存储器扩展

信息工程学院　陈梅

实验举例

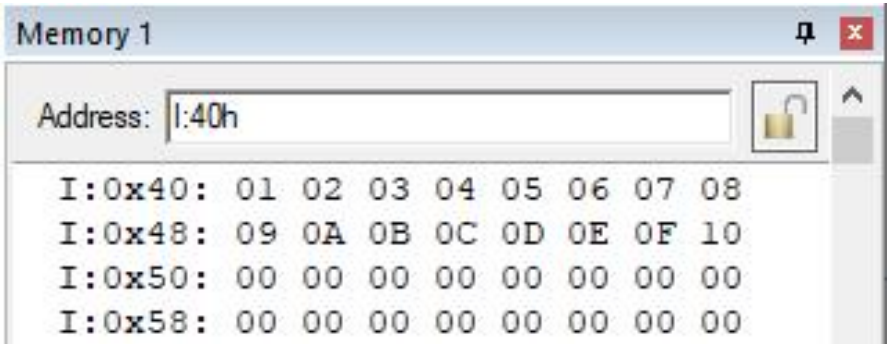
【例3- 24&3-31】设单片机片内数据存储器首地址为40H，片外数据存储器首地址为4000H，存取数据字节个数16个。先将片内数据存储区的这16个字节的内容设置为01H~10H，再将片内首地址为30H开始的16个单元的内容传送到片外首地址为4000H开始的数据存储区中保存。

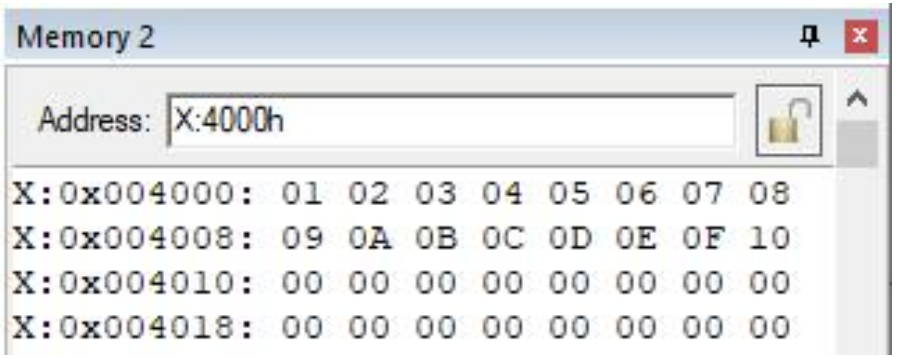
要求：汇编语言实验或C51编程使用绝对地址访问的三种方式之一实现。

①使用指针　　②使用C51运行库中预定义宏　　③使用C51扩展关键字_at_

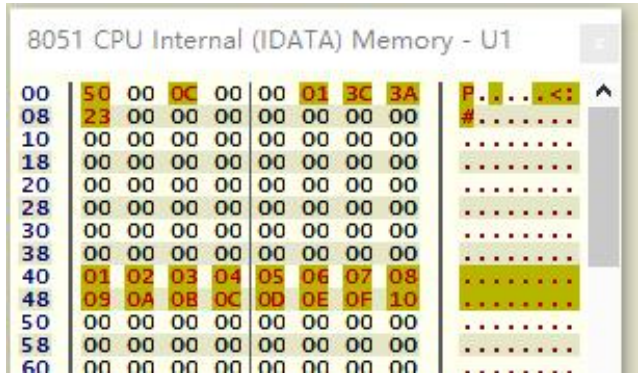
不同存储空间的数据的查看在Memory窗口输入相应的字母和数字：

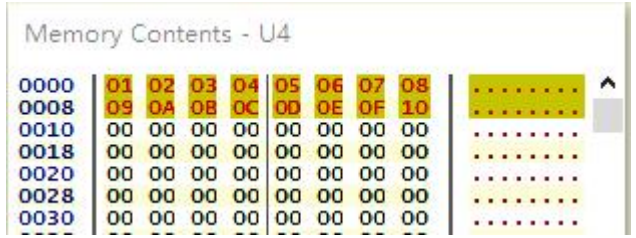
查看片内数据存储单元，输入：i:0x40 d:0x40或i:40H d:40H
查看片外数据存储单元，输入：x:0x4000 或x:4000H
查看程序存储单元，输入：c:0x0000 或c:0000H

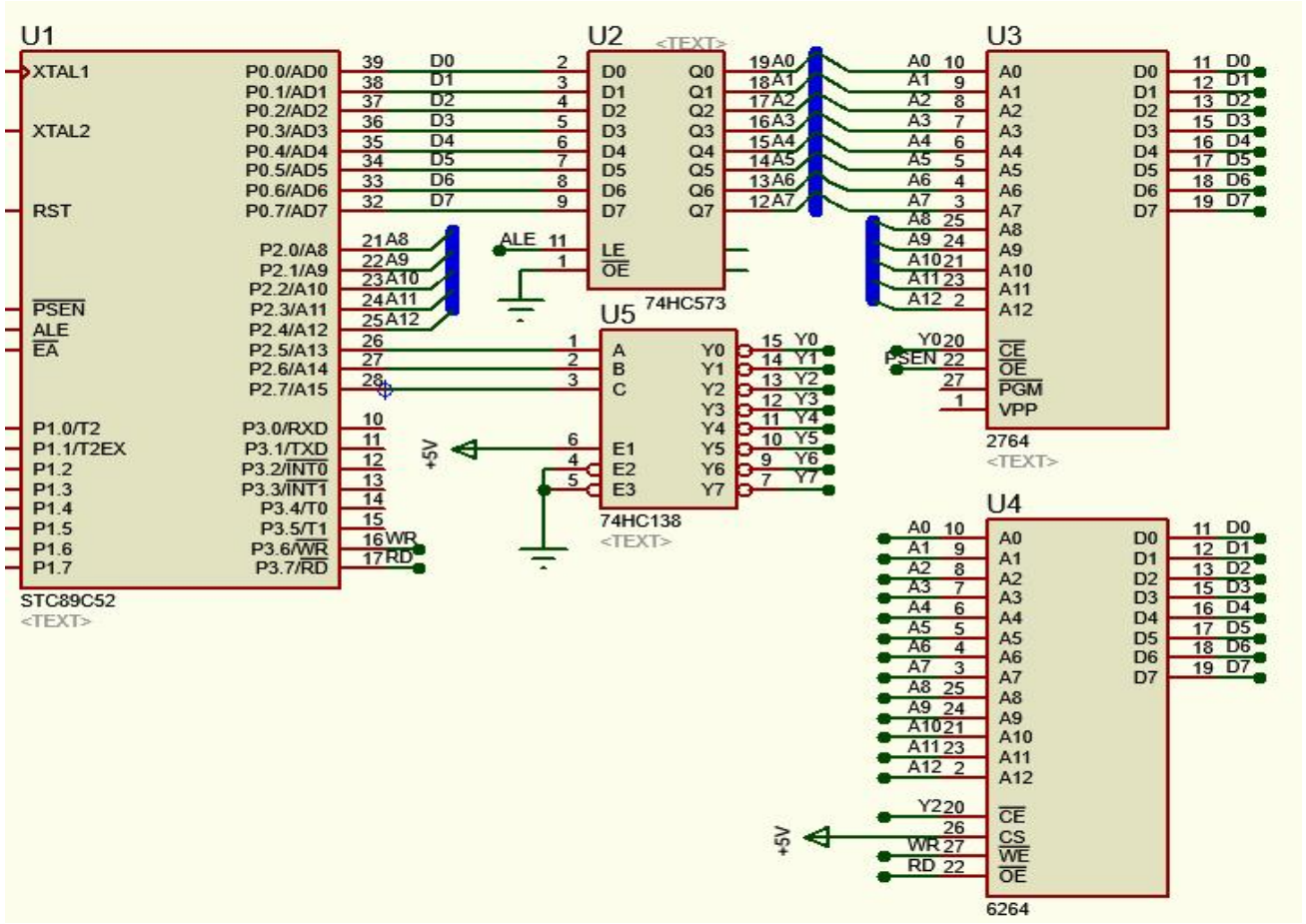




【例3- 24&3-31】设单片机片内数据存储器首地址为40H，片外数据存储器首地址为4000H，存取数据字节个数16个。先将片内数据存储区的这16个字节的内容设置为01H~10H，再将片内首地址为30H开始的16个单元的内容传送到片外首地址为4000H开始的数据存储区中保存。





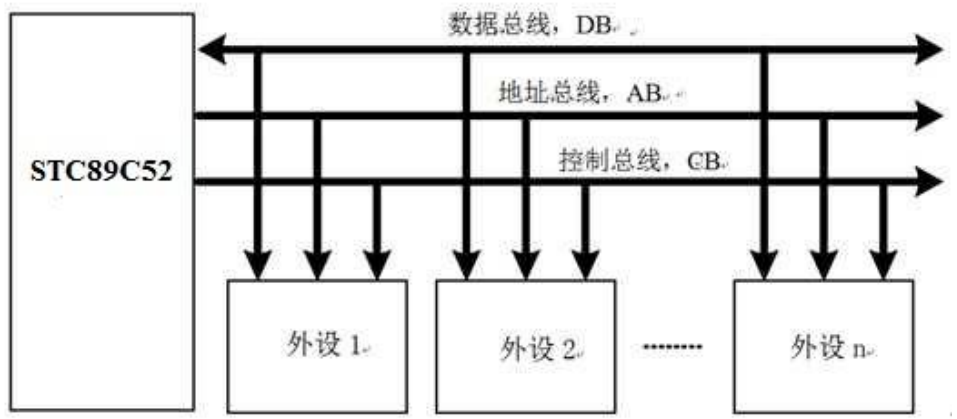


STC89C52存储器扩展 要点

要点:

- 系统扩展结构
- 地址锁存与地址空间分配
- 程序存储器的扩展
- 数据存储器的扩展
- EPROM和RAM的综合扩展

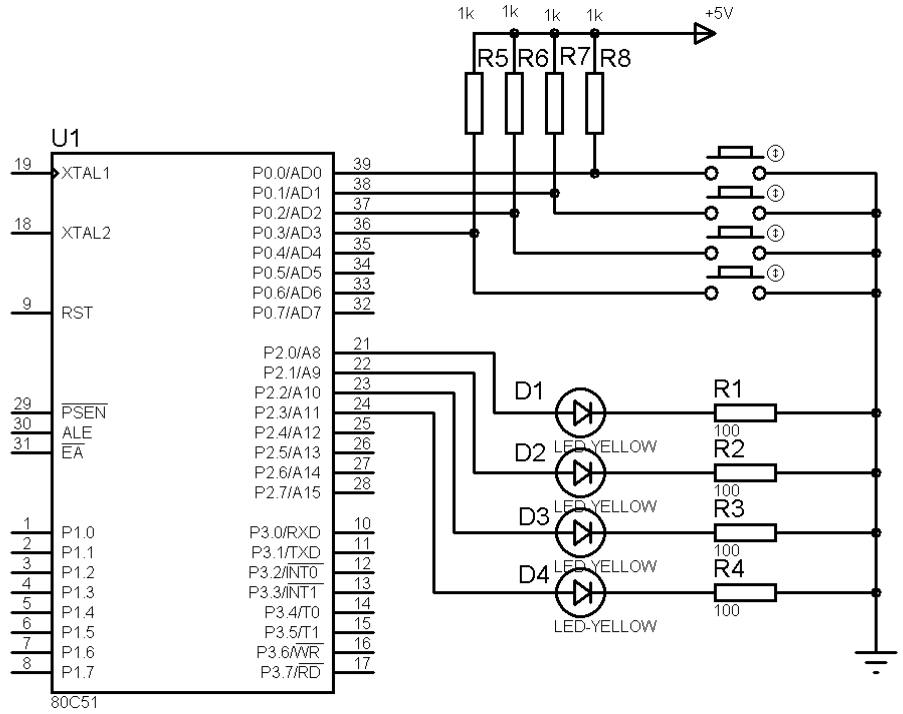
STC89C52单片机与外部设备的两种连接方式



总线方式

采用片外RAM指令访问外设

例如：**MOVX A, @DPTR**



非总线方式

采用片内RAM指令访问外设

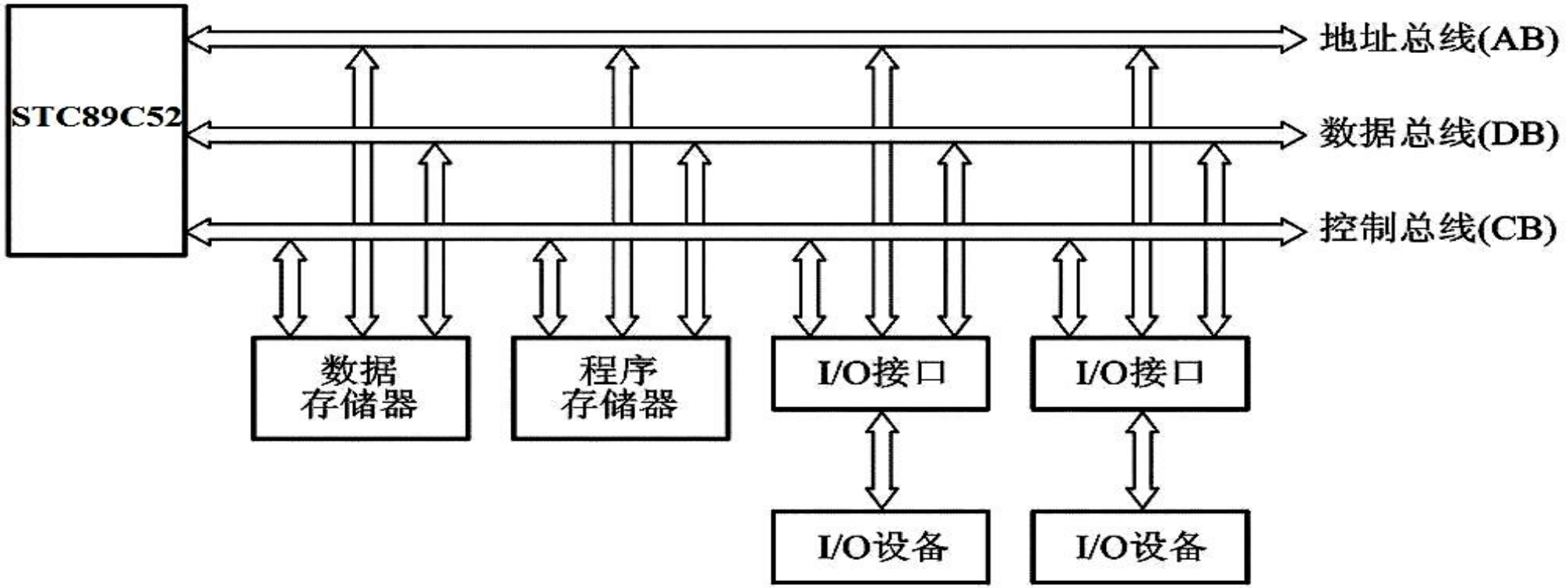
例如：**MOV A, P0**

系统扩展主要包括：

- **存储器扩展：当单片机存储空间不够时，进行程序或数据存储扩展。**
- **I/O接口扩展：当单片机与外设信息的类型或传送速度不匹配时，进行I/O接口的扩展。**

系统扩展首先要构造**系统总线**。

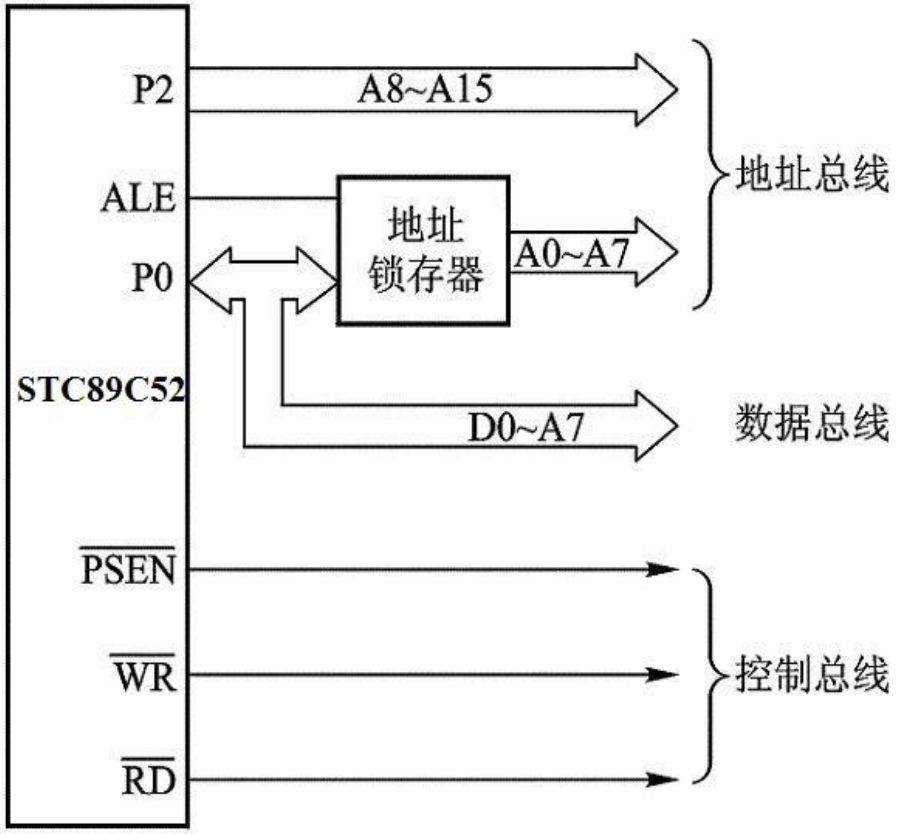
系统扩展结构



把具有共性的连线归并成一组公共连线，即总线（BUS）

地址总线(AB) 、数据总线(DB)、控制总线(CB).

```
uchar idata dADDR[16]   _at_   0x40;
uchar xdata xADDR[16]  _at_   0x4000;

xADDR[0]=dADDR[0];
```




```
MOV    R0,#40H
MOV    DPTR,#4000H
MOV    A，@R0

MOVX   @DPTR，A
```

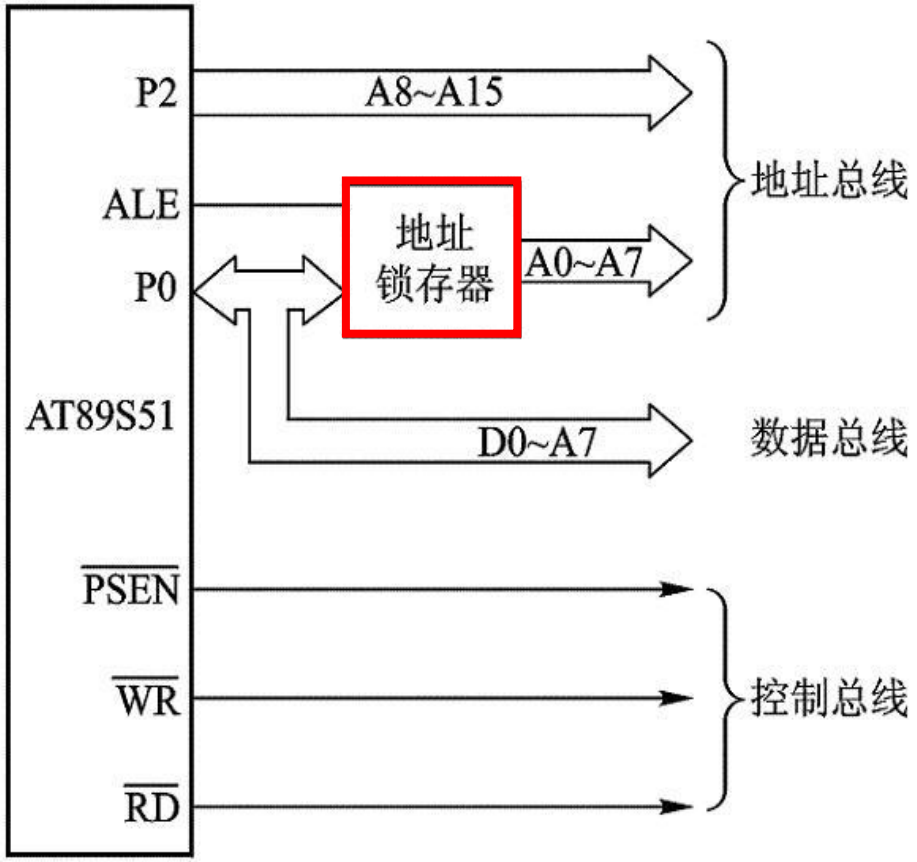
数据总线——P0口

地址总线——P0+P2

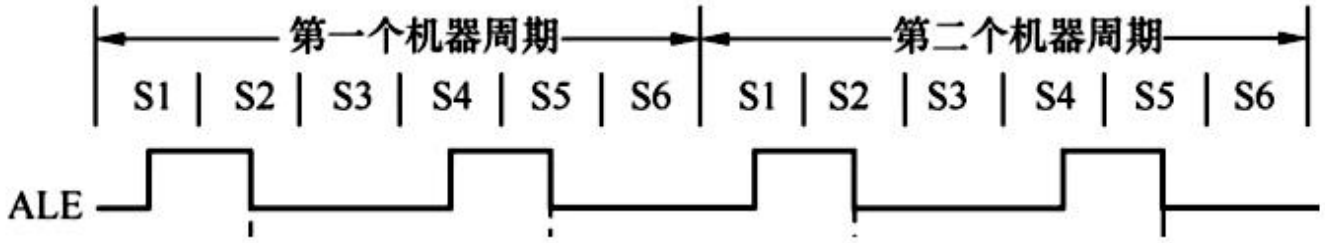
控制总线——P3口部分引脚

系统扩展后，四个并行I/O口就只剩P1和P3的部分口线，作为普通I/O使用。

为什么要用地址锁存器？

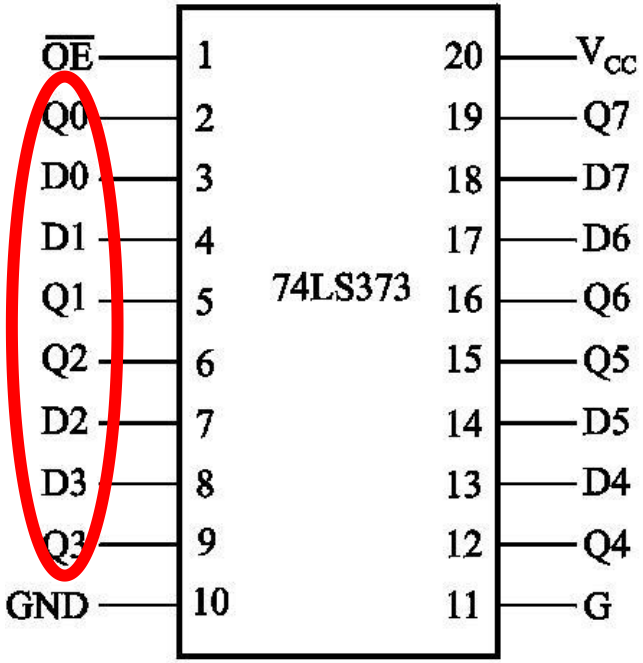


- P0口先将低8位地址信号通过锁存器输出，P0口再输出8位数据信号；
- 同时因为锁存器的输出端与P0口隔离，因此锁存器输出的低8位地址信号不会被8位数据信息覆盖，与P2口输出的高8位地址一起产生16位地址信号；
- 在对外进行数据传送时，地址必须保持不变；低8位地址和8位数据在P0口线上分时复用；



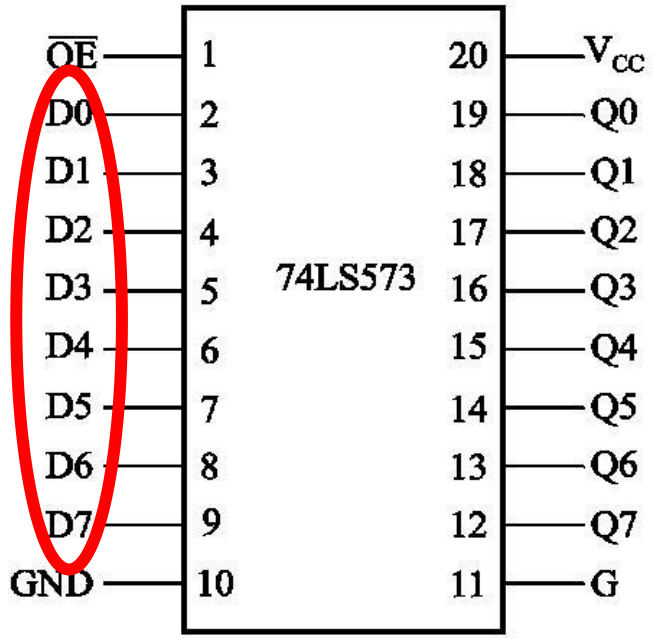
在每个机器周期，ALE两次有效，可以利用地址锁存器在ALE的下降沿将P0口输出的地址信息锁存，当ALE转为低电平时，P0输出8位数据信息。

常用的锁存器74LS373 & 74LS573



说明：

- 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
- $\overline{OE}$ ：数据输出允许信号



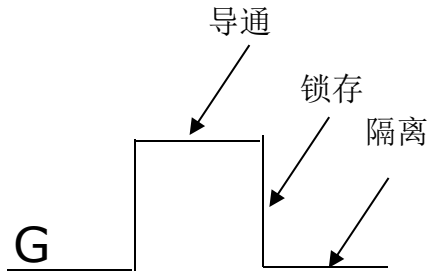


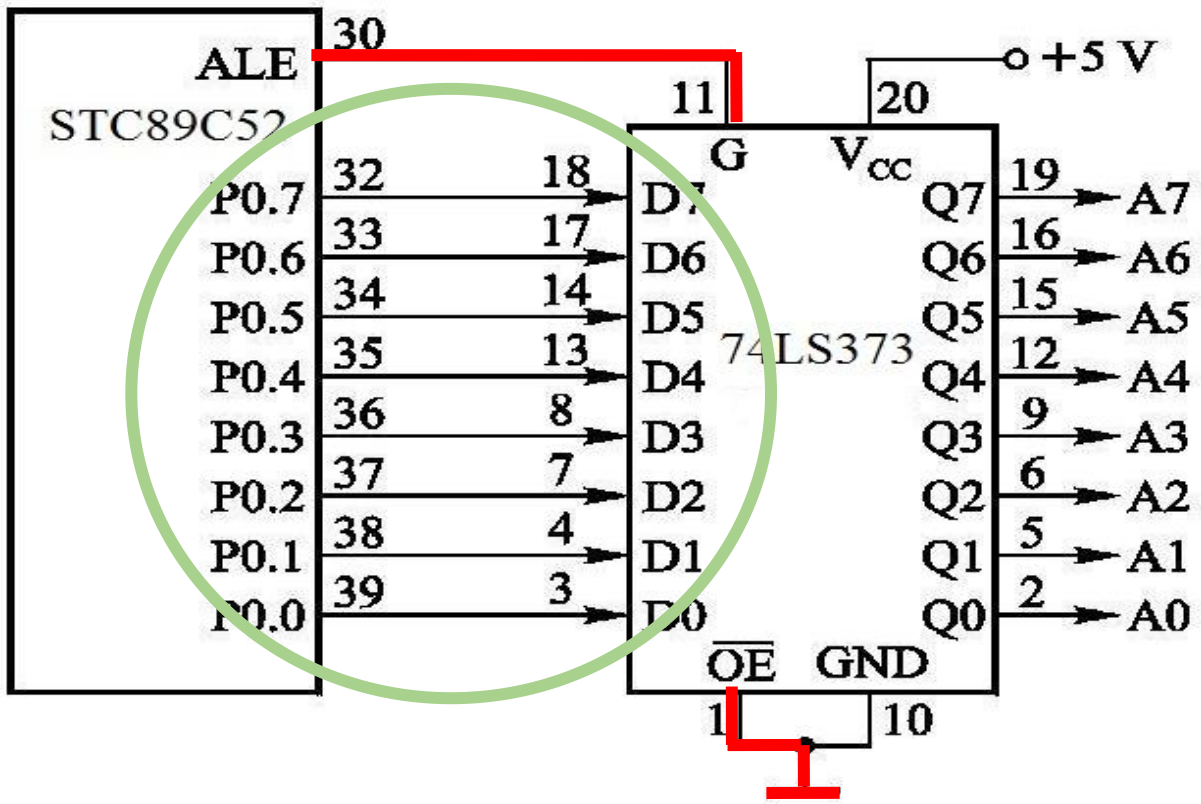
表 8-3　74LS373 功能表

$\overline{OE}$	G	D	Q
0	1	1	1
0	1	0	0
0	0	×	不变
1	×	×	高阻态

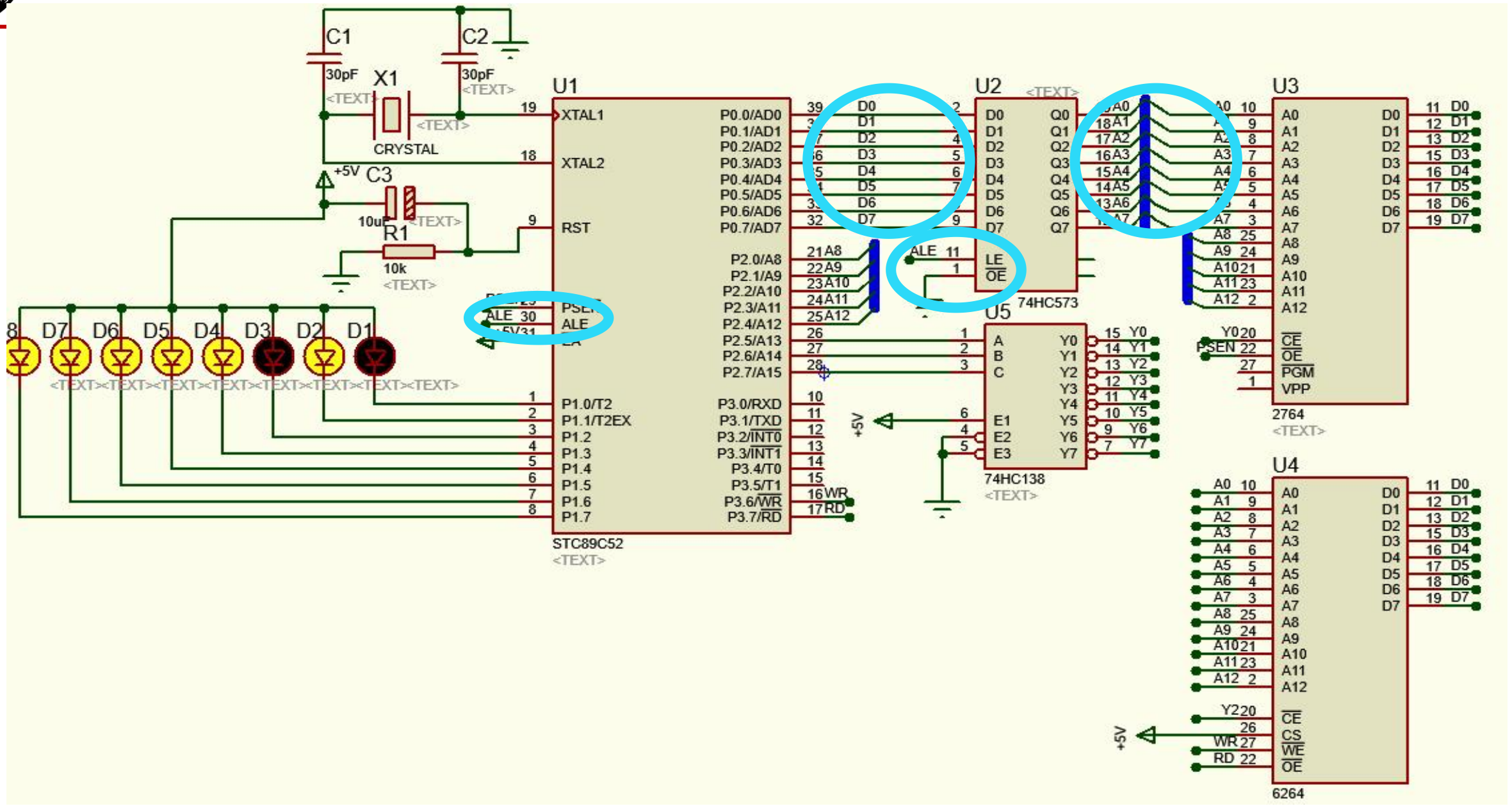
锁存器的一般接法

一般接法

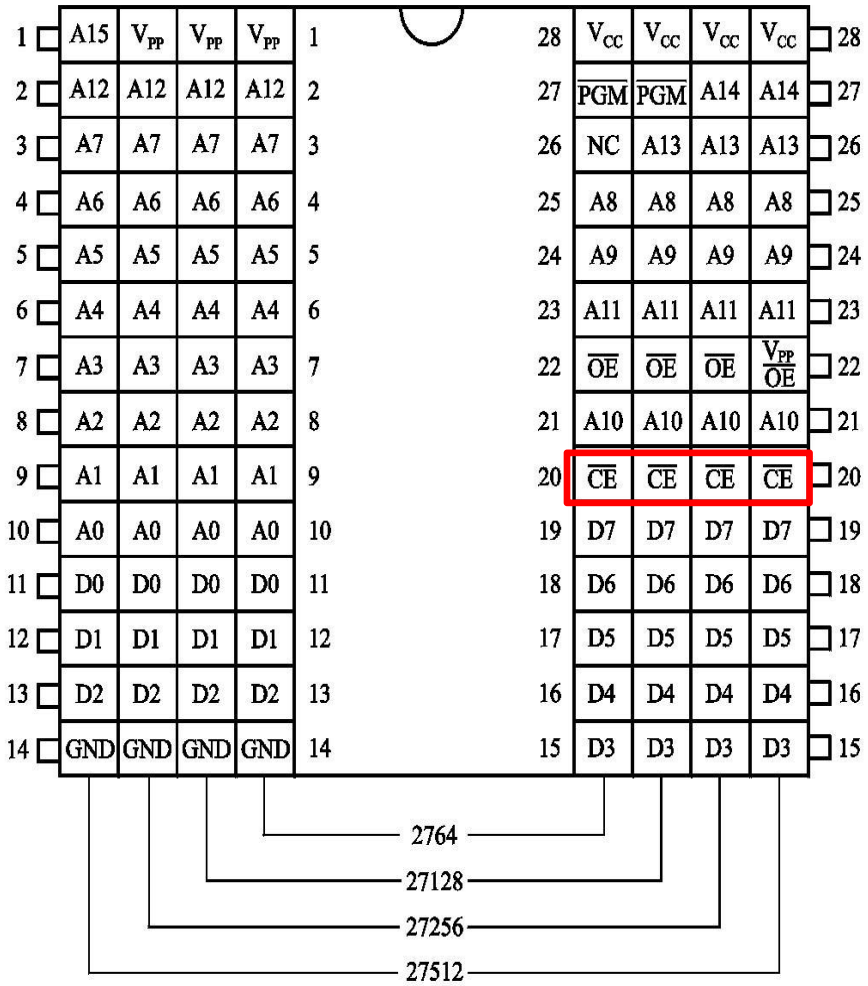
- D7～D0：8位数据输入线
- Q7～Q0：8位数据输出线
- G：数据输入锁存选通信号
- $\overline{OE}$：数据输出允许信号



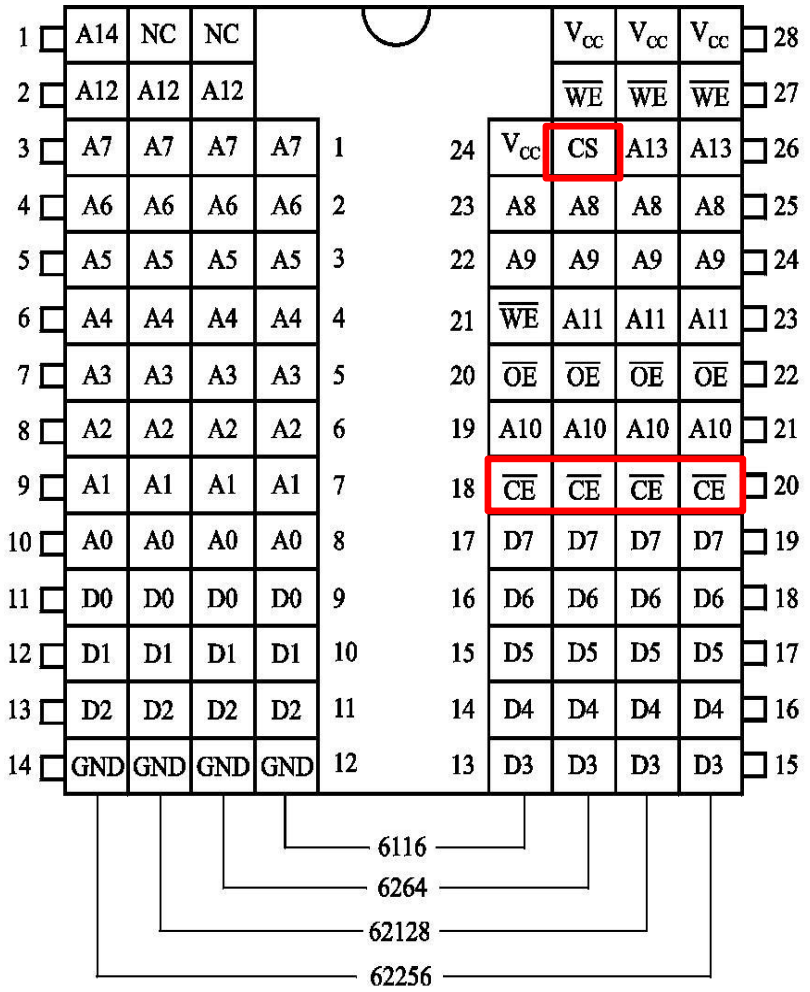
锁存器的一般接法


C1
30pF
C2
30pF
X1
CRYSTAL
C3
10uF
R1
10k
+5V
U1
STC89C52
XTAL1
XTAL2
RST
ALE
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
U2
74HC573
LE
OE
U3
2764
CE
OE
PGM
VPP
U4
6264
CS
WE
U5
74HC138
A
B
C
E1
E2
E3
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
PSEN
D8
D7
D6
D5
D4
D3
D2
D1

程序存储器



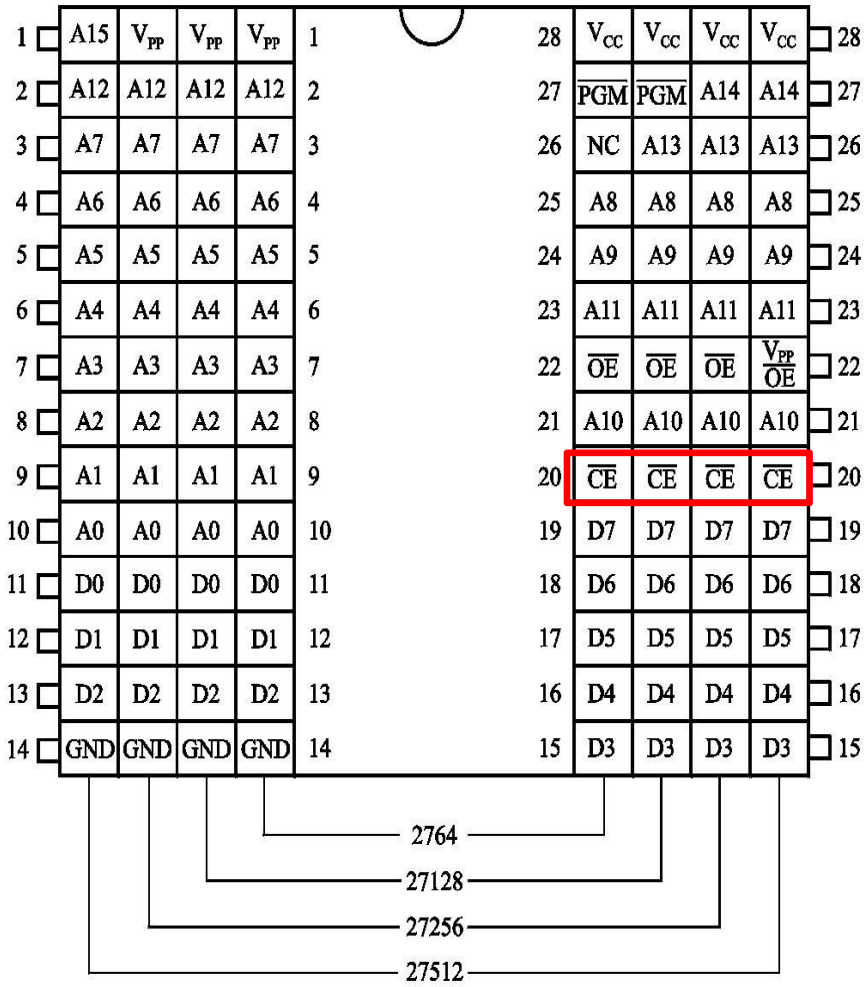
数据存储器



型号“27或62”后面的数字表示其位存储容量，除以8即换算成字节容量。

例如： “27128”或“62128”的字节容量为128/8 =16KB

程序存储器

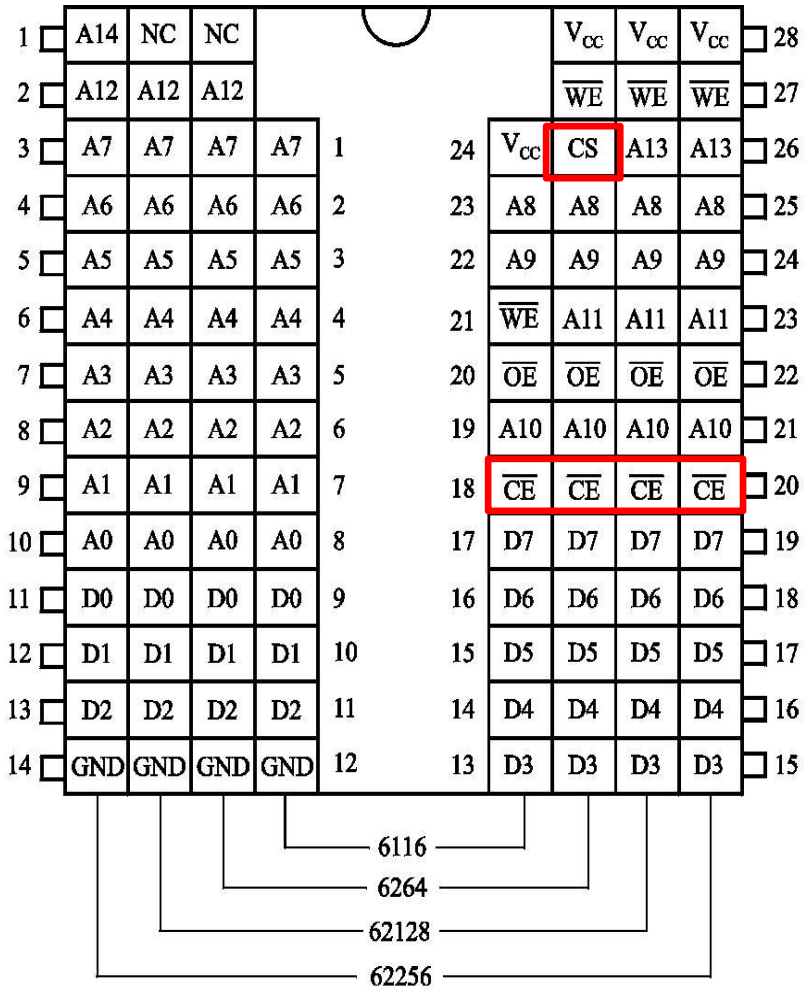


- D7～D0：数据线
- A0～A15：地址线

它的根数由芯片的存储容量决定，用于进行单元选择。

- $\overline{CE}$ ：片选信号，低电平有效
- $\overline{OE}$ ：输出允许，低电平有效

数据存储器



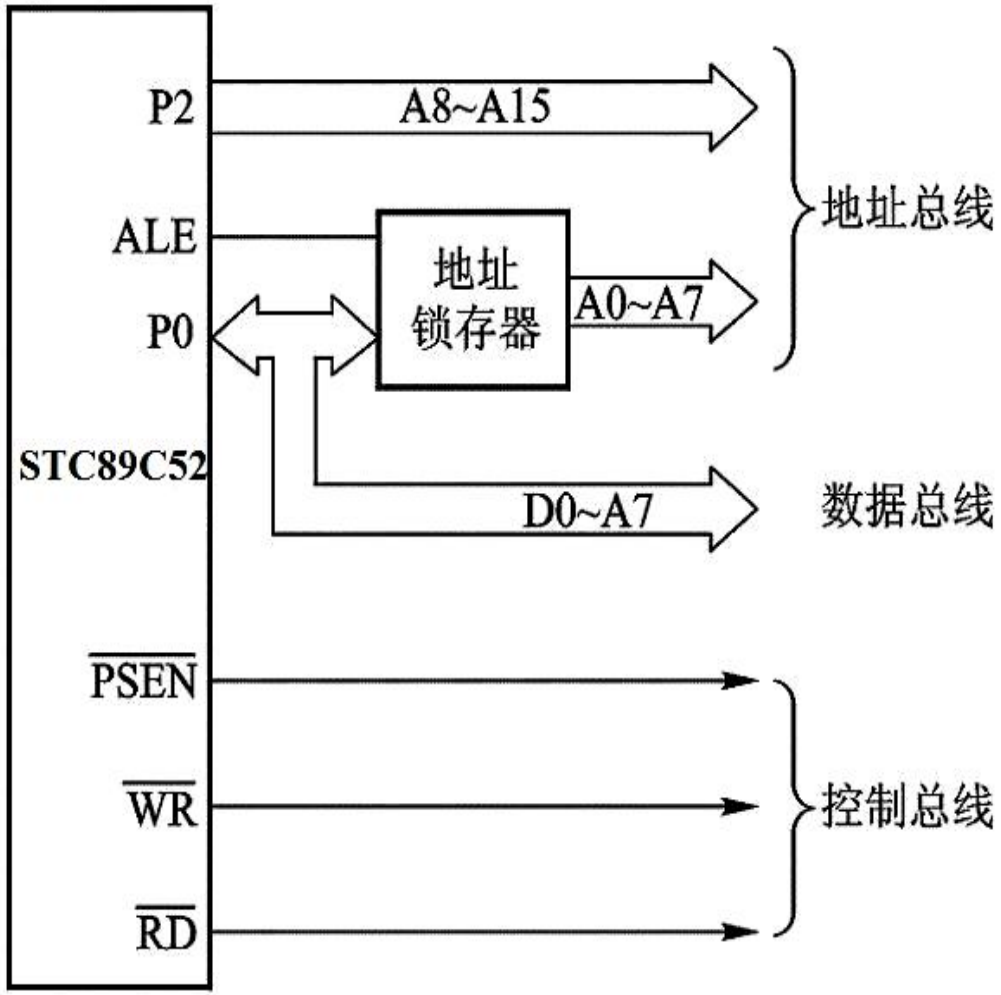
特殊：6264芯片，要CS引脚为高且 $\overline{CE}$ 引脚为低时才选中该片。

- D0～D7：数据线
- A0～A15：地址线

它的根数由芯片的存储容量决定，用于进行单元选择。

- $\overline{CE}$ ：片选信号。
- $\overline{OE}$ ：读信号，低电平有效。
- $\overline{WE}$ ：写信号，低电平有效。

扩展存储器的电路设计主要思路——三总线



- 数据总线：8位
  - P0口直接连接到存储器的D0~D7
- 地址总线：16位分为高8位和低8位。
  - 低8位由P0口通过锁存器连接到存储器的A0~A7.
  - 高8位由P2口采用线选直接连接或译码输出连接到存储器的A7~A15
- 控制总线：分为数据存储和程序存储
  - ALE作为P0口发出的低8位地址锁存控制
  - /EA引脚根据程序存储器确定应该选择什么电平
  - 程序存储扩展时，/PSEN引脚与程序存储器的/OE引脚相连
  - 数据存储扩展时，/WR和/RD引脚与数据存储器的/WE和/OE引脚相连

地址空间分配

如何分配片外的64KB空间，避免单片机发出一个地址时，同时访问不同存储器的存储单元，发生数据冲突。这就是存储器地址空间分配问题。

STC89C52单片机发出的地址码用于选择某个存储器单元。

外扩多片存储芯片时，单片机必须进行两种选择：

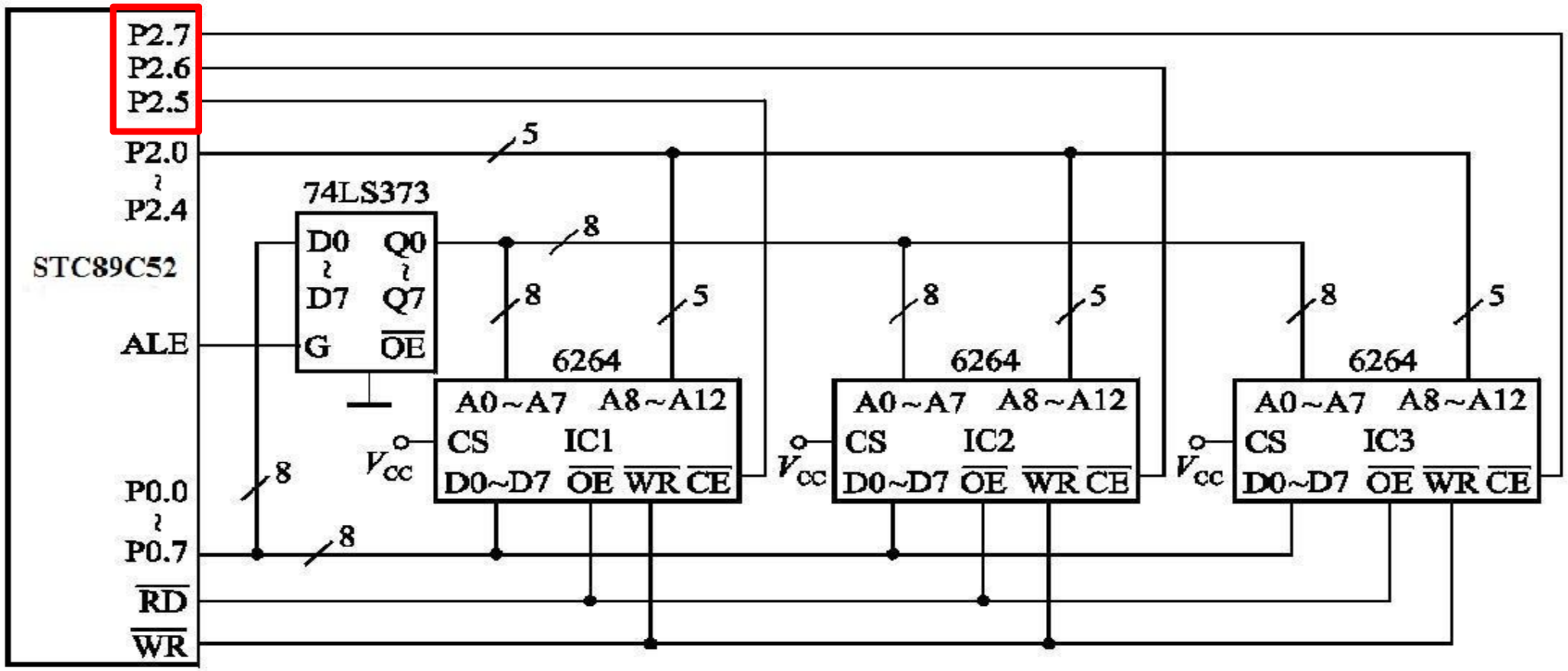
一是“片选”，存储器芯片有片选引脚，单片机控制使该引脚有效，从而选中该存储器芯片，未被选中的芯片不能被访问。

二是“单元选择“，存储器芯片有地址线引脚，单片机发出地址信号使用该引脚有效，从而选中该芯片中的某一单元。

地址空间分配方法有两种：线选法和译码法

1．线选法

直接以系统的高位地址线（P2口某几根多余的地址线）与存储器芯片的“片选”端直接连接。



例：扩展三片SRAM6264(8Kx8位)

线选法

分析:1）数据线8条：P0.7~P0.0接D7~D0

2）地址线低位：2^{13}=8K,13条： P2.4~P2.0P0.7~P0.0 接A_{12}~A_0

3）地址线高位：P2.5~P2.7 采用线选法与片选线 /CE相连

P2.5选6264(1)；P2.6选6264(2)；P2.7选6264(3)

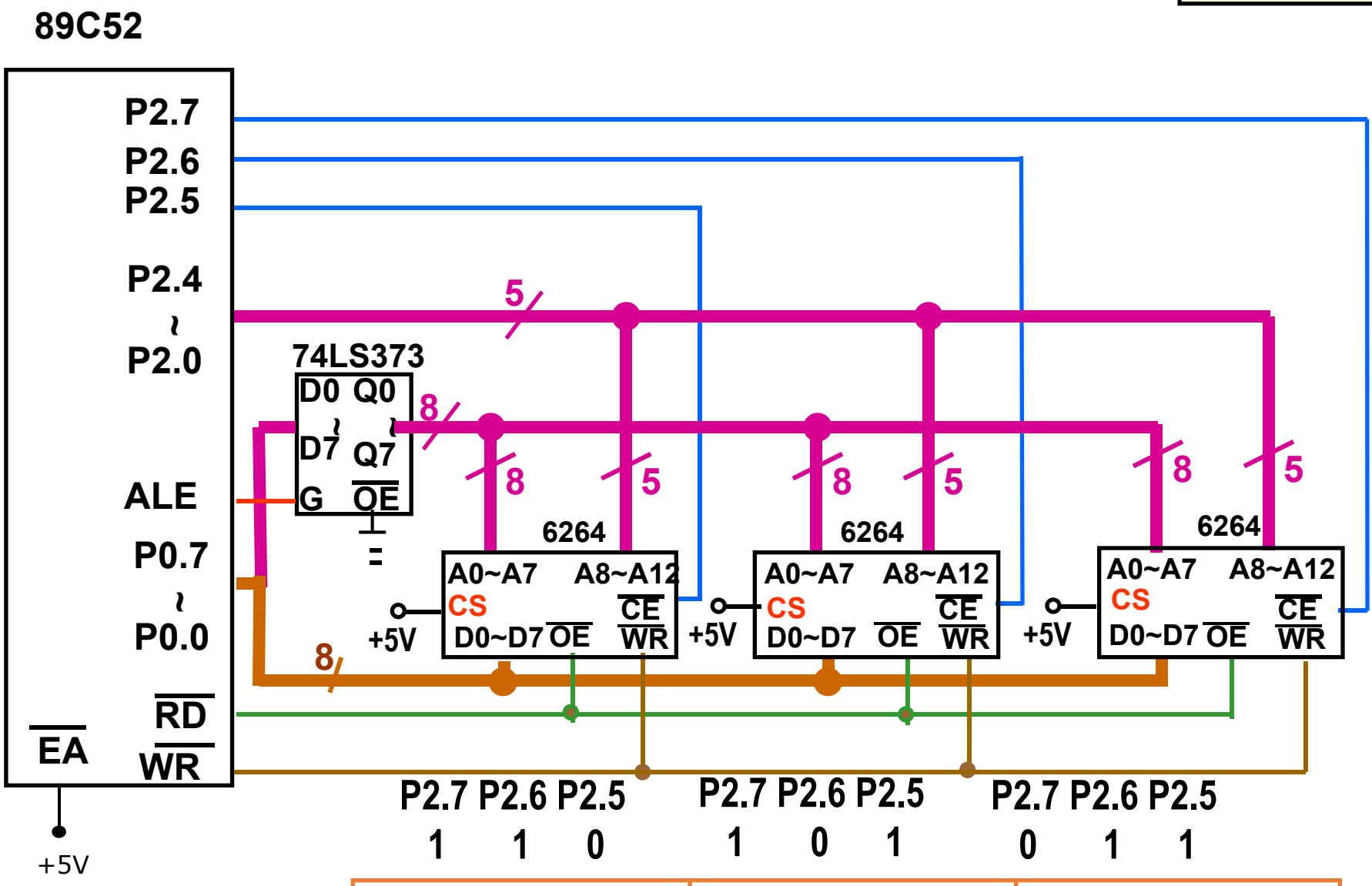
P2.7 P2.6 P2.5 线选	P2.4~P2.0	P0.7~P0.0	选中芯片	地址范围	存储容量
1 1 0	* ****	**** ****	IC1	C000H~DFFFH	8K
1 0 1	* ****	**** ****	IC2	A000H~BFFFH	8K
0 1 1	* ****	**** ****	IC3	6000H~7FFFH	8K

4）控制线： /WR接存储芯片的/WE；/RD接芯片的/OE端

/EA接地；ALE接锁存器

扩展3片6264与单片机的接口电路

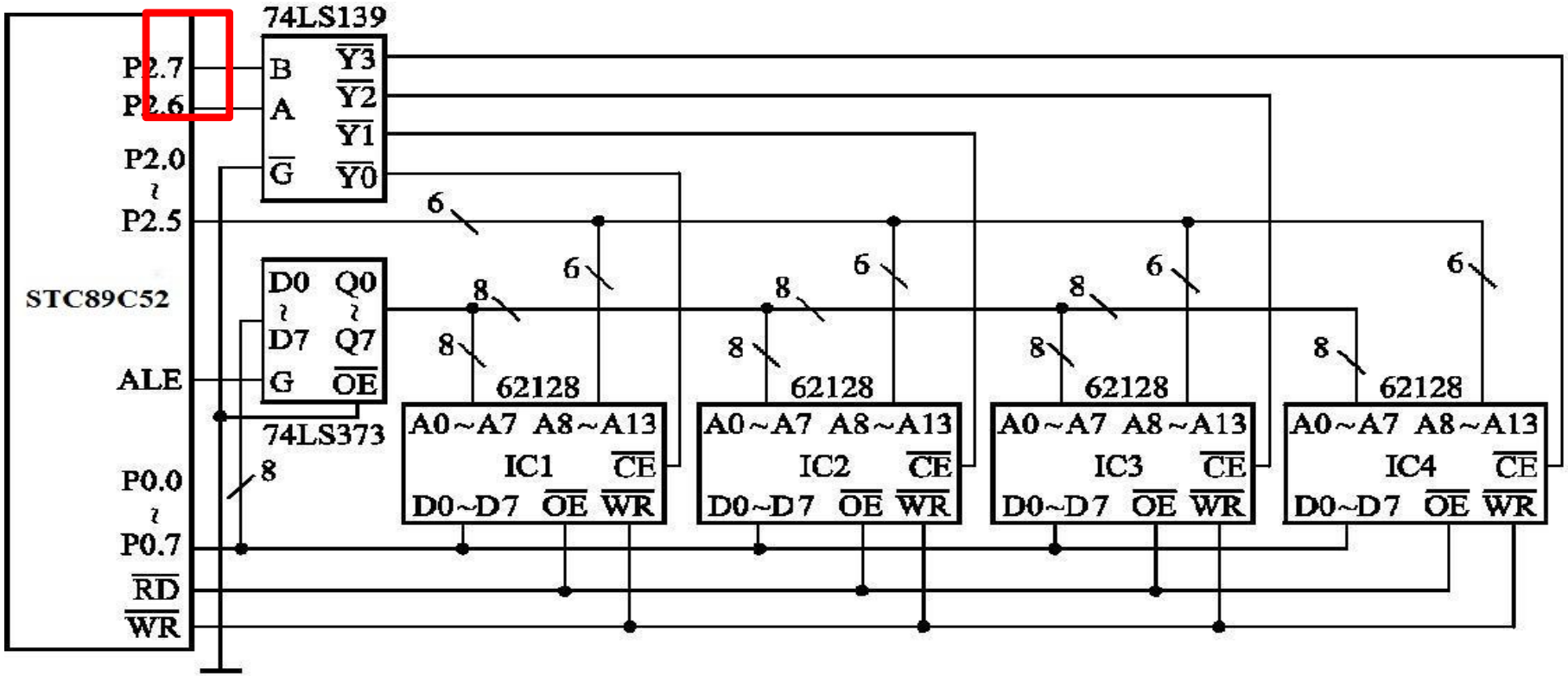
线选法



地址空间分配方法有两种：线选法和译码法

2、译码法

单片机的高位地址通过译码器译码输出作为存储器芯片的片选信号。



常用的译码器芯片：74LS138（3-8线译码器）、74LS139（双2-4线译码器）

译码器74LS138

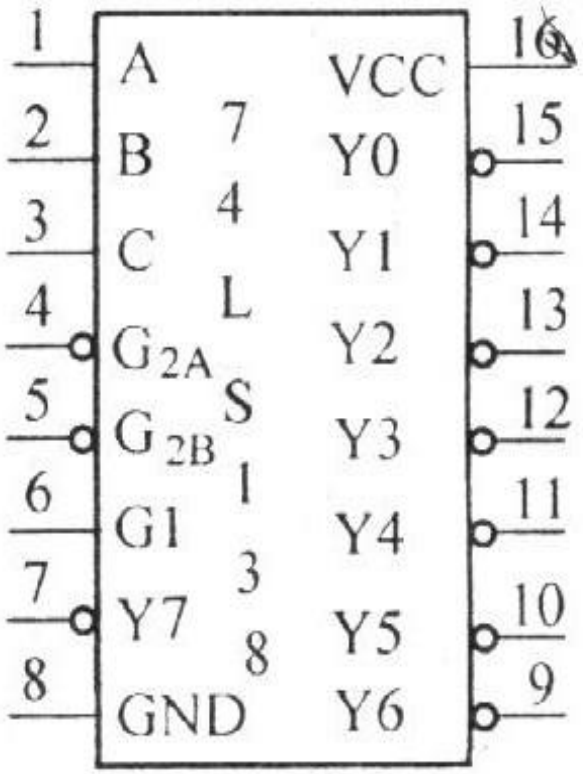


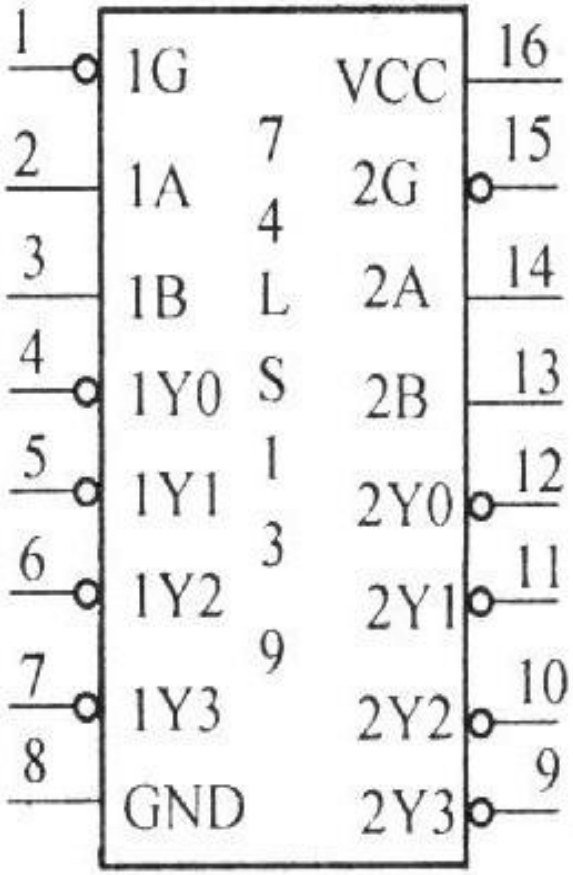
表 8-1　74LS138 真值表

输入端						输出端							
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	$\overline{Y7}$	$\overline{Y6}$	$\overline{Y5}$	$\overline{Y4}$	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
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	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	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
其他状态			×	×	×	1	1	1	1	1	1	1	1

注：1 表示高电平，0 表示低电平，×表示任意

3-8线译码器74LS138：3个数据输入端，译码产生八种状态

译码器74LS139



输入端			输出端			
允许	选择					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

双2-4线译码器74LS139，含两个完全独立的译码器

例：89C52扩展4片SRAM62128(16Kx8位)

译码法

分析: 1) 数据线：P0.7~P0.0接D7~D0

2）地址线低位：2^{14}=16K,14条： P2.5~P2.0P0.7~P0.0 接A_{13}~A_0

3）地址线高位：P2.6~P2.7 采用译码输出与片选线 /CE相连

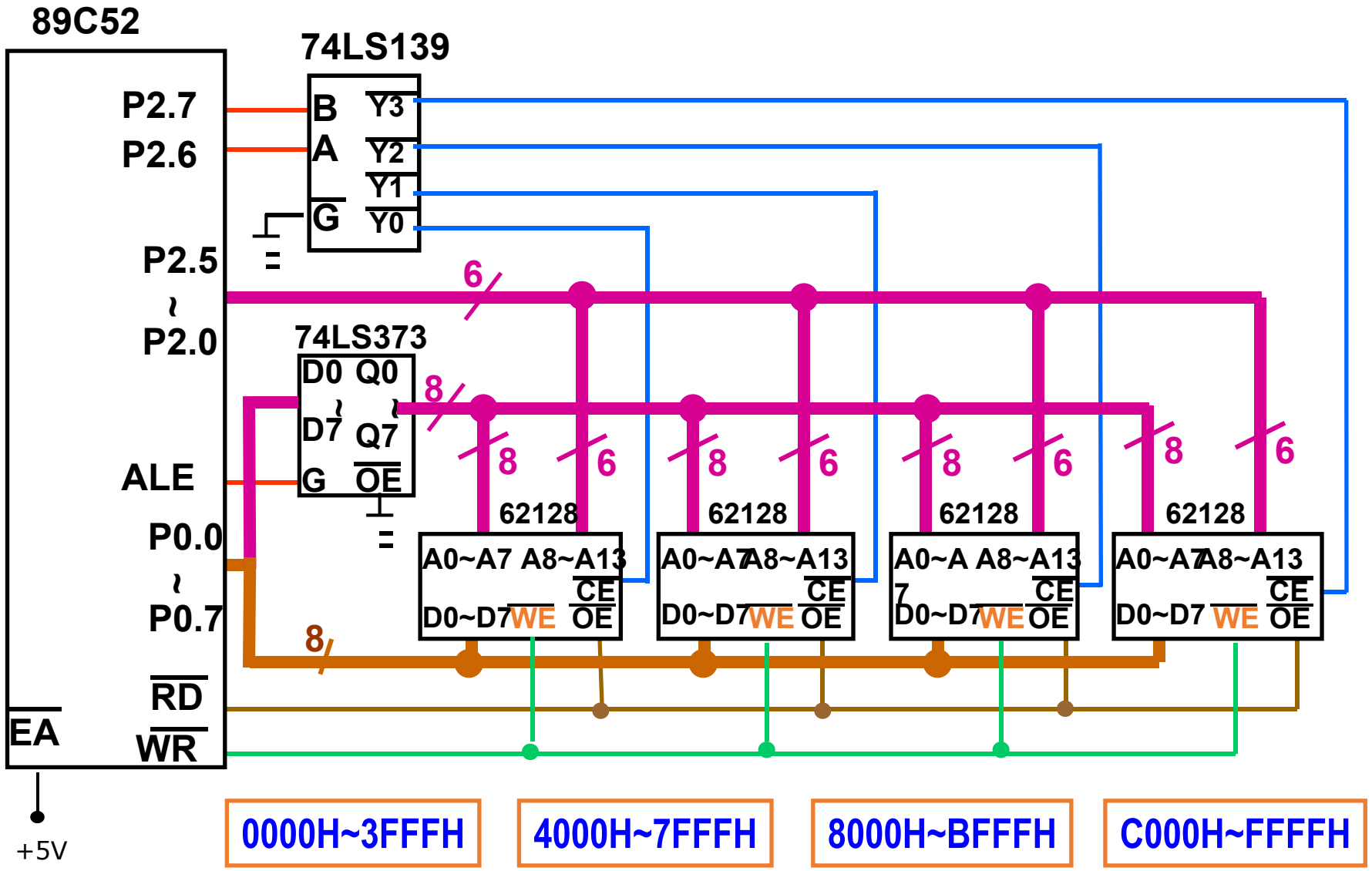
P2.7	P2.6	译码输出	选中芯片	地址范围	容量
0	0	Y0	IC1	0000H—3FFFH	16K
0	1	Y1	IC2	4000H—7FFFH	16K
1	0	Y2	IC3	8000H—BFFFH	16K
1	1	Y3	IC4	C000H--FFFFH	16K

4）控制线: /WR接存储芯片的/WE；/RD接芯片的/OE端

/EA接地；ALE接锁存器

扩展4片62128与单片机的接口电路

译码法



各存储器芯片地址范围

译码法特点：

1）适用于外扩存储器芯片较多的情况；

2）需设计译码电路，增加成本；

3）有效地利用存储器空间，避免了空间分散，适用于多芯片的存储器扩展。。

线选法特点：

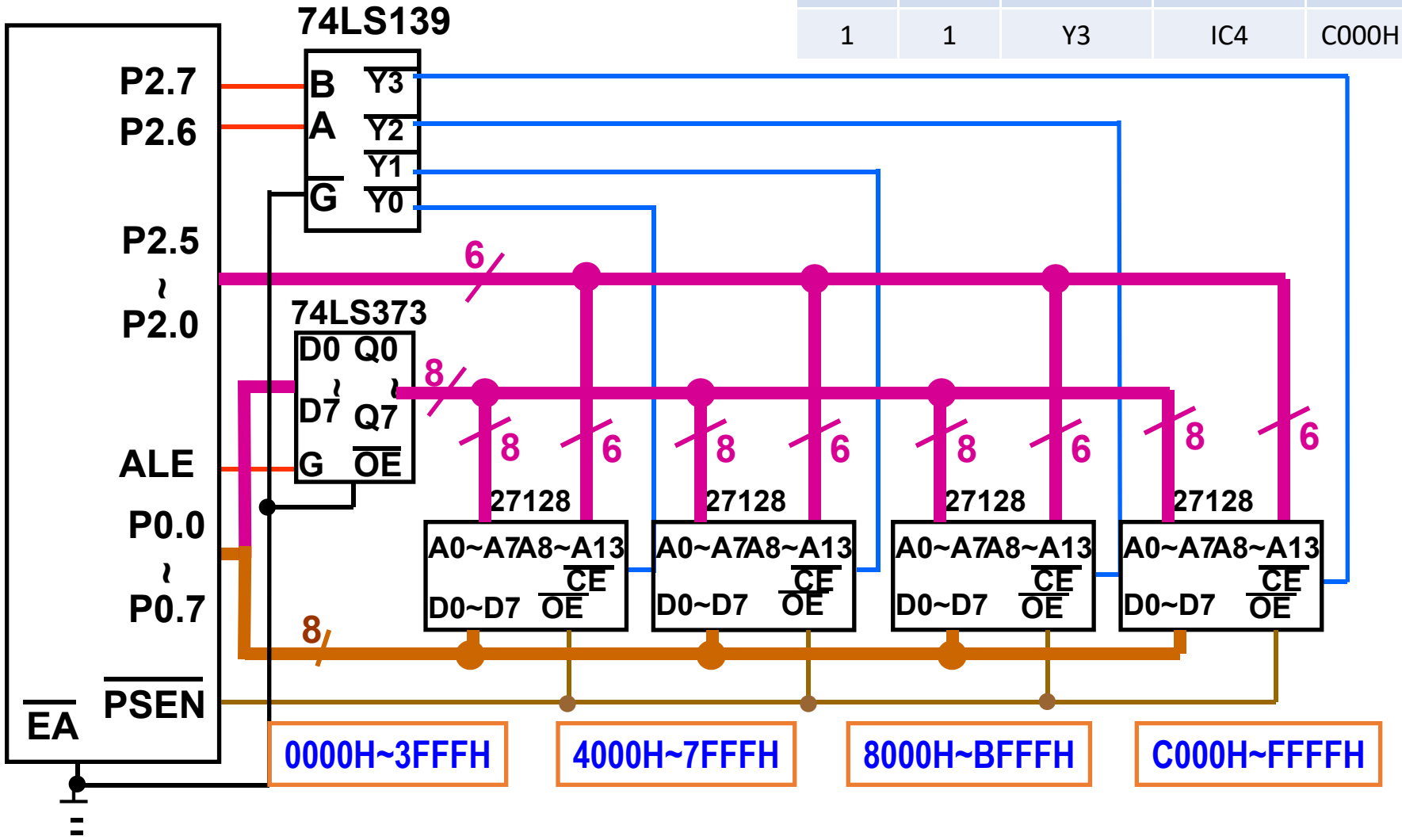
1）电路简单，不需要增加地址译码器硬件电路，体积小，成本低。

2）可寻址的芯片数目受到限制，适用于外扩存储器芯片较少的情况；

3）地址空间不连续，每个存储单元的地址不唯一，这会给程序设计带来不便，只适用于外扩芯片数目不多的单片机系统的存储器扩展。

问：外扩4片27128与62128的区别？

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



各存储器芯片地址范围

程序和数据存储器的综合扩展

在单片机应用系统中， 对外部扩展的存储器空间访问：

- **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。
- 对**片外数据存储器的读和写**由单片机的$\overline{\text{RD}}$(P3.7)和 $\overline{\text{WR}}$(P3.6)控制。
- 对**片外程序存储器的读取**由单片机的读选通 $\overline{\text{PSEN}}$ 控制。

说明：即使数据存储器与程序存储器**地址空间相同**，但由于控制信号不同，**不会发生总线冲突**。

片外ROM和RAM的最大容量都是64KB，地址为0000H～FFFFH。

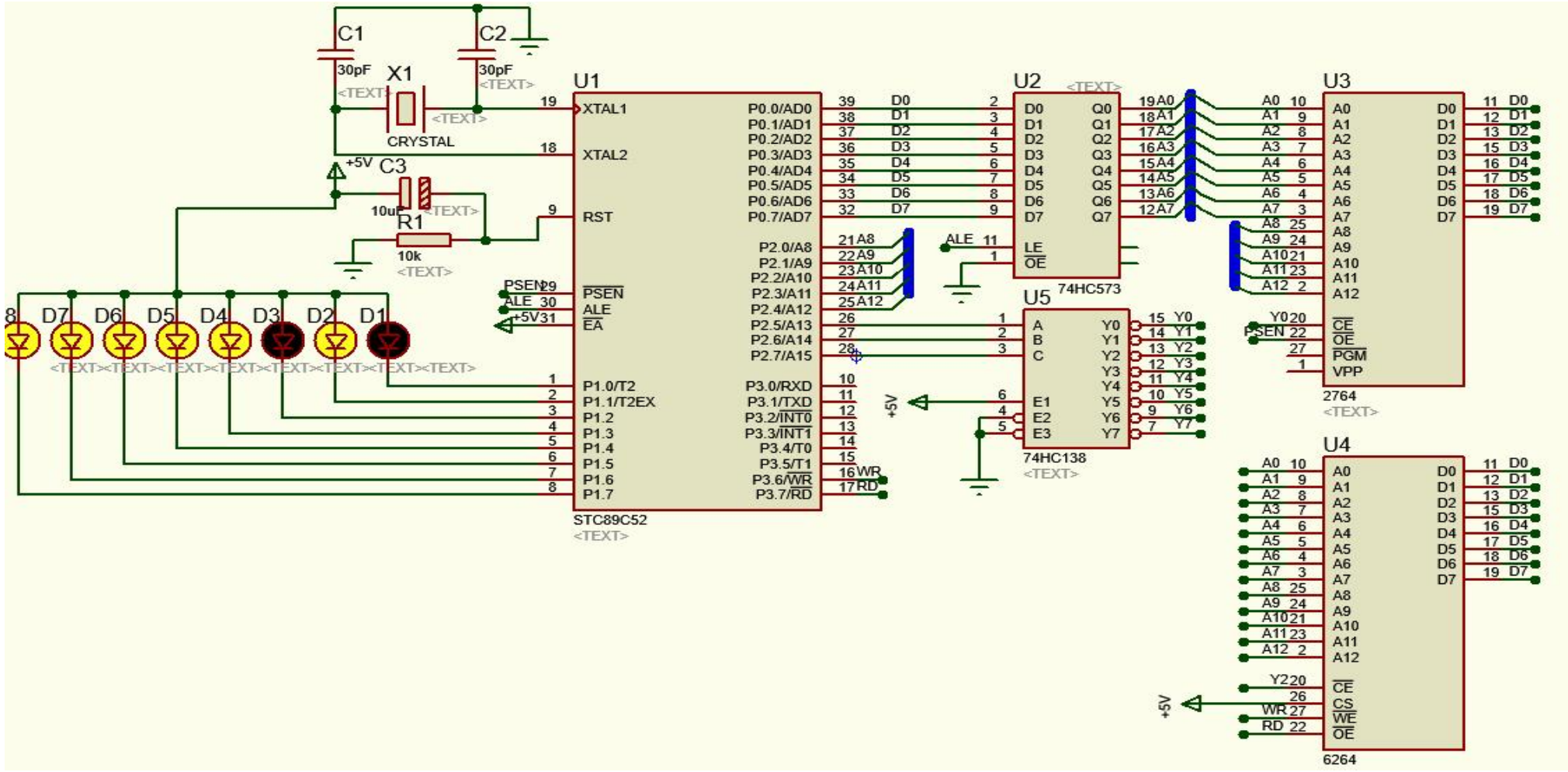
CPU对ROM的读操作由 $\overline{\text{PSEN}}$ 控制，指令用MOVC类；

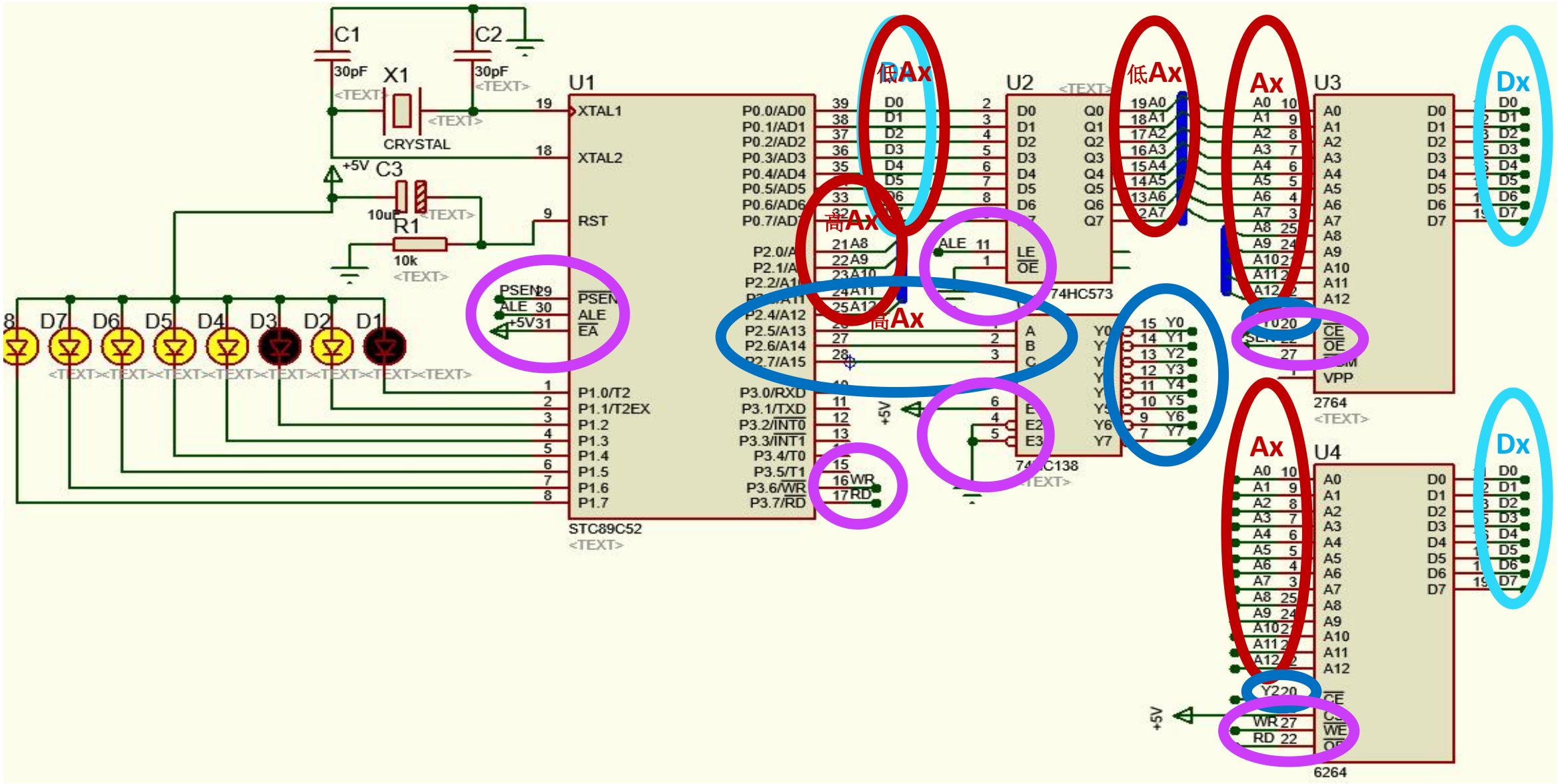
CPU对RAM读、写操作分别用 $\overline{\text{RD}}$ 和 $\overline{\text{WR}}$ 控制，指令用MOVX），

所以，由于STC89C52采用不同的控制信号和指令，地址不会冲突。

存储器扩展实验

【例3- 24&3-31】设单片机片内数据存储器首地址为40H，片外数据存储器首地址为4000H，存取数据字节个数16个。先将片内数据存储区的这16个字节的内容设置为01H~10H，再将片内首地址为30H开始的16个单元的内容传送到片外首地址为4000H开始的数据存储区中保存。




C1
C2
30pF
30pF
X1
CRYSTAL
+5V
C3
10uF
R1
10k
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
STC89C52
低Ax
高Ax
高Ax
U2
74HC573
ALE
LE
OE
74HC138
U3
2764
VPP
CE
OE
Ax
Dx
U4
6264
WR
RD
WE
D8
D7
D6
D5
D4
D3
D2
D1

【例3- 24&3-31】设单片机片内数据存储器首地址为40H，片外数据存储器首地址为4000H，存取数据字节个数16个。先将片内数据存储区的这16个字节的内容设置为01H~10H，再将片内首地址为30H开始的16个单元的内容传送到片外首地址为4000H开始的数据存储区中保存。

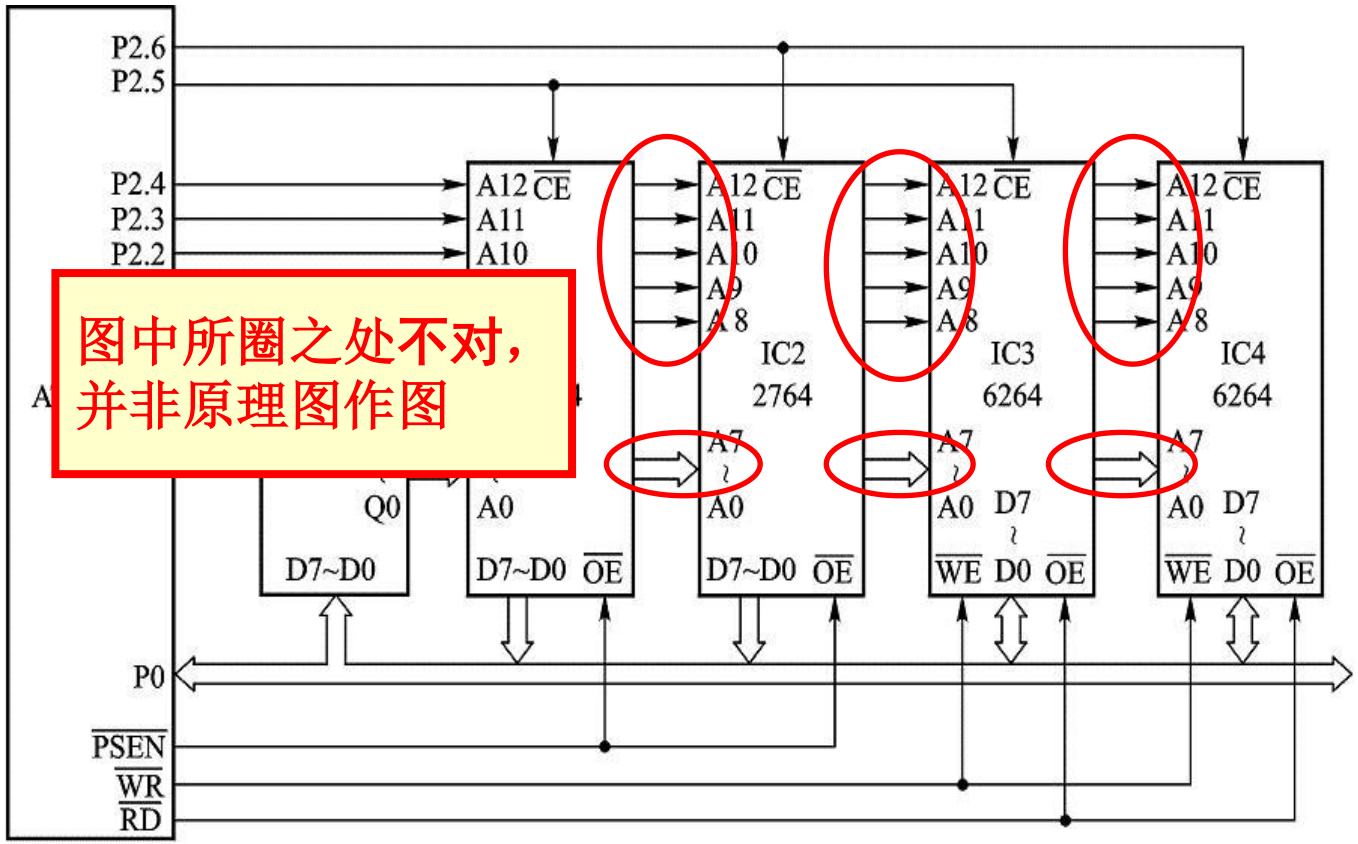
```
#include<reg52.h>
#define uint unsigned int
#define uchar unsigned char
#define Numbers  16
uchar idata dADDR[50] _at_ 0x40;
uchar xdata xADDR[50] _at_ 0x4000;

void delay(uint z)//ms延时函数
{     uint i;
      for(i=0;i<z;i++);
}
void Init(uchar lenth)//片内数据区初始化
{      uchar i;
       for(i=0;i<lenth;i++){
            dADDR[i]=i+1;}
}
void movcd(uchar lenth)//片内传送到片外函数
{      uchar i;
       for(i=0;i<lenth;i++){
          xADDR[i]=dADDR[i];}
}
```

```
void Display(uchar lenth)//传送数据送P1口
{       uchar i;
        for(i=0;i<lenth;i++){
                P1=xADDR[i];
                delay(20000);
        }
}


/*************主函数******************/
void main()
{     uint i;
      Init(Numbers);//片内数据区初始化
      movcd(Numbers);//片内传送到片外
      while(1){
          Display(Numbers);//数据送P1口
      }
}
```

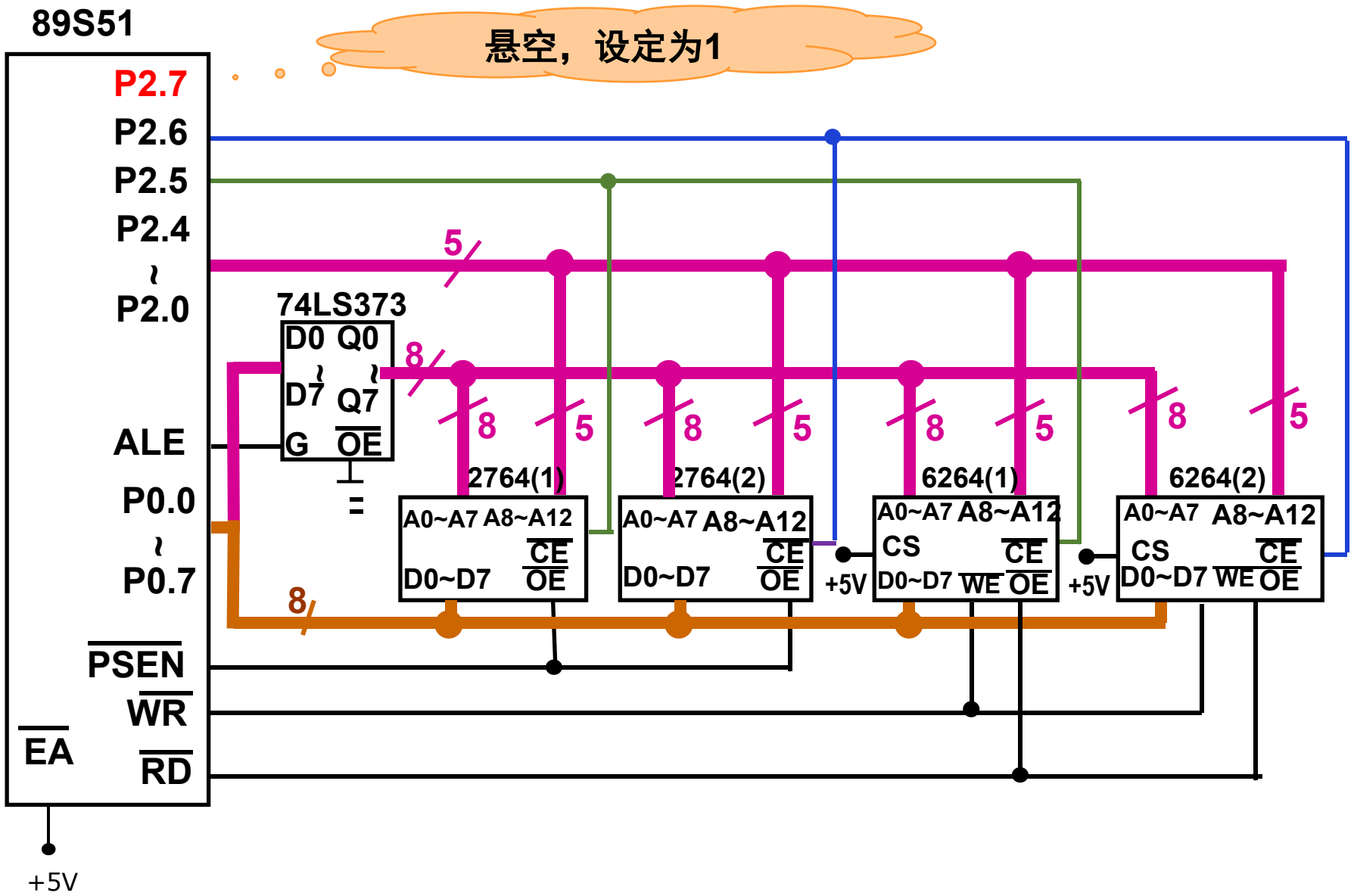
【例】采用线选法扩展2片SRAM 6264的和2片EPROM 2764。

要求：1）给出硬件接口电路。2）确定各芯片的地址范围。

3）编写程序将片外数据存储器中C000H～C0FFH单元设置为00H-FFH，并读取这些数据送P1口显示。



扩展2片6264和2片2764与单片机接口电路



分析：

- 两片6264和两片2764均为8K，因此单元选择需要地址线13条，即P2.4~P2.0 P0.7~P0.0即可。
- 高3位P2.5~P2.7地址线剩余：P2.6P2.5用于选择是2764还是6264，即P2.6=1，P2.5=0时，选中第一组2764或6264；当P2.6=0，P2.5=1时，选中第二组2764或6264。
- P2.7多余，悬空。

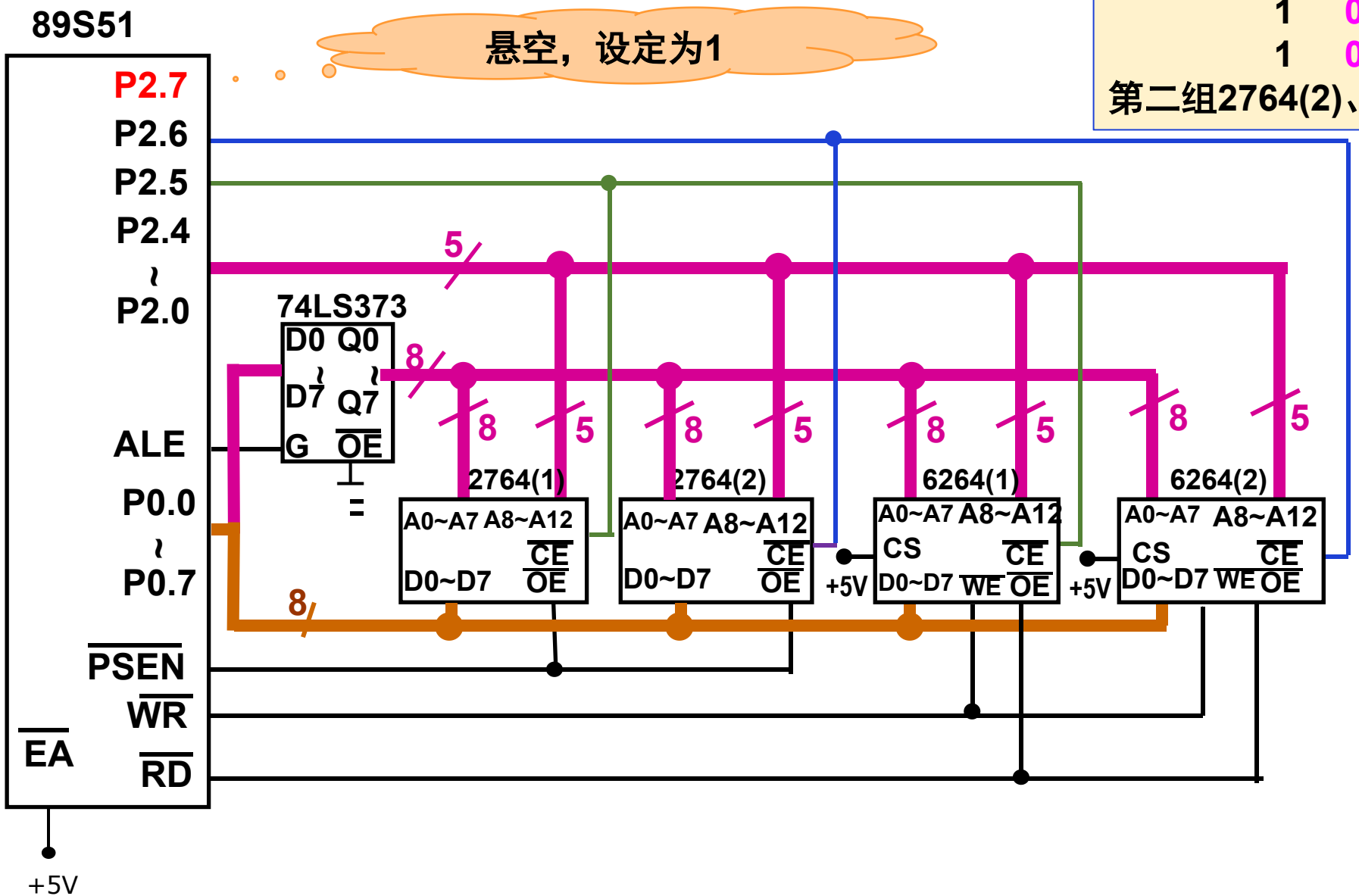
若P2.7状态定为1, 对应的存储空间为：

P2.7	P2.6	P2.5	P2.4~P2.0	P0.7~P0.0
1	1	0	0 ……	0
1	1	0	1 ……	1

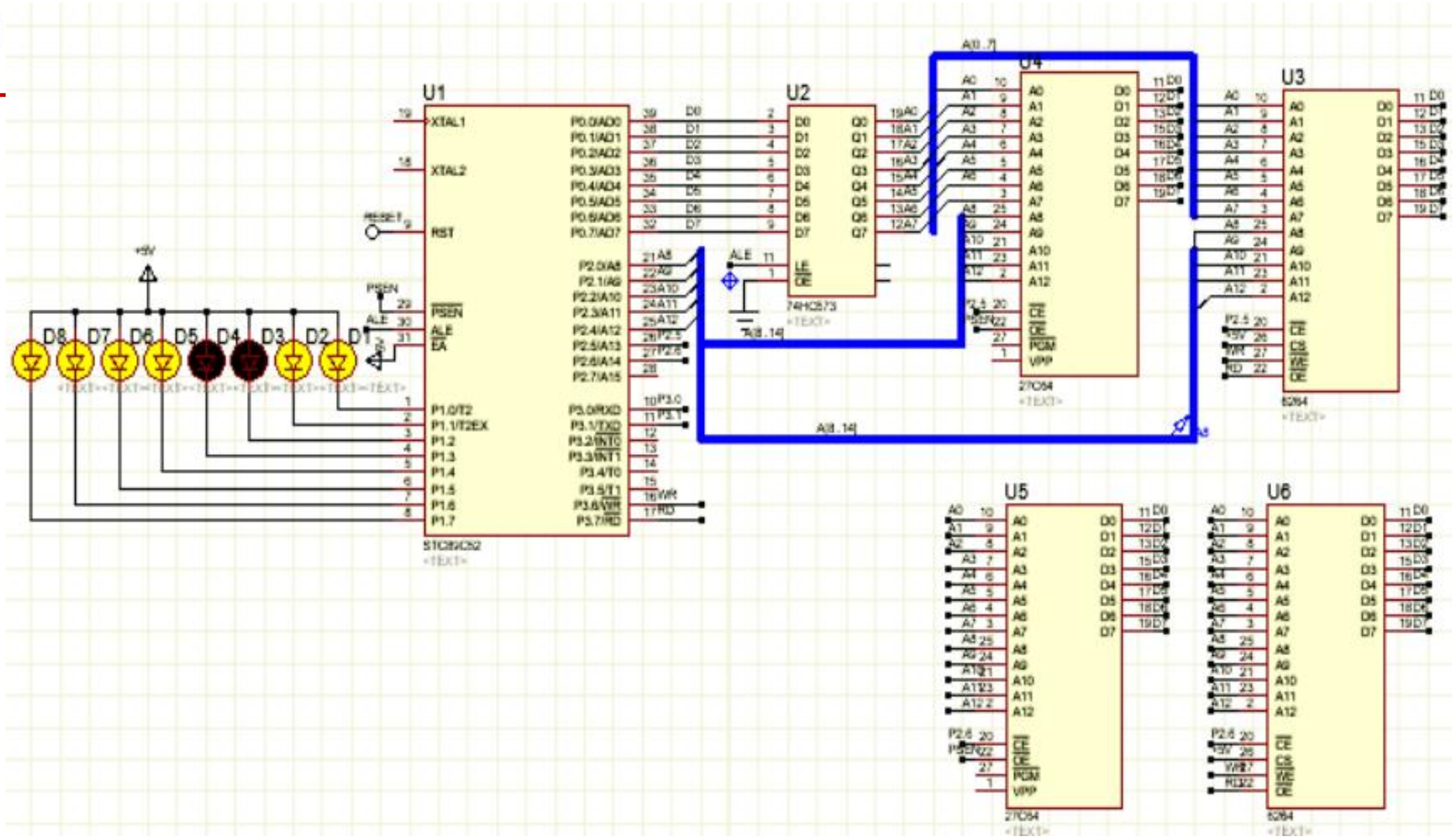
第一组2764(1)、6264(1)地址：C000H~DFFFH

1	0	1	0 ……	0
1	0	1	1 ……	1

第二组2764(2)、6264(2)地址：A000H~BFFFH



虽然每次同时选中两个芯片，具体对哪个芯片进行操作还要通过$\overline{PSEN}$、$\overline{RD}$、$\overline{WR}$这3个信号控制线来控制。它们是在执行指令时产生的，任意时刻只能执行一条指令，所以只能有一个信号有效，不可能同时有效，所以不会发生数据冲突。


U1
STC89C52
U2
74HC573
U3
U4
U5
U6
27C64
8264
D8 D7 D6 D5 D4 D3 D2 D1
+5V
RESET
PSEN
ALE
EA
XTAL1
XTAL2
RST
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
A8..14
A0..7

编写程序：将片外数据存储器中C000H～C0FFH单元设置为00H-FFH.

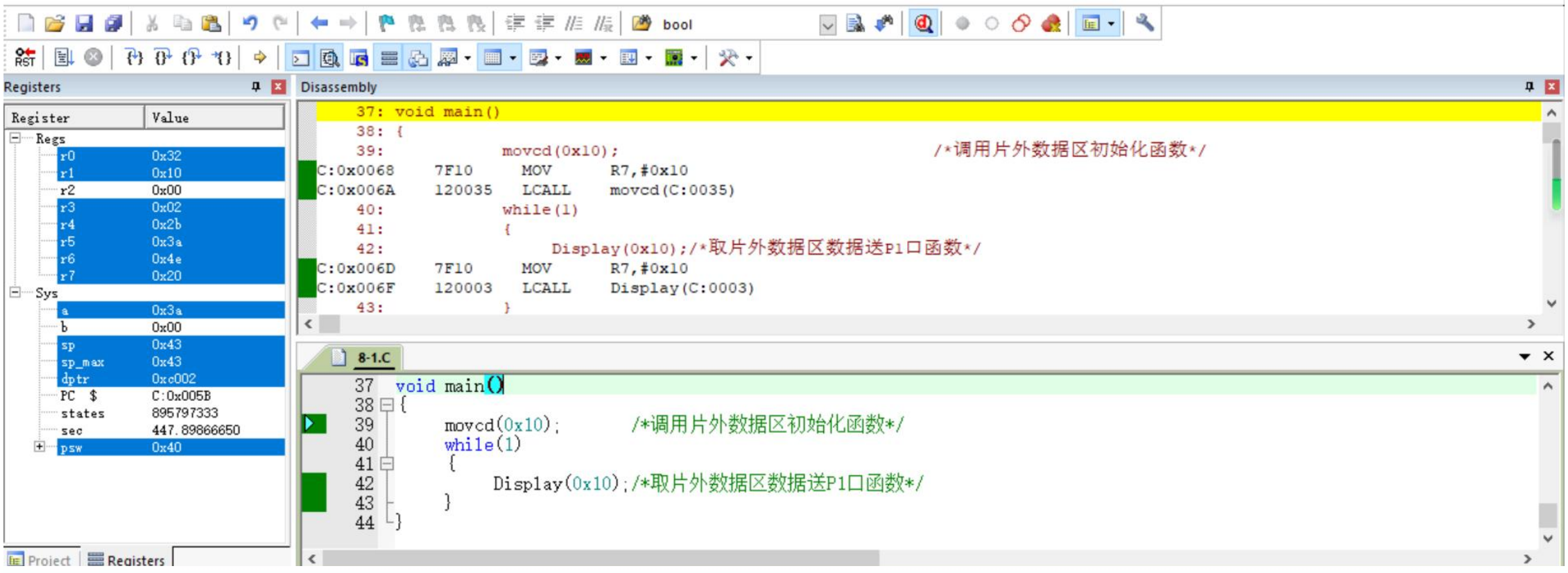
```
        ORG   0000H
        LJMP  MAIN
        ORG   0100H
MAIN:   MOV   R2,#00H
        MOV   DPTR,#0C000H
        MOV   A,#00H
LOOP：  MOVX  @DPTR,A
        INC   DPTR
        INC   A
        DJNZ  R2,LOOP
        SJMP  $
        END
```

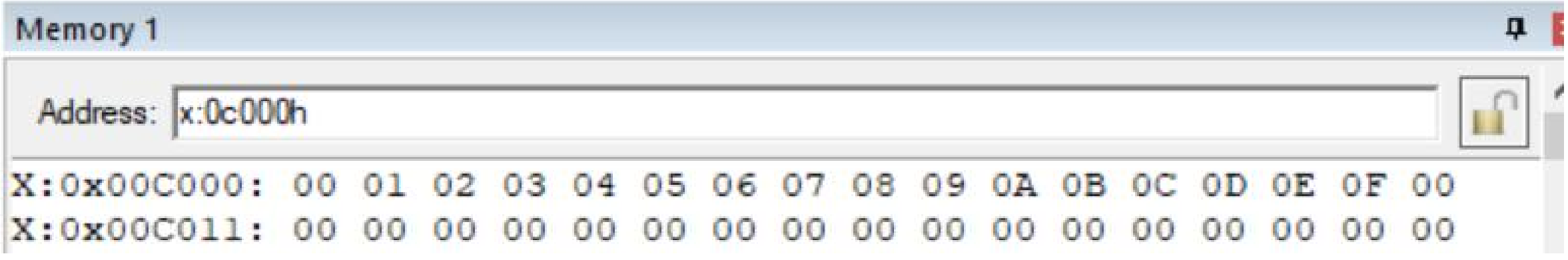
```
        ORG   0000H
        LJMP  MAIN
        ORG   0100H
MAIN:   MOV   DPTR,#0C000H
        MOV   A,#00H
LOOP：  MOVX  @DPTR,A
        INC   DPTR
        INC   A
        MOV   R7,DPL
        CJNE  R7, #0, LOOP
HERE：  SJMP  HERE
        END
```

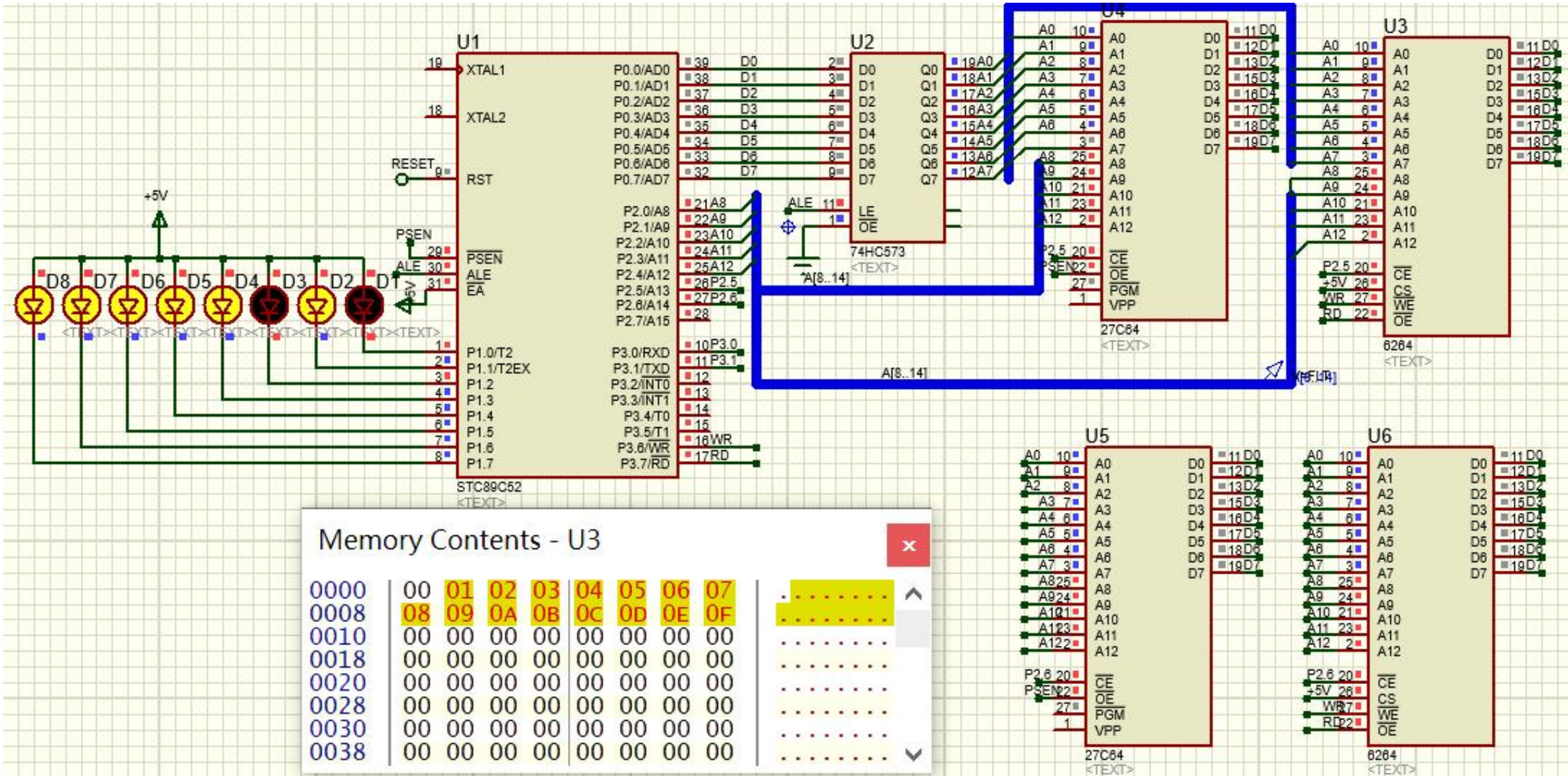
编写程序：将片外数据存储器中C000H～C0FFH单元设置为00H-0FH. 并读取这些数据送P1口显示。

```
#include<reg52.h>
#include<intrins.h>
#include <absacc.h>
#define uint unsigned int
#define uchar unsigned char
uchar xdata xADDR[16] _at_ 0xC000

void delay(uint z)
{
      uint i;
      for(i=0;i<z;i++);
}

void movcd(uchar lenth)
{
      uint i;
      for(i=0;i<lenth;i++){
         xADDR[i]=i;
      }
}
```

```
void Display(uchar lenth)
{
        uint i;
        for(i=0;i<lenth;i++)
        {
          P1=xADDR[i];
          delay(20000);
        }
  }

void main()
{
        movcd(0x10);
        while(1) {
          Display(0x10);
         }
}
```


Registers
Register
Value
Regs
r0 0x32
r1 0x10
r2 0x00
r3 0x02
r4 0x2b
r5 0x3a
r6 0x4e
r7 0x20
Sys
a 0x3a
b 0x00
sp 0x43
sp_max 0x43
dptr 0xc002
PC $ C:0x005B
states 895797333
sec 447.89866650
psw 0x40
Project
Registers
Disassembly
37: void main()
38: {
39: movcd(0x10); /*调用片外数据区初始化函数*/
C:0x0068 7F10 MOV R7,#0x10
C:0x006A 120035 LCALL movcd(C:0035)
40: while(1)
41: {
42: Display(0x10);/*取片外数据区数据送P1口函数*/
C:0x006D 7F10 MOV R7,#0x10
C:0x006F 120003 LCALL Display(C:0003)
43: }
8-1.C
37 void main()
38 {
39 movcd(0x10); /*调用片外数据区初始化函数*/
40 while(1)
41 {
42 Display(0x10);/*取片外数据区数据送P1口函数*/
43 }
44 }

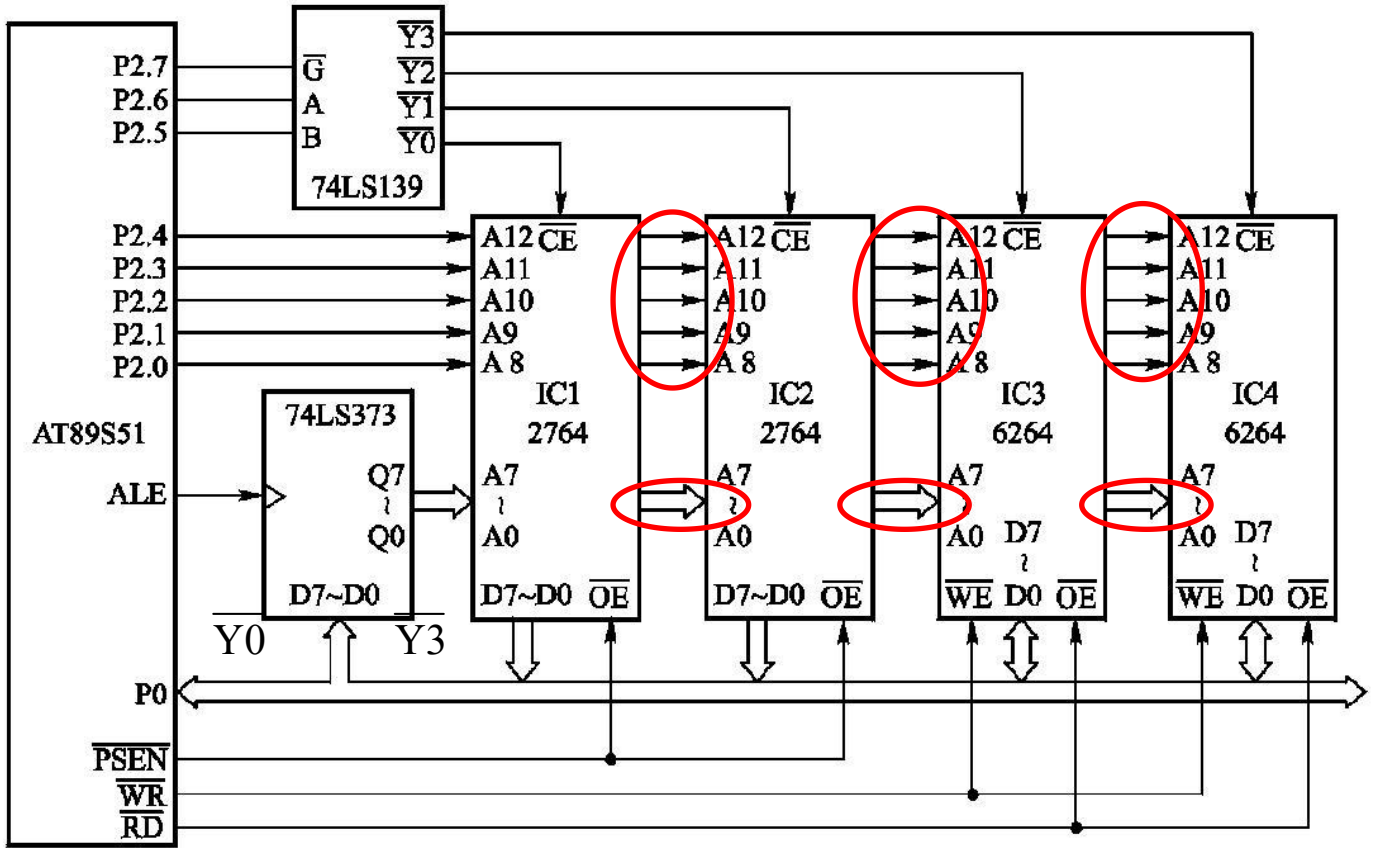

Memory 1
Address: x:0c000h
X:0x00C000: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 00
X:0x00C011: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00


U1
STC89C52
U2
74HC573
U3
6264
U4
27C64
U5
27C64
U6
6264
D8 D7 D6 D5 D4 D3 D2 D1
+5V
RESET
PSEN
ALE
EA
A[8..14]
Memory Contents - U3
0000 00 01 02 03 04 05 06 07
0008 08 09 0A 0B 0C 0D 0E 0F
0010 00 00 00 00 00 00 00 00
0018 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 00
0028 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00
0038 00 00 00 00 00 00 00 00

【例】采用译码法扩展2片SRAM 6264的和2片EPROM 2764。

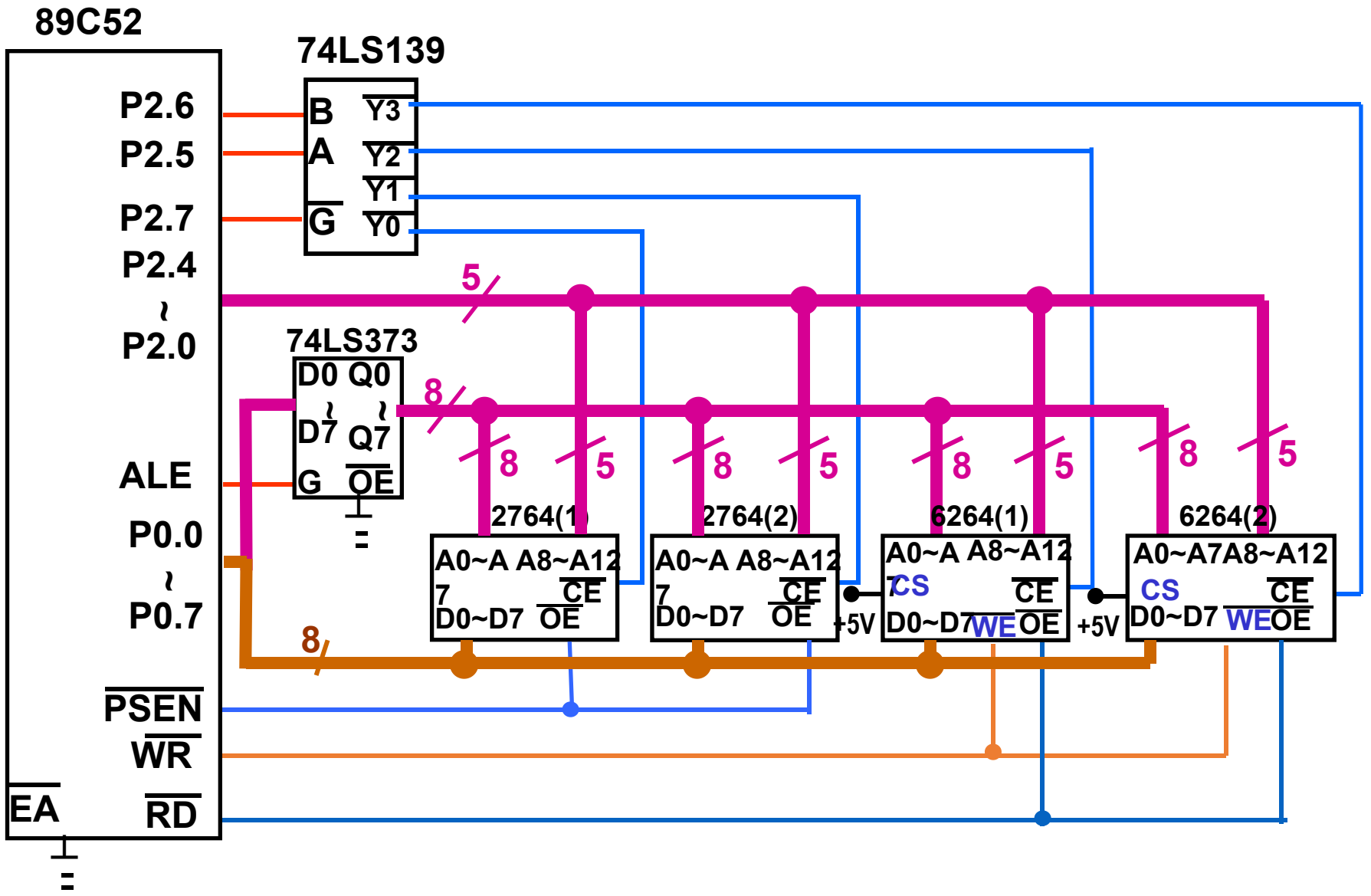
要求：1）给出硬件接口电路。 2）确定各芯片的地址范围。

3）编写程序：将片外第一片数据存储器6264中前256个单元全部清“0”。



图中所圈之处不对，并非原理图作图

方法二:译码法

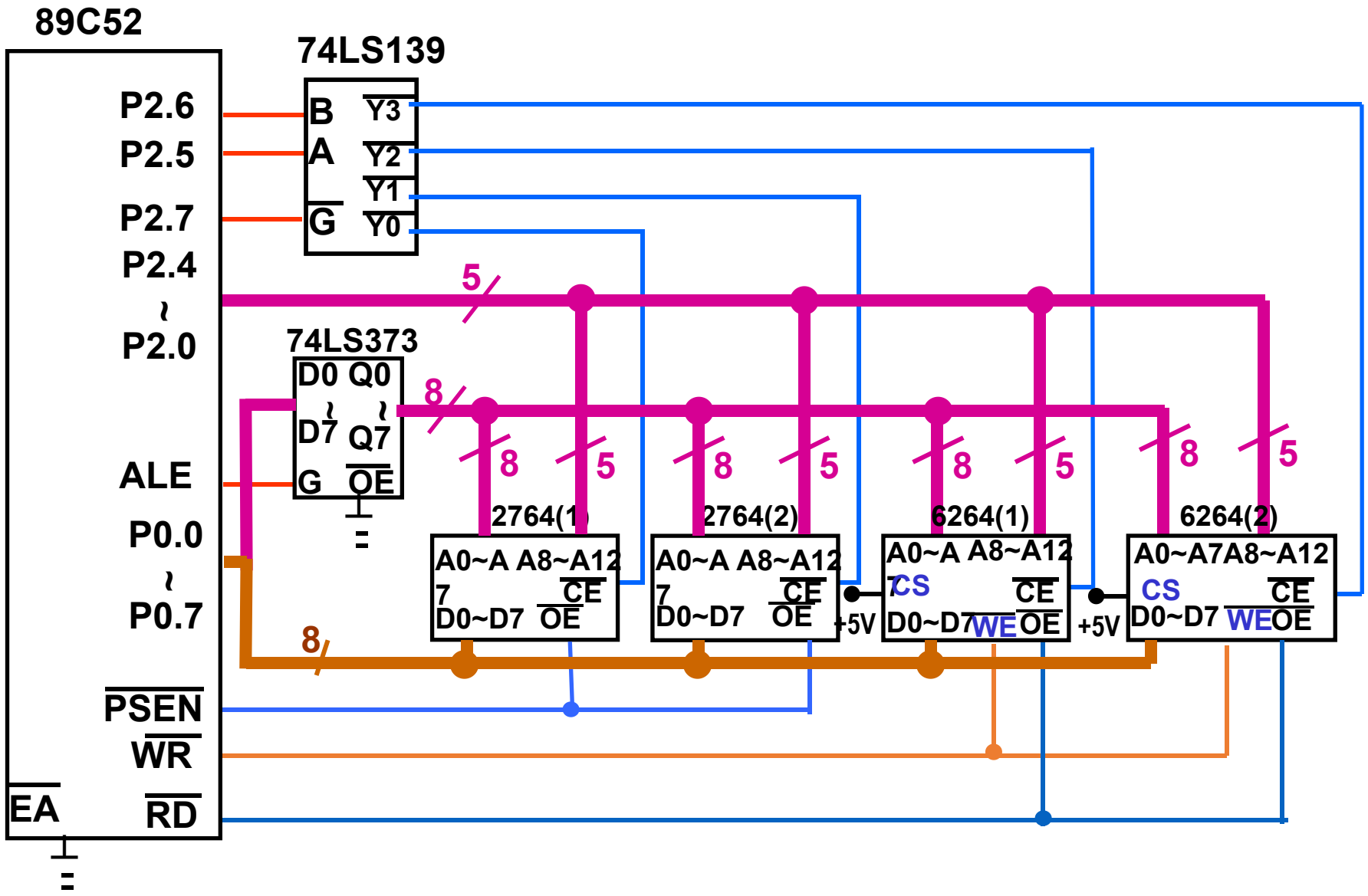


分析：

- 两片6264和两片2764均为8K，因此单元选择需要地址线13条，即P2.4~P2.0 P0.7~P0.0即可。

- 高3位P2.5~P2.7地址线剩余：连接到74LS139译码器的3个输入端B、A、$\overline{G}$，输出端Y0～Y3分别连接4片存储器芯片的片选端。

方法二:译码法



G	B	A		
P2.7	P2.6	P2.5	P2.4~P0.0	芯片
0	0	0	……	2764(1)
0	0	1	……	2764(2)
0	1	0	……	6264(1)
0	1	1	……	6264(2)

各芯片存储空间为：

2764(1): 0000H-----1FFFH

2764(2): 2000H-----3FFFH

6264(1): 4000H-----5FFFH

6264(2): 6000H-----7FFFH

问：下列代码中的两个地址是哪个芯片的存储空间？

```
ORG  0000H
.....
MOV  DPTR, #4000H
MOVX  A, @DPTR
```

3）编写程序：将片外第一片数据存储器6264中前256个单元全部清“0”。

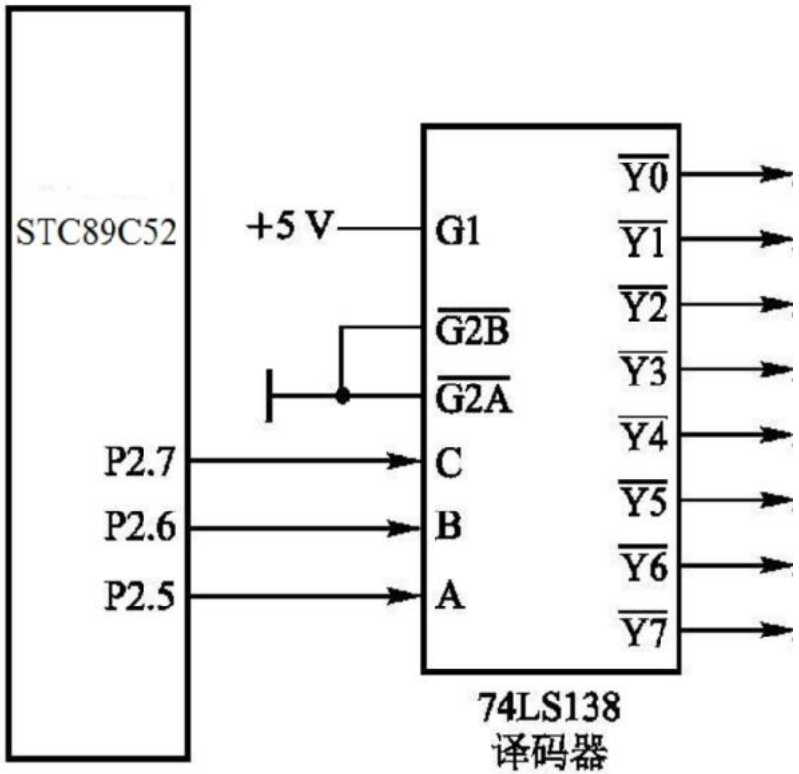
```
#include<reg52.h>
#include<intrins.h>
#include <absacc.h>
#define uint unsigned int
#define uchar unsigned char
uchar xdata xADDR[256] _at_ 0x4000

void movcd(uchar lenth)
{
      uint i;
      for(i=0;i<lenth+1;i++){
        xADDR[i]=0;
      }
}
void main()
{
        movcd(0xff);
        while(1) { ;
         }
}
```

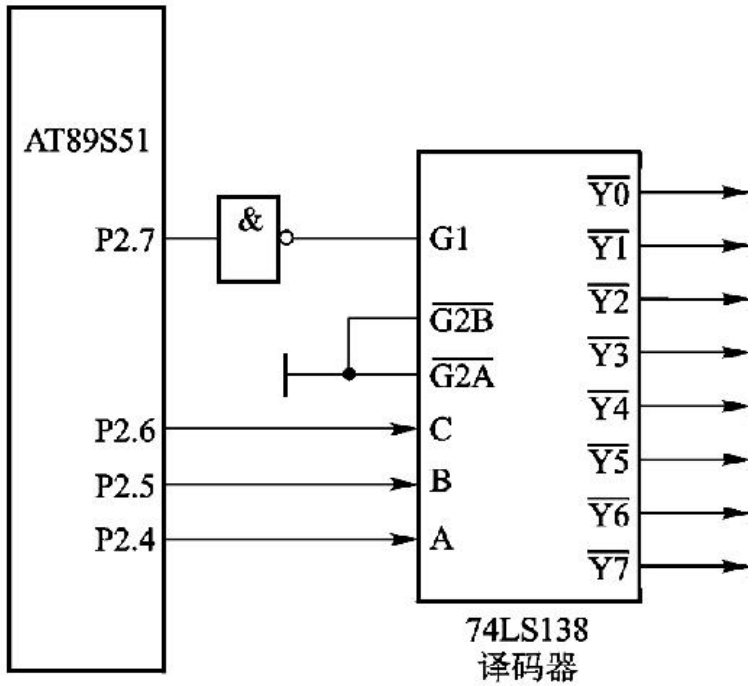
```
        ORG   0000H
        LJMP  MAIN
        ORG   0100H
MAIN:   MOV   R7, #00H
        MOV   DPTR, #4000H
        CLR   A
LOOP:   MOVX  @DPTR, A
        INC   DPTR
        DJNZ  R7, LOOP
        SJMP  $
        END
```

【例】假设实现：将程序存储器2764中以TAB为首址的64个单元的内容依次传送到第一片6264。设置DPTR指向待传送的数据块的首地址#TAB，循环次数为64。

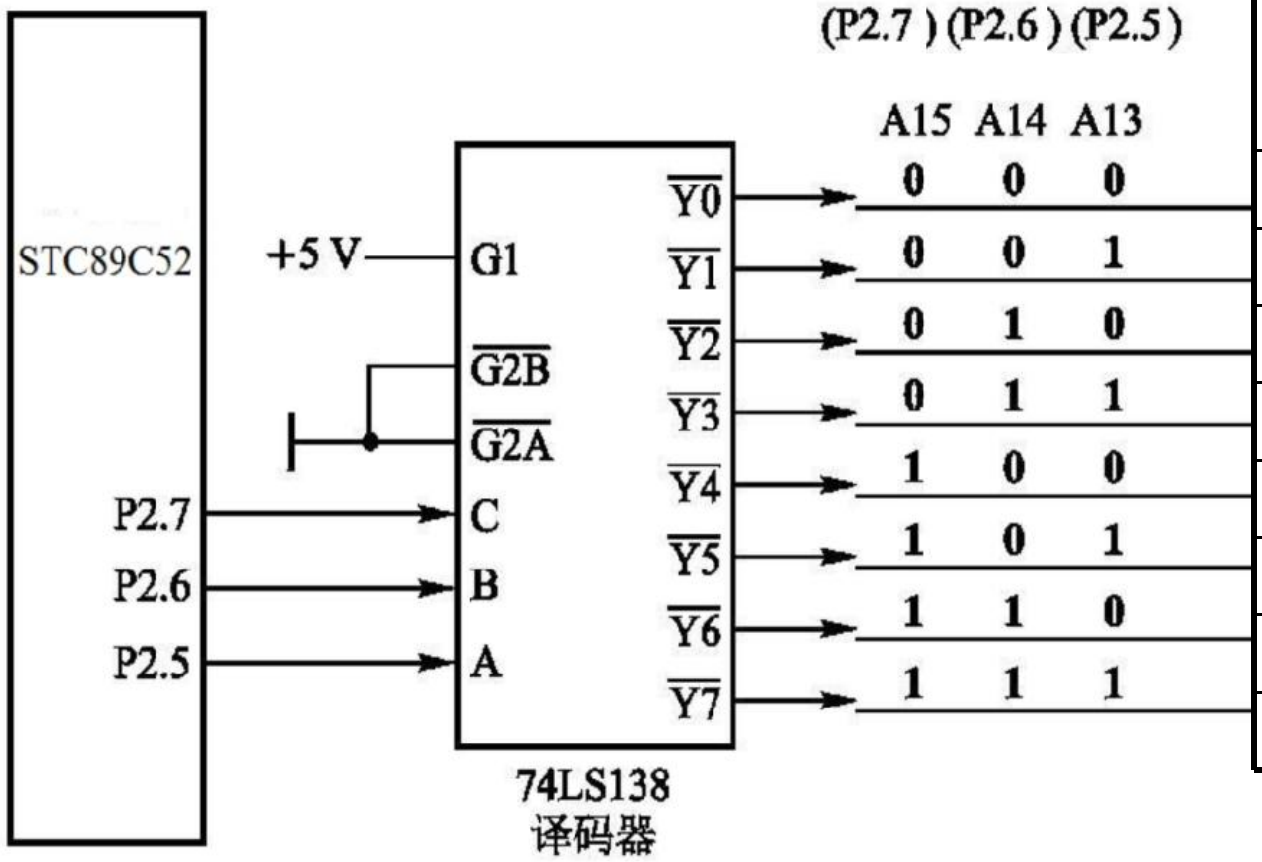
```
        ORG    0000H
        MOV    DPTR,#TAB      ;要传数据的首地址#TAB送入DPTR
        MOV    R0,#0          ;R0的初始值为0，也是RAM地址低字节
        MOV    P2,#40H        ;片外存储单元地址高字节
AGIN：  MOV    A,R0
        MOVC   A,@A+DPTR      ;把以TAB为首址64个单元内容送入A
        MOVX   @R0,A          ;程序存储器中表的内容送入外部RAM单元
        INC    R0             ;循环次数加1，也即外部RAM单元的地址指针加1
        CJNE   R0,#64,AGIN    ;判64个单元的数据是否已经传送完
        SJMP   $
TAB:    DB  01H,02H,……       ;外部程序存储器中要传送的64个单元的内容
        END
```

问：两电路中/Y0~/Y7输出的地址范围分别是多少？



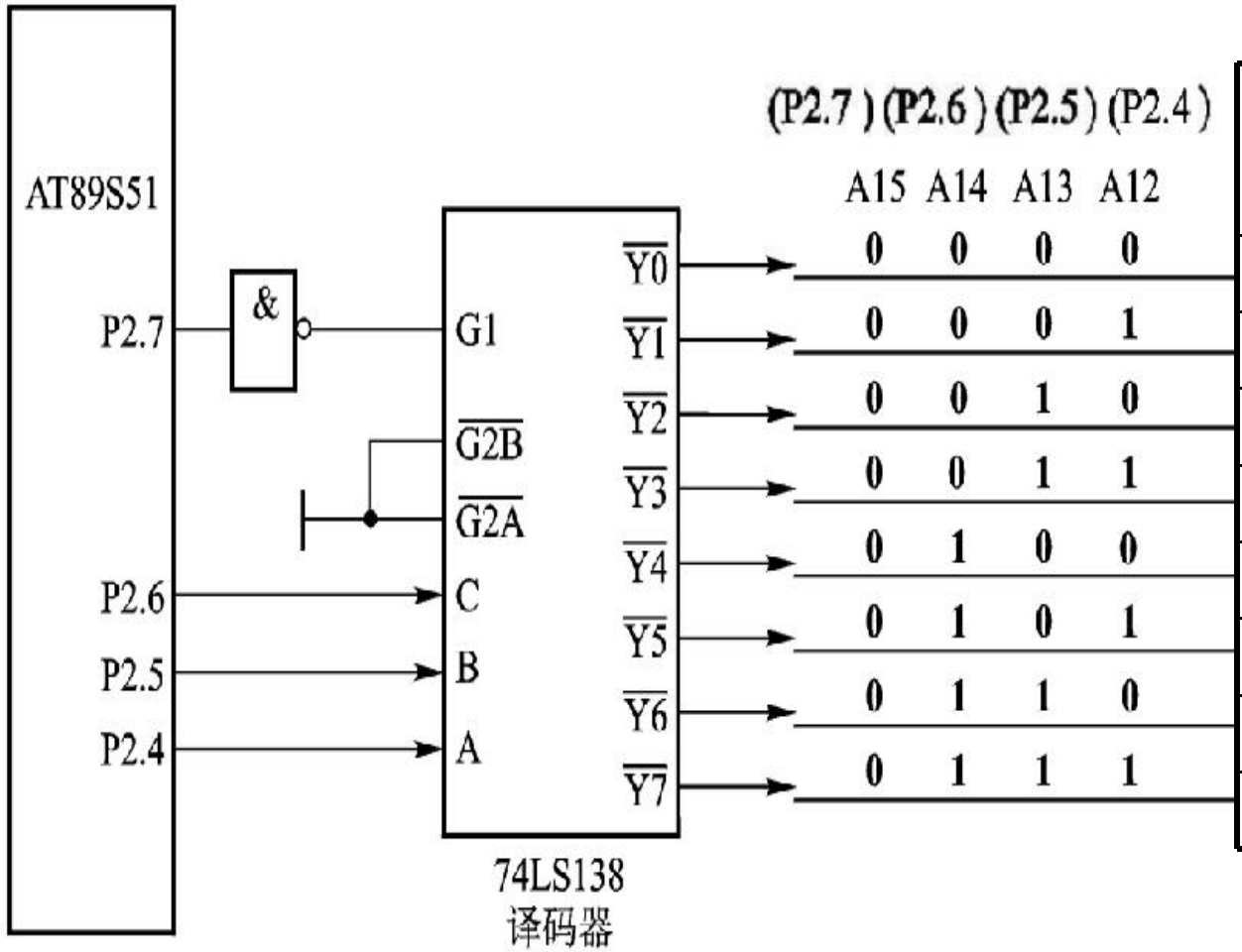


问：电路中138扩展存储器的总空间是多少KB？



A15…… A8 A7……A0 P2.7 P2.6 P2.5 **P2.4~P2.0** **P0.7~P0.0** C B A					地址范围
0 0 0	* ****	****	****		0000H~1FFFH
0 0 1	* ****	****	****		2000H~3FFFH
0 1 0	* ****	****	****		4000H~5FFFH
0 1 1	* ****	****	****		6000H~7FFFH
1 0 0	* ****	****	****		8000H~9FFFH
1 0 1	* ****	****	****		A000H~BFFFH
1 1 0	* ****	****	****		C000H~DFFFH
1 1 1	* ****	****	****		E000H~FFFFH

问：电路中138扩展存储器的总空间是多少KB？



A15……A8 P2.7 **P2.6 P2.5 P2.4** P2.3~P2.0 0 **C B A**					A7……A0 P0.7~P0.0		地址范围
0	0	0	0	****	****	****	0000H~0FFFH
0	0	0	1	****	****	****	1000H~1FFFH
0	0	1	0	****	****	****	2000H~2FFFH
0	0	1	1	****	****	****	3000H~3FFFH
0	1	0	0	****	****	****	4000H~4FFFH
0	1	0	1	****	****	****	5000H~5FFFH
0	1	1	0	****	****	****	6000H~6FFFH
0	1	1	1	****	****	****	7000H~7FFFH

STC89C52存储器扩展 要点

要点:

- 存储器扩展三总线的连接特点
- 地址空间分配：线选法和译码法
- 程序和数据存储器的扩展的区别
- 程序和数据存储器的综合扩展应用