

# **8051 INTERFACING TO EXTERNAL MEMORY**

# Memory Capacity

- The number of bits that a semiconductor memory chip can store
  - 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

# Memory Organization

- Memory chips are organized into a number of locations within the IC
  - Each location can hold 1 bit, 4 bits, 8 bits, or even 16 bits
    - The number of locations within a memory IC depends on the address pins
    - The number of bits that each location can hold is always equal to the number of data pins
      - A memory chip contain  $2^x$  location, where  $x$  is the number of address pins
      - Each location contains  $y$  bits, where  $y$  is the number of data pins on the chip
      - The entire chip will contain  $2^x \times y$  bits

# Speed

- One of the most important characteristics of a memory chip is the speed at which its data can be accessed
  - To access the data, the address is presented to the address pins
  - The READ pin is activated
    - After a certain amount of time has elapsed, the data shows up at the data pins
    - The shorter this elapsed time, the better, and consequently, the more expensive the memory chip
  - The speed of the memory chip is commonly referred to as its access time

### Example 14-1

A given memory chip has 12 address pins and 4 data pins. Find:  
(a) the organization, and (b) the capacity.

#### **Solution:**

- (a) This memory chip has 4096 locations ( $2^{12} = 4096$ ), and each location can hold 4 bits of data. This gives an organization of  $4096 \times 4$ , often represented as 4Kx4.
- (b) The capacity is equal to 16K bits since there is a total of 4K locations and each location can hold 4 bits of data.

### Example 14-2

A 512K memory chip has 8 pins for data. Find:  
(a) the organization, and (b) the number of address pins for this memory chip.

#### **Solution:**

- (a) A memory chip with 8 data pins means that each location within the chip can hold 8 bits of data. To find the number of locations within this memory chip, divide the capacity by the number of data pins.  $512K/8 = 64K$ ; therefore, the organization for this memory chip is 64Kx8.
- (b) The chip has 16 address lines since  $2^{16} = 64K$ .

# 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

# PROM (Programmable ROM)

- PROM refers to the kind of ROM that the user can burn information into
  - PROM is a user-programmable memory
    - For every bit of the PROM, there exists a fuse
    - If the information burned into PROM is wrong, that PROM must be discarded since its internal fuses are blown permanently
    - PROM is also referred to as OTP (one-time programmable) Programming ROM, also called burning ROM
    - It requires special equipment called a ROM burner or ROM programmer



# EPROM (Erasable Programmable ROM)

- EPROM was invented to allow making changes in the contents of PROM after it is burned
  - In EPROM, one can program the memory chip and erase it thousands of times
  - A widely used EPROM is called UVEPROM
    - UV stands for ultra-violet
      - They have a window that is used to shine ultraviolet (UV) radiation to erase its contents
    - The only problem with UV-EPROM is that erasing its contents can take up to 20 minutes



# EPROM (Erasable Programmable ROM) (cont.)

- To program a UV-EPROM chip, the following steps must be taken:
  - Its contents must be erased
    - Removed from its socket on the system board
    - Placed in EPROM erasure equipment to expose it to UV radiation for 15-20 minutes
  - Program the chip
    - Place it in the ROM burner
      - To burn code or data into EPROM, the ROM burner uses 12.5 volts,  $V_{pp}$ , or higher, depending on the EPROM type
  - Place it back into its system board socket

# EPROM (Erasable Programmable ROM) (cont.)

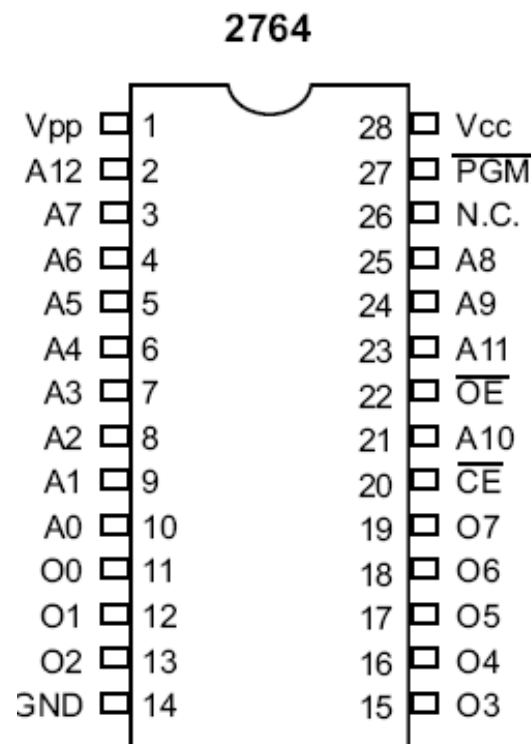
- There is an EPROM programmer (burner)
- There is also separate EPROM erasure equipment
  - The major disadvantage is that it cannot be programmed while in the system board
- Notice the pattern of the IC numbers
  - Ex. 27128-25 refers to UV-EPROM that has a capacity of 128K bits and access time of 250 nanoseconds
  - 27xx always refers to UV-EPROM chips

### Example 14-3

For ROM chip 27128, find the number of data and address pins.

#### Solution:

The 27128 has a capacity of 128K bits. It has 16Kx8 organization (all ROMs have 8 data pins), which indicates that there are 8 pins for data and 14 pins for address ( $2^{14} = 16K$ ).



# EEPROM (Electrically Erasable Programmable ROM)

- EEPROM has several advantage:
  - Its method of erasure is electrical and instant
    - As opposed to the 20-minute erasure time required for UVEPROM
  - One can select which byte to be erased
    - In contrast to UV-EPROM, in which the entire contents of ROM are erased
  - One can program and erase its contents while it is still in the system board
    - The designer incorporate into the system board the circuitry to program the EEPROM

# Flash Memory EPROM

- Flash EPROM has become a popular user-programmable memory chip since the early 1990s
  - The process of erasure of the entire contents takes less than a second, or might in a flash
    - The erasure method is electrical
    - It is commonly called flash memory
  - The major difference between EEPROM and flash memory is
    - Flash memory's contents are erased, then the entire device is erased

# Flash Memory EPROM (cont.)

- There are some flash memories are recently made so that the erasure can be done block by block
- One can erase a desired section or byte on EEPROM
- It is believed that flash memory will replace part of the hard disk as a mass storage medium
  - The flash memory can be programmed while it is in its socket on the system board
  - Widely used as a way to upgrade PC BIOS ROM



# Flash Memory EPROM (cont.)

- Flash memory is semiconductor memory with access time in the range of 100 ns
  - Compared with disk access time in the range of tens of milliseconds
- Flash memory's program/erase cycles must become infinite, like hard disks
  - Program/erase cycle refers to the number of times that a chip can be erased and programmed before it becomes unusable
  - The program/erase cycle is 100,000 for flash and EEPROM, 1000 for UV-EPROM

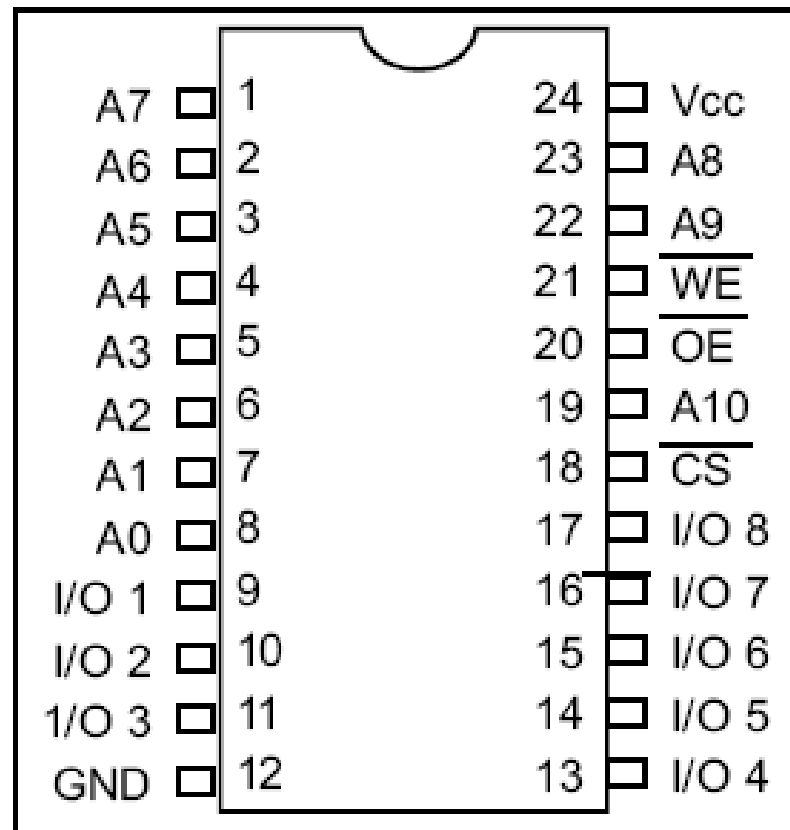


# RAM (Random Access Memory)

- 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)

# SRAM (Static RAM)

- Storage cells in static RAM memory are made of flip-flops
  - They do not require refreshing in order to keep their data
  - The problem with the use of flip-flops for storage cells is:
    - Each cell require at least 6 transistors to build, and the cell holds only 1 bit of data
    - In recent years, the cells have been made of 4 transistors, given birth to a high-capacity SRAM
      - Still too many
      - Its capacity is far below DRAM



**Figure 14-2. 2Kx8 SRAM Pins**

# NV-RAM (Nonvolatile RAM)

- It combines the best of RAM and ROM
  - The read and write ability of RAM, plus the nonvolatility of ROM
  - NV-RAM chip internally is made of the following components
    - It uses extremely power-efficient SRAM cells built out of CMOS
    - It uses an internal lithium battery as a backup energy source
    - It uses an intelligent control circuitry
      - The main job of this control circuitry is to monitor the Vcc pin constantly to detect loss of the external power supply

# DRAM (Dynamic RAM)

- Dynamic RAM uses a capacitor to store each bit
  - It cuts down the number of transistors needed to build the cell
  - It requires constant refreshing due to leakage
  - The advantages and disadvantages:
    - Major advantages are high capacity, cheaper cost per bit, and lower power consumption per bit
    - The disadvantages is
      - It must be refreshed periodically, due to the fact that the capacitor cell loses its charge
      - While it is being refreshed, the data cannot be accessed

# Packing Issue in DRAM

- A problem of packing a large number of cells into a single chip with the normal number of pins assigned to addresses
  - Large number of pins defeats the purpose of high density and small packaging
    - A 64K-bit chip ( $64K \times 1$ ) must have 16 address lines and 1 data line, requiring 16 pins to send in address
      - DRAM memory chips can have any of the x1, x4, x8, x16 organizations
  - The method used is to split the address in half
    - To send in each half of the address through the same pins, thereby requiring fewer address pins

# Packing Issue in DRAM (cont.)

- Internally, the DRAM structure is divided into a square of rows and columns
  - The first half of the address is called row
  - The second half is called column
    - The first half of the address is sent in through the address pins
      - By activating RAS (row address strobe)
      - The internal latches inside DRAM grab the first half
    - After that, the second half of the address is sent in through the same pins
      - By activating CAS (column address strobe)
      - The internal latches inside DRAM latch the second half



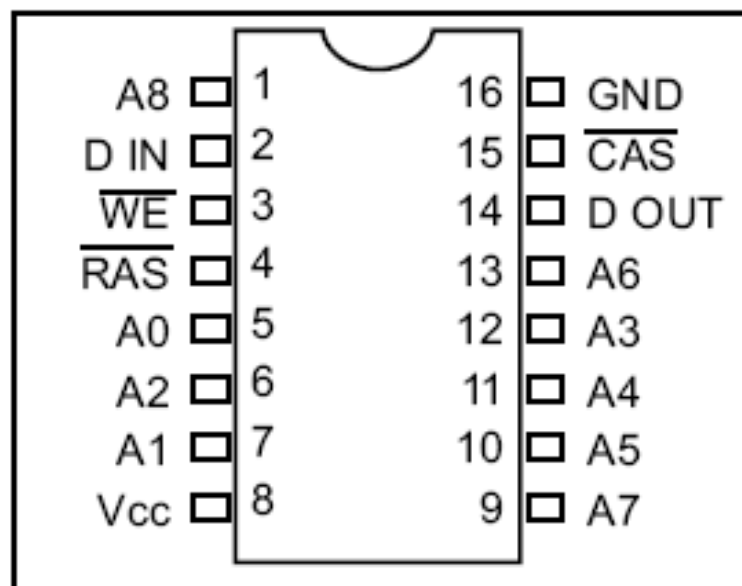


Figure 14-3. 256Kx1 DRAM

### Example 14-5

Discuss the number of pins set aside for addresses in each of the following memory chips.

(a) 16Kx4 DRAM

(b) 16Kx4 SRAM

#### Solution:

Since  $2^{14} = 16K$ :

- (a) For DRAM we have 7 pins (A0 - A6) for the address pins and 2 pins for RAS and CAS.
- (b) For SRAM we have 14 pins for address and no pins for RAS and CAS since they are associated only with DRAM. In both cases we have 4 pins for the data bus.

# Memory Address Decoding

- The CPU provides the address of the data
  - 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)
    - Must be activated for the memory's contents to be accessed
    - Sometimes the chip select is also referred to as chip enable (CE)
- In connecting a memory chip to the CPU, note the following points:

# Memory Address Decoding (cont.)

- 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:
  - 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 (cont.)

- Memories are divided into blocks
  - The output of the decoder selects a given memory block
    - Using simple logic gates
    - Using the 74LS138
    - Using programmable logics
- The simplest way of decoding circuitry is the use of NAND or other gates
  - The output of a NAND gate is active low
  - The CS pin is also active low
    - Makes them a perfect match

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

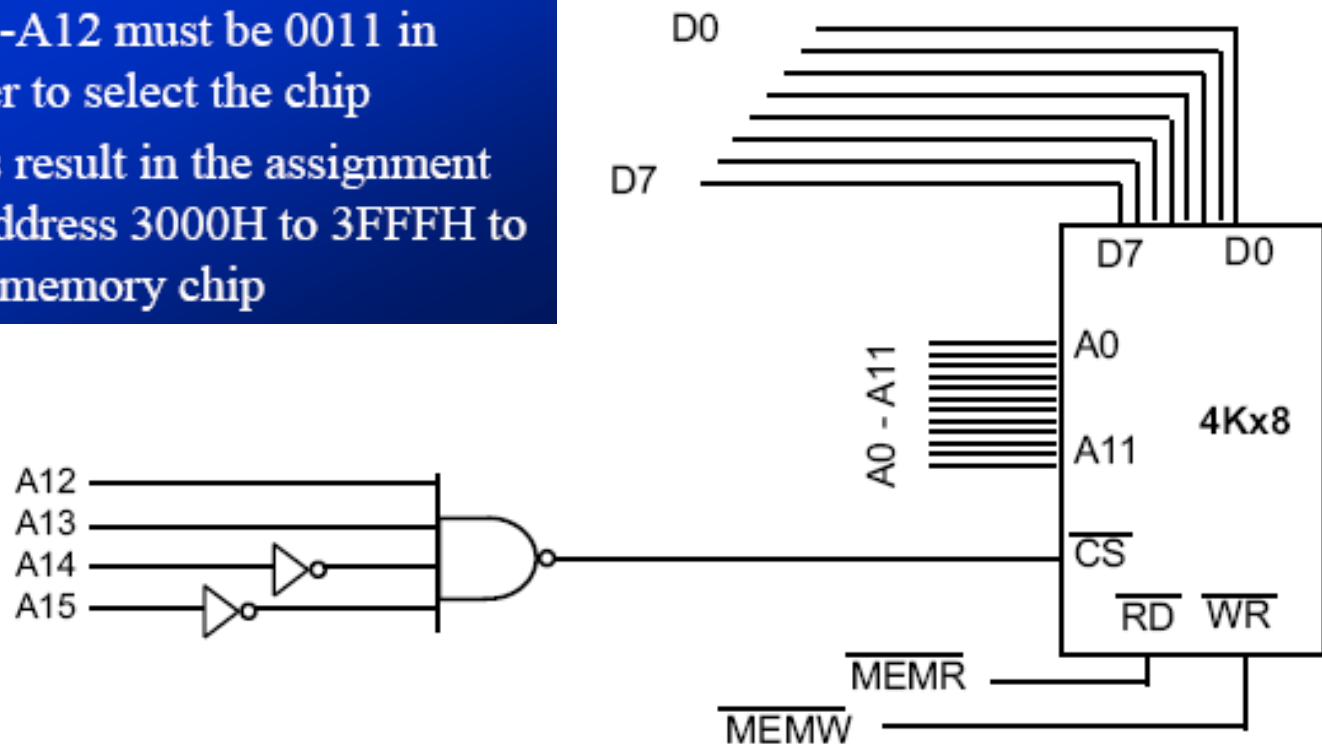


Figure 14-4. Logic Gate as Decoder

# 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
      - 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



# Using 74LS138 3-8 Decoder (cont.)

- If any one of the inputs  $G1$ ,  $G2A$ , or  $G2B$  is not connected to an address signal, they must be activated permanently either by  $V_{cc}$  or ground
  - Depending on the activation level

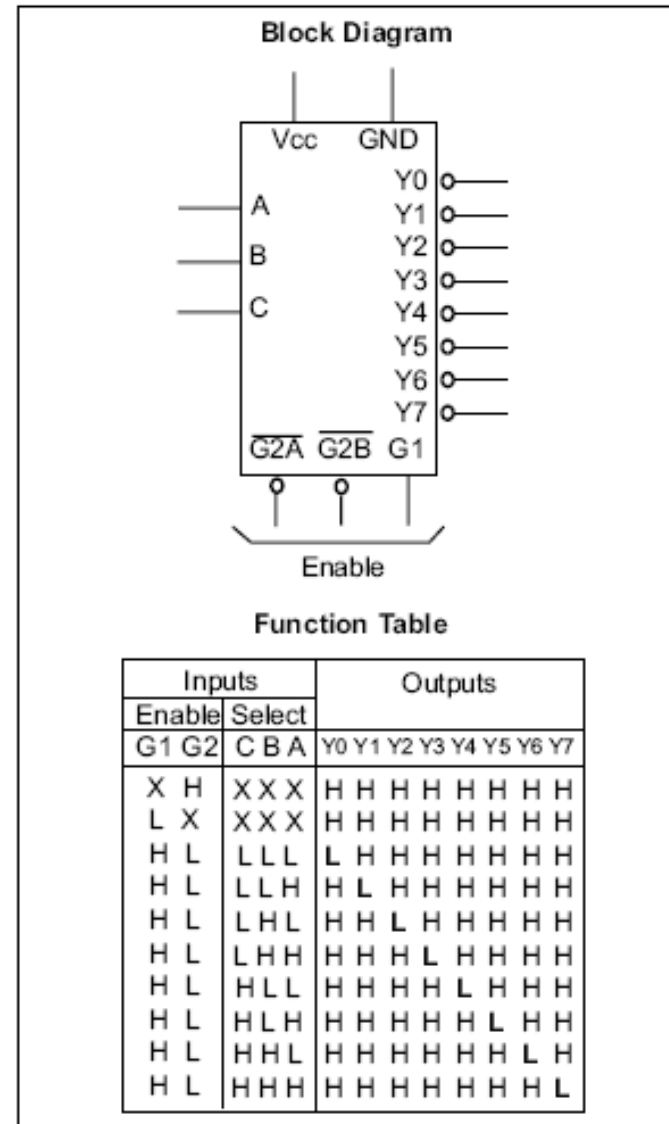


Figure 14-5. 74LS138 Decoder

(Reprinted by permission of Texas Instruments, Copyright Texas Instruments, 1988)



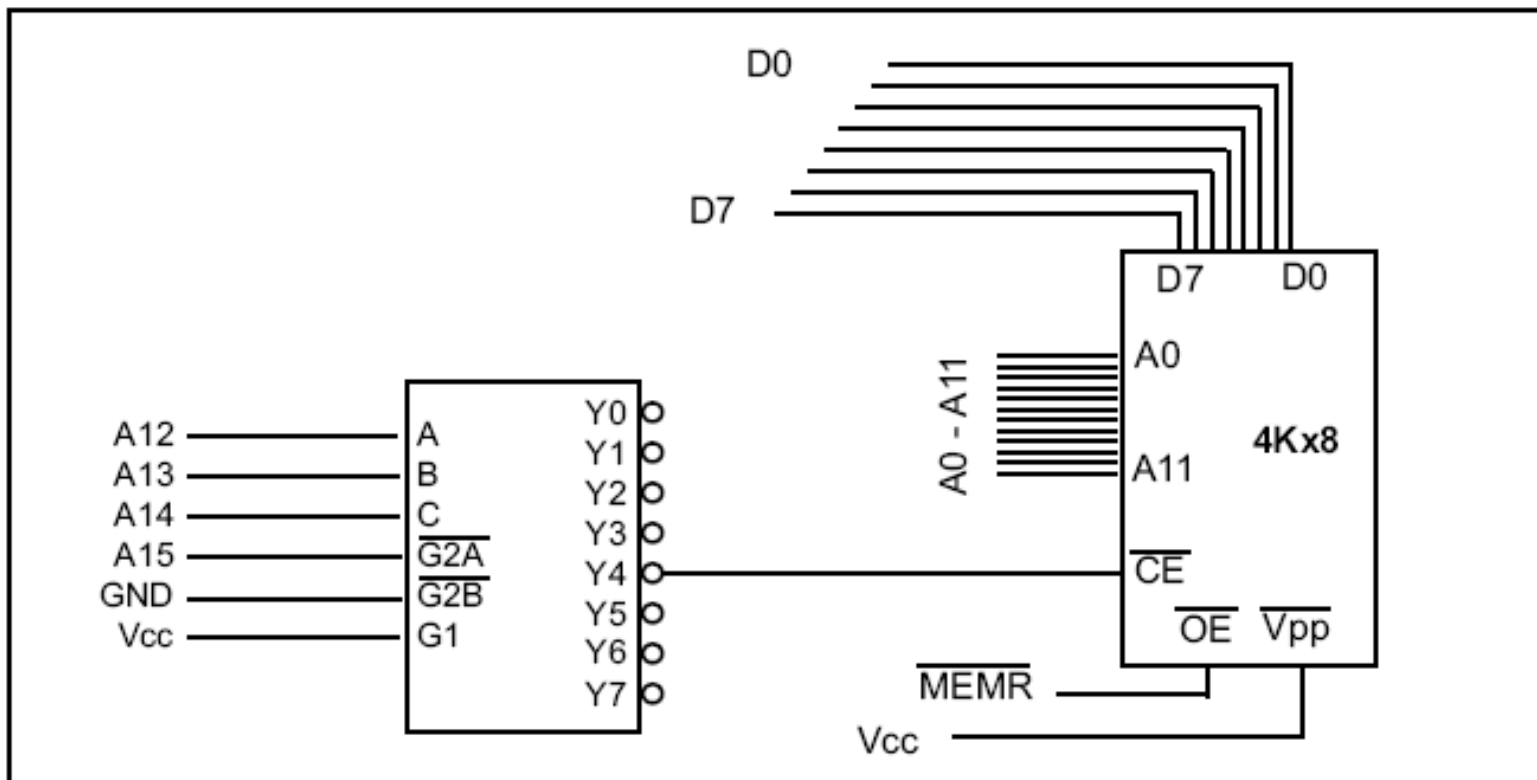


Figure 14-6. 74LS138 as Decoder



# 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
      - 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

# Interfacing External ROM

- The 8031 chip is a ROMless version of the 8051
  - It is exactly like any member of the 8051 family as far as executing the instructions and features are concerned
    - 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 8051 allows the program size to be as large as 64K bytes

# EA Pin

- For 8751/89C51/DS5000-based system, connect 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, EA must be connected to GND

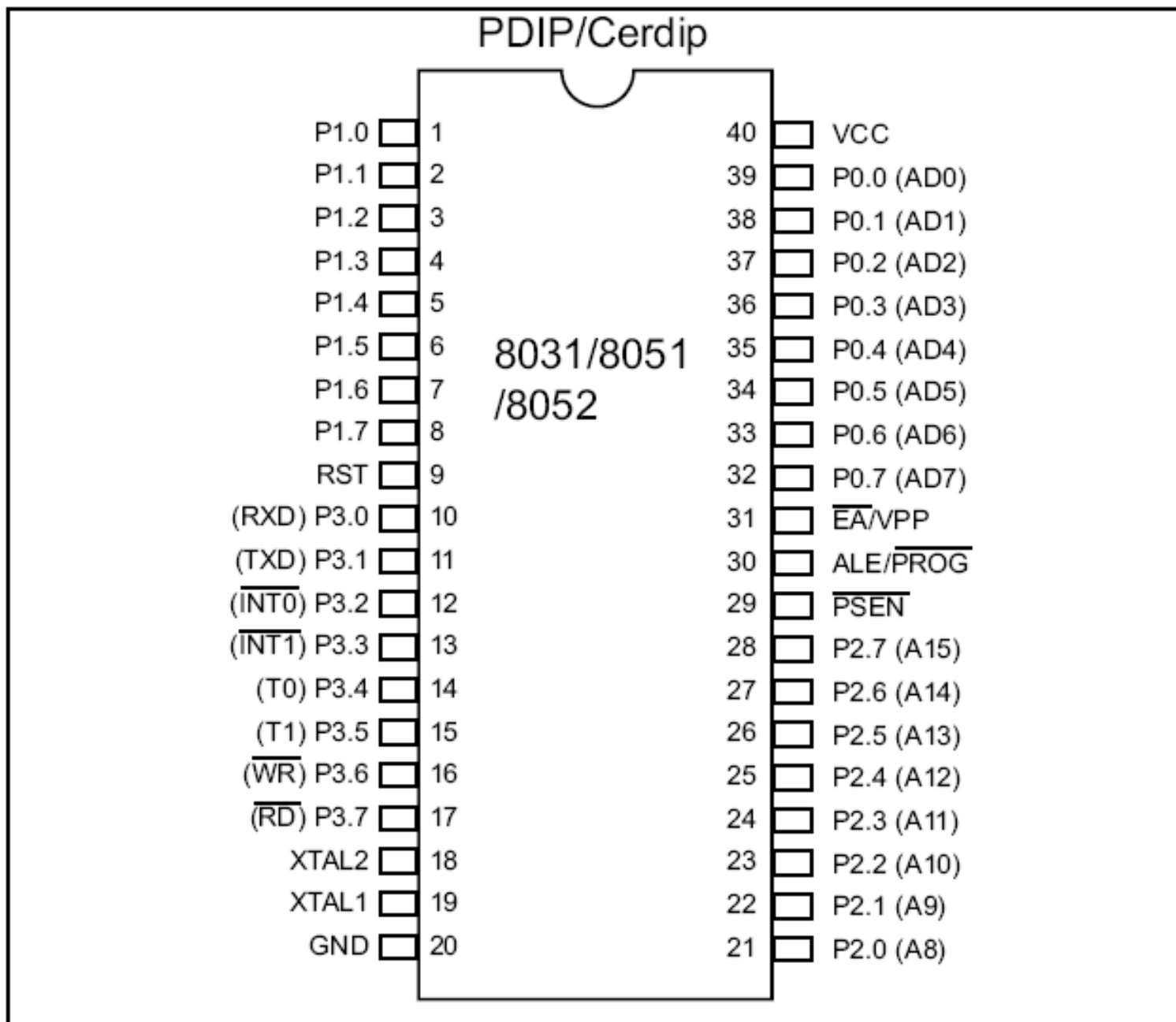


Figure 14-7. 8051 Pin Diagram

# P0 and P2 in Providing Address

- 8031/51 is capable of accessing up to 64K bytes of program code
  - Since the PC (program counter) is 16-bit
  - 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
    - P2 provides the upper 8 bit address A8 – A15
  - P0 is also used to provide the 8-bit data bus D0 – D7
    - P0 is used for both the address and data paths
      - Address/data multiplexing



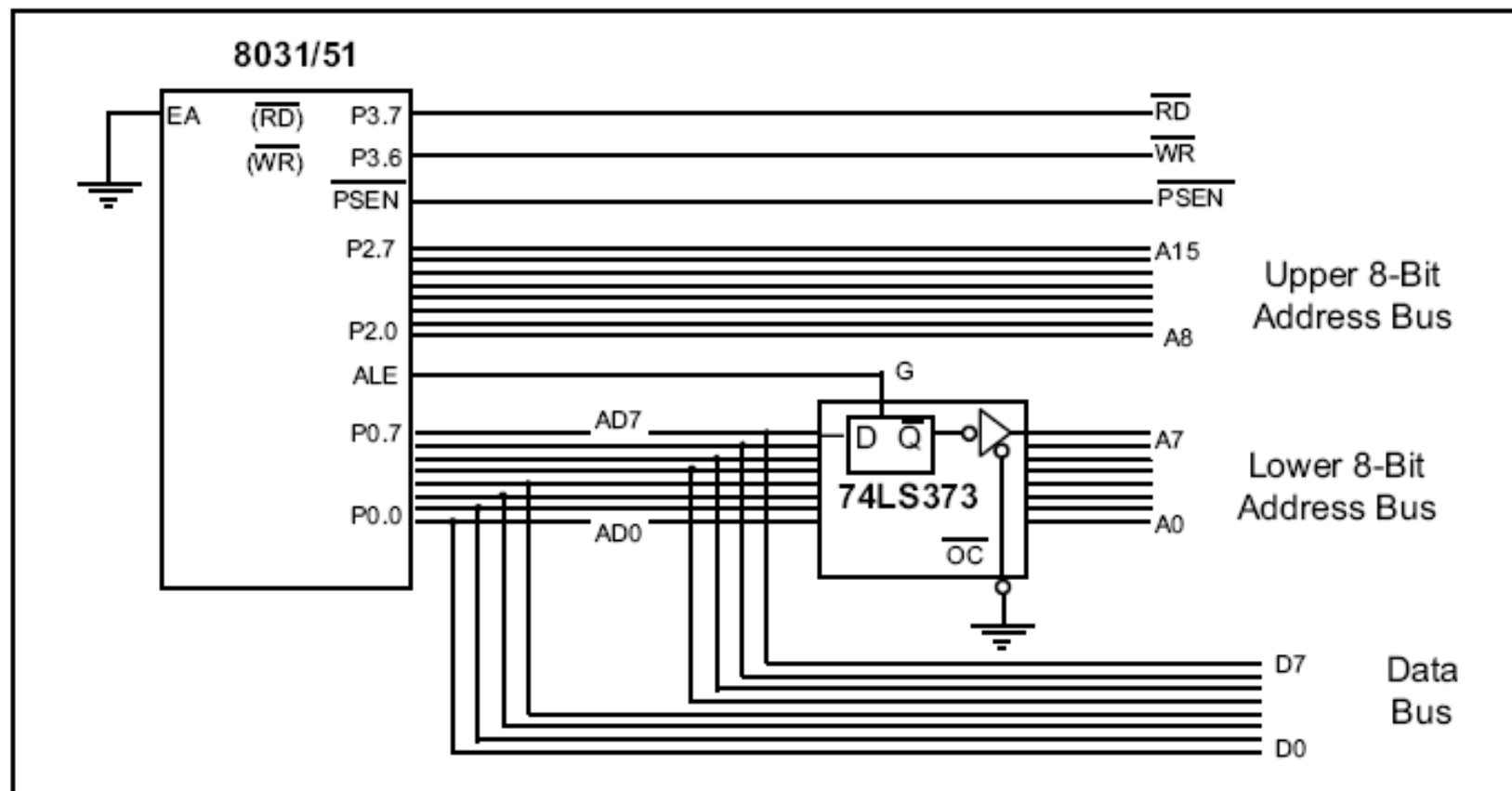
# ALE Pin

- 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
  - Connect P0 to a 74LS373
  - Use the ALE pin to latch the address
    - Normally  $ALE = 0$ , and P0 is used as a data bus
      - Sending data out or bringing data in
    - To use P0 as an address bus, it puts the addresses A0 – A7 on the P0 pins and activates  $ALE = 1$



# PSEN Pin

- PSEN (program store enable) signal is an output signal for the 8031/51
  - It must be connected to the OE pin of a ROM containing the program code
    - When the EA pin is connected to GND, the 8031/51 fetches opcode from external ROM by using PSEN
- In systems based on the 8751/89C51/DS5000 where EA is connected to  $V_{cc}$ 
  - These chips do not activate the PSEN pin
    - The on-chip ROM contains program code



**Figure 14-10. Data, Address, and Control Buses for the 8031**  
*(For reset and crystal connection, see Chapter 4.)*

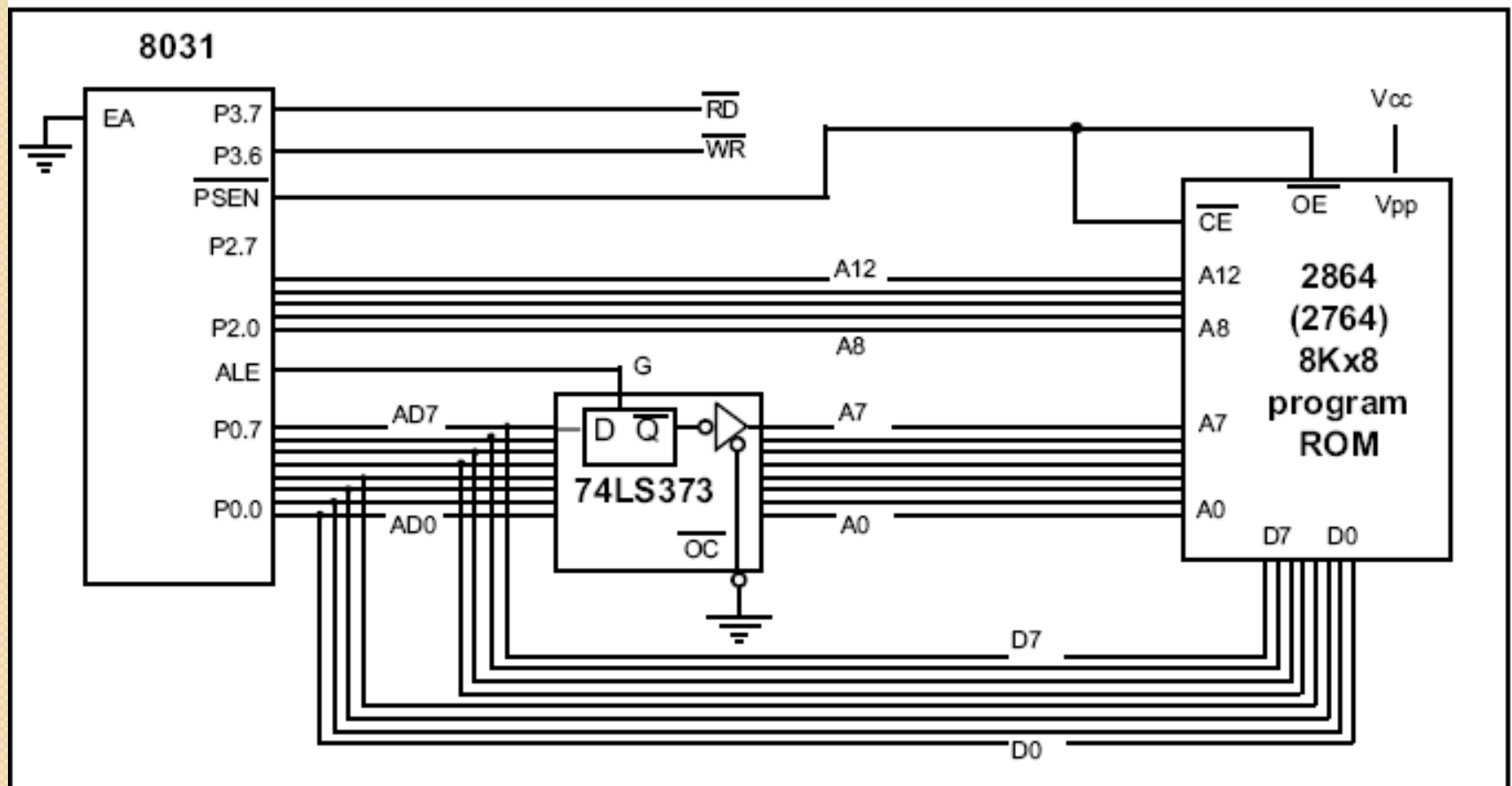


Figure 14-11. 8031 Connection to External Program ROM

# On-Chip and Off-Chip Code ROM

- In an 8751 (89C51) system we could use on-chip ROM for boot code and an external ROM will contain the user's program
  - We still have  $EA = V_{cc}$ ,
    - Upon reset 8051 executes the on-chip program first
    - When it reaches the end of the on-chip ROM, it switches to external ROM for rest of program

Discuss the program ROM space allocation for each of the following cases.

- (a)  $EA = 0$  for the 8751 (89C51) chip.
- (b)  $EA = V_{cc}$  with both on-chip and off-chip ROM for the 8751.
- (c)  $EA = V_{cc}$  with both on-chip and off-chip ROM for the 8752.

**Solution:**

- (a) When  $EA = 0$ , the EA pin is strapped to GND, and all program fetches are directed to external memory regardless of whether or not the 8751 has some on-chip ROM for program code. This external ROM can be as high as 64K bytes with address space of 0000 – FFFFH. In this case an 8751(89C51) is the same as the 8031 system.
- (b) With the 8751 (89C51) system where  $EA=V_{cc}$ , it fetches the program code of address 0000 – 0FFFH from on-chip ROM since it has 4K bytes of on-chip program ROM and any fetches from addresses 1000H – FFFFH are directed to external ROM.
- (c) With the 8752 (89C52) system where  $EA=V_{cc}$ , it fetches the program code of addresses 0000 – 1FFFH from on-chip ROM since it has 8K bytes of on-chip program ROM and any fetches from addresses 2000H – FFFFH are directed to external ROM



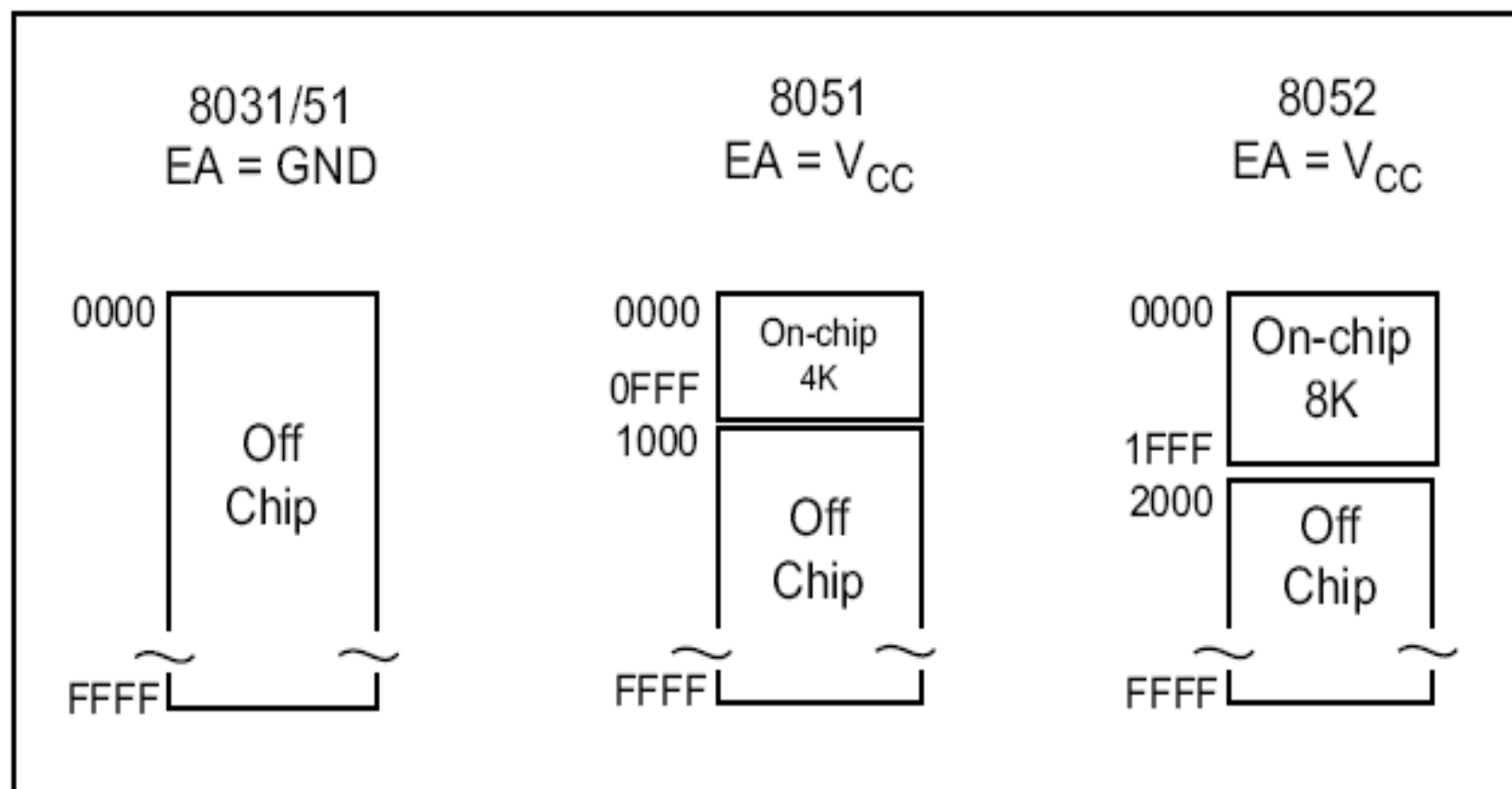


Figure 14-12. On-chip and Off-chip Program Code Access

# Data Memory Space

- 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
    - We can place data in the code space
      - 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`
      - X stands for eXternal – The data memory space must be implemented externally

# 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

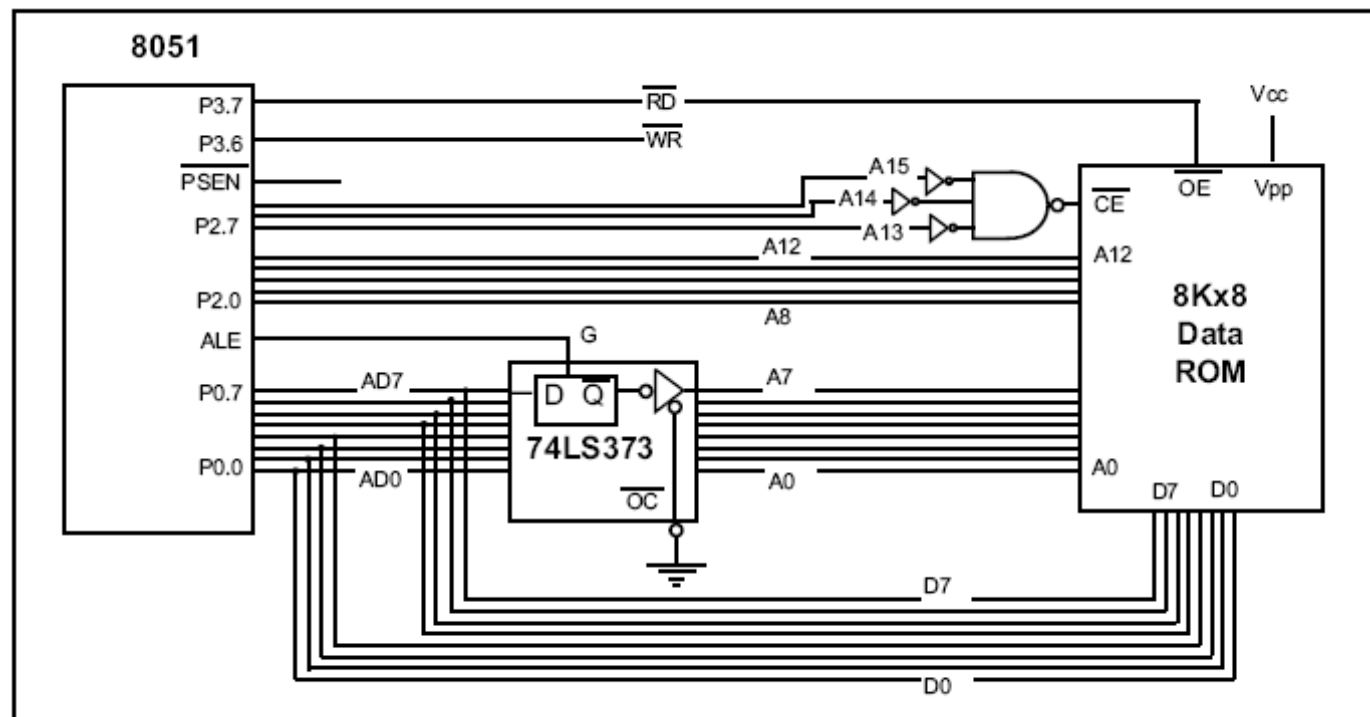


Figure 14-13. 8051 Connection to External Data ROM

# MOVX Instruction

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

## Example 14-9

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 them to P1.

### Solution:

```
MYXDATA    EQU    1000H
COUNT     EQU    30
            ...
            MOV    DPTR,#MYXDATA    ;pointer to external data
            MOV    R2,#COUNT        ;counter
AGAIN:      MOVX   A,@DPTR           ;get byte from external mem
            MOV    P1,A              ;issue it to P1
            INC    DPTR              ;next location
            DJNZ   R2,AGAIN          ;until all are read
```

### Example 14-10

External data ROM has a look-up table for the squares of numbers 0 - 9. Since the internal RAM of the 8031/51 has a shorter access time, write a program to copy the table elements into internal RAM starting at address 30H. The look-up table address starts at address 0 of external ROM.

#### Solution:

```
TABLE      EQU    000H
RAMTBLE    EQU    30H
COUNT     EQU    10

    . . .
    MOV     DPTR,#TABLE      ;pointer to external data
    MOV     R5,#COUNT      ;counter
    MOV     R0,#RAMTBLE     ;pointer to internal RAM
BACK:      MOVX  A,@DPTR      ;get byte from external mem
           MOV   @R0,A        ;store it in internal RAM
           INC   DPTR         ;next data location
           INC   R0           ;next RAM location
           DJNZ  R5,BACK      ;until all are read
```

### Example 14-11

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 activate OE, while CE is activated by a simple decoder.

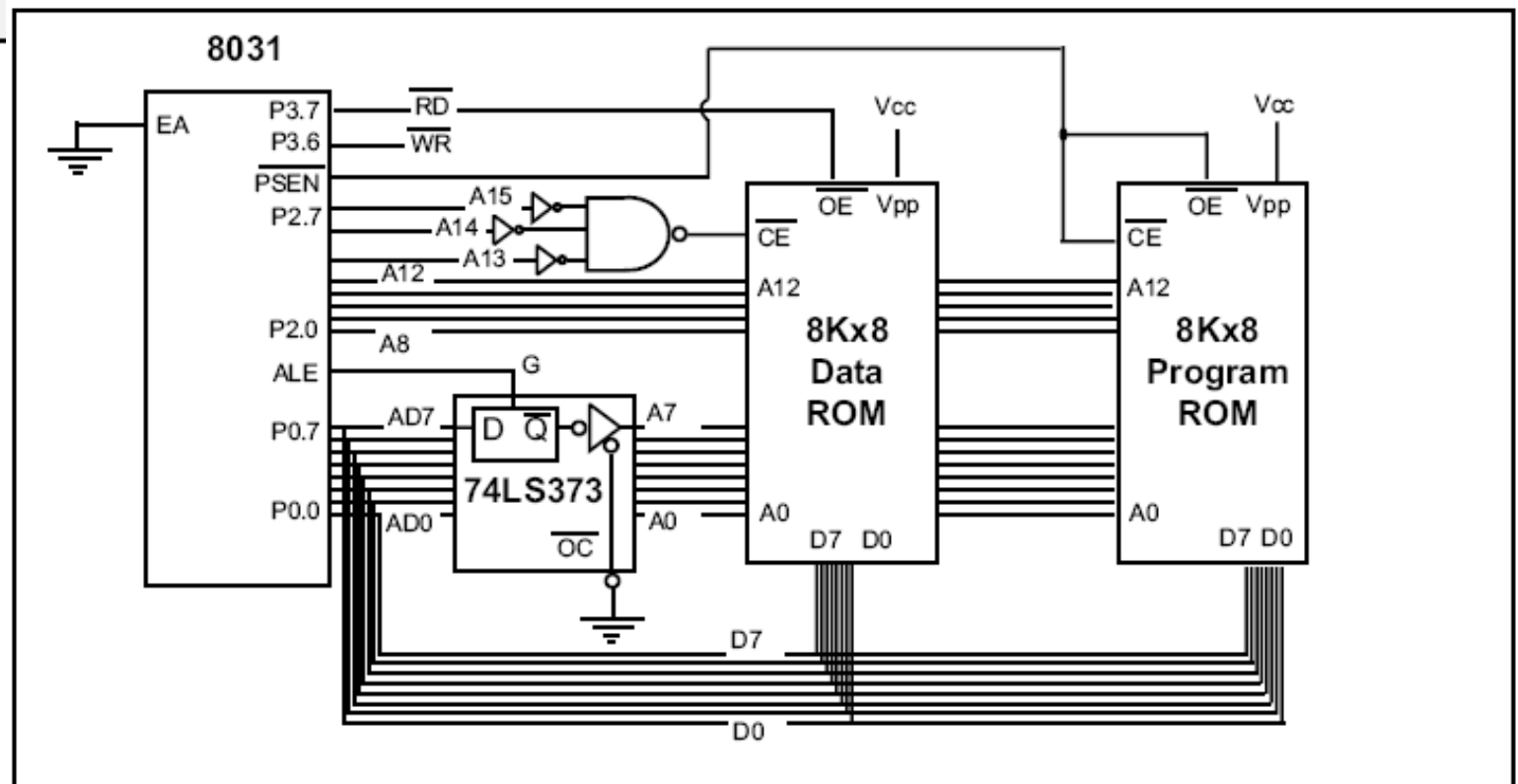


Figure 14-14. 8031 Connection to External Data ROM and External Program ROM



# External Data RAM

- To connect the 8051 to an external SRAM, we must use both RD (P3.7) and WR (P3.6)
- In writing data to external data RAM, we use the instruction `MOVX @DPTR,A`



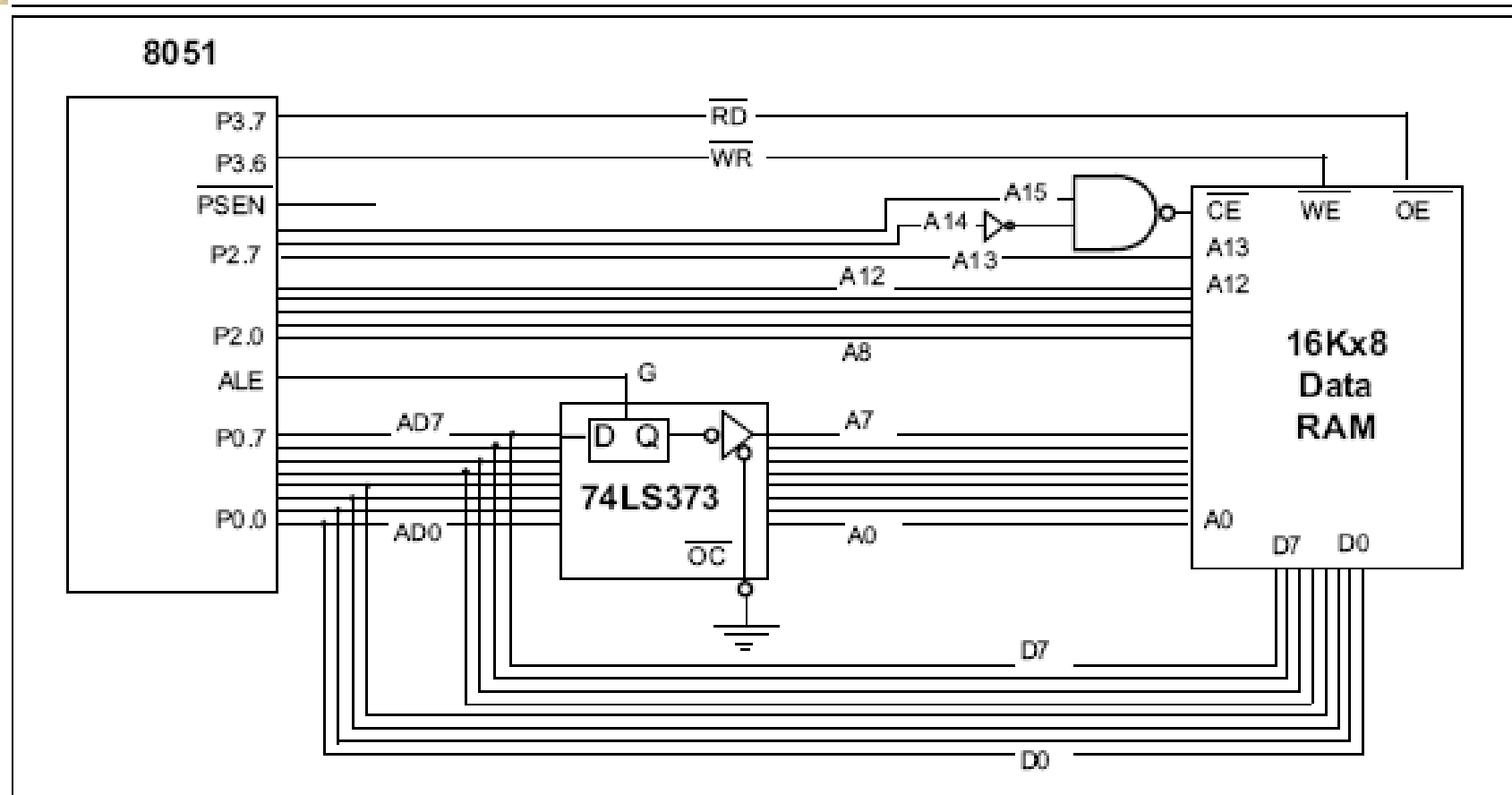


Figure 14-15. 8051 Connection to External Data RAM

### Example 14-12

- (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
```

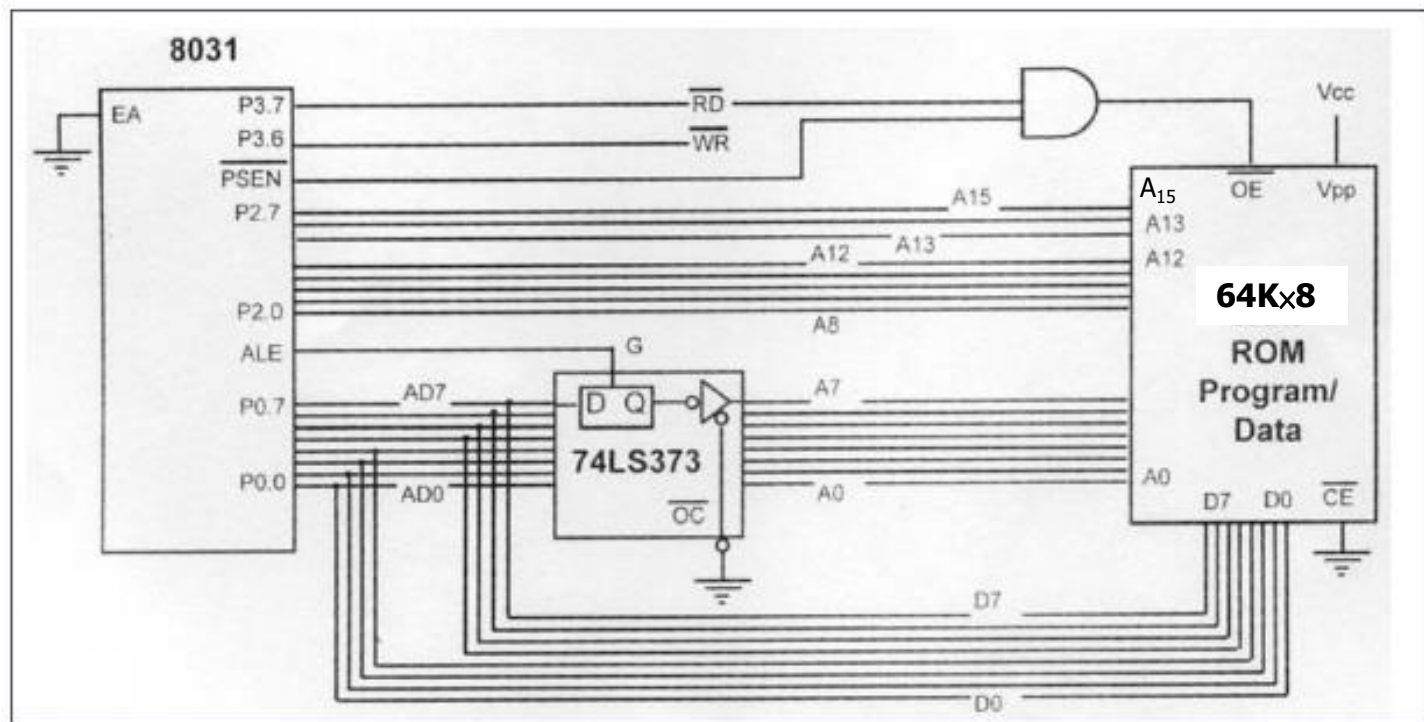
```

                MOV    DPTR,#RAMDATA ;pointer to external NV-RAM
                MOV    R3,#COUNT    ;counter
AGAIN:          MOV    A,P1           ;read data from P1
                MOVX   @DPTR,A        ;save it external NV-RAM
                ACALL  DELAY           ;wait before next sample
                INC    DPTR           ;next data location
                DJNZ   R3,AGAIN        ;until all are read
HERE:           SJMP   HERE           ;stay here when finished
```

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

# Single External ROM for Code and Data

- An 8031-based system connected to a single  $64\text{K} \times 8$  (27512) external ROM chip
  - The single external ROM chip is used for both program code and data storage
    - For example, the space 0000 – 7FFFH is allocated to program code
    - Address space 8000H – FFFFH is set aside for data
    - In accessing the data, we use the MOVX instruction
  - To allow a single ROM chip to provide both program code space and data space, we use an AND gate to signal the OE

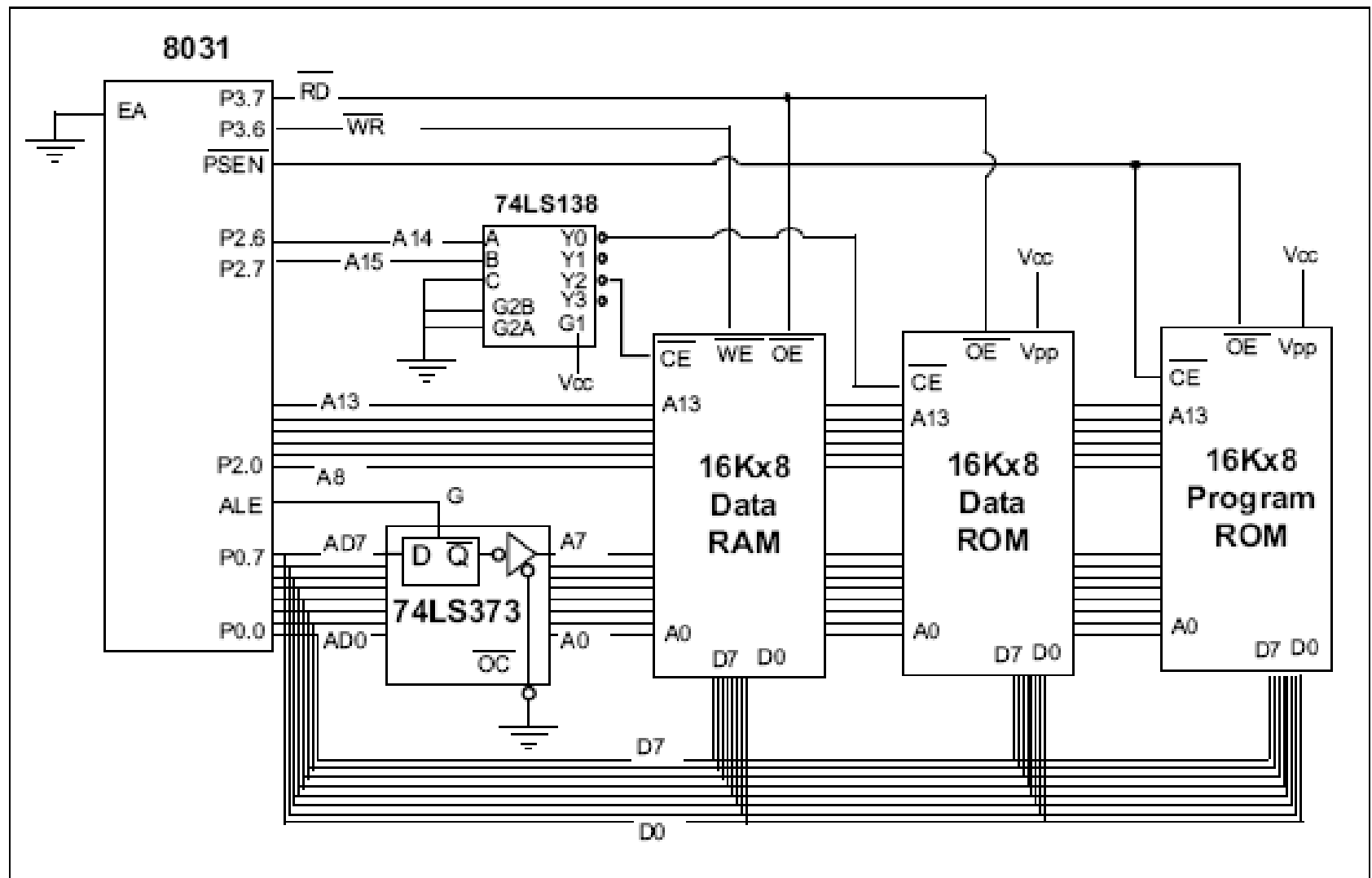


### Example 14-13

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  $G1 = V_{cc}$ ,  $G2A = GND$ ,  $G2B = GND$ , and the C input of the 74LS138 is also grounded since we use Y0 - Y3 only.

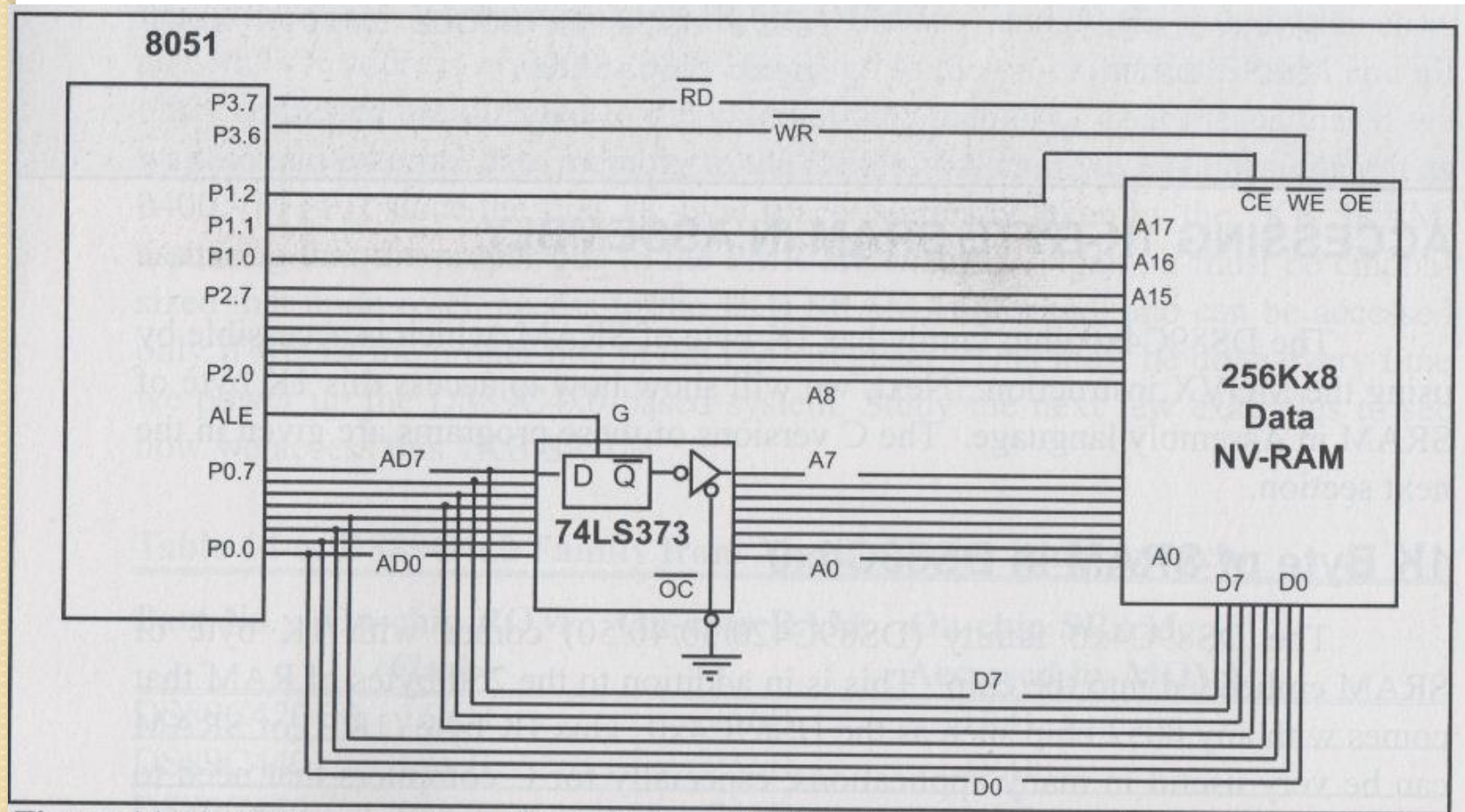


**Figure 14-17. 8031 Connection to External Program ROM, Data RAM, and Data ROM**

# Interfacing to Large External Memory

- In some applications we need a large amount of memory to store data
  - The 8051 can support only 64K bytes of external data memory since DPTR is 16-bit
  - To solve this problem
    - Connect A0 – A15 of the 8051 directly to the external memory's A0 – A15 pins
    - Use some of the P1 pins to access the 64K bytes blocks inside the single 256K×8 memory chip





**Figure 14-18. 8051 Accessing 256Kx8 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 :

<b>Chip select</b>	<b>A17</b>	<b>A16</b>	<b>Block address space</b>
<b>P1.2</b>	<b>P1.1</b>	<b>P1.0</b>	
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
MOVB     @DPTR,A
INC      DPTR       ;next location
...
```