DELHI SKILL AND ENTREPRENEURSHIP UNIVERSITY

End-Semester Examination

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 from Section A (Short Answer).
3. Attempt questions from Section B (Descriptive) as indicated.
4. Assume necessary data if missing.

SECTION A – Short Answer Questions

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

Q1. Identify the addressing modes of the following 8085 instructions:

(i) `MOV A, M`

(ii) `MVI B, 45H`

(iii) `LDA 2050H`

Answer: *(i)* ***Register Indirect Addressing Mode****: The address of the operand is specified by a register pair (H-L).*
(ii) ***Immediate Addressing Mode****: The 8-bit data (45H) is specified within the instruction itself.*
(iii) ***Direct Addressing Mode****: The 16-bit address (2050H) of the operand is given in the instruction.*

Q2. Explain the function of the ALE (Address Latch Enable) pin in 8085.

Answer: *ALE is a positive-going pulse generated every time the 8085 begins an operation (machine cycle). It indicates that the bits on AD7–AD0 are address bits. It is used to latch the low-order address from the multiplexed bus, separating the address from data.*

Q3. Differentiate between `PUSH` and `POP` instructions with respect to the Stack Pointer (SP).

Answer: *`PUSH`: Decrements the Stack Pointer (SP) by 2 and copies the contents of a register pair onto the stack top.*
`POP`: Copies the contents of the stack top to a register pair and increments the Stack Pointer (SP) by 2.

Q4. If `CS = 2000H` and `IP = 1234H`, calculate the physical address in 8086.

Answer: *The Physical Address in 8086 is calculated as:* $PhysicalAddress = (SegmentAddress \times 10H) + OffsetAddress$.
$CS \times 10H = 20000H$
$IP = 1234H$
$PhysicalAddress = 20000H + 1234H =$ **21234H**.

Q5. What is the function of the Instruction Queue in 8086?

Answer: *The Instruction Queue (6 bytes long) allows the Bus Interface Unit (BIU) to pre-fetch instructions while the Execution Unit (EU) executes the current instruction. This implementation of* ***pipelining*** *speeds up processing by reducing the wait time for instruction fetching.*

Q6. Explain the significance of the `TEST` pin in 8086.

Answer: *The `TEST` input is examined by the `WAIT` instruction. If the `TEST` pin is LOW, execution continues; if HIGH, the processor waits in an idle state. This is primarily used to synchronize the processor with an external coprocessor (like 8087).*

Q7. List the operating modes of the 8255 PPI.

Answer: *1.* ***BSR Mode****: Bit Set/Reset mode (for Port C only).*
2. ***I/O Mode****:*

- *Mode 0: Basic Input/Output (No handshaking).*
- *Mode 1: Input/Output with Handshaking.*
- *Mode 2: Bidirectional I/O with Handshaking (Port A only).*

Q8. Distinguish between memory-mapped I/O and I/O-mapped I/O.

Answer: *In* ***Memory-mapped I/O****, peripherals are treated as memory locations (using full 16/20-bit addresses) and accessed via memory instructions (MOV). In* ***I/O-mapped I/O****, peripherals have a separate 8-bit or 16-bit address space accessed via special instructions (IN, OUT).*

Q9. State three key differences between a Microprocessor and a Microcontroller.

Answer: *1.* ***Components****: Microprocessors have CPU only (external RAM/ROM needed); Microcontrollers have CPU, RAM, ROM, and I/O on-chip.*
2. ***Application****: Microprocessors are for general-purpose computing; Microcontrollers are for specific embedded control tasks.*
3. ***Cost/Power****: Microcontrollers are generally cheaper and consume less power than general-purpose microprocessors.*

Q10. What is the function of the PSW (Program Status Word) register in 8051?

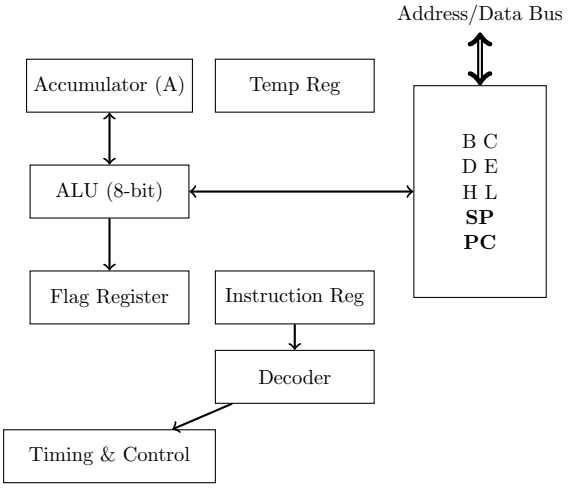
Answer: *The PSW is an 8-bit register that contains status flags (Carry, Auxiliary Carry, Overflow, Parity) and register bank select bits (RS0, RS1). It reflects the current state of the CPU and determines which of the four register banks (0-3) is currently active.*

SECTION B – Descriptive Questions

Attempt all questions. Include diagrams where necessary.

Q11. Draw the functional block diagram of the 8085 Microprocessor and explain the function of the Accumulator, PC, and SP.

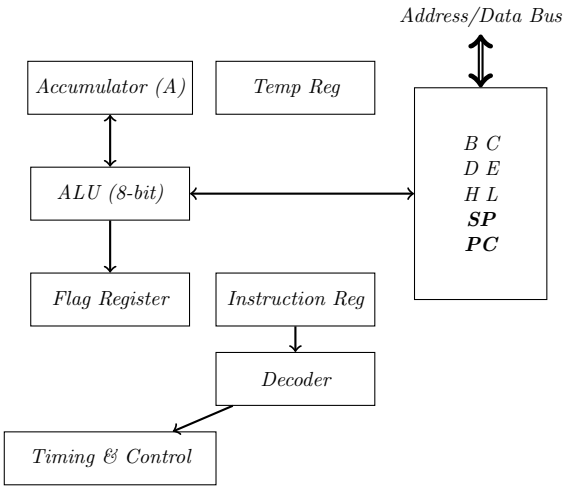
Answer:



1. Accumulator (A): It is an 8-bit register that is part of the ALU. It stores one of the operands for arithmetic and logical operations. The final result of an operation is also stored in the Accumulator.

2. Program Counter (PC): It is a 16-bit special-purpose register. It holds the memory address of the *next* instruction to be executed. The microprocessor increments the PC automatically after fetching an instruction byte.

3. Stack Pointer (SP): It is a 16-bit register used as a memory pointer. It points to the top location of the stack memory. It is decremented during `PUSH` operations and incremented during `POP` operations. *The 8085 is an 8-bit microprocessor with a 16-bit address bus. It consists of the ALU, Timing and Control Unit, Instruction Register, and General Purpose Registers.*



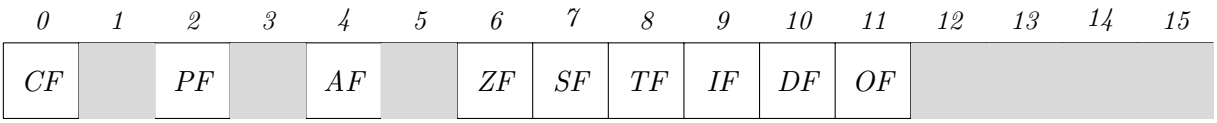
1. Accumulator (A): *It is an 8-bit register that is part of the ALU. It stores one of the operands for arithmetic and logical operations. The final result of an operation is also stored in the Accumulator.*

2. Program Counter (PC): *It is a 16-bit special-purpose register. It holds the memory address of the next instruction to be executed. The microprocessor increments the PC automatically after fetching an instruction byte.*

3. Stack Pointer (SP): *It is a 16-bit register used as a memory pointer. It points to the top location of the stack memory. It is decremented during* `PUSH` *operations and incremented during* `POP` *operations.*

Q12. Draw the Flag Register of 8086 and explain the function of the Conditional Flags.

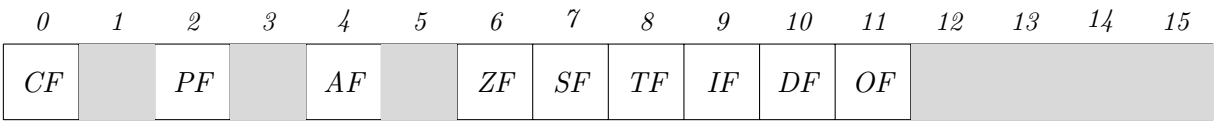
Answer:



Conditional Flags:

- ***CF (Carry Flag):*** *Set if there is a carry out of MSB or borrow into MSB.*
- ***PF (Parity Flag):*** *Set if the result has even parity (even number of 1s).*
- ***AF (Auxiliary Carry):*** *Set if carry generates from bit D3 to D4 (used in BCD).*
- ***ZF (Zero Flag):*** *Set if the result of an operation is zero.*
- ***SF (Sign Flag):*** *Set if the MSB of the result is 1 (negative).*
- ***OF (Overflow Flag):*** *Set if the signed result exceeds the capacity of the register.*

The 8086 has a 16-bit Flag Register containing 9 active flags: 6 Conditional (Status) flags and 3 Control flags.



Conditional Flags:

- ***CF (Carry Flag):*** *Set if there is a carry out of MSB or borrow into MSB.*
- ***PF (Parity Flag):*** *Set if the result has even parity (even number of 1s).*
- ***AF (Auxiliary Carry):*** *Set if carry generates from bit D3 to D4 (used in BCD).*
- ***ZF (Zero Flag):*** *Set if the result of an operation is zero.*
- ***SF (Sign Flag):*** *Set if the MSB of the result is 1 (negative).*
- ***OF (Overflow Flag):*** *Set if the signed result exceeds the capacity of the register.*

Q13. Explain the concept of Memory Banking in 8086. How are Even and Odd banks organized and accessed?

Answer:

Organization:

- ***Even Bank (Low Bank):*** *Contains even addresses (*$00000H, 00002H, \ldots$*). Connected to data lines* $D_0 - D_7$*. Enabled by* $A_0 = 0$*.*
- ***Odd Bank (High Bank):*** *Contains odd addresses (*$00001H, 00003H, \ldots$*). Connected to data lines* $D_8 - D_{15}$*. Enabled by* $\overline{BHE} = 0$*.*

Access Mechanism: *The processor uses the signals* $\overline{BHE}$ *(Bus High Enable) and* A_0 *to determine access:*

$\overline{\mathbf{BHE}}$	$\mathbf{A_0}$	**Operation**
0	0	Whole Word Access (16-bit) at Even Address
0	1	Byte Access (Odd Address/High Byte)
1	0	Byte Access (Even Address/Low Byte)
1	1	No Operation

This architecture allows 8086 to read a 16-bit word in ***one machine cycle*** *if the word starts at an even address (Aligned), but takes* ***two cycles*** *if it starts at an odd address (Unaligned). The 8086 has a 16-bit data bus but accesses memory which is byte-organized. To transfer 16 bits in a single cycle, the 1MB memory is divided into two banks of 512KB each: the* ***Even Bank*** *and the* ***Odd Bank****.*

Organization:

- ***Even Bank (Low Bank):*** *Contains even addresses (*$00000H, 00002H, \ldots$*). Connected to data lines* $D_0 - D_7$*. Enabled by* $A_0 = 0$*.*
- ***Odd Bank (High Bank):*** *Contains odd addresses (*$00001H, 00003H, \ldots$*). Connected to data lines* $D_8 - D_{15}$*. Enabled by* $\overline{BHE} = 0$*.*

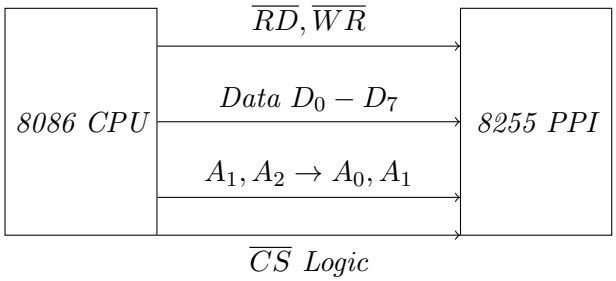
Access Mechanism: *The processor uses the signals* $\overline{BHE}$ *(Bus High Enable) and* A_0 *to determine access:*

$\overline{\mathbf{BHE}}$	$\mathbf{A_0}$	**Operation**
0	0	Whole Word Access (16-bit) at Even Address
0	1	Byte Access (Odd Address/High Byte)
1	0	Byte Access (Even Address/Low Byte)
1	1	No Operation

This architecture allows 8086 to read a 16-bit word in ***one machine cycle*** *if the word starts at an even address (Aligned), but takes* ***two cycles*** *if it starts at an odd address (Unaligned).*

Q14. Design an interface between 8086 and 8255 PPI. Explain the control word format for I/O Mode.

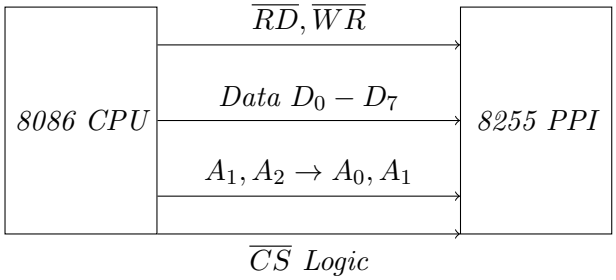
Answer:



Control Word Format (I/O Mode): *The Control Word Register (CWR) determines the mode of Ports A, B, and C.*

- ***D7 (1):*** *Must be 1 for I/O Mode.*
- ***D6, D5 (Mode Select A):*** *00=Mode 0, 01=Mode 1, 1x=Mode 2.*
- ***D4 (Port A):*** *1=Input, 0=Output.*
- ***D3 (Port C Upper):*** *1=Input, 0=Output.*
- ***D2 (Mode Select B):*** *0=Mode 0, 1=Mode 1.*
- ***D1 (Port B):*** *1=Input, 0=Output.*
- ***D0 (Port C Lower):*** *1=Input, 0=Output.*

8255 PPI Interface with 8086: *The 8255 is mapped to the lower order of the data bus ($D_0 - D_7$) for even address mapping or can be tailored. Typically, A_1 and A_2 of the CPU connect to A_0 and A_1 of the 8255 to select ports.*



Control Word Format (I/O Mode): *The Control Word Register (CWR) determines the mode of Ports A, B, and C.*

- ***D7 (1):*** *Must be 1 for I/O Mode.*
- ***D6, D5 (Mode Select A):*** *00=Mode 0, 01=Mode 1, 1x=Mode 2.*
- ***D4 (Port A):*** *1=Input, 0=Output.*
- ***D3 (Port C Upper):*** *1=Input, 0=Output.*
- ***D2 (Mode Select B):*** *0=Mode 0, 1=Mode 1.*
- ***D1 (Port B):*** *1=Input, 0=Output.*
- ***D0 (Port C Lower):*** *1=Input, 0=Output.*

Q15. Explain the principle of interfacing a Stepper Motor with a microprocessor.

Answer: *A stepper motor converts digital pulses into mechanical shaft rotation.*

(a) ***Driver Circuit:*** *The microprocessor cannot drive the motor coils directly due to current requirements. A driver IC (like ULN2003) or transistor array is used.*

(b) ***Sequence Generation:*** *To rotate the motor, the coils (windings) must be energized in a specific sequence (e.g., A-B-C-D for full step).*

(c) ***Speed Control:*** *The speed is controlled by the delay between the steps sent by the program.*

(d) ***Interfacing:*** *Usually connected via 8255 Port A or B. Writing a sequence like $33H, 66H, CCH, 99H$ rotates the motor.*

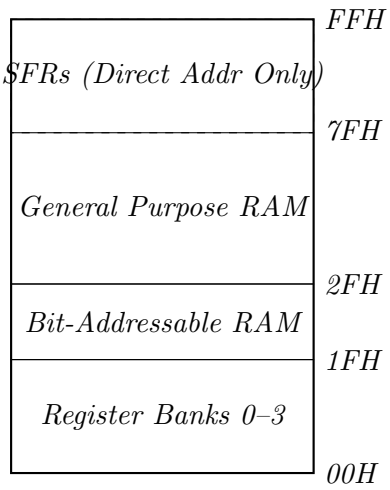
Q16. Describe the internal memory organization of the 8051 Microcontroller (RAM and ROM).

Answer:

1. Internal ROM (Program Memory):

- *Standard 8051 has 4KB of on-chip ROM.*
- *Address range:* $0000H$ *to* $0FFFH$.
- *If the* $\overline{EA}$ *(External Access) pin is High, the CPU executes from internal ROM. If Low, it fetches from external ROM.*

2. Internal RAM (Data Memory): *The 8051 has 128 bytes of internal RAM (Address* $00H$ *to* $7FH$*), divided into three sections:*



(a) ***Register Banks (***$00H - 1FH$***):*** *Four banks (0-3), each having 8 registers (R0-R7).*

(b) ***Bit-Addressable Area (***$20H - 2FH$***):*** *16 bytes where individual bits can be accessed using instructions like* `SETB`, `CLR`.

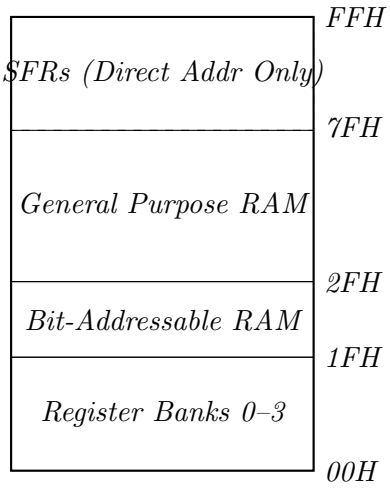
(c) ***Scratch Pad RAM (***$30H - 7FH$***):*** *General storage for data and stack.*

The 8051 has a Harvard architecture (separate code and data memory).

1. Internal ROM (Program Memory):

- *Standard 8051 has 4KB of on-chip ROM.*
- *Address range:* $0000H$ *to* $0FFFH$.
- *If the* $\overline{EA}$ *(External Access) pin is High, the CPU executes from internal ROM. If Low, it fetches from external ROM.*

2. Internal RAM (Data Memory): *The 8051 has 128 bytes of internal RAM (Address* $00H$ *to* $7FH$*), divided into three sections:*



(a) ***Register Banks*** *(*$00H - 1FH$*):* *Four banks (0-3), each having 8 registers (R0-R7).*

(b) ***Bit-Addressable Area*** *(*$20H - 2FH$*):* *16 bytes where individual bits can be accessed using instructions like* `SETB`*,* `CLR`*.*

(c) ***Scratch Pad RAM*** *(*$30H - 7FH$*):* *General storage for data and stack.*

Q17. Explain the TMOD register in 8051. Write an assembly program to generate a square wave of 50% duty cycle on Pin P1.5 using Timer 0 in Mode 1.

Answer:

Assembly Program (Square Wave on P1.5):

```
HERE: MOV TL0, 0F2H ; Load Low byte (Example values for delay) MOV TH0, 0FFH
; Load High byte CPL P1.5 ; Toggle Port 1.5 ACALL DELAY ; Call delay routine
SJMP HERE ; Repeat
DELAY: SETB TR0 ; Start Timer 0 WAIT: JNB TF0, WAIT ; Wait for Overflow Flag
(TF0) CLR TR0 ; Stop Timer CLR TF0 ; Clear Flag RET
```

TMOD (Timer Mode) Register: *It is an 8-bit register controlling Timer 0 and Timer 1.*

- *Upper Nibble (Timer 1), Lower Nibble (Timer 0).*
- *Bits:* ***GATE*** *(External control),* ***C/$\overline{T}$*** *(Counter/Timer selector),* ***M1, M0*** *(Mode bits).*
- *Modes: 00 (13-bit), 01 (16-bit), 10 (8-bit Auto-reload), 11 (Split).*

Assembly Program (Square Wave on P1.5):

```
HERE: MOV TL0, 0F2H ; Load Low byte (Example values for delay) MOV TH0, 0FFH
; Load High byte CPL P1.5 ; Toggle Port 1.5 ACALL DELAY ; Call delay routine
SJMP HERE ; Repeat
DELAY: SETB TR0 ; Start Timer 0 WAIT: JNB TF0, WAIT ; Wait for Overflow Flag
(TF0) CLR TR0 ; Stop Timer CLR TF0 ; Clear Flag RET
```

Q18. Compare 8085 and 8086 microprocessors based on:

(a) *Data Bus Width*

(b) Address Bus Width

(c) Pipelining

(d) Memory Capacity

(e) Operating Frequency (Typical)

Answer:

Feature	***8085***	***8086***
Data Bus	*8-bit*	*16-bit*
Address Bus	*16-bit*	*20-bit*
Pipelining	*Not Supported*	*Supported (Instruction Queue)*
Memory	*64 KB*	*1 MB*
Frequency	*3 MHz*	*5 MHz, 8 MHz, 10 MHz*