

## Subject Name (Gujarati)

1333202 -- Summer 2025

# Semester 1 Study Material

### Detailed Solutions and Explanations

**પ્રશ્ન 1(અ) [3 ગુણ]**

8085 નું બસ ઓર્ગનાઈઝેશન દોરો.

જવાબ

## 8085 MICROPROCESSOR

[illegible]

### 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(બ) [4 ગુણ]**

**માઈક્રોપ્રોસેસરની માઈક્રોકંટ્રોલર સાથે સરખામણી કરો.**

જવાબ

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 કમ્યુટિંગ	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(ક) [7 ગુણ]

8085 માઈક્રોપ્રોસેસરના દરેક બ્લોક દોરો અને સમજાવો.

#### જવાબ

```
graph TB
    A[Accumulator] --- ALU[Arithmetic Logic Unit]
    B[Register Array] --- ALU
    ALU --- F[Flag Register]

    PC[Program Counter] --- AB[Address Buffer]
    SP[Stack Pointer] --- AB
    AB --- ADDR[Address Bus 16-bit]

    IR[Instruction Register] --- ID[Instruction Decoder]
    ID --- CU[Control Unit]
    CU --- CB[Control Bus]

    DB[Data Buffer] --- DATA[Data Bus 8-bit]

    T[Timing & Control] --- CU
```

#### 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 એકદર ઓપરેશનને કંટ્રોલ કરે છે

#### મેમરી ટ્રીક

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

### પ્રશ્ન 1(ક) OR [7 ગુણ]

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

#### જવાબ

#### 8085 PIN DIAGRAM

X1	1	+{--}{--}{--}{--}{--}{--}{--}{--}	40	Vcc
X2	2		39	HOLD
RESET	3		38	HLDA
SOD	4		37	CLK(OUT)
SID	5	8085	36	RESET IN
TRAP	6		35	READY
RST7.5	7		34	IO/M
RST6.5	8		33	S1

```

RST5.5  9 |          | 32  RD
INTR    10 |          | 31  WR
INTA    11 |          | 30  ALE
AD0{-7 12{-}19|          | 23{-}29 A8{-}A15}
Vss     20 +{-{-}-{-}-{-}-{-}-{-}-+ 21  A15{-}A8}

```

#### 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(અ) [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(બ) [4 ગુણ]

RISC અને CISC વચ્ચે તફાવત આપો.

#### જવાબ

લક્ષણ	RISC	CISC
<b>Instructions</b>	સરળ, fixed format	જટિલ, variable format
<b>Execution</b>	Single cycle execution	Multiple cycle execution
<b>Addressing</b>	થોડા addressing modes	ઘણા addressing modes
<b>Memory</b>	Load/Store architecture	Memory-to-memory operations
<b>Compiler</b>	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(ક) [7 ગુણ]

Von-Neumann અને Harvard Architecture વચ્ચે તફાવત આપો.

જવાબ		
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

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting}[]
graph TD
    subgraph "Von{-Neumann}"
        CPU1[CPU] --{-}{-}{-} MEM1[Combined Memory{br/{}Data + Instructions}]
    end

    subgraph "Harvard"
        CPU2[CPU] --{-}{-}{-} IMEM[Instruction Memory]
        CPU2 --{-}{-}{-} DMEM[Data Memory]
    end
{Highlighting}
{Shaded}
```

મેમરી ટ્રીક

“VH - Von has one bus, Harvard has two”

પ્રશ્ન 2(અ) OR [3 ગુણ]

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

જવાબ	
વ્યાખ્યાઓ:	<ul style="list-style-type: none"><li>• <b>T-State:</b> Time state - મૂળભૂત timing unit, એક clock period</li><li>• <b>Instruction Cycle:</b> એક instruction નું સંપૂર્ણ execution</li><li>• <b>Machine Cycle:</b> એક memory operation માટે જરૂરી T-states નું જૂથ</li></ul>
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(બ) OR [4 ગુણ]

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



- **SP (81H):** Stack Pointer register
- લાક્ષણિકતાઓ:**
- **Address Range:** Internal RAM માં 80H થી FFH
  - **Bit Addressable:** કેટલાક SFRs individual bit access આપે છે
  - **Function Specific:** દરેકનું dedicated hardware function હોય છે

## મેમરી ટ્રીક

“APS - ACC, PSW, Stack Pointer”

**પ્રશ્ન 3(બ) [4 ગુણ]**

**Program Counter (PC) અને Data Pointer (DPTR) Register સમજાવો.**

જાદીયા

**Program Counter (PC):**

- **Size:** 16-bit register
- **Function:** આગળના instruction નું address ધરાવે છે
- **Auto-increment:** Instruction fetch પછી automatically increment થાય છે
- **Range:** 0000H થી FFFFH

**Data Pointer (DPTR):**

- **Size:** 16-bit register (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)

PC:    +{-{-}}{-}{-}{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+

|     PCH   |     PCL   |   16{-bit}

$$+ \{-[-]\{-\}\{-\}\{-\}\{-\}\{-\}\{-\}\{-\} + \{-\}\{-\}\{-\}\{-\}\{-\}\{-\}\{-\}\{-\} +$$

DPTR: +{-{-}}{-}{-}{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+

|     DPH    |     DPL    |   16{-bit }   

$$+ \{-\{-\}\{-\}\{-\}\{-\}\{-\}\{-\}\{-\}\{-\} + \{-\}\{-\}\{-\}\{-\}\{-\}\{-\}\{-\}\{-\} +$$

83H	82H
-----	-----

### મેમરી ટ્રીક

“PD - PC Points to Program, DPTR Points to Data”

**પ્રશ્ન 3(ક) [7 ગુણ]**

8051 નું આર્કિટેક્ચર દોરો અને સમજાવો.

**જાદીયા**

```
graph TB
    subgraph "8051 Architecture"
        ALU[8{-bit ALU}]
        ACC[Accumulator A]
        B[B Register]
        PSW[Program Status Word]

        SP[Stack Pointer]
        DPTR[Data Pointer DPTR]
        PC[Program Counter PC]

        ROM[4KB ROM{br/0000{-}OFFF}]
        RAM[128B RAM{br/00{-}7F}]
        SFR[SFR Area{br/80{-}FF}]
    end
```

```

P0[Port 0]
P1[Port 1]
P2[Port 2]
P3[Port 3]

T0[Timer 0]
T1[Timer 1]
UART[Serial Port]
INT[Interrupt Control]
end

ALU {-{-}{-} ACC}
ALU {-{-}{-} B}
ALU {-{-}{-} PSW}

```

#### 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:** Communication માટે full duplex UART
- **Interrupts:** Priority levels સાથે 5 interrupt sources

#### વિશેષ લક્ષણો:

- **Boolean Processor:** Bit manipulation capabilities
- **Addressing Modes:** 8 અલગ addressing modes
- **Power Management:** Idle અને power-down modes

#### મેમરી ટ્રીક

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

### પ્રશ્ન 3(અ) 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 ના OE pin સાથે જોડાય છે
- **XTAL1 & XTAL2 (Pins 19, 18):** Crystal connections
  - Clock generation માટે external crystal જોડાય છે
  - સામાન્ય frequency: 11.0592 MHz અથવા 12 MHz

Crystal Oscillator Connection:

```

XTAL1 {-{-}{-}{-}[Crystal]{-}{-}{-}{-} XTAL2}
|                               |
[C1]                           [C2]
|                               |
GND                           GND

```

#### મેમરી ટ્રીક

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

**પ્રશ્ન 3(બ) OR [4 ગુણ]**

8051 માઈક્રોકંટ્રોલરનું આંતરિક RAM ઓર્ગેનાઈઝેશન સમજાવો.

ଝରାଘ

### 8051 Internal RAM Organization (128 Bytes)

[illegible]

### RAM વિભાગો:

- **Register Banks:** 4 banks  $\times 8\text{registers}(00H - 7FH)$
- **Bit Addressable:** Individual bit access સાથે 16 bytes (20H-2FH)
- **General Purpose:** User data માટે 80 bytes (30H-7FH)
- **Stack Area:** સામાન્યતે 08H થી ઉપર શરૂ થાય છે

### Addressing:

- **Direct:** वास्तविक address वापरते (MOV 30H, A)
- **Indirect:** Register pointer वापरते (MOV @R0, A)

## મેમરી ટ્રીક

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

**પ્રશ્ન 3(ક) OR [7 ગુણ]**

8051 નો પીન ડાયાગ્રામ દોરો અને કોઈ પણ 4 પીન સમજાવો.

ଓଡ଼ିଆ

## 8051 PIN DIAGRAM

P1.0	1	+{--}{--}{--}{--}{--}{--}{--}{--}	40	Vcc
P1.1	2		39	P0.0/AD0
P1.2	3		38	P0.1/AD1
P1.3	4		37	P0.2/AD2
P1.4	5	8051	36	P0.3/AD3
P1.5	6		35	P0.4/AD4
P1.6	7		34	P0.5/AD5
P1.7	8		33	P0.6/AD6
RESET	9		32	P0.7/AD7
P3.0/RXD	10		31	EA/VPP
P3.1/TXD	11		30	ALE/PROG
P3.2/INT0	12		29	PSEN
P3.3/INT1	13		28	P2.7/A15
P3.4/T0	14		27	P2.6/A14



P3.5/T1	15	26	P2.5/A13
P3.6/WR	16	25	P2.4/A12
P3.7/RD	17	24	P2.3/A11
XTAL2	18	23	P2.2/A10
XTAL1	19	22	P2.1/A9
VSS	20+{-{-}{-}{-}{-}{-}{-}{-}{-}+ 21		P2.0/A8}

**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(અ) [3 ગુણ]**

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

정답이

```

ORG 0000H
MOV R0, \#05H    ;
MOV R1, \#03H    ;
MOV A, R0        ; R0  accumulator
MOV B, R1        ; R1  B register
MUL AB           ; A   B
MOV R2, A        ; LSB  R2
MOV R3, B        ; MSB  R3
END

```

**પ્રોગ્રામ ફ્લો:**

- **Operands લોડ કરો** R0 અને R1 માં
- **ટ્રાન્સફર કરો** ગુણાકાર માટે A અને B registers માં
- **Execute કરો** MUL AB instruction
- **સ્ટોર કરો** 16-bit પરિણામ (A=LSB, B=MSB)

**પરિણામ સ્ટોરેજ:**

- **R2:** Product ની lower 8 bits
- **R3:** Product ની upper 8 bits

મેમરી ટ્રીક
-------------

“LTSE - Load, Transfer, multiply, Store result”

**પ્રશ્ન 4(બ) [4 ગુણ]**

ડેટા ટ્રાન્સફર ઇન્સ્ટ્રક્શનની યાદી આપો. કોઈ પણ બે ડેટા ટ્રાન્સફર ઇન્સ્ટ્રક્શન ઉદાહરણ સહિત સમજાવો.

정답이

ડેટા ટ્રાન્સફર ઇન્સ્ટ્રક્શન:	
Instruction	કાર્ય
MOV	Registers/memory વચ્ચે data move કરે છે
MOVS	External memory થી data move કરે છે
MOVC	Code byte ને accumulator માં move કરે છે

PUSH	Data ને stack પર push કરે છે
POP	Stack માંથી data pop કરે છે
XCH	Accumulator સાથે register exchange કરે છે
XCHD	Lower nibble exchange કરે છે

વિગતવાર ઉદાહરણો:

#### 1. MOV Instruction:

```
MOV A, \#50H      ; Immediate data 50H  accumulator
MOV R0, A          ; Accumulator content  R0  copy
MOV 30H, A         ; Accumulator content  address 30H
```

#### 2. PUSH/POP Instructions:

```
PUSH ACC           ; Accumulator  stack  push
PUSH 00H           ; R0 content  stack  push
POP 01H            ; Stack content  R1  pop
POP ACC            ; Stack content  accumulator  pop
```

#### મેમરી ટ્રીક

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

#### પ્રશ્ન 4(ક) [7 ગુણ]

8051 ના એડ્રેસિંગ મોડ્સને વ્યાખ્યાયિત કરો અને સમજાવો.

#### જવાબ

#### 8051 એડ્રેસિંગ મોડ્સ:

મોડ	વર્ણન	ઉદાહરણ	ઉપયોગ
<b>Immediate</b>	Data instruction નો ભાગ છે	MOV A, #50H	સ્થિર મૂલ્યો
<b>Register</b>	Register નો સીધો ઉપયોગ	MOV A, R0	ઝડપી access
<b>Direct</b>	સીધું address વાપરે છે	MOV A, 30H	RAM locations
<b>Indirect</b>	Register ને pointer તરીકે વાપરે છે	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:

```
; Immediate Addressing
MOV A, \#25H          ; 25H  A

; Register Addressing
MOV A, R1              ; R1  A   copy

; Direct Addressing
MOV A, 40H             ; Address 40H

; Indirect Addressing
MOV R0, \#40H          ; R0 40H  point
MOV A, @R0             ; R0    pointed address

; Indexed Addressing
MOV DPTR, \#TABLE      ; Table  point
MOV A, \#02H           ; Index value
MOVC A, @A+DPTR        ; TABLE+2
```

### મેમરી ટ્રીક

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

### પ્રશ્ન 4(અ) OR [3 ગુણ]

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

#### જવાબ

```
ORG 0000H
MOV R0, \#85H          ;
MOV A, R0              ; accumulator copy
CPL A                  ; bits complement (1's complement)}
INC A                  ; 2's complement 1  }
MOV R1, A              ; R1
END
```

#### Algorithm:

- **Step 1:** R0 માંથી ડેટાને accumulator માં લોડ કરો
- **Step 2:** CPL A વાપરીને બધા bits complement કરો
- **Step 3:** 2's complement માટે INC A વાપરીને 1 ઉમેરો
- **Step 4:** પરિણામ પાછું સ્ટોર કરો

#### ચકાસણી:

```
: 85H = 10000101B
1's Comp: 7AH = 01111010B
2's Comp: 7BH = 01111011B
```

### મેમરી ટ્રીક

``CCI - Complement, aCd 1, Include result``

### પ્રશ્ન 4(બ) OR [4 ગુણ]

લોજિકલ ઇન્સ્ટ્રક્શનની યાદી આપો. કોઈ પણ બે લોજિકલ ઇન્સ્ટ્રક્શન ઉદાહરણ સહિત સમજાવો.

## જવાબ

### લોજિકલ ઇન્સ્ટ્રક્શન:

Instruction	કાર્ય
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):

```
MOV A, \#0F0H      ; A = 11110000B
ANL A, \#0AAH       ; 10101010B   AND
; :
```

A = 10100000B = A0H

ઉપયોગ: વિશિષ્ટ bits masking, અનચાહતા bits clear કરવા

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

```
MOV A, \#0F0H      ; A = 11110000B
ORL A, \#00FH       ; 00001111B   OR
; :
```

A = 11111111B = FFH

ઉપયોગ: વિશિષ્ટ bits setting, bit patterns combine કરવા

## મેમરી ટ્રીક

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

## પ્રશ્ન 4(ક) OR [7 ગુણ]

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

## જવાબ

### ઇન્સ્ટ્રક્શન સમજાવટ:

#### 1. ADDC (Add with Carry):

```
MOV A, \#80H
ADDC A, \#90H      ; A = A + 90H + Carry flag
```

કાર્ય: Source, destination, અને carry flag ઉમેરે છે

#### 2. INC (Increment):

```
INC A              ; A = A + 1
INC R0             ; R0 = R0 + 1
INC 30H            ; (30H) = (30H) + 1
```

કાર્ય: Operand માં 1 વધારે છે

#### 3. DEC (Decrement):

```
DEC A              ; A = A {- 1}
```

```
DEC R1          ; R1 = R1 {- 1}
DEC 40H          ; (40H) = (40H) {- 1 }
```

કાર્ય: Operand માંથી 1 ઓછું કરે છે

#### 4. JZ (Jump on Zero):

```
DEC A
JZ ZERO\_LABEL ; A = 0      jump
```

કાર્ય: Zero flag set હોય ત્યારે conditional jump

#### 5. SUBB (Subtract with Borrow):

```
MOV A, \#50H
SUBB A, \#30H ; A = A {- 30H {-} Carry flag}
```

કાર્ય: Accumulator માંથી source અને carry ઓછું કરે છે

#### 6. NOP (No Operation):

```
NOP ; , 1 cycle
```

કાર્ય: Timing delay આપે છે, placeholder

#### 7. RET (Return):

```
CALL SUBROUTINE
...
SUBROUTINE:
    MOV A, \#10H
    RET ; Caller
```

કાર્ય: Subroutine માંથી calling address પર પાછા જાય છે

### મેમરી ટ્રીક

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

## પ્રશ્ન 5(અ) [3 ગુણ]

DJNZ અને CJNE ઇન્સ્ટ્રક્શન યોગ્ય ઉદાહરણ સહિત સમજાવો.

### જવાબ

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

```
MOV R0, \#05H ; Counter initialize
LOOP:
    MOV A, \#00H ; operation
    DJNZ R0, LOOP ; R0 , zero      jump
```

કાર્ય: Decrement અને conditional jump operations combine કરે છે

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

```
MOV A, \#30H
CJNE A, \#30H, NOT\_EQUAL ; A 30H compare
MOV R0, \#01H ; Equal case
SJMP CONTINUE
NOT\_EQUAL:
    MOV R0, \#00H ; Not equal case
CONTINUE:
```

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

ઉપયોગો:

- DJNZ: Loop control, counting operations
- CJNE: Decision making, condition checking

## મેમરી ટ્રીક

“DC - Decrement count, Compare jump”

**પ્રશ્ન 5(બ) [4 ગુણ]**

ટાઈમર ૦ નો ઉપયોગ કરી ૩૦ મિલી સેકન્ડનો ટાઈમ ડિલે જનરેટ કરવા માટે એસેમ્બલી પ્રોગ્રામ બનાવો. ક્રિસ્ટલ ફ્રિક્વન્સી ૧૨ મેગા હર્ટઝ ગણતરીમાં લેવી.

જગદીય

```

ORG 0000H
MAIN:
    CALL DELAY\_30MS    ; 30ms delay call
    SJMP MAIN           ; Repeat

DELAY\_30MS:
    MOV TMOD, \#01H     ; Timer 0, Mode 1 (16{-bit})
    MOV TH0, \#8AH      ; 30ms high byte
    MOV TLO, \#23H      ; Low byte
    SETB TR0            ; Timer 0 start

WAIT:
    JNB TFO, WAIT       ; Timer overflow
    CLR TR0             ; Timer stop
    CLR TFO             ; Timer flag clear
    RET

END

```

**30ms delay માટે ગણતરી:**

Crystal Frequency = 12 MHz  
 Machine Cycle = 12/12 MHz = 1  $\mu$ s  
 30ms = 30,000  $\mu$ s = 30,000 machine cycles

Timer Count = 65536 - 30000 = 35536 = 8A23H  
 TH0 = 8AH, TLO = 23H

**Timer Configuration:**

- **TMOD:** Timer mode register configuration
- **TH0/TLO:** Timer 0 high/low byte registers
- **TR0:** Timer 0 run control bit
- **TFO:** Timer 0 overflow flag

મેમરી ટ્રીક

“CLSW - Calculate, Load, Start, Wait”

**પ્રશ્ન 5(ક) [7 ગુણ]**

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

જાણી

```
8051 to LCD Interfacing (4{-bit mode})

8051                                16x2 LCD
{-{-}}{-}}{-}                    {-}}{-}}{-}}{-}}{-}}{-}}{-}}
P2.7 {-{-}}{-}}{-}}{-}}{-}}{-}}{-}} D7 (Pin 14)}
P2.6 {-{-}}{-}}{-}}{-}}{-}}{-}}{-}} D6 (Pin 13) }
P2.5 {-{-}}{-}}{-}}{-}}{-}}{-}}{-}} D5 (Pin 12)}
P2.4 {-{-}}{-}}{-}}{-}}{-}}{-}}{-}} D4 (Pin 11)}
```







[R] = Current limiting resistor (330Ω)

For Common Cathode:

Common pin (Pin 3,8) {-}{-}{-} GND}

For Common Anode:

Common pin (Pin 3,8) {-}{-}{-} +5V}

#### ડિસ્પ્લે કોન્ફિગરેશન:

અક્ષર	Common Cathode કોડ	Common Anode કોડ
0	3FH	C0H
1	06H	F9H
2	5BH	A4H
3	4FH	B0H
4	66H	99H
5	6DH	92H
6	7DH	82H
7	07H	F8H
8	7FH	80H
9	6FH	90H

#### નમૂના પ્રોગ્રામ:

```
ORG 0000H
MOV DPTR, \#DIGIT\_TABLE ; Lookup table point
MOV A, \#05H ; 5
MOVC A, @A+DPTR ; 7{-segment }
MOV P1, A ; Display
SJMP $ ;
```

DIGIT\\_TABLE:

```
DB 3FH, 06H, 5BH, 4FH, 66H ; 0,1,2,3,4
DB 6DH, 7DH, 07H, 7FH, 6FH ; 5,6,7,8,9
END
```

#### ઇન્ટરફેસ ઘટકો:

- **Current Limiting Resistors:** LED current મર્યાદિત કરવા 330Ω
- **Common Connection:** GND ને cathode અથવા +5V ને anode
- **Data Lines:** Segments a-g અને decimal point માટે 8 bits

#### મલ્ટિપલ ડિજિટ્સ માટે Multiplexing:

- **Digit Select:** Digit selection માટે વધારાના pins
- **Time Division:** Digits વચ્ચે ઝડપથી switch કરવું
- **Persistence of Vision:** એકસાથે display નો ભ્રમ બનાવે છે

#### મેમરી ટ્રીક

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