Lecture 7: Memory Address Decoding

Prof. Xiangzhong FANG
xzfang@sjtu.edu.cn

The 80x86 IBM PC and Compatible Computers

Chapter 10

Memory and Memory Interfacing

Chapter 11

I/O and the 8255

Terms about Memory - Revisit

- Capacity: how many bits that a memory module contains
  - E.g., 64M (bits)
- Organization: how many bits can be accessed simultaneously (a memory unit, memory word) & thus how many locations of a memory module
  - E.g., 64M X 1, 16M X 4, 8M X 8
  - 2^*a* X *d*
- Access time: how long does it take from putting the address on the address pins to getting the data from the data pins

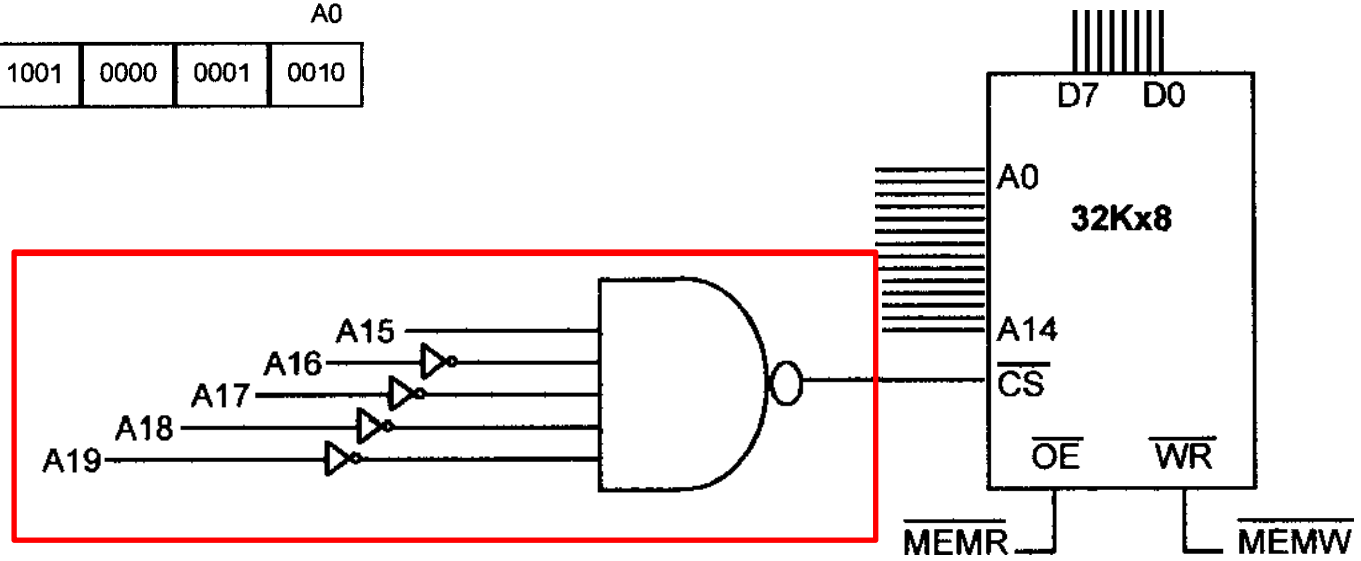
Memory Address Decoding

- According to **your instructions** that access memory
  - E.g., MOV AX, [0012H]
- **CPU** calculates the physical address and put corresponding signals on the address bus
  - E.g., if DS=0900H, *what's the PA?*
- **Memory address decoding circuitry** locates the specific memory chip that stores the desired data
  - Examine address decoding using logic gates and 74LS138 decoder chips

Memory Address Decoding

PA of 0900:0012 = 09012H

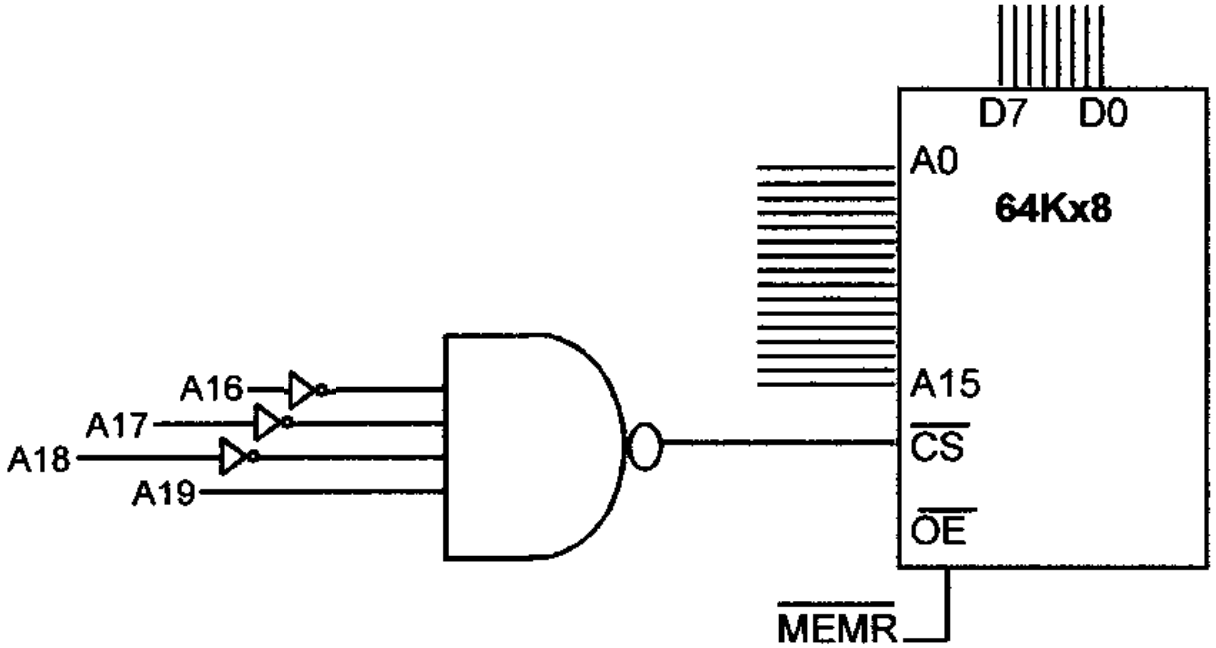
A19				A0
0000	1001	0000	0001	0010



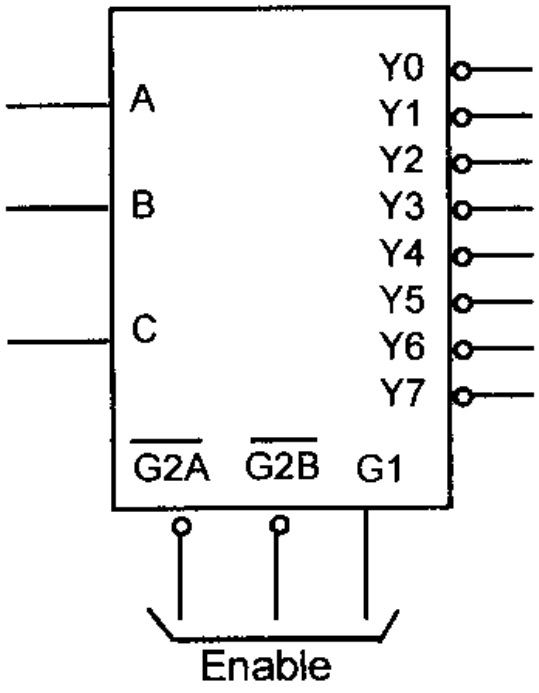
A19				A0	
0000	1000	0000	0000	0000	=08000H
0000	1111	1111	1111	1111	= 0FFFFH

Quiz

What's the address range?



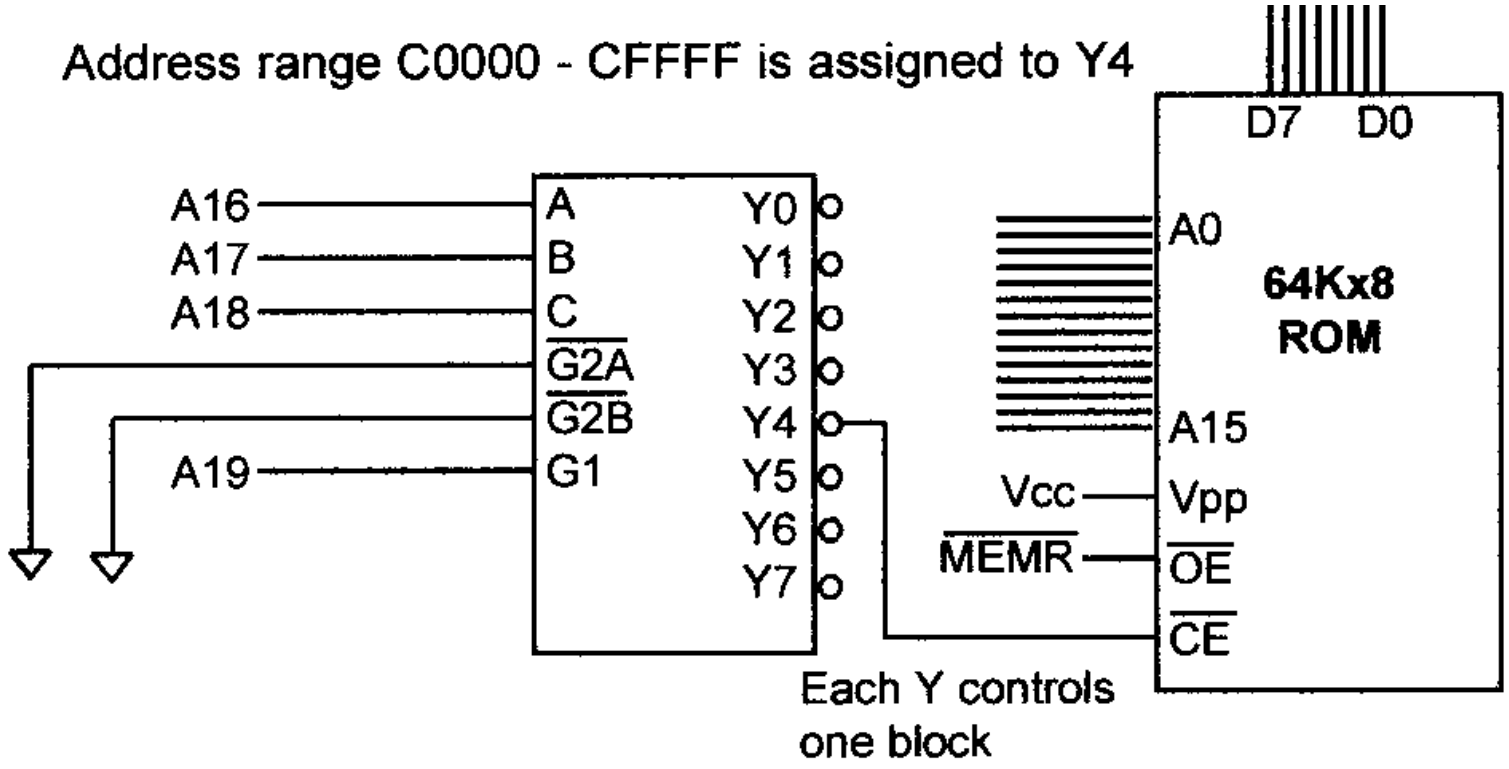
The 74LS138 Decoder Chip



Function Table

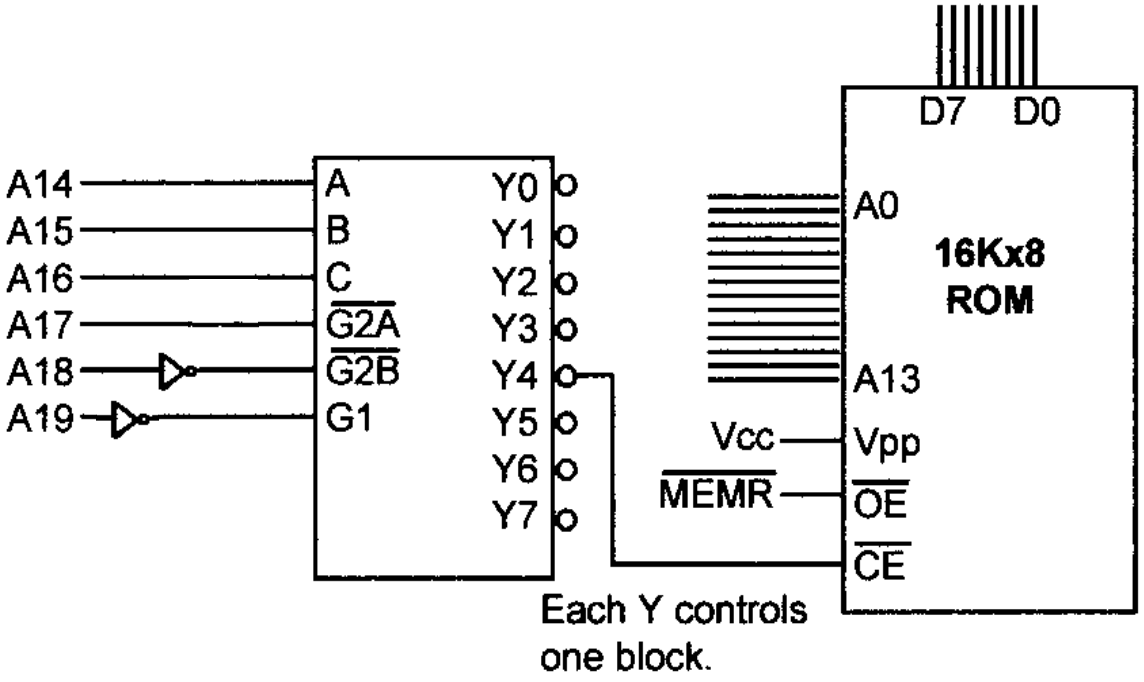
Inputs			Outputs
Enable		Select	
G1	G2	C B A	Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7
X	H	X X X	H H H H H H H H
L	X	X X X	H H H H H H H H
H	L	L L L	L H H H H H H H
H	L	L L H	H L H H H H H H
H	L	L H L	H H L H H H H H
H	L	L H H	H H H L H H H H
H	L	H L L	H H H H L H H H
H	L	H L H	H H H H H L H H
H	L	L L L	H H H H H H L H
H	L	H H H	H H H H H H H L

Using 74LS138 to Decode



Quiz 2

What's the address range for Y4 and Y7?

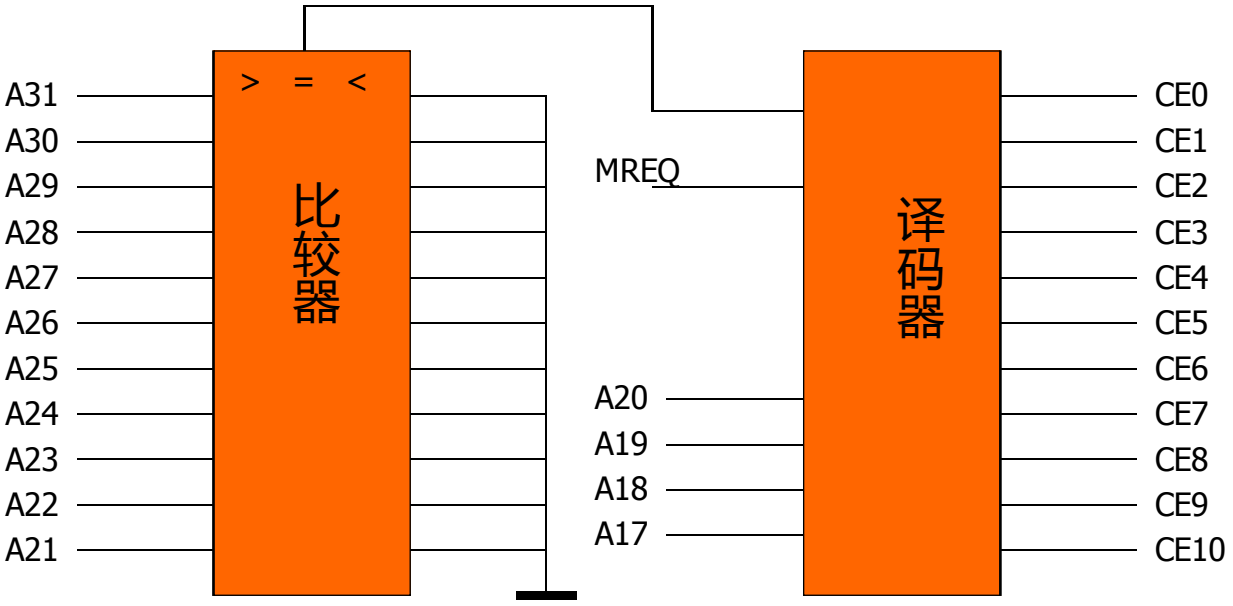


More on Address Decoding

- Absolute address decoding
  - All address lines are decoded
- Linear select decoding
  - Only selected lines are decoded
  - Cheap
  - But with *aliases:* the same port (memory unit) with multiple addresses
    - *Why this happens?*

全译码的实现

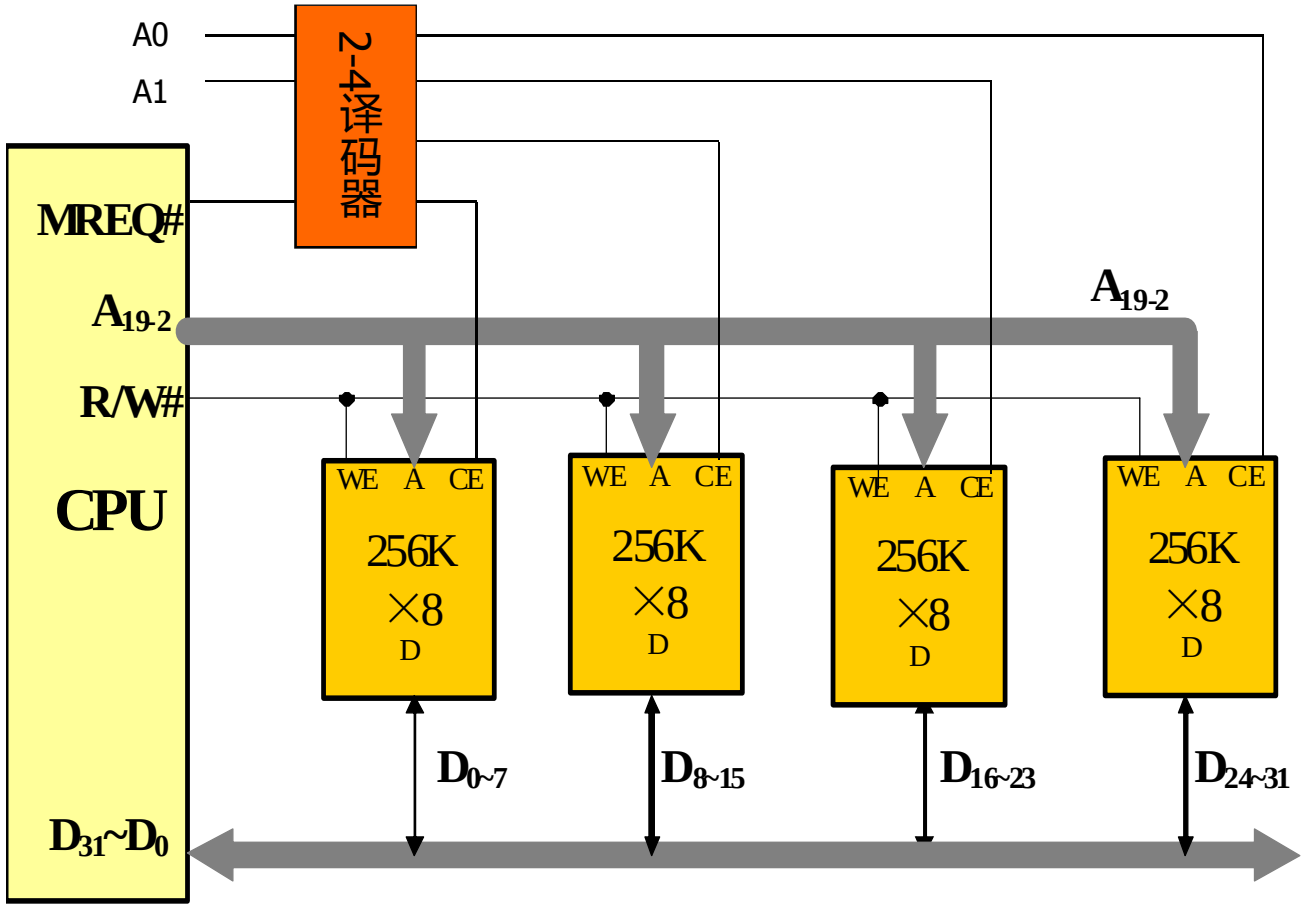
- 与门（常用）
- 地址比较器



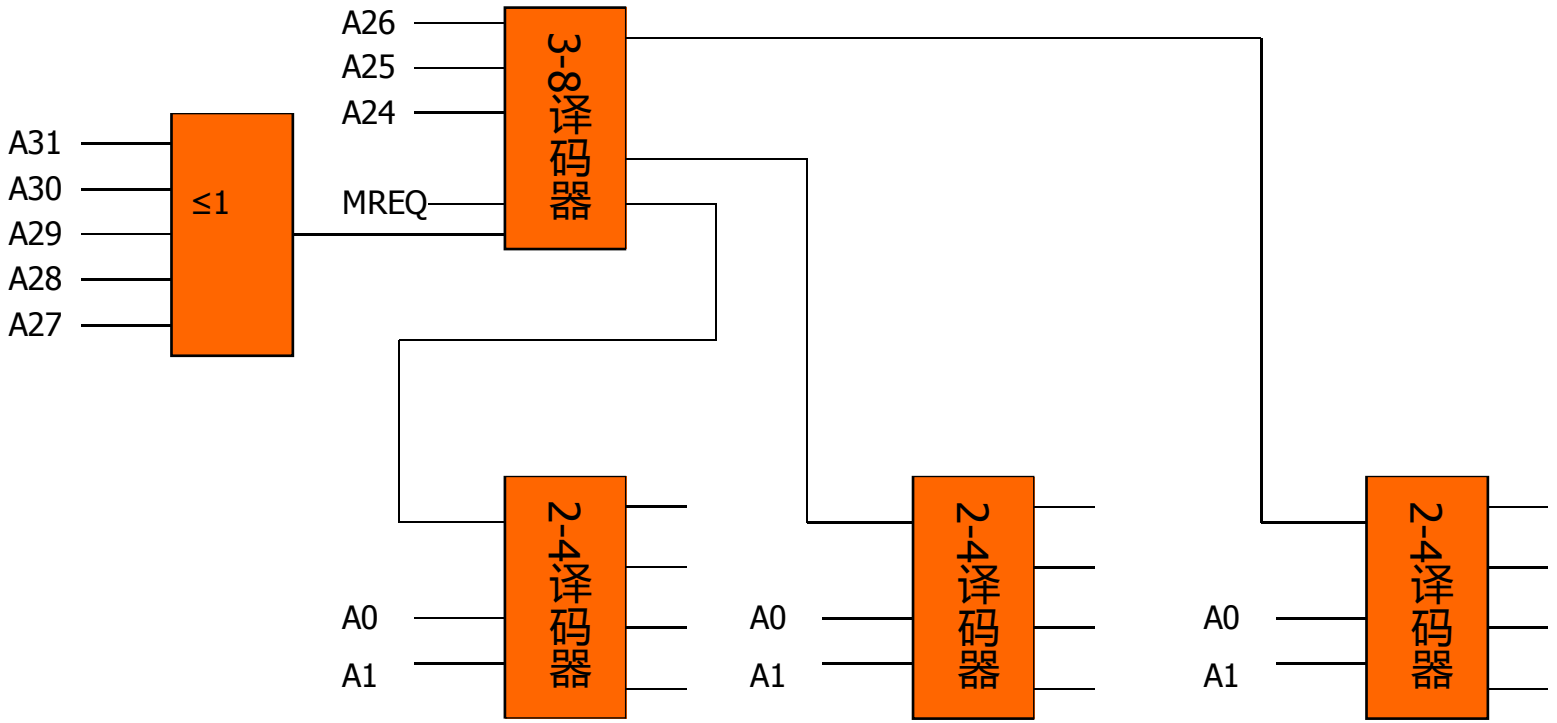
字选择与字节选择

- **字选择**
  - 以字为单位访问存储器
  - 位数等于存储器的字长
  - 低位地址不需要
- **字节选择**
  - 可以以字节为单位访问存储器
  - 低位地址用于选择字节

字节选择的实现


A0
A1
2-4译码器
MREQ#
A19-2
R/W#
CPU
D31~D0
A19-2
WE A CE
256K
×8
D
WE A CE
256K
×8
D
WE A CE
256K
×8
D
WE A CE
256K
×8
D
D0~7
D8~15
D16~23
D24~31

字位扩展中的字节选择


A26
A25
A24
3-8译码器
A31
A30
A29
A28
A27
≤1
MREQ
2-4译码器
A0
A1
2-4译码器
A0
A1
2-4译码器
A0
A1

例1 某计算机的主存地址空间中，从地址0000_{16}到$3FFF_{16}$为ROM存储区域，从4000_{16}到$5FFF_{16}$为保留地址区域，暂时不用，从6000_{16}到$FFFF_{16}$为RAM地址区域。RAM的控制信号为CS#和WE#，CPU的地址线为A15~A0，数据线为8位的线路D7~D0，控制信号有读写控制R/W#和访存请求MREQ#，要求：

(1) 画出地址译码方案。

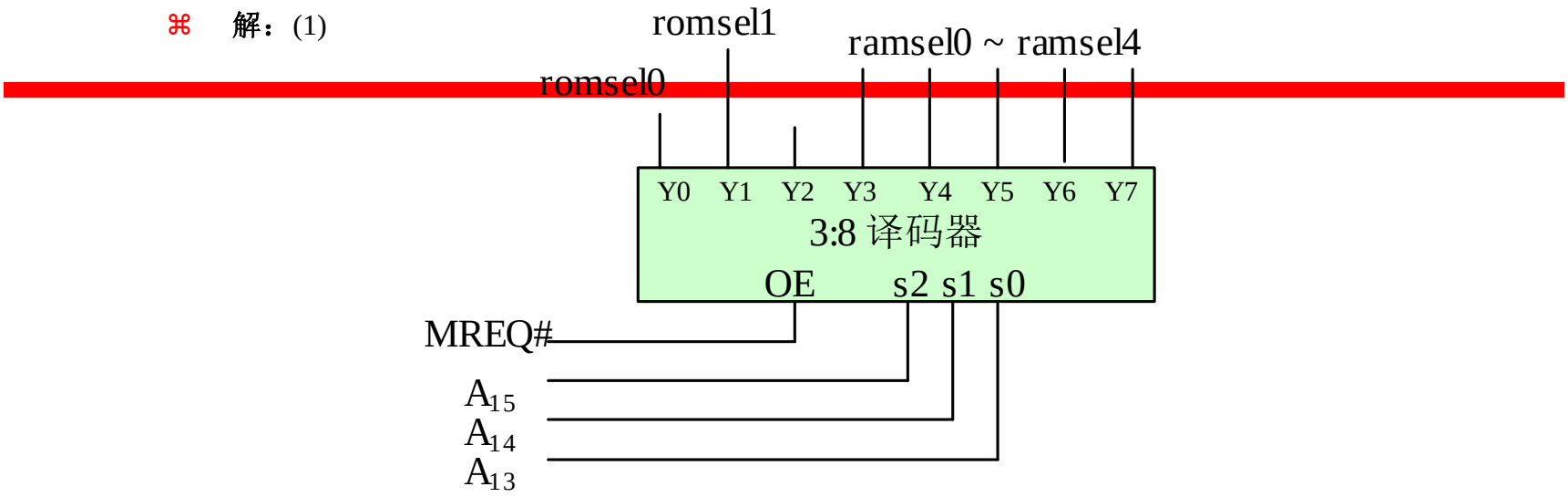
(2) 如果ROM和RAM存储器芯片都采用8K×1的芯片，试画出存储器与CPU的连接图。

(3) 如果ROM存储器芯片采用8K×8的芯片，RAM存储器芯片采用4K×8的芯片，试画出存储器与CPU的连接图。

(4) 如果ROM存储器芯片采用16K×8的芯片，RAM存储器芯片采用8K×8的芯片，试画出存储器与CPU的连接图。

(1) 画出地址译码方案

⌘ 解：(1)



译码器的输出信号逻辑表达式为：

$$romsel0 = \overline{A15} * \overline{A14} * \overline{A13} * \overline{MREQ\#}$$

$$romsel1 = \overline{A15} * \overline{A14} * A13 * \overline{MREQ\#}$$

$$ramsel0 = \overline{A15} * A14 * A13 * \overline{MREQ\#}$$

$$ramsel1 = A15 * \overline{A14} * \overline{A13} * \overline{MREQ\#}$$

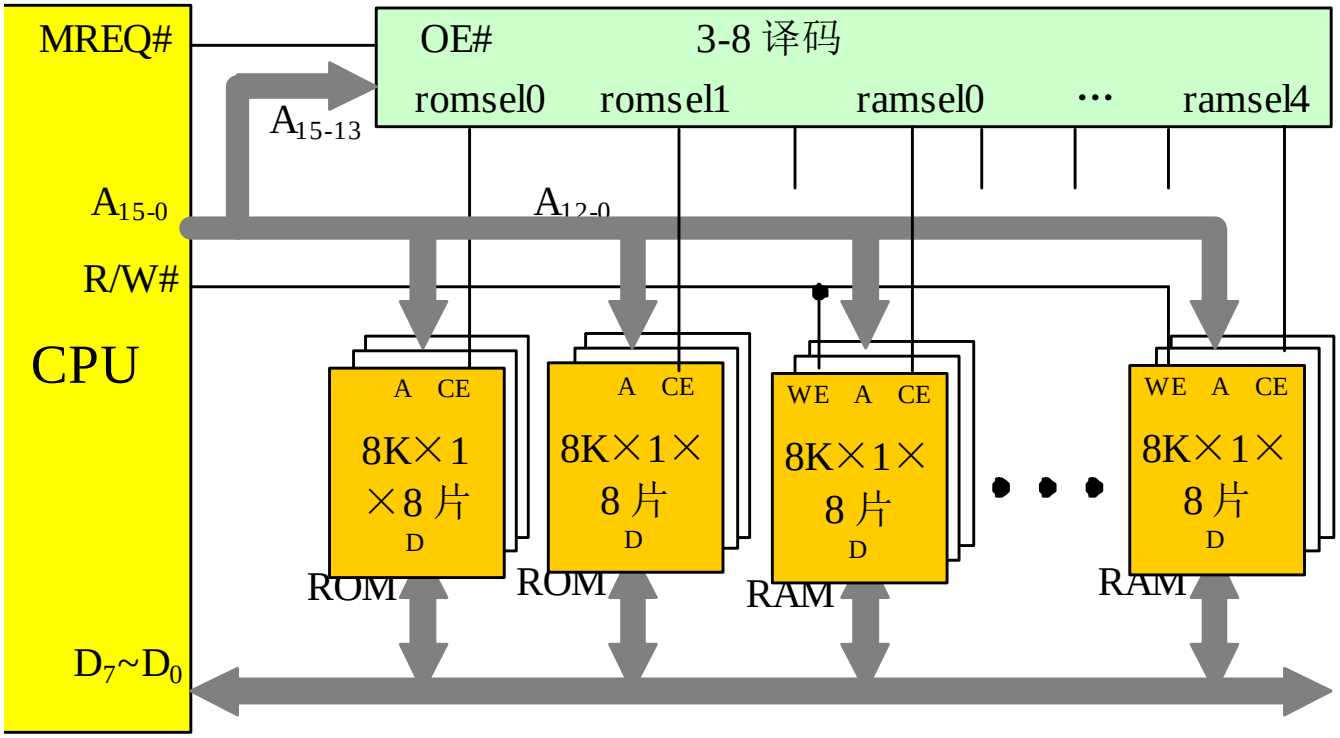
$$ramsel2 = A15 * \overline{A14} * A13 * \overline{MREQ\#}$$

$$ramsel3 = A15 * A14 * \overline{A13} * \overline{MREQ\#}$$

$$ramsel4 = A15 * A14 * A13 * \overline{MREQ\#}$$

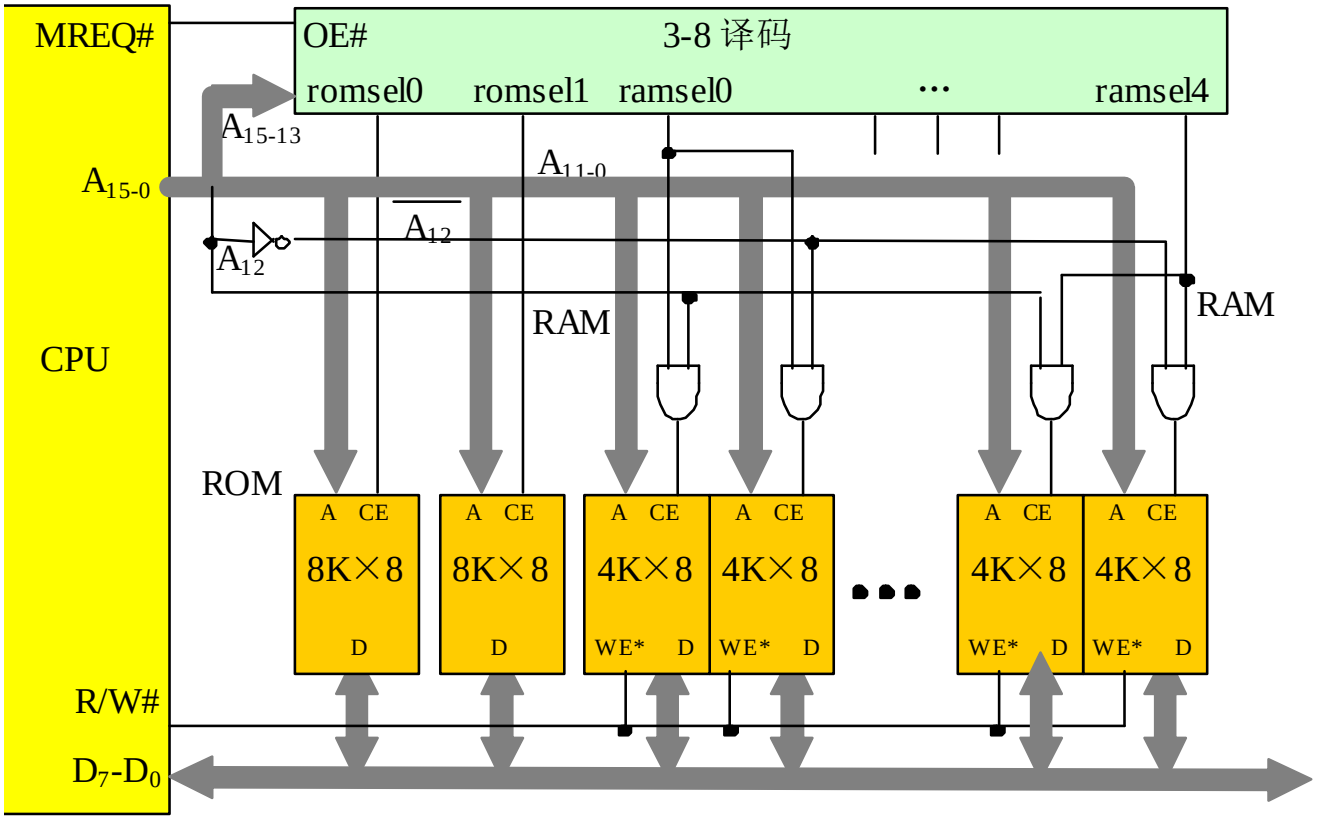
(2) 如果ROM和RAM存储器芯片都采用8K×1的芯片，试画出存储器与CPU的连接图。

解： **(2) 8KB的存储区域可以用8片存储器芯片构成一组实现。8K×1的存储器芯片的地址线需要13条，即A12~0。**



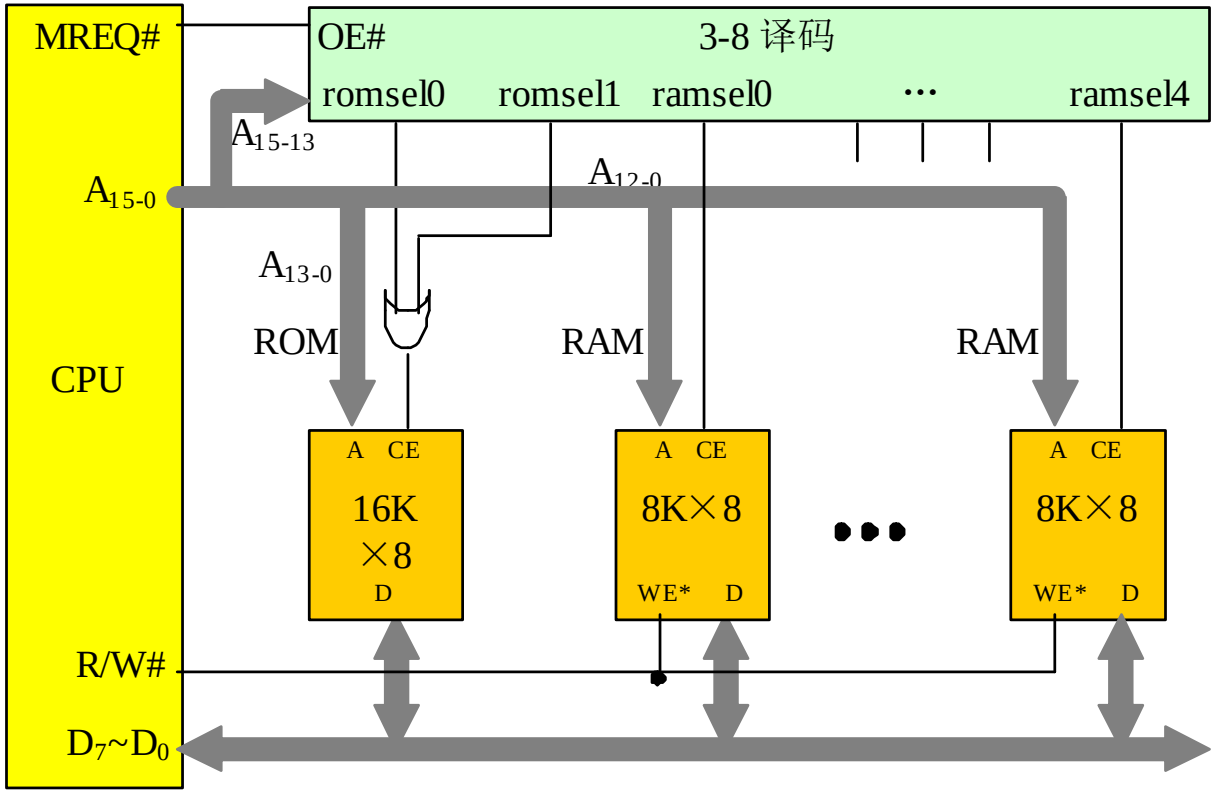
(3) 如果ROM存储器芯片采用8K×8的芯片，RAM存储器芯片采用4K×8的芯片，试画出存储器与CPU的连接图。

解：（3）



(4) 如果ROM存储器芯片采用16K×8的芯片，RAM存储器芯片采用8K×8的芯片，试画出存储器与CPU的连接图。

解：（4）



例2 某计算机系统的主存采用32位字节地址空间和64位数据线访问存储器，若使用64M位的DRAM芯片组成该机所允许的最大主存空间，并采用内存条的形式，问：

(1) 若每个内存条为64M×32位，共需多少内存条？

(2) 每个内存条内共有多少片DRAM芯片？

(3) 主存共需多少DRAM芯片？

(4) CPU如何有选择地访问各内存条？

解：(1) 主存最大空间为2^{32}=4GB，每个内存条的容量为64×4B = 256MB，主存需要的内存条数量为4GB/256MB=16条。

(2) 每个芯片的容量为8MB，内存条需要的芯片数量为256MB/8MB = 32片。

(3) 整个主存需要的内存芯片数量是16×32=512片。

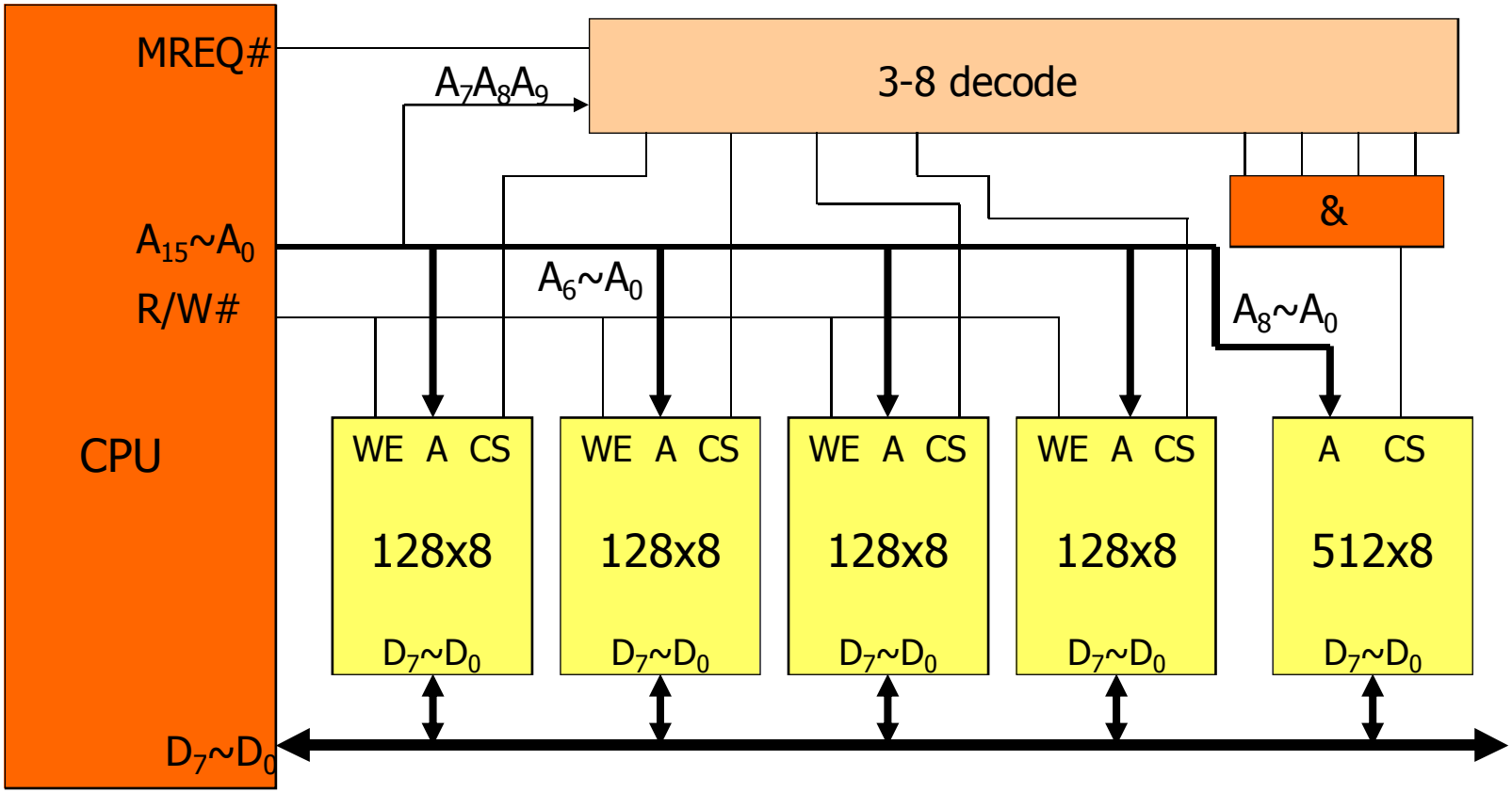
(4) 由于CPU字长为64位，内存条需要进行位扩展，即2个32位的内存条构成一组64位的存储单元组，16个内存条构成8组，为选择这8组内存条，CPU地址中需要用最高3位地址作为产生选择信号的地址码。

例3 假定计算机系统需要512字节RAM和512字节ROM容量。使用的RAM芯片是128字×8位，ROM芯片为512字×8位。RAM芯片有CS*及WE*控制端，ROM芯片有CS*控制端，CPU有地址线A15~A0、数据线D7~D0、读写控制线RW*等，试确定各存储器芯片的地址区间，指出存储器以及各存储器芯片需要的地址线数量，并画出存储器与CPU的连接图。

解：各存储器芯片的地址区间：

元件	16 进制地址范围	二进制地址值
RAM1	0000~007F	0 0 0 x x x x x x x
RAM2	0080~00FF	0 0 1 x x x x x x x
RAM3	0100~017F	0 1 0 x x x x x x x
RAM4	0180~01FF	0 1 1 x x x x x x x
ROM	0200~03FF	1 x x x x x x x x x

- 存储器的总容量为1KB，需要10条地址线。
- RAM芯片需要7条信号线(2^7=128)，ROM芯片需要9条地址线(2^9=512)。
- 存储器与CPU的连接图



Data Integrity

- Checksum byte for ROM
- Parity bit for DRAM
- CRC for disks and the Internet

Checksum Byte

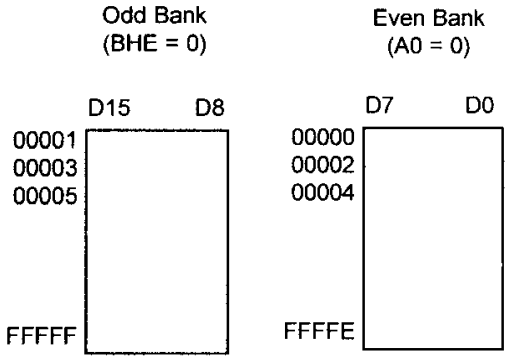
- Check the integrity of a series of bytes
  - Calculation
    - Add all bytes together and drop all carries
    - Take the 2's complement of the sum
  - Store the checksum byte together with data
  - check the integrity by adding data and the checksum together
  - *E.g., 38H, 23H, 33H, 07H, what is the checksum byte?*

Parity bit

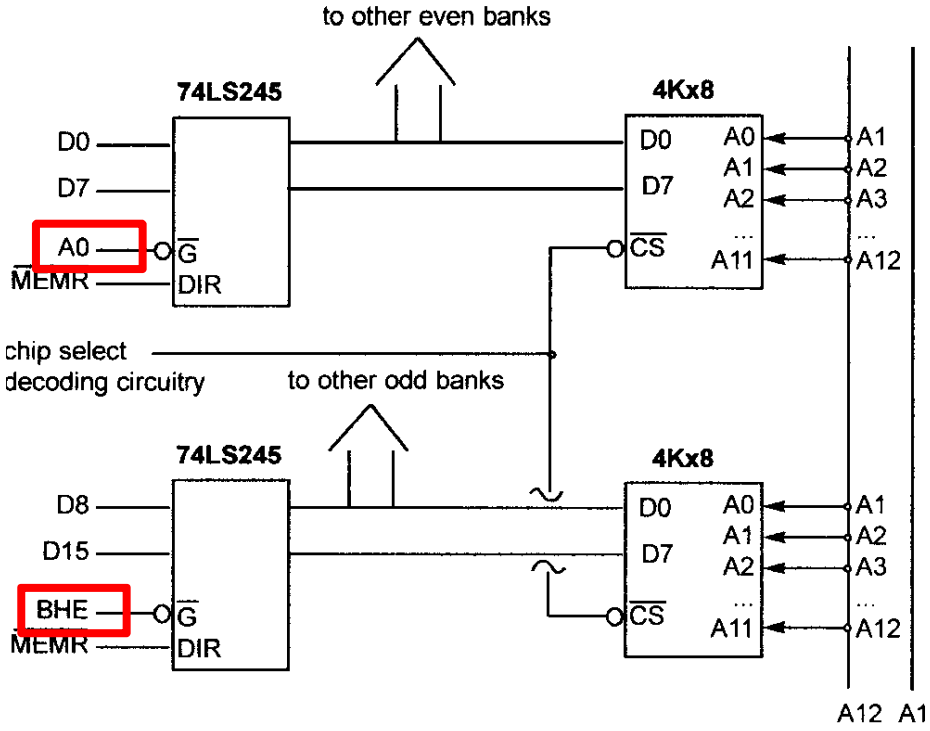
- Check the integrity of a series of bits (a byte)
  - Calculation
    - Even number of 1s, the bit is set

Memory Organization in 8086

- Even and odd banks

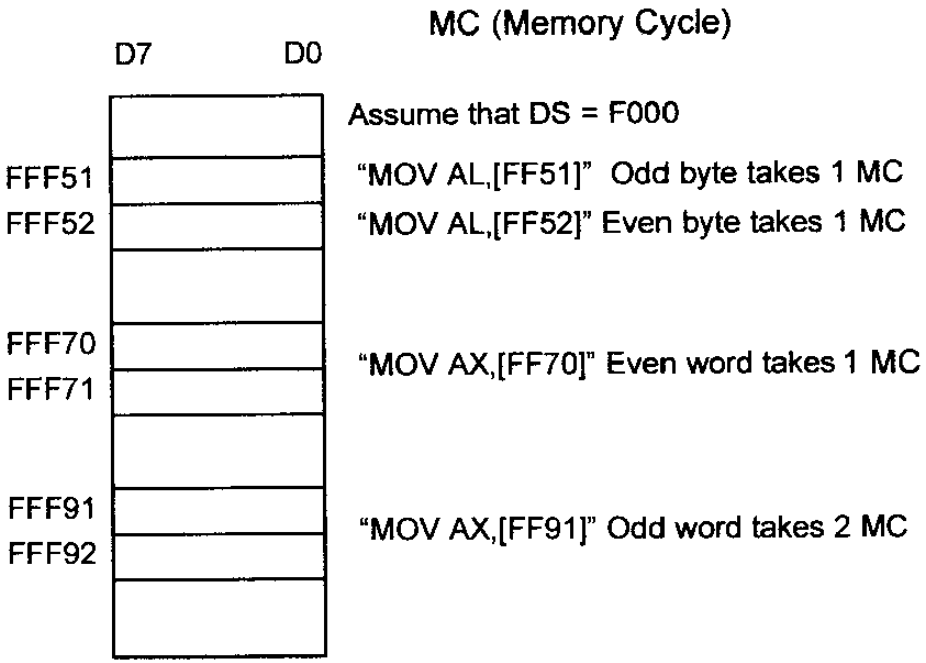


BHE	A0		
0	0	Even word	D0 - D15
0	1	Odd byte	D8 - D15
1	0	Even byte	D0 - D7
1	1	None	



Accessing Odd & Even Words

- Memory cycle : the bus cycle time used for accessing memory
  - Read cycle vs. write cycle



I/O in X86 family – The Other Space from Memory

- X86 microprocessors have an I/O space in addition to memory space
- Use special I/O instructions accessing I/O devices at ports
- Memory can contain machine codes and data, I/O ports only contain data
- Also referred to as *peripheral I/O* or *isolated I/O*

I/O Instructions – 8-Bit Instance

	Inputting Data		Outputting Data	
Format:	IN	dest,source	OUT	dest,source
(1)	IN	AL,port#	OUT	port#,AL

- port# ranges from 00h to 0ffh, 256 ports in total

(2)	MOV	DX,port#	MOV	DX,port#
	IN	AL,DX	OUT	DX,AL

- port# ranges from 0000h to 0ffffh, 65536 ports in all

- **Note: no segment concept for port addresses**

I/O Example

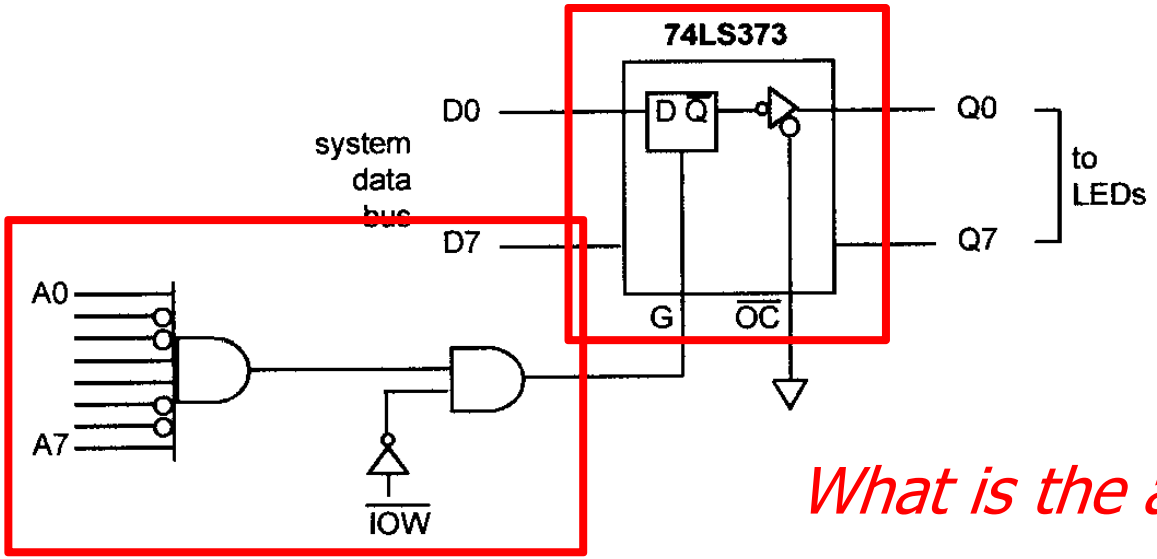
In a given 8088-based system, port address 22H is an input port for monitoring the temperature. Write Assembly language instructions to monitor that port continuously for the temperature of 100 degrees. If it reaches 100, then BH should contain 'Y'.

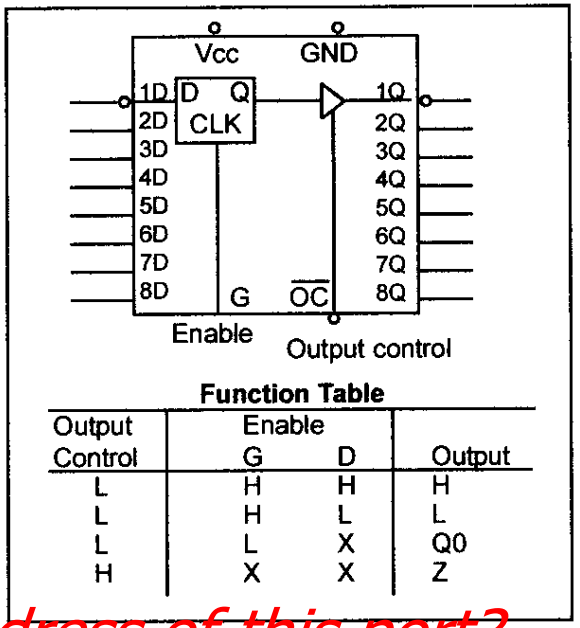
Solution:

```
BACK:    IN    AL,22H
         CMP   AL,100
         JNZ   BACK
         MOV   BH,'Y
```

Output Port Design

- Latch the data coming from the CPU
- Address decoding





Function Table

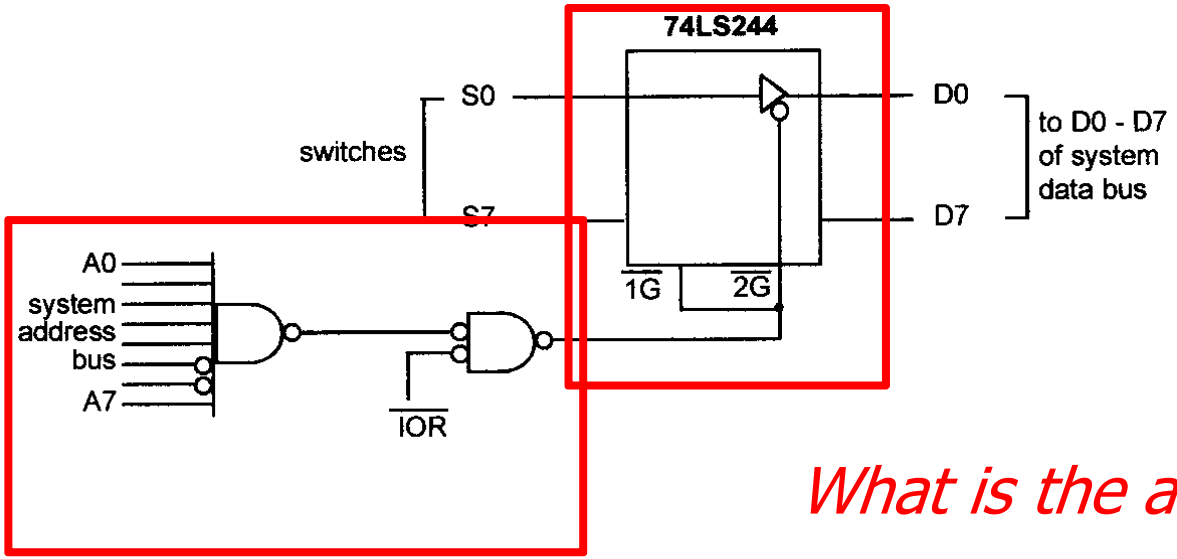
Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	Q0
H	X	X	Z

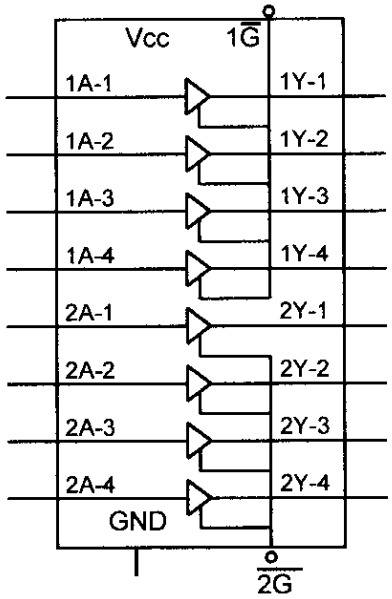
What is the address of this port?

74LS373 D Latch

Input Port Design

- Use *tri-state buffer* to connect to system data bus
- Address decoding





What is the address or this port?