

# Subject Name Solutions

4341101 – Summer 2023

Semester 1 Study Material

*Detailed Solutions and Explanations*

## Question 1(a) [3 marks]

Compare Microprocessor and Microcontroller.

Solution			
	Feature	Microprocessor	Microcontroller
Definition		CPU on single chip	Complete computer on single chip
Memory		External RAM/ROM needed	Built-in RAM/ROM
Applications		General computing, PCs	Embedded systems, IoT
Examples		Intel 8085, 8086	8051, Arduino, PIC
Cost		Higher	Lower

### Mnemonic

“PCRAM” - “Processors Connect to RAM, Microcontrollers Already have Memory”

## Question 1(b) [4 marks]

Compare RISC and CISC.

Solution			
	Feature	RISC (Reduced Instruction Set Computer)	CISC (Complex Instruction Set Computer)
Instructions		Few, simple instructions	Many, complex instructions
Execution Time		Fixed (1 clock cycle)	Variable (multiple cycles)
Memory Access		Only through load/store	Multiple memory access modes
Pipelining		Easy implementation	Difficult implementation
Examples		ARM, MIPS	Intel x86, 8085
Hardware		Simple, less transistors	Complex, more transistors
Register Set		Large number of registers	Fewer registers

### Mnemonic

“RISC-Fast, CISC-Many” (RISC uses Fast execution, CISC has Many instructions)

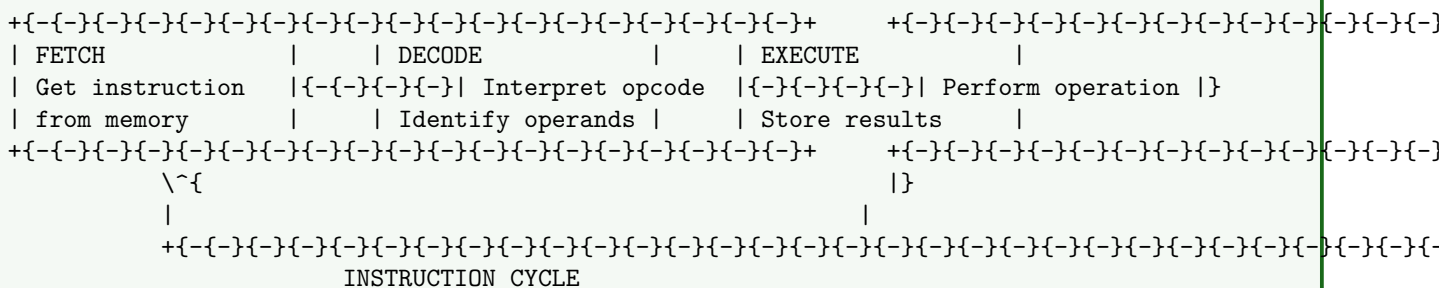
## Question 1(c) [7 marks]

Define: Microprocessor, Operand, Instruction Cycle, Opcode, ALU, Machine Cycle, T-State

Solution	
Term	Definition
Microprocessor	CPU on a single integrated circuit that processes instructions

<b>Operand</b>	Data value used in an instruction operation
<b>Instruction Cycle</b>	Complete process to fetch, decode and execute an instruction
<b>Opcode</b>	Operation code that tells CPU what operation to perform
<b>ALU</b>	Arithmetic Logic Unit that performs mathematical computations
<b>Machine Cycle</b>	Basic operation like memory read/write (subset of instruction cycle)
<b>T-State</b>	Time state - smallest unit of time in processor operation (clock period)

Diagram:



### Mnemonic

“My Old Intel Chip Only Makes Trouble” (Microprocessor, Operand, Instruction, Opcode, ALU, Machine, T-state)

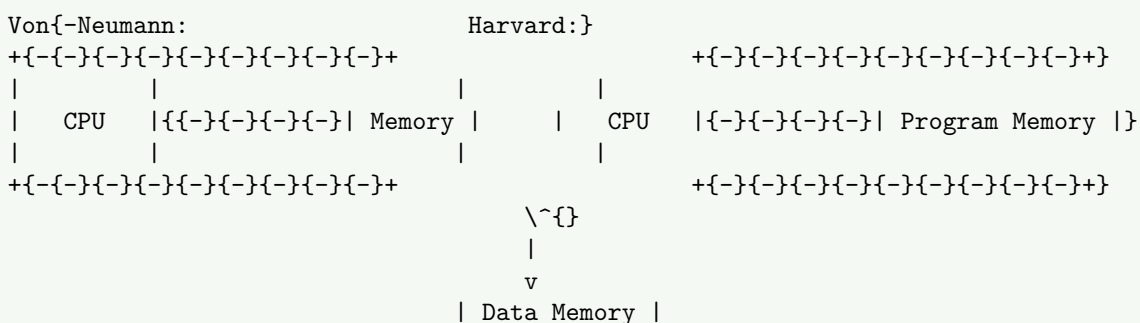
Question 1(c OR) [7 marks]

**Compare Von-Neumann and Harvard architecture.**

## Solution

Feature	Von-Neumann Architecture	Harvard Architecture
Memory Buses	Single memory bus for instructions and data	Separate buses for program and data memory
Execution	Sequential execution	Parallel fetch and execute possible
Speed	Slower due to bus bottleneck	Faster due to simultaneous access
Complexity	Simpler design	More complex design
Applications	General-purpose computing	DSP, microcontrollers, embedded systems
Security	Less secure (code can be modified as data)	More secure (code separation from data)
Example	Most PCs, 8085, 8086	8051, PIC, ARM Cortex-M

Diagram:



### Mnemonic

“Harvard Has Separate Streets” (Harvard Has Separate memory paths)

Question 2(a) [3 marks]

Draw Flag Register of 8085 microprocessor & explain it.

## Solution

Diagram:

[illegible]

Flag	Name	Purpose
S	Sign	Set if result is negative (bit 7=1)
Z	Zero	Set if result is zero
AC	Auxiliary Carry	Set if carry from bit 3 to bit 4
P	Parity	Set if result has even parity
CY	Carry	Set if carry from bit 7 or borrow to bit 7

## Mnemonic

“Smart Zombies Always Prefer Candy” (Sign, Zero, Auxiliary, Parity, Carry)

**Question 2(b) [4 marks]**

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

## Solution

Diagram:

[illegible]

- **Need:** 8085 has multiplexed pins (AD0-AD7) to save pins
- **Process:**
  1. CPU places address on AD0-AD7 pins
  2. ALE (Address Latch Enable) signal goes HIGH
  3. Address latch (74LS373) captures lower address bits
  4. ALE goes LOW, latching the address
  5. AD0-AD7 pins now free for data transfer

### Mnemonic

“ALE Latches, Data Follows” (Address Latch Enable captures address first, then data)

Question 2(c) [7 marks]

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

Solution

Diagram:

8085 CPU

REGISTERS: A (Accum.), B, C, D, E, H, L, SP, PC, Flags

CONTROL UNIT: Instruction, Decoder, Timing, Control

ALU

Address Bus (16-bit), Data Bus (8-bit), Control Bus

Memory, I/O Devices

- **Main Components:**
  - **Registers:** Storage locations (A, B-L, SP, PC, Flags)
  - **ALU:** Performs arithmetic and logical operations
  - **Control Unit:** Generates timing and control signals
  - **Bus:** Address bus (16-bit), Data bus (8-bit), Control bus
- **Key Features:**
  - 8-bit data bus, 16-bit address bus (64KB addressable memory)
  - 6 general-purpose registers (B,C,D,E,H,L) and accumulator
  - 5 flags for status information

Mnemonic

“RABC” - “Registers, ALU, Buses, Control” (main components)

Question 2(a OR) [3 marks]

Explain Bus Organization of 8085 microprocessor.

Solution		
Bus Type	Width	Function
Address Bus	16-bit (A0-A15)	Carries memory/I/O device addresses
Data Bus	8-bit (D0-D7)	Transfers data between CPU & memory/I/O

## Control Bus

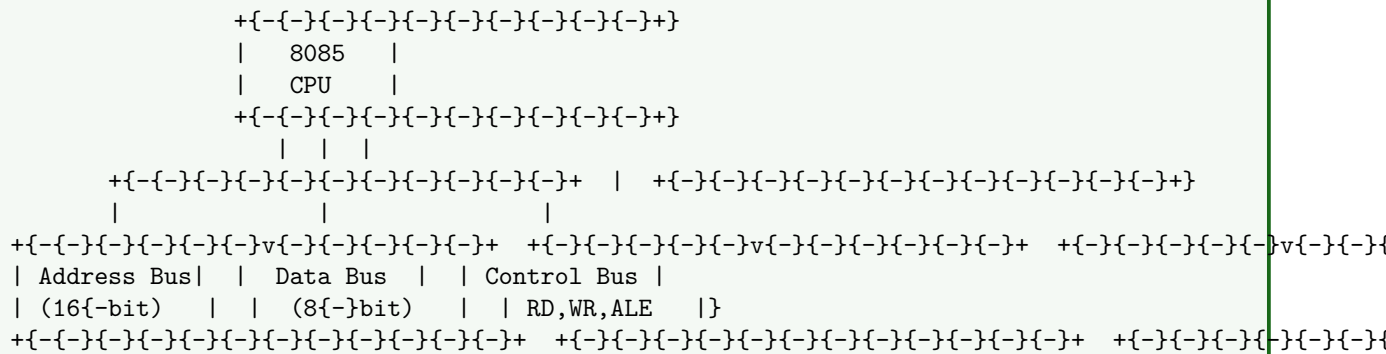
Various signals

Coordinates system operations

### Key Control Signals:

- **RD**: Read signal (active low)
- **WR**: Write signal (active low)
- **ALE**: Address Latch Enable
- **IO/M**: Distinguishes I/O (high) from memory (low) operations

### Diagram:



## Mnemonic

“ADC” - “Address finds, Data travels, Control coordinates”

## Question 2(b OR) [4 marks]

Explain: Program Counter & Stack pointer

### Solution

Register	Size	Function
<b>Program Counter (PC)</b>	16-bit	Holds address of next instruction to execute
<b>Stack Pointer (SP)</b>	16-bit	Points to the top of the stack in memory

- Automatically increments after instruction fetch
- Modified by jump/call instructions
- Controls program execution sequence
- Initially set to 0000H on reset

- Points to last data item pushed onto stack
- Stack works in LIFO (Last In First Out) manner
- Used during subroutine calls and interrupts
- Stack grows downward in memory (decrements)

[illegible]

“PC Previews, SP Stacks” (PC points to next instruction, SP manages stack)

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

+5V	{-}{-}{-}{-}		{-}{-}{-}{-} GND}
X1	{-}{-}{-}{-}		{-}{-}{-}{-} X2}
RESET	{-}{-}{-}{-}		{-}{-}{-}{-} READY}
HOLD	{-}{-}{-}{-}{-}{-}		{-}{-}{-}{-}{-} CLK OUT}
HLDA	{{-}}{-}{-}{-}{-}{-}{-}	8085	{-}{-}{-}{-}{-} RESET IN}
INTR	{{-}}{-}{-}{-}{-}{-}{-}{-}		{-}{-}{-}{-}{-} RST 7.5}
INTA	{{-}}{-}{-}{-}{-}{-}{-}{-}{-}		{-}{-}{-}{-}{-} RST 6.5}
SOD	{{-}}{-}{-}{-}{-}{-}{-}		{-}{-}{-}{-}{-} RST 5.5}
SID	{{-}}{-}{-}{-}{-}{-}{-}		{-}{-}{-}{-}{-} TRAP}
RD	{{-}}{-}{-}{-}{-}{-}{-}{-}{-}		}
WR	{{-}}{-}{-}{-}{-}{-}{-}{-}{-}		}
IO/M	{{-}}{-}{-}{-}{-}{-}{-}{-}{-}		}

Pin diagram of the 8085 microprocessor showing 40 pins. The pins are arranged in two rows of 20. The top row contains pins 1-20 and the bottom row contains pins 40-21. The pins are labeled as follows: Pin 1: ALE, Pin 2: S1, Pin 3: S0, Pin 4: A15-A8, Pin 5: AD7-AD0, Pin 6: +, Pin 7: -, Pin 8: -, Pin 9: -, Pin 10: -, Pin 11: -, Pin 12: -, Pin 13: -, Pin 14: -, Pin 15: -, Pin 16: -, Pin 17: -, Pin 18: -, Pin 19: -, Pin 20: +, Pin 21: ALE, Pin 22: S1, Pin 23: S0, Pin 24: A15-A8, Pin 25: AD7-AD0, Pin 26: +, Pin 27: -, Pin 28: -, Pin 29: -, Pin 30: -, Pin 31: -, Pin 32: -, Pin 33: -, Pin 34: -, Pin 35: -, Pin 36: -, Pin 37: -, Pin 38: -, Pin 39: -, Pin 40: +.

1. **Power & Clock:** Vcc, GND, X1, X2, CLK
2. **Address/Data:** A8-A15, AD0-AD7 (multiplexed)
3. **Control:** ALE, RD, WR, IO/M
4. **Interrupt:** INTR, INTA, RST 5.5/6.5/7.5, TRAP
5. **DMA:** HOLD, HLDA
6. **Serial I/O:** SID, SOD
7. **Status:** S0, S1

**Mnemonic**

“PACI-DHS” (Power, Address, Control, Interrupt, DMA, Hardware status, Serial)

**Mnemonic**

“PACI-DHS” (Power, Address, Control, Interrupt, DMA, Hardware status, Serial)

**Question 3(a) [3 marks]**

### Explain Stack, Stack Pointer and Stack operation.

## Solution

[illegible]

+{ - { - } { - } { - } { - } { - } { - } { - } +	{ - } { - }	2000H	+{ - } { - } { - } { - } { - } { - } { - } { - } +		+{ - } { - } { - } { - } { - } { - } { - } { - }
+{ - { - } { - } { - } { - } { - } { - } { - } +	{ - } { - }	1FFFH	+{ - } { - } { - } { - } { - } { - } { - } { - } +		+{ - } { - } { - } { - } { - } { - } { - } { - }
		76H	{ - } { - } SP		{ - } { - } SP}
+{ - { - } { - } { - } { - } { - } { - } { - } +	{ - } { - }	1FFE H	+{ - } { - } { - } { - } { - } { - } { - } { - } +		+{ - } { - } { - } { - } { - } { - } { - } { - }
	{ - } { - } SP				}
+{ - { - } { - } { - } { - } { - } { - } { - } +			+{ - } { - } { - } { - } { - } { - } { - } { - } +		+{ - } { - } { - } { - } { - } { - } { - } { - }

**Mnemonic**

“LIFO Saves Push-Pop” (Last-In-First-Out with Push and Pop operations)

**Mnemonic**

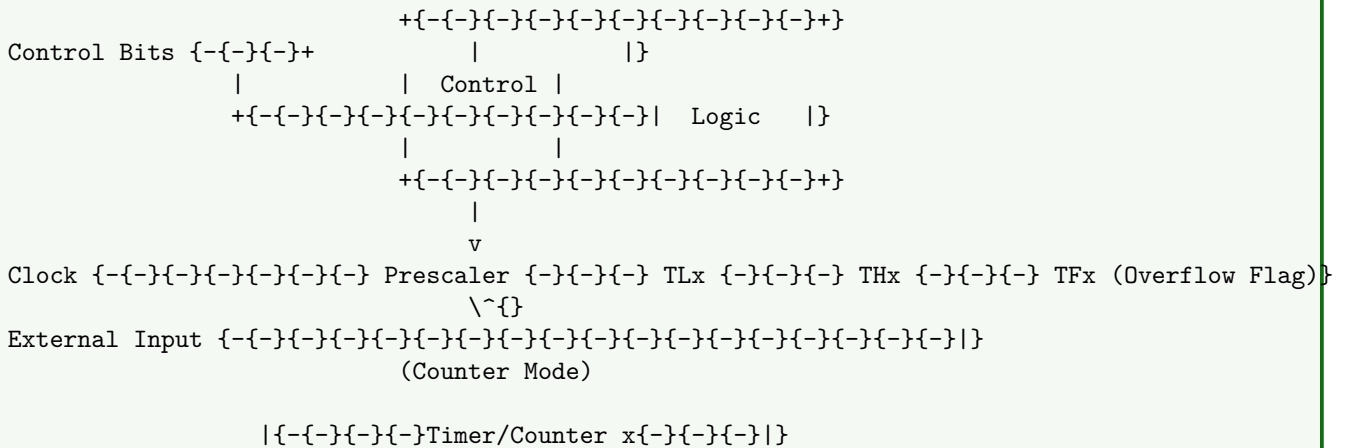
“LIFO Saves Push-Pop” (Last-In-First-Out with Push and Pop operations)

Question 3(b) [4 marks]

Draw Timers/Counters logic diagram of 8051 microcontroller and explain it.

## Solution

### Diagram:



- **8051 has 2 16-bit timers/counters:** Timer 0 and Timer 1
- **Each timer has two 8-bit registers:** THx (High byte) and TLx (Low byte)
- **4 Operating Modes:**
  - Mode 0: 13-bit timer
  - Mode 1: 16-bit timer
  - Mode 2: 8-bit auto-reload
  - Mode 3: Split timer mode
- **Two Functions:**
  - Timer: Counts internal clock pulses
  - Counter: Counts external events

## Mnemonic

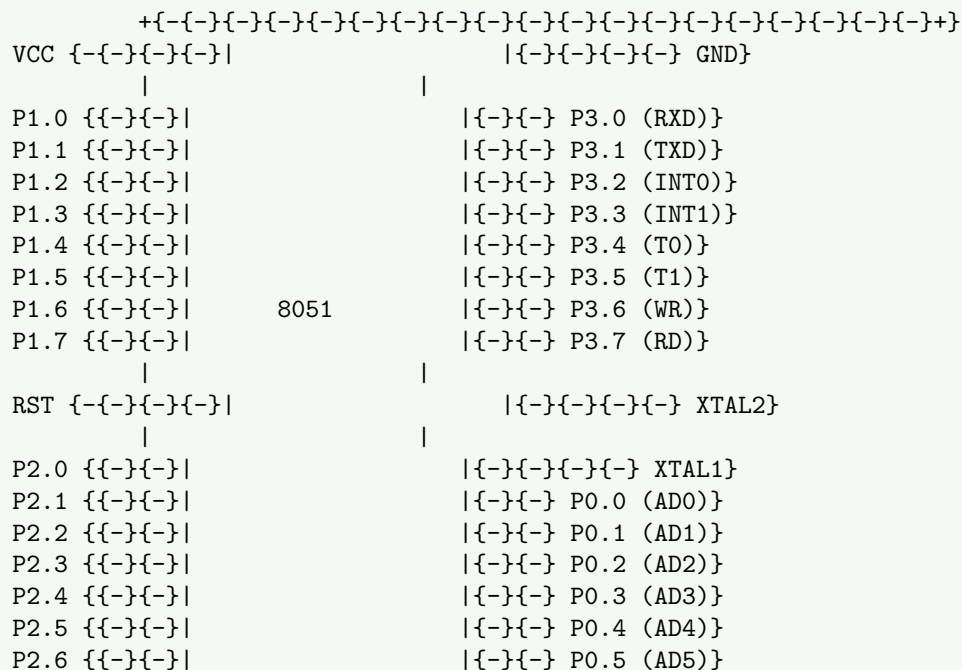
“TIME-C” (Timer Input, Mode select, External count)

## Question 3(c) [7 marks]

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

## Solution

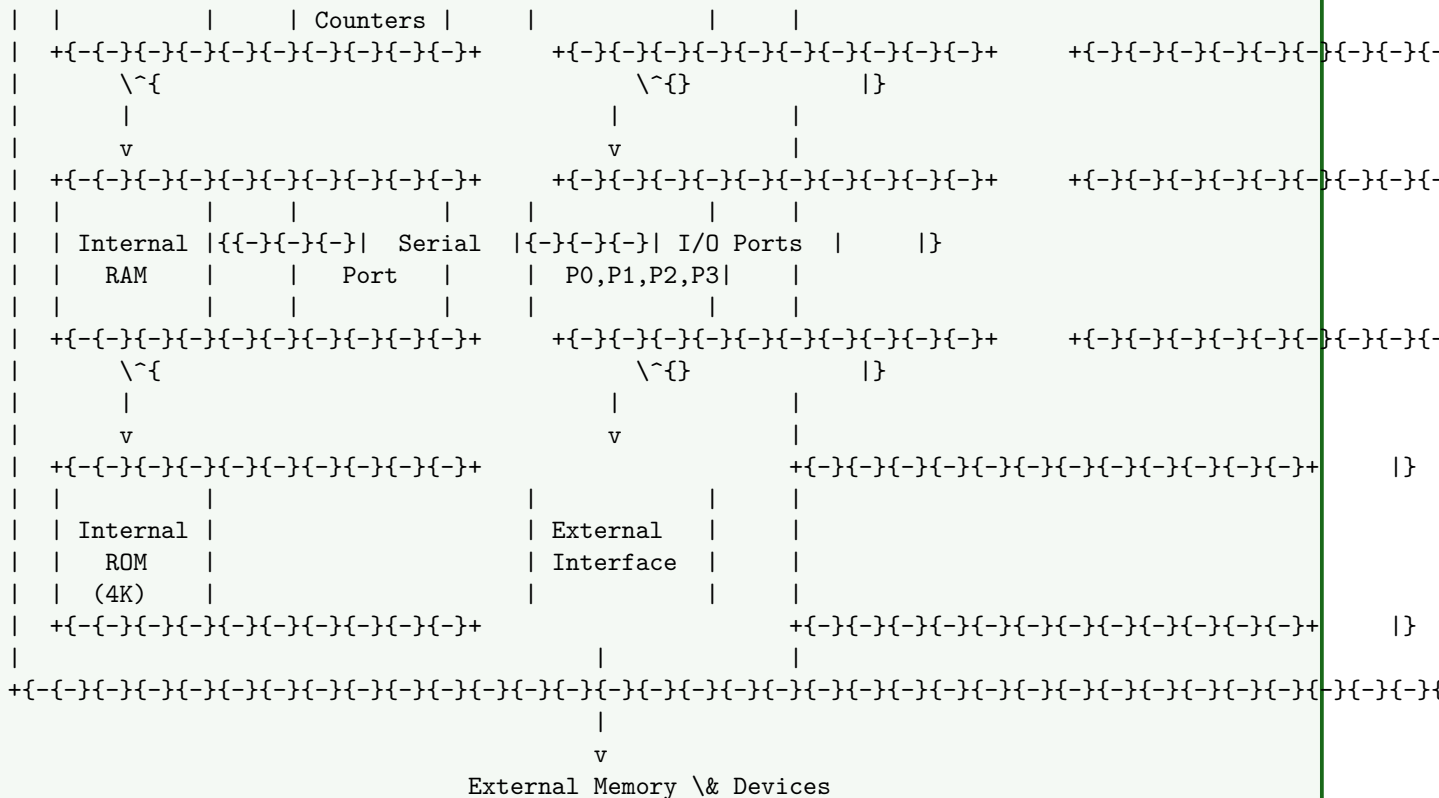
### Diagram:





Pin diagram of the 8051 microcontroller showing 40 pins. Pins 1-12 are on the left, 13-24 on the top, 25-36 on the right, and 37-40 on the bottom. Pin 1 is GND, Pin 40 is VCC. Pins 2-3 are P0. Pins 4-5 are P2.7 and P2.6. Pins 6-7 are P2.5 and P2.4. Pins 8-9 are P2.3 and P2.2. Pins 10-11 are P2.1 and P2.0. Pins 12-13 are PSEN and ALE. Pins 14-15 are P0.7 and P0.6. Pins 16-17 are P0.5 and P0.4. Pins 18-19 are P0.3 and P0.2. Pins 20-21 are P0.1 and P0.0. Pins 22-23 are EA and GND. Pins 24-25 are VCC and GND. Pins 26-27 are XTAL2 and XTAL1. Pins 28-29 are GND and VCC. Pins 30-31 are P3.7 and P3.6. Pins 32-33 are P3.5 and P3.4. Pins 34-35 are P3.3 and P3.2. Pins 36-37 are P3.1 and P3.0. Pins 38-39 are P3.7 and P3.6.





#### Key Components:

- **CPU:** 8-bit processor with ALU, registers, and control logic
- **Memory:**
  - 4KB internal ROM (program memory)
  - 128 bytes internal RAM (data memory)
- **I/O:** Four 8-bit I/O ports (P0-P3)
- **Timers:** Two 16-bit timers/counters
- **Serial Port:** Full-duplex UART
- **Interrupts:** Five interrupt sources with two priority levels

#### Mnemonic

“BASICS” (Bus, Architecture with CPU, Serial port, I/O ports, Counters/timers, Special functions)

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

Write an 8051 Assembly Language Program to Exchange lower nibbles of register R5 and R6: put the lower nibble of R5 into R6, and the lower nibble of R6 into R5.

#### Solution

```

; Exchange lower nibbles of R5 and R6
MOV A, R5      ; Copy R5 to accumulator
ANL A, \#0FH   ; Mask upper nibble (keep only lower nibble)
MOV B, A       ; Save R5's lower nibble in B

MOV A, R6      ; Copy R6 to accumulator
ANL A, \#0FH   ; Mask upper nibble (keep only lower nibble)
MOV C, A       ; Save temporarily in a free register (R7)

MOV A, R5      ; Get R5 again
ANL A, \#F0H   ; Keep only upper nibble of R5
ORL A, C       ; Combine with lower nibble from R6
MOV R5, A      ; Store result back in R5

```





		2041H: 00   (No carry)
v	v	+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{+}
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{+}		}
	ADD	
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{+}		

**Mnemonic**

“MASIM” (Move, Add, Store, Increment, Move again)

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

**For 8051 Microcontroller with a crystal frequency of 12 MHz, generate a delay of 5ms.**

## Solution

```

; Delay of 5ms with 12MHz Crystal (1 machine cycle = 1 s)
DELAY: MOV R7, \#5      ; 5 loops of 1ms each
LOOP1: MOV R6, \#250     ; 250 x 4 s = 1000 s = 1ms
LOOP2: NOP               ; 1 s
      NOP               ; 1 s
      DJNZ R6, LOOP2    ; 2 s (if jump taken)
      DJNZ R7, LOOP1    ; Repeat 5 times for 5ms
      RET              ; Return from subroutine

```

Diagram:

```
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
| Start      |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
    |
    v
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
| R7 = 5      |{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+          |}
    |                      |
    v                      |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+          |}
| R6 = 250     |{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+   |}
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+           |   |}
    |              |      |
    v              |      |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+           |   |}
| 2 NOPs        |      |      |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+           |   |}
    |              |      |
    v              |      |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+           |   |}
| Decrement R6 |{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+   |}
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+           |}
    |              |
    v              |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+           |}
| Decrement R7 |{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
    |
    v
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
| Return       |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
```

Calculation:

- 12MHz crystal = 1 s machine cycle
- Inner loop: 2 NOPs (2 s) + DJNZ (2 s) = 4 s per iteration
- 250 iterations  $\times 4s = 1000s = 1ms$
- Outer loop: 5 iterations  $\times 1ms = 5ms$

### Mnemonic

“LOON-5” (LOOp Nested for 5ms)

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

Explain any seven Arithmetic Instructions with example for 8051 Microcontroller.

### Solution

Instruction	Function	Example	Flag Affected
<b>ADD A,src</b>	Add source to A	ADD A,R0 (A=A+R0)	C, OV, AC
<b>ADDC A,src</b>	Add source + carry to A	ADDC A,#25H (A=A+25H+C)	C, OV, AC
<b>SUBB A,src</b>	Subtract source + borrow from A	SUBB A,@R1 (A=A-@R1-C)	C, OV, AC
<b>INC</b>	Increment by 1	INC R3 (R3=R3+1)	None
<b>DEC</b>	Decrement by 1	DEC A (A=A-1)	None
<b>MUL AB</b>	Multiply A and B	MUL AB (B:A=A)	C, OV
<b>DIV AB</b>	Divide A by B	DIV AB (A=quotient, B=remainder)	C, OV

+{-}+		+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}
ADD A,RO	MUL AB	DIV AB

[illegible]

“ACID-IBM” (Add, Carry add, Inc, Dec, Mul, Borrow subtract, Divide)

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

Field	Applications
Consumer Electronics	TV, washing machine, microwave, remote control
Automotive	Engine control, anti-lock braking, airbag systems
Industrial	Automation, robotics, process control
Medical	Patient monitoring, medical instruments, implants
Home Automation	Smart lighting, security systems, HVAC control
Communication	Mobile phones, routers, modems
Aerospace	Navigation systems, flight control, satellite systems



[illegible]

**Mnemonic**

“CHAIM-MA” (Consumer, Home, Automotive, Industrial, Medical, Mobile, Aerospace)

“CHAIM-MA” (Consumer, Home, Automotive, Industrial, Medical, Mobile, Aerospace)

Question 5(b) [4 marks]

### Interface Relay with 8051 microcontroller.

**Solution**

[illegible]

**Components Required:**

- 8051 microcontroller
- ULN2003 or similar driver IC
- Relay (5V or 12V)
- Protection diode (1N4007)
- Power supply

- Components Required:**
- 8051 microcontroller
  - ULN2003 or similar driver IC
  - Relay (5V or 12V)
  - Protection diode (1N4007)
  - Power supply

**Working:**

1. 8051 sends control signal from P1.0
2. Driver amplifies current to drive relay
3. Protection diode prevents back EMF damage
4. Relay switches connected devices

- Working:**
1. 8051 sends control signal from P1.0
  2. Driver amplifies current to drive relay
  3. Protection diode prevents back EMF damage
  4. Relay switches connected devices

**Mnemonic**

“DRIPS” (Driver, Relay, Input from  $\mu\text{C}$ , Protection diode, Switching)

**Mnemonic**

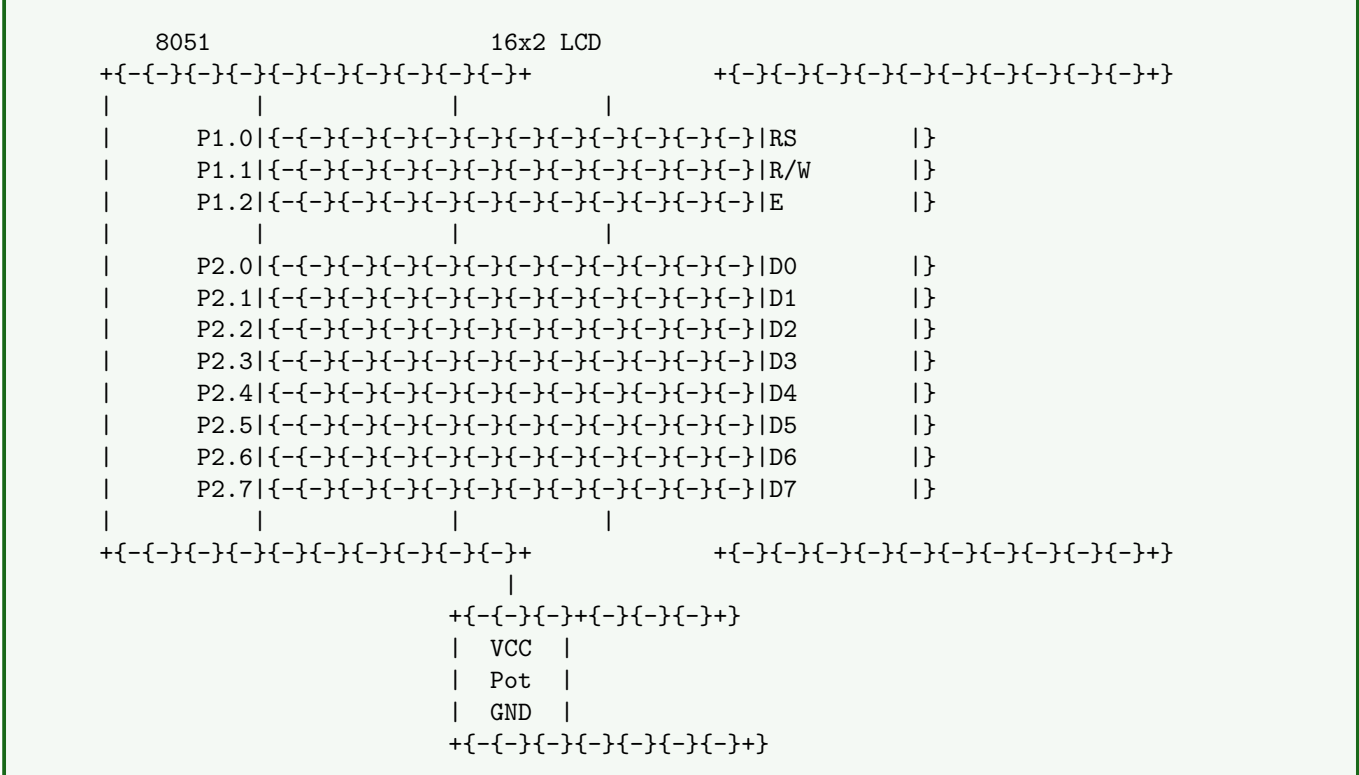
“DRIPS” (Driver, Relay, Input from  $\mu\text{C}$ , Protection diode, Switching)

Question 5(c) [7 marks]

Interface LCD with 8051 microcontroller.

### Solution

Diagram:



**Connections:**

- **Control Lines:**
  - P1.0  $\rightarrow RS(RegisterSelect)$
  - P1.1  $\rightarrow R/W(Read/Write)$
  - P1.2  $\rightarrow E(Enable)$
- **Data Lines:**
  - P2.0-P2.7  $\rightarrow D0 - D7(8-bit databus)$

### Code to Initialize LCD:

```
MOV A, \#38H      ; 2 lines, 5x7 matrix
ACALL COMMAND     ; Send command

MOV A, \#0EH      ; Display ON, cursor ON
ACALL COMMAND     ; Send command

MOV A, \#01H      ; Clear LCD
ACALL COMMAND     ; Send command

MOV A, \#06H      ; Increment cursor
ACALL COMMAND     ; Send command
```

**Mnemonic**

“CIDER-8” (Control lines, Initialize, Data bus, Enable, Register select, 8-bit mode)

**Mnemonic**

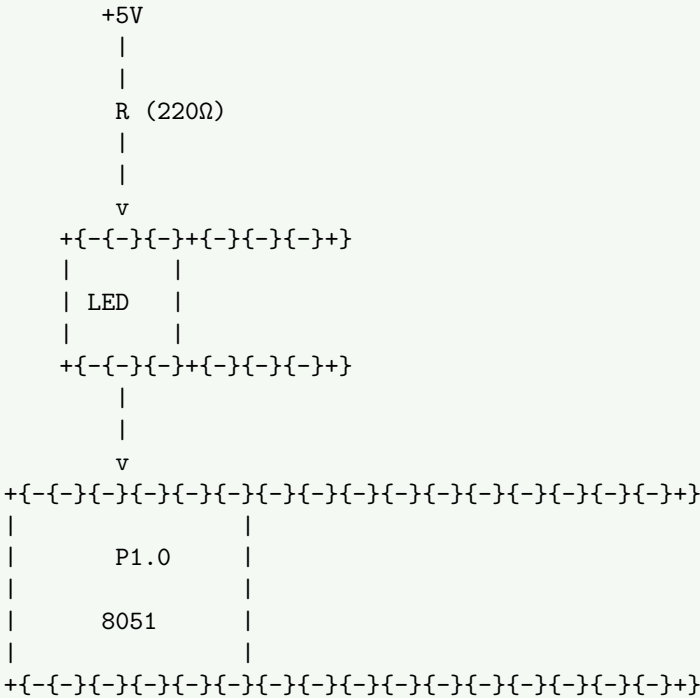
“CIDER-8” (Control lines, Initialize, Data bus, Enable, Register select, 8-bit mode)

Question 5(a OR) [3 marks]

Draw Interfacing of LED with 8051 microcontroller.

## Solution

Diagram:



## Components:

- 8051 microcontroller
- LED
- Current limiting resistor ( $220\Omega$ )
- Power supply

### Working Principle:

- Active-Low configuration: LED ON when pin = 0
- P1.0 drives LED through current limiting resistor
- Maximum current should not exceed 20mA per pin

### Code for LED Blinking:

```

MAIN: CLR P1.0      ; Turn ON LED (active low)
      CALL DELAY    ; Wait
      SETB P1.0     ; Turn OFF LED
      CALL DELAY    ; Wait
      SJMP MAIN      ; Repeat

```

### Mnemonic

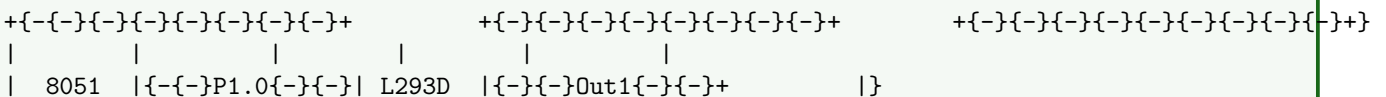
“CIRCLE” (Current limiting Resistor, IO pin, Cathode to LED, LED to Earth/ground)

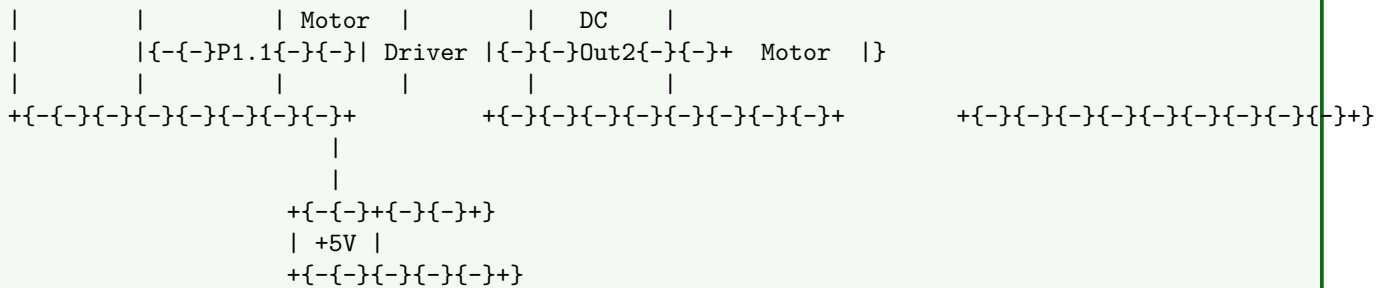
**Question 5(b OR) [4 marks]**

### Interface DC Motor with 8051 microcontroller.

### Solution

**Diagram:**





#### Components:

- 8051 microcontroller
- L293D motor driver IC
- DC motor
- Power supply

#### Control Logic:

P1.0	P1.1	Motor Action
0	0	Stop (Brake)
0	1	Clockwise
1	0	Counter-clockwise
1	1	Stop (Free-running)

#### Code for Motor Control:

```
MOV P1, \#02H ; P1.0=0, P1.1=1 (Clockwise)
CALL DELAY ; Run for some time
MOV P1, \#01H ; P1.0=1, P1.1=0 (Counter-clockwise)}
CALL DELAY ; Run for some time
MOV P1, \#00H ; P1.0=0, P1.1=0 (Stop)
```

#### Mnemonic

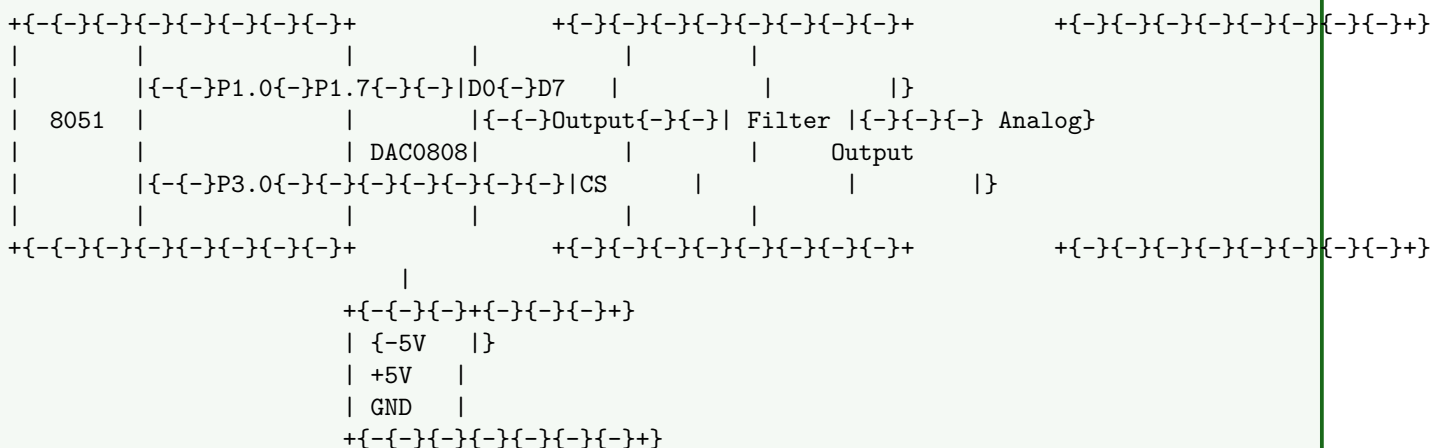
“DICER” (Driver chip, Input from  $\mu C$ , Control logic, Enable motor, Rotation)

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

Interface DAC0808 with 8051 microcontroller.

#### Solution

##### Diagram:



#### Components:

- 8051 microcontroller
- DAC0808 (8-bit digital-to-analog converter)

- Operational amplifier (for output buffering)
- RC filter (for smoothing)
- Reference voltage source

**Connections:**

- P1.0-P1.7  $\rightarrow$  *D0 – D7(8 – bitdigitalinput)*
- P3.0  $\rightarrow$  *CS(ChipSelect)*
- DAC output  $\rightarrow$  *filter*  $\rightarrow$  *finalanalogoutput*

**Sample Code for Ramp Signal Generation:**

```

START: MOV R0, \#00H      ; Start from 0
LOOP:  MOV P1, R0         ; Output value to DAC
        CALL DELAY        ; Wait
        INC R0            ; Increment value
        SJMP LOOP         ; Loop to create ramp

```

**Applications:**

- Waveform generation
- Programmable voltage source
- Motor speed control
- Audio applications

**Mnemonic**

“DACR” (Digital input, Analog output, Conversion, Reference voltage)