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

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

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}
| FETCH              |     | DECODE             |     | EXECUTE            |
| Get instruction    |{-{-}{-}{-}| Interpret opcode  |{-}{-}{-}{-}| Perform operation |}
| from memory        |     | Identify operands |     | Store results      |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}
           \^{                                                 |}
           |                                                 |
           +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}
                        INSTRUCTION CYCLE
```

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:

```
Von{-Neumann:                   Harvard:}
+{-{-}{-}{-}{-}{-}{-}{-}{-}+                    +{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
|         |                      |         |
|   CPU   |{{-}{-}{-}{-}| Memory |     |   CPU   |{-}{-}{-}{-}| Program Memory |}
|         |                      |         |
+{-{-}{-}{-}{-}{-}{-}{-}{-}+                    +{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                                    \^{}
                                    |
                                    v
                               | Data Memory |
```

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:

```
+{-{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+}
| S | Z | {- | AC| {-} | P | {-} | CY|}
+{-{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+}
  7   6   5   4   3   2   1   0
      FLAG REGISTER (F)
```

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:

```
                   +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
A15{-A8 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|          |}
                   |          |{-{-}{-}{-}{-}{-}{-}{-}{-}{-} A15{-}A8 (Higher Address)}
                   |          |
AD7{-AD0 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}| 8085 CPU |{-}{-}{-}{-}+}
                   |          |     |
                   |          |     |    +{-{-}{-}{-}{-}{-}{-}{-}{-}+}
                   +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}| 74LS373|{-}{-}{-}{-}{-} A7{-}A0 (Lower
                        |                |  Latch  |
                        |                +{-{-}{-}{-}{-}{-}{-}{-}{-}+}
                        |                    \^{}
                        |                    |
                     ALE {-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}}
```

- **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    |       | CONTROL UNIT |    |
|  |{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|       |{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|   |}
|  | A (Accum.)  |{{-}{-}{-}{-}{-}| Instruction  |   |}
|  | B,C,D,E,H,L |       | Decoder      |    |
|  | SP, PC      |       |              |    |
|  | Flags       |       | Timing \&     |    |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+       | Control      |   |}
|       \^{               +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+   |}
|       |                   \^{            |}
|       v                   |            |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+             |            |}
|  |    ALU      |{{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+           |}
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+                           |}
|       \^{                              |}
+{-{-}{-}{-}{-}{-}{-}|{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-
       |
+{-{-}{-}{-}{-}{-}{-}|{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-
|       v                                   |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+   |}
|  | Address Bus |{-{-}{-}{-}{-}{-}| Memory \&     |   |}
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+       | I/O Devices  |   |}
|  | Data Bus    |{{-}{-}{-}{-}{-}|             |   |}
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+   |}
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-
```

- **Main Components**:
  - **Registers**: Storage locations (A, B-L, SP, PC, Flags)
  - **ALU**: Performs arithmetic and logical operations
  - **Control Unit**: Generates timing and control signals
  - **Buses**: 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:

```
                    +{-{-}{-}{-}{-}{-}{-}{-}{-}+}
                    |   8085   |
                    |   CPU    |
                    +{-{-}{-}{-}{-}{-}{-}{-}{-}+}
                       |  |  |
           +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+  |  +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
           |              |               |
    +{-{-}{-}{-}{-}{-}v{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}v{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}v{-}{-}{
    | Address Bus|  |  Data Bus   |  | Control Bus |
    | (16{-bit)   |  |  (8{-}bit)   |  | RD,WR,ALE   |}
    +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}{
```

Mnemonic

"ADC" - "Address finds, Data travels, Control coordinates"

Question 2(b OR) [4 marks]

Explain: Program Counter & Stack pointer

Solution

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

Program Counter (PC):

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

Stack Pointer (SP):

- 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)

Diagram:

```
Memory:             PC:              SP:
+{-{-}{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}+      +{-}{-}{-}{-}{-}{-}{-}+}
| Instr 1 |{{-}{-}{-}{-}{-}{-}{-}| 2001H  |      | 3FFEH  |{-}{-}{-}+}
+{-{-}{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}+      +{-}{-}{-}{-}{-}{-}{-}+   |}
| Instr 2 |                             |
+{-{-}{-}{-}{-}{-}{-}{-}+       Stack:                    |}
    ...             ...                 |
+{-{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}+                    |}
| Data 1 |         | Empty  |             |
+{-{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}+                    |}
| Data 2 |{{-}{-}{-}{-}{-}{-}{-}| Data A |{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
+{-{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}+}
```

Mnemonic

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

Question 2(c OR) [7 marks]

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

Solution

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 {{-}{-}{-}{-}{-}{-}{-}|                   |}
                 |                   |
```

```
     ALE {{-}{-}{-}{-}{-}{-}{-}{-}|                   |}
                 |                   |
      S1 {{-}{-}{-}{-}{-}{-}{-}{-}|                   |}
                 |                   |
      S0 {{-}{-}{-}{-}{-}{-}{-}{-}|                   |}
                 |                   |
    A15{-A8 {-}{-}{-}{-}{-}|                   |}
                 |                   |
  AD7{-AD0 {-}{-}{-}{-}{-}|                   |}
                +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
```

Pin Groups:

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)

Question 3(a) [3 marks]

Explain Stack, Stack Pointer and Stack operation.

Solution

Term	Definition
Stack	Memory area used for temporary storage in LIFO order
Stack Pointer	16-bit register that points to the top item in stack
PUSH	Operation that stores data on stack (SP decrements)
POP	Operation that retrieves data from stack (SP increments)

Diagram:

```
Memory:                 Stack Operations:

Before PUSH:            PUSH B (76H):           POP H:
+{-{-}{-}{-}{-}{-}{-}+ {-}{-} 2000H   +{-}{-}{-}{-}{-}{-}{-}+            +{-}{-}{-}{-}{-}{-}{-
|        |              |        |              |        |
+{-{-}{-}{-}{-}{-}{-}+ {-}{-} 1FFFH   +{-}{-}{-}{-}{-}{-}{-}+            +{-}{-}{-}{-}{-}{-}{-
|        |              |   76H  | {{-}{-} SP       |        | {-}{-} SP}
+{-{-}{-}{-}{-}{-}{-}+ {-}{-} 1FFEH   +{-}{-}{-}{-}{-}{-}{-}+            +{-}{-}{-}{-}{-}{-}{-
|        | {{-}{-} SP      |        |              |        |}
+{-{-}{-}{-}{-}{-}{-}+            +{-}{-}{-}{-}{-}{-}{-}+            +{-}{-}{-}{-}{-}{-}{-
```

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:

```
                            +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
Control Bits {-{-}{-}+           |           |}
                |           |  Control |
                +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}|  Logic   |}
                            |          |
                            +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                                 |
                                 v
Clock {-{-}{-}{-}{-}{-}{-} Prescaler {-}{-}{-} TLx {-}{-}{-} THx {-}{-}{-} TFx (Overflow Flag)}
                                 \^{}
External Input {-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|}
                            (Counter Mode)

                  |{-{-}{-}{-}Timer/Counter x{-}{-}{-}|}
```

- **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:

```
                 +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
        VCC {-{-}{-}{-}|                   |{-}{-}{-}{-} GND}
                 |                   |
        P1.0 {{-}{-}|                   |{-}{-} P3.0 (RXD)}
        P1.1 {{-}{-}|                   |{-}{-} P3.1 (TXD)}
        P1.2 {{-}{-}|                   |{-}{-} P3.2 (INT0)}
        P1.3 {{-}{-}|                   |{-}{-} P3.3 (INT1)}
        P1.4 {{-}{-}|                   |{-}{-} P3.4 (T0)}
        P1.5 {{-}{-}|                   |{-}{-} P3.5 (T1)}
        P1.6 {{-}{-}|      8051         |{-}{-} P3.6 (WR)}
        P1.7 {{-}{-}|                   |{-}{-} P3.7 (RD)}
                 |                   |
        RST {-{-}{-}{-}|                   |{-}{-}{-}{-} XTAL2}
                 |                   |
        P2.0 {{-}{-}|                   |{-}{-}{-}{-} XTAL1}
        P2.1 {{-}{-}|                   |{-}{-} P0.0 (AD0)}
        P2.2 {{-}{-}|                   |{-}{-} P0.1 (AD1)}
        P2.3 {{-}{-}|                   |{-}{-} P0.2 (AD2)}
        P2.4 {{-}{-}|                   |{-}{-} P0.3 (AD3)}
        P2.5 {{-}{-}|                   |{-}{-} P0.4 (AD4)}
        P2.6 {{-}{-}|                   |{-}{-} P0.5 (AD5)}
```

```
      P2.7 {{-}{-}|                   |{-}{-} P0.6 (AD6)}
               |                   |{{-}{-} P0.7 (AD7)}
      PSEN {-{-}{-}|                   |}
               |                   |
      ALE {-{-}{-}{-}|                   |{-}{-}{-}{-} EA}
               +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
```

Pin Groups:

1. **Port Pins**:
   - P0 (Port 0): 8-bit bidirectional, multiplexed address/data
   - P1 (Port 1): 8-bit bidirectional I/O
   - P2 (Port 2): 8-bit bidirectional, higher address byte
   - P3 (Port 3): 8-bit bidirectional with alternate functions
2. **Power & Clock**: VCC, GND, XTAL1, XTAL2
3. **Control Signals**:
   - RST: Reset input
   - ALE: Address Latch Enable
   - PSEN: Program Store Enable
   - EA: External Access

Mnemonic

"PORT-CAPS" (Ports 0-3, Clock, Address latch, Program store, Supply)

Question 3(a OR) [3 marks]

Explain Serial communication modes of 8051 microcontroller.

Solution

Mode	Description	Baud Rate	Data Bits
Mode 0	Shift register	Fixed (FOSC/12)	8 bits
Mode 1	8-bit UART	Variable	10 bits (8+start+stop)
Mode 2	9-bit UART	Fixed (FOSC/32 or FOSC/64)	11 bits (9+start+stop)
Mode 3	9-bit UART	Variable	11 bits (9+start+stop)

Key Components:

- **SBUF**: Serial buffer register
- **SCON**: Serial control register
- **P3.0 (RXD)**: Receive pin
- **P3.1 (TXD)**: Transmit pin

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
| Timer 1 |{-{-}{-}{-}| Baud  |{-}{-}{-}{-}|           |     +{-}{-}{-}{-}{-}+}
+{-{-}{-}{-}{-}{-}{-}{-}+     | Rate   |     |  Serial   |{-}{-}{-}{-}| TXD |{-}{-}}
                | Gen    |     |  Control  |     +{-{-}{-}{-}{-}+}
                +{-{-}{-}{-}{-}{-}+     |  Logic    |}
                              |           |     +{-{-}{-}{-}{-}+}
                              |           |{{-}{-}{-}{-}| RXD |{-}{-}}
                              +{-{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}+}
                                    |
                                    v
                                 +{-{-}{-}{-}{-}+}
                                 | SBUF |
                                 +{-{-}{-}{-}{-}+}
```

Mnemonic

"SMART" (Serial Modes Are Rate and Timing dependent)

Question 3(b OR) [4 marks]

Explain Internal RAM Organization of 8051 microcontroller.

Solution

Diagram:

```
8051 Internal RAM (128 bytes):
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+ 7FH}
|                    |
|  General Purpose   |
|       RAM          |
|                    |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+ 30H}
|    Bit{-Addressable|}
|       Area         |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+ 20H}
|                    |
|  Register Banks    |
|    (R0{-R7)        |}
|                    |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+ 00H}
```

Memory Region	Address Range	Description
Register Banks	00H-1FH	Four banks (0-3) of 8 registers each
Bit-Addressable	20H-2FH	16 bytes (128 bits) individually addressable
General Purpose	30H-7FH	Scratch pad RAM for variables
SFR	80H-FFH	Special Function Registers (not in RAM)

Key Features:

- Only one register bank active at a time (selected by PSW bits)
- Each bit in bit-addressable area has its own address (20H.0-2FH.7)
- Stack can be located anywhere in internal RAM

Mnemonic

"RGB-S" (Registers, General purpose, Bit-addressable, SFRs)

Question 3(c OR) [7 marks]

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

Solution

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{
|                     8051 ARCHITECTURE                  |
|                                                        |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}{-}{-
|  |          |     |          |     |             |     |
|  |   CPU    |{{-}{-}{-}|  Timers/ |     |  Interrupts|     |}
```

```
|  |          |     | Counters |     |           |     |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-
|      \^{                                 \^{}            |}
|      |                           |           |
|      v                           v           |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-
|  |          |     |          |     |           |     |
|  | Internal |{{-}{-}{-}|  Serial  |{-}{-}{-}| I/O Ports  |     |}
|  |   RAM    |     |   Port   |     |  P0,P1,P2,P3|     |
|  |          |     |          |     |           |     |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-
|      \^{                                 \^{}            |}
|      |                           |           |
|      v                           v           |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+                 +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+     |}
|  |          |                    |           |     |
|  | Internal |                    | External  |     |
|  |   ROM    |                    | Interface |     |
|  |  (4K)    |                    |           |     |
|  +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+                 +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+     |}
|                                |           |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{
                                |
                                v
                      External Memory \& Devices
```

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
```

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

Diagram:

```
Initially:         After Exchange:
R5: 1010 1100     R5: 1010 0011
    ↑↑↑↑ ↑↑↑↑         ↑↑↑↑ ↑↑↑↑
     |    |            |    |
     |    +{-{-}+         |    |}
     |       |         |    |
     |    +{-{-}v         |    v}
     |    |            |    From R6
     v    v            v
R6: 0011 0011     R6: 0011 1100
```

Mnemonic

"MAMS" (Mask, And, Move, Swap)

Question 4(b) [4 marks]

Write an 8051 Assembly Language Program to blink LED interfaced at port P1.0 at time interval of 1ms.

Solution

```
      ORG 0000H        ; Start at memory location 0000H
MAIN: CPL P1.0         ; Complement P1.0 (toggle LED)
      ACALL DELAY      ; Call delay subroutine
      SJMP MAIN        ; Loop forever

DELAY: MOV R7, \#2      ; Load R7 for outer loop (2)
DELAY1: MOV R6, \#250   ; Load R6 for inner loop (250)
DELAY2: NOP            ; No operation (consume time)
        NOP            ; Additional delay
        DJNZ R6, DELAY2 ; Decrement R6 \& loop until zero
        DJNZ R7, DELAY1 ; Decrement R7 \& loop until zero
        RET            ; Return from subroutine
```

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+          +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
|          |          |          |
| P1.0 = 1 |{-{-}{-}{-}{-}{-}{-}{-}| P1.0 = 0 |}
|          |   1ms    |          |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+          +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
      \^{                  |}
      |                    |
      |                    |
      +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
             1ms
```

Mnemonic

"TCDL" (Toggle, Call, Delay, Loop)

Question 4(c) [7 marks]

List Addressing Modes of 8051 Microcontroller and explain them with at least one example.

Solution

Addressing Mode	Description	Example
Register	Uses registers (R0-R7)	`MOV A, R0` (Move R0 to A)
Direct	Uses direct memory address	`MOV A, 30H` (Move data from 30H to A)
Register Indirect	Uses register as pointer	`MOV A, @R0` (Move data from address in R0 to A)
Immediate	Uses constant data	`MOV A, #25H` (Load A with 25H)
Indexed	Base address + offset	`MOVC A, @A+DPTR` (Code memory access)
Bit	Operates on individual bits	`SETB P1.0` (Set bit 0 of Port 1)
Implied	Implicit operand	`RRC A` (Rotate A right through carry)

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{
| Register         |    | Direct           |    | Indirect         |
| MOV A, R5        |    | MOV A, 40H       |    | MOV A, @R1       |
| +{-{-}{-}+     +{-}{-}{-}+  |    | +{-}{-}{-}+     +{-}{-}{-}+  |    | +{-}{-}{-}+    +{-}{-}{-}+    |
| | A |{{-}{-}{-}{-}| R5|  |    | | A |{-}{-}{-}{-}|40H|  |    | | A |{-}{-}{-}| X |   |}
| +{-{-}{-}+     +{-}{-}{-}+  |    | +{-}{-}{-}+     +{-}{-}{-}+  |    | +{-}{-}{-}+    +{-}{-}{-}+    |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+    +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{
                                                 |             |     |
                                                 |         +{-{-}+{-}{-}+  |}
                                                 |         | R1=X|  |
                                                 |         +{-{-}{-}{-}{-}+  |}
                                                 +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
```

Mnemonic

"RIDDIBM" (Register, Immediate, Direct, Data, Indirect, Bit, iMplied)

Question 4(a OR) [3 marks]

Write an 8051 Assembly Language Program to add the byte in register R2 and R3, put the result in external RAM 2040h (LSB) and 2041h (MSB).

Solution

```
MOV A, R2      ; Move R2 to accumulator
ADD A, R3      ; Add R3 to accumulator
MOV DPTR, \#2040H ; Set DPTR to external RAM address 2040H
MOVX @DPTR, A  ; Store the result (LSB) at 2040H

MOV A, \#00H    ; Clear accumulator
ADDC A, \#00H   ; Add carry flag to accumulator
INC DPTR       ; Increment DPTR to 2041H
MOVX @DPTR, A  ; Store the result (MSB) at 2041H
```

Diagram:

```
  R2    R3           External RAM
+{-{-}{-}{-}+ +{-}{-}{-}{-}+        +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
| 25H| | 45H|         | 2040H: 6A |  (25H + 45H = 6AH)
+{-{-}{-}{-}+ +{-}{-}{-}{-}+        +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
```

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

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}
| ADD A,R0           |     | MUL AB             |     | DIV AB             |
|                    |     |                    |     |                    |
|

A = 25H, R0 = 15H |     |

A = 05H,

B = 03H  |     |

A = 14H,

B = 05H  |

|

A = 25H + 15H     |     | B:A = 05H  03H   |     |

A = 14H  05H     |

|

A = 3AH           |     |

B = 00H,

A = 0FH  |     |

A = 04H,

B = 00H  |

+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+     +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}
```

Mnemonic

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

Question 5(a) [3 marks]

List Applications of microcontroller in various fields.

Solution

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

Diagram:

```
                   +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                   | Microcontroller   |
                   | Applications      |
                   +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                            |
         +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-
         |                  |                  |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-
| Consumer      |  | Industrial     |  | Communication |
| Electronics   |  | Automation     |  | Systems       |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-
         |                  |                  |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-
| Automotive    |  | Medical        |  | Home          |
| Systems       |  | Devices        |  | Automation    |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+  +{-
```

Mnemonic

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

Question 5(b) [4 marks]

Interface Relay with 8051 microcontroller.

Solution

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}+        +{-}{-}{-}{-}{-}{-}{-}{-}+        +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-
|        |         |        |         |            |
|        |         |        |         |            |
|  8051  |{-{-}{-}P1.0{-}{-}| Driver |{-}{-}{-}(+){-}{-}{-}| Relay Coil |}
|        |         | ULN2003|    |    |            |
|        |         |        |    |    |            |
+{-{-}{-}{-}{-}{-}{-}{-}+        +{-}{-}{-}{-}{-}{-}{-}{-}+    |    +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-
                                 |          |
                                 |          |
                        +{-{-}{-}{-}{-}{-}{-}{-}+         |}
                        |                    |
                        | +5V                |
                        +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                                 |
                            Protection
                              Diode
```

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

Mnemonic

"DRIPS" (Driver, Relay, Input from µC, Protection diode, Switching)

Question 5(c) [7 marks]

Interface LCD with 8051 microcontroller.

Solution

Diagram:

```
       8051                    16x2 LCD
    +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+             +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
    |          |             |          |
    |      P1.0|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|RS        |}
    |      P1.1|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|R/W       |}
    |      P1.2|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|E         |}
    |          |             |          |
    |      P2.0|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D0        |}
    |      P2.1|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D1        |}
    |      P2.2|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D2        |}
    |      P2.3|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D3        |}
    |      P2.4|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D4        |}
    |      P2.5|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D5        |}
    |      P2.6|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D6        |}
    |      P2.7|{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}|D7        |}
    |          |             |          |
    +{-{-}{-}{-}{-}{-}{-}{-}{-}{-}+             +{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                              |
                          +{-{-}{-}+{-}{-}{-}+}
                          |  VCC  |
                          |  Pot  |
                          |  GND  |
                          +{-{-}{-}{-}{-}{-}{-}+}
```

Connections:

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

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)

Question 5(a OR) [3 marks]

Draw Interfacing of LED with 8051 microcontroller.

Solution

Diagram:

```
       +5V
        |
        |
        R (220Ω)
        |
        |
        v
    +{-{-}{-}+{-}{-}{-}+}
    |       |
    | LED   |
    |       |
    +{-{-}{-}+{-}{-}{-}+}
        |
        |
        v
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
|                |
|       P1.0     |
|                |
|      8051      |
|                |
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
```

Components:

- 8051 microcontroller
- LED
- Current limiting resistor (220Ω)
- 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:

```
+{-{-}{-}{-}{-}{-}{-}{-}+         +{-}{-}{-}{-}{-}{-}{-}{-}+        +{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
|        |         |        |        |         |
|  8051  |{-{-}P1.0{-}{-}| L293D  |{-}{-}Out1{-}{-}+         |}
```

```
|        |         | Motor  |        |   DC    |
|        |{-{-}P1.1{-}{-}| Driver |{-}{-}Out2{-}{-}+  Motor  |}
|        |         |        |        |         |
+{-{-}{-}{-}{-}{-}{-}{-}+        +{-}{-}{-}{-}{-}{-}{-}{-}+       +{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                      |
                      |
                   +{-{-}+{-}{-}+}
                   | +5V |
                   +{-{-}{-}{-}{-}+}
```

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 µC, Control logic, Enable motor, Rotation)

Question 5(c OR) [7 marks]

Interface DAC0808 with 8051 microcontroller.

Solution

Diagram:

```
+{-{-}{-}{-}{-}{-}{-}{-}+             +{-}{-}{-}{-}{-}{-}{-}{-}+         +{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
|        |              |        |          |        |
|        |{-{-}P1.0{-}P1.7{-}{-}|D0{-}D7   |          |        |}
|  8051  |              |        |{-{-}Output{-}{-}| Filter |{-}{-}{-} Analog}
|        |              | DAC0808|          |        |     Output
|        |{-{-}P3.0{-}{-}{-}{-}{-}{-}{-}{-}|CS      |          |        |}
|        |              |        |          |        |
+{-{-}{-}{-}{-}{-}{-}{-}+             +{-}{-}{-}{-}{-}{-}{-}{-}+         +{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
                           |
                        +{-{-}{-}+{-}{-}{-}+}
                        | {-5V   |}
                        | +5V   |
                        | GND   |
                        +{-{-}{-}{-}{-}{-}{-}+}
```

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 $\rightarrow D0 - D7 (8 - bit digital input)$
- P3.0 $\rightarrow CS (Chip Select)$
- DAC output $\rightarrow filter \rightarrow final analog output$

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)