



MICROPROCESSOR AND INTERFACING

CSE2006

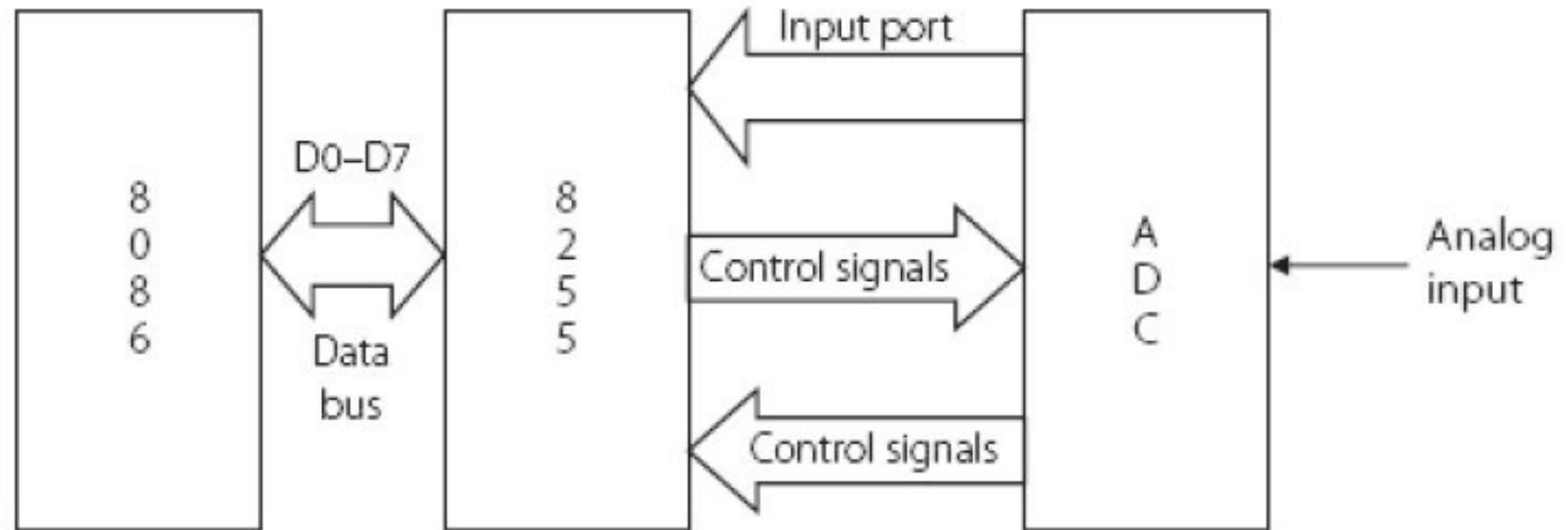
Dr. Arvind Kumar
| School of Electronics Engineering, VIT Vellore |



- ANALOG-TO-DIGITAL (ADC)
- DIGITAL-TO-ANALOG (DAC) CONVERTERS

ADC (Analog-to Digital Converter)

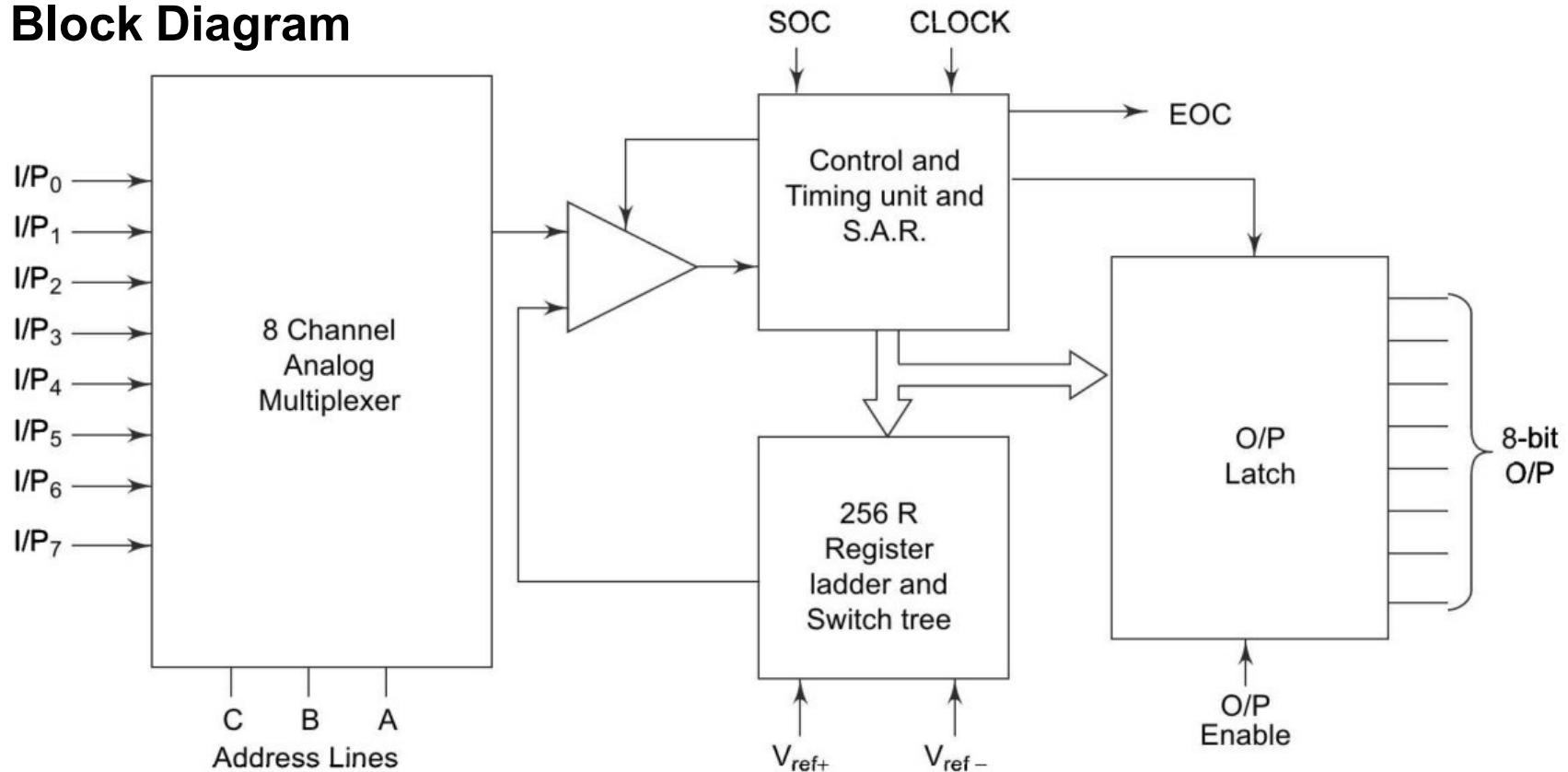
Interface with 8086



SOC: Start of Conversion
EOC : End of Conversion

ADC 0808 / 0809

Block Diagram



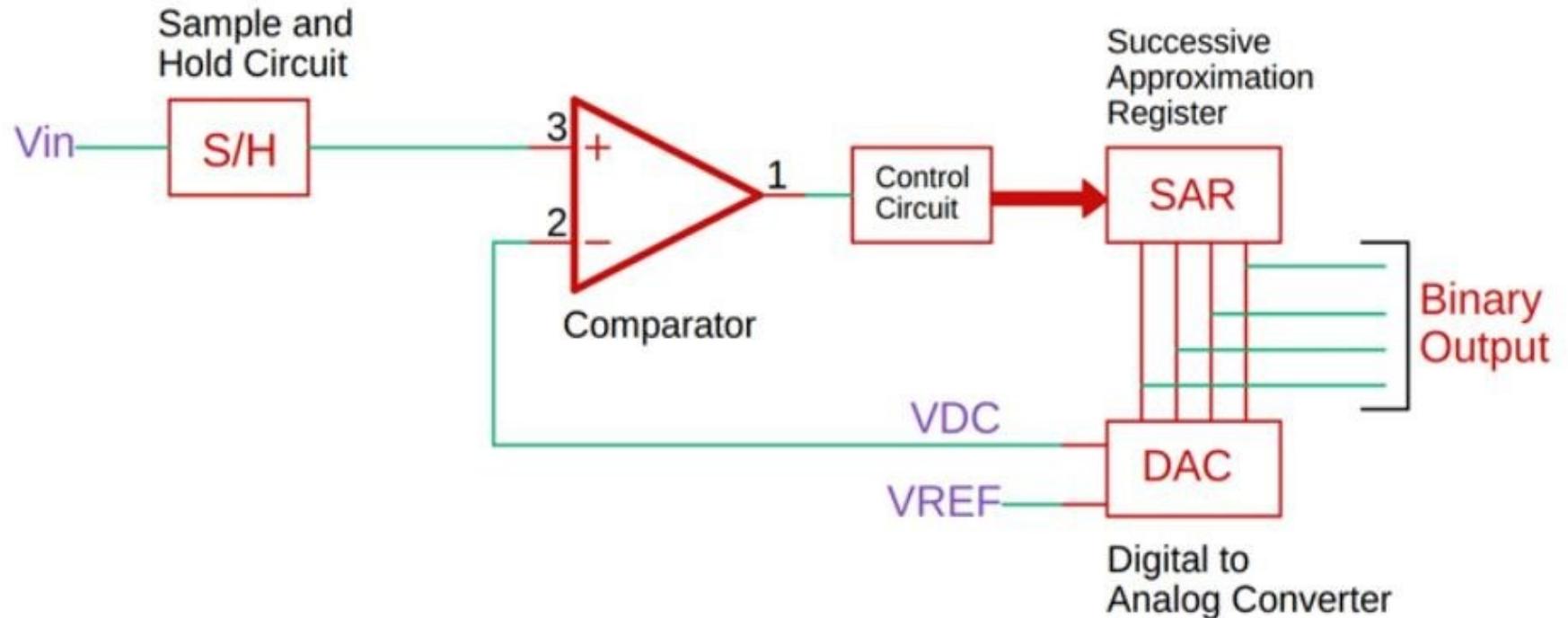
Low-cost ADC, Power 15 mW,
 Compatible with a wide range of microprocessors. Power Supply 5 V
 Moderate speed 100 μ s
 Moderate accuracy Error \pm LSB

ADC 0808 / 0809

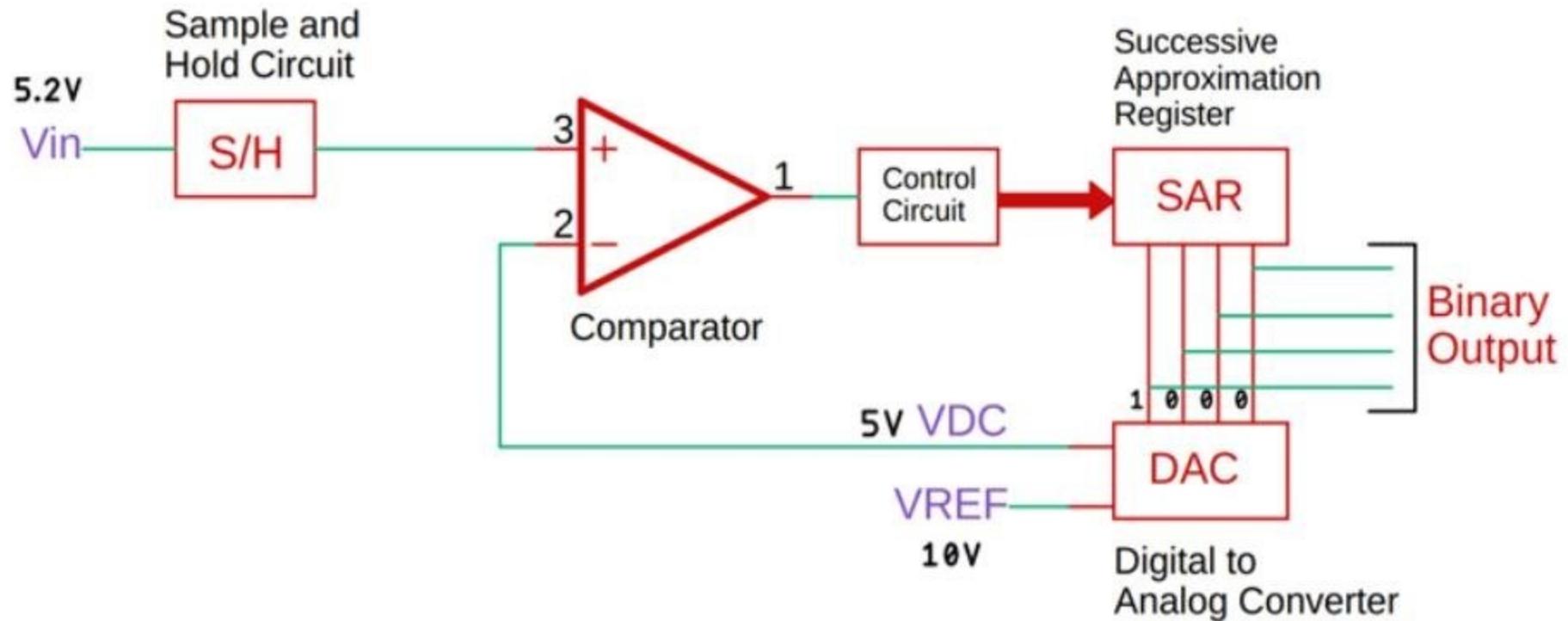
Pin Diagram

I/P ₃ →	1	28 ← I/P ₂	
I/P ₄ →	2	27 ← I/P ₁	
I/P ₅ →	3	26 ← I/P ₀	I/P ₀ –I/P ₇ Analog inputs
I/P ₆ →	4	25 ← ADD A	ADD A, B, C Address lines for selecting analog inputs
I/P ₇ →	5	24 ← ADD B	O ₇ –O ₀ Digital 8-bit output with O ₇ MSB and O ₀ LSB
SOC →	6	23 ← ADD C	SOC Start of conversion signal pin
EOC →	7	22 ← ALE	EOC End of conversion signal pin
O ₃ →	8 ADC 0808	21 ← O ₇ MSB	OE Output latch enable pin, if high enable output
OE →	9	20 ← O ₆	CLK Clock input for ADC
CLK →	10	19 ← O ₅	V _{CC} , GND Supply pins +5V and GND
V _{CC} →	11	18 ← O ₄	V _{ref+} and V _{ref-} Reference voltage positive (+5 Volts maximum)
V _{ref+} →	12	17 ← O ₀ LSB	and Reference voltage negative (0V minimum)
GND →	13	16 ← V _{ref-}	
O ₁ →	14	15 ← O ₂	

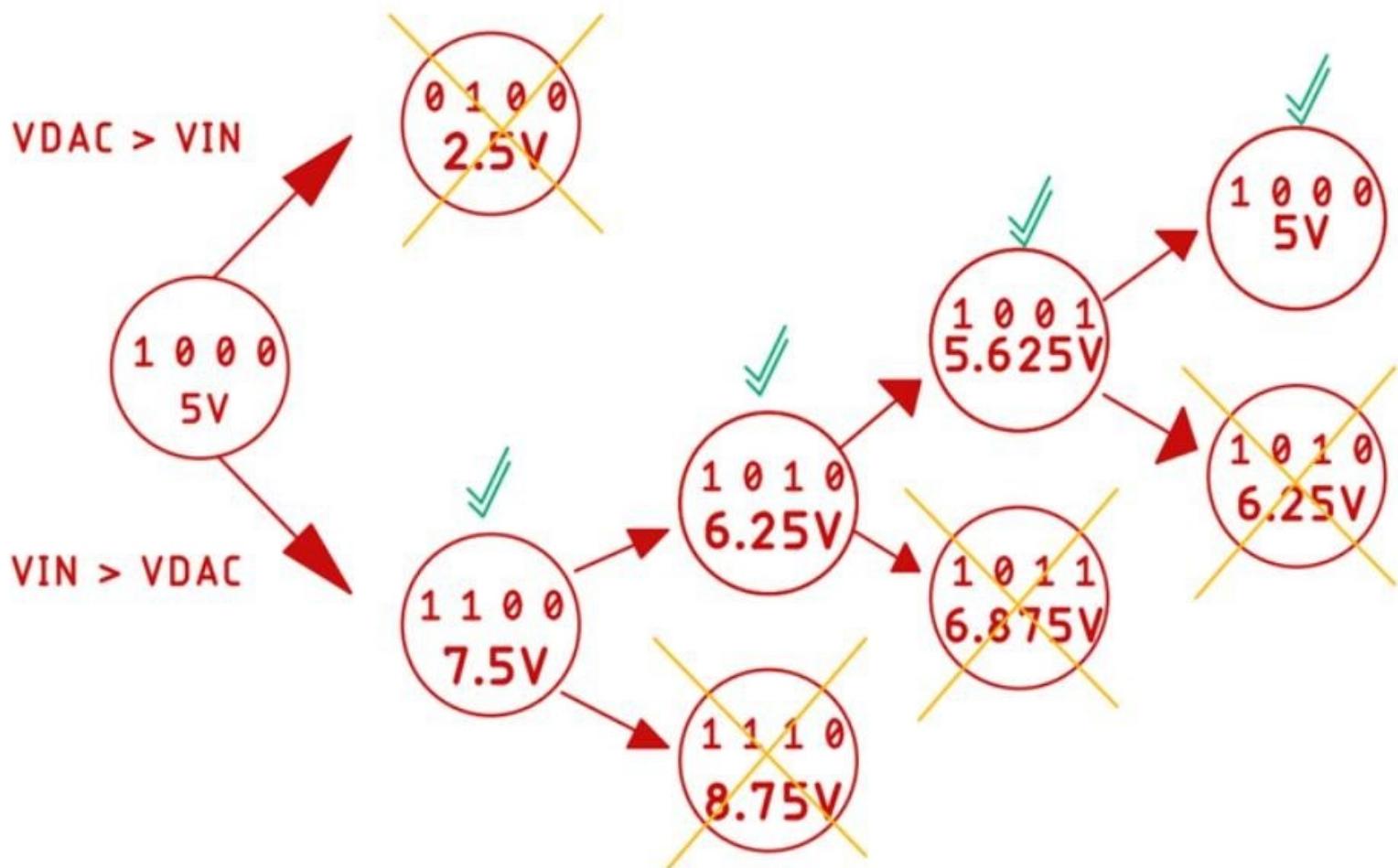
Successive Approximation ADC



Successive Approximation ADC

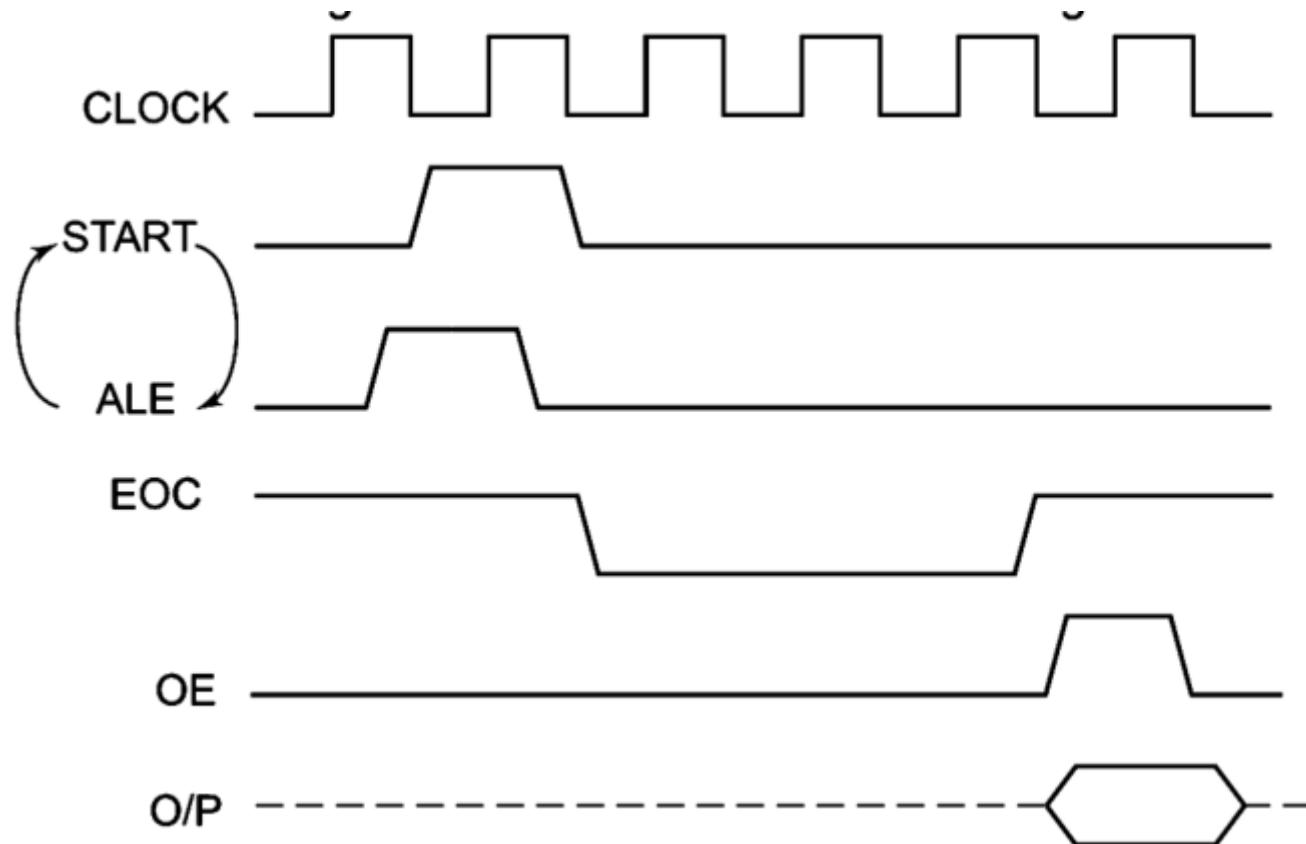


Successive Approximation ADC



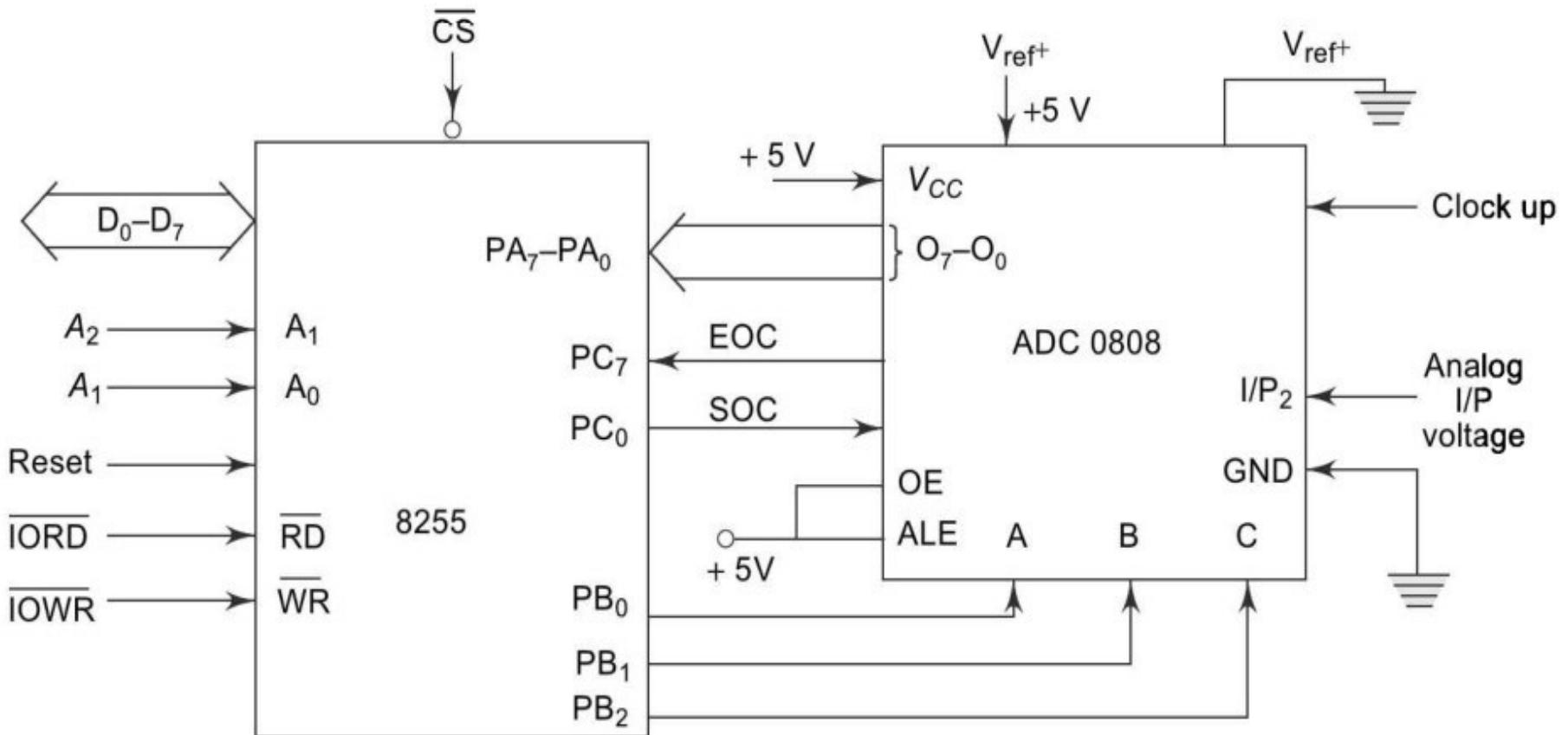
ADC 0808 / 0809

Timing Diagram



ADC 0808 / 0809

Interface ADC 0808 with 8086 using 8255 ports. Use Port A of 8255 for transferring digital data output of ADC to the CPU and Port C for control signals. Assume that an analog input is present at I/P₂ of the ADC and a clock input of suitable frequency is available for ADC. Draw the schematic and write required ALP.





ADC 0808 / 0809

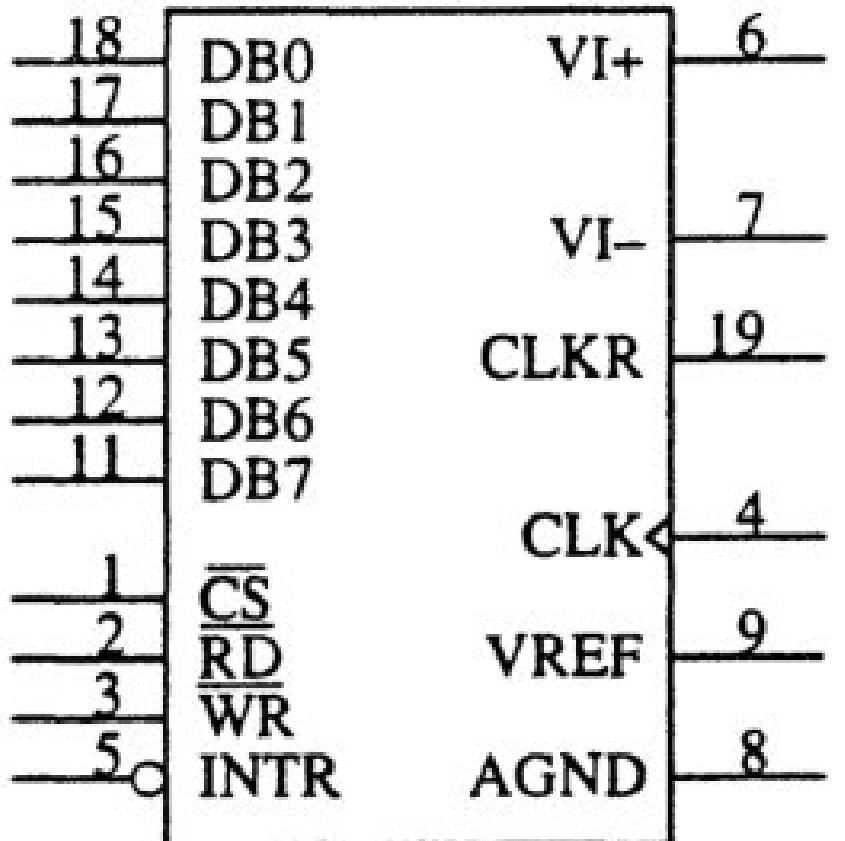
Interface ADC 0808 with 8086 using 8255 ports. Use Port A of 8255 for transferring digital data output of ADC to the CPU and Port C for control signals. Assume that an analog input is present at I/P₂ of the ADC and a clock input of suitable frequency is available for ADC. Draw the schematic and write required ALP.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Control word
1	0	0	1	1	0	0	0	= 98 H D7=1; I/O Mode. D6=0 and D5=0; Port A Mode

The required ALP is given as follows:

```
MOV AL,98 H           ; Initialise 8255 as       .  
OUT CWR,AL           ; discussed above        D4=1; Port A is input port  
MOV AL,02H            ; Select I/P2 as analog    D3=1; Port C (Upper) is input  
OUT PORT B,AL         ; input                         D2=0; Port B Mode 0.  
MOV AL,00H            ; Give start of conversion D0=0; Port C (Lower) is output  
OUT PORT C,AL         ; pulse to the ADC.          D1=0; Port B is output  
MOV AL,01 H           ;  
OUT PORT C,AL         ;  
MOV AL,00H            ;  
OUT PORT C,AL         ;  
WAIT :   IN AL,PORTC      ; Check for EOC by       .  
          RCL                ; reading port C upper and  
          JNC WAIT             ; rotating through carry.  
          IN AL,PORTA          ; If EOC, read digital equivalent in  
                                ; AL  
          HLT                 ; Stop
```

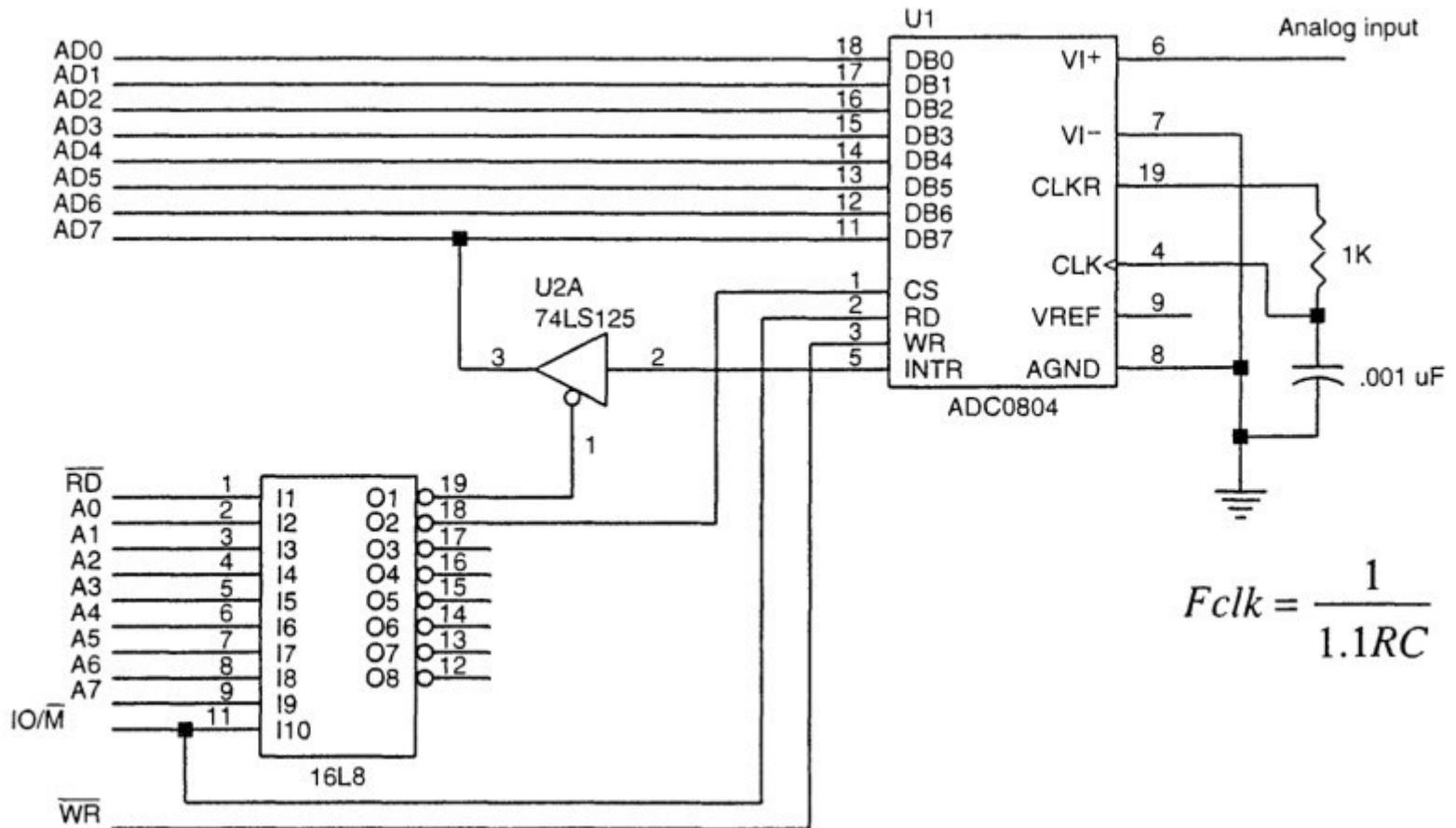
ADC 0804



ADC0804

SOC: Start of Conversion WR' and CS'
EOC : End of Conversion INTR

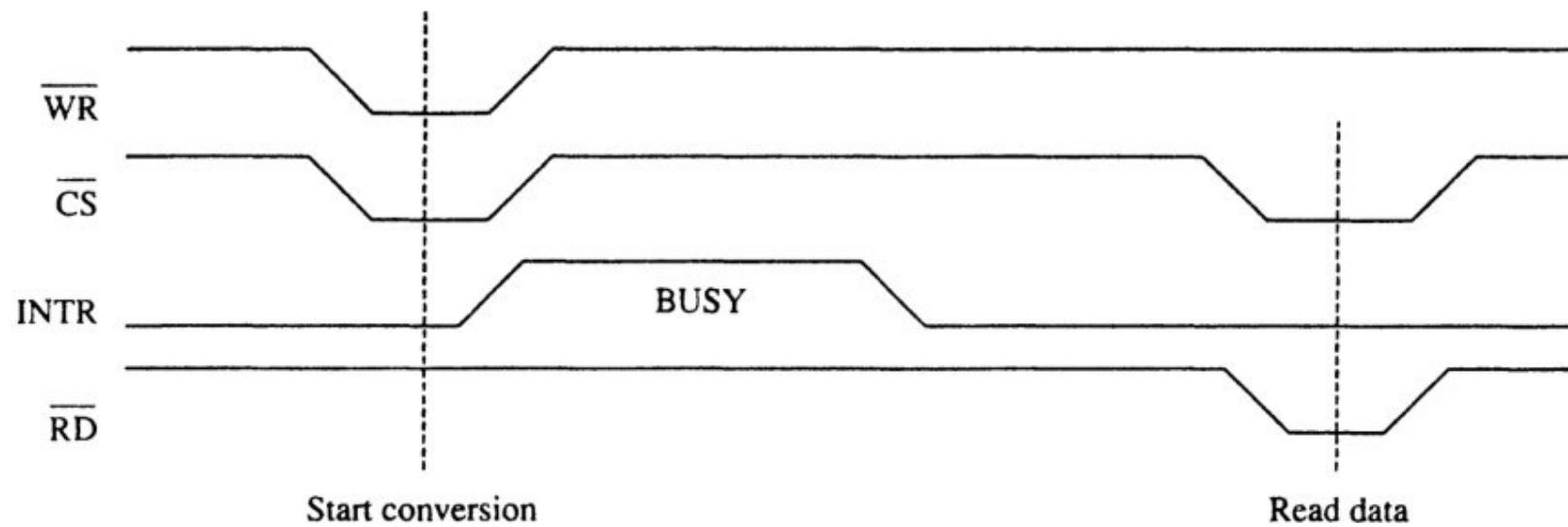
ADC 0804



Permissible range of clock frequencies is 100 KHz - 1460 KHz.
 desirable to use a frequency as close as possible to 1460 KHz so conversion time is minimized

ADC 0804

Timing Diagram





DAC0830

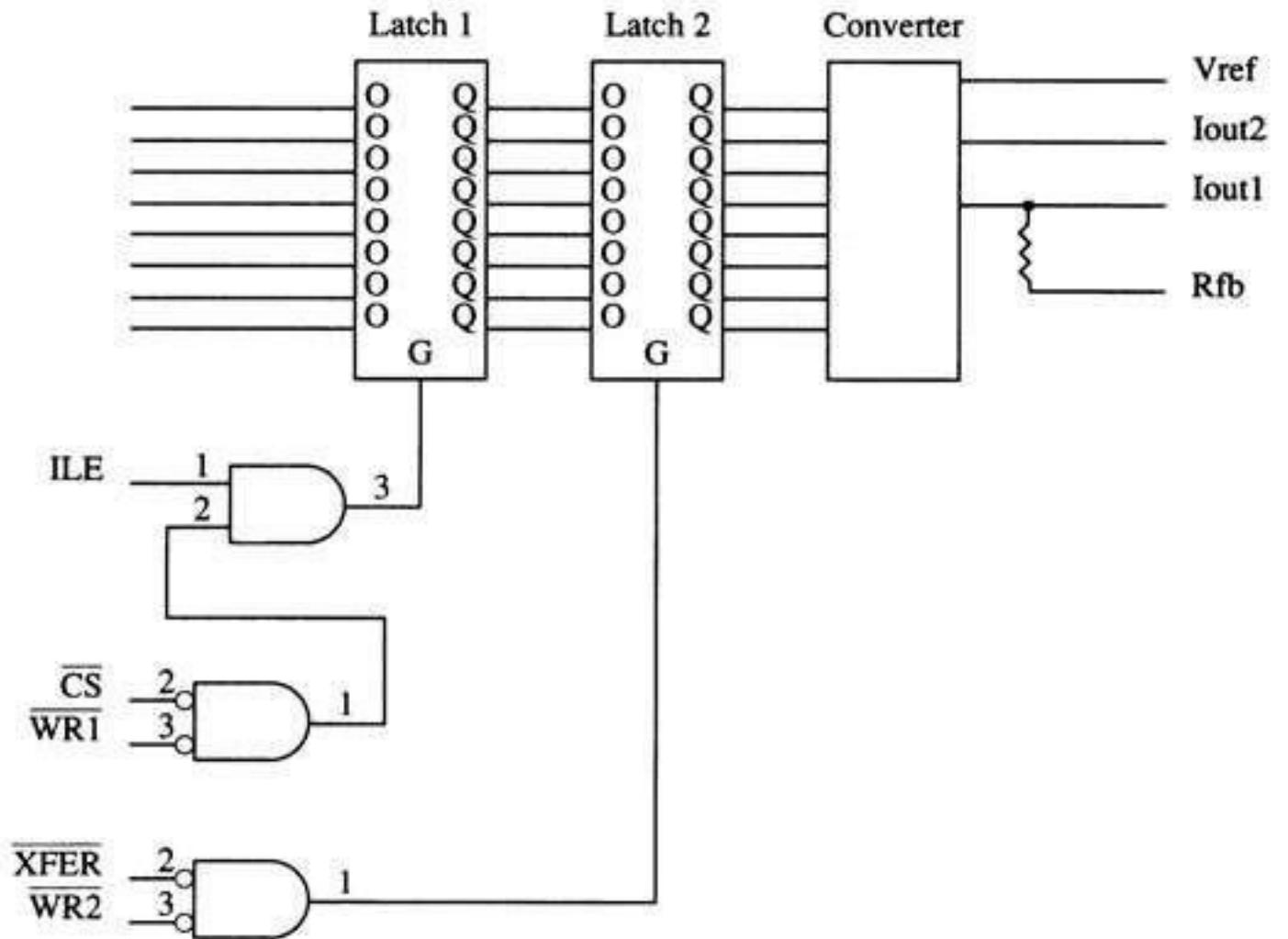
- A fairly common and low-cost digital-to-analog converter is the DAC0830.
- An 8-bit converter that transforms an 8-bit binary number into an analog voltage.
- Other converters are available that convert from 10-, 12-, or 16-bit binary numbers into analog voltages.



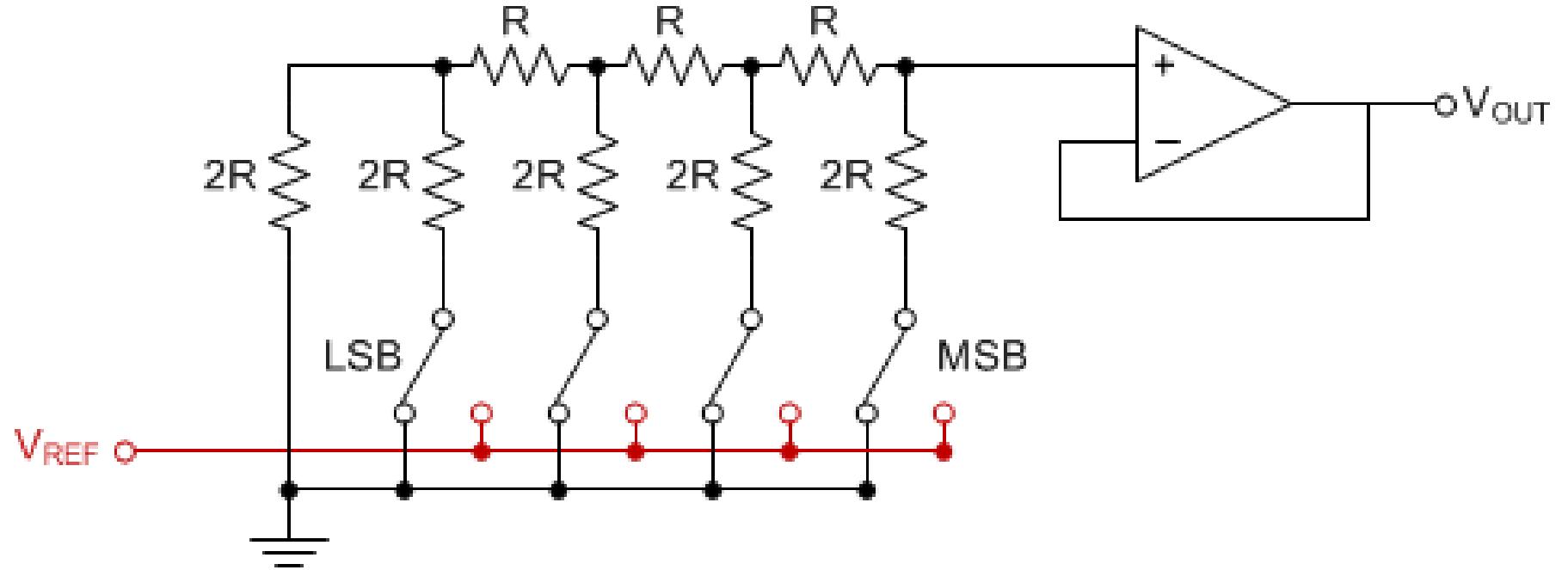
DAC0830

- The number of voltage steps generated by the converter is equal to the number of binary input combinations.
 - an 8-bit converter generates 256 voltage levels
 - a 10-bit converter generates 1024 levels
- The DAC0830 is a medium-speed converter that transforms a digital input to an analog output in approximately $1.0 \mu\text{s}$.

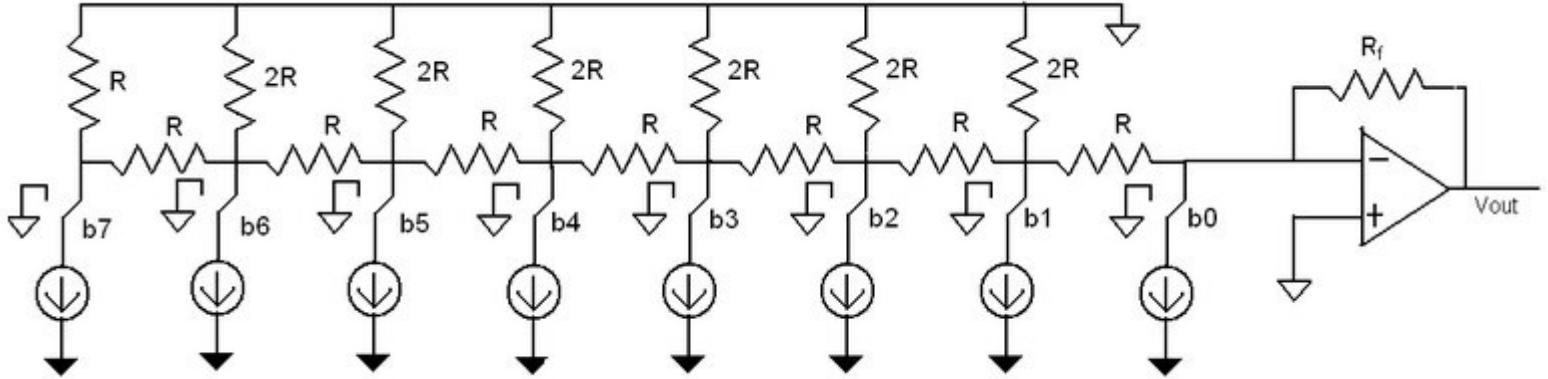
DAC0830



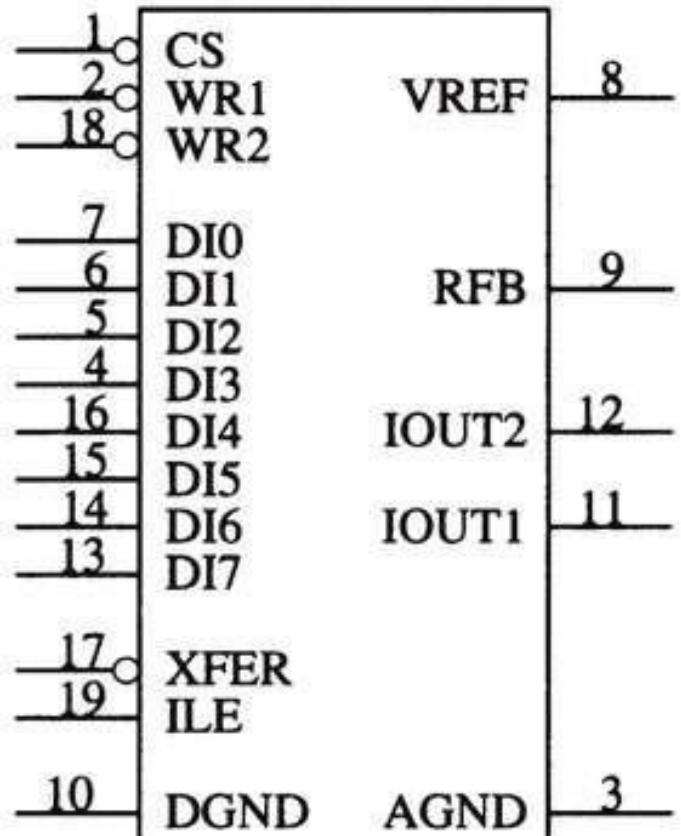
R-2R Ladder DAC



R-2R Ladder DAC



DAC0830



DAC0830

Because this is an 8-bit converter, its output step voltage is defined as $-V_{REF}$ (reference voltage), divided by 255. The step voltage is often called the resolution of the converter.



interfacing Adc with 8086

- ❖ In most cases, **PPI 8255** is used to interface the ADC to the microprocessor
- ❖ ADC is considered as **input device** to the microprocessor that sends an initializing signal to the ADC to start the analog signal conversion
- ❖ There is a pulse which is called the **start of conversion (SOC)** signal
- ❖ A2D conversion is a **slow process** and the microprocessor has to wait until the conversion is over.
- ❖ Once the conversion is over, the **end of signal (EOS)** is sent to inform the microprocessor about the same and the result is ready at the output buffer of ADC



Continued

- ❖ Issuing SOC, reading EOC, getting the results are all done by PPI 8255
- ❖ The time taken by the ADC for the conversion of the analog signal to digital signal is called **conversion delay**
- ❖ It can range from a few **microseconds** to a few hundred milli seconds
- ❖ Selection of appropriate ADC – speed, resolution and cost factor.



General algorithm for interfacing

- ❖ Ensure the stability of the analog input given to the ADC
- ❖ Issue SOC of ADC
- ❖ Read EOC to mark the end of conversion
- ❖ Read the digital output from the output buffer of ADC
- ❖ The analog input must be a constant value from the start to end of conversion process
- ❖ This is done using a sampling and hold circuit
- ❖ The 8086 gives the hold signal to the sample and hold circuit



ADC 0808/0809

- ❖ Successive optimization converters – one of the fastest techniques
- ❖ Conversion delay is **100 microsecs for 640kHz**
- ❖ Do not need an external zero or full scale adjustments
- ❖ 8 input analog signals can be provided
- ❖ Out of these 8 inputs, only 1 input can be selected for conversion by using address lines ADD A, ADD B and ADD C

Address lines for conversion

Analog I/P selected	Address lines		
	C	B	A
I/P 0	0	0	0
I/P 1	0	0	1
I/P 2	0	1	0
I/P 3	0	1	1
I/P 4	1	0	0
I/P 5	1	0	1
I/P 6	1	1	0
I/P 7	1	1	1

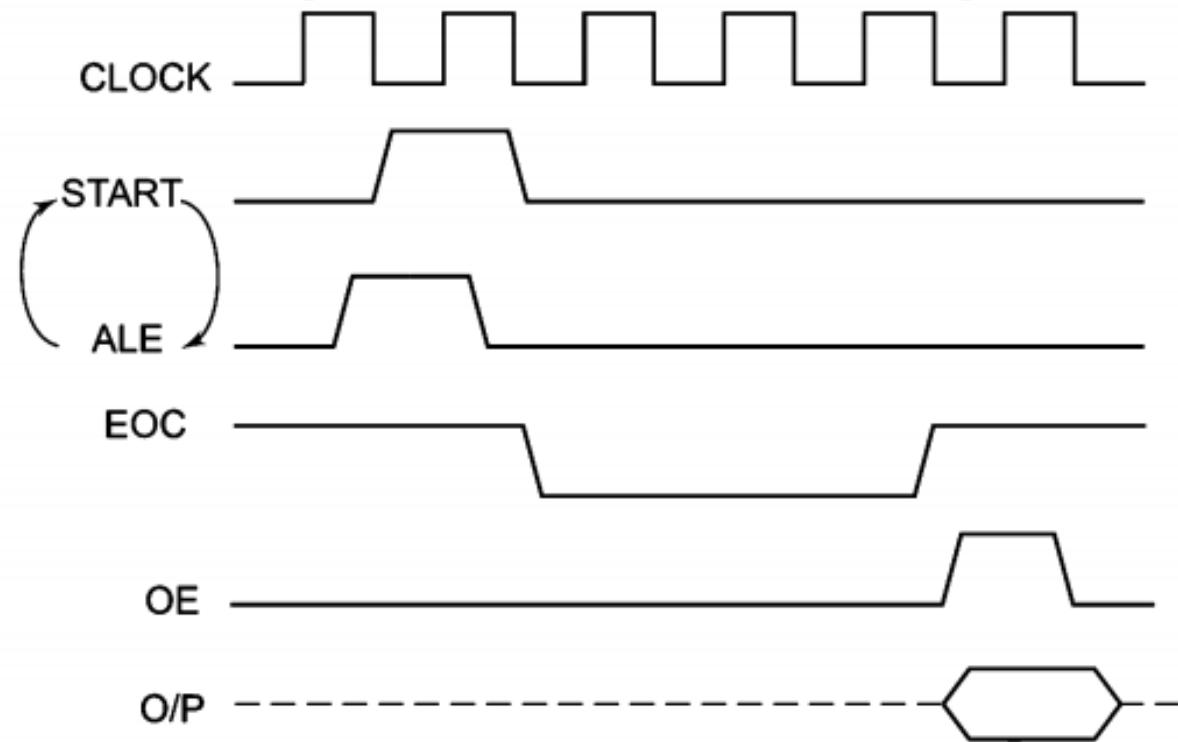
Pin diagram

I/P ₃ →	1	28 ← I/P ₂	
I/P ₄ →	2	27 ← I/P ₁	
I/P ₅ →	3	26 ← I/P ₀	I/P ₀ –I/P ₇
I/P ₆ →	4	25 ← ADD A	ADD A, B, C
I/P ₇ →	5	24 ← ADD B	O ₇ –O ₀
SOC →	6	23 ← ADD C	SOC
EOC →	7	22 ← ALE	EOC
O ₃ →	8 ADC 0808	21 ← O ₇ MSB	OE
OE →	9	20 ← O ₆	CLK
CLK →	10	19 ← O ₅	V _{CC} , GND
V _{CC} →	11	18 ← O ₄	V _{ref+} and V _{ref-}
V _{ref+} →	12	17 ← O ₀ LSB	
GND →	13	16 ← V _{ref-}	
O ₁ →	14	15 ← O ₂	

Example problem for interfacing

Interface ADC 0808 with 8086 using 8255 ports. Use Port A of 8255 for transferring digital data output of ADC to the CPU and Port C for control signals. Assume that an analog input is present at I/P₂ of the ADC and a clock input of suitable frequency is available for ADC. Draw the schematic and write required ALP.

The timing diagram of different signals of ADC0808 is shown in Fig. 5.38.



Solution Figure 5.39 shows the interfacing connections of ADC0808 with 8086 using 8255. The analog input I/P₂ is used and therefore address pins A,B,C should be 0,1,0 respectively to select I/P₂. The OE and ALE pins are already kept at +5V to select the ADC and enable the outputs. Port C upper acts as the input port to receive the EOC signal while port C lower acts as the output port to send SOC to the ADC. Port A acts as a 8-bit input data port to receive the digital data output from the ADC. The 8255 control word is written as follows:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Control word
1	0	0	1	1	0	0	0	= 98 H

The required ALP is given as follows:

```

MOV AL,98 H           ; Initialise 8255 as
OUT CWR,AL           ; discussed above
MOV AL,02H            ; Select I/P2 as analog
OUT PORT B,AL         ; input
MOV AL,00H            ; Give start of conversion
OUT PORT C,AL         ; pulse to the ADC.
MOV AL,01 H           ;
OUT PORT C,AL         ;
MOV AL,00H            ;

```

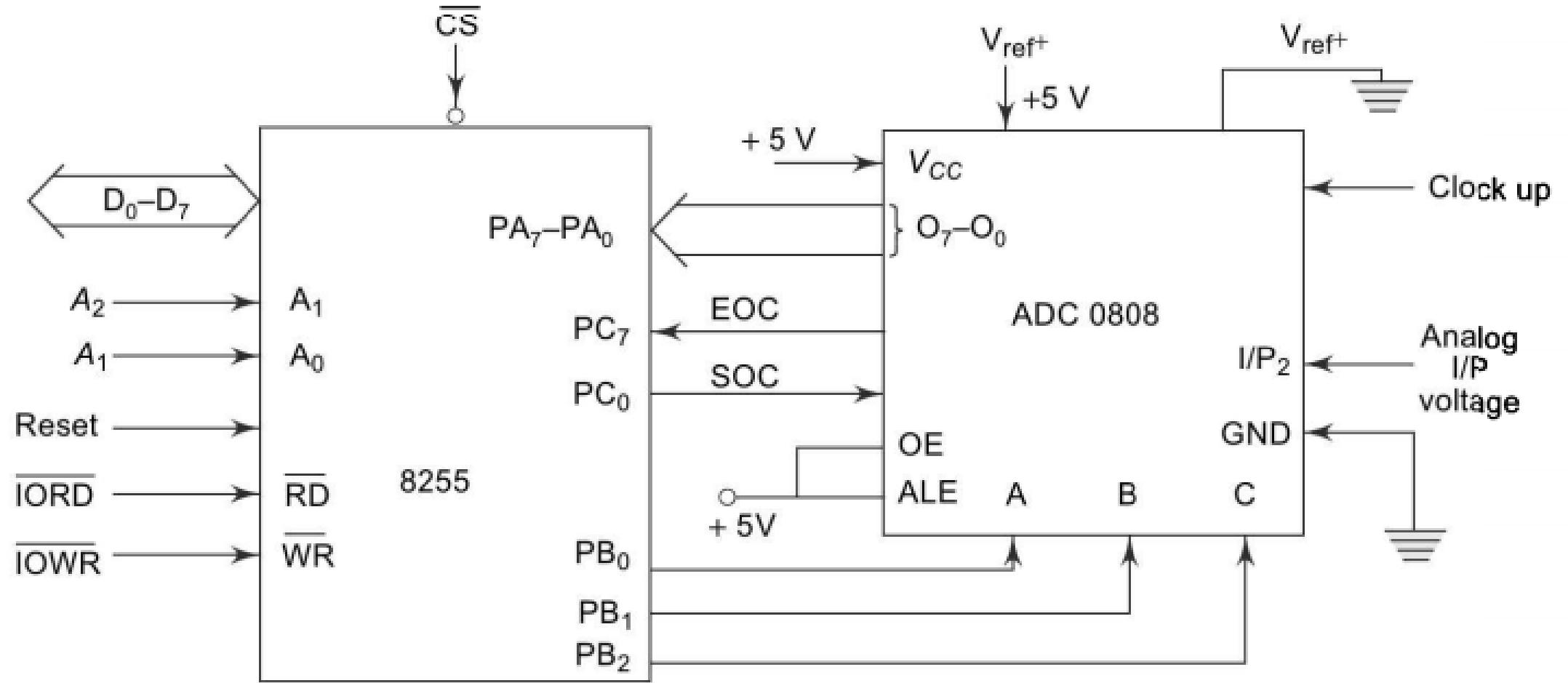
Activate
Go to Sett



continued

```
        OUT PORT C,AL      ;  
WAIT :   IN AL,PORTC    ; Check for EOC by  
                  ; reading port C upper and  
                  ; rotating through carry.  
                  ; If EOC, read digital equivalent in  
                  ; AL  
        IN AL,PORTA    ; Stop  
        HLT
```

Interfacing 8086 with 0808



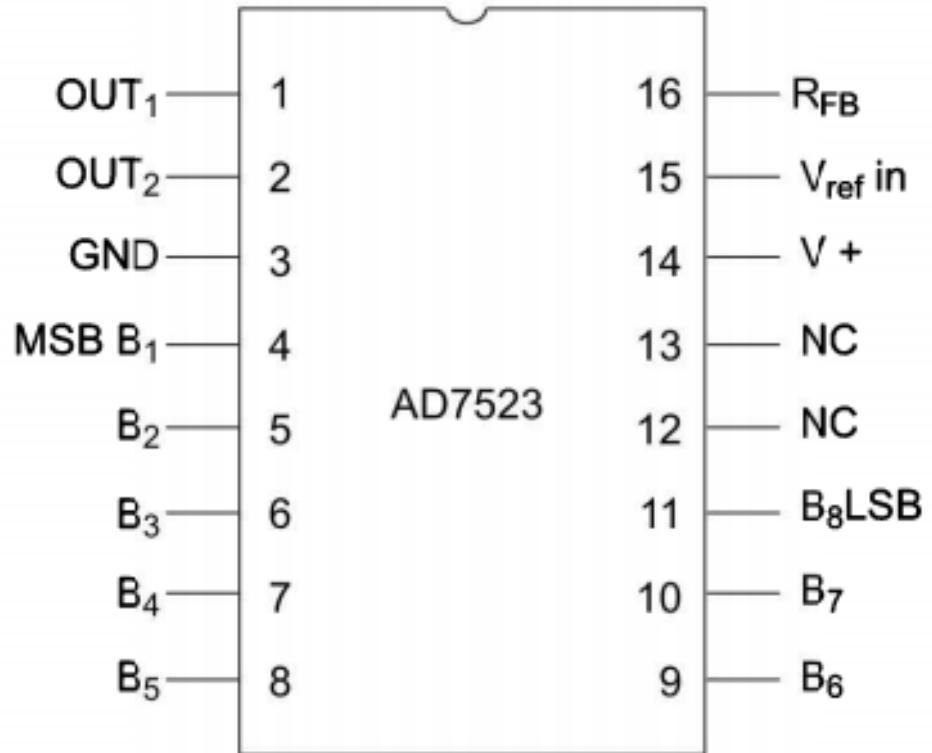


Interfacing digital to analog converters

- ❖ DACs convert binary numbers into their analog equivalent voltages
- ❖ Applications of DAC
 - ❖ Digitally controlled gains
 - ❖ Motor speed controls
 - ❖ Programmable gain amplifiers

AD 7523 8-bit multiplying dac

- ❖ Intersil's AD 7523 is a 16 pin, multiplying DAC containing R-2R ladder with $R=10\text{ K}$ for digital to analog conversion along with single pole double throw NMOS switches to connect the digital inputs to the ladder.



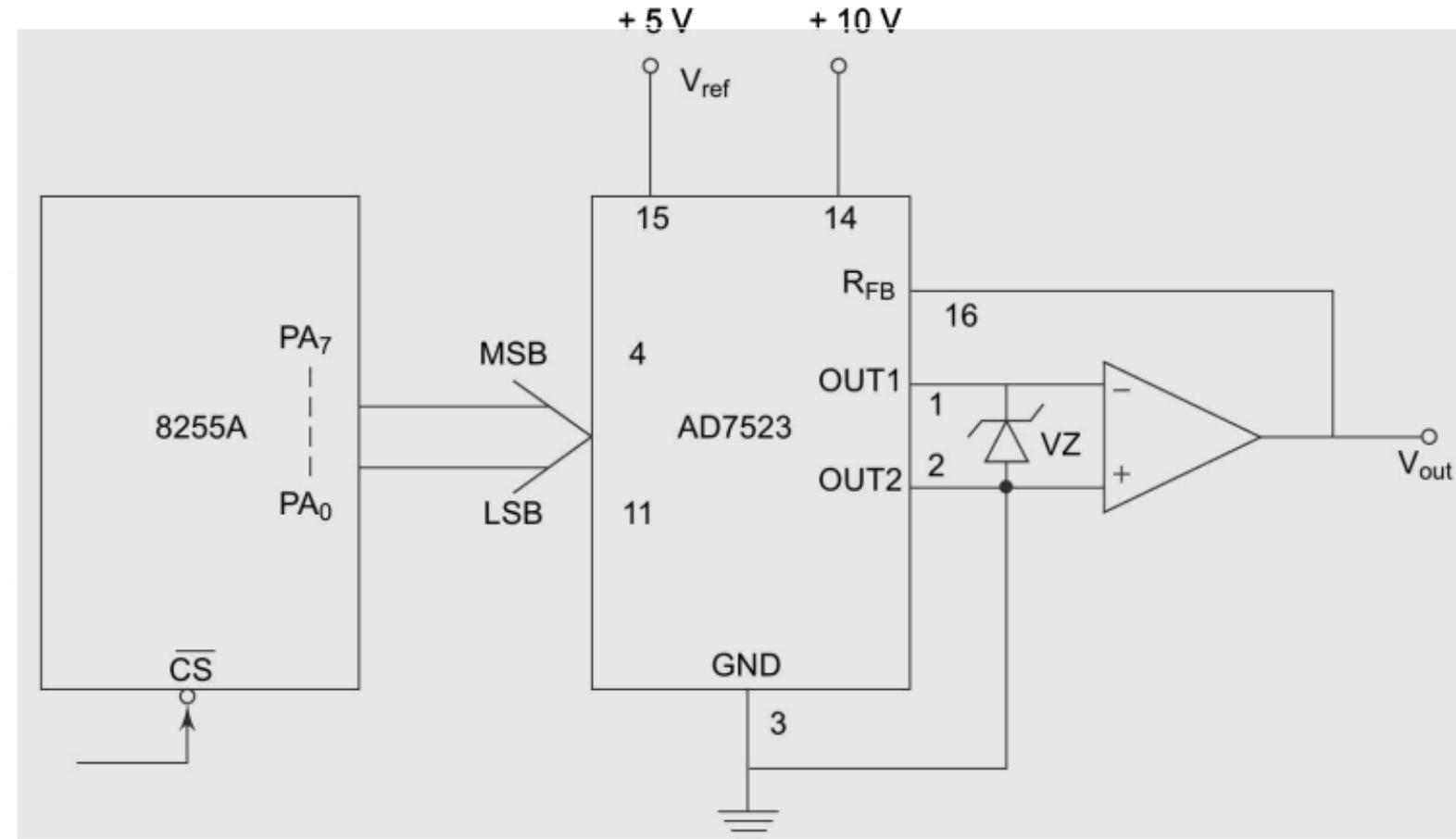


continued

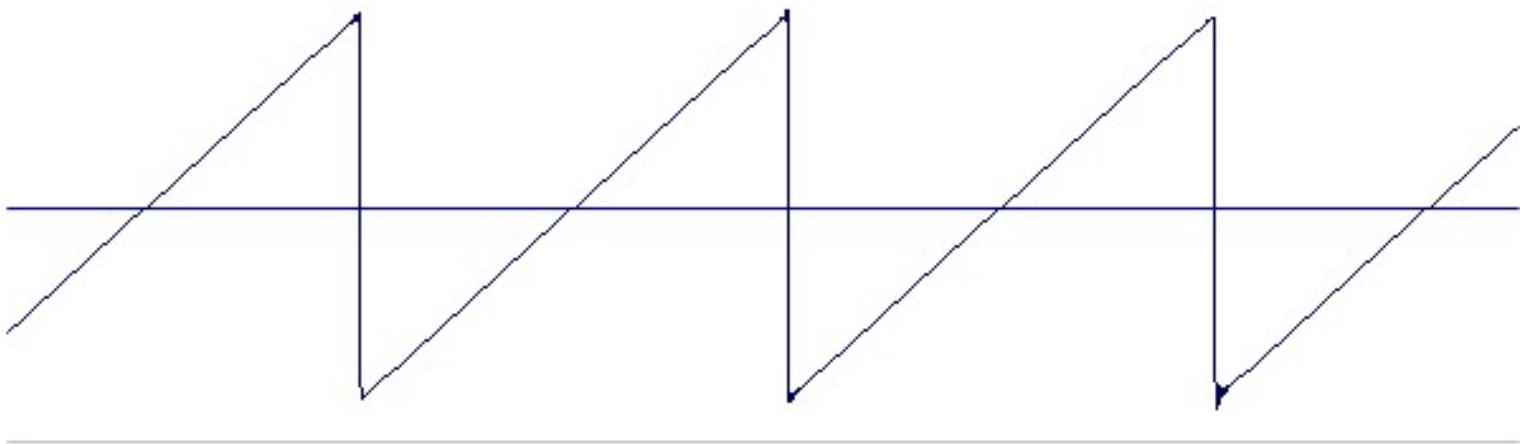
- ❖ The supply range extends from +5V to +15V
- ❖ The maximum analog output value is +10V, when all the digital input values are at logic high state
- ❖ To save the DAC from negative transients, a Zener diode is connected between OUT1 and OUT2.
- ❖ An OPAMP is used as a current-to-voltage converter at the output of AD 7523 in order to generate an equivalent output voltage for the current produced.

Example problem to interface dac with 8086

Interface DAC AD7523 with an 8086 CPU running at 8 MHz and write an assembly language program to generate a sawtooth waveform of period 1 ms with V_{max} 5V.



Sawtooth wave



```

ASSUME      CS : CODE
CODE        SEGMENT
START:      MOV AL,80 H          ; Initialise port A as output
            OUT CWR,AL         ; port
AGAIN:      MOV AL,00H          ; Start the ramp from 0V
BACK :       OUT PORTA,AL      ; Input 00H to DAC
            INC AL             ; Increment AL to increase ramp output
            CMP AL,0F2H          ; Is upper limit reached?
            JB BACK             ; If not, then increment the ramp
            JMP AGAIN            ; Else start again from 00H
CODE        ENDS
END START
  
```

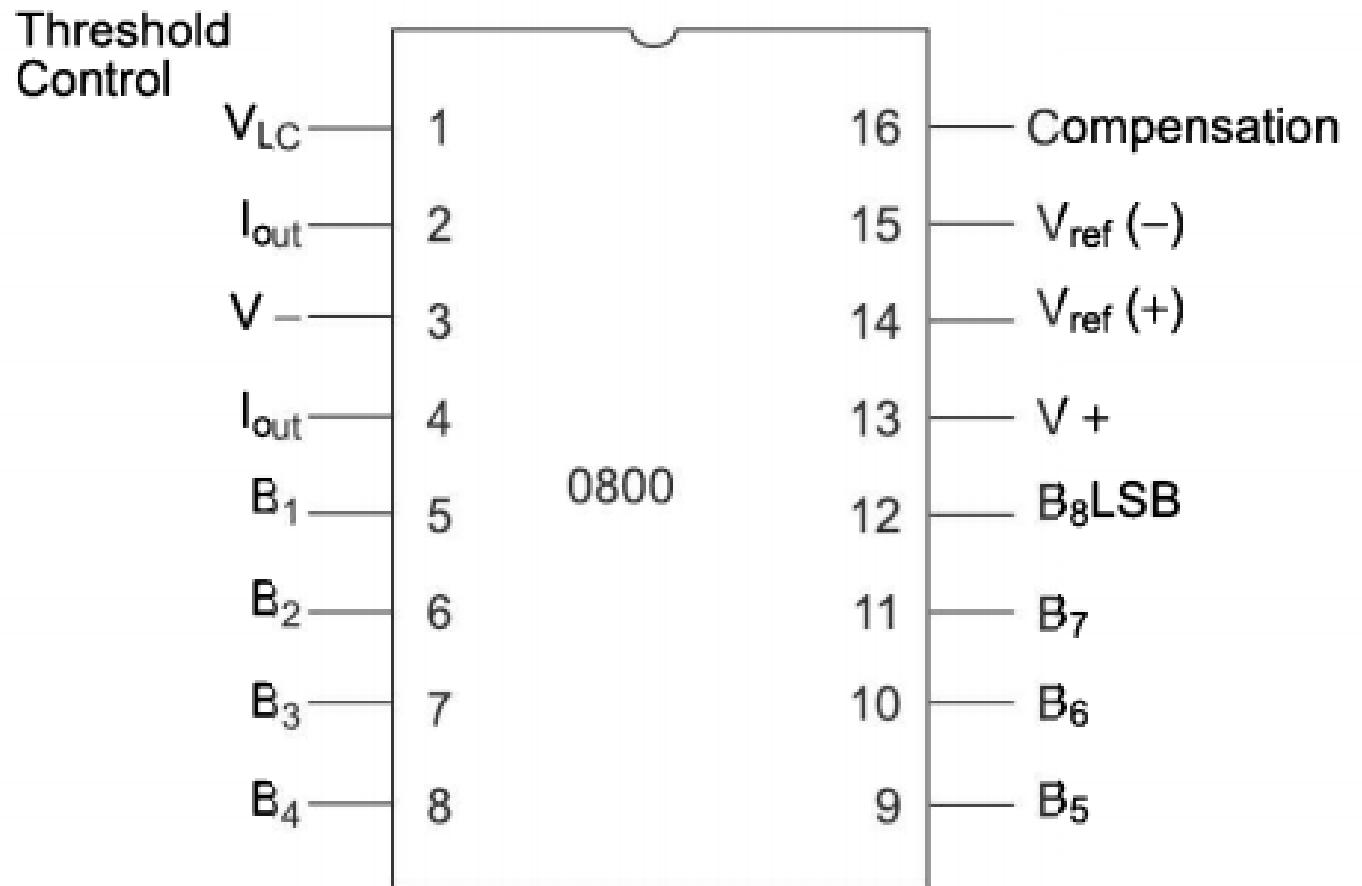
- ❖ Port A is designed to send the digital data as inputs to the converter
- ❖ The ramp starts from 0V and hence AL is moved with 00H.
- ❖ To increment the ramp, the value of AL must be incremented till it reaches FFH
- ❖ After that the sawtooth again starts from 00H and this procedure repeats



Dac 0800 8-bit digital to analog converter

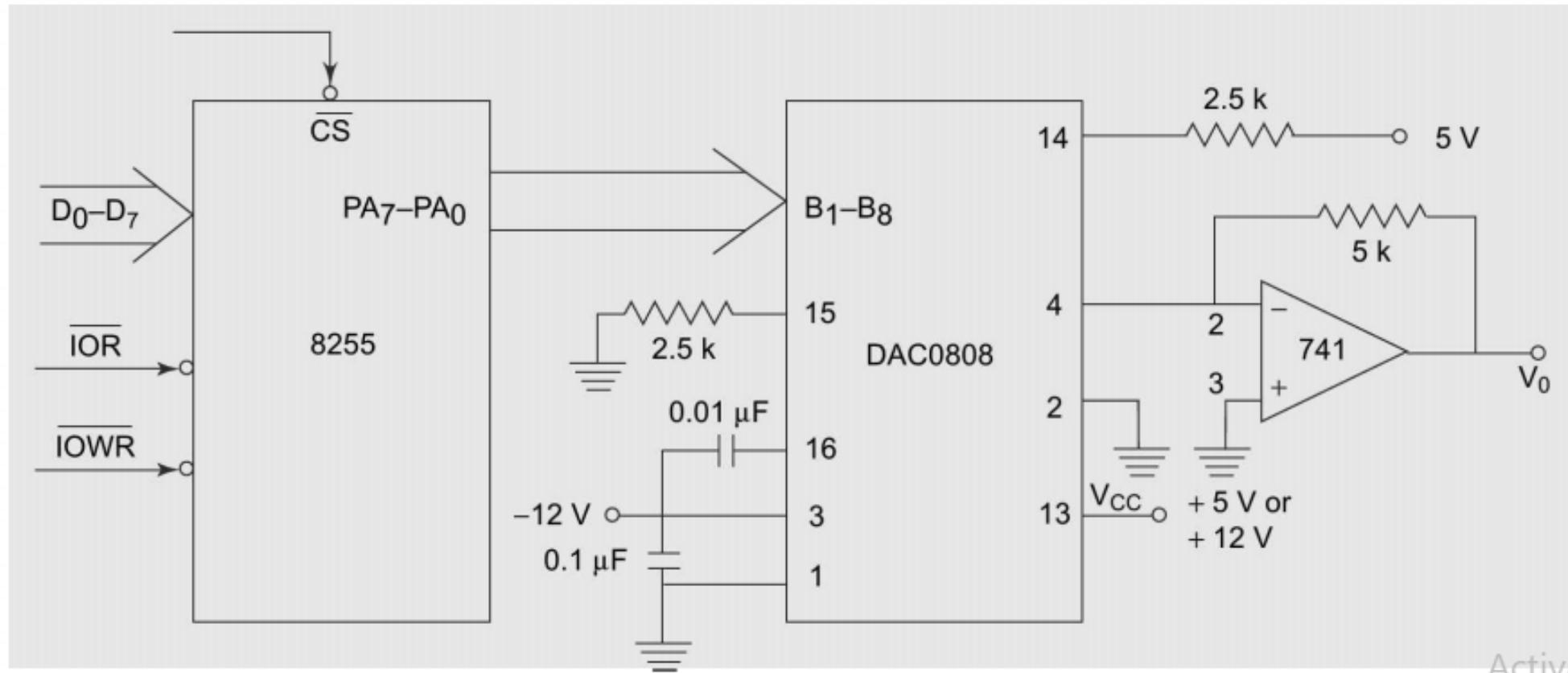
- ❖ DAC 0800 is a monolithic 8-bit DAC manufactured by National Semiconductor.
- ❖ It can work at various voltages from 4.8V to 18V, usually, 5V or 12 V

Pin diagram



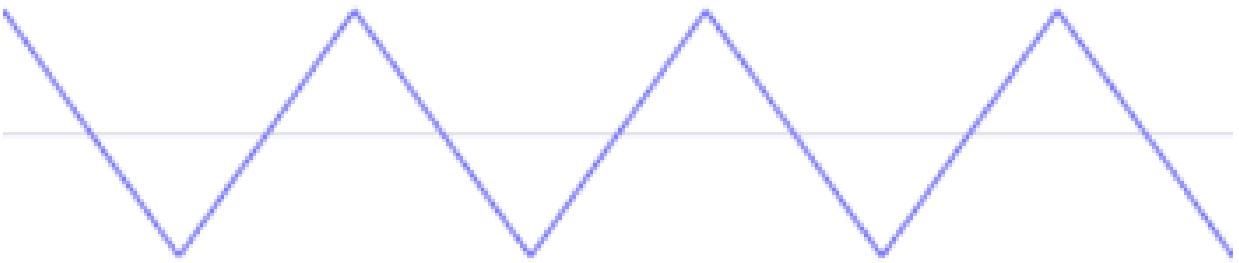
example

Write an assembly language program to generate a triangular wave of frequency 500 Hz using the interfacing circuit given in Fig. 5.48. The 8086 system operates at 8 MHz. The amplitude of the triangular wave should be +5 V.



Activate

Triangular wave



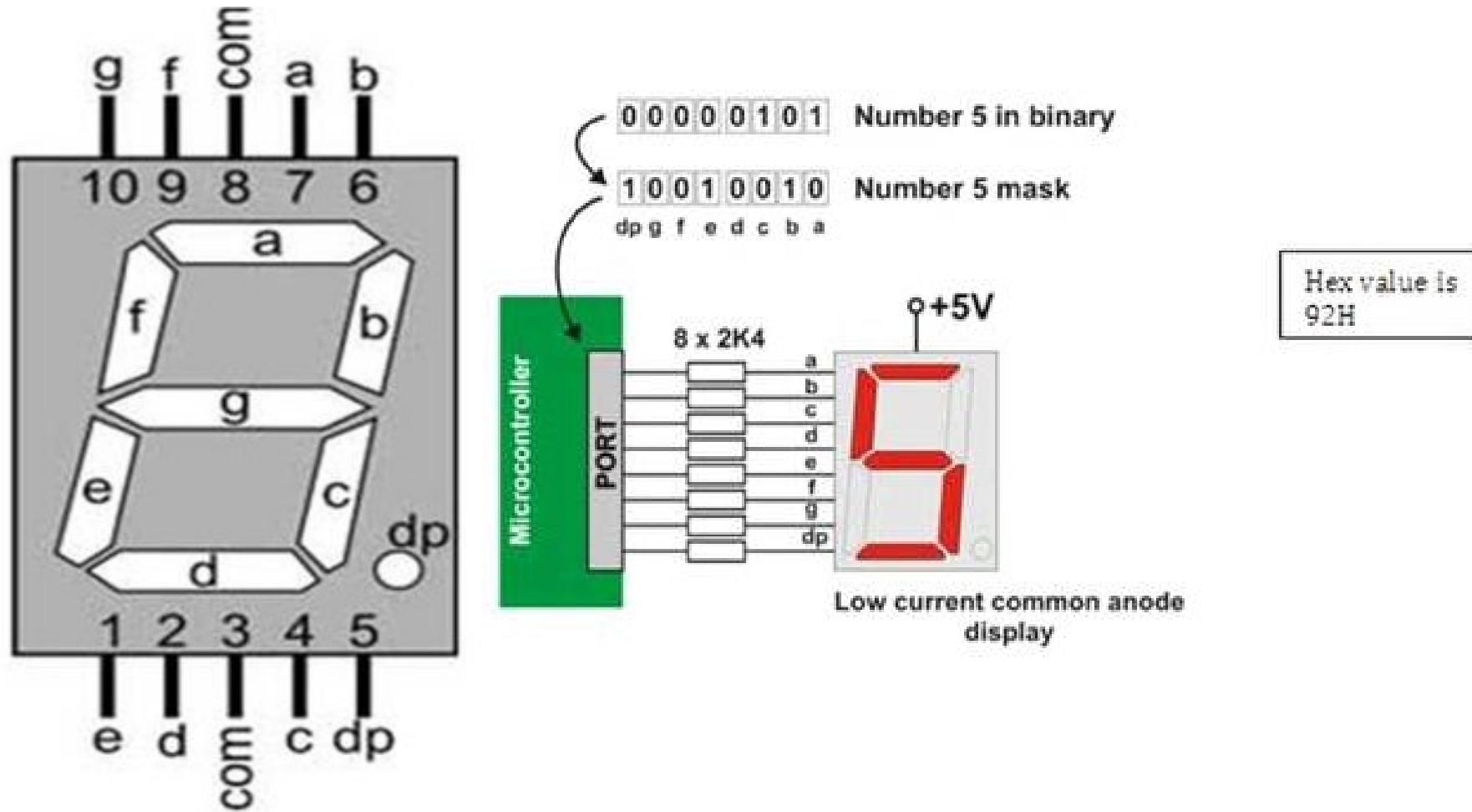
$N = 0$



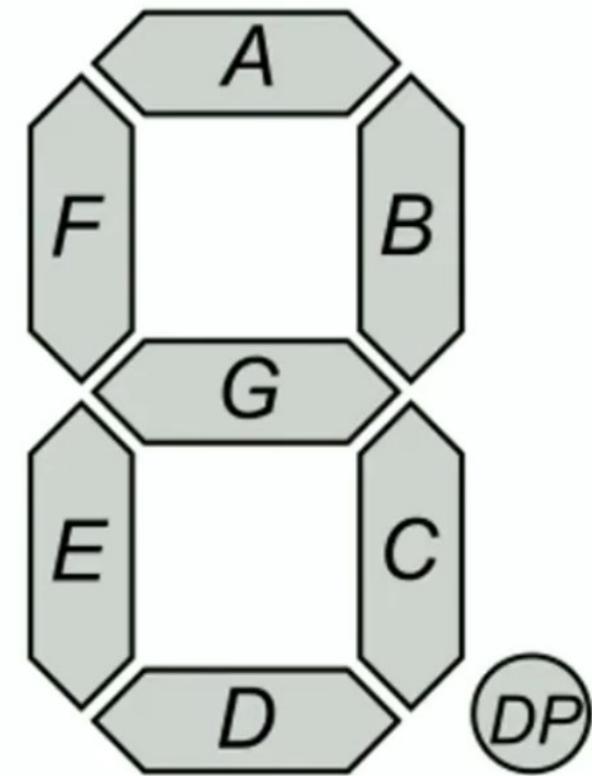
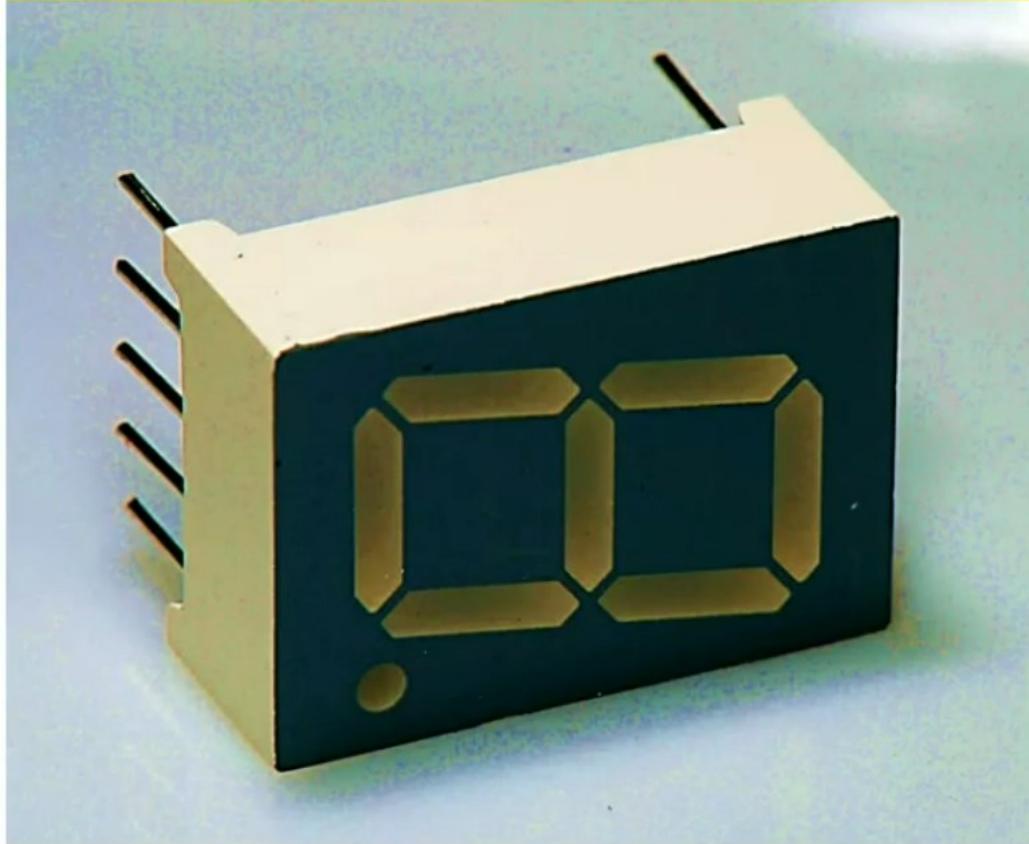
Solution The V_{ref+} should be tied to +5 V to generate a wave of +5V amplitude. The required frequency of the output is 500 Hz, i.e. the period is 2 ms. Assuming the wave to be generated is symmetric, the waveform will rise for 1 ms and fall for 1 ms. This will be repeated continuously. In the previous program, we have already written an instruction sequence for period 1 ms. Using the same instruction sequence one can derive this triangular waveform. The ALP is given as follows:

```
ASSUME    CS : CODE
CODE      SEGMENT
START :   MOV AL,80 H          ; Initialise 8255 ports
          OUT CWR,AL          ; suitably.
          MOV AL,00H            ; Start rising ramp from
BACK :    OUT PORT A,AL       ; OV by sending 00H to DAC.
          INC AL                ; Increment ramp till 5V
          CMP AL,FFH            ; i.e. FFH.
          JB BACK               ; If it is FFH then,
BACK1 :   OUT PORT A,AL      ; Output it and start the falling
          DEC AL                ; ramp by decrementing the
          CMP AL,00              ; counter till it reaches
          JA BACK1              ; zero. Then start again
          JMP BACK              ; for the next cycle.
CODE      ENDS
END START
```

Interfacing 7-segment display with 8086

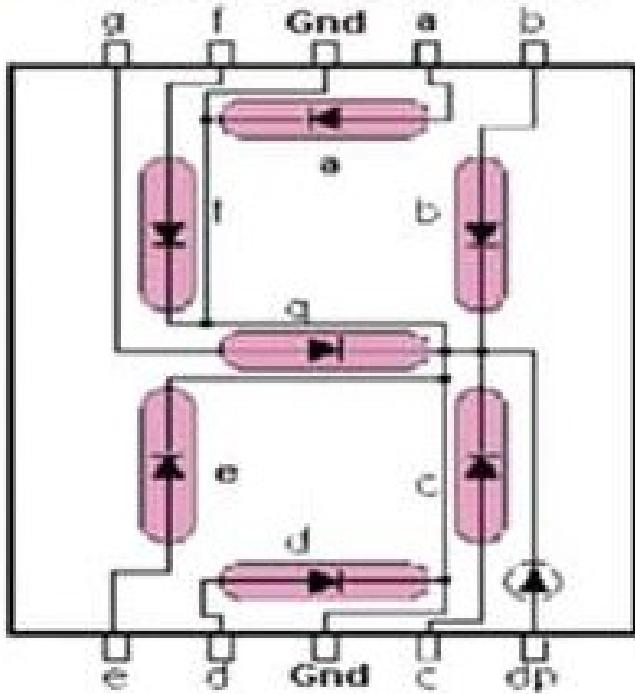


Physical View of Seven Segment Display

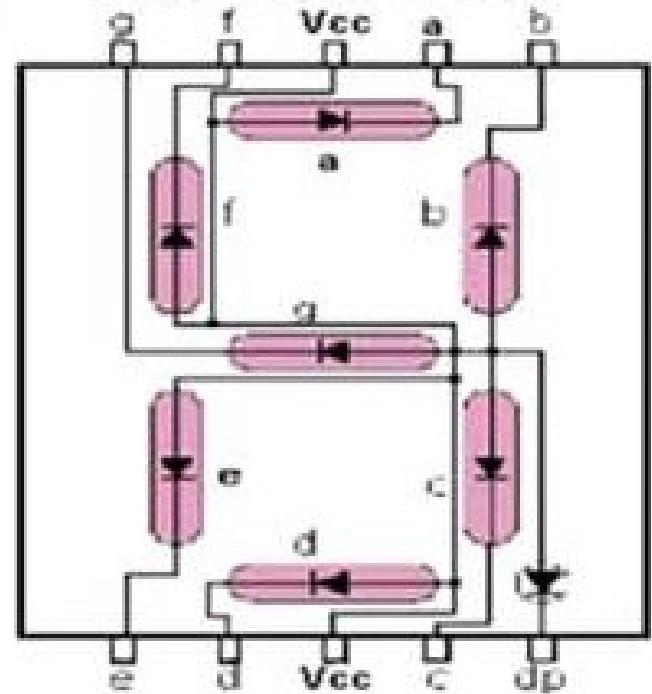


Types of 7-segment displays

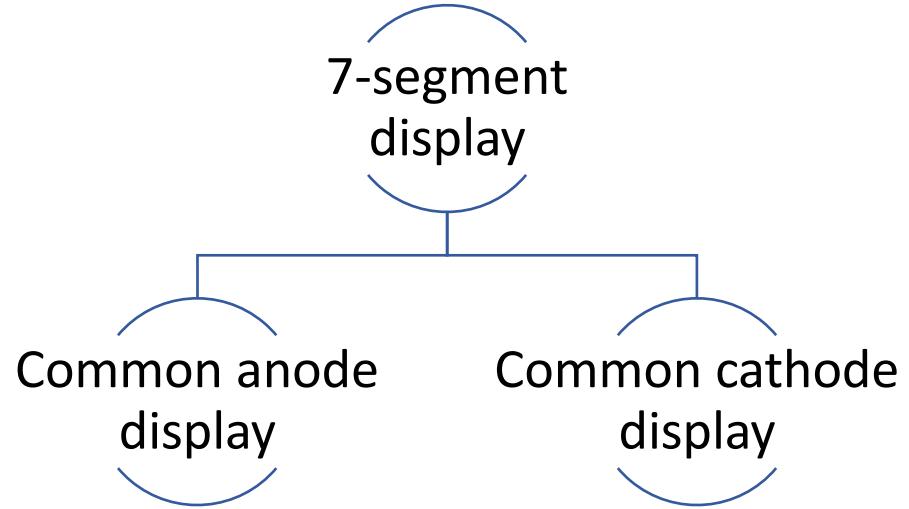
Common Cathode



Common Anode



continued



- ❖ Basically LEDs have anode and cathode
- ❖ In Common anode display, anodes of all LEDs are connected to VCC, cathodes are connected to the microprocessor port via 8255.
- ❖ In common cathode display, cathodes are connected to VCC and anodes are connected to the microprocessor port via 8255.

continued

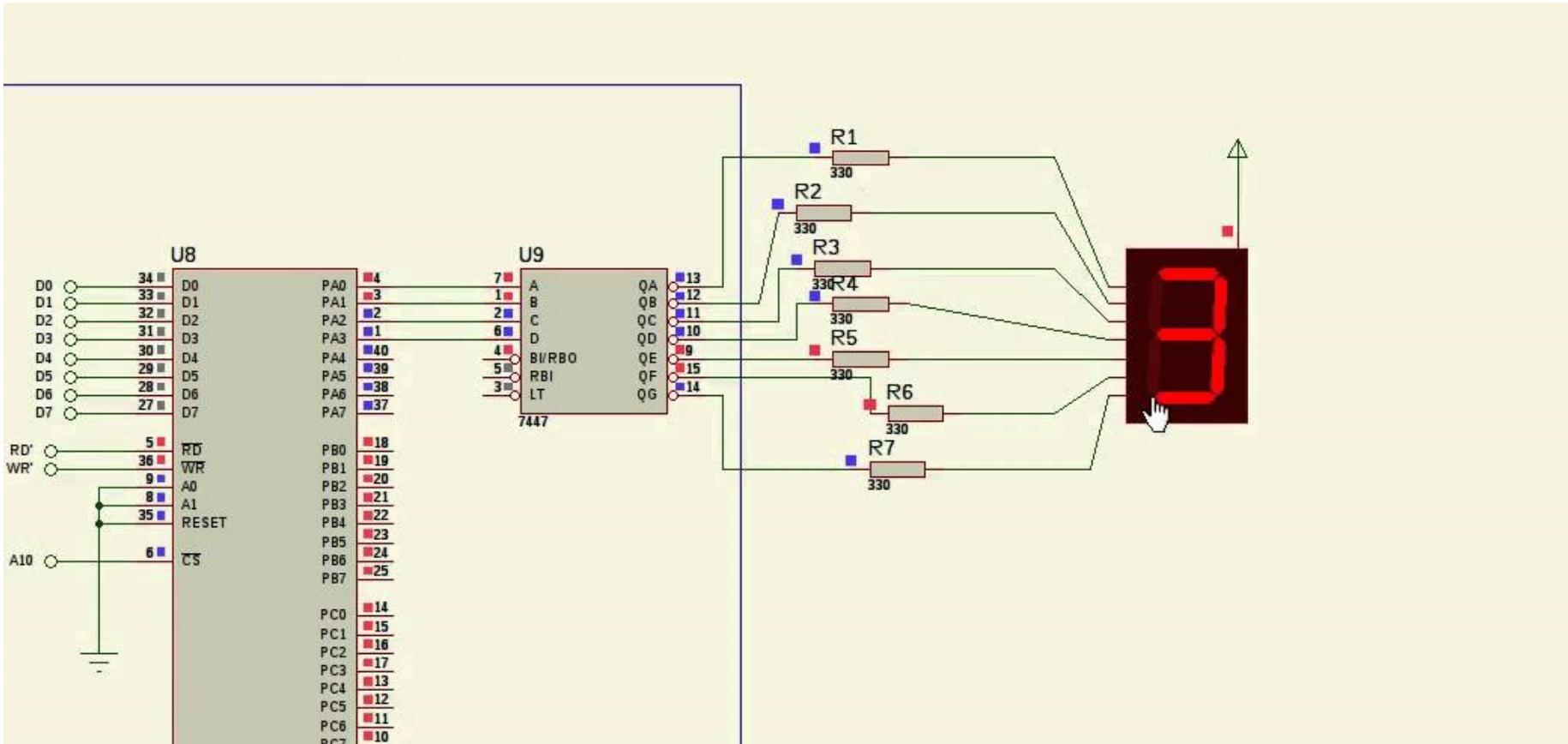
DIODE	DATA	STATUS
COMMON CATHODE	1	ON
	0	OFF
COMMON ANODE	0	ON
	1	OFF

DISPLAY FORMAT

- ❖ Let us consider common anode display. The connection table is as follows
- ❖ Logic 0 – turn on segment, logic 1 – turn off segment

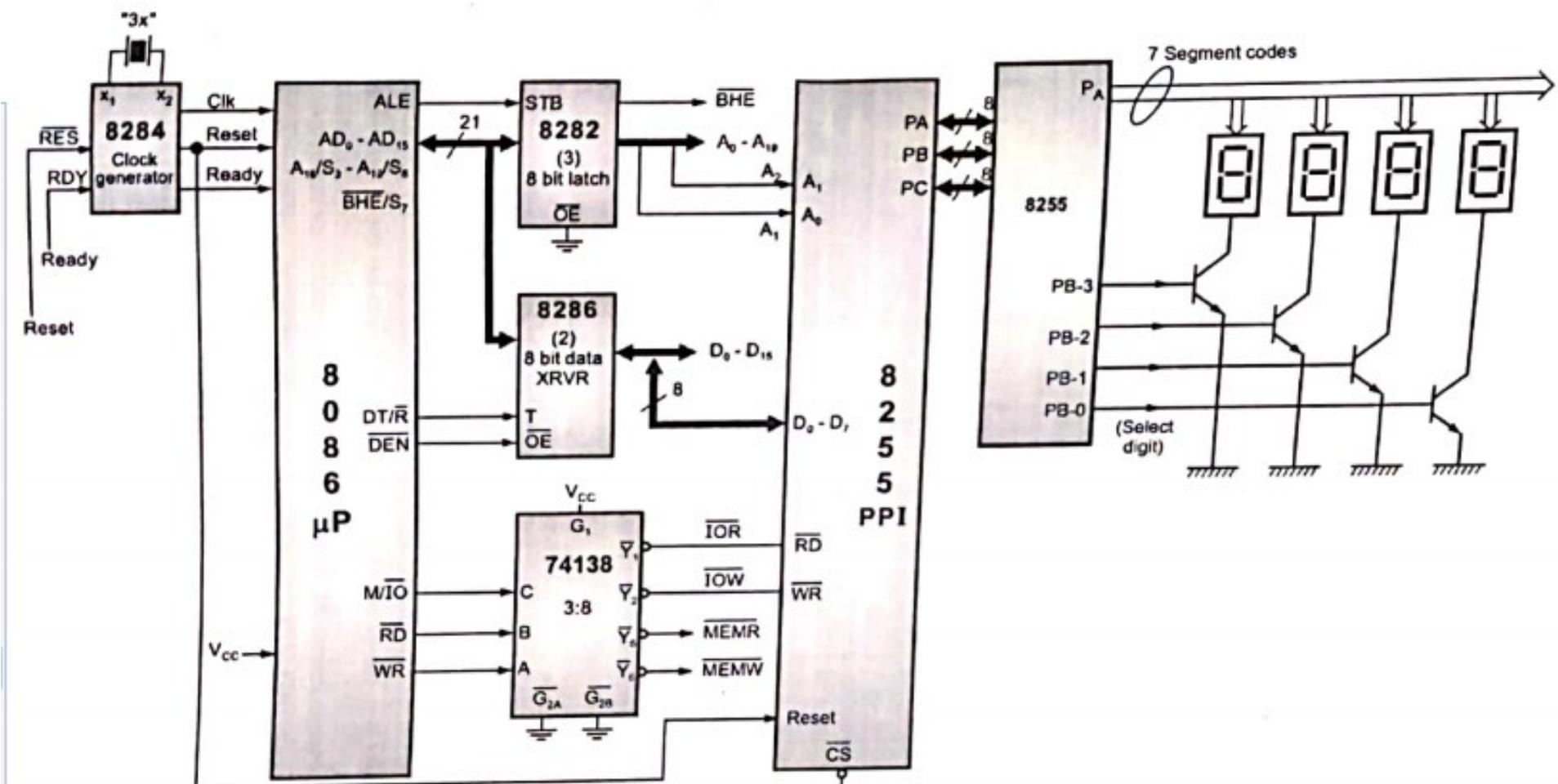
D7	D6	D5	D4	D3	D2	D1	D0
DP	G	F	E	D	C	B	A

7 – segment display interfacing



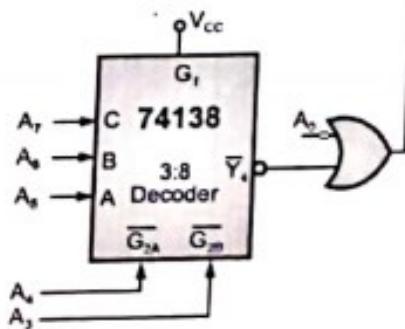
Equivalent hex values

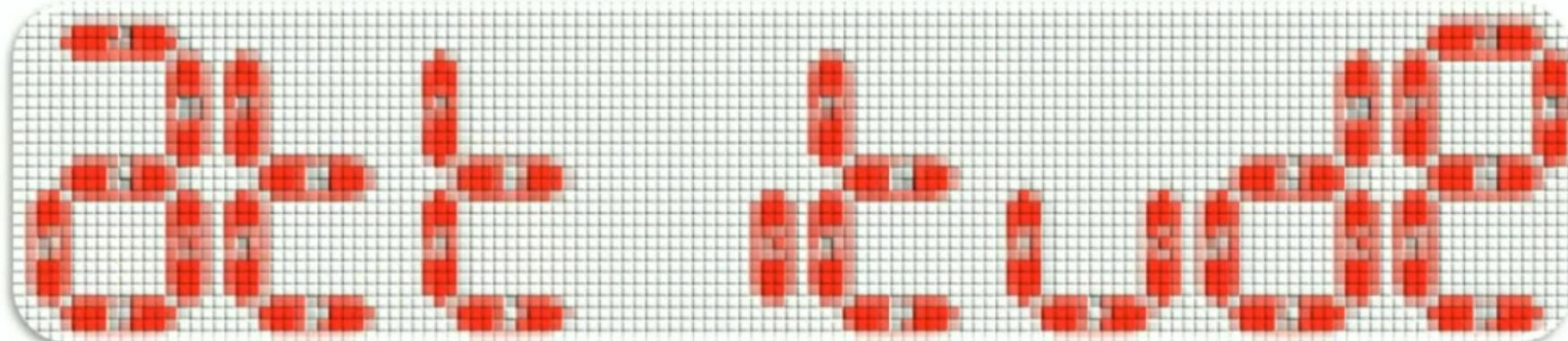
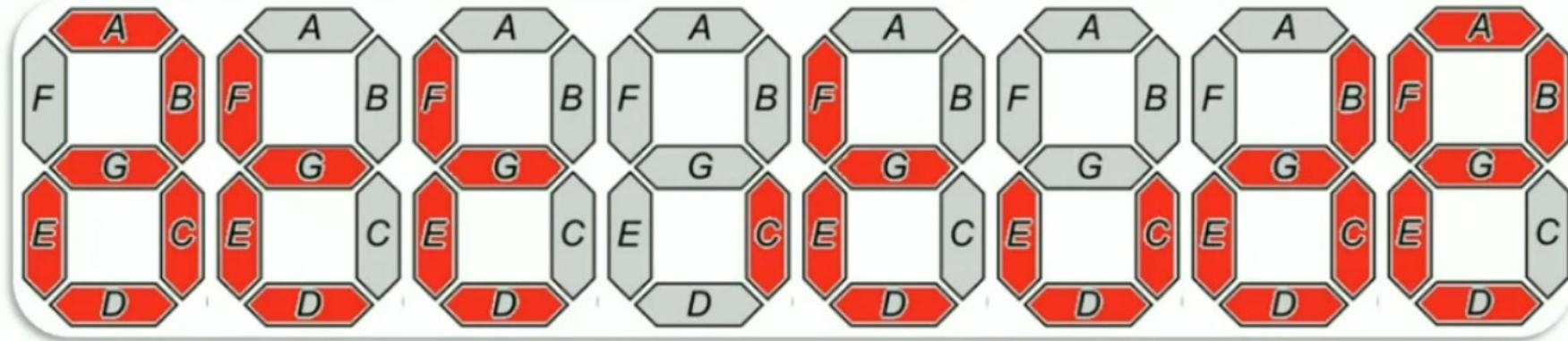
DIGIT	a	b	c	d	e	f	g	HEX Value
0	0	0	0	0	0	0	1	0x40
1	1	0	0	1	1	1	1	0xF9
2	0	0	1	0	0	1	0	0x24
3	0	0	0	0	1	1	0	0x30
4	1	0	0	1	1	0	0	0x19
5	0	1	0	0	1	0	0	0x12
6	0	1	0	0	0	0	0	0x02
7	0	0	0	1	1	1	1	0xF8
8	0	0	0	0	0	0	0	0x00
9	0	0	0	1	1	0	0	0x10



I/O map of 8255 at address 80H

	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	
PA	1	0	0	0	0	0	0	0	80H
PB	1	0	0	0	0	0	1	0	82H
PC	1	0	0	0	0	1	0	0	84H
C.W.	1	0	0	0	0	1	1	0	86H

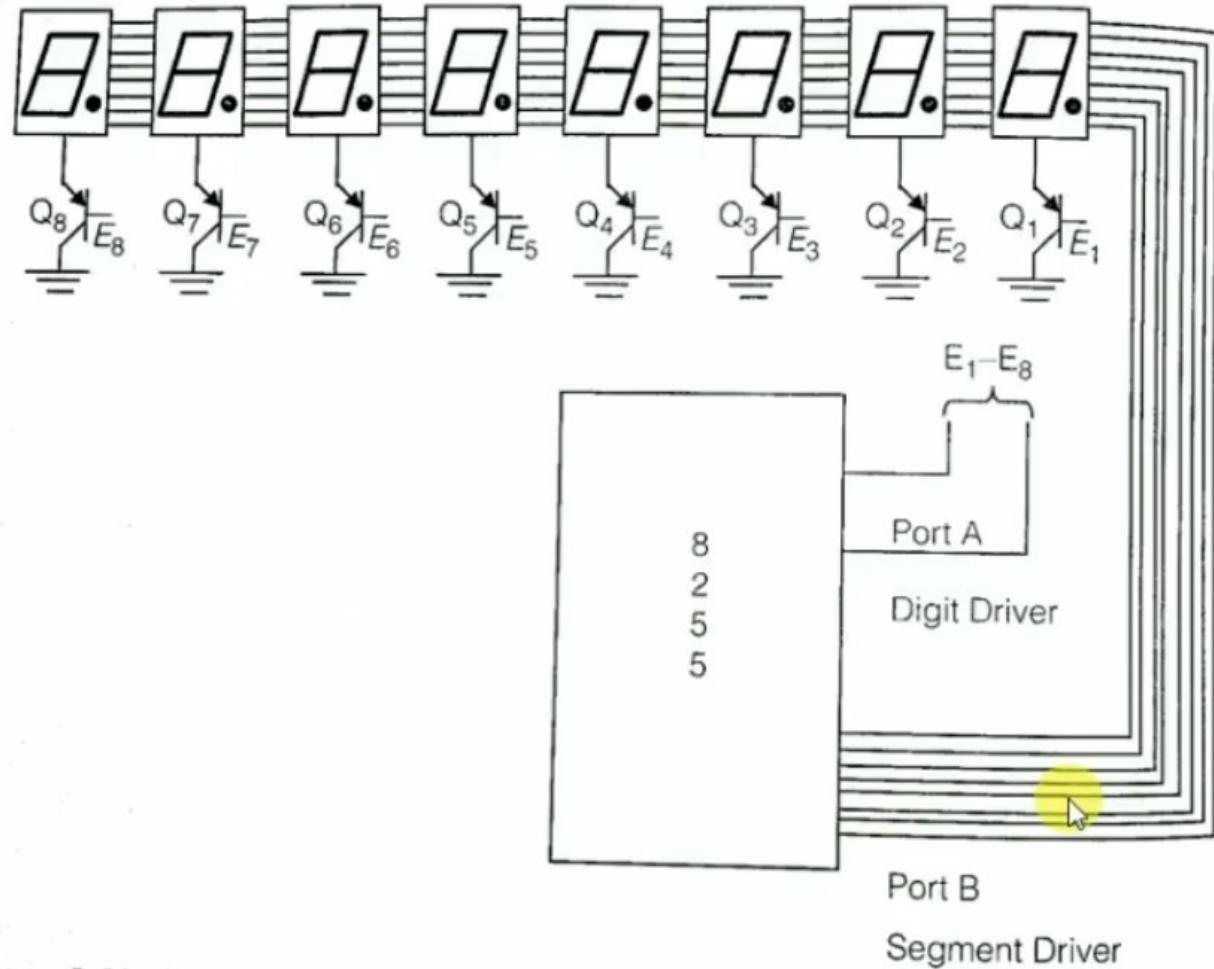




	hex	h	g	f	e	d	c	B	a
a	5F	0	1	0	1	1	1	1	1
t	78	0	1	1	1	1	0	0	0
t	78	0	1	1	1	1	0	0	0
i	04	0	0	0	0	0	1	0	0
t	78	0	1	1	1	1	0	0	0
u	3E	0	0	1	1	1	1	1	0
d	5E	0	1	0	1	1	1	1	0
e	5B	0	1	0	1	1	0	1	1



Interfacing Multiple Seven Segment Display



Program : Display word “attitude”

```
CR EQU 0C6H           ;address of the control register
PA EQU 0C0H           ;address of Port A
PB EQU 0C2H           ;address of Port B → 

TABLE DB      5FH,78H,78H,04,78H,3EH,5EH,7BH

MOV AL, 80H           ;CW with ports A and B as outputs
OUT CR, AL            ;send it to the control register

STRT: LEA SI, TABLE   ;let SI point to the display data
    MOV CX, 8           ;CX to contain the number of digits
    MOV BL, 01111111    ;bit sequence for leftmost digit

AGN:  MOV AL, BL        ;copy it to AL
      OUT PA, AL         ;send it to Port A
      MOV AL, [SI]         ;copy display data to AL
      OUT PB, AL         ;send it to Port B
      CALL DELAY          ;call a delay
      INC SI              ;increment pointer
      ROR BL, 1            ;activate next digit
```

1	0	0	0	0	0	0	0
I/O Mode	PA Mode-0		PA -O/P	PCU-O/P	PB -Mode-0	PB -O/P	PCL-O/P

✓ CR EQU 0C6H ;address of the control register
 ✓ PA EQU 0C0H ;address of Port A
 ✓ PB EQU 0C2H ;address of Port B

TABLE DB 5FH, 78H, 78H, 04, 78H, 3EH, 5EH, 7BH

MOV AL, 80H ;CW with ports A and B as outputs
 OUT CR, AL ;send it to the control register

STRT: LEA SI, TABLE ;let SI point to the display data
 MOV CX, 8 ;CX to contain the number of digits
 MOV BL, 01111111 ;bit sequence for leftmost digit

AGN: MOV AL, BL ;copy it to AL
 OUT PA, AL ;send it to Port A
 MOV AL, [SI] ;copy display data to AL
 OUT PB, AL ;send it to Port B
 CALL DELAY ;call a delay
 INC SI ;increment pointer
 ROR BL, 1 ;activate next digit

Program : Display word “attitude”



```
LOOP AGN        ;repeat until CX = 0
    JMP START
    DELAY PROC NEAR
    MOV DX, 300      ;repeat the sequence of display
    ;delay loop
    ;calculate delay value
BACK: DEC DX
    JNZ BACK
    RET
    DELAY ENDP

END
```



Key Board Interfacing to 8086 MP through 8255 PPI





4 × 4 Matrix Keyboard Interfacing

☞ Working of a Key

- A keyboard is the most basic input device.
- When a Key is connected to a port pin, the port pin must be configured as an input pin. We read the value of the pin to establish if a key is pressed.
- Most circuits use an active low logic for a key.
- The Keys are connected to Vcc via a pull up resistor. The default value on the keys is a 1. When a Key is pressed, it gets connected to GND making its value 0.



4 × 4 Matrix Keyboard Interfacing

☞ Concept of a matrix Keyboard

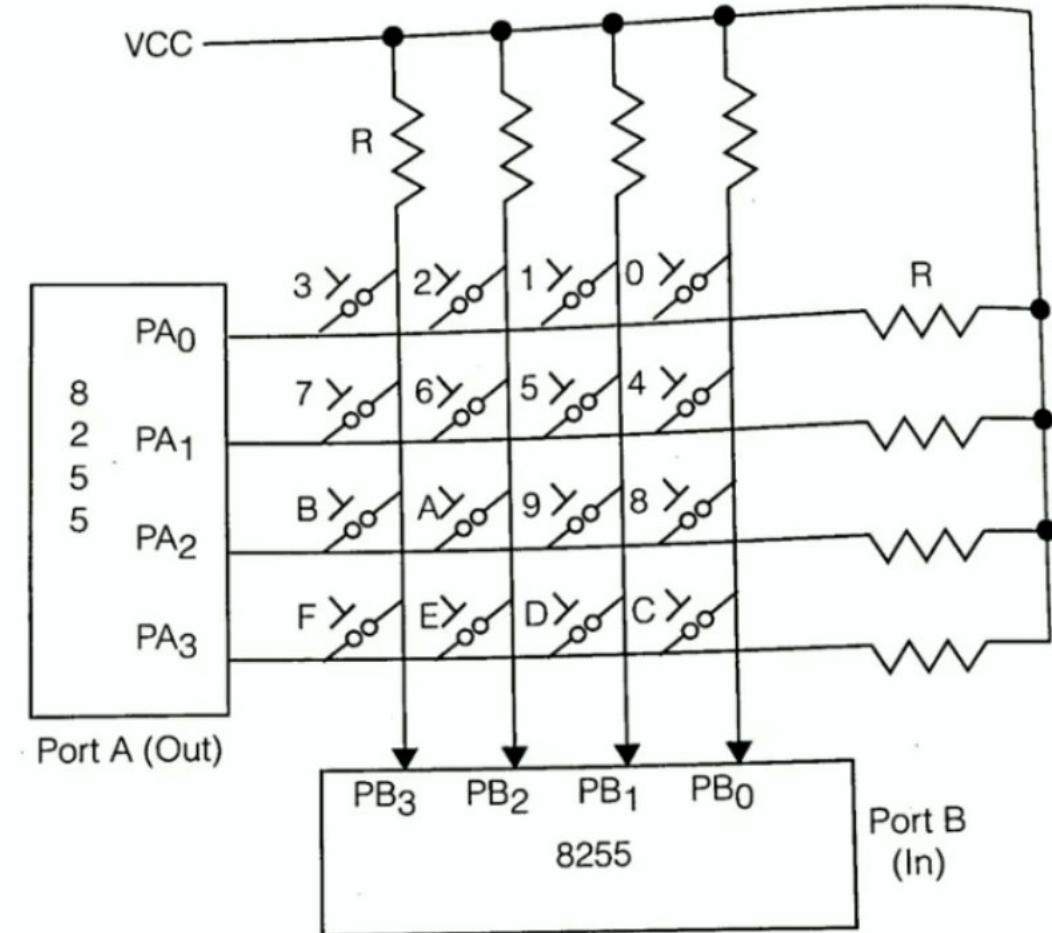
- For a small keypad, like a car remote key, we need only 2-3 keys. We could simply connect them on each line of a port.
- But for a large keyboard like a TV Remote or the computer's keyboard, the number of keys is large. We cannot afford one port pin each for every key.
- So we connect the keys in a matrix form. This gives us more keys using less lines. By identifying the row and the column we can figure out which key is pressed.



☞ 4x4 Matrix Keyboard

- We use four lines of a port, say PA.0... PA.3 as rows.
- We use four lines of a different port, say PB.0... PB.3 as columns. Rows will be Output whereas Columns will be Input.
- The columns will have a default value of 1 due to the Vcc with a pull up resistor. This gives us 16 intersecting points, on each we plant a key.
- When the key is “Pressed”, the corresponding row and column will come into contact.

Interfacing of Hex Key Board





☞ Program Logic

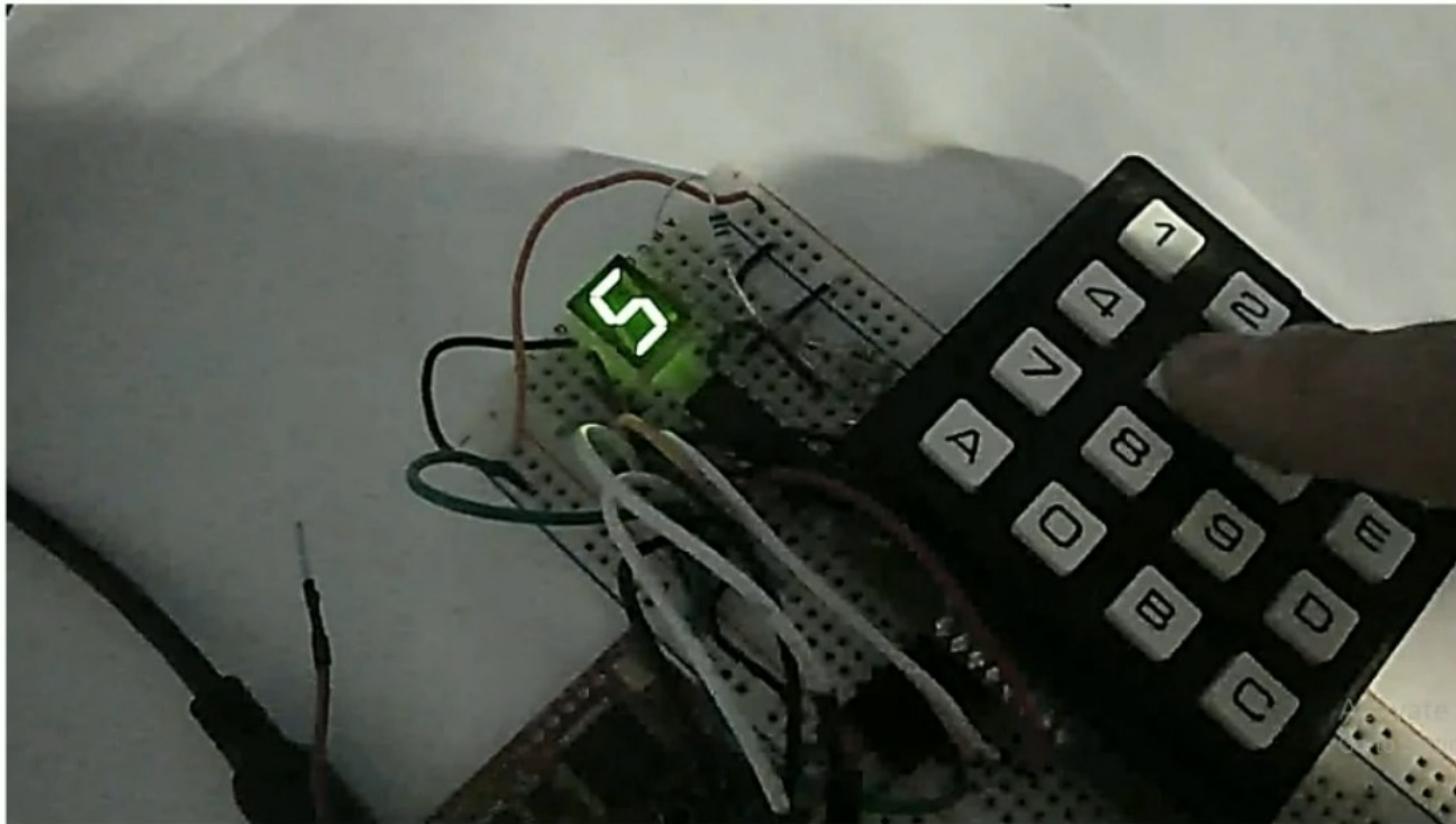
- First off, we need to know IF A KEY IS PRESSED?
- For this we keep sending 0s on all rows and read the columns.
- If columns are still all 1's it means no key is pressed. Repeat this process. Else move to Row Check. Once we are sure that a key is pressed, we check each row individually.
- This is done by sending a 0 on every row and checking the columns till the 0 comes on the columns. Once the row is identified we check the column by rotating the column data till a 0 is found.
- This finally identifies the key.



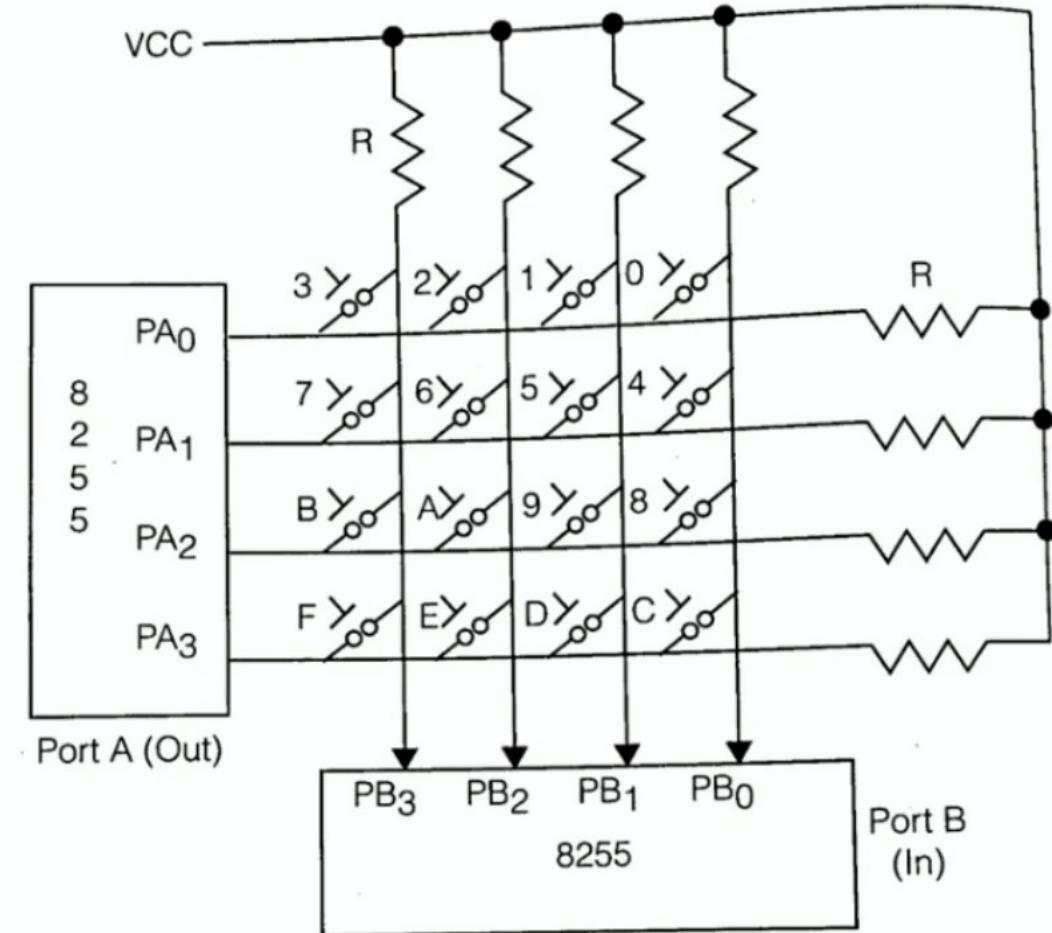
☞ Key De-bounce

- Practically every key uses a spring mechanism and hence will bounce.
- This means when a key is pressed there are microscopic vibrations which fluctuate the key value between 0 and 1 till it finally settles to a stable 0. Reading the key during this bouncing period may give you a wrong result. Avoid this bouncing period by calling a delay before performing row and column checking. This allows the key to stabilize in the “pressed” state of logic 0. The delay is typically 10-20 milliseconds. This is called key de-bouncing.

Hex Key Board and Seven segment Display



Interfacing of Hex Key Board



Column Identification

C 3	C 2	C 1	C 0
1	1	1	1

R 0	PA0	0
R 1	PA1	0
R 2	PA2	0
R 3	PA3	0

3	2	1	0
7	6	5	4
B	A	9	8
F	E	D	C

1	1	1	1
PB 3	PB 2	PB 1	PB 0

Column Identification

C 3	C 2	C 1	C 0
1	1	1	1

R 0	PA0	0
R 1	PA1	0
R 2	PA2	0
R 3	PA3	0

3	2	1	0
7	6	5	4
B	A	9	8
F	E	D	C

1	1	0	1
PB 3	PB 2	PB 1	PB 0

Defining CWR , Port A and Port B of 8255

8255 PPI Programming							
Control Word Register (0C6H)							
	8			2			
1	0	0	0	0	0	1	0
I/O	MODE - 0	PORT A		MODE - 0	PORT B		
	PORT A	O/P		PORT B	I/P		

ROW_0 DB 0, 1, 2, 3

ROW_1 DB 4, 5, 6, 7

ROW_2 DB 8, 9, 0AH, 0BH

ROW_3 DB 0CH, 0DH, 0EH, 0FH

KEY DB 0

CR EQU 0C6H ;address of the control register

PA EQU 0C0H ;address of Port A

PB EQU 0C2H ;address of Port B

MOV AL, 82H ;8255 control word

OUT CR, AL ;send to control register

Activate Windows
Go to Settings to activate Windows.

Column Reading

KEYP:

C3	C2	C1	C0	
1	1	1	1	
0	3	2	1	0
0	7	6	5	4
0	B	A	9	8
0	F	E	D	C
1	1	0	1	

```

MOV AL, 0          ;AL = 0
OUT PA, AL         ;send zeros to all the rows
IN AL, PB          ;read in the columns
AND AL, 0FH         ;mask the upper nibble
CMP AL, 0FH         ;compare with 0FH
JZ KEYP            ;if equal, no key, keep checking
CALL DELAY         ;if key press, wait for debounce

IN AL, PB          ;read in the columns
AND AL, 0FH         ;mask the upper nibble
CMP AL, 0FH         ;compare with 0FH
JZ KEYP            ;if equal, no key, check again
;Next, ground one row at a time
  
```

	CMP AL,OFH							
PORT B	1	1	1	1	1	1	0	1
0FH	0	0	0	0	1	1	1	1
CMP	Upper Nibble Masked				Not Equal			

Column Reading

KEYP:

C3	C2	C1	C0	
1	1	1	1	
0	3	2	1	0
0	7	6	5	4
0	B	A	9	8
0	F	E	D	C
1	1	0	1	

```

MOV AL, 0          ;AL = 0
OUT PA, AL         ;send zeros to all the rows
IN AL, PB          ;read in the columns
AND AL, 0FH         ;mask the upper nibble
CMP AL, 0FH         ;compare with 0FH
JZ KEYP            ;if equal, no key, keep checking
CALL DELAY         ;if key press, wait for debounce
IN AL, PB          ;read in the columns
AND AL, 0FH         ;mask the upper nibble
CMP AL, 0FH         ;compare with 0FH
JZ KEYP            ;if equal, no key, check again
;Next, ground one row at a time
  
```

	CMP AL,OFH							
PORT B	1	1	1	1	1	1	0	1
OFH	0	0	0	0	1	1	1	1
CMP	Upper Nibble Masked				Not Equal			

Activate Windows
Go to Settings to activate Windows.

Row Reading - Row 0

ROW0:

		C3	C2	C1	C0
R0	0	3	2	1	0
R1	1	7	6	5	4
R2	1	B	A	9	8
R3	1	F	E	D	C
		1	1	1	1
PB3	PB2	PB1	PB0		
C3	C2	C1	C0		

```

MOV AL, 0FEH      ;send '0' to Row0 alone
OUT PA, AL
IN AL, PB         ;read in the columns
AND AL, 0FH        ;mask the upper nibble
CMP AL, 0FH        ;compare with 0FH
JZ ROW1           ;no keypress in Row0, check Row1
LEA SI, ROW_0     ;not equal, SI to point to Row0 data
JMP COL_ID        ;go to finding the column
  
```

AL	F				E			
	1	1	1	1	1	1	1	0
					R3	R2	R1	R0

		CMP AL,OFH							
PORT B		1	1	1	1	1	1	1	Activate Windows
OFH		0	0	0	0	1	1	1	Go to Settings to activate Windows.
CMP		Upper Nibble Masked				Equal			



Weblink for 7 segment display and 4x4 keyboard interfacing

- <https://www.youtube.com/watch?v=Q1I4-8fVShA>
- <https://www.youtube.com/watch?v=7V69bSFBnVQ>

https://nptel.ac.in/content/storage2/courses/106108100/pdf/Teacher_Slides/mod3/M3L7.pdf



Govt. of India recognizes
VIT as an
Institution of Eminence (IoE)

THANK YOU !



ARVINDKR.NITT@GMAIL.COM

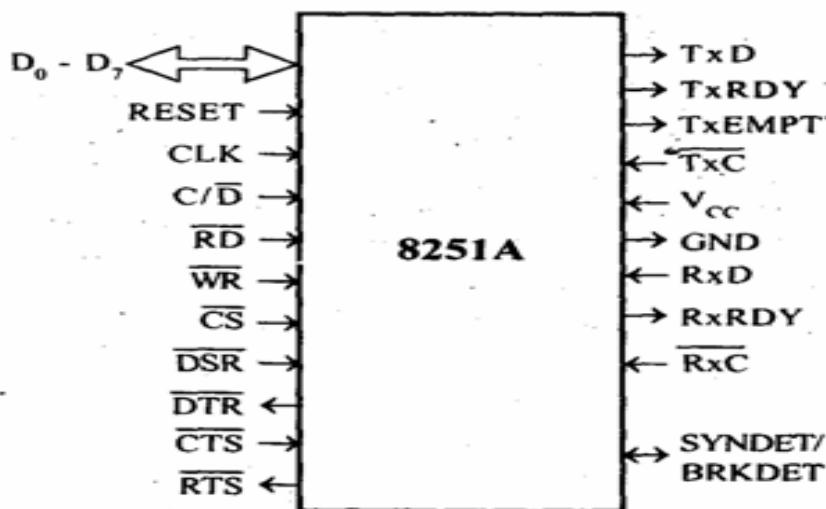
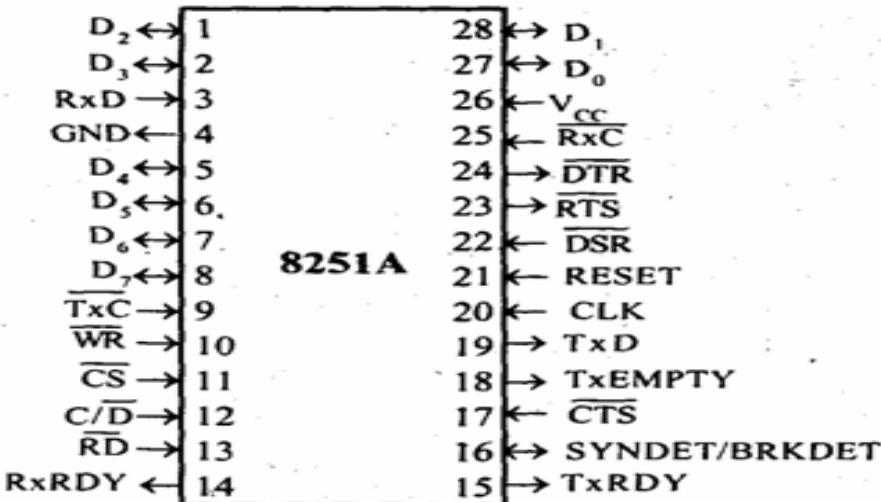


[HTTPS://SITES.GOOGLE.COM/VIEW/ARVINDK](https://sites.google.com/view/arvindk)

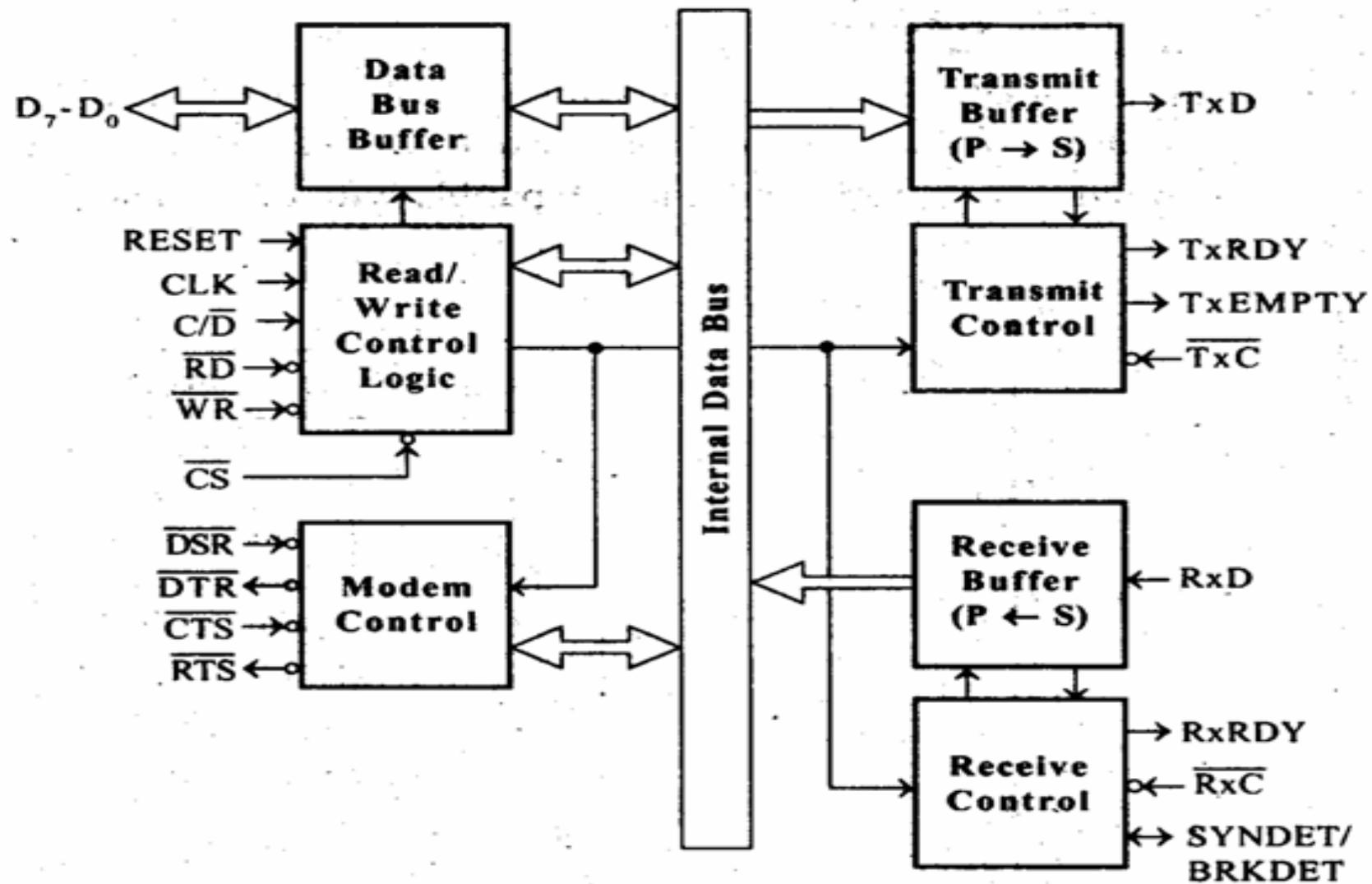
8251

- The 8251A is a programmable serial communication interface chip designed for synchronous and asynchronous serial data communication.
- It supports the serial transmission of data.
- It is packed in a 28 pin DIP.

Pin details



Architecture



Arch - details

- The functional block diagram of 825 1A consists five sections.

They are:

- Read/Write control logic
- Transmitter
- Receiver
- Data bus buffer
- Modem control.

Read/Write control logic

- The Read/Write Control logic interfaces the 8251A with CPU, determines the functions of the 8251A according to the control word written into its control register.
- It monitors the data flow.
- This section has three registers and they are control register, status register and data buffer.
- The active low signals RD, WR, CS and C/D(Low) are used for read/write operations with these three registers.

Read/Write control logic

- When C/D(low) is high, the control register is selected for writing control word or reading status word.
- When C/D(low) is low, the data buffer is selected for read/write operation.
- When the reset is high, it forces 8251A into the idle mode.
- The clock input is necessary for 8251A for communication with CPU and this clock does not control either the serial transmission or the reception rate.

Transmitter

- The transmitter section accepts parallel data from CPU and converts them into serial data.
- The transmitter section is double buffered, i.e., it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits.
- When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register.

Transmitter

- If buffer register is empty, then TxRDY is goes to high.
- If output register is empty then TxEMPTY goes to high.
- The clock signal, TxC (low) controls the rate at which the bits are transmitted by the USART.
- The clock frequency can be 1,16 or 64 times the baud rate.

Receiver

- The receiver section accepts serial data and convert them into parallel data
- The receiver section is double buffered, i.e., it has an input register to receive serial data and convert to parallel, and a buffer register to hold the parallel data.
- When the RxD line goes low, the control logic assumes it as a START bit, waits for half a bit time and samples the line again.
- If the line is still low, then the input register accepts the following bits, forms a character and loads it into the buffer register.

Receiver

- The CPU reads the parallel data from the buffer register.
- When the input register loads a parallel data to buffer register, the RxRDY line goes high.
- The clock signal RxC (low) controls the rate at which bits are received by the USART.
- During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.
- During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.

Modem control

- The MODEM control unit allows to interface a MODEM to 8251A and to establish data communication through MODEM over telephone lines.
- This unit takes care of handshake signals for MODEM interface.
- The 825 1A can be either memory mapped or I/O mapped in the system.
- 8251A in I/O mapped in the system is shown in the figure.
- Using a 3-to-8 decoder generates the chip select signals for I/O mapped devices.
- The address lines A4, A5 and A6 are decoded to generate eight chip select signals (IOCS-0 to IOCS-7) and in this, the chip select signal IOCS-2 is used to select 8251A.
- The address line A7 and the control signal IO / M(low) are used as enable for decoder.
- The address line A0 of 8085 is connected to C/D(low) of 8251A to provide the internal addresses.

Modem control

- The data lines D0 - D7 are connected to D0 - D7 of the processor to achieve parallel data transfer.
- The RESET and clock signals are supplied by the processor. Here the processor clock is directly connected to 8251A. This clock controls the parallel data transfer between the processor and 8251A.
- The output clock signal of 8085 is divided by suitable clock dividers like programmable timer 8254 and then used as clock for serial transmission and reception.

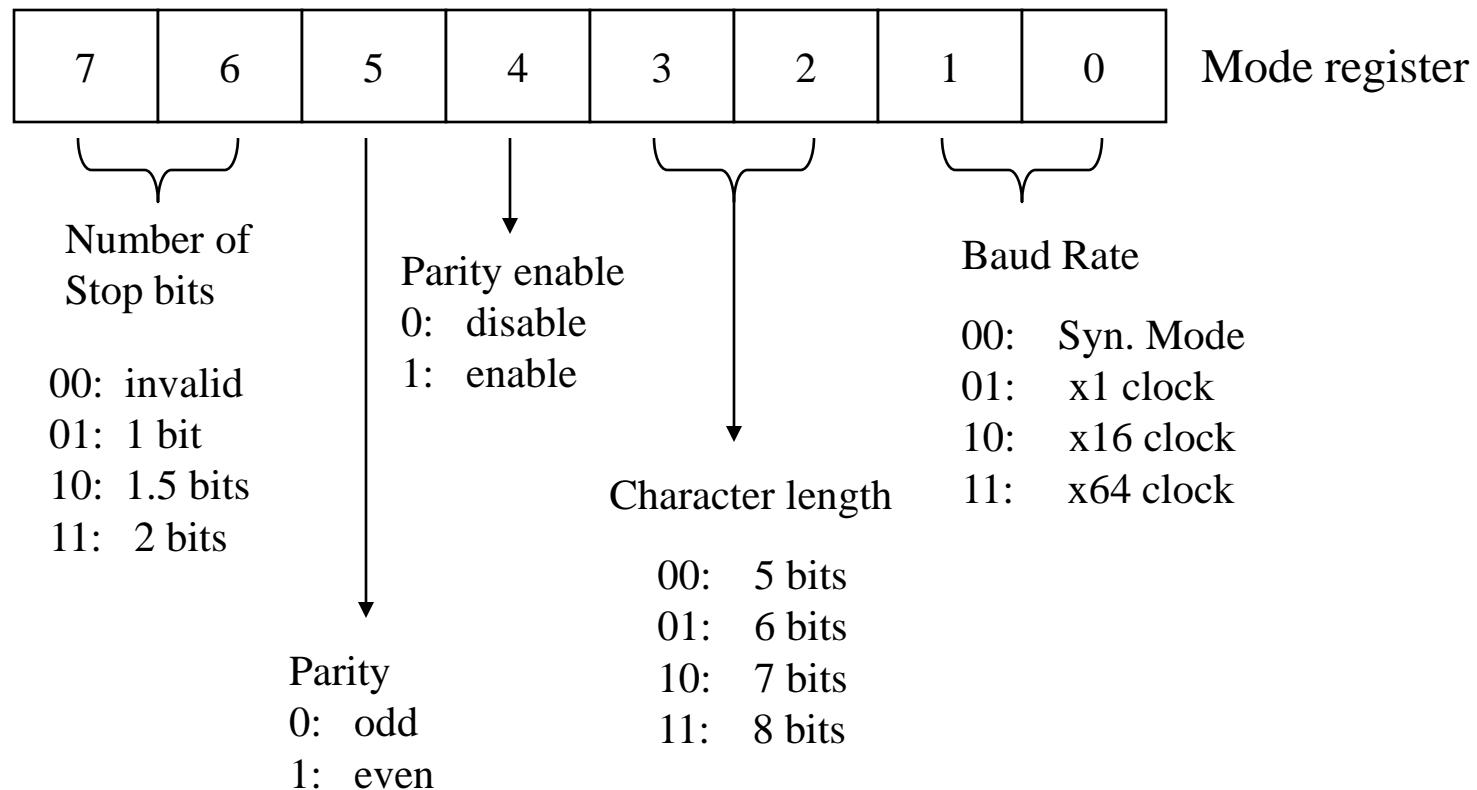
Modem control

- The TTL logic levels of the serial data lines and the control signals necessary for serial transmission and reception are converted to RS232 logic levels using MAX232 and then terminated on a standard 9-pin D-type connector.
- In 8251A the transmission and reception baud rates can be different or same.
- The device which requires serial communication with processor can be connected to this 9-pin D-type connector using 9-core cable
- The signals TxEMPTY, TxRDY and RxRDY can be used as interrupt signals to initiate interrupt driven data transfer scheme between processor and 8251

Modem control

- The CPU reads the parallel data from the buffer register.
- When the input register loads a parallel data to buffer register, the RxRDY line goes high.
- The clock signal RxC (low) controls the rate at which bits are received by the USART.
- During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.
- During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.

8251 mode register



8251 command register

EH	IR	RTS	ER	SBRK	RxE	DTR	TxE
----	----	-----	----	------	-----	-----	-----

TxE: transmit enable

DTR: data terminal ready

RxE: receiver enable

SBPRK: send break character

ER: error reset

RTS: request to send

IR: internal reset

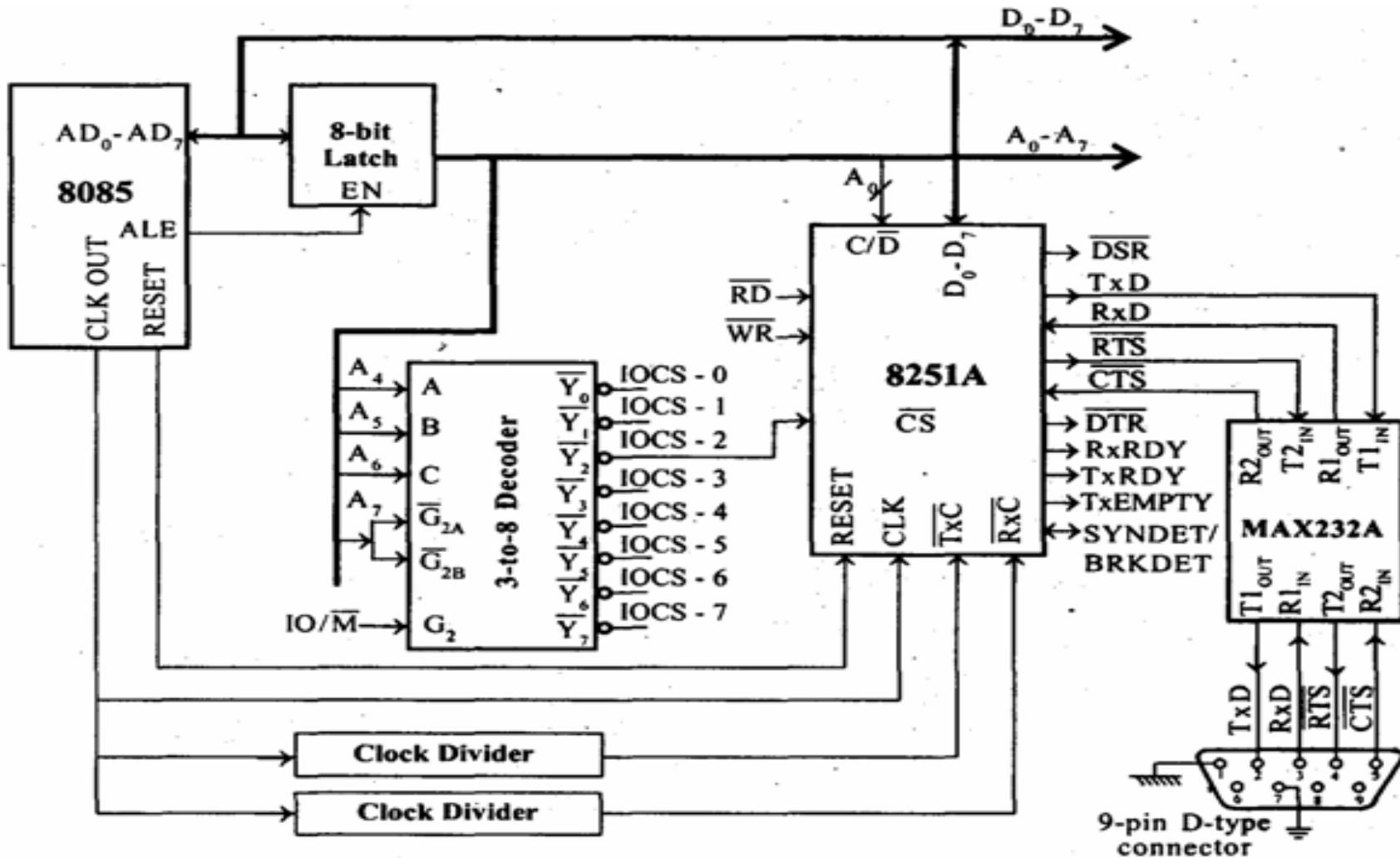
EH: enter hunt mode

8251 status register

DSR	SYNDET	FE	OE	PE	TxEMPTY	RxRDY	TxRDY
-----	--------	----	----	----	---------	-------	-------

TxRDY: transmit ready
RxRDY: receiver ready
TxEMPTY: transmitter empty
PE: parity error
OE: overrun error
FE: framing error
SYNDET: sync. character detected
DSR: data set ready

8251 interfaced with 8085



8255 PPI

- PPI

Programmable Peripheral
Interface

Intel 8255 PPI

PPI – Programmable Peripheral Interface

It is an I/O port chip used for interfacing I/O devices with microprocessor

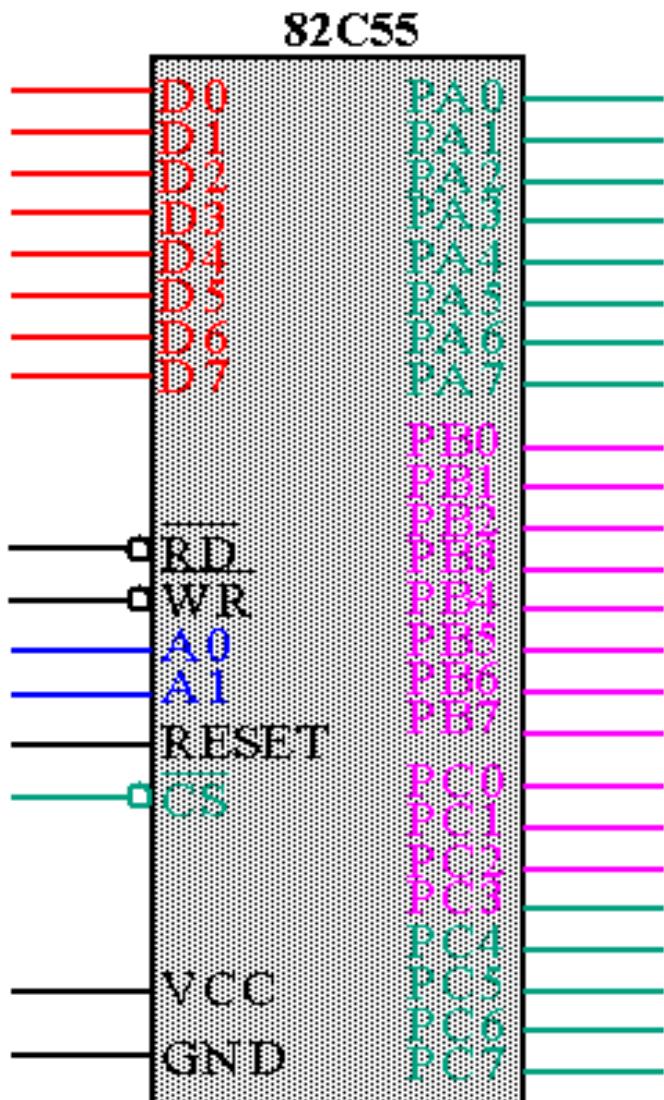
Very commonly used peripheral chip

Knowledge of 8255 essential for students in the Microprocessors lab for Interfacing experiments

About 82C55

- The 82C55 is a popular interfacing component, that can interface any TTL-compatible I/O device to a microprocessor.
- It is used to interface to the keyboard and a parallel printer port in PCs (usually as part of an integrated chipset).
- Requires insertion of wait states if used with a microprocessor using higher than an 8 MHz clock.
- PPI has 24 pins for I/O that are programmable in groups of 12 pins and has three distinct modes of operation.

82C55 : Pin Layout



Group A

Port A (PA7-PA0) and upper half of port C (PC7 - PC4)

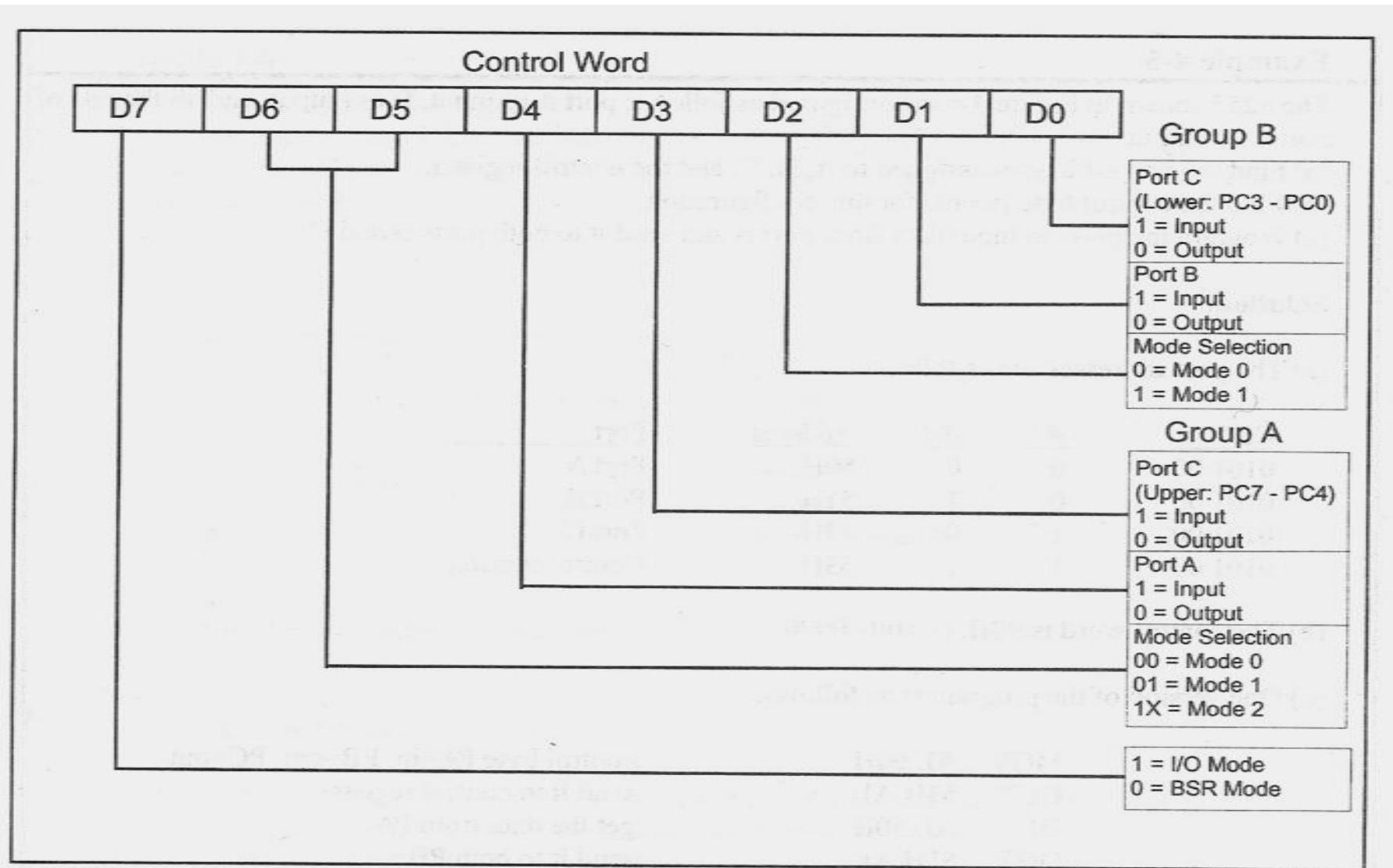
Group B

Port B (PB7-PB0) and lower half of port C (PC3 - PC0)

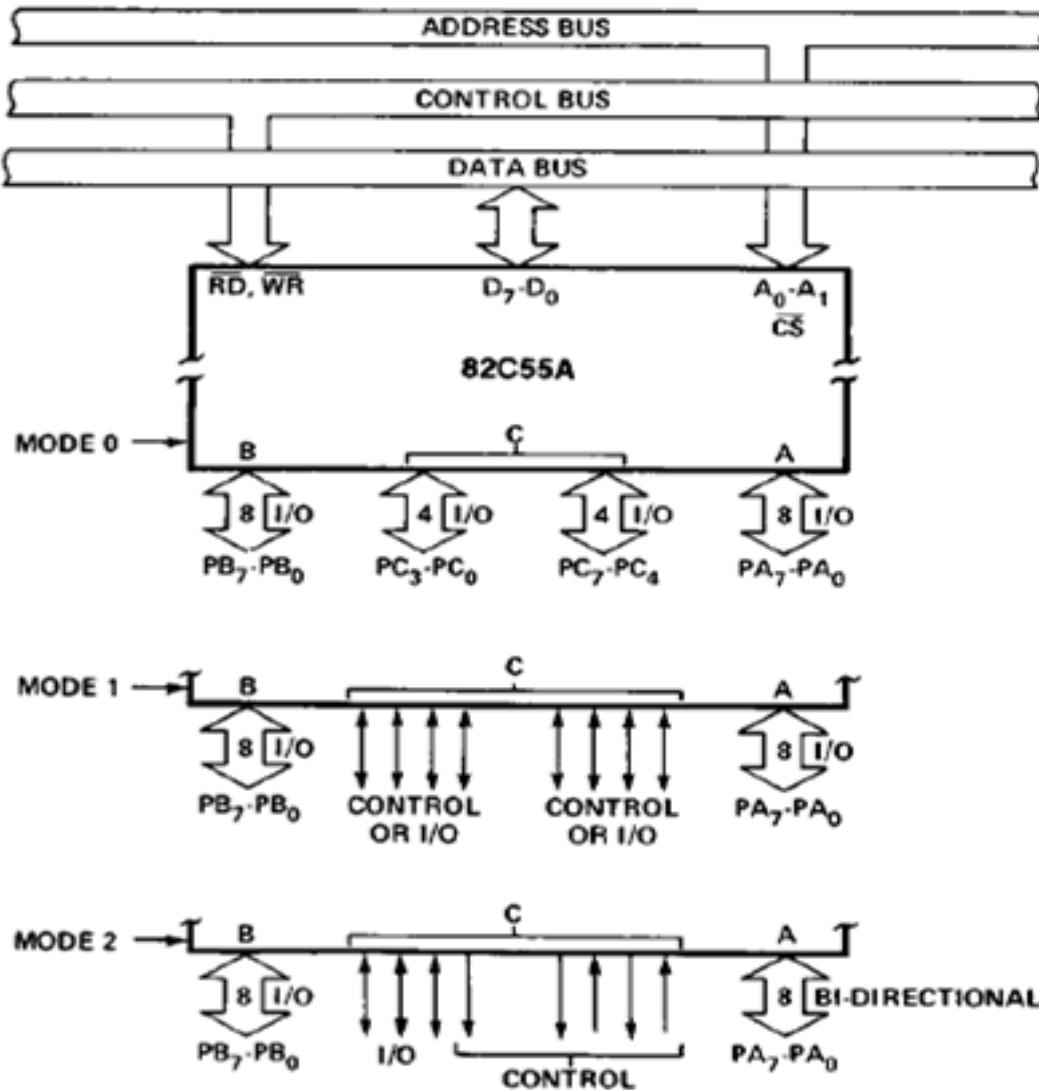
I/O Port Assignments

A_1	A_0	Function
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Command Register

8255 Control Word



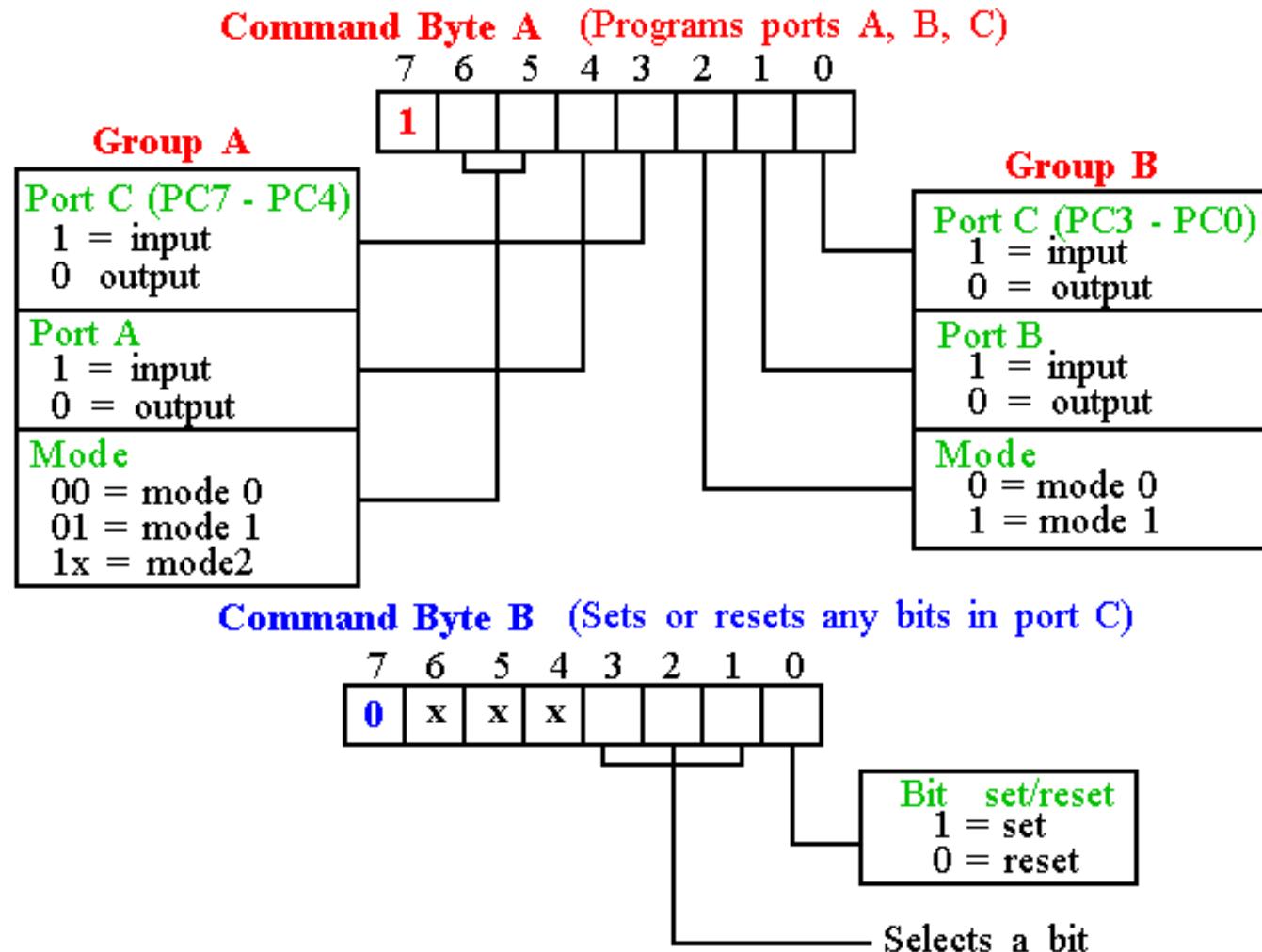
Basic Mode Definitions and Bus Int



- Mode 0
 - Basic I/O
- Mode 1
 - Strobe I/O
- Mode 2
 - Bi-Dir Bus

Programming 8255

- 8255 has three operation modes: *mode 0, mode 1, and mode 2*



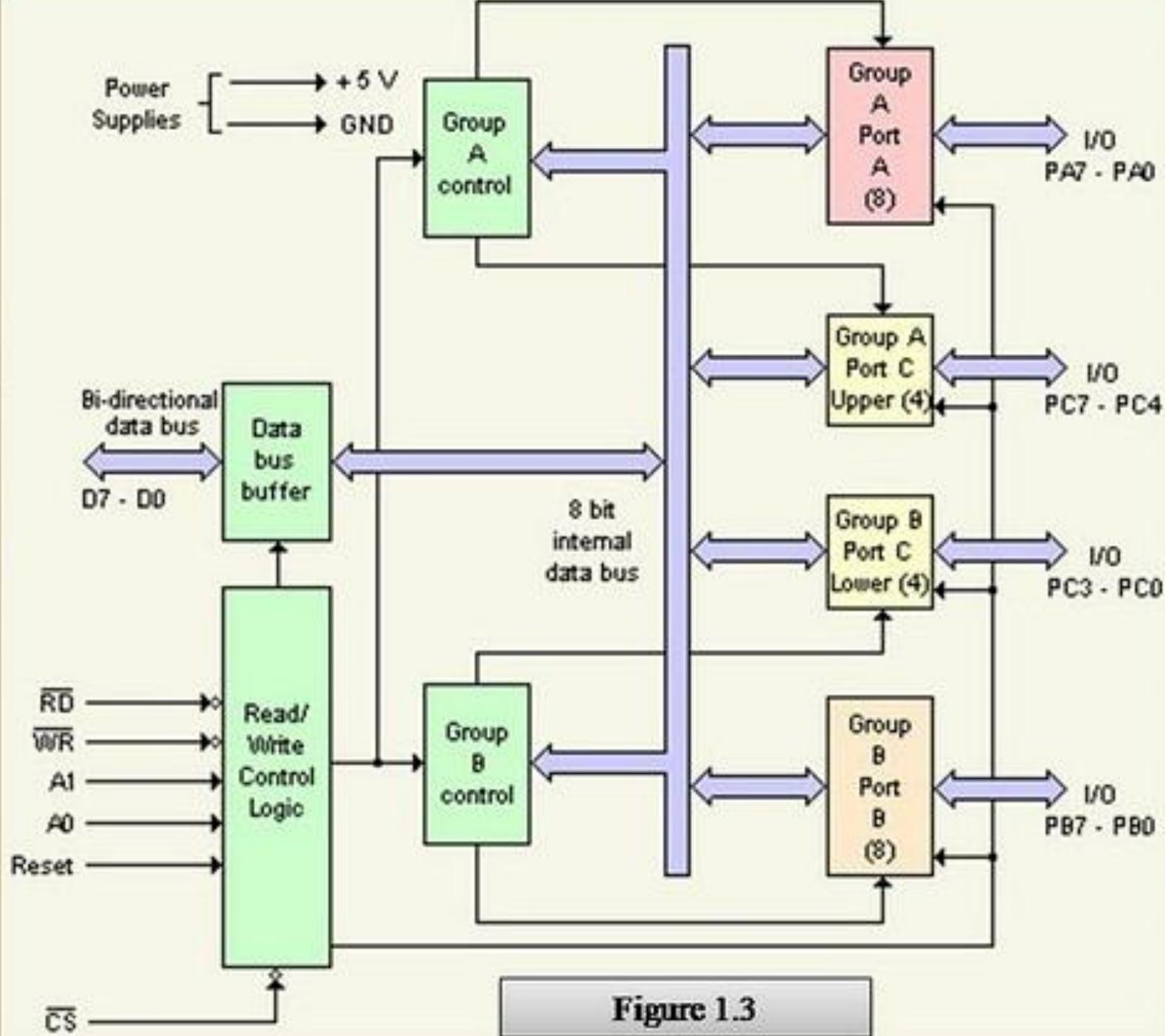


Figure 1.3

8255 PPI contd.

3 ports in 8255 from user's point of view

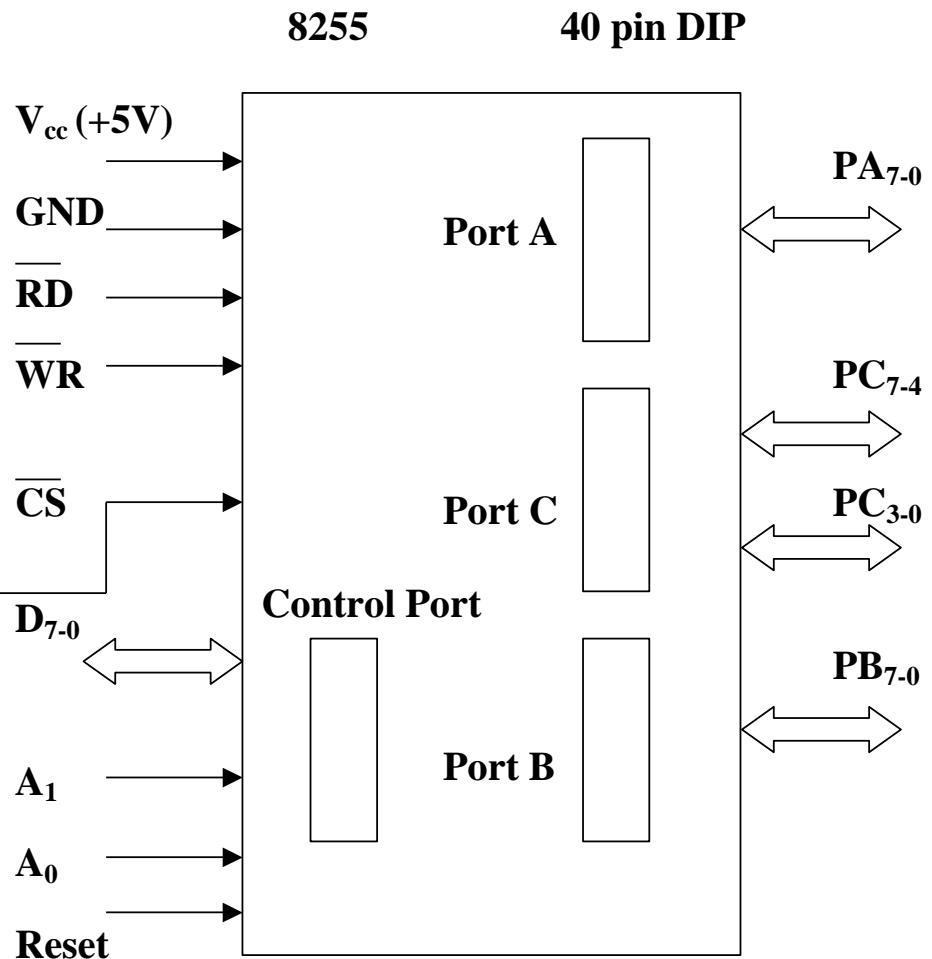
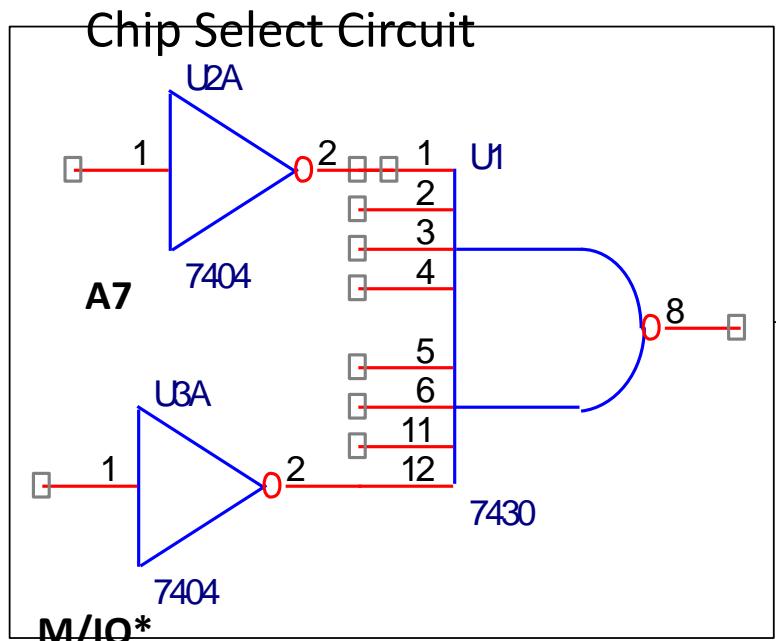
- Port A, Port B and Port C.

Port C composed of two independent 4-bit ports

- PC7-4 (PC Upper) and PC3-0 (PC Lower)

A1	A0	Selected port
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control port

Intel 8255 PPI



A7=0, A6=1, A5=1, A4=1, A3=1, A2=1, & M/IO*= 0

8255 PPI Contd.

There is also a Control port from the Processor point of view. Its contents decides the working of 8255.

When CS (Chip select) is 0, 8255 is selected for communication by the processor. The chip select circuit connected to the CS pin assigns addresses to the ports of 8255.

For the chip select circuit shown, the chip is selected when A7=0, A6=1, A5=1, A4=1, A3=1, A2=1, & M/IO*= 0

Port A, Port B, Port C and Control port will have the addresses as 7CH, 7DH, 7EH, and 7FH respectively.

8255 PPI Contd.

Mode 0: Simple Input or Output

In this mode, ports A, B are used as two simple 8-bit I/O ports
port C as two 4-bit ports.

Each port can be programmed to function as simply an input port or an output port. The input/output features in Mode 0 are as follows.

1. Outputs are latched.
2. Inputs are not latched.
3. Ports don't have handshake or interrupt capability.

8255 PPI Contd.

Mode 1: Input or Output with Handshake

In this mode, handshake signals are exchanged between the MPU and peripherals prior to data transfer.

The features of the mode include the following:

1. Two ports (A and B) function as 8-bit I/O ports.
They can be configured as either as input or output ports.
2. Each port uses three lines from Port C as handshake signals.
The remaining two lines of Port C can be used for simple I/O operations.
3. Input and Output data are latched.
4. Interrupt logic is supported.

8255 PPI Contd.

Mode 2: Bidirectional Data Transfer

This mode is used primarily in applications such as data transfer between two computers.

In this mode, Port A can be configured as the bidirectional port Port B either in Mode 0 or Mode 1.

Port A uses five signals from Port C as handshake signals for data transfer.

The remaining three signals from port C can be used either as simple I/O or as handshake for port B.

8255 Handshake signals

Where are the Handshake signals?

Port C pins act as handshake signals, when Port A and Port B are configured for other than Mode 0.

Port A in Mode 2 and Port B in Mode 1 is possible, as it needs only $5+3 = 8$ handshake signals

After Reset of 8255, Port A , Port B , and Port C are configured for Mode 0 operation as input ports.

8255 Handshake signals Contd.

PC2-0 are used as handshake signals by Port B when configured in Mode 1. This is immaterial whether Port B is configured as i/p or o/p port.

PC5-3 are used as handshake signals by Port A when configured as i/p port in Mode 1.

PC7,6,3 are used as handshake signals by Port A when configured as o/p port in Mode 1.

PC7-3 are used as handshake signals by Port A when configured in Mode 2.

8255 PPI Contd.

Port A can work in Mode 0, Mode 1, or Mode 2

Port B can work in Mode 0, or Mode 1

Port C can work in Mode 0 only, if at all

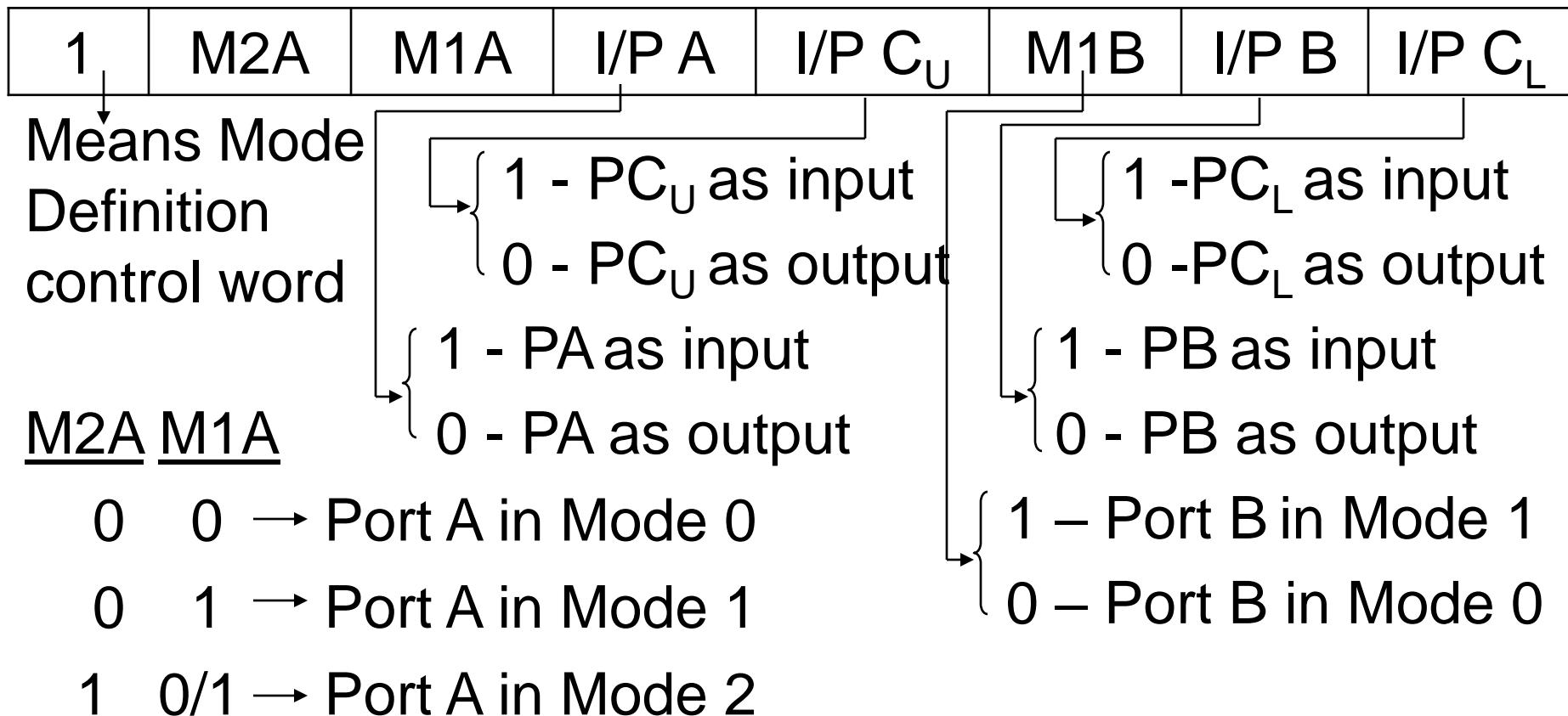
Port A, Port B and Port C can work in Mode 0

Port A and Port B can work in Mode 1

Only Port A can work in Mode 2

8255 MD Control word

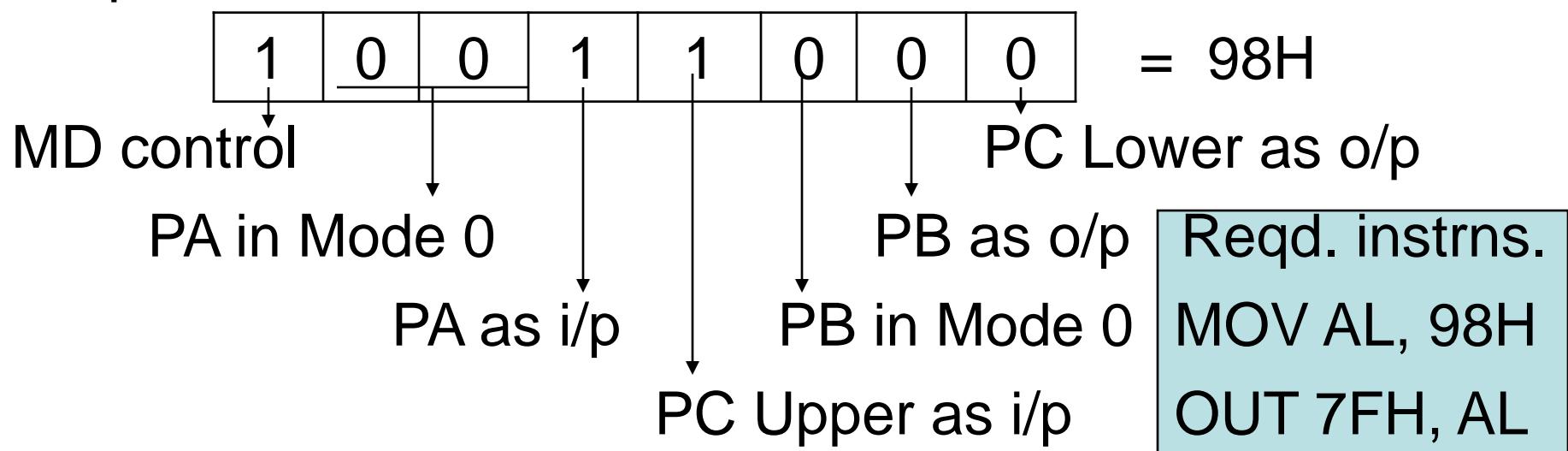
Control port having Mode Definition (MD) control word



8255 MD Control word Contd.

Ex. 1: Configure Port A as i/p in Mode 0, Port B as o/p in mode 0, Port C (Lower) as o/p and Port C (Upper) as i/p ports.

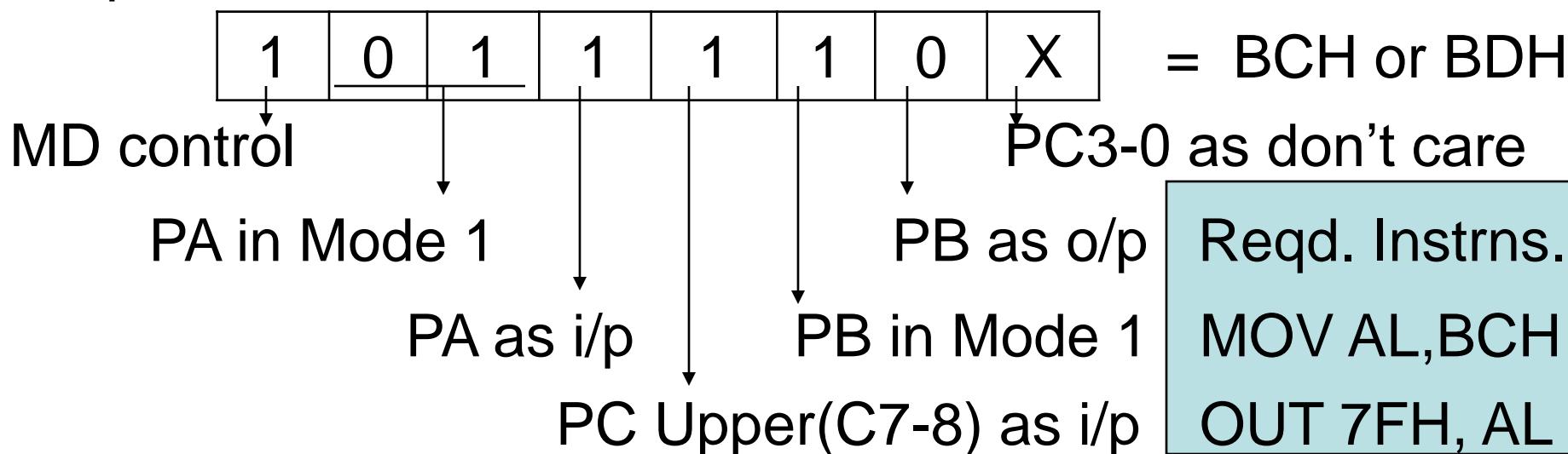
Required MD control word:



8255 MD Control word Contd.

Ex. 2: Configure Port A as i/p in Mode 1, Port B as o/p in mode 1, Port C7-8 as i/p ports. (PC5-0 are handshake lines, some i/p lines and others o/p. So they are shown as X)

Required MD control word:



8255 Contd.

There are 2 control words in 8255

Mode Definition (MD) Control word and
Port C Bit Set / Reset (PCBSR) Control Word

MD control word configures the ports of 8255

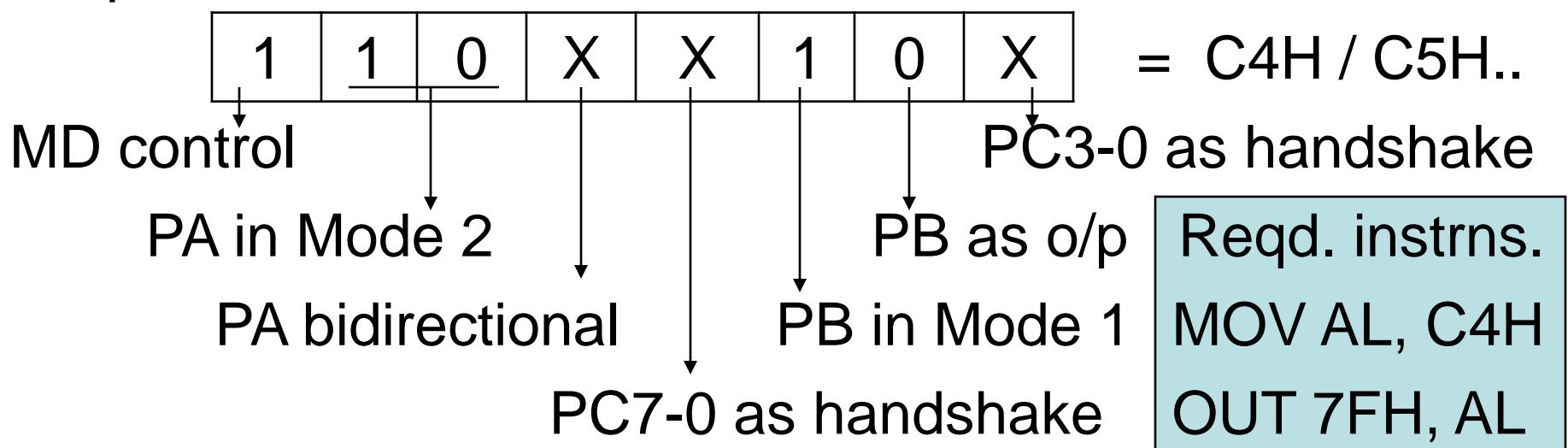
- as i/p or o/p in Mode 0, 1, or 2

PCBSR control word is used to set to 1 or reset to 0
any one selected bit of Port C

8255 MD Control word Contd.

Ex. 3: Configure Port A in Mode 2, Port B as o/p in mode 1.
(PC5-0 are handshake lines for Port A and PC2-0 are handshake signals for port B)

Required MD control word:



8255 PCBSR Control word

Control port having Port C Bit Set / Reset control word

PC bit set / reset control word							
0	X	X	X	SB2	SB1	SB0	S/R*
PC bit set / reset control	Don't cares	Select bit of PC to be set / reset		1 - Set to 1			
		0 0 0	→ Bit 0 of Port C	0 - Reset to 0			
		0 0 1	→ Bit 1 of Port C				
		:					
		:					
		1 1 1	→ Bit 7 of Port C				

8253 / 8254 Timer

- To program a given counter to divide the CLK input frequency, one must send the divisor to that specific counter's register.
- Although all three counters share the same control register, the divisor registers are separate for each counter
- Example: given the port addresses for
8253/54: Counter 0: 94H Counter 1:
95H
Counter 2: 96H Control Reg: 97H

8253 / 8254 Timer

- Task1: program counter 0 for binary counter for mode 3 to divide CLK0 by number 4282 (BCD)

MOV AL, 0011 0111B

OUT 97H, AL

MOV AX, 4282H (BCD needs H)

OUT 94H, AL (Low Byte)

MOV AL, AH

OUT 94H, AL (High Byte)

- OUT0 = CLK0 / 4282

Shape of the 8253/54 Output

- Given $\text{CLK} = 1.193 \text{ MHz}$, the clock period of input frequency is 838 ns
- If the number N loaded into the counter is even, both high and low pulse are the same length, which is $N/2 * 838 \text{ ns}$
- If the number N loaded into the counter is odd, the high pulse is $(N+1)/2 * 838 \text{ ns}$ and the low pulse is $(N-1)/2 * 838 \text{ ns}$
- → If N is odd, the high portion of the output square wave is slightly wider than the low portion

8253/54 Operation Modes

- Mode 0: Interrupt on terminal count
 - The output is initially low, and remain low for the duration of the count if GATE=1. When the terminal count is reached, the output will go high and remain high until a new control word or new count number is loaded
 - Width of low pulse = $N * T$, where T is clock period
 - Example: GATE=1 and CLK = 1 MHz
Clock count N = 1000

8253/54 Operation Modes

- Mode 0: Interrupt on terminal count
 - If GATE becomes low at the middle of the count, the count will stop and the output will be low. The count resumes when the GATE becomes high again → This in effect adds to the total time the output is low.
- Mode 1: HW triggered / programmable one shot
 - The triggering must be done through the GATE input by sending a 0-to-1 pulse to it.
 - Steps: 1) Load the count register
 - 2) A 0-to-1 pulse must be sent to the GATE input to trigger the count

8253/54 Operation Modes

- Mode 1: HW triggered / programmable one shot
 - In Mode 1, after sending the 0-to-1 pulse to GATE, OUT becomes low and stays low for a duration of $N*T$, then becomes high and stays high until the GATE is triggered again
 - If during the activation, a retriggered happened, then restart the down counting
- Mode 2: Rate Generator (Divide-by-N counter)
 - In Mode2, if GATE=1, OUT will be high for $N*T$, goes low only for one clock pulse, then counter is reloaded automatically, and the process

8253/54 Operation Modes

- Mode 3: Square wave rate generator
 - Most commonly used
- Mode 4: Software triggered strobe
 - Similar to Mode2, except that the counter is not reloaded automatically
 - In Mode4, if GATE=1, the output will go high when loading the count, it will stay high for duration $N*T$. After the count reaches zero, it becomes low for one clock pulse, then goes high again and stays high until a new command word or new count is loaded
 - To repeat the strobe, the count must be reloaded

8253/54 Operation Modes

- Mode 5: Hardware triggered strobe
 - Similar to Mode4, except that the triggering must be done with the GATE input
 - The count starts only when a 0-to-1 pulse is sent to the GATE input
 - If GATE retriggered during the counting, it will restart the down counting

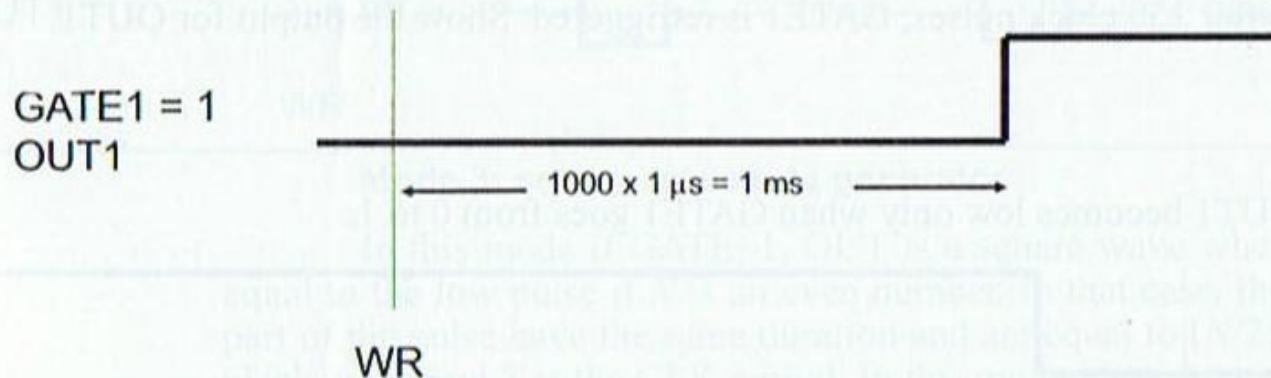
8253 / 8254 Timer

Example 5-7

Assume that GATE1 = 1 and CLK1 = 1 MHz, and the clock count N = 1000. Show the output of OUT1 if it is programmed in mode 0.

Solution:

The clock period of CLK1 is $1 \mu\text{s}$; therefore, OUT1 is low for $1000 \times 1 \mu\text{s} = 1 \text{ ms}$, before it goes high, as shown in the following diagram.



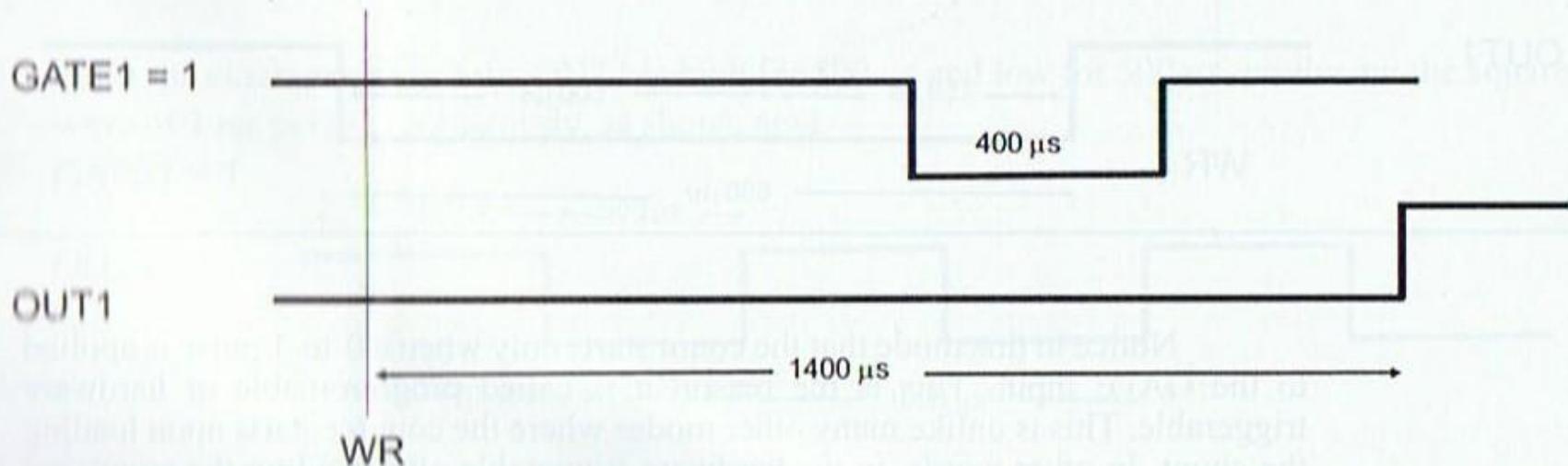
8253 / 8254 Timer

Example 5-8

In Example 5-7, assume that GATE1 becomes zero for 400 μ s. What is the width of the low pulse for OUT1?

Solution:

It is $1000 \mu\text{s} + 400 \mu\text{s} = 1400 \mu\text{s}$, as shown next.



Mode 1: programmable one-shot

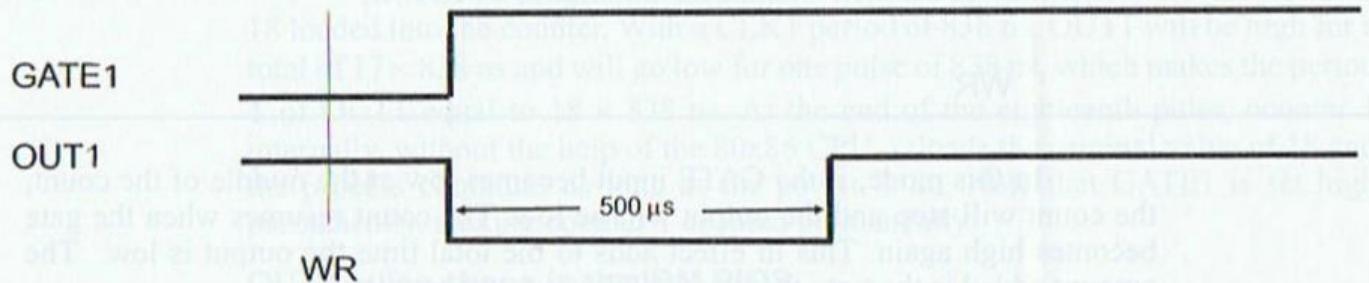
8253 / 8254 Timer

Example 5-9

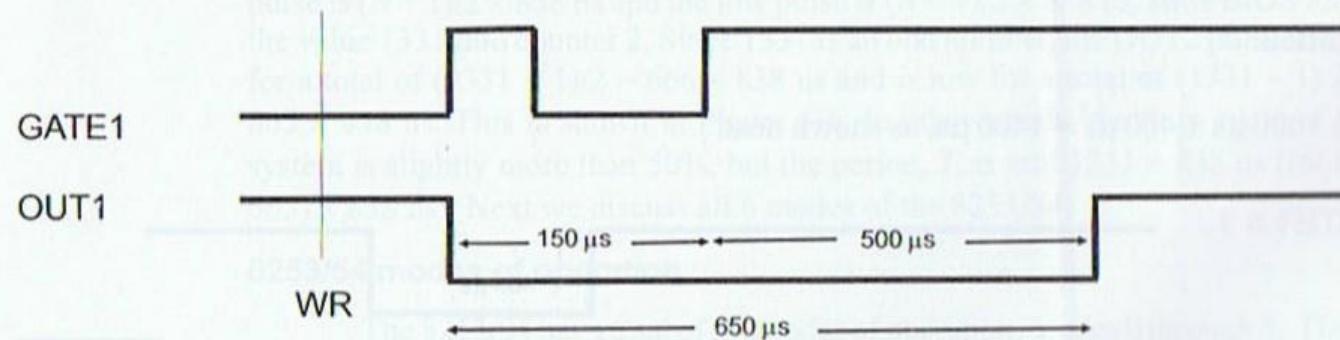
- If $\text{CLK1} = 1 \text{ MHz}$ and $N = 500$, show the output for OUT1 if it is programmed for mode 1.
- Assume that after 150 clock pulses, GATE1 is retriggered. Show the output for OUT1.

Solution:

- Notice that OUT1 becomes low only when GATE1 goes from 0 to 1.



- If GATE1 is retriggered after 150 clock pulses, COUNT1 is reloaded with $N = 500$ and the count starts all over again, making the OUT1 pulse duration 650 μs as shown next.



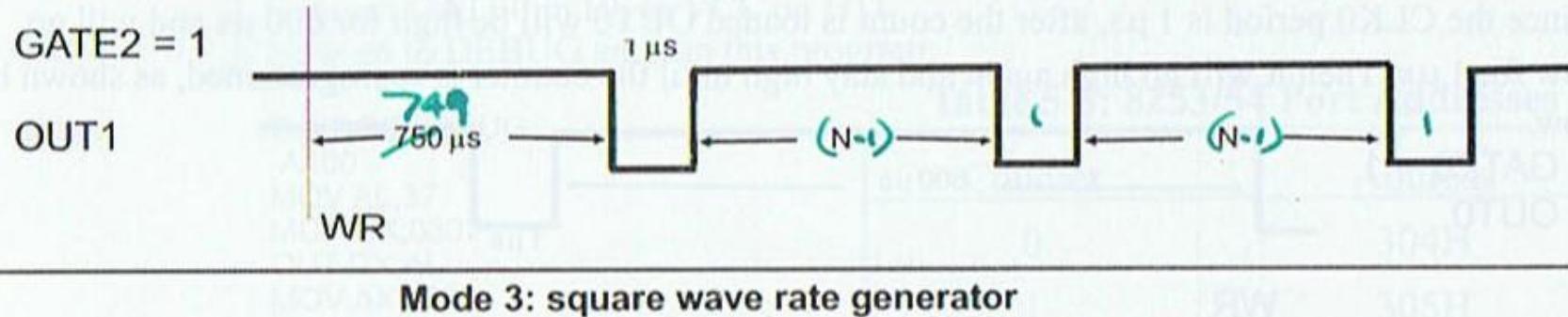
8253 / 8254 Timer

Example 5-10

If $\text{CLK2} = 1 \text{ MHz}$, $\text{GATE2} = 1$, and $N = 750$, show OUT2 if COUNT2 is programmed for mode 2.

Solution:

Notice that the count is reloaded automatically and the counter continues to produce OUT2 .



8253 / 8254 Timer

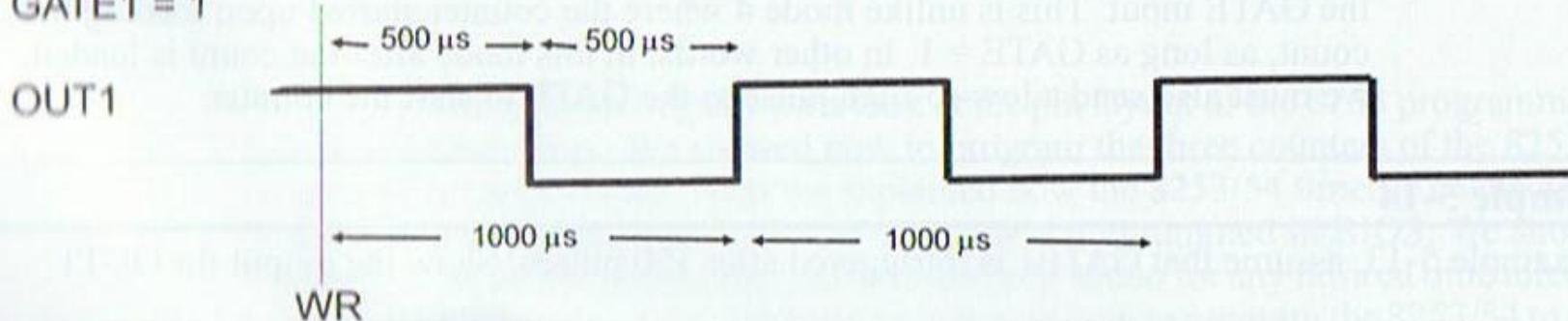
Example 5-11

If $\text{CLK2} = 1 \text{ MHz}$, $\text{GATE1} = 1$, $N = 1000$, show OUT1 if COUNT1 is programmed for mode 3.

Solution:

Since the clock period is $1 \mu\text{s}$, OUT1 is high for $500 \mu\text{s}$ and low for $500 \mu\text{s}$, producing the square wave of 1 ms period continuously, as shown next.

$\text{GATE1} = 1$



Mode 4: software triggered strobe

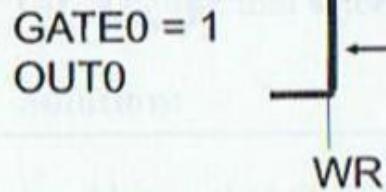
8253 / 8254 Timer

Example 5-12

If $\text{CLK}_0 = 1 \text{ MHz}$, $\text{GATE}_0 = 1$, and $N = 600$, show the shape of OUT_0 where counter 0 is programmed for mode 4.

Solution:

Since the CLK_0 period is $1 \mu\text{s}$, after the count is loaded OUT_0 will be high for $600 \mu\text{s}$ and will go low for $1 \mu\text{s}$. Then it will go high again and stay high until the counter is reprogrammed, as shown below.



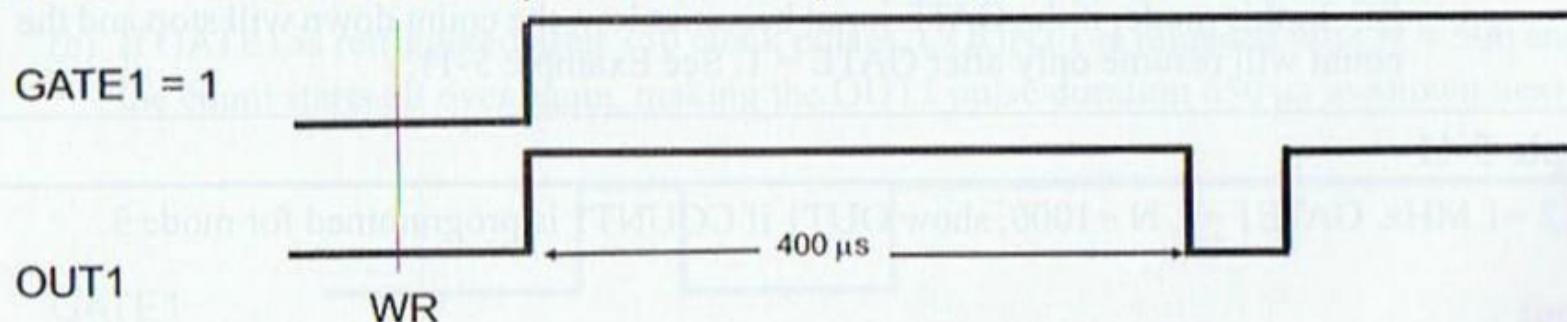
8253 / 8254 Timer

Example 5-13

If $\text{CLK1} = 1 \text{ MHz}$, and $N = 400$, show the output for OUT1 if it is programmed for mode 5.

Solution:

Notice that the count starts only when the 0-to-1 pulse is applied to GATE1.



Mode 5: hardware triggered strobe

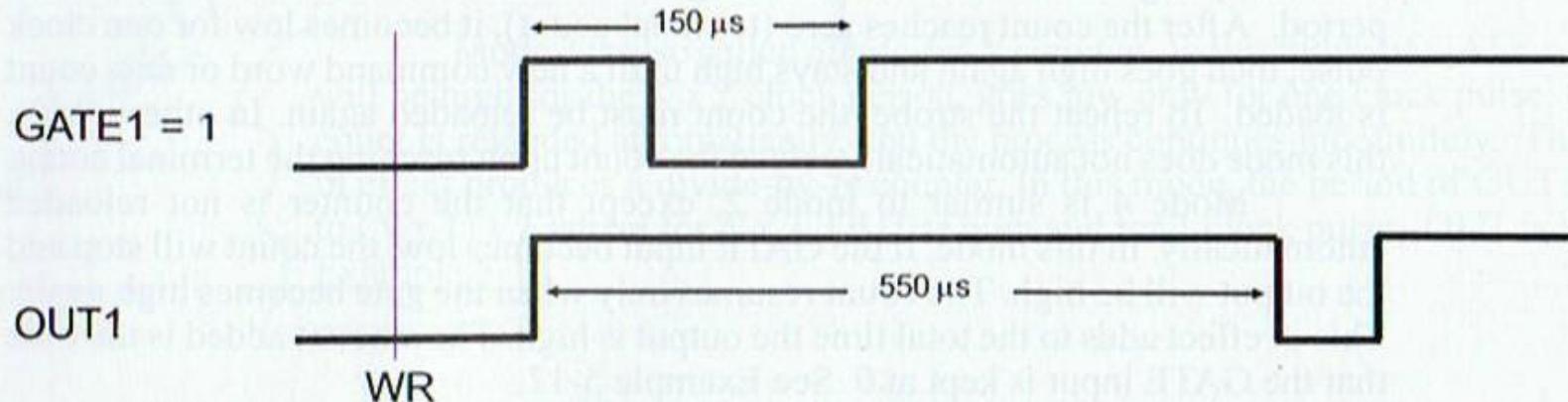
8253 / 8254 Timer

Example 5-14

In Example 5-13, assume that GATE1 is retriggered after 150 pulses. Show the output for OUT1.

Solution:

If GATE1 is retriggered after 150 clock pulses into the countdown, COUNT1 is reloaded with $N = 400$ and the counts begins again, making the OUT1 pulse duration 550 μs , as shown next.



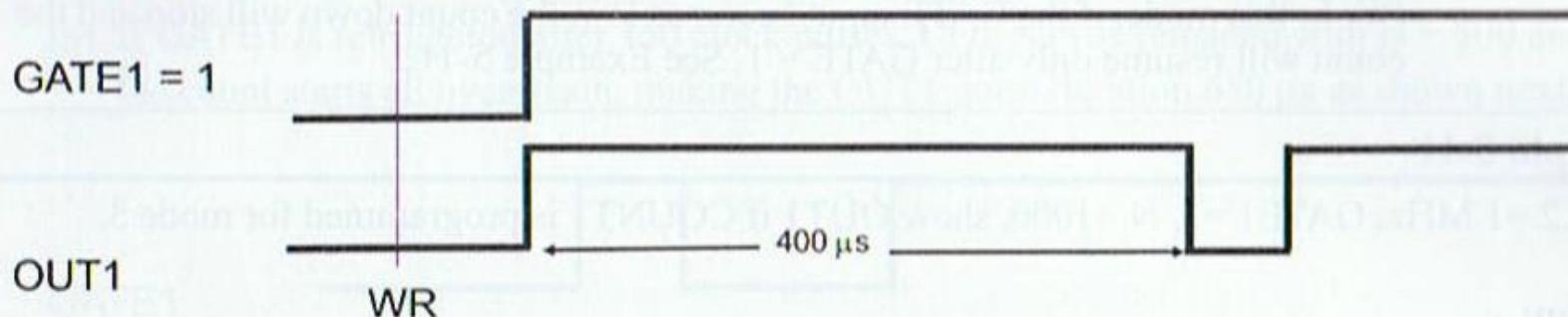
8253 / 8254 Timer

Example 5-13

If $\text{CLK1} = 1 \text{ MHz}$, and $N = 400$, show the output for OUT1 if it is programmed for mode 5.

Solution:

Notice that the count starts only when the 0-to-1 pulse is applied to GATE1.



Mode 5: hardware triggered strobe

8259

- 8259 is Programmable Interrupt Controller (PIC)
- It is a tool for managing the interrupt requests.
- 8259 is a very flexible peripheral controller chip: PIC can deal with up to 64 interrupt inputs
 - interrupts can be masked
 - various priority schemes can also programmed.
- originally (in PC XT) it is available as a separate IC
- Later the functionality of (*two PICs*) is in the motherboards chipset.

Pin description

- 8-bit bi-directional data bus, one address line is needed, the direction of data flow is controlled by RD and WR.
- CS is as usual connected to the output of the address decoder.
- Interrupt requests are output on INT which is connected to the INTR of the processor. Int. acknowledgment is received by INTA.
- IR0-IR7 allow 8 separate interrupt requests to be inputted to the PIC.
- sp/en=1 for master , sp/en=0 for slave.
- CAS0-3 inputs/outputs are used when more than one PIC to cascaded.

FIGURE 9-4 Block diagram and pin definitions for the 8259A Programmable Interrupt Controller (PIC). (Courtesy of Intel Corporation.)

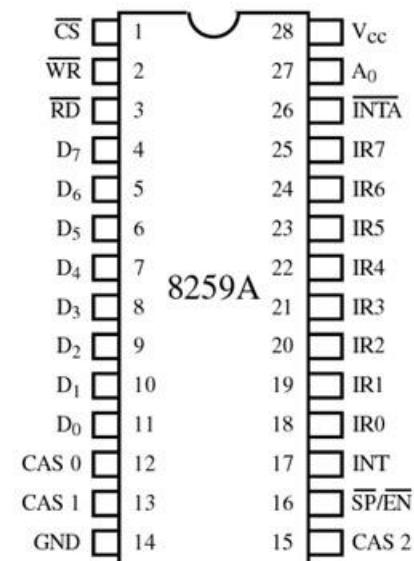
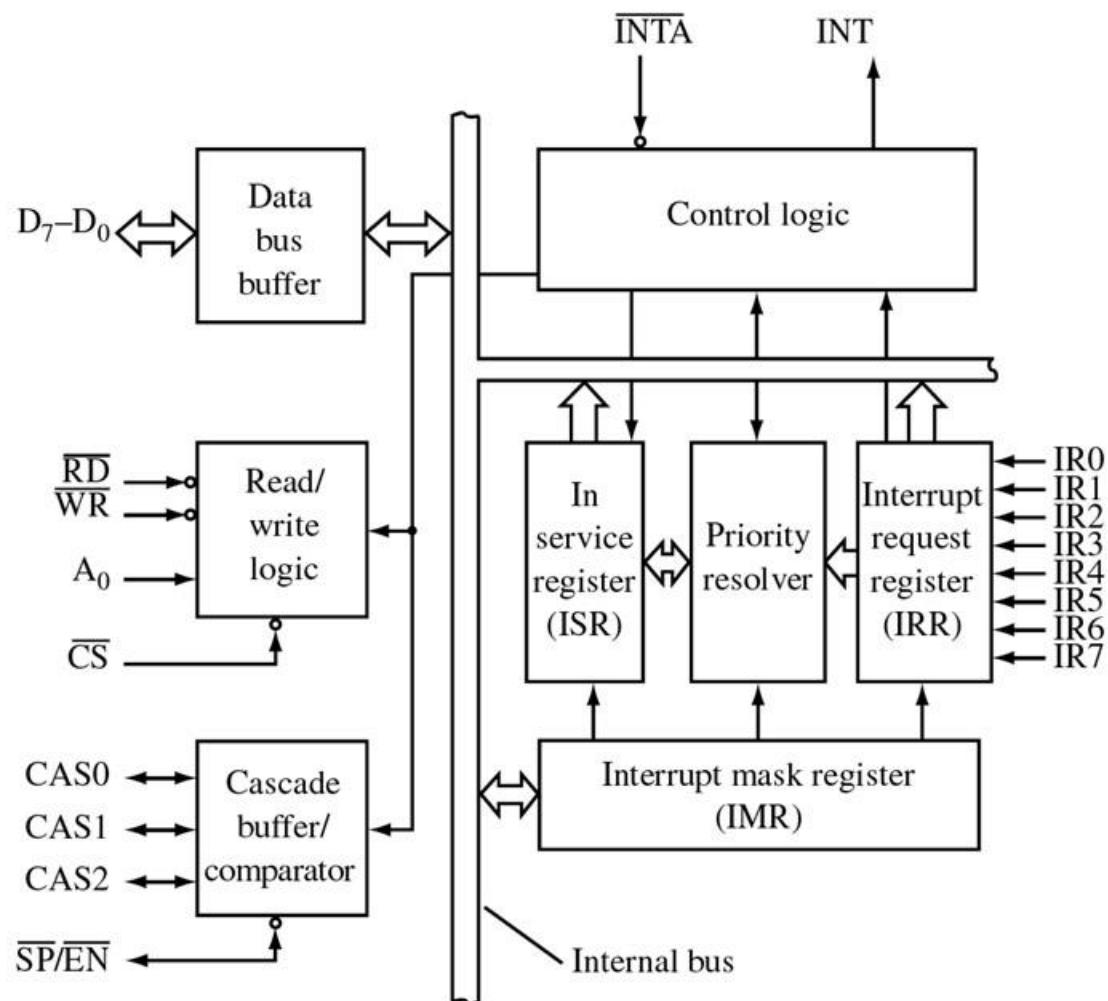


FIGURE 9-5 Interfacing the PIC to the 386 and 486 processors. Two I/O ports are required.

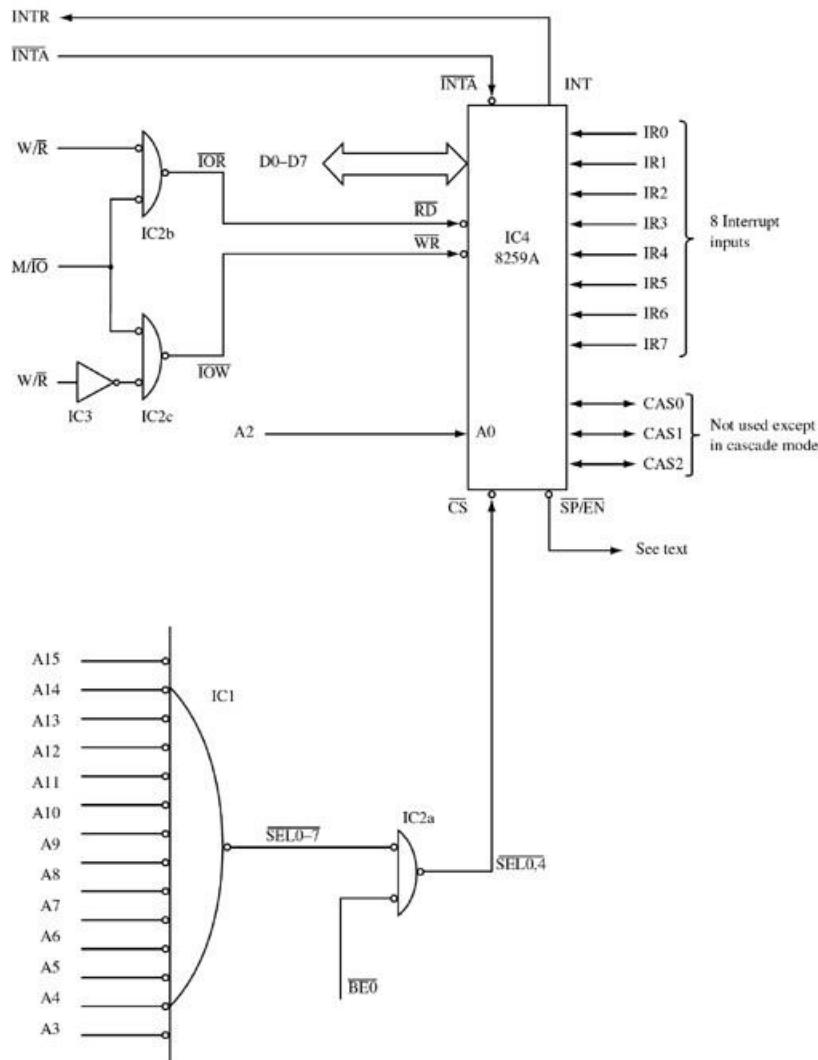
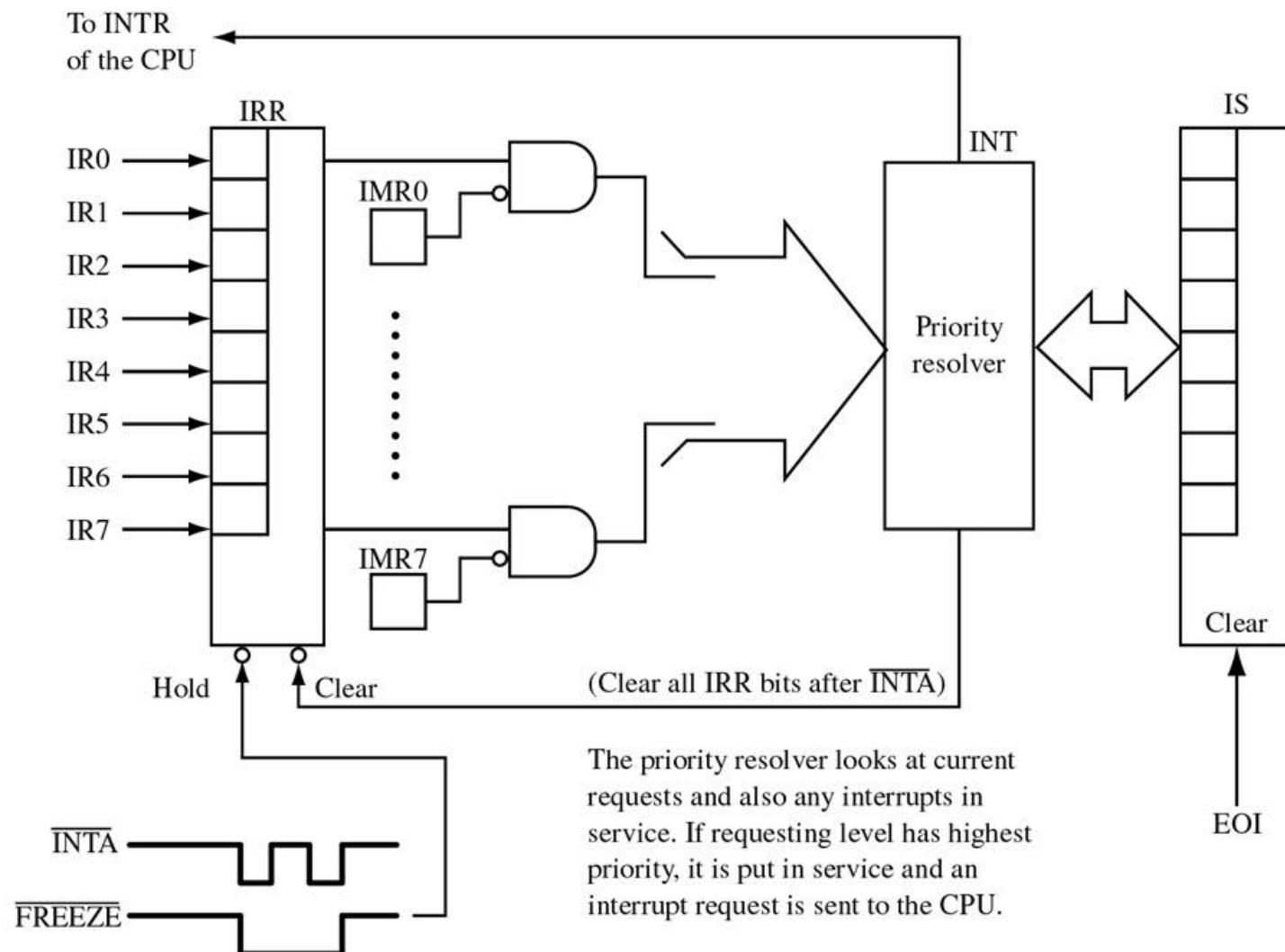
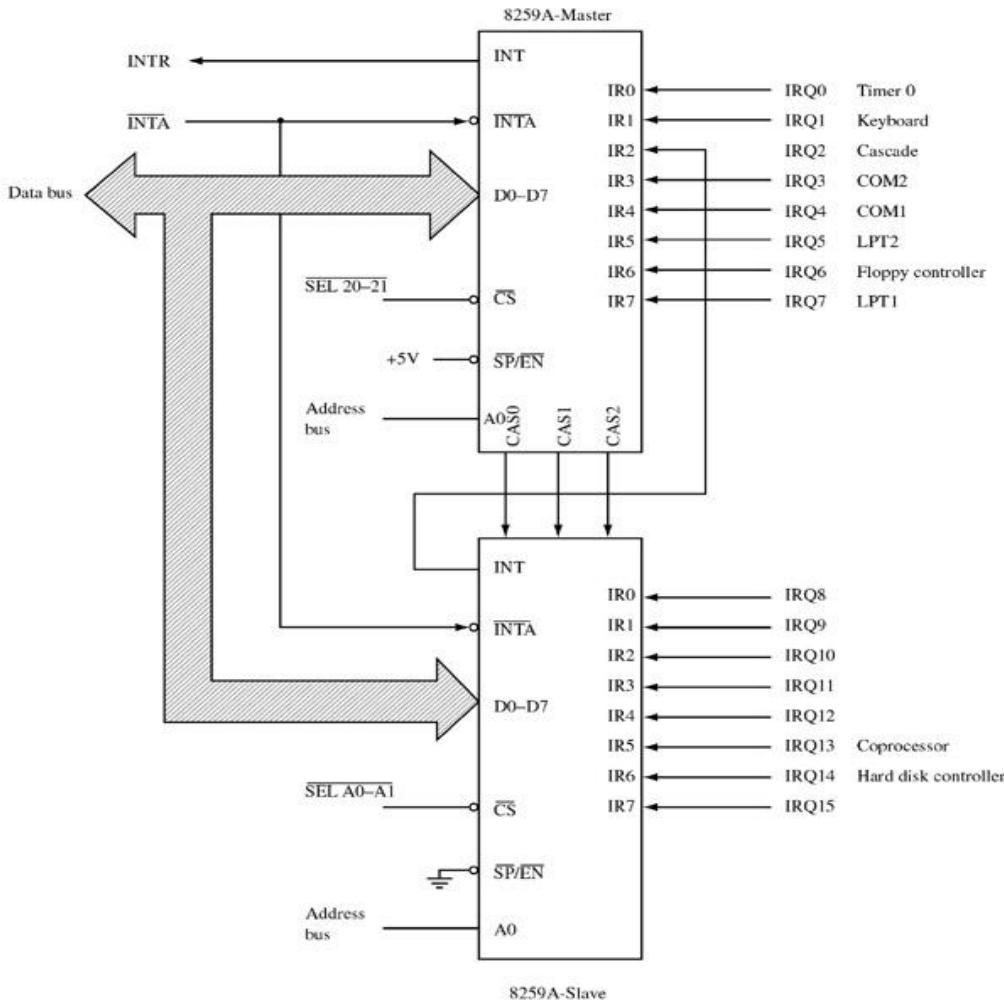


FIGURE 9-7 All interrupt requests must pass through the PIC's interrupt request register (IRR) and interrupt mask register (IMR). If put in service, the appropriate bit of the in-service (IS) register is set.



Example of two cascaded PICs



OPERATION

- PIC is to be initialized and programmed to control its operation.
- The operation in simple words:

When an interrupt occurs , the PIC determines the highest priority, activates the processor via its INTR input, and sends the type number onto the data bus when the processor acknowledges the interrupt.
- Priority:

What is used in PC is *fully nested mode*. That is the lowest numbered IRQ input has highest priority. Lower priority interrupts will not be forwarded to the processor until the higher priority interrupts have been serviced.

FIGURE 9-8 (a) Simultaneous interrupt requests arrive on IR4 and IR6. IR4 has highest priority and its IS bit is set as the IR4 service routine is put in service. (b) The IR4 service routine issues a rotate-on-nonspecific-EOI command, resetting IS4 and assigning it lowest priority. IR6 is now placed in service. (c) The IR6 service routine issues a rotate-on-nonspecific-EOI command, resetting IS6 and assigning it lowest priority.

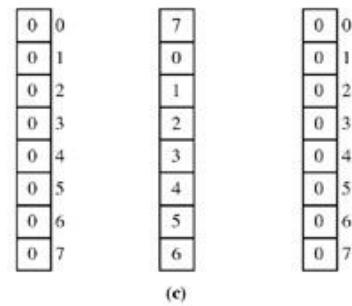
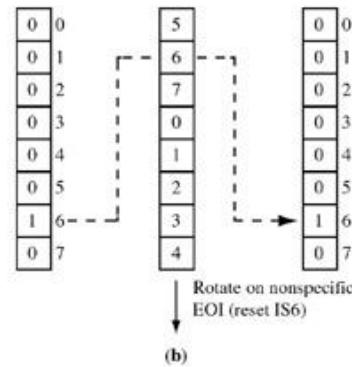
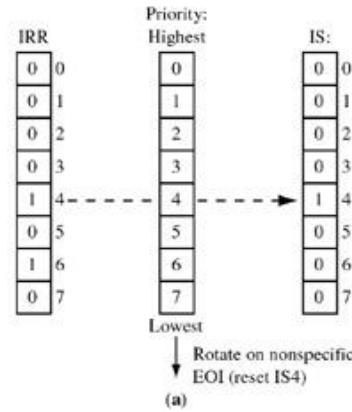
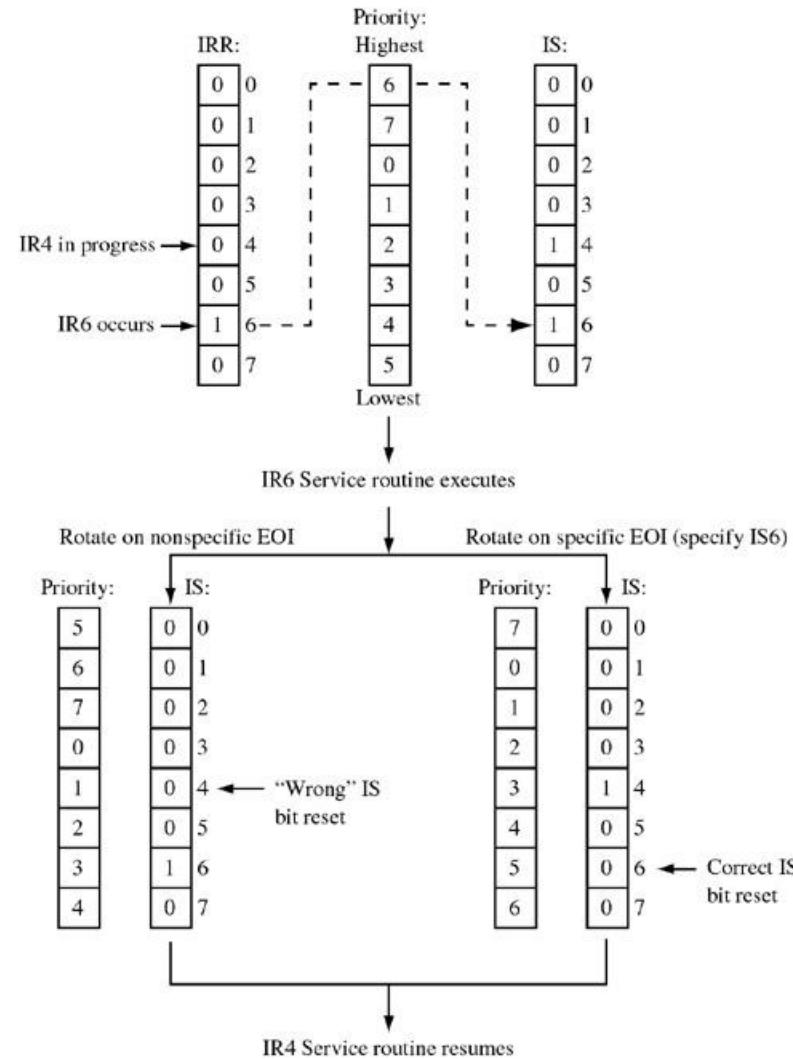
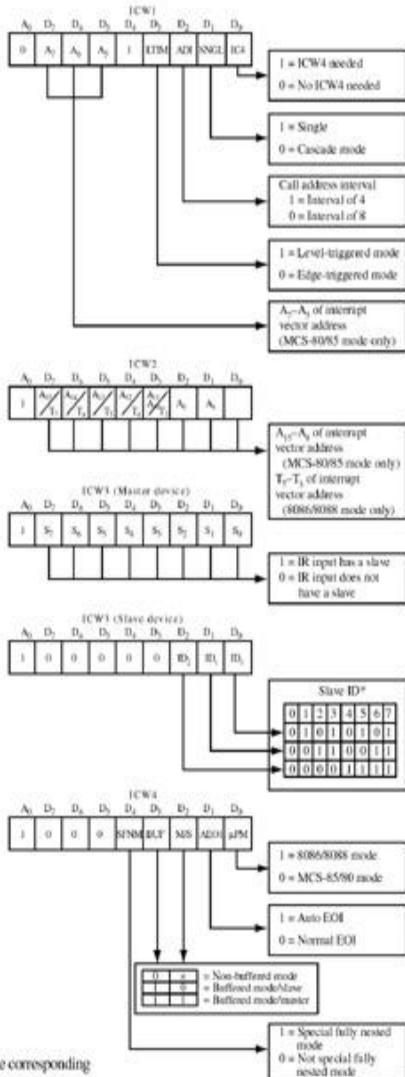


FIGURE 9-9 Example illustrating the difference between the rotate-on-nonspecific-EOI command and the rotate-on-specific-EOI command.



Modes

- Fully Nested mode
- Special Fully Nested mode
- Nonspecific Rotating
- Specific Rotating
- Special Mask
- Polling



* Slave ID is equal to the corresponding master IR input.

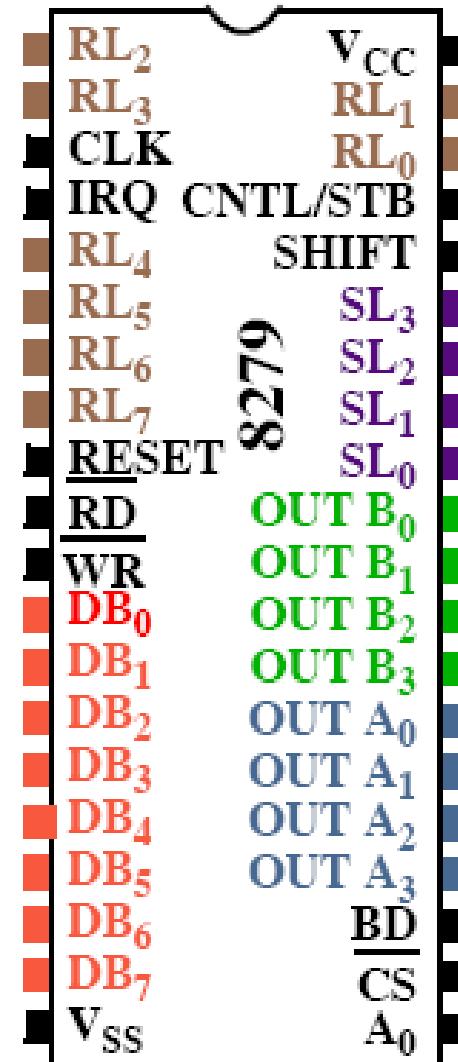
Features of 8279

The important features of 8279 are,

- Simultaneous keyboard and display operations.
- Scanned keyboard mode.
- Scanned sensor mode.
- 8-character keyboard FIFO.
- 1 6-character display.
- Right or left entry 1 6-byte display RAM.
- Programmable scan timing.

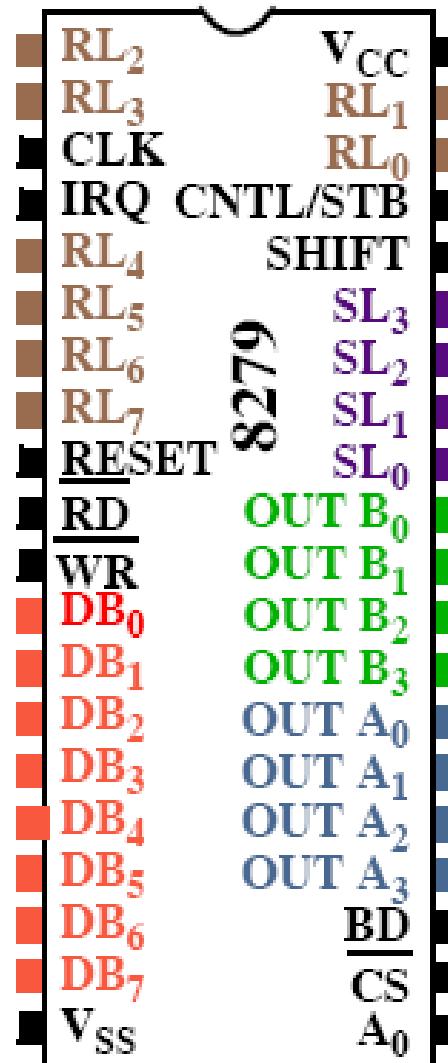
Pin details

- A0: Selects data (0) or control/status (1) for reads and writes between micro and 8279.
- BD: Output that blanks the displays.
- CLK: Used internally for timing. Max is 3 MHz.
- CN/ST: Control/strobe, connected to the control key on the keyboard.
- CS: Chip select that enables programming, reading the keyboard, etc.
- DB7-DB0: Consists of bi-directional pins that connect to data bus on micro.

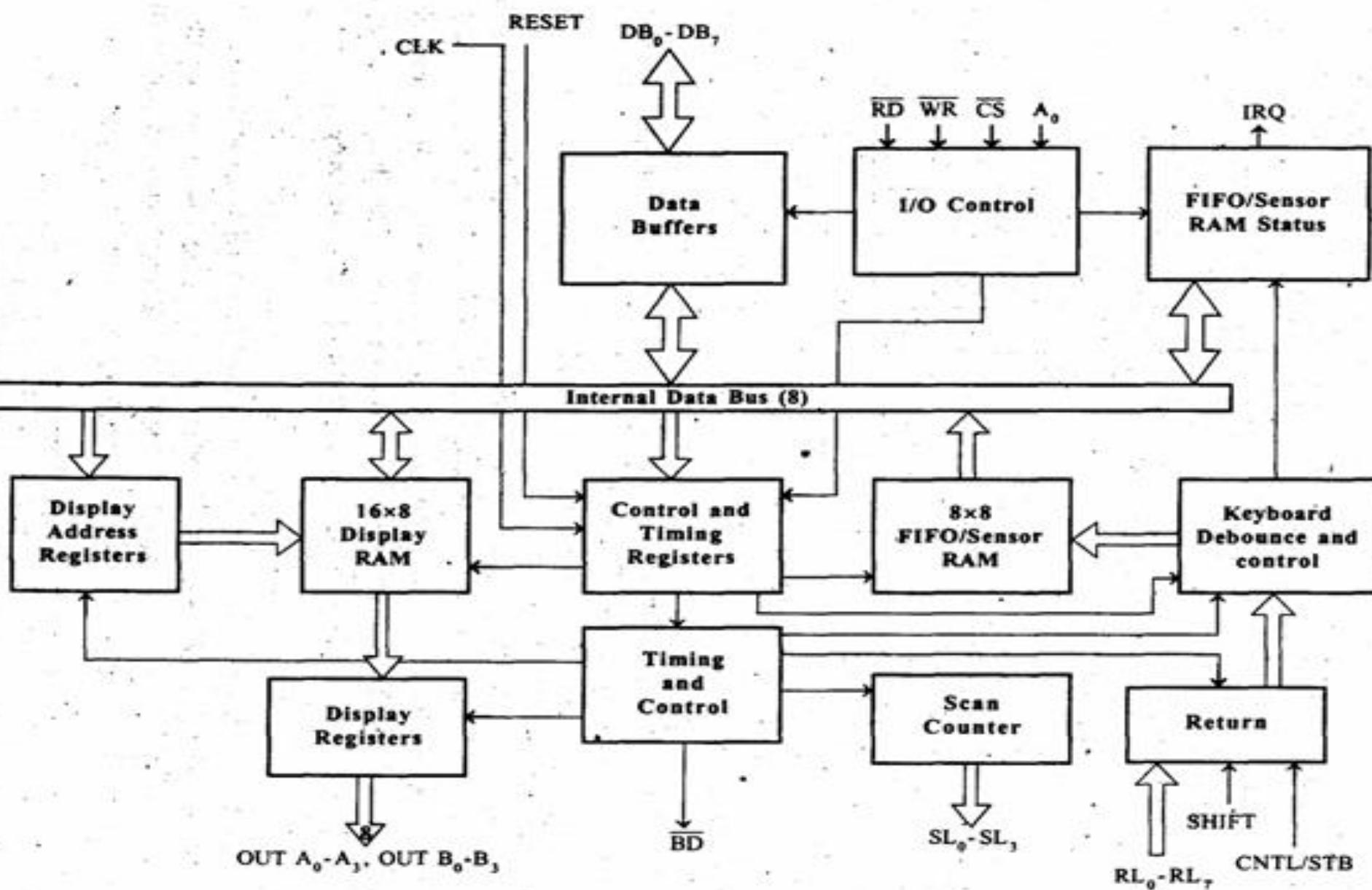


Pin details

- **IRQ:** Interrupt request, becomes 1 when a key is pressed, data is available.
- **OUT A3-A0/B3-B0:** Outputs that sends data to the most significant/least significant nibble of display.
 - RD(WR) Connects to micro's IORC or RD signal, reads data/status registers.
 - **RESET:** Connects to system RESET.
 - **RL7-RL0:** Return lines are inputs used to sense key depression in the keyboard matrix.
 - **Shift:** Shift connects to Shift key on keyboard.
 - **SL3-SL0:** Scan line outputs scan both the keyboard and displays.



Block diagram of 8279



Sections

- Keyboard
- Display
- Scan
- CPU interface

Keyboard section

- The keyboard section consists of eight return lines RL0 - RL7 that can be used to form the columns of a keyboard matrix.
- It has two additional input : shift and control/strobe. The keys are automatically debounced.
- The two operating modes of keyboard section are 2-key lockout and N-key rollover.
- In the 2-key lockout mode, if two keys are pressed simultaneously, only the first key is recognized.
- In the N-key rollover mode simultaneous keys are recognized and their codes are stored in FIFO.
- The keyboard section also have an 8 x 8 FIFO (First In First Out) RAM.
- The FIFO can store eight key codes in the scan keyboard mode. The status of the shift key and control key are also stored along with key code.
- The 8279 generate an interrupt signal when there is an entry in FIFO.

Display section

- The display section has eight output lines divided into two groups A0-A3 and B0-B3.
- The output lines can be used either as a single group of eight lines or as two groups of four lines, in conjunction with the scan lines for a multiplexed display.
- The output lines are connected to the anodes through driver transistor in case of common cathode 7-segment LEDs.
- The cathodes are connected to scan lines through driver transistors.
- The display can be blanked by BD (low) line.
- The display section consists of 16 x 8 display RAM. The CPU can read from or write into any location of the display RAM.

Scan section

- The scan section has a scan counter and four scan lines, SL0 to SL3.
- In decoded scan mode, the output of scan lines will be similar to a 2-to-4 decoder.
- In encoded scan mode, the output of scan lines will be binary count, and so an external decoder should be used to convert the binary count to decoded output.
- The scan lines are common for keyboard and display.
- The scan lines are used to form the rows of a matrix keyboard and also connected to digit drivers of a multiplexed display, to turn ON/OFF.

CPU interface section

- The CPU interface section takes care of data transfer between 8279 and the processor.
- This section has eight bidirectional data lines DB0 to DB7 for data transfer between 8279 and CPU.
- It requires two internal address A =0 for selecting data buffer and A = 1 for selecting control register of 8279.
- The control signals WR (low), RD (low), CS (low) and A0 are used for read/write to 8279.
- It has an interrupt request line IRQ, for interrupt driven data transfer with processor.
- The 8279 require an internal clock frequency of 100 kHz. This can be obtained by dividing the input clock by an internal prescaler.
- The RESET signal sets the 8279 in 16-character display with two -key lockout keyboard modes.

Control Word Description:

First three bits given below select one of 8 control registers (opcode).

➤ 000DD MMMM

Mode set: Opcode 000.

DD sets displays mode.

MMM sets keyboard mode.

DD field selects either:

- 8- or 16-digit display
- Whether new data are entered to the rightmost or leftmost display position.

<i>DD</i>	<i>Function</i>
00	8-digit display with left entry
01	16-digit display with left entry
10	8-digit display with right entry
11	16-digit display with right entry

Control Word Description:

MMM field:

MMM

<i>DD</i>	<i>Function</i>
000	Encoded keyboard with 2-key lockout
001	Decoded keyboard with 2-key lockout
010	Encoded keyboard with N-key rollover
011	Decoded keyboard with N-key rollover
100	Encoded sensor matrix
101	Decoded sensor matrix
110	Strobed keyboard, encoded display scan
111	Strobed keyboard, decoded display scan

- **Encoded Mode:** SL outputs are active-high, follow binary bit pattern 0-7 or 0-15 depending on 8 or 16 digit display.
- **Decoded Mode:** SL outputs are active-low (only one of the four outputs will be low at any time). Pattern output: 1110, 1101, 1011, 0111.

I/O Interface

Control Word Description:

- **Strobe** : An active high pulse on the CN/ST input pin strobes data from the RL pins into an internal FIFO for reading by micro later.
- **2-key lockout/N-key rollover**: Prevents 2 keys from being recognized if pressed simultaneously/Accepts all keys pressed from 1st to last.

Write display format

➤ 100ZAAAA

write display Selects address – to write address of one of the Display. Z selects auto-increment so subsequent writes go to subsequent display positions.

Clear Display format

- 1100CCFA
- The clear control word clears the display, FIFO or both
- Bit F clears FIFO and the display RAM status, and sets address pointer to 000.
- If CC are 00 or 01, all display RAM locations become 00000000.
- If CC is 10, --> 00100000,
- if CC is 11, --> 11111111.

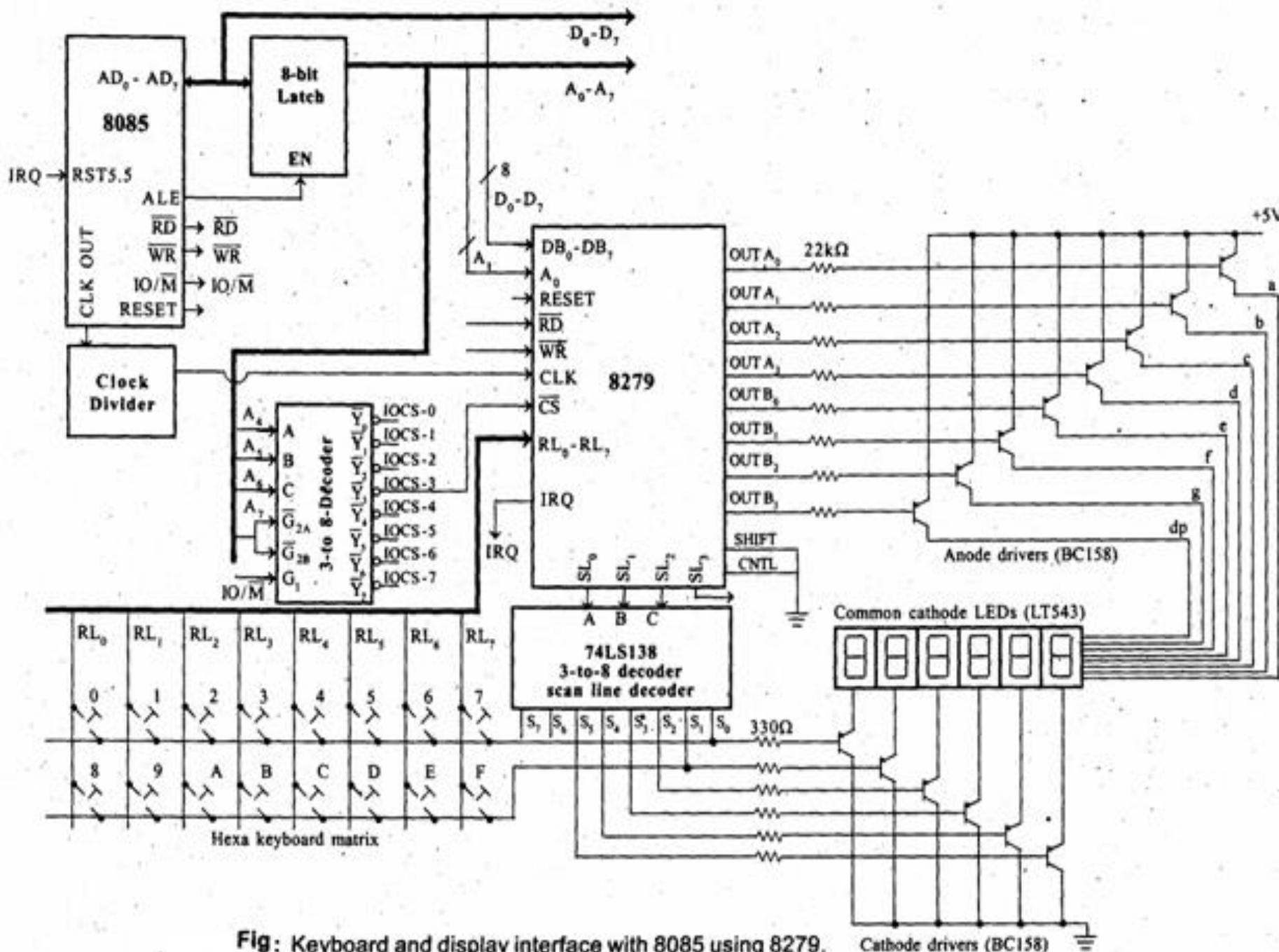


Fig: Keyboard and display interface with 8085 using 8279.

Cathode drivers (BC158)

Basic DMA concept

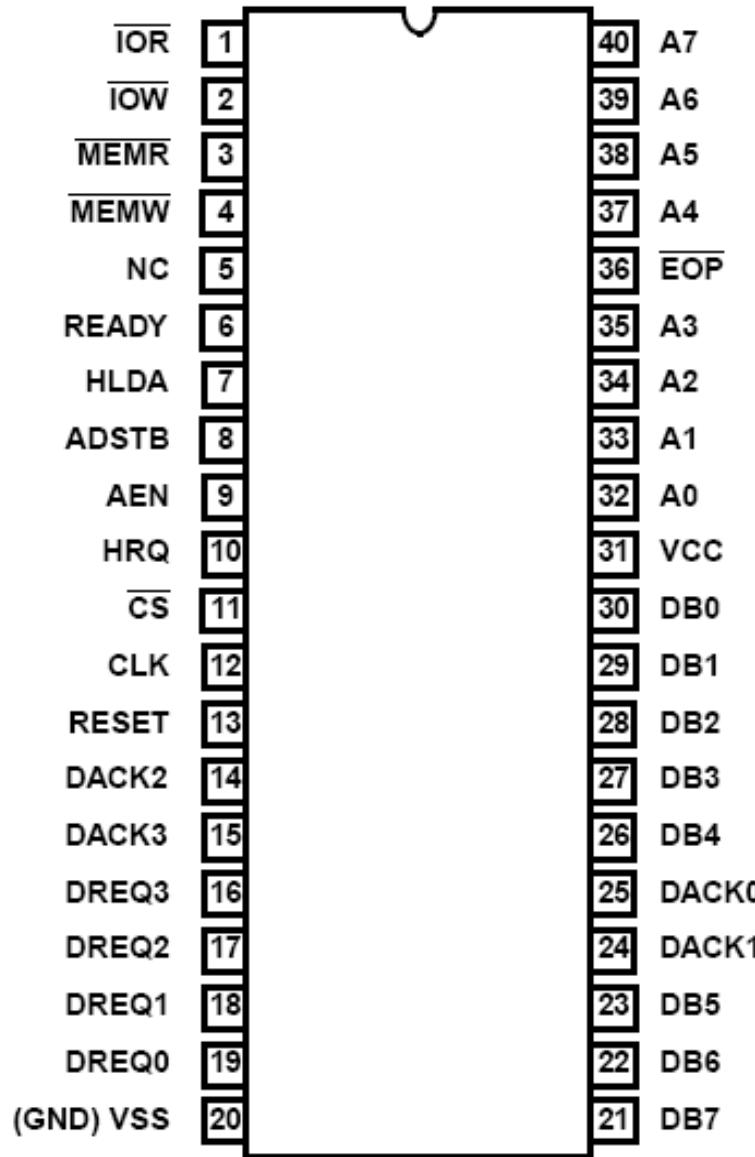
- **Direct memory access (DMA)** is a feature of modern computer systems that allows certain hardware subsystems to read/write data to/from memory without microprocessor intervention, allowing the processor to do other work.
- Used in disk controllers, video/sound cards etc, or between memory locations.
- Typically, the CPU initiates DMA transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller once the operation is complete.
- Can create cache coherency problems (the data in the cache may be different from the data in the external memory after DMA)

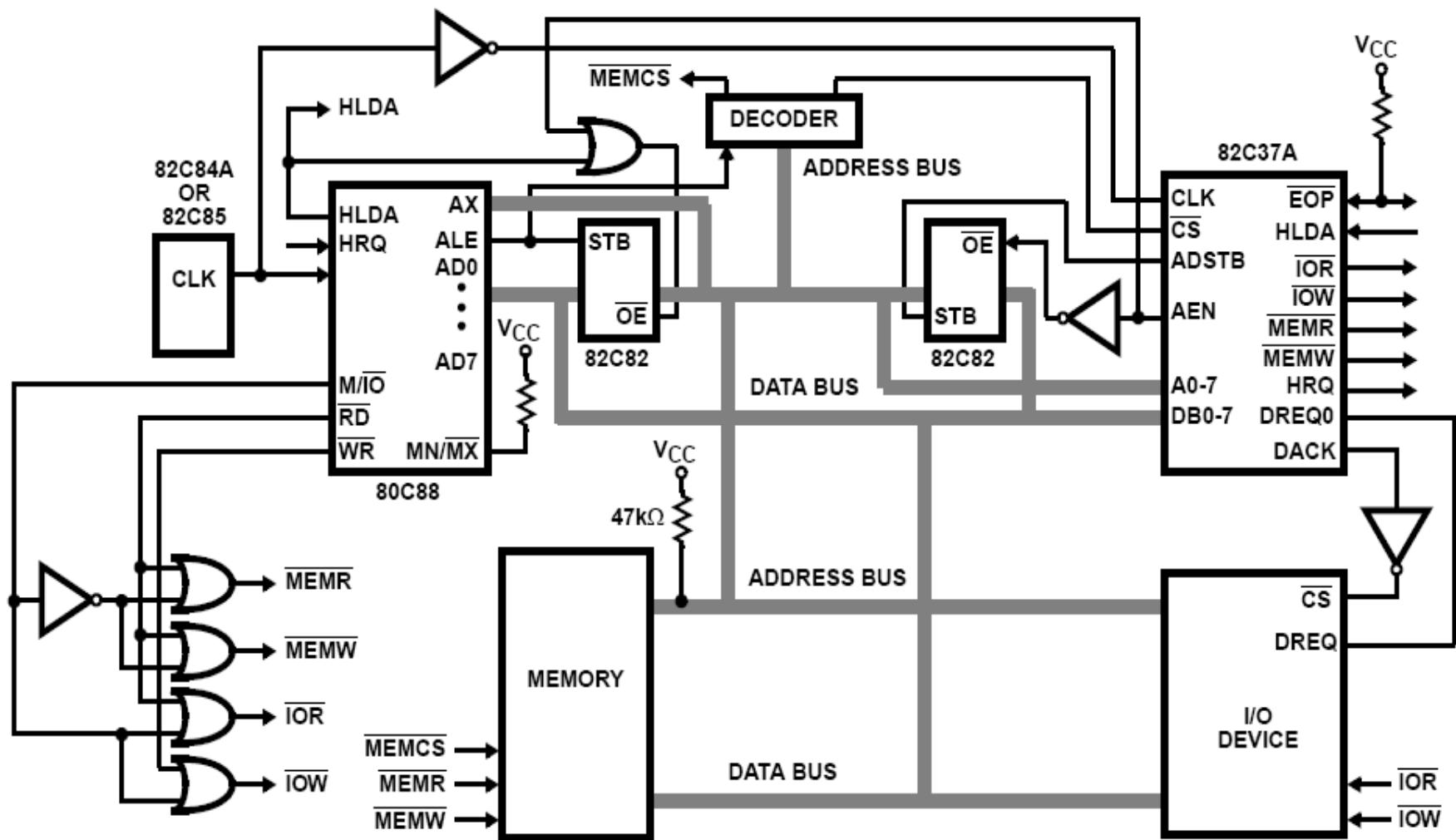
The 8237 DMA controller

- Supplies memory and I/O with control signals and addresses during DMA transfer
- 4-channels (expandable)
 - 0: DRAM refresh
 - 1: Free
 - 2: Floppy disk controller
 - 3: Free
- 1.6MByte/sec transfer rate
- 64 KByte section of memory address capability with single programming
- “fly-by” controller (data does not pass through the DMA-only memory to I/O transfer capability)
- Initialization involves writing into each channel:
 - i) The address of the first byte of the block of data that must be transferred (called the base address).
 - ii) The number of bytes to be transferred (called the word count).

8237 pins

- CLK: System clock
- CS': Chip select (decoder output)
- RESET: Clears registers, sets mask register
- READY: 0 for inserting wait states
- HLDA: Signals that the µp has relinquished buses
- DREQ3 – DREQ0: DMA request input for each channel
- DB7-DB0: Data bus pins
- IOR': Bidirectional pin used during programming
and during a DMA write cycle
- IOW': Bidirectional pin used during programming
and during a DMA read cycle
- EOP': End of process is a bidirectional signal used as input to terminate
a DMA process or as output to signal the end of the DMA transfer
- A3-A0: Address pins for selecting internal registers
- A7-A4: Outputs that provide part of the DMA transfer address
- HRQ: DMA request output
- DACK3-DACK0: DMA acknowledge for each channel.
- AEN: Address enable signal
- ADSTB: Address strobe
- MEMR': Memory read output used in DMA read cycle
- MEMW': Memory write output used in DMA write cycle



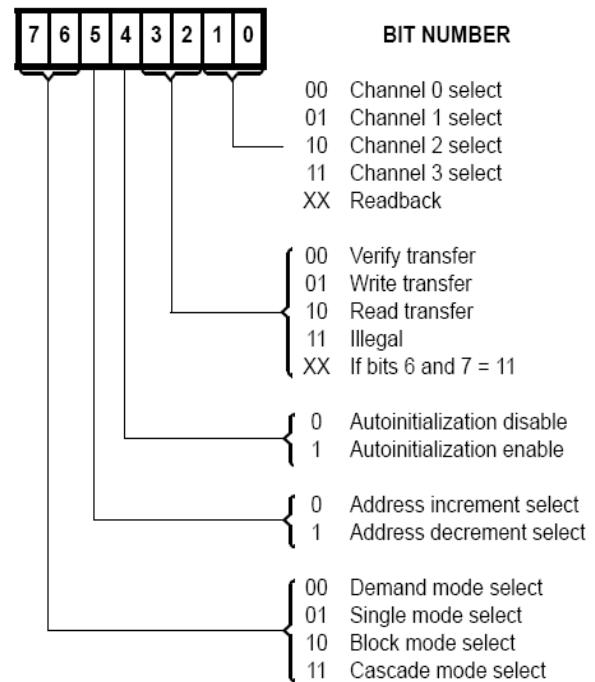
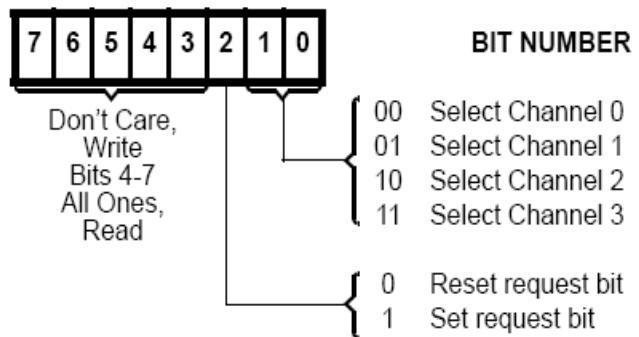


DMA ARCHITECTURE

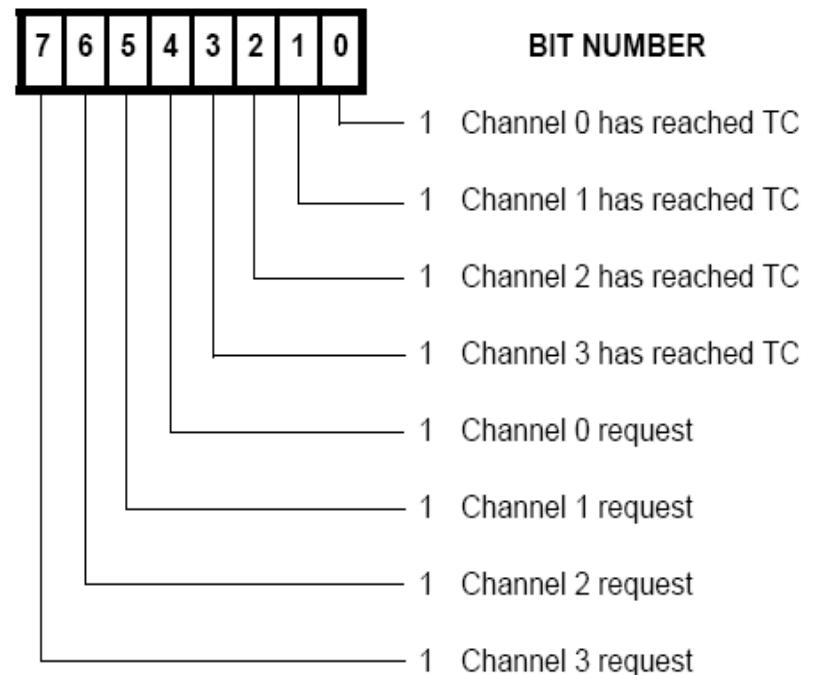
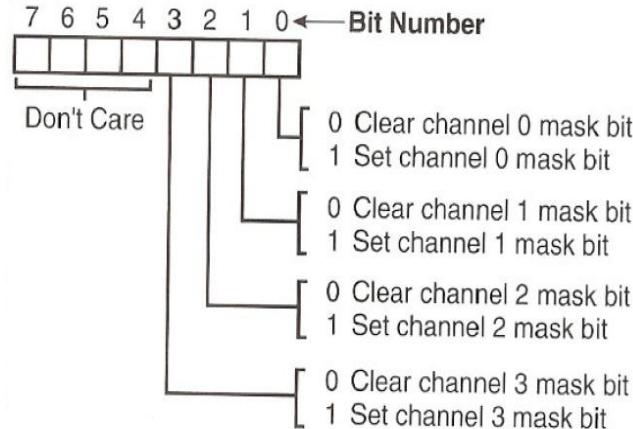
8237 registers

- CAR (Current Address Register): holds the 16-bit memory address used for the DMA transfer (one for each channel), either incremented or decremented during the operation
- CWCR (Current Word Count Register): Programs a channel for the number of bytes (up to 64K) transferred during a DMA operation
- BA (Base Address) and WC (Word Count): Used when auto-initialization is selected for a channel, to reload the CAR and CWCR when DMA is complete.
- CR (Command Register): Programs the operation of the controller

- MR (Mode Register):
- Programs the mode of operation for a channel (or for each channel).



- MR (Mask Register):



- SR (Status Register): Shows the status of each DMA channel

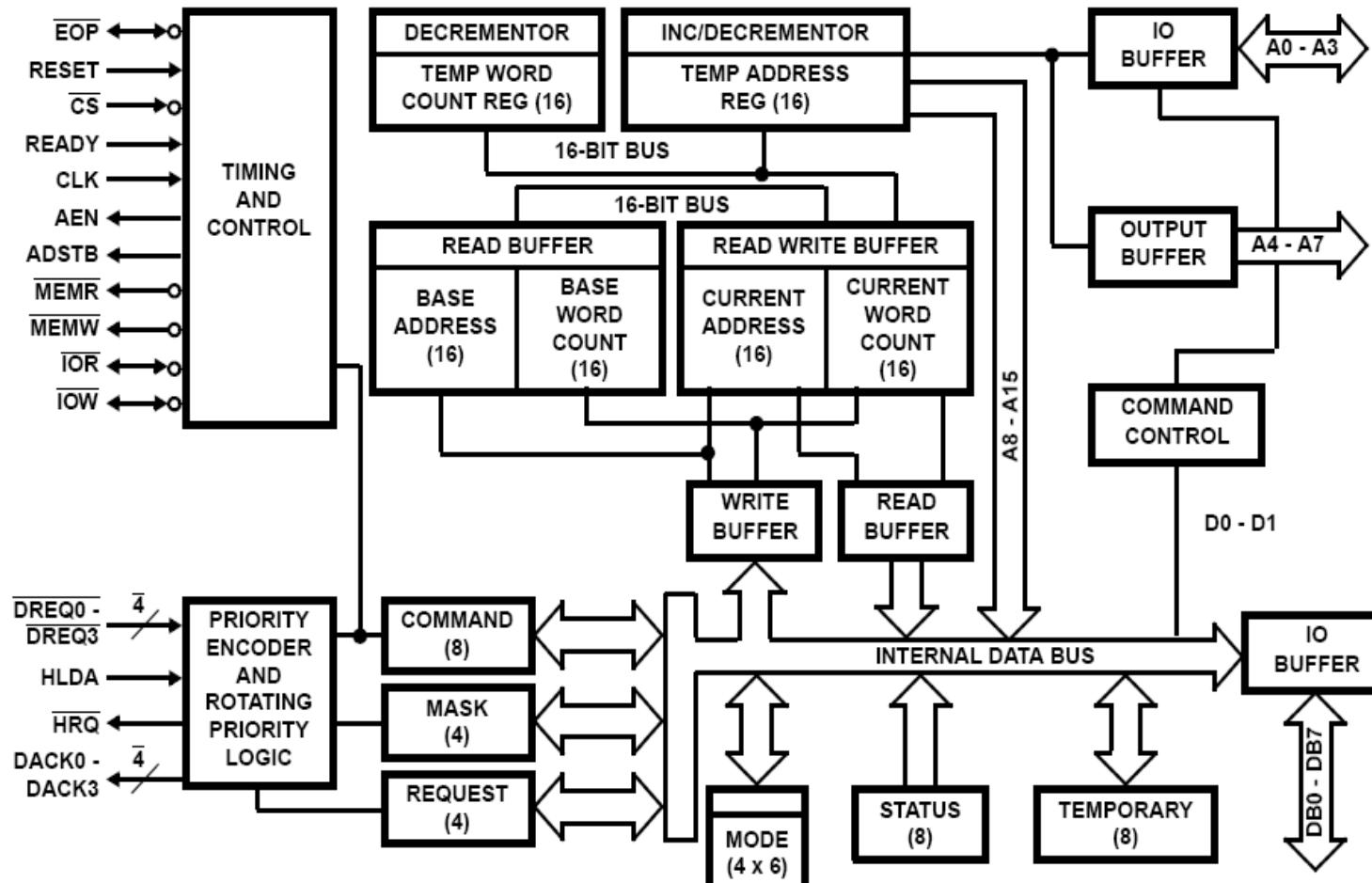
8237 Software commands

OPERATION	A3	A2	A1	A0	\overline{IOR}	\overline{IOW}
Read Status Register	1	0	0	0	0	1
Write Command Register	1	0	0	0	1	0
Read Request Register	1	0	0	1	0	1
Write Request Register	1	0	0	1	1	0
Read Command Register	1	0	1	0	0	1
Write Single Mask Bit	1	0	1	0	1	0
Read Mode Register	1	0	1	1	0	1
Write Mode Register	1	0	1	1	1	0
Set First/Last F/F	1	1	0	0	0	1
Clear First/Last F/F	1	1	0	0	1	0
Read Temporary Register	1	1	0	1	0	1
Master Clear	1	1	0	1	1	0
Clear Mode Reg. Counter	1	1	1	0	0	1
Clear Mask Register	1	1	1	0	1	0
Read All Mask Bits	1	1	1	1	0	1
Write All Mask Bits	1	1	1	1	1	0

8237 Software commands

- **Clear First/Last Flip-Flop** - This command is executed prior to writing or reading new address or word count information to the 82C37. This command initializes the flipflop to a known state (low byte first) so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.
- **Set First/Last Flip-Flop** - This command will set the flip-flop to select the high byte first on read and write operations to address and word count registers.
- **Master Clear** - This software instruction has the same effect as the hardware Reset. The Command, Status, Request, and Temporary registers, and Internal First/Last Flip-Flop and mode register counter are cleared and the Mask register is set. The 82C37A will enter the idle cycle.
- **Clear Mask Register** - This command clears the mask bits of all four channels, enabling them to accept DMA requests.
- **Clear Mode Register Counter** - Since only one address location is available for reading the Mode registers, an internal two-bit counter has been included to select Mode registers during read operation. To read the Mode registers, first execute the Clear Mode Register Counter command, then do consecutive reads until the desired channel is read. Read order is channel 0 first, channel 3 last. The lower two bits on all Mode registers will read as ones.

8237 block diagram



Initiating a DMA transaction

- Save the current interrupt status and disable interrupts by executing the CLI instruction
- Disable the channel that will be used for the transaction
- Reset the flip-flop by writing a value of 0X to the register
- Set the Mode Register
- Set the Page Register
- Set the Offset Register
- Set the Block Size Register
- Enable the channel that will be used for the transaction
- Restore the interrupt status

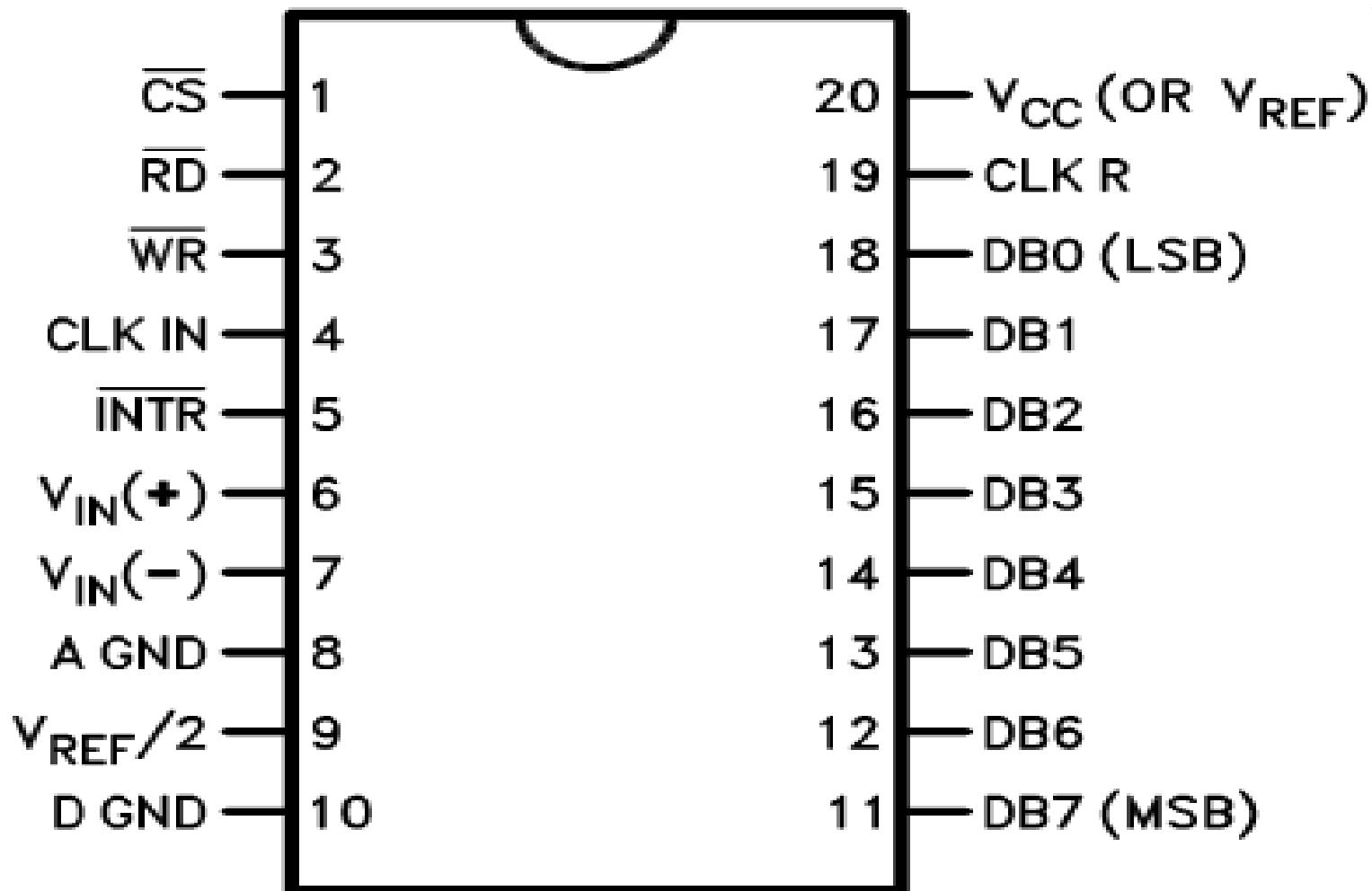
A/D Interfacing

- After the conversion is over, the **ADC sends end of conversion EOC signal to inform the microprocessor** that the conversion is over and the **result is ready at the output buffer of the ADC.**
- These tasks of issuing an **SOC pulse to ADC**, **reading EOC signal from the ADC** and **reading the digital output of the ADC** are carried out by the **CPU using 8255 I/O ports.**

A/D Interfacing

- The time taken by the ADC from the **active edge of SOC pulse till the active edge of EOC signal is called as the conversion delay of the ADC.**
- Successive approximation techniques and dual slope integration techniques are the most popular techniques used in the integrated ADC chip.
- The analog to **digital converter chips 0808 and 0809 are 8-bit CMOS**, successive approximation converters.

ADC 0804



ADC Interfacing

- CS :Active low input used to activate the ADC0804 chip.

RD (data enable) : Active low input used to get converted data out of the ADC0804 chip. When CS = 0, if a high-to-low pulse is applied to the RD pin, the 8-bit digital output shows up at the D0-D7 data pins.

WR (start conversion): Active low input used to inform the ADC0804 to start the conversion process. If CS = 0 when WR makes a low-to-high transition, the ADC0804 starts converting the analog input value of Vin to an 8-bit digital number. When the data conversion is complete, the INTR pin is forced low by the ADC0804.

ADC Interfacing

- CLK IN and CLK R : Connect to external capacitor and resistor for self-clocking, $f = 1/(1.1RC)$. The clock affect the conversion time and this time cannot be faster than 110 micros.

INTR (end of conversion) This is an active low output pin. When the conversion is finished, it goes low to signal the CPU that the converted data is ready to be picked up. After INTR goes low, we make CS = 0 and send a high-to-low pulse to the RD pin to get the data out of the ADC0804 chip.

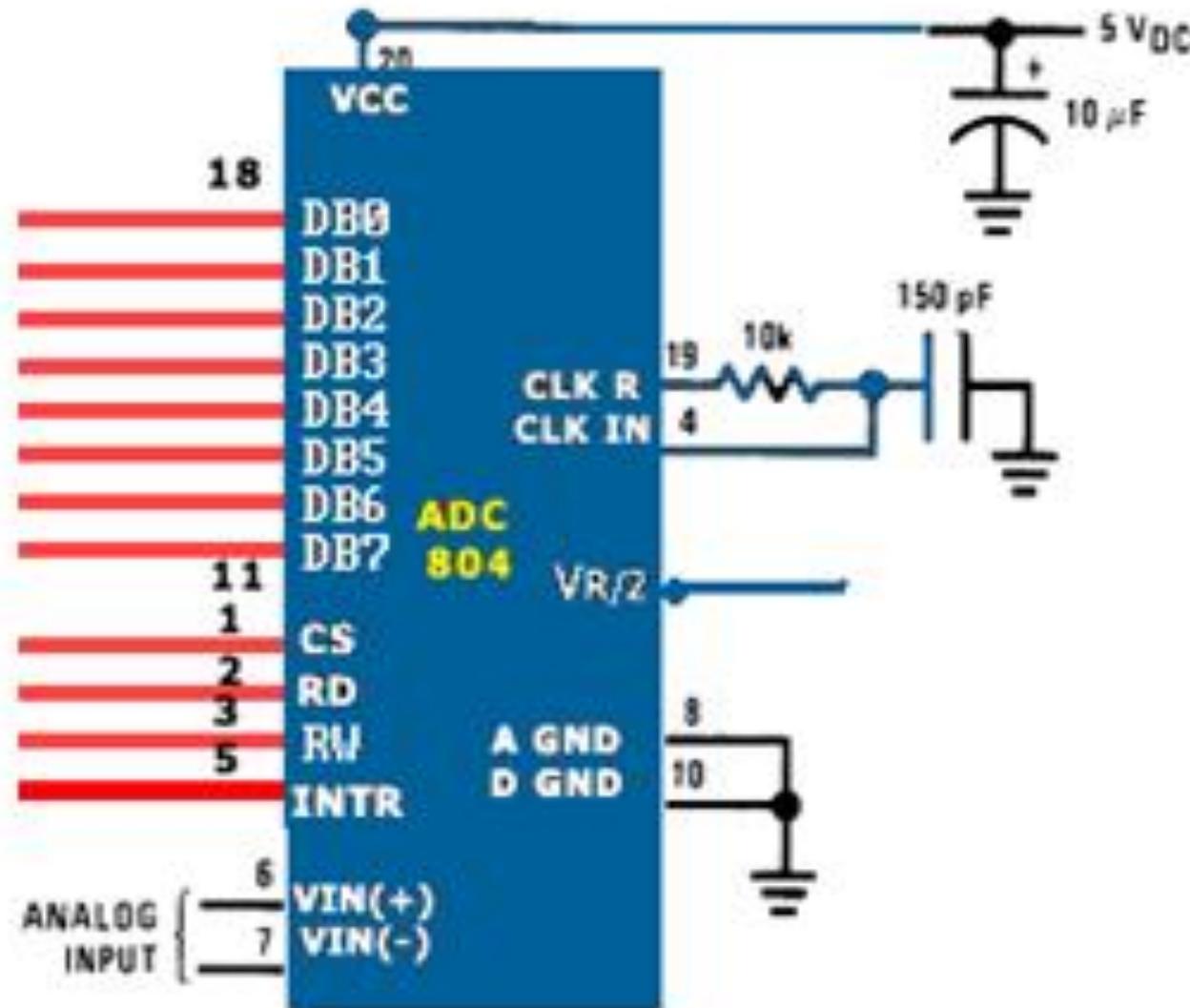
ADC Interfacing

- **Vin (+) and Vin (-)**: These are the differential analog inputs where $V_{in} = V_{in\ (+)} - V_{in\ (-)}$. Often the $V_{in\ (-)}$ pin is **connected to ground** and the $V_{in\ (+)}$ pin is used as the analog input to be converted to digital.
- **VCC** : This is the +5V power supply. It is also used as a reference voltage when the $V_{ref/2}$ (pin 9) input is open.

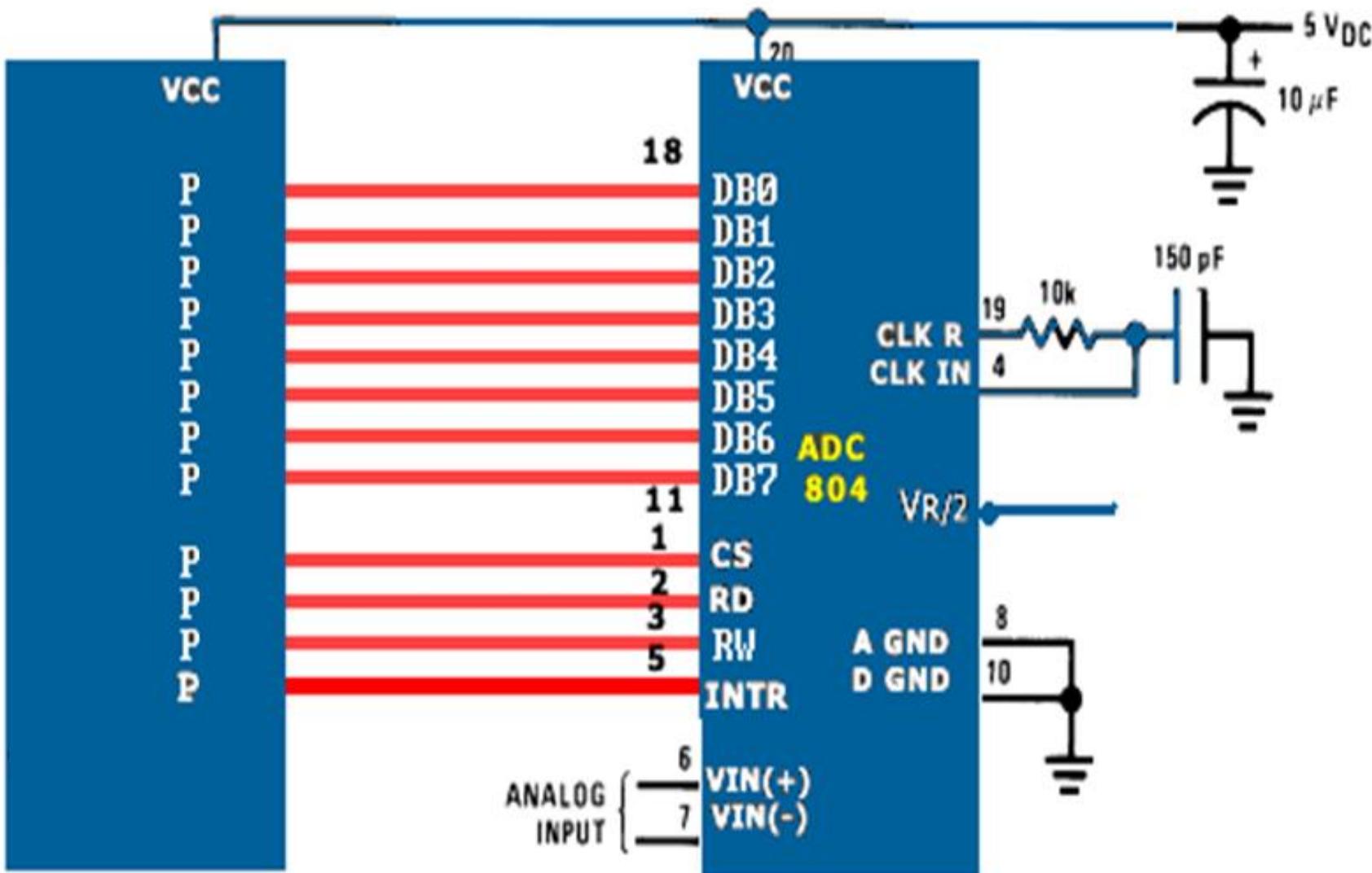
ADC Interfacing

- Vref/2 :- Input voltage pin used for the reference voltage. If this pin is open, the analog input voltage for the the ADC is ranged from 0 to 5 volts. This is optional input pin. It is used only when the input signal range is small. When pin 9 is at 2V, the range is 0-4V, i.e. Twice the voltage at pin 9. Pin 6 (V+), Pin 7(V-): The actual input is the difference in voltages applied to these pins. The analogue input can range from 0 to 5V.

ADC Interfacing

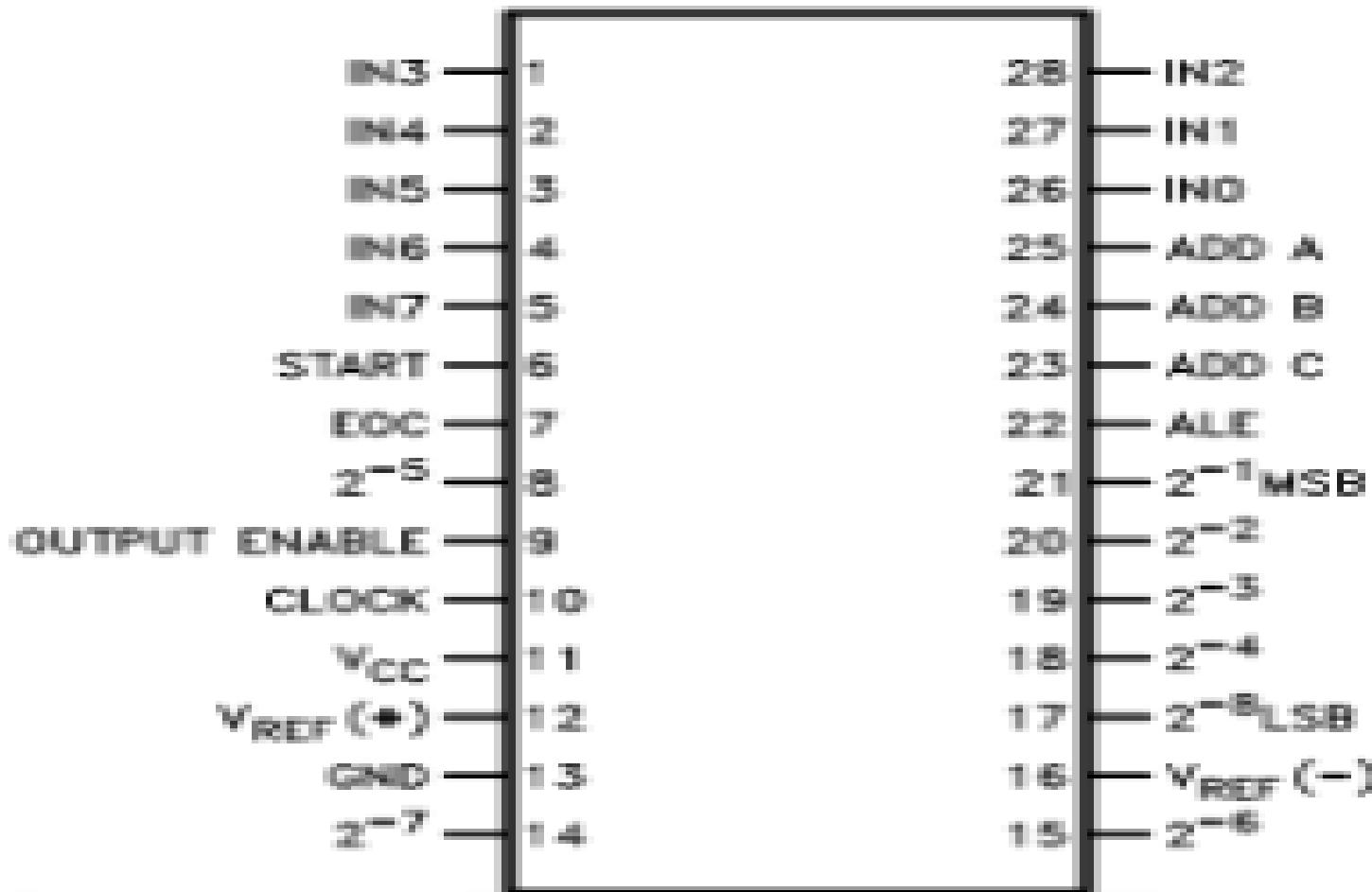


ADC Interfacing



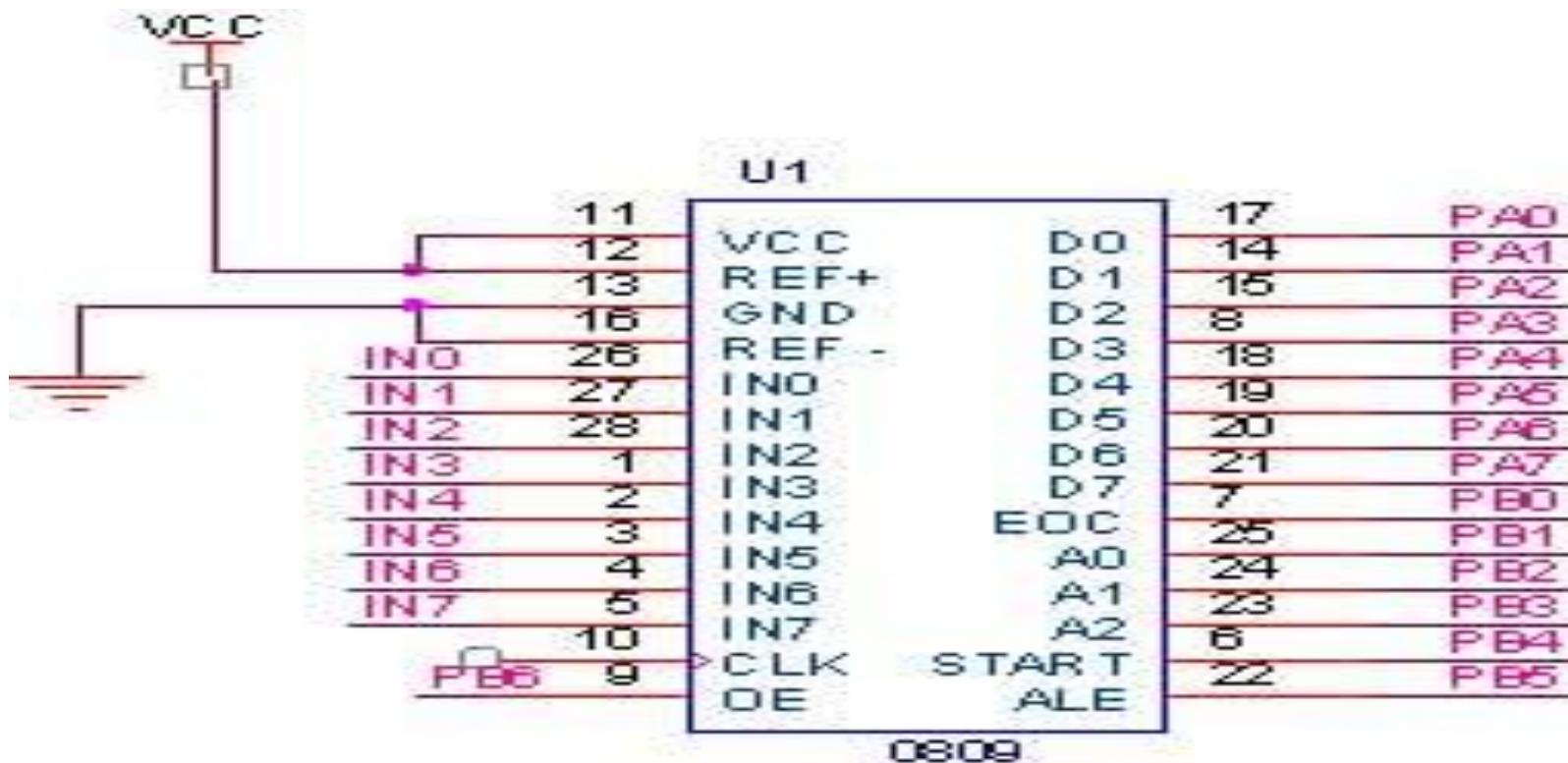
A/D Interfacing [0808]

Dual-In-Line Package



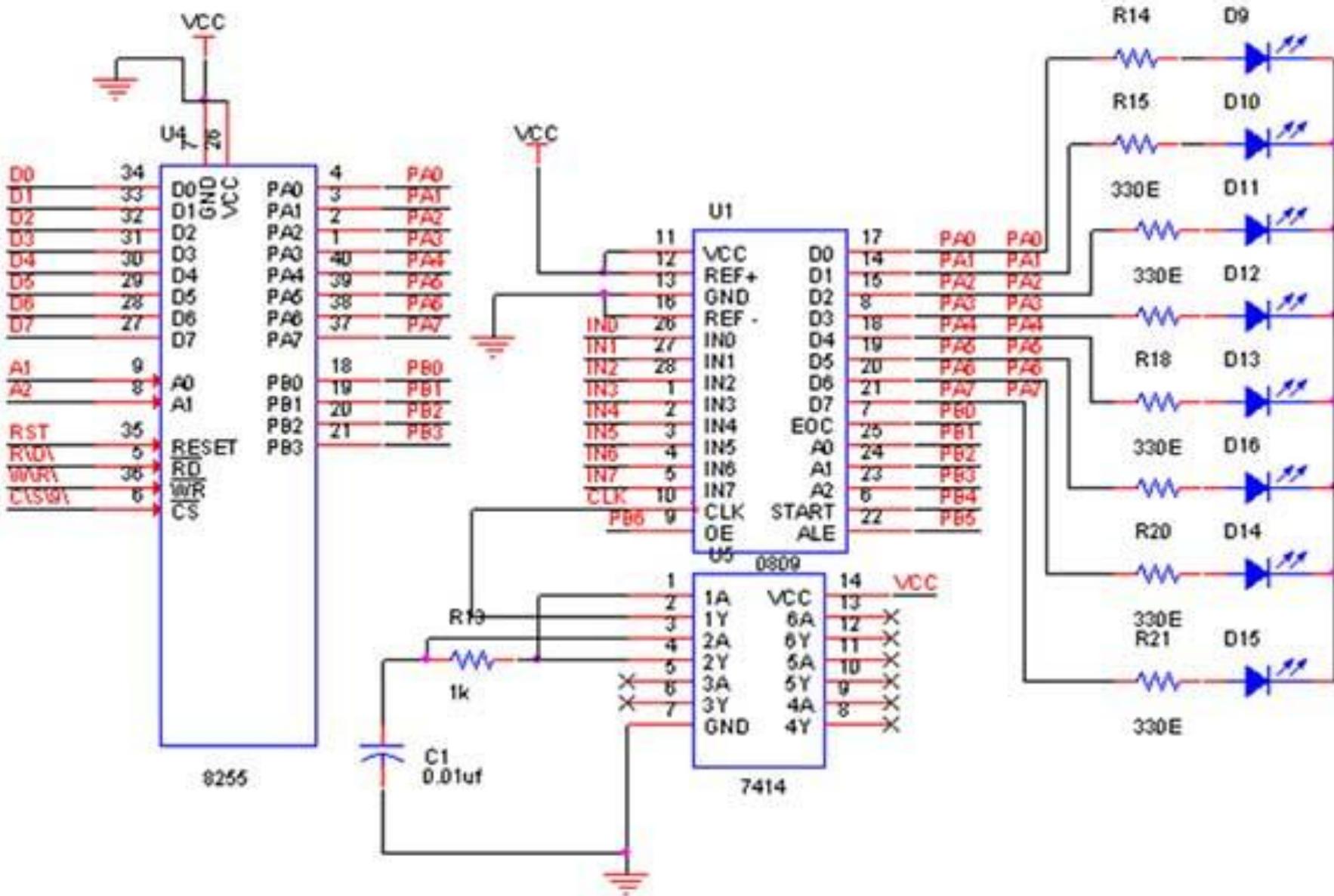
A/D Interfacing

- The ADC 0808 is 8-channel 8-bit ADC chip.
It has 8 analog inputs i.e. IN0-IN7.



PA0 – PA7 is connected to 8 no's of LED

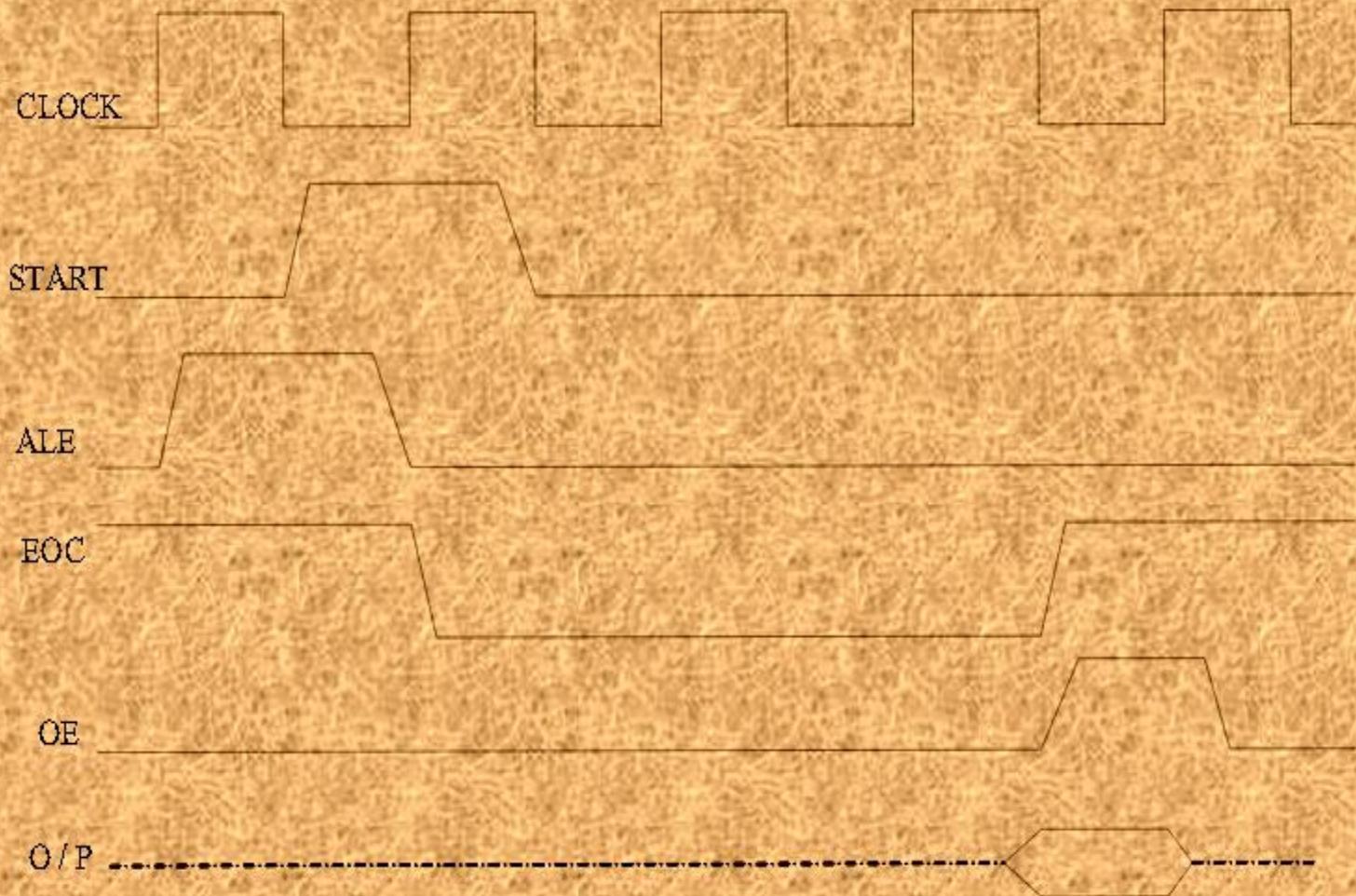
A/D Interfacing



A/D Interfacing

LABEL	OPCODE	OPERAND	COMMENT
	MVI	A,98H	Initialize 8255
	OUT	83H (CWR)	
	MVI	A,0BH	Selects Channel IN3 and gets High to Low SOC and ALE signals
	OUT	82H(PORT C)	
	MVI	A,03H	
	OUT	82H(PORT C)	
L1:	IN	82H(PORT C)	Checks EOC
	RAL		Is conversion complete? if no then jump to L1
	JNC	L1	
	MVI	A,01H	Is conversion complete? if yes then enable Output
	OUT	81H(PORT B)	
	IN	80H(PORT A)	Read data from Port A
	STA	2050H	Store Data
	HLT		Stop

A/D Interfacing



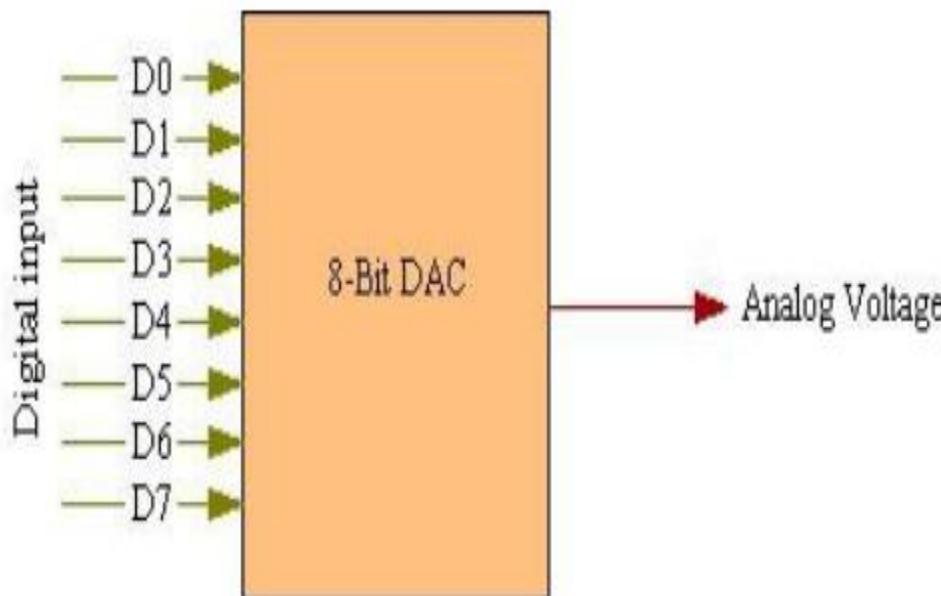
Timing Diagram of ADC 0808

3/9/14

D/A INTERFACING

D/A Interfacing

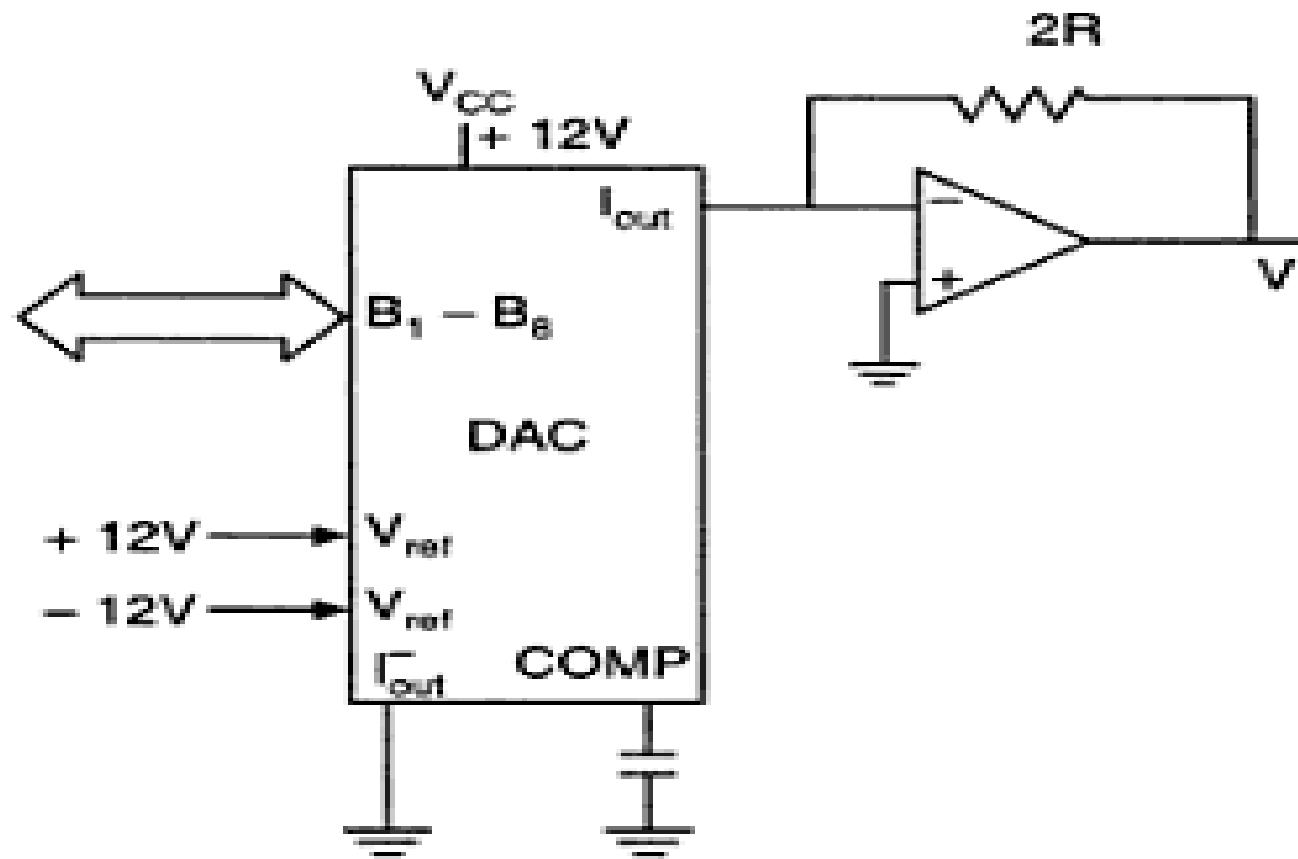
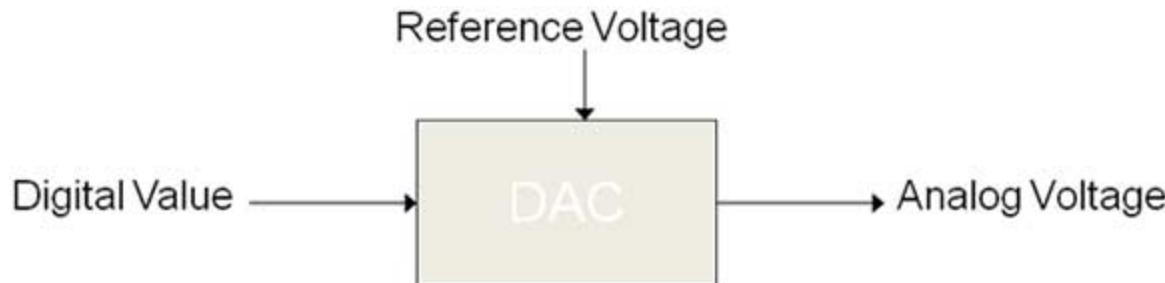
- The digital to analog converters **convert binary number into their equivalent voltages.**
- The DAC find applications in areas like digitally controlled **speed** **controls,** **amplifiers** etc.



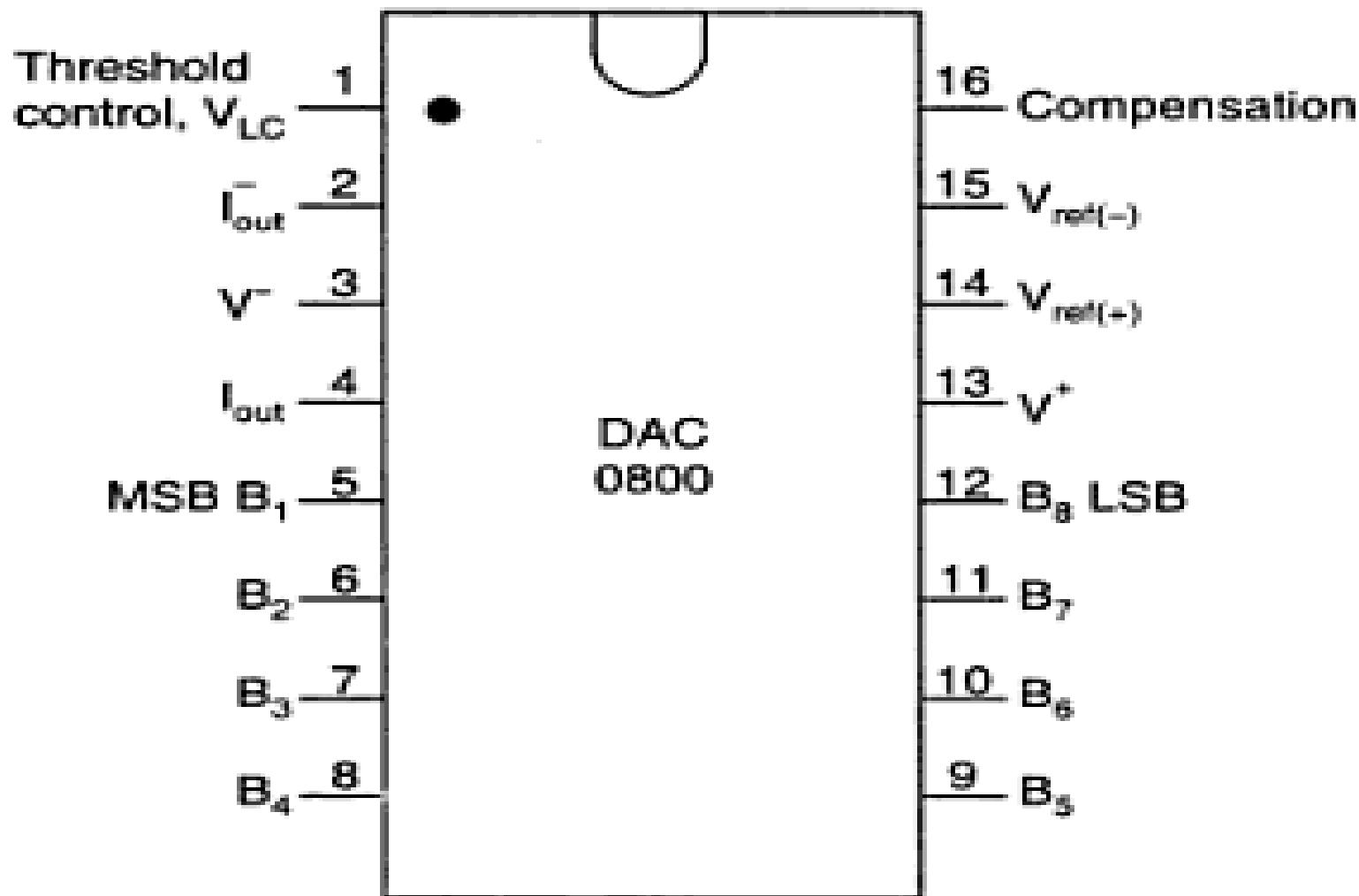
D/A Interfacing

	DAC0800	DAC0802
Resolution (Bits)	8	8
DAC: Channels	1	1
Architecture	Multiplying DAC	Multiplying DAC
Interface	Parallel	Parallel
Output Type	Current	Current
Output Range Min. (V or mA)	0	0
Output Range Max. (V or mA)	2	2
Settling Time (μ s)	0.1	0.1
Reference: Type	Ext	Ext
Rating	Catalog	Catalog
Pin/Package	16PDIP 16SOIC	16SOIC
Approx. Price (US\$)	0.51 1ku	0.69 1ku
Operating Temperature Range (C)	0 to 70	0 to 70

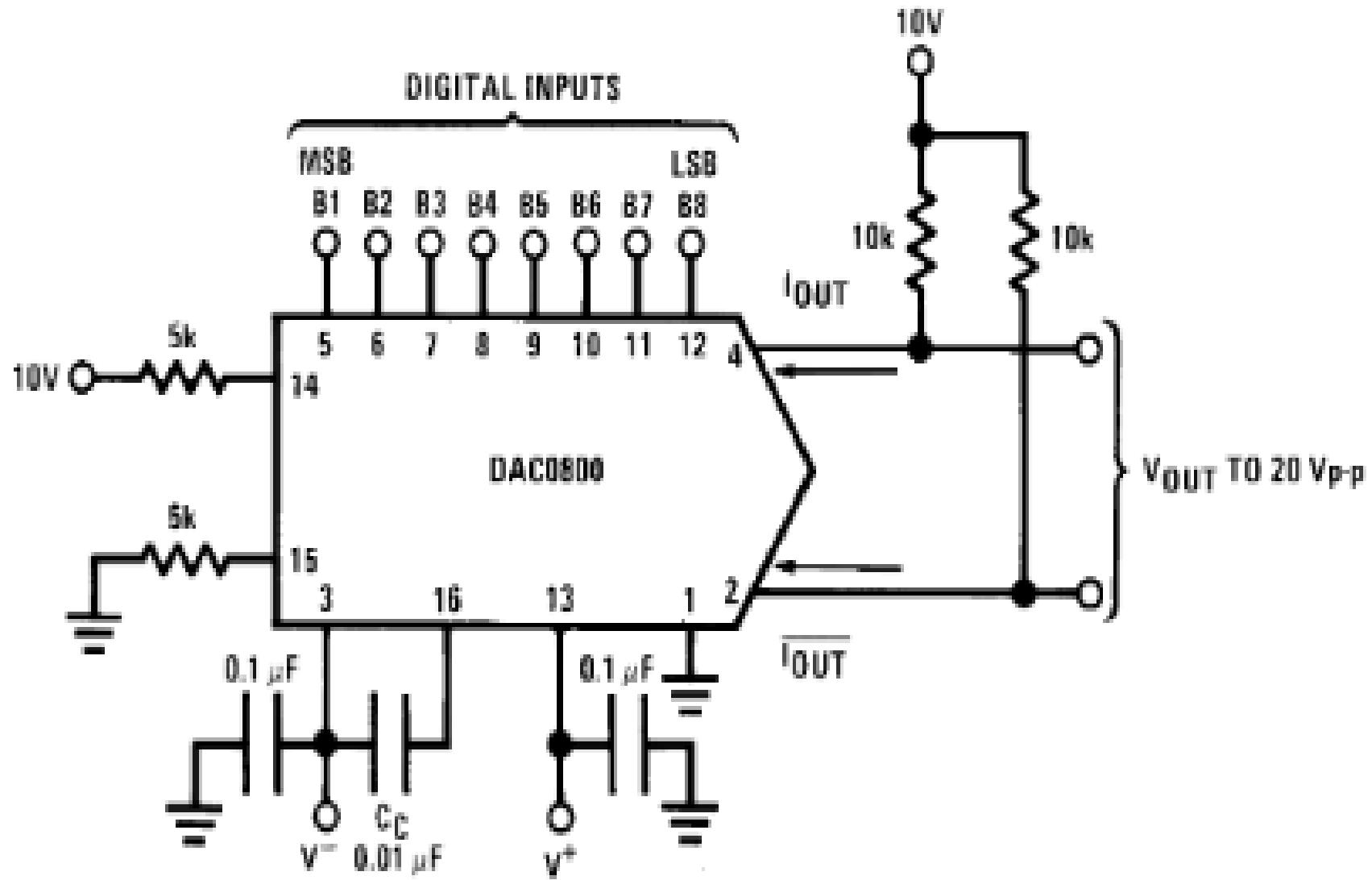
D/A Interfacing



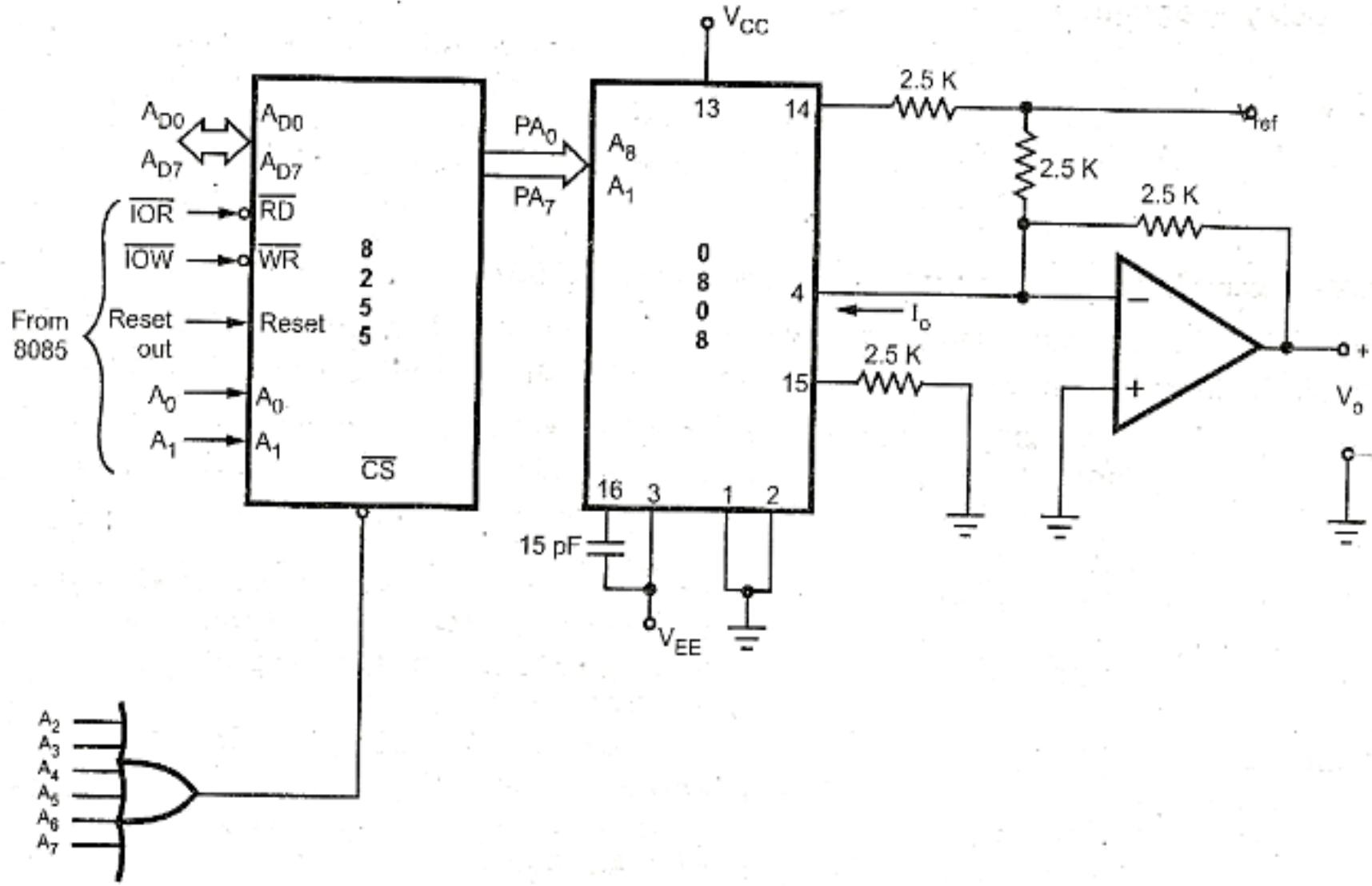
D/A Interfacing



D/A Interfacing



D/A Interfacing



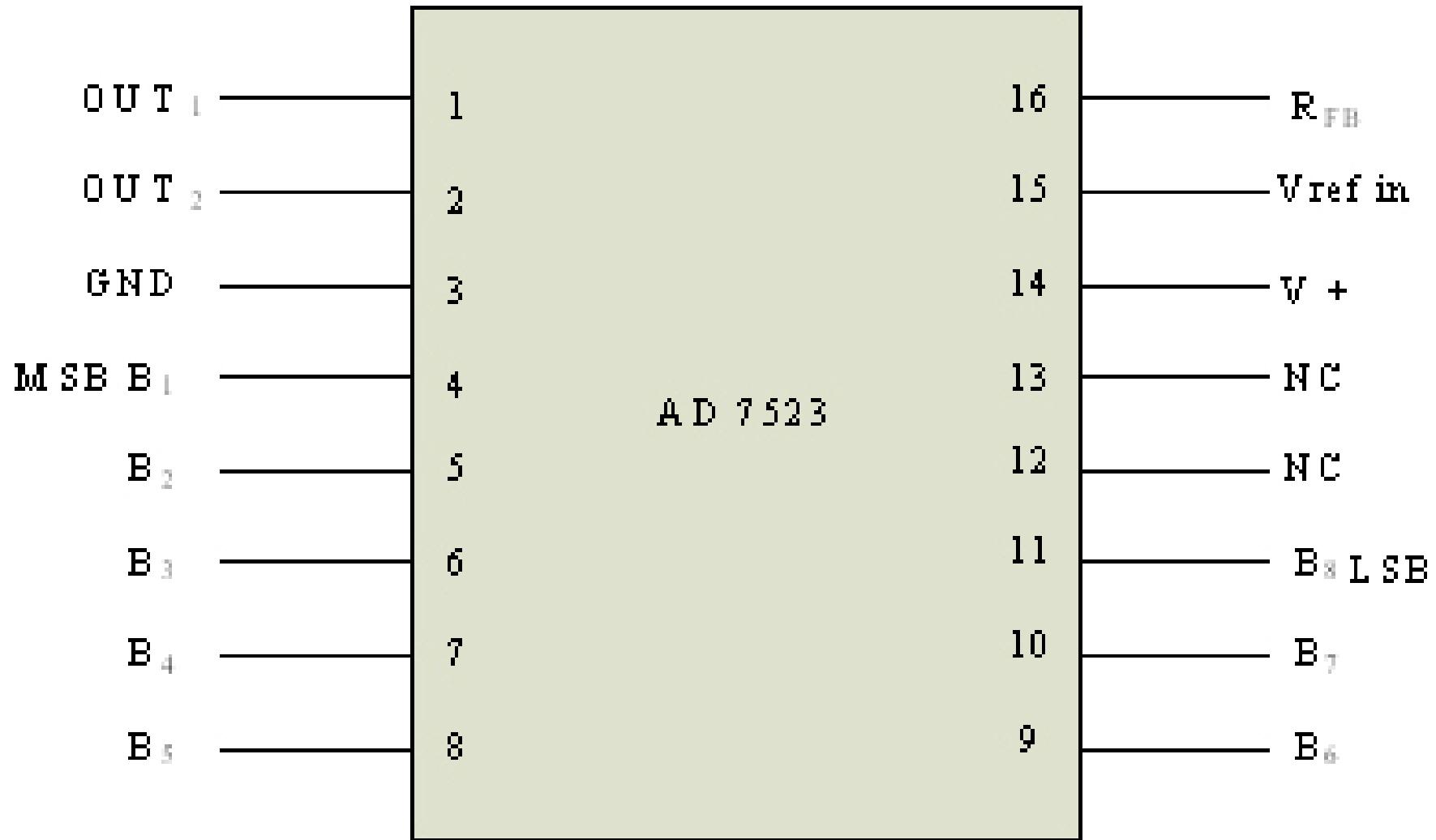
D/A Interfacing

MEMORY ADDRESS	OPCODE	MNEMONICS
9100	3E 80	MVI A,80
9102	D3 23	OUT CNTRL
9104	3E 00	START: MVI A, 00
9106	D3 20	LOOP1: OUT PORTA
9108	3C	INR A
9109	C2 06 91	JNZ LOOP1
910C	C3 04 91	JMP START

D/A Interfacing

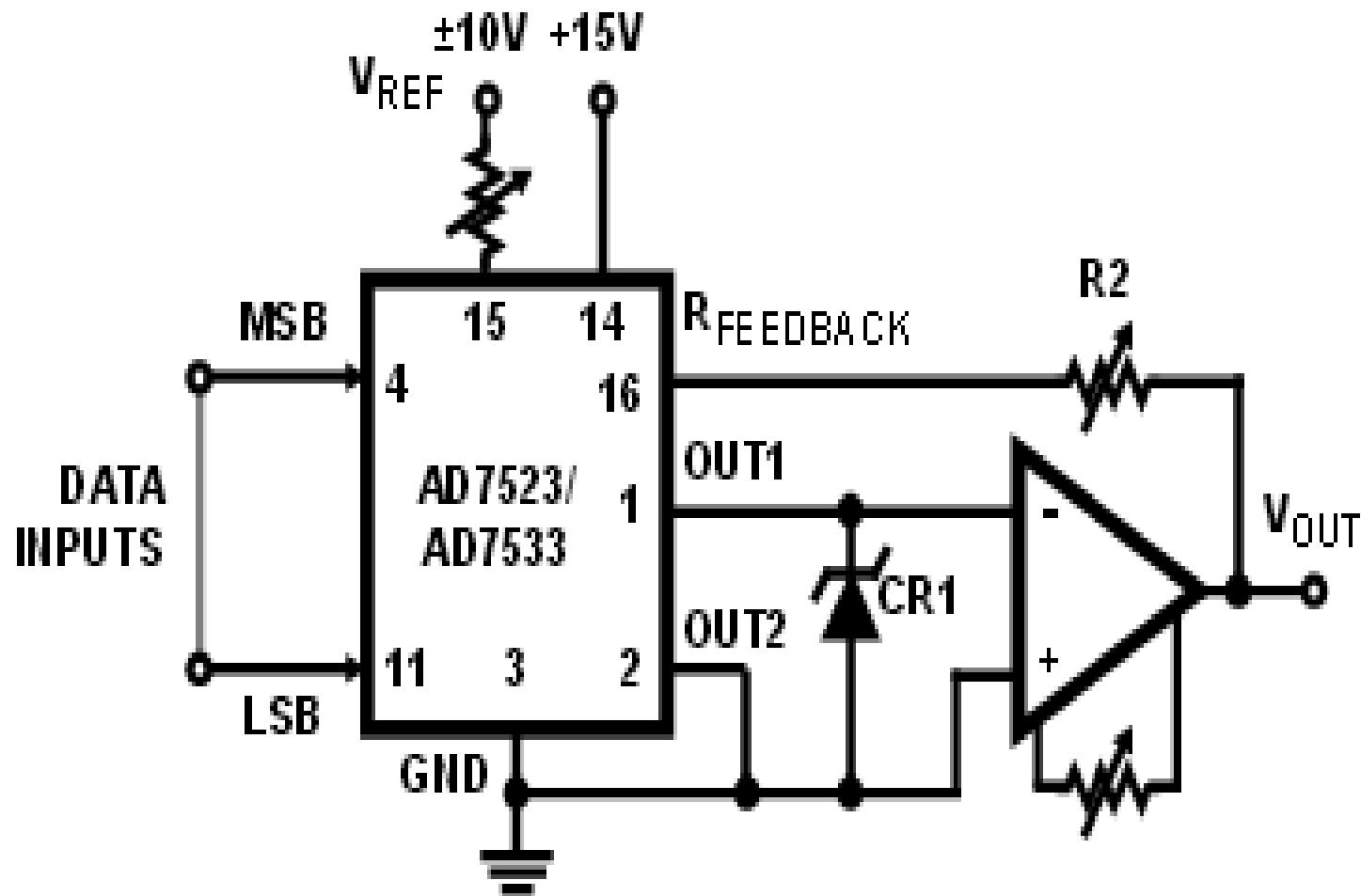
- AD 7523 8-bit Multiplying DAC : This is a 16 pin DIP, multiplying digital to analog converter, containing R-2R ladder for D-A conversion along with single pole double thrown NMOS switches to connect the digital inputs to the ladder.
- supply range is from +5V to +15V, while
- Vref may be any where between -10V to +10V.
- The maximum analog output voltage will be any where between -10V to +10V, when all the digital inputs are at logic high state.

D/A Interfacing

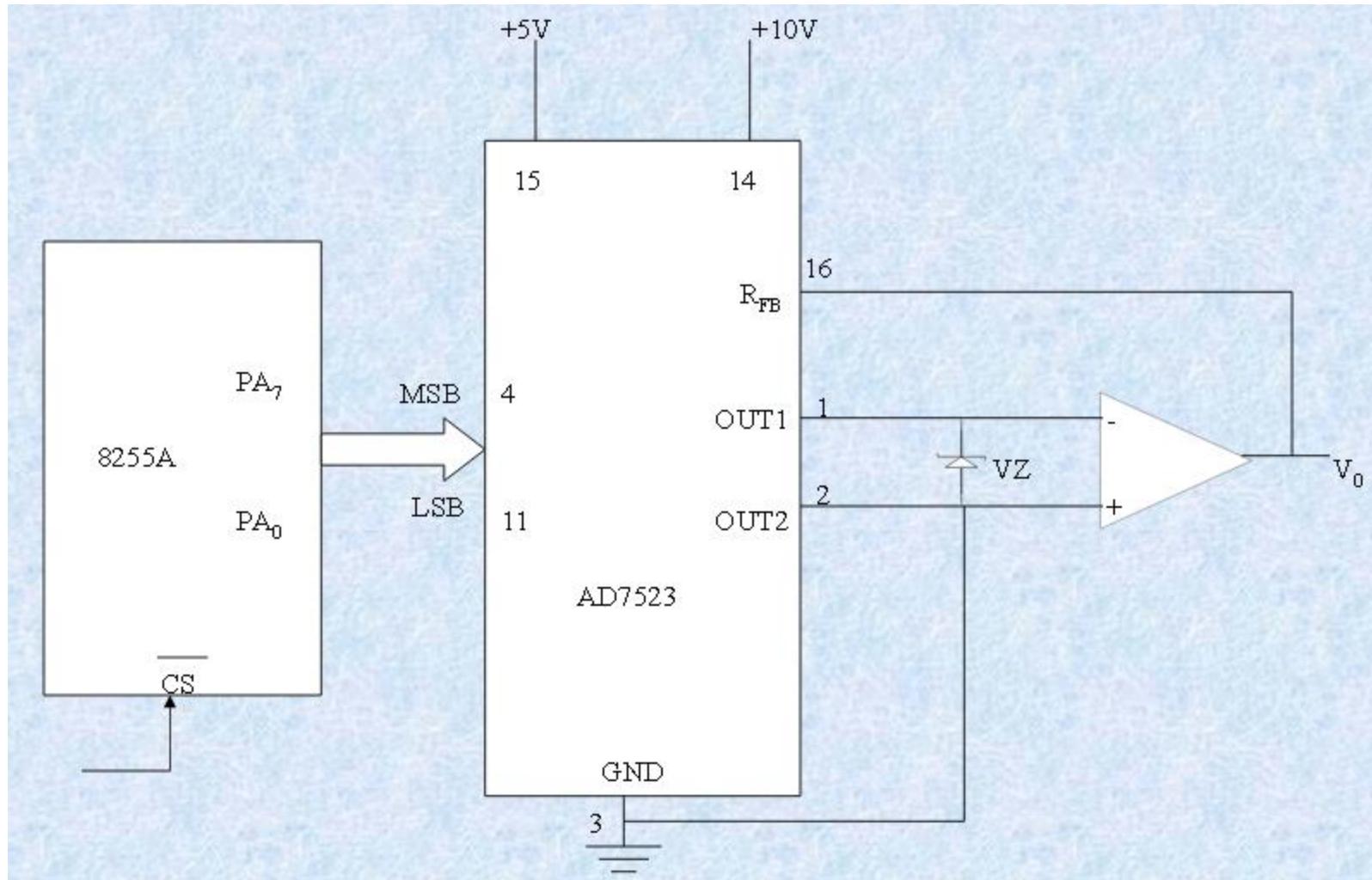


Pin Diagram of AD 7523

D/A Interfacing



D/A Interfacing



D/A Interfacing

UNIPOLAR BINARY CODE - AD7523

DIGITAL INPUT MSB LSB	ANALOG OUTPUT
11111111	$-V_{REF} \left(\frac{255}{256} \right)$
10000001	$-V_{REF} \left(\frac{129}{256} \right)$
10000000	$-V_{REF} \left(\frac{128}{256} \right) = -\frac{V_{REF}}{2}$
01111111	$-V_{REF} \left(\frac{127}{256} \right)$
00000001	$-V_{REF} \left(\frac{1}{256} \right)$
00000000	$-V_{REF} \left(\frac{0}{256} \right) = 0$

Peripheral Interfacing

Data Communications

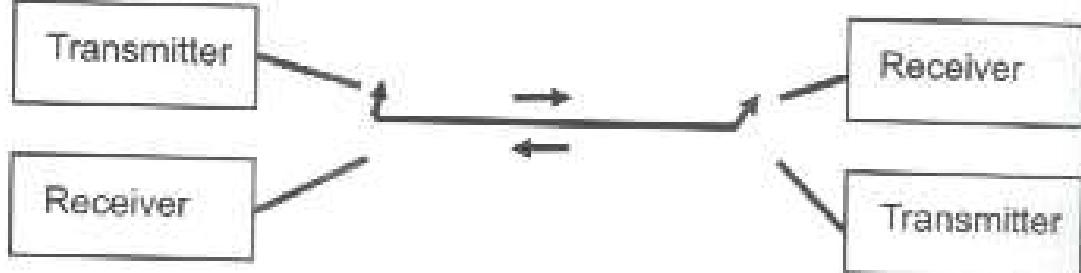
- Data communications refers to the ability of one computer to exchange data with another computer or a peripheral
- Physically, the data comm. path may be a short, 5 to 10 feet ribbon cable connecting a microcomputer and parallel printer; or it might be a high speed telecommunications port connecting two computers thousands of miles apart.
- Standard data communication interfaces and standards are needed
- Centronic's parallel printer interface
- RS-232 defines a serial communications standard
- We focus on serial I/O this week
- **8251 USART (Universal Synchronous/Asynchronous Receiver/Transmitter) is the key component for converting parallel data to serial form and vice versa**
- Two types of serial data communications are widely used
 - Asynchronous communications
 - Synchronous communications

Types of Transmission

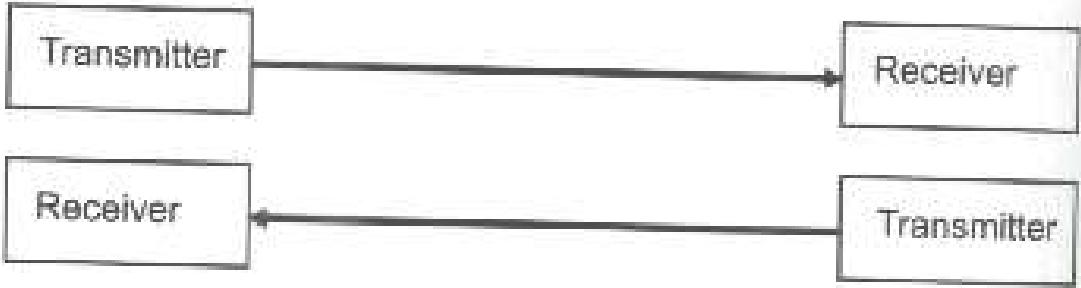
Simplex



Half Duplex

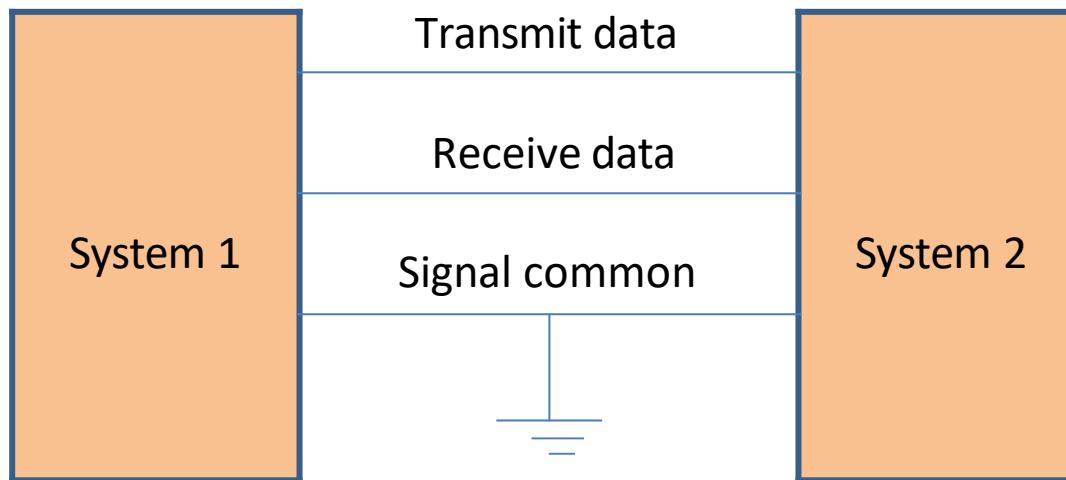


Full Duplex



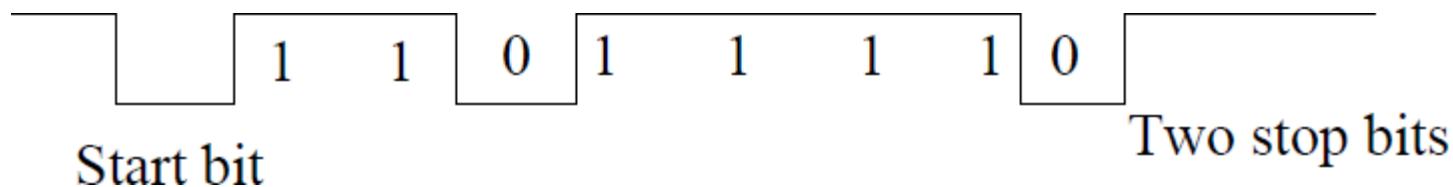
Asynchronous Communications

- Eliminates the need for a clock signal between two microprocessor based systems



Asynchronous Communications

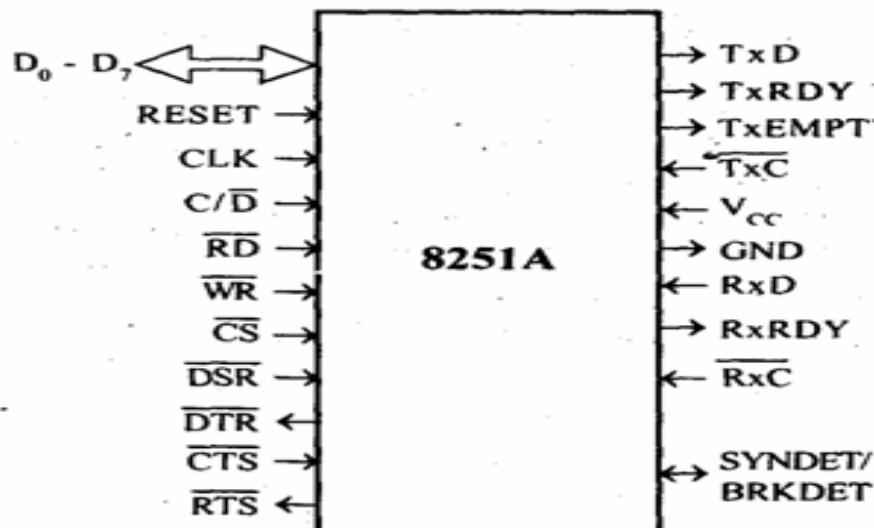
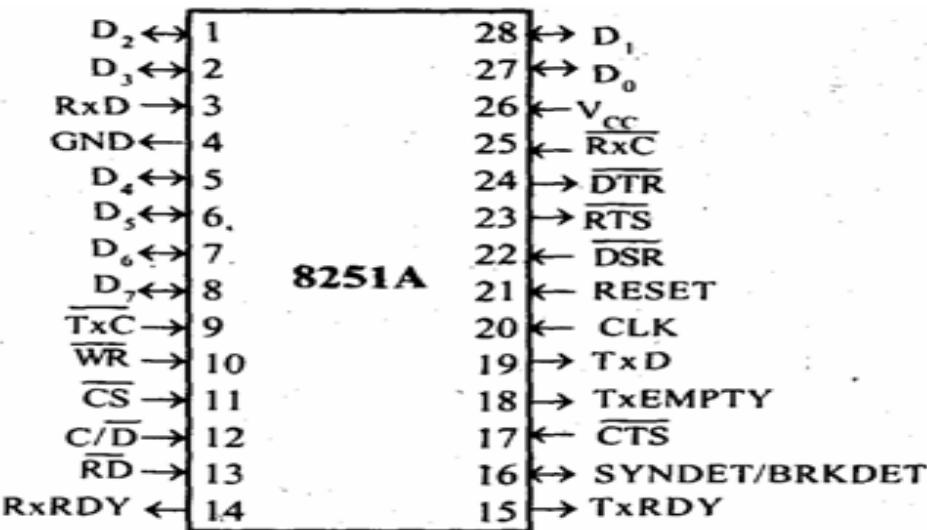
- Data to be transmitted is sent out one character at a time and the receiver end of the communication line synchronization is performed by examining synchronization bits that are included at the beginning and at the end of each character



8251

- The 8251A is a programmable serial communication interface chip designed for synchronous and asynchronous serial data communication.
- It supports the serial transmission of data.
- It is packed in a 28 pin DIP.

Pin details



Pin	Description
D ₀ -D ₇	Parallel data
C/D	Control register or Data buffer select
RD	Read control
WR	Write control
CS	Chip Select
CLK	Clock pulse (TTL)
RESET	Reset
TxC	Transmitter Clock
TxD	Transmitter Data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready
TxRDY	Transmitter Ready
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET/BRKDET	Synchronous Detect / Break Detect
RTS	Request To Send Data
CTS	Clear To Send Data
TxEMPTY	Transmitter Empty
V _{CC}	Supply (+5V)
GND	Ground (0 V)

Pin	Description
D0 - D7	parallel data
C/D	Control register or Data buffer select
RD	Read Control
WR	Write control
CS	Chip Select
CLK	clock pulse
RESET	Reset
TxC	Transmitter Clock
TxD	transmitted data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready
TxRDY	Transmitter Ready
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET/	Synchronous Detect/
BRKDET	DetectBreak
RTS	Request to send Data
CTS	Clear to send Data
TxEMPTY	Transmitter Empty
Vcc	Vcc (5V)
GND	Ground(0V)

2 Signal Description of 8251

D0 – D7: This is an 8-bit data bus used to read or write status, command word or data from or to the 8251A.

C / D: (Control Word/Data): This input pin, together with RD and WR inputs, informs the 8251A that the word on the data bus is either a data or control word/status information. If this pin is 1, control / status is on the bus, otherwise data is on the bus.

RD: This active-low input to 8251A is used to inform it that the CPU is reading either data or status information from its internal registers. This active-low input to 8251A is used to inform it that the CPU is writing data or control word to 8251A.

WR: This is an active-low chip select input of 8251A. If it is high, no read or write operation can be carried out on 8251. The data bus is tristated if this pin is high.

CLK: This input is used to generate internal device timings and is normally connected to clock generator output. This input frequency should be at least 30 times greater than the receiver or transmitter data bit transfer rate.

RESET: A high on this input forces the 8251A into an idle state. The device will remain idle till this input signal again goes low and a new set of control word is written into it. The minimum required reset pulse width is 6 clock states, for the proper reset operation.

TXC (Transmitter Clock Input): This transmitter clock input controls the rate at which the character is to be transmitted. The serial data is shifted out on the successive negative edge of the TXC.

TXD (Transmitted Data Output): This output pin carries serial stream of the transmitted data bits along with other information like start bit, stop bits and parity bit, etc.

RXC (Receiver Clock Input): This receiver clock input pin controls the rate at which the character is to be received.

RXD (Receive Data Input): This input pin of 8251A receives a composite stream of the data to be received by 8251 A.

RXD (Receive Data Input): This input pin of 8251A receives a composite stream of the data to be received by 8251 A.

RXRDY (Receiver Ready Output): This output indicates that the 8251A contains a character to be read by the CPU.

TXRDY - Transmitter Ready: This output signal indicates to the CPU that the internal circuit of the transmitter is ready to accept a new character for transmission from the CPU.

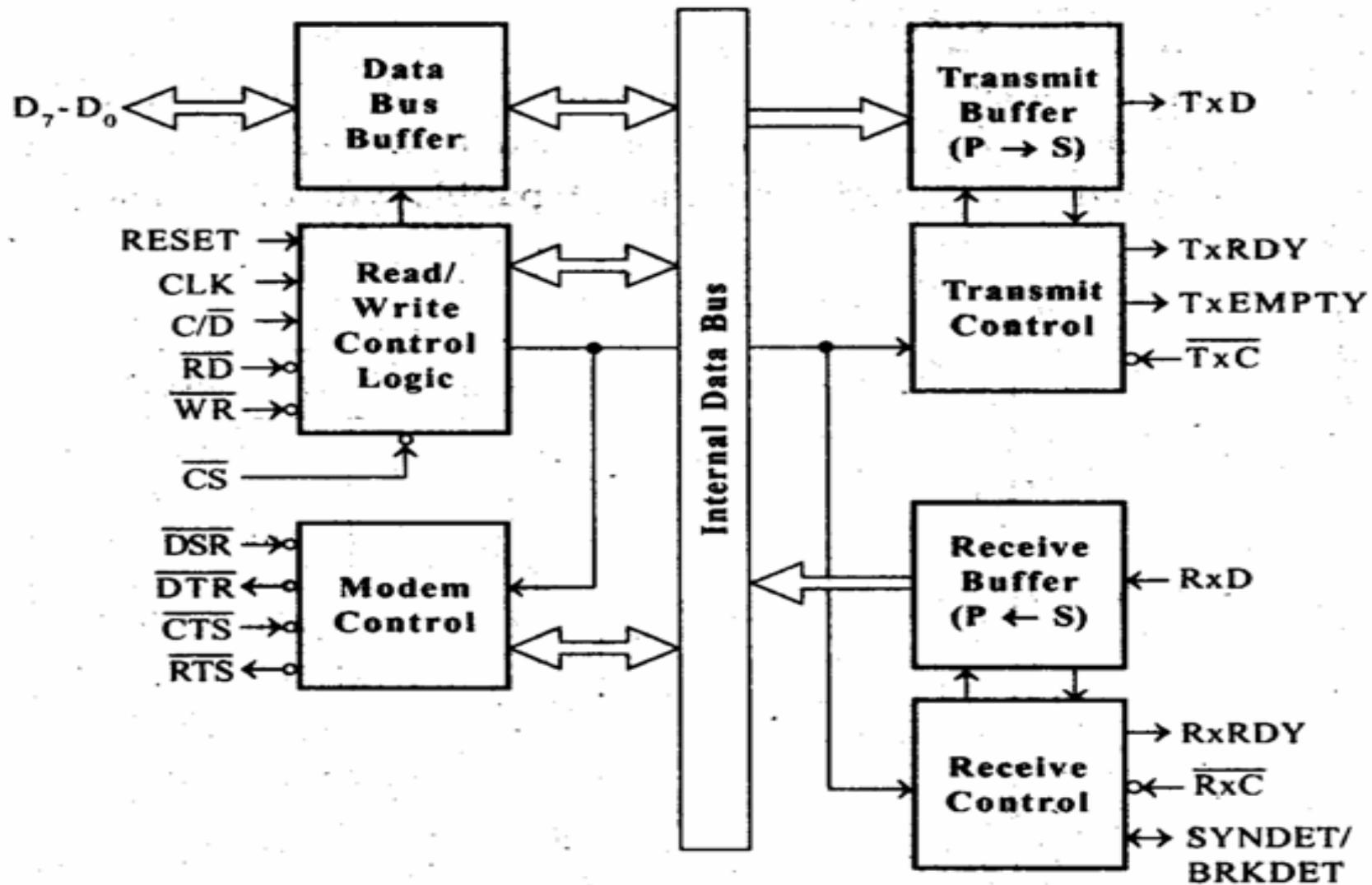
DSR - Data Set Ready: This is normally used to check if data set is ready when communicating with a modem.

DTR - Data Terminal Ready: This is used to indicate that the device is ready to accept data when the 8251 is communicating with a modem.

RTS - Request to Send Data: This signal is used to communicate with a modem.

TXE- Transmitter Empty: The TXE signal can be used to indicate the end of a transmission mode.

Architecture



Arch - details

- The functional block diagram of 825 1A consists five sections.

They are:

- Read/Write control logic
- Transmitter
- Receiver
- Data bus buffer
- Modem control.

Read/Write control logic

- The Read/Write Control logic interfaces the 8251A with CPU, determines the functions of the 8251A according to the control word written into its control register.
- It monitors the data flow.
- This section has three registers and they are control register, status register and data buffer.
- The active low signals RD, WR, CS and C/D(Low) are used for read/write operations with these three registers.

Read/Write control logic

- When C/D(low) is high, the control register is selected for writing control word or reading status word.
- When C/D(low) is low, the data buffer is selected for read/write operation.
- When the reset is high, it forces 8251A into the idle mode.
- The clock input is necessary for 8251A for communication with CPU and this clock does not control either the serial transmission or the reception rate.

Transmitter

- The transmitter section accepts parallel data from CPU and converts them into serial data.
- The transmitter section is double buffered, i.e., it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits.
- When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register.

Transmitter

- If buffer register is empty, then TxRDY is goes to high.
- If output register is empty then TxEMPTY goes to high.
- The clock signal, TxC (low) controls the rate at which the bits are transmitted by the USART.
- The clock frequency can be 1,16 or 64 times the baud rate.

Receiver

- The receiver section accepts serial data and convert them into parallel data
- The receiver section is double buffered, i.e., it has an input register to receive serial data and convert to parallel, and a buffer register to hold the parallel data.
- When the RxD line goes low, the control logic assumes it as a START bit, waits for half a bit time and samples the line again.
- If the line is still low, then the input register accepts the following bits, forms a character and loads it into the buffer register.

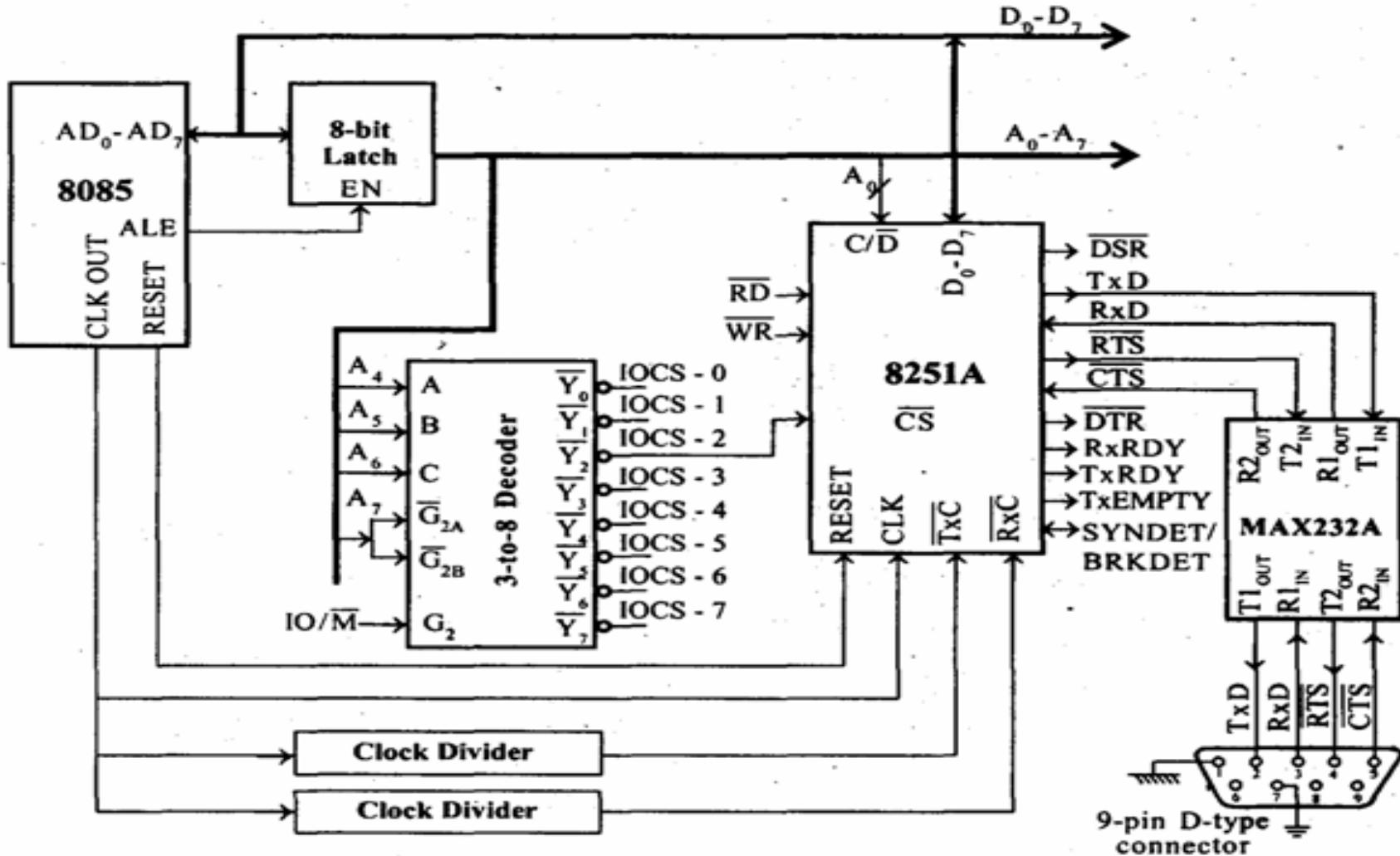
Receiver

- The CPU reads the parallel data from the buffer register.
- When the input register loads a parallel data to buffer register, the RxRDY line goes high.
- The clock signal RxC (low) controls the rate at which bits are received by the USART.
- During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.
- During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.

Modem control

- The MODEM control unit allows to interface a MODEM to 8251A and to establish data communication through MODEM over telephone lines.
- This unit takes care of handshake signals for MODEM interface.
- The 825 1A can be either memory mapped or I/O mapped in the system.
- 8251A in I/O mapped in the system is shown in the figure.
- Using a 3-to-8 decoder generates the chip select signals for I/O mapped devices.
- The address lines A4, A5 and A6 are decoded to generate eight chip select signals (IOCS-0 to IOCS-7) and in this, the chip select signal IOCS-2 is used to select 8251A.
- The address line A7 and the control signal IO / M(low) are used as enable for decoder.
- The address line A0 of 8085 is connected to C/D(low) of 8251A to provide the internal addresses.

8251 interfaced with 8085



Modem control

- The data lines D0 - D7 are connected to D0 - D7 of the processor to achieve parallel data transfer.
- The RESET and clock signals are supplied by the processor. Here the processor clock is directly connected to 8251A. This clock controls the parallel data transfer between the processor and 8251A.
- The output clock signal of 8085 is divided by suitable clock dividers like programmable timer 8254 and then used as clock for serial transmission and reception.

Modem control

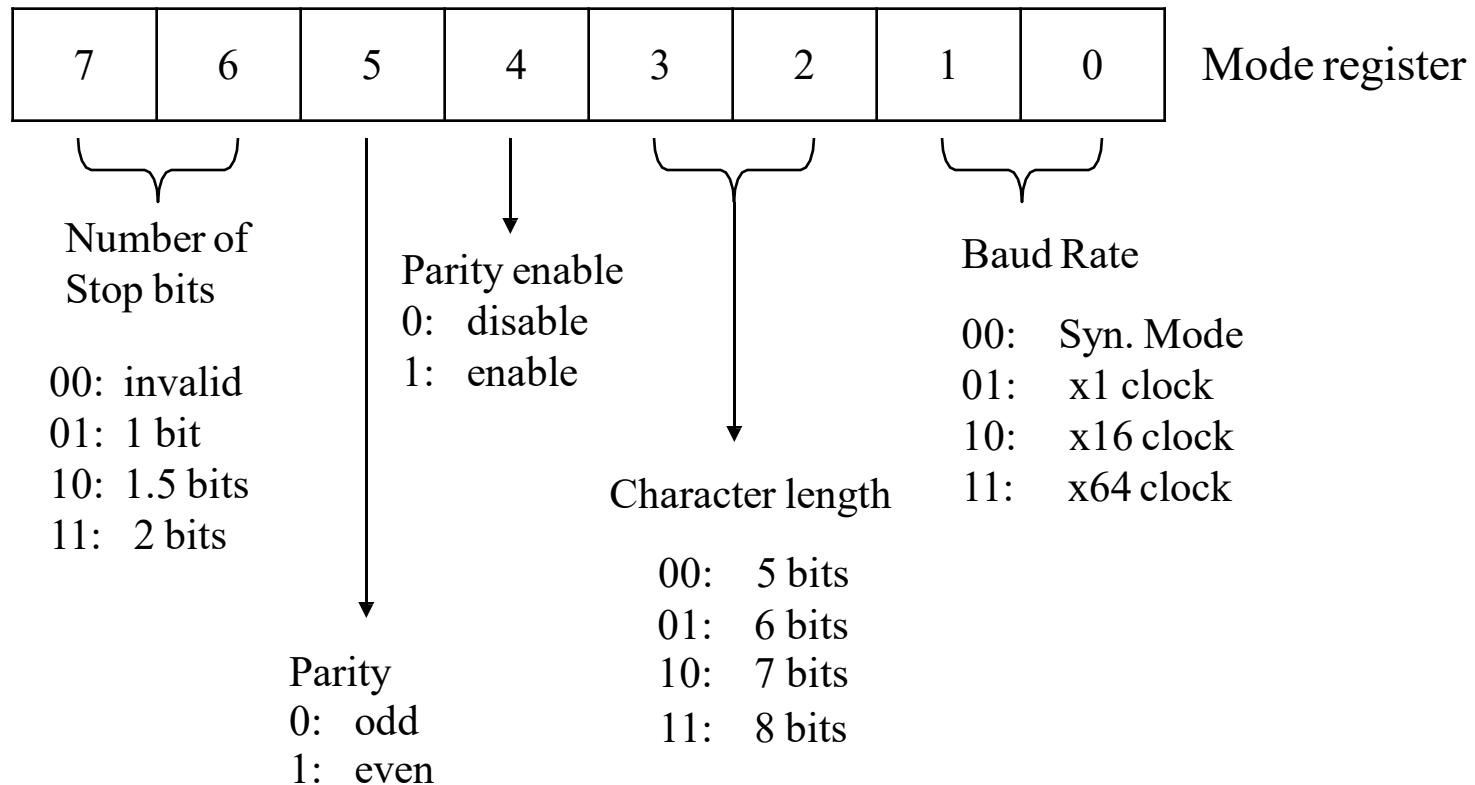
- The TTL logic levels of the serial data lines and the control signals necessary for serial transmission and reception are converted to RS232 logic levels using MAX232 and then terminated on a standard 9-pin D-type connector.
- In 8251A the transmission and reception baud rates can be different or same.
- The device which requires serial communication with processor can be connected to this 9-pin D-type connector using 9-core cable
- The signals TxEMPTY, TxRDY and RxRDY can be used as interrupt signals to initiate interrupt driven data transfer scheme between processor and 8251

Modem control

- The CPU reads the parallel data from the buffer register.
- When the input register loads a parallel data to buffer register, the RxRDY line goes high.
- The clock signal RxC (low) controls the rate at which bits are received by the USART.
- During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.
- During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.

A0	RD	WR	Task	Port Address
0	0	1	Read Data Word	90H
0	1	0	Write Data Word	90H
1	0	1	Read Status Word	91H
1	1	0	Write Control Word	91H

8251 mode register



8251 command register

EH	IR	RTS	ER	SBRK	RxE	DTR	TxE
----	----	-----	----	------	-----	-----	-----

TxE: transmit enable

DTR: data terminal ready

RxE: receiver enable

SBPRK: send break character

ER: error reset

RTS: request to send

IR: internal reset

EH: enter hunt mode

8251 status register

DSR	SYNDET	FE	OE	PE	TxEMPTY	TxRDY
-----	--------	----	----	----	---------	-------

TxRDY:	transmit ready
RxRDY:	receiver ready
TxEMPTY:	transmitter empty
PE:	parity error
OE:	overrun error
FE:	framing error
SYNDET:	sync. character detected
DSR:	data set ready

8251 interfaced with 8085

