Principle and Interface Techniques of Microcontroller

--8051 Microcontroller and Embedded Systems Using Assembly and C

LI, Guang (李光) **Prof. PhD, DIC, MIET**

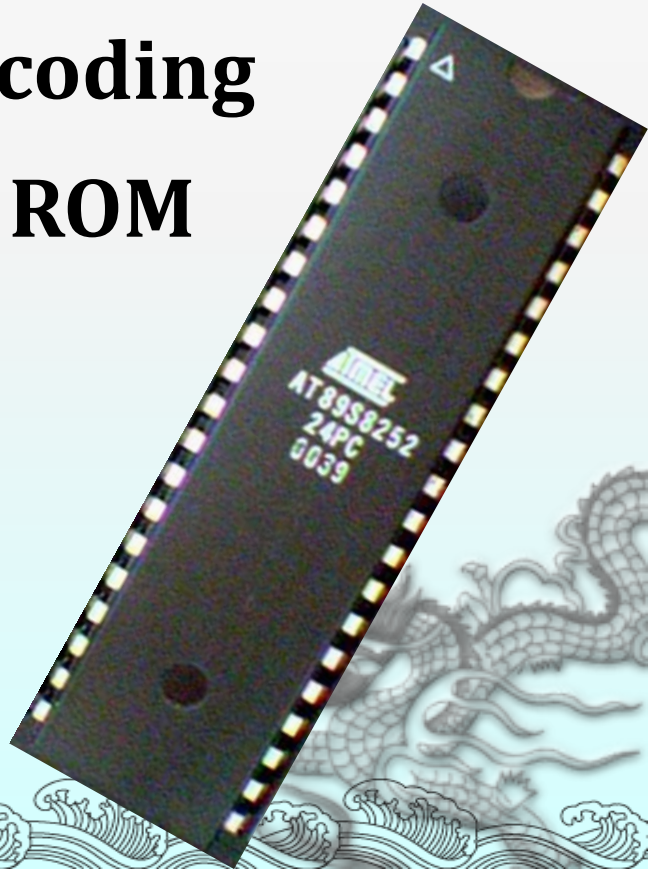
WANG, You (王酉) **PhD, MIET**

杭州 • 浙江大学 • 2015

Chapter 12
8031/51 Interfacing to External Memory

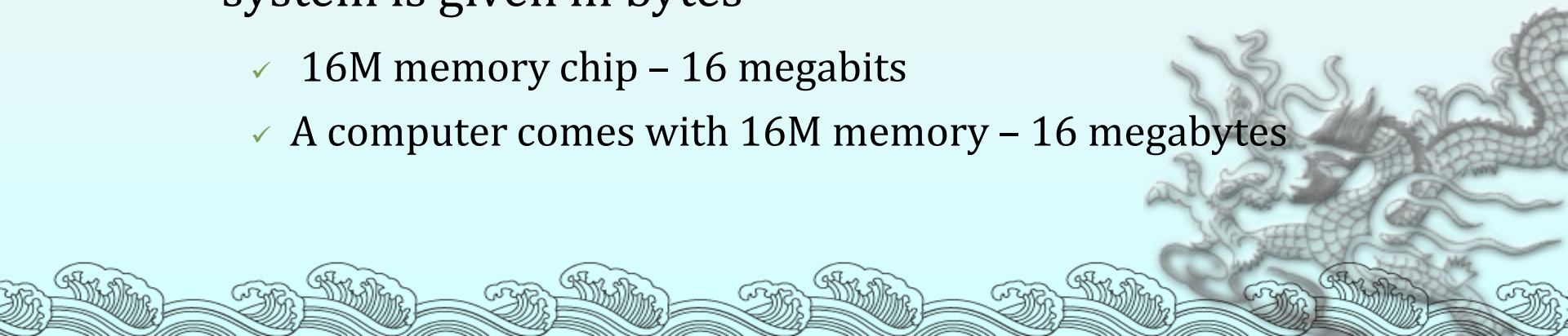
Outline

- § 12-1 Semiconductor Memory
- § 12-2 8051 Data Memory Space
- § 12-3 Memory Address Decoding
- § 12-4 Interfacing External ROM



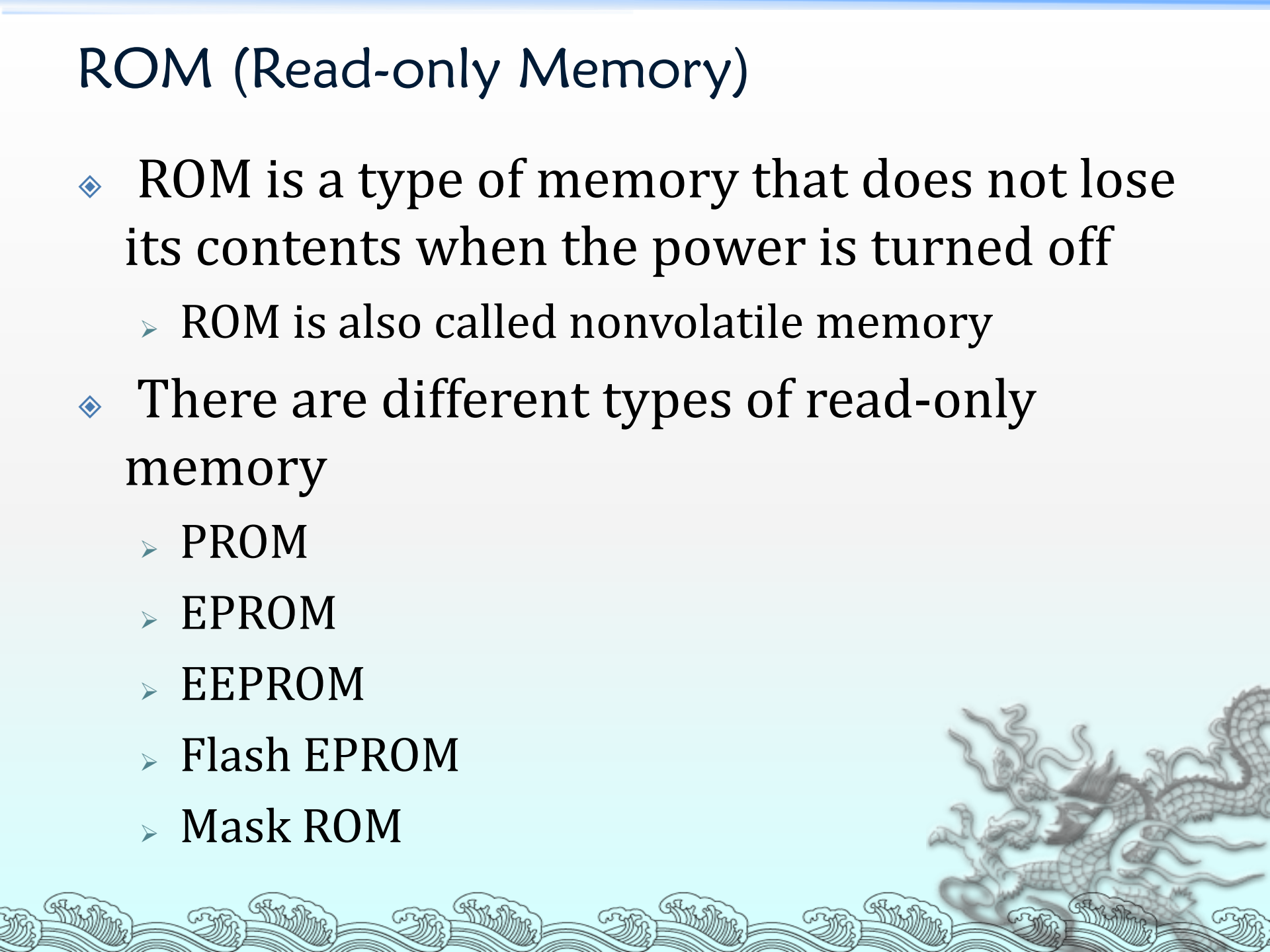
§ 12-1 Semiconductor Memory

- The number of bits that a semiconductor memory chip can store is called chip capacity
  - It can be in units of Kbits (kilobits), Mbits (megabits), and so on
- This must be distinguished from the storage capacity of computer systems
  - While the memory capacity of a memory IC chip is always given bits, the memory capacity of a computer system is given in bytes
    - 16M memory chip – 16 megabits
    - A computer comes with 16M memory – 16 megabytes



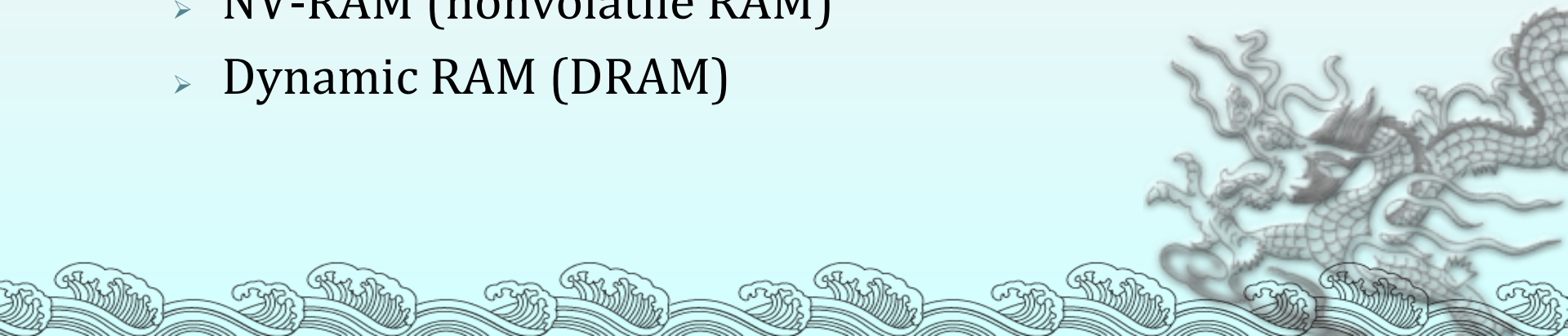
ROM (Read-only Memory)

- ROM is a type of memory that does not lose its contents when the power is turned off
  - ROM is also called nonvolatile memory
- There are different types of read-only memory
  - PROM
  - EPROM
  - EEPROM
  - Flash EPROM
  - Mask ROM



RAM (Random Access Memory)

- RAM memory is called volatile memory since cutting off the power to the IC will result in the loss of data
  - Sometimes RAM is also referred to as RAWM (read and write memory), in contrast to ROM, which cannot be written to
- There are three types of RAM
  - Static RAM (SRAM)
  - NV-RAM (nonvolatile RAM)
  - Dynamic RAM (DRAM)



§ 12-2 8051 Data Memory Space

- The 8051 has 128K bytes of address space
  - 64K bytes are set aside for program code
    - ✓ Program space is accessed using the program counter (PC) to locate and fetch instructions
    - ✓ In some example we placed data in the code space and used the instruction

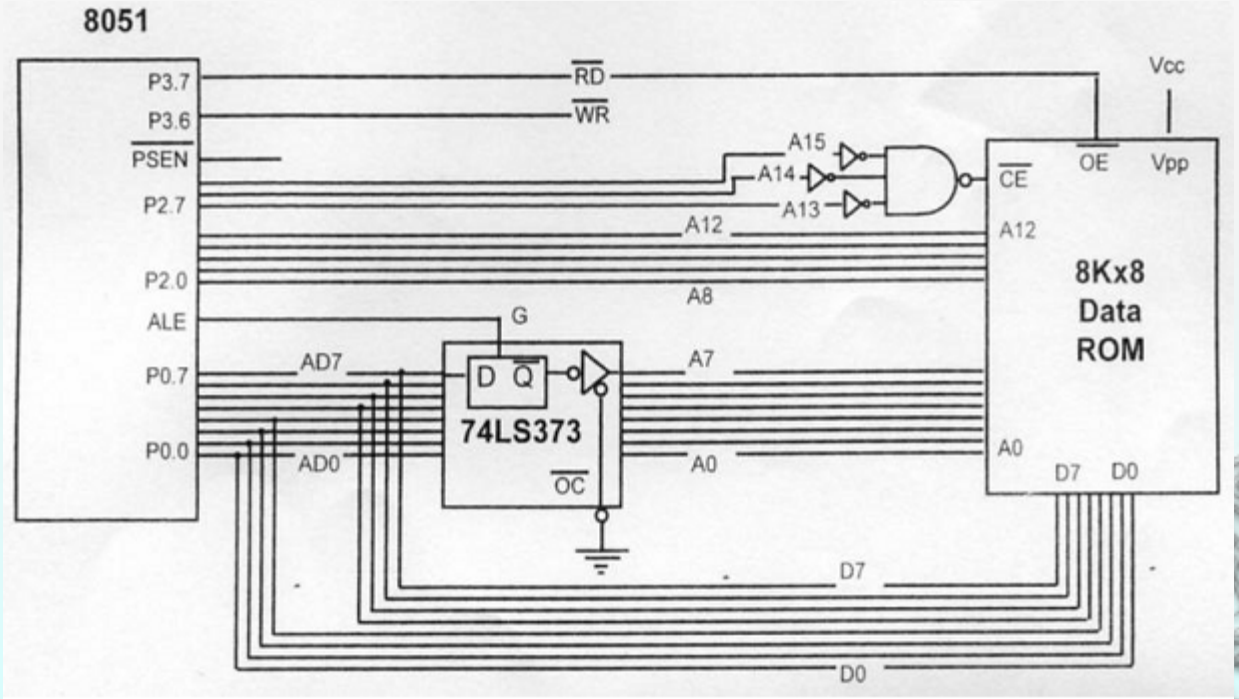
      MOVC A,@A+DPTR to get data, where C stands for code
  - The other 64K bytes are set aside for data
    - ✓ The data memory space is accessed using the DPTR register and an instruction called MOVX, where X stands for external

      – The data memory space must be implemented externally

      MOVX A,@DPTR

External ROM for Data

- We use RD to connect the 8031/51 to external ROM containing data
  - For the ROM containing the program code, PSEN is used to fetch the code



8051 Connection to External Data ROM

MOVX Instruction

- MOVX is a widely used instruction allowing access to external data memory space
  - To bring externally stored data into the CPU, we use the instruction MOVX A,@DPTR

An external ROM uses the 8051 data space to store the look-up table (starting at 1000H) for DAC data. Write a program to read 30 Bytes of these data and send it to P1.

Solution:

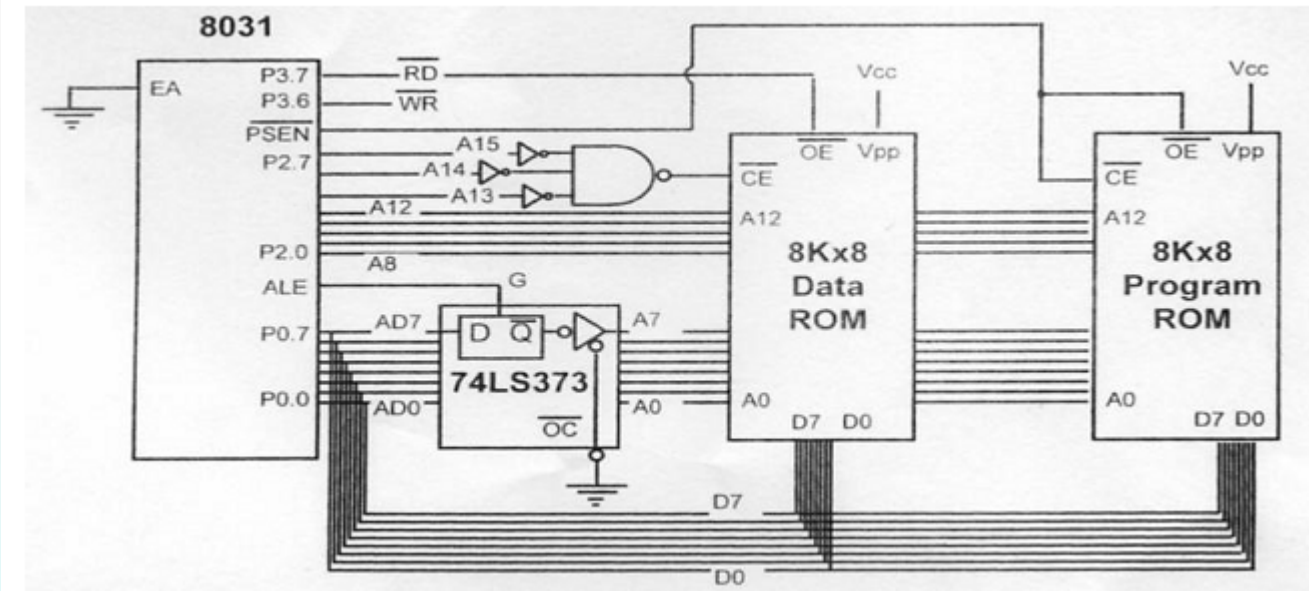
```
        MYXDATA EQU 1000H
        COUNT   EQU 30
         …
        MOV DPTR,#MYXDATA
        MOV R2,#COUNT
AGAIN:  MOVX A,@DPTR
        MOV P1,A
         INC DPTR
        DJNZ R2,AGAIN
```

Although both MOVC A,@A+DPTR and MOVX A,@DPTR look very similar, one is used to get data in the code space and the other is used to get data in the data space of the microcontroller

Show the design of an 8031-based system with 8K bytes of program ROM and 8K bytes of data ROM.

Solution:

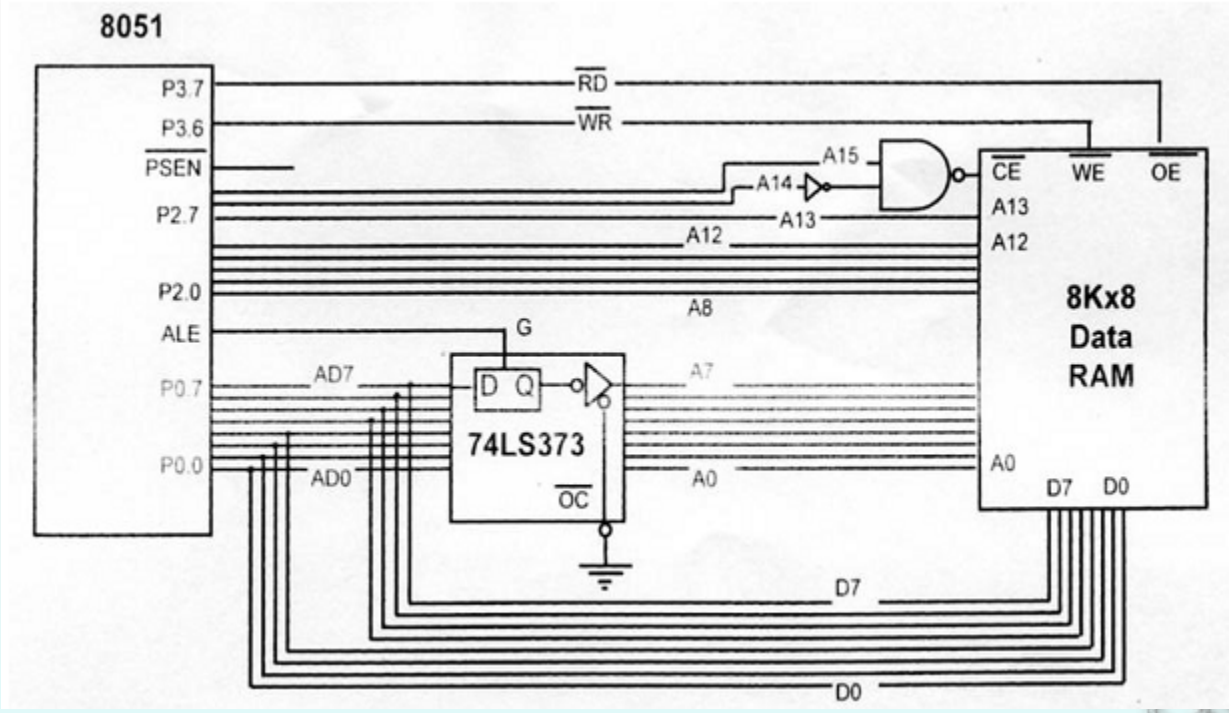
Figure 14-14 shows the design. Notice the role of PSEN and RD in each ROM. For program ROM, PSEN is used to activate both OE and CE. For data ROM, we use RD to active OE, while CE is activated by a Simple decoder.



8031 Connection to External Data ROM and External Program ROM

External Data RAM

- To connect the 8051 to an external SRAM, we must use both RD (P3.7) and WR (P3.6)



8051 Connection to External Data RAM

- In writing data to external data RAM, we use the instruction MOVX @DPTR,A

(a) Write a program to read 200 bytes of data from P1 and save the data in external RAM starting at RAM location 5000H.

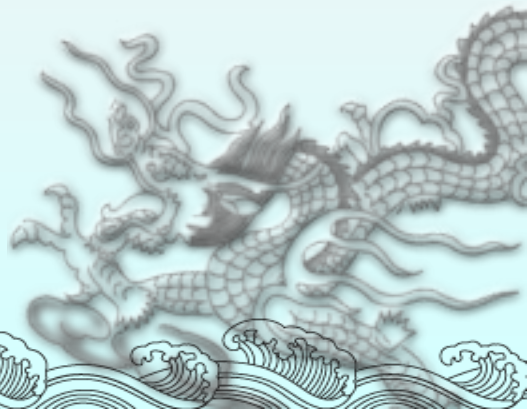
(b) What is the address space allocated to data RAM in Figure 14-15?

Solution:

(a)

```
        RAMDATA  EQU  5000H
        COUNT  EQU  200
        MOV DPTR,#RAMDATA
        MOV R3,#COUNT
AGAIN:  MOV A,P1
        MOVX @DPTR,A
         ACALL DELAY
         INC DPTR
         DJNZ R3,AGAIN
HERE:    SJMP HERE
```

(b) The data address space is 8000H to BFFFH.

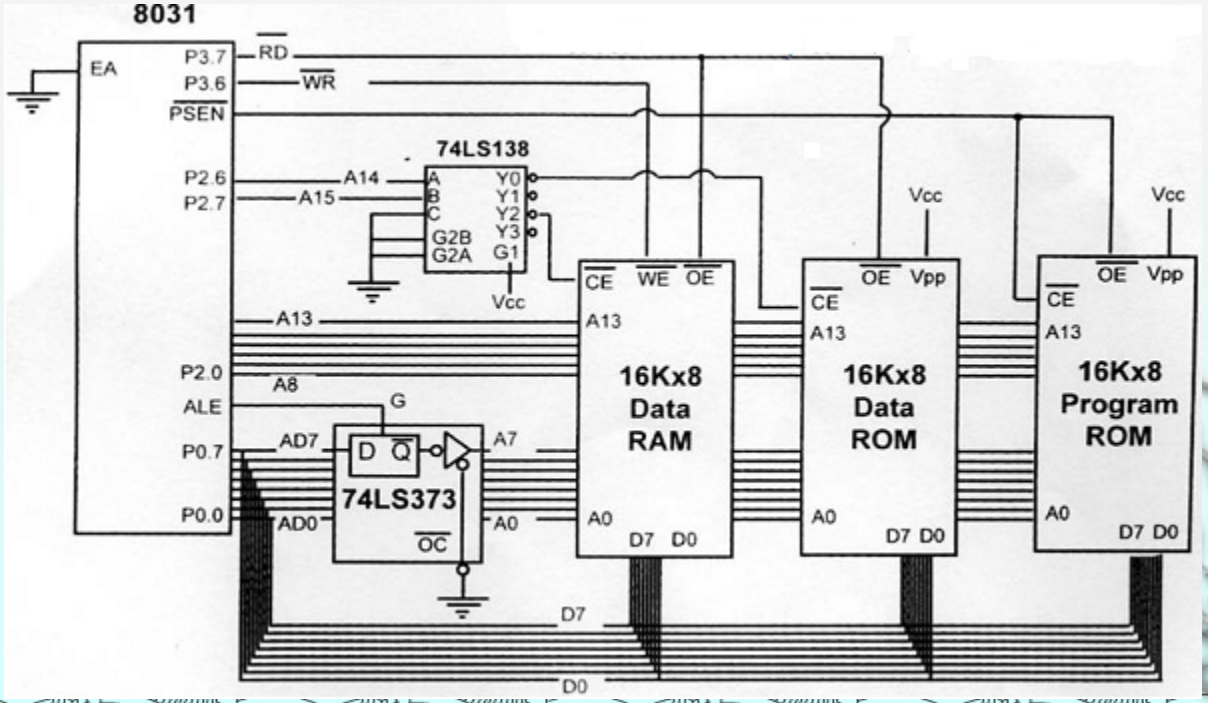


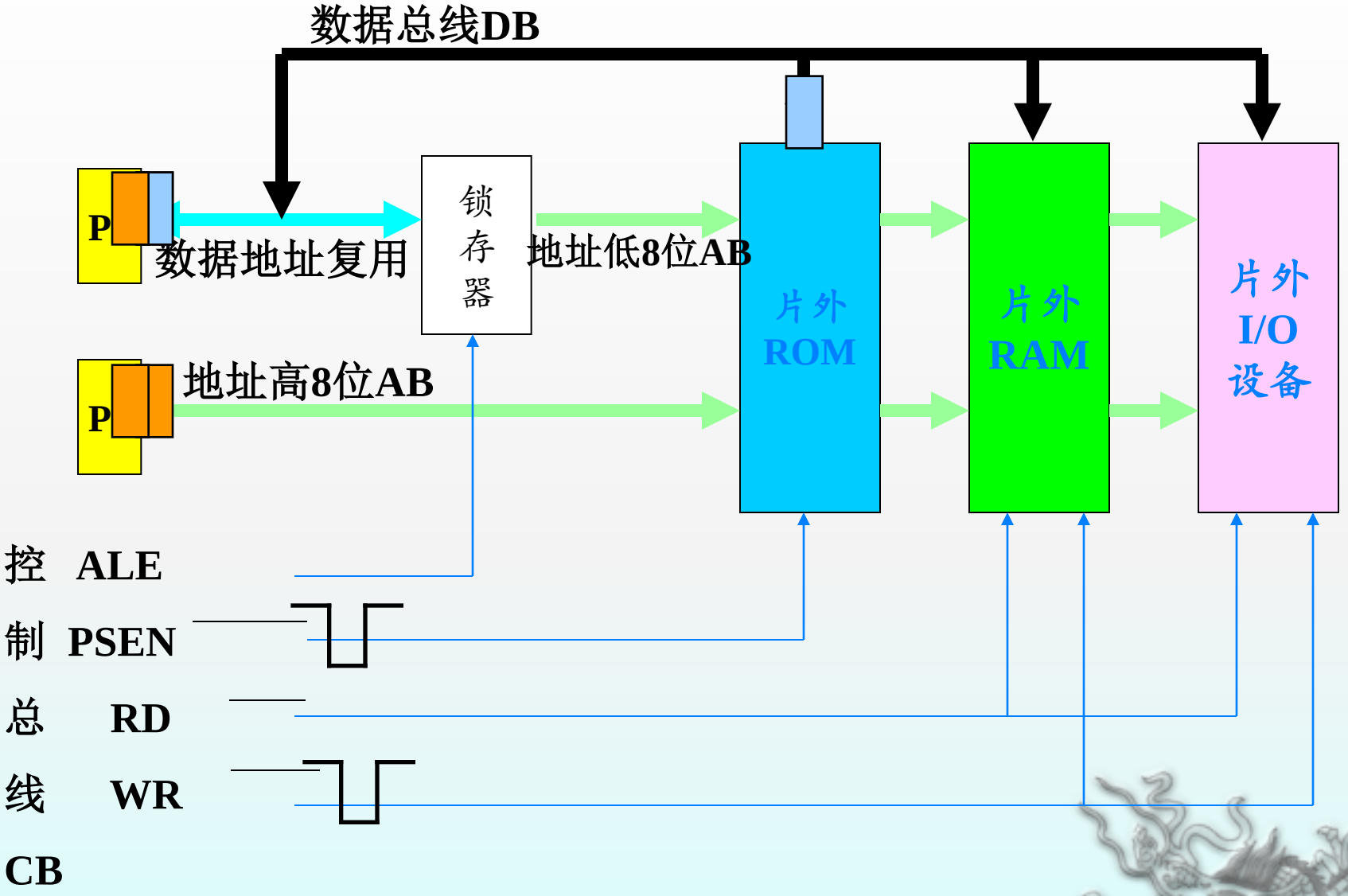
Assume that we need an 8031 system with 16KB of program space, 16KB of data ROM starting at 0000, and 16K of NV-RAM starting at 8000H. Show the design using a 74LS138 for the address decoder.

Solution:

The solution is diagrammed in Figure 14-17. Notice that there is no need for a decoder for program ROM, but we need a 74LS138 decoder For data ROM and RAM. Also notice that G1 = Vcc, G2A = GND, G2B = GND, and the C input of the 74LS138 is also grounded since we Use Y0 – Y3 only.

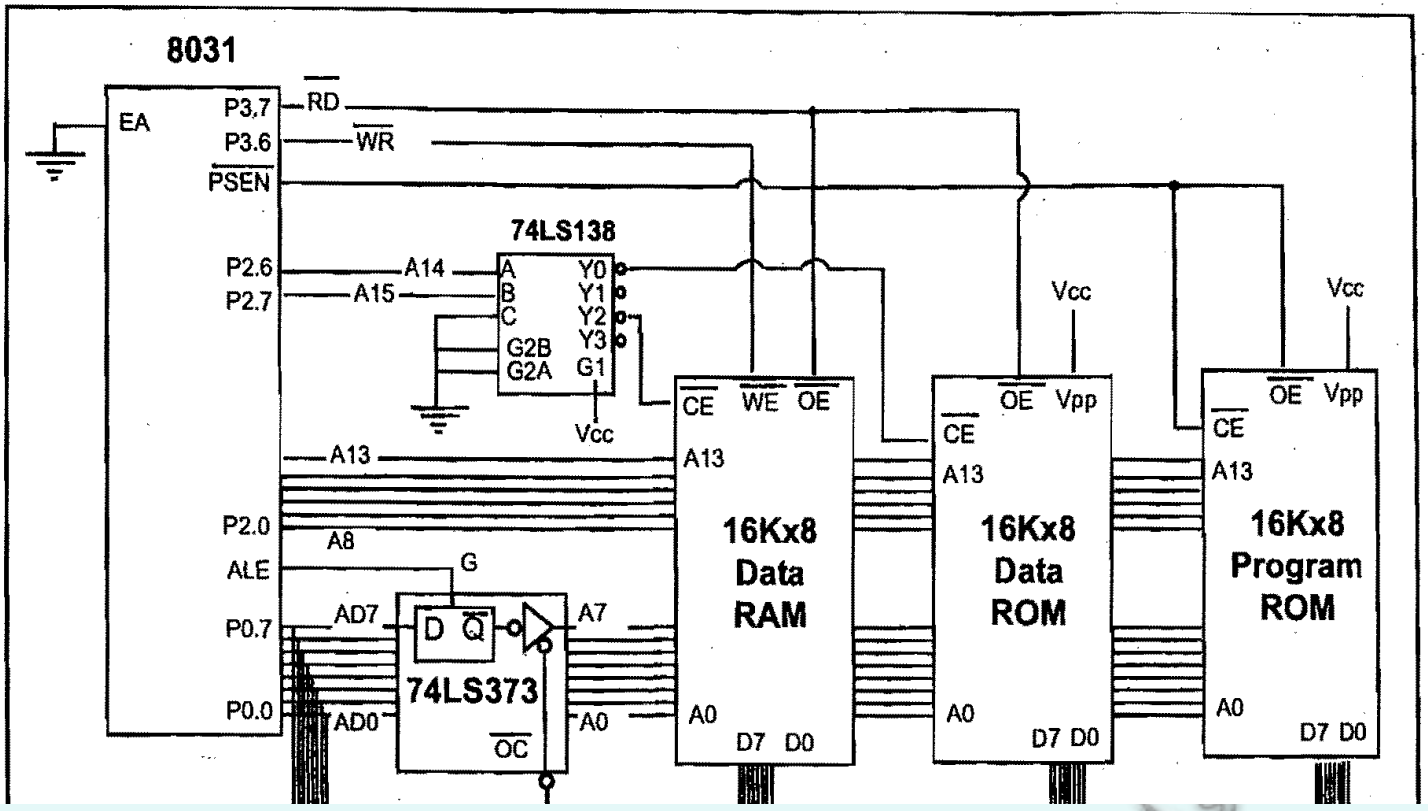
8031 Connection to External Program ROM, Data RAM, and Data ROM





片外三总线与片外芯片的连接关系图

Interfacing to Large External Memory



8051 Accessing 256K*8 External NV-RAM

In a certain application, we need 256K bytes of NV-RAM to store data collected by an 8051 microcontroller. (a) Show the connection of an 8051 to a single 256K×8 NV-RAM chip. (b) Show how various blocks of this single chip are accessed

Solution:

(a) The 256K×8 NV-RAM has 18 address pins (A0 – A17) and 8 data lines. As shown in Figure 14-18, A0 – A15 go directly to the memory chip while A16 and A17 are controlled by P1.0 and P1.1, respectively. Also notice that chip select of external RAM is connected to P1.2 of the 8051.

(b) The 256K bytes of memory are divided into four blocks, and each block is accessed as follows :

Chip select P1.2	A17 P1.1	A16 P1.0	Block address space
0	0	0	00000H - 0FFFFH
0	0	1	10000H - 1FFFFH
0	1	0	20000H - 2FFFFH
0	1	1	30000H - 3FFFFH
1	x	x	External RAM disabled

For example, to access the 20000H – 2FFFFH address space we need the following :

```
CLR     P1.2        ;enable external RAM
MOV     DPTR,#0  ;start of 64K memory block
CLR     P1.0     ;A16 = 0
SETB    P1.1     ;A17 = 1 for 20000H block
MOV     A, SBUF   ;get data from serial port
MOVX  @DPTR,A
INC     DPTR     ;next location
...
```

§ 12-3 Memory Address Decoding

- The CPU provides the address of the data desired, but it is the job of the decoding circuitry to locate the selected memory block
  - Memory chips have one or more pins called CS (chip select), which must be activated for the memory's contents to be accessed
  - Sometimes the chip select is also referred to as chip enable (CE)

Memory Address Decoding

- In connecting a memory chip to the CPU, note the following points
  - The data bus of the CPU is connected directly to the data pins of the memory chip
  - Control signals RD (read) and WR (memory write) from the CPU are connected to the OE (output enable) and WE (write enable) pins of the memory chip
  - In the case of the address buses, while the **lower bits of the address from the CPU go directly to the memory chip address pins, the upper ones are used to activate the CS pin of the memory chip**

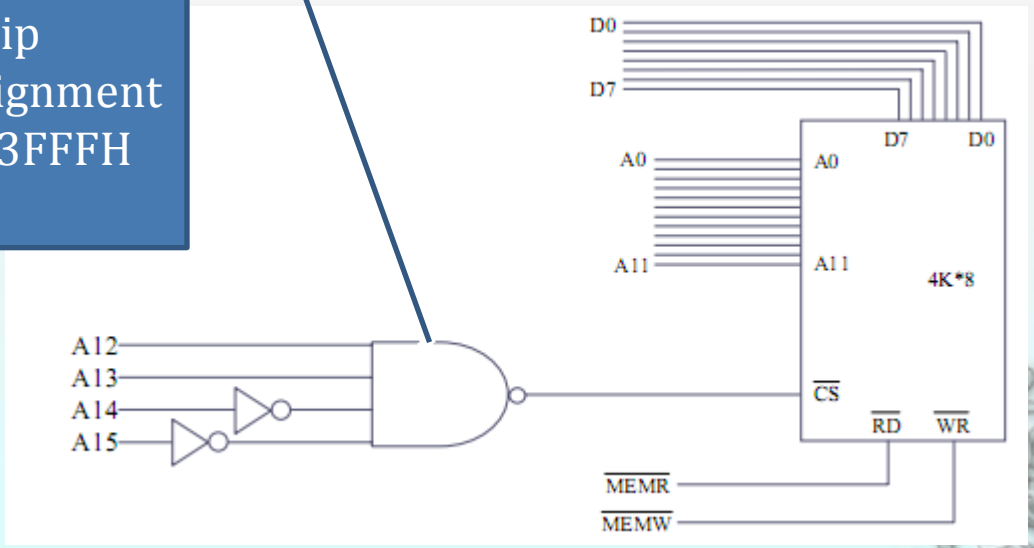
Memory Address Decoding

- Normally memories are divided into blocks and the output of the decoder selects a given memory block
  - Using simple logic gates
  - Using the 74LS138
  - Using programmable logics

Simple Logic Gate Address Decoder

- The simplest way of decoding circuitry is the use of NAND or other gates
  - The fact that the output of a NAND gate is active low, and that the CS pin is also active low makes them a perfect match

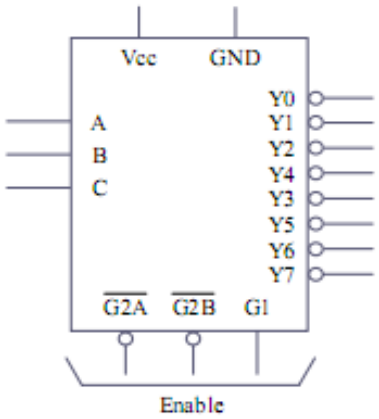
A15-A12 must be 0011 in order to select the chip
This result in the assignment of address 3000H to 3FFFH to this memory chip



Using 74LS138 3-8 Decoder

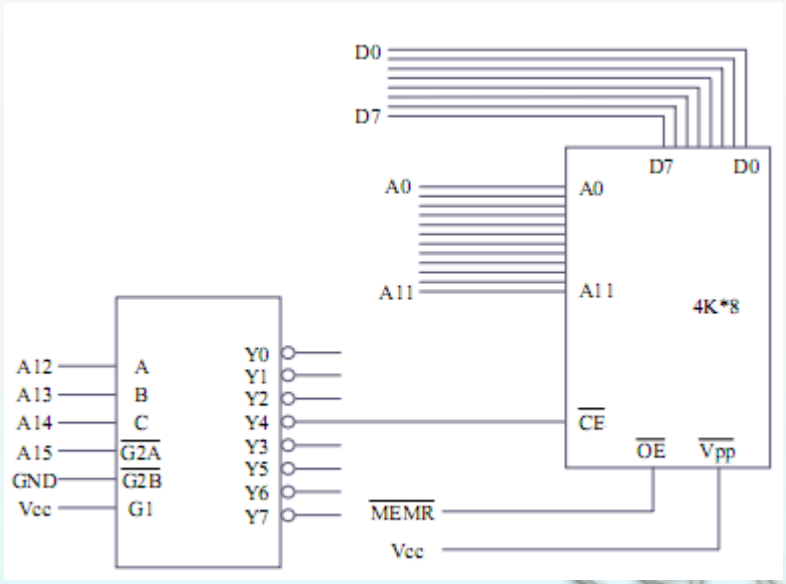
- This is one of the most widely used address decoders
  - The 3 inputs A, B, and C generate 8 active-low outputs Y0 – Y7
    - Each Y output is connected to CS of a memory chip, allowing control of 8 memory blocks by a single 74LS138
  - In the 74LS138, where A, B, and C select which output is activated, there are three additional inputs, G2A, G2B, and G1
    - G2A and G2B are both active low, and G1 is active high
    - If any one of the inputs G1, G2A, or G2B is not connected to an address signal, they must be activated permanently either by Vcc or ground, depending on the activation level

74LS138 Decoder



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	H H L	H	H	H	H	H	H	L	H
H	L	H H H	H	H	H	H	H	H	H	L



Looking at the design in Figure 14-6, find the address range for the Following. (a) Y4, (b) Y2, and (c) Y7.

Solution :

(a) The address range for Y4 is calculated as follows.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1

The above shows that the range for Y4 is 4000H to 4FFFH. In Figure 14-6, notice that A15 must be 0 for the decoder to be activated. Y4 will be selected when A14 A13 A12 = 100 (4 in binary). The remaining A11-A0 will be 0 for the lowest address and 1 for the highest address.

(b) The address range for Y2 is 2000H to 2FFFH.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

(c) The address range for Y7 is 7000H to 7FFFH.

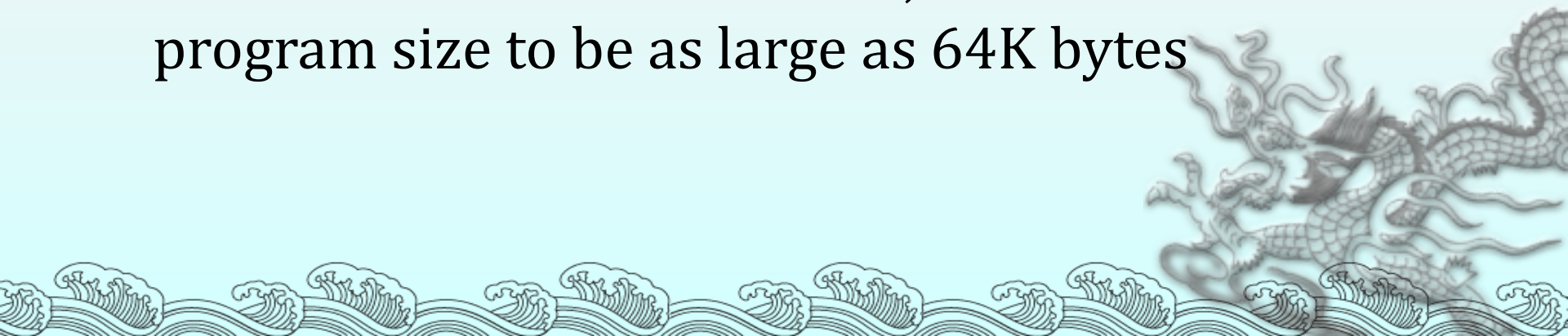
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Using Programmable Logic

- Other widely used decoders are programmable logic chips such as PAL and GAL chips
  - One disadvantage of these chips is that one must have access to a PAL/GAL software and burner, whereas the 74LS138 needs neither of these
  - The advantage of these chips is that they are much more versatile since they can be programmed for any combination of address ranges

§ 12-4 Interfacing External ROM

- The 8031 chip is a ROM less version of the 8051
  - It is exactly like any member of the 8051 family as far as executing the instructions and features are concerned, but it has no on-chip ROM
  - To make the 8031 execute 8051 code, it must be connected to external ROM memory containing the program code
- 8031 is ideal for many systems where the on-chip ROM of 8051 is not sufficient, since it allows the program size to be as large as 64K bytes



EA Pin

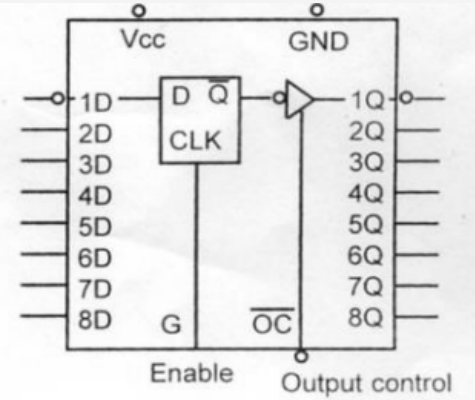
- For 8751/89C51/DS5000-based system, we connected the EA pin to Vcc to indicate that the program code is stored in the microcontroller's on-chip ROM
  - To indicate that the program code is stored in external ROM, this pin must be connected to GND

P0 and P2 in Providing Address

- Since the PC (program counter) of the 8031/51 is 16-bit, it is capable of accessing up to 64K bytes of program code
  - In the 8031/51, port 0 and port 2 provide the 16-bit address to access external memory
    - P0 provides the lower 8 bit address A0 – A7, and P2 provides the upper 8 bit address A8 – A15
    - P0 is also used to provide the 8-bit data bus D0 – D7
  - P0.0 – P0.7 are used for both the address and data paths
    - address/data multiplexing

P0 and P2 in Providing Address

- ALE (address latch enable) pin is an output pin for 8031/51
  - ALE = 0, P0 is used for data path
  - ALE = 1, P0 is used for address path
- To extract the address from the P0 pins we connect P0 to a 74LS373 and use the ALE pin to latch the address



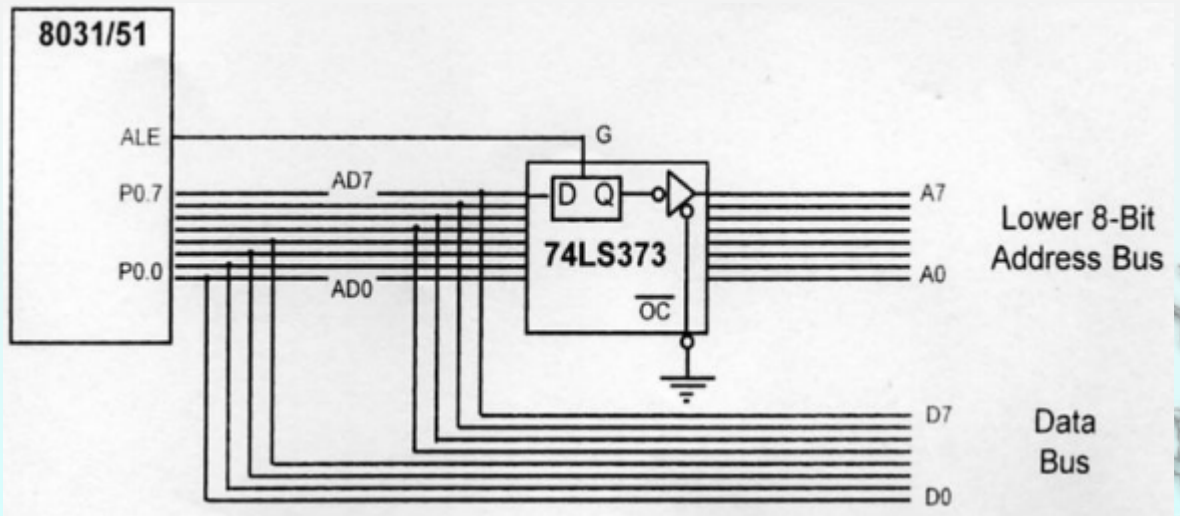
Funtion Table

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

P0 and P2 in Providing Address

- Normally ALE = 0, and P0 is used as a data bus, sending data out or bringing data in
- Whenever the 8031/51 wants to use P0 as an address bus, it puts the addresses A0 – A7 on the P0 pins and activates ALE = 1

Address/Data Multiplexing



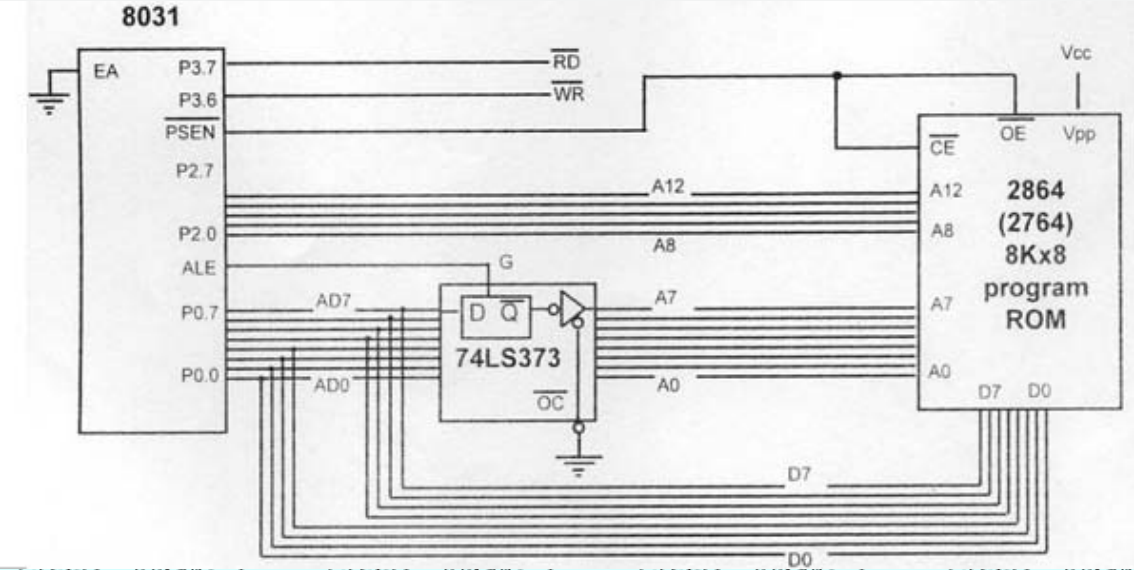
PSEN

- PSEN (program store enable) signal is an output signal for the 8031/51 microcontroller and must be connected to the OE pin of a ROM containing the program code
- It is important to emphasize the role of EA and PSEN when connecting the 8031/51 to external ROM
  - When the EA pin is connected to GND, the 8031/51 fetches opcode from external ROM by using PSEN

PSEN

- The connection of the PSEN pin to the OE pin of ROM
  - In systems based on the 8751/89C51/ DS5000 where EA is connected to Vcc, these chips do not activate the PSEN pin
    - ✓ This indicates that the on-chip ROM contains program code

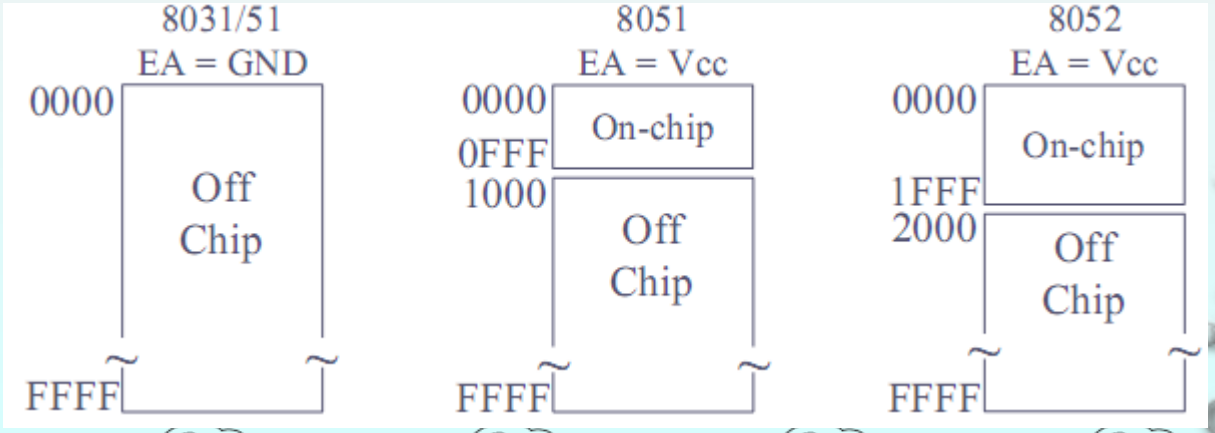
Connection to External Program ROM



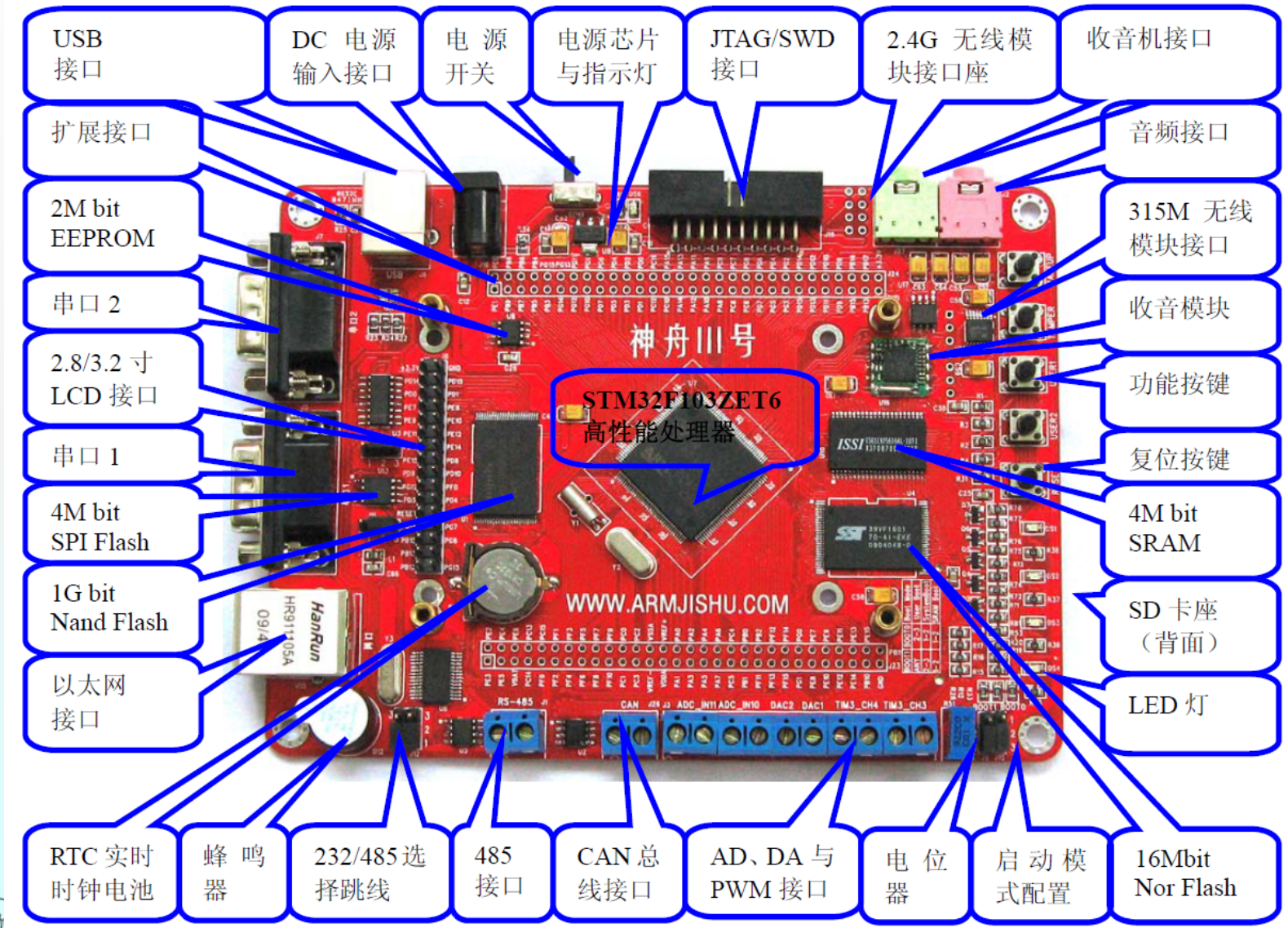
On-Chip and Off-Chip Code ROM

- In an 8751 system we could use on-chip ROM for boot code and an external ROM will contain the user's program
  - We still have EA = Vcc,
    - ✓ Upon reset 8051 executes the on-chip program first, then
    - ✓ When it reaches the end of the on-chip ROM, it switches to external ROM for rest of program

On-chip and Off-chip Program Code Access

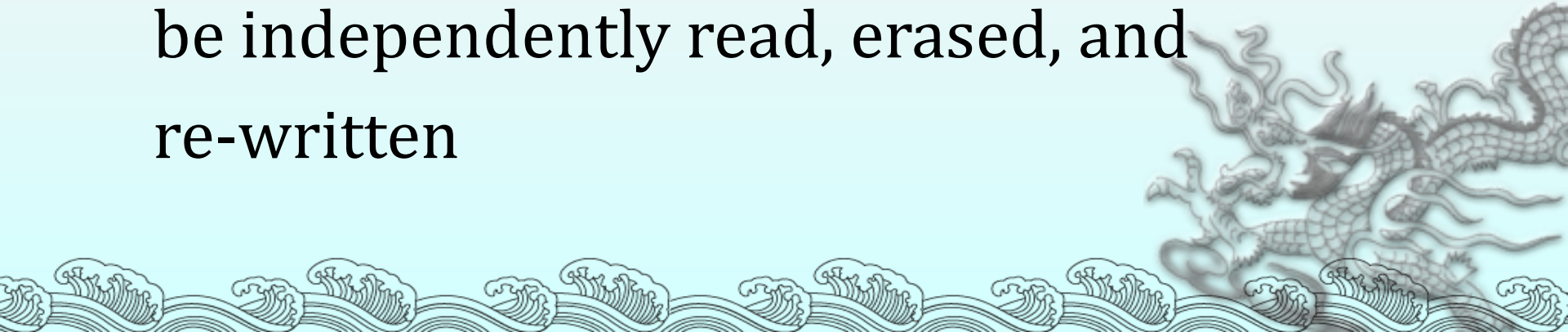


§ 12-5 New memory chips



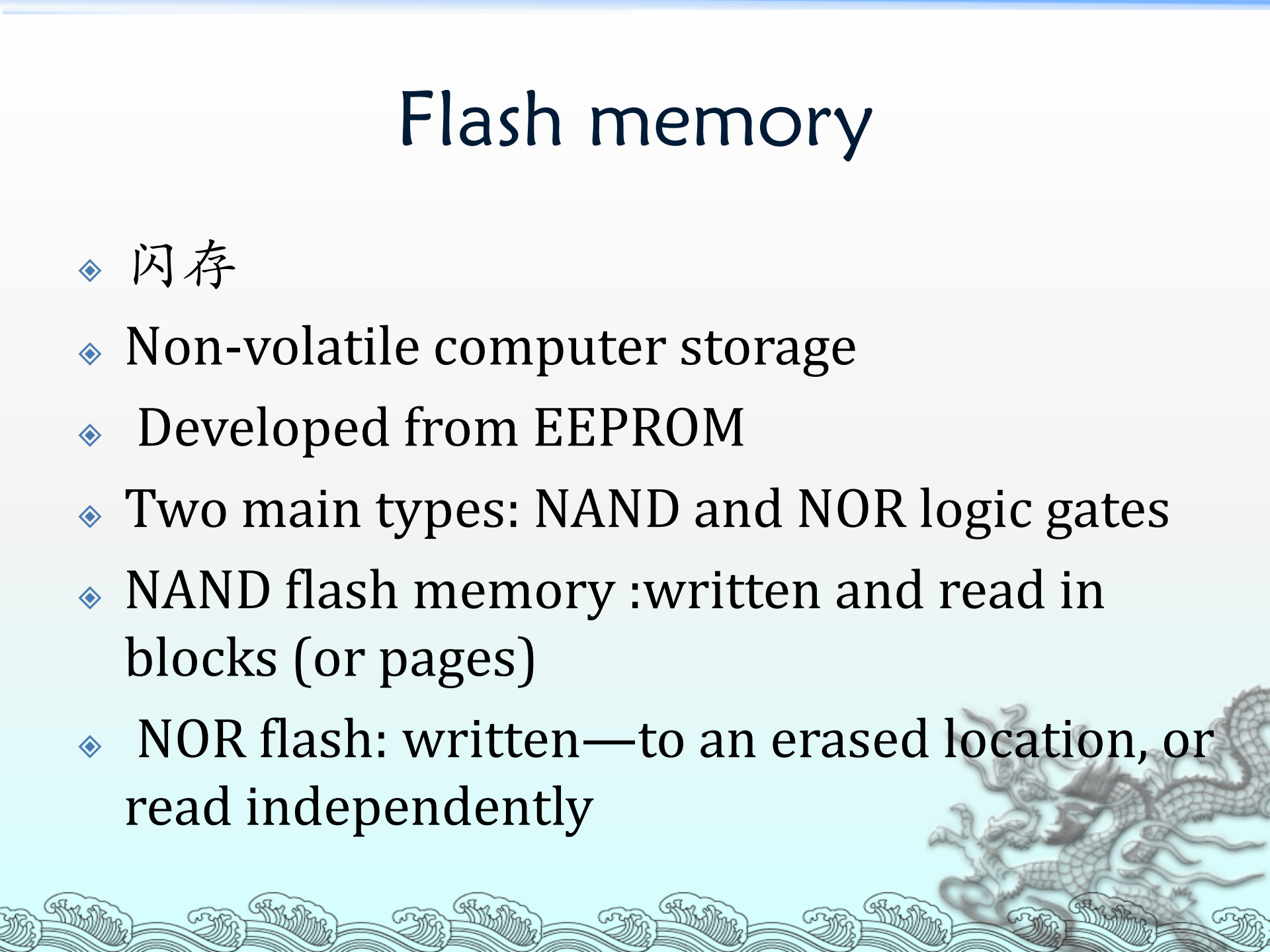
EEPROM

- **E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory
- Store small amounts of data that must be saved when power is removed
- e.g., calibration tables or device configuration
- Individual bytes in a traditional EEPROM can be independently read, erased, and re-written



Flash memory

- 闪存
- Non-volatile computer storage
- Developed from EEPROM
- Two main types: NAND and NOR logic gates
- NAND flash memory :written and read in blocks (or pages)
- NOR flash: written—to an erased location, or read independently

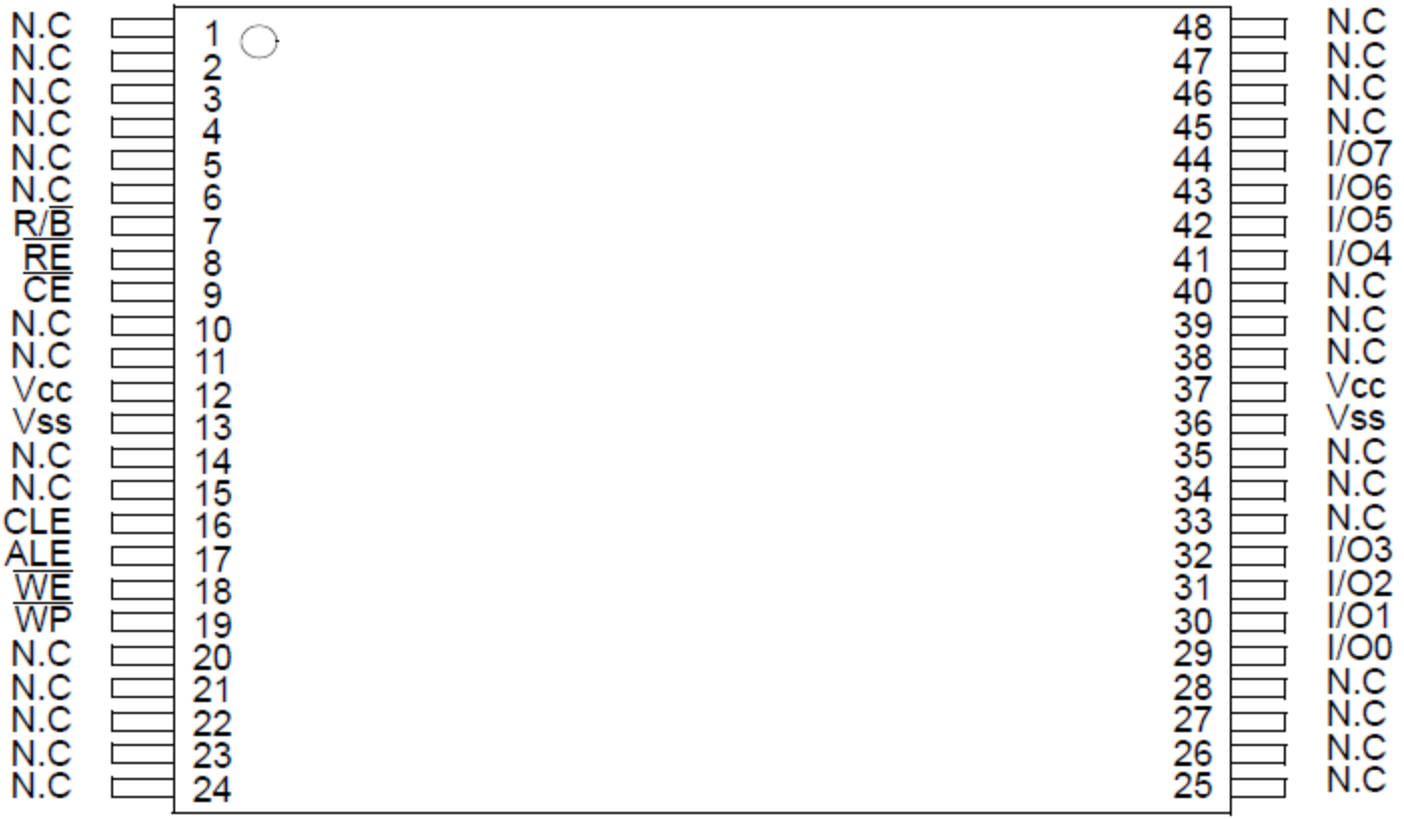


SRAM

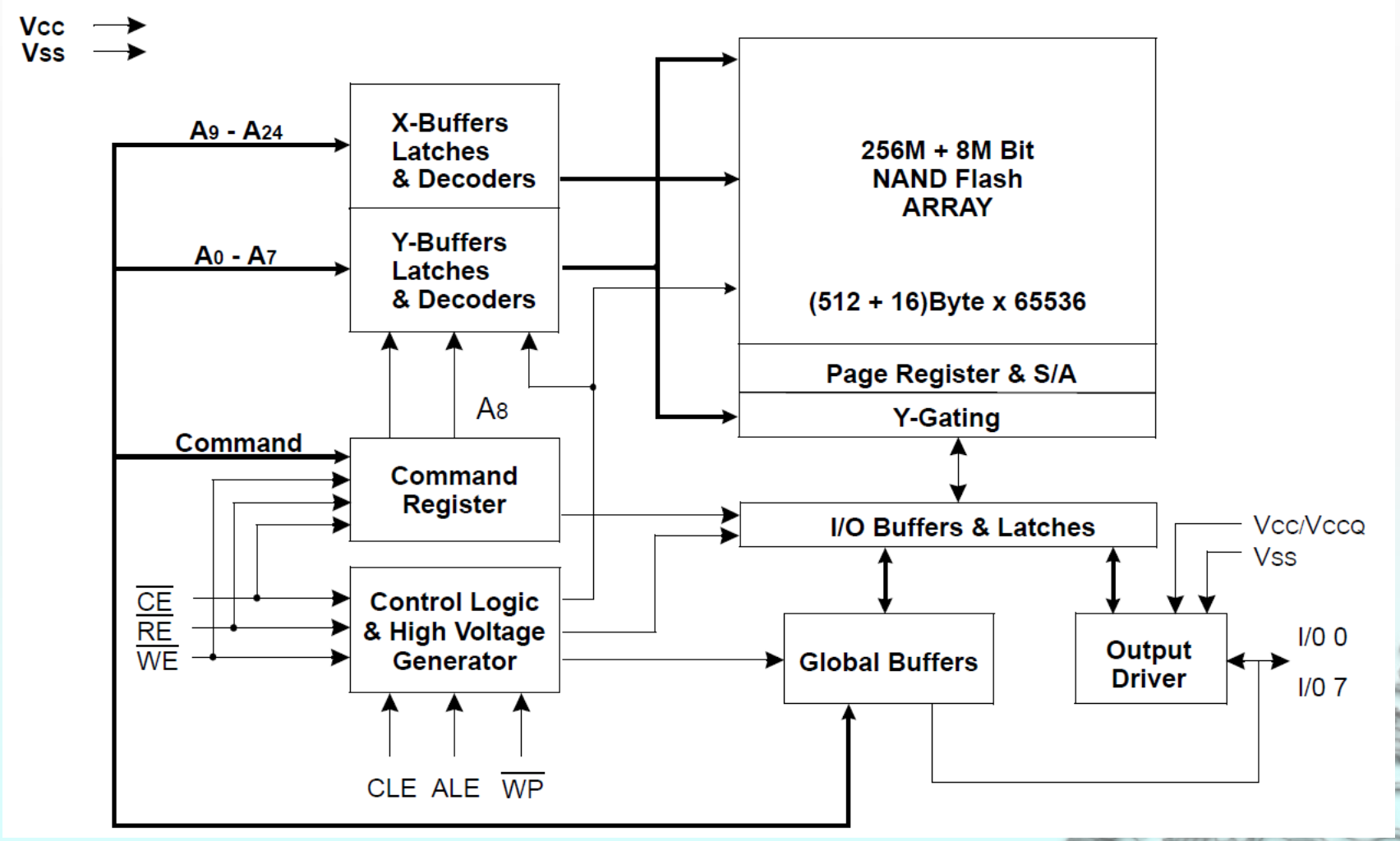
- Static random-access memory
- Bistable latching circuitry (flip-flop)
- Volatile, data lost when the memory is not powered

Flash memory——32M Bytes

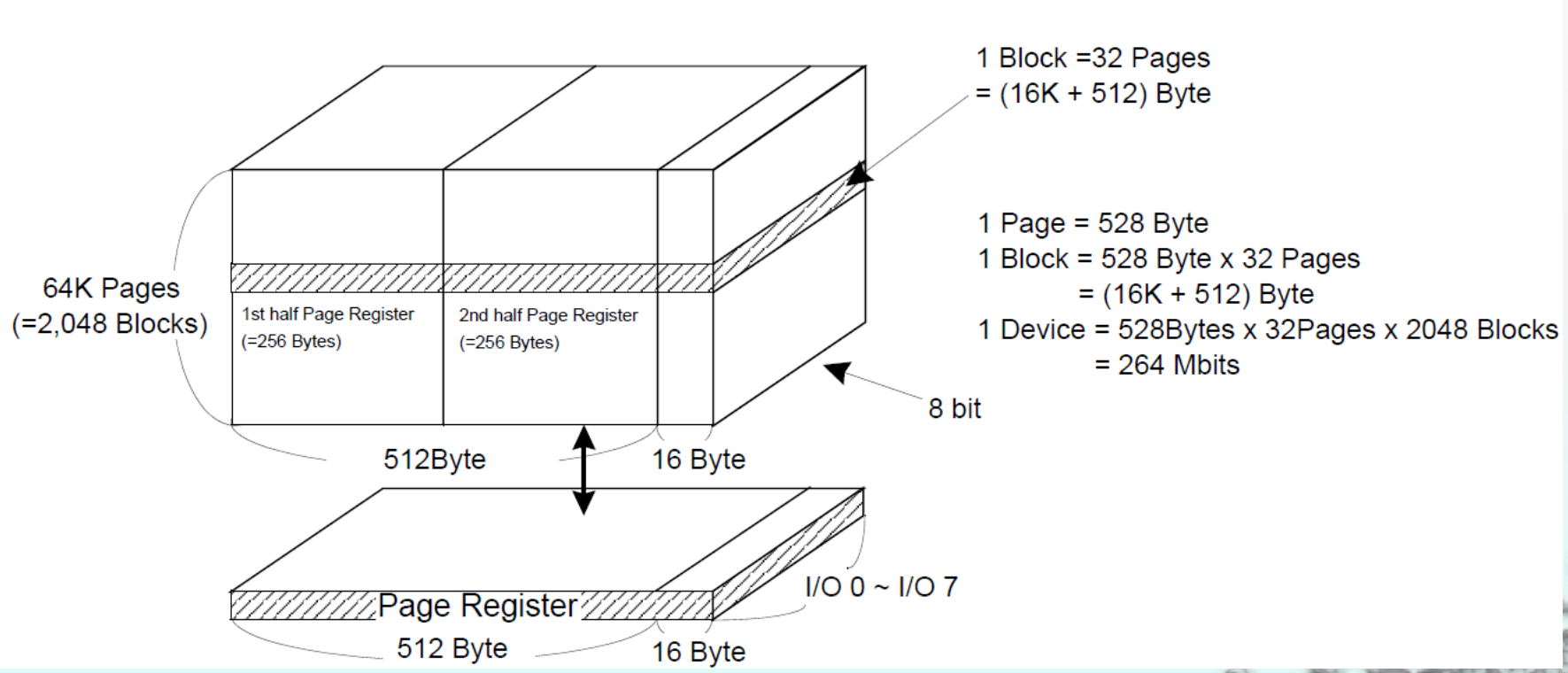
K9F5608D(U)0D-PCB0/PIB0



Function block diagram



Array organization



How to access a byte

	I/O 0	I/O 1	I/O 2	I/O 3	I/O 4	I/O 5	I/O 6	I/O 7	
1st Cycle	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	**Column Address**
2nd Cycle	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	**Row Address (Page Address)**
3rd Cycle	A_{17}	A_{18}	A_{19}	A_{20}	A_{21}	A_{22}	A_{23}	A_{24}	

NOTE : Column Address : Starting Address of the Register.

00h Command(Read) : Defines the starting address of the 1st half of the register.

01h Command(Read) : Defines the starting address of the 2nd half of the register.

* A_8 is set to "Low" or "High" by the 00h or 01h Command.

* The device ignores any additional input of address cycles than required.