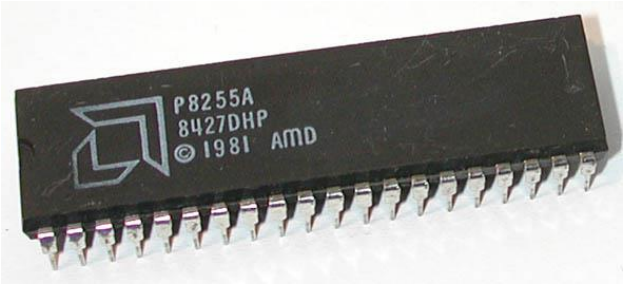
Programmable Peripheral Interface (PPI) – 8255A

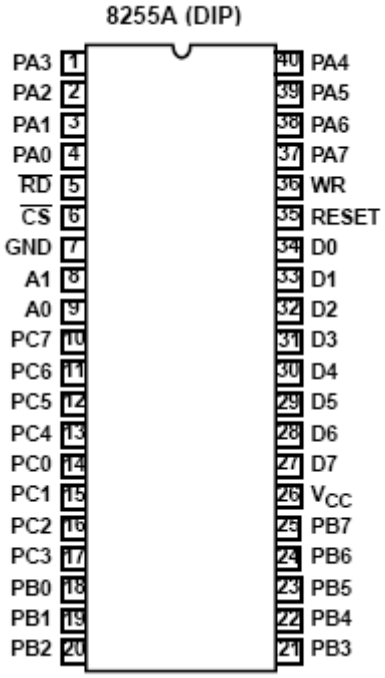
CEN433

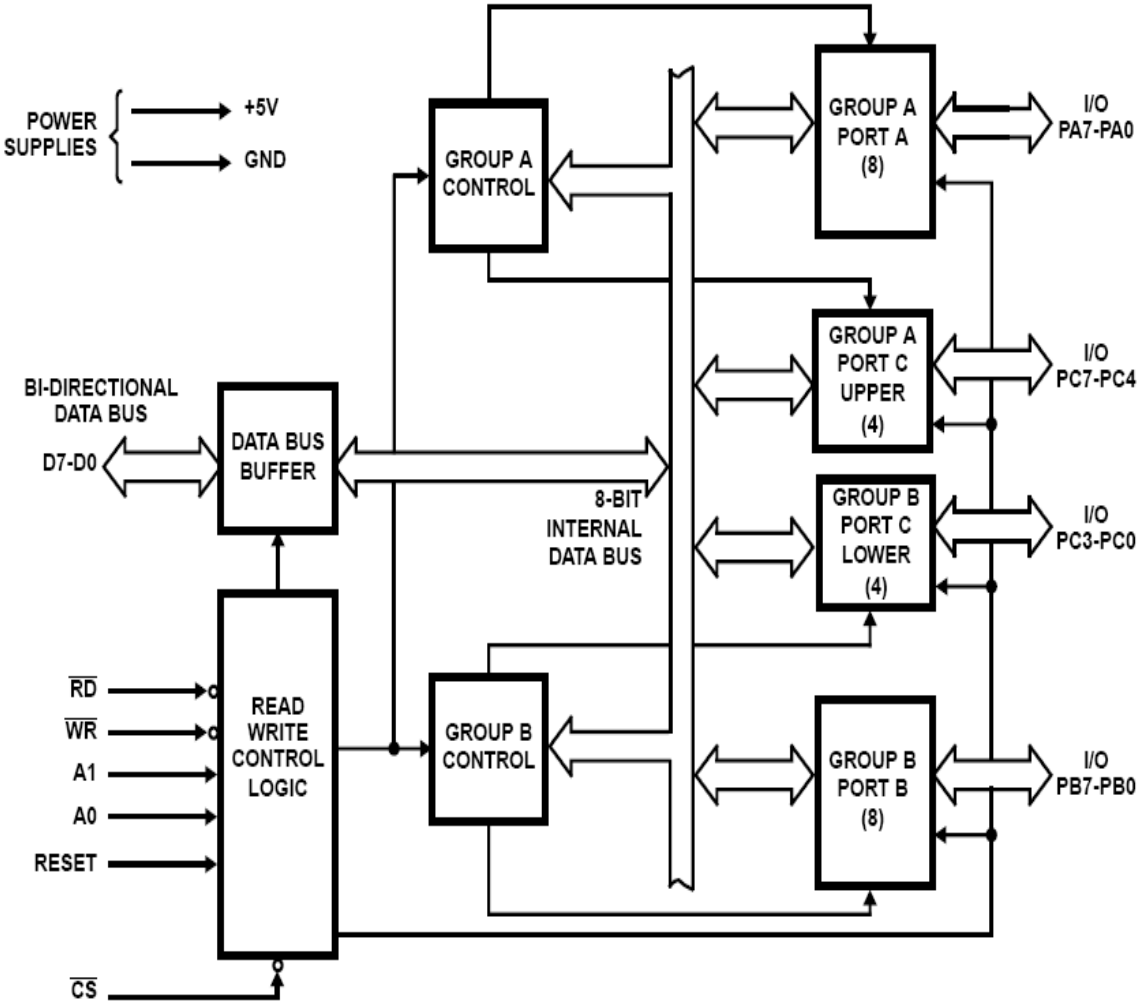
King Saud University

Dr. Mohammed Amer Arafah

Functional Diagram



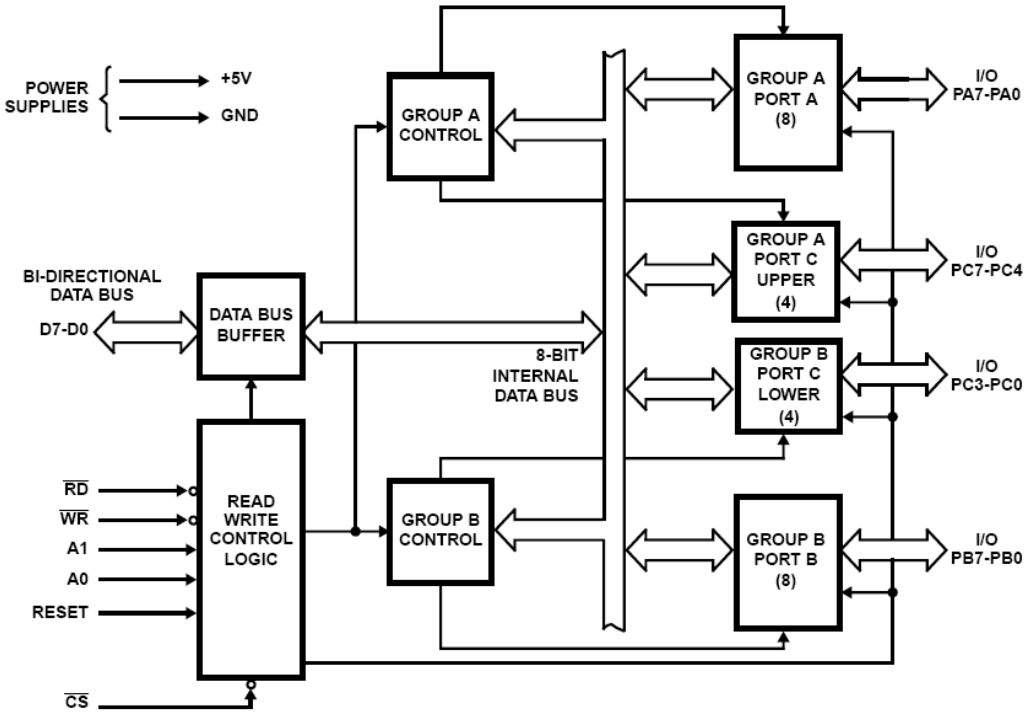




Pin Description

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
V_{CC}	26		V_{CC}: The +5V power supply pin. A 0.1μF capacitor between pins 26 and 7 is recommended for decoupling.
GND	7		GROUND
D0-D7	27-34	I/O	DATA BUS: The Data Bus lines are bidirectional three-state pins connected to the system data bus.
RESET	35	I	RESET: A high on this input clears the control register and all ports (A, B, C) are set to the input mode with the "Bus Hold" circuitry turned on.
$\overline{CS}$	6	I	CHIP SELECT: Chip select is an active low input used to enable the 82C55A onto the Data Bus for CPU communications.
$\overline{RD}$	5	I	READ: Read is an active low input control signal used by the CPU to read status information or data via the data bus.
$\overline{WR}$	36	I	WRITE: Write is an active low input control signal used by the CPU to load control words and data into the 82C55A.
A0-A1	8, 9	I	ADDRESS: These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word register. A0 and A1 are normally connected to the least significant bits of the Address Bus A0, A1.
PA0-PA7	1-4, 37-40	I/O	PORT A: 8-bit input and output port. Both bus hold high and bus hold low circuitry are present on this port.
PB0-PB7	18-25	I/O	PORT B: 8-bit input and output port. Bus hold high circuitry is present on this port.
PC0-PC7	10-17	I/O	PORT C: 8-bit input and output port. Bus hold circuitry is present on this port.

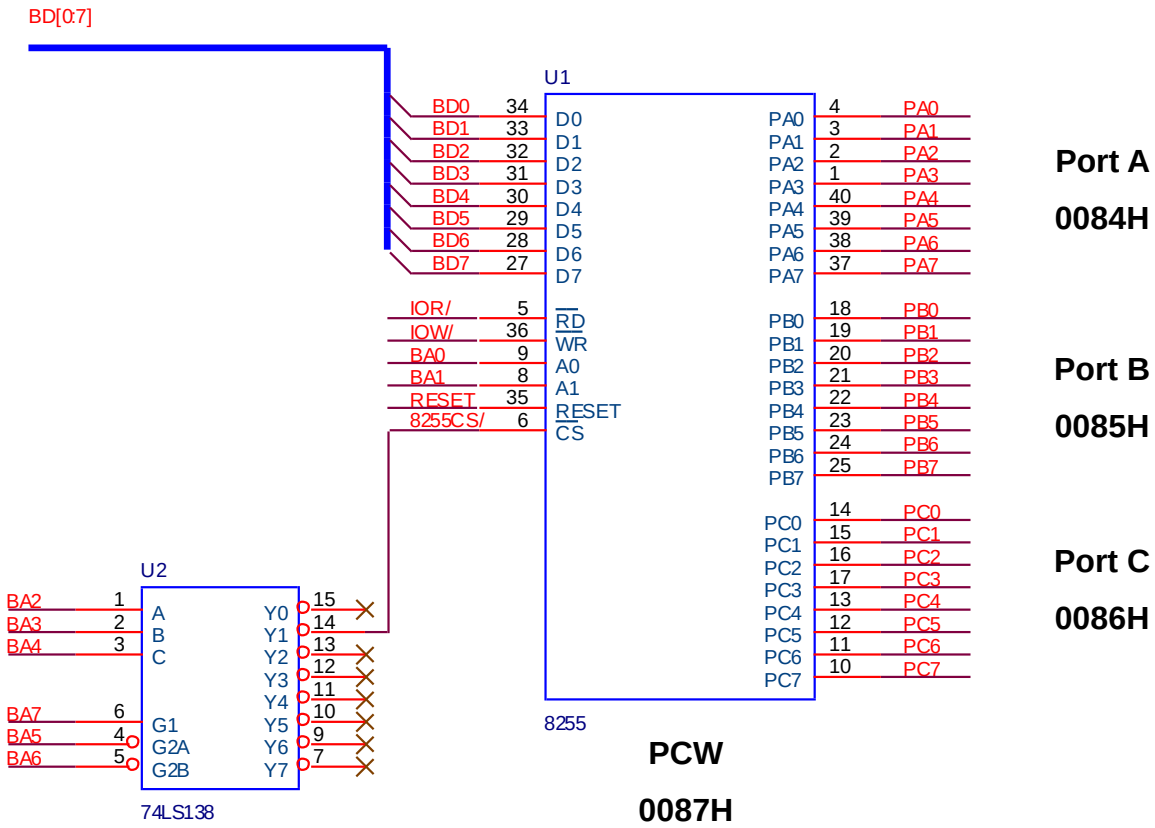
8255A Basic Operation



A_1	A_0	Port
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control Word

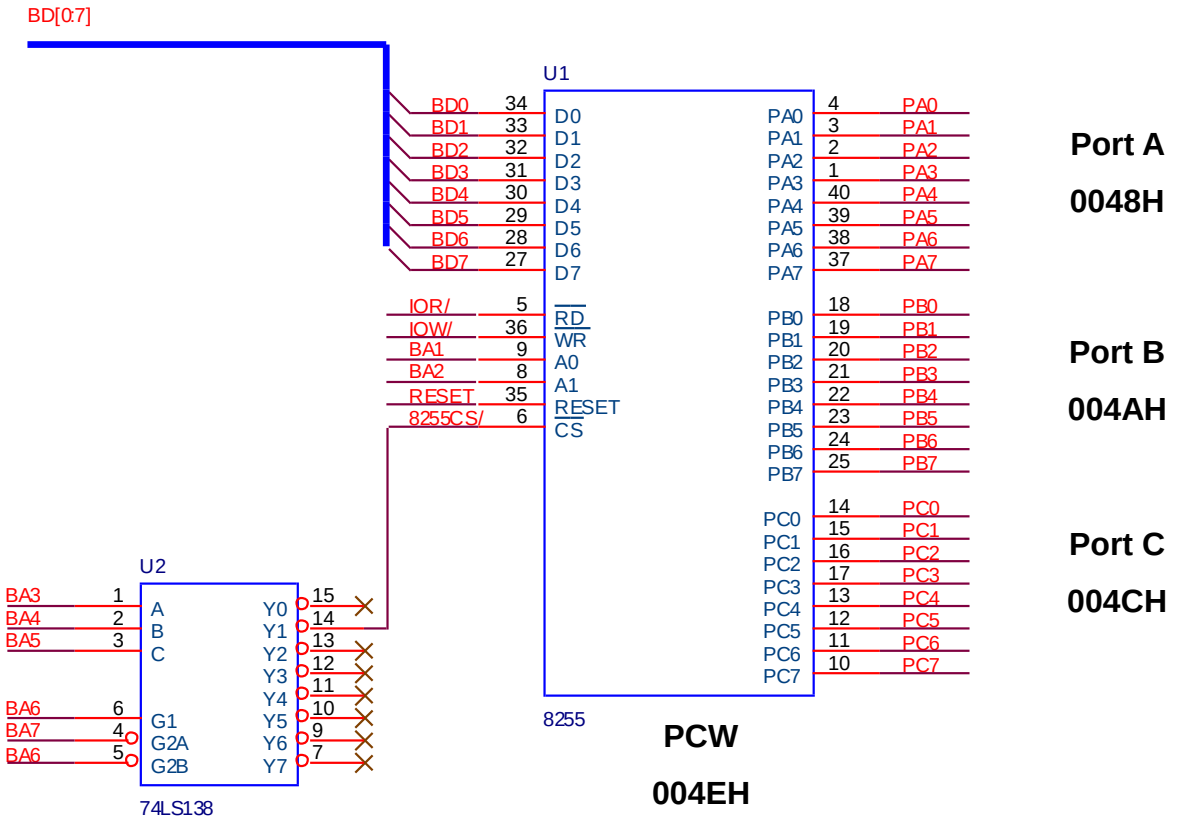
A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					DISABLE FUNCTION
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

Interfacing 8255A to Buffered 8088 System



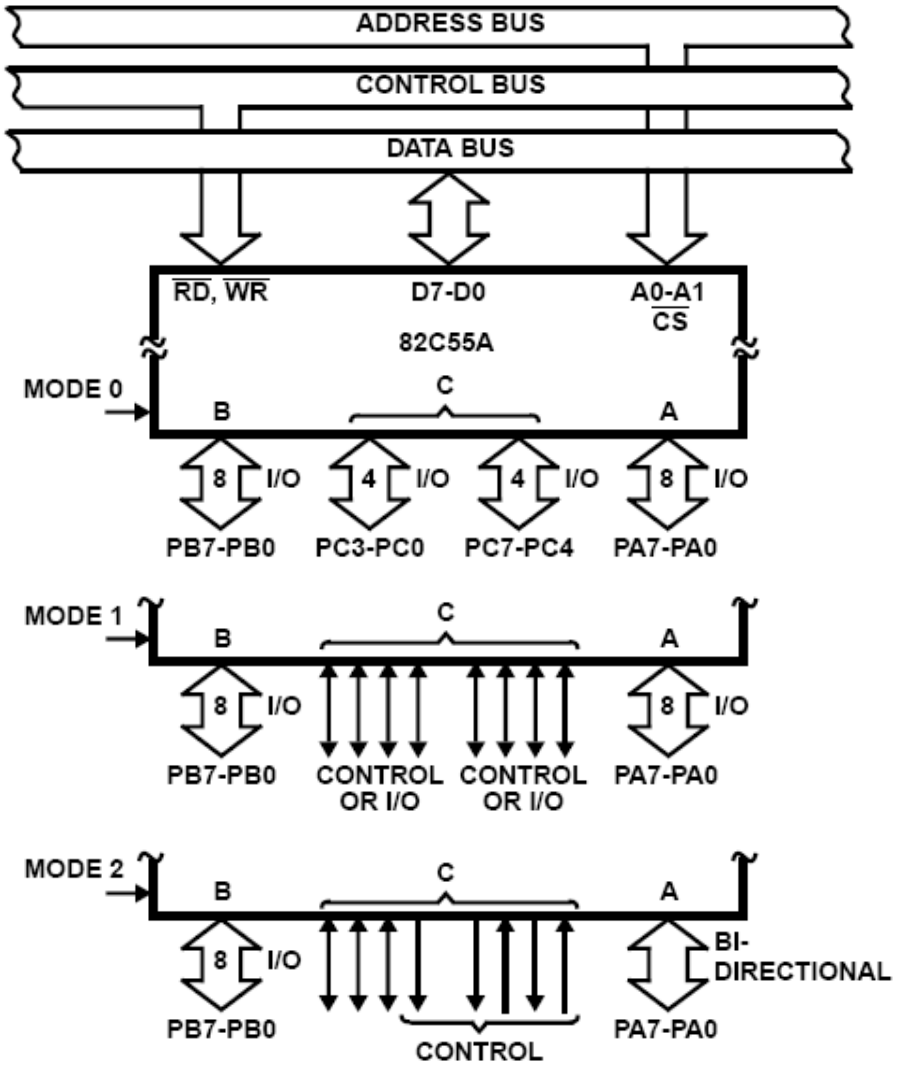
BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
x	x	x	x	x	x	x	x	1	0	0	0	0	1	-	-
								Enable Signals			Y1			Ports	

Interfacing 8255A to Buffered 8086 System

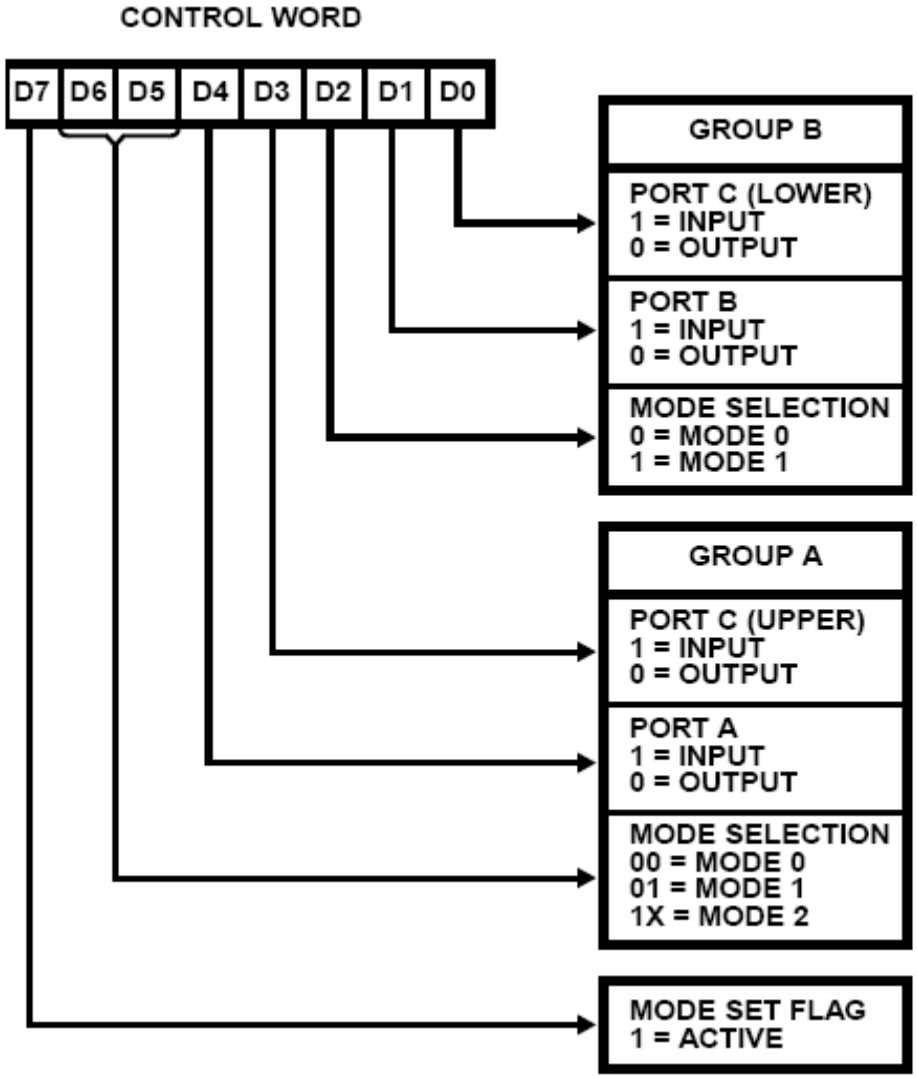


BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
x	x	x	x	x	x	x	x	0	1	0	0	1	-	-	0
								Enable Signals		Y1			Ports		Enable

Mode Definitions and Bus Interface

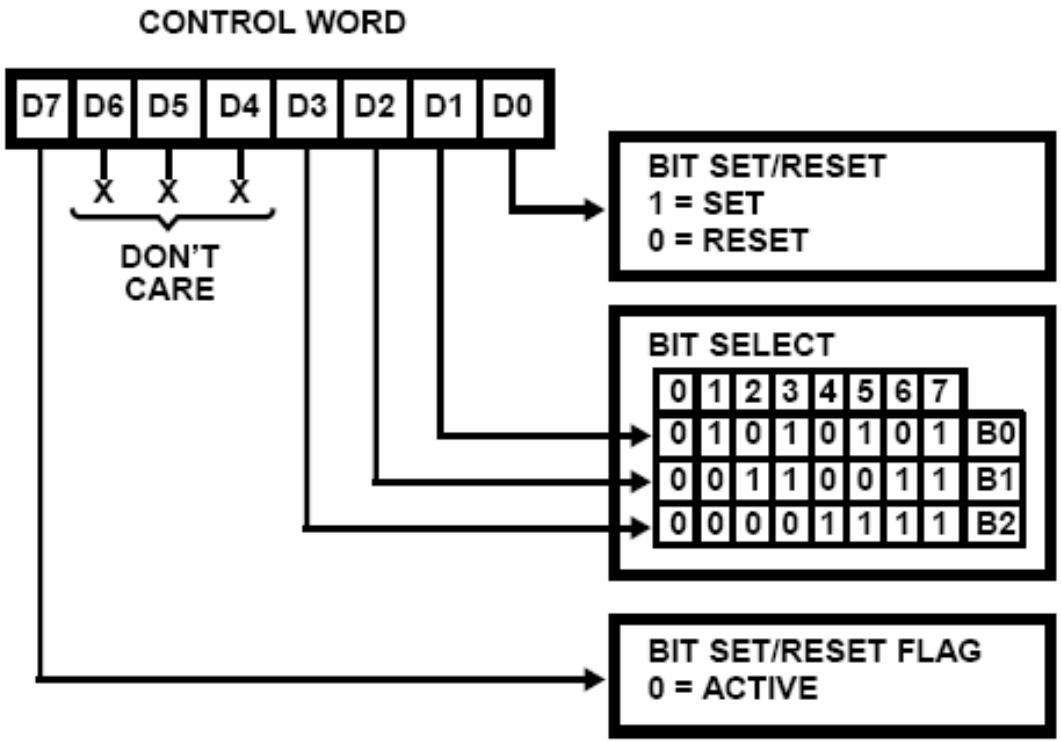


Mode Definition Format




```
; Example
MOV     DX, PCW
MOV     AL, 10011001B
OUT     DX, AL
```

Bit Set/Reset Format




```
; Set PC6
MOV     DX, PCW
MOV     AL, 00001101B
OUT     DX, AL

; Reset PC6
MOV     DX, PCW
MOV     AL, 00001100B
OUT     DX, AL
```

Mode 0 (Basic Input/Output)

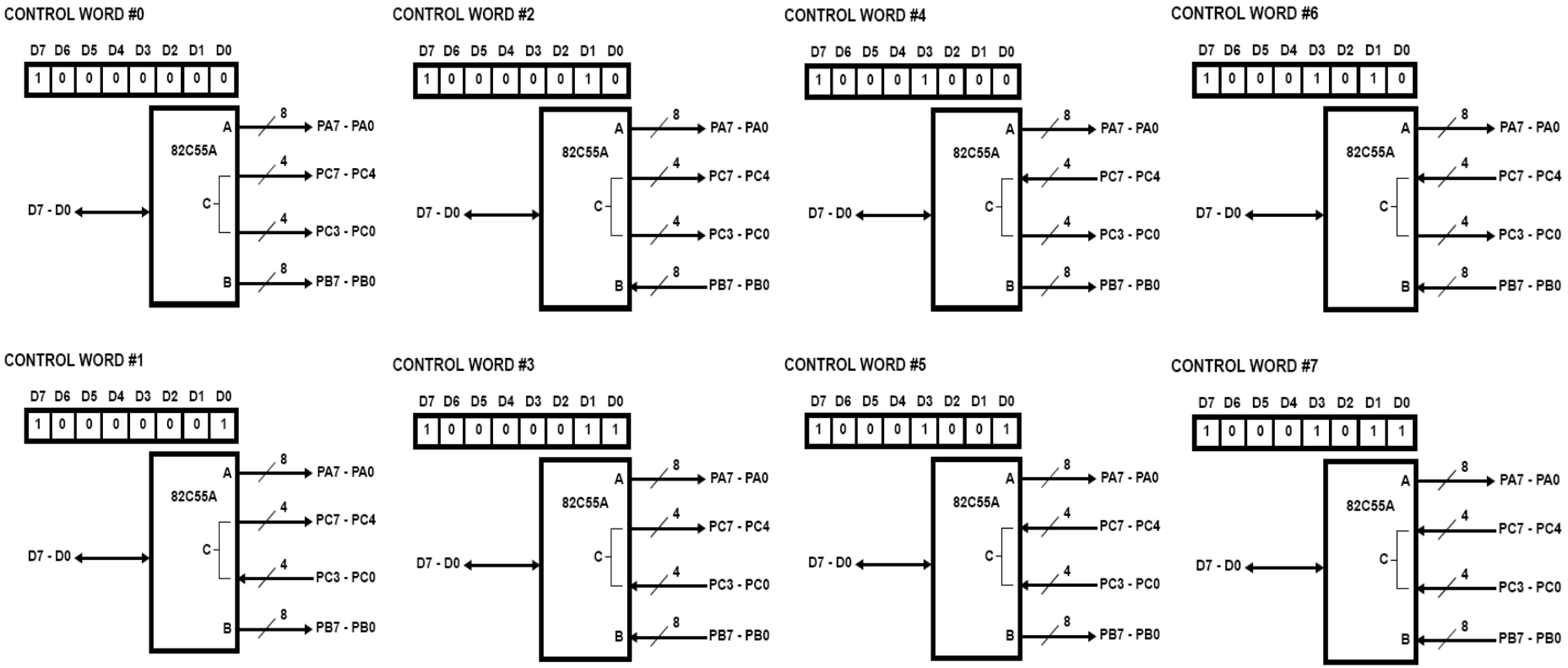
- This functional configuration provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specific port.

- **Mode 0 Basic Functional Definitions:**
  - Two 8-bit ports and two 4-bit ports.
  - Any Port can be input or output.
  - Outputs are latched.
  - Input are not latched (tri-stated).
  - 16 different Input/Output configurations possible.

Mode 0 Port Definition

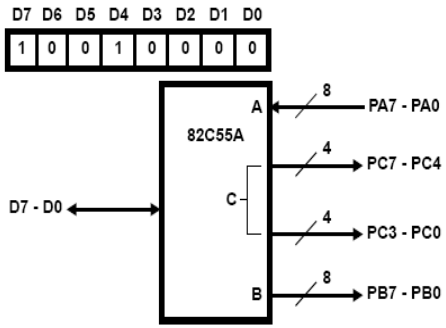
A		B		GROUP A		#	GROUP B	
D4	D3	D1	D0	PORT A	PORT C (Upper)		PORT B	PORT C (Lower)
0	0	0	0	Output	Output	0	Output	Output
0	0	0	1	Output	Output	1	Output	Input
0	0	1	0	Output	Output	2	Input	Output
0	0	1	1	Output	Output	3	Input	Input
0	1	0	0	Output	Input	4	Output	Output
0	1	0	1	Output	Input	5	Output	Input
0	1	1	0	Output	Input	6	Input	Output
0	1	1	1	Output	Input	7	Input	Input
1	0	0	0	Input	Output	8	Output	Output
1	0	0	1	Input	Output	9	Output	Input
1	0	1	0	Input	Output	10	Input	Output
1	0	1	1	Input	Output	11	Input	Input
1	1	0	0	Input	Input	12	Output	Output
1	1	0	1	Input	Input	13	Output	Input
1	1	1	0	Input	Input	14	Input	Output
1	1	1	1	Input	Input	15	Input	Input

Mode 0 Configurations

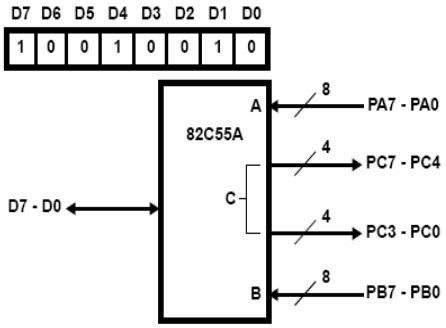


Mode 0 Configurations

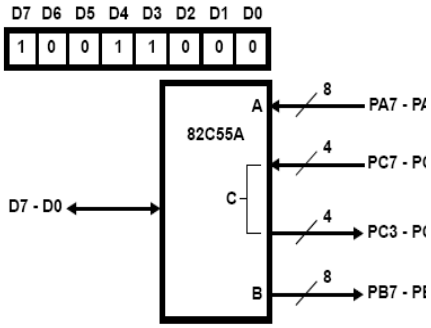
CONTROL WORD #8



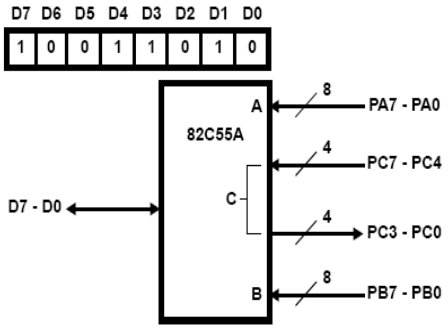
CONTROL WORD #10



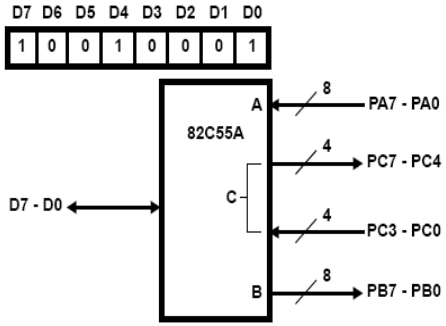
CONTROL WORD #12



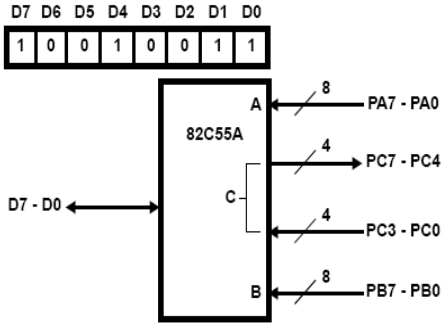
CONTROL WORD #14



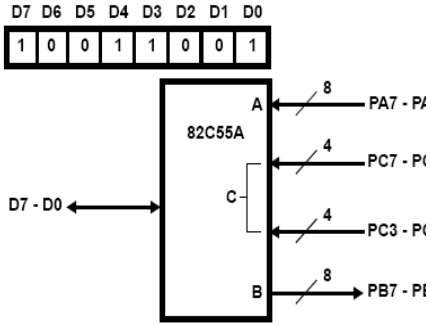
CONTROL WORD #9



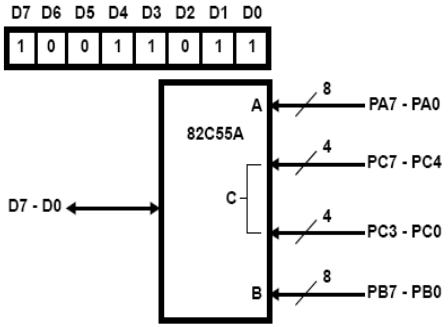
CONTROL WORD #11



CONTROL WORD #13

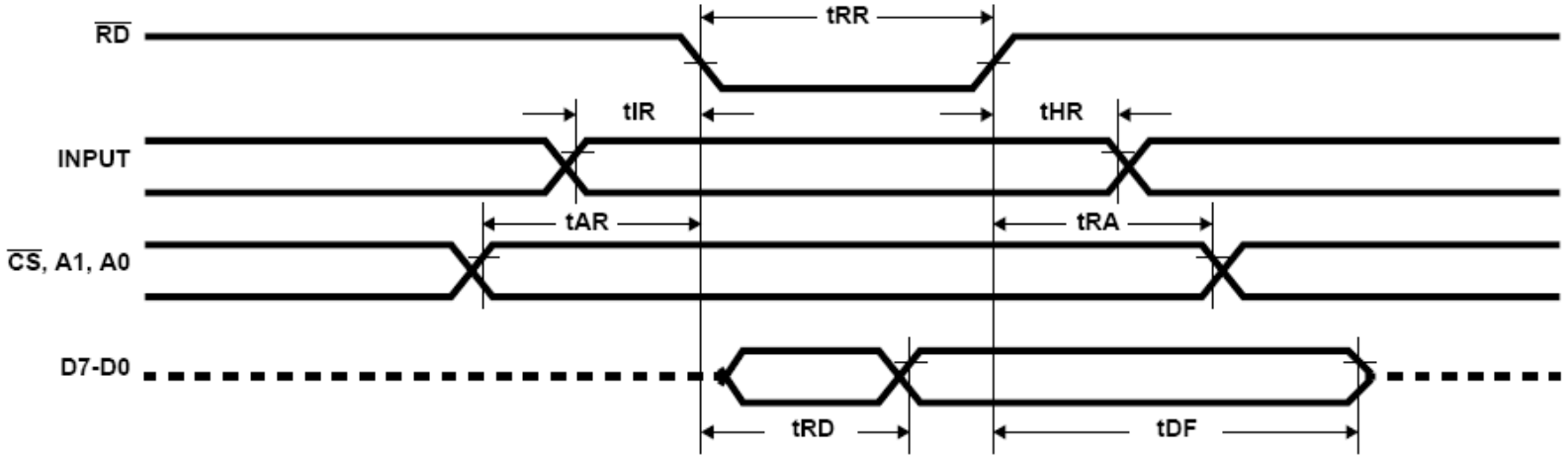


CONTROL WORD #15

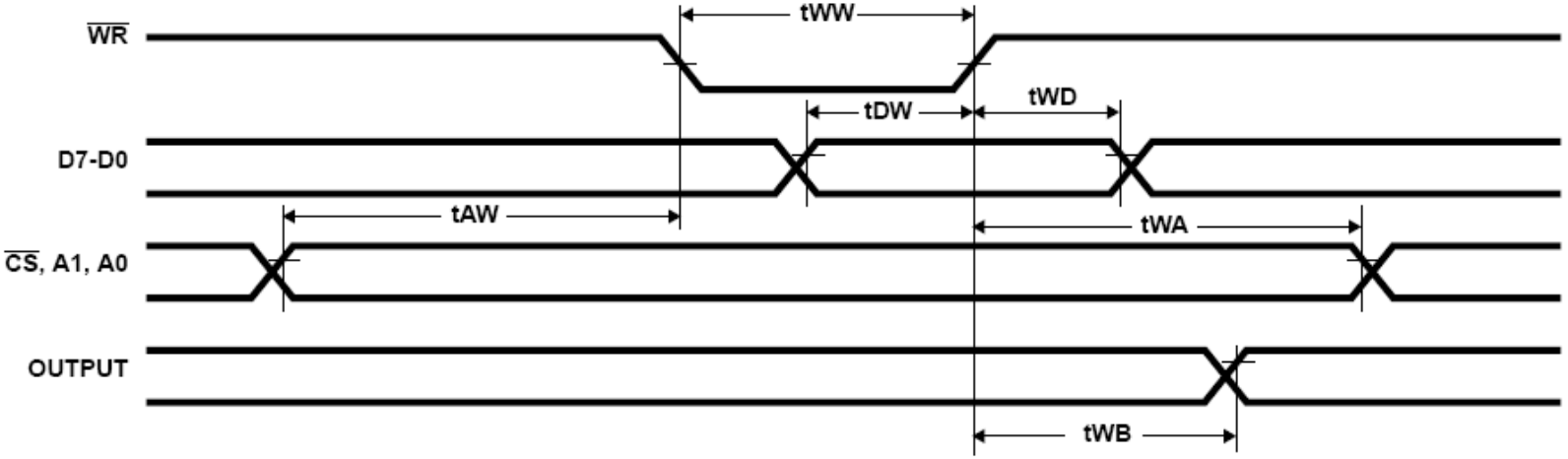


Mode 0 Configurations

Mode 0 (Basic Input)



Mode 0 (Basic Output)



Decoding Circuit for I/O Chips

Example 1:

- A system has 8 I/O chips.
- The dedicated I/O space is: 8000H – 9FFFH.
- Some of the I/O chips require 16 contiguous I/O ports.

Solution:

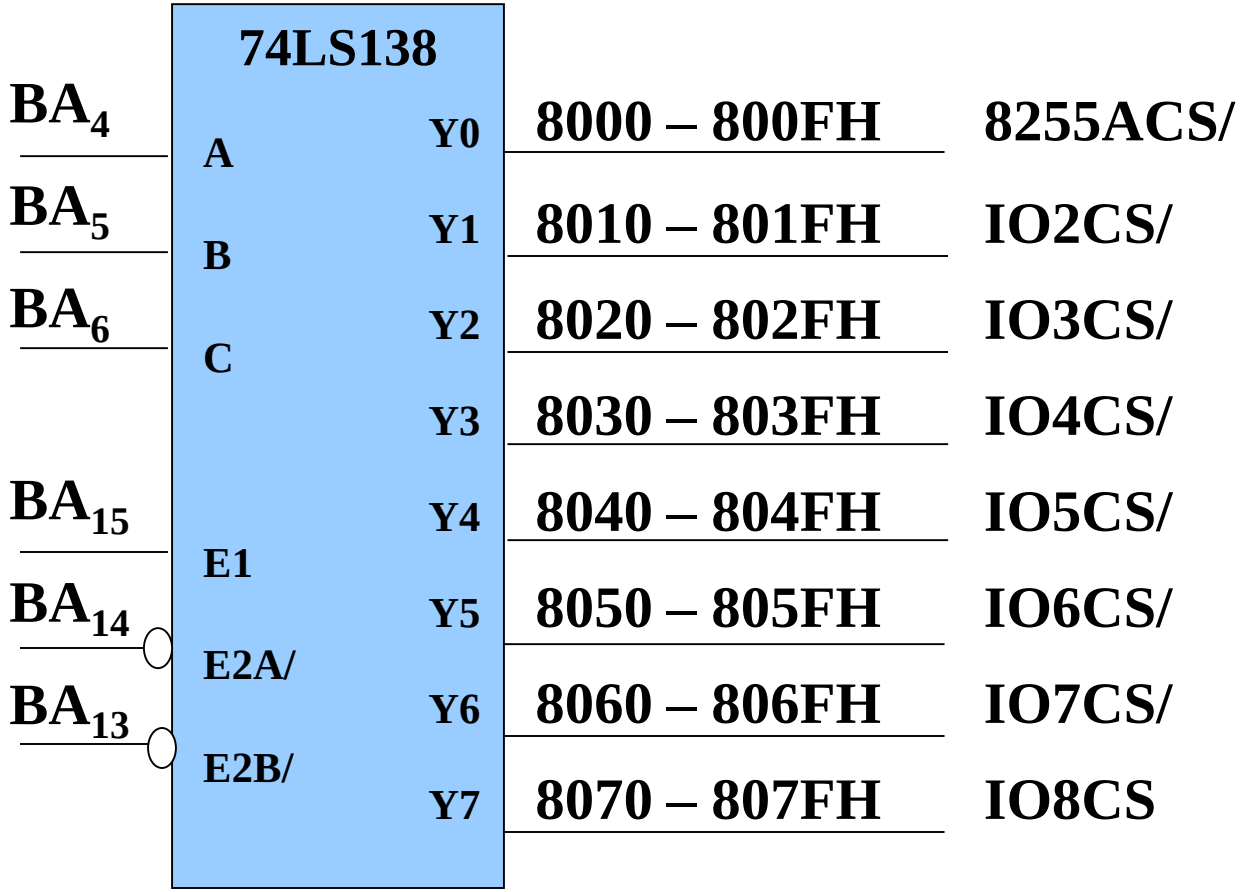
8000 – 9FFF

1000 1001 → BA15, BA14, and BA13 are constants.

BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
1	0	0	x	x	x	x	x	x	0	0	0	-	-	-	-
Constants									Eight I/O Chips			16 contiguous I/O ports			

Decoding Circuit for I/O Chips

Solution of Example 1:



Port A	8000H
Port B	8001H
Port C	8002H
CW	8003H

Decoding Circuit for I/O Chips

Example 2:

- A system has 4 I/O chips.
- The dedicated I/O space is: 4000H – 40FFH.
- Some of the I/O chips require 8 contiguous I/O ports.

Solution:

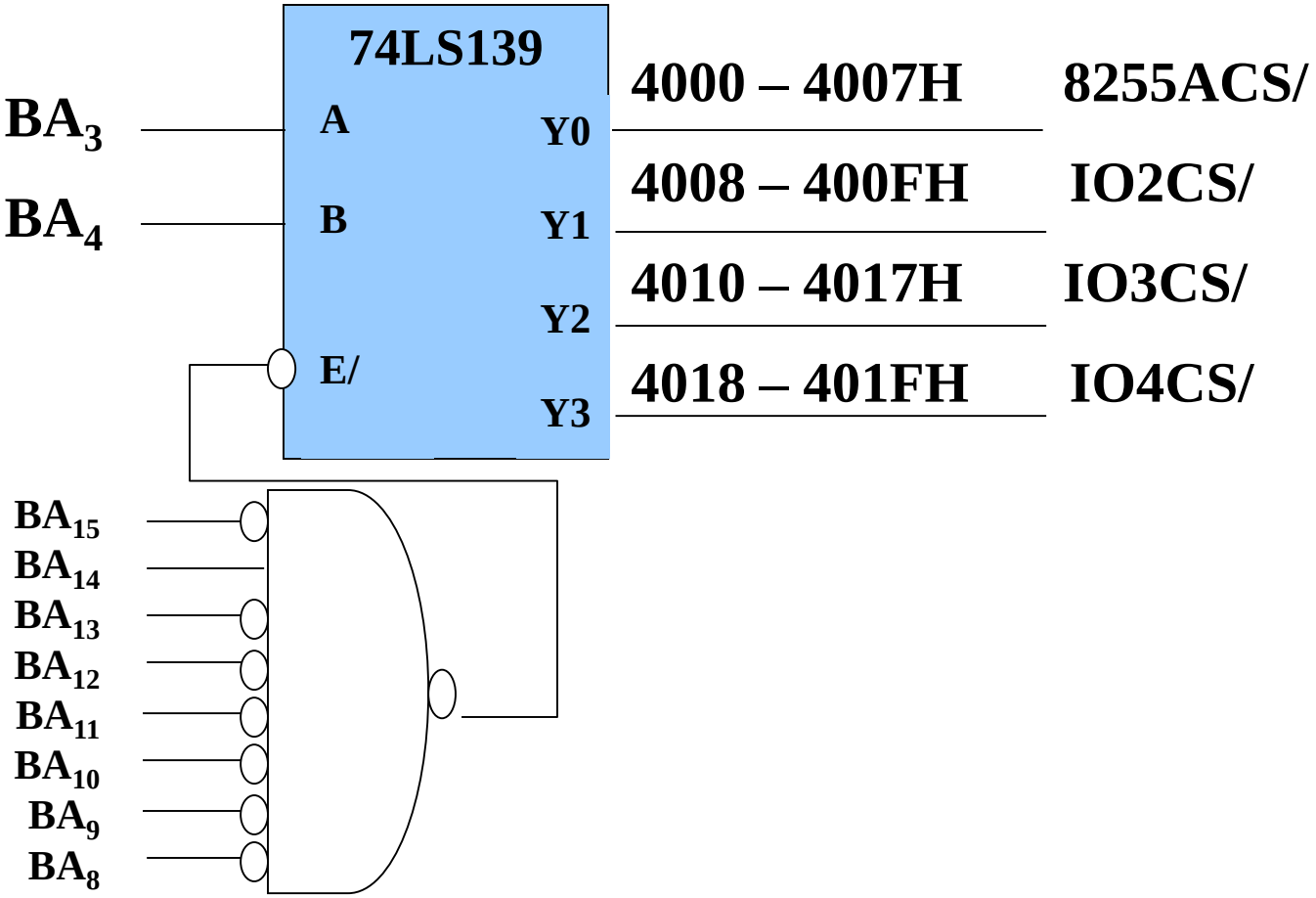
4000 – 40FF

→ **BA15 – BA8 are constants.**

BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
0	1	0	0	0	0	0	0	x	x	x	0	0	-	-	-
Constants											Four I/O Chips		8 contiguous I/O ports		

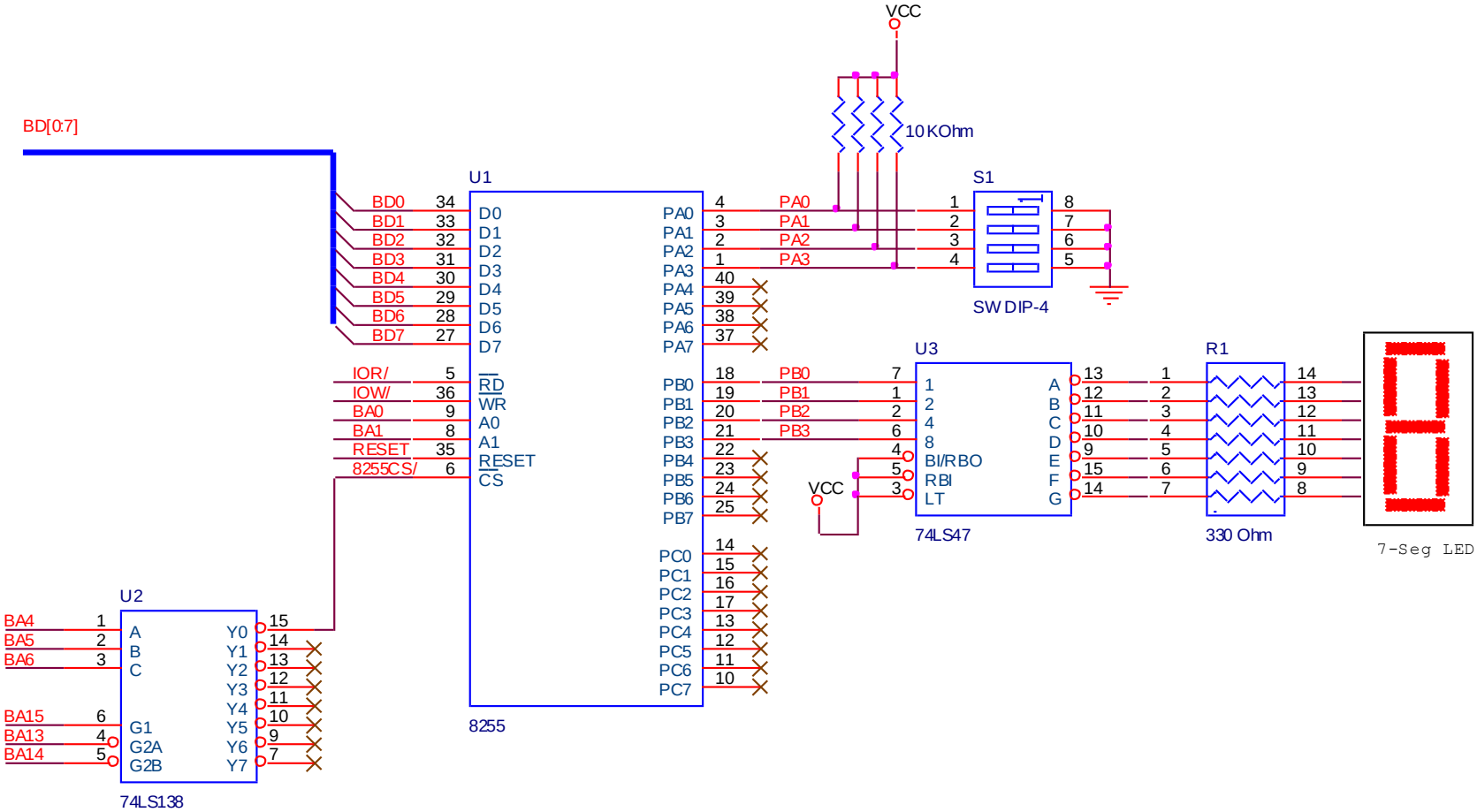
Decoding Circuit for I/O Chips

Solution of Example 2:



Port A	4000H
Port B	4001H
Port C	4002H
CW	4003H

Application 1: Basic Input/Output

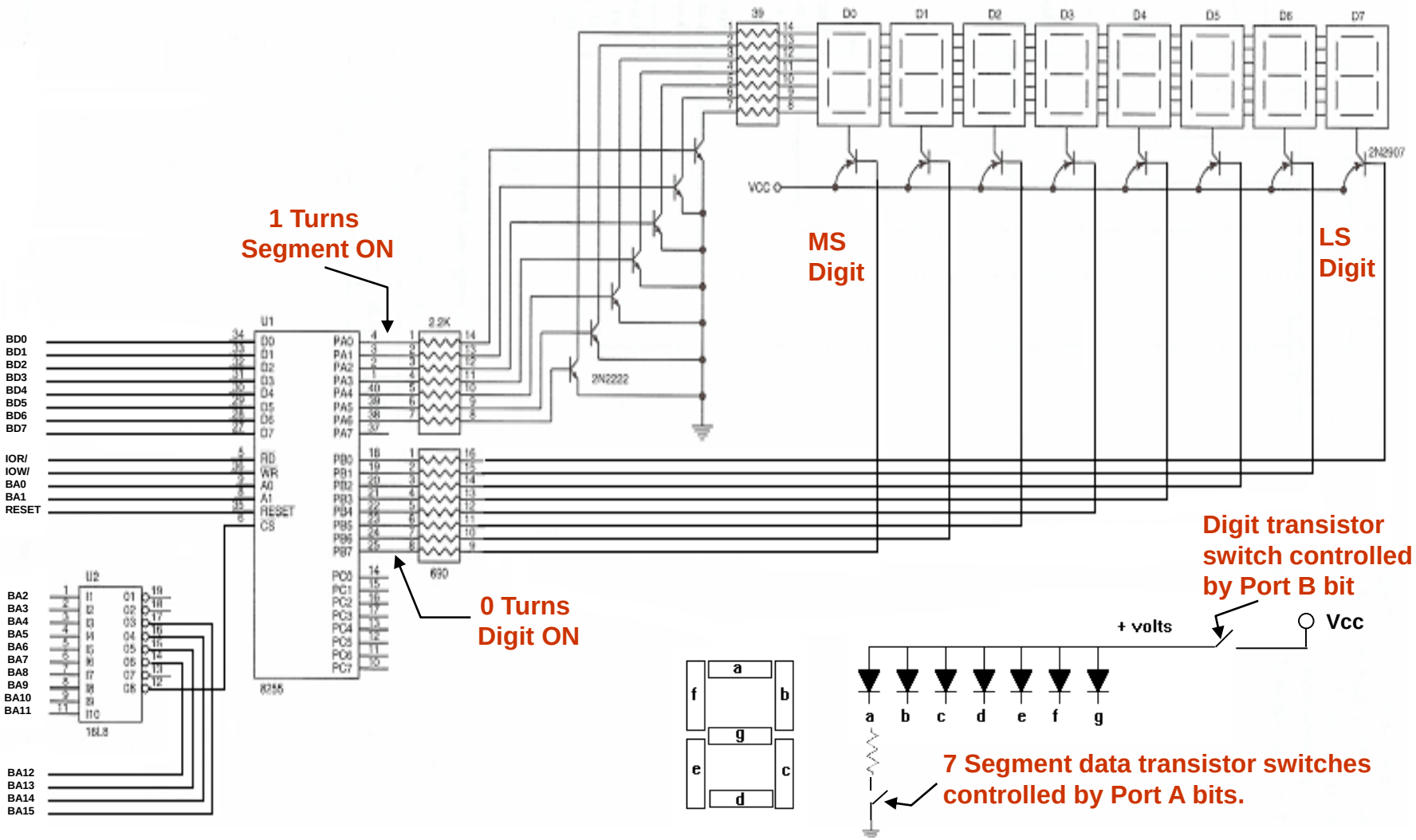


BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
1	0	0	x	x	x	x	x	x	0	0	0	x	x	-	-
Enable Signals									Y1					Ports	

Application 1: Basic Input/Output

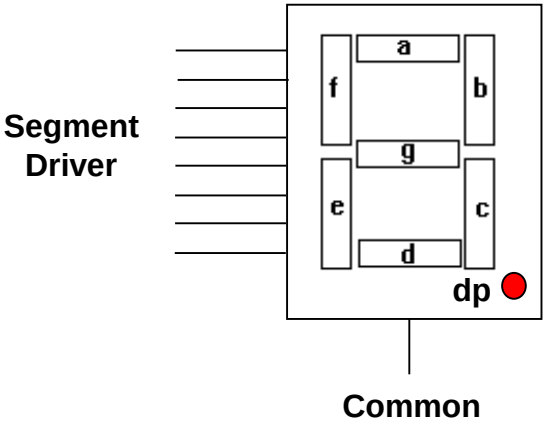
```
            PORTA   EQU     8000H
            PORTB   EQU     8001H
            PORTC   EQU     8002H
            PCW     EQU     8003H

; 8255A Initialization
            MOV     DX, PCW
            MOV     AL, 10011001B
            OUT     DX, AL

; To read port A unconditionally
again:      MOV     DX, PORTA
            IN      AL, DX

; To write the value to port B unconditionally
            INC     DX
            OUT     DX, AL
            JMP     again
```

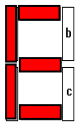
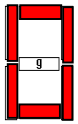
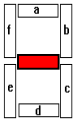
Application 2: 7-Segement Display



An 8-digit LED display interfaced to an 8088 microprocessor through 8255A

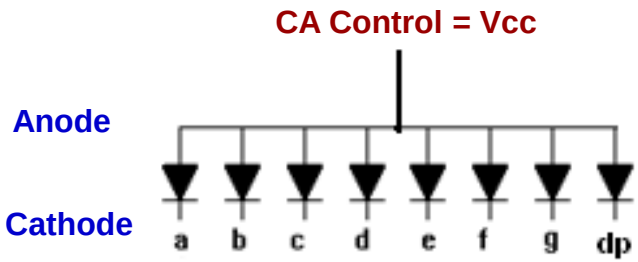
Application 2: 7-Segement Display



Character	dp	g	f	e	d	c	b	a	Hex
	0	1	1	1	1	0	0	1	79H
	0	0	1	1	1	1	1	1	3FH
	0	1	0	0	0	0	0	0	40H

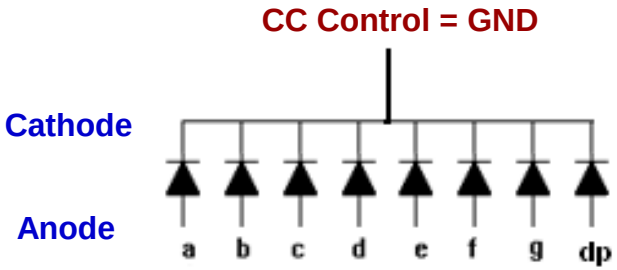
Two Types:

Common Anode (CA)



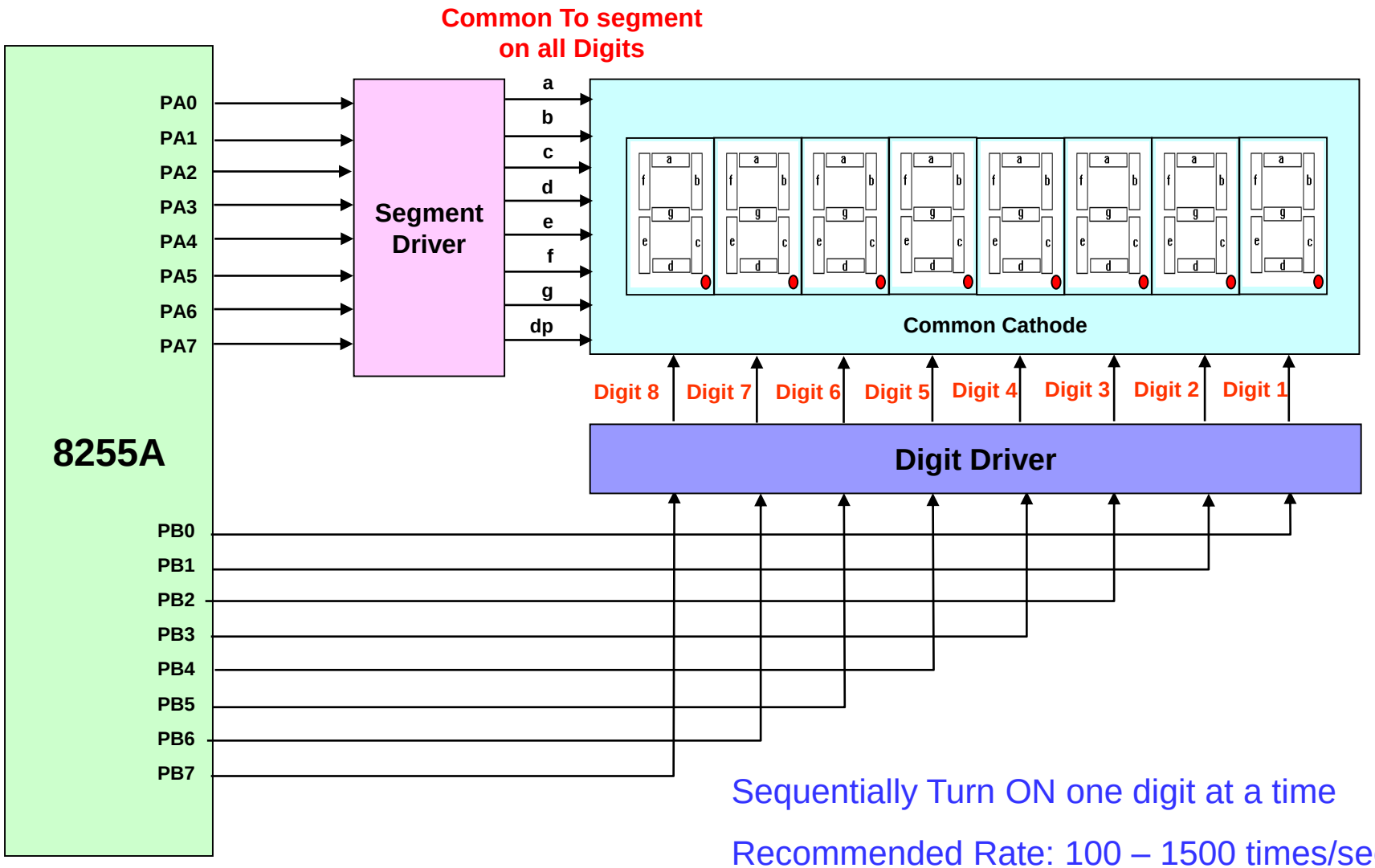
Select Segments:
Switched Resistors to GND

Common Cathode (CC)



Select Segments:
Switched Resistors to Vcc

Application 2: 7-Segement Display



Sequentially Turn ON one digit at a time

Recommended Rate: 100 – 1500 times/sec

Application 2: 7-Segement Display

A_1	A_0	Address	Port
0	0	700H	Port A
0	1	701H	Port B
1	0	702H	Port C
1	1	703H	Control Word

PLD Program for Address Decoding

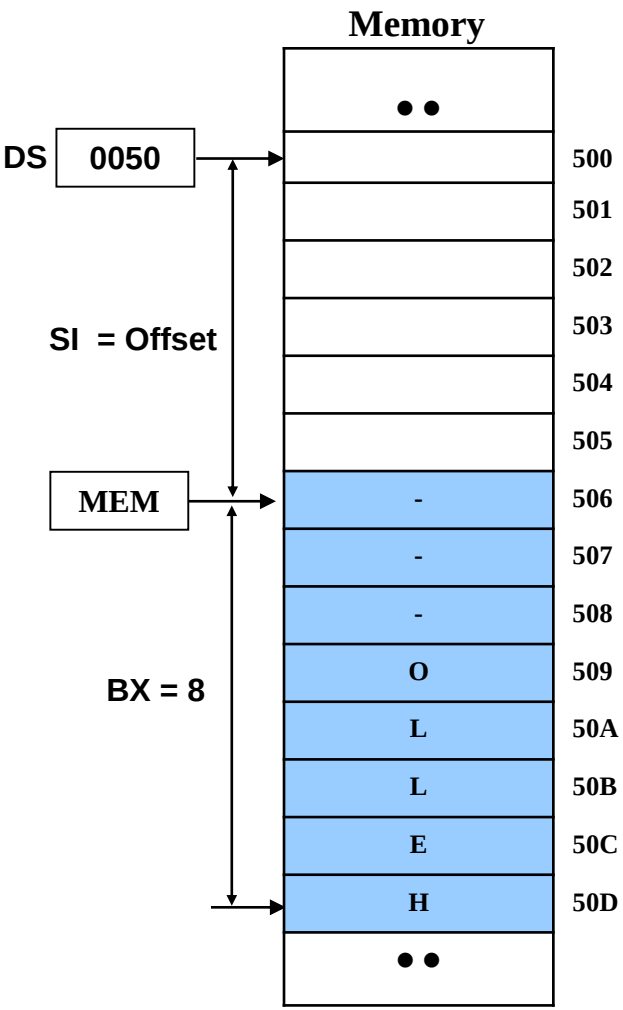
```
library ieee;
use ieee.std_logic_1164.all;
entity DECODER_11_21 is
port (
        A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2: in STD_LOGIC;
        D0: out STD_LOGIC
);
end;
architecture V1 of DECODER_11_17 is
begin
            D0 <=     A15 or A14 or A13 or A12 or A11 or not A10
                      or not A9 or not A8 or A7 or A6 or A5 or A4 or A3 or A2;
end V1;
```

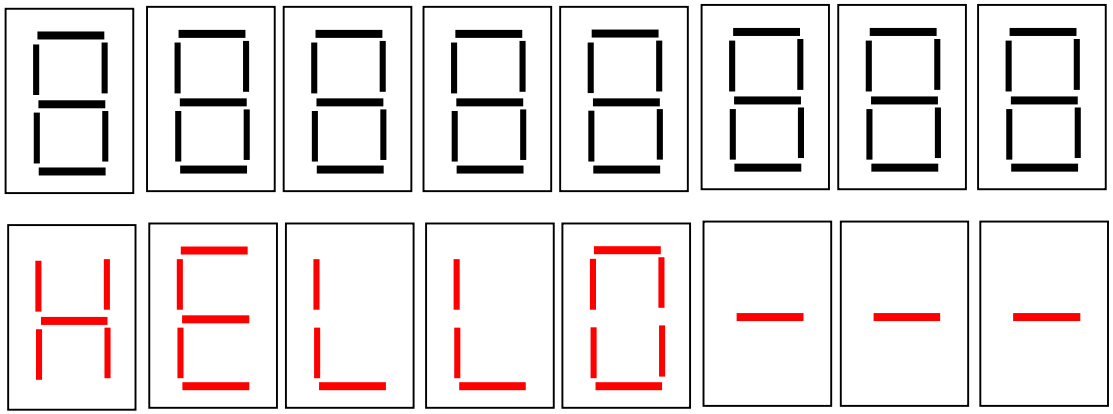
BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
0	0	0	0	0	1	1	1	0	0	0	0	0	0	-	-

14-bit address decoding Using PLD

On chip Selection

Application 2: 7-Segement Display





MEM = 506

MOV SI, OFFSET MEM - 1 → SI = (506 - 1) - 500 = 5

● ● ●

MOV AL, [BX + SI] → DS = 0050 0

BX = 0008

SI = 0005

0050D

Content of the address 0050D is 'H'.

Application 2: 7-Segement Display

```
; Program the 82C55 for Port A and Port B are output ports in mode 0
            MOV DX, 703H                    ; Address of Command Port into DX
            MOV AL, 80H                     ; 80H Data into AL
            OUT DX, AL              ; Write 80H into Command Port to program PPI

; An assembly language procedure that multiplexes the 8-digit display.
; This procedure must be called often enough for the display to appear stable
DISP        PROC        NEAR USES AX BX DX SI
            PUSHF
            MOV  BX, 8                      ;load counter BX with # of display digits
            MOV  AH, 01111111B              ;load initial digit selection pattern to enable MS digit
            MOV  SI, OFFSET MEM - 1         ;Load SI with offset (MEM) - 1
            MOV  DX, 701H                   ;address Port B (for Port A: decrement DX)
;Sequentially display all 8 digits starting with MS digit
            .REPEAT
                        MOV  AL, AH
                        OUT  DX, AL ;send digit selection pattern to Port B
                        DEC  DX             ;Address Port A (to send Digit Data)
                        MOV  AL, [BX+SI]    ;Load digit data from memory into AL
                        OUT  DX, AL         ;send digit data to Port A
                        CALL DELAY          ;wait 1.0 ms leaving displayed digit ON
                        ROR  AH, 1          ;adjust selection pattern to point to next digit
                        INC  DX             ;Address port B
                        DEC  BX             ;decrement counter for data of next digit.
            .UNTIL BX == 0
            POPF
            RET
DISP        ENDP
```

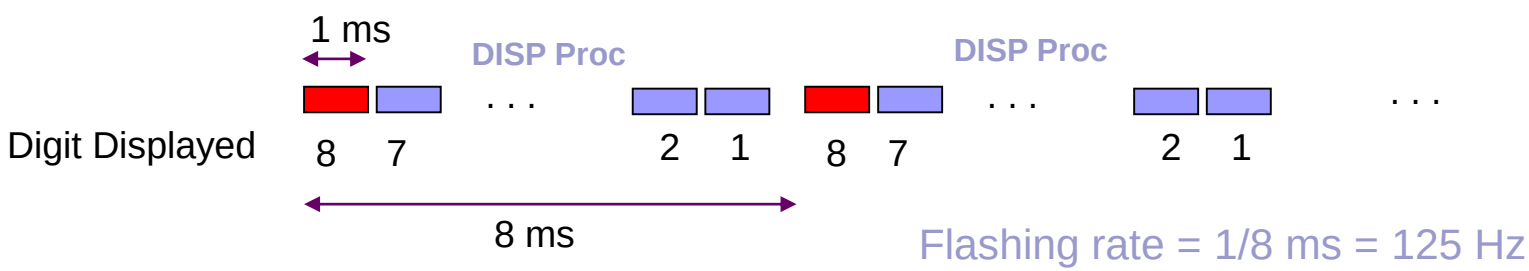
Application 2: 7-Segement Display

```
; Delay Loop
DELAY     PROC      NEAR USES CX
          MOV CX, XXXX              ; XXXX determines delay, = Delay required / loop exec time
D1:
          LOOP D1
          RET
DELAY     ENDP
```

Loop execution time is calculated from instruction data and the clock frequency.
An 80486 executes “LOOP D1” in 7 clock cycles
With a 20 MHz clock, loop exec time = 7 x 50 = 350 ns
XXXX = 1ms/350ns

Display Flashing Rate:

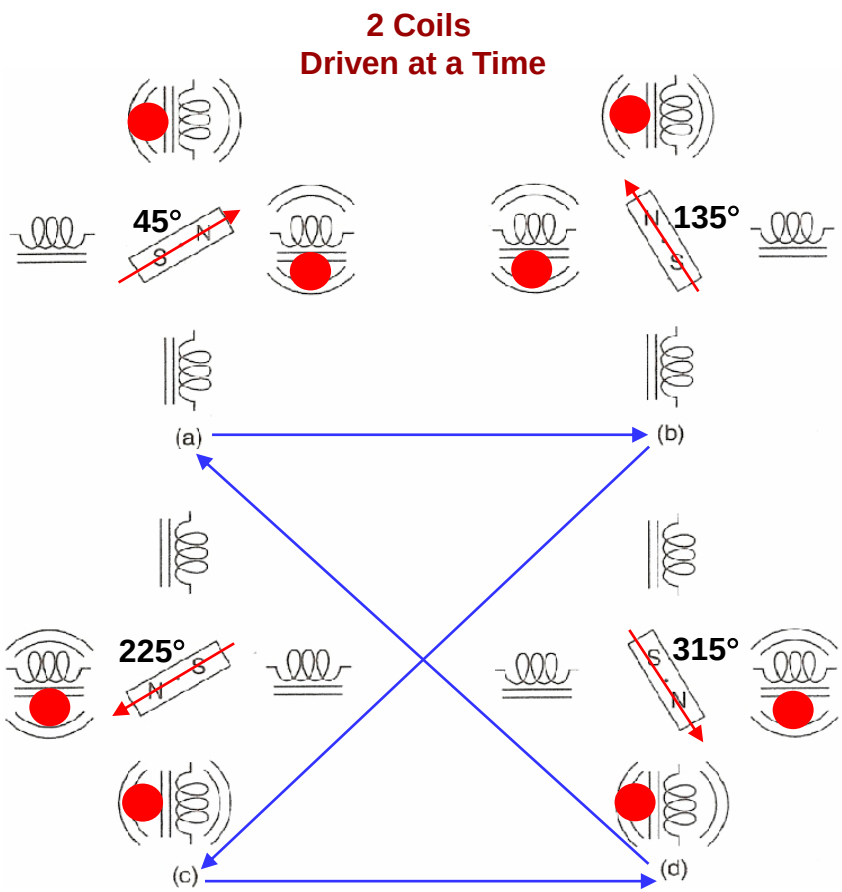
- Assume the DISP Procedure is called continuously
- Ignore loop execution times relative to delay time (e.g. 350 ns << 1 ms)



Flashing rate = 1/8 ms = 125 Hz

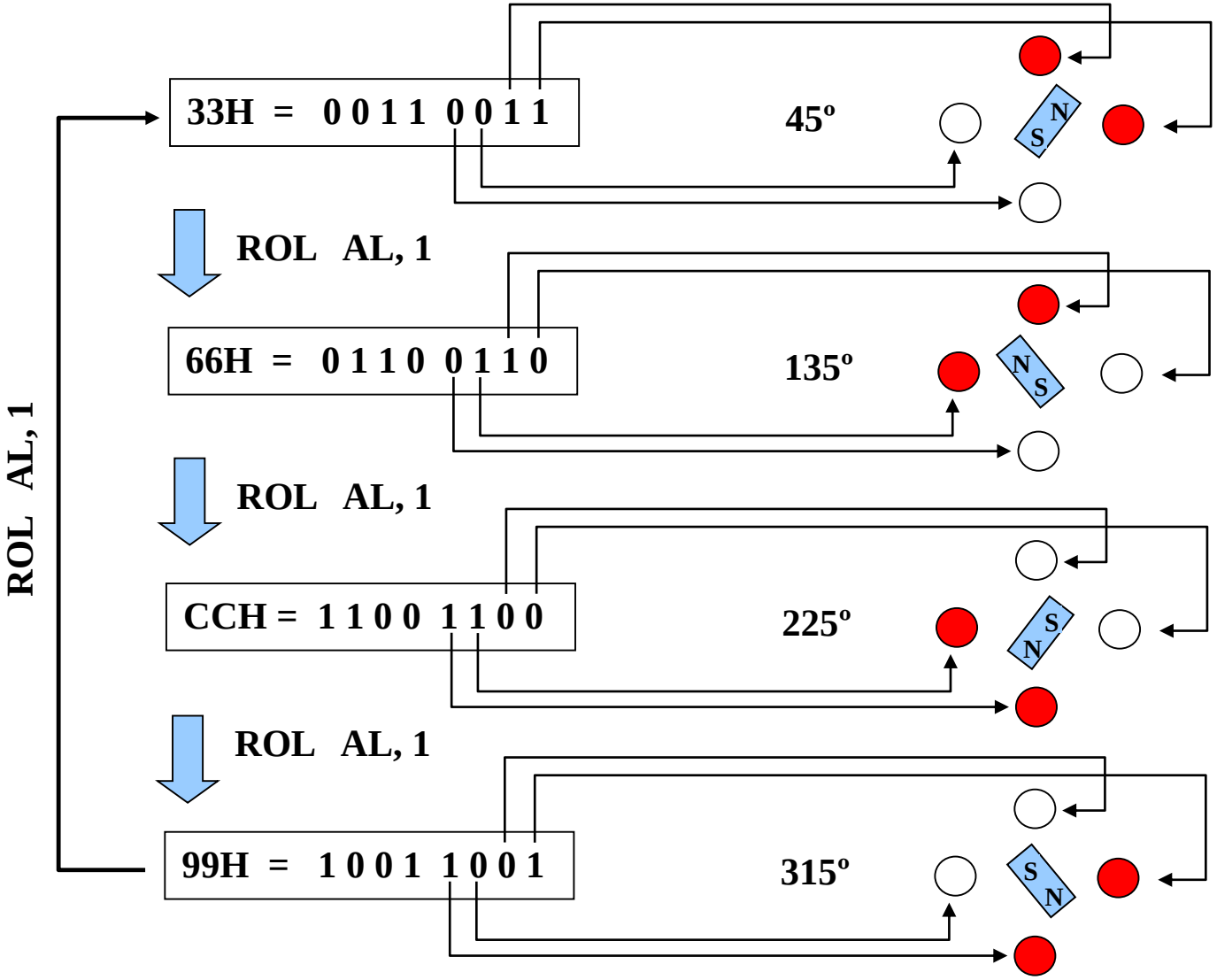
Application 3: Stepper Motor

- Stepper motor is digital in nature.
- It rotates in a sequence of discrete steps controlled by sequentially energizing a set of coils (windings).
- Step angles vary from 1° to 15 ° depending on precision required (and cost)

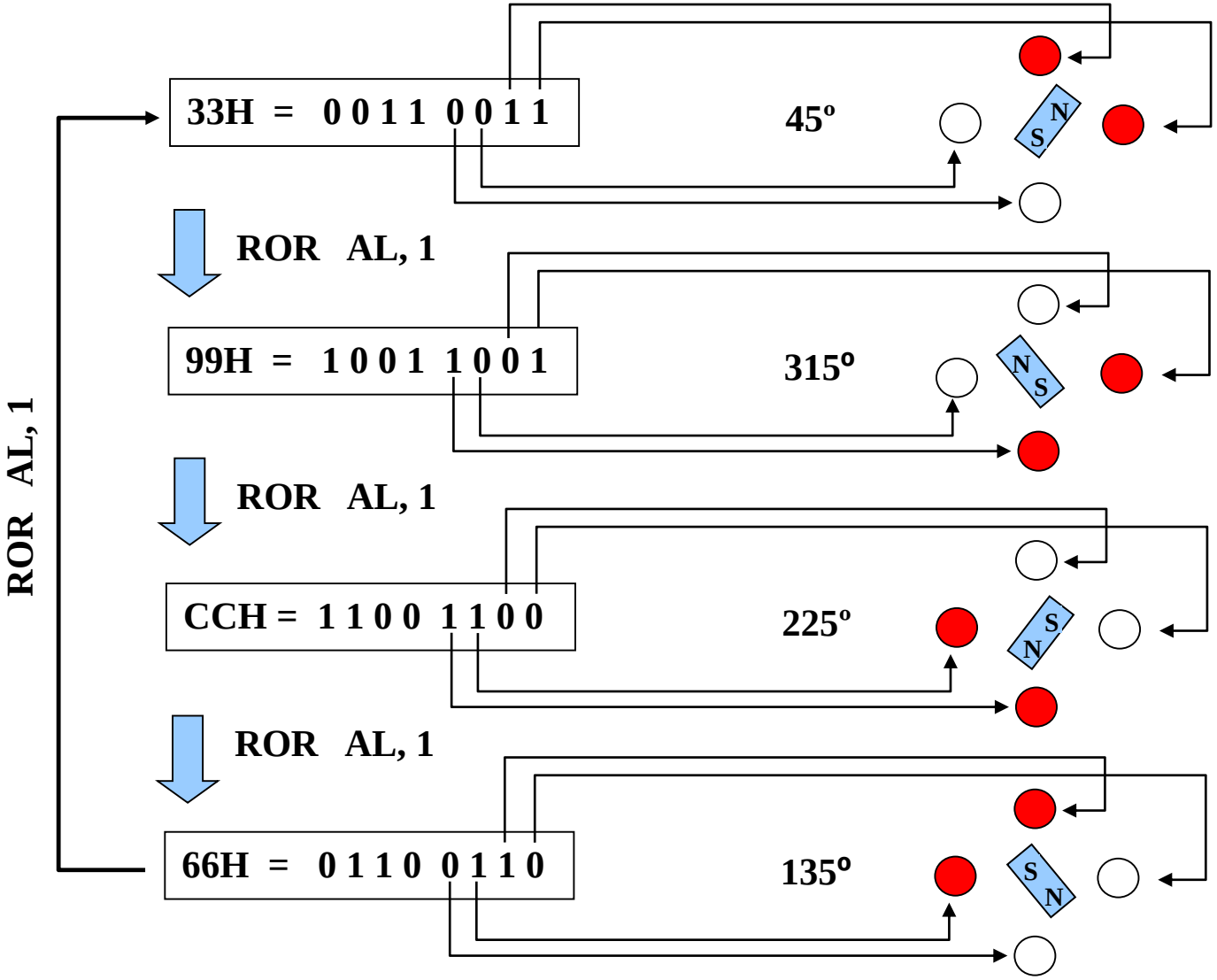


- **N Pole lies between the two energized coils**
- **Rotation Direction: Anti-clock wise**
- **Step angle: 90°**

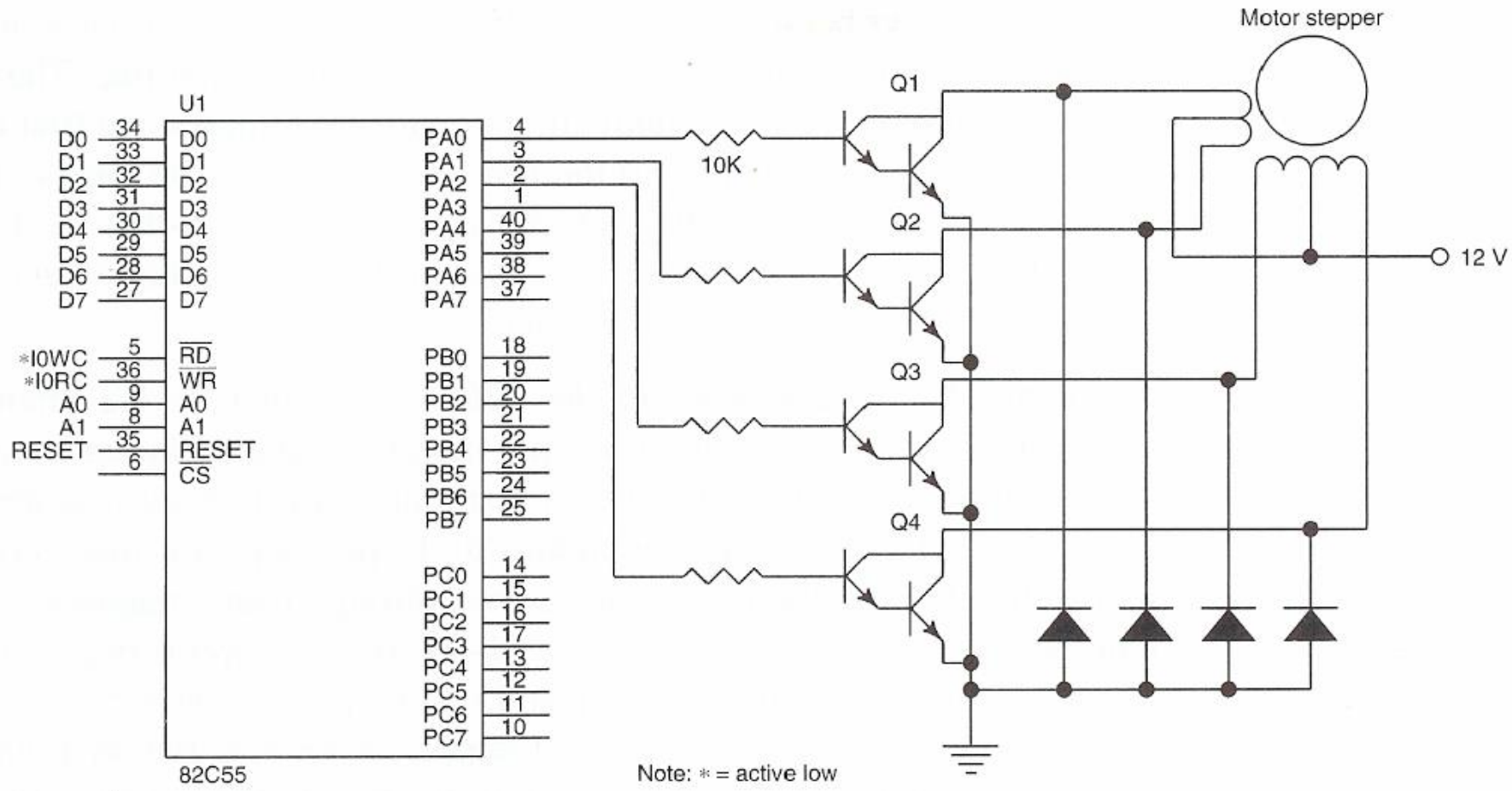
Application 3: Stepper Motor



Application 3: Stepper Motor



Application 3: Stepper Motor



Application 3: Stepper Motor

```
                PORTA       EQU         40H
;An assembly language procedure that controls the stepper motor
STEP            PROC        NEAR USES CX AX
                MOV  AL, POS                                    ;get position
                OR   CX, CX                                     ;set flag bits
                .IF !ZERO?
                            .IF !SIGN?                          ;if not sign → Rotate left
                                        .REPEAT
                                                    ROL  AL,1   ;rotate step left
                                                    MOV  DX, PORTA
                                                    OUT  DX, AL
                                                    CALL DELAY  ; wait 1 ms for motor to move
                                        .UNTIL  CXZ
                            .ELSE
                                        AND CX, 7FFFH           ;make CX positive
                                        .REPEAT
                                                    ROR  AL,1   ;rotate step right
                                                    MOV  DX, PORTA
                                                    OUT  PORT,AL
                                                    CALL DELAY   ;wait 1 ms for motor to move
                                        .UNTILCXZ
                            .ENDIF
                .ENDIF
                MOV  POS,AL                                     ; Save POSN for next step
                RET
STEP            ENDP
```

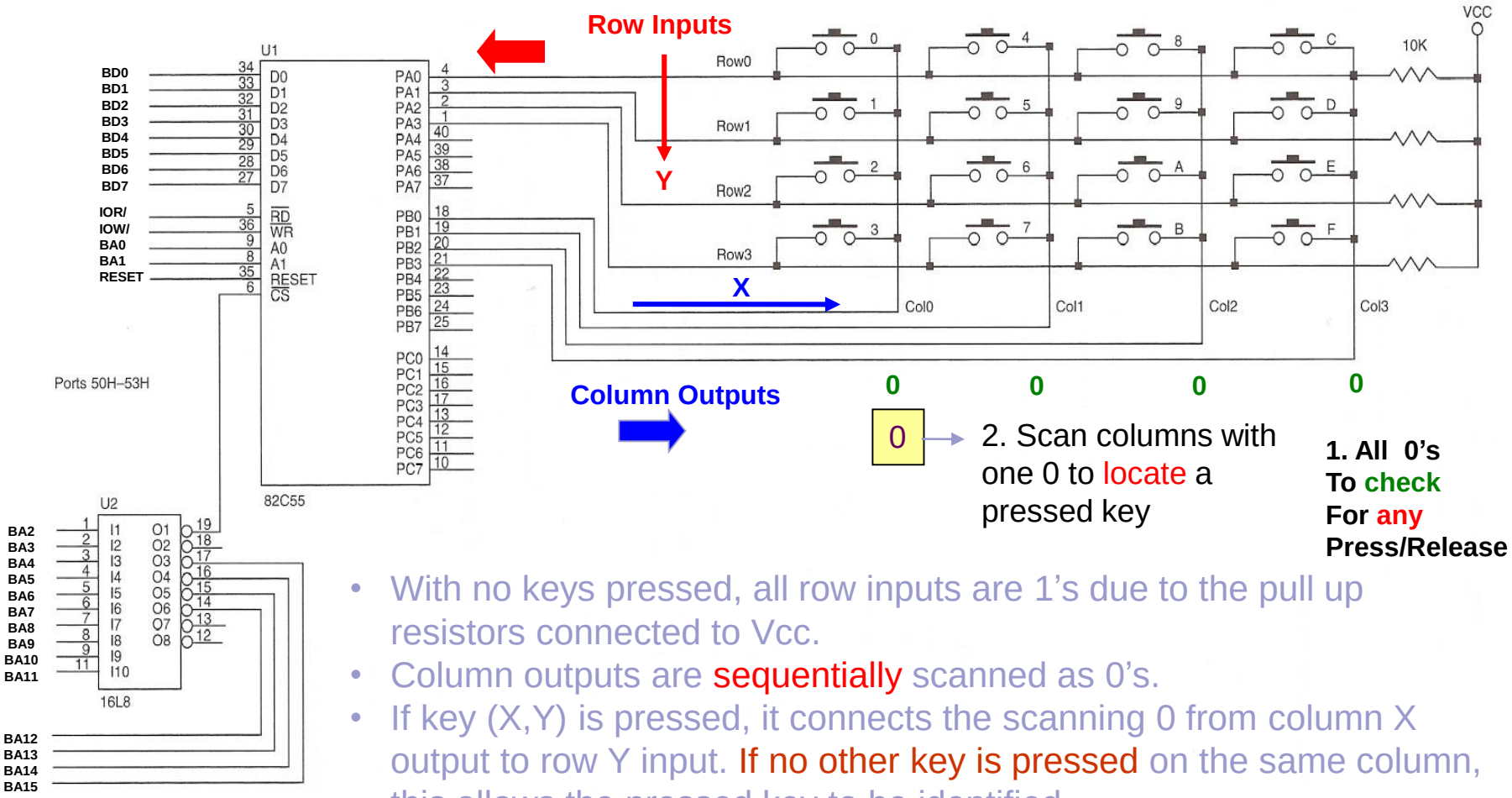
CX Positive: Rotate Anti-clockwise

CX Negative: Rotate Clockwise

CX has:
Sign of Rotation
0: ACW 1:CW
# of Steps

e.g. 0000H (0)
0005H (5,ACW)
8007H (7,CW)

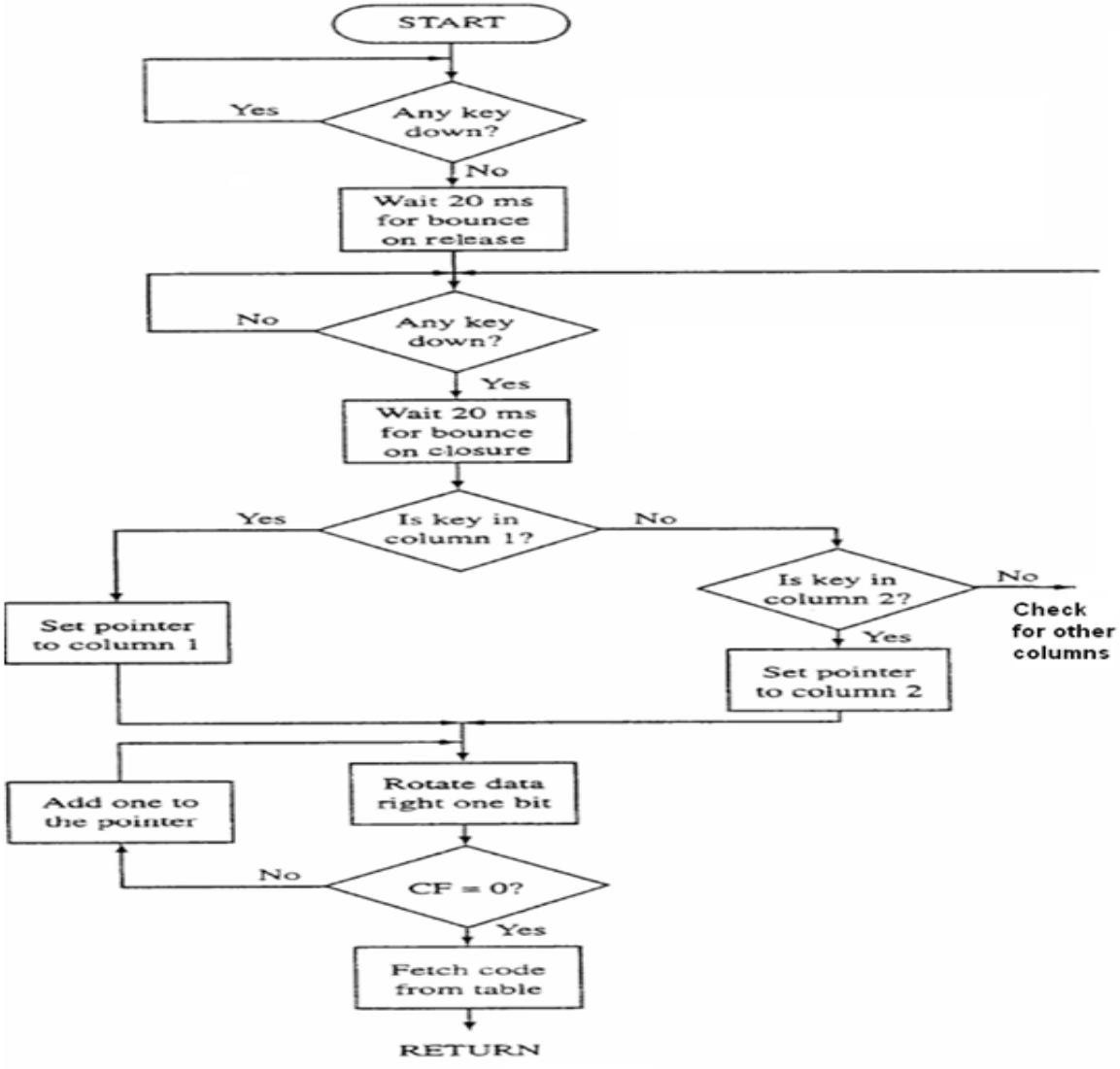
Application 4: Interfacing a 4x4 Key Matrix



- With no keys pressed, all row inputs are 1's due to the pull up resistors connected to Vcc.
- Column outputs are sequentially scanned as 0's.
- If key (X,Y) is pressed, it connects the scanning 0 from column X output to row Y input. If no other key is pressed on the same column, this allows the pressed key to be identified.

An 4 ×4 keyboard matrix interfaced to an 8088 microprocessor through 8255A

Application 4: Interfacing a 4x4 Key Matrix



Application 4: Interfacing a 4x4 Key Matrix

```
;KEY scans the keyboard and returns the key code in AL.
                COLS        EQU         4
                ROWS        EQU         4
                PORTA       EQU         50H
                PORTB       EQU         51H
KEY             PROC        NEAR USES CX BX
                MOV  BL,FFH                         ;compute row mask
                SHL  BL,ROWS
                MOV  AL,0
                OUT  PORTB,AL                       ;place all zeros on Port B
                .REPEAT                             ;wait for release
                            .REPEAT
                                        CALL SCAN
                            .UNTIL ZERO?
                            CALL DELAY10            ; Release debounce
                            CALL SCAN
                .UNTIL ZERO?
                .REPEAT                             ;wait for key press (to be determined)
                            .REPEAT
                                        CALL SCAN
                            .UNTIL !ZERO?           ; (not zero, i.e. not = FF)
                            CALL DELAY10            ; Press debounce
                            CALL SCAN               , scan again after things have settled
                .UNTIL !ZERO?
                MOV  CX,00FEH
                .WHILE 1                            ;find column
                            MOV  AL,CL
                            OUT  PORTB,AL
                            CALL SHORTDELAY         ;Wait till data outputted to PortB have settled!
                            CALL SCAN
                            .BREAK !ZERO?           ;Key found at this column- Quit WHILE1!
                            ADD  CH,COLS            ;Key not found at this row- move on to next row – add COLS to CH
                            ROL  CL,1               ;AL = 11111101 on 2nd trial
                .ENDW
                .WHILE 1                            ;find row from pattern Read into PortA in SCAN
                            SHR  AL,1
                            .BREAK .IF !CARRY?      ; LSB of AL is shifted into the carry flag by SHR! So we stop on 1st zero bit
                            INC  CH                 ; for each shift until row is found
                .ENDW
                MOV  AL,CH                          ;get key code into AL: AL = CH = (COLS) X + Y = 4 X + Y; X = 0, 1,..,3 , Y = 0, 1,.., 3
                RET
KEY             ENDP
SCAN            PROC        NEAR
                IN   AL,PORTA                       ;read rows
                OR   AL,BL
                CMP  AL,0FFH                        ;test for no keys
                RET
SCAN            ENDP
```

Keep calling SCAN
Until FF (no key pressed)
i.e. wait for key release

Keep calling SCAN
Until (Not FF) (a key pressed)
i.e. wait for key stroke

1st Column

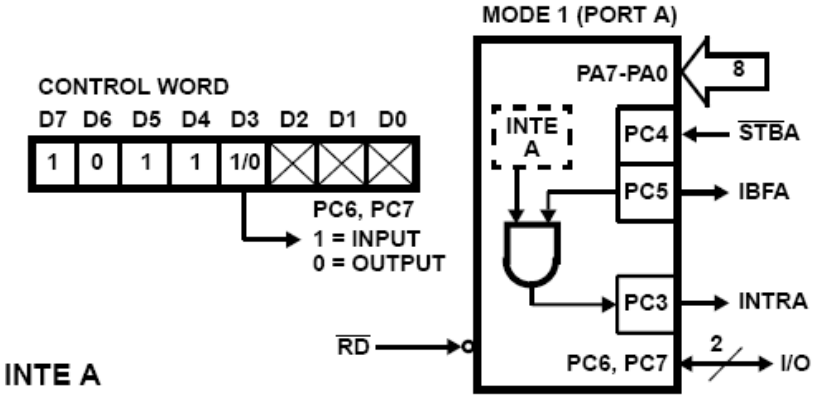
AL = 11111110

Mode 1 (Strobed Input/Output)

- This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or “hand shaking” signals. In mode 1, port A and port B use the lines on port C to generate or accept these “hand shaking” signals.

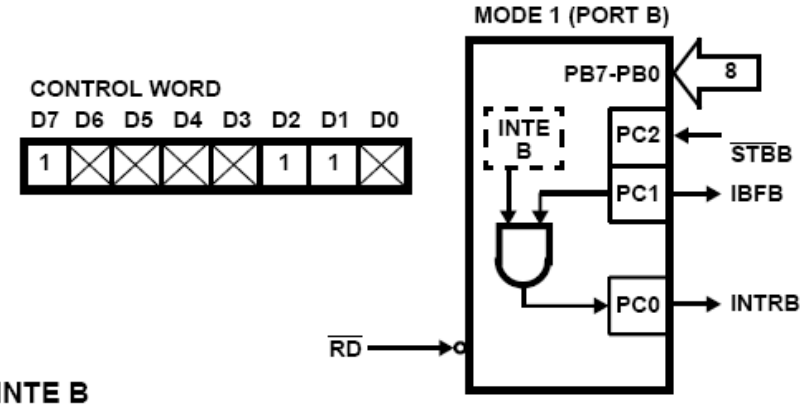
- **Mode 1 Basic Function Definitions:**
  - Two Groups (Group A and Group B).
  - Each group contains one 8-bit port and one 5-bit (Group A) or 3-bit (Group B) control/data port.
  - The 8-bit data port can be either input or output. Both inputs and outputs are latched.

Mode 1 (Strobed Input)



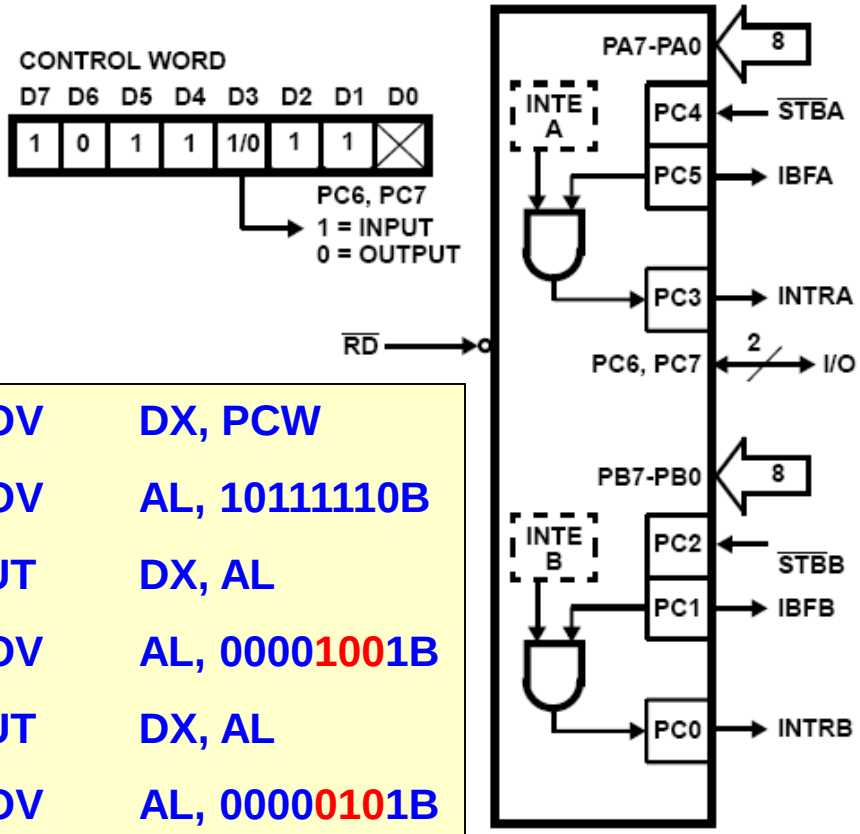
INTE A

Controlled by bit set/reset of PC4.



INTE B

Controlled by bit set/reset of PC2.




```
MOV     DX, PCW
MOV     AL, 10111110B
OUT     DX, AL
MOV     AL, 00001001B
OUT     DX, AL
MOV     AL, 00000101B
OUT     DX, AL
```

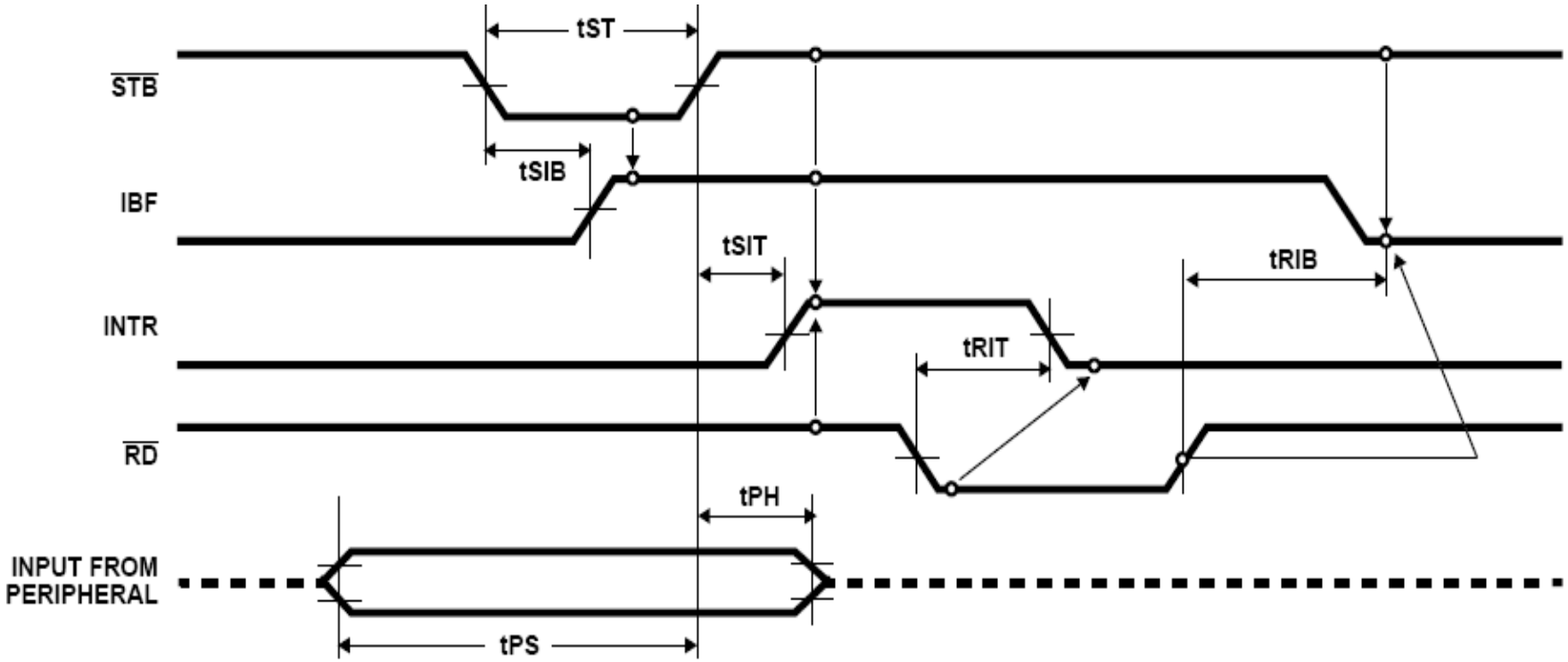
Input Control Signal Definition

- **STB (Strobe Input)**
  A “low” on this input loads data into the input latch.

- **IBF (Input Buffer Full F/F)**
- A “high” on this output indicates that the data has been loaded into the input latch: in essence, and acknowledgment. IBF is set by STB input being low and is reset by the rising edge of the RD input.

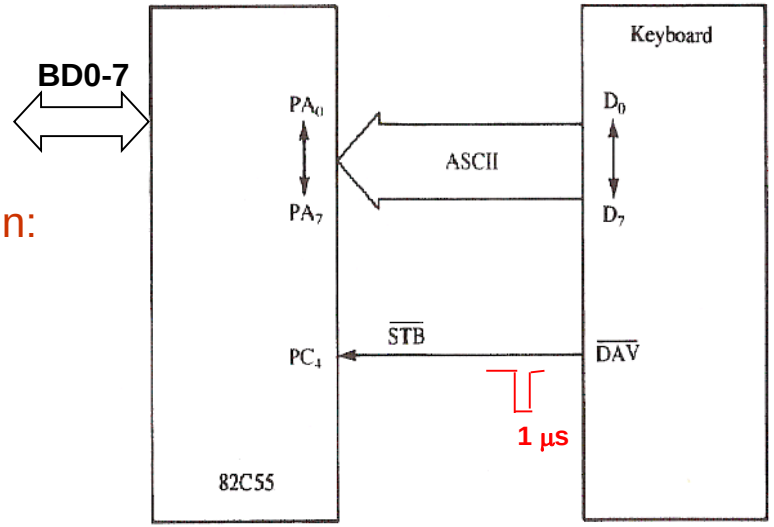
- **INTR (Interrupt Request)**
  A “high” on this output can be used to interrupt the CPU when and input device is requesting service. INTR is set by the condition: STB is a “one”, IBF is a “one” and INTE is a “one”. It is reset by the falling edge of RD. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

Mode 1 (Strobed Input)



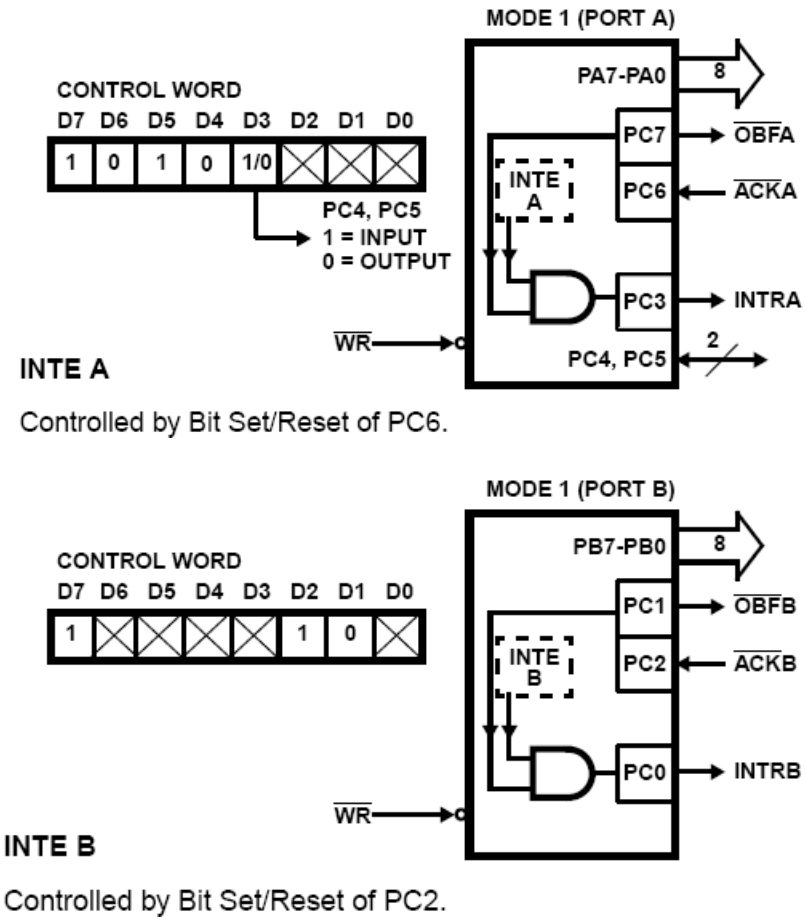
Interfacing a Keyboard to μP using Port A in Mode 1

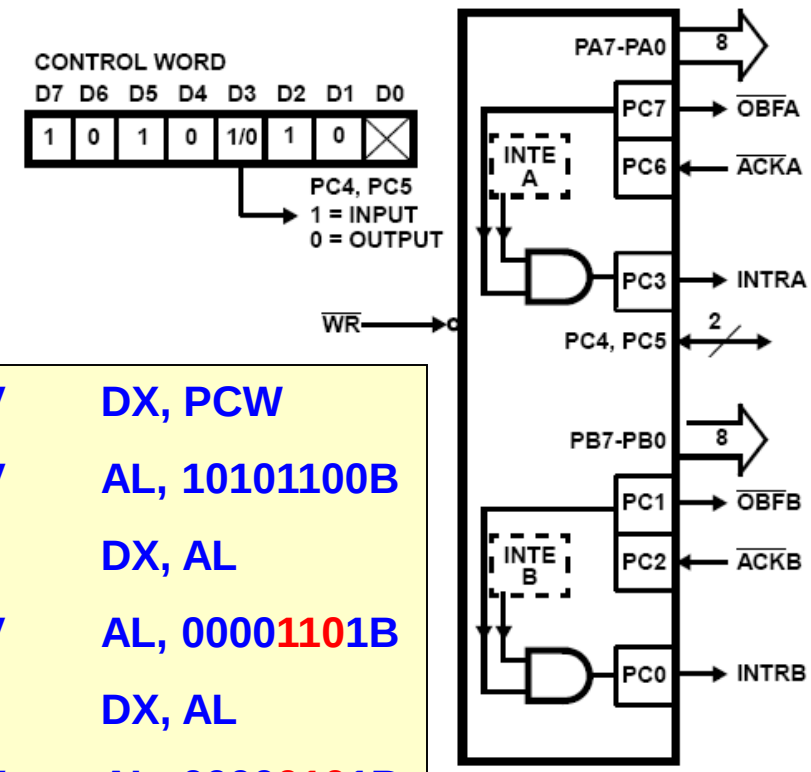
```
;A procedure that reads the keyboard and returns the ASCII key code in AL
        BIT3    EQU     08H         ;00001000 PC3 (INTR_A)
        PORTC   EQU     22H
        PORTA   EQU     20H
        READ    PROC    NEAR
                .REPEAT             ;poll IBF bit
                        MOV  DX, PORTC
                        IN     AL,  DX
                        TEST AL,  BIT3
                .UNTIL !ZERO?       ; Quit polling when bit 3 read is not ZERO (INTR_A=1)
                MOV  DX, PORTA
                IN   AL, DX         ; get ASCII value of key pressed from keyboard
                RET
        READ    ENDP
```



8255 should be programmed for operation in:
- Group A in Mode 1
- Port A is input

Mode 1 (Strobed Output)






```
MOV     DX, PCW
MOV     AL, 10101100B
OUT     DX, AL
MOV     AL, 00001101B
OUT     DX, AL
MOV     AL, 00000101B
OUT     DX, AL
```

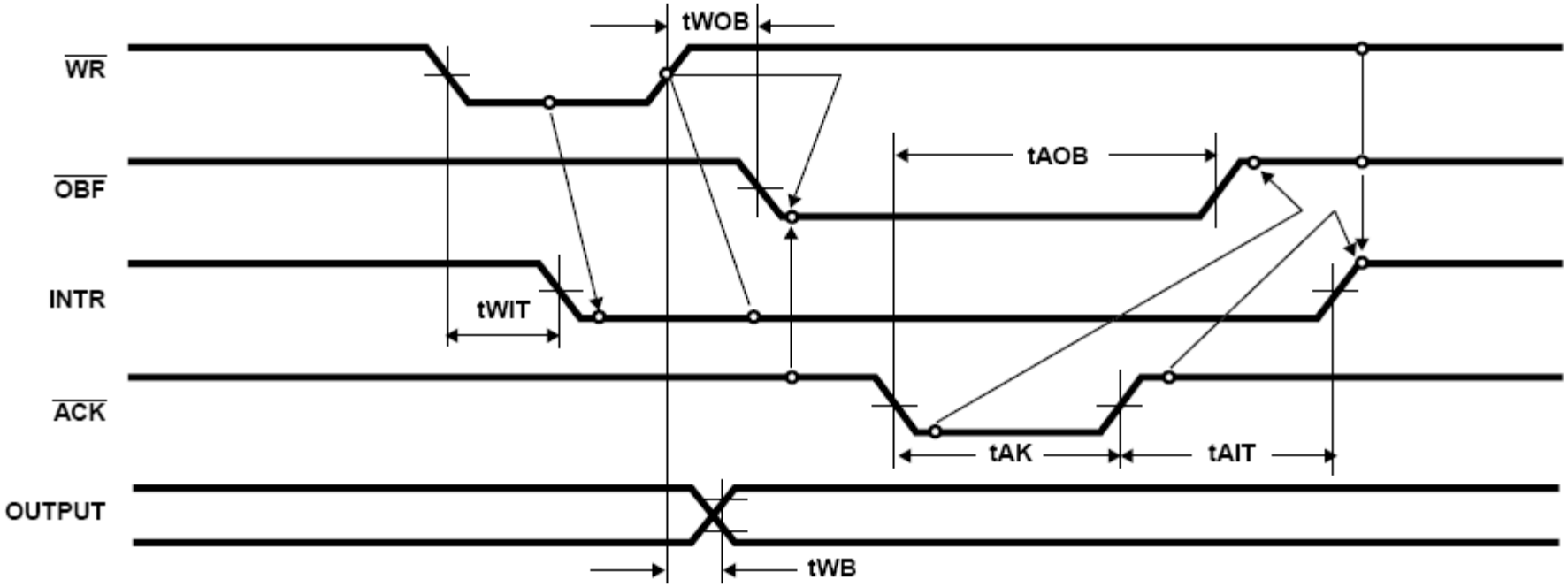
Output Control Signal Definition

- **OBF** – (Output Buffer Full F/F). The OBF output will go “low” to indicate that the CPU has written data out to be specified port. This does not mean valid data is sent out of the part at this time since OBF can go true before data is available. Data is guaranteed valid at the rising edge of OBF. The OBF F/F will be set by the rising edge of the WR input and reset by ACK input being low.

- **ACK** – (Acknowledge Input). A “low” on this input informs the 82C55A that the data from Port A or Port B is ready to be accepted. In essence, a response from the peripheral device indicating that it is ready to accept data.

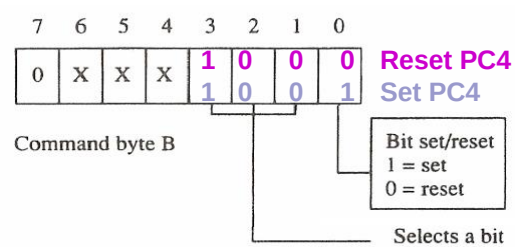
- **INTR** - (Interrupt Request). A “high” on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when ACK is a “one”, OBF is a “one” and INTE is a “one”. It is reset by the falling edge of WR.

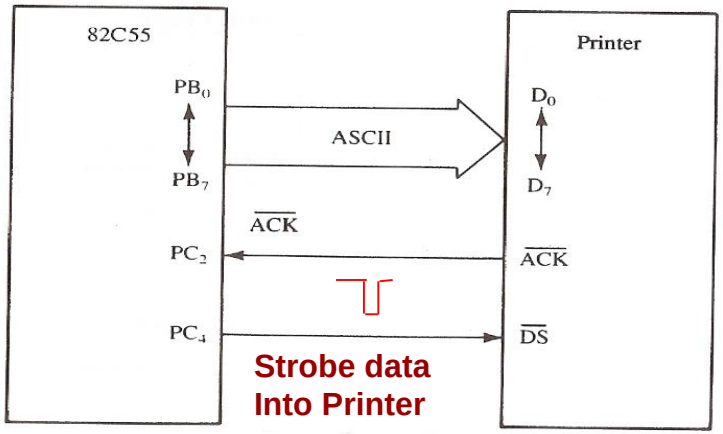
Mode 1 (Strobed Output)



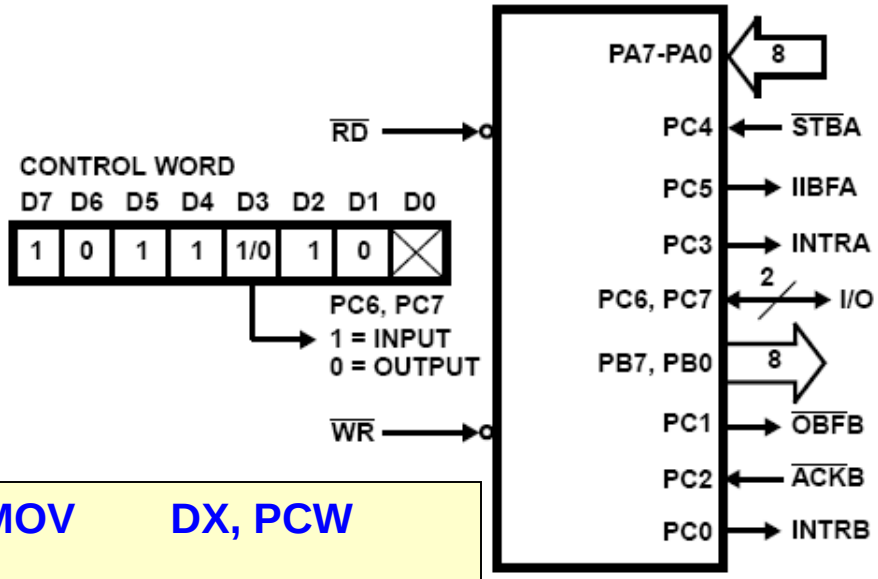
Interfacing a Printer to μP using Port B in Mode 1

```
;A procedure that transfers an ASCII character from AH to the printer connected to port B
BIT0    EQU     1           ; Bit PC0 = INTR_B
PORTC   EQU     62H
PORTB   EQU     61H
CMD     EQU     63H
PRINT   PROC    NEAR
        .REPEAT             ;Wait for printer ready to receive a new char- Poll INTR_B till high
                MOV  DX, PORTC
                IN     AL,  DX
                TEST AL, BIT0
        .UNTIL !ZERO?       ; INTR_B =1 No data in output buffer, so can write into it!
        MOV  DX, PORTB
        MOV  AL, AH
        OUT  DX, AL     ; Write ASCII char data into port latch
        MOV  AL, 8      ;Generate data strobe pulse on PC4
        OUT  CMD, AL
        MOV  AL, 9
        OUT  CMD, AL
        RET
PRINT   ENDP
```

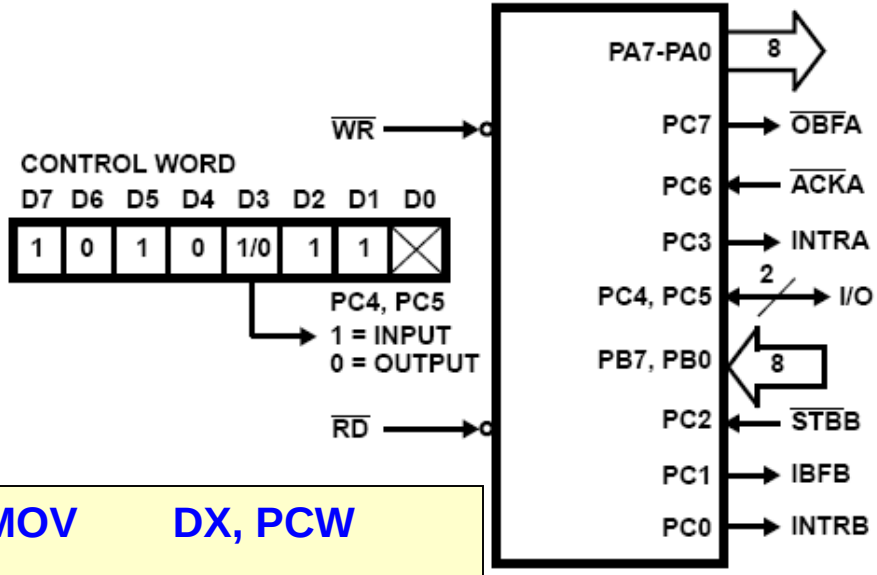




Combinations of Mode 1




```
MOV     DX, PCW
MOV     AL, 10111100B
OUT     DX, AL
MOV     AL, 00001001B
OUT     DX, AL
MOV     AL, 00000101B
OUT     DX, AL
```

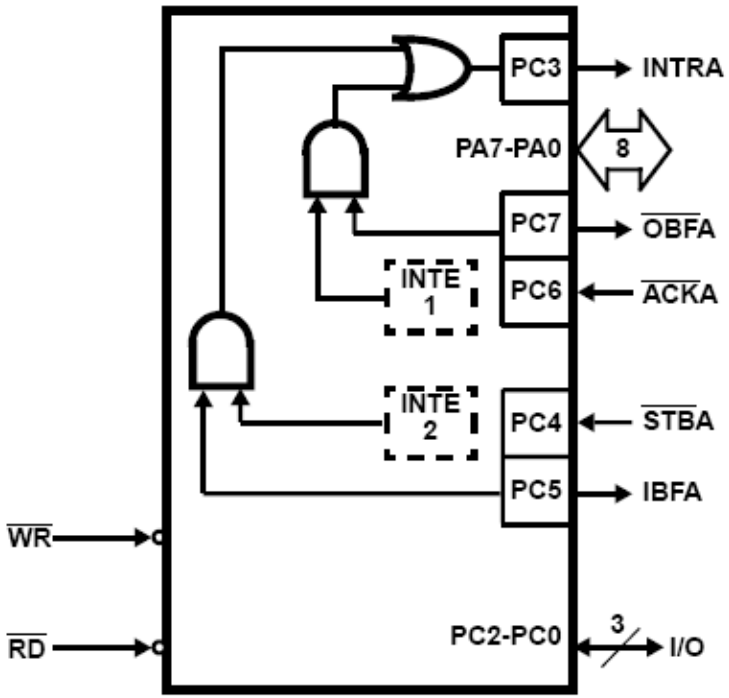


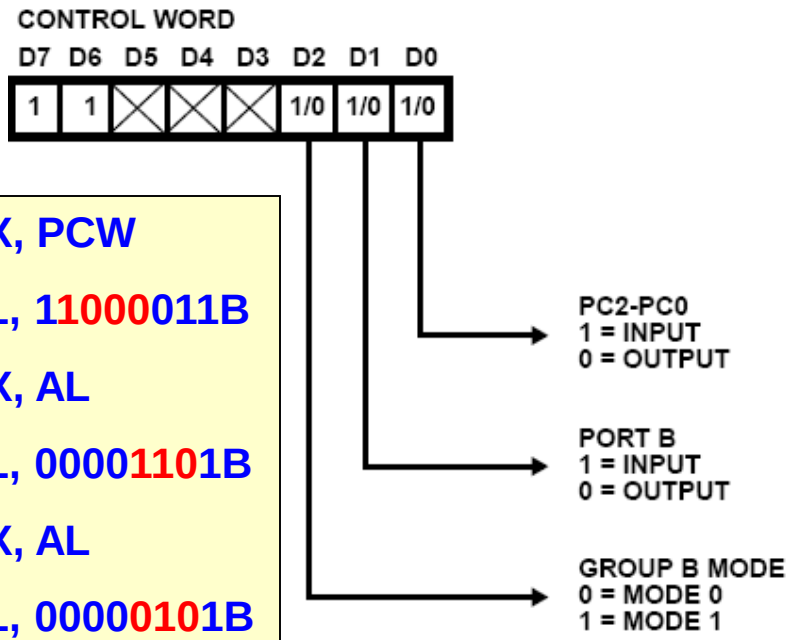

```
MOV     DX, PCW
MOV     AL, 10101110B
OUT     DX, AL
MOV     AL, 00001101B
OUT     DX, AL
MOV     AL, 00000101B
OUT     DX, AL
```

Mode 2 (Strobed Bi-Directional Bus I/O)

- The functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bi-directional bus I/O). “Hand shaking” signals are provided to maintain proper bus flow discipline similar to Mode 1. Interrupt generation and enable/disable functions are also available.
- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Mode 2 (Strobed Bi-Directional Bus I/O)






```
MOV     DX, PCW
MOV     AL, 11000011B
OUT     DX, AL
MOV     AL, 00001101B
OUT     DX, AL
MOV     AL, 00000101B
OUT     DX, AL
```

Bi-Directional Bus I/O Control Signal Definition

- **INTR -** (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

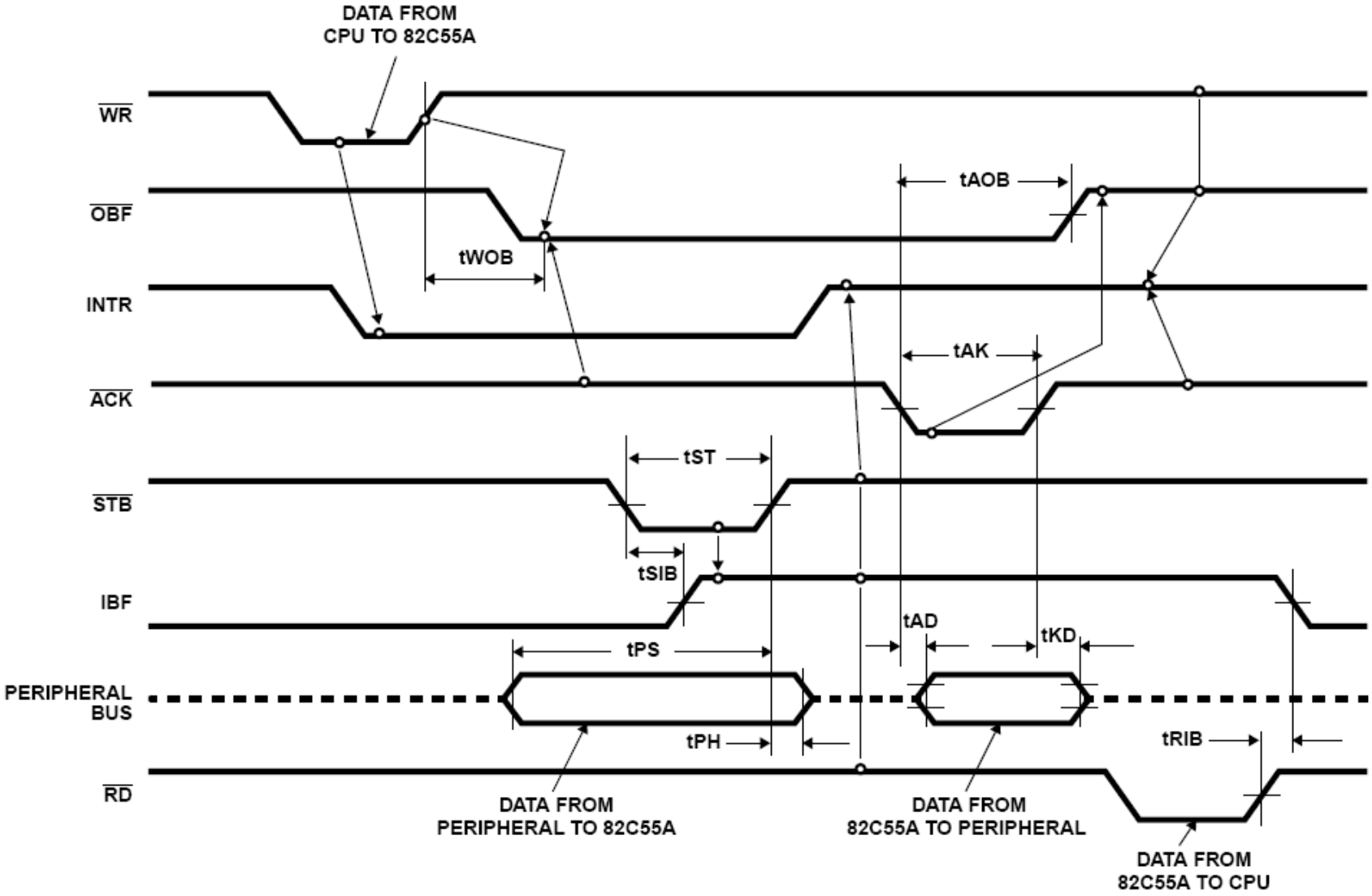
Output Operations:

- **OBF -** (Output Buffer Full). The OBF output will go “low” to indicate that the CPU has written data out to port A.
- **ACK -** (Acknowledge). A “low” on this input enables the three-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.
- **INTE 1** - (The INTE flip-flop associated with OBF). Controlled by bit set/reset of PC4.

Input Operations

- **STB** - (Strobe Input). A “low” on this input loads data into the input latch.
- **IBF** - (Input Buffer Full F/F). A “high” on this output indicates that data has been loaded into the input latch.
- **INTE 2** - (The INTE flip-flop associated with IBF). Controlled by bit set/reset of PC4.

Mode 2 (Strobed Bi-Directional Bus I/O)



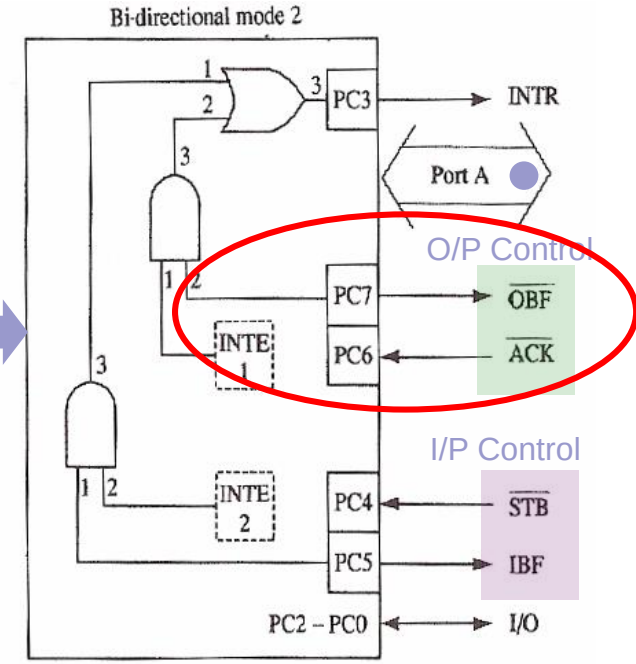
Mode 2 Example: Processor Sends Data to External Device on the Bidirectional Bus

Mode 2 (Strobed Bi-Directi

To send Data from processor to external device:

1. Processor checks if #OBF = 1 (No data pending in port)
2. Processor OUTs data to port (Writes it into Port A latch- not on external bus yet)
3. Port automatically lowers #OBF O/P to alert device to take data
4. Device detects 3 and lowers #ACK input to port
5. This raises #OBF high
6. #ACK enables Port external bus to carry latch data so it can be taken by device
7. After device takes data it raises #ACK high

From µP

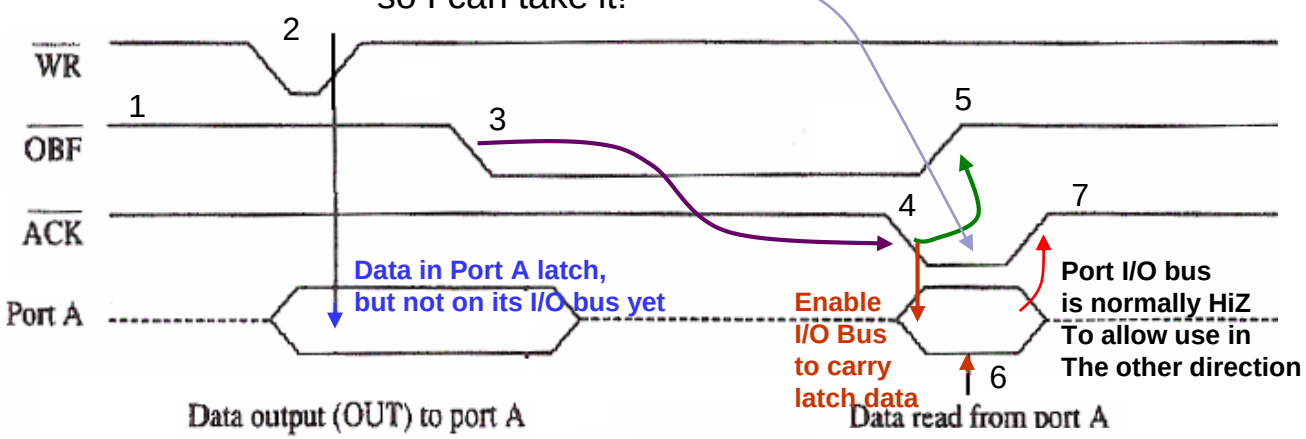


;A procedure transmits AH through the bidirectional bus

```
BIT7 EQU 80H
PORTC EQU 62H
PORTA EQU 60H

TRANSPROC NEAR

        .REPEAT
            IN AL,PORTC
            TEST AL,BIT7
        .UNTIL !ZERO?
        MOV AL,AH
        OUT PORTA,AL
        RET

TRANS   ENDP
```

Wait for #OBF =1
Result = 1 (Not zero)

I understand you have data for me in your latch. Please put it on the bus so I can take it!



By Microprocessor

By external device
(May need a strobe pulse from µP)

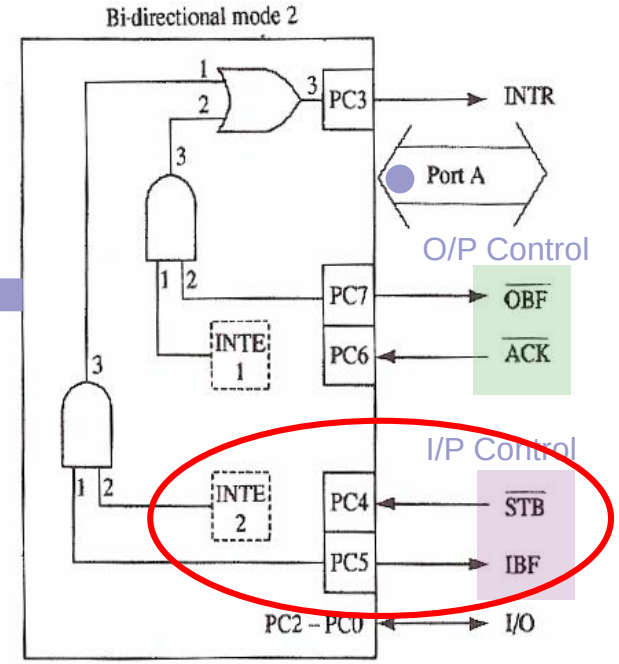
Mode 2 Example: Processor Receives Data from External Device on the Bidirectional Bus

Mode 2 (Strobed Bi-Directi

To Receive Data:

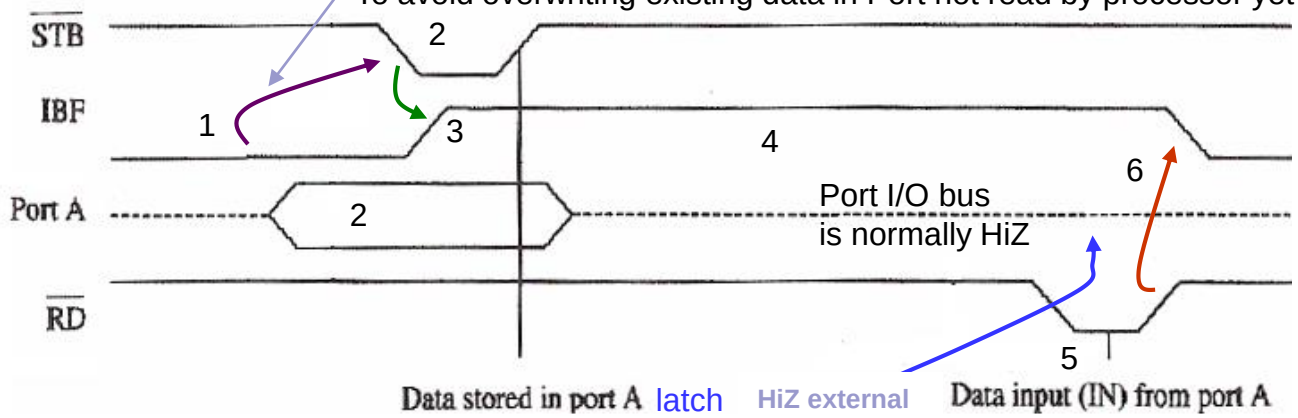
1. External device sending data checks if #IBF = 0 (No pending data in port latch not read by processor) (Hardware Polling)
2. Then it puts its data on external bus and strobes it into port latch using #STB
3. IBF automatically goes high until data is read by processor
4. Processor polls IBF for IBF = 1 to make sure data is in port latch (software polling)
5. Processor reads data from port
6. This automatically lowers IBF to enable further writes

To µP

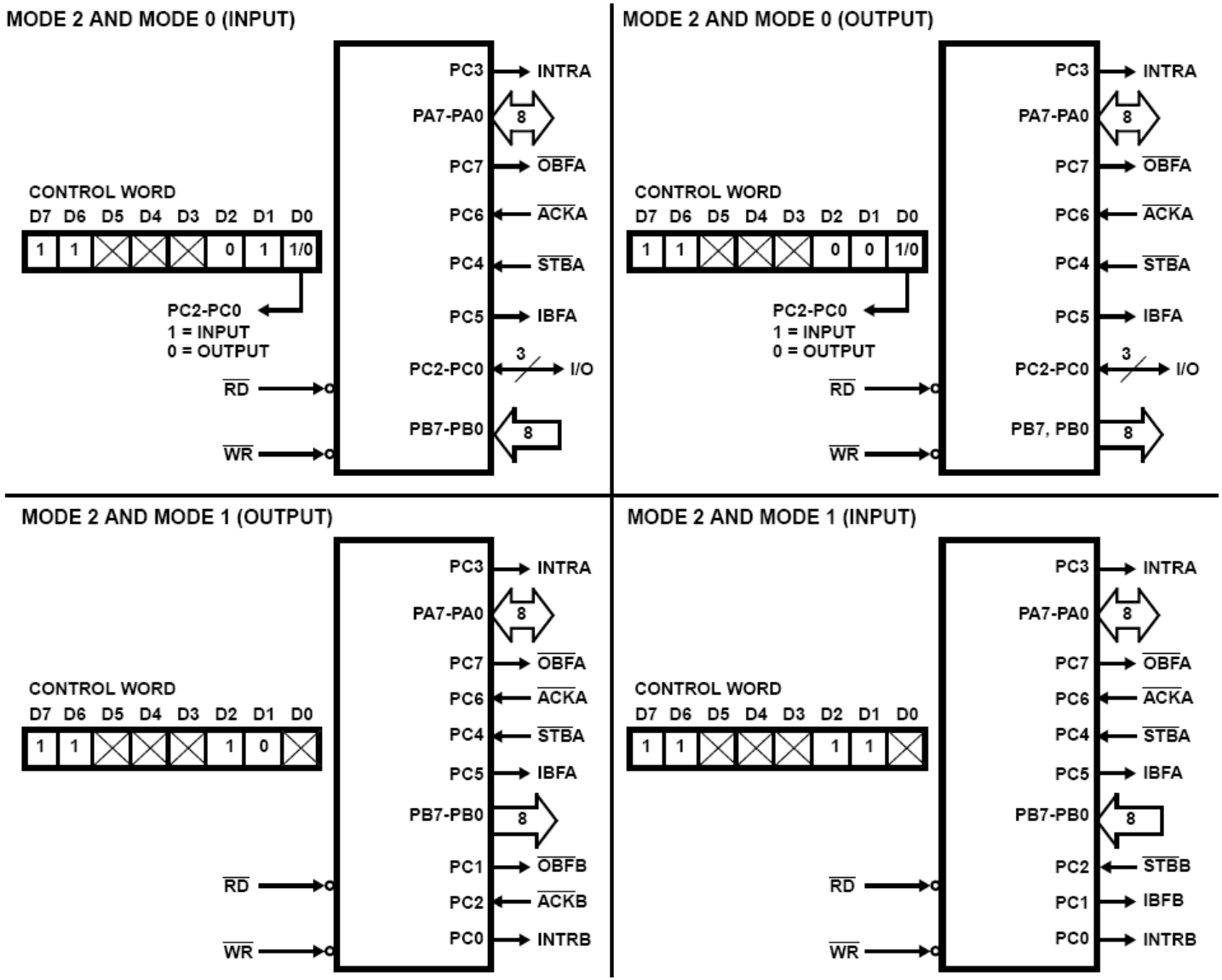



```
;A procedure that reads data from the bidirectional bus into AL

BIT5 EQU 20H
PORTC EQU 62H
PORTA EQU 60H

READ PROC NEAR
        .REPEAT  ;Wait for IBF = 1
        IN AL,PORTC
Step 4  TEST AL,BIT5
        .UNTIL !ZERO?
        IN AL,PORTA
        RET
                Step 5

READ PROC NEAR
```

Hardware Polling for IBF = 0 by device to generate #STB
To avoid overwriting existing data in Port not read by processor yet



Mode 2 Combinations



Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA0	In	Out	In	Out	↔	
PA1	In	Out	In	Out	↔	
PA2	In	Out	In	Out	↔	
PA3	In	Out	In	Out	↔	
PA4	In	Out	In	Out	↔	
PA5	In	Out	In	Out	↔	
PA6	In	Out	In	Out	↔	
PA7	In	Out	In	Out	↔	
PB0	In	Out	In	Out		Mode 0 or Mode 1 Only
PB1	In	Out	In	Out		
PB2	In	Out	In	Out		
PB3	In	Out	In	Out		
PB4	In	Out	In	Out		
PB5	In	Out	In	Out		
PB6	In	Out	In	Out		
PB7	In	Out	In	Out		
PC0	In	Out	INTRB	INTRB	I/O	
PC1	In	Out	IBFB	$\overline{OBF}$B	I/O	
PC2	In	Out	$\overline{STB}$B	$\overline{ACK}$B	I/O	
PC3	In	Out	INTRA	INTRA	INTRA	
PC4	In	Out	$\overline{STB}$A	I/O	$\overline{STB}$A	
PC5	In	Out	IBFA	I/O	IBFA	
PC6	In	Out	I/O	$\overline{ACK}$A	$\overline{ACK}$A	
PC7	In	Out	I/O	$\overline{OBF}$A	$\overline{OBF}$A	

Status Word

INPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	IBFA	INTEA	INTRA	INTEB	IBFB	INTRB
GROUP A					GROUP B		

OUTPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
OBFA	INTEA	I/O	I/O	INTRA	INTEB	OBFB	INTRB
GROUP A					GROUP B		

MODE 1 STATUS WORD FORMAT

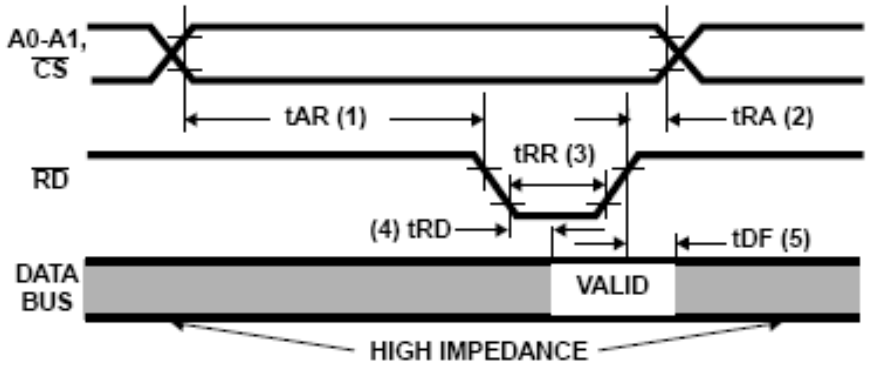
D7	D6	D5	D4	D3	D2	D1	D0
OBFA	INTE1	IBFA	INTE2	INTRA	X	X	X
GROUP A					GROUP B		

(Defined by Mode 0 or Mode 1 Selection)

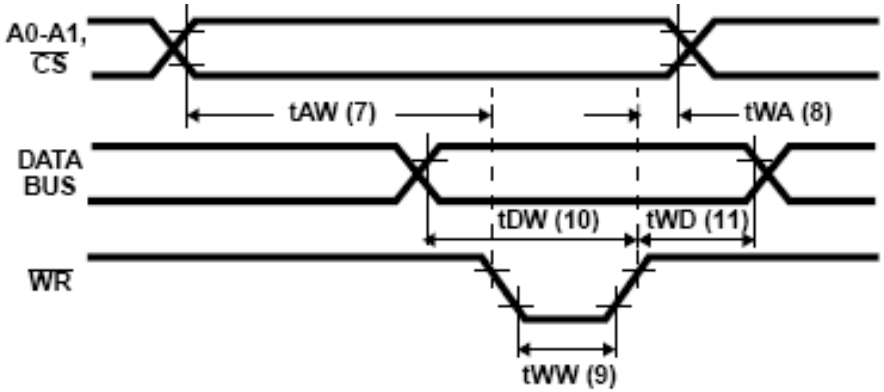
MODE 2 STATUS WORD FORMAT

SYMBOL	PARAMETER	82C55A-5		82C55A		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
READ TIMING							
(1) tAR	Address Stable Before $\overline{RD}$	0	-	0	-	ns	
(2) tRA	Address Stable After $\overline{RD}$	0	-	0	-	ns	
(3) tRR	$\overline{RD}$ Pulse Width	250	-	150	-	ns	
(4) tRD	Data Valid From $\overline{RD}$	-	200	-	120	ns	1
(5) tDF	Data Float After $\overline{RD}$	10	75	10	75	ns	2
(6) tRV	Time Between $\overline{RD}$s and/or $\overline{WR}$s	300	-	300	-	ns	
WRITE TIMING							
(7) tAW	Address Stable Before $\overline{WR}$	0	-	0	-	ns	
(8) tWA	Address Stable After $\overline{WR}$	20	-	20	-	ns	
(9) tWW	$\overline{WR}$ Pulse Width	100	-	100	-	ns	
(10) tDW	Data Valid to $\overline{WR}$ High	100	-	100	-	ns	
(11) tWD	Data Valid After $\overline{WR}$ High	30	-	30	-	ns	
OTHER TIMING							
(12) tWB	$\overline{WR}$ = 1 to Output	-	350	-	350	ns	1
(13) tIR	Peripheral Data Before $\overline{RD}$	0	-	0	-	ns	
(14) tHR	Peripheral Data After $\overline{RD}$	0	-	0	-	ns	
(15) tAK	ACK Pulse Width	200	-	200	-	ns	
(16) tST	STB Pulse Width	100	-	100	-	ns	
(17) tPS	Peripheral Data Before STB High	20	-	20	-	ns	
(18) tPH	Peripheral Data After STB High	50	-	50	-	ns	
(19) tAD	ACK = 0 to Output	-	175	-	175	ns	1
(20) tKD	ACK = 1 to Output Float	20	250	20	250	ns	2
(21) tWOB	$\overline{WR}$ = 1 to OBF = 0	-	150	-	150	ns	1
(22) tAOB	ACK = 0 to OBF = 1	-	150	-	150	ns	1
(23) tSIB	STB = 0 to IBF = 1	-	150	-	150	ns	1
(24) tRIB	$\overline{RD}$ = 1 to IBF = 0	-	150	-	150	ns	1
(25) tRIT	$\overline{RD}$ = 0 to INTR = 0	-	200	-	200	ns	1
(26) tSIT	STB = 1 to INTR = 1	-	150	-	150	ns	1
(27) tAIT	ACK = 1 to INTR = 1	-	150	-	150	ns	1
(28) tWIT	$\overline{WR}$ = 0 to INTR = 0	-	200	-	200	ns	1
(29) tRES	Reset Pulse Width	500	-	500	-	ns	1, (Note)

Timing Parameters

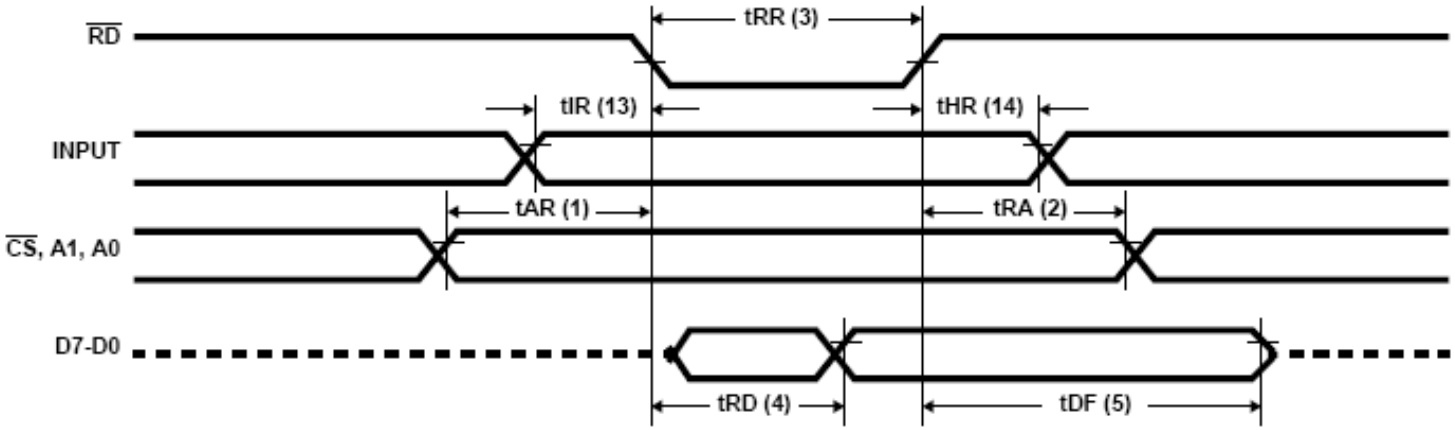


READ TIMING

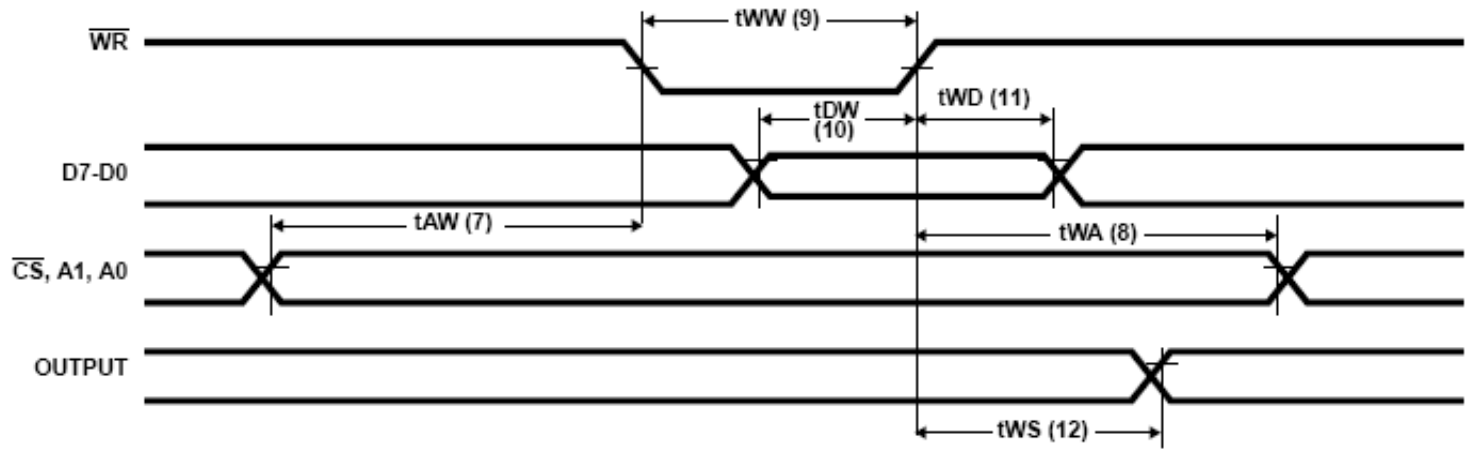


WRITE TIMING

Mode 0 Timing Parameters

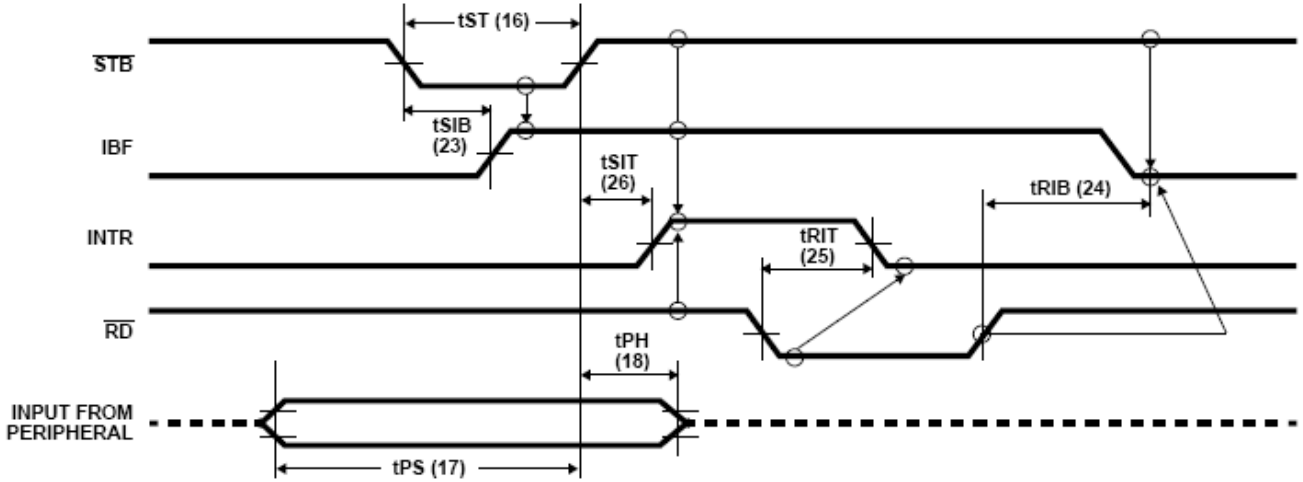


MODE 0 (BASIC INPUT)

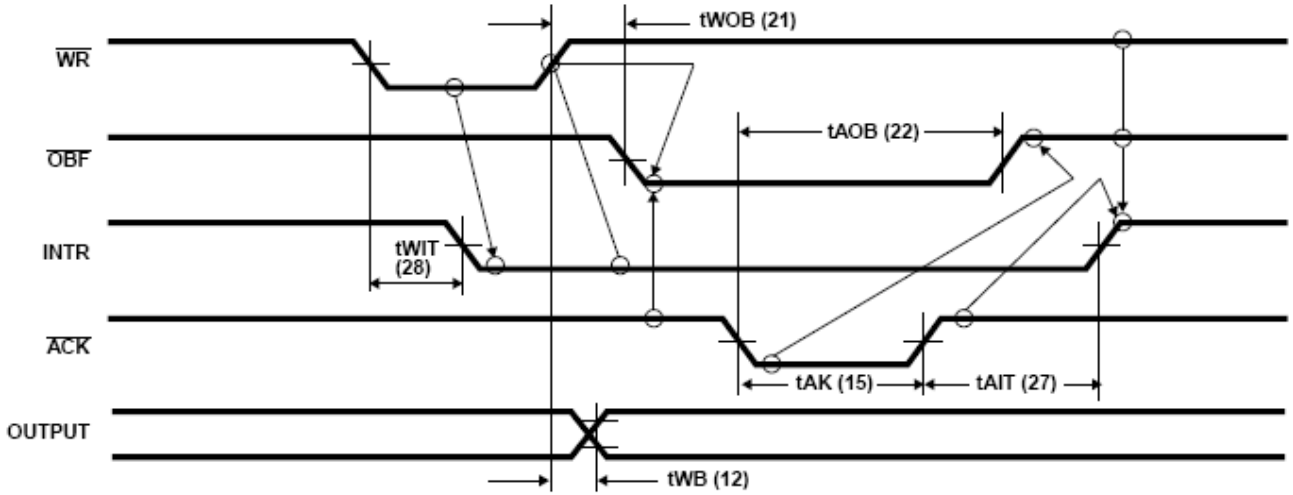


MODE 0 (BASIC OUTPUT)

Mode 1 Timing Parameters

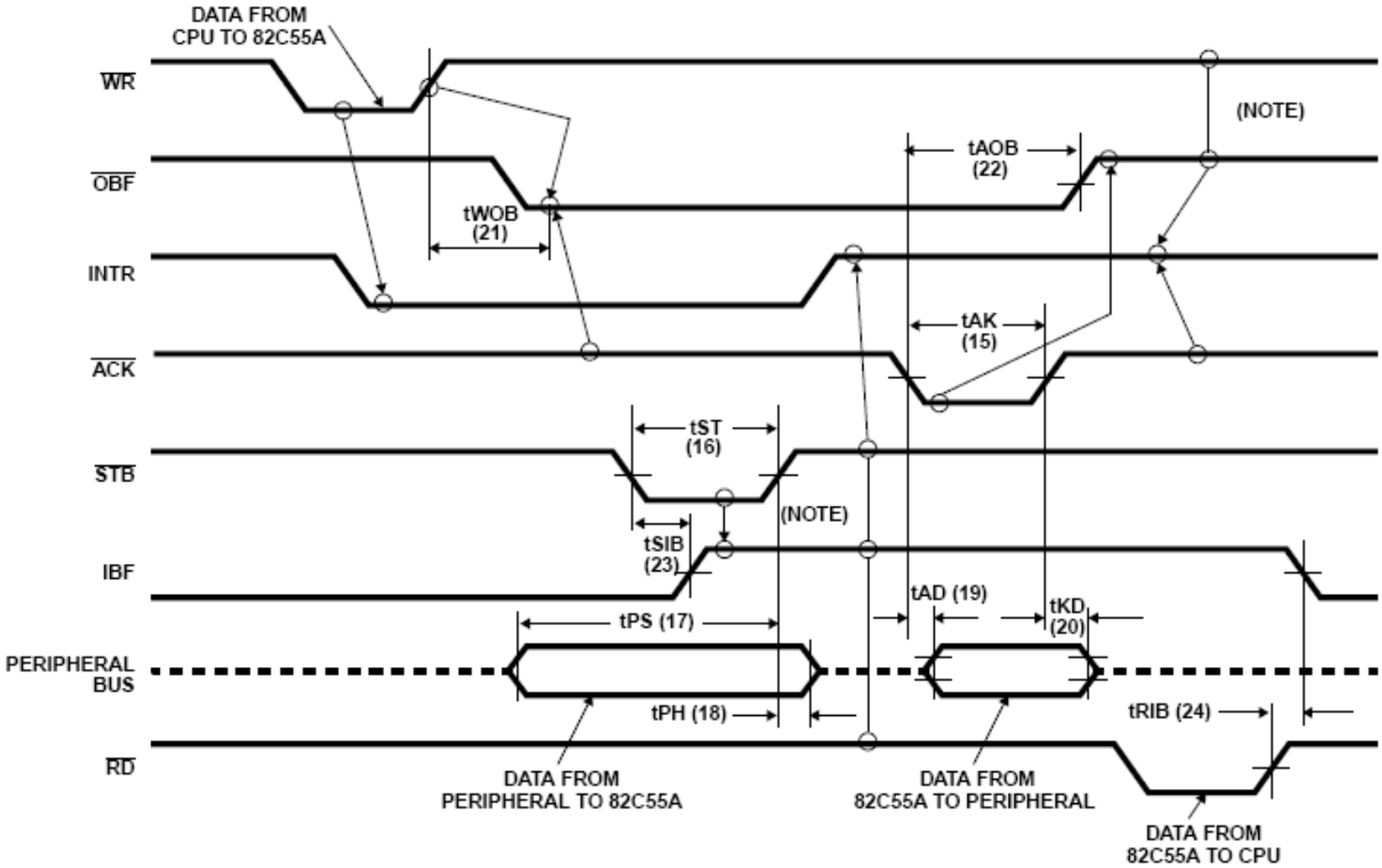


MODE 1 (STROBED INPUT)



MODE 1 (STROBED OUTPUT)

Mode 2 Timing Parameters



MODE 2 (BI-DIRECTIONAL)