

# 第六章 接口技术

## 接口基础



# 学习目标

- 了解接口的基本构成、数据传送方式以及控制方式
- 理解不同接口寻址方式的特点
- 理解接口译码原理、掌握接口译码电路的设计
- 掌握接口设计的一般原则



# 一、接口的基本概念

- 接口是一组电路，是中央处理器与存储器、输入输出设备等设备之间协调动作的控制电路。
- 并不局限在中央处理器与存储器或外设之间，也可在存储器与外设之间



# 接口功能

- 任何接口通常都包括以下5种功能：
  - 控制和定时
  - 与微处理器通信
  - 与外设通信
  - 数据缓冲
  - 错误检测

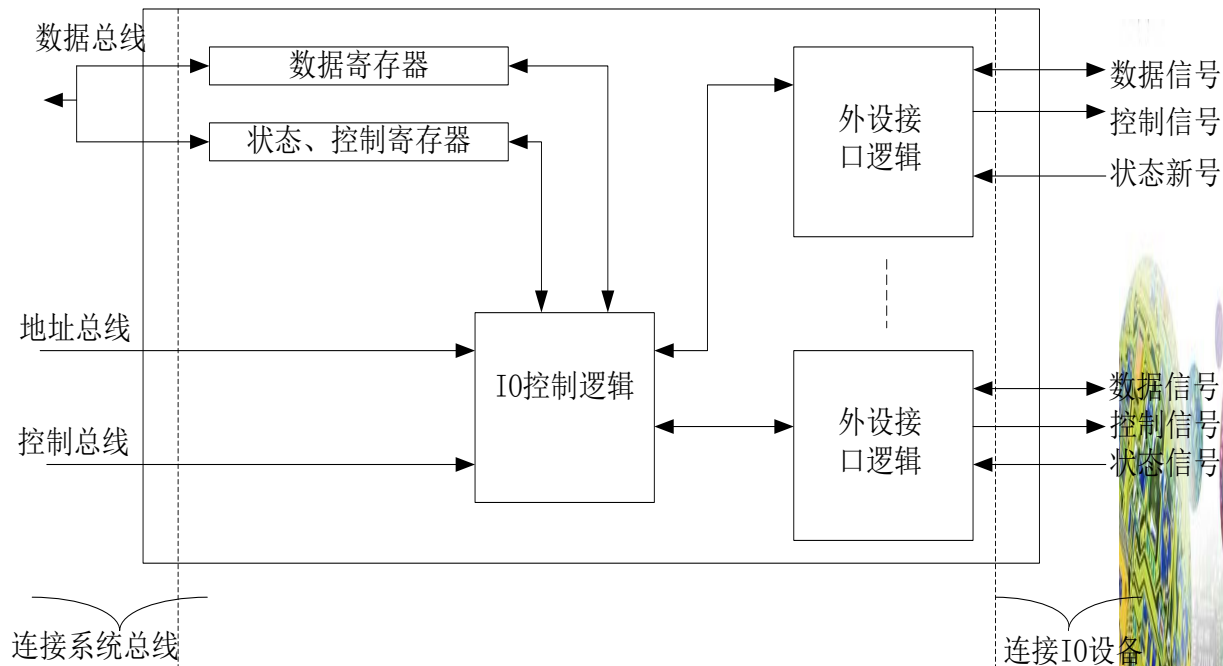


- 微处理器与IO接口之间通信以及IO设备与IO接口之间通信包括以下几个方面：
  - 命令译码
  - 数据交换
  - 状态反馈
  - 地址译码

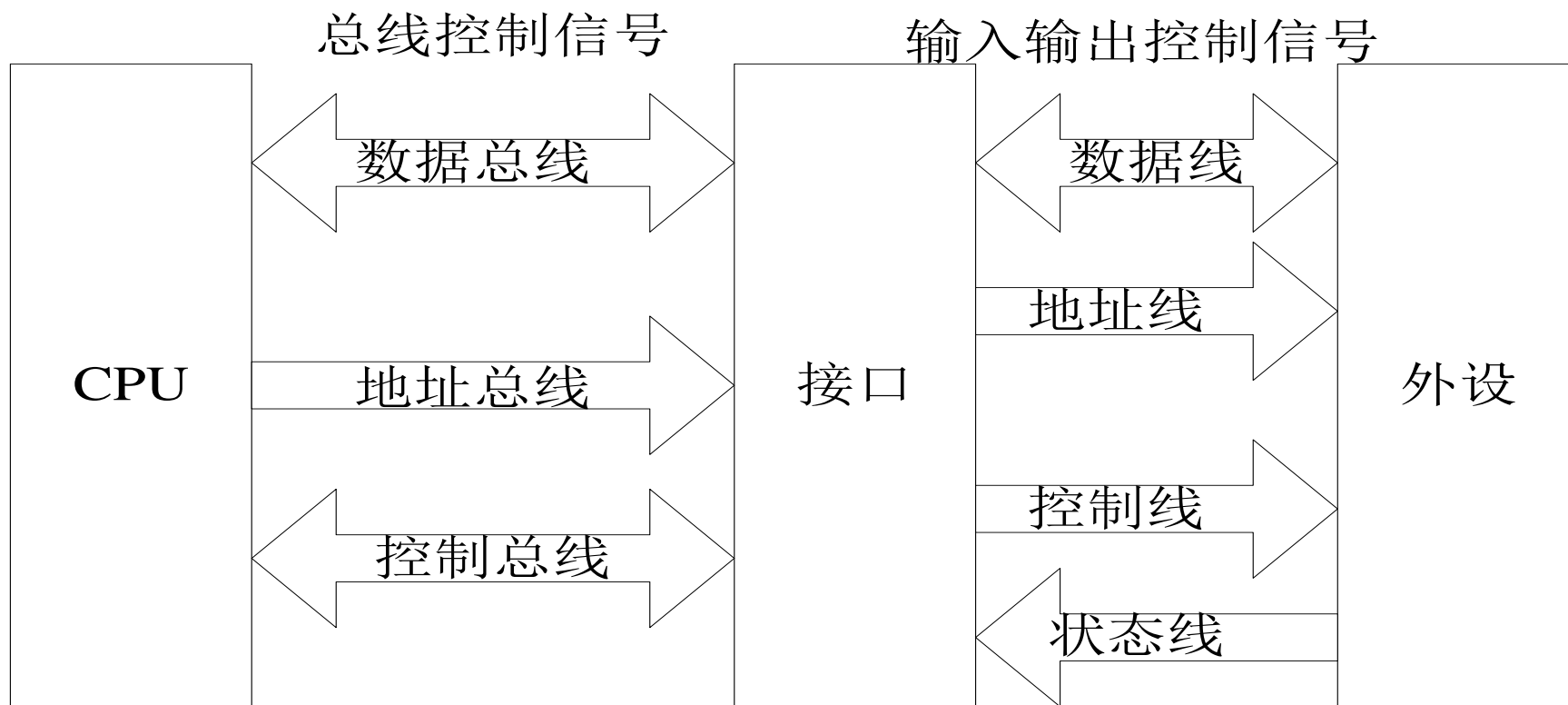


## 二、接口基本结构

- 控制逻辑电路
- 状态设置和存储电路
- 数据存储和缓冲电



# 接口信号





### 三、接口与外设间的数据传送方式

- 并行数据传送
  - 并行数据的每一位都对应独立的传输线路
- 串行数据传送
  - 将构成字符的每个二进制数据位，按一定的顺序逐位进行传送





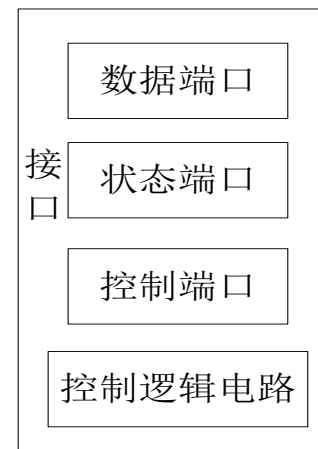
# 接口控制方式

- 查询
  - 在数据传送之前通过接口的状态设置存储电路询问外设，待外设允许传送数据后才传送数据的操作方式
  - “无条件”传送方式
- 中断
  - 外设要与中央处理器传送数据时，外设向中央处理器发出请求，中央处理器响应后再传送数据
- DMA
  - 数据不经过中央处理器在存储器和外设之间直接传送



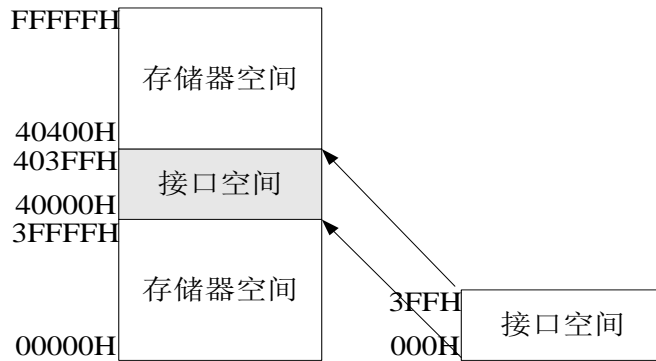
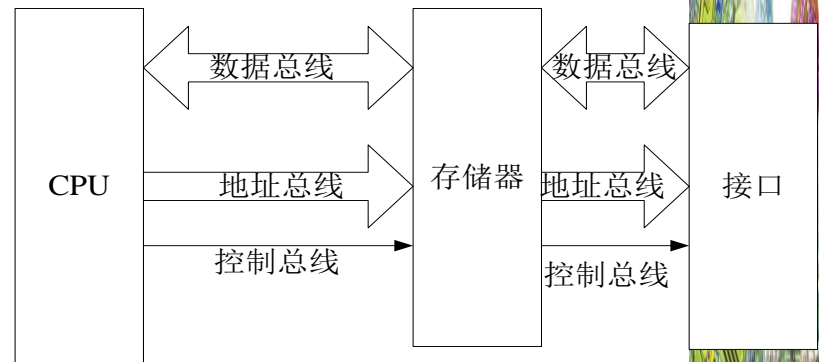
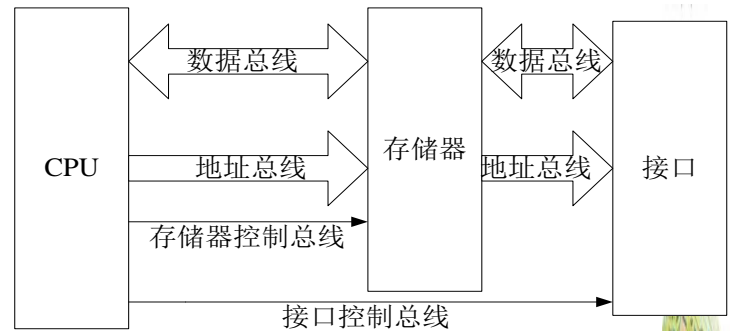
## 四、IO接口寻址方式

- 接口基本结构中的寄存器叫做端口
- 根据寄存器的不同功能，把这些寄存器分别叫做控制端口、状态端口和数据端口



# IO接口结构

- 标准（独立）I/O接口结构计算机系统
- 存储器映像I/O接口结构计算机系统



# 标准I/O 寻址方式特点

- I/O设备的地址空间和存储器地址空间是独立的、分开的。即I/O接口地址不占用存储器地址空间。
- 微处理器对I/O设备的管理是利用专用的IN(输入)和OUT(输出)指令来实现数据传送的。
- CPU对I/O设备的读/写控制是用I/O读/写控制信号( $\overline{IOR}$ 、 $\overline{IOW}$ )。
- 微机系统的微处理器都采用标准I/O寻址方式



# 存储器映像I/O寻址方式

- I/O接口与存储器共用同一个地址空间。
  - = I/O设备的每一个寄存器占用存储器空间的  
一个地址。
  - 存储器与I/O设备之间的唯一区别是其所占用的地址不同。
  - 存储器与I/O设备之间的唯一区别是其所占用的地址不同。
- CPU利用对存储器存储单元进行操作的指令来实现对I/O设备的管理。
- CPU利用对存储器存储单元进行操作的指令来实现对I/O设备的管理。
- 实现用存储器读/写控制信号( $\overline{MEMR}$ 、 $\overline{MEMW}$ )对I/O设备进行读/写控制。
- CPU用存储器读/写控制信号(、)对I/O设备进行读/写控制。
- 嵌入式微处理器基本上都采用存储器映像IO寻址方式。
- 嵌入式微处理器基本上都采用存储器映像IO寻址方式。

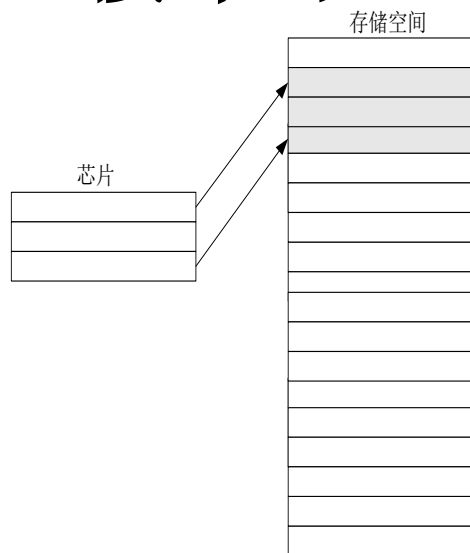


## 五、接口译码原理

- 直接译码
  - 直接采用地址总线来实现存储单元选址的译码方式
- 间接译码
  - 利用专门地址端口来实现地址译码

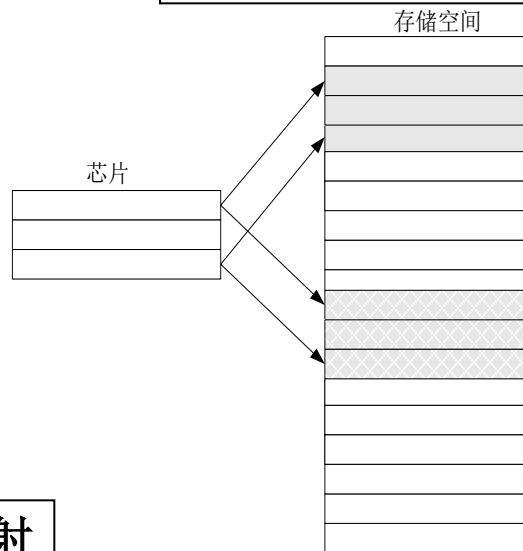


# 直接译码

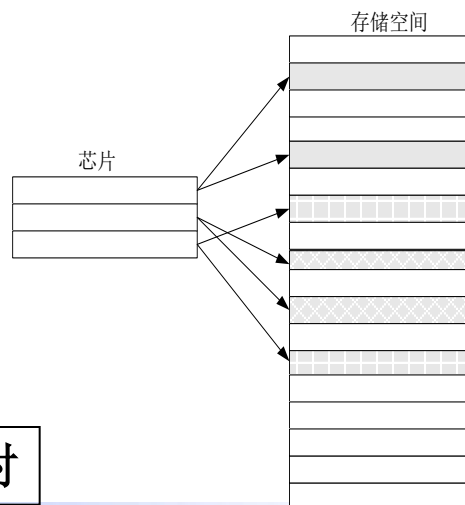


一对一整体映射

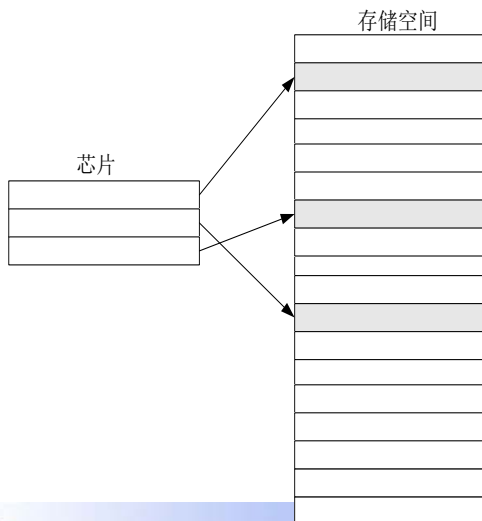
一对多整体映射



一对多离散映射



一对一离散映射





# 一对一整体映射举例

- 系统地址总线所有剩余高位地址都必须经过译码之后才能连接到芯片的片选控制端叫做**全译码法**。
- 例6.1 将容量为256K的存储芯片一对一整体映射到存储空间为4G的计算机存储系统中，且要求映射地址范围为0xfff40000~0xfff7ffff，如何实现地址译码？

- 256K的存储芯片具有18位地址总线：A0~A17。
- 4G存储空间的计算机系统具有32位地址总线：A0~A31。其中A0~A17与存储芯片的18位地址总线A0~A17直接相连。
- 剩余的高位地址线为A18~A31，由于地址范围为0xfff40000~0xfff7ffff。由此可知在整个地址范围内A19为0，A18,A20~A31都为1。因此需要将A18~A31译码产生存储芯片片选信号，假设片选信号为CS，且低电平有效，那么
- $\overline{CS} = A19 + A18 \cdot A20 \cdot A21 \cdot A22 \cdot A23 \cdot A24 \cdot A25 \cdot A26 \cdot A27 \cdot A28 \cdot A29 \cdot A30 \cdot A31$

# 一对多整体映射举例

- 部分高位地址线没有参与译码形成芯片片选控制信号。
  - 仅一位高位地址线参与译码，就叫做**线选法**；
  - 是多位但不是全部高位地址线参与译码，就叫做**部分译码法**。



- 例6.2将容量为256K的存储芯片一对多整体映射到存储空间为4G的计算机存储系统中，且要求地址范围为0xfff00000~0xfff3ffff或0xfff40000~0xfff7ffff，该如何实现地址译码？

$A_{18}: 0, 1$

$A_{19}: 0$

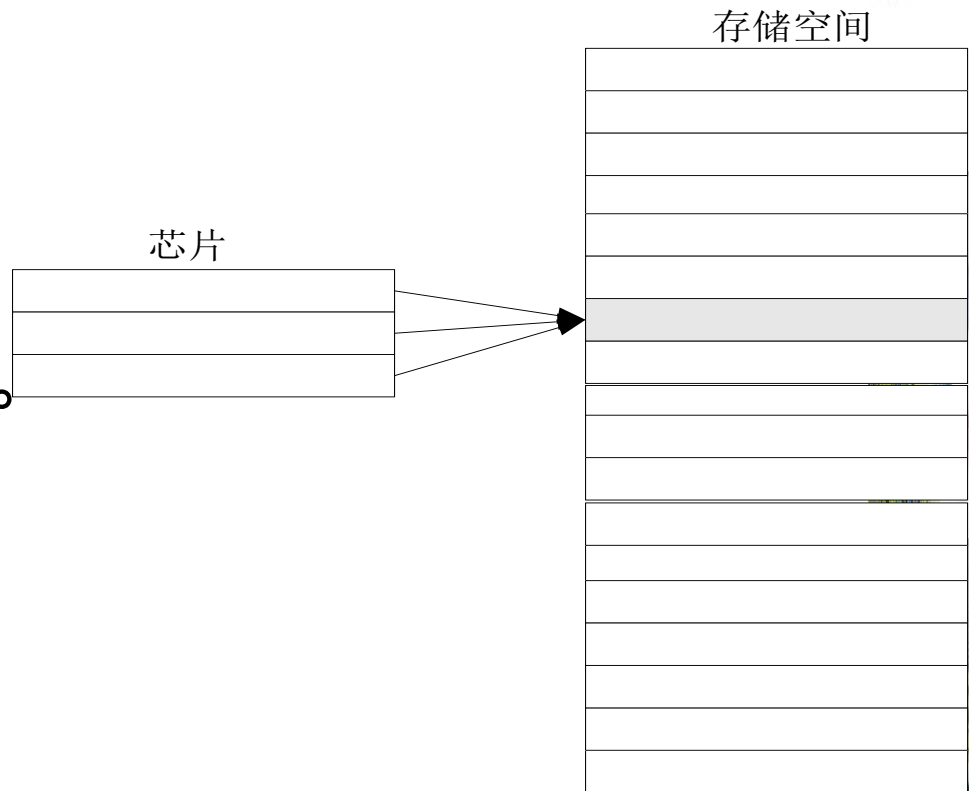
$A_{20} \sim A_{31}: 1111 \ 1111 \ 1111$

$$\overline{CS} = \overline{A_{19} + A_{20}A_{21}A_{22}A_{23}A_{24}A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}}$$

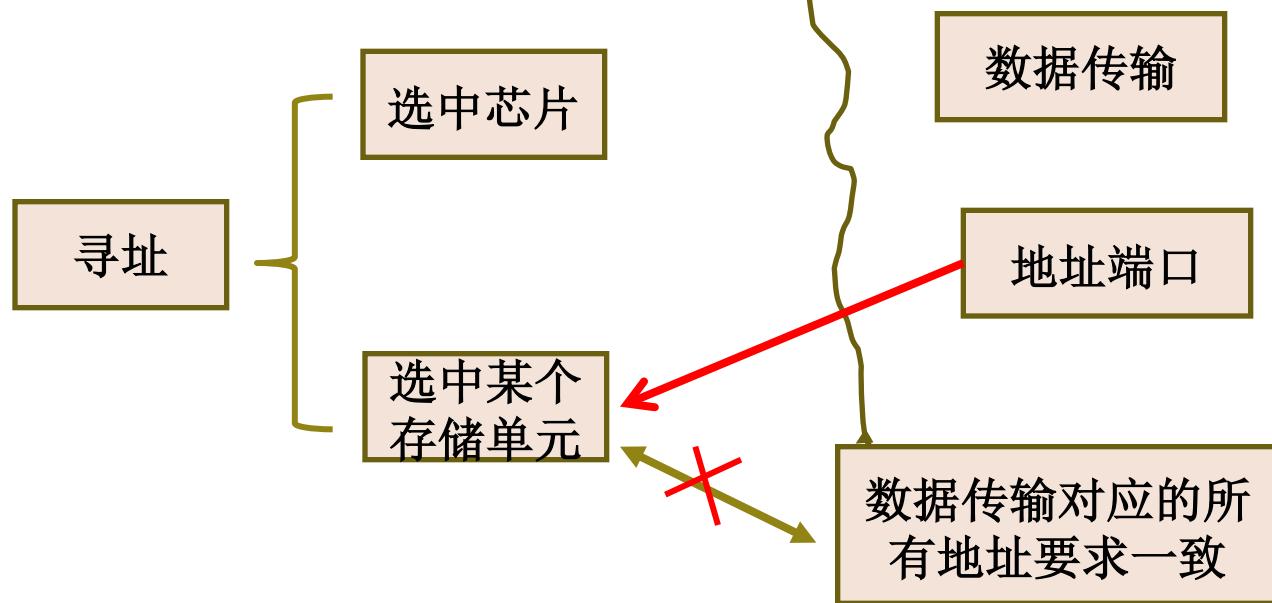


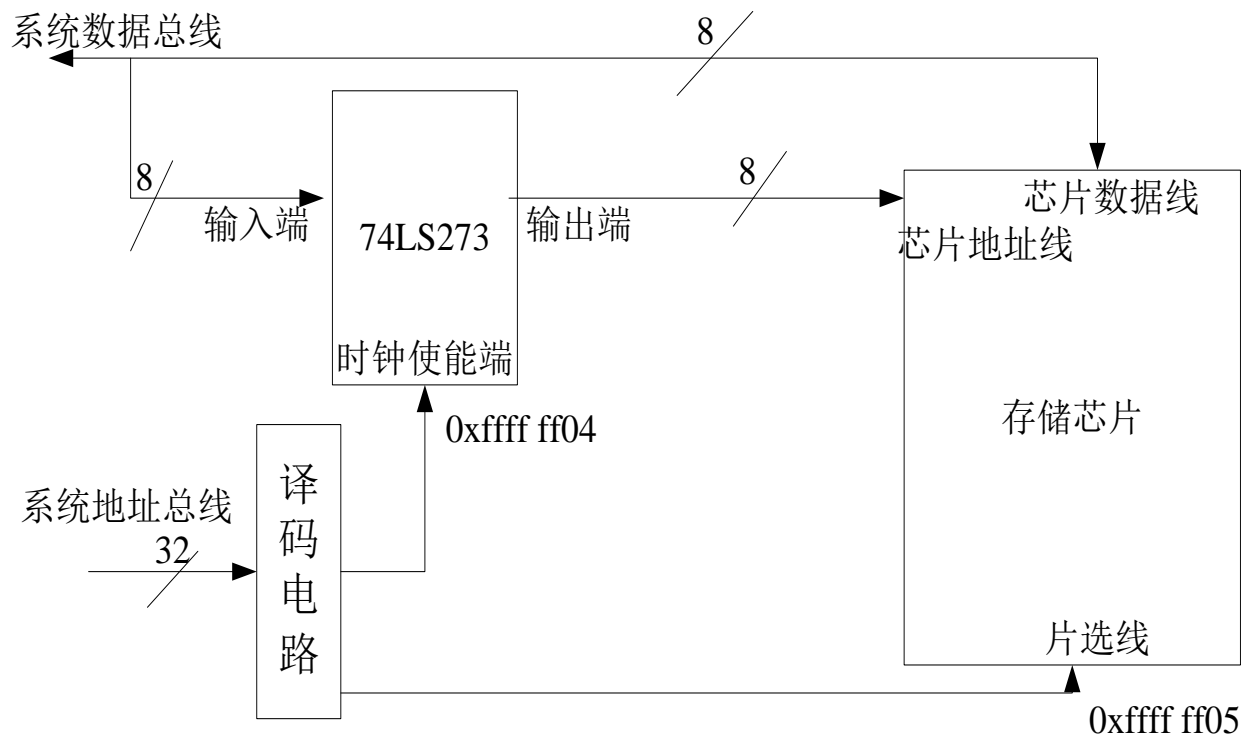
# 多对一整体映射

- 微处理器给出的一个地址对应了多个物理存储空间，必须采用IO接口来实现芯片内各个存储单元的访问。该类接口包含一个数据端口和一个地址端口



- 例6.3 一个具有256个存储单元的芯片，要求将其所有的存储单元都映射到具有32位地址总线、32位数据总线的计算机系统存储空间中的一个存储单元中。假设该存储单元地址为0xffff ff05，并要求能对芯片内任意一个存储单元进行访问，试设计接口。





$$Y = \prod_{k=8}^{31} A_k$$

$$\text{CLK} = \overline{Y \cdot \overline{A7} \cdot \overline{A6} \cdot \overline{A5} \cdot \overline{A4} \cdot \overline{A3} \cdot A2 \cdot \overline{A1} \cdot \overline{A0}}$$

$$\overline{\text{CS}} = \overline{Y \cdot \overline{A7} \cdot \overline{A6} \cdot \overline{A5} \cdot \overline{A4} \cdot \overline{A3} \cdot A2 \cdot \overline{A1} \cdot A0}$$





# 采用MIPS汇编指令来实现对存储芯片0x80单元字节类型数据的访问

*lui \$s0, 0xffff*

*ori \$t0, \$s0, 0xff04* #将地址端口的地址0xffff ff04保存到\$t0中

*addi \$s1, \$zero, 0x80* #将芯片存储单元地址0x80保存到寄存器\$s1中

*sbu \$s1, 0(\$t0)* #将0x80输出到地址端口，锁存在74LS273的输出端

*ori \$t0, \$s0, 0xff05* #将数据端口的地址0xffff ff05保存到\$t0中

*lbu \$s2, 0(\$t0)* #从数据端口中读取无符号字节类型数据



# 采用Xilinx C语言来实现

```
int AddressPort=0xffffffff04;
```

```
int DataPort=0xffffffff05;
```

```
Xil_Out8(AddressPort,0x80);
```

```
char data=Xil_In8(DataPort);
```

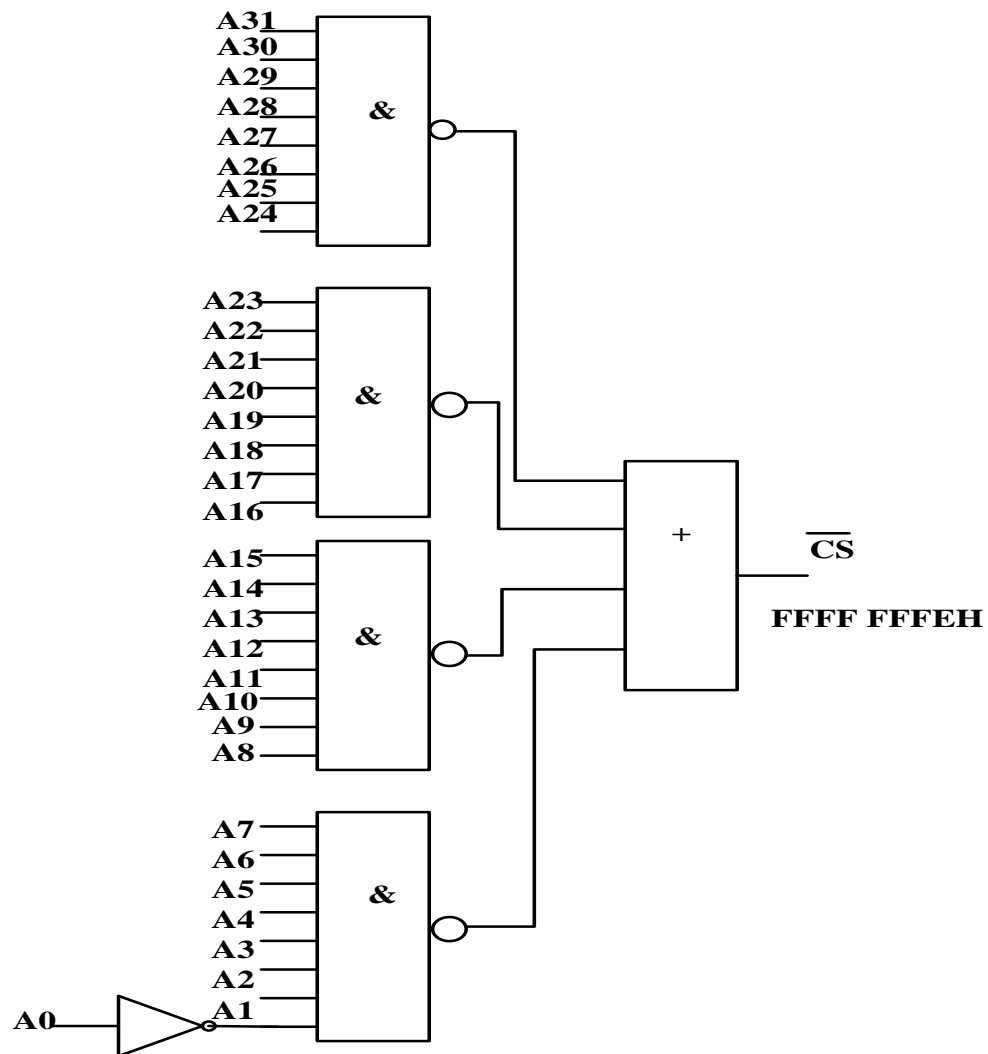


# 接口译码电路

- 逻辑门电路
  - AND ,NAND ,OR ,NOR...
- 专用译码芯片
  - 138, ...
- 可编程逻辑器件
  - CPLD,GAL,FPGA

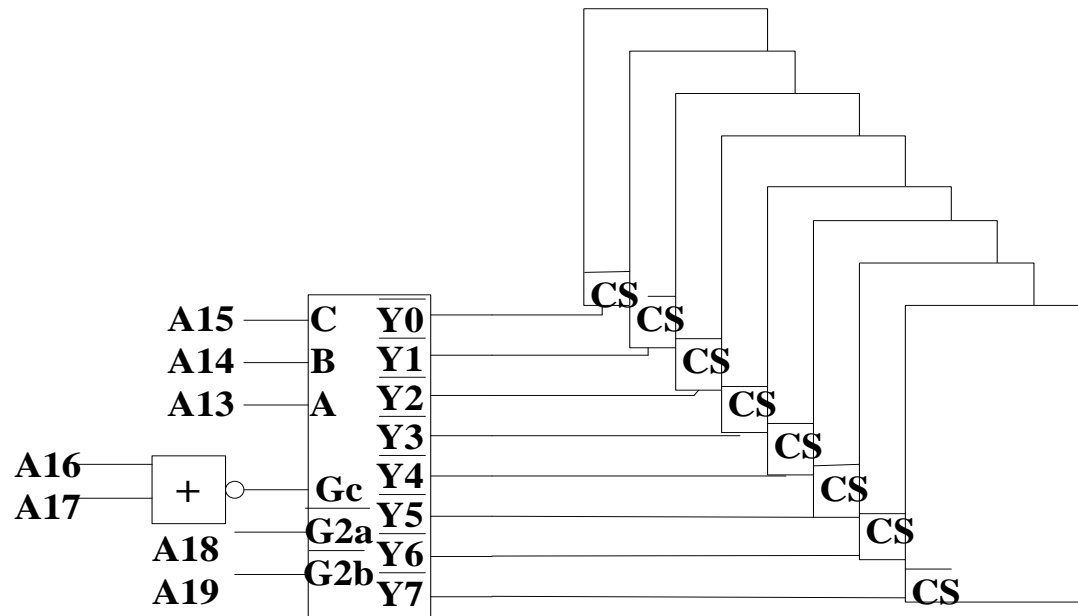


# 逻辑门



# 138译码器

输入						输出							
允许			选择										
Gc	G2b	G2a	C	B	A	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
1	0	0	1	1	1	0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
0	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1

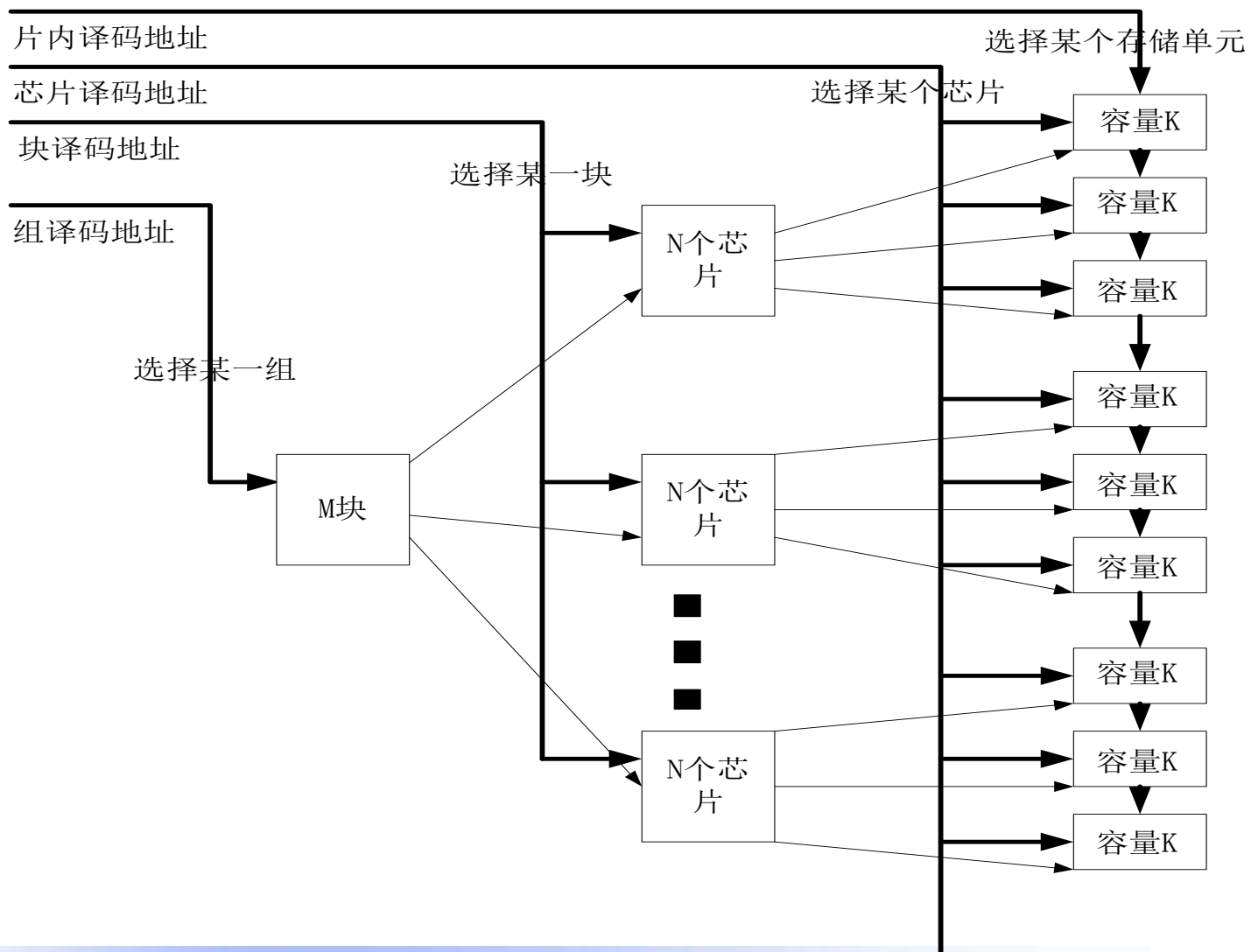


# FPGA译码

```
module DECODER(  
    input [19:13] A, //输入地址信号  
    output [7:0] CS //输出片选信号  
);  
reg [7:0] CS1; //设置输出寄存器  
    assign CS[7:0]=CS1[7:0]; //输出引脚与寄存器相连  
    always @(A)  
begin  
    if (A[19:16] != 4'b0000)  
        CS1[7:0] <= 8'b11111111; //A[19:16] != 0000, 所有CS无效  
    else  
        case (A[15:13])  
            3'b000 : CS1[7:0] <= 8'b11111110;  
            3'b001 : CS1[7:0] <= 8'b11111101;  
            3'b010 : CS1[7:0] <= 8'b11111011;  
            3'b011 : CS1[7:0] <= 8'b11110111;  
            3'b100 : CS1[7:0] <= 8'b11101111;  
            3'b101 : CS1[7:0] <= 8'b11011111;  
            3'b110 : CS1[7:0] <= 8'b10111111;  
            3'b111 : CS1[7:0] <= 8'b01111111;  
        endcase  
    end  
endmodule
```



# 分级译码



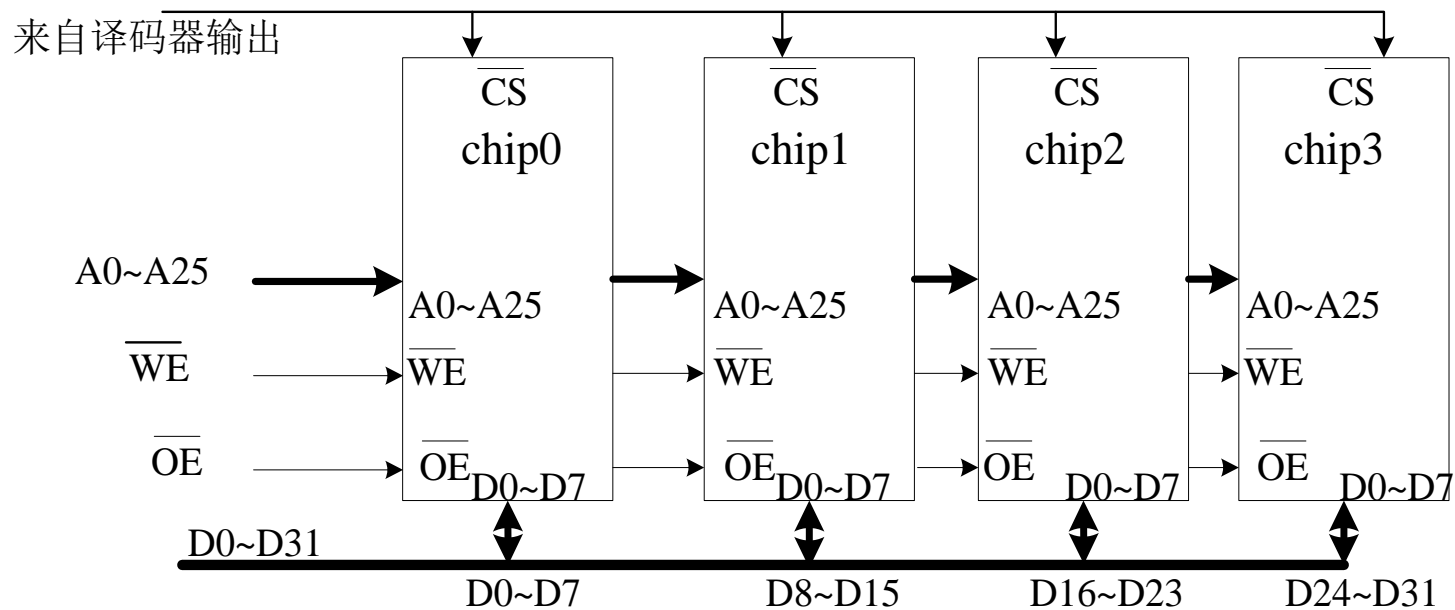
# 接口总线连接原则

- 地址总线连接原则
  - 地址总线中的低位地址总线与接口的地址总线相连，
  - 地址总线中剩余的**高位地址总线**通过译码电路**译码**之后连接到接口的控制信号上
- 数据总线连接原则
  - 接口位宽与数据总线位宽**一致**，那么仅需要将接口数据线与数据总线**对应位直接相连**即可
  - 接口位宽**大于**数据总线位宽，在数据总线宽度一定的情况下，需要**复用部分数据总线**来传输高位接口数据，
  - 8位数据总线接口芯片来构建更高位宽的接口，则需要多个同类型接口芯片采用并联方式，即**字长扩展**。由1位，2位，4位等不同位宽芯片来构建8位宽整数倍的接口同样也是采用这种方式。

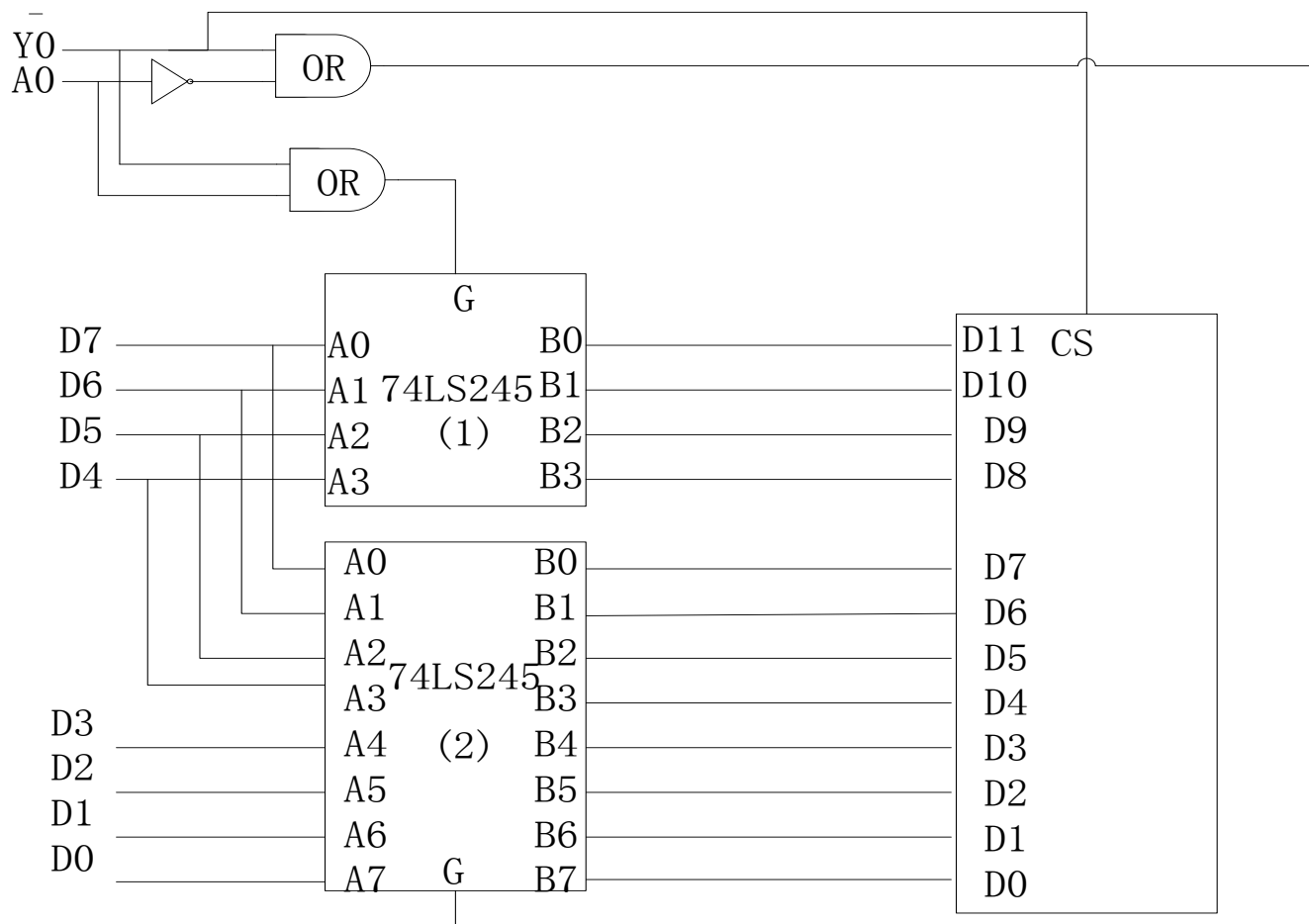




# 字长扩展多芯片并联



# 复用数据总线



# 通过8位数据总线输出12位数据

```
int data=0x789;
```

```
int port0=0x378,port1=0x379;
```

```
char byte0,byte1;
```

```
byte0=(char) data; //byte0=0x89
```

```
data=data>>4; //data=0x0078
```

```
byte1=(char)(data) ; //byte1=0x78
```

```
Xil_Out8(port0,byte1); //输出0x78, 但是低4位无效
```

```
Xil_Out8(port1,byte0); //输出0x89
```



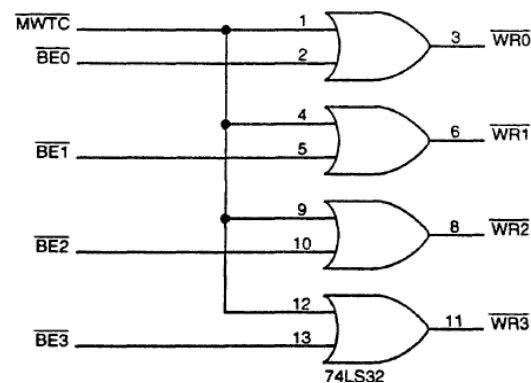
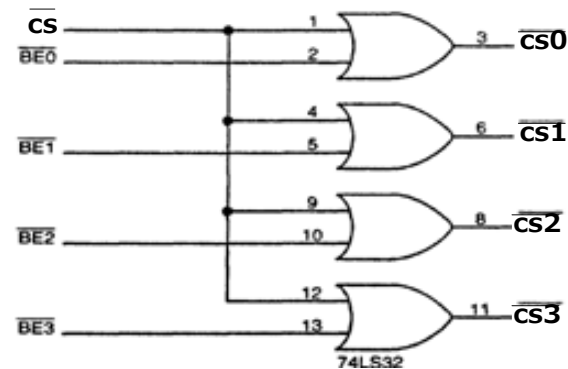
# 控制总线连接原则

- 接口控制总线主要有读写控制信号、片选信号等信号
  - 独立IO结构计算机系统中，则需分别对IO接口和存储器接口的读写控制信号与总线提供的相对应读写控制信号相连：如IO接口的读写控制信号与微处理器提供的 $\overline{IOR}$ ， $\overline{IOW}$ 相连；存储器接口的读写控制信号与总线提供的 $\overline{MEMR}$ ， $\overline{MEMW}$ 相连
  - 存储器映像IO结构，不需要区分是IO接口还是存储器接口，只需要直接将接口的读写控制信号与总线提供的读写控制信号相连
- 片选信号通常来自高位地址译码



# 不同位宽接口兼容

- 高位地址与字节使能信号译码产生不同的片选控制信号
- 写控制信号与字节使能信号译码产生不同的字节写控制信号





# 作业

- 4,
- 5

