Microprocessor and Microcontroller (4341101) - Summer 2023 Solution

Milav Dabgar

June 15, 2023

Question 1 [a marks]

3 Compare Microprocessor and Microcontroller.

Solution

Answer:

Table 1. Microprocessor vs Microcontroller

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 marks]

4 Compare RISC and CISC.

Solution

Answer:

Table 2. RISC vs CISC

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 marks]

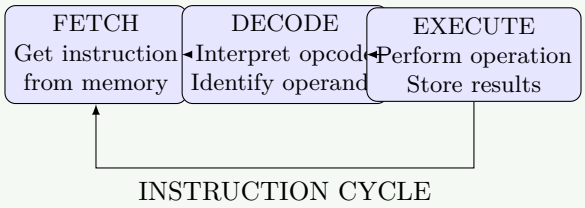
7 Define: Microprocessor, Operand, Instruction Cycle, Opcode, ALU, Machine Cycle, T-State

Solution

Answer:

Table 3. Definitions

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)



Mnemonic

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

OR

Question 1 [c marks]

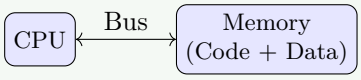
7 Compare Von-Neumann and Harvard architecture.

Solution

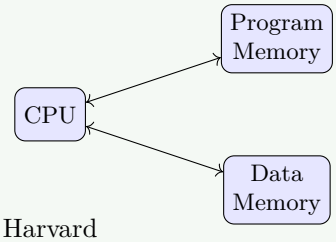
Answer:

Table 4. Von-Neumann vs Harvard Architecture

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



Von-Neumann



Harvard

Mnemonic

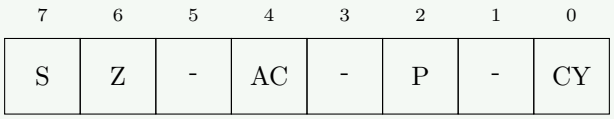
"Harvard Has Separate Streets" (Harvard Has Separate memory paths)

Question 2 [a marks]

3 Draw Flag Register of 8085 microprocessor & explain it.

Solution

Answer:



FLAG REGISTER (F)

Table 5. Flag Register Bits

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

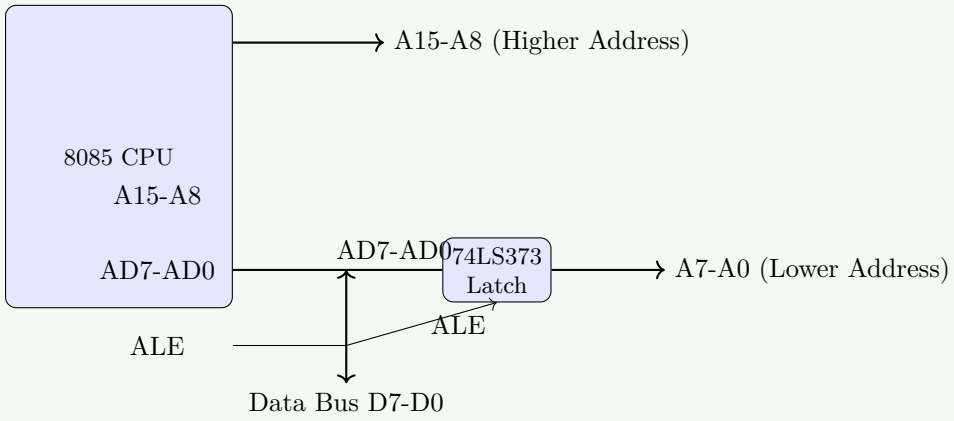
"Smart Zombies Always Prefer Candy" (Sign, Zero, Auxiliary, Parity, Carry)

Question 2 [b marks]

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

Solution

Answer:



- **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

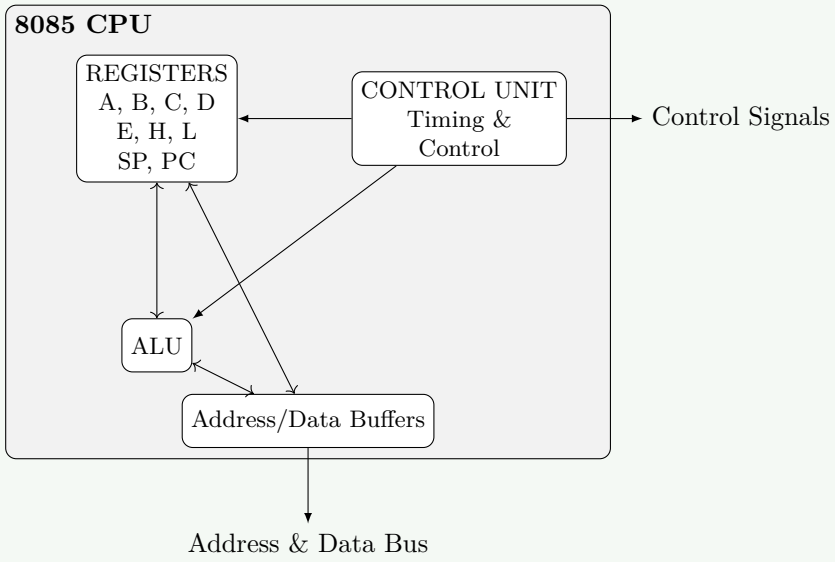
"ALE Latches, Data Follows" (Address Latch Enable captures address first, then data)

Question 2 [c marks]

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

Solution

Answer:



- **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)

OR

Question 2 [a marks]

3 Explain Bus Organization of 8085 microprocessor.

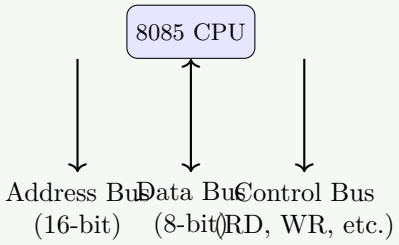
Solution

Answer:

Table 6. 8085 Bus Organization

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



Mnemonic

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

OR

Question 2 [b marks]

4 Explain: Program Counter & Stack pointer

Solution

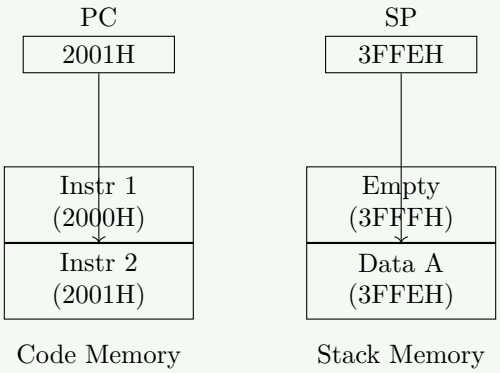
Answer:

Table 7. PC vs SP

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)



Mnemonic

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

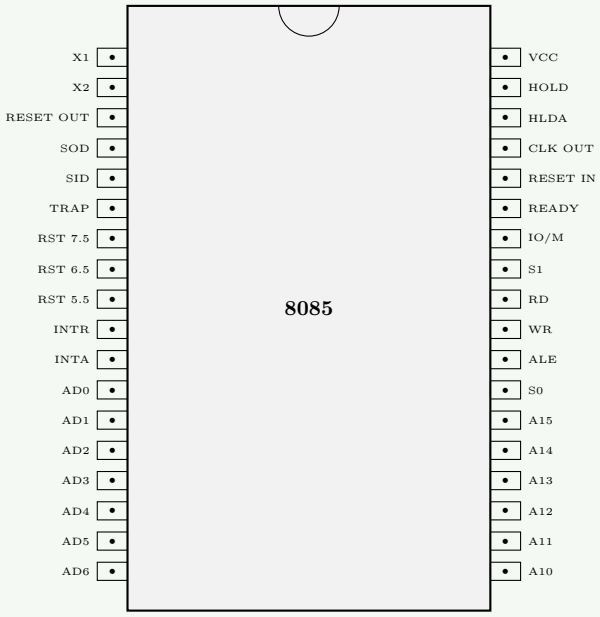
OR

Question 2 [c marks]

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

Solution

Answer:



Pin Groups:

1. **Power & Clock**: Vcc, GND, X1, X2, CLK
2. **Address/Data**: A8-A15, AD0-AD7 (multiplexed)
3. **Control**: ALE, $\overline{RD}$, $\overline{WR}$, $IO/\overline{M}$
4. **Interrupt**: INTR, $\overline{INTA}$, 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 marks]

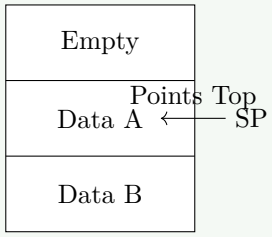
3 Explain Stack, Stack Pointer and Stack operation.

Solution

Answer:

Table 8. Stack Concepts

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)



Mnemonic

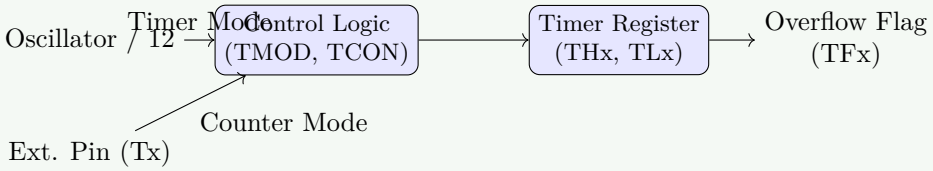
"LIFO Saves Push-Pop" (Last-In-First-Out with Push and Pop operations)

Question 3 [b marks]

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

Solution

Answer:



- **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

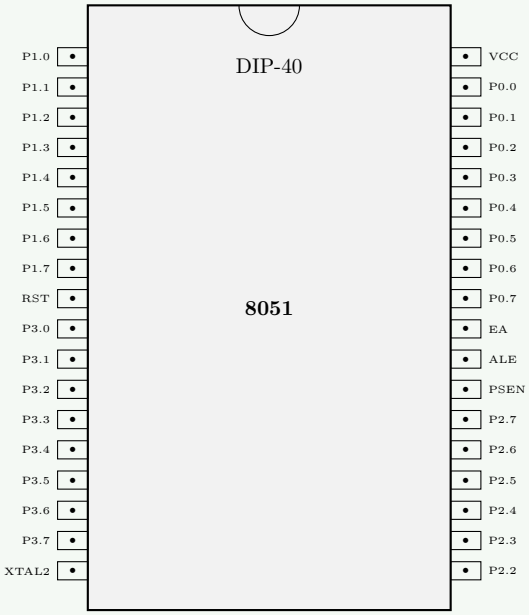
"TIME-C" (Timer Input, Mode select, External count)

Question 3 [c marks]

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

Solution

Answer:



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
   - $\overline{PSEN}$: Program Store Enable
   - $\overline{EA}$: External Access

Mnemonic

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

OR

Question 3 [a marks]

3 Explain Serial communication modes of 8051 microcontroller.

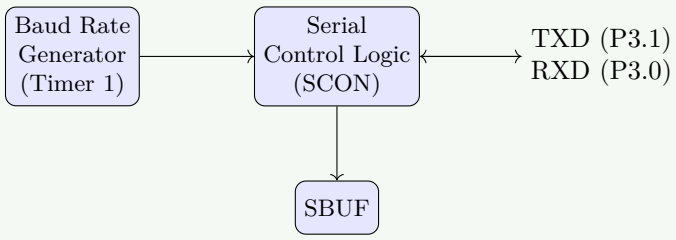
Solution

Answer:

Table 9. Serial Modes

Mode	Description	Baud Rate	Data Bits
Mode 0	Shift register	Fixed ($F_{OSC}/12$)	8 bits
Mode 1	8-bit UART	Variable	10 bits (8+start+stop)
Mode 2	9-bit UART	Fixed ($F_{OSC}/32$ or $F_{OSC}/64$)	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



Mnemonic

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

OR

Question 3 [b marks]

4 Explain Internal RAM Organization of 8051 microcontroller.

Solution

Answer:

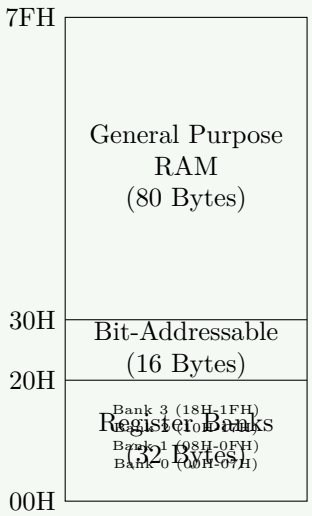


Table 10. Internal RAM Map

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)

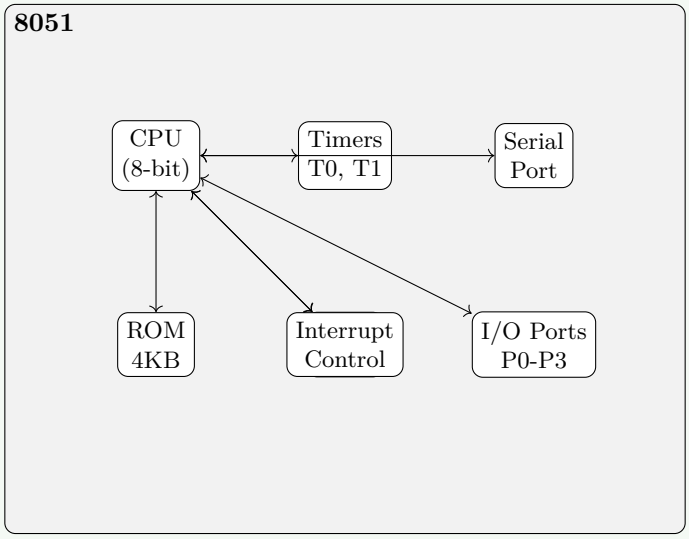
OR

Question 3 [c marks]

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

Solution

Answer:



External Bus Interface

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 marks]

3 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

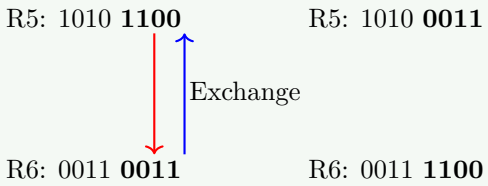
Answer:

```
      ; 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
```



Mnemonic

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

Question 4 [b marks]

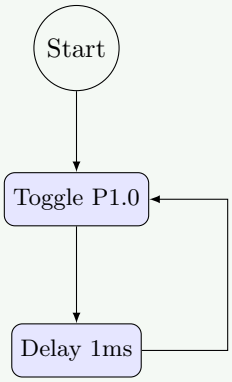
4 Write an 8051 Assembly Language Program to blink LED interfaced at port P1.0 at time interval of 1ms.

Solution

Answer:

```
      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
```

```
5
6  DELAY: MOV R7, #2       ; Load R7 for outer loop (2)
7  DELAY1: MOV R6, #250    ; Load R6 for inner loop (250)
8  DELAY2: NOP             ; No operation (consume time)
9          NOP             ; Additional delay
10         DJNZ R6, DELAY2 ; Decrement R6 & loop until zero
11         DJNZ R7, DELAY1 ; Decrement R7 & loop until zero
12         RET             ; Return from subroutine
```



Mnemonic

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

Question 4 [c marks]

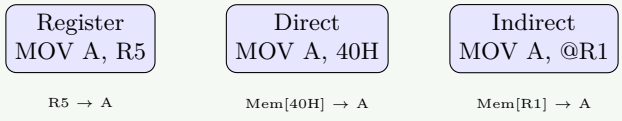
7 List Addressing Modes of 8051 Microcontroller and explain them with at least one example.

Solution

Answer:

Table 11. 8051 Addressing Modes

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)



Mnemonic

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

OR

Question 4 [a marks]

3 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

Answer:

```
      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
```

- Uses `MOVX` for external RAM access
- Uses `ADDC` to handle potential carry (result > 255)

Mnemonic

"MASIM" (Move, Add, Store, Increment, Move again)

OR

Question 4 [b marks]

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

Solution

Answer:

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

Calculation:

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

Mnemonic

"LOON-5" (LOOp Nested for 5ms)

OR

Question 4 [c marks]

7 Explain any seven Arithmetic Instructions with example for 8051 Microcontroller.

Solution

Answer:

Table 12. Arithmetic Instructions

Instruction	Function	Example	Flag Affected
ADD A,src	Add source to A	`ADD A,R0`	C, OV, AC
ADDC A,src	Add source + carry to A	`ADDC A,#25H`	C, OV, AC
SUBB A,src	Subtract source + borrow from A	`SUBB A,@R1`	C, OV, AC
INC	Increment by 1	`INC R3`	None
DEC	Decrement by 1	`DEC A`	None
MUL AB	Multiply A and B	`MUL AB`	C, OV
DIV AB	Divide A by B	`DIV AB`	C, OV

Mnemonic

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

Question 5 [a marks]

3 List Applications of microcontroller in various fields.

Solution

Answer:

Table 13. Microcontroller Applications

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

Mnemonic

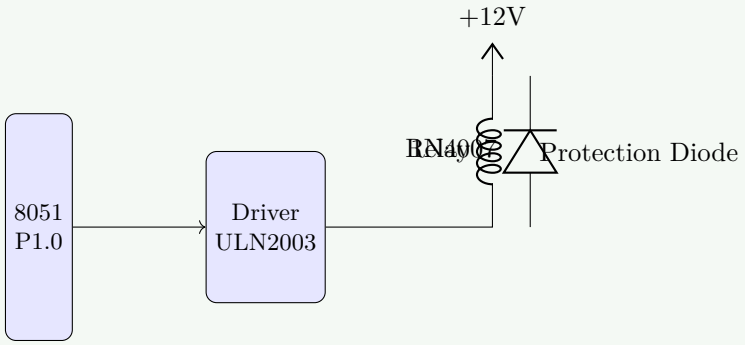
"CHAIM-MA" (Consumer, Home, Automotive, Industrial, Medical, Mobile, Aerospace)

Question 5 [b marks]

4 Interface Relay with 8051 microcontroller.

Solution

Answer:



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

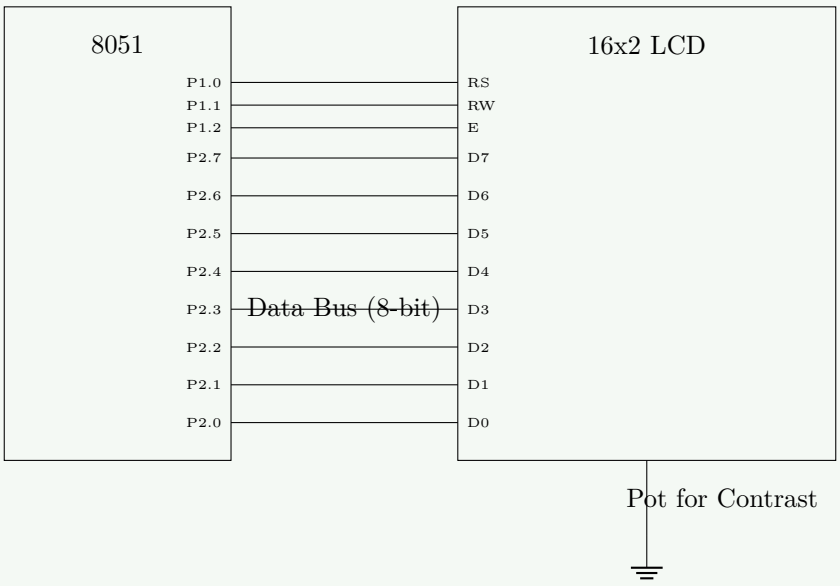
"DRIPS" (Driver, Relay, Input from µC, Protection diode, Switching)

Question 5 [c marks]

7 Interface LCD with 8051 microcontroller.

Solution

Answer:



Connections:
- **Control Lines**: RS (P1.0), RW (P1.1), E (P1.2)
- **Data Lines**: D0-D7 connected to Port 2

Initialization Code:

```
MOV A, #38H      ; 2 lines, 5x7 matrix
ACALL COMMAND
MOV A, #0EH      ; Display ON, cursor ON
ACALL COMMAND
MOV A, #01H      ; Clear LCD
ACALL COMMAND
MOV A, #06H      ; Increment cursor
ACALL COMMAND
```

Mnemonic

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

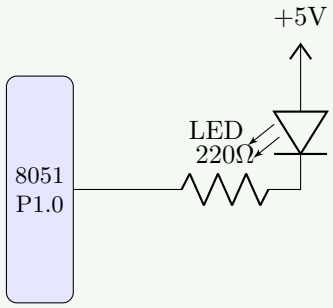
OR

Question 5 [a marks]

3 Draw Interfacing of LED with 8051 microcontroller.

Solution

Answer:



Working Principle:

- Active-Low configuration: LED ON when pin = 0
- Current limiting resistor (220Ω) protects LED
- Max current approx 10-20mA

Mnemonic

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

OR

Question 5 [b marks]

4 Interface DC Motor with 8051 microcontroller.

Solution

Answer:

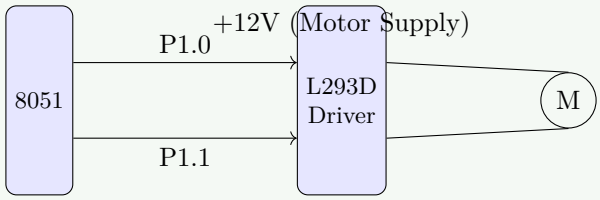


Table 14. Motor Control Logic

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

Mnemonic

"DICER" (Driver chip, Input from µC, Control logic, Enable motor, Rotation)

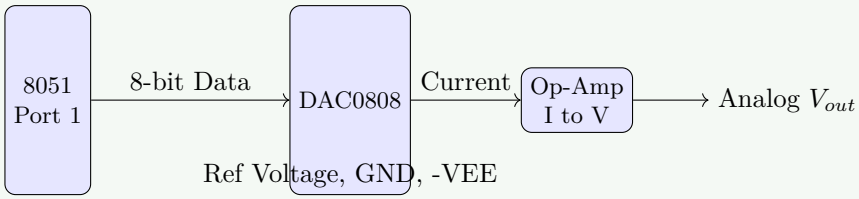
OR

Question 5 [c marks]

7 Interface DAC0808 with 8051 microcontroller.

Solution

Answer:



Connections:
- P1.0-P1.7 → D0-D7 inputs of DAC
- DAC output is current, converted to voltage by Op-Amp

Applications: Waveform generation, Motor speed control

Mnemonic

"DACR" (Digital input, Analog output, Conversion, Reference voltage)