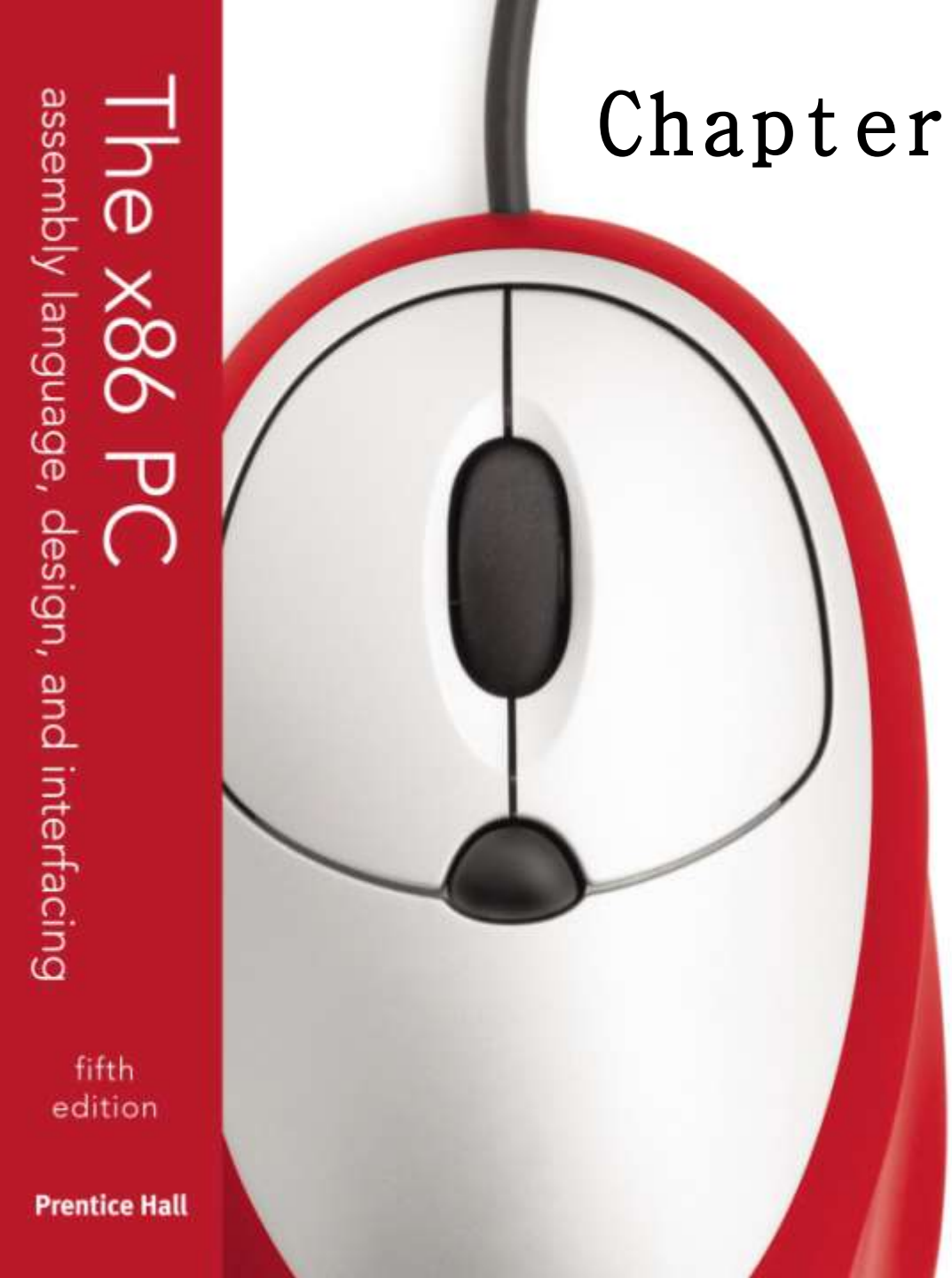


Chapter ELEVEN

8255 I/O PROGRAMMING

The x86 PC

assembly language, design, and interfacing

fifth edition

MUHAMMAD ALI MAZIDI
JANICE GILLISPIE MAZIDI
DANNY CAUSEY

OBJECTIVES

this chapter enables the student to:

- Code Assembly language instructions to read and write data to and from I/O ports.
- Diagram the design of peripheral I/O using the 74LS373 output latch and the 74LS244 input buffer.
- Describe the I/O address map of x86 PCs.
- List the differences in memory-mapped I/O versus peripheral I/O.
- Describe the purpose of the 8255 programmable peripheral interface chip.

11.1: 8088 INPUT/OUTPUT INSTRUCTIONS

- All x86 processors, 8088 to Pentium®, can access external devices called *ports* using I/O instructions.
  - Memory can contain both Instructions and data.
  - I/O ports contain data only
    - Two instructions: “OUT” and “IN” send data from the accumulator (AL or AX) to ports or bring data from ports into the accumulator.
- 8088 I/O operation is applicable to all x86 CPUs.
  - The 8-bit port uses the D0–D7 data bus for I/O devices.

11.1: 8088 INPUT/OUTPUT INSTRUCTIONS
8-bit data ports

- Register **AL** is used as the source/destination for IN/OUT instructions.
  - To input or output data from any other registers, the data must first be moved to the AL register.
    - Instructions OUT and IN have the following formats:

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

11.1: 8088 INPUT/OUTPUT INSTRUCTIONS
how to use I/O instructions

- I/O instructions are used in programming 8- and 16-bit peripheral devices.
  - Printers, hard disks, and keyboards.
- For an 8-bit port, use *immediate addressing*:
  - For more ports, use 16-bit port address instruction.

```
MOV    AL,36H        ;AL=36H
OUT    43H,AL        ;send value 36H to port address 43H
```

11.1: 8088 INPUT/OUTPUT INSTRUCTIONS
how to use I/O instructions

- 16-bit port address instruction using *register indirect addressing* mode with register **DX**.
  - This program toggles port address 300H continuously.

```
BACK: MOV   DX,300H      ;DX = port address 300H
      MOV   AL,55H
      OUT   DX,AL        ;toggle the bits
      MOV   AL,0AAH
      OUT   DX,AL        ;toggle the bits
      JMP   BACK
```

  - Only **DX** can be used for 16-bit I/O addresses.
  - Use register AL for 8-bit data.

11.1: 8088 INPUT/OUTPUT INSTRUCTIONS
how to use I/O instructions

Example 11-1 shows decision making based on the data that was input.

Example 11-1

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:

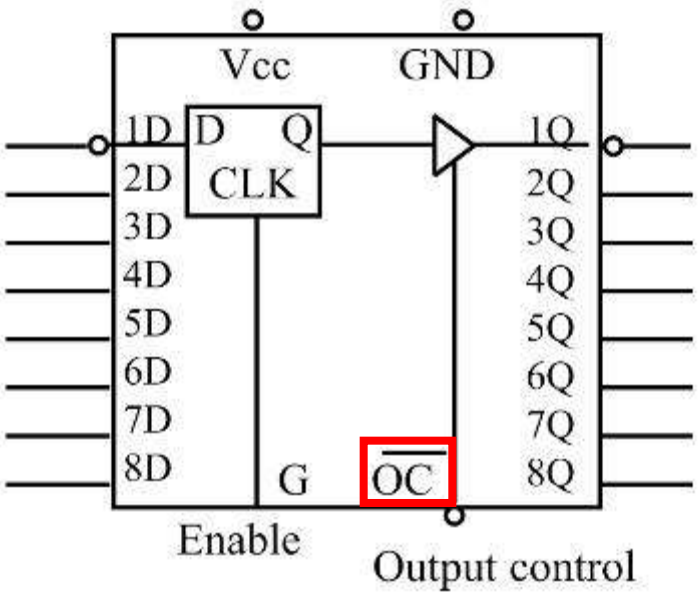
```
BACK:   IN    AL,22H   ;get the temperature from port # 22H
        CMP   AL,100       ;is temp = 100?
        JNZ   BACK         ;if not, keep monitoring
        MOV   BH,'Y        ;temp = 100, load 'Y' into BH
```

11.2: I/O ADDRESS DECODING AND DESIGN

- The concept of address bus decoding for I/O instructions is exactly the same as for memory.
  - 1. The control signals $\overline{\textbf{IOR}}$ and $\overline{\textbf{IOW}}$ are used along with the decoder.
  - 2. For an 8-bit port address, **A0–A7** is decoded.
  - 3. If the port address is 16-bit (using **DX**), **A0–A15** is decoded.

11.2: I/O ADDRESS DECODING AND DESIGN using 74LS373 in an output port design

- 74LS373 can be used as a latching system for simple I/O ports.
  - Pin **OC** must be grounded.



Function Table

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

Figure 11-1 74LS373 D Latch

11.2: I/O ADDRESS DECODING AND DESIGN using 74LS373 in an output port design

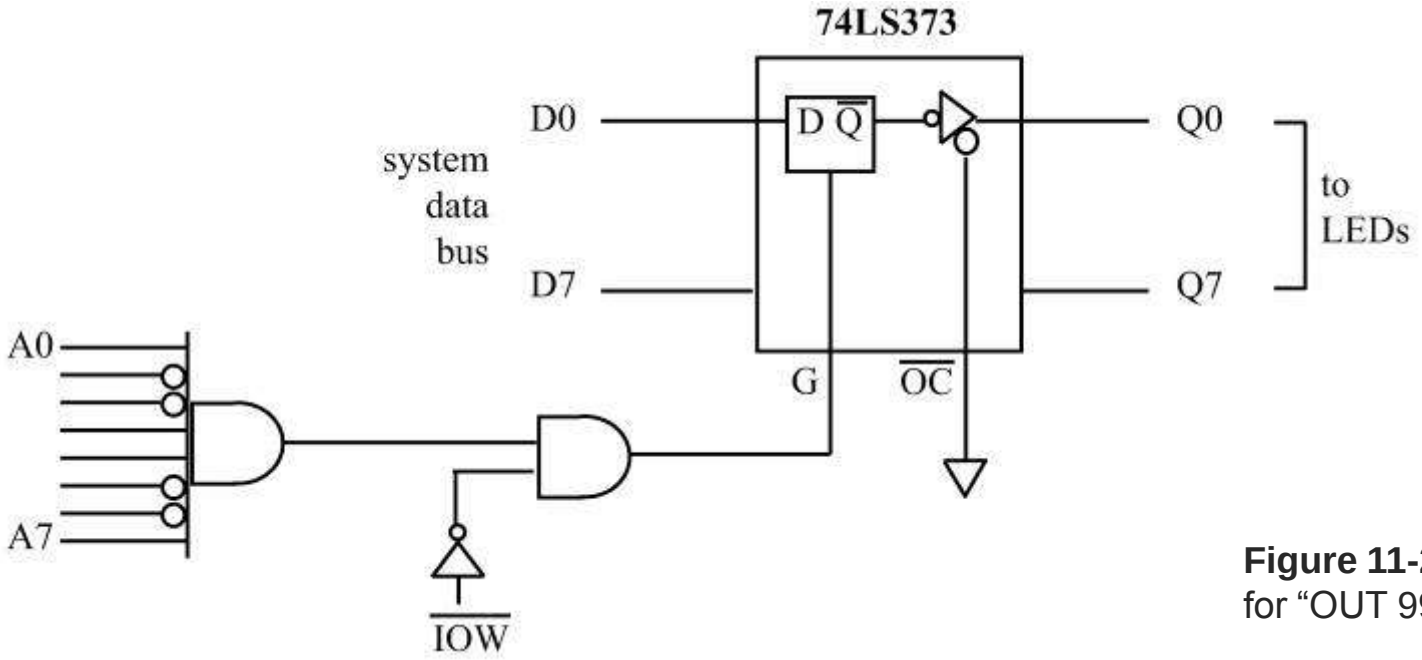


Figure 11-2 Design for "OUT 99H, AL".

- For an output latch, it is common to AND the output of the address decoder with control signal **IOW**.
  - To provide the latching action.

11.2: I/O ADDRESS DECODING AND DESIGN using 74LS373 in an output port design

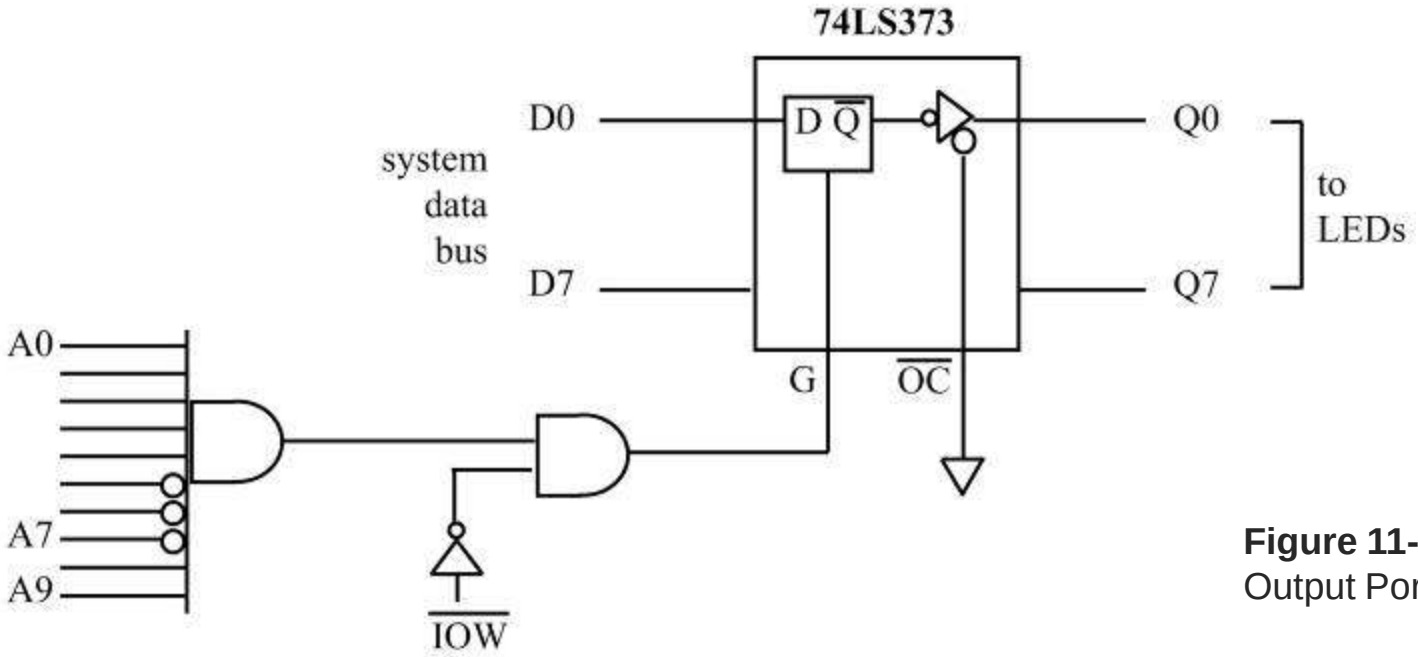


Figure 11-3 Design for Output Port Address 31H.

Example 11-2

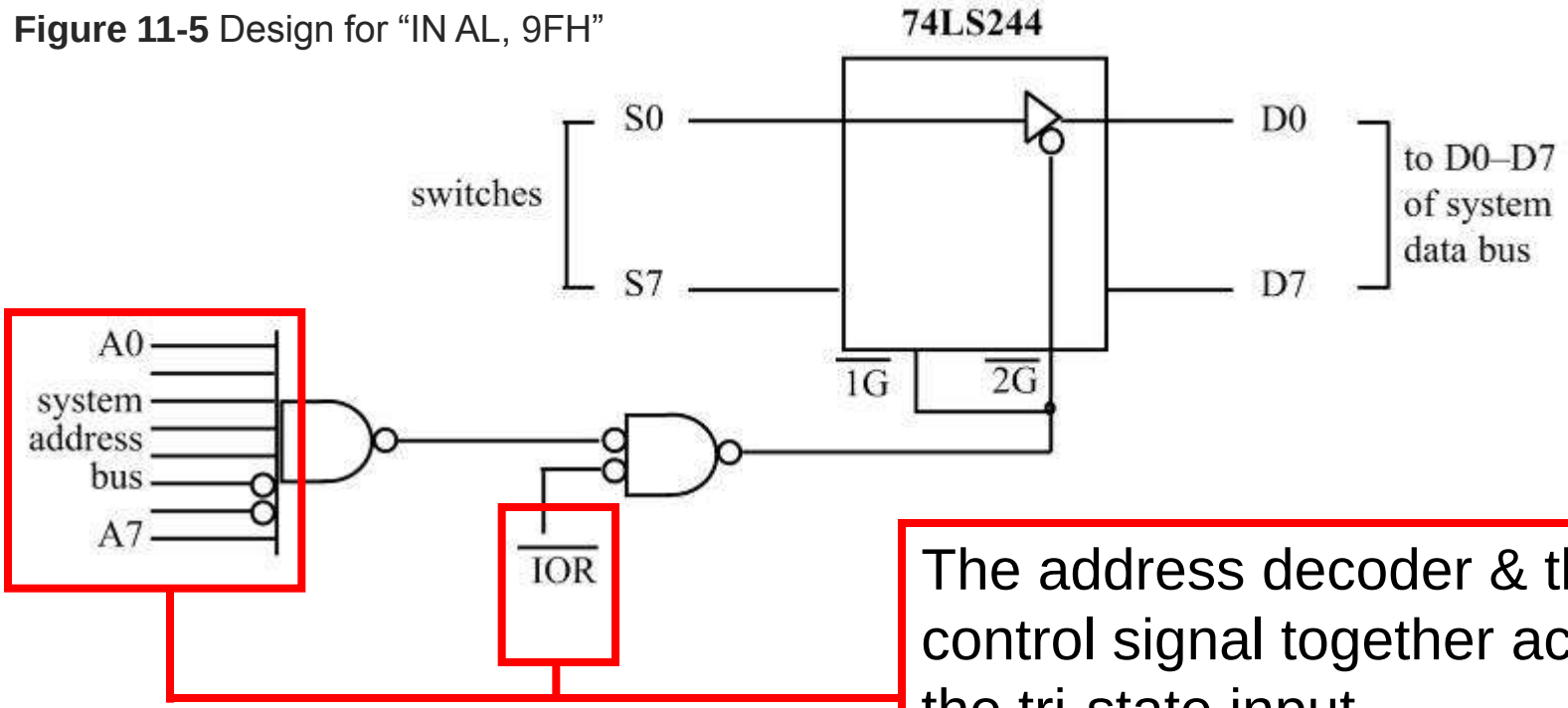
Show the design of an output port with an I/O address of 31FH using the 74LS373.

Solution:

31F9H is decoded, then ANDed with IOW to activate the G pin of the 74LS373 latch. This is shown in Figure 11-3.

11.2: I/O ADDRESS DECODING AND DESIGN
IN port design using 74LS244

Figure 11-5 Design for "IN AL, 9FH"



The address decoder & the **IOR** control signal together activate the tri-state input.

- 74LS244 is widely used for buffering and providing high driving capability for unidirectional buses.

11.2: I/O ADDRESS DECODING AND DESIGN
memory-mapped I/O

- Some processors do not have IN & OUT instructions, but use *Memory-mapped I/O*.
  - A memory location is assigned as an input or output port.
  - Instructions access memory locations to access I/O ports.
    - Instead of **IN** and **OUT** instructions.
  - The entire 20-bit address, **A0–A19**, must be decoded.
    - The **DS** register must be loaded prior to accessing memory-mapped I/O.
  - In memory-mapped I/O interfacing, control signals **MEMR** and **MEMW** are used.
    - Memory I/O ports can number as high as 2^{20} (1,048,576).

11.2: I/O ADDRESS DECODING AND DESIGN
memory-mapped I/O

- Some processors do not have IN & OUT instructions, but use *Memory-mapped I/O*.
  - Memory-mapped I/O can perform arithmetic & logic operations on I/O data directly without first moving them into the accumulator.
  - Memory-mapped I/O uses memory address space, which could lead to memory space fragmentation.

Example 11-3

Show the design of "IN AL,9FH" using the 74LS244 as a tri-state buffer.

Solution:

9FH is decoded, then ANDed with IOR. To activate OC of the 74LS244, it must be inverted since OC is an active-low pin. This is shown in Figure 11-5.

11.3: I/O ADDRESS MAP OF x86 PCs

Hex Range	Device
000–01F	DMA controller 1, 8237A-5
020–03F	Interrupt controller 1, 8259A, Master
040–05F	Timer, 8254-2
060–06F	8042 (keyboard)
070–07F	Real-time clock, NMI mask
080–09F	DMA page register, 74LS612
0A0–0BF	Interrupt controller 2, 8237A-5
0C0–0DF	DMA controller 2, 8237A-5
0F0	Clear math coprocessor busy
0F1	Reset math coprocessor
0F8–0FF	Math coprocessor
1F0–1F8	Fixed disk
200–207	Game I/O
20C–20D	Reserved
21F	Reserved
278–27F	Parallel printer port 2
2B0–2DF	Alternate enhanced graphics adapter
2E1	GPIB (adapter 0)
2E2 & 2E3	Data acquisition (adapter 0)
2F8–2FF	Serial port 2
300–31F	Prototype card
360–363	PC network (low address)
364–367	Reserved
368–36B	PC network (high address)
36C–36F	Reserved

Hex Range	Device
378–37F	Parallel printer port 1
380–38F	SDLC, bisynchronous 2
390–393	Cluster
3A0–3AF	Bisynchronous 1
3B0–3BF	Monochrome display/printer adapter
3C0–3CF	Enhanced graphics adapter
3D0–3DF	Color graphics monitor adapter
3F0–3F7	Disk controller
3F8–3FF	Serial port 1
6E2 & 6E3	Data acquisition (adapter 1)
790–793	Cluster (adapter 1)
AE2 & AE3	Data acquisition (adapter 2)
B90–B93	Cluster (adapter 2)
EE2 & EE3	Data acquisition (adapter 3)
1390–1393	Cluster (adapter 3)
22E1	GPIB (adapter 1)
2390–2393	Cluster (adapter 4)
42E1	GPIB (adapter 2)
62E1	GPIB (adapter 3)
82E1	GPIB (adapter 4)
A2E1	GPIB (adapter 5)
C2E1	GPIB (adapter 6)
E2E1	GPIB (adapter 7)

See the entire I/O map on page 296 of your textbook.

11.3: I/O ADDRESS MAP OF x86 PCs absolute vs. linear address decoding

- In decoding addresses, either all or a selected number of them are decoded.
  - In *absolute* decoding, all address lines are decoded.
  - If only selected address pins are decoded, it is called *linear select* decoding.
- Linear select is cheaper, but creates aliases, the same port with multiple addresses.
  - If you see a large gap in the I/O address map of the x86 PC, it is due to the address aliases of the original PC.

11.3: I/O ADDRESS MAP OF x86 PCs
prototype addresses 300-31FH in x86 PC

- Prototype cards at **300H–31FH** can be data acquisition boards used to monitor analog signals.
  - Temperature, pressure, etc., inputs use signals on the 62-pin section of the ISA expansion slot.

IOR and **IOW**. Both *active-low*.

When **AEN** = 0, the CPU is using the bus.

A0–A9 for address decoding.

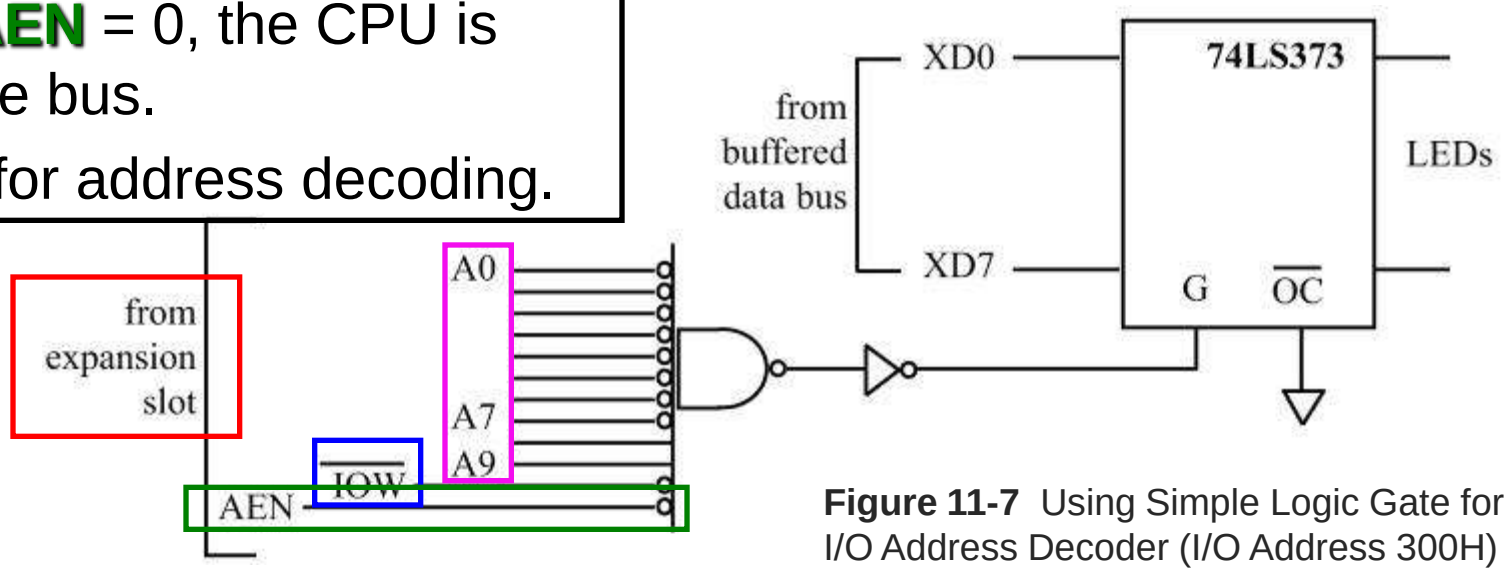
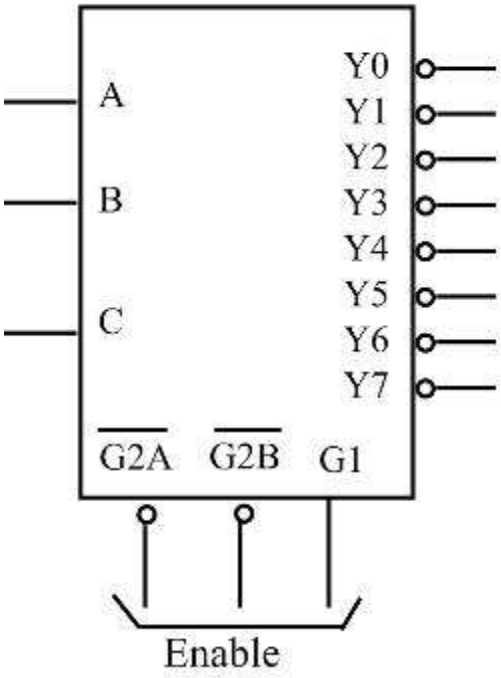


Figure 11-7 Using Simple Logic Gate for I/O Address Decoder (I/O Address 300H)

11.3: I/O ADDRESS MAP OF x86 PCs
74LS138 as a decoder

- NANDs, inverters, and 74LS138 chips for decoders can be applied to I/O address decoding.

Block Diagram



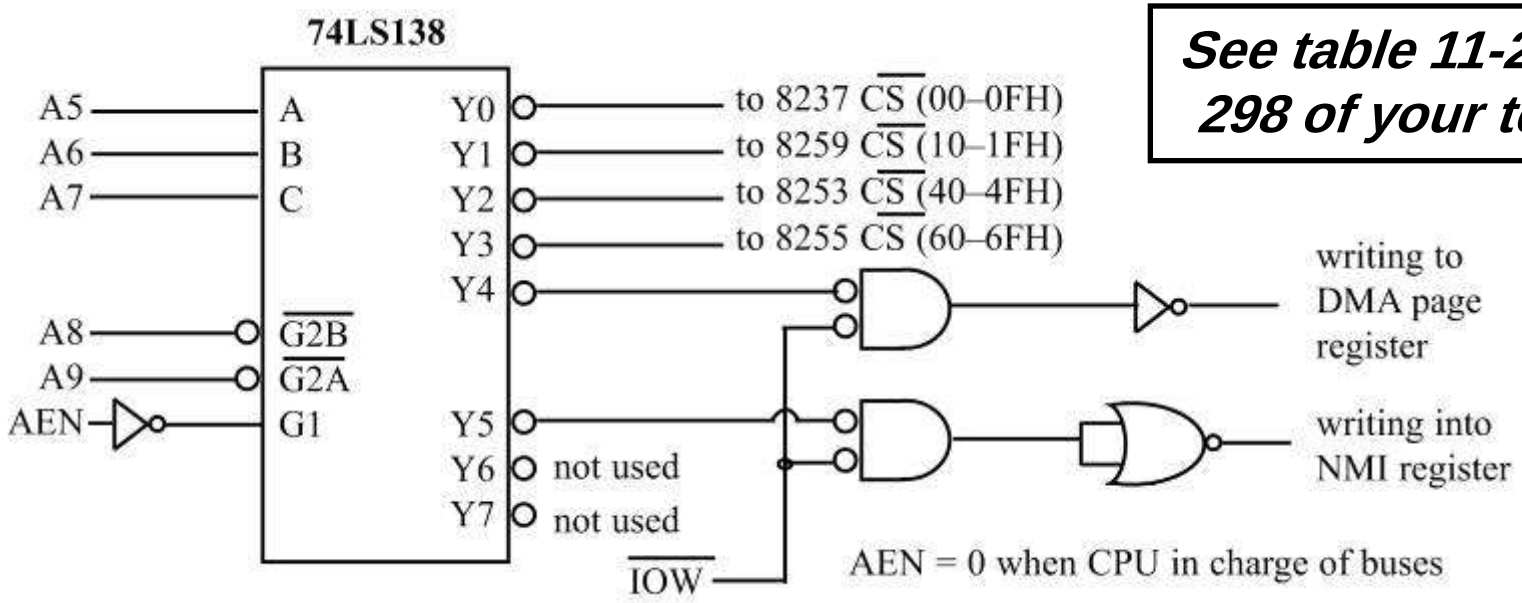
Function Table

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

Figure 11-8 74S138 Decoder

11.3: I/O ADDRESS MAP OF x86 PCs
74LS138 as IBM PC I/O address decoder

- A0 to A4 go to individual peripheral input addresses.
- A5, A6, & A7 handle output selection of outputs Y0 to Y7.
- Pins A8, A9, & AEN all must be *low* to enable 74LS138.
  - AEN is low only when the x86 is in control of the system bus.



See table 11-2 on page 298 of your textbook.

Figure 11-10 PC/XT Port Address Coding

11.4: PROGRAMMING & INTERFACING THE 8255

- The 8255 is a widely used 40-pin, DIP I/O chip.
  - It has three separately accessible programmed ports, A, B & C.
  - Each port can be programmed to be input or output.
  - Ports can also be changed dynamically.

Port A (PA0–PA7)
Port B (PB0–PB7)
Port C (PC0–PC7)

These 8-bit ports can be all input *or* all output.

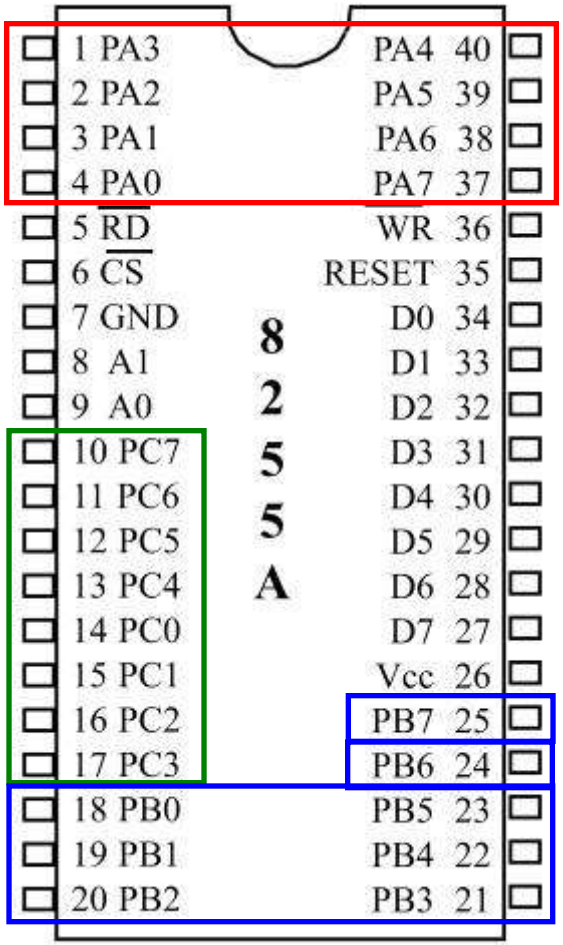
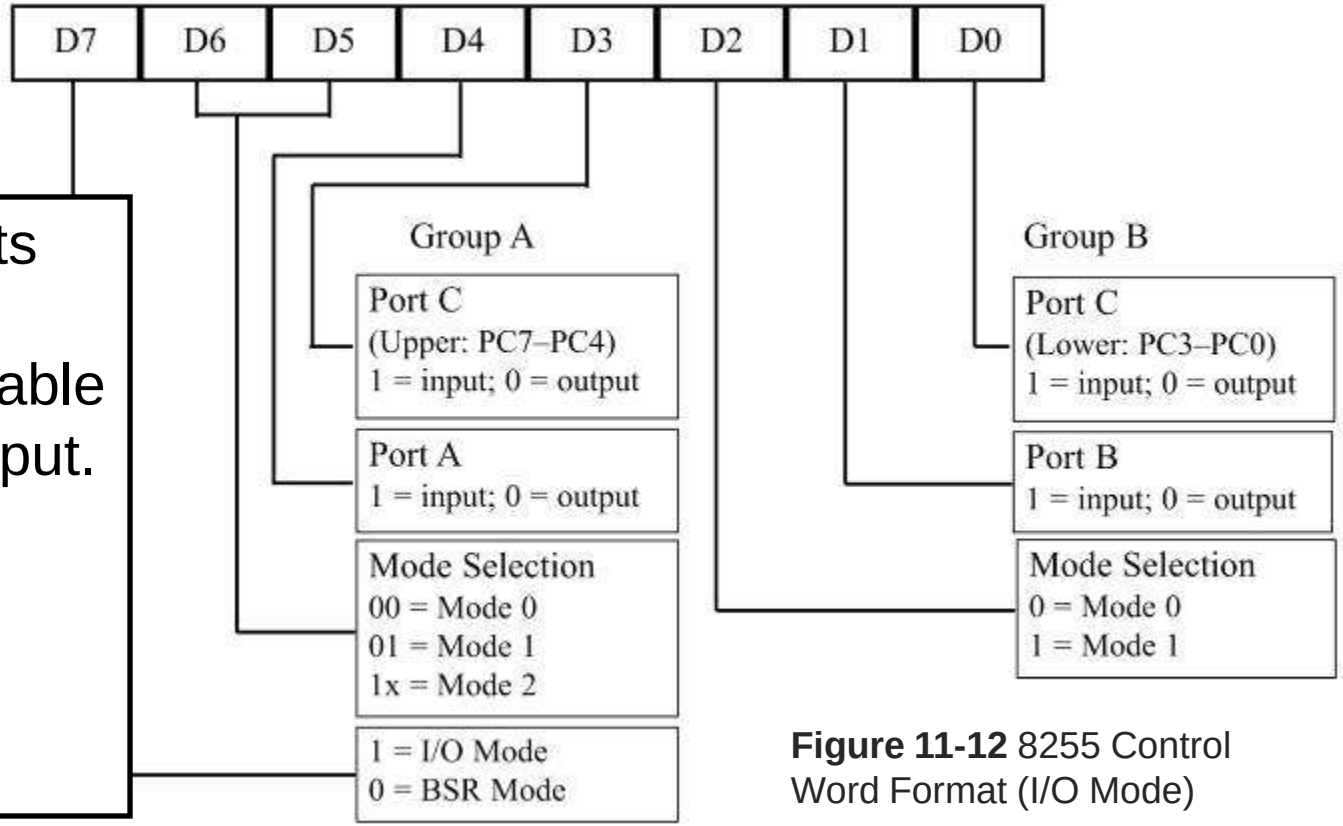


Figure 11-11 8255 PPI Chip

11.4: PROGRAMMING & INTERFACING THE 8255
mode selection of the 8255A

- 8255 ports can be programmed in various modes.
  - The *simple I/O mode*, Mode 0, is most widely used.



Any of the ports A, B, CL & CU are programmable as input *or* output.

All bits are out or all are in.

No control of individual bits

Figure 11-12 8255 Control Word Format (I/O Mode)

ENDS ; ELEVEN

The x86 PC

assembly language, design, and interfacing

fifth edition

Prentice Hall

The x86 PC

assembly language, design, and interfacing

fifth edition

MUHAMMAD ALI MAZIDI
JANICE GILLISPIE MAZIDI
DANNY CAUSEY