# The Peripheral Interface Adapter (PIA)

# The Peripheral Interface Adapter (PIA)

- A peripheral interface adapter contains two or three parallel I/0 ports or locations that can be programmed to handle input or output data.
- Some of these adapters also contain either RAM or ROM memory.
- Still others contain a timer or programmable modulus counter that can generate a variety of different signals.
- The timer is also used to count external events in some applications.

# The Peripheral Interface Adapter (PIA)

- These devices allow almost any TTL-compatible input or output device to be interfaced to the microprocessor.
- They contain the basic input and output circuitry discussed previously in this chapter.
- In addition to the basic I/0 circuitry, PIAs contain a handshaking or synchronization mechanism that is typically used with asynchronous external I/0 devices to synchronize them with the microprocessor.
- Peripheral interface adapters are available for multiplexed data buses or standard nonmultiplexed data buses.
- The multiplexed bus version is normally set up to work with isolated I/0, and the nonmultiplexed version works with either isolated or memory-mapped I/0.

# A Typical Port Pin Connection

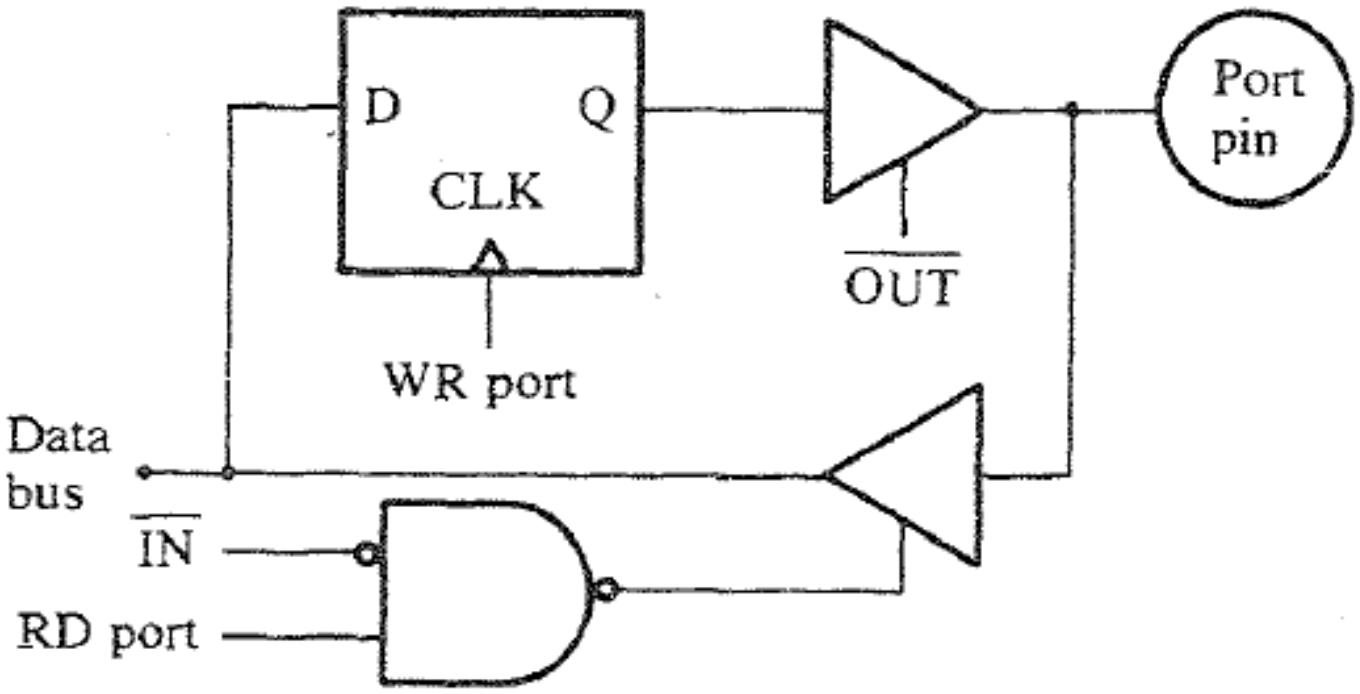
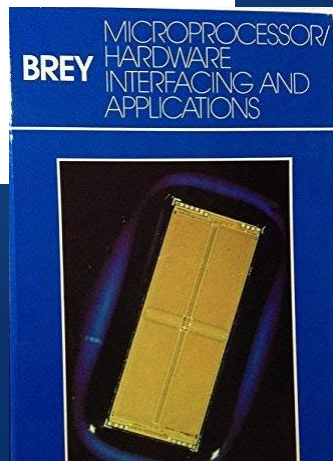
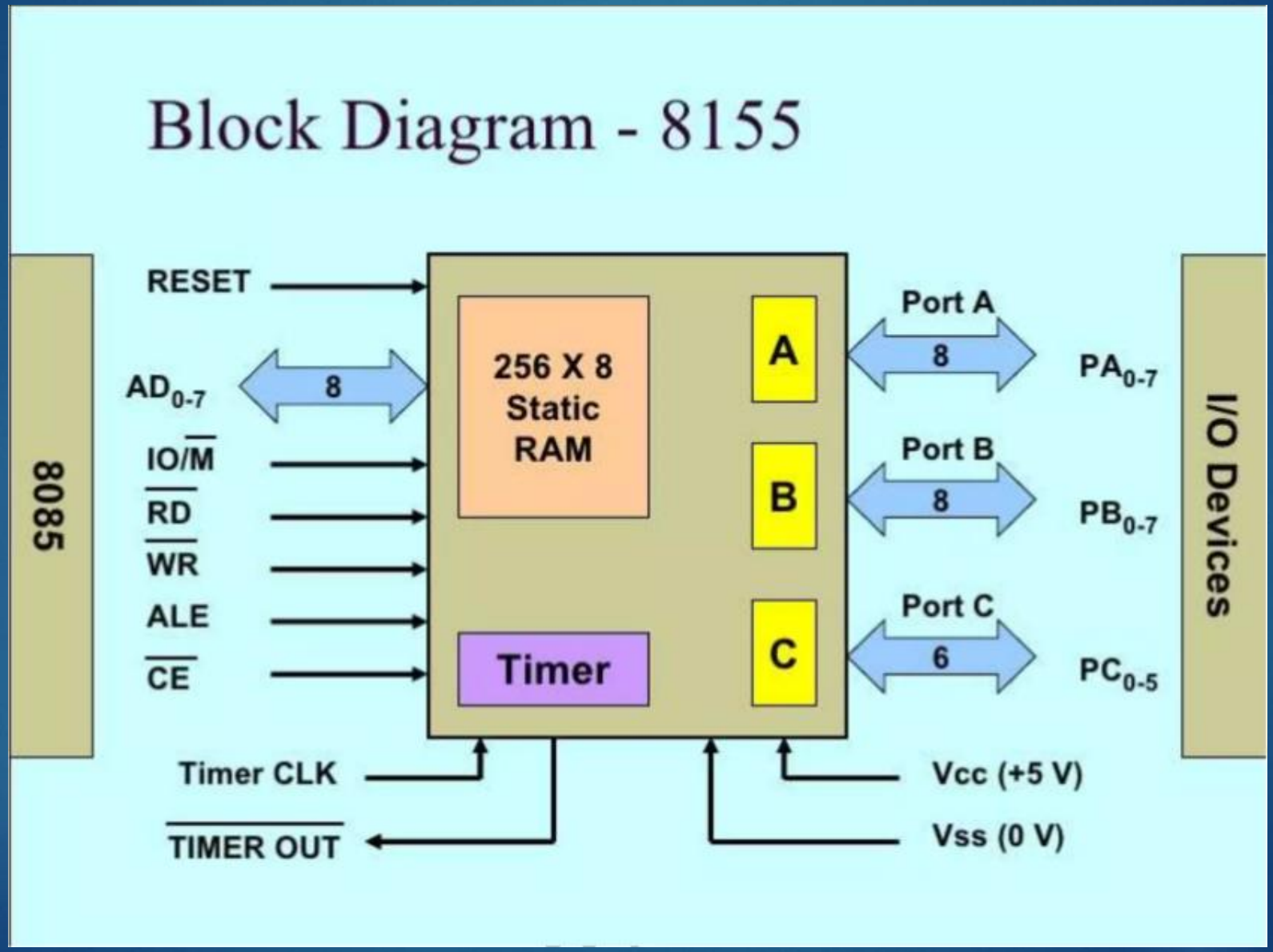



**FIGURE 6–18** A typical internal representation of a programmable bidirectional PIA port pin connection.

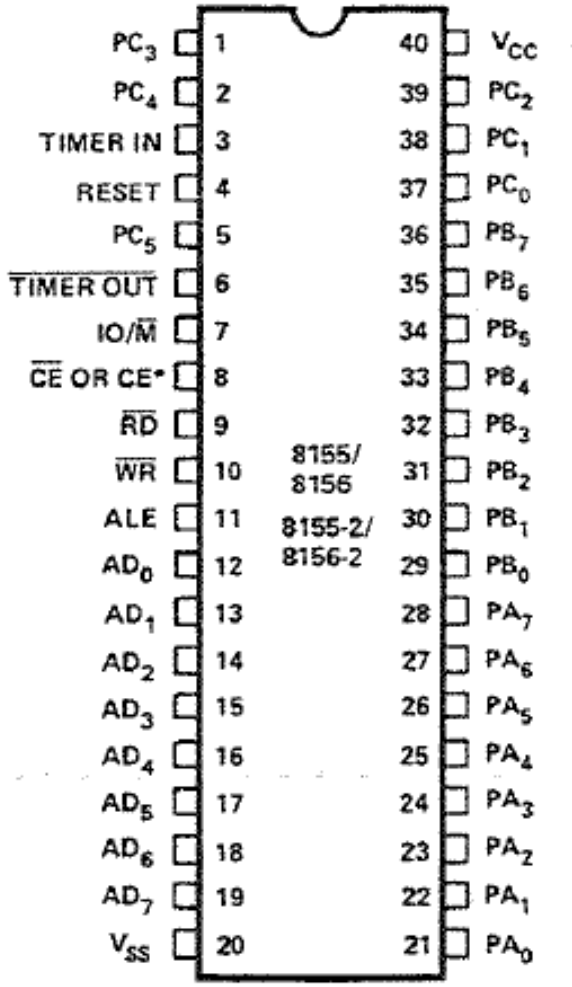


# 8155

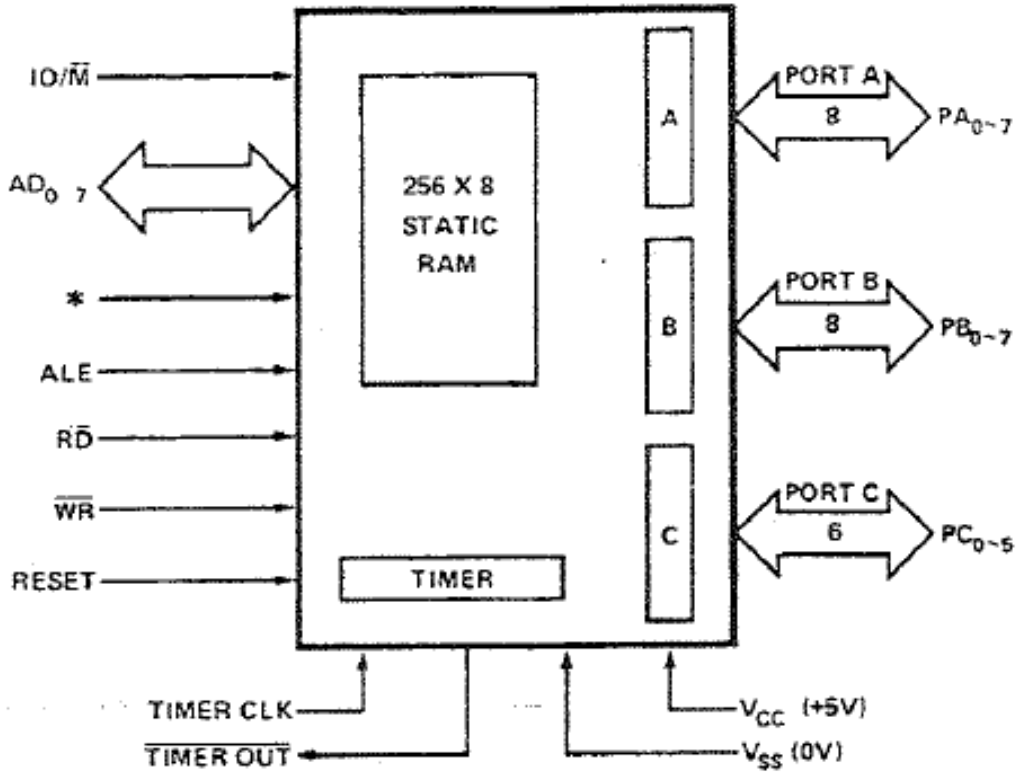

Block Diagram - 8155
8085
RESET
AD0-7
8
IO/M
RD
WR
ALE
CE
256 X 8
Static
RAM
Timer
A
B
C
Port A
8
PA0-7
Port B
8
PB0-7
Port C
6
PC0-5
I/O Devices
Timer CLK
TIMER OUT
Vcc (+5 V)
Vss (0 V)

# 8155

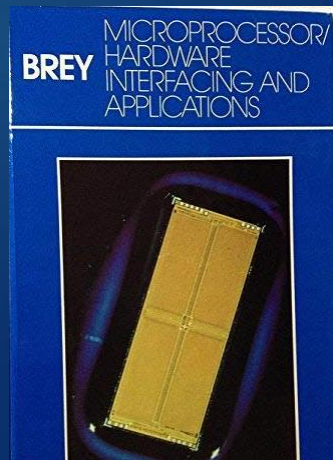
## PIN CONFIGURATION




## BLOCK DIAGRAM



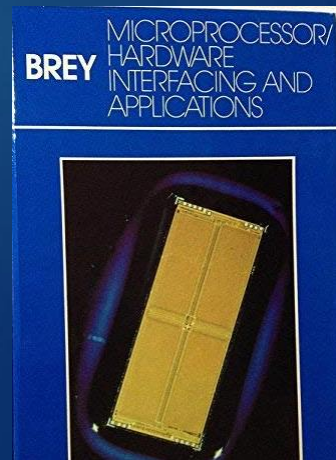

*: 8155/8155-2 = $\overline{CE}$, 8156/8156-2 = CE



# 8155

**TABLE 6–1** 8155 I/O port assignments.

| Port Address | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A15 | A14 | A13 | A12 | A11 or A3 | A10 | A9 | A8 | Selected Device |
| A7 | A6 | A5 | A4 | | A2 | A1 | A0 | |
| X | X | X | X | X | 0 | 0 | 0 | Command/Status |
| X | X | X | X | X | 0 | 0 | 1 | Port A |
| X | X | X | X | X | 0 | 1 | 0 | Port B |
| X | X | X | X | X | 0 | 1 | 1 | Port C |
| X | X | X | X | X | 1 | 0 | 0 | LSB of timer |
| X | X | X | X | X | 1 | 0 | 1 | MSB of timer |



# 8155

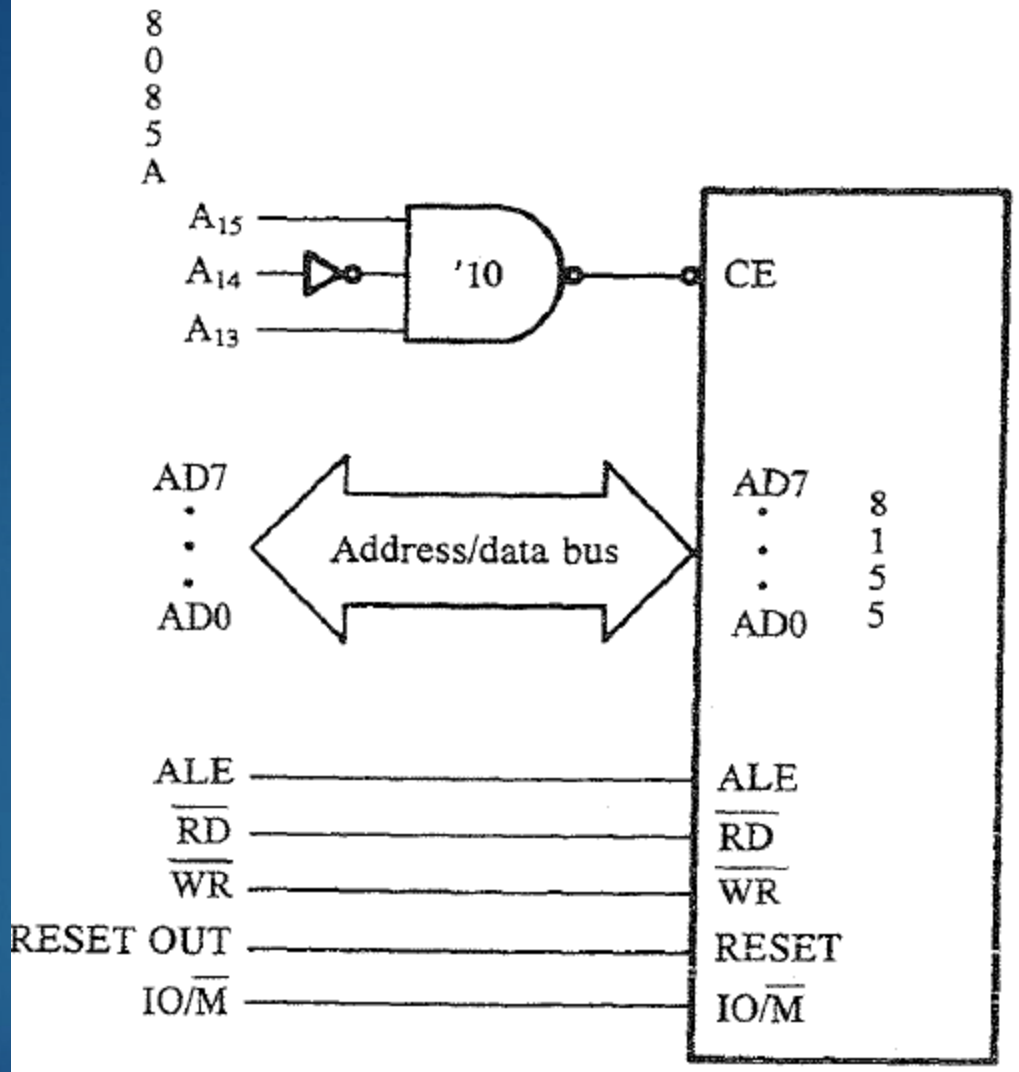
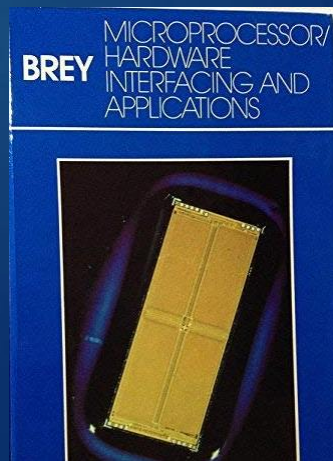



**FIGURE 6–20** An example decoder and interconnection diagram from the 8155 to the 8085A microprocessor.



# 8155

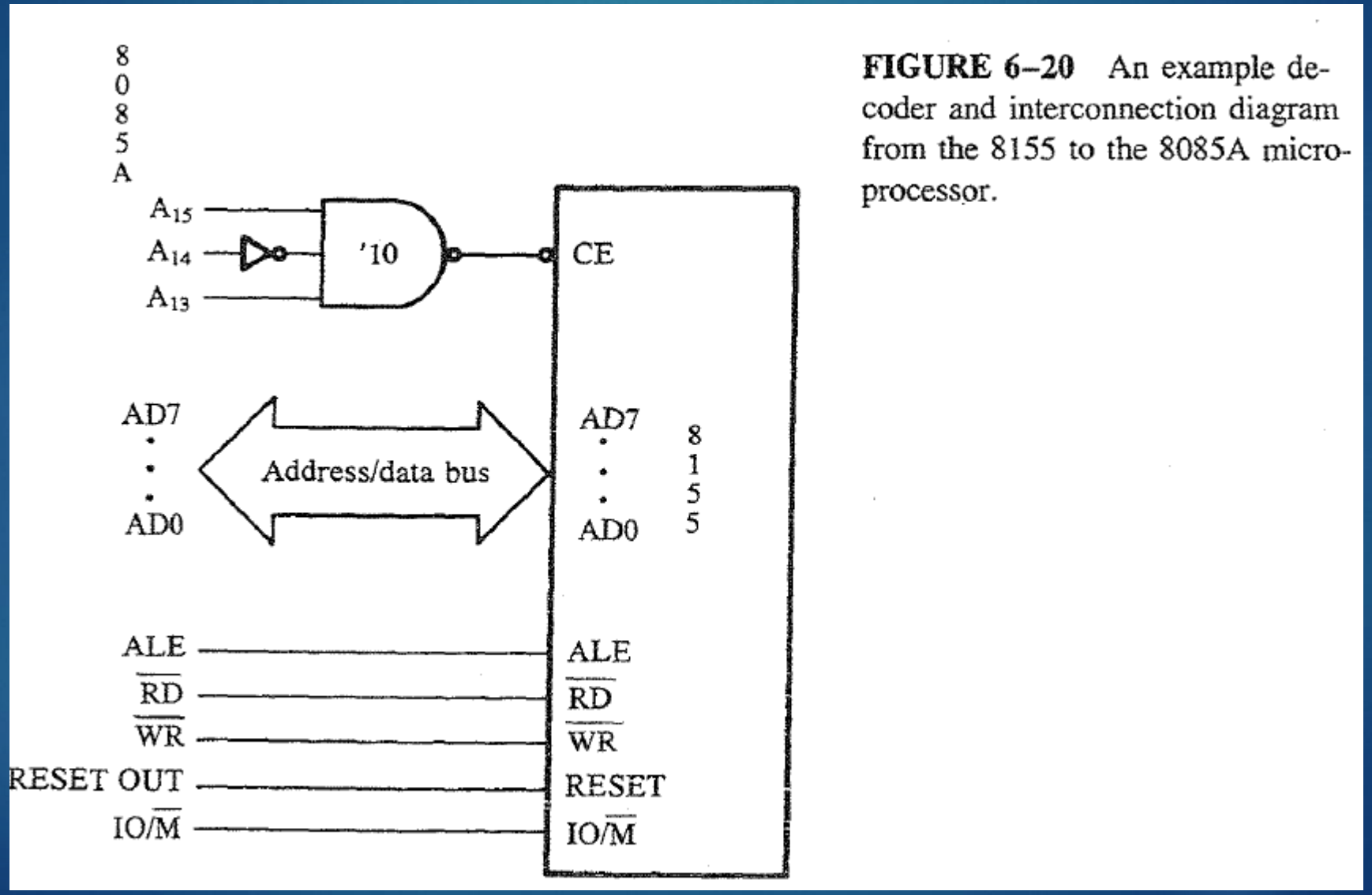
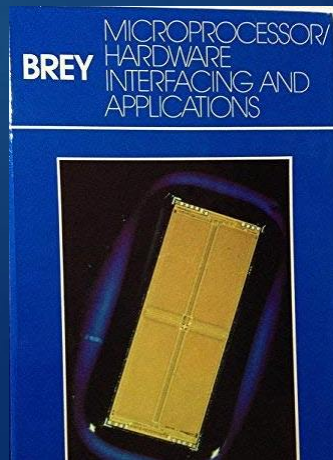



**FIGURE 6–20** An example decoder and interconnection diagram from the 8155 to the 8085A microprocessor.



# 8155

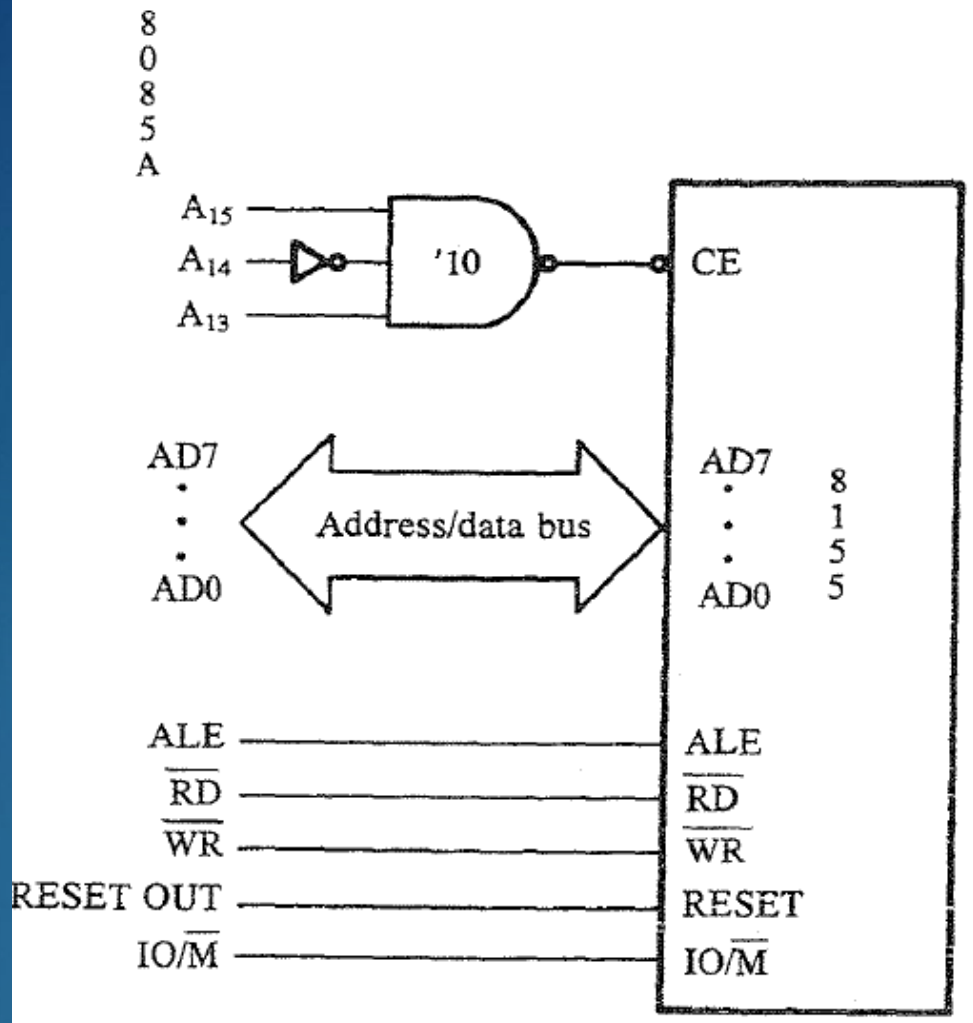
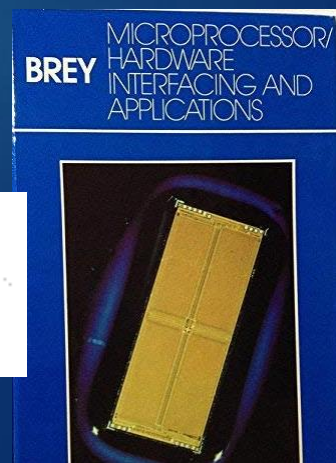



**FIGURE 6–20** An example decoder and interconnection diagram from the 8155 to the 8085A microprocessor.

MEMORY = 101X XXXX XXXX XXXX = A000H to BFFFH

I/O = 101X XXXX = A0H to BFH

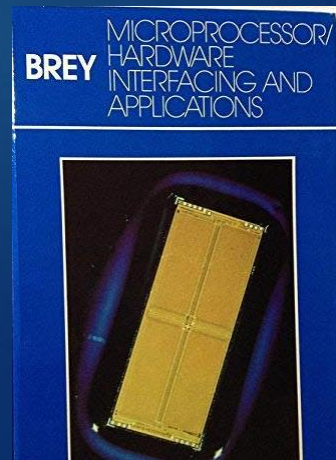


# 8155 Command Register

- The 8155 is a programmable I/O device that must be programmed for normal operation so that the I/O pins can be controlled and the timer can be set with a count and started if needed.
- The command register, which accomplishes the programming, is the internal register that directs the operation of the 8155.

**Command Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

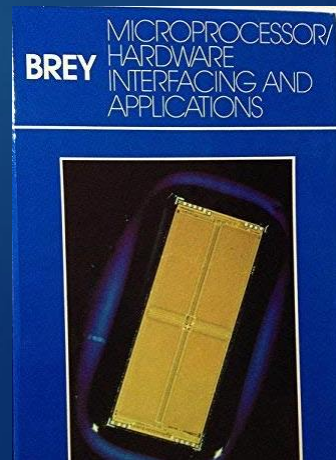


# 8155 Command Register

**Command Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

| Bit Positions | Function |
|---|---|
| 7 and 6 | Program the operation of the Timer<br>0 0 = Timer NOP<br>0 1 = Stop timer<br>1 0 = Stop timer after terminal count<br>1 1 = Start timer |
| 5 | Programs the port B interrupt function<br>0 = Disable interrupt port B<br>1 = Enable interrupt port B |
| 4 | Programs the port A interrupt function<br>0 = Disable interrupt port A<br>1 = Enable interrupt port A |
| 3 and 2 | Program the operation of port C<br>0 0 = Input<br>0 1 = ALT mode 3<br>1 0 = ALT mode 4<br>1 1 = Output |
| 1 | Programs the operation of port B<br>0 = Input<br>1 = Output |
| 0 | Programs the operation of port A<br>0 = Input<br>1 = Output |



# 8155 Timer

The tier count can range in value from 2H to 3FFFH and can generate a squarewave or pulse. The duration of the pulse output is equal to one input clock period, while the square wave is symmetrical.

**TABLE 6–3** 8155 timer ports.

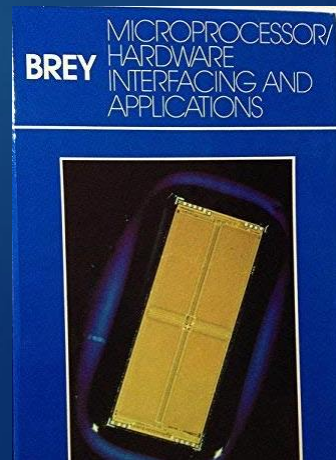
**Port XXXX X100 (LSB)**

| C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
|---|---|---|---|---|---|---|---|

**Port XXXX X101 (MSB)**

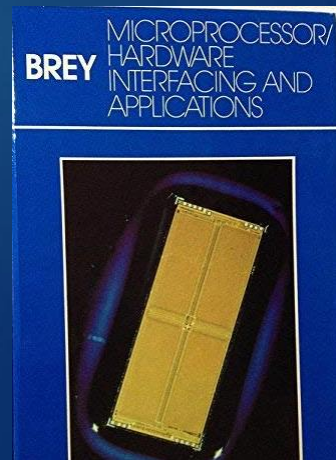
| M2 | M1 | C13 | C12 | C11 | C10 | C9 | C8 |
|---|---|---|---|---|---|---|---|

| Bit Assignment | Function |
|---|---|
| C13 to C0 | 14-bit count |
| M2 and M1 | Timer waveform control |
| | 0 0 = Single square wave |
| | 0 1 = Continuous square wave |
| | 1 0 = Single pulse |
| | 1 1 = Continuous pulse |



# 8155 - Timer

- The timer count can range in value from 2H to 3FFFH and can generate a squarewave or pulse.
- The duration of the pulse output is equal to one input clock period, while the square wave is symmetrical.



# 8155 - Timer

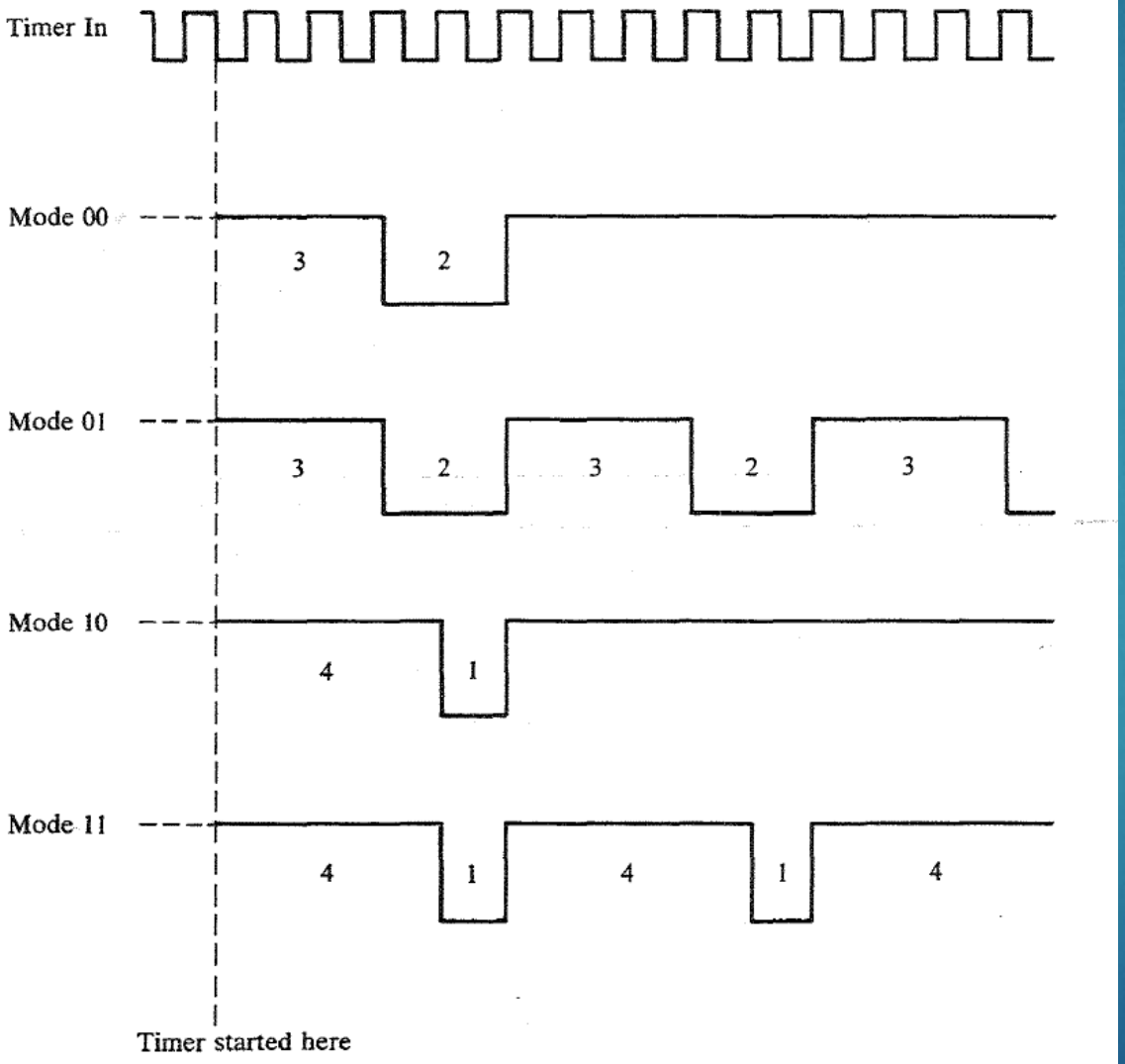



**FIGURE 6–21** The operation of the 8155 timer using a count of five.

**TABLE 6–3** 8155 timer ports.

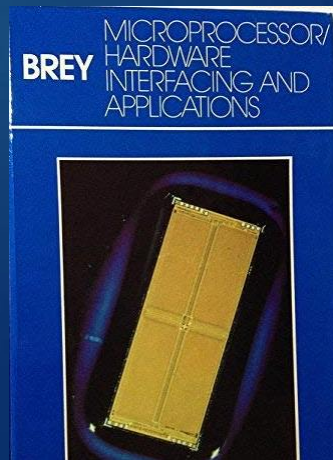
**Port XXXX X100 (LSB)**

| C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
|---|---|---|---|---|---|---|---|

**Port XXXX X101 (MSB)**

| M2 | M1 | C13 | C12 | C11 | C10 | C9 | C8 |
|---|---|---|---|---|---|---|---|

| Bit Assignment | Function |
|---|---|
| C13 to C0 | 14-bit count |
| M2 and M1 | Timer waveform control |
| | 0 0 = Single square wave |
| | 0 1 = Continuous square wave |
| | 1 0 = Single pulse |
| | 1 1 = Continuous pulse |



# 8155 - Example

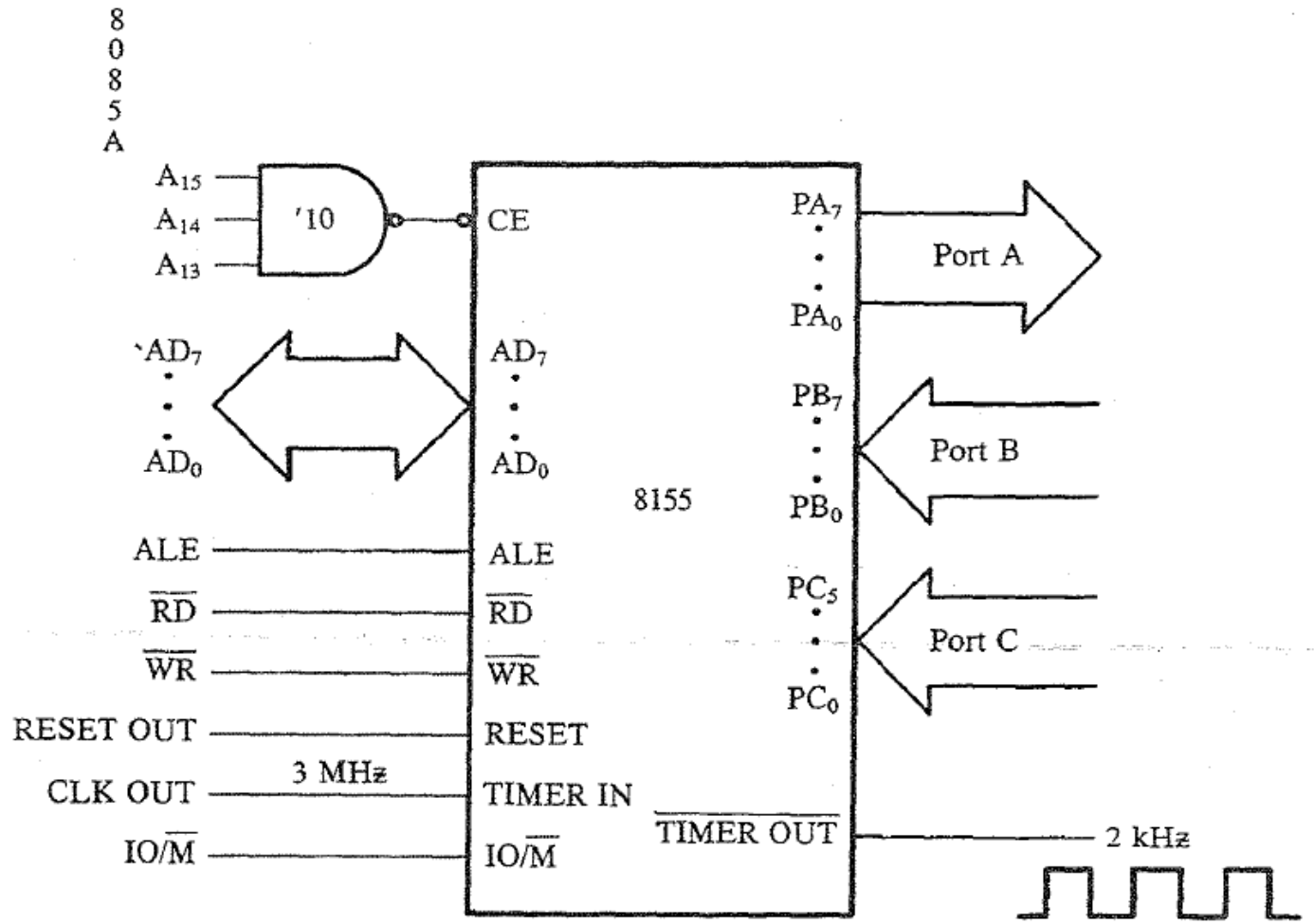
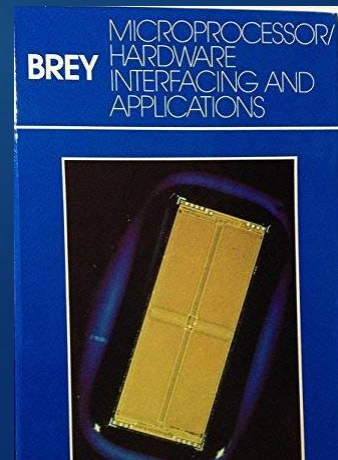
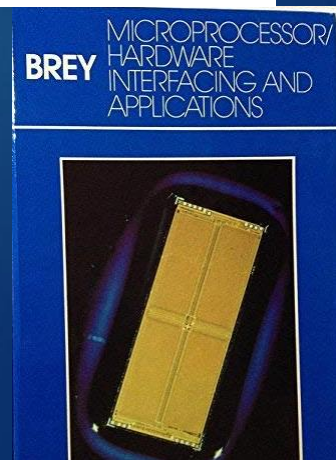



**FIGURE 6–22** An example 8155 interface with Port A designated as an output, Ports B and C designated as inputs, and the timer set up to divide the clock by 1500.



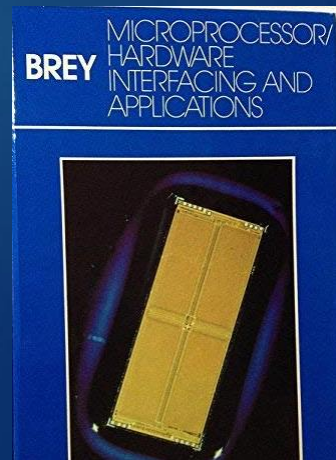
# 8155 - Example

```
                        ;8155 initialization dialog for the 8085A
                        ;
                        ;this software must be executed before any
                        ;8155 operation can be performed
                        ;
0000  3EDC      RESET:    MVI  A,0DCH
0002  D3E4                OUT  0E4H            ;set LSB of count
0004  3E45                MVI  A,45H
0006  D3E5                OUT  0E5H            ;set mode and MSB of count
0008  3EC1                MVI  A,11000001B
000A  D3E0                OUT  0E0H            ;program command register
```



# Handshaking with the 8155

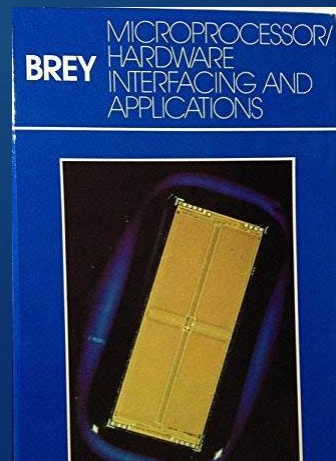
- **Handshaking** is a term that describes a communications protocol between two separate digital systems.
- This handshake, in many cases, is accomplished through two wires or signal lines.
- Handshaking synchronizes a device such as a printer to the microprocessor.
- This synchronization is accomplished, in many cases, with a pair of wires and some software.
- A wire from the microprocessor signals the printer that data are being sent to the printer.
- The printer, when ready to receive more data, sends a signal back to the microprocessor.
- This "handshake" between the microprocessor and printer synchronizes their operation.
- The microprocessor waits for the ready signal from the printer before sending another byte of information.



# 8155 – Handshaking - Alternate Modes of Operation

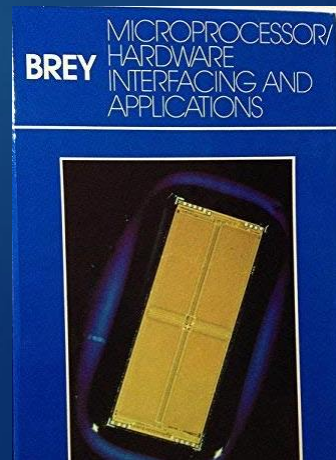
**TABLE 6–4** 8155 port C alternate modes ALT3 and ALT4.

| Pin # | ALT3 Pin Name | ALT4 Pin Name |
|---|---|---|
| PC0 | Port A INTR | Port A INTR |
| PC1 | Port A BF | Port A BF |
| PC2 | Port A $\overline{\text{STB}}$ | Port A $\overline{\text{STB}}$ |
| PC3 | Output pin | Port B INTR |
| PC4 | Output pin | Port B BF |
| PC5 | Output pin | Port B $\overline{\text{STB}}$ |



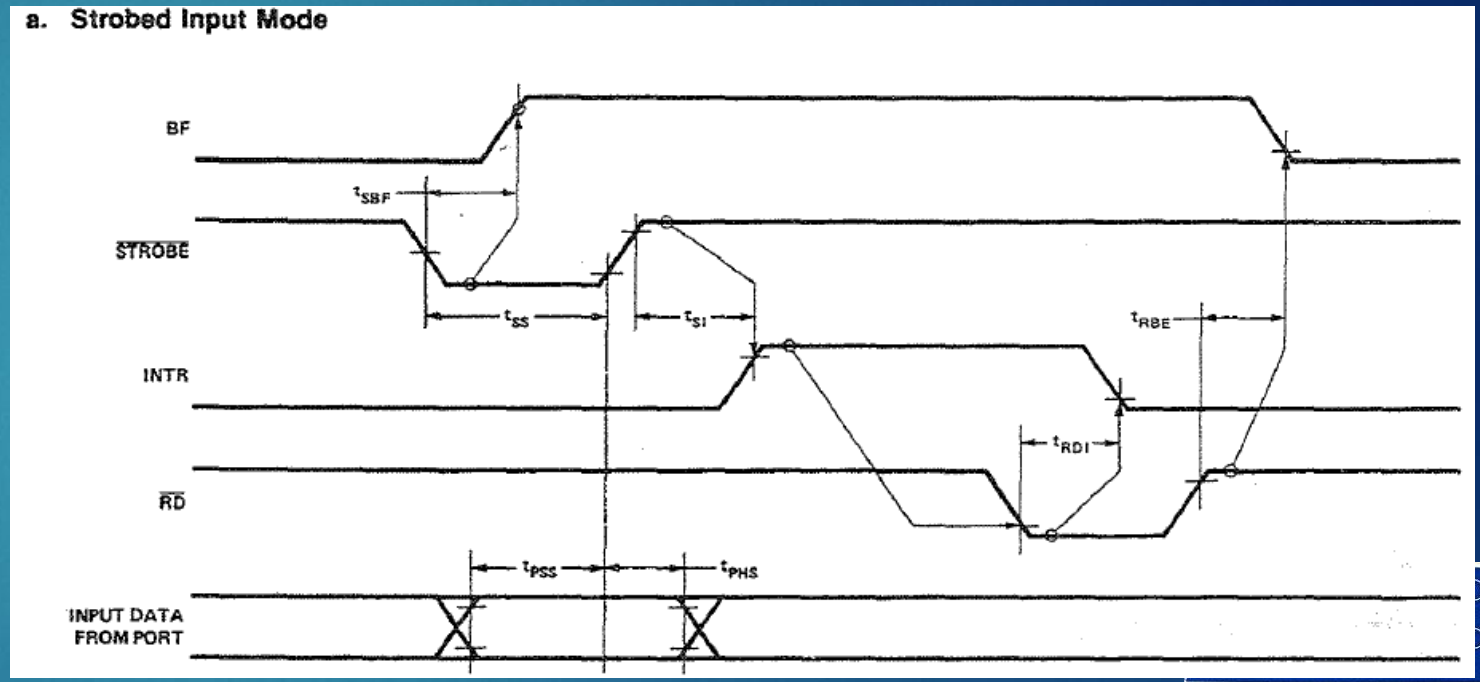
# 8155 – Handshaking - Alternate Modes of Operation

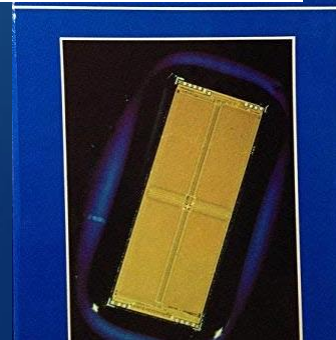
- The **STB, or strobe signal**, strobes data into or out of a port;
- The **BF, or buffer full signal**, indicates whether or not data are present inside the I/O port;
- and **INTR**, or interrupt request, becomes a one after the arrival of the STB signal. INTR causes an interrupt to occur in a microprocessor.



# 8155 Strobed Input Operation

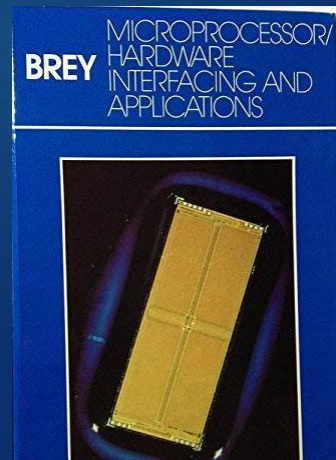
- Strobed input operation of the 8155 is illustrated in timing diagram. The strobe input, which comes · from the external device, causes the data to be held in an internal latch, forces the buffer full flag (BF) high, and forces the INTR signal high. The software provided by the user tests BF to determine if data have been strobed into the port by the STB signal.
- When BF is detected high, the software reads or inputs data from this port. The data are then transferred into the accumulator of the microprocessor, and the buffer full flag is cleared by the 8155. Buffer full is cleared whenever the data are read from the port.



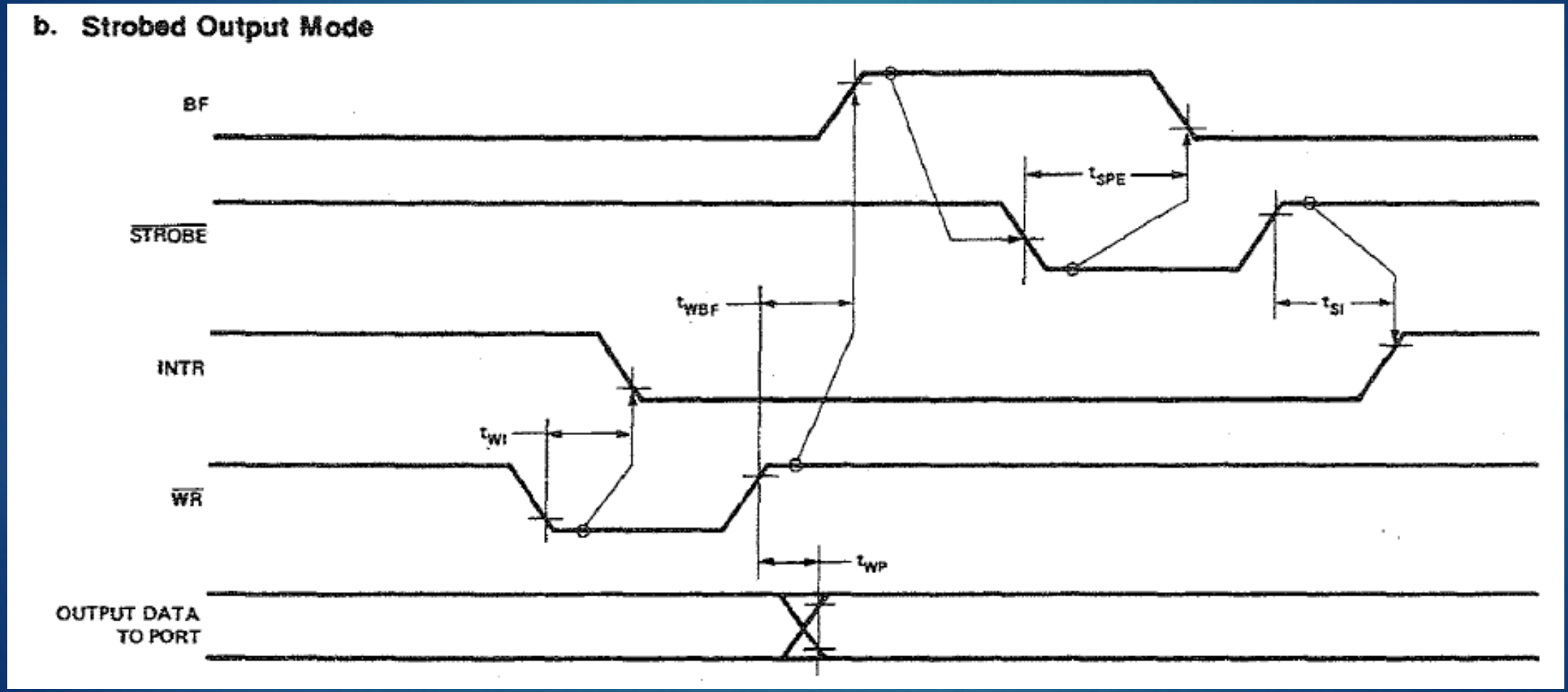



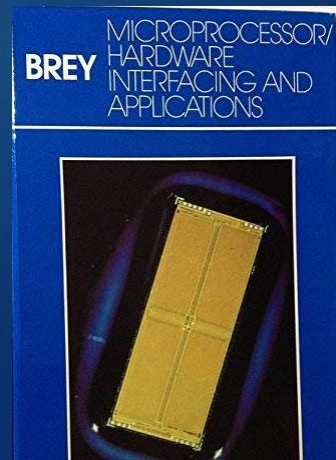
# 8155 Strobed Output Operation

- Strobed output operation is pictured in the timing diagram. The order of operation for strobed output is the reverse of strobed input. The data are first written into the port by the microprocessor, which causes the data to be held in an internal latch, clears INTR, and forces buffer full (BF) high.
- Buffer full indicates to the external device that data are present in the port. The external device responds by sending a strobe to the port, which indicates that it has received the information. The strobe then forces buffer full low and also sets the INTR signal.
- The software associated with this port checks the buffer full flag to determine if the external device has removed the data from the port. If not, the software waits for the buffer full flag to be cleared by the external strobe signal.
- This type of operation is useful with the printer interface discussed earlier in the text.



# 8155 Strobed Output Operation



# 8155 Keyboard Interface

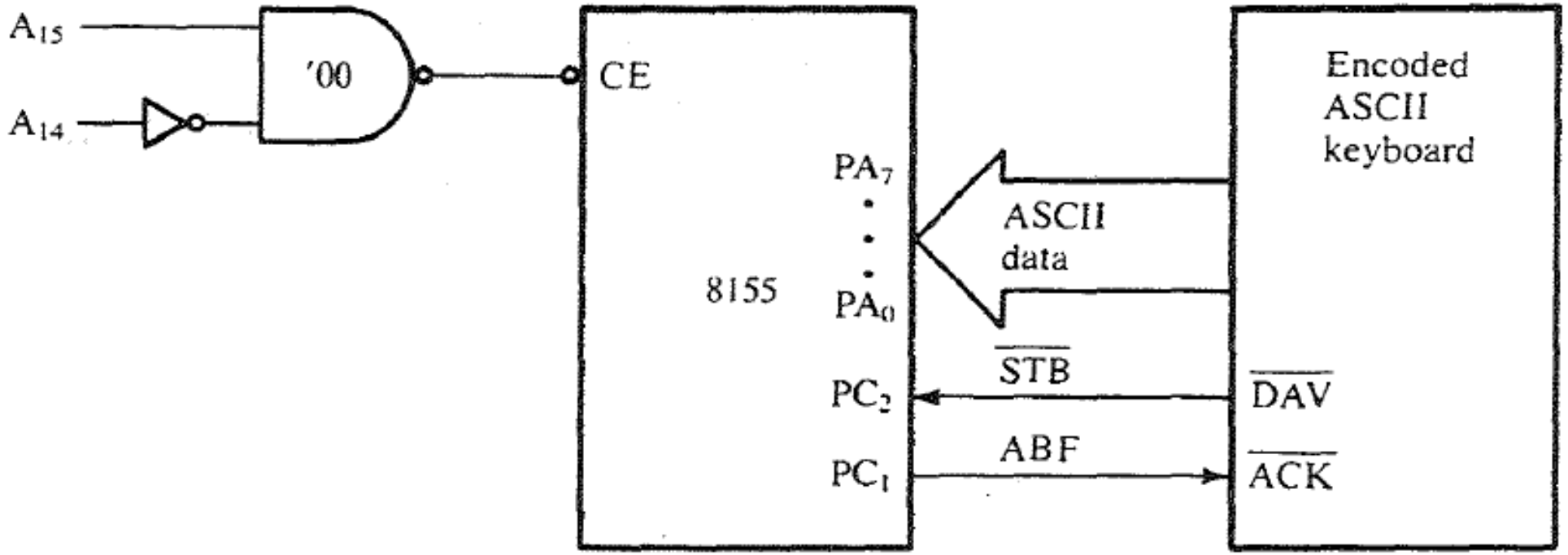
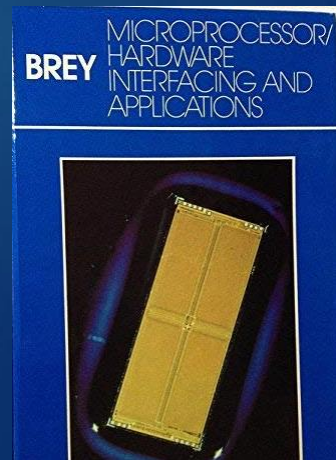
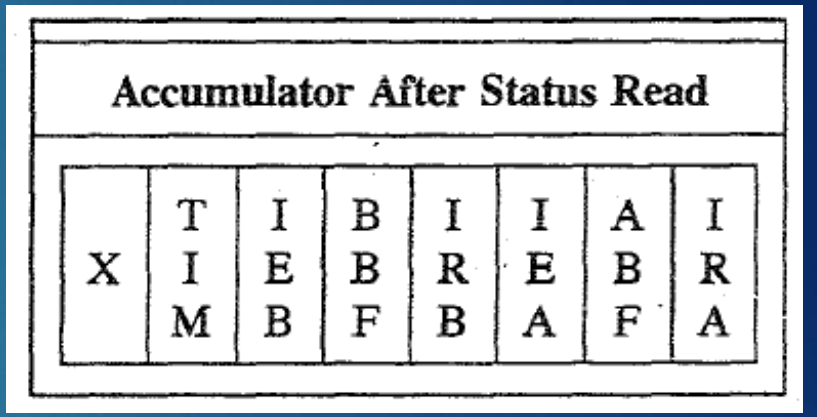



**FIGURE 6–24** An example of the 8155 connected to an ASCII encoded keyboard using strobed input operation.



# 8155 Status Register Bit Assignment

- **TIM** Whenever the internal timer reaches its terminal count, this bit position becomes a logic one. For example, if the timer is programmed to divide by ten, this bit becomes a logic one on the tenth clock pulse.
- **IEB** Whenever this bit position is a logic one, it indicates that the port B interrupt has been enabled through the command register.
- **BBF** The port B buffer full flag (BBF) indicates that data have been strobed into the port B latch for an input operation or that data have been extracted from the port for an output operation.
- **IRB** The interrupt request bit reflects the condition of the port B INTR pin.
- **IEA** This position indicates that interrupt port A has been enabled using the command register.
- **ABF** The port A buffer full flag (ABF) indicates that data have been strobed into port A for an input operation or extracted from port A for an output operation.
- **IRA** This position reflects the condition of the port A INTR signal.



| Accumulator After Status Read | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | TIM | IEB | BBF | IRB | IEA | ABF | IRA |

# The Keyboard Software

- To develop the software for this keyboard, the port A buffer full flag must be located. This flag bit is located in the status register of the 8155 and can be examined by inputting the status register at port number XXXX X000. In this example, figure 6-24, the I/0 port assignment is I0XX X000 (80H) for the command/status register.

```
                        ;8155 initialization dialog
                        ;
0000  3E04              INIT:    MVI    A,00000100B     ;select ALT3
0002  D380                      OUT    80H             ;Port A = Input
                        ;
0100                             ORG    100H
                        ;
                        ;INKEY subroutine
                        ;
0100  DB80              INKEY:   IN     80H             ;get status word
0102  E602                      ANI    2               ;isolate ABF
0104  CA0001                    JZ     INKEY           ;if no data
0107  DB81                      IN     81H             ;get data
0109  C9                        RET                    ;return
```