DELHI SKILL AND ENTREPRENEURSHIP UNIVERSITY

End-Semester Examination – Set 2

B.Tech CSE – Microprocessors and Microcontrollers (BT-EC-ES511)

Max. Marks: 100 Time: 3 Hours

Instructions:

1. This question paper contains two sections: Section A and Section B.
2. Attempt **all** questions.
3. Figures to the right indicate full marks.
4. Assume necessary data if missing.

SECTION A – Short Answer Questions

Attempt all questions. Answers should be concise (3–4 lines).

Q1. Distinguish between Maskable and Non-Maskable interrupts in 8085 with examples.

Answer: ***Maskable Interrupts*** *(e.g., RST 7.5, RST 6.5, INTR) can be disabled or delayed by software instructions (EI/DI).*
Non-Maskable Interrupts *(e.g., TRAP) have the highest priority and cannot be disabled by software; they are typically used for critical power-failure events.*

Q2. Define 'Machine Cycle' and 'T-State' in the context of microprocessor timing.

Answer: *A* ***Machine Cycle*** *is the time required by the microprocessor to complete one operation of accessing memory or I/O (e.g., Opcode Fetch).*
A ***T-State*** *is one subdivision of the operation performed in one clock period. A machine cycle consists of 3 to 6 T-States.*

Q3. What is the function of the segment registers (CS, DS, SS, ES) in 8086?

Answer: *These 16-bit registers hold the base addresses of memory segments:*

- ***CS (Code Segment):*** *Points to the program code.*
- ***DS (Data Segment):*** *Points to the data variables.*
- ***SS (Stack Segment):*** *Points to the stack memory.*
- ***ES (Extra Segment):*** *Used for additional data storage.*

Q4. Explain the purpose of the $\overline{\text{BHE}}$ (Bus High Enable) signal in 8086.

Answer: *$\overline{BHE}$ is an active-low signal used to enable the most significant data bus half ($D_8 - D_{15}$) during a data transfer. It is used in conjunction with A_0 to select between odd bank, even bank, or both (16-bit word transfer).*

Q5. How does the `READY` signal synchronize the processor with slow peripherals?

Answer: *The `READY` input is sampled by the processor. If `READY` is low, the processor enters a "Wait State" (T_W) and extends the bus cycle, allowing slower memories or I/O devices extra time to place data on the bus.*

Q6. List the major features of the 8259 Programmable Interrupt Controller (PIC).

Answer:

(a) *Manages up to 8 vectored priority interrupts.*

(b) *Can be cascaded to support up to 64 interrupts.*

(c) *Supports different priority modes (Fixed, Rotating).*

(d) *Programmable to work with 8085 or 8086 processors.*

Q7. What is the resolution of an n-bit Analog-to-Digital Converter (ADC)?

Answer: *Resolution is the smallest change in analog input voltage that causes a change of 1 LSB in the digital output.*

$$Resolution = \frac{V_{ref}}{2^n}$$

Where V_{ref} is the reference voltage and n is the number of bits.

Q8. What is the function of the BSR (Bit Set/Reset) mode in 8255 PPI?

Answer: *The BSR mode is used specifically to set or reset individual bits of **Port C**. It does not affect Ports A or B. It is useful for control applications where single-bit manipulation (like turning on an LED connected to PC3) is required.*

Q9. Identify the alternate functions of Port 3 pins P3.0 and P3.1 in 8051.

Answer: *Port 3 pins have multifunctional roles:*

- ***P3.0 (RXD):** Serial Data Input pin.*
- ***P3.1 (TXD):** Serial Data Output pin.*

These are used for UART serial communication.

Q10. Explain the role of the DPTR (Data Pointer) register in 8051.

Answer: *DPTR is the only user-accessible 16-bit register in 8051, consisting of DPH (High byte) and DPL (Low byte). It is primarily used to point to data in external memory or code memory (e.g., lookup tables) for 16-bit addressing.*

SECTION B – Descriptive Questions

Attempt all questions. Include diagrams where necessary.

Q11. Write an assembly language program for the 8085 microprocessor to find the largest number in a block of data stored in memory. Assume the block length is at memory location 2050H and data starts from 2051H.

Answer:

Assembly Code:

```
LOOP: INX H ; Point to next number CMP M ; Compare A with Memory (A - M) JNC
SKIP ; If Carry=0 (A >= M), Jump to SKIP MOV A, M ; If Carry=1 (A < M), Update
A with new max
SKIP: DCR C ; Decrement counter JNZ LOOP ; If C != 0, repeat loop
STA 3050H ; Store the result (Largest Number) HLT ; Terminate Program
```

Algorithm:

(a) Load the counter (length of array) from 2050H.

(b) Load the starting address of the data (2051H).

(c) Move the first element to the Accumulator (Assume it is the largest).

(d) Decrement counter.

(e) Compare the next number in memory with the Accumulator.

(f) If Memory ¿ Accumulator, move Memory to Accumulator. Else, continue.

(g) Decrement counter. If not zero, repeat loop.

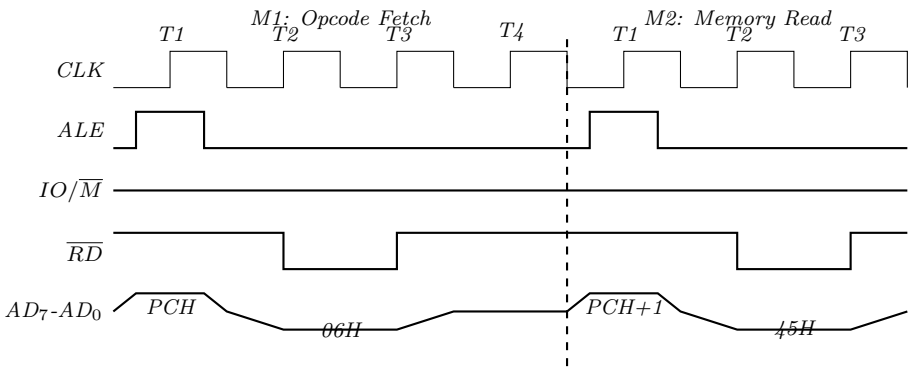
(h) Store the result (largest number) at 3050H.

Assembly Code:

```
LOOP: INX H ; Point to next number CMP M ; Compare A with Memory (A - M) JNC
SKIP ; If Carry=0 (A >= M), Jump to SKIP MOV A, M ; If Carry=1 (A < M), Update
A with new max
SKIP: DCR C ; Decrement counter JNZ LOOP ; If C != 0, repeat loop
STA 3050H ; Store the result (Largest Number) HLT ; Terminate Program
```

Q12. Draw and explain the Timing Diagram for the instruction `MVI B, 45H`.

Answer:

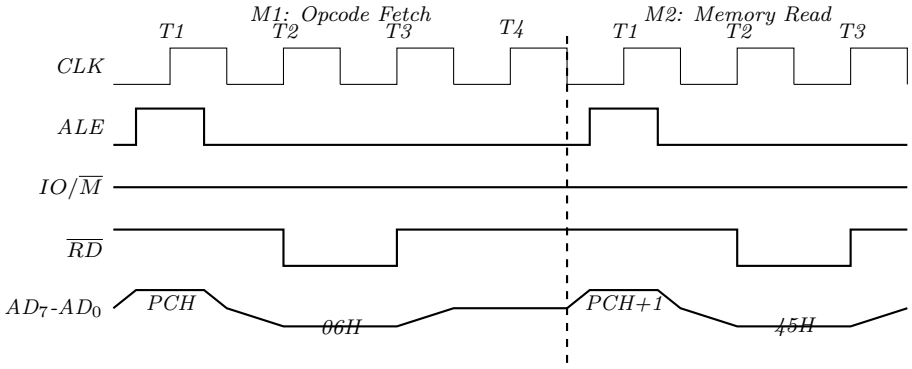


The instruction `MVI B, 45H` (Opcode 06H) is a 2-byte instruction requiring 2 Machine Cycles:

(a) ***M1 - Opcode Fetch (4 T-States):*** *Fetches 06H. ALE indicates address. RD reads opcode.*

(b) ***M2 - Memory Read (3 T-States):*** *Reads immediate data 45H.*

Total T-States $= 4 + 3 = 7$.



Q13. Explain the Addressing Modes of 8086 with one example for each.

Answer:

(a) ***Immediate Addressing:*** *The operand is part of the instruction.*
Ex: `MOV AX, 1234H` (Loads 1234H into AX)

(b) ***Register Addressing:*** *The operand is in a register.*
Ex: `MOV AX, BX` (Copies content of BX to AX)

(c) ***Direct Addressing:*** *The effective address (offset) is given directly.*
Ex: `MOV AX, [1000H]` (Moves content of DS:1000H to AX)

(d) ***Register Indirect Addressing:*** *The address is held in BX, BP, SI, or DI.*
Ex: `MOV AX, [BX]` (Moves content of memory pointed by BX to AX)

(e) ***Based Addressing:*** *Offset is sum of Base Register (BX/BP) and displacement.*
Ex: `MOV AX, [BX + 04H]` (Accesses array structures)

(f) ***Indexed Addressing:*** *Offset is sum of Index Register (SI/DI) and displacement.*
Ex: `MOV AX, [SI + 02H]`

(g) ***Based Indexed Addressing:*** *Combination of Base and Index registers.*
Ex: `MOV AX, [BX + SI]` (Useful for 2D arrays)

The 8086 provides several methods to specify the operand for an instruction:

(a) ***Immediate Addressing:*** *The operand is part of the instruction.*
Ex: `MOV AX, 1234H` (Loads 1234H into AX)

(b) ***Register Addressing:*** *The operand is in a register.*
Ex: `MOV AX, BX` (Copies content of BX to AX)

(c) ***Direct Addressing:*** *The effective address (offset) is given directly.*
Ex: `MOV AX, [1000H]` (Moves content of DS:1000H to AX)

(d) ***Register Indirect Addressing:*** *The address is held in BX, BP, SI, or DI.*
Ex: `MOV AX, [BX]` (Moves content of memory pointed by BX to AX)

(e) ***Based Addressing:*** *Offset is sum of Base Register (BX/BP) and displacement. Ex:* `MOV AX, [BX + 04H]` *(Accesses array structures)*

(f) ***Indexed Addressing:*** *Offset is sum of Index Register (SI/DI) and displacement. Ex:* `MOV AX, [SI + 02H]`

(g) ***Based Indexed Addressing:*** *Combination of Base and Index registers. Ex:* `MOV AX, [BX + SI]` *(Useful for 2D arrays)*

Q14. Compare Minimum Mode and Maximum Mode of 8086. List the control signals generated in each mode.

Answer:

- ***Minimum Mode (MN/$\overline{MX}$ = 5V):***
  - *The processor generates all control signals internally.*
  - *Suitable for small, single-processor systems.*
  - ***Signals:*** *$\overline{WR}$, M/$\overline{IO}$, DT/$\overline{R}$, $\overline{DEN}$, ALE, $\overline{INTA}$.*
  - *The DMA is handled by HOLD and HLDA pins.*
- ***Maximum Mode (MN/$\overline{MX}$ = GND):***
  - *Control signals are generated externally by the* ***8288 Bus Controller****.*
  - *Suitable for multiprocessor or coprocessor (8087) systems.*
  - ***Status Signals:*** *$\overline{S_0}, \overline{S_1}, \overline{S_2}$ are sent to the 8288 to generate control commands (MRDC, MWTC, etc.).*
  - *Instead of HOLD/HLDA, it uses $\overline{RQ}$/$\overline{GT}$ (Request/Grant) lines for bus arbitration.*

The 8086 is configured by the ***MN/$\overline{MX}$*** *pin (Pin 33).*

- ***Minimum Mode (MN/$\overline{MX}$ = 5V):***
  - *The processor generates all control signals internally.*
  - *Suitable for small, single-processor systems.*
  - ***Signals:*** *$\overline{WR}$, M/$\overline{IO}$, DT/$\overline{R}$, $\overline{DEN}$, ALE, $\overline{INTA}$.*
  - *The DMA is handled by HOLD and HLDA pins.*
- ***Maximum Mode (MN/$\overline{MX}$ = GND):***
  - *Control signals are generated externally by the* ***8288 Bus Controller****.*
  - *Suitable for multiprocessor or coprocessor (8087) systems.*
  - ***Status Signals:*** *$\overline{S_0}, \overline{S_1}, \overline{S_2}$ are sent to the 8288 to generate control commands (MRDC, MWTC, etc.).*
  - *Instead of HOLD/HLDA, it uses $\overline{RQ}$/$\overline{GT}$ (Request/Grant) lines for bus arbitration.*

Q15. Draw the block schematic for interfacing an ADC 0808/0809 with a microprocessor.

Answer:

[¡-¿, double, thick] ((adc.west) + (0, 0.5)) – ++(-2,0) node[left] Data Bus ($D_0 - D_7$); [left, anchor=east] at ((adc.west) + (0, 0.5)) D0-D7;

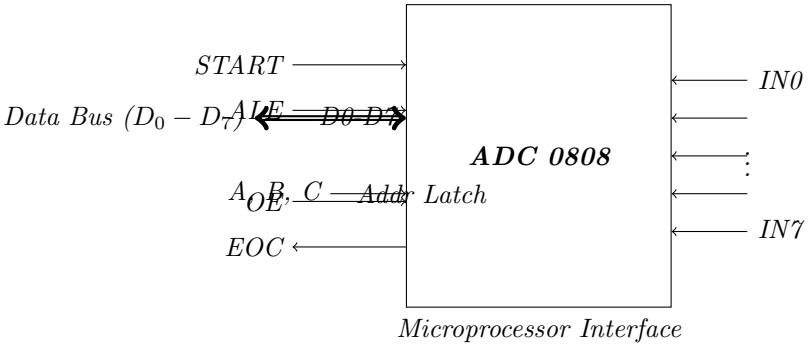
[¡-] ((adc.west)−(0, 0.5)) – ++(-1,0) node[left] A, B, C; [right, anchor=west] at ((adc.west)−(0.8, 0.5)) Addr Latch;

[¡-] ((adc.northwest)!0.2!(adc.southwest)) – ++(-1.5,0) node[left] START; [¡-] ((adc.northwest)!0.35!(adc – ++(-1.5,0) node[left] ALE; [¡-] ((adc.northwest)!0.65!(adc.southwest)) – ++(-1.5,0) node[left] OE; [-¿] ((adc.northwest)!0.8!(adc.southwest)) – ++(-1.5,0) node[left] EOC;

[¡-] ((adc.east)+(0, 1)) – ++(1,0); [¡-] ((adc.east)+(0, 0.5)) – ++(1,0); [¡-] ((adc.east)+(0, 0)) – ++(1,0); [¡-] ((adc.east) + (0, −0.5)) – ++(1,0); [¡-] ((adc.east) + (0, −1)) – ++(1,0); [right] at ((adc.east) + (1, 1)) IN0; [right] at ((adc.east) + (1, −1)) IN7; at ((adc.east) + (1, 0)) ⋮;

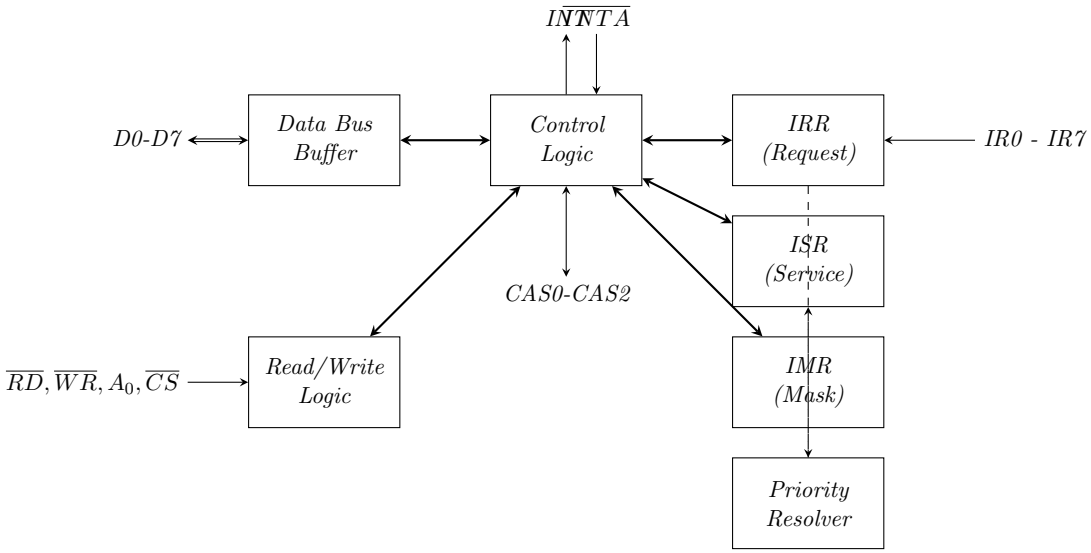
[below] at (adc.south) Microprocessor Interface;

Process: Select channel via A,B,C. Pulse ALE & START. Monitor EOC. When EOC goes High, assert OE to read data. *The ADC 0808 is an 8-bit, 8-channel multiplexed ADC.*



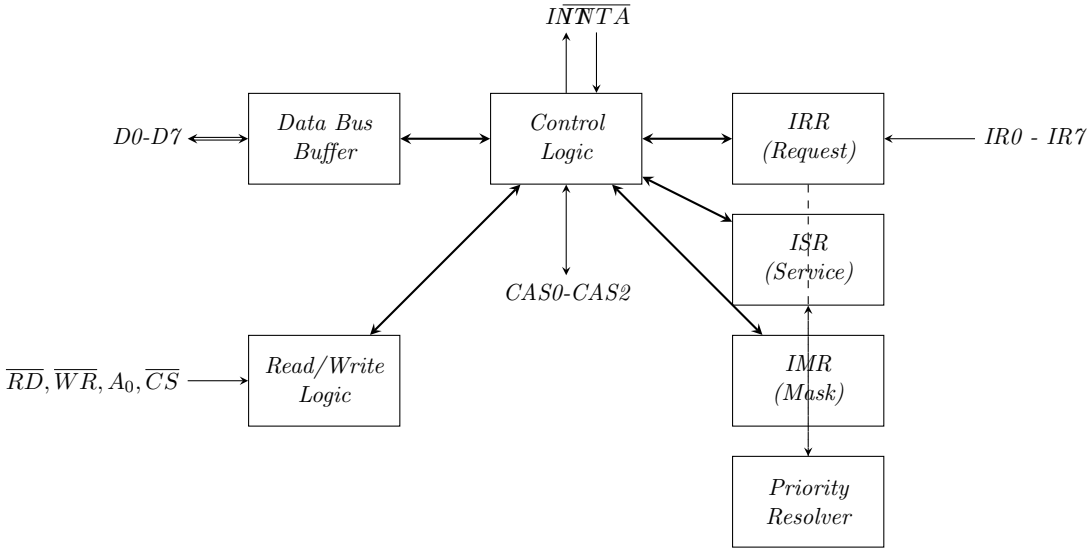
Process: *Select channel via A,B,C. Pulse ALE & START. Monitor EOC. When EOC goes High, assert OE to read data.*

Draw the block diagram of 8259A PIC and explain the function of IRR, ISR, and IMR.
Answer:



- ***IRR (Interrupt Request Register):*** *Stores bits for interrupts that are requesting service but not yet acknowledged.*
- ***ISR (In-Service Register):*** *Stores bits for interrupts currently being processed. Used to prevent lower priority interrupts from interrupting.*
- ***IMR (Interrupt Mask Register):*** *Stores masking bits. If a bit is set (1), that interrupt line is disabled (masked).*
- ***Priority Resolver:*** *Compares bits in IRR with ISR and IMR to determine if the new interrupt should be passed to the CPU.*

The 8259A handles priority interrupts.



- ***IRR (Interrupt Request Register):*** *Stores bits for interrupts that are requesting service but not yet acknowledged.*
- ***ISR (In-Service Register):*** *Stores bits for interrupts currently being processed. Used to prevent lower priority interrupts from interrupting.*
- ***IMR (Interrupt Mask Register):*** *Stores masking bits. If a bit is set (1), that interrupt line is disabled (masked).*
- ***Priority Resolver:*** *Compares bits in IRR with ISR and IMR to determine if the new interrupt should be passed to the CPU.*

Explain Serial Communication in 8051. Detail the format of the SCON register. How is the Baud Rate determined?
Answer:

SCON Register (8-bit):

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

- ***SM0, SM1:*** *Mode Selection. (00: Shift Reg, 01: 8-bit UART, 10: 9-bit Fixed, 11: 9-bit Variable).*
- ***REN:*** *Receive Enable (Must be 1 to receive).*
- ***TB8/RB8:*** *9th data bit for modes 2 and 3.*
- ***TI:*** *Transmit Interrupt flag (Set when stop bit sent).*
- ***RI:*** *Receive Interrupt flag (Set when stop bit received).*

Baud Rate: *In Mode 1, Baud Rate is determined by Timer 1 overflow.*

$$Baud\ Rate = \frac{K \times F_{osc}}{32 \times 12 \times (256 - TH1)}$$

Where $K = 2^{SMOD}$. For 9600 baud (11.0592MHz), TH1 is loaded with `-3 (FDH)`*. The 8051 uses pins TXD (P3.1) and RXD (P3.0). It is controlled by the* ***SCON (Serial Control)*** *register.*

SCON Register (8-bit):

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

- ***SM0, SM1:*** *Mode Selection. (00: Shift Reg, 01: 8-bit UART, 10: 9-bit Fixed, 11: 9-bit Variable).*
- ***REN:*** *Receive Enable (Must be 1 to receive).*
- ***TB8/RB8:*** *9th data bit for modes 2 and 3.*
- ***TI:*** *Transmit Interrupt flag (Set when stop bit sent).*
- ***RI:*** *Receive Interrupt flag (Set when stop bit received).*

Baud Rate: *In Mode 1, Baud Rate is determined by Timer 1 overflow.*

$$Baud\ Rate = \frac{K \times F_{osc}}{32 \times 12 \times (256 - TH1)}$$

Where $K = 2^{SMOD}$. For 9600 baud (11.0592MHz), TH1 is loaded with `-3 (FDH)`*.*

Describe the Interrupt Structure of 8051. Explain the functions of the IE (Interrupt Enable) and IP (Interrupt Priority) registers.
Answer:

1. IE (Interrupt Enable) Register (A8H):

EA	-	*ET2*	*ES*	*ET1*	*EX1*	*ET0*	*EX0*

- ***EA:*** *Global Enable. If 0, all interrupts are disabled.*
- ***ES:*** *Serial Port Interrupt.*
- ***ET0/ET1:*** *Timer 0/1 Overflow Interrupts.*
- ***EX0/EX1:*** *External Interrupts ($\overline{INT0}, \overline{INT1}$).*

2. IP (Interrupt Priority) Register (B8H):

-	-	*PT2*	*PS*	*PT1*	*PX1*	*PT0*	*PX0*

Setting a bit to 1 assigns ***High Priority*** *to that interrupt. High priority interrupts can interrupt low priority handlers. The 8051 has 5 interrupt sources: External 0, Timer 0, External 1, Timer 1, and Serial Port.*

1. IE (Interrupt Enable) Register (A8H):

EA	-	*ET2*	*ES*	*ET1*	*EX1*	*ET0*	*EX0*

- ***EA:*** *Global Enable. If 0, all interrupts are disabled.*
- ***ES:*** *Serial Port Interrupt.*
- ***ET0/ET1:*** *Timer 0/1 Overflow Interrupts.*
- ***EX0/EX1:*** *External Interrupts (*$\overline{INT0}, \overline{INT1}$*).*

2. IP (Interrupt Priority) Register (B8H):

-	-	*PT2*	*PS*	*PT1*	*PX1*	*PT0*	*PX0*

Setting a bit to 1 assigns ***High Priority*** *to that interrupt. High priority interrupts can interrupt low priority handlers.*