

## Question 1(a) [3 marks]

Compare RISC and CISC.

Answer:

Feature	RISC	CISC
Instructions	Simple, fixed-length	Complex, variable-length
Execution	Single cycle	Multiple cycles
Addressing modes	Few	Many
Registers	Many	Few
Design focus	Hardware simplicity	Code density

**Mnemonic:** "RISCs Complete Instructions Simply"

## Question 1(b) [4 marks]

Compare Von-Neumann and Harvard architecture.

Answer:

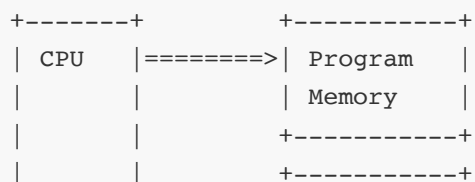
Feature	Von-Neumann	Harvard
Memory	Single shared memory	Separate program & data memory
Bus	Single bus for data & instructions	Separate buses
Speed	Slower (memory bottleneck)	Faster (parallel access)
Complexity	Simpler design	More complex
Applications	General computing	Real-time systems

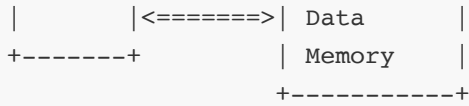
**Diagram:**

Von-Neumann:



Harvard:





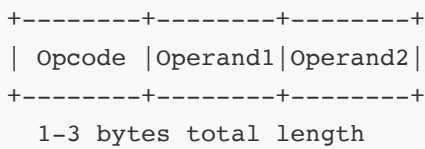
**Mnemonic:** "Harvard Has Separate Spaces"

## Question 1(c) [7 marks]

**Explain: 8085 Instruction Format, Control Unit, Machine Cycle, ALU**

**Answer:**

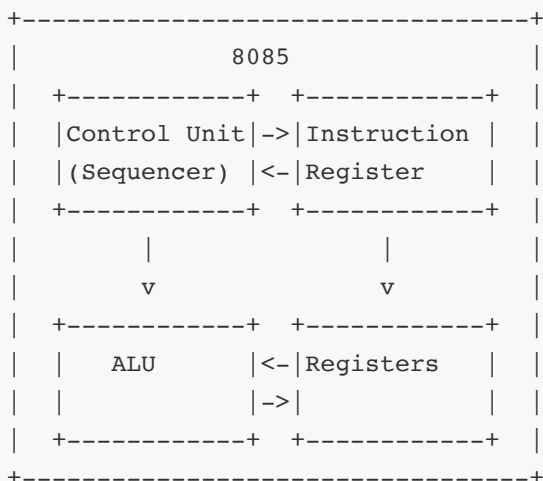
**Instruction Format:**



Component	Function
<b>Instruction Format</b>	1-3 byte structure with opcode and operands
<b>Control Unit</b>	Fetches, decodes instructions; generates signals
<b>Machine Cycle</b>	Basic operation cycle (T-states)
<b>ALU</b>	Performs arithmetic and logical operations

- **Instruction Format:** Contains opcode (3-8 bits) and 0-2 operands
- **Control Unit:** Heart of processor that coordinates all operations
- **Machine Cycle:** Consists of fetch, decode, execute phases
- **ALU:** Handles ADD/SUB/AND/OR/XOR operations on data

**Diagram:**



**Mnemonic:** "CIMA: Control Interprets, Machine Acts"

## Question 1(c OR) [7 marks]

**Compare Microprocessor and Microcontroller.**

**Answer:**

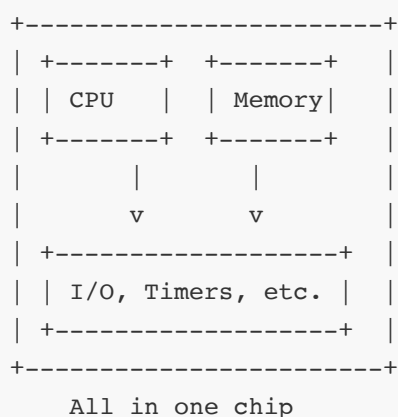
Feature	Microprocessor	Microcontroller
Design	CPU only	CPU + peripherals
Memory	External	Internal (RAM/ROM)
I/O ports	Limited	Many built-in
Cost	Higher	Lower
Applications	General computing	Embedded systems
Power consumption	Higher	Lower
Example	Intel 8085/8086	Intel 8051

**Diagram:**

Microprocessor System:



Microcontroller:



**Mnemonic:** "Micro-P Processes, Micro-C Controls"

## Question 2(a) [3 marks]

**Explain Instruction Fetching, Decoding and Execution Operation in microprocessor.**

**Answer:**

Phase	Operation
Fetching	CPU gets instruction from memory using PC
Decoding	Determines operation type and operands
Execution	Performs the actual operation

**Diagram:**

```

+-----+   +-----+   +-----+
| Fetch |--->| Decode |--->| Execute |
+-----+   +-----+   +-----+

```

**Mnemonic:** "FDE: First Get, Then Understand, Finally Do"

## Question 2(b) [4 marks]

**Explain Bus Organization of 8085 microprocessor.**

**Answer:**

Bus Type	Width	Function
Address Bus	16-bit	Carries memory addresses (A0-A15)
Data Bus	8-bit	Transfers data (D0-D7)
Control Bus	Various lines	Manages data flow (RD, WR, IO/M)
Multiplexed	AD0-AD7	Lower address bits + data bits

**Diagram:**

```

8085 Microprocessor
|
|---- Address Bus (16-bit) ----> Memory
|                                Location
|---- Data Bus (8-bit) <-----> Data
|
|---- Control Bus -----> Control
|                          Signals

```

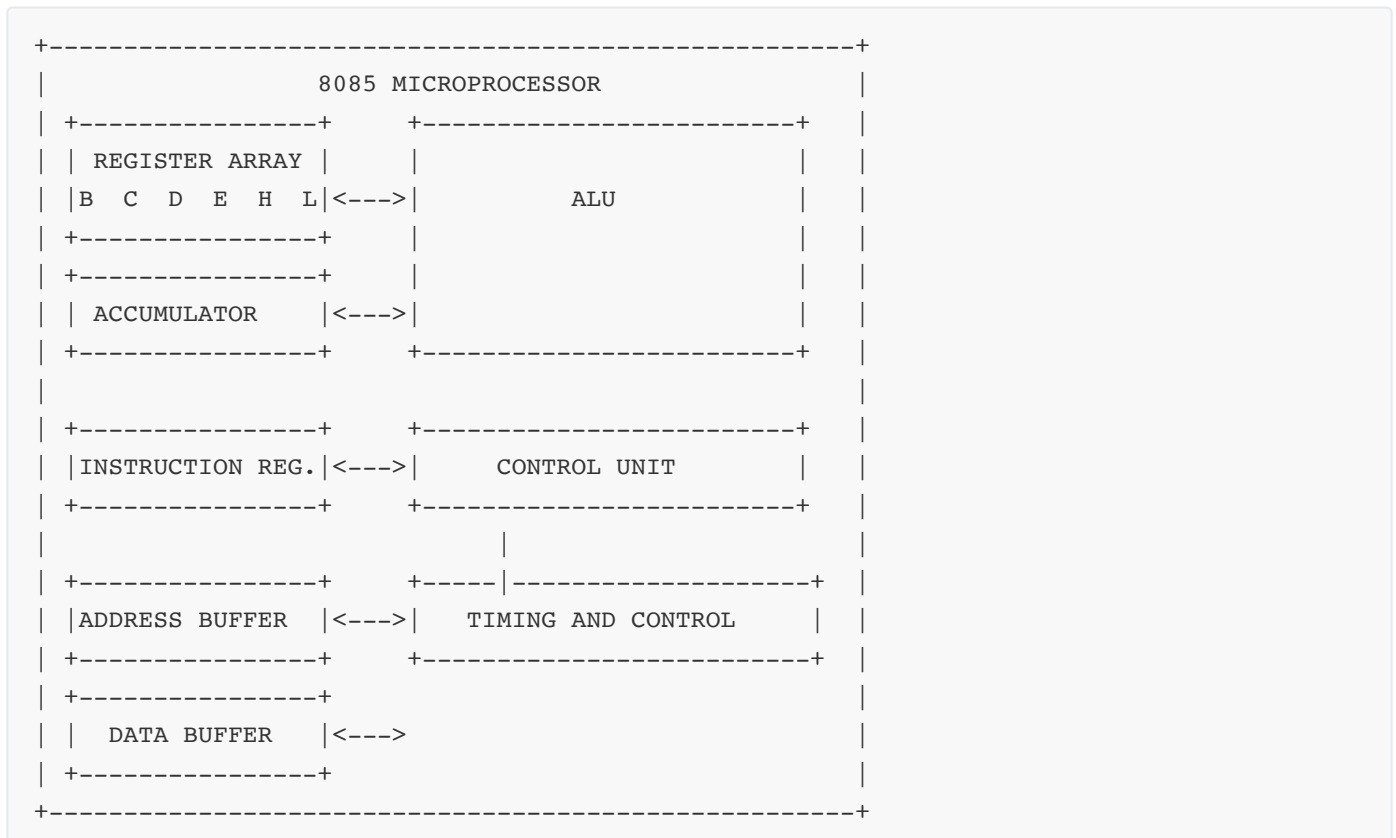
**Mnemonic:** "ADC: Address points, Data flows, Control directs"

## Question 2(c) [7 marks]

**Describe architecture of 8085 microprocessor with the help of neat diagram.**

**Answer:**

Component	Function
ALU	Arithmetic & logical operations
Register Array	Temporary data storage (B,C,D,E,H,L)
Accumulator	Main register for arithmetic
Control Unit	Instruction control & timing
Instruction Register	Holds current instruction
Timing & Control	Generates timing signals
Address Buffer	Manages address bus
Data Buffer	Handles data bus transfers

**Diagram:**

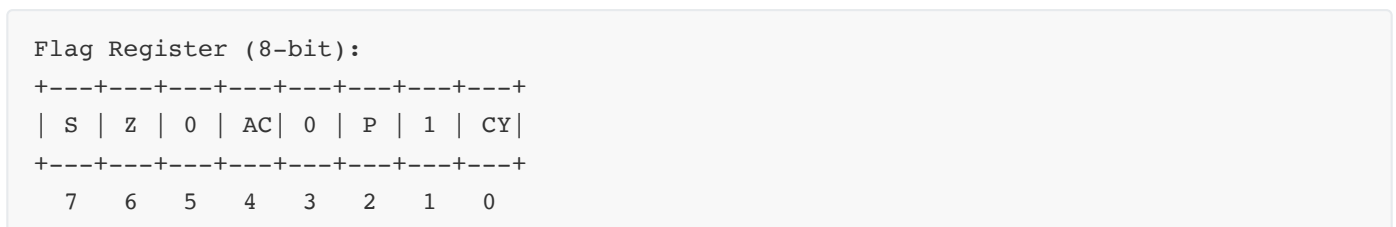
- **ALU:** Performs arithmetic & logical operations
- **Control Unit:** Fetches & decodes instructions
- **Registers:** Store data temporarily during processing
- **Buses:** Carry address, data and control signals

**Mnemonic:** "AR CBD: Architecture Registers Control Buses Data"

## Question 2(a OR) [3 marks]

**Explain De-multiplexing of Address and Data buses for 8085 Microprocessor.****Answer:**

Step	Action
1	ALE signal goes high
2	Lower address (A0-A7) appears on AD0-AD7
3	Latch captures address using ALE
4	ALE goes low, AD0-AD7 now carries data

**Diagram:****Mnemonic:** "ALAD: ALE Latches Address before Data"**Question 2(b OR) [4 marks]****Draw Flag Register of 8085 microprocessor & explain it.****Answer:**

Flag	Name	Set when
S	Sign	Bit 7 of result is 1 (negative)
Z	Zero	Result is zero
AC	Auxiliary Carry	Carry from bit 3 to bit 4
P	Parity	Result has even parity (even 1s)
CY	Carry	Carry generated from bit 7

**Mnemonic:** "SuZie AC's Perfect CarrY"

## Question 2(c OR) [7 marks]

Describe Pin diagram of 8085 microprocessor with the help of neat diagram.

Answer:

Pin Group	Function
Address/Data	Multiplexed AD0-AD7, A8-A15
Control	RD, WR, IO/M, S0, S1, ALE, CLK
Interrupts	INTR, RST 5.5-7.5, TRAP
DMA	HOLD, HLDA
Power	Vcc, Vss
Serial I/O	SID, SOD
Reset	RESET IN, RESET OUT

Diagram:

```

      +-----+
X1  -->| 1    40 |<-- Vcc
X2  -->| 2    39 |<-- HOLD
RESET OUT-->| 3    38 |<-- HLDA
RESET IN  -->| 4    37 |<-- CLK
IO/M  -->| 5    36 |<-- RESET IN
S1    -->| 6    35 |<-- READY
RD    -->| 7    34 |<-- IO/M
WR    -->| 8    33 |<-- S1
ALE   -->| 9    32 |<-- RD
S0    -->| 10   31 |<-- WR
A15   -->| 11   30 |<-- ALE
A14   -->| 12   29 |<-- S0
A13   -->| 13   28 |<-- A15
A12   -->| 14   27 |<-- A14
A11   -->| 15   26 |<-- A13
A10   -->| 16   25 |<-- A12
A9    -->| 17   24 |<-- A11
A8    -->| 18   23 |<-- A10
AD7   -->| 19   22 |<-- A9
AD6   -->| 20   21 |<-- A8
      +-----+

```

- **Address/Data Pins:** Multiplexed pins save physical pins
- **Control Pins:** Coordinate memory and I/O operations
- **Interrupt Pins:** Allow external device interrupts

- **Serial Pins:** Provide basic serial communication

**Mnemonic:** "ACID-PS: Address-Control-Interrupt-DMA-Power-Serial"

## Question 3(a) [3 marks]

**Explain Stack, Stack Pointer and Stack operation.**

**Answer:**

Term	Description
Stack	LIFO memory area for temporary data storage
Stack Pointer	16-bit register that points to stack top
Operations	PUSH (store), POP (retrieve)

**Diagram:**

```

Memory:      Stack Pointer:
+-----+    +-----+
|         |<----| SP  |
+-----+    +-----+
| Data |
+-----+    PUSH: SP--, M[SP]=data
| Data |    POP:  data=M[SP], SP++
+-----+

```

**Mnemonic:** "SP Points to LIFO Lane"

## Question 3(b) [4 marks]

**Draw Pin diagram of 8051 microcontroller.**

**Answer:**

```

      8051 Microcontroller
      +-----+
P1.0--| 1          40 |--VCC
P1.1--| 2          39 |--P0.0/AD0
P1.2--| 3          38 |--P0.1/AD1
P1.3--| 4          37 |--P0.2/AD2
P1.4--| 5          36 |--P0.3/AD3
P1.5--| 6          35 |--P0.4/AD4
P1.6--| 7          34 |--P0.5/AD5
P1.7--| 8          33 |--P0.6/AD6
RST  --| 9          32 |--P0.7/AD7
P3.0/RXD| 10        31 |--EA/VPP
P3.1/TXD| 11        30 |--ALE/PROG
P3.2/INT0| 12        29 |--PSEN

```



P3.3/INT1	13	28	--P2.7/A15
P3.4/T0	14	27	--P2.6/A14
P3.5/T1	15	26	--P2.5/A13
P3.6/WR	16	25	--P2.4/A12
P3.7/RD	17	24	--P2.3/A11
XTAL2 --	18	23	--P2.2/A10
XTAL1 --	19	22	--P2.1/A9
VSS --	20	21	--P2.0/A8
+-----+			

Pin Group	Function
P0	Port 0, multiplexed with address/data
P1	Port 1, general purpose I/O
P2	Port 2, upper address and I/O
P3	Port 3, special functions and I/O

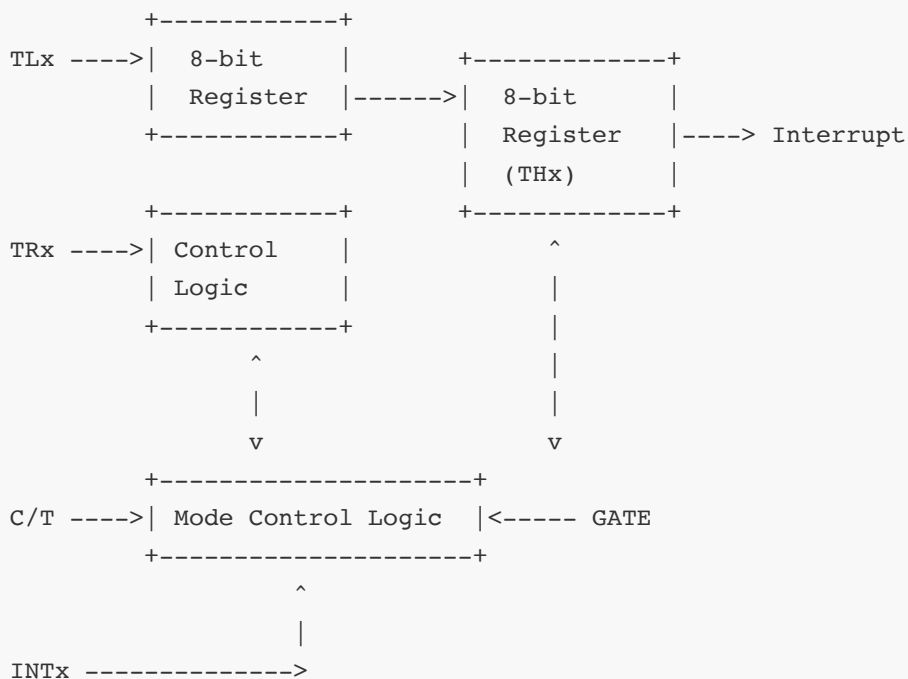
**Mnemonic:** "PORT 0123: Data-General-Address-Special"

### Question 3(c) [7 marks]

**Draw Timers/Counters logic diagram of 8051 microcontroller and explain its operation in various modes.**

**Answer:**

### Timer/Counter Diagram:



Mode	Operation
Mode 0	13-bit timer (5-bit TL, 8-bit TH)
Mode 1	16-bit timer (8-bit TL, 8-bit TH)
Mode 2	8-bit auto-reload (TL counts, TH reloads)
Mode 3	Split timer (Timer 0 only)

- **Timer:** Uses internal clock, counts machine cycles
- **Counter:** Uses external input, counts external events
- **Control Bits:** TMOD register sets mode, TCON controls operation
- **Modes:** Different configurations for different timing needs

**Mnemonic:** "MARC: Mode Auto-Reload Count"

## Question 3(a OR) [3 marks]

**List Common features of Microcontrollers.**

**Answer:**

Feature	Purpose
CPU Core	Process instructions
Memory (RAM/ROM)	Store program and data
I/O Ports	Interface with external devices
Timers/Counters	Measure time intervals
Interrupts	Handle asynchronous events
Serial Communication	Transfer data with other devices

**Mnemonic:** "CRITICS: CPU ROM I/O Timers Interrupts Comm Serial"

## Question 3(b OR) [4 marks]

**Explain Internal RAM Organization of 8051 microcontroller.**

**Answer:**

RAM Area	Address Range	Usage
Register Banks	00H-1FH	R0-R7 (4 banks)
Bit-addressable	20H-2FH	128 bits (0-7FH)
Scratch Pad	30H-7FH	General purpose
SFRs	80H-FFH	Control registers

**Diagram:**

```

8051 Internal RAM (128 bytes):
+-----+ 00H
| Register Bank 0 |
+-----+ 08H
| Register Bank 1 |
+-----+ 10H
| Register Bank 2 |
+-----+ 18H
| Register Bank 3 |
+-----+ 20H
| Bit-addressable |
+-----+ 30H
|               |
| Scratch Pad    |
|               |
+-----+ 80H

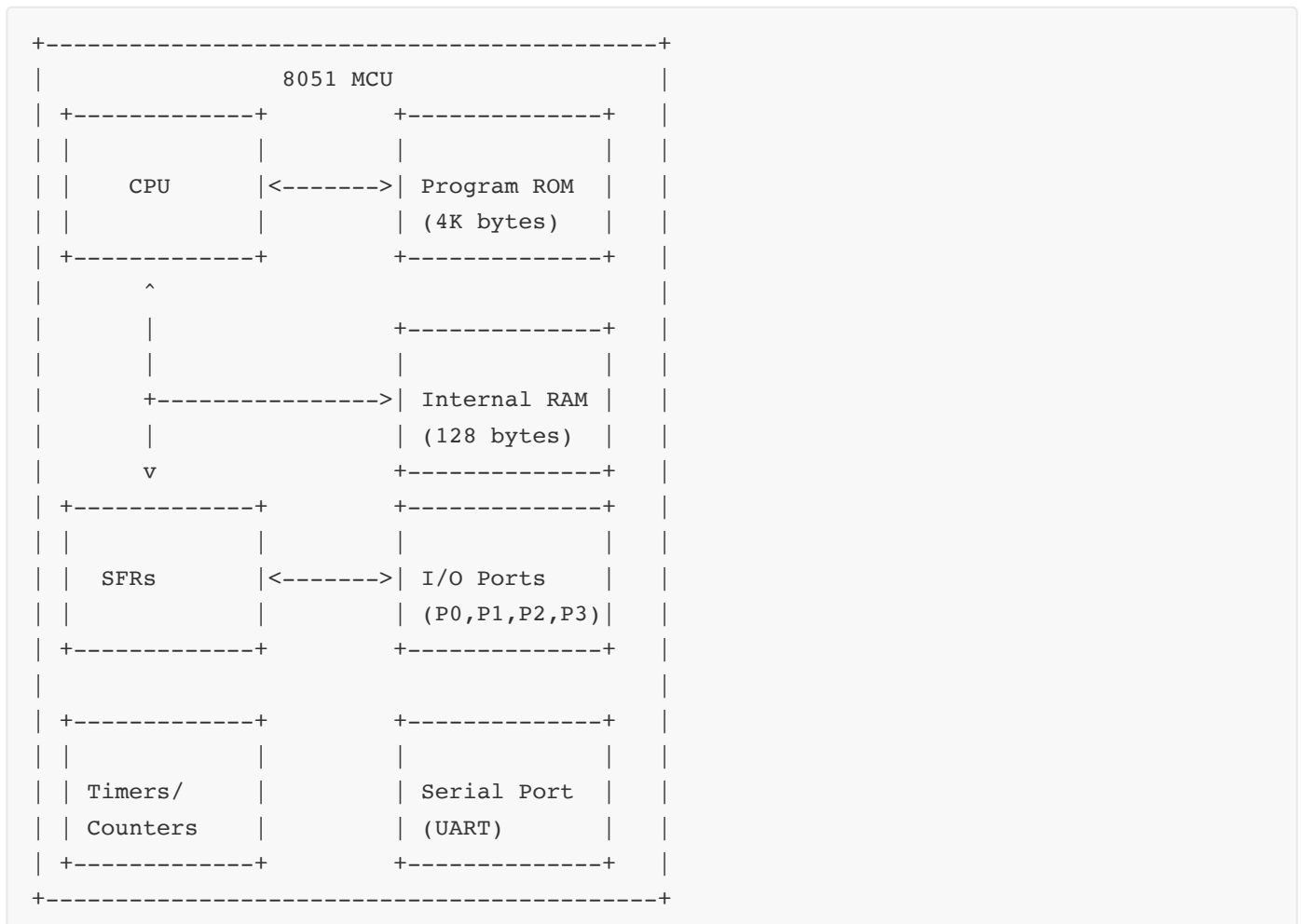
```

**Mnemonic:** "RBBS: Registers Bits Buffer Scratch"

## Question 3(c OR) [7 marks]

**Explain architecture of 8051 microcontroller with the help of neat diagram.****Answer:**

Component	Function
CPU	8-bit processor with ALU
Memory	4K ROM, 128 bytes RAM
I/O Ports	Four 8-bit ports (P0-P3)
Timers	Two 16-bit timers/counters
Serial Port	Full-duplex UART
Interrupts	Five interrupt sources
Special Function Registers	Control registers

**Diagram:**

- **Harvard Architecture:** Separate program and data memory
- **CISC Design:** Rich instruction set (over 100 instructions)
- **In-built Peripherals:** No need for external components
- **Single-chip Solution:** Complete system on one chip

**Mnemonic:** "CAPITALS: CPU Architecture Ports I/O Timer ALU LS-Interface Serial"

## Question 4(a) [3 marks]

Write an 8051 Assembly Language Program to Copy the data from external RAM Location 0123h to TL0 and Data from external RAM location 0234h to TH0.

**Answer:**

```

MOV  DPTR, #0123H    ; Load DPTR with source address 0123H
MOVB A, @DPTR        ; Read data from external RAM
MOV  TL0, A          ; Copy to Timer 0 low byte

MOV  DPTR, #0234H    ; Load DPTR with source address 0234H
MOVB A, @DPTR        ; Read data from external RAM
MOV  TH0, A          ; Copy to Timer 0 high byte

```

**Key Steps:**

- Use DPTR to address external RAM
- MOVX instruction for external memory access
- Direct transfer to timer registers

**Mnemonic:** "DRAM: DPTR Read Address Move"

## Question 4(b) [4 marks]

---

**Write an 8051 Assembly Language Program to blink LED interfaced at port P1.3 at time interval of 1ms.**

**Answer:**

```

AGAIN:  SETB P1.3      ; Turn ON LED at P1.3
        ACALL DELAY    ; Call delay subroutine
        CLR  P1.3      ; Turn OFF LED at P1.3
        ACALL DELAY    ; Call delay subroutine
        SJMP AGAIN     ; Repeat forever

DELAY:  MOV  R7, #250   ; Load R7 for outer loop
OUTER:  MOV  R6, #1     ; Load R6 for inner loop
INNER:  DJNZ R6, INNER  ; Decrement R6 until zero
        DJNZ R7, OUTER  ; Decrement R7 until zero
        RET            ; Return from subroutine

```

**Key Steps:**

- Toggle P1.3 pin to blink LED
- Nested delay loop for timing
- Infinite loop for continuous blinking

**Mnemonic:** "STACI: Set-Timer-And-Clear-Infinitely"

## Question 4(c) [7 marks]

---

**List Addressing Modes of 8051 Microcontroller and explain all of them with the help of example.**

**Answer:**

Addressing Mode	Example	Description
Immediate	MOV A, #25H	Data is in instruction
Register	MOV A, R0	Data is in register
Direct	MOV A, 30H	Data is at RAM address
Indirect	MOV A, @R0	R0/R1 contains address
Indexed	MOVC A, @A+DPTR	Access program memory
Bit	SETB P1.3	Access individual bits
Relative	SJMP LABEL	Jumps with 8-bit offset

**Examples:**

- **Immediate:** MOV A, #55H (Load A with 55H)
- **Register:** ADD A, R3 (Add R3 to A)
- **Direct:** MOV 40H, A (Store A at address 40H)
- **Indirect:** MOV @R0, #5 (Store 5 at address in R0)
- **Indexed:** MOVC A, @A+DPTR (Read code memory)
- **Bit:** CLR C (Clear carry flag)
- **Relative:** JZ LOOP (Jump if A is zero)

**Mnemonic:** "I'M DIRBI: Immediate Register Direct Bit Indexed"

## Question 4(a OR) [3 marks]

Write an 8051 Assembly Language Program to Subtract the content of RAM location 11h from RAM location 14h; put result in RAM location 3Ch.

**Answer:**

```

MOV  A, 14H      ; Load content of RAM location 14H to A
CLR  C           ; Clear carry flag
SUBB A, 11H      ; Subtract content of 11H with borrow
MOV  3CH, A      ; Store result in RAM location 3CH

```

**Key Steps:**

- Load minuend into accumulator
- Clear carry for correct subtraction
- Use SUBB for subtraction with borrow
- Store result in destination

**Mnemonic:** "LCSS: Load-Clear-Subtract-Store"

## Question 4(b OR) [4 marks]

Write an 8051 Assembly Language Program to generate a square wave of 50% duty cycle on bit 3 of Port 1 using Timer 0 in Mode 1.

Answer:

```

MOV  TMOD, #01H    ; Timer 0, Mode 1 (16-bit)
AGAIN: MOV  TH0, #0FCH ; Load high byte
MOV  TL0, #18H     ; Load low byte (-1000 in 16-bit)
SETB TR0          ; Start timer
JNB  TF0, $        ; Wait for overflow
CLR  TR0          ; Stop timer
CLR  TF0          ; Clear timer flag
CPL  P1.3         ; Toggle P1.3
SJMP AGAIN        ; Repeat

```

Key Steps:

- Configure Timer 0 in Mode 1
- Preload timer with value for 1ms delay
- Wait for timer overflow
- Toggle output bit for square wave

Mnemonic: "MSTCCS: Mode-Set-Timer-Check-Clear-Switch"

## Question 4(c OR) [7 marks]

Explain any seven Logical Instructions with example for 8051 Microcontroller.

Answer:

Instruction	Example	Operation
ANL	ANL A, #3FH	Logical AND
ORL	ORL P1, #80H	Logical OR
XRL	XRL A, R0	Logical XOR
CLR	CLR A	Clear (set to 0)
CPL	CPL P1.0	Complement (invert)
RL	RL A	Rotate left
RR	RR A	Rotate right

Examples:

- **ANL:** `ANL A, #0FH` (A = A AND 0FH, masks high nibble)

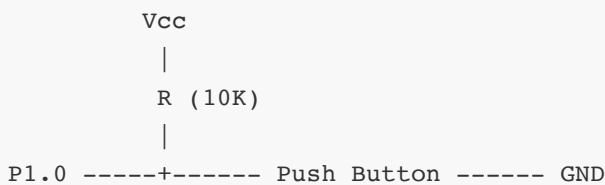
- **ORL:** ORL 20H, A (20H = 20H OR A, sets bits)
- **XRL:** XRL A, #55H (A = A XOR 55H, toggles bits)
- **CLR:** CLR C (Clear carry flag, C = 0)
- **CPL:** CPL A (Complement A, A = NOT A)
- **RL:** RL A (Rotate A left one bit)
- **RR:** RR A (Rotate A right one bit)

**Mnemonic:** "A-OX-CCR: AND OR XOR Clear Complement Rotate"

## Question 5(a) [3 marks]

**Draw Interfacing of Push button Switch with 8051 microcontroller.**

**Answer:**



Component	Connection
Push Button	Between P1.0 and GND
Pull-up Resistor	10K between P1.0 and VCC
Port Pin	P1.0 configured as input

**Key Points:**

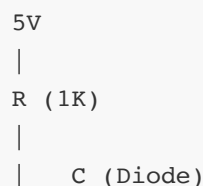
- Active-low configuration (button press gives 0)
- Pull-up resistor prevents floating input
- Can connect to any I/O pin

**Mnemonic:** "PIP: Pull-up-Input-Press"

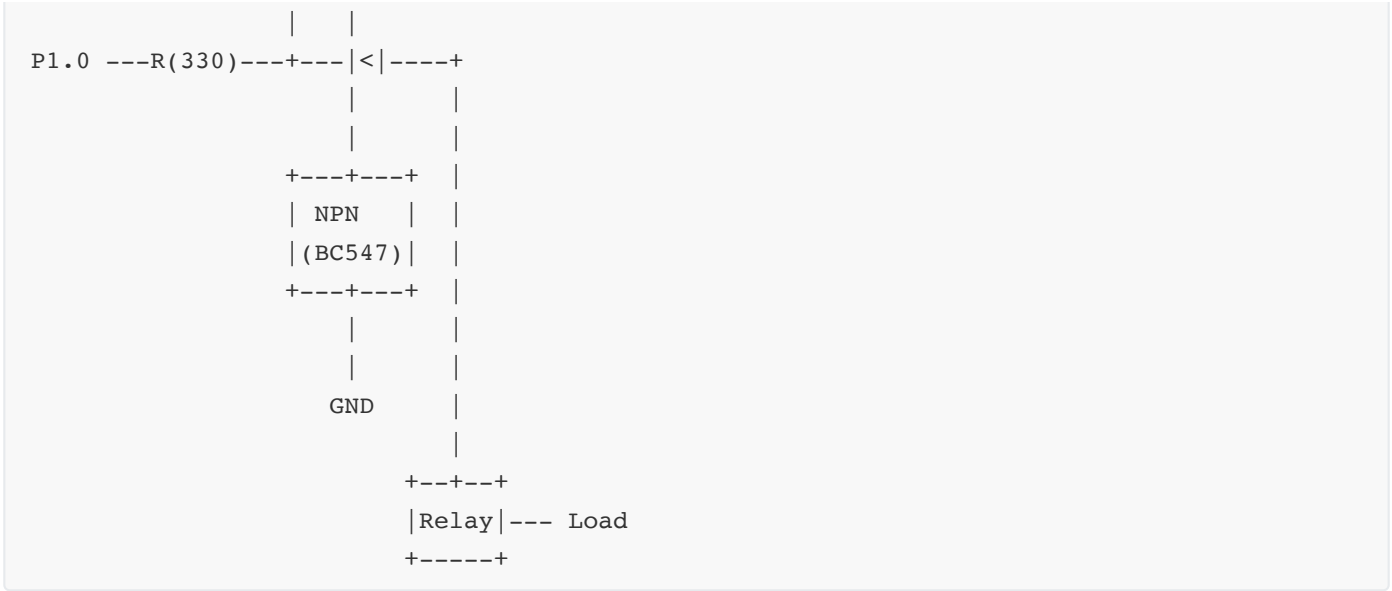
## Question 5(b) [4 marks]

**Interface Relay with 8051 microcontroller.**

**Answer:**







Component	Purpose
NPN Transistor	Current amplification
Diode	Back EMF protection
Resistors	Current limiting
Relay	High-power switching

Key Steps:

- Port pin drives transistor base
- Transistor switches relay coil
- Diode protects against back EMF
- Relay contacts switch high-power load

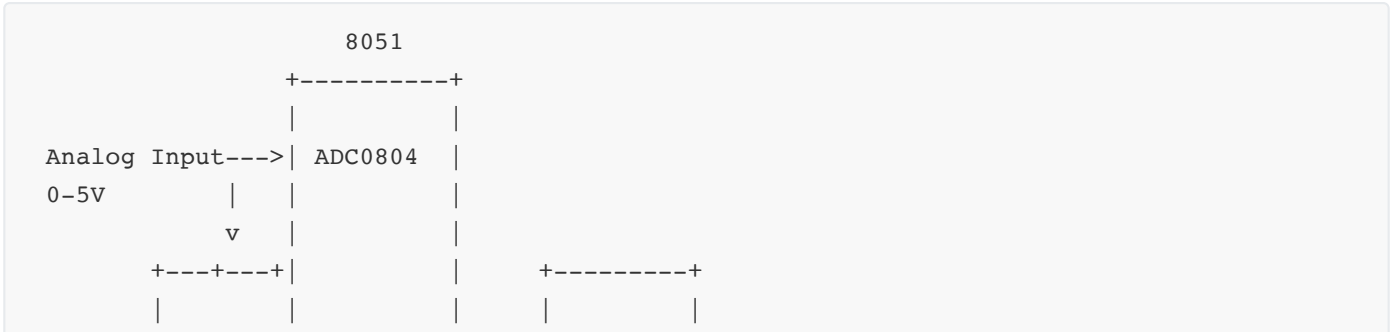
Mnemonic: "TRIP: Transistor-Relay-Interface-Protection"

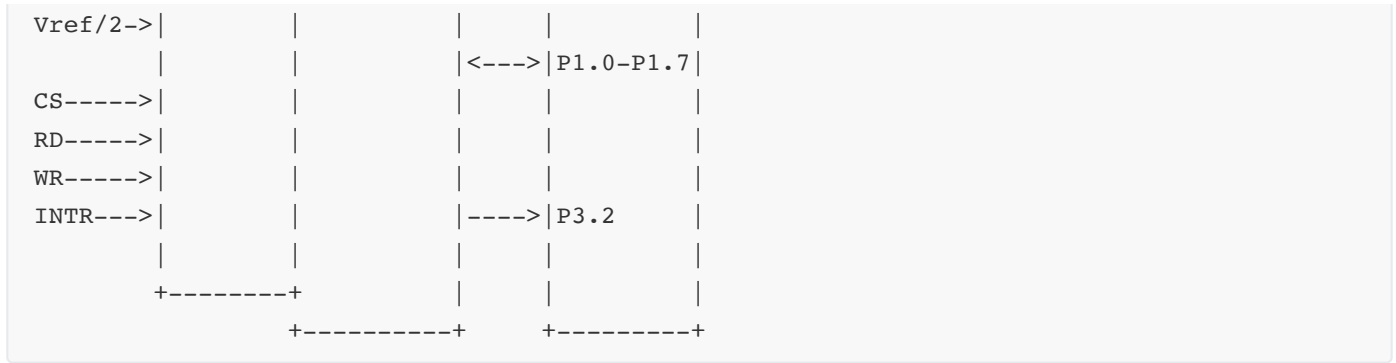
Question 5(c) [7 marks]

Interface ADC0804 with 8051 microcontroller.

Answer:

Circuit Diagram:





Connection	8051 Pin	ADC0804 Pin
Data Bus	P1.0-P1.7	D0-D7
CS	P3.0	CS
RD	P3.1	RD
WR	P3.2	WR
INTR	P3.3	INTR

- **ADC0804:** 8-bit A/D converter with 0-5V input range
- **Interface:** Connect data pins to Port 1, control to Port 3
- **Operation:** Write to ADC to start conversion, wait for INTR, read result
- **Resolution:** 8-bit (256 steps) for 0-5V gives ~19.5mV per step

**Mnemonic:** "CRIW: Control-Read-Interrupt-Write"

## Question 5(a OR) [3 marks]

**List Applications of microcontroller in various fields.**

**Answer:**

Field	Applications
Industrial	Motor control, automation, PLCs
Medical	Patient monitoring, diagnostic equipment
Consumer	Washing machines, microwaves, toys
Automotive	Engine control, ABS, airbag systems
Communication	Mobile phones, modems, routers
Security	Access control, alarm systems

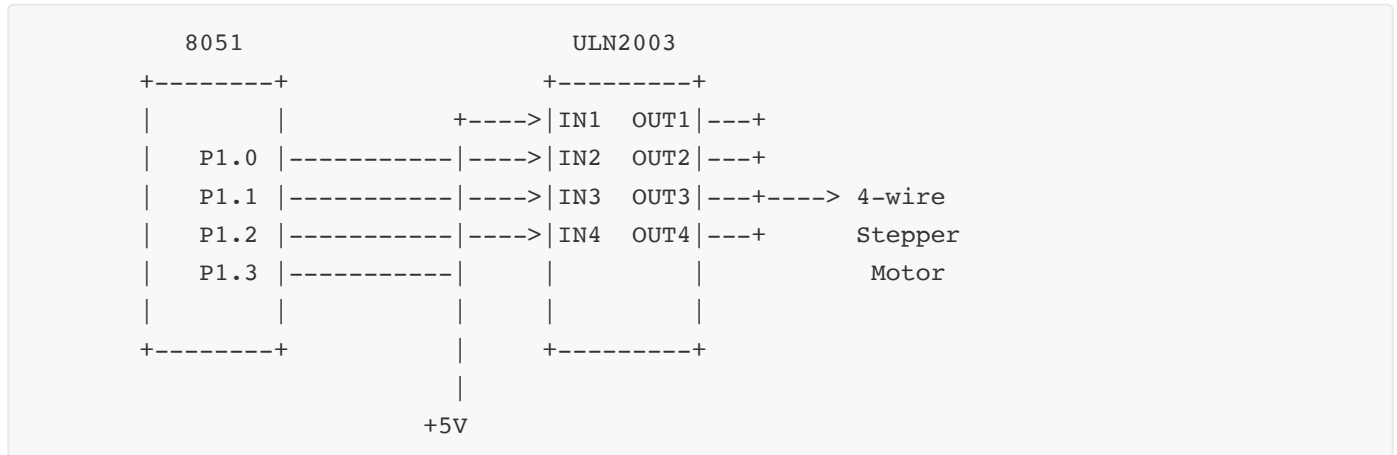
**Mnemonic:** "I-MACS: Industrial-Medical-Automotive-Consumer-Security"

## Question 5(b OR) [4 marks]

Interface Stepper motor with 8051 microcontroller.

Answer:

Circuit Diagram:



Component	Purpose
ULN2003	Driver IC with Darlington arrays
Port pins	P1.0-P1.3 for 4 motor phases
Power supply	Separate supply for motor

Code Structure:

```

; Clockwise Rotation Sequence
STEP_SEQ: DB 0000_1000B ; Step 1
          DB 0000_1100B ; Step 2
          DB 0000_0100B ; Step 3
          DB 0000_0110B ; Step 4
  
```

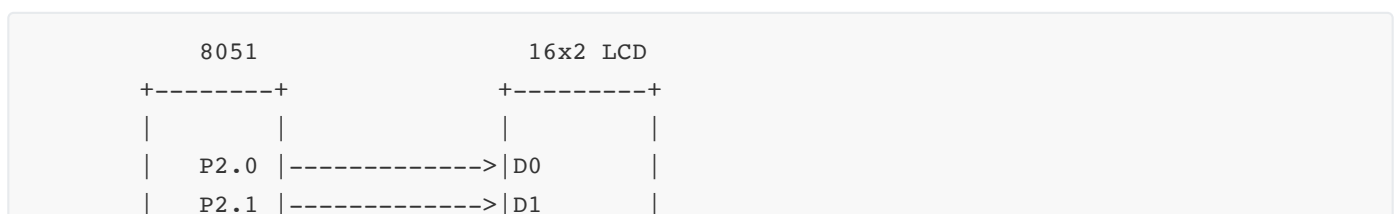
Mnemonic: "PDCS: Port-Driver-Current-Sequence"

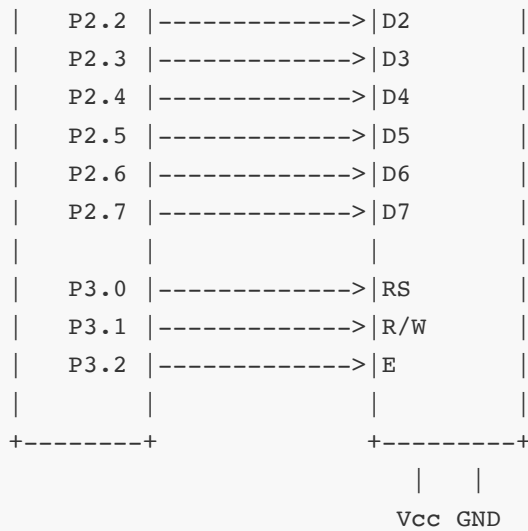
## Question 5(c OR) [7 marks]

Interface LCD with 8051 microcontroller.

Answer:

Circuit Diagram:





Connection	Purpose
Data Pins (D0-D7)	Connect to P2.0-P2.7
RS	Register Select (0=Command, 1=Data)
R/W	Read/Write (0=Write, 1=Read)
E	Enable signal (Active High)

### Basic Commands:

```

0x01 - Clear Display
0x02 - Return Home
0x0C - Display ON, Cursor OFF
0x38 - 8-bit, 2 line, 5x7 dots
  
```

- **Initialization:** Configure LCD for 8-bit mode, 2 lines
- **Writing:** Send data with RS=1, control with RS=0
- **Timing:** E pulse must meet timing requirements
- **Contrast:** Adjust with potentiometer on VEE pin

**Mnemonic:** "DICE: Data-Instruction-Control-Enable"