

# Microprocessor & Microcontroller Systems (1333202) - Summer 2025 Solution

Milav Dabgar

May 13, 2025

## પ્રશ્ન 1(A) [3 ગુણ]

8085 નું બસ ઓર્ગેનાઈઝેશન દોરો.

જવાબ

આકૃતિ 1. 8085 Bus Organization

બસના પ્રકારો:

- **Address Bus:** મેમરી એડ્રેસિંગ માટે 16-bit એકદિશીય બસ.
- **Data Bus:** ડેટા ટ્રાન્સફર માટે 8-bit દ્વિદિશીય બસ.
- **Control Bus:** RD, WR, ALE, IO/M જેવા કંટ્રોલ સિગ્નલ્સ.

મેમરી ટ્રીક

ADC - Address, Data, Control

## પ્રશ્ન 1(B) [4 ગુણ]

માઈક્રોપ્રોસેસરની માઈક્રોકંટ્રોલર સાથે સરખામણી કરો.

જવાબ

કોષ્ટક 1. માઈક્રોપ્રોસેસર vs માઈક્રોકંટ્રોલર

લક્ષણ	માઈક્રોપ્રોસેસર	માઈક્રોકંટ્રોલર
આર્કિટેક્ચર	બાહ્ય ઘટકોની જરૂર.	એક જ ચિપ પર બધા ઘટકો.
મેમરી	બાહ્ય RAM/ROM જરૂરી.	આંતરિક RAM/ROM ઉપલબ્ધ.
કિંમત	વધુ સિસ્ટમ કોસ્ટ.	ઓછી સિસ્ટમ કોસ્ટ.
પાવર	વધુ પાવર વપરાશ.	ઓછો પાવર વપરાશ.
સાઈઝ	મોટું સિસ્ટમ સાઈઝ.	કોમ્પેક્ટ સિસ્ટમ.
ઉપયોગ	સામાન્ય હેતુ કમ્પ્યુટિંગ.	એમ્બેડેડ કંટ્રોલ એપ્લિકેશનો.

મુખ્ય મુદ્દાઓ:

- માઈક્રોપ્રોસેસર: માત્ર CPU, બાહ્ય સપોર્ટ ચિપ્સ જરૂરી.
- માઈક્રોકંટ્રોલર: ચિપ પર સંપૂર્ણ કમ્પ્યુટર સિસ્ટમ.

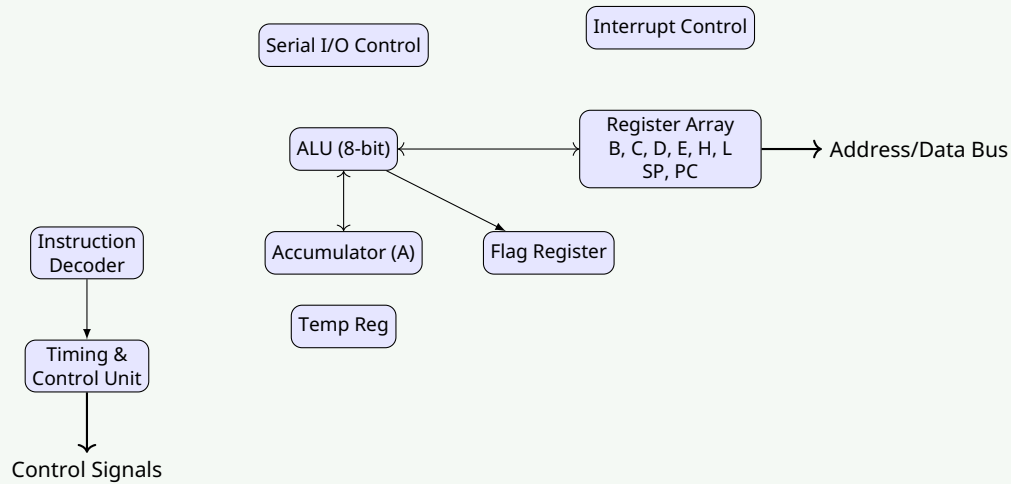
### મેમરી ટ્રીક

MICRO - Memory Internal, Compact, Reduced cost, Optimized

## પ્રશ્ન 1(C) [7 ગુણ]

8085 માઈક્રોપ્રોસેસરના દરેક બ્લોક દોરો અને સમજાવો.

### જવાબ



આકૃતિ 2. 8085 Block Diagram

### બ્લોકના કાર્યો:

- **ALU:** અંકગણિત અને તાર્કિક ઓપરેશન કરે છે.
- **Accumulator:** ડેટા પ્રોસેસિંગ માટે પ્રાથમિક કામકાજ રજિસ્ટર.
- **Register Array:** B, C, D, E, H, L સામાન્ય હેતુ રજિસ્ટરો.
- **Program Counter:** આગળના instruction નું address ધરાવે છે.
- **Stack Pointer:** મેમરીમાં stack ના ટોપને પોઈન્ટ કરે છે.
- **Control Unit:** પ્રોસેસરના એકંદર ઓપરેશનને કંટ્રોલ કરે છે.

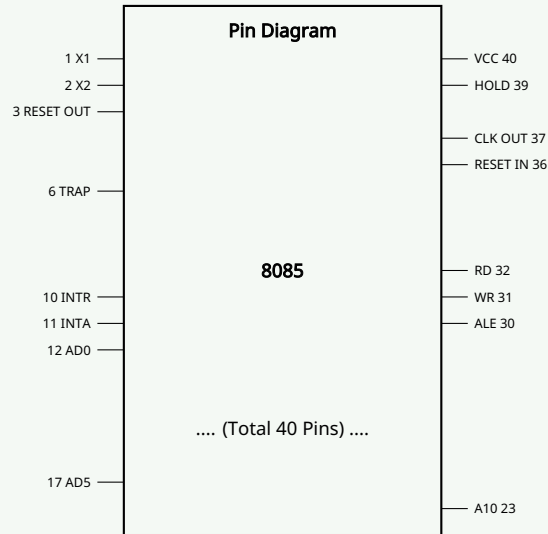
### મેમરી ટ્રીક

APRIL - ALU, Program counter, Registers, Instruction decoder, Logic control

## પ્રશ્ન 1(C) OR [7 ગુણ]

8085 માઈક્રોપ્રોસેસરનો પીન ડાયાગ્રામ દોરો અને કોઈ પણ 4 પીન સમજાવો.

## જવાબ



આકૃતિ 3. 8085 Pin Diagram

## પીન સમજાવટ:

- **ALE (Pin 30):** Address Latch Enable - multiplexed bus પર address અને data અલગ કરે છે.
- **RD (Pin 32):** Read control signal - active low, read operation દર્શાવે છે.
- **WR (Pin 31):** Write control signal - active low, write operation દર્શાવે છે.
- **RESET (Pin 36):** Reset input - low થાય ત્યારે processor initialize કરે છે.

## મેમરી ટ્રીક

ARWA - ALE, Read, Write, rAset

## પ્રશ્ન 2(A) [3 ગુણ]

વ્યાખ્યા આપો: (1) Opcode (2) Operand

## જવાબ

## વ્યાખ્યાઓ:

- **Opcode:** Operation Code - કરવાનું operation સ્પષ્ટ કરે છે (ADD, MOV, JMP).
- **Operand:** જે ડેટા અથવા address પર operation કરવાનું છે.

## ઉદાહરણ:

MOV A, B

```

| | |
| | +-- Operand 2 (Source)
| +-- Operand 1 (Destination)
+-- Opcode

```

## મેમરી ટ્રીક

OO - Operation + Operand

## પ્રશ્ન 2(B) [4 ગુણ]

RISC અને CISC વચ્ચે તફાવત આપો.

જવાબ

કોષ્ટક 2. RISC vs CISC

લક્ષણ	RISC	CISC
Instructions	સરળ, fixed format.	જટિલ, variable format.
Execution	Single cycle execution.	Multiple cycle execution.
Addressing	થોડા addressing modes.	ઘણા addressing modes.
Memory	Load/Store architecture.	Memory-to-memory operations.
Compiler	જટિલ compiler design.	સરળ compiler design.

મુખ્ય મુદ્દાઓ:

- RISC: Reduced Instruction Set Computer - સરળ, ઝડપી.
- CISC: Complex Instruction Set Computer - feature rich.

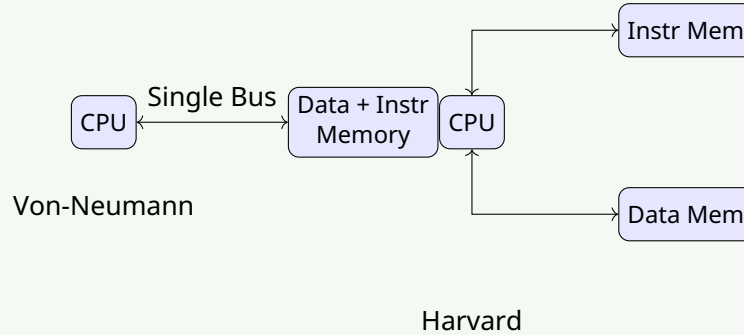
મેમરી ટ્રીક

RISC is SLIM - Simple, Load-store, Instruction reduced, Memory efficient

## પ્રશ્ન 2(C) [7 ગુણ]

Von-Neumann અને Harvard Architecture વચ્ચે તફાવત આપો.

જવાબ



આકૃતિ 4. Von-Neumann vs Harvard

કોષ્ટક 3. Architecture Comparison

લક્ષણ	Von-Neumann	Harvard
Memory	data અને instructions માટે single memory.	data અને instructions માટે અલગ memory.
Bus Structure	Single bus system.	Dual bus system.
Access	data અને instructions ને sequential access.	simultaneous access શક્ય.
Cost	ઓછી કિંમત.	વધુ કિંમત.
Speed	bus conflicts કારણે ધીમું.	parallel access કારણે ઝડપી.
Examples	8085, સામાન્ય computers.	8051, DSP processors.

## મેમરી ટ્રીક

VH - Von has one bus, Harvard has two

## પ્રશ્ન 2(A) OR [3 ગુણ]

વ્યાખ્યા આપો: (1) T-State (2) Instruction Cycle (3) Machine Cycle

## જવાબ

## વ્યાખ્યાઓ:

- **T-State:** Time state - મૂળભૂત timing unit, એક clock period.
- **Instruction Cycle:** એક instruction નું સંપૂર્ણ execution.
- **Machine Cycle:** એક memory operation માટે જરૂરી T-states નું જૂથ.

## સંબંધ:

- Instruction Cycle = Multiple Machine Cycles
- Machine Cycle = Multiple T-States (3-6 T-states)

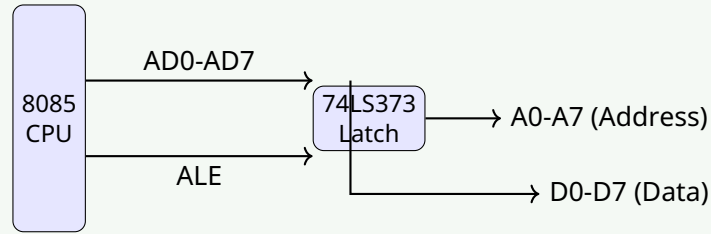
## મેમરી ટ્રીક

TIM - T-state, Instruction cycle, Machine cycle

## પ્રશ્ન 2(B) OR [4 ગુણ]

8085 ના Address અને Data Bus નું De-Multiplexing સમજાવો.

## જવાબ



આકૃતિ 5. Demultiplexing Circuit

## પ્રક્રિયા:

- **Step 1:** T1 દરમિયાન, AD0-AD7 માં lower 8-bit address હોય છે.
- **Step 2:** ALE high થાય છે, external latch માં address latch થાય છે.
- **Step 3:** બાકીના T-states માટે AD0-AD7 data bus બને છે.

## જરૂરી ઘટકો:

- **74LS373:** Address latching માટે Octal latch IC.
- **ALE:** Timing માટે Address Latch Enable signal.

## મેમરી ટ્રીક

LAD - Latch Address with Data separation

## પ્રશ્ન 2(C) OR [7 ગુણ]

8085 નો Flag Register દોરો અને સમજાવો.

જવાબ

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

આકૃતિ 6. Flag Register Format

Flag વર્ણન:

- CY (D0): Carry flag - carry આવે ત્યારે set થાય છે.
- P (D2): Parity flag - even parity માટે set થાય છે.
- AC (D4): Auxiliary carry - BCD operations માટે set થાય છે.
- Z (D6): Zero flag - પરિણામ zero હોય ત્યારે set થાય છે.
- S (D7): Sign flag - પરિણામ negative હોય ત્યારે set થાય છે.

Flag Operations:

- Conditional Jumps: Flag status પર આધારિત (JZ, JC, JP).
- Arithmetic Results: ALU operations પછી automatically update થાય છે.

મેમરી ટ્રીક

SZAPC - Sign, Zero, Auxiliary, Parity, Carry

## પ્રશ્ન 3(A) [3 ગુણ]

SFR એટલે શું? કોઈ પણ ત્રણ SFR નું લિસ્ટ આપો.

જવાબ

SFR વ્યાખ્યા: Special Function Register - માઈક્રોકંટ્રોલરમાં ચોક્કસ કાર્યો માટે સમર્પિત રજિસ્ટરો.

ત્રણ SFRs:

- ACC (E0H): Accumulator register.
- PSW (D0H): Program Status Word.
- SP (81H): Stack Pointer register.

લાક્ષણિકતાઓ:

- Address Range: Internal RAM માં 80H થી FFH.
- Bit Addressable: કેટલાક SFRs વ્યક્તિગત bit access આપે છે.
- Function Specific: દરેકનું સમર્પિત હાર્ડવેર કાર્ય છે.

મેમરી ટ્રીક

APS - ACC, PSW, Stack Pointer

## પ્રશ્ન 3(B) [4 ગુણ]

Program Counter (PC) અને Data Pointer (DPTR) રજિસ્ટર સમજાવો.

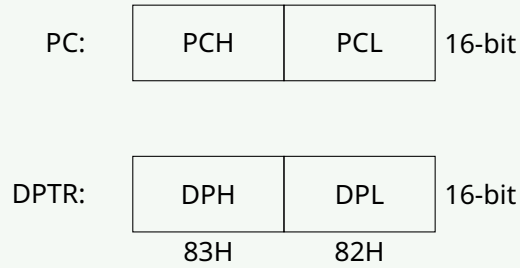
## જવાબ

**Program Counter (PC):**

- **Size:** 16-bit રજિસ્ટર.
- **Function:** execute થનાર આગામી instruction નું address ધરાવે છે.
- **Auto-increment:** Instruction fetch પછી આપમેળે increment થાય છે.
- **Range:** 0000H થી FFFFH.

**Data Pointer (DPTR):**

- **Size:** 16-bit રજિસ્ટર (DPH + DPL).
- **Function:** External data memory locations ને point કરે છે.
- **Usage:** External memory access માટે MOVX instructions સાથે વપરાય છે.
- **Components:** DPH (83H) અને DPL (82H).



આકૃતિ 7. PC અને DPTR Registers

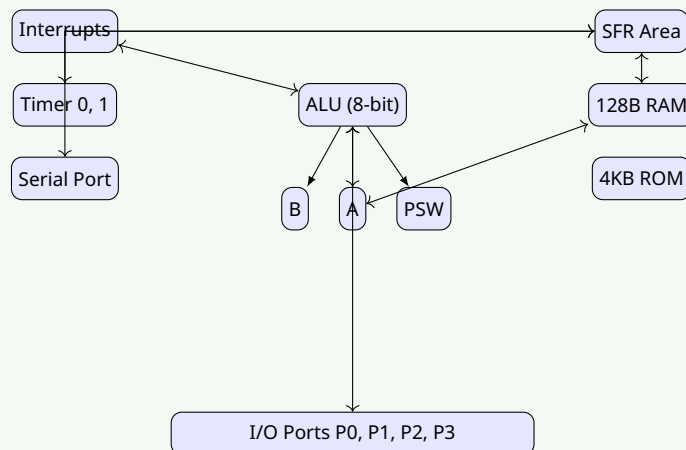
## મેમરી ટ્રીક

PD - PC Points to Program, DPTR Points to Data

## પ્રશ્ન 3(C) [7 ગુણ]

8051 નું આર્કિટેક્ચર દોરો અને સમજાવો.

## જવાબ



આકૃતિ 8. 8051 Architecture Block Diagram

**આર્કિટેક્ચર ઘટકો:**

- **CPU:** Accumulator અને B register સાથે 8-bit ALU.
- **Memory:** 4KB internal ROM, 128B internal RAM.
- **I/O Ports:** ચાર 8-bit bidirectional ports (P0-P3).
- **Timers:** બે 16-bit timers/counters (T0, T1).

- **Serial Port:** કમ્યુનિકેશન માટે Full duplex UART.
- **Interrupts:** Priority levels સાથે 5 interrupt sources.

ખાસ વિશેષતાઓ:

- **Boolean Processor:** Bit manipulation ક્ષમતાઓ.
- **Addressing Modes:** 8 વિવિધ addressing modes.
- **Power Management:** Idle અને power-down modes.

#### મેમરી ટ્રીક

MIPTIS - Memory, I/O, Processor, Timers, Interrupts, Serial

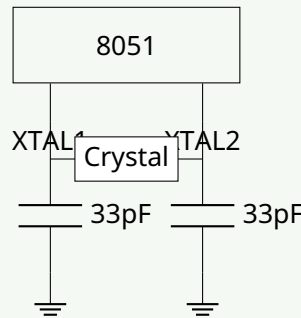
### પ્રશ્ન 3(A) OR [3 ગુણ]

8051 ની નીચેની પીન સમજાવો: (1) ALE (2) PSEN (3) XTAL1 & XTAL2

#### જવાબ

પીન કાર્યો:

- **ALE (Pin 30):** Address Latch Enable
  - Lower address byte latch કરવા માટે output pulse.
  - Oscillator frequency ના 1/6 પર active high signal.
- **PSEN (Pin 29):** Program Store Enable
  - External program memory read માટે active low output.
  - External EPROM ની OE pin સાથે જોડાયેલ.
- **XTAL1 & XTAL2 (Pins 19, 18):** Crystal જોડાણો
  - Clock generation માટે external crystal જોડો.
  - સામાન્ય frequency: 11.0592 MHz અથવા 12 MHz.



આકૃતિ 9. Crystal Oscillator Connection

#### મેમરી ટ્રીક

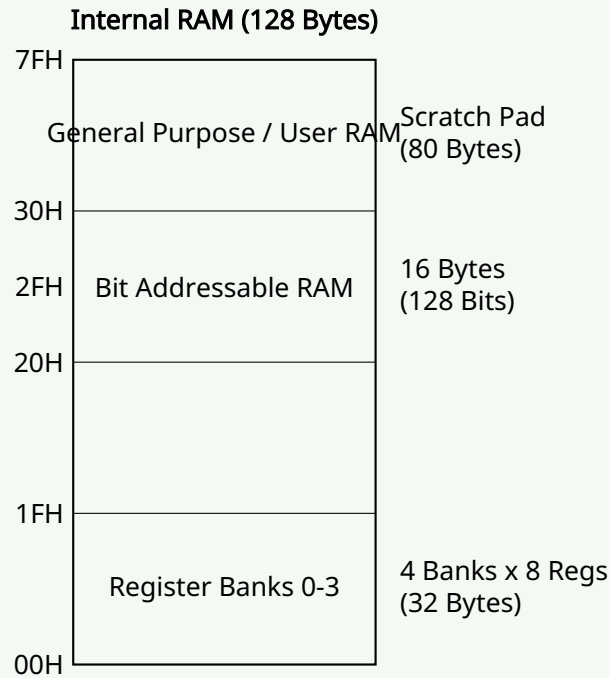
APX - ALE latches Address, PSEN enables Program, XTAL generates clock

### પ્રશ્ન 3(B) OR [4 ગુણ]

8051 માઈક્રોકંટ્રોલરનું ઈન્ટરનલ રેમ ઓર્ગેનાઈઝેશન સમજાવો.



## જવાબ



આકૃતિ 10. 8051 Internal RAM Organization

**RAM વિભાગો:**

- **Register Banks:** 4 banks  $\times$  8 registers દરેક (00H-1FH).
- **Bit Addressable:** વ્યક્તિગત bit access સાથે 16 bytes (20H-2FH).
- **General Purpose:** User data માટે 80 bytes (30H-7FH).
- **Stack Area:** સામાન્ય રીતે 08H થી ઉપર તરફ હોય છે.

**Addressing:**

- **Direct:** વાસ્તવિક address નો ઉપયોગ કરીને (MOV 30H, A).
- **Indirect:** Register pointer નો ઉપયોગ કરીને (MOV @R0, A).

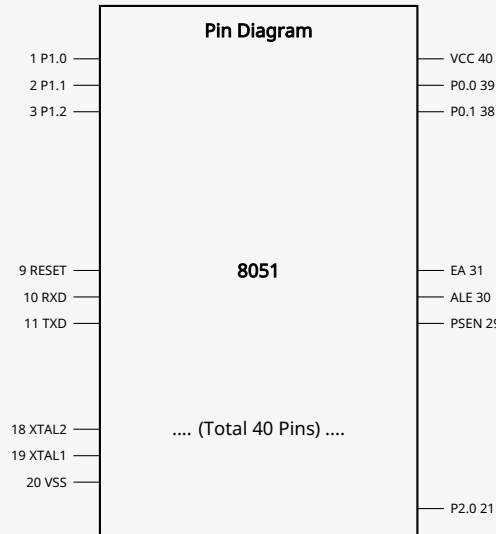
**મેમરી ટ્રીક**

RBGS - Register banks, Bit addressable, General purpose, Stack

**પ્રશ્ન 3(C) OR [7 ગુણ]**

8051 નો પીન ડાયાગ્રામ દોરો અને સમજાવો.

## જવાબ



આકૃતિ 11. 8051 Pin Diagram

## પીન સમજાવટ:

- **RESET (Pin 9):** Reset input - Active high, microcontroller initialize કરે છે.
- **EA/VPP (Pin 31):** External Access - Program memory selection control કરે છે.
- **P0 (Pins 32-39):** Port 0 - External memory માટે Multiplexed address/data bus.
- **P2 (Pins 21-28):** Port 2 - External memory માટે High-order address bus.

## મેમરી ટ્રીક

REPP - REset, External Access, Port 0, Port 2

## પ્રશ્ન 4(A) [3 ગુણ]

R0 રજિસ્ટરમાં સ્ટોર થયેલ ડેટાનો R1 રજિસ્ટરમાં સ્ટોર થયેલ ડેટા સાથે ગુણાકાર કરવાનો પ્રોગ્રામ લખો. પરિણામ R2 રજિસ્ટર (LSB) અને R3 રજિસ્ટર (MSB) માં સ્ટોર કરો.

## જવાબ

## Listing 1. Multiplication Program

```

1  ORG 0000H
2  MOV R0, #05H ; Load first number
3  MOV R1, #03H ; Load second number
4  MOV A, R0 ; R0 ને accumulator માં move કરો
5  MOV B, R1 ; R1 ને B register માં move કરો
6  MUL AB ; A અને B નો ગુણાકાર કરો
7  MOV R2, A ; LSB ને R2 માં store કરો
8  MOV R3, B ; MSB ને R3 માં store કરો
9  END

```

## Program Flow:

- Load operands R0 અને R1 માં.
- Multiplication માટે A અને B registers માં transfer કરો.
- MUL AB instruction execute કરો.
- 16-bit result (A=LSB, B=MSB) store કરો.

## Result Storage:

- R2: Product ની lower 8 bits.
- R3: Product ની higher 8 bits.

### મેમરી ટ્રીક

LTSE - Load, Transfer, multiply, Store result

## પ્રશ્ન 4(B) [4 ગુણ]

Data Transfer Instructions નું વિસ્તર આપો અને કોઈ પણ બે યોગ્ય ઉદાહરણ સાથે સમજાવો.

### જવાબ

#### Data Transfer Instructions:

Instruction	Function
MOV	Registers/memory વચ્ચે data move કરે છે.
MOVX	External memory માંથી data move કરે છે.
MOVC	Code byte ને accumulator માં move કરે છે.
PUSH	Stack પર data push કરે છે.
POP	Stack માંથી data pop કરે છે.
XCH	Accumulator ને register સાથે exchange કરે છે.
XCHD	Lower nibble exchange કરે છે.

#### વિગતવાર ઉદાહરણો:

##### 1. MOV Instruction:

- 1 `MOV A, #50H` ; Immediate data 50H accumulator માં load કરો
- 2 `MOV R0, A` ; Accumulator content ને R0 માં copy કરો
- 3 `MOV 30H, A` ; Accumulator content ને address 30H પર store કરો

##### 2. PUSH/POP Instructions:

- 1 `PUSH ACC` ; Accumulator ને stack પર push કરો
- 2 `PUSH 00H` ; R0 content ને stack પર push કરો
- 3 `POP 01H` ; Stack content ને R1 માં pop કરો
- 4 `POP ACC` ; Stack content ને accumulator માં pop કરો

### મેમરી ટ્રીક

Move Makes Programs Possible - MOV, MOVX, PUSH, POP

## પ્રશ્ન 4(C) [7 ગુણ]

8051 ના Addressing Modes વ્યાખ્યાયિત કરો અને સમજાવો.

### જવાબ

#### 8051 Addressing Modes:

Mode	Description	Example	Usage
<b>Immediate</b>	Data instruction નો ભાગ છે.	MOV A, #50H	Constant values.
<b>Register</b>	સીધો register નો ઉપયોગ.	MOV A, R0	Fast access.
<b>Direct</b>	Direct address નો ઉપયોગ.	MOV A, 30H	RAM locations.
<b>Indirect</b>	Pointer તરીકે register.	MOV A, @R0	Dynamic addressing.
<b>Indexed</b>	Base + offset addressing.	MOVC A, @A+DPTR	Table lookup.
<b>Relative</b>	PC + offset.	SJMP LOOP	Branch instructions.
<b>Absolute</b>	Direct jump address.	LJMP 1000H	Long jumps.
<b>Bit</b>	Individual bit access.	SETB P1.0	Control operations.

### વિગતવાર ઉદાહરણો:

```

1 ; Immediate Addressing
2 MOV A, #25H ; 25H ને A માં Load કરો
3
4 ; Register Addressing
5 MOV A, R1 ; R1 ને A માં Copy કરો
6
7 ; Direct Addressing
8 MOV A, 40H ; Address 40H થી Load કરો
9
10 ; Indirect Addressing
11 MOV R0, #40H ; R0 points to 40H
12 MOV A, @R0 ; R0 દ્વારા pointed address થી Load કરો
13
14 ; Indexed Addressing
15 MOV DPTR, #TABLE ; Table ને point કરો
16 MOV A, #02H ; Index value
17 MOVC A, @A+DPTR ; TABLE+2 થી Load કરો

```

### મેમરી ટ્રીક

IRIDRAB - Immediate, Register, Indirect, Direct, Relative, Absolute, Bit

## પ્રશ્ન 4(A) OR [3 ગુણ]

R0 રજિસ્ટરમાં સ્ટોર થયેલ ડેટાનો 2's Complement શોધવાનો પ્રોગ્રામ લખો.

### જવાબ

Listing 2. 2's Complement Program

```

1 ORG 0000H
2 MOV R0, #85H ; Test data load કરો
3 MOV A, R0 ; Data ને accumulator માં copy કરો
4 CPL A ; bits ને complement (1's complement) કરો
5 INC A ; 2's complement મેળવવા 1 ઉમેરો
6 MOV R1, A ; Result R1 માં store કરો
7 END

```

### Algorithm:

- **Step 1:** R0 માંથી ડેટા accumulator માં load કરો.
- **Step 2:** CPL A વાપરીને તમામ bits complement કરો.
- **Step 3:** INC A વાપરીને 1 ઉમેરો.

- Step 4: પરિણામ પાછું સ્ટોર કરો.

**Verification:**

Original: 85H = 10000101B

1's Comp: 7AH = 01111010B

2's Comp: 7BH = 01111011B

**મેમરી ટ્રીક**

CCI - Complement, aCd 1, Include result

**પ્રશ્ન 4(B) OR [4 ગુણ]**

Logical Instructions નું લિસ્ટ આપો અને કોઈ પણ બે યોગ્ય ઉદાહરણ સાથે સમજાવો.

**જવાબ****Logical Instructions:**

Instruction	Function
ANL	Logical AND operation.
ORL	Logical OR operation.
XRL	Logical XOR operation.
CPL	Complement operation.
RL/RLC	Rotate left.
RR/RRC	Rotate right.
SWAP	Swap nibbles.

**વિગતવાર ઉદાહરણો:****1. ANL (AND Logic):**

```

1  MOV A, #0F0H    ; A = 11110000B
2  ANL A, #0AAH    ; 10101010B સાથે AND કરો
3                  ; પરિણામ: A = 10100000B = A0H

```

ઉપયોગ: ચોક્કસ bits ને mask કરવા, ન જોઈતા bits clear કરવા.

**2. ORL (OR Logic):**

```

1  MOV A, #0F0H    ; A = 11110000B
2  ORL A, #00FH    ; 00001111B સાથે OR કરો
3                  ; પરિણામ: A = 11111111B = FFH

```

ઉપયોગ: ચોક્કસ bits set કરવા, bit patterns ભેગા કરવા.

**મેમરી ટ્રીક**

AXOR - AND masks, XOR toggles, OR sets, Rotate shifts

**પ્રશ્ન 4(C) OR [7 ગુણ]**

નીચેની Instructions સમજાવો: (1)ADDC (2) INC (3) DEC (4) JZ (5) SUBB (6) NOP (7) RET

## જવાબ

## Instruction સમજૂતી:

1. **ADDC (Add with Carry):** Source, destination, અને carry flag નો સરવાળો કરે છે.

```
1 MOV A, #80H
2 ADDC A, #90H ; A = A + 90H + Carry flag
3
```

2. **INC (Increment):** Operand માં 1 નો વધારો કરે છે.

```
1 INC A ; A = A + 1
2 INC R0 ; R0 = R0 + 1
3
```

3. **DEC (Decrement):** Operand માં 1 નો ઘટાડો કરે છે.

```
1 DEC A ; A = A - 1
2 DEC R1 ; R1 = R1 - 1
3
```

4. **JZ (Jump on Zero):** Zero flag set હોય ત્યારે conditional jump.

```
1 DEC A
2 JZ ZERO_LABEL ; જો A = 0 તો Jump કરો
3
```

5. **SUBB (Subtract with Borrow):** Accumulator માંથી source અને carry બાદ કરે છે.

```
1 MOV A, #50H
2 SUBB A, #30H ; A = A - 30H - Carry flag
3
```

6. **NOP (No Operation):** Timing delay પૂરો પાડે છે, placeholder છે.

```
1 NOP ; કંઈ કરશો નહીં, 1 cycle વાપરો
2
```

7. **RET (Return):** Subroutine માંથી calling address પર return થાય છે.

```
1 CALL SUBROUTINE
2 ...
3 SUBROUTINE:
4 RET ; Caller પર પાછા ફરો
5
```

## મેમરી ટ્રીક

AIDS NR - Add, Increment, Decrement, Subtract, No-op, Return

## પ્રશ્ન 5(A) [3 ગુણ]

DJNZ અને CJNE Instructions યોગ્ય ઉદાહરણ સાથે સમજાવો.

## જવાબ

## DJNZ (Decrement and Jump if Not Zero):

```

1  MOV R0, #05H    ; Counter initialize કરો
2  LOOP:
3  MOV A, #00H     ; કોઈ operation
4  DJNZ R0, LOOP   ; R0 ઘટાડો, zero ન હોય તો jump કરો

```

કાર્ય: Decrement અને conditional jump operations ને જોડે છે.

## CJNE (Compare and Jump if Not Equal):

```

1  MOV A, #30H
2  CJNE A, #30H, NOT_EQUAL ; A ને 30H સાથે compare કરો
3  MOV R0, #01H           ; સમાન હોય
4  SJMP CONTINUE
5  NOT_EQUAL:
6  MOV R0, #00H           ; સમાન ન હોય
7  CONTINUE:

```

કાર્ય: બે operands compare કરે છે અને સમાન નથી તો jump કરે છે.

## મેમરી ટ્રીક

DC - Decrement count, Compare jump

## પ્રશ્ન 5(B) [4 ગુણ]

ટાઈમર 0 નો ઉપયોગ કરી 30 મિલી સેકન્ડનો ટાઈમ ડિલે જનરેટ કરવા માટે એસેમ્બલી પ્રોગ્રામ બનાવો. ક્રિસ્ટલ ફ્રિક્વન્સી 12 મેગા હર્ટઝ ગણતરીમાં લેવી.

## જવાબ

## Listing 3. 30ms Delay Program

```

1  ORG 0000H
2  MAIN:
3  CALL DELAY_30MS ; 30ms delay call કરો
4  SJMP MAIN      ; Repeat કરો
5
6  DELAY_30MS:
7  MOV TMOD, #01H ; Timer 0, Mode 1 (16-bit)
8  MOV TH0, #8AH  ; 30ms માટે high byte લોડ કરો
9  MOV TL0, #23H  ; Low byte લોડ કરો
10 SETB TR0       ; Timer 0 start કરો
11 WAIT:
12 JNB TF0, WAIT  ; Timer overflow માટે રાહ જુઓ
13 CLR TR0        ; Timer stop કરો
14 CLR TF0        ; Timer flag clear કરો
15 RET
16 END

```

## 30ms delay માટે ગણતરી:

- Crystal Frequency = 12 MHz
- Machine Cycle =  $12/12 \text{ MHz} = 1 \mu\text{s}$
- 30ms માટે =  $30,000 \mu\text{s} = 30,000 \text{ machine cycles}$
- Timer Count =  $65536 - 30000 = 35536 = 8A23H$
- TH0 = 8AH, TL0 = 23H

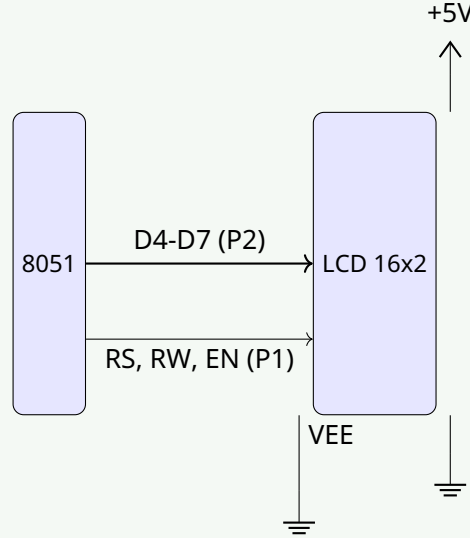
## મેમરી ટ્રીક

CLSW - Calculate, Load, Start, Wait

## પ્રશ્ન 5(C) [7 ગુણ]

8051 માઈક્રોકંટ્રોલર સાથે LCD નો ઇન્ટરફેસિંગ ડાયાગ્રામ દોરો અને ઇન્ટરફેસિંગ માટે જરૂરી LCD ની તમામ પીનો સમજાવો.

## જવાબ



આકૃતિ 12. LCD Interfacing

## LCD પીન કાર્યો:

- RS (Pin 4): Register Select - 0=Command, 1=Data.
- RW (Pin 5): Read/Write - 0=Write, 1=Read.
- EN (Pin 6): Enable - Data transfer માટે high to low pulse.
- D4-D7 (Pins 11-14): Commands/data માટે 4-bit data lines.

## મૂળભૂત LCD Commands:

- 0x38: Function set (8-bit, 2 lines).
- 0x0E: Display ON, cursor ON.
- 0x01: Clear display.
- 0x80: Cursor ને first line પર set કરો.

## મેમરી ટ્રીક

REED - RS selects, RW reads, EN enables, Data transfers

## પ્રશ્ન 5(A) OR [3 ગુણ]

65h મેમરી લોકેશન પર સ્ટોર થયેલ ડેટાનું 75h મેમરી લોકેશન પર સ્ટોર થયેલ ડેટા સાથે OR ઓપરેશન કરો અને પરિણામ R6 રજિસ્ટરમાં સ્ટોર કરવાનો પ્રોગ્રામ લખો.



## જવાબ

Listing 4. OR Operation Program

```

1  ORG 0000H
2  MOV 65H, #0F0H ; 65H પર ટેસ્ટ ડેટા સ્ટોર કરો
3  MOV 75H, #0AAH ; 75H પર ટેસ્ટ ડેટા સ્ટોર કરો
4
5  MOV A, 65H ; 65H માંથી ડેટાને accumulator માં લોડ કરો
6  ORL A, 75H ; 75H પરના ડેટા સાથે OR કરો
7  MOV R6, A ; પરિણામ R6 register માં સ્ટોર કરો
8  END

```

## ઉદાહરણ ગણતરી:

65H પરનો ડેટા: F0H = 11110000B

75H પરનો ડેટા: AAH = 10101010B

OR પરિણામ: FAH = 11111010B

## મેમરી ટ્રીક

LOS - Load, OR, Store result

## પ્રશ્ન 5(B) OR [4 ગુણ]

P1.3 પર 2 કિલો હર્ટઝનો સ્ક્વેર વેવ જનરેટ કરવા માટે એસેમ્બલી પ્રોગ્રામ લખો. ક્રિસ્ટલ ફ્રિક્વન્સી 11.0592 મેગા હર્ટઝ ગણતરીમાં લેવી.

## જવાબ

Listing 5. Square Wave Program

```

1  ORG 0000H
2  MAIN:
3  SETB P1.3 ; P1.3 ને high કરો
4  CALL DELAY_250US ; અડધા period માટે delay
5  CLR P1.3 ; P1.3 ને low કરો
6  CALL DELAY_250US ; અડધા period માટે delay
7  SJMP MAIN ; સતત repeat કરો
8
9  DELAY_250US:
10 MOV TMOD, #01H ; Timer 0, Mode 1
11 MOV TH0, #0FEH ; High byte લોડ કરો
12 MOV TL0, #0CBH ; Low byte લોડ કરો
13 SETB TR0 ; Timer 0 start કરો
14 WAIT:
15 JNB TF0, WAIT ; Overflow માટે રાહ જુઓ
16 CLR TR0 ; Timer stop કરો
17 CLR TF0 ; Flag clear કરો
18 RET
19 END

```

## 2KHz Square Wave માટે ગણતરી:

- Frequency = 2KHz, Period = 500 $\mu$ s. Half Period = 250 $\mu$ s.
- Crystal = 11.0592 MHz. Machine Cycle = 1.085 $\mu$ s.
- Timer Count = 250/1.085  $\approx$  230.
- Timer Value = 65536 - 230 = 65306 = F ECBH.

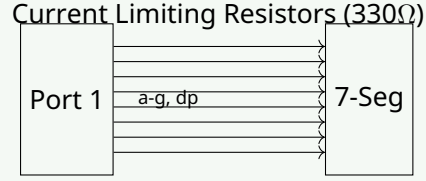
## મેમરી ટ્રીક

SCDW - Set high, Clear low, Delay, Wait

## પ્રશ્ન 5(C) OR [7 ગુણ]

8051 માઈક્રોકંટ્રોલર સાથે 7-Segment ડિસ્પ્લેનો ઇન્ટરફેસિંગ દોરો અને સમજાવો.

## જવાબ



આકૃતિ 13. 7-Segment Interface

ડિસ્પ્લે કોન્ફિગરેશન:

Digit	Common Cathode	Common Anode
0	3FH	C0H
1	06H	F9H
2	5BH	A4H

નમૂના પ્રોગ્રામ:

```

1  MOV DPTR, #DIGIT_TABLE ; Lookup table ને point કરો
2  MOV A, #05H           ; અંક 5 દર્શાવો
3  MOVC A, @A+DPTR       ; 7-segment કોડ મેળવો
4  MOV P1, A             ; Display ને મોકલો

```

## મેમરી ટ્રીક

CRAM - Common connection, Resistors limit, Address segments, Multiplex digits