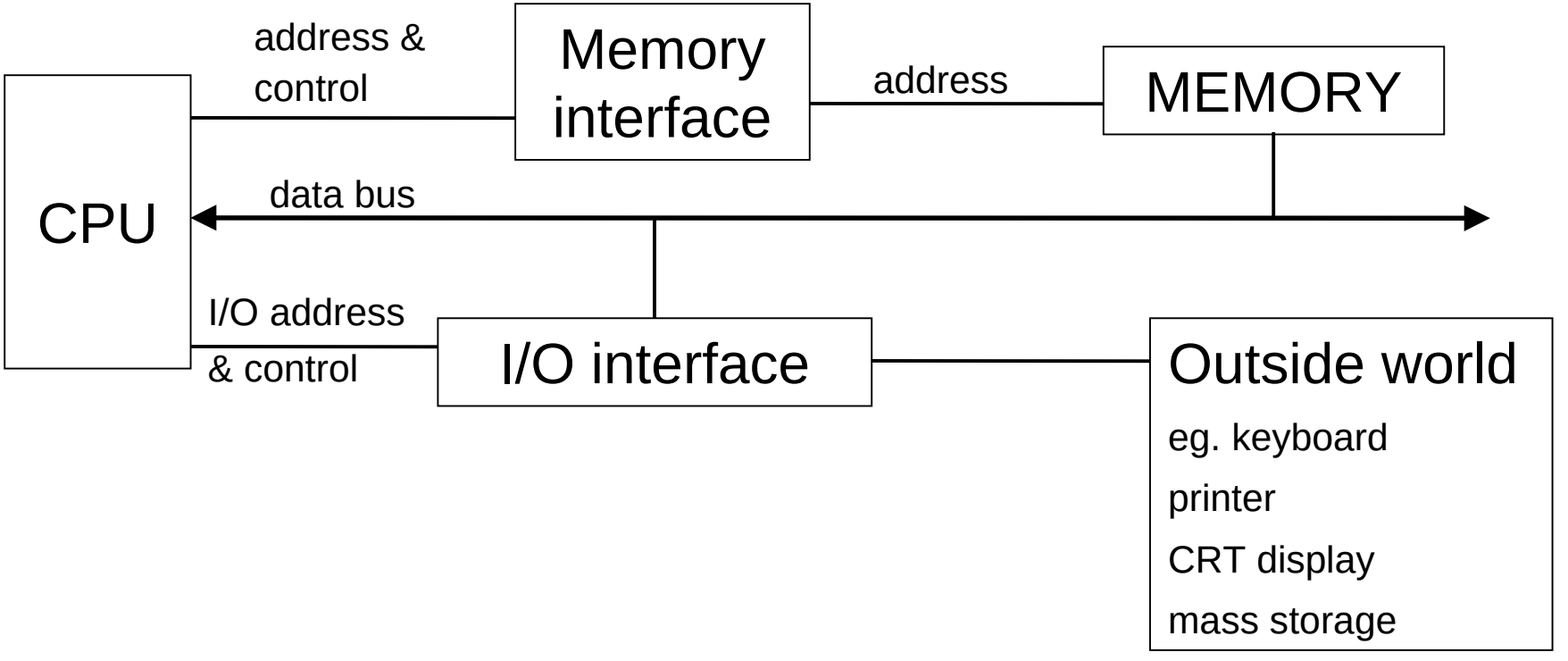
Introduction to Input-Output



I/O interface

- Input-output involves the transfer to (or from) peripheral devices from (or to) the data bus of the cpu eg. data transfer to the CRT display, from the keyboard, to/from a hard disk drive, to/from a modem to another computer system.
- Input-output operations fall into one of the following types:
  - *Programmed I/O* - cpu polls peripherals to check if I/O is needed
  - *Interrupt I/O* - peripheral sends an interrupt request to cpu for I/O
  - *Direct memory access (DMA)* - peripheral writes directly to memory
- 8088 uses **Isolated I/O** (I/O addresses are not part of memory address) as distinct from **Memory mapped I/O** (peripherals are mapped to locations in the memory address space).

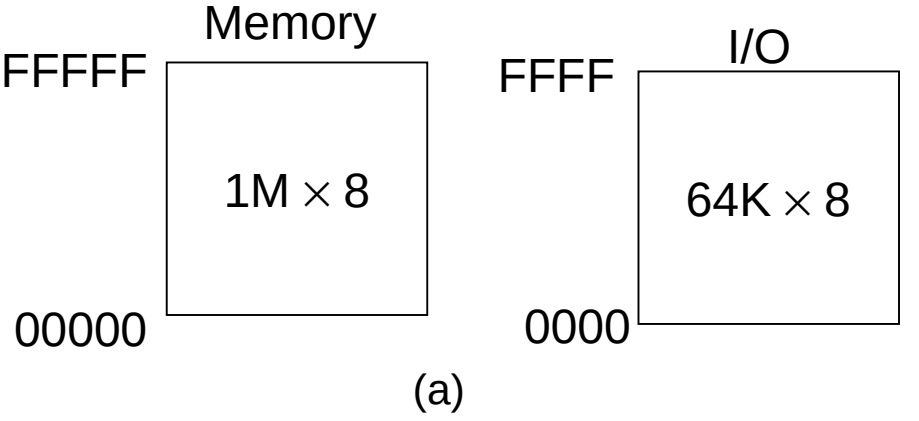
Q: pros and cons of isolated I/O and memory mapped I/O?

Memory and I/O Map

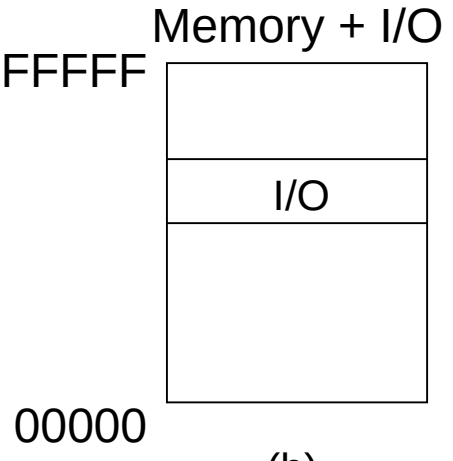
The Memory and I/O maps for the 8086/8088 microprocessor.

(a) Isolated I/O.

(b) Memory mapped I/O.

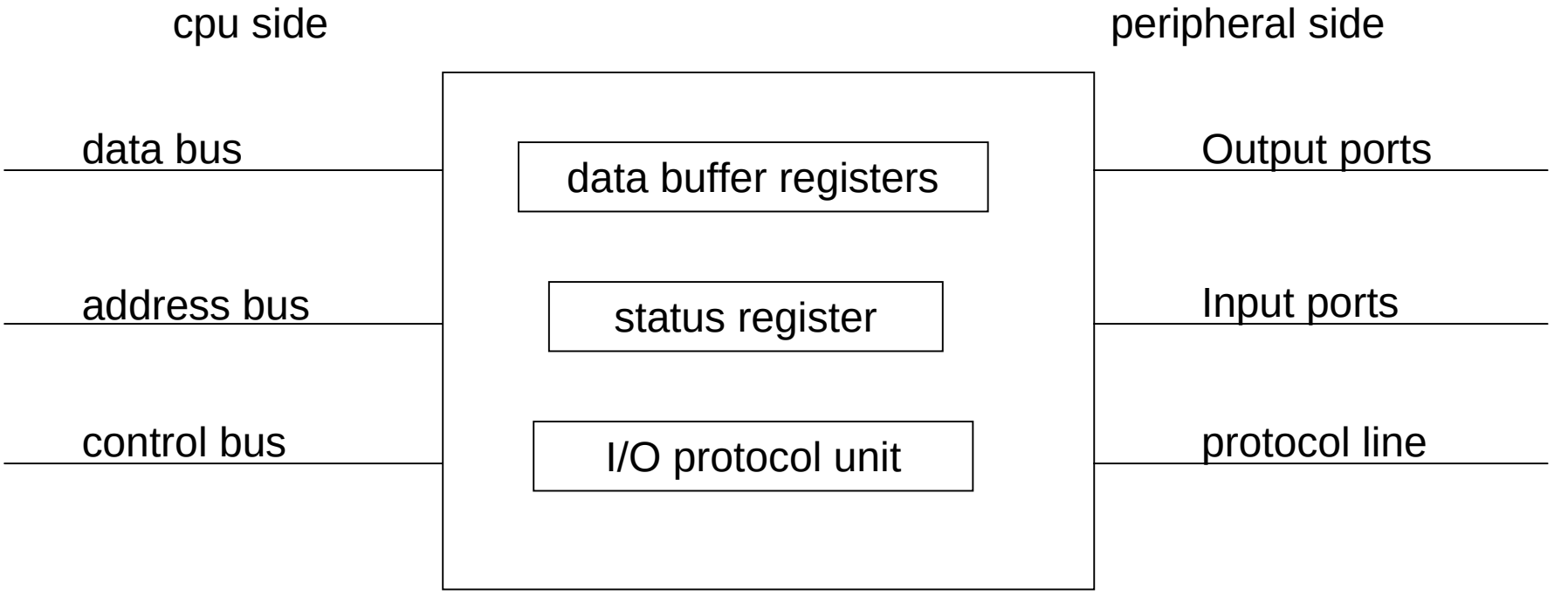


(a)



(b)

I/O interface



example of an I/O interface

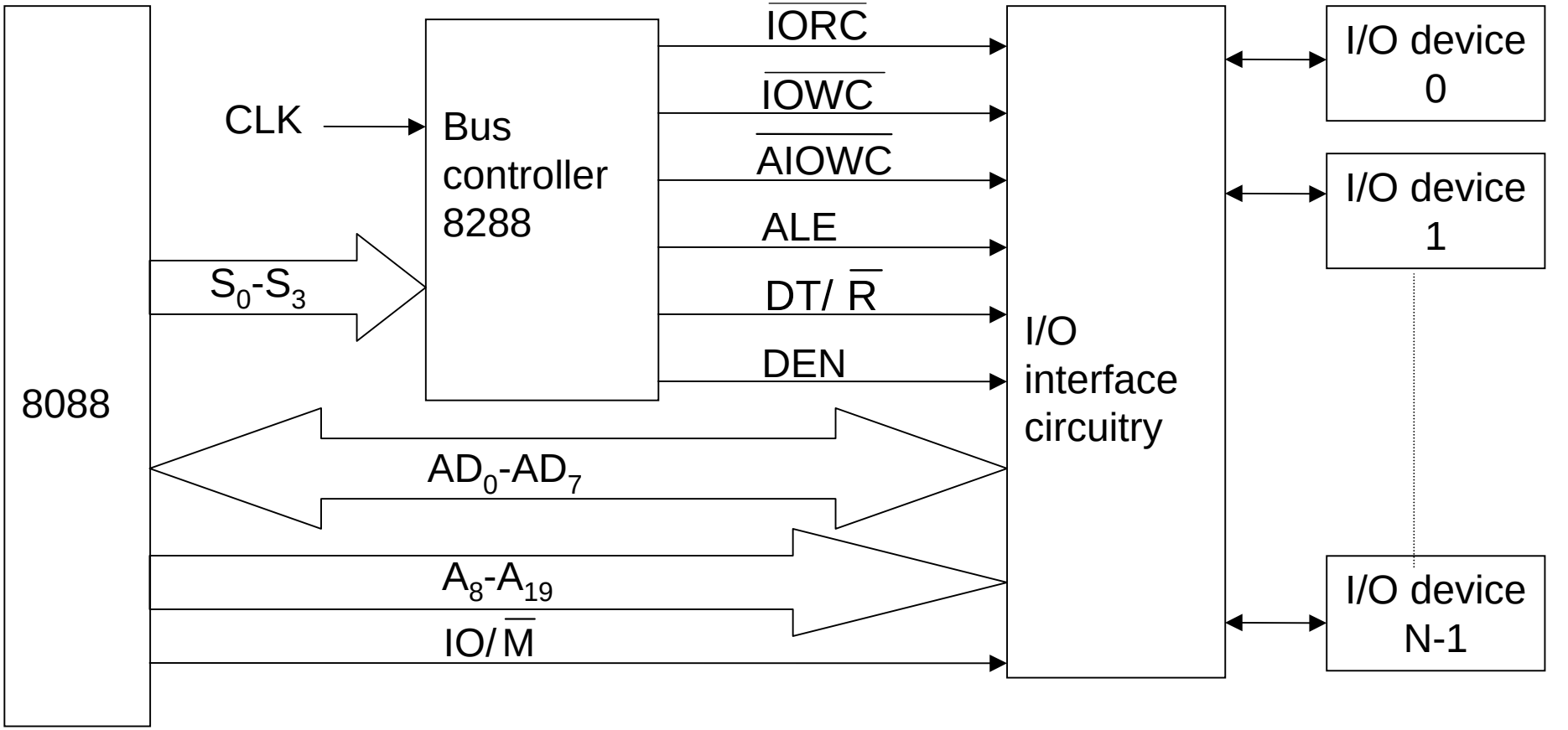
Typically, not all data, address or control lines are needed.
Input and output ports maybe the same.

I/O interface

- I/O interface functions can include
  - data storage buffer for sending and receiving data
  - low-level communications protocol (handshaking)
  - data format conversion (eg. parallel/serial)
  - error detection
  - addressing of different peripherals

- I/O interface are typically implemented by LSI (large scale integration) - many different types are available from different manufacturers.
- Data can be transferred through I/O interface by either programmed I/O or interrupt I/O. DMA typically needs a separate controller.

8088 maximum mode I/O interface



Generalized I/O interface connections to maximum mode 8088

Parallel and Serial Data Transfer

- Data Transfer between the I/O interface and the peripheral can involve either parallel or serial data transmission, depending on the peripheral and the actual implementation of the I/O interface.

- Parallel data transfer involves using at least 8 separate lines for the 8 data bits in a byte. Normally, other lines are needed for the communications protocol (eg. STB [data strobe] line to indicate when data is valid, ACK line to acknowledge data has been read).

  STB: sent by sender ACK:sent by receiver

- Parallel data transfer is usually fast. Each data bit typically needs it's own ground return line to reduce noise. A popular parallel data transfer interface standard is the CENTRONICS type interface which uses a 36-pin connector. The centronics interface is commonly used in printers.

Parallel and Serial Data Transfer (cont.)

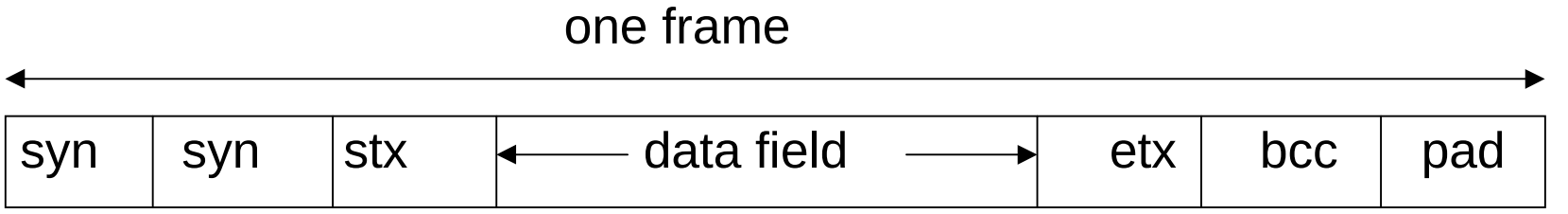
- Serial data transfer involves sending the data on a single line, bit by bit. The I/O interface converts the data from parallel to serial or vice-versa using shift registers.

- Serial connections are commonly used for data transfer over longer distances (eg over telephone line). A popular standard for serial data transmission is the RS232C standard.

Serial I/O

- Serial I/O can be either

(1) Synchronous - data are sent in blocks, with start and end-of-block markers. Individual characters within a block do not need start and stop bits since the receiver identifies every 8 bits as one character, eg.



syn = sync character (ascii code 16)

stx = start of text (ascii code 02)

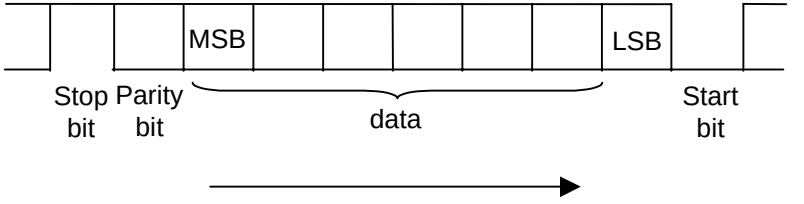
etx = end of text (ascii code 03)

bcc = block check characters (error detection)

pad = end of frame pad (ascii code ff)

Serial I/O

(2) Asynchronous - no block synchronization bits. Each character is identified by the start and stop bit(s) (stop bits can be 1, $1\frac{1}{2}$, 2 bits) inserted at the start and end of each character.



- Synchronous serial data transfer is more efficient (ie. faster) since asynchronous transfer “wastes” about 30% of the bits for start and stop bits in sending a 7-bit ASCII code.

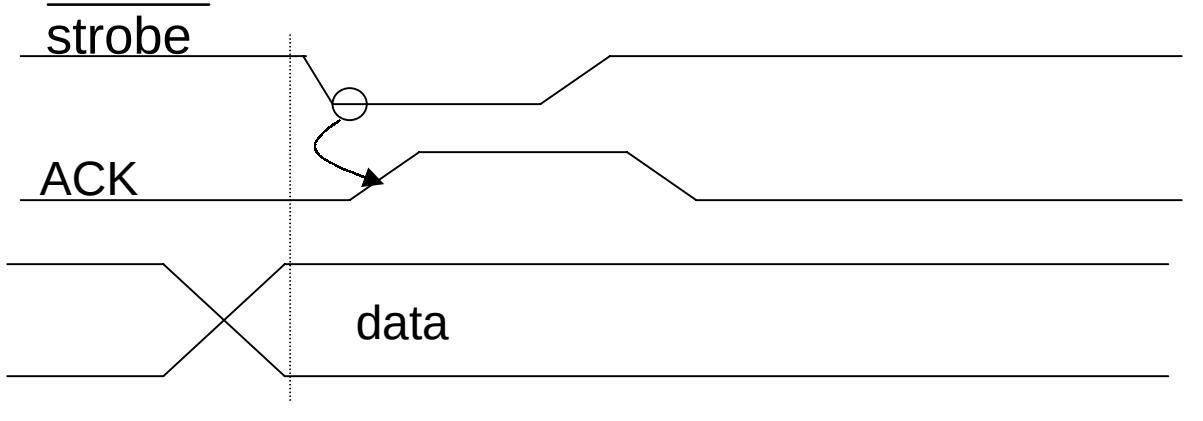
- An example of asynchronous serial data transfer is the RS232 serial port found in most computers.

8255 Programmable Peripheral Interface (PPI)

- Intel 8255A is a general purpose parallel I/O interface. It provides three I/O port (A, B and C) and can be programmed as
  - Simple parallel I/O - no handshaking implemented
  - Single handshake I/O - uses STB-ACK handshake
  - Double handshake I/O - uses STB-ACK and STB-ACK

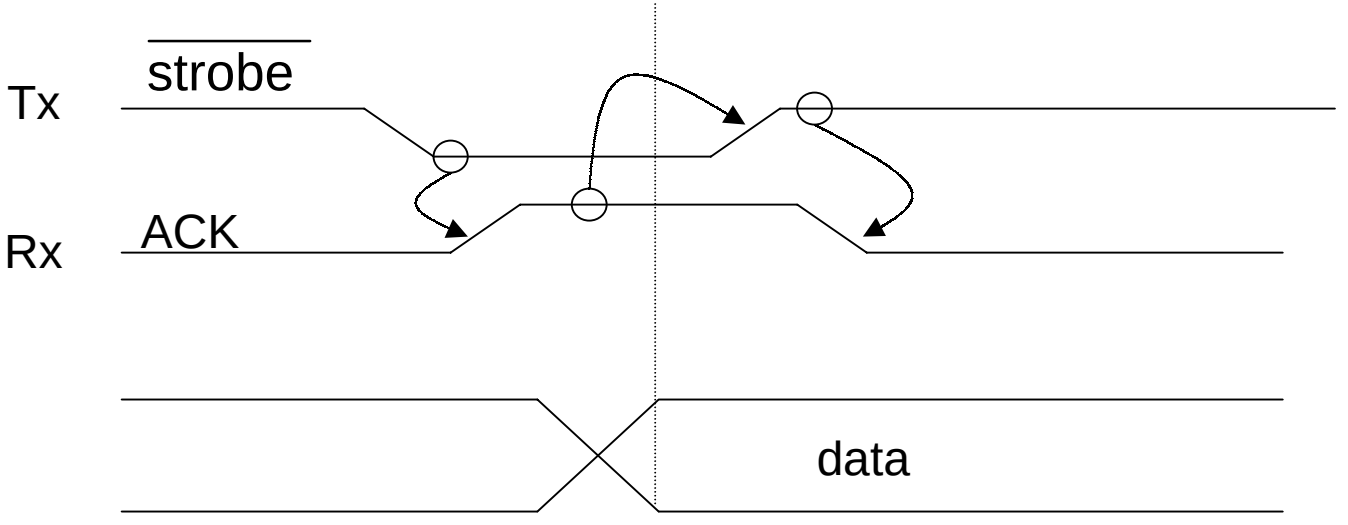
simple

(no handshake)



Single Handshake

8255 Programmable Peripheral Interface (PPI) (cont.)



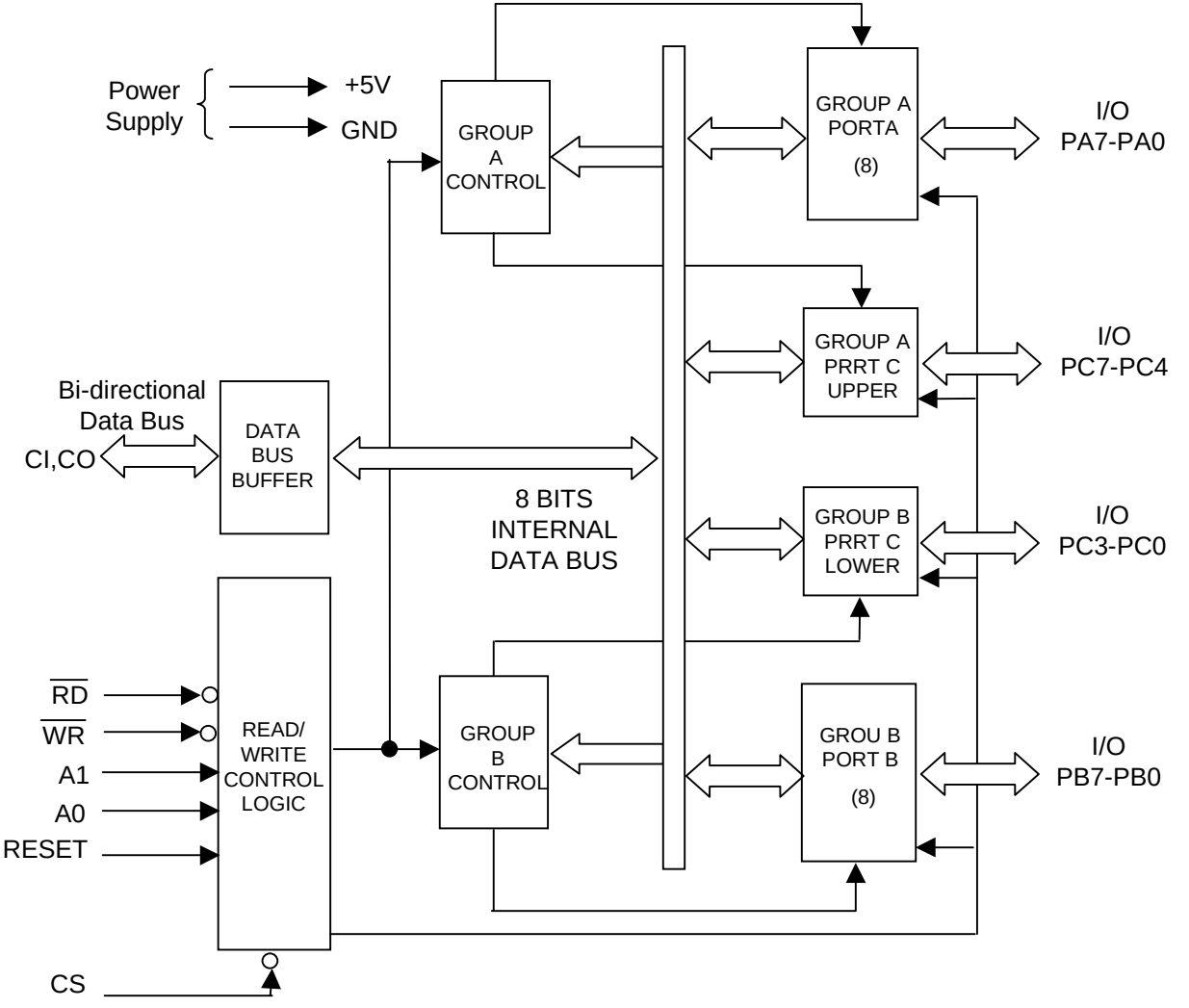
Double Handshake

Data transferred after the first STB and ACK.

- 8255A's mode of operation is determined by the contents of its control register (see Intel data sheet for further details).
- Port A and Port B can be set to different mode and input/output independently.

8255A Programmable Parallel Port Device

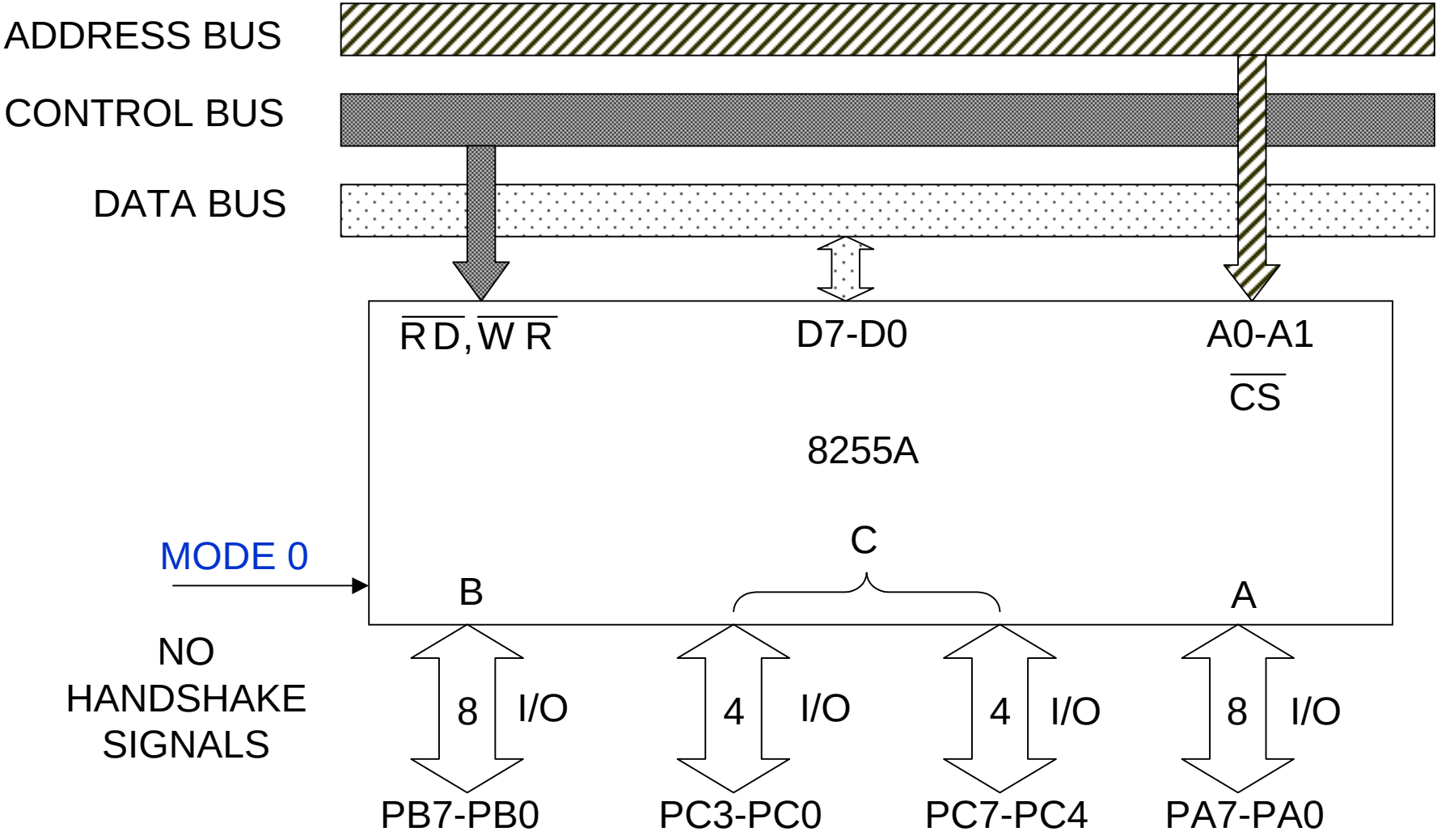
Internal structure block diagram



8255A Mode 0 Operation

- In Mode 0 operation, no handshaking will be used.
- If both port A and port B are initialized as mode 0 operation, port C can be used together as an additional 8-bit port, or two 4-bit ports.
- When used as an outputs, port C line can be set/reset by sending special control word to the control register address.
- The two halves of port C are independent and can be set as input or output port independently.

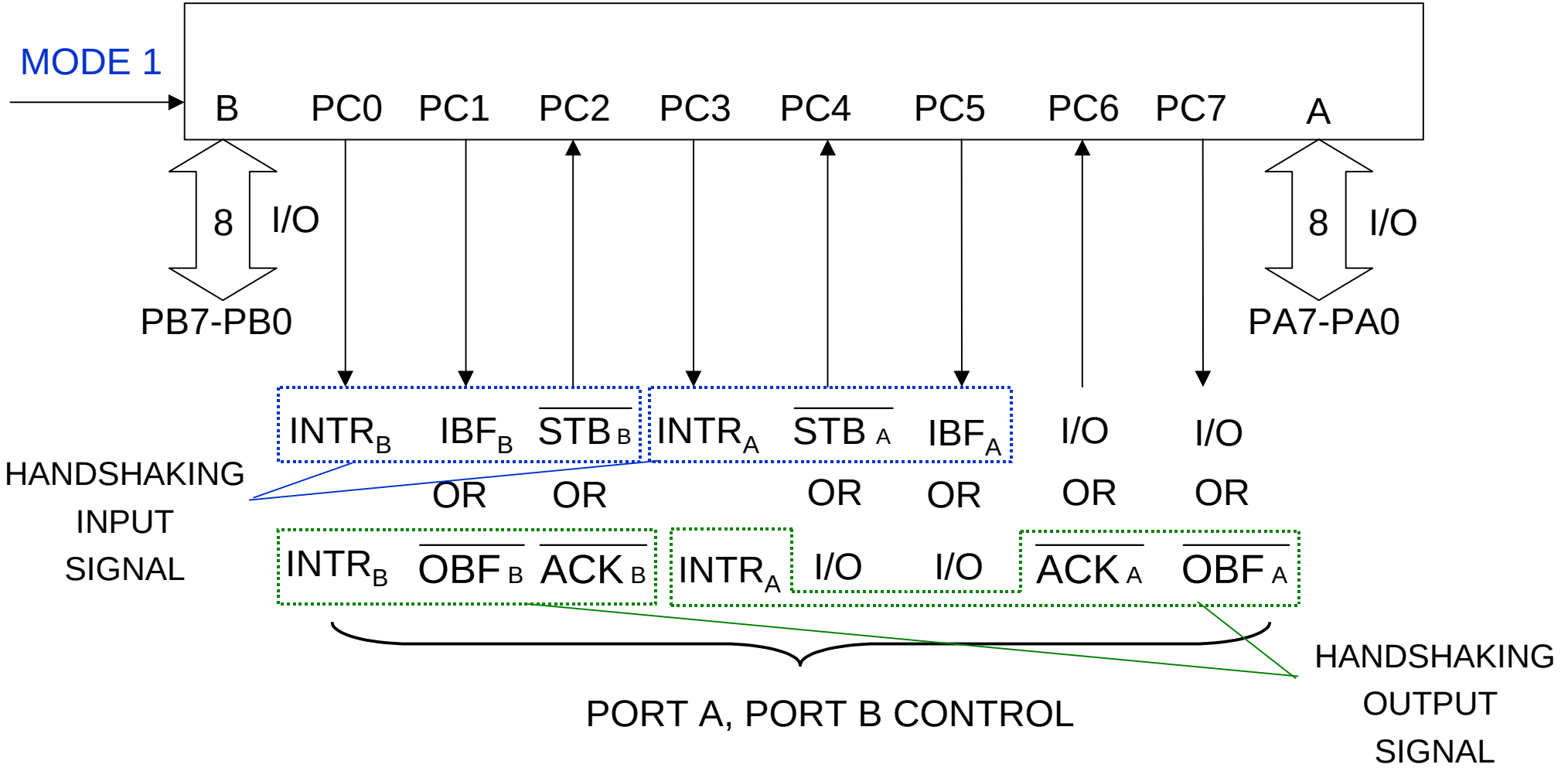
8255A Mode 0 Operation



8255A Mode 1 Operation

- In Mode 1 operation, single handshaking (strobed) is used.
- In this mode, some of the port C pins are used for handshaking purpose.
  - Port A set to mode 1 & input port: PC3, PC4, and PC5 pins are used.
  - Port A set to mode 1 & output port: PC3, PC6, and PC7 pins are used.
  - Port B set to mode 1 & input port: PC0, PC1, and PC2 pins are used.
  - Port B set to mode 1 & output port: PC0, PC1, and PC2 pins are used.

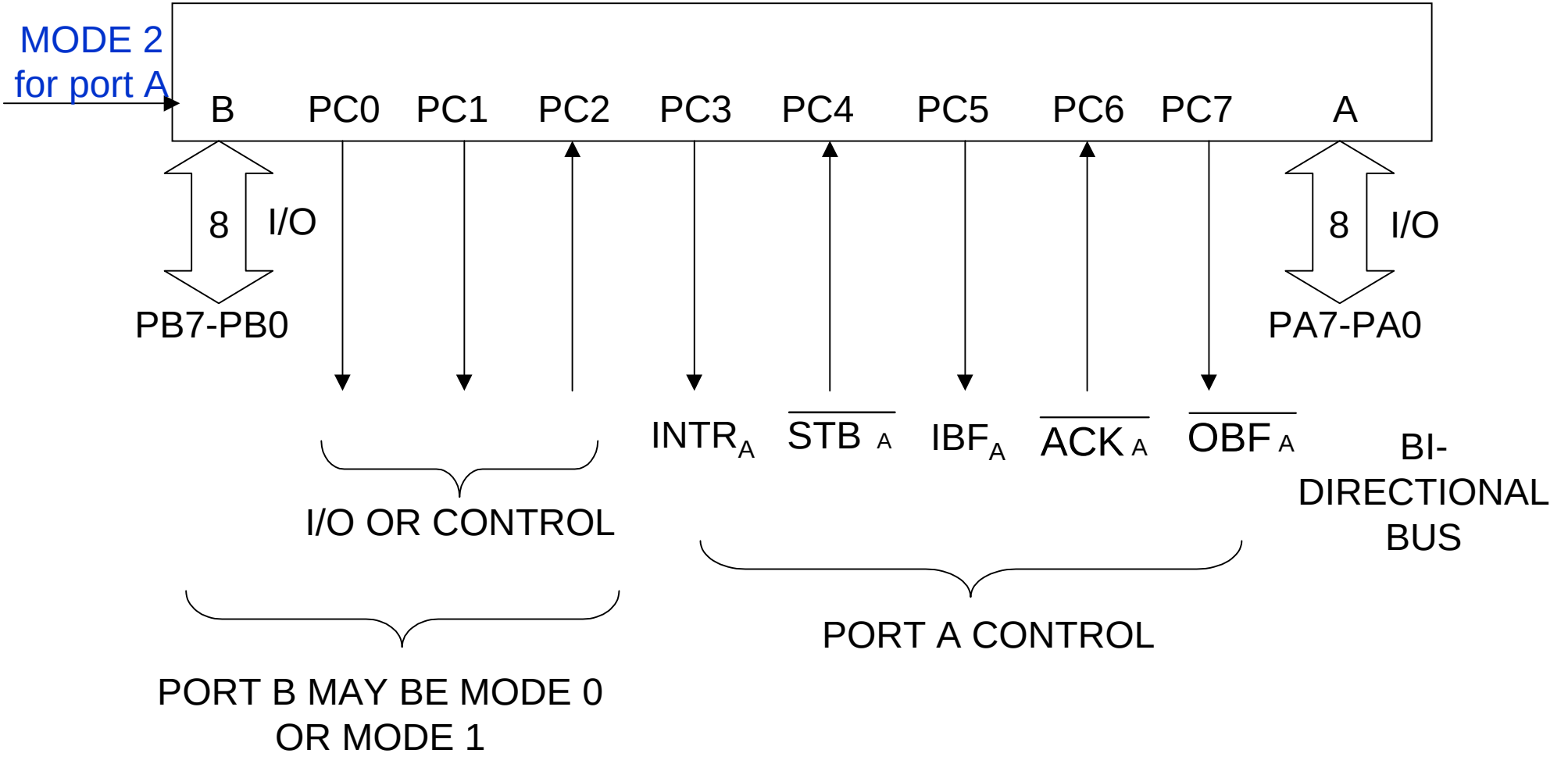
8255A Mode 1 Operation



8255A Mode 2 Operation

- Only port A can be initialized in mode 2.
- In mode 2, port A can be used as *bi-directional handshake data transfer*.
- In this mode, PC3-PC7 are used for handshake lines.
- PC0-PC2 can be used as I/O pins if port B is set to mode 0.
- PC0-PC2 can be used as handshake lines if port B is set to mode 1.

Summary of Port C Usage



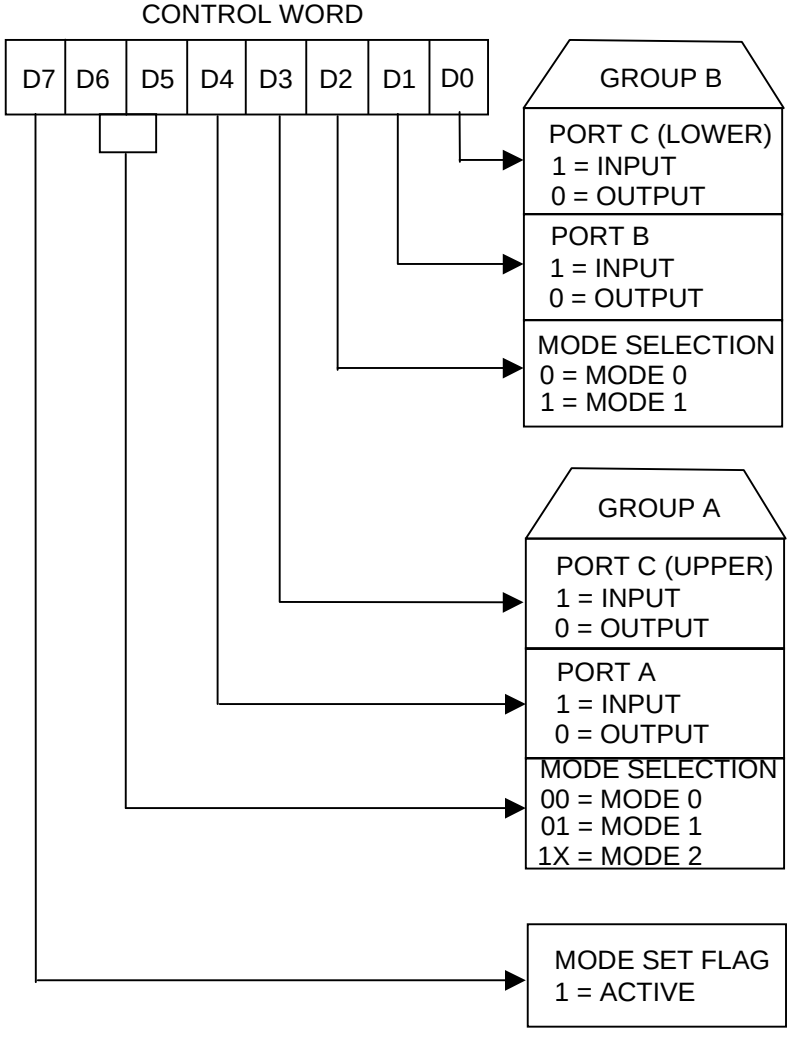
8255A Control Words

- Two control word formats are used:
  (1) mode-set control word format : to set the modes of each port
  (2) port C bit set/rest control word format : to set the particular bit in port C.
- These two formats are differentiated by the MSB of the control word.
- The control words can be sent to the corresponding address where 8255A is assigned to.

  Ex. Assume 8255A is located at FFF8H, control register address is FFFEH and the control word is to be set to 10001110B

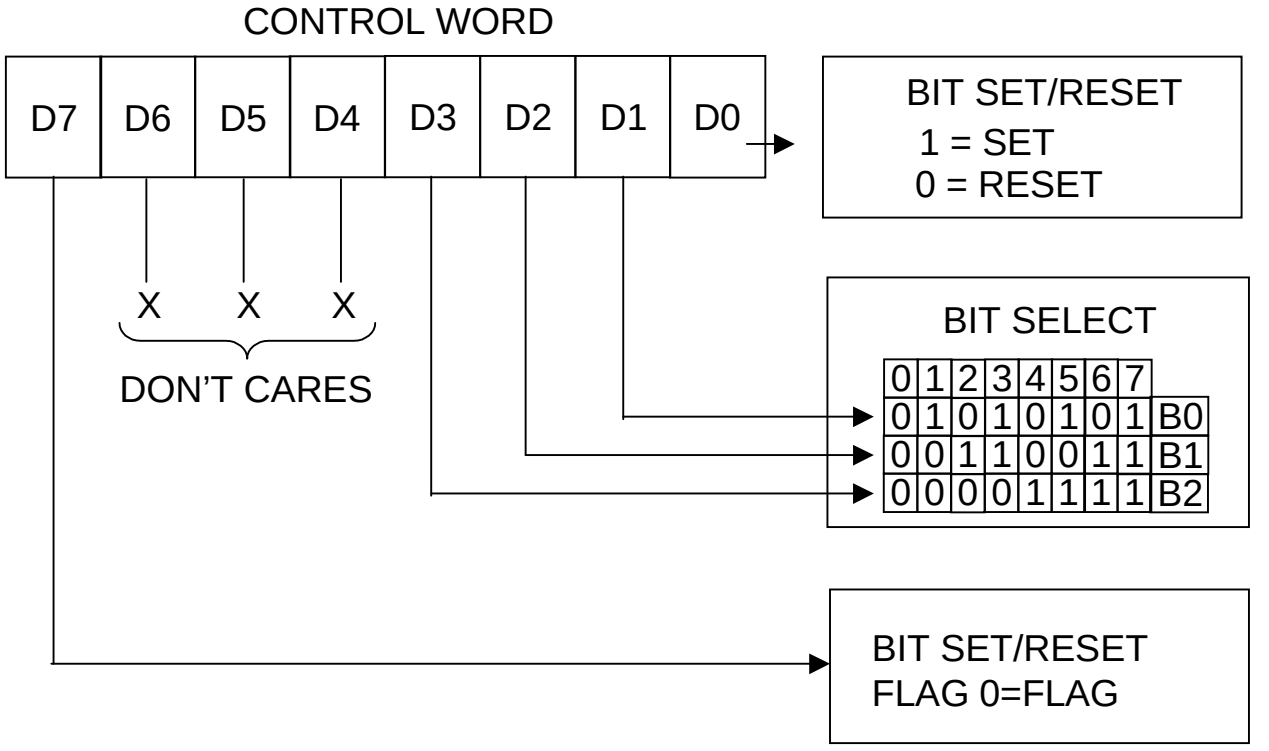
  → MOV AL, 10001110B
  MOV DX, 0FFFEH
  OUT DX, AL

8255A Control Word Formats



(a) Mode-set control word

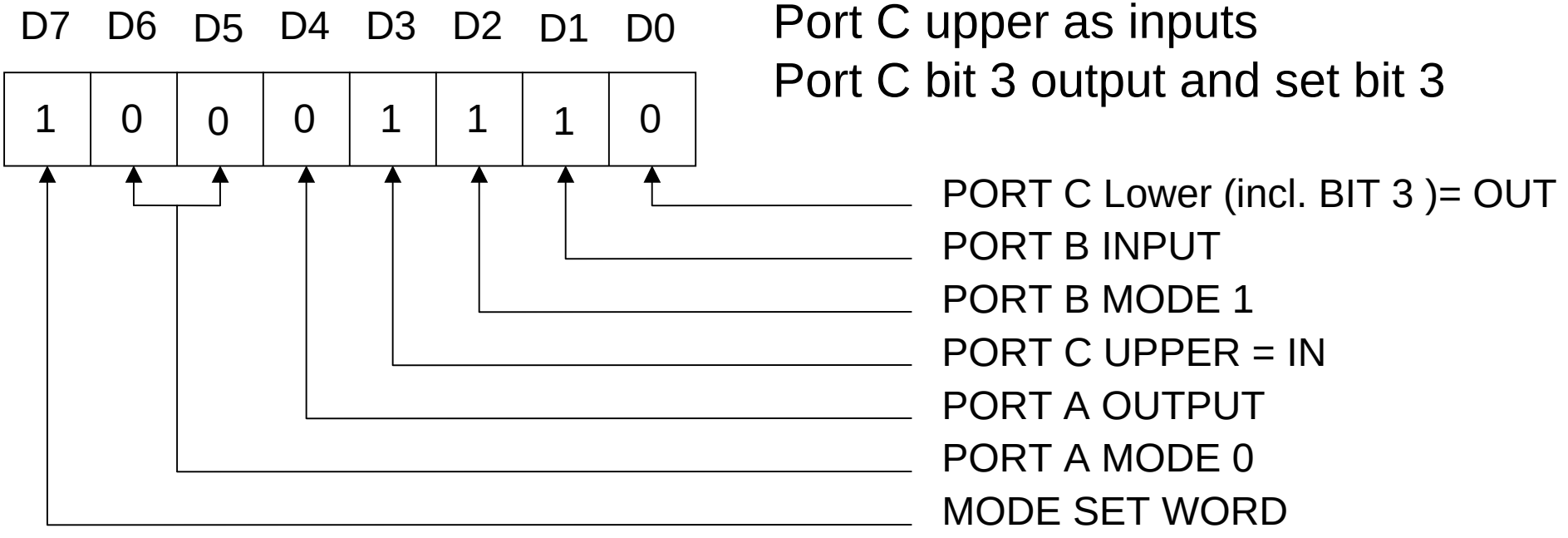
8255A Control Word Formats



(b) Port C bit set/reset control word

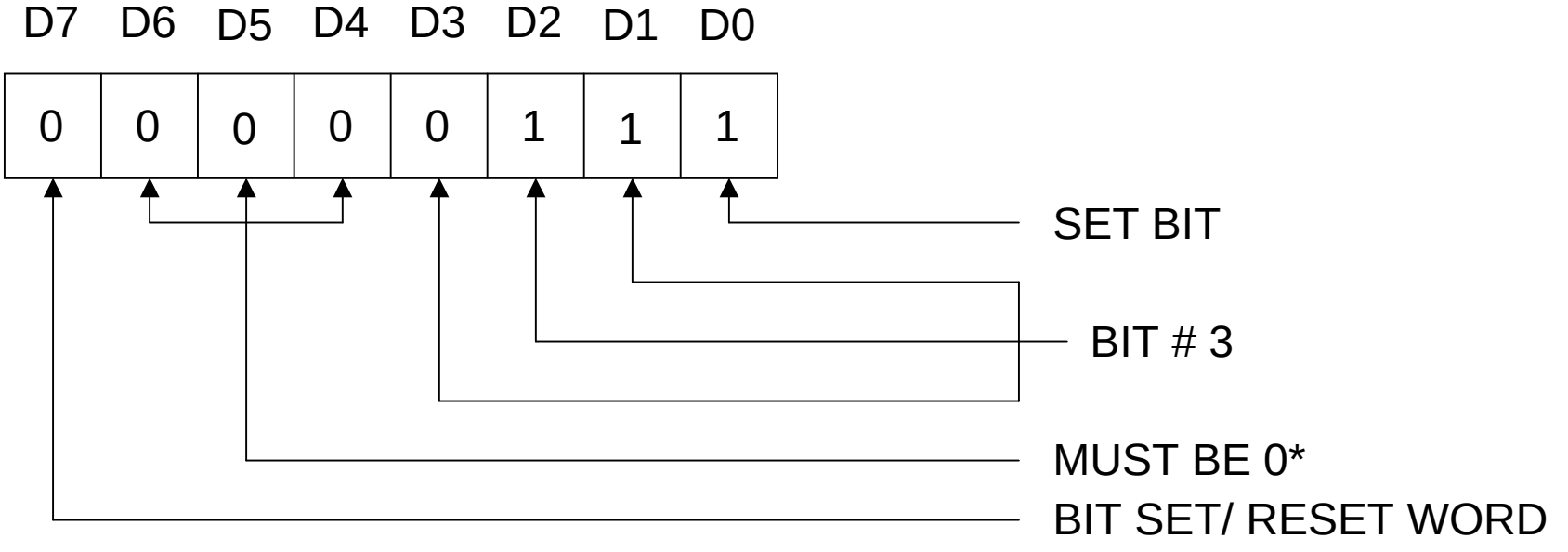
Control Word examples for 8255A

Suppose we want to set
- Port B as mode 1 input
- Port A as mode 0 output
- Port C upper as inputs
- Port C bit 3 output and set bit 3



(a) mode-set control word

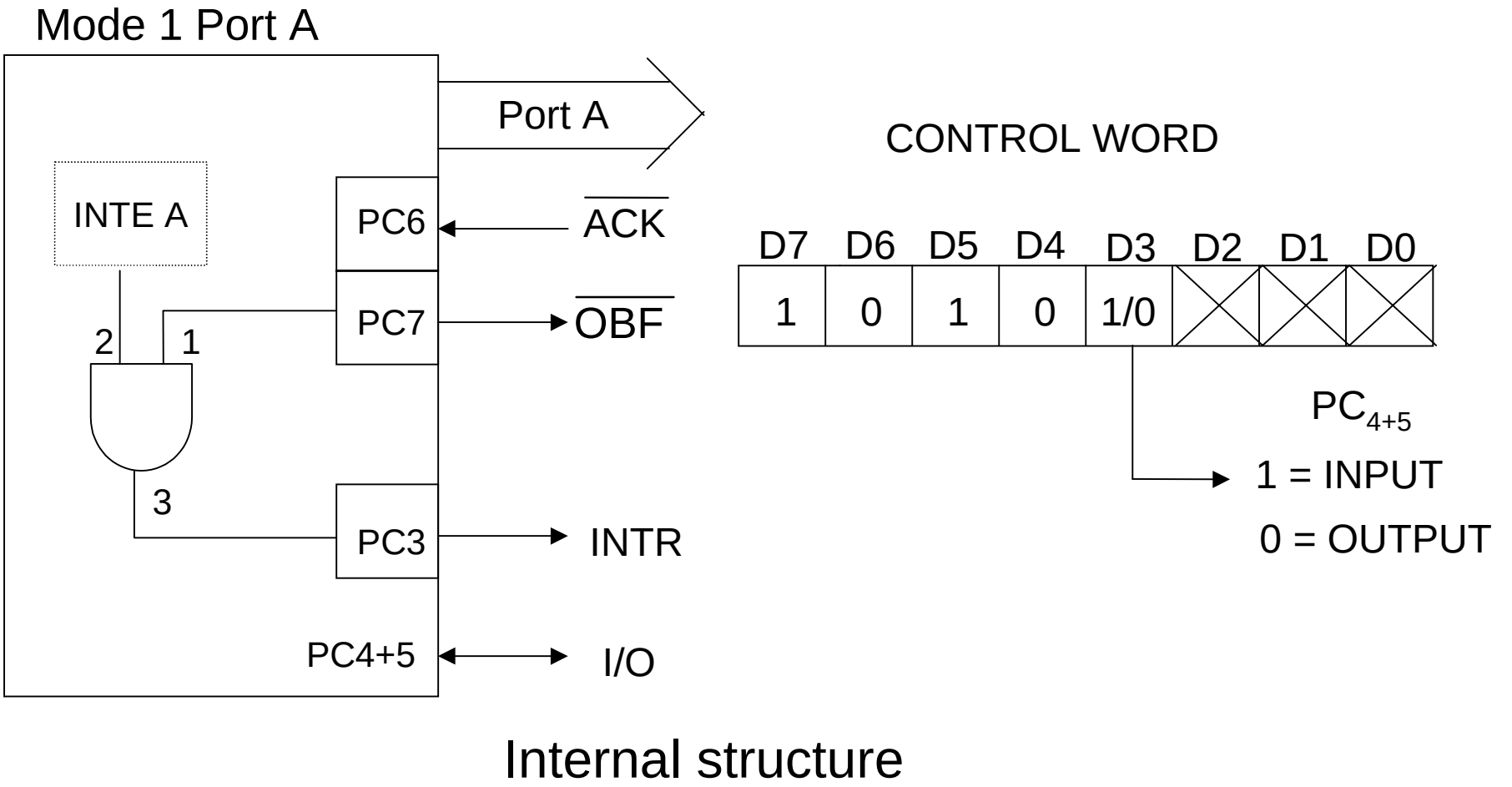
Control Word examples for 8255A (cont.)



*:D6-D4 are set to 0 for simplicity and compatibility with future product.

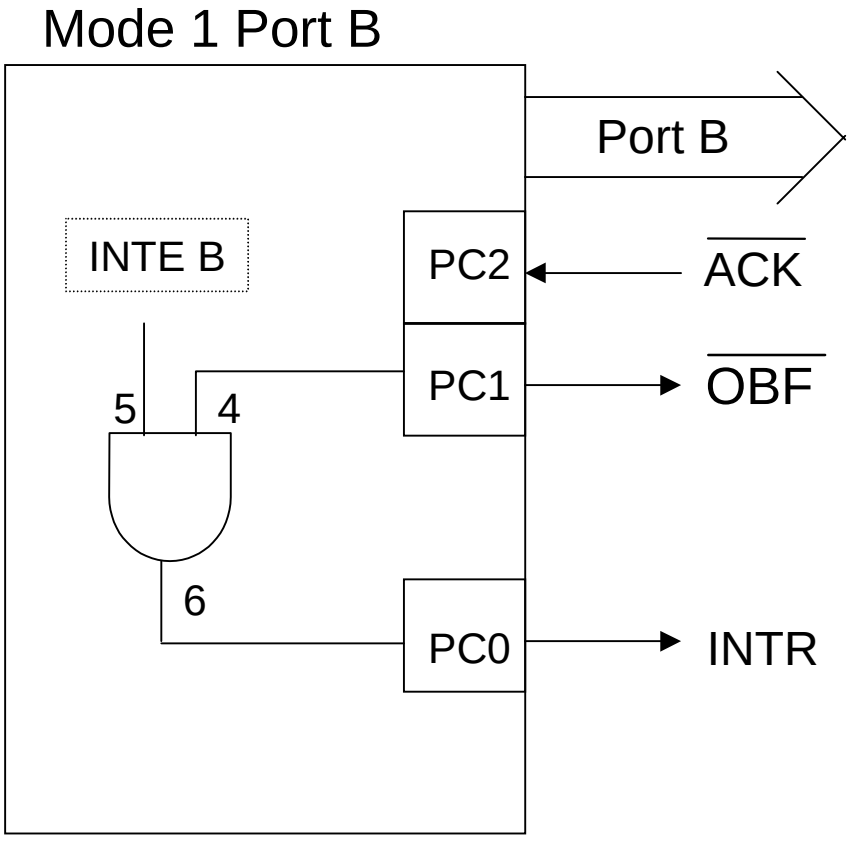
(b) Port C bit set/reset control word to set bit 3

Strobed Output Operation of 8255A (Mode 1 Port A)



Internal structure

Strobed Output Operation (Mode 1) of 8255A



CONTROL WORD

D7	D6	D5	D4	D3	D2	D1	D0
1	X	X	X	X	1	0	X

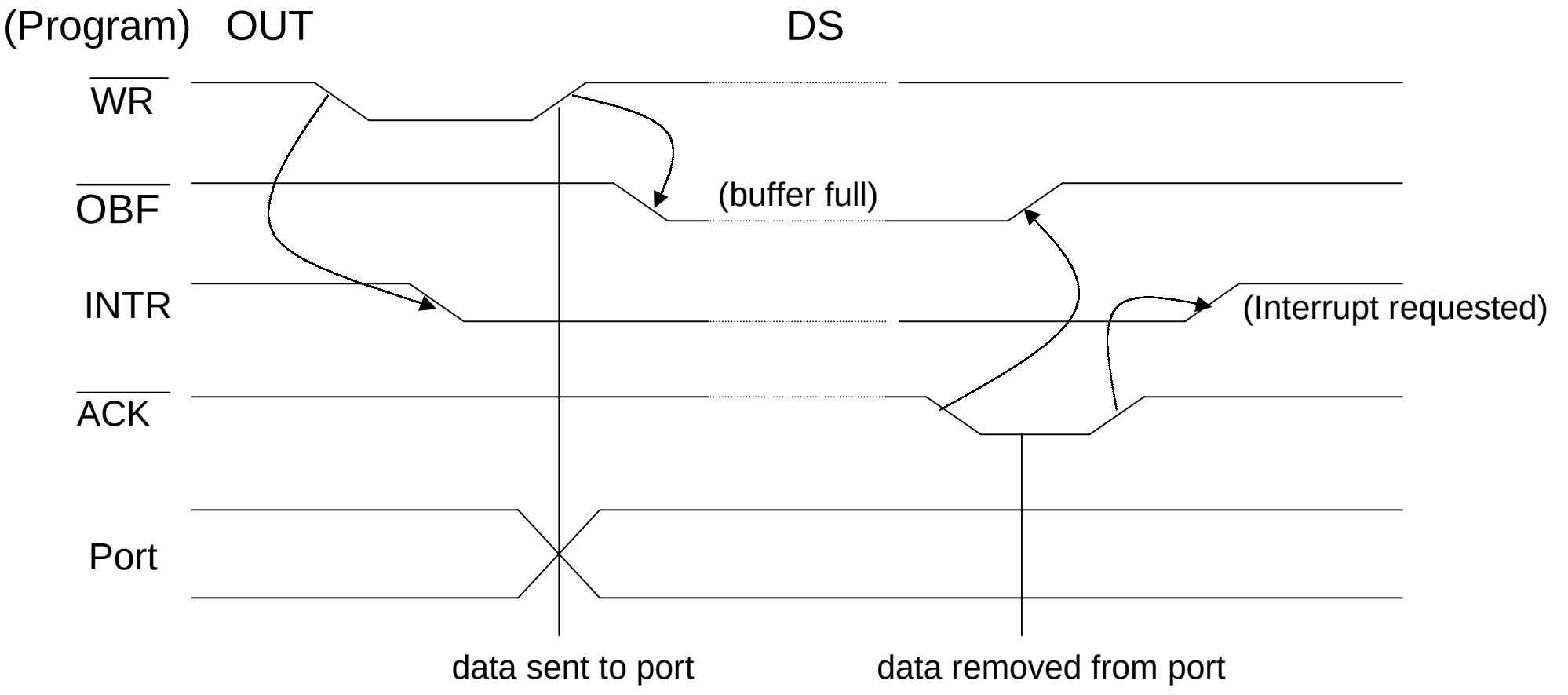
Internal structure

Signal Definition of Mode 1 Strobed Output

- Whenever data are written to a port programmed as a strobed output port, the $\overline{OBF}$ signal becomes a logic 0 to indicate that data are present in the port latch and is ready for external device to access. The external device strobes the $\overline{ACK}$ signal to indicate it has received the data. The $\overline{ACK}$ returns the $\overline{OBF}$ to logic 1.

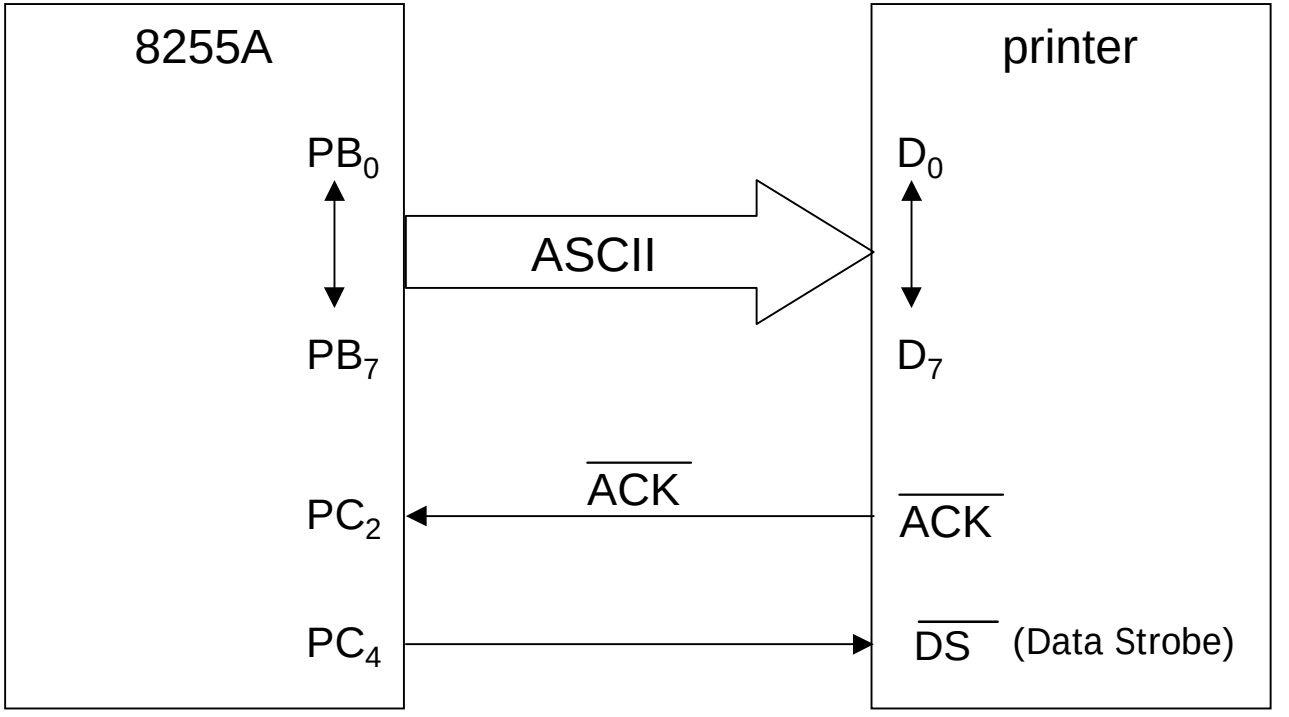
- $\overline{OBF}$ an output that goes low whenever data are output (OUT) to the port A or port B latch
- $\overline{ACK}$ an Acknowledge signal that cause $\overline{OBF}$ pin to return to a logic 1. This signal is a response from external device to indicate it has received data from 8255 port.
- INTR a signal that often interrupts the processor when the external device receives the data and sends back the $\overline{ACK}$ signal.
- INTE an internal bit programmed to enable or disable the INTR pin. INTE A is programmed as PC6 (for output mode) or PC4 (for input mode) and INTEB is PC2 (for both input/output mode).

Strobed Output Operation (Mode 1) of 8255A



Timing diagram

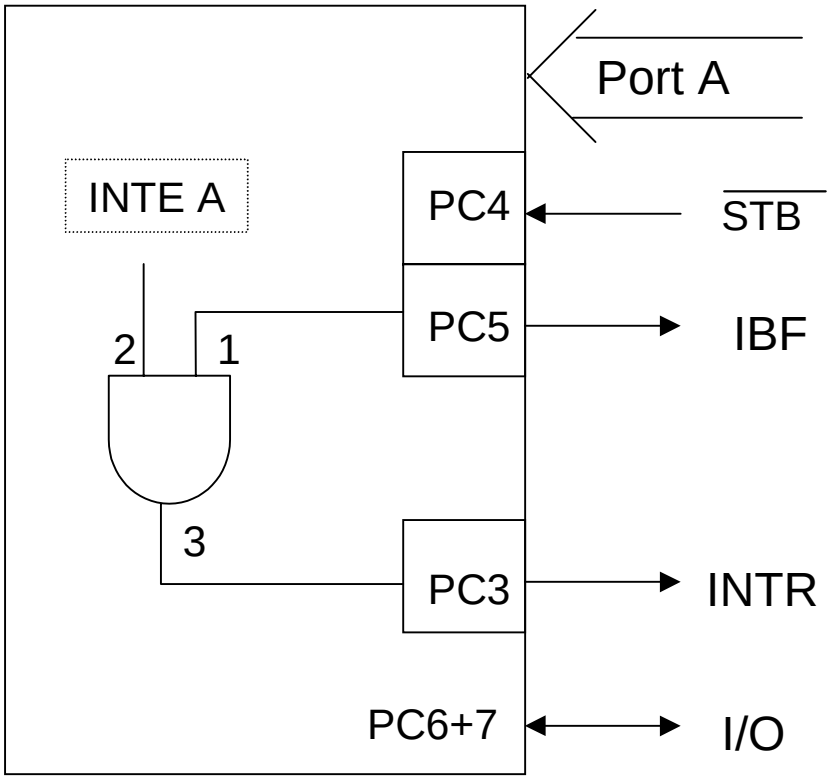
Example of Strobed Output Mode of Operation for 8255A



8255A connected to a parallel printer interface

Strobed Input Operation (Mode 1) of 8255A

Mode 1 Port A



CONTROL WORD

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	1	1/0	X	X	X

PC_{6+7}

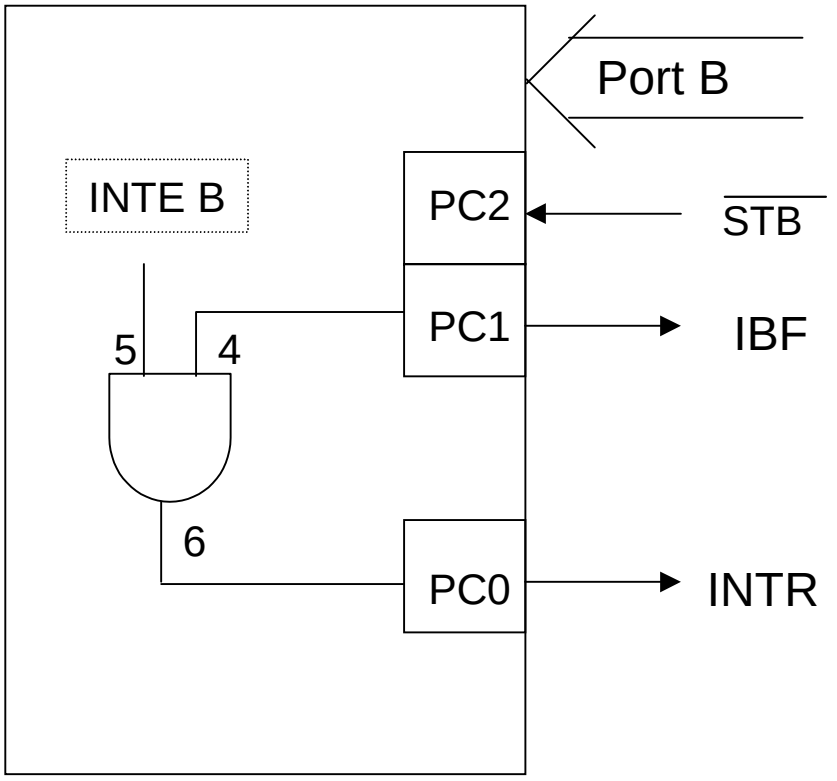
1 = INPUT

0 = OUTPUT

Internal Structure

Strobed Input Operation (Mode 1) of 8255A

Mode 1 Port B

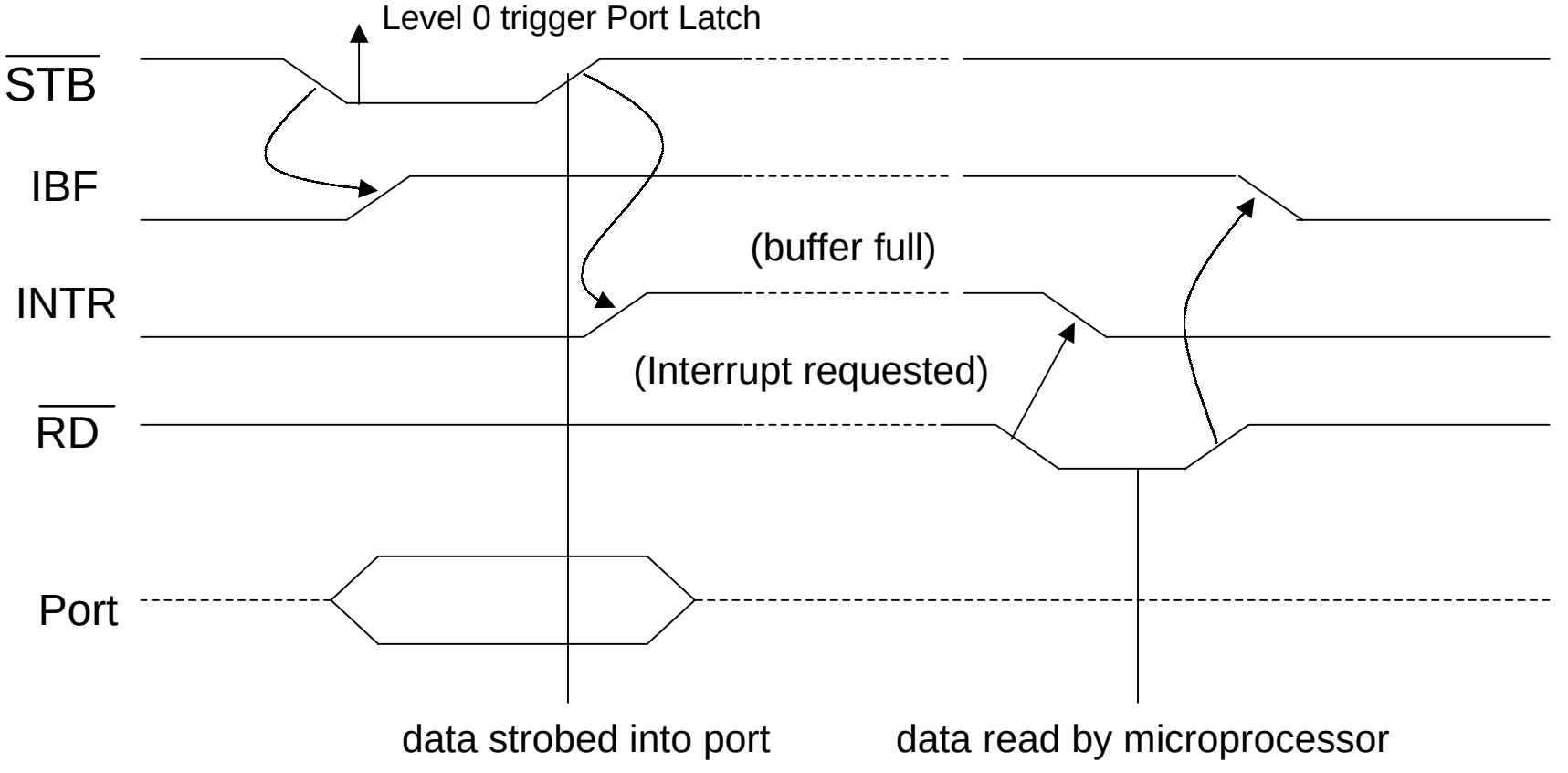


CONTROL WORD

D7	D6	D5	D4	D3	D2	D1	D0
1	X	X	X	X	1	1	X

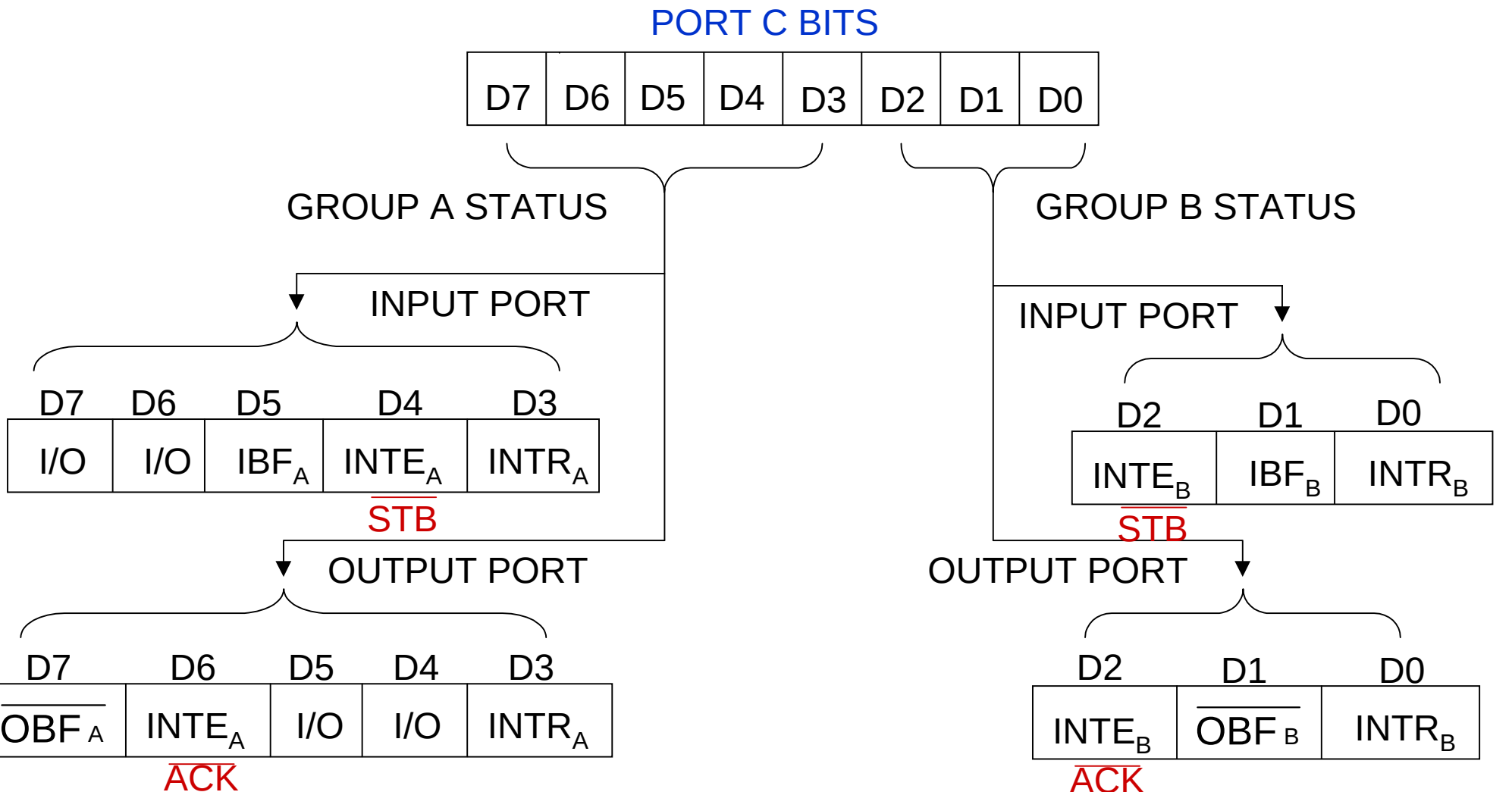
Internal Structure

Strobed Input Operation (Mode 1) of 8255A



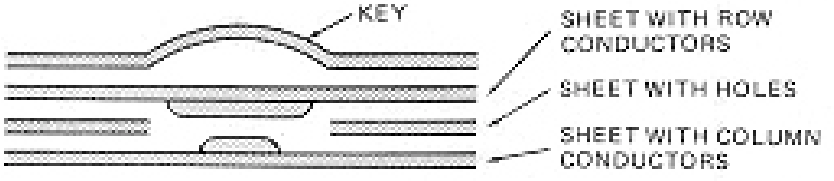
Timing diagram

8255A Status Word Format for Mode 1 Input and Output

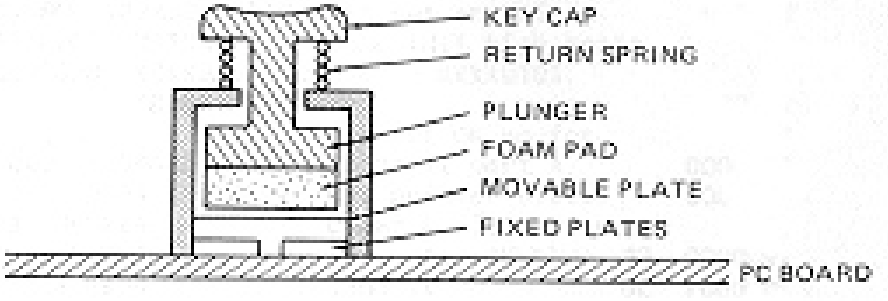


Interfacing a Microprocessor to Keyboard

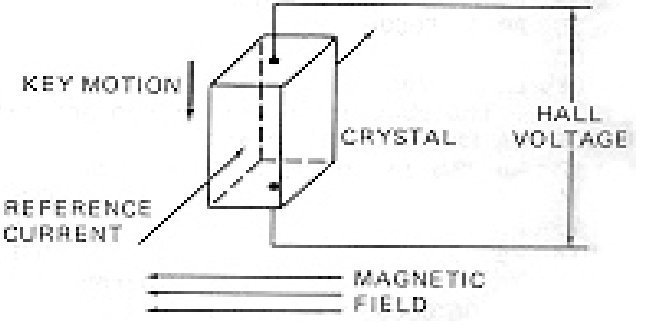
Keyswitch Types



(a) Membrane



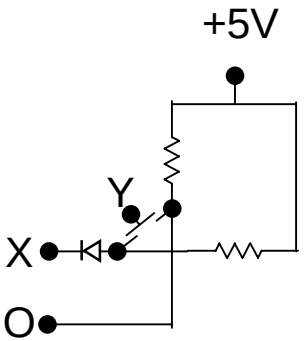
(b) Capacitive
20M key strokes



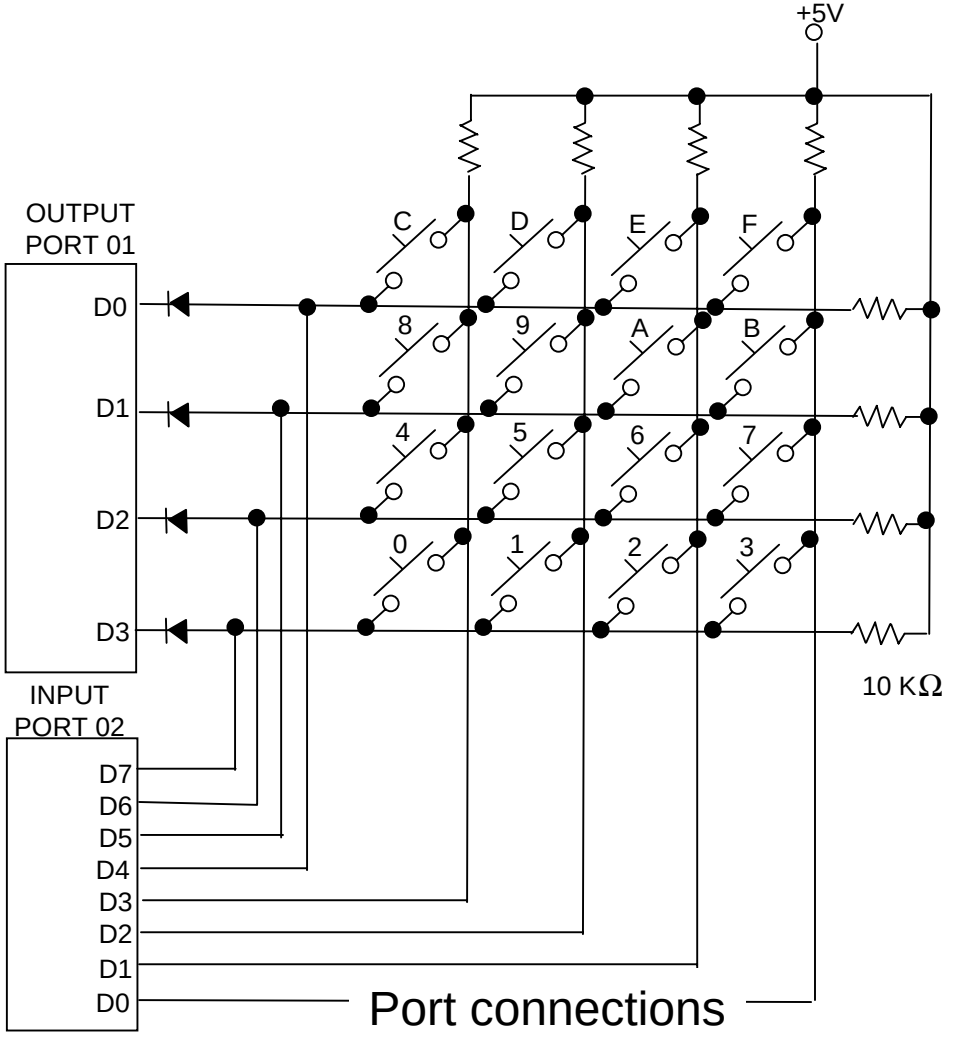
(c) Hall Effect
(100M key strokes)

(D. Hall Fig 9-18)

Detecting a Matrix Keyboard Keypress



X \ Y	open	close
high	high	high
low	high	low



Q:

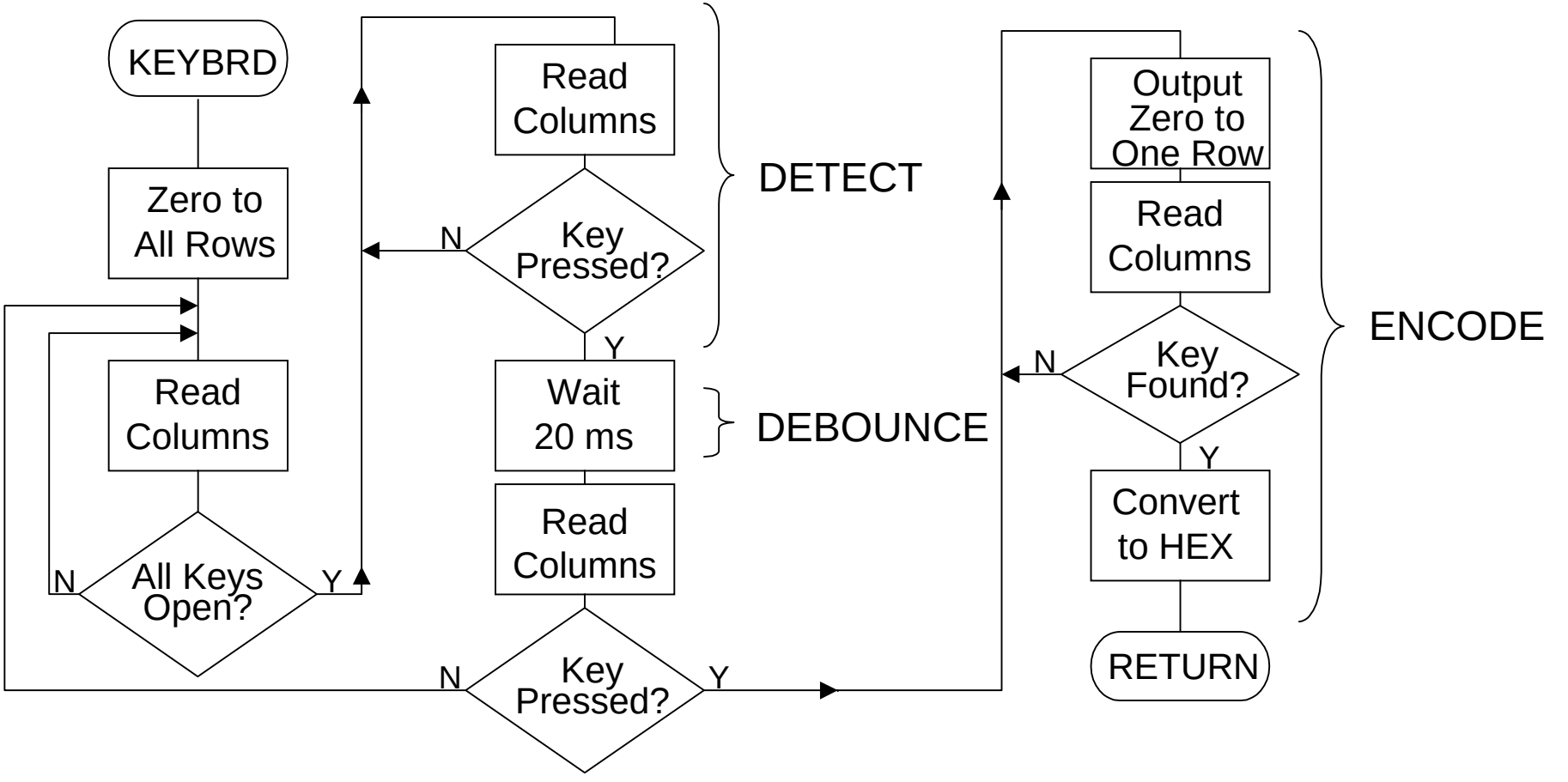
Assume key “9” is pressed. For the following cases, what are the values of D3-D0 at Port 2?

(1). Port 1, D3-D0=0000

(2). Port 1, D3-D0=1110

(3). Port 1, D3-D0=1101

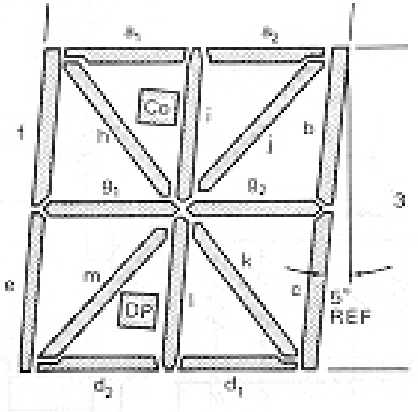
Detecting a Matrix Keyboard Keypress



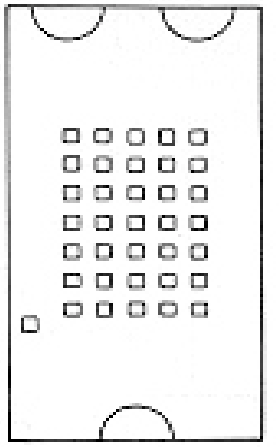
Flowchart for procedure

(D. Hall)

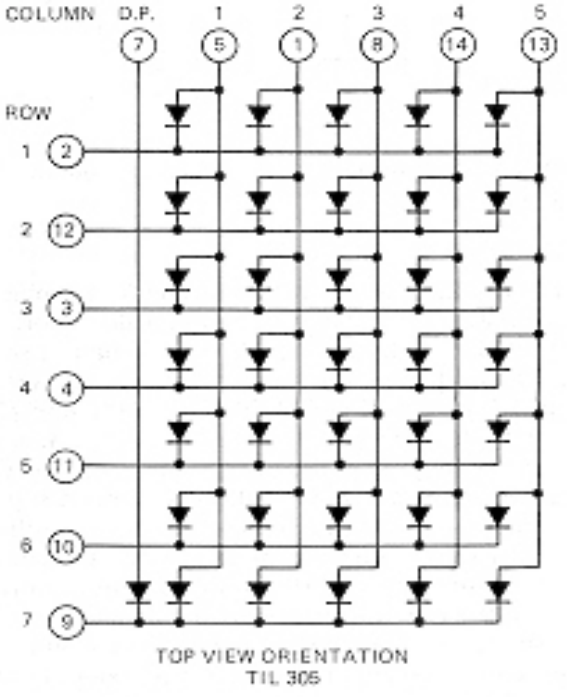
Interfacing to Alphanumeric Display



(a) 18-segment display



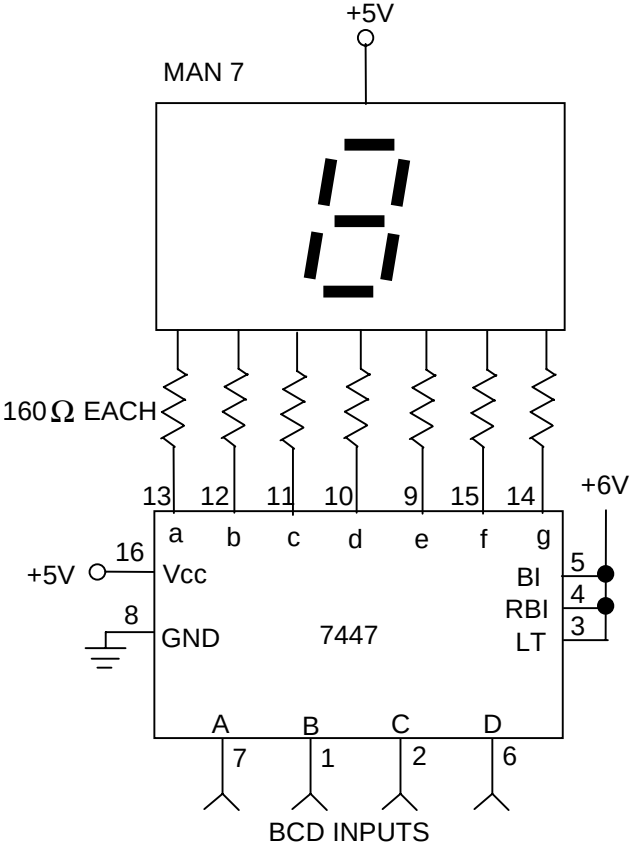
(b) 5 by 7 dot matrix display format



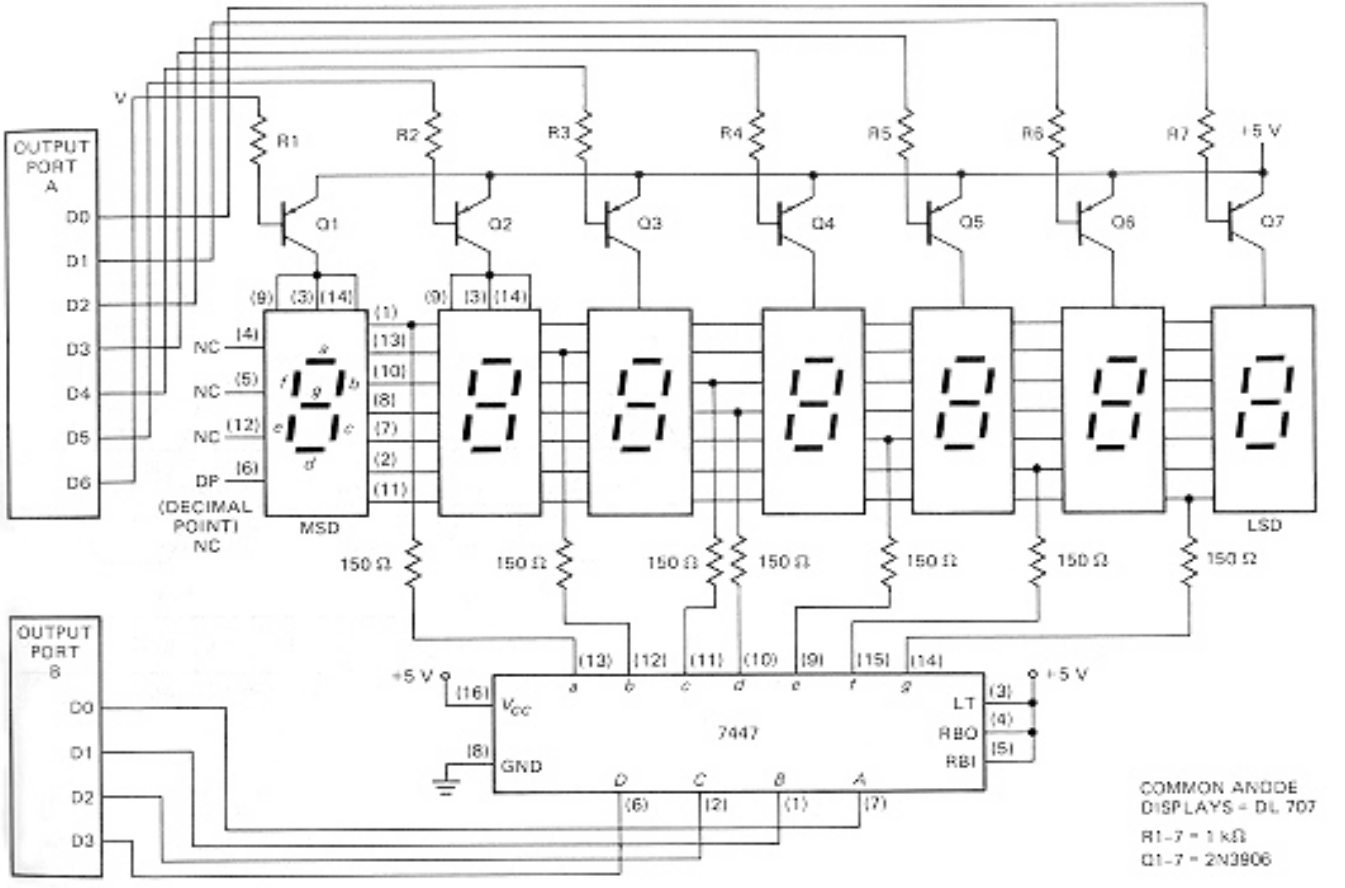
(c) 5 by 7 dot matrix circuit connections

(D Hall Fig 9-23)

Circuit Driving a 7-Segment LED with 7447



Multiplexing 7-segment Displays with a Microcomputer



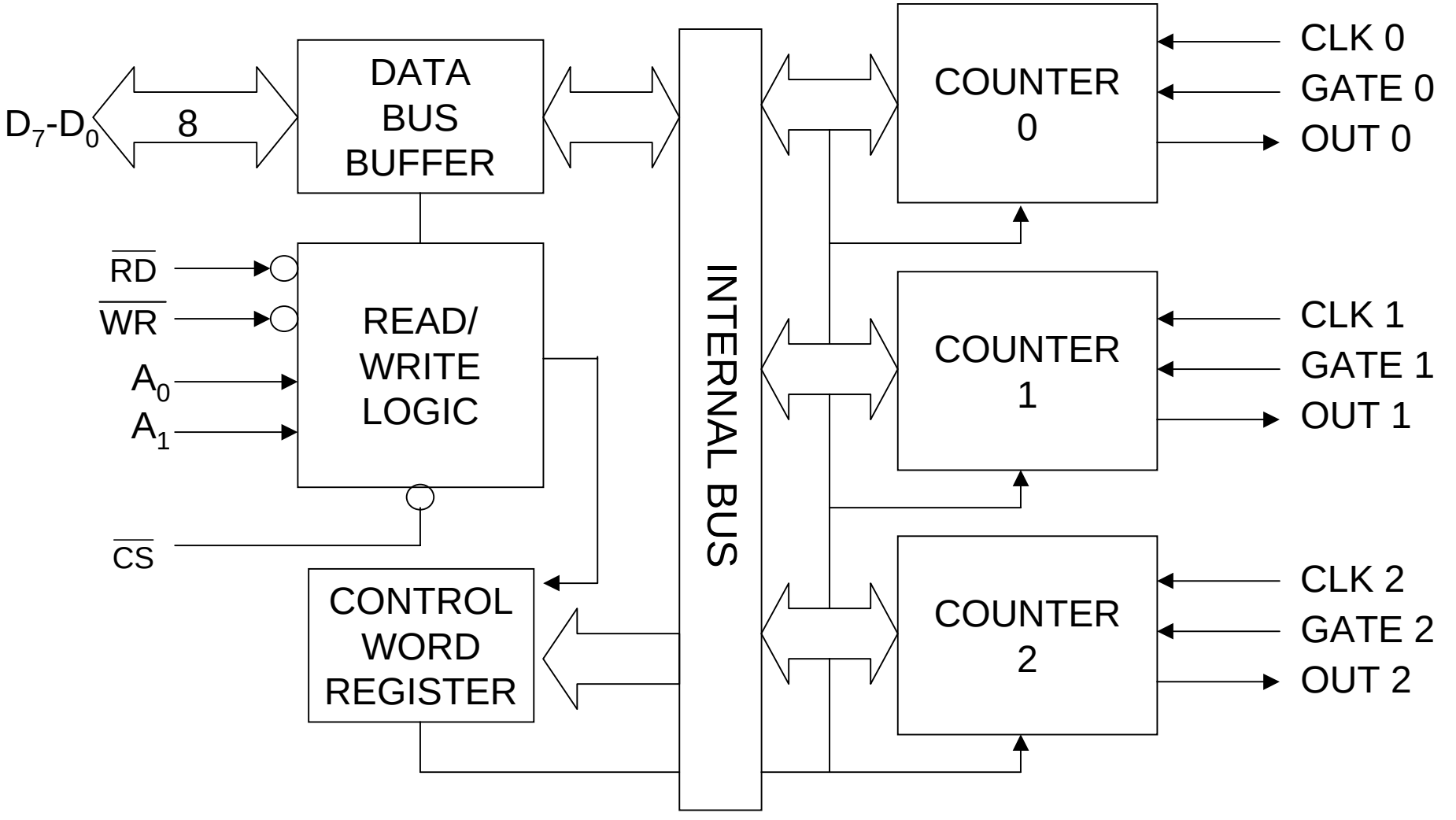
Q: Only one 7447 for all 7 digits. → display the same values for all?

A: multiplex method

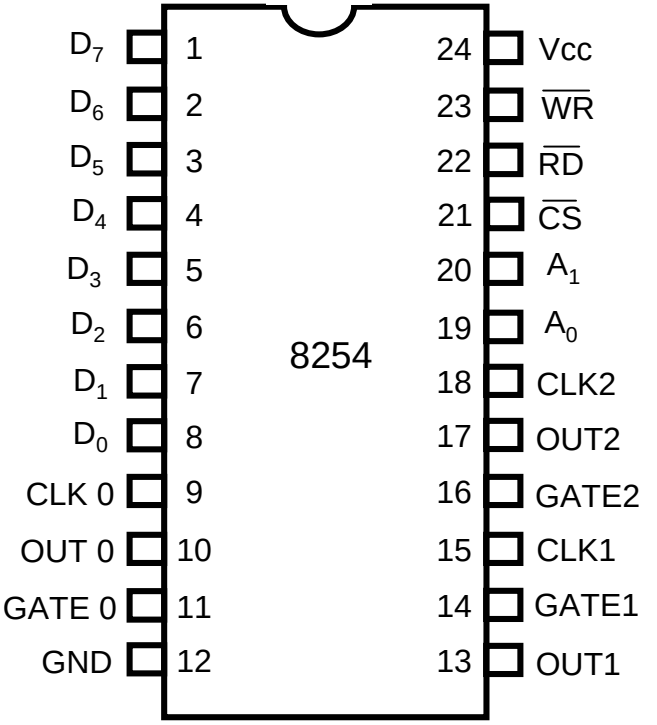
8254 Software-Programmable Timer/Counter

- 8254 is very versatile and can be used in many applications.
- There are several modes of operation for different applications.
- Intel 8253 and 8254 are almost pin-to-pin compatible except
  - The maximum input clock frequency for 8253 and 8254 is 2.6 MHz and 8 MHz, respectively. (10MHz for 8254-2)
  - 8254 has a read-back feature which allows you to latch the count in all the counters and the status of the counter at any point. 8253 does not have this feature.
- 8254 contains three 16-bit counters. The counter can be programmed to load the initial count, start and stop the count.
- 8254 has an 8-bit interface to data bus, and two address input A_0 and A_1 to address each of the three counters.

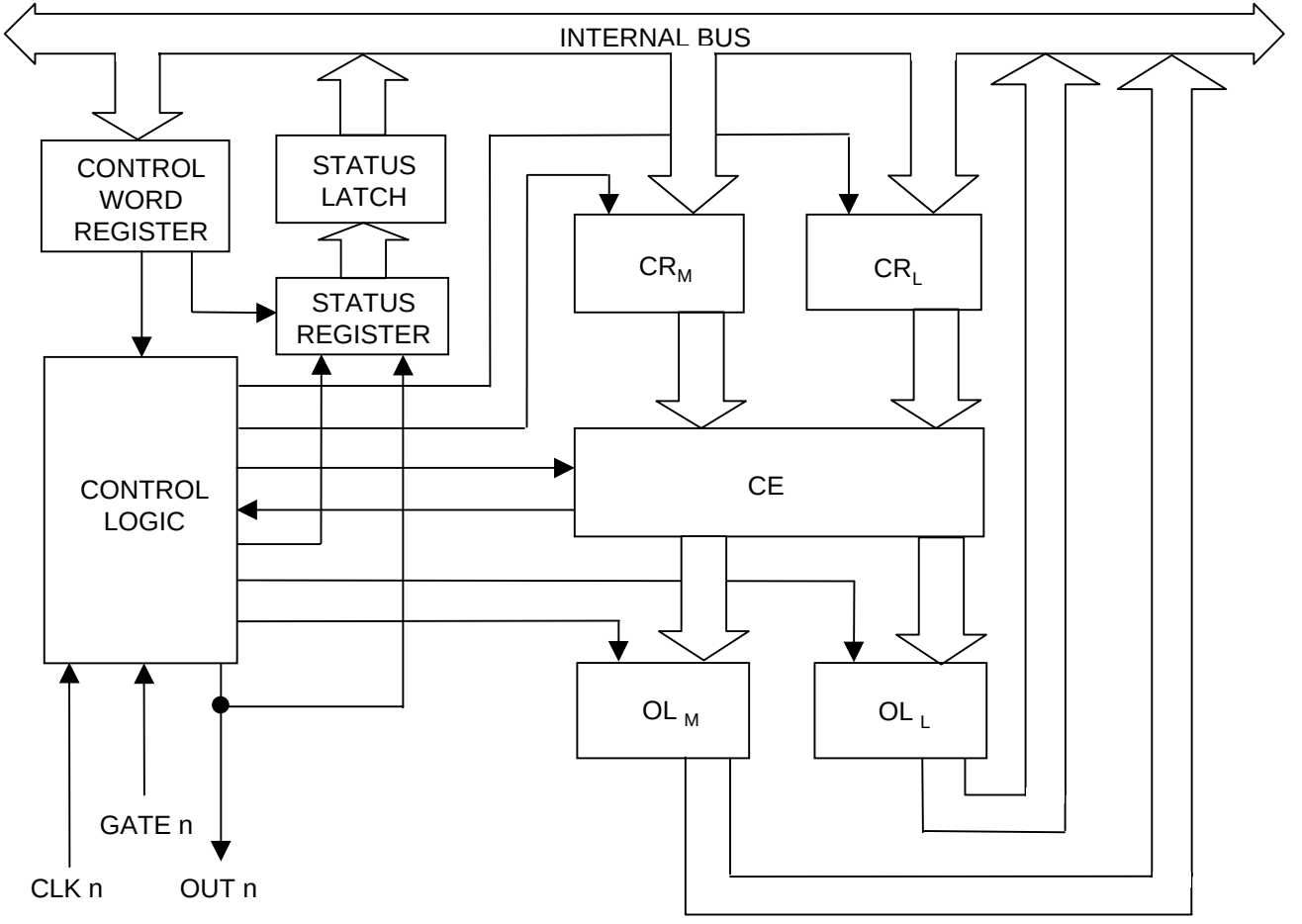
8254 Block Diagram



8254 Pin Configuration



Internal Block Diagram of a Counter



Initializing 8254

- When power on, programmable peripheral devices such as 8254 are usually in *undefined* state. → need initialization
- Initialization steps:
  1. Determine the base address of the device from the address decode circuitry or the address decoder truth table.
  2. Determine the internal address for each 8254 internal device (control register, port, counters, status register, etc.)
  3. Add each of the internal address to the system base address to determine the system address of each device.
  4. Look in data sheet for the device for the format of the control word(s) that you have to send to the device to initialize it.
  5. Construct the control word required to initialize the device.
  6. Send the control word to the device. In case of the 8254, you need to send the starting count to each of the counter registers.

8254 Addresses

Internal

A1	A0	SELECTS
0	0	COUNTER 0
0	1	COUNTER 1
1	0	COUNRER 2
1	1	CONTROL WORD REGISTER

System*

SYSTEM ADDRESS	8254 PART
F F 0 1	COUNTER 0 Register
F F 0 3	COUNTER 1 Register
F F 0 5	COUNTER 2 Register
F F 0 7	CONTROL Register

Note : This system address is circuit-dependent.
In this example, 8254's A1 and A0 are connected to CPU's pin A2 and A1.

8254 Control Word Format

$A_1A_0 = 11$ $\overline{CS} = 0$ $\overline{WR} = 0$ $\overline{RD} = 1$

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

(1) SC - Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

8254 Control Word Format (cont.)

(2) RW - Read/ Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first then most significant byte

(4) BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

8254 Control Word Format (cont.)

(3) M - Mode

M2	M1	M0	
0	0	0	Mode 0 - Interrupt on Terminal Count
0	0	1	Mode 1 - Hardware One-Shot
×	1	0	Mode 2 - Pulse Generator
×	1	1	Mode 3 - Square Wave Generator
1	0	0	Mode 4 - Software Triggered Strobe
1	0	1	Mode 5 - Hardware Triggered Strobe

NOTE:

Don’t Care bits (×) should be 0 to insure compatibility with future Intel products

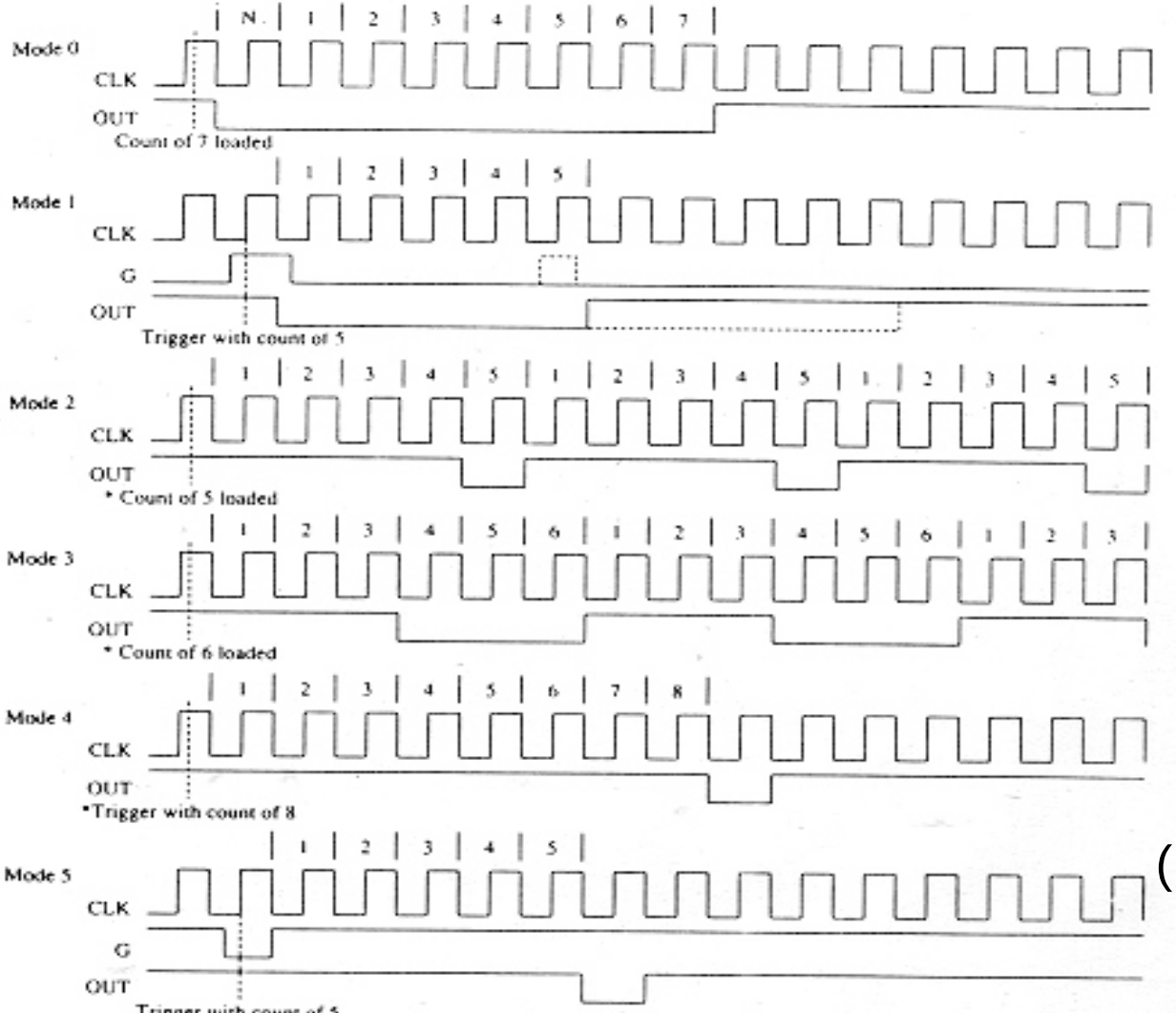
Example of a control word

Task: Use counter 0 of the 8254 to divide a clock signal at 2.45 MHz to 78.6 kHz (1/32).

```
MOV   AL, 00010111B     ; Control word for counter 0
                        ; Read/write LSB only, mode 3, BCD countdown
                        ; 00   01   011   1
                        ;                 BCD countdown
                        ;            Mode 3
                        ;       R/W LSB only
                        ;  Select counter 0
MOV   DX, 0FF07H        ; Point at 8254 control register
OUT   DX, AL            ; Send control word
MOV   AL, 32H           ; Load lower byte of count
MOV   DX, 0FF01H        ; Point to counter 0 count register
OUT   DX, AL            ; Send count to count register
```

(see page 54)

Six Modes of Operation for 8254 Programmable Interval Timer



(Brey Fig10-35)

8254 Mode Operation

- **Mode 0:** Allows 8254 counter to be used as an event counter. The output becomes a logic 0 when the control word is written and remains until N plus the number of programmed counts.

- **Mode 1:** Causes the counter to function as a retriggerable monostable multivibrator (one shot). In this mode, the G input triggers the counter so that it develops a pulse at the OUT connection that becomes a logic 0 for the duration of the counter. If the count is 10, then the OUT connection goes low for 10 clock period when triggered. If the G input occurs within the duration of the output pulse, the counter is again *reloaded* with the count and the OUT connection continues for the total length of the count.

- **Mode 2:** Allows the counter to generate a series of continuous pules that are one clock pulse in width. The separation between pulses is determined by the count. (periodic pulse generator)

8254 Mode Operation (cont.)

- **Mode 3:** Generates a continuous *square-wave* at the OUT connection, provided the G pin is a logic 1. If the count is even, the output is high for one-half of the count and low for *one-half* of the count. If the count is *odd*, the output is high for one clocking period longer than it is low.

- **Mode 4:** Allows the counter to produce a single pulse at the output. If the count is 10, the output is high for 10 clocking period and then low for 1 clocking period. (software-triggered strobe) (Writing the count to the counter register starts the count.)

- **Mode 5:** A hardware triggered one-shot that functions as mode 4 except that it is started by a trigger pulse on the G pin instead of by software. This mode is also similar to mode 1 because it is retriggerable. (hardware-triggered strobe) (Count will be transferred to counter register after trigger goes high.)