Microprocessor and Microcontroller (4341101) - Summer 2025 Solution

Milav Dabgar

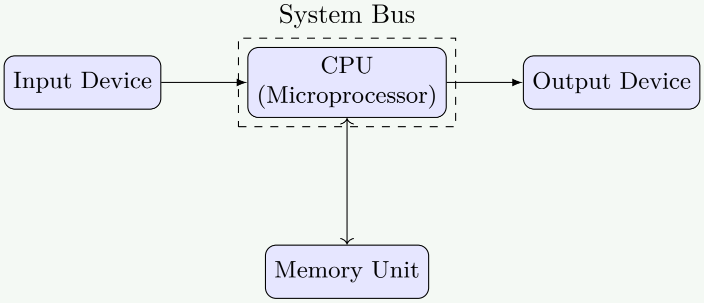
May 13, 2025

Question 1 [a marks]

3 Define Microprocessor and draw its block diagram.

Solution

Answer: A **microprocessor** is a programmable digital device that performs arithmetic and logical operations on data according to stored instructions.
Block Diagram:



- **CPU**: Central Processing Unit performs all operations
- **Memory**: Stores programs and data
- **Control Unit**: Controls instruction execution sequence

Mnemonic

"My Computer Processes Instructions" (Memory-CPU-Program-Instructions)

Question 1 [b marks]

4 Explain operand and opcode with proper instruction example.

Solution

Answer: **Opcode** specifies the operation to be performed. **Operand** specifies the data on which operation is performed.
Example Table:

Table 1. Instruction Parts

Instruction	Opcode	Operand	Function
`MOV A,B`	MOV	A,B	Move B to A
`ADD A,#05H`	ADD	A,#05H	Add 05H to A

- **Opcode**: Operation code (MOV, ADD, SUB)
- **Operand**: Data or address (A, B, #05H)
- **Format**: Opcode + Operand = Complete Instruction

Mnemonic

“Operation On Data” (Opcode-Operand-Data)

Question 1 [c marks]

7 Compare Microprocessor and Microcontroller.

Solution

Answer:

Table 2. Comparison

Parameter	**Microprocessor**	**Microcontroller**
Definition	CPU only	CPU + Memory + I/O
Memory	External RAM/ROM	Internal RAM/ROM
I/O Ports	External interface	Built-in ports
Cost	Higher system cost	Lower system cost
Power	Higher consumption	Lower consumption
Speed	Faster processing	Moderate speed
Applications	Computers, laptops	Washing machine, microwave

- **Microprocessor**: General purpose computing
- **Microcontroller**: Specific embedded applications
- **Integration**: Microcontroller has everything on single chip

Mnemonic

“Micro Means More Integration” (Microcontroller-Memory-More-Integration)

OR

Question 1 [c marks]

7 Compare RISC and CISC.

Solution

Answer:

Table 3. RISC vs CISC

Parameter	**RISC**	**CISC**
Instructions	Simple, few	Complex, many
Instruction Size	Fixed length	Variable length
Execution Time	Single cycle	Multiple cycles
Memory Access	Load/Store only	Any instruction
Registers	More registers	Fewer registers
Pipeline	Efficient pipelining	Complex pipelining
Examples	ARM, MIPS	x86, 8085

- **RISC**: Reduced Instruction Set Computer
- **CISC**: Complex Instruction Set Computer
- **Performance**: RISC faster, CISC more flexible

Mnemonic

"Reduced Instructions Speed Computing" (RISC-Instructions-Speed-Computing)

Question 2 [a marks]

3 Explain Bus Organization of 8085 microprocessor.

Solution

Answer: 8085 has **three types** of buses for communication with external devices.

Table 4. Bus Organization

Bus Type	**Lines**	**Function**
Address Bus	16 lines (A0-A15)	Memory addressing
Data Bus	8 lines (D0-D7)	Data transfer
Control Bus	Multiple lines	Control signals

- **Address Bus**: Unidirectional, 64KB memory addressing
- **Data Bus**: Bidirectional, 8-bit data transfer
- **Control Bus**: Read, Write, IO/M signals

Mnemonic

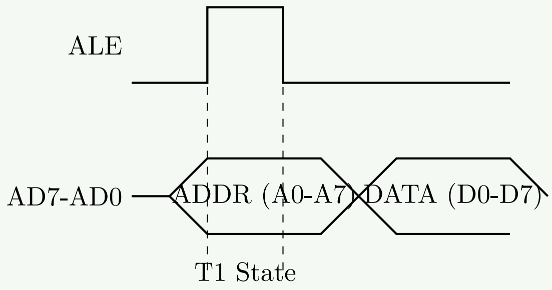
"Address Data Control" (ADC)

Question 2 [b marks]

4 Explain function of ALE signal with diagram.

Solution

Answer: **ALE (Address Latch Enable)** separates address and data on multiplexed bus.
ALE Timing Diagram:



- **High ALE**: Address is available on AD0-AD7
- **Low ALE**: Data is available on AD0-AD7
- **Function**: Latches lower address byte
- **Frequency**: ALE = Clock frequency ÷ 2

Mnemonic

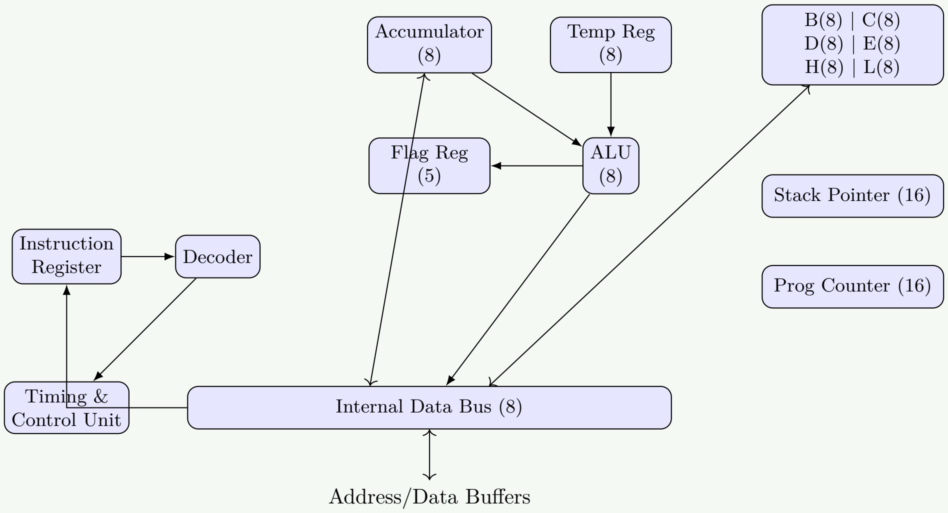
"Address Latch Enable" (ALE)

Question 2 [c marks]

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

Solution

Answer:
Diagram:



Key Components:
- **ALU**: Performs arithmetic and logical operations
- **Registers**: Store temporary data (A, B, C, D, E, H, L)
- **Program Counter**: Points to next instruction
- **Stack Pointer**: Points to stack top
- **Control Unit**: Generates control signals

Mnemonic

"All Registers Program Stack Control" (A-R-P-S-C)

OR

Question 2 [a marks]

3 Draw Flag Register of 8085 microprocessor & explain it.

Solution

Answer:
Flag Register Format:

D0	D1	D2	D3	D4	D5	D6	D7
C	1	P	0	AC	0	Z	S

Flag Functions:

- **S (Sign)**: Set if result is negative
- **Z (Zero)**: Set if result is zero
- **AC (Auxiliary Carry)**: Set for BCD operations (carry from D3 to D4)
- **P (Parity)**: Set for even parity
- **C (Carry)**: Set when carry/borrow occurs

Mnemonic

"Some Zero Auxiliary Parity Carry" (SZAPC)

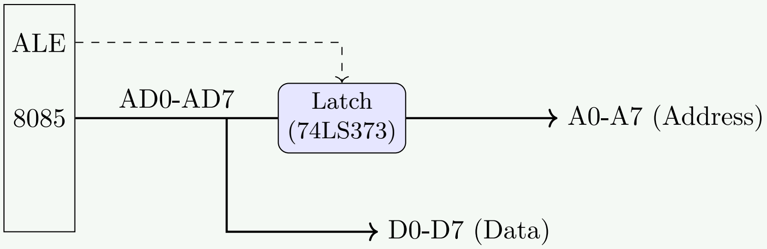
OR

Question 2 [b marks]

4 Explain De-multiplexing of Address and Data buses for 8085 Microprocessor.

Solution

Answer: **De-multiplexing** separates address and data signals from AD0-AD7 lines.
De-multiplexing Circuit:



- **ALE High**: Address latched in external latch (A0-A7)
- **ALE Low**: Data flows through buffer (D0-D7)
- **74LS373**: Common latch IC used
- **Benefit**: Separate address and data buses

Mnemonic

"Address Latch External Demultiplex" (ALED)

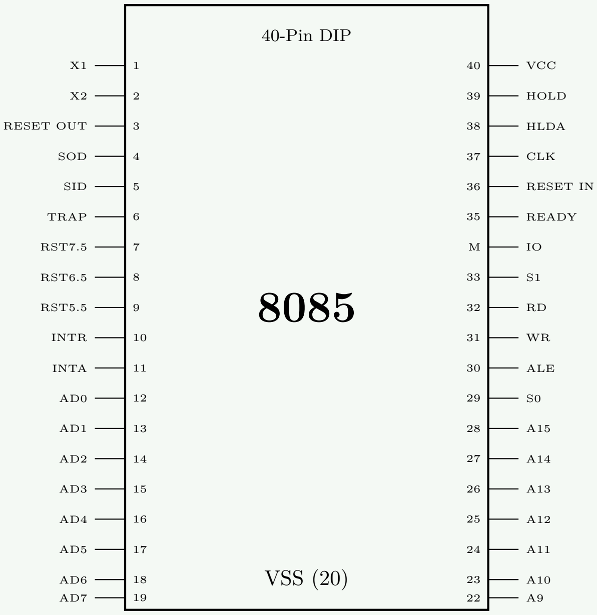
OR

Question 2 [c marks]

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

Solution

Answer:



Pin Categories:

- **Power**: VCC (+5V), VSS (GND)
- **Clock**: X1, X2 (Crystal), CLK OUT
- **Address/Data**: AD0-AD7 (Multiplexed), A8-A15 (High Address)
- **Control**: ALE, RD, WR, IO/M, S0, S1
- **Interrupt**: INTR, INTA, RST7.5, RST6.5, RST5.5, TRAP

Mnemonic

"Power Clock Address Control Interrupt" (PCACI)

Question 3 [a marks]

3 Write a function of DPTR and PC.

Solution

Answer:

Table 5. Functions

Register	Function	Size
DPTR	Data Pointer	16-bit
PC	Program Counter	16-bit

- **DPTR Functions**:

  - Access external data memory (RAM/ROM)
  - Holds 16-bit address for `MOVX A,@DPTR` or `MOVC A,@A+DPTR`
- **PC Functions**:
  - Points to the address of the **next instruction** to be executed
  - Auto-increments after each instruction fetch

Mnemonic

"Data Program Counter" (DPC)

Question 3 [b marks]

4 Draw PCON SFR of 8051 and Explain function of each bit.

Solution

Answer:
PCON Register (87H):

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	-	-	-	GF1	GF0	PD	IDL

Bit Functions:

- **SMOD (D7)**: Serial port baud rate doubler (if set, baud rate is doubled in Mode 1, 2, 3)
- **GF1, GF0 (D3, D2)**: General purpose user flags
- **PD (D1)**: Power Down mode (Oscillator stops, minimal power)
- **IDL (D0)**: Idle mode (Clock to CPU stops, peripherals active)

Mnemonic

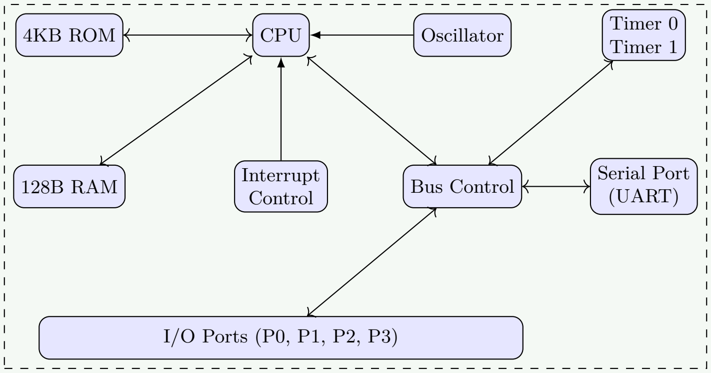
"Serial General Power Idle" (SGPI)

Question 3 [c marks]

7 Explain architecture of 8051 microcontroller with the help of neat diagram.

Solution

Answer:
Diagram:



Major Blocks:

- **CPU**: 8-bit processor with ALU and Accumulator
- **Memory**: 4KB Internal ROM (Code), 128B Internal RAM (Data)
- **Timers**: Two 16-bit timers/counters (T0, T1)
- **Serial Port**: Full duplex UART (TXD, RXD)
- **I/O Ports**: Four 8-bit bidirectional ports (P0-P3)
- **Interrupts**: 5 interrupt sources (External 0/1, Timer 0/1, Serial)

Mnemonic

"CPU Memory Timer Serial IO Interrupt" (CMTSII)

OR

Question 3 [a marks]

3 List common features of 8051 microcontroller.

Solution

Answer: **Common Features:**

- **CPU**: 8-bit microcontroller
- **Memory**: 4KB On-chip Program Memory (ROM), 128 Bytes On-chip Data Memory (RAM)
- **I/O Ports**: 32 I/O lines arranged as four 8-bit ports (P0-P3)
- **Timers**: Two 16-bit timers/counters (T0 and T1)
- **Serial Port**: One Full duplex UART serial channel
- **Interrupts**: 5 interrupt sources (2 External, 2 Timers, 1 Serial)
- **Clock**: On-chip oscillator (typically 12MHz)

Mnemonic

"CPU Memory IO Timer Serial Interrupt Clock" (CMITSIC)

OR

Question 3 [b marks]

4 Draw IP SFR of 8051 and Explain function of each bit.

Solution

Answer:
IP Register (B8H) - Interrupt Priority:

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	PS	PT1	PX1	PT0	PX0

Bit Functions:

- **PS**: Serial Port Interrupt Priority
- **PT1**: Timer 1 Interrupt Priority
- **PX1**: External Interrupt 1 Priority
- **PT0**: Timer 0 Interrupt Priority
- **PX0**: External Interrupt 0 Priority

Priority Levels: 1 = High Priority, 0 = Low Priority

Mnemonic

“Priority Serial Timer External” (PSTE)

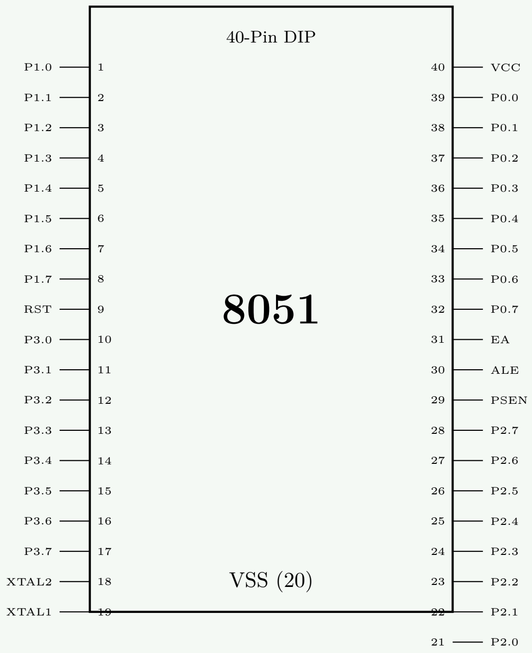
OR

Question 3 [c marks]

7 With the help of neat diagram explain Pin diagram of 8051 microcontroller.

Solution

Answer:



Pin Groups:

- **Power**: VCC (40), VSS (20)
- **Clock**: XTAL1, XTAL2 (Oscillator)
- **Reset**: RST (High active reset)
- **Ports**:
  - P0 (32-39): Address/Data bus
  - P1 (1-8): I/O only
  - P2 (21-28): High Address
  - P3 (10-17): Special functions (Serial, Interrupts, Timers)
- **Control**: ALE, PSEN, EA

Mnemonic

“Power Clock Reset Ports Control” (PCRPC)

Question 4 [a marks]

3 Explain arithmetic instructions with example.

Solution

Answer:
Arithmetic Instructions:

Table 6. Arithmetic Instructions

Instruction	Function	Example
ADD	Addition	`ADD A,#10H`
SUBB	Subtraction	`SUBB A,R0`
MUL	Multiplication	`MUL AB`
DIV	Division	`DIV AB`
INC	Increment	`INC A`
DEC	Decrement	`DEC R1`

- `ADD A,#10H`: Add 10H to accumulator
- **Flags**: Affected by arithmetic operations (C, AC, OV, P)

Mnemonic

"Add Subtract Multiply Divide Increment Decrement" (ASMIDI)

Question 4 [b marks]

4 Write an 8051 Assembly Language Program to Find 2's complement of a value stored at memory location 65H. Put the result on same location.

Solution

Answer:

```
ORG 0000H           ; Program start address
MOV A,65H           ; Load value from location 65H
CPL A               ; Complement the value (1's complement)
ADD A,#01H          ; Add 1 to get 2's complement
MOV 65H,A           ; Store result back to 65H
SJMP $              ; Stop program
END
```

Program Steps:

- **Load**: Get value from memory location 65H
- **Complement**: Generate 1's complement using CPL
- **Add 1**: Convert to 2's complement
- **Store**: Put result back to same location

Mnemonic

"Load Complement Add Store" (LCAS)

Question 4 [c marks]

7 List Addressing Modes of 8051 Microcontroller and explain them with example.

Solution

Answer:

Table 7. Addressing Modes

Mode	Description	Example	Usage
Immediate	Data directly in instruction	`MOV A,#25H`	Constant data
Register	Data in register	`MOV A,R0`	Fast access
Direct	Memory address specified	`MOV A,30H`	RAM access
Indirect	Address stored in register	`MOV A,@R0`	Pointer/Array access
Indexed	Base address + offset	`MOVC A,@A+DPTR`	Table lookup
Relative	Jump amount relative to PC	`SJMP LOOP`	Branching
Bit	Operations on single bit	`SETB P1.0`	Bit manipulation

Examples:

- `MOV A,#25H`: Load immediate value 25H
- `MOV A,@R0`: Load data from address held in R0
- `SJMP LOOP`: Jump to label LOOP (relative to current PC)

Mnemonic

"Immediate Register Direct Indirect Indexed Relative Bit" (IRDIIRB)

OR

Question 4 [a marks]

3 Explain logical instruction with example.

Solution

Answer:
Logical Instructions:

Table 8. Logical Instructions

Instruction	Function	Example
ANL	AND operation	`ANL A,#0FH`
ORL	OR operation	`ORL A,R1`
XRL	XOR operation	`XRL A,#55H`
CPL	Complement	`CPL A`
RL	Rotate Left	`RL A`
RR	Rotate Right	`RR A`

- `ANL A,#0FH`: AND accumulator with 0FH (Masking example)
- **Applications**: Bit masking, data manipulation, flag testing

Mnemonic

"AND OR XOR Complement Rotate" (AOXCR)

OR

Question 4 [b marks]

4 Write an 8051 Assembly Language Program to Multiply the number in register R3 by the number in register R0 and put the result in internal RAM location 10h(MSB) and 11h(LSB).

Solution

Answer:

```
ORG 0000H           ; Program start address
MOV A,R3            ; Move multiplicand (R3) to Accumulator
MOV B,R0            ; Move multiplier (R0) to B register
MUL AB              ; Multiply A by B (Product: B=High, A=Low)
MOV 10H,B           ; Store MSB (B) to location 10H
MOV 11H,A           ; Store LSB (A) to location 11H
SJMP $              ; Stop program
END
```

Program Flow:

- **Load**: Move multiplicand and multiplier to specific registers (A and B)
- **Multiply**: Execute `MUL AB` to perform 8-bit × 8-bit multiplication
- **Result**: 16-bit product is stored in B (High Byte) and A (Low Byte)
- **Store**: Save MSB and LSB to specified RAM locations

Mnemonic

"Load Multiply Store Result" (LMSR)

OR

Question 4 [c marks]

7 Explain data transfer instruction with example.

Solution

Answer:
Data Transfer Instructions:

Table 9. Data Transfer Instructions

Category	Instruction	Example	Function
Register	MOV	`MOV A,R0`	Register to register
Immediate	MOV	`MOV A,#25H`	Immediate to register
Direct	MOV	`MOV A,30H`	Memory to register
Indirect	MOV	`MOV A,@R0`	Indirect addressing
External	MOVX	`MOVX A,@DPTR`	External data memory
Code	MOVC	`MOVC A,@A+DPTR`	Program (Code) memory
Stack	PUSH/POP	`PUSH ACC`	Stack operations

Examples:

- `MOV A,R0`: Move content of R0 to Accumulator
- `MOVX A,@DPTR`: Read data from external RAM at address in DPTR
- `PUSH ACC`: Push Accumulator content onto the Stack

Mnemonic

"Move Data Between Locations" (MDBL)

Question 5 [a marks]

3 Explain the 8051 flags with the help of PSW format.

Solution

Answer:
PSW Register (D0H):

D0	D1	D2	D3	D4	D5	D6	D7
P	-	OV	RS0	RS1	F0	AC	C

Flag Functions:

- **C (Carry - D7)**: Set when carry/borrow occurs in arithmetic
- **AC (Auxiliary Carry - D6)**: Set when carry from D3 to D4 (BCD arithmetic)
- **F0 (D5)**: User defined flag
- **RS1, RS0 (D4, D3)**: Register Bank Select (00=Bank0, 01=Bank1, 10=Bank2, 11=Bank3)
- **OV (Overflow - D2)**: Set when signed arithmetic overflow occurs
- **P (Parity - D0)**: Set to 1 if Accumulator has odd number of 1s (Even Parity needed)

Mnemonic

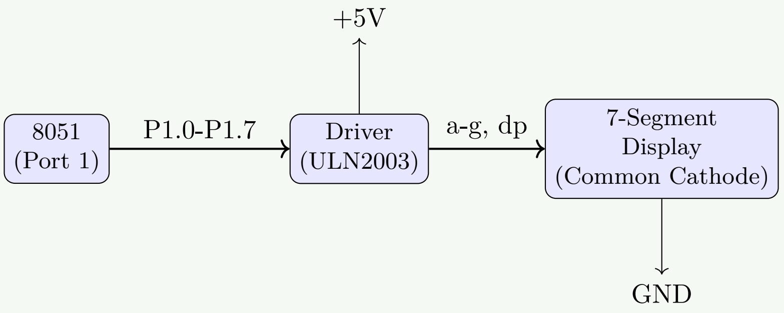
"Carry Auxiliary Overflow Parity Register" (CAOPR)

Question 5 [b marks]

4 Draw and explain diagram Interfacing 7 segment with microcontroller.

Solution

Answer:
7-Segment Interface (Common Cathode):



Components:

- **ULN2003/Resistors**: Used as current driver/limiter because 8051 ports generally cannot drive LED segments directly (or use logic low to drive Common Anode).
- **Display**: Common Cathode type requires Logic 1 (High) to turn on segment (via driver).

Mnemonic

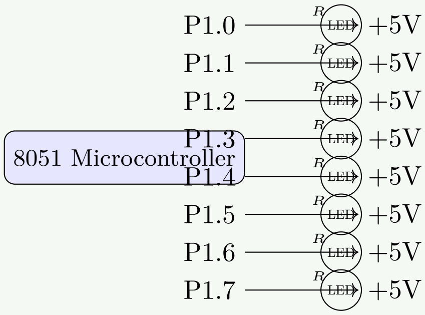
"Port Driver Display Ground" (PDDG)

Question 5 [c marks]

7 Interface 8 LEDs with microcontroller and write a program to turn on and off.

Solution

Answer:
LED Interface Circuit:



Note: Diagram shows Common Anode configuration (Active Low) for simple driving.
Assembly Program:

```
ORG 0000H           ; Start address
MAIN:
    MOV P1,#00H     ; Turn ON all LEDs (Logic 0 for Active Low)
                    ; If Active High: MOV P1,#0FFH
    ACALL DELAY     ; Wait
    MOV P1,#0FFH    ; Turn OFF all LEDs (Logic 1)
                    ; If Active High: MOV P1,#00H
    ACALL DELAY     ; Wait
    SJMP MAIN       ; Repeat continuously

DELAY:
    MOV R2,#250     ; Outer loop
D1: MOV R3,#250     ; Inner loop
D2: DJNZ R3,D2      ; Decrement inner
    DJNZ R2,D1      ; Decrement outer
    RET             ; Return
END
```

Mnemonic

"Light Emitting Display Interface" (LEDI)

OR

Question 5 [a marks]

3 List Applications of microcontroller in various fields.

Solution

Answer:

Table 10. Applications

Field	**Applications**
Home Appliances	Washing machine, Microwave, AC, TV Remote
Automotive	Engine Control Unit (ECU), ABS, Airbags, Dashboard
Industrial	Process control, Robotics, Sensors, Automation
Medical	Pacemaker, Blood pressure monitor, Ventilators
Communication	Mobile phones, Modems, Routers
Security	Access control systems, Burglar alarms, CCTV
Entertainment	Gaming consoles, Music players, Toys

Mnemonic

"Home Auto Industrial Medical Communication Security Entertainment" (HAIMCSE)

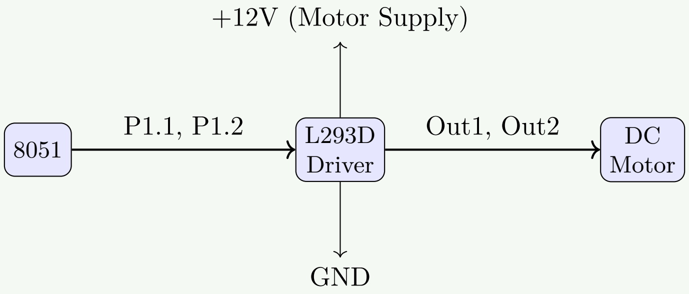
OR

Question 5 [b marks]

4 Draw and explain diagram interfacing of DC motor with 8051.

Solution

Answer:
DC Motor Interface (using L293D):



Operation using H-Bridge (L293D):

- **Forward**: P1.1 = 1, P1.2 = 0
- **Reverse**: P1.1 = 0, P1.2 = 1
- **Stop**: P1.1 = 0, P1.2 = 0 (or 1, 1)

Mnemonic

"Driver Control Motor Direction" (DCMD)

OR

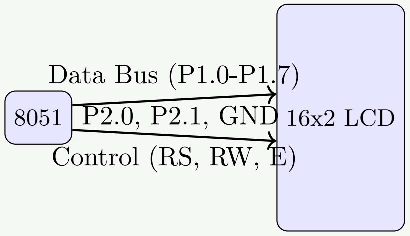
Question 5 [c marks]

7 Interface LCD with microcontroller and write a program to display "Microprocessor and Microcon-

troller”.

Solution

Answer:
LCD Interface (16x2):



Assembly Program:

```
ORG 0000H
    ACALL LCD_INIT      ; Initialize LCD
    MOV DPTR,#MSG       ; Point to message
DISP_LOOP:
    CLR A
    MOVC A,@A+DPTR      ; Get character
    JZ STOP             ; If 0, stop
    ACALL SEND_DATA     ; Display char
    INC DPTR            ; Next char
    SJMP DISP_LOOP      ; Repeat
STOP: SJMP $

LCD_INIT:
    MOV A,#38H          ; 2 lines, 5x7 matrix
    ACALL SEND_CMD
    MOV A,#0FH          ; Display ON, Cursor ON
    ACALL SEND_CMD
    MOV A,#01H          ; Clear Display
    ACALL SEND_CMD
    RET

SEND_CMD:
    MOV P1,A            ; Send command to Data Port
    CLR P2.0            ; RS=0 for Command
    CLR P2.1            ; RW=0 for Write
    SETB P2.2           ; E=1
    CLR P2.2            ; E=0 (Latch)
    ACALL DELAY
    RET

SEND_DATA:
    MOV P1,A            ; Send data to Data Port
    SETB P2.0           ; RS=1 for Data
    CLR P2.1            ; RW=0 for Write
    SETB P2.2           ; E=1
    CLR P2.2            ; E=0 (Latch)
    ACALL DELAY
    RET

DELAY: MOV R3,#50       ; Simple delay
       DJNZ R3,$
       RET

MSG: DB "Microprocessor and Microcontroller",0h
END
```

Mnemonic

"Init Send Data Display Message" (ISDDM)