

# Microprocessor and Microcontroller (4341101) - Winter 2024 Solution

Milav Dabgar

January 15, 2024

## Question 1 [a marks]

3 List Common features of Microcontrollers.

### Solution

Answer:

Table 1. Common Features

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

### Mnemonic

“CRITICS” (CPU-ROM-I/O-Timers-Interrupts-Communication-Serial)

## Question 1 [b marks]

4 Explain the functions of ALU.

### Mnemonic

“ALFS” (Arithmetic-Logic-Flags-Status)

## Question 1 [c marks]

7 Define: Memory, Operand, Instruction Cycle, Opcode, CU, Machine Cycle, CISC

### Solution

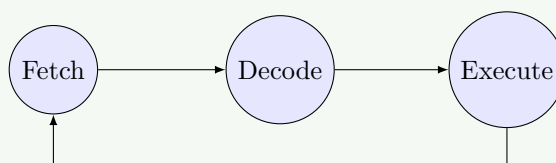
Answer:

Table 3. Definitions

Term	Definition
<b>Memory</b>	Storage unit that holds data and instructions (Valid Address Space).
<b>Operand</b>	Data value or address used in an operation.
<b>Instruction Cycle</b>	Complete process of fetching and executing an instruction.
<b>Opcode</b>	Operation code that specifies the instruction type (e.g., MOV, ADD).
<b>CU</b>	Control Unit that coordinates processor operations.
<b>Machine Cycle</b>	Basic operation cycle consisting of T-states required to access memory or I/O.
<b>CISC</b>	Complex Instruction Set Computer with rich instruction set.

- **Memory:** Organized array of storage cells with unique addresses.
- **Operand:** Data elements that instructions operate upon.
- **Instruction Cycle:** Fetch-decode-execute sequence for each instruction.
- **Opcode:** Binary code that tells processor what operation to perform.

**Instruction Cycle Diagram:**



### Mnemonic

“MO-ICO-MC” (Memory-Operand-Instruction-Control-Operation-Machine-Complex)

OR

## Question 1 [c marks]

7 i) Define: Microprocessor. ii) Compare Von-Neumann and Harvard architecture.

### Solution

**Answer:**

i) **Microprocessor Definition:** An integrated circuit containing the CPU functionality of a computer, capable of fetching, decoding, and executing instructions with ALU and control circuitry on a single chip.

ii) **Von-Neumann vs Harvard Architecture:**

**Table 4.** Comparison

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

**Diagrams:**



**Mnemonic**

“Harvard Has Separate Spaces”

**Question 2 [a marks]**

3 Explain various Registers of 8085 microprocessor.

**Solution**

Answer:

**Table 5.** 8085 Registers

Register	Size	Function
<b>Accumulator (A)</b>	8-bit	Main register for arithmetic & logic operations.
<b>General Purpose</b>	8-bit	B, C, D, E, H, L (Temporary data storage). Pairs: BC, DE, HL.
<b>Program Counter (PC)</b>	16-bit	Points to the address of the next instruction.
<b>Stack Pointer (SP)</b>	16-bit	Points to the top of the stack memory.
<b>Flag Register</b>	8-bit	Stores status flags (Z, S, P, CY, AC).

**Mnemonic**

“AGSF” (Accumulator-General-Stack-Flags)

**Question 2 [b marks]**

4 Explain Fetching, Decoding and Execution of Instruction.

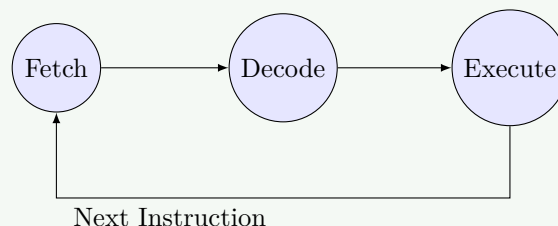
**Solution**

Answer:

**Table 6.** Instruction Phases

Phase	Activity	Hardware Involved
<b>Fetching</b>	Processor gets instruction from memory address pointed by PC.	PC, Address bus, Memory, Data Bus
<b>Decoding</b>	Processor identifies the operation and operands.	Instruction Register, Decoder, Control Unit
<b>Execution</b>	Processor performs the specified operation (Arithmetic, Logic, Move).	ALU, Registers, Data bus

Diagram:



- **Fetching:** PC contents are placed on address bus. Memory reads op-code and places it on data bus. Op-code moves to Instruction Register (IR).

- **Decoding:** Control unit interprets the op-code to determine what action to take.
- **Execution:** Control unit generates signals to perform the action (e.g., enable ALU, read/write memory).

### Mnemonic

“FDE” (Fetch-Decode-Execute)

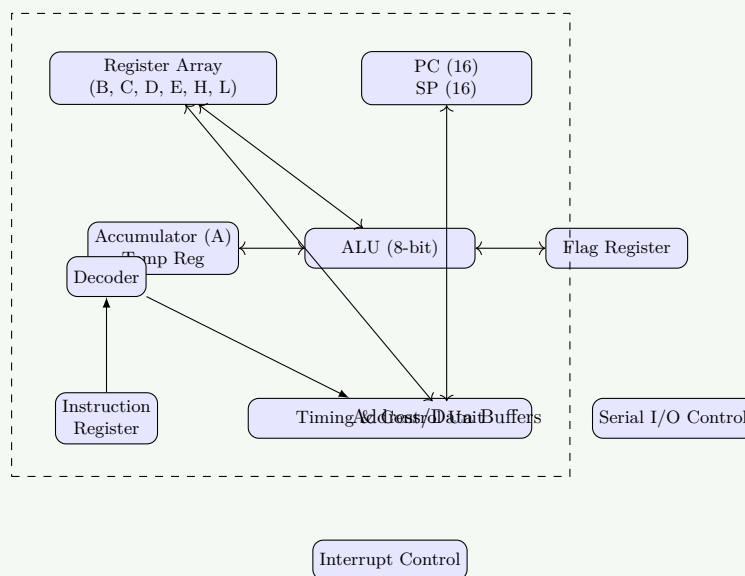
## Question 2 [c marks]

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

### Solution

Answer:

Diagram:



- **ALU:** Performs arithmetic and logical operations in 8-bit.
- **Register Array:** Includes temporary registers (B, C, D, E, H, L) and special purpose registers (PC, SP).
- **Control Unit:** Generates timing and control signals (RD, WR, ALE, etc.).
- **Instruction Register & Decoder:** Fetches op-code, decodes it, and executes.
- **Interrupt Control:** Handles hardware interrupts (INTR, RST, TRAP).
- **Serial I/O:** Handles SID (Serial Input Data) and SOD (Serial Output Data).

### Mnemonic

“RAID” (Registers-ALU-Instructions-Decoders)

OR

## Question 2 [a marks]

3 Compare Microprocessor & Microcontroller.

**Solution****Answer:****Table 7.** Comparison

Feature	Microprocessor	Microcontroller
<b>Design</b>	CPU only (External components needed)	CPU + Peripherals (System on Chip)
<b>Memory</b>	External RAM/ROM	Internal RAM/ROM
<b>I/O ports</b>	Limited/None (External interfacing)	Many built-in I/O ports
<b>Applications</b>	General purpose computing (Laptops, PCs)	Embedded systems (Washing machines, Remotes)
<b>Cost</b>	Higher system cost	Lower system cost
<b>Example</b>	Intel 8085, 8086, Core i7	Intel 8051, AVR, PIC

**Mnemonic**

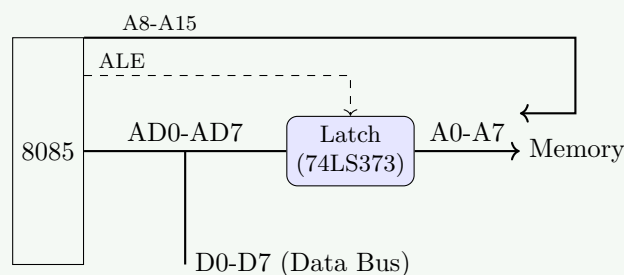
“Micro-P Processes, Micro-C Controls”

**OR****Question 2 [b marks]****4 Explain De-multiplexing of Address and Data buses for 8085 Microprocessor.****Solution**

**Answer:** Address lines A0-A7 and Data lines D0-D7 are multiplexed as AD0-AD7 to save pins. They must be separated (demultiplexed) to talk to memory.

**Steps:**

1. **ALE High:** Microprocessor sends ALE=1. AD0-AD7 carries address. Latch is enabled.
2. **Latch:** External Latch (e.g., 74LS373) captures the address (A0-A7).
3. **ALE Low:** ALE=0. Latch holds the address. AD0-AD7 lines are now free to carry Data (D0-D7).

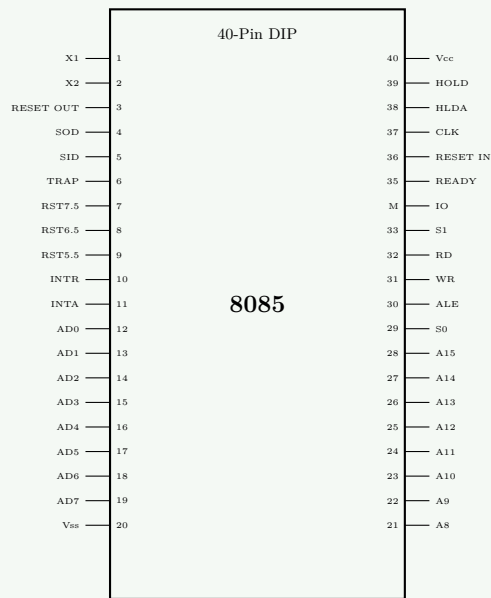
**Diagram:****Mnemonic**

“ALAD” (ALE-Latches-Address-Data)

**OR****Question 2 [c marks]****7 Describe Pin diagram of 8085 microprocessor with the help of neat diagram.**

## Solution

Answer:



- **Address/Data:** AD0-AD7 (Multiplexed), A8-A15 (High Address).
- **Control & Status:** ALE (Address Latch Enable), RD (Read), WR (Write), IO/M (Select IO or Memory), S0, S1 (Status).
- **Interrupts:** INTR, INTA, RST 5.5, RST 6.5, RST 7.5, TRAP.
- **Serial I/O:** SID (Input), SOD (Output).
- **DMA:** HOLD, HLDA.
- **Power/Clock:** Vcc (+5V), Vss (GND), X1, X2 (Crystal).

## Mnemonic

“ACID-PS” (Address-Control-Interrupt-DMA-Power-Serial)

## Question 3 [a marks]

3 Explain interrupts of 8051 microcontroller.

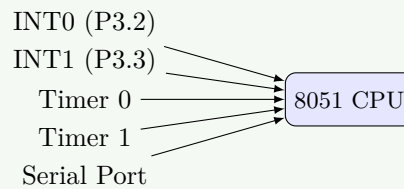
## Solution

Answer:

Table 8. 8051 Interrupts

Interrupt	Vector	Priority	Source
External 0	0003H	1 (Highest)	Pin INT0 (P3.2)
Timer 0	000BH	2	Timer 0 overflow (TF0)
External 1	0013H	3	Pin INT1 (P3.3)
Timer 1	001BH	4	Timer 1 overflow (TF1)
Serial	0023H	5 (Lowest)	RI or TI (Serial Port)

Diagram:



### Mnemonic

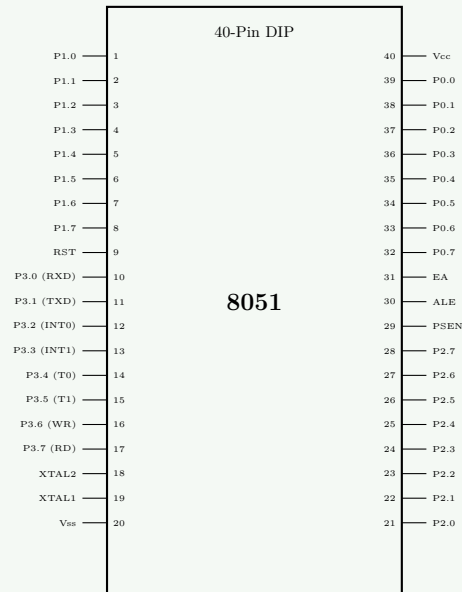
“ETTES” (External-Timer-Timer-External-Serial)

## Question 3 [b marks]

4 Draw Pin diagram of 8051 microcontroller.

### Solution

Answer:



- **P0.0-P0.7:** Port 0 (Address/Data AD0-AD7).
- **P1.0-P1.7:** Port 1 (I/O).
- **P2.0-P2.7:** Port 2 (Address A8-A15).
- **P3.0-P3.7:** Port 3 (Special functions like RD, WR, INT).
- **Power:** Vcc (40), Vss (20).
- **Control:** RST, ALE, PSEN, EA.

### Mnemonic

“PORT 0123”

## Question 3 [c marks]

7 Explain Internal RAM Organization of 8051 microcontroller.

- **Register Banks (00H-1FH):** 4 Banks of 8 registers each. Selected by RS0, RS1 bits in PSW.

- **Bit Addressable (20H-2FH):** 16 bytes where each bit can be individually accessed (set/cleared).
- **General Purpose (30H-7FH):** 80 bytes used for data storage and stack.
- **SFRs (80H-FFH):** Mapped in upper 128 bytes (Direct Access only).

OR

### Question 3 [a marks]

3 List SFRs with their addresses.

#### Solution

Answer:

Table 9. Special Function Registers

SFR	Address	Function
P0	80H	Port 0
SP	81H	Stack Pointer
DPH:DPL	83H:82H	Data Pointer (16-bit)
PCON	87H	Power Control
TCON	88H	Timer Control
TMOD	89H	Timer Mode
TL0	8AH	Timer 0 Low Byte
TL1	8BH	Timer 1 Low Byte
TH0	8CH	Timer 0 High Byte
TH1	8DH	Timer 1 High Byte
P1	90H	Port 1
SCON	98H	Serial Control
SBUF	99H	Serial Buffer
P2	A0H	Port 2
IE	A8H	Interrupt Enable
P3	B0H	Port 3
IP	B8H	Interrupt Priority
PSW	D0H	Program Status Word
ACC (A)	E0H	Accumulator
B	F0H	B Register

#### Mnemonic

“PDPT-SP” (Ports-Data-Program-Timers-Serial-Prioritized)

OR

### Question 3 [b marks]

4 Explain Timers/Counters logic diagram of 8051 microcontroller.

OR

### Question 3 [c marks]

7 Explain block diagram of 8051 microcontroller.

!

### Question 4 [a marks]

3 Write an 8051 Assembly Language Program to add two bytes of data and store result in R4 register.

#### Solution

##### Answer:

b0 A, 25H ; Load first value (25H) into Accumulator MOV R3, 18H ; Load second value (18H) into R3 ADD A, R3 ; Add R3 to Accumulator (A = A + R3) MOV R4, A ; Store result from A into R4

##### Steps:

1. Load first operand in A.
2. Load second operand in R3.
3. Perform ADD operation.
4. Move result to R4.

#### Mnemonic

“LLAS” (Load-Load-Add-Store)

### Question 4 [b marks]

4 Write an 8051 Assembly Language Program to OR the contents of Port-1 and Port-2 then put the result in external RAM location 0200H.

#### Solution

##### Answer:

```
1  MOV A, P1      ; Read Port 1 into Accumulator
2  ORL A, P2      ; OR Accumulator with Port 2
3  MOV DPTR, #0200H ; Load Data Pointer with 0200H
4  MOVX @DPTR, A   ; Move content of A to External RAM at 0200H
```

##### Steps:

1. Read P1 to A.
2. Logical OR with P2.
3. Set DPTR to address.
4. Write A to external memory using MOVX.

#### Mnemonic

“PORT” (Port-OR-Register-Transfer)

### Question 4 [c marks]

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

**Solution****Answer:****Table 10.** Addressing Modes

Mode	Example	Description
<b>Immediate</b>	MOV A, #25H	Data is provided directly in the instruction (#).
<b>Register</b>	MOV A, R0	Data is in one of the registers (R0-R7).
<b>Direct</b>	MOV A, 30H	Address of the data is given directly.
<b>Indirect</b>	MOV A, @R0	Address is stored in a register (@R0 or @R1).
<b>Indexed</b>	MOVC A, @A+DPTR	Access data from Code memory (Base + Offset).
<b>Bit</b>	SETB P1.3	Operation on a single bit.
<b>Relative</b>	SJMP LABEL	Jump to a relative address (offset).

**Mnemonic**

“I’M DIRBI” (Immediate-Register-Direct-Indirect-Relative-Bit-Indexed)

**OR****Question 4 [a marks]****3** Explain following instructions: (i) DJNZ (ii) POP (iii) CJNE.**Solution****Answer:**

- **DJNZ (Decrement and Jump if Not Zero):**
  - Syntax: DJNZ Rn, rel
  - Operation: Decrement register Rn. If result is not 0, jump to relative address. Used for loops.
  - Example: DJNZ R7, LOOP
- **POP:**
  - Syntax: POP direct
  - Operation: Pop data from Stack to direct memory address. SP is decremented.
  - Example: POP 30H
- **CJNE (Compare and Jump if Not Equal):**
  - Syntax: CJNE A, #data, rel
  - Operation: Compare A with data. If not equal, jump. Sets Carry flag if A < data.
  - Example: CJNE A, #25H, NEXT

**Mnemonic**

“DPC” (Decrement-Pop-Compare)

**OR****Question 4 [b marks]****4** For 8051 Microcontroller with a crystal frequency of 12 MHz, generate a delay of 4ms.

## Solution

### Answer: Calculation:

- Crystal Freq = 12 MHz.
- Machine Cycle Freq = 12 MHz / 12 = 1 MHz.
- Time for 1 Machine Cycle =  $1/1\text{MHz} = 1\mu\text{s}$ .
- Required Delay = 4 ms = 4000  $\mu\text{s}$  = 4000 Machine Cycles.

### Program:

```

1      MOV R7, #16          ; Outer Loop: 16
2  DELAY1:
3      MOV R6, #250         ; Inner Loop: 250
4  DELAY2:
5      NOP                  ; 1 Cycle
6      NOP                  ; 1 Cycle
7      DJNZ R6, DELAY2      ; 2 Cycles. Total Inner = 4 * 250 = 1000
8      DJNZ R7, DELAY1      ; Total = 16 * 1000 = 16000 cycles
9      RET

```

Note: The above calculation in MDX ( $16 \times 250 \times 4$ ) gives 16000 cycles = 16ms. To get 4ms, Outer loop should be 4.

### Corrected for 4ms:

```

1      MOV R7, #08          ; Outer Loop
2  DELAY_LOOP:
3      MOV R6, #250         ; Inner Loop (250 x 2 = 500us)
4      DJNZ R6, $           ; 2 cycles per loop
5      DJNZ R7, DELAY_LOOP ; 8 x 500 = 4000 us = 4ms
6      RET

```

## Mnemonic

“LNDD” (Load-NOP-Decrement-Decrement)

OR

## Question 4 [c marks]

7 Explain any seven Logical instructions with example for 8051 Microcontroller.

## Solution

### Answer:

Table 11. Logical Instructions

Instruction	Example	Operation
ANL	ANL A, #0FH	Logical AND. Masking bits.
ORL	ORL P1, #80H	Logical OR. Setting bits.
XRL	XRL A, R0	Logical XOR. Toggling bits.
CLR	CLR A	Clear Accumulator (A=00H).
CPL	CPL A	Complement Accumulator (Invert bits).
RL	RL A	Rotate Left (circular shift).
RR	RR A	Rotate Right (circular shift).

**Mnemonic**

“A-OX-CCR” (AND-OR-XOR-Clear-Complement-Rotate)

**Question 5 [a marks]**

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

**Solution**

**Answer:**

- **Industrial:** Motor control, Automation, PLCs.
- **Medical:** Patient monitoring, X-ray machines.
- **Consumer:** Washing machines, Microwave ovens, Toys.
- **Automotive:** ECU, ABS, Airbags.
- **Communication:** Modems, Routers, Mobile phones.
- **Security:** CCTV, Biometric systems.

**Mnemonic**

“I-MACS” (Industrial-Medical-Automotive-Consumer-Security)

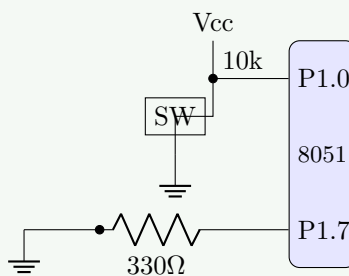
**Question 5 [b marks]**

**4 Interface Push button Switch and LED with 8051 microcontroller.**

**Solution**

**Answer:**

**Circuit Diagram:**



**Program:**

```

1  AGAIN:
2      JB P1.0, LED_OFF ; If P1.0 is High (Not pressed), Jump
3      SETB P1.7         ; If Low (Pressed), Turn ON LED
4      SJMP AGAIN
5  LED_OFF:
6      CLR P1.7          ; Turn OFF LED
7      SJMP AGAIN

```

**Mnemonic**

“PLIC” (Push-LED-Input-Control)

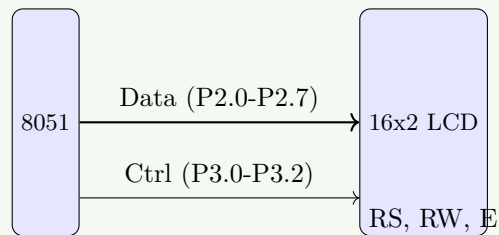
## Question 5 [c marks]

7 Interface LCD with microcontroller and write a program to display "HELLO".

### Solution

Answer:

Circuit Diagram:



Program:

```

1  MOV A, #38H      ; Init LCD 2 lines, 5x7
2  ACALL CMD
3  MOV A, #0EH      ; Display ON, Cursor ON
4  ACALL CMD
5  MOV A, #'H'      ; Data 'H'
6  ACALL DISP
7  MOV A, #'E'
8  ACALL DISP
9  MOV A, #'L'
10 ACALL DISP
11 MOV A, #'L'
12 ACALL DISP
13 MOV A, #'O'
14 ACALL DISP
15 SJMP $

16
17 CMD: MOV P2, A    ; Send Command
18     CLR P3.0      ; RS=0
19     CLR P3.1      ; RW=0
20     SETB P3.2     ; E=1
21     ACALL DELAY
22     CLR P3.2     ; E=0
23     RET
24
25 DISP: MOV P2, A   ; Send Data
26     SETB P3.0     ; RS=1
27     CLR P3.1      ; RW=0
28     SETB P3.2     ; E=1
29     ACALL DELAY
30     CLR P3.2
31     RET

```

### Mnemonic

“DICE” (Data-Instruction-Control-Enable)

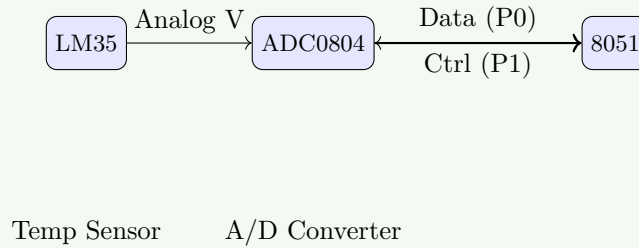
OR

## Question 5 [a marks]

3 Draw Interfacing of LM35 with 8051 microcontroller.

### Solution

Answer:



### Mnemonic

“TAC” (Temperature-Analog-Convert)

OR

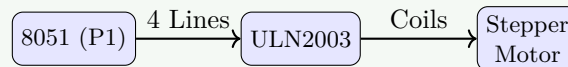
## Question 5 [b marks]

4 Interface Stepper motor with 8051 microcontroller.

### Solution

Answer:

Circuit:



Logic:

- Send sequence: 0x08, 0x0C, 0x04, 0x06, 0x02, 0x03, 0x01, 0x09.
- Delay between steps determines speed.

### Mnemonic

“PDCS” (Port-Driver-Current-Sequence)

OR

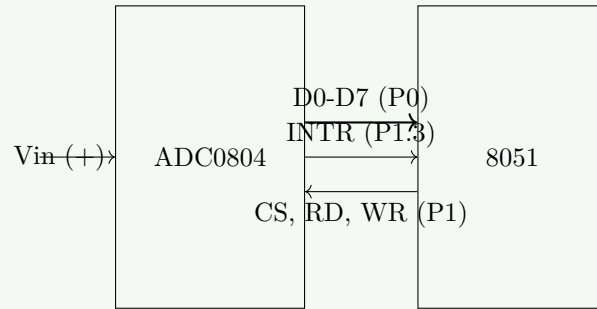
## Question 5 [c marks]

7 Interface ADC0804 with 8051 microcontroller.

### Solution

**Answer:**

**Interfacing:**



**Control Signals:**

1. **CS=0**: Select Chip.
2. **WR=0 then 1**: Start Conversion.
3. **Wait for INTR=0**: Conversion Complete.
4. **RD=0**: Read Data.

### Mnemonic

“CRIW” (Control-Read-Interrupt-Write)

