数见表8-4（P143，略）。

图8-11中的引脚功能如下：

A0~A15：地址线引脚。数目决定存储容量来定，用来进行单元选择。

D7~D0：数据线引脚

CE*：片选输入端

OE*：输出允许控制端

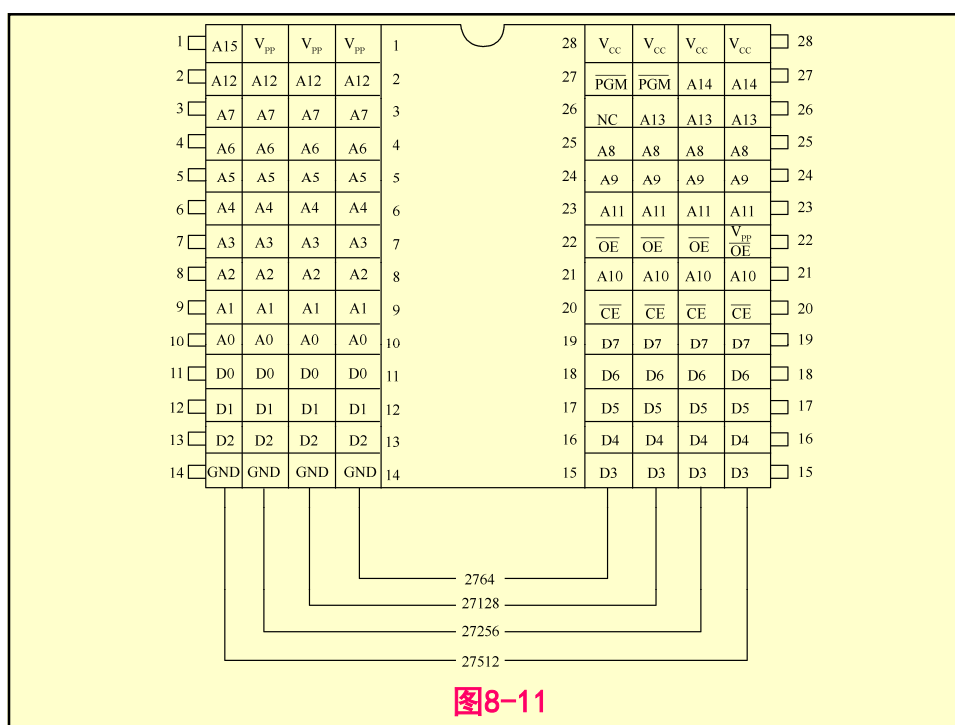
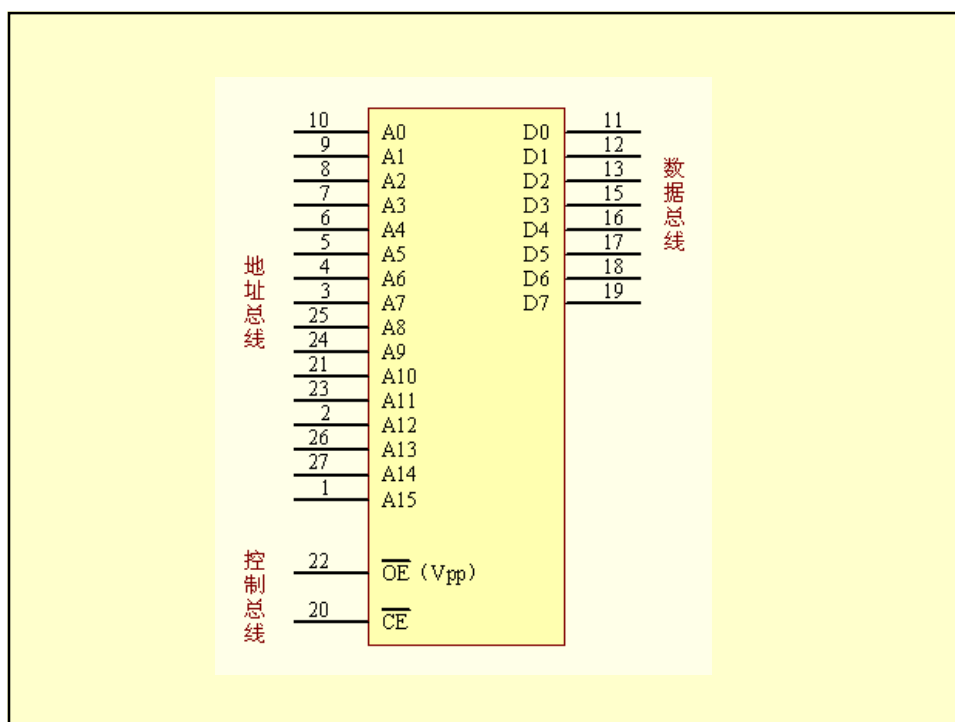
PGM*：编程时，加编程脉冲的输入端

Vpp：编程时，编程电压（+12V或+25V）输入端

Vcc：+5V，芯片的工作电压。

GND：数字地。

NC：无用端



2. EPROM芯片的工作方式

5种工作方式如表8-5所示。

(1) 读出方式

片选控制线为低, 同时输出允许控制线为低, V_{pp} 为+5V, 指定地址单元的内容从D7~D0上读出。

(2) 未选中方式

片选控制线为高电平。

(3) 编程方式

V_{pp} 端加上规定高压, CE^* 和 OE^* 端加合适电平(不同的芯片要求不同), 就能将数据线上的数据写入到指定的地址单元。

(4) 编程校验方式

(5) 编程禁止方式

输出呈高阻状态, 不写入程序。

8.3.2 程序存储器的操作时序

1. 访问程序存储器的控制信号

(1) ALE

(2) $PSEN^*$

(3) EA^*

如果指令是从片外EPROM中读取, ALE用于低8位地址锁存, $PSEN^*$ 接外扩EPROM的 OE^* 脚。

P0口: 分时低8位地址总线 and 数据总线, **P2口**: 高8位地址线。

2. 操作时序

(1) 应用系统中无片外RAM

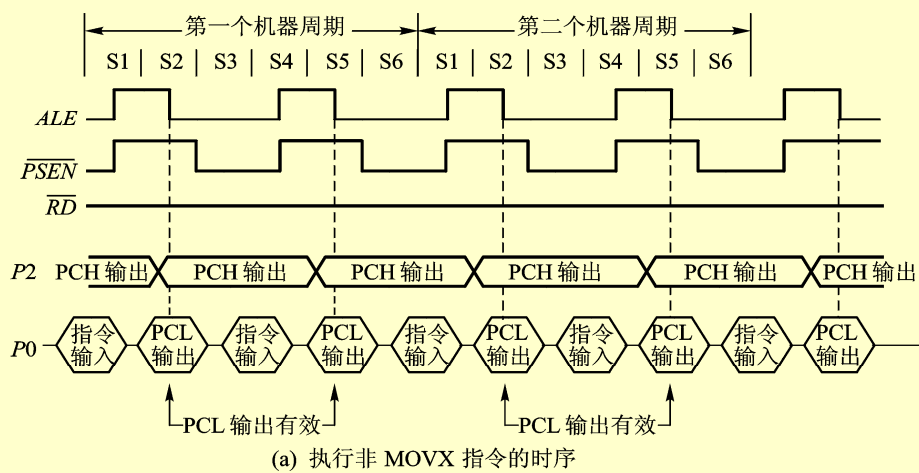


图8-12 (a)

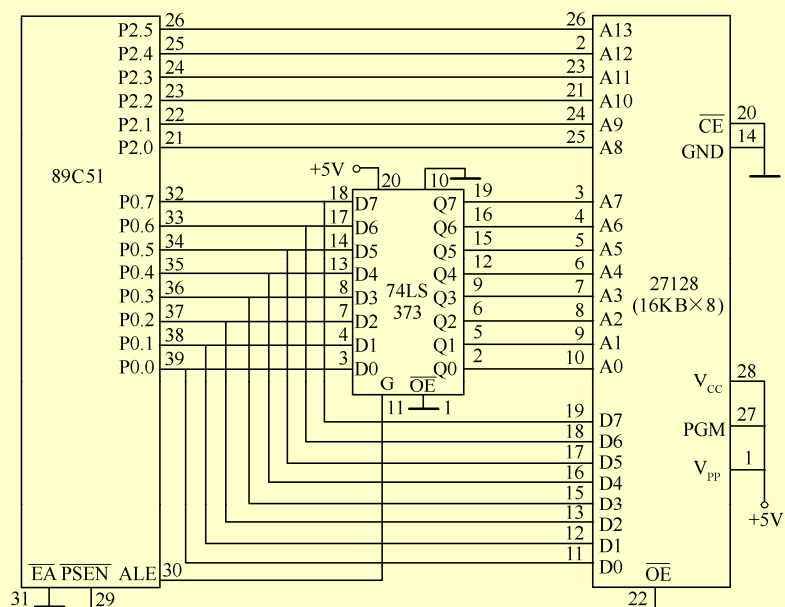


图8-13

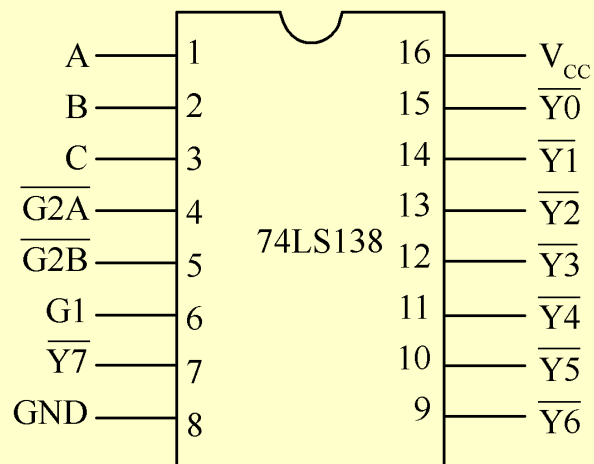


图8-3

表8-1 74LS138译码器真值表

输 入						输 出							
G1	G2A*	G2B*	C	B	A	Y7*	Y6*	Y5*	Y4*	Y3*	Y2*	Y1*	Y0*
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	0	1	1
1	0	0	0	1	0	1	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	0	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1
其它状态			×	×	×	1	1	1	1	1	1	1	1

(2) 74LS139

74LS139是**双2-4译码器**。两个译码器完全独立，分别有各自的数据输入端、译码状态输出端以及数据输入允许端。其引脚如**图8-4**所示，真值表如**表8-2**所示（见P138）。

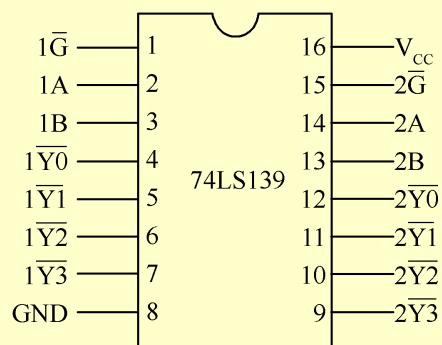


图8-4

下面以74LS138为例，介绍如何进行地址分配。

例 要扩8片8KB的RAM 6264，如何通过74LS138把64KB空间分配给各个芯片？

64KB地址空间的分配如**图8-5**所示。

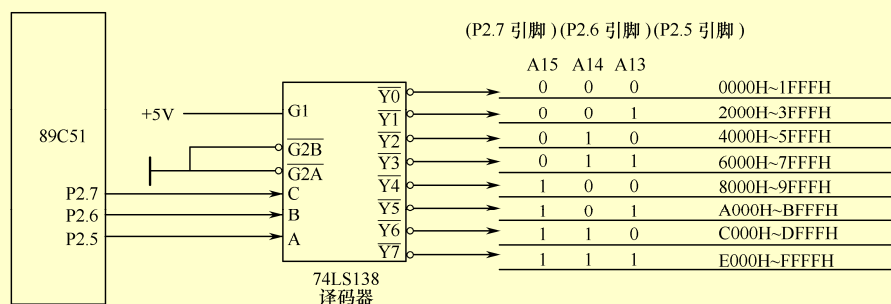


图8-5

采用**全地址译码方式**，单片机发地址码时，每次只能选中一个存储单元。同类存储器间不会产生地址重叠的问题。

如果用**74LS138**把**64K**空间全部划分为每块**4KB**，如何划分呢？

由于**4KB**空间需要**12**条地址线进行“**单元选择**”，而译码器的输入有**3**条地址线（**P2.6**~**P2.4**），**P2.7**没有参加译码，**P2.7**发出的**0**或**1**决定了选择**64KB**存储器空间的前**32KB**还是后**32KB**，由于**P2.7**没有参加译码，就**不是全译码方式**，这样前后两个**32KB**空间就重叠了。

那么，这**32KB**空间利用**74LS138**译码器可划分为**8**个**4KB**空间。

如果把**P2.7**通过一个非门与**74LS138**译码器的**G1**端连接起来，如**图8-6**所示，就不会发生两个**32KB**空间重叠的问题了。

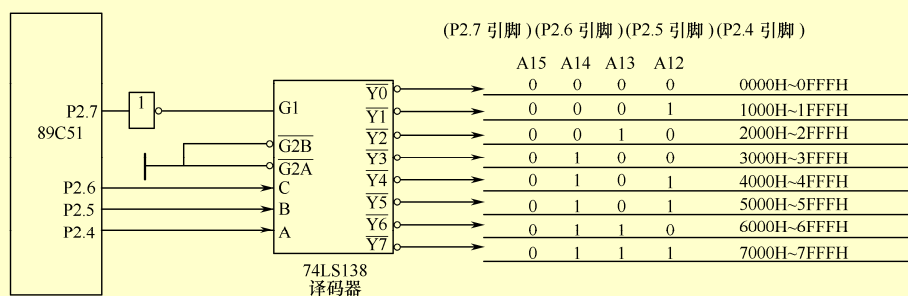


图8-6

8.2.2 外部地址锁存器

地址锁存器芯片：74LS373、8282、74LS573等。

1. 锁存器74LS373

带有三态门的8D锁存器，其引脚及内部结构如**图8-7**和**图8-8**。

89C51与74LS373的连接如**图8-9**所示。

3. 使用多片EPROM的扩展电路

89C51扩展4片27128。

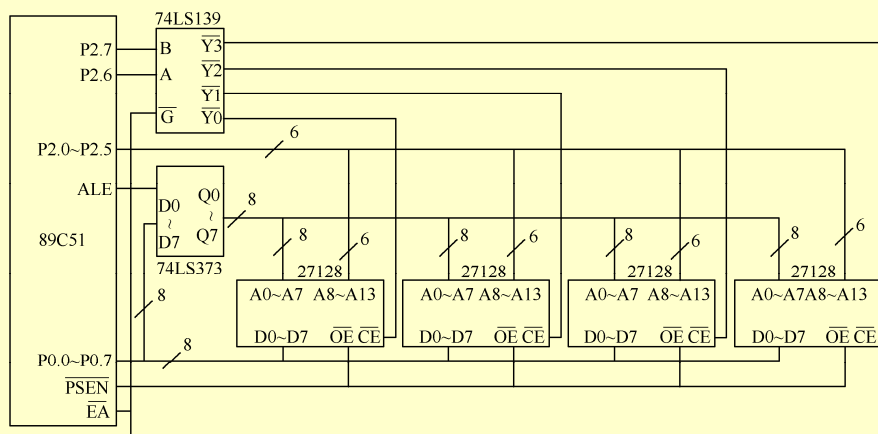


图8-14

图8-14中的片选控制信号由译码器产生。4片27128各自所占的地址空间，请读者自己分析。

8.4 静态数据存储器的扩展

在单片机应用系统中，外扩的数据存储器都采用静态数据存储器（SRAM），所以只讨论SRAM与89C51的接口。

所扩展的数据存储器空间地址由P2口提供高8位地址，P0口分时提供低8位地址和8位双向数据总线。片外数据存储器RAM的读和写由89C51的RD*（P3.7）和WR*（P3.6）信号控制，而片外程序存储器EPROM的输出允许端（OE*）由89C51的程序存储器读选通信号PSEN*控制。

尽管与EPROM的地址空间范围都是相同的，但由于控制信号不同，故不会发生总线冲突。

8.4.1 常用的静态RAM（SRAM）芯片

典型型号有:6116、6264、62128、62256。+5V电源供电，双列直插封装，6116为24引脚封装，6264、62128、62256为28引脚封装，引脚如图8-15。

各引脚功能：

A0~A14：地址输入线。

D0~D7：双向三态数据线。

CE*：片选信号输入。对于6264芯片，当26脚(CS)为高电平时,且CE*为低电平时才选中该片。

OE*：读选通信号输入线。

WE*：写允许信号输入线，低电平有效。

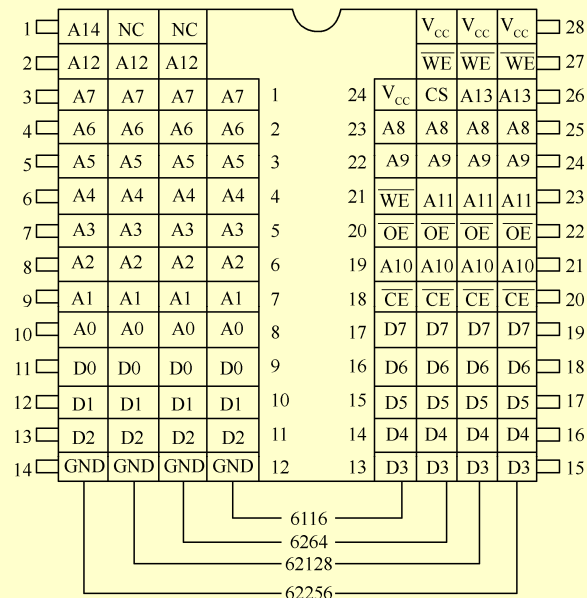


图8-15

Vcc: 工作电源+5V

GND: 地

工作方式有**读出**、**写入**、**维持**三种，这些工作方式的操作控制如表8-6 (P148)。

8.4.2 外扩数据存储器的读写操作时序

1. 读片外RAM操作时序

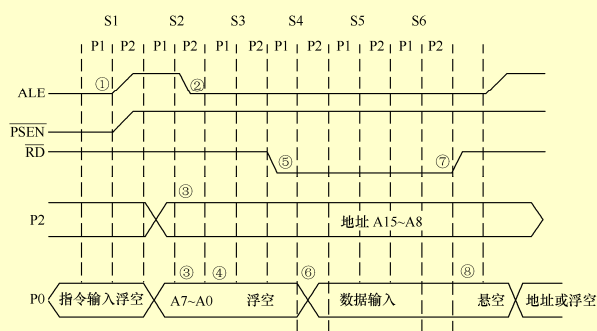


图8-16

2. 写片外RAM操作时序

写是CPU主动把数据送上P0口总线。故在时序上，CPU先向P0口总线上送完8位地址后，在S3状态就将数据送到P0口总线。

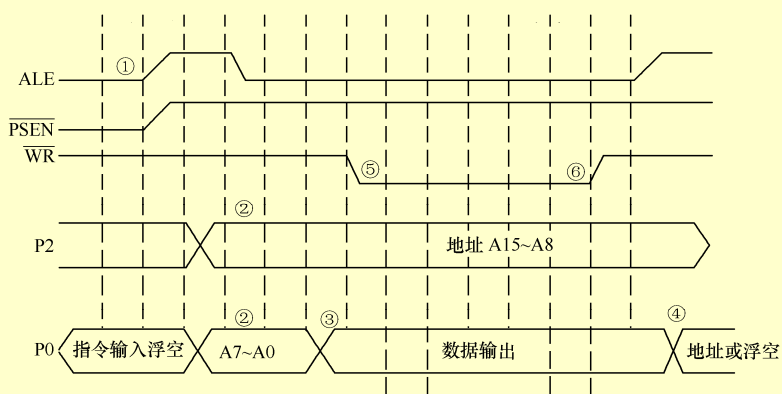


图8-17

8.4.3 89C51与RAM的接口电路设计

图8-18为线选法扩展外部数据存储器的电路。

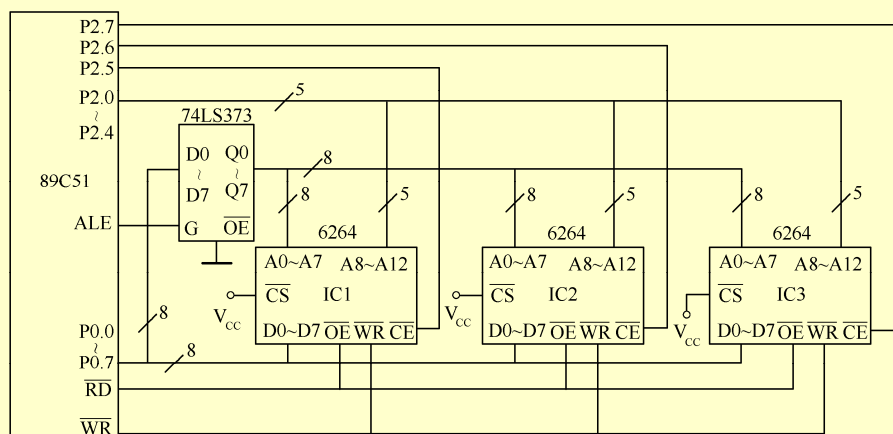


图8-18

地址线为A0~A12，故8031剩余地址线为三根。用线选法可扩展3片6264。3片6264对应的存储器空间如表8-7。

表8-7

P2.7	P2.6	P2.5	选中芯片	地址范围	存储容量
1	1	0	IC1	C000H~DFFFH	8K
1	0	1	IC2	A000H~BFFFH	8K
0	1	1	IC3	6000H~7FFFH	8K

译码选通法扩展,如图8-19所示。

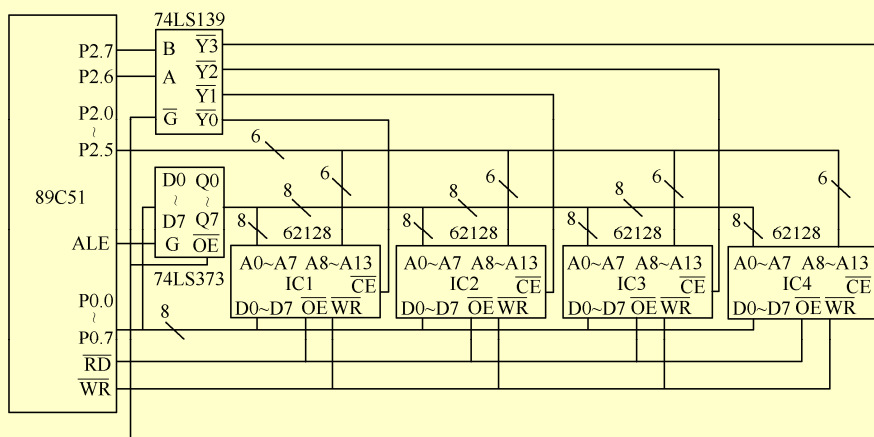


图8-19

各62128芯片的地址分配见表8-9。

表8-9 各片62128地址分配

P2.6	P2.7	译码输出	选中芯片	地址范围	存储容量
0	0	Y0*	IC1	0000H-3FFFH	16K
0	1	Y1*	IC2	4000H-7FFFH	16K
1	0	Y2*	IC3	8000H-BFFFH	16K
1	1	Y3*	IC4	C000H-FFFFH	16K

例8-1 编写程序将片外数据存储器中5000H~50FFH单元全部清零

方法1:

用DPTR作为数据区地址指针，同时**使用字节计数器**。

```

MOV DPTR, #5000H      ; 设置数据块指针的初值
MOV R7, #00H          ; 设置块长度计数器初值
CLR    A
LOOP:  MOVX @DPTR, A    ; 把某一单元清零
      INC  DPTR        ; 地址指针加1
      DJNZ R7, LOOP    ; 数据块长度减1，若不为
                        ; 0则继续清零
      HERE: SJMP HERE  ; 执行完毕，原地踏步

```

方法2:

用DPTR作为数据区地址指针，但不使用字节计数器，而是**比较特征地址**。

```

MOV    DPTR, #5000H
CLR    A
LOOP:  MOVX @DPTR, A
      INC  DPTR
      MOV  R7, DPL
      CJNE R7, #0, LOOP ; 与末地址+1比较
      HERE: SJMP  HERE

```

8.6 EPROM和RAM的综合扩展

8.6.1 综合扩展的硬件接口电路

例8-2 采用**线选法**扩展2片8KB的RAM和2片8KB的EPROM。RAM选6264，EPROM选2764。扩展接口电路见**图8-20**。

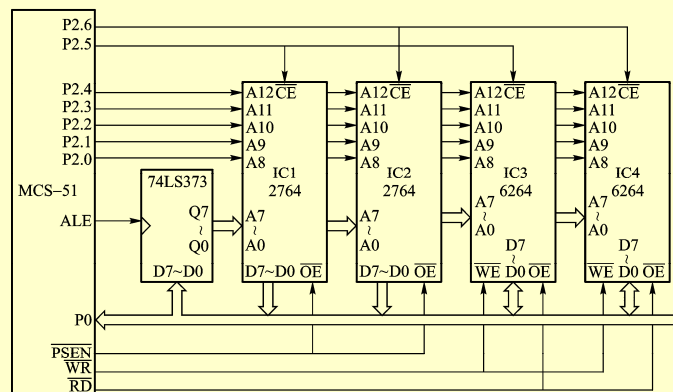


图8-20

(1) 控制信号及片选信号

IC2和IC4占用地址空间为2000H~3FFFH共8KB。同理IC1、IC3地址范围4000H~5FFFH (P2.6=1、P2.5=0、P2.7=0)。线选法地址不连续，地址空间利用不充分。

(2) 各芯片地址空间分配

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
1	0	1	×	×	×	×	×	×	×	×	×	×	×	×	×

IC2和IC4占用的地址空间为A000H~BFFFH共8KB。

同理IC1、IC3的地址范围为C000H~DFFFH。4片存储器各自所占的地址空间如表8-9所示。

例8-3 采用译码器法扩展2片8KB EPROM，2片8KB RAM。EPROM选用2764，RAM选用6264。共扩展4片芯片。扩展接口电路见图8-21。

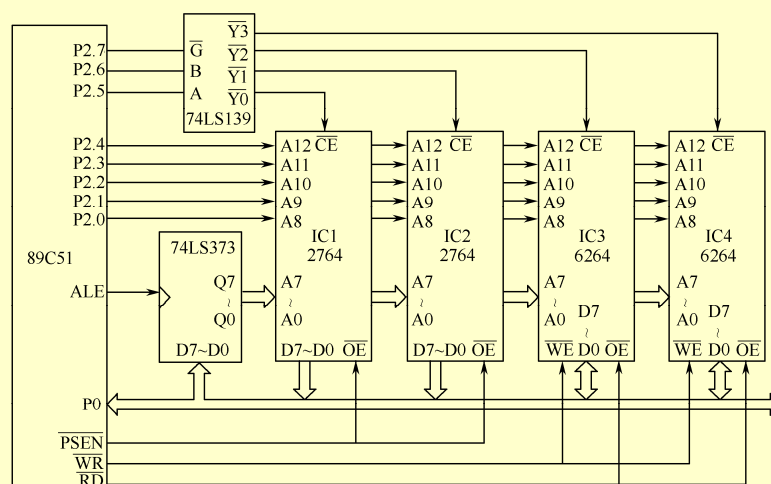


图8-21

各存储器地址范围如下：

表8-9

芯片	地址范围
IC4	6000H~7FFFH
IC3	4000H~5FFFH
IC2	2000H~3FFFH
IC1	0000H~1FFFH

可见译码法进行地址分配，各芯片地址空间是连续的。

8.5.2 外扩存储器电路的工作原理及软件设计

1. 单片机片外程序区读指令过程

2. 单片机片外数据区读/写数据过程

例如，把片外6000H单元的数据送到片内RAM 50H单元，程序如下：

```
MOV    DPTR, #6000H
```

```
MOVX   A, @DPTR
```

```
MOV    50H, A
```

例如，把片内50H单元的数据送到片外4000H单元中，程序如下：

```
MOV    A, 50H
```

```
MOV    DPTR, #4000H
```

```
MOVX   @DPTR, A
```

89C51单片机读写片外数据存储器中的内容，除用MOVX

A, @DPTR和MOVX @DPTR, A外，还可使用MOVX A, @Ri和MOVX @Ri, A。这时通过P0口输出Ri中的内容（低8位地址），而把P2口原有的内容作为高8位地址输出。

例8-4 将程序存储器中以TAB为首址的32个单元的内容依次传送到外部RAM以7000H为首地址的区域去。

DPTR指向标号TAB的首地址。**R0**既指示外部RAM的地址，又表示数据标号TAB的位移量。本程序的循环次数为32，R0的值：0~31，R0的值达到32就结束循环。程序如下：

```

                MOV    P2, #70H
                MOV    DPTR, #TAB
                MOV    R0, #0
AGIN:          MOV    A, R0
                MOVC   A, @A+DPTR
                MOVX   @R0, A
                INC    R0
                CJNE   R0, #32, AGIN
HERE:          SJMP   HERE
TAB:           DB     .....

```

8.6 AT89C51/89C55单片机的片内闪烁存储器

AT89C51/89C52/89C55是低功耗、高性能的片内含有4KB/8KB/20KB闪烁可编程/擦除只读存储器，芯片内的闪存允许在线编程或采用通用的编程器对其重复编程。

8.6.1 89C51的性能及片内闪烁存储器

1. 89C51的主要性能

- (1) 与MCS-51微控制器系列产品兼容。
- (2) 片内有4KB可在线重复编程的闪烁存储器
(Flash Memory)
- (3) 存储器可循环写入/擦除10000次。
- (4) 存储器数据保存时间为10年。
- (5) 宽工作电压范围：Vcc可为+2.7~6V。

- (6) 全静态工作：可从0Hz~16MHz。
- (7) 程序存储器具有3级加密保护。
- (8) 空闲状态维持低功耗和掉电状态保存存储器内容。

2. 片内闪烁存储器(Flash Memory)

目前，美国ATMEL公司生产的带有片内Flash存储器的AT89C51/89C52/89C55单片机，由于价格便宜，且与MCS-51系列单片机兼容，受到了我国广大工程技术人员的欢迎。使用该系列单片机，可省去外扩程序存储器的工作，设计者只需了解片内闪烁存储器的特性以及如何对其编程。

8.6.2 片内闪烁存储器的编程

89C51的I/O口P0、P1、P2和P3除具有与8031相同的一些性能和用途外，在对Flash编程时，P0口还可接收代码字节，但在程序校验时需要外加上拉负载电阻。

在Flash编程和程序校验期间，P1口接收低位地址字节，P2口接收高位地址位和一些控制信号，P3口也接收Flash编程和校验用的控制信号。此时，ALE/PROG*引脚是编程脉冲输入（PROG*）端。

该芯片内有三个加密位，其状态可以是编程（P）或不编程（U），各状态提供的功能见表8-11。

如果加密位LB1被编程，则EA*脚的电平在复位时被采样并锁存。

若器件在加电时不进行复位，那么该锁存器初始化为一个随机值，并在复位有效前始终保持该值。

为使器件工作正常，EA*的锁存值必须与引脚的当前逻辑电平一致。

89C51的三个加密位可以不被编程（U）或被编程（P），以获得表8-11所示的特性。

表8-11

类型	程序加密位			保护功能
	LB1	LB2	LB3	
1	U	U	U	无程序加密特性
2	P	U	U	可对外部程序存储器执行MOVX指令，不允许从内部存储器取代码字节。在幅为脉冲期间 \overline{OE} 被采样并锁存。禁止EEPROM的进一步编程。
3	P	P	U	与类型2相同，同时定时校验。
4	P	P	P	与类型3相同，同时外部执行被禁止。

对89C51片内的Flash存储器编程，设计者只需在市场上购买相应的编程器。一种是在购买单片机仿真开发系统时就带有编程器，另一种是单独购买编程器。

开发者可按照编程器的说明进行操作，如想对写入的内容加密，只需按照编程器的菜单，选择加密功能选项即可。

