

# **Model Answer, Statistics and Verdict of Mid Semester Exam**

Dr A Sahu  
Dept of Comp Sc & Engg.  
IIT Guwahati

# Outline

- Very Easy
  - Question 1
- Easy
  - Question 2
- OK, If u have read the book/slide
  - Question 4
  - Question 5
- A Bit difficult one
  - Question 3 : Two IC 8255, 8254, 3x2 7SegLED,  
Binary to BCD conversion

# Question No 1

- [10] Answers the following short questions:
  - [2] What is memory mapped I/O and IO mapped IO. Explain data movements (instruction and addressing) from IO to CPU and addressing methods in both the cases?
    - *In I/O mapped IO: addresses are 8 bit, IO can be performed by IN, OUT instruction, Higher address are same as lower address*
    - *In memory mapped IO: addresses are 16 bit, IO can be performed by LD, ST instruction, MOV M R, MOV R M*

# Question No 1

- [10] Answers the following short questions:
  - [1.5] What are different conditions to satisfy for data transfer from CPU to I/O or vice versa?
    - *Unconditional, status check, polling, interrupt*, Interrupt, Ready signal, Handshake-DMA(*hold signal*)...*Answer is in questions itself*
  - [1.5] What is difference between Status check, Polling and interrupt?
    - **Polling and Status check are same.. Periodically Check/Poll the device ready/finish status and run take the action if status is Ok..** There is diff between ready signal based (MPU check its own bit/pin, device updates to MPU ready bits) and polling (MPU check the device ready signal).
    - **Interrupt : MPU do its own work, Device interrupt the MPU when it is ready and ISR is executed when it receive an intr.**

# Question No 1

- [10] Answers the following short questions:
  - [2] Explain steps in transferring data from Memory to IO using DMA controller using HOLD & HLODA.
  - **HOLD: DMA to CPU**
    - DMA Send HOLD High to CPU
    - I (DMA) want BUS Cycles
  - **HOLDA**
    - CPU send HOLDA
    - BUS is granted to DMA to do the transfer
    - DMA is from Slaves to Master mode
  - **HOLD Low to CPU**
    - I (DMA) finished the transfer
  - **Cycle Stealing if One BUS**

# Question No 1

- [10] Answers the following short questions:
  - [3] What algorithms/methodologies are used for Analog to Digital conversion?
    - Counter or Tracking ADC, Successive Approximation ADC (Most Commonly Used), Parallel/Flash ADC (Fast Conversion)
  - What are different performance characteristics measures of DAC?
    - Resolution, Reference Voltages, Settling Time, Linearity, Speed, Errors

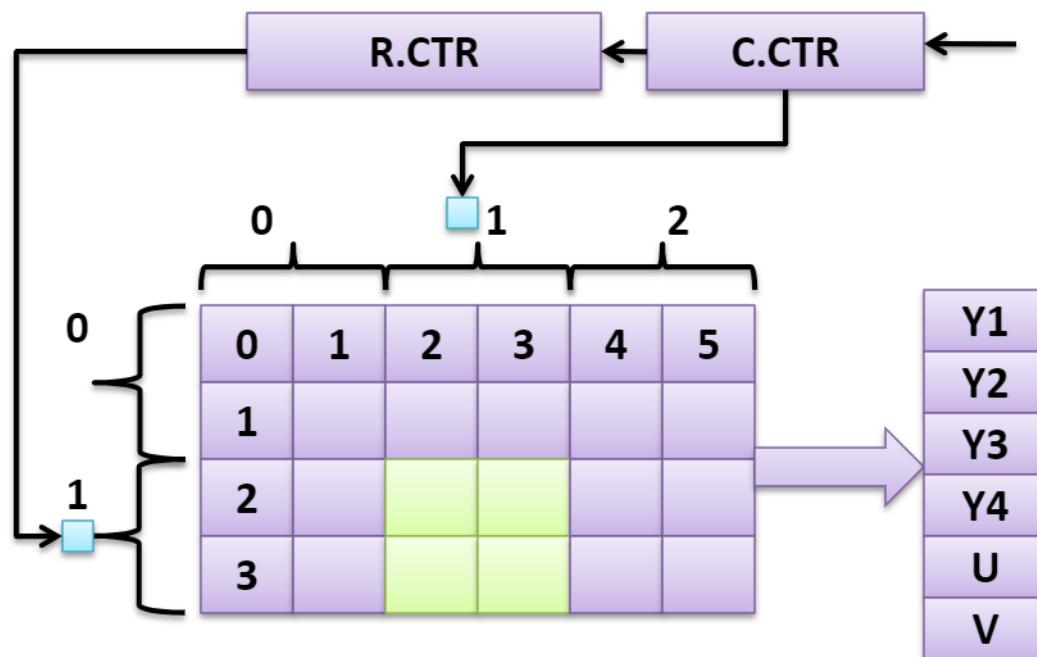
# Question No :2

- Suppose a graphics display having resolution 1600x1200 (1.9MegaPixel) with Frame buffer containing YUV values instead of RGB color values. **The human eye has fairly little color sensitivity:** the accuracy of the brightness information of the luminance channel (Y) has far more impact on the image discerned than that of the other two (U and V). In case of RGB we store 24 (8+8+8) bits per pixel in frame buffer , but in YUV case we store  $4 \times 8 + 8 + 8 = 48$  bits for group of 4 pixels ( that is 4 Y values 8 bit each + 1 U value of 8 bit + 1 V value of 8 bit for group of 4 adjacent pixels ). The RGB values can be computed from YUV values by a decoder to render the screen. (Assume YUV to RGB decoder is a black box, you don't require to draw the internal and you can use 4 such decoder).

Design a circuit block diagram to refresh the above designed graphics screen at **30frame/S**. What should be Clock values for the both row and column counter to perform at this rate? What should be maximum delay of such YUV to RGB decoder?

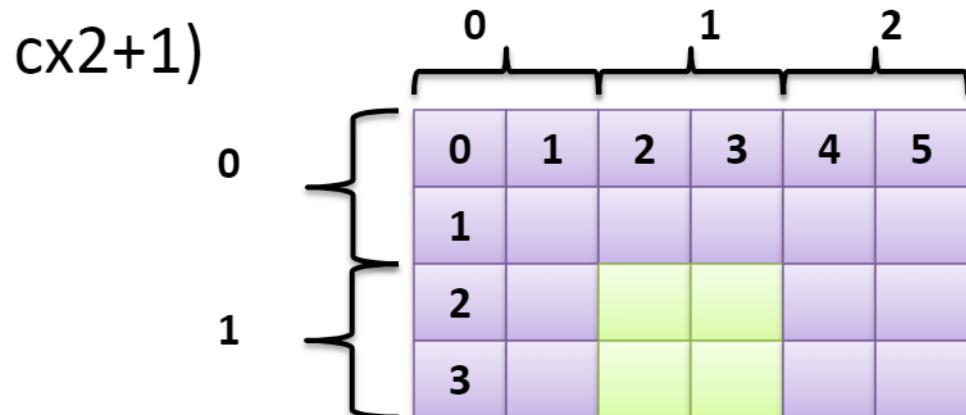
# Model Answer to question 2

- YUG coding instead of RGB, 4 Adjacent pixel (two row & 2 col)
- Store Y U,V in different Array
- Use same Col-Row counter, access U, V by  $(r/2,c/2)$  and but access Y by  $(r,c)$



# Model Answer to question 2

- YUG coding instead of RGB, 4 Adjacent pixel (two row & 2 col)
- Store Y1,Y2,Y3,Y4,U,V of Group in a single address
- Effective row and column
  - $1600/2=800$  rows,  $1200/2=600$  rows
  - Render  $(rx2,cx2)$ ,  $(rx2,cx2+1)$ ,  $(rx2+1,cx2)$ ,  $(rx2+1,cx2+1)$



- Use Shift register for Multiplication
- Last bit flipper for addition

# Clock and Decoder Delay

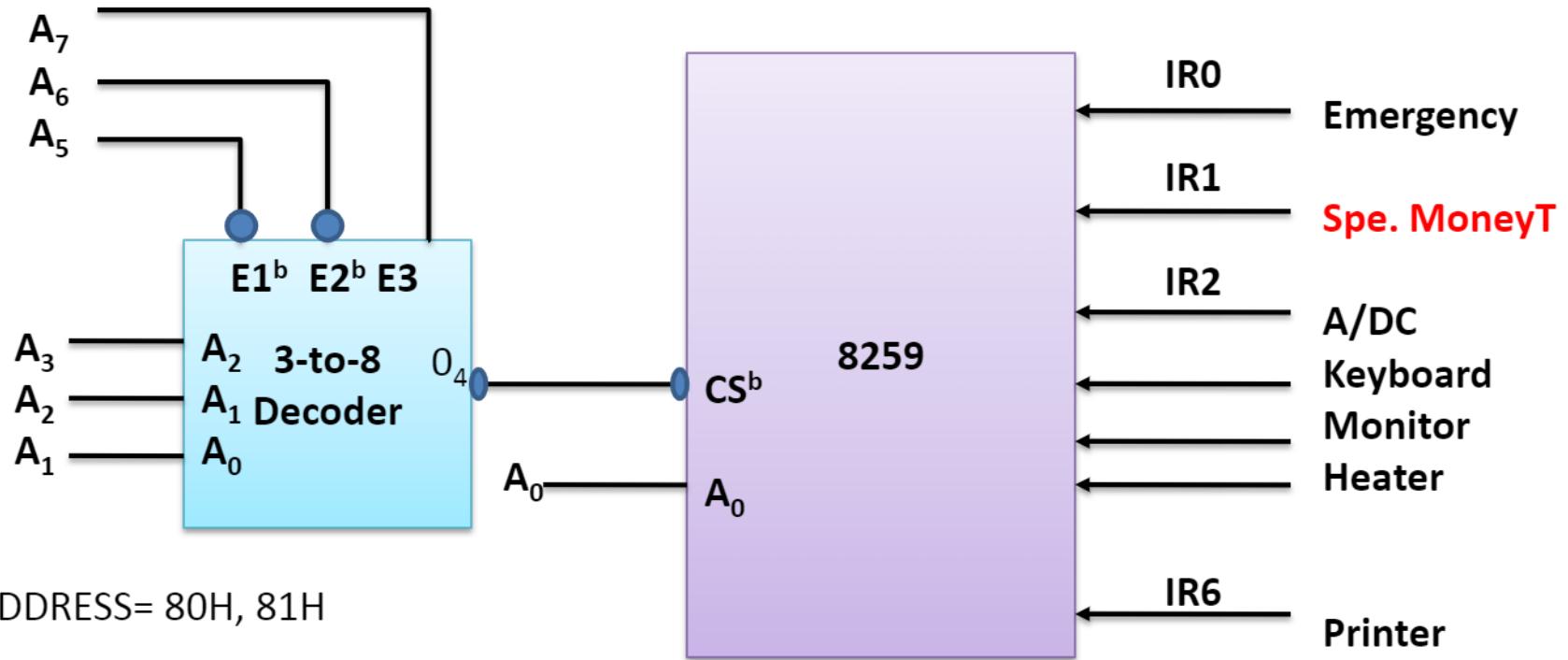
- 30F/S, 1.9MP
- Clock Speed= $(30 \times 1.9 \times 10^6) = 57.6\text{MHz}$
- YUV to RGB Decoder speed  $\geq 1/30 \times 1.9 \times 10^6$   
 $= 1.736 \times 10^{-8}\text{S}$
- Use of 4 decoder: Speed can be reduced to  
 $(1.736/4) \times 10^{-8}\text{S}$

# Question 4

[10] Design interface and write interface program to interface seven I/Os devices namely Emergency actions, A/D converter, Heater, Keyboard, Display, Special money transaction device and printer using **8259** interrupt controller. Design interface the circuit to work at address 80H for ICW1 and 81H for ICW2. Special money transaction device have lower priority then Emergency actions. You have to ensure that Special money transaction device should not be interrupted once its ISR started till the end of its ISR using mask bit. Assume interrupt vector address for each interrupts and use properly in your interface program.

Determine the values of ICWs, OCWs in this case for your interface structure. Draw the interface diagram and write interfacing (8085 assembly) program (to initialize ICWs & OCWs) to perform this work.

# Example: Setting of control word



AD0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7= 1	M6= 1	M5= 1	M4= 1	M3= 1	M2= 1	M1= 0	M0= 1
Interrupt Masks: 1= Mask Set, 0 =Mask reset								

OCW1=FDH

# Initialization words (ICW1 & ICW2)

AD0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	1	0	1	1	0
A7, A6,A5 Lower address bit of Vector Address			0 for Edge Trigger		Call Address interval =4	1=single 0=Cascade		

76H

AD0	D7	D6	D5	D4	D3	D2	D1	D0
1	T7	T6	T5	T4	T3	T2	T1	T0
	0	0	1	0	0	0	0	0
T7=T0 is the assign to IR0, Vector address for ISR Lower Byte of call address								

20H

Vector Address 2060, 2064....	0100 0000	0 1 1	00000
----------------------------------	-----------	-------	-------

## Program to initialize

DI

MVI A, **76H** ;move ICW1 byte to ACC

OUT **80H** ; initialize 8259A ICW1

MVI A, **20H** ; Mov ICW2 byte to ACC

OUT **81H** ; Initialize 8259A ICW2

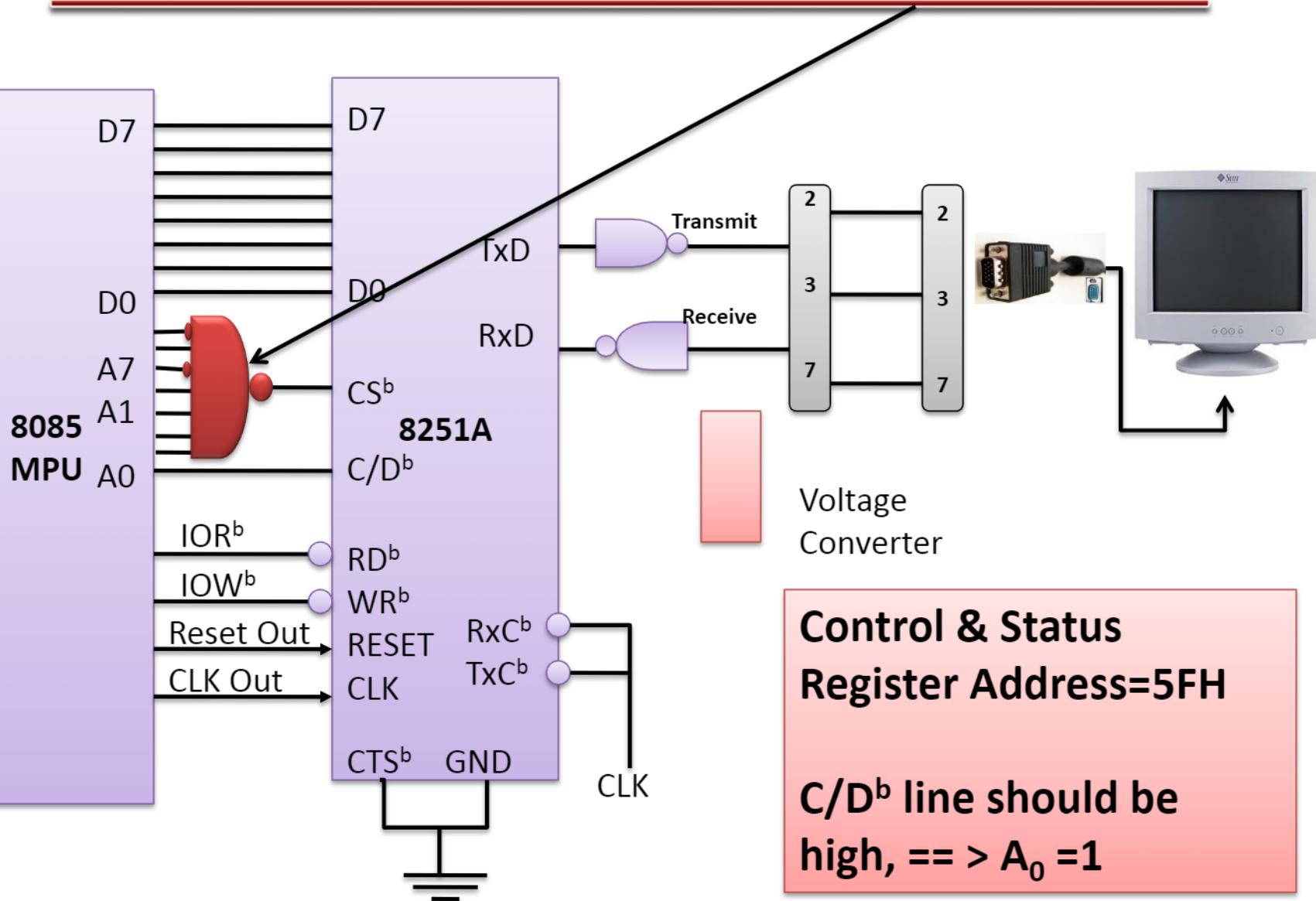
MVI A, **FDH** ; Put the OCW1

OUT **80H**

# Question 5

- [10] Interface a LCD monitor via RS-232 port using a 8251 USART controller. Design the interface diagram so that you should be able to address the SUART at 5EH and 5FH for data and control respectively. Specify initialization instructions and status word to transmit characters with Async mode with 9600 baud rate, 8 bit character length, 2 stop bits and without any parity check.
- Write instructions to initialize USART and read status word and setup a loop until the USART transmitter is ready. Using this write an interface program that will send 10 characters from memory to LCD monitor for display.

# Interface : For Address 5FH & 5EH



# Initialization of UART: 9600baud, 8 bit char , 2 stop bit, WO parity

Mode Word	D7	D6	D5	D4	D3	D2	D1	D0	CEH
	1	1	0	0	1	1	1	0	
	Two Stop bits		No parity		7 bit characters			Baud=TxC/16 =153.6k/16 =9600	
COMMAND WORD	D7	D6	D5	D4	D3	D2	D1	D0	11H
	X	0	X	1	X	0	X	1	
				ERR Reset		Receive Disable		Transmit Enable	
STATUS	D7	D6	D5	D4	D3	D2	D1	D0	01H
	X	X	X	X	X	X	X	1	
								Transmit Ready	

# Initialization instructions

SETUP:	MVI A, <b>C EH</b> ; <i>load mode word</i>
	OUT <b>5 FH</b> ; <i>Write mode word in control register</i>
	MVI A, <b>11 H</b> ; <i>load command word to enable TX</i>
	OUT <b>5 FH</b> ; <i>Enable the transmitter</i>
STATUS:	IN <b>5 FH</b> ; <i>Read the status register</i>
	ANI <b>01 H</b> ; <i>Mask all bit except D0</i>
	JZ STATUS ; <i>if D0=0 the TX buffer if full</i>

# Write a program to display message

## at CRT terminal

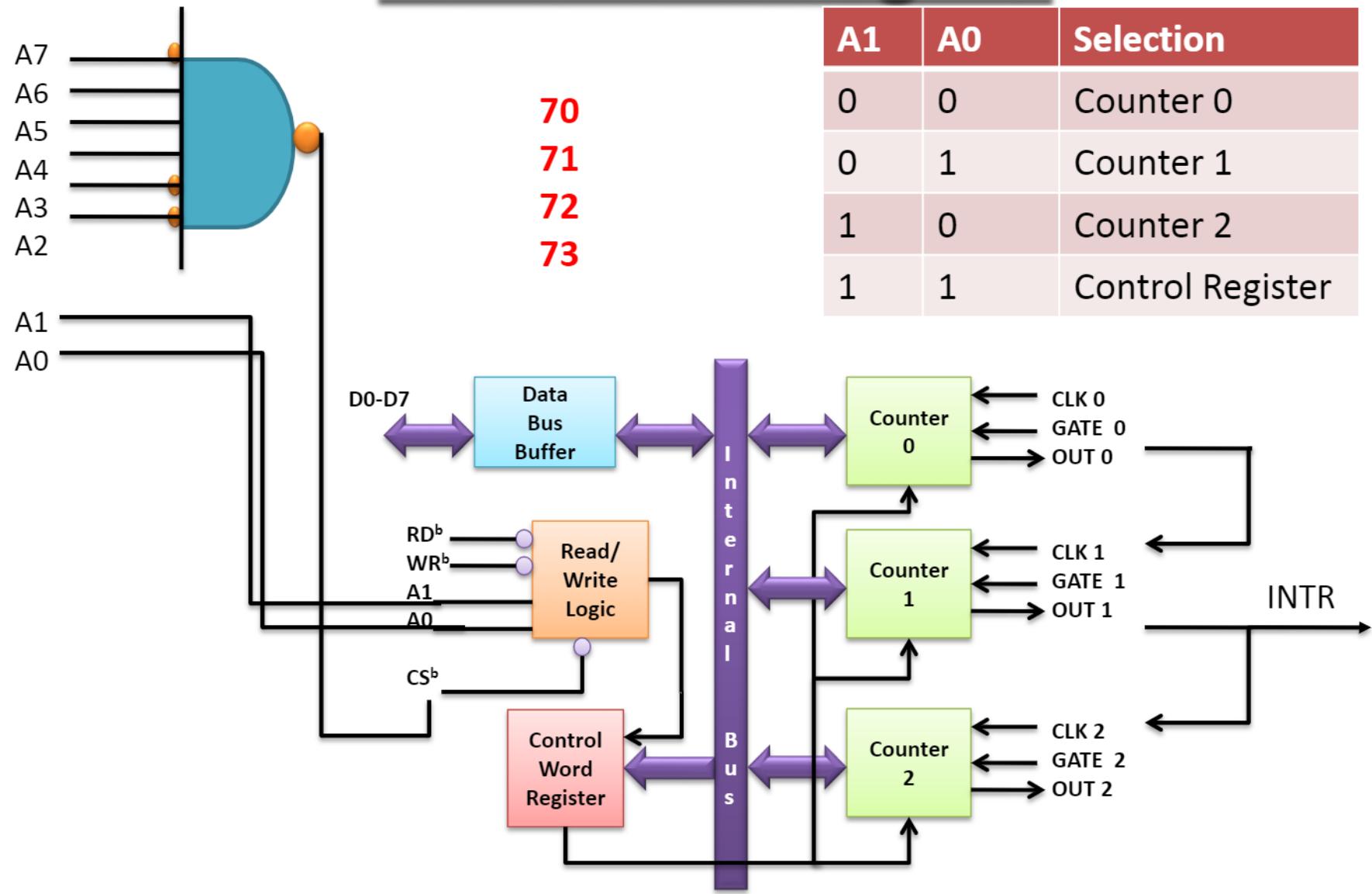
LXI	H 2070H ; Meory ptr for Message
MOV	C, M ; Set up Ctr register
MVI	A,40; Reset 8251
OUT	<b>5FH</b>
MVI	A, <b>CEH</b> ; Initialize 8251
OUT	<b>5FH</b>
MVI	A, <b>11</b> ; initialize for transmit
OUT	<b>5FH</b>

STATUS:	IN	<b>5FH</b>
	ANI	01H ;Ckeck TxRDY
	JZ	STATUS ; is txRDY 1 ? If not wait
	INX	H ; Pont to Next Char
	MOV	A,M ; place the Char in ACC
	OUT	<b>5EH</b> ; Send the Char to Transmitter
	DCR	C ; DCr cnt
	JNZ	STATUS ;Again Send the rest of Char
	HLT	

# Question 3

- [10] Design a real time clock with hours, minute and second with 6 seven segment LED display. Interface an **8254 PIT** and a group of 6 seven segment LEDS using an **8255 PPC** to 8085 microprocessor. Generate 1 Second clock pulse using two counter (ctr0 & ctr1) of 8254PIT and one counter (12hr.60min.60Sec=43200 < max of 16 bit number 65368) for real time clock (ctr2). After every 1 second (generate interrupt to read) read the RTC counter value and format the value in proper format (BCDs HH:MM:SS) and send to output port using 8255.
- Draw the interface diagram and write interfacing (8085 assembly) program to perform this work.

# 8254 Block Diagram



# Write a SR to generate an interrupt

## every 1 Second

- Assume Clock Freq=2MHz
- Count is too large
- Counter 1 load with 50,000 to generate 25ms
  - CNTLOAD= $50,000_{10}$ =C350H
- Counter 2 load with 40 to generate  $25\text{ms} \times 40 = 1\text{s}$  pulse (CNTLOAD= $40_{10}$ =28H)
- CNT3Load =  $12 \times 60 \times 60 = 43200_{10}$ =A8C0H
- Counter1 output is to counter 2
- Counter2 output is to counter 3
- All Counter 1, Counter 2 & Counter 3 in Mode 2

# Instruction to set up 1s interrupt

MVI	A , 74H	; Mode for 1 <sup>st</sup> CTR
OUT	73H	;Write in control register
MVI	A,94H	; Mode for 2 <sup>nd</sup> CTR
OUT	73H	; Write to control register
MVI	A,F4H	; Mode for 3 <sup>rd</sup> CTR
OUT	73H	; Write to control register
MVI	A,50	; low byte of CTR1=C350
OUT	70H	; load to CTR1 low byte
MVI	A,C3	; high byte of CTR1=C350
OUT	70H	; load to CTR1 high byte
MVI	A,28H	; Count for Counter 2
OUT	71H	; Load Counter 2
MVI	A,C0	; low byte of CTR3=A8C0H
OUT	72H	; load to CTR1 low byte
MVI	A,A8	; high byte of CTR3=A8C0H
OUT	72H	; load to CTR1 high byte

# Control word

- Counter 1 (74H)

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC2	RW1	RW0	M2	M1	M0	BCD
01		Load 16 bit (11)		010 (mode 2)			0

- Counter 2 (94H)

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC2	RW1	RW0	M2	M1	M0	BCD
10		Load 8 bit (01)		010 (mode 2)			0

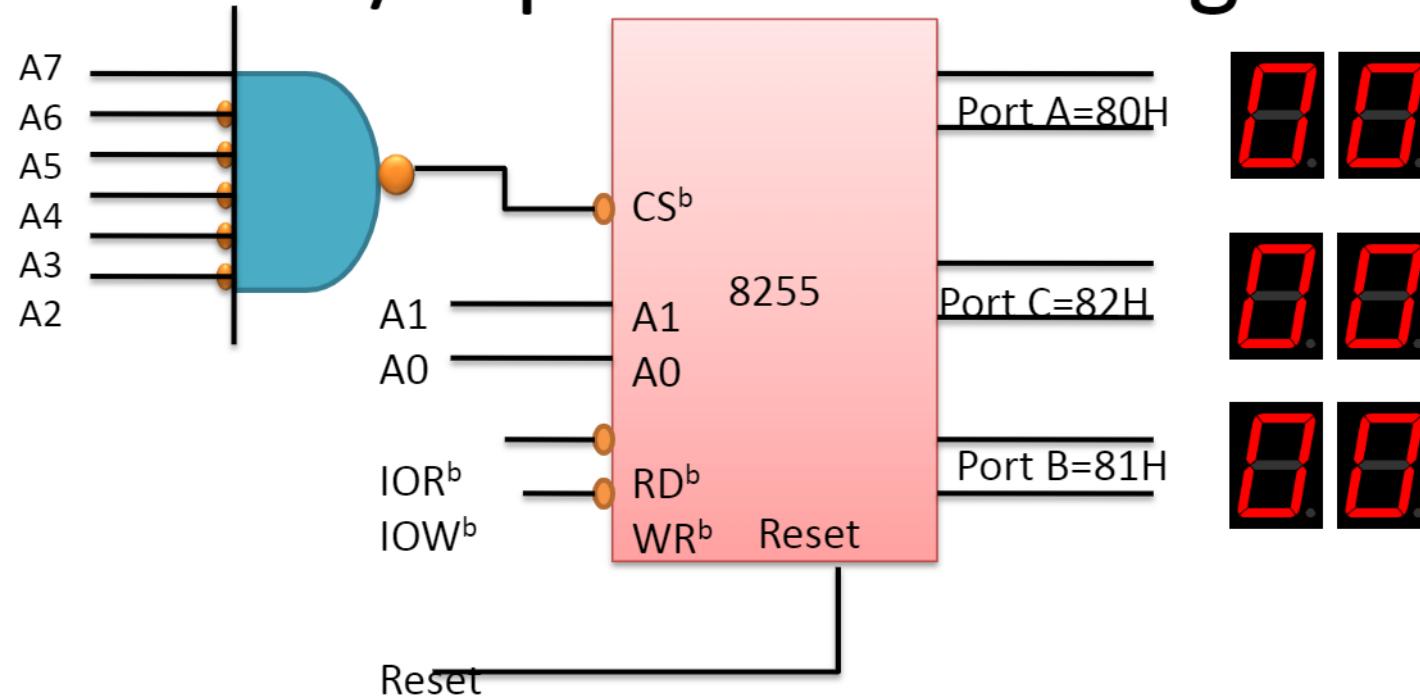
- Counter 3 (F4H)

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC2	RW1	RW0	M2	M1	M0	BCD
11		Load 16 bit (11)		010 (mode 2)			0

# Require 8255

- Port A : Output (Sec)
- Port B: Out put (Min)
- Port C: Output (Hr)
- In mode 0
  - Two 8-bit ports and two 4-bit ports
  - Any port can be input or output
  - Outputs are latched, inputs are not latched

# I/O port Addressing-8255



CS <sup>b</sup>	A1	A0	HEX Address	Port
A7 A6 A5 A4 A3 A2 1 0 0 0 0 0	0	0	= 80H	A
	0	1	= 81H	B
	1	0	= 82H	C
	1	1	= 83H	Control Register

# 8255: Mode 0, Example 1

- Configure
  - Port A, B and C as out port
- Control word
- Address: 80H, 81H ,82H for ABC

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0
I/O function	Port A in Mode 0	Port A as O/P	Port C <sub>U</sub> As O/P	Port B in Mode 0	Port B As O/P	Port C <sub>L</sub> As O/P	

80H

# Interface Program

MVI A,80H ; Load acc with Control word  
OUT 83H ; Load control register with 83  
at port address 83

CALL BUSYWAIT()

ISRDISPLY:

CALL GETTIMESTOREAT2000()

MOV A, 2000H

OUT 81H ; Write to LEDs

MOV A, 2001H

OUT 82H ; Write to LED

MOV A, 2002H

OUT 83H ; Write to LED

RET

**Thanks**