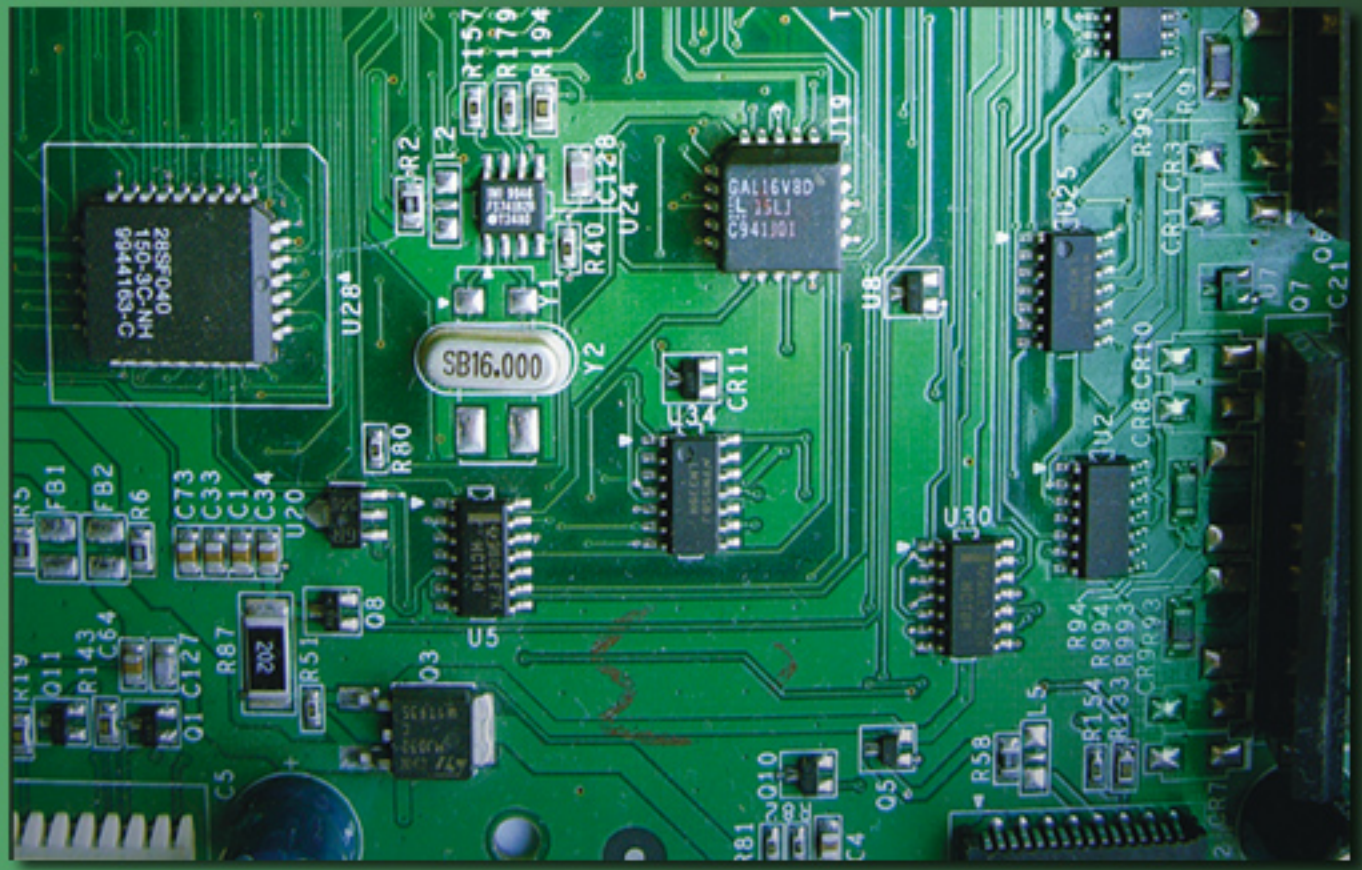
The Intel Microprocessors

8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions

Architecture, Programming, and Interfacing



EIGHTH EDITION

Barry B. Brey

PEARSON

Chapter 11: Basic I/O Interface

Introduction

- This chapter outlines some of the basic methods of communications, both serial and parallel, between humans or machines and the microprocessor.
- We first introduce the basic I/O interface and discuss decoding for I/O devices.
- Then, we provide detail on parallel and serial interfacing, both of which have a variety of applications.

Objectives

Upon completion of this lecture, you will be able to:

- Explain the operation of the basic input and output interfaces.
- Decode an 8-, 16-, and 32-bit I/O device so that they can be used at any I/O port address.
- Define handshaking and explain how to use it with I/O devices.

11–1 INTRO TO I/O INTERFACE

- I/O instructions (IN, INS, OUT, and OUTS) are explained.
- Also isolated (direct or I/O mapped I/O) and memory-mapped I/O, the basic input and output interfaces, and handshaking.
- Knowledge of these topics makes it easier to understand the connection and operation of the programmable interface components and I/O techniques.

The I/O Instructions

- One type of instruction transfers information to an I/O device (OUT).
- Another reads from an I/O device (IN).
- Instructions are also provided to transfer strings of data between memory and I/O.
  - INS and OUTS, found except the 8086/8088

- Instructions that transfer data between an I/O device and the microprocessor's accumulator (AL, AX, or EAX) are called **IN** and **OUT**.
- The I/O address is stored in register DX as a 16-bit address or in the byte (p8) immediately following the opcode as an 8-bit address.
  - Intel calls the 8-bit form (p8) a **fixed address** because it is stored with the instruction, usually in a ROM
- The 16-bit address is called a **variable address** because it is stored in a DX, and then used to address the I/O device.

Other instructions that use DX to address I/O are the INS and OUTS instructions.

- I/O ports are 8 bits in width.
  - a 16-bit port is actually two consecutive 8-bit ports being addressed
  - a 32-bit I/O port is actually four 8-bit ports

- When data are transferred using IN or OUT, the I/O address, (**port number** or simply port), appears on the address bus.
- External I/O interface decodes the port number in the same manner as a memory address.
  - the 8-bit fixed port number (p8) appears on address bus connections A_7–A_0 with bits A_{15}–A_8 equal to 00000000_2
  - connections above A_{15} are undefined for I/O instruction

- The 16-bit variable port number (DX) appears on address connections A_{15}–A_0.
- The first 256 I/O port addresses (00H–FFH) are accessed by both fixed and variable I/O instructions.
  - any I/O address from 0100H to FFFFH is only accessed by the variable I/O address
- In a PC computer, all 16 address bus bits are decoded with locations 0000H–03FFH.
  - used for I/O inside the PC on the ISA (**industry standard architecture**) bus

- INS and OUTS instructions address an I/O device using the DX register.
  - but do not transfer data between accumulator and I/O device as do the IN/OUT instructions
  - Instead, they transfer data between memory and the I/O device
- Pentium 4 and Core2 operating in the 64-bit mode have the same I/O instructions.
- There are no 64-bit I/O instructions in the 64-bit mode.
  - most I/O is still 8 bits and likely will remain so

TABLE 11-1: Input/Output instructions.

Instruction	Data Width	Function
IN AL, p8	8	A byte is input into AL from port p8
IN AX, p8	16	A word is input into AX from port p8
IN EAX, p8	32	A doubleword is input into EAX from port p8
IN AL, DX	8	A byte is input into AL from the port addressed by DX
IN AX, DX	16	A word is input into AX from the port addressed by DX
on addressed by DI, then DI = DI ± 2		memory locati
is input from the port addressed by DI and stored into the extra segment on addressed by DI, then DI = DI ± 4	INSD	32	A doubleword memory locati
ut from AL into port p8	OUT p8, AL	8	A byte is outp
ut from AL into port p8	OUT p8, AX	16	A word is outp
is output from EAX into port p8	OUT p8, EAX	32	A doubleword
ut from AL into the port addressed by DX	OUT DX, AL	8	A byte is outp
ut from AX into the port addressed by DX	OUT DX, AX	16	A word is outp
is output from EAX into the port addressed by DX	OUT DX, EAX	32	A doubleword
ut from the data segment memory location addressed by SI into the port DX, then SI = SI ± 1	OUTSB	8	A byte is outp addressed by
ut from the data segment memory location addressed by SI into the d by DX, then SI = SI ± 2	OUTSW	16	A word is outp port addresse
is output from the data segment memory location addressed by SI into ssed by DX, then SI = SI ± 4	OUTSD	32	A doubleword the port addre

Isolated and Memory-Mapped I/O

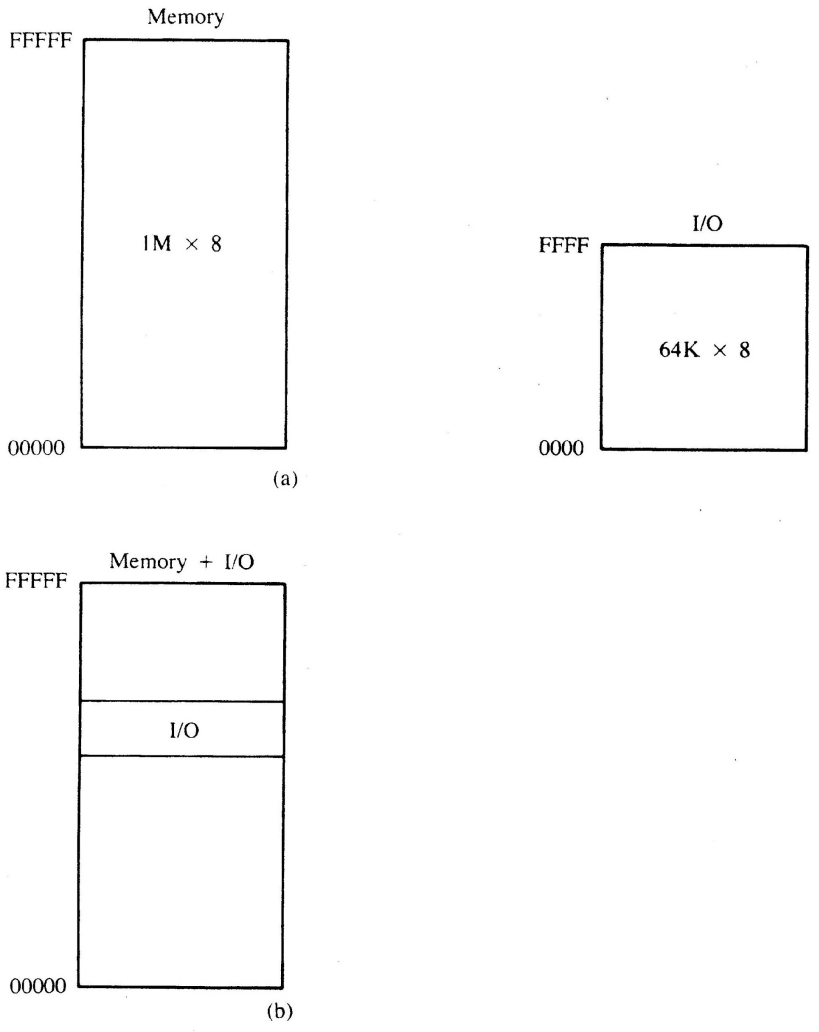
- Two different methods of interfacing I/O: **isolated I/O** and **memory-mapped I/O**.
- In isolated I/O, the IN, INS, OUT, and OUTS transfer data between the microprocessor's accumulator or memory and the I/O device.
- In memory-mapped I/O, any instruction that references memory can accomplish the transfer.
- The PC does not use memory-mapped I/O.

Isolated I/O

- The most common I/O transfer technique used in the Intel-based system is isolated I/O.
  - *isolated* describes how I/O locations are isolated from memory in a separate I/O address space
- Addresses for isolated I/O devices, called ports, are separate from memory.
- Because the ports are separate, the user can expand the memory to its full size without using any of memory space for I/O devices.

- A disadvantage of isolated I/O is that data transferred between I/O and microprocessor must be accessed by the IN, INS, OUT, and OUTS instructions.
- Separate control signals for the I/O space are developed (using M/$\overline{IO}$ and W/$\overline{R}$), which indicate an I/O read ($\overline{IORC}$) or an I/O write ($\overline{RD}$) operation.
- These signals indicate an I/O port address, which appears on the address bus, is used to select the I/O device.

Figure 11–1 The memory and I/O maps for the 8086/8088 microprocessors. (a) Isolated I/O. (b) Memory-mapped I/O.

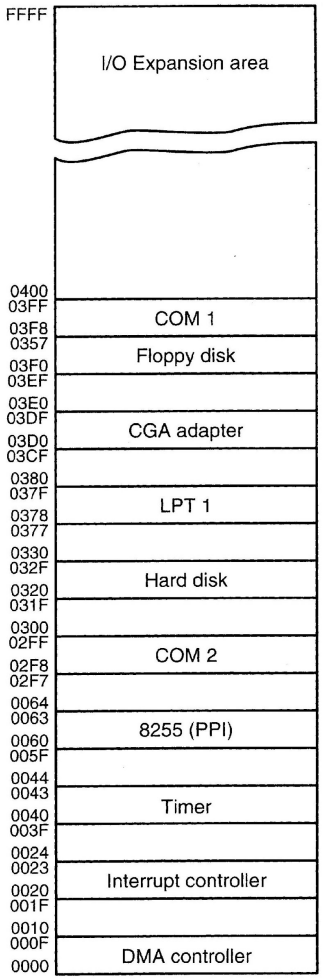


- in the PC, isolated I/O ports are used to control peripheral devices
- an 8-bit port address is used to access devices located on the system board, such as the timer and keyboard interface
- a 16-bit port is used to access serial and parallel ports, video and disk drive systems

Memory-Mapped I/O

- Memory-mapped I/O does not use the IN, INS, OUT, or OUTS instructions.
- It uses any instruction that transfers data between the microprocessor and memory.
  - treated as a memory location in memory map
- Advantage is any memory transfer instruction can access the I/O device.
- Disadvantage is a portion of memory system is used as the I/O map.
  - reduces memory available to applications

Personal Computer I/O Map



- the PC uses part of I/O map for dedicated functions, as shown here
- I/O space between ports 0000H and 03FFH is normally reserved for the system and ISA bus
- ports at 0400H–FFFFH are generally available for user applications, main-board functions, and the PCI bus
- 80287 coprocessor uses 00F8H–00FFH, so Intel reserves I/O ports 00F0H–00FFH

Figure 11–2 I/O map of a personal computer illustrating many of the fixed I/O areas.

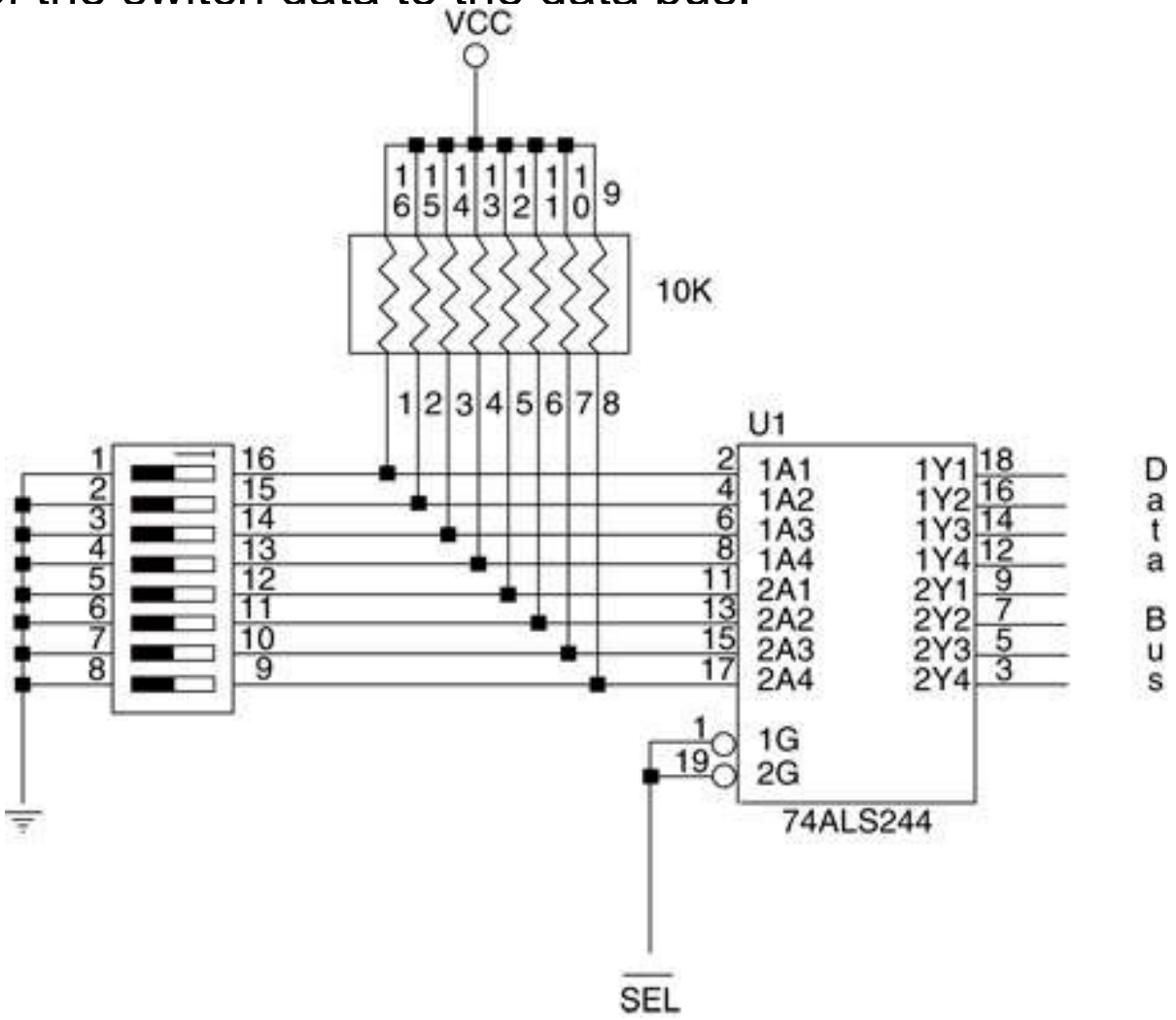
Basic Input and Output Interfaces

- The basic input device is a set of three-state buffers.
- The basic output device is a set of data latches.
- The term IN refers to moving data *from* the I/O device *into* the microprocessor and
- The term OUT refers to moving data *out* of the microprocessor *to* the I/O device.

The Basic Input Interface

- Three-state buffers are used to construct the 8-bit input port depicted in Figure 11–3.
- External TTL data are connected to the inputs of the buffers.
  - buffer outputs connect to the data bus
- The circuit of allows the processor to read the contents of the eight switches that connect to any 8-bit section of the data bus when the select signal becomes a logic 0.

Figure 11–3 The basic input interface illustrating the connection of eight switches. Note that the 74ALS244 is a three-state buffer that controls the application of the switch data to the data bus.

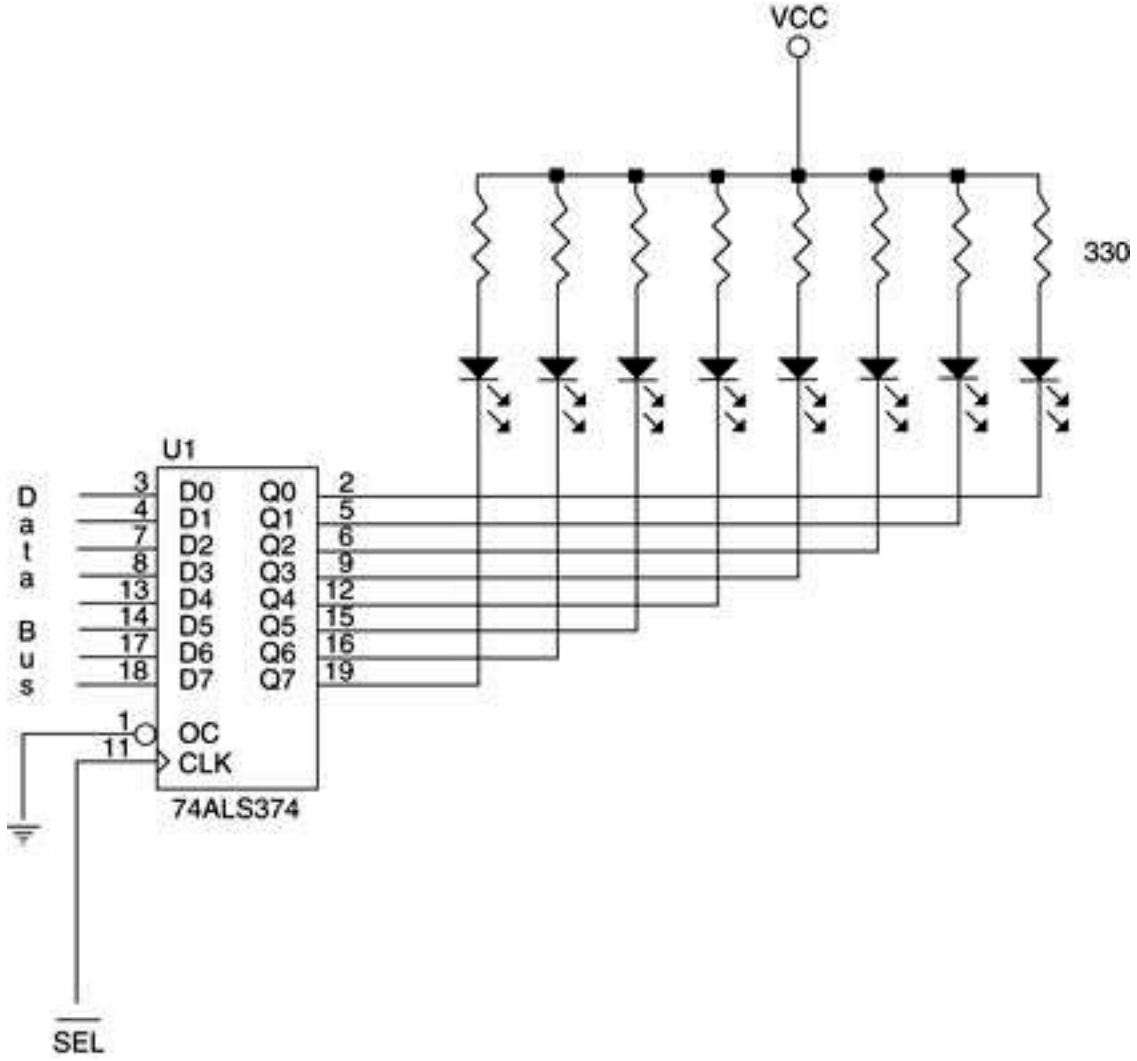


- When the IN instruction executes, contents of the switches copy to the AL register.
- This basic input circuit is not optional and must appear any time input data are interfaced to the microprocessor.
- Sometimes it appears as a discrete part of the circuit, as shown in Figure 11–3.
  - also built into a programmable I/O devices
- Sixteen- or 32-bit data can also be interfaced but is not nearly as common as 8-bit data.

The Basic Output Interface

- Receives data from the processor and usually must hold it for some external device.
  - latches or flip-flops, like buffers in the input device, are often built into the I/O device
- Fig 11–4 shows how eight light-emitting diodes (LEDs) connect to the processor through a set of eight data latches.
- The latch stores the number output by the microprocessor from the data bus so that the LEDs can be lit with any 8-bit binary number.

Figure 11–4 The basic output interface connected to a set of LED displays.

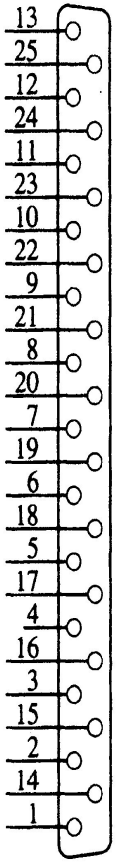


- Latches hold the data because when the processor executes an OUT, data are only present on the data bus for less than 1.0 μs.
  - the viewer would never see the LEDs illuminate
- When the OUT executes, data from AL, AX, or EAX transfer to the latch via the data bus.
- Each time the OUT executes, the $\overline{\text{SEL}}$ signal activates, capturing data to the latch.
  - data are held until the next OUT
- When the output instruction is executed, data from the AL register appear on the LEDs.

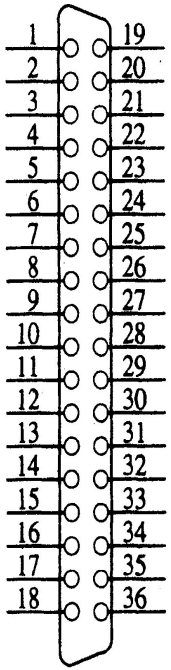
Handshaking

- Many I/O devices accept or release information slower than the microprocessor.
- A method of I/O control called **handshaking** or **polling**, synchronizes the I/O device with the microprocessor.
- An example is a parallel printer that prints a few hundred characters per second (CPS).
- The processor can send data much faster.
  - a way to slow the microprocessor down to match speeds with the printer must be developed

- Fig 11–5 illustrates typical input and output connections found on a printer.
  - data transfers via data connections (D_7–D_0)
- ASCII data are placed on D_7–D_0, and a pulse is then applied to the $\overline{STB}$ connection.
  - BUSY indicates the printer is busy
  - $\overline{STB}$ is a clock pulse used to send data to printer
- The strobe signal sends or clocks the data into the printer so that they can be printed.
  - as the printer receives data, it places logic 1 on the BUSY pin, indicating it is printing data



Connector DB25



Connector CENT36

DB25 Pin number	CENT36 Pin number	Function	DB25 Pin number	CENT36 Pin number	Function
1	1	$\overline{\text{Data Strobe}}$	12	12	Paper empty
2	2	Data 0 (D0)	13	13	Select
3	3	Data 1 (D1)	14	14	Afd
4	4	Data 2 (D2)	15	32	$\overline{\text{Error}}$
5	5	Data 3 (D3)	16	—	$\overline{\text{RESET}}$
6	6	Data 4 (D4)	17	31	Select in
7	7	Data 5 (D5)	18—25	19—30	Ground
8	8	Data 6 (D6)	—	17	Frame ground
9	9	Data 7 (D7)	—	16	Ground
10	10	$\overline{\text{Ack}}$	—	33	Ground
11	11	Busy			

FIGURE 11-5 The DB25 connector found on computers and the Centronics 36-pin connector found on printers for the Centronics parallel printer interface.

- The software polls or tests the BUSY pin to decide whether the printer is busy.
  - If the printer is busy, the processor waits
  - if not, the next ASCII character goes to the printer
- This process of interrogating the printer, or any asynchronous device like a printer, is called handshaking or polling.

- **EXAMPLE 11–1**
- An assembly language procedure that prints the ASCII contents of BL.

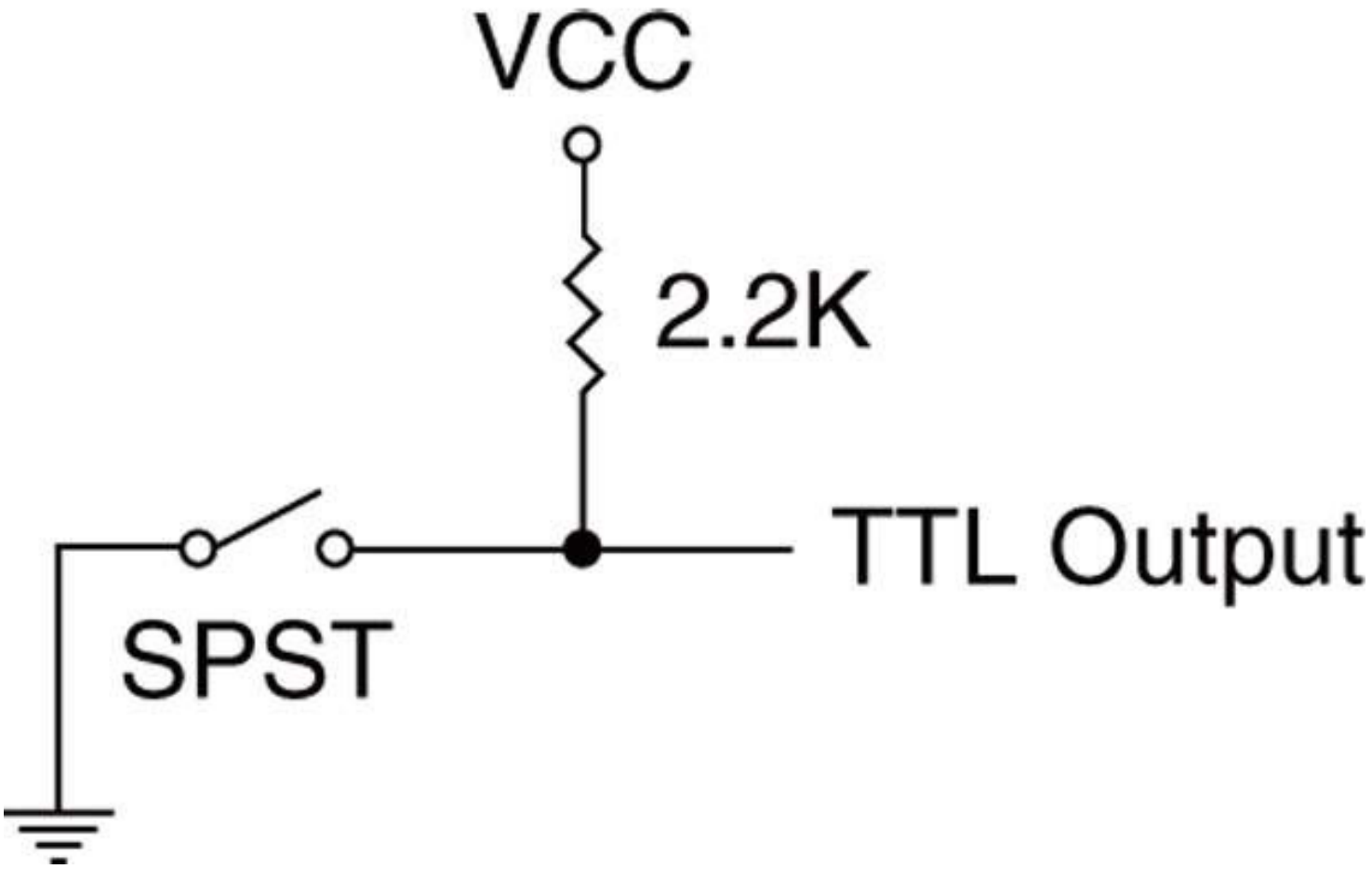
```
PRINT PROC NEAR
    .REPEAT ;test the busy flag
        IN AL, BUSY
        TEST AL, BUSY_BIT
    .UNTIL ZERO
    MOV AL, BL ;position data in AL
    OUT PRINTER, AL ;print data
    RET
PRINT ENDP
```

Input Devices

- Input devices are already TTL and compatible, and can be connected to the microprocessor and its interfacing components.
  - or they are switch-based
- Switch-based devices are either open or connected; These are not TTL levels.
  - TTL levels are a logic 0 (0.0 V–0.8 V)
  - or a logic 1 (2.0 V–5.0 V)
- Using switch-based device as TTL-compatible input requires conditioning applied.

- Fig 11–6 shows a toggle switch properly connected to function as an input device.
- A pull-up resistor ensures when the switch is open, the output signal is a logic 1.
  - when the switch is closed, it connects to ground, producing a valid logic 0 level
- A standard range of values for pull-up resistors is between 1K Ohm and 10K Ohm.

Figure 11–6 A single-pole, single-throw switch interfaced as a TTL device.

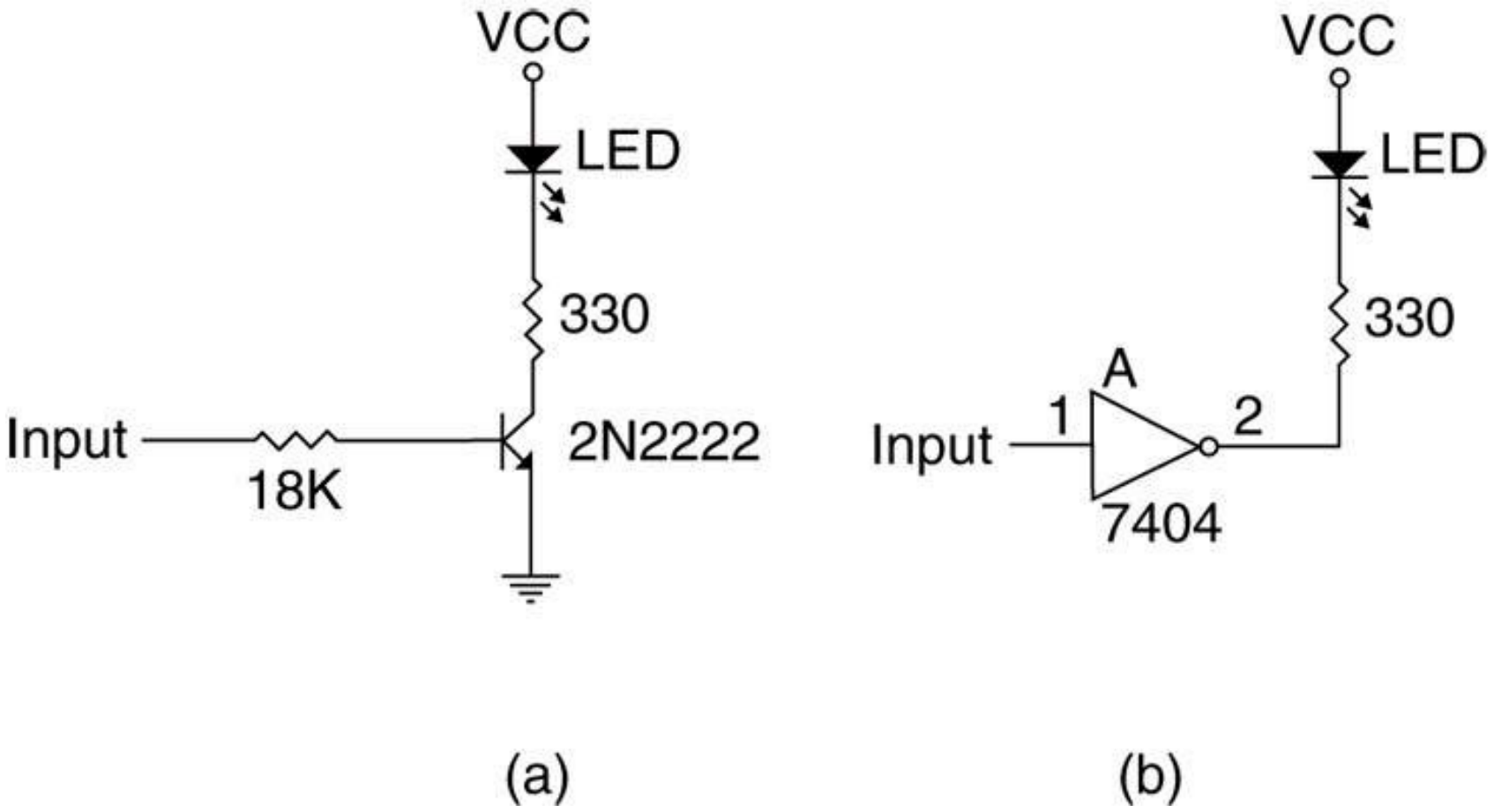


Output Devices

- Output devices are more diverse than input devices, but many are interfaced in a uniform manner.
- Before an output device can be interfaced, we must understand voltages and currents from the microprocessor or TTL interface.
- Voltages are TTL-compatible from the microprocessor of the interfacing element.
  - logic 0 = 0.0 V to 0.4 V
  - logic 1 = 2.4 V to 5.0 V

- Currents for a processor and many interfacing components are less than for standard TTL.
  - Logic 0 = 0.0 to 2.0 mA
  - logic 1 = 0.0 to 400 µA
- Fig 11–8 shows how to interface a simple LED to a microprocessor peripheral pin.
  - a transistor driver is used in 11–8(a)
  - a TTL inverter is used in 11–8(b)
- The TTL inverter (standard version) provides up to 16 mA of current at a logic 0 level
  - more than enough to drive a standard LED

Figure 11–8 Interfacing an LED: (a) using a transistor and (b) using an inverter.



- TTL input signal has minimum value of 2.4 V
- Drop across emitter-base junction is 0.7 V.
- The difference is 1.7 V
  - the voltage drop across the resistor
- The value of the resistor is 1.7 V ÷ 0.1 mA or 17K Ω.
  - as 17K Ω is not a standard value, an 18K Ω resistor is chosen

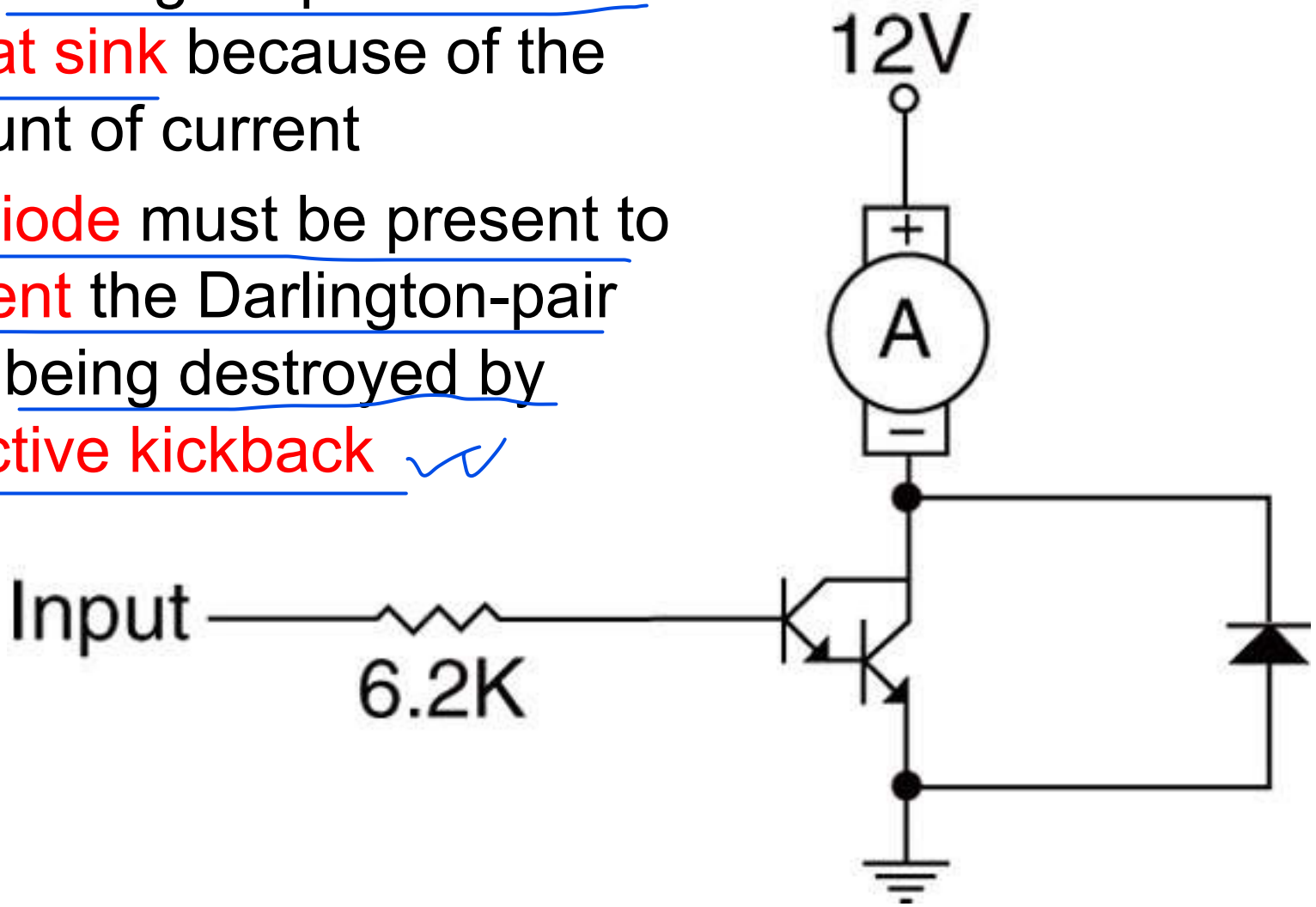
- In 11–8(a), we elected to use a switching transistor in place of the TTL buffer.
  - 2N2222 is a good low-cost, general-purpose switching transistor with a minimum gain of 100
  - collector current is 10 mA; so base current will be 1/100 of collector current of 0.1 mA
- To determine the value of the base current–limiting resistor, use the 0.1 mA base current and a voltage drop of 1.7 V across the base current–limiting resistor.

- Suppose we need to interface a 12 V DC 1A motor to the microprocessor.
- We cannot use a TTL inverter:
  - 12 V signal would burn out the inverter
  - current far exceeds 16 mA inverter maximum
- We cannot use a 2N2222 transistor:
  - maximum current is 250 mA to 500 mA, depending on the package style chosen
- The solution is to use a Darlington-pair, such as a TIP120.
  - costs 25¢, can handle 4A current with heat sink

- Fig 11–9 illustrates a motor connected to the Darlington-pair with a minimum current gain of 7000 and a maximum current of 4A.
- Value of the bias resistor is calculated exactly the same as the one used in the LED driver.
- The current through the resistor is 1.0 A ÷ 7000, or about 0.143 mA.
- Voltage drop is 0.9 V because of the two diode drops (base/emitter junctions).
- The value of the bias resistor is 0.9 V ÷ 0.143 mA or 6.29K Ω.

Figure 11–9 A DC motor interfaced to a system by using a Darlington-pair.

- The Darlington-pair must use a heat sink because of the amount of current
- the diode must be present to prevent the Darlington-pair from being destroyed by inductive kickback



11–2 I/O PORT ADDRESS DECODING

- Very similar to memory address decoding, especially for memory-mapped I/O devices.
- The difference between memory decoding and isolated I/O decoding is the number of address pins connected to the decoder.
- In the personal computer system, we always decode all 16 bits of the I/O port address.

Decoding 8-Bit I/O Port Addresses

- Fixed I/O instruction uses an 8-bit I/O port address that on A_{15}–A_0 as 0000H–00FFH.
  - we often decode only address connections A_7–A_0 for an 8-bit I/O port address
- The DX register can also address I/O ports 00H–FFH.
- If the address is decoded as an 8-bit address, we can never include I/O devices using a 16-bit address.
  - the PC never uses or decodes an 8-bit address

- Figure 11–10 shows a 74ALS138 decoder that decodes 8-bit I/O ports F0H - F7H.
  - identical to a memory address decoder except we only connect address bits A_7–A_0 to the inputs of the decoder
- Figure 11–11 shows the PLD version, using a GAL22V10 (a low-cost device) for this decoder.
- The PLD is a better decoder circuit because the number of integrated circuits has been reduced to one device.

Figure 11–10 A port decoder that decodes 8-bit I/O ports. This decoder generates active low outputs for ports F0H–F7H.

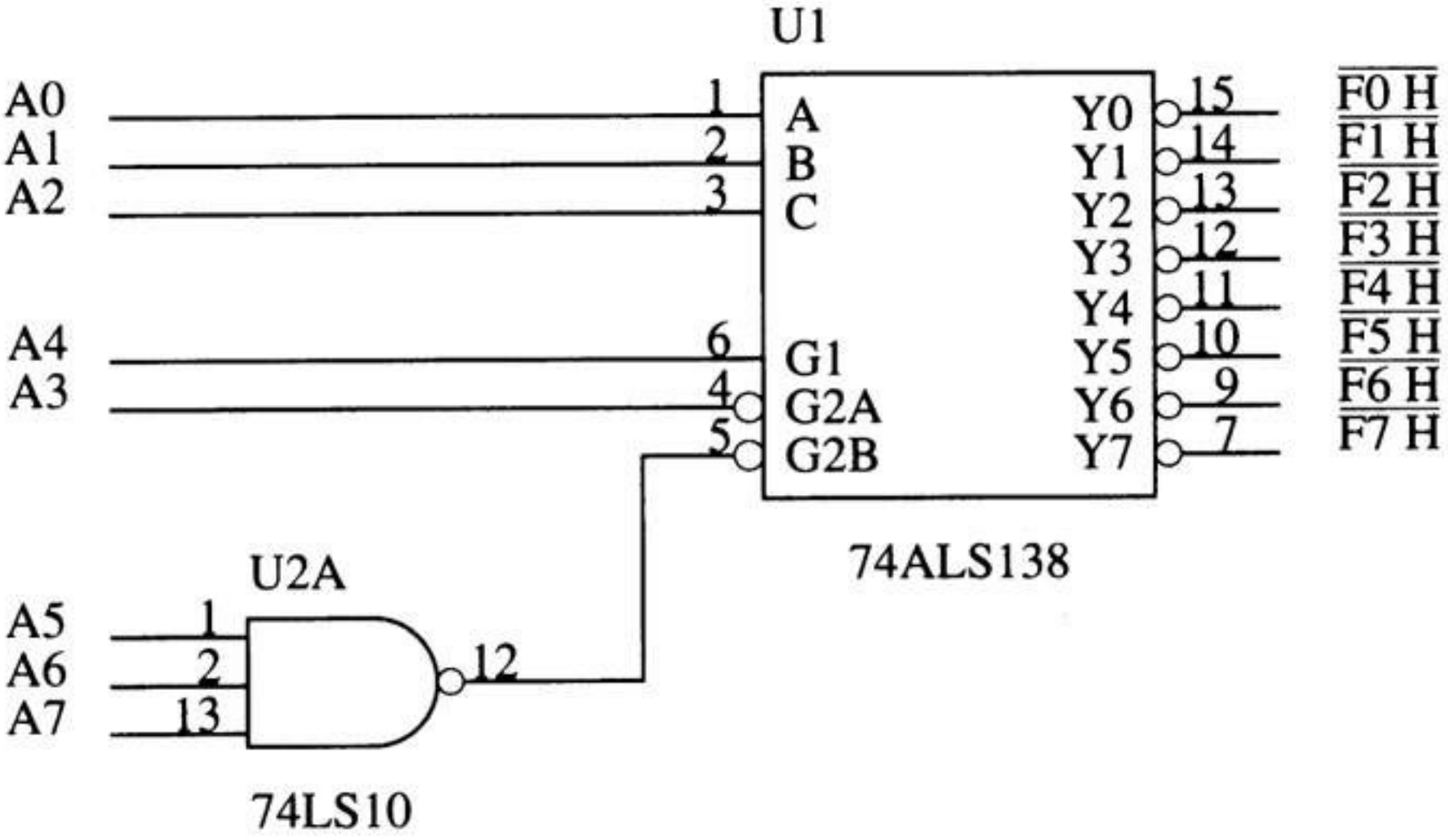
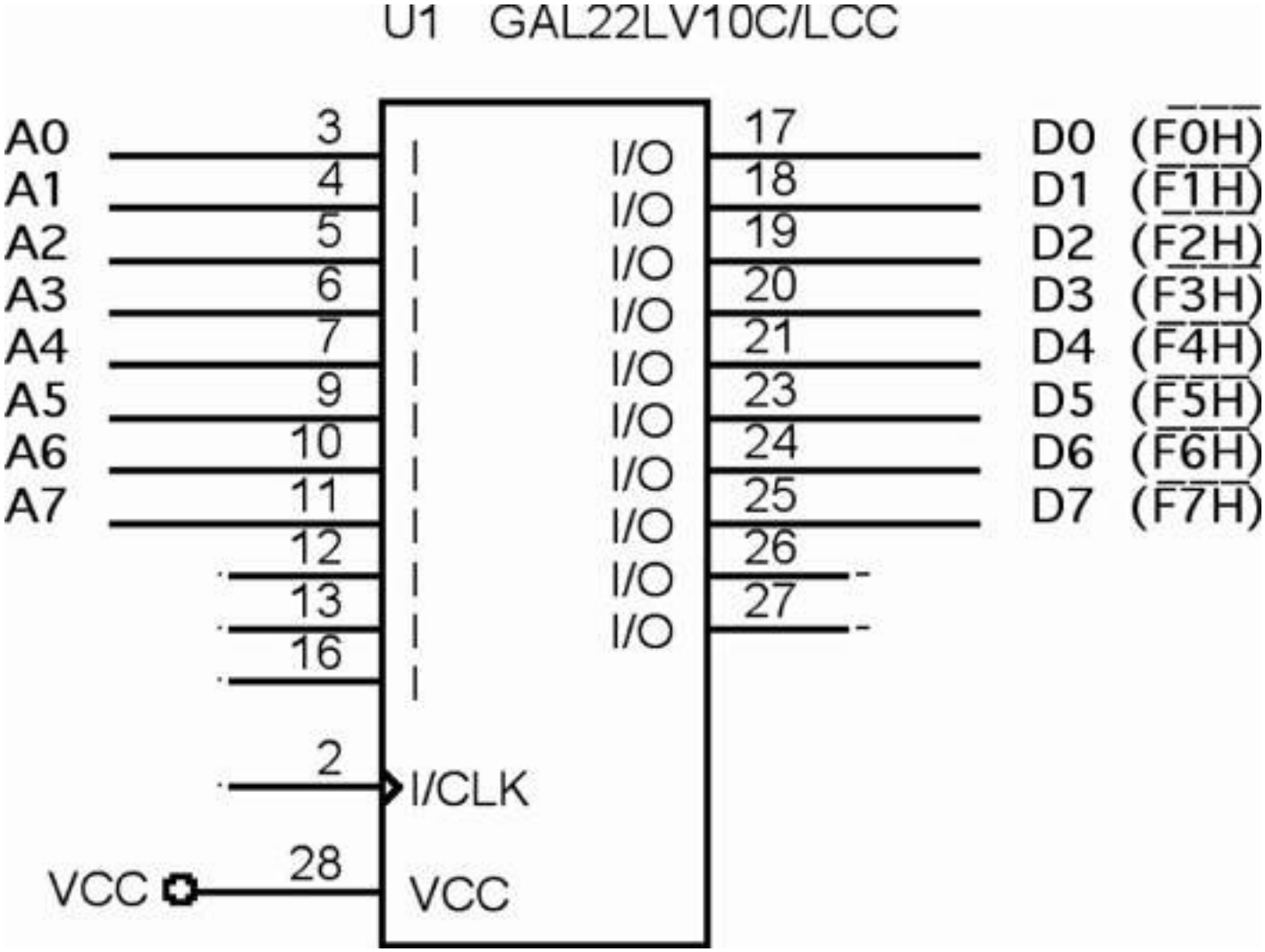


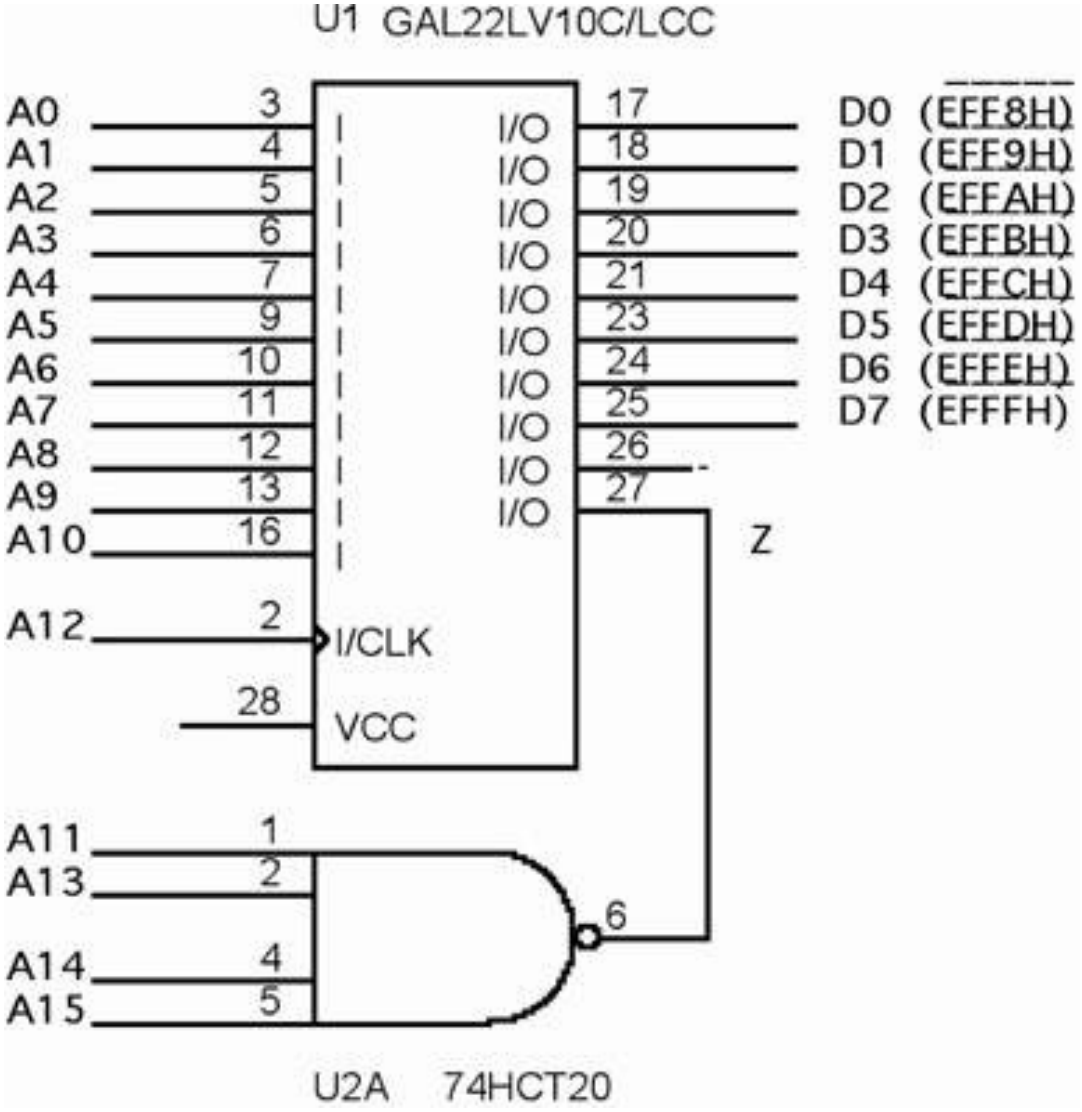
Figure 11–11 A PLD that generates part selection signals



Decoding 16-Bit I/O Port Addresses

- PC systems typically use 16-bit I/O addresses.
  - 16-bit addresses rare in embedded systems
- The difference between decoding an 8-bit and a 16-bit I/O address is that eight additional address lines (A_{15}–A_8) must be decoded.
- Figure 11–12 illustrates a circuit that contains a PLD and a 4-input NAND gate used to decode I/O ports EFF8H–EFFFH.
- PLD generates address strobes for I/O ports

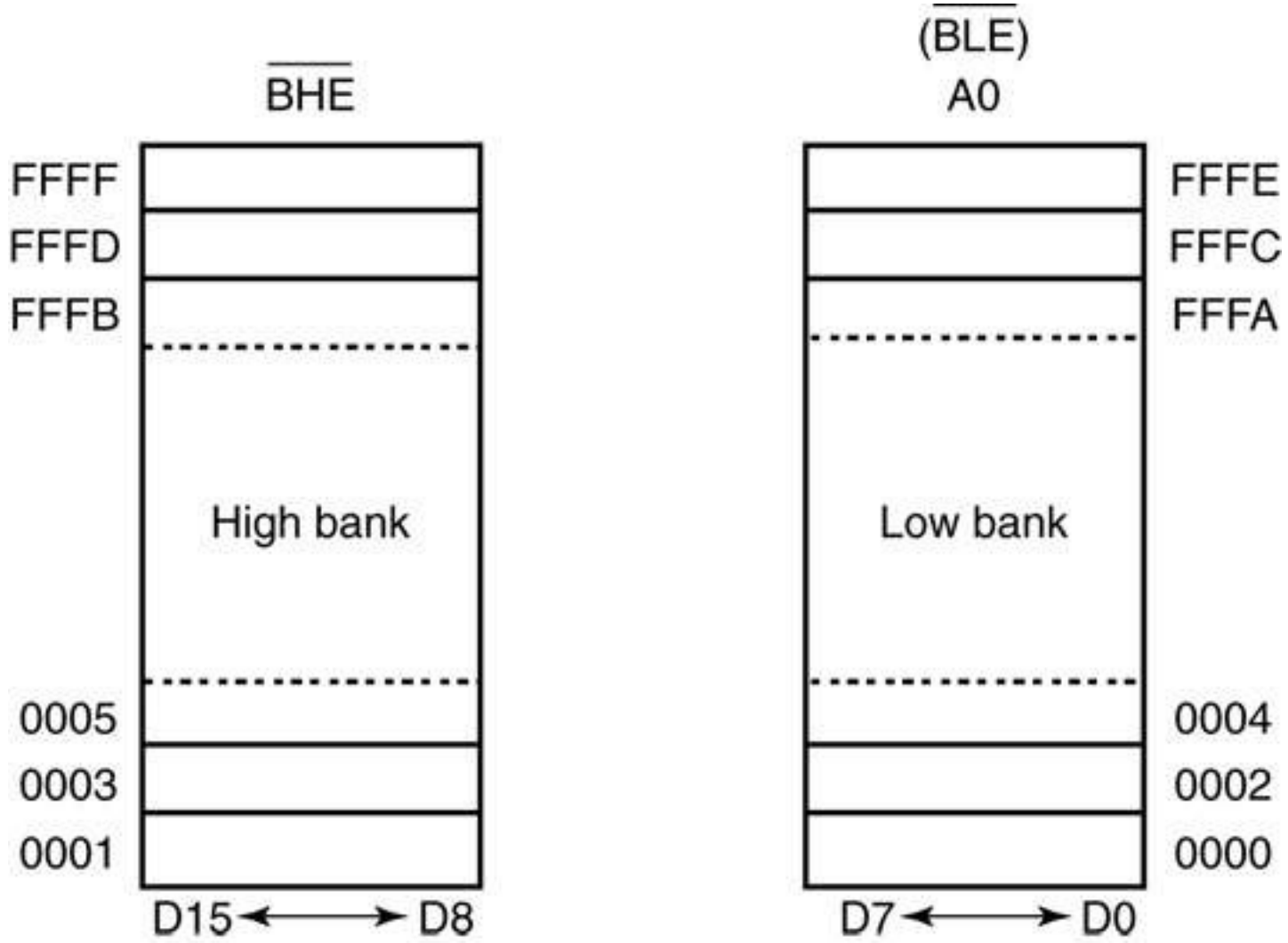
Figure 11–12 A PLD that decodes 16-bit I/O ports EFF8H through EFFFH.



8- and 16-Bit Wide I/O Ports

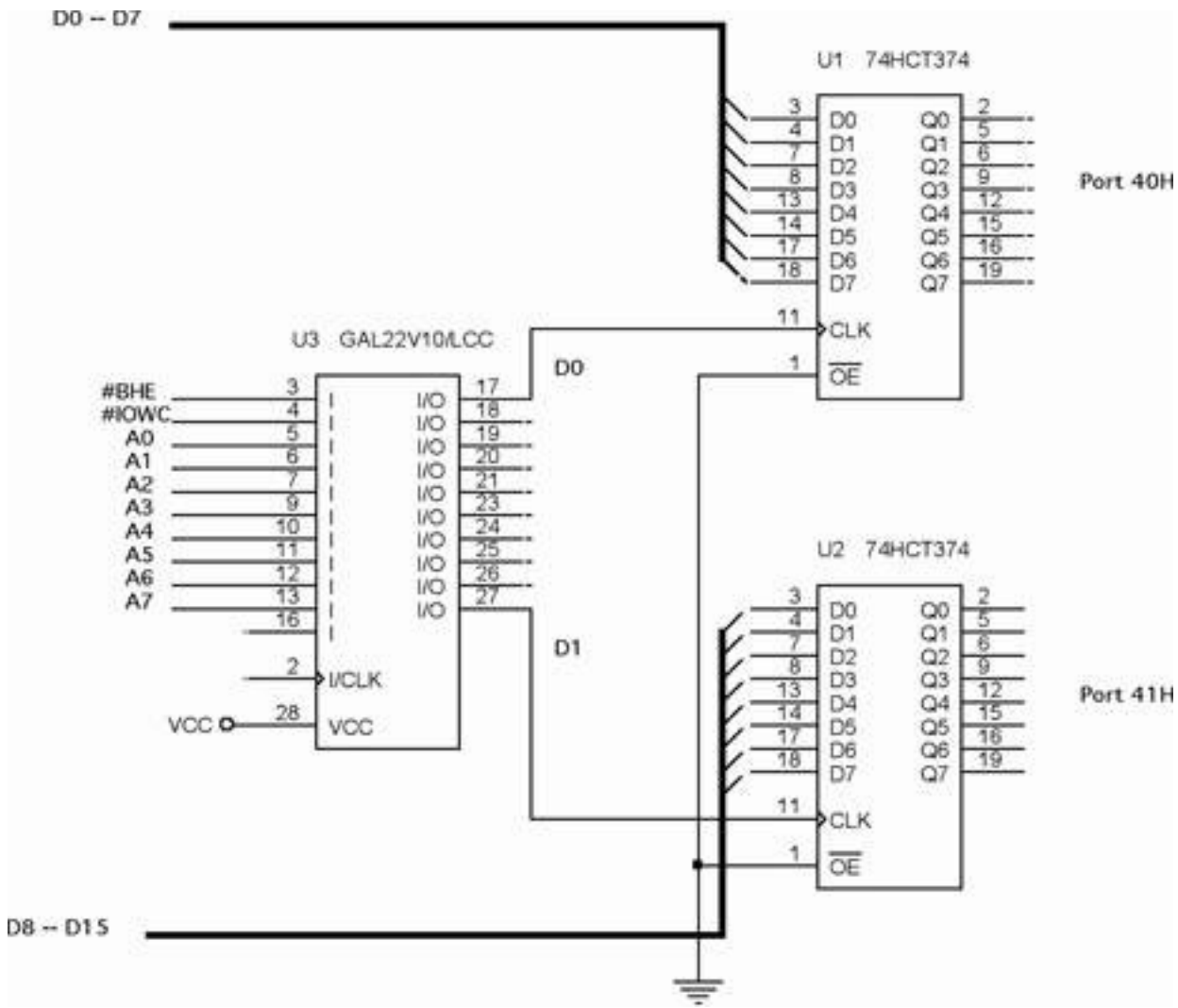
- Data transferred to an 8-bit I/O device exist in one of the I/O banks in a 16-bit processor such as 80386SX.
- The I/O system on such a microprocessor contains two 8-bit memory banks.
- Fig 11–13 shows separate I/O banks for a 16-bit system such as 80386SX.
- Because two I/O banks exist, any 8-bit I/O write requires a separate write.

Figure 11–13 The I/O banks found in the 8086, 80186, 80286, and 80386SX.



- I/O reads don't require separate strobes.
  - as with memory, the processor reads only the byte it expects and ignores the other byte
  - a read can cause problems when an I/O device responds incorrectly to a read operation
- Fig 11–14 shows a system with two different 8-bit output devices, located at 40H and 41H.
- These are 8-bit devices and appear in different I/O banks.
  - thus, separate I/O write signals are generated to clock a pair of latches that capture port data

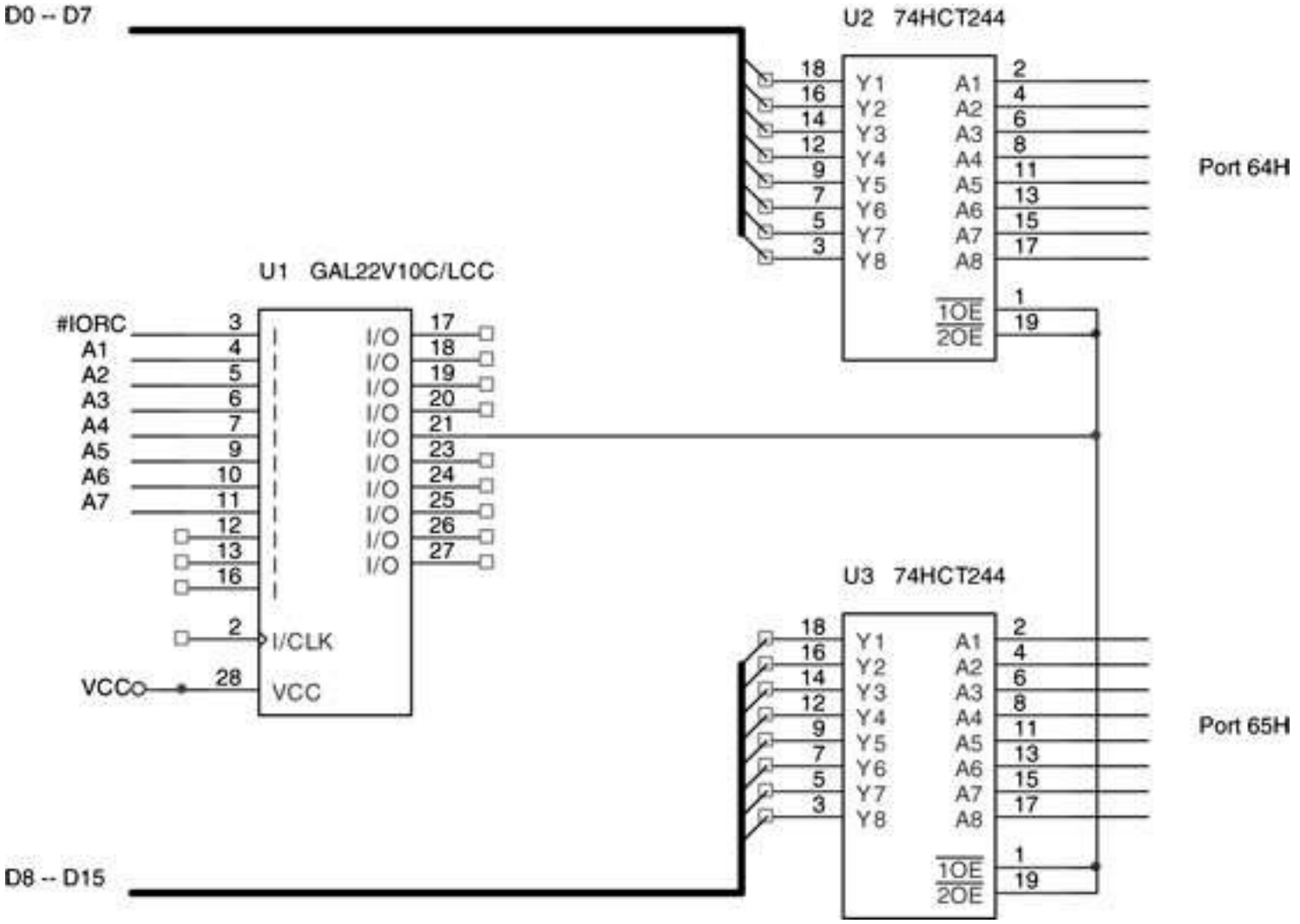
Figure 11–14 An I/O port decoder that selects ports 40H and 41H for output data.



- all I/O ports use 8-bit addresses
- ports 40H & 41H can be addressed as separate 8-bit ports
- or as one 16-bit port

- Fig 11–15 shows a 16-bit device connected to function at 8-bit addresses 64H & 65H.
- The PLD decoder does not have a connection for address bits $\overline{\text{BLE}}$ (A_0) and $\overline{\text{BHE}}$ because the signals don't apply to 16-bit-wide devices.
- The program for the PLD, illustrated in Example 11–5, shows how the enable signals are generated for the three-state buffers (74HCT244) used as input devices.

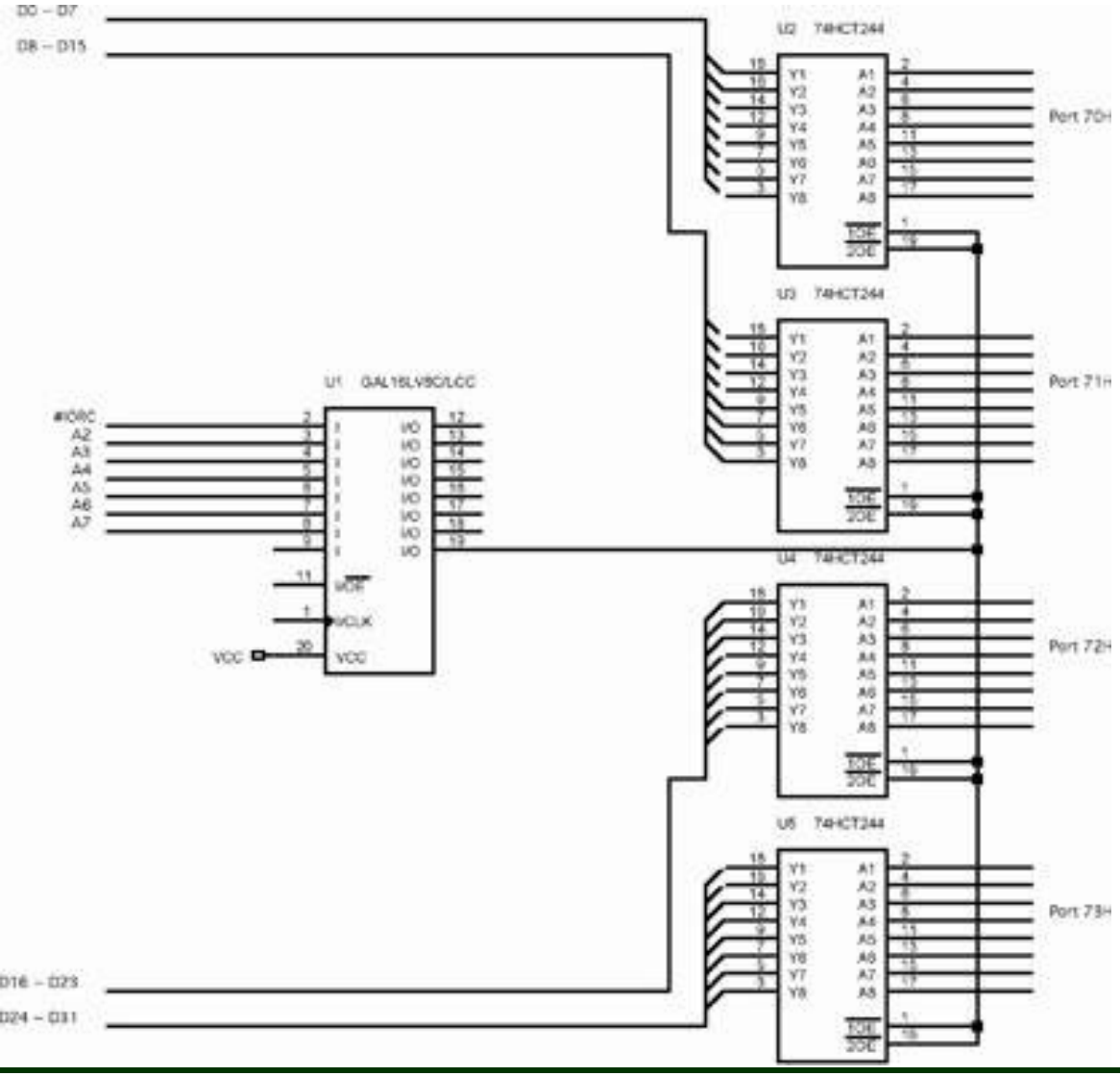
Figure 11–15 A 16-bit-wide port decoded at I/O addresses 64H and 65H.



32-Bit-Wide I/O Ports

- May eventually become common because of newer buses found in computer systems.
- The EISA system bus supports 32-bit I/O as well as the VESA local and current PCI bus.
  - not many I/O devices are 32 bits in width
- Fig 11–16 shows a 32-bit input port for 80386DX - 80486DX microprocessor.
- The circuit uses a single PLD to decode the I/O ports and four 74HCT244 buffers to connect the I/O data to the data bus.

Figure 11–16 A 32-bit-wide port decoded at 70H through 73H for the 80486DX microprocessor.

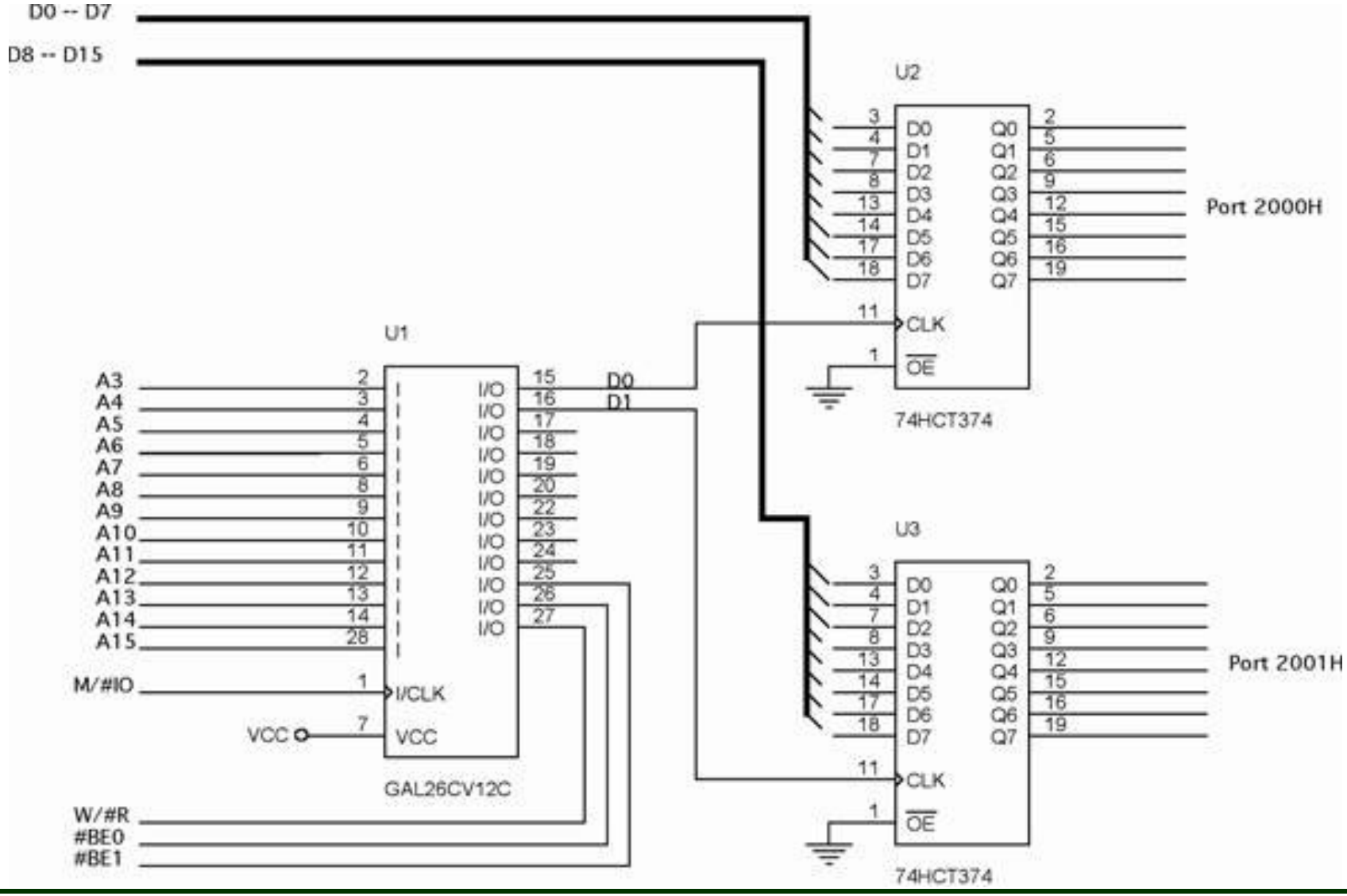


- I/O ports decoded by this interface are the 8-bit ports 70H–73H
- When writing to access this port, it is crucial to use the address 70H for 32-bit input
- as instruction IN EAX, 70H

- With the Pentium–Core2 and their 64-bit data buses, I/O ports appear in various banks, as determined by the I/O port address.
- A 32-bit I/O access in the Pentium system can appear in any four consecutive I/O banks.
  - 32-bit ports 0100H–0103H appear in banks 0–3
- I/O address range must begin at a location where the rightmost two bits are zeros.
  - 0100H–0103H is allowable
  - 0101H–0104H is not

- Widest I/O transfers are 32 bits, and there are no I/O instructions to support 64-bit transfers.
  - true for Pentium 4 or Core2 in the 64-bit mode
- Suppose we need to interface a 16-bit-wide output port at I/O address 2000H and 2001H.
  - interface is illustrated in Figure 11–17
  - PLD program is listed in Example 11–7
- The problem that can arise is when the I/O port spans across a 64-bit boundary.
  - example, a 16-bit- port at 2007H and 2008H

Figure 11–17 A Pentium 4 interfaced to a 16-bit-wide I/O port at port addresses 2000H and 2001H.



SUMMARY

- The 8086-Core2 microprocessors have two basic types of I/O instructions: IN and OUT.
- The IN instruction inputs data from an external I/O device into either the AL (8-bit) or AX (16-bit) register.
- The IN instruction is available as a fixed port instruction, a variable port instruction, or a string instruction (80286-Pentium 4) INSB or INSW.

SUMMARY (*cont.*)

- The OUT instruction outputs data from AL or AX to an external I/O device and is available as a fixed, variable, or string instruction OUTSB or OUTSW.
- The fixed port instruction uses an 8-bit I/O port address, while the variable and string I/O instructions use a 16-bit port number found in the DX register.

SUMMARY (*cont.*)

- Isolated I/O, sometimes called direct I/O, uses a separate map for the I/O space, freeing the entire memory for use by the program.
- Isolated I/O uses the IN and OUT instructions to transfer data between the I/O device and the micro-processor.

SUMMARY (*cont.*)

- Memory-mapped I/O uses a portion of the memory space for I/O transfers.
- In addition, any instruction that addresses a memory location using any addressing mode can be used to transfer data between the microprocessor and the I/O device using memory-mapped I/O.

SUMMARY (*cont.*)

- All input devices are buffered so that the I/O data are connected only to the data bus during the execution of the IN instruction.
- The buffer is either built into a programmable peripheral or located separately.
- All output devices use a latch to capture output data during the execution of the OUT instruction.

SUMMARY (*cont.*)

- Handshaking or polling is the act of two independent devices synchronizing with a few control lines.
- This communication between the computer and the printer is a handshake or a poll.
- Interfaces are required for most switch-based input devices and for most output devices that are not TTL-compatible.

SUMMARY (*cont.*)

- The I/O port number appears on address bus connections A7-A0 for a fixed port I/O instruction and on A15-A0 for a variable port I/O instruction (note that A15-A8 contains zeros for an 8-bit port).
- In both cases, address bits above A15 are undefined.

SUMMARY (*cont.*)

- Because the 8086/80286/80386SX microprocessors contain a 16-bit data bus and the I/O addresses reference byte-sized I/O locations, I/O space is also organized in banks, as is the memory system.
- In order to interface an 8-bit I/O device to the 16-bit data bus, we often require separate write strobes (an upper and a lower) for I/O write operations.