SUMMER – 14 EXAMINATION

Model Answer

Subject Code : **12187** Page No: 1/28

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.

2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.

3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).

4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.

5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.

6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.

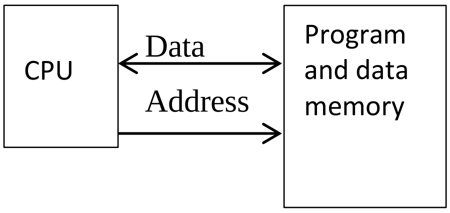
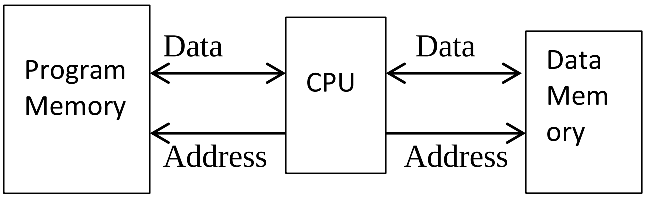
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Marks

1. A) Solve any three **(12)**

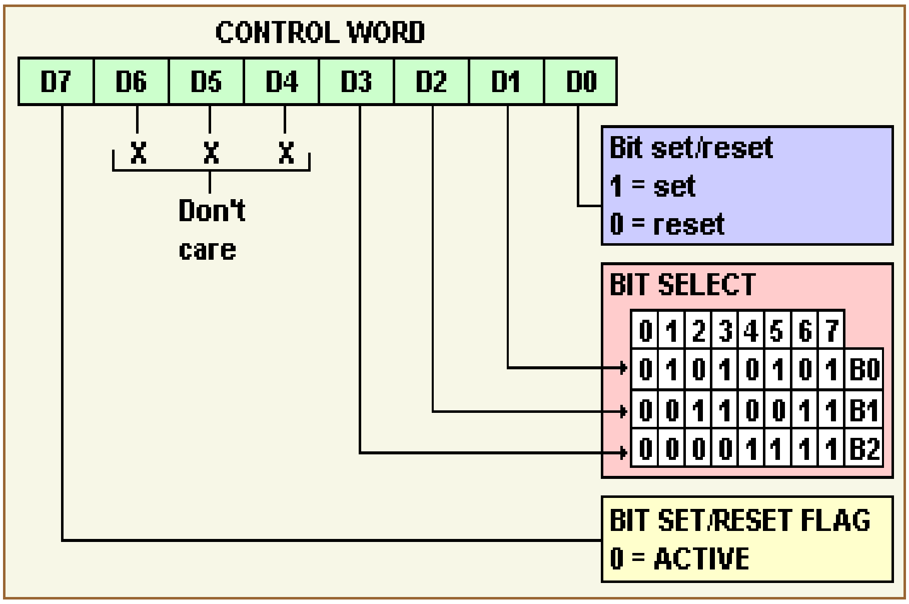
a) Differentiate between Von Neumann and Harvard architecture.

Ans : Any four points – 1 mark each.

	Von Neumann architecture	**Harvard architecture**
1		 Program Memory Data Address CPU Data Address Data Mem ory
2	The Van Neumann architecture uses single memory for their instructions and data.	The Harvard architecture uses physically separate memories for their instructions and data.
3	Requires single bus for instructions and data	Requires separate & dedicated buses for memories for instructions and data.
4	Its design is simpler	Its design is complicated
5	Instructions and data have to be fetched in sequential order limiting the operation bandwidth.	Instructions and data can be fetched simultaneously as there is separate buses for instruction and data which increasing operation bandwidth.
6	Program segments & memory blocks for data & stacks have separate sets of addresses.	Vectors & pointers, variables program segments & memory blocks for data & stacks have different addresses in the program.

b) Explain BSR mode of 8255. Write control word in BSR mode to set PC_4 bit of Port C.

Ans : BSR Mode diagram – 1 mark, Explanation – 1 mark, Correct Control Word – 2 marks.



0	1	2	3	4	5	6	7	
0	1	0	1	0	1	0	1	B0
0	0	1	1	0	0	1	1	B1
0	0	0	0	1	1	1	1	B2

Explanation :

- Any of the eight bits of Port C can be Set or Reset using a single output instruction.
- This feature reduces software requirements in Control-based applications.
- When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Set/Reset operation just as if they were data output ports.

Control word in BSR mode to set PC_4 bit of Port C.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	0	0	1

BSR mode PC4 set

= 09 H

c) Explain 8051 microcontroller as Boolean processor.

Ans : Explanation : 2 Marks ; Any two Boolean instructions : 1 mark each.

- The 8051 instruction is set is optimized for the one- bit operation so often desired in real-world, real- time control applications.
- The Boolean processor provides direct support for bit manipulation. This leads to more efficient programs that need to deal with binary input and output conditions inherent in digital control problems.
- Bit addressing can be used for test pin monitoring or program control flags. For example, instructions for Boolean function are as given below.

(a) ORL P0, #1 ; Set P0.0

(b) XRL P0, #1 ; Toggle P0.0

(c) ANL C, P1.4 ; AND the bit on P1.1 with carry

(d) ANL C, ! (P1.4) ; AND inverted bit on P1.4 with carry

d) Write an assembly language program to add two numbers AC H and gDH. Write the status of different flags after this addition.

Ans : Correct program – 2 marks, correct status of flags – ½ mark each.

Program :

```
MOV A, #  OACH;  Load first number in Accumulator.
ADD A, # 9D H;  Add first number with second number.
Loop :  AJMP Loop
```

Status of different flags after addition

AC = 1010 1100

+ 9D = + 1001 1101

———— ————————

149 $\boxed{1}$ **0100 1001**

1 4 9

Carry (CY) = 1

Auxiliary (AC) = 1

Overflow (OV) = 1

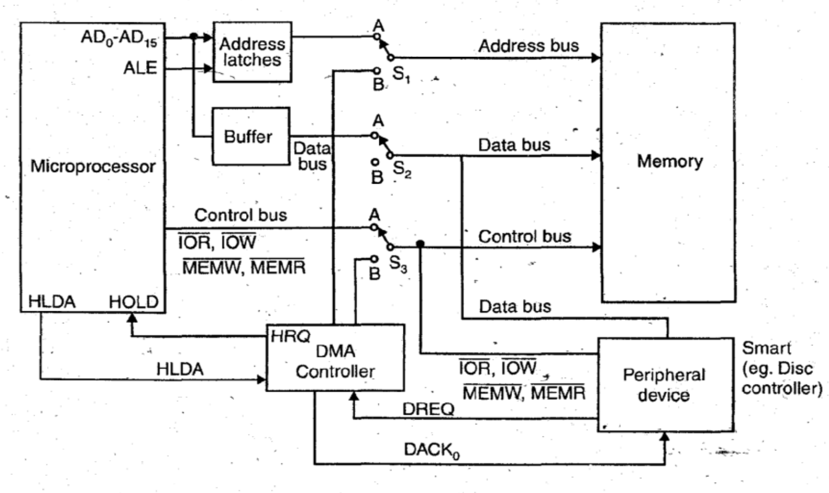
Parity bit (P) = 0

1B) Attempt any one : 6

a) Explain DMA controlled data transfer technique.

Ans : Suitable Diagram: 2 Marks; need of DMA:1 Mark; Types Explanation: 3 Marks

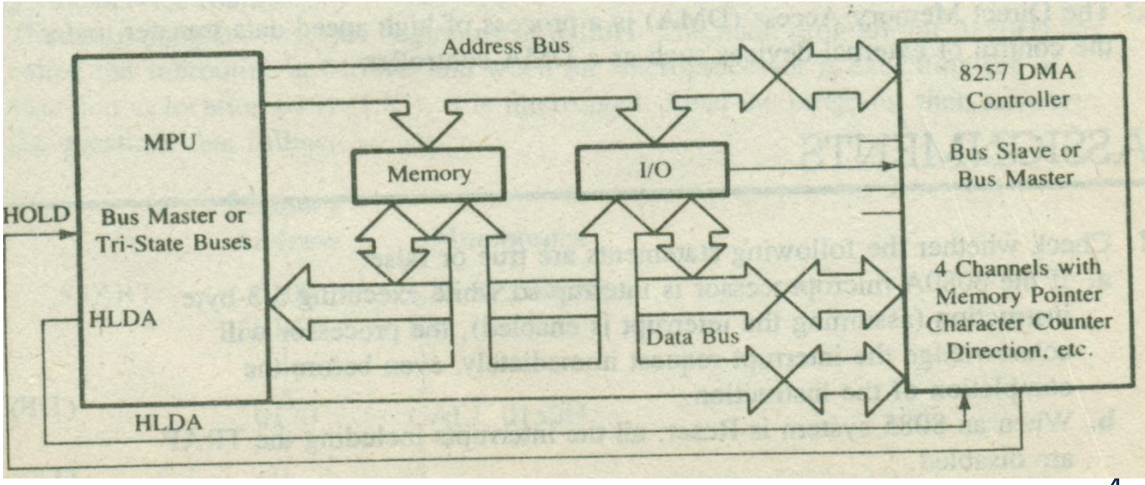
OR



DMA controller scheme

Fig: Direct Memory Access

OR



Address Bus
8257 DMA Controller
Bus Slave or Bus Master
MPU
Memory
I/O
HOLD
Bus Master or Tri-State Buses
HLDA
4 Channels with Memory Pointer Character Counter Direction, etc.
Data Bus
HLDA

Fig: DMA Data Transfer

- **Need:** In situation in which the microprocessor controlled data transfer is too slow, the DMA is generally used. E.g. data transfer between a floppy disk & R/W memory of the system.
- In this data transfer method, the data transfer operation is carried out by the DMA controller which is another master in the microprocessor based system.
- The data is transferred directly between I/O device and memory and data transfer is controlled by either I/O device or DMA controller.
- Microprocessor does not participate in this data transfer method.
- Whenever there is request from the I/O device, then DMA controller takes the control of all system buses i.e. address bus, data bus and control bus and perform data transfer operation directly between I/O device and memory.
- This method is used when the large amount of data is required to be transfer.
- IN IBM PC, Hard disk drive, floppy disk drive CD- ROM etc. devices uses this method of data transfer.
- In this method , when an I/O device wishes for data transfer, an I/O device itself generate request signal DREQ to DMA controller.
- In response to DREQ, the DMA controller send HOLD signal to the microprocessor.
- After receiving HOLD signal, microprocessor performs current operation completely and transfer the control of all system buses i.e. address bus, data bus and control bus to DMA controller.
- The DMA controller start data transfer operation.
- The speed of the data transfer is faster as compare to programmed I/O data transfer method.
- The three data transfer schemes of DMA are as given below:

1. Single Byte Transfer
2. Block Transfer
3. Hidden or Transparent DMA

Single Byte Transfer:

- Only one byte data is Transferred at a time – data transfer speed is low.
- In this mode of DMA data transfer, only one byte of data is transferred at a time, hence the data transfer speed is slow.
- DMA controller send HOLD signal to microprocessor and wait for HLDA (Acknowledge) signal.
- After receiving HLDA signal from microprocessor, the DMA enter into master mode and gain the control of all system buses and execute only one DMA cycle to transfer one byte of data.

- After transferring one byte of data, the DMA controller disable HOLD signal, enter into slave mode and transfer the control of system buses to the microprocessor.
- This process is repeated to transfer all data bytes. Means, the DMA controller enables and disable HOLD signal for all data bytes transfer

Block Transfer:

- In this mode of DMA data transfer, the block of data bytes is transferred continuously.
- During the DMA data transfer, the microprocessor is disconnected from the system buses, hence the microprocessor can not executes its own programs.
- N number of DMA cycle are added into the machine cycle of the microprocessor where N indicates numbers of bytes to be transferred.
- In this mode, the DMA controller sends HOLD signal to the microprocessor to gain the control of the system buses and wait for HLDA signal.
- After receiving HLDA signal, the DMA controller enters into the master mode and starts data transfer operation.
- After transferring all data bytes of the block, the DMA controller disable HOLD signal and enter into the slave mode.
- This mode DMA data transfer is faster than single byte mode.

Hidden or Transparent DMA:

- In the machine cycle of microprocessor, there are some states during which all buses are not used by the microprocessor means it floats system buses.
- During these states, the microprocessor is isolated from the system buses and DMA controller transfer data between I/O device and memory.
- This is slowest DMA data transfer. In this method, additional logic i.e. hardware is required to detect the idle states when the microprocessor floats its buses.

b) List various files required during assembly programming using 8051. Explain the steps to create an executable assembly language program with the help of flowchart.

Ans : List of various files : 2 marks, steps: 2 marks, flow chart :2 marks.

The list of various files required during assembly programming using 8051 are

- .asm
- .1st
- .obj
- .abs
- .hex

Step 1 : First of all, text editor is required to type a program. So, we can use any editor which produces ASCII file for example MS-DOS EDIT or WINDOWS NOTPAD. After typing program in the editor, program must be saved with some file name with extension **.asm** or **.src** depending on the assembler used. The **'asm'** extension file is required by assembler in the next step.

Step 2 : The **'asm'** source file containing the program code created in step 1 is fed to an 8051 assembler. The assembler converts the instruction into machine code and produces an object file with **óbj'** extension and list file with '1st'extension.

Step 3 : Assembler requires a third step called as linking. The link program takes one or more object file and produces an absolute object file with the extension **ábs'**. This file is used by the 8051 trainer that has a monitor program.

Step 4 : Next the **ábs'** file is fed into a program called as object to hex converter which create a file with extension **'hex'** that is ready to burn into the ROM. This program comes along with the assembler.

In recent WINDOWS based assembler comes with all steps which have been discussed in step 1 to 4.

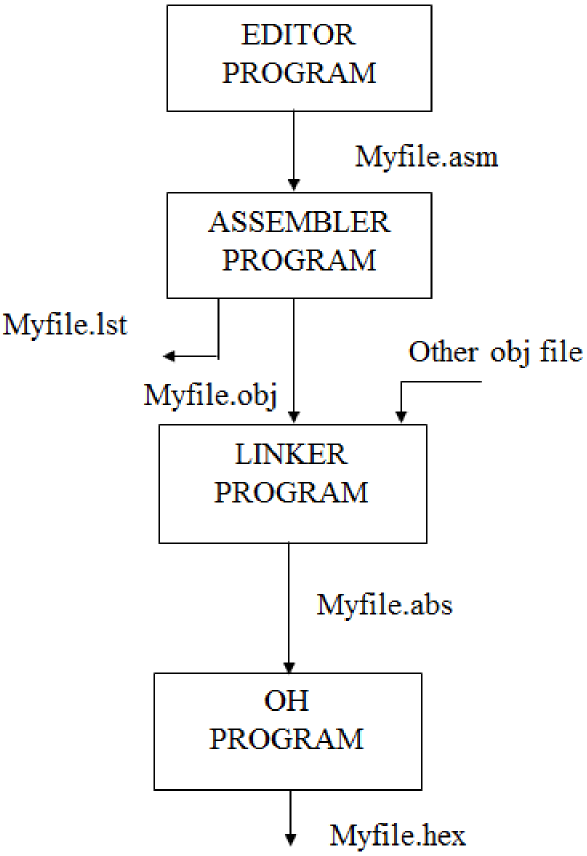


Figure: Flowchart indicating Steps to create an execute assembly language program

2. Attempt any four : 16

a) Explain read-modify-write feature of 8051 with one example.

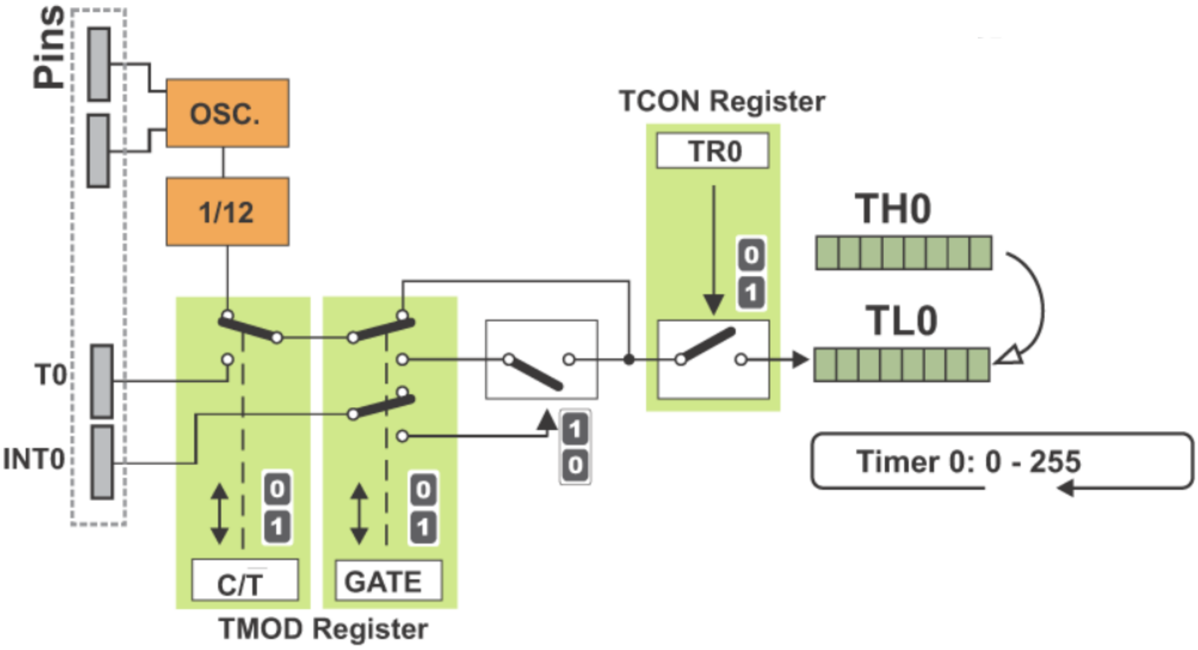
Ans : - Explanation : 2 marks, Any suitable Example: 2 marks.

The ports in the 8051 can be assessed by the read-modify-write technique. This feature saves many lines of code by combining in a single instruction all three actions of (1) reading the port, (2) modifying its value, and (3) writing to the port. The following code first places 01010101 (binary) into port 1. Next, the instruction “XLR P1, #0FFH” performs an XOR logic operation on P1 with 1111 1111 (binary), and then writes the result back into P1.

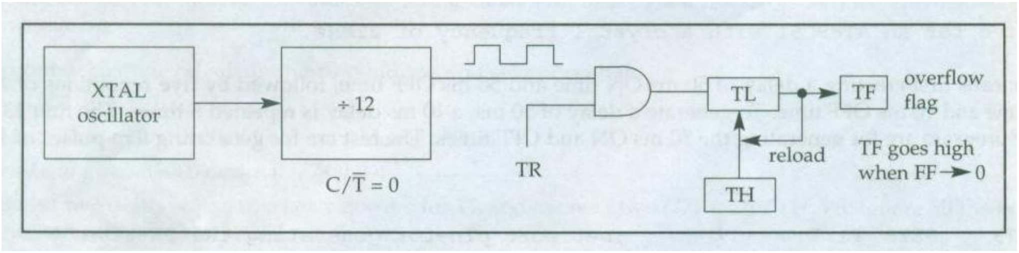
```
               MOV   P1, #55H             ; P1  =  01010101
AGAIN :        XLR    P1, #OFFH           ; EX-OR  P1 with 11111111
               ACALL DELAY
               SJMP  AGAIN
```

b) Describe the operation of timer of 8051 in mode 2.

Ans : Diagram : 2marks, Operation : 2 marks.



OR



Operation :

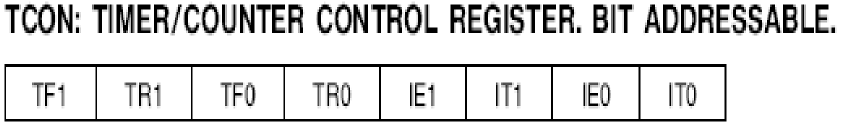
The following are the characteristics and operations of mode 2.

1. It is an 8-bit timer; therefore, it allows only values of 00 to FFH to be loaded into the timer's register TH.
2. After TH is loaded with the 8-bit value, the 8051 gives a copy of it to TL. Then, the timer must be started. This is done by the instructions 'SETB TRO' for Timer 0 and "SETB TR1"FOR Timer 1. This is just lik mode 1.
3. After the timer is started, it starts to count up by incrementing the TL register. It counts up until it reaches its limit of FFH. When it rolls over from FFH to 00, it sets high the TF (timer flag). If we are using Timer 0, TF0 goes high; if we are using Timer 1, TF1 is raised.
4. When the TL register rolls from FFH to 0 and TF is set to 1, TL is reloaded automatically with the original value kept by the TH register. To repeat the process, we must simply clear TF and let it go without any need by the programmer to reload the original value. This makes mode 2 an auto-reload, in contrast with mode 1 in which the programmer has to reload TH and TL.

It must be emphasized that mode 2 is an 8-bit timer. However, it has an auto-reloading capability. In auto-reload, TH is loaded with the initial count and a copy of it is given to TL. This reloading leaves TH unchanged, still holding a copy of the original value.

c) Draw the format of TCON register and explain the function of each bit.

Ans : format : 2 Marks Explanation : 2 Marks



TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 TCON. 7 Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.

TR1 TCON. 6 Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.

TF0 TCON. 5 Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.

TR0 TCON. 4 Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.

IE1 TCON. 3 External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.

IT1 TCON. 2 Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

IE0 TCON. 1 External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected.
Cleared by hardware when interrupt is processed.

IT0 TCON. 0 Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

d) Write a program to generate continuous ware of 2 KHz on P1.5 using mode 2 of timer 0. The crystal frequency is 1.0592 MHz.

Ans : Calculation of count : 2 marks, correct program : 2 marks.

- The count to be loaded in timer register must be calculated first.
- As we know the required frequency of square wave is 2 KHz and frequency applied to XTAL is 11.0592 MHz.

$T_{SQUARE} = 1/f = (1/2) \times 10^{-3}$

$= 500$ μsec (Period of Square Wave)

$T_{ON} = T_{OFF} = T_{SQUARE}/2 = 500/2 = 250$ μsec

So, $T_{XTAL} = 1 / f_{XTAL} = (1/11.0592) \times 10^{-6}$

$= 1.085$

Count $= 250 / 1.085 = 230$

- The count which is to be loaded in timer register = 256 – 230 = 26 in decimal i.e. 1AH.
- Now, load TH with 1AH.

Program :

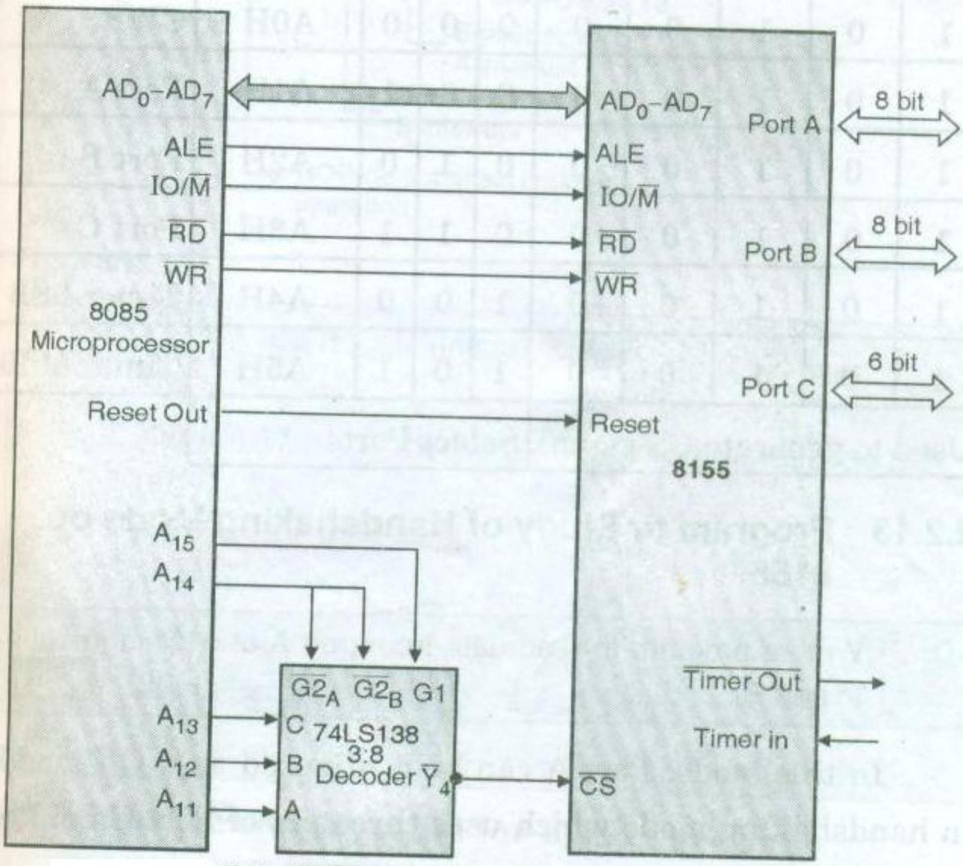
```
MOV  TMOD  #02H           ; Set Timer 0 in Mode 2 i.e. 8 bit timer.
MOV  THO, # 1AH           ; Load TH register with 8 bit Count
SETB  TRO                 ; Start Timer 0
POLL: JNB  TFO, POLL      ; Poll till timer roll over
CPL  P1.0                 ; Complement Port 1.5 line to get high or low
CLR   TFO                 ; Clear Timer Flag 0
SJMP POLL                 ; Re-poll timer
```

e) List any four features of IC 8155. Draw complete interfacing diagram of IC 8085 with IC 8155 in I/O mapped I/O mode.

Ans : Any four features – 2 marks. Interfacing diagram – 2 marks.

Features :

- 2K static RAM memory organized as 256 X 8.
- Completely Static Design.
- Internal Address Latches for de-multiplexing of AD_0-AD_7.
- Two Programmable 8-Bit I/O Ports.
- One Programmable 6-Bit I/O Port.
- Programmable 14-Bit Binary Counter/Timer.
- Multiplexed Address and Data Bus.
- Compatible with 8085 microprocessor.
- Require single +5 Volt power supply.



Interfacing of 8155 with 8085 in I/O mapped I/O

f) Draw the PSW format of 8051 microcontroller and explain each bit in detail.

Ans : format : 2 marks, explanation : 2 marks.

CY	AC	F0	RS	RS0	OV	--	P

CY	PSW.7	Carry flag
AC	PSW.6	Auxiliary carry flag
F0	PSW.5	Available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1
RS0	PSW.3	Register Bank selector bit 0
OV	PSW.2	Overflow flag
--	PSW.1	User-definable bit
P	PSW.0	Parity flag. Set / cleared by hardware each instruction cycle to indicate an odd/even number of 1 bits in the accumulator.

RSI	RS0	Register Bank	Address
0	0	0	00H - 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

Explanation :

CY, the carry flag

This flag is set whenever there is a carry out from the D7 bit. This flag bit is affected after an 8-bit addition or subtraction. It can also be set to 1 or 0 directly by an instruction such as “SETB C” and “CLR C” where “SETB C” stands for “set bit carry” and “CLR C” for “clear carry”.

AC, the auxiliary carry flag :

If there is a carry from D3 to D4 during an ADD or SUB operation, this bit is set; otherwise, it is cleared. This flag is used by instructions that perform BCD (binary coded decimal) arithmetic.

FO – Flag 0 :

This is a general-purpose bit available to the user.

RSO, RS1 – Register bank selects bits :

These two bits are used to select one of the four register banks in internal RAM as shown in Table. By writing zeroes and ones to these bits, a group of registers R_0-R_7 can be used out of four register banks in internal RAM.

P, the parity flag

The parity flag reflects the number of 1s in the A (accumulator) register only. If the A register contains an odd number of 1s, then P = 1. Therefore, P=0 if A has an even number of 1s.

OV, the overflow flag

This flag is set whenever the result of a signed number operation is too large, causing the high-order bit to overflow into the sign bit. In general, the carry flag is used to detect errors in unsigned arithmetic operations.

3. Attempt any four: 16

a) Write an assembly language program to find the largest number among the block of 10 bytes stored in internal memory starting at 50 H, and store the largest number at location number 60 H.
Ans : Correct program with comments : 4 marks

```
              CLR PSW.3          ;Select bank 0
              CLR PSW.4
             MOV R1,0AH        ;Initialize byte counter
             MOV R0,#40H       ;Initialize memory pointer
             DEC R1             ; Decrement byte counter by 1
              MOV 60H,@R0      ; Store number in memory location 60h
     UP:     INC  R0            ;Increment memory pointer by 1
              MOV A,@R0         ; Read next number
              CJNE A,60H,DN    ;If number next number, then go to DN
             AJMP NEXT         ; else go to NEXT
     DN:      JC NEXT            ; If next number >number then go to NEXT
             MOV  60H,A        ;Else replace next number with number
   NEXT:      DJNZ R1,UP       ;Decrement byte counter by 1,if byte counter ≠ 0 the go to UP
  LOOP:      AJMP LOOP         ;Stop.
```

b) Explain memory organization of 8051 in detail.
Ans : Diagram: 2 marks, Explanation : 2 marks.

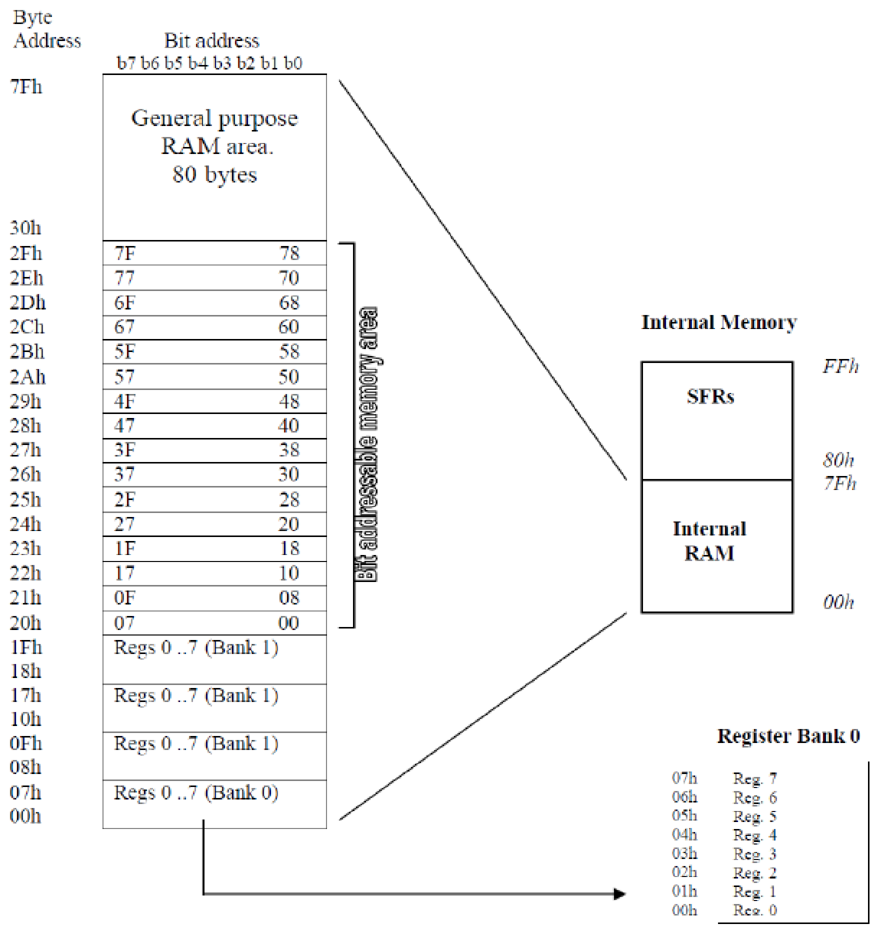


Fig:- Internal memory organisation

Internal ROM The 8051 has 4K (4096 locations) of on-chip ROM. This is used for storing the system program. 212 = 4096, therefore the internal ROM address bus is 12 bits wide and internal ROM locations go from 000H to FFFH.

Internal RAM

There are 256 bytes of internal RAM on the 8051. 28 = 256, therefore the internal RAM address bus is 8 bits wide and internal RAM locations go from 00H to FFH.

Register Banks

There are four register banks from 00H to 1FH. On power-up, registers R0 to R7 are located at 00H to 07H. However, this can be changed so that the register set points to any of the other three banks (if you change to Bank 2, for example, R0 to R7 is now located at 10H to 17H).

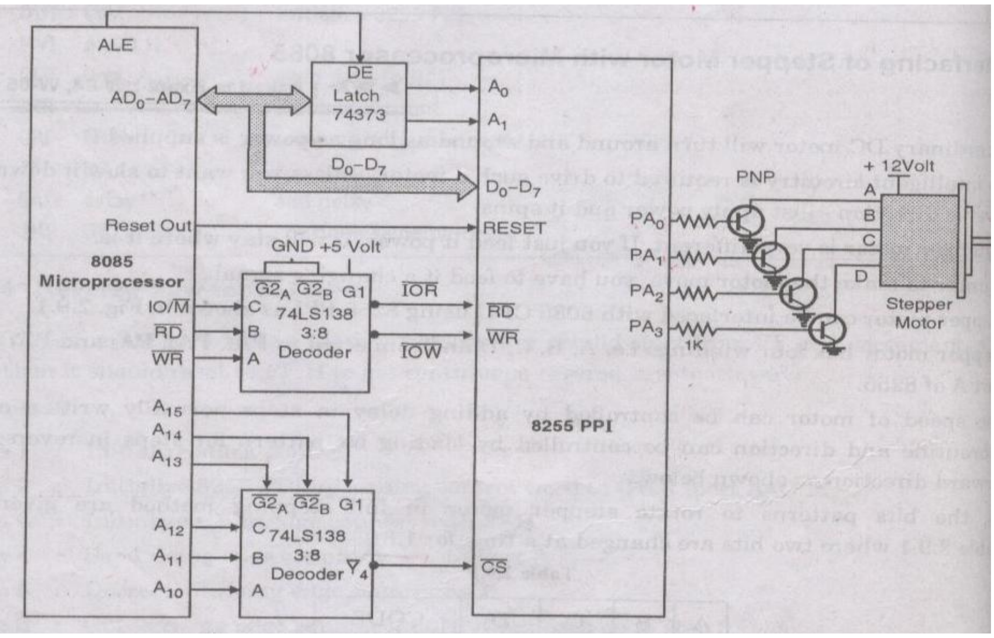
Bit-addressable Locations

The 8051 contains 210 bit-addressable locations of which 128 are at locations 20H to 2FH while the rest are in the SFRs. Each of the 128 bits from 20H to 2FH have a unique number (address) attached to them, as shown in the table above. The 8051 instruction set allows you to set or reset any single bit in this section of RAM. With the general purpose RAM from 30H to 7FH and the register banks from 00H to 1FH, you may only read or write a full byte (8 bits) at these locations. However, with bit-addressable RAM (20H to 2FH) you can read or write any single bit in this region by using the unique address for that bit. We will later see that this is a very powerful feature.

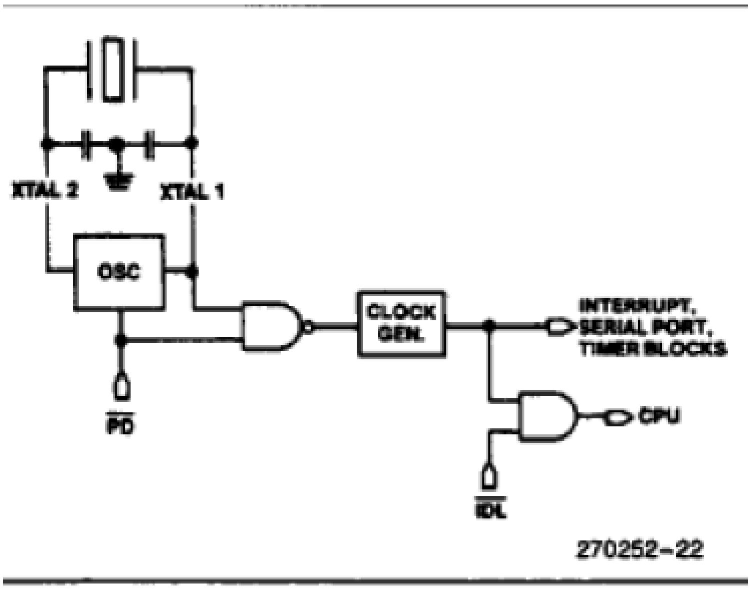
General Purpose RAM
These 80 bytes of Internal RAM memory are available for general-purpose data storage. The general purpose RAM can be accessed using direct or indirect addressing mode instructions.
Special Function Registers (SFRs)

Locations 80H to FFH contain the special function registers. As you can see from the diagram above, not all locations are used by the 8051 (eleven locations are blank). These extra locations are used by other family members (8052, etc.) for the extra features these microcontrollers possess.

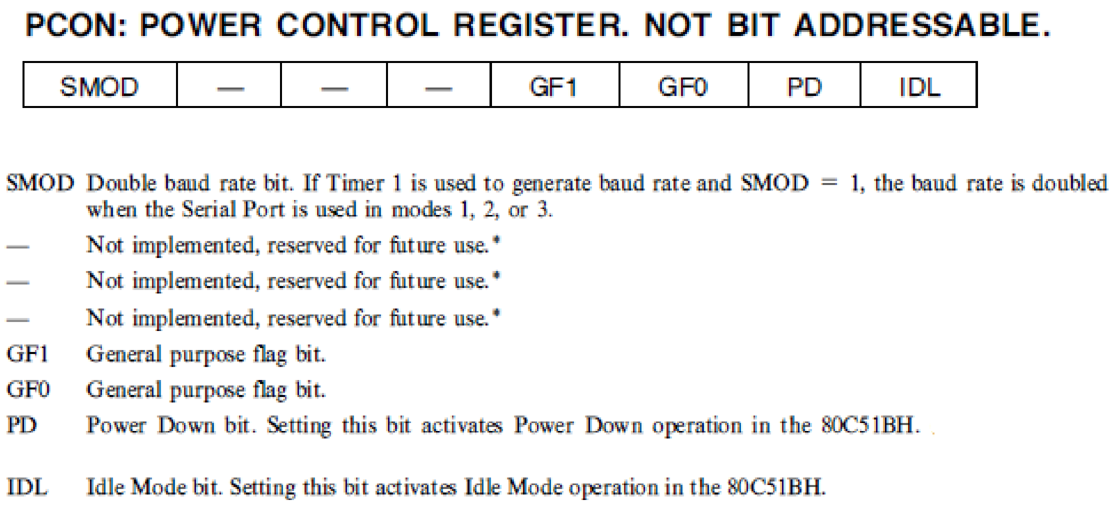
c) Draw schematic diagram to interface stepper motor with 8085 microprocessor.
Ans : correct interfacing diagram: 4 marks.



d) Explain the power saving operation of 8051 in detail.
Ans : Diagram : 1 mark, PCON format: 1mark, Explanation : 2 marks



Format of PCON:



PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD	Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.
—	Not implemented, reserved for future use.*
—	Not implemented, reserved for future use.*
—	Not implemented, reserved for future use.*
GF1	General purpose flag bit.
GF0	General purpose flag bit.
PD	Power Down bit. Setting this bit activates Power Down operation in the 80C51BH.
IDL	Idle Mode bit. Setting this bit activates Idle Mode operation in the 80C51BH.

IDLE MODE

In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions.

The CPU status is preserved in its entirety, the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical state they had at the time idle mode was activated. ALE and PSEN hold at logic high levels.

There are two ways to terminate the idle mode.

i) Activation of any enabled interrupt will cause PCON.O to be cleared and idle mode is terminated.

ii) Hard ware reset: that is signal at RST pin clears IDEAL bit IN PCON register directly. At this time, CPU resumes the program execution from where it left off.

POWER DOWN MODE

An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In the Power Down mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Special Function Register are maintained held. The port pins output the values held by their respective SFRS. ALE and PSEN are held low.

Termination from power down mode: an exit from this mode is hardware reset.

Reset defines all SFRs but doesn't change on chip RAM.

e) Write an assembly language program to get X value from P1 and send X^2 to P2 continuously using look up table.

Ans : correct program : 4 marks.

```
 MOV DPTR,#2000 H      ;initialize memory pointer
  MOV A,P1             ;load X value from P1
 MOVC A,@A+DPTR        ; Copy the code byte found at the ROM address formed
                       by adding A and the DPTR, to A
MOV P2,A              ;Load value X2 from accumulator to P2.
LOOP: AJMP LOOP
```

Addr	content
2000	0^2=0
2001	1^2=1
2002	2^2=4
2003	3^2=9
2004	4^2=16

Q 4) Attempt any three: 12

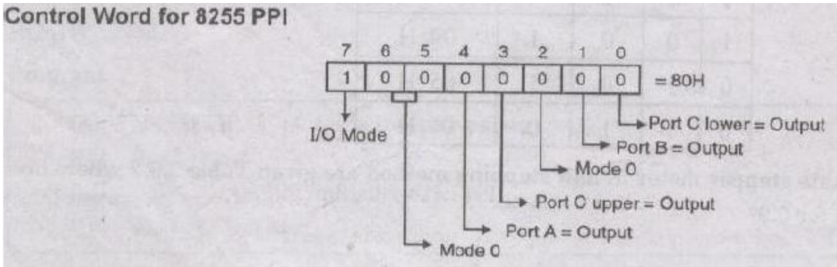
a) Write an assembly language program to rotate stepper motor in clockwise direction using half stepping method.

Ans : correct program : 2 marks, bits pattern : 2 marks

Stepper motor has four windings i.e. A, B, C, D and connected to PA0, PA1, PA2, PA3 of a port of 8255.

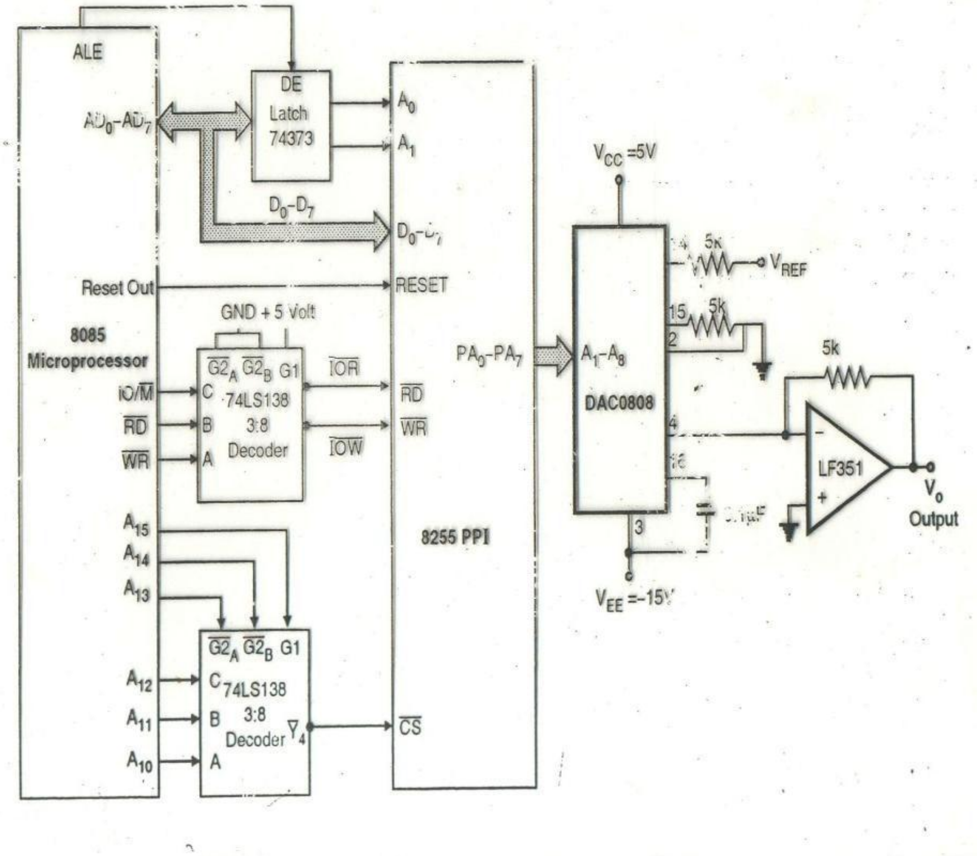
The bits pattern to rotate stepper motor in 1.8 degree i.e. in half stepping method is as follows:

A	B	C	D	CODE
1	0	1	0	0A H
1	0	0	0	08 H
1	0	0	1	09 H
0	0	0	1	01 H
0	1	0	1	05 H
0	1	0	0	04 H
0	1	1	0	06 H
0	0	1	0	02 H
1	0	1	0	0A H



```
         LXI SP, FFE0 H          ; Initialize stack pointer
         MVI A , 80H
         OUT CWR                 ; Initalize 8255 PPI
UP:      LXI H, C200 H           ; Initalize memory pointer for look up table
         MVI C, 08 H             ; Initalize byte counter
UP 1:    MOV A, M                ; Read bits pattern i.e. step code
         OUT PORT_A              ; write to port A
         CALL DELAY              ; add delay between steps
         INX H                   ; increment memory pointer for look up table
         DCR C                   ; DECREMENT BYTE COUNTER BY 1
         JNZ UP1                 ; if byte counter ≠ 0 then go to UP1
         JMP UP                  ; Continuous rotation
```

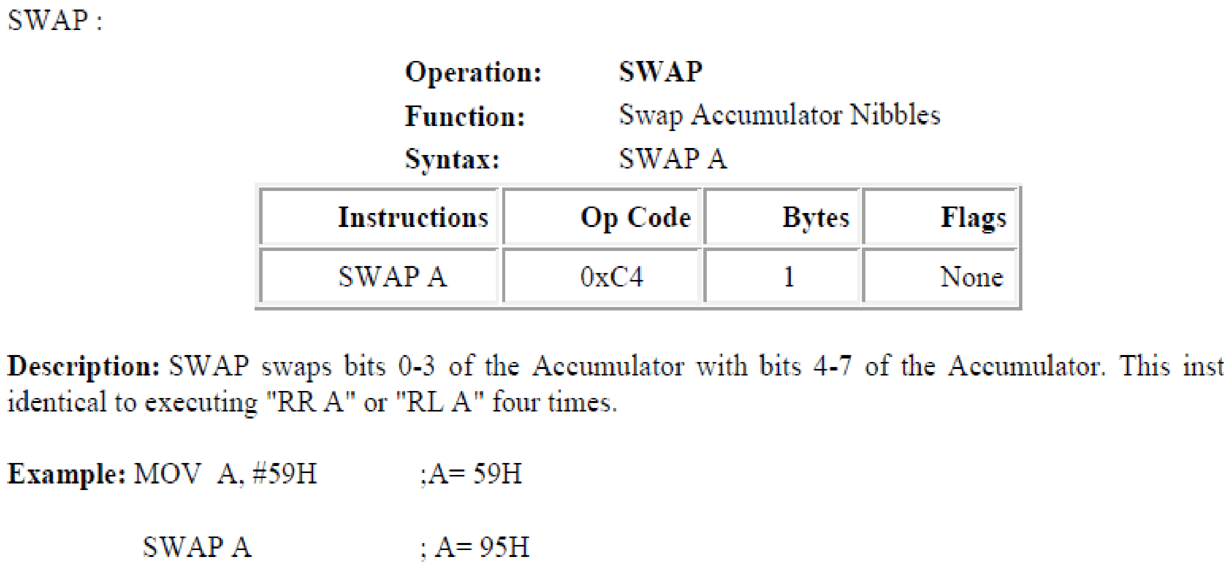
b) Draw complete interfacing diagram of D to A converter with 8085.
Ans : correct interfacing diagram: 4 marks



c) Describe the following instructions of 8051:
Ans : Each instruction: 1 mark

(Note: Only 1 mark for each instruction. so either description in sentence **or** example by considering data, should be given marks.)

i)**SWAP A**



SWAP :

Operation: **SWAP**
Function: Swap Accumulator Nibbles
Syntax: SWAP A

Instructions	Op Code	Bytes	Flags
SWAP A	0xC4	1	None

Description: SWAP swaps bits 0-3 of the Accumulator with bits 4-7 of the Accumulator. This inst identical to executing "RR A" or "RL A" four times.

Example: MOV A, #59H ;A= 59H

SWAP A ; A= 95H

ii)**DIV AB**

Operation: **DIV**
Function: Divide Accumulator by B
Syntax: DIV AB

Instructions	OpCode	Bytes	Flags
DIV AB	0x84	1	C,OV

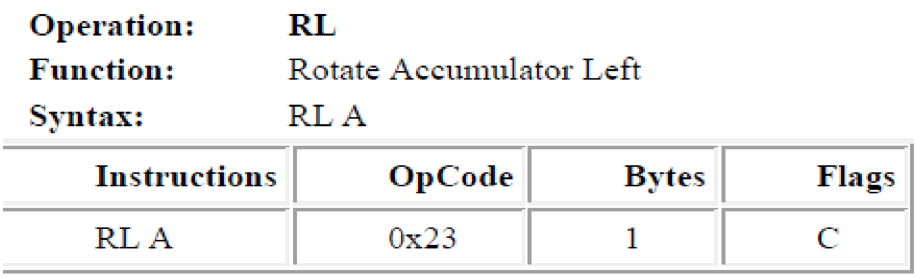
Description: Multiples the unsigned value of the Accumulator by the unsigned value of the "B" register. The least significant byte of the result is placed in the Accumulator and the most-significant-byte is placed in the "B" register.

The **Carry Flag (C)** is always cleared.

The **Overflow Flag (OV)** is set if the result is greater than 255 (if the most-significant byte is not zero), otherwise it is cleared

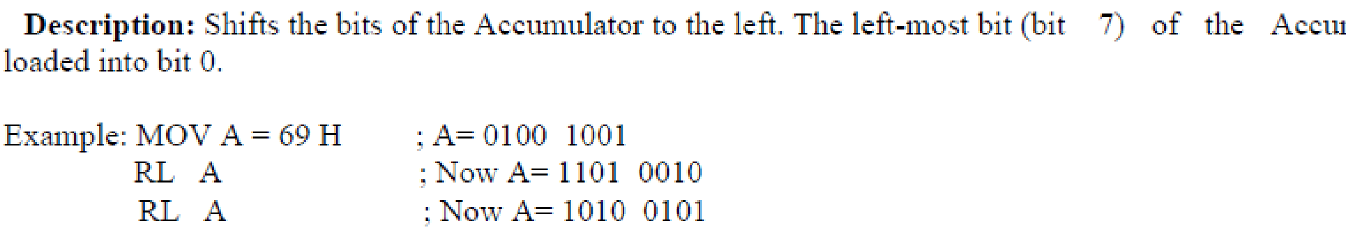
Example: MOV A, #5
MOV B, # 7
DIV AB ; A= 35=23H, B=0

iii) **RL A**



Operation: **RL**
Function: Rotate Accumulator Left
Syntax: RL A

Instructions	OpCode	Bytes	Flags
RL A	0x23	1	C



Description: Shifts the bits of the Accumulator to the left. The left-most bit (bit 7) of the Accu loaded into bit 0.

```
Example: MOV A = 69 H     ; A= 0100  1001
         RL   A           ; Now A= 1101  0010
         RL   A           ; Now A= 1010  0101
```

iv) **RRC A**

Operation: **RRC**
Function: Rotate accumulator to the right through carry

Syntax: RRC A

Instructions	OpCode	Bytes	Flags
RRC A	0x13	1	C,OV

Description: Rotate the contents of accumulator bit by bit to the right through carry.
Example: MOV A=AB H ;A=10101011
RRC A ;NOW A=01010101

d) List the addressing modes of 8051 microcontroller with one suitable instruction each.
Ans: List Addressing modes:-2 Marks, One example of each---- 2 Marks.

There are a number of addressing modes available to the 8051 instruction set, as follows:

1. Immediate Addressing mode
2. Register Addressing mode
3. Direct Addressing mode
4. Register Indirect addressing mode
5. Relative Addressing mode
6. Indexed Addressing mode

1) Immediate Addressing mode:
For example the instruction:
MOV A, #25H ; Load 25H into A

2) Register Addressing Mode:
An example instruction using register addressing is :
ADD A, R5 ; Add the contents of register R5 to contents of A (accumulator)
.

3) Direct Addressing Mode:
For example consider the instruction:
MOV R0, 40H ; Save contents of RAM location 40H in R0.

4) Register Indirect Addressing Mode:
An example instruction, which uses indirect addressing, is as follows:
MOV A, @R0 ; move contents of RAM location whose

5) Relative Addressing Mode:
Consider the following example:
SJMP LABEL_X
An advantage of relative addressing is that the program code is easy to relocate in memory in that the addressing is relative to the position in memory.

6) Indexed Addressing Mode:

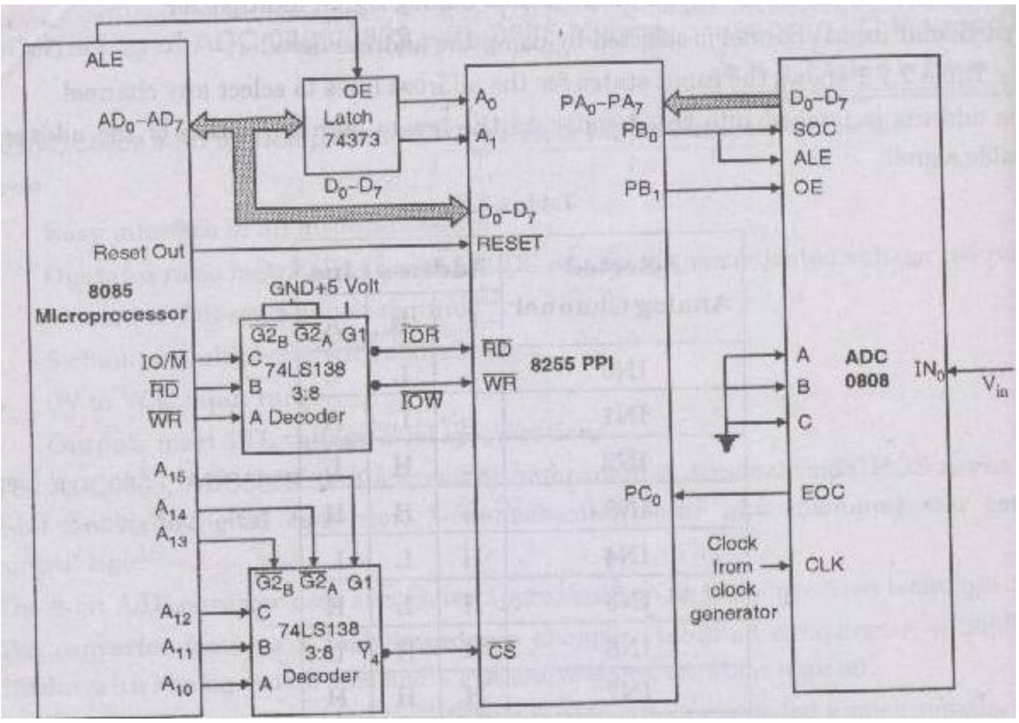
Consider the example instruction:

MOVC A, @A+DPTR
MOVC is a move instruction, which moves data from the external code memory space. The address operand in this example is formed by adding the content of the DPTR register to the accumulator value. Here the DPTR value is referred to as the base address and the accumulator value us referred to as the index address.

Q 4B) Attempt any one: 6

a) Draw complete interfacing diagram of A to D converter with 8085 microprocessor using 8255.Write a program to read analog input.

Ans : Suitable diagram -- 3Marks, Correct program—3 Marks



PROGRAM:

```
MVI A, 98 H       ; Initialize ports of 8255
OUT CWR
MVI A, 01H
OUT Port B        ; Generate A Trigger Pulse For Start Conversion
CALL DELAY
MVI A, 00H
OUT Port B
UP: IN Port C     ; Check For End of Conversion
RLC
JNC UP
MVI A, O2H        ; Send Output Enable to ADC
OUT Port B
IN Port A         ; Read from port A
HLT
```

b) Define directives. Explain the following directives with examples.

i) ORG ii) DB iii) EQU iv) END

i) ORG:- Origin

It is used to indicate the beginning of address.

Syntax:

ORG	Address

The address can be given in either hex or decimal there should be a space of at least one character between ORG & address fields. Some assemblers use ORG should not begin in label field.

ii) DB:- Data Byte

Syntax:

LABEL	DB	BYTE

Where byte is an 8-bit number represented in either binary, Hex, decimal or ASCII form. There should be at least one space between label & DB.

The colon (:) must present after label. This directive can be used at the beginning of program. The label will be used in program instead of actual byte. There should be at least one space between DB & a byte.

iii) EQU: Equate

It is used to define constant without occupying a memory location.

Syntax:

Name	EQU	Constant

By means of this directive, a numeric value is replaced by a symbol. For e.g. MAXIMUM EQU 99

After this directive every appearance of the label “MAXIMUM” in the program, the assembler will

interpret as number 99 (MAXIMUM=99).

iv) END:

This directive must be at the end of every program.meaning that in the source code anything after the

END directive is ignored by the assembler.

This indicates to the assembler the end of the source file(asm).

Once it encounters this directive, the assembler will stop interpreting program into machine code. e.g. END ; End of the program.

Q.5. Attempt any FOUR **16**

a) **Write assembly language program to generate square wave of 2KHz using timer mode of 8051.Assume input clock frequency is 10 KHz.**

Ans:

NOTE: CLOCK FREQUENCY OF MICROCONTROLLER IS IN THE RANGE OF 1MHz TO 16MHzAND GIVEN FRQUENCY IS IN KHz , WHICH IS VERY LOW.SO MARKS SHOULD BE GIVEN FOR THE STEPS.

Ans : **Calculations --- 1Mark, Delay --- 1 Mark, Program --- 2 Marks**

Crystal frequency= 10 KHz

I/P clock = 10 X 10^3 = 10 KHz

1/12x10 KHz = 833.33 Hz

Tin = 1.2msec

For 2 kHz square wave

Fout = 2 KHz

Tout = 1/2 X 103

Tout = 500µ sec

Consider half of it = Tout = 250µ sec=0.250msec

N = Tout / Tin = 0.250/1.2= 0.208ms=0.208*10^{-3} µ sec

65536-XX= YYYY H

Program:

```
ORG 0000
MOV TMOD, #01H ; Mode1
MOV TL0, #0YYH ; Lower byte Of Timer0
MOV TH0, #0YYH ; higher byte Of Timer0
CPL P1.4 ; Toggle P1.4
ACALL DELAY
SJMP HERE

        Delay using Timer0
        DELAY: SETB TR0 ; Start Timer 0
        AGAIN: JNB TF0, AGAIN
        CLR TR0 ; Stop Timer 0
        CLR TF0
        RET
        END
```

b) Describe the function of following handshaking signal of IC8255.

i. IBF **III. $\overline{\text{OBF}}$**

ii. $\overline{\text{ACK}}$ **IV. $\overline{\text{STB}}$**

Ans : Description of each signal : 1mark.

i. IBF (Input buffer full) – If this signal rises to logic 1, it indicates that data has been loaded into latches, i.e. it works as an acknowledgement. IBF is set by a low on STB and is reset by the rising edge of RD input.

ii. $\overline{\text{STB}}$(Strobe input) – If this lines falls to logic low level, the data available at 8-bit input port is loaded into input latches.

iii. $\overline{\text{OBF}}$ (Output buffer full) – This status signal, whenever falls to low, indicates that CPU has written data to the specified output port. The OBF flip-flop will be set by a rising edge of WR signal and reset by a low going edge at the ACK input.

iv. $\overline{\text{ACK}}$ (Acknowledge input) – ACK signal acts as an acknowledgement to be given by an output device. ACK signal, whenever low, informs the CPU that the data transferred by the CPU to the output device through the port is received by the output device.

c) Differentiate between :

i. Microprocessor and microcontroller

ii. RISC and CISC.

Ans : Microprocessor and microcontroller

Consider any four points. (1/2 mark each)

Sr. No	Parameter	Microprocessor	Microcontroller
1.	No. of instructions used	Many instructions to read/ write data to/ from external memory.	Few instruction to read/ write data to/ from external memory
2.	Memory	Do not have inbuilt RAM or ROM.	Inbuilt RAM or ROM
		Program and data are stored in same memory.	Separate memory to store program and data
3.	Registers	Microprocessor contains general purpose registers, Stack pointer register, Program counter register	Microcontroller contains general purpose registers, Stack pointer register, Program counter register additional to that it contains Special Function Registers (SFRs) for Timer , Interrupt and serial communication etc.
4.	Timer	Do not have inbuilt Timer.	Inbuilt Timer
5.	I/O ports	I/O ports are not available requires extra device like 8155 or 8255.	I/O ports are available
6.	Serial port	Do not have inbuilt serial port, requires extra devices like 8250 or 8251.	Inbuilt serial port

7.	Multifunction pins	Less Multifunction pins on IC.	Many multifunction pins on the IC
8.	Boolean Operation	Boolean operation is not possible directly.	Boolean Operation i.e. operation on individual bit is possible directly
9.	Applications	General purpose, Computers and Personal Uses.	Single purpose(dedicated application), Automobile companies, embedded systems, remote control devices.

ii. RISC and CISC

Ans: Consider any four points: (1/2 mark each)

Sr.No	RISC(Reduced Instruction Set Computer)	CISC(Complex Instruction Set Computer)
1	Emphasis on software	Emphasis on hardware
2	Single clock reduced instruction only	Includes multi clock complex instruction
3	Register to Register: „‟Load” and “Store” are independent instructions	Memory to memory:” Load and “Store” incorporated in instructions.
4	Low cycles per second, large code sizes	Small code sizes, high cycles per second
5	Spends more transistors on memory registers	Transistors used for strong complex instructions.

d) Define stack of 8051 microcontroller. Explain PUSH and POP instructions with examples.

Ans: Definition : 1 marks, Explanation : 3 marks.

Stack: The stack is a part of RAM used by the CPU to store information temporarily. This information may either be data or an adress. The CPU needs this storage area as there are only a limited amount of registers. The register used to access the stack is called the stack pointer which is an 8-bit register. So, it can take values of 00 to FF H. When the 8051 is powered up, the SP register contains the value 07, i.e, the RAM location value 08 is the first location being used for the stack by the 8051 controller. There are two important functions to handle this stack. One is the PUSH and other is the POP. The loading of data from the CPU registers to the stack is done by the PUSH and loading of the contents of the stack back into a CPU register is done by POP.

```
EX. MOV R6 ,#35 H
MOV R1 ,#21 H
PUSH 6
PUSH 1
```

In the above instructions the contents of the Registers R6 and R1 are moved to stack and they occupy the 08 and 09 locations of the stack. Now the contents of the SP are incremented by two and it is 0A.

Similarly POP 3 instruction pops the contents of the stack into R3 register. Now the contents of the SP is decremented by 1. In 8051 the RAM locations 08 to IF (24 bytes) can be used for the stack. In any program if we need more than 24 bytes of stack, we can change the SP point to RAM locations 30-75 H. This can be done with the instruction MOV SP,#XX.

e) Write 8051 assembly language program to arrange the given 10 numbers in ascending order.

Ans: Correct program : 4 marks

```
        MOV R0, #0AH           ; Initialize the pass counter
UP1:     MOV DPTR, #3000H   ; Initialize the memory pointer
        MOV R1, #0AH           ; Initialize the byte counter
UP:      MOV R2, DPL           ; Save the lower byte address
        MOVX A, @DPTR          ; Read number from array
        MOV 0F0H, A            ; Copy number to B register
        INC DPTR               ; Increment memory pointer
        MOVX A,@DPTR           ; Read next number from array
        CJNE A , 0F0, DWN      ; Compare  number with next number
        AJMP SKIP
DWN:    JNC SKIP               ; If number> next number then go to SKIP
        MOV DPL, R2            ; Else exchange the number with next number
        MOVX @DPTR, A
        INC DPTR
        MOV A, 0F0H
        MOVX @DPTR, A
SKIP:     DJNZ R1 UP           ; Decrement byte counter if not zero, go to UP
        DJNZ R0 UP1            ; Decrement pass counter if not zero, go to UP1
LOOP:    AJMP LOOP             ; Stop
```

f) What is in register A after the execution of the following code?

```
        MOV A,# 85H
        SWAP A
        ANL A,#0F0H
```

Ans. 4 marks.

MOV A, # 85H ------ Contents in A = 85H (1 mark)

SWAP A ------- Contents in A = 58H (1 ½ mark)

ANL A, #0F0H ------- Contents in A = 50H (1 ½ mark)

Q.6. Attempt any FOUR **16**

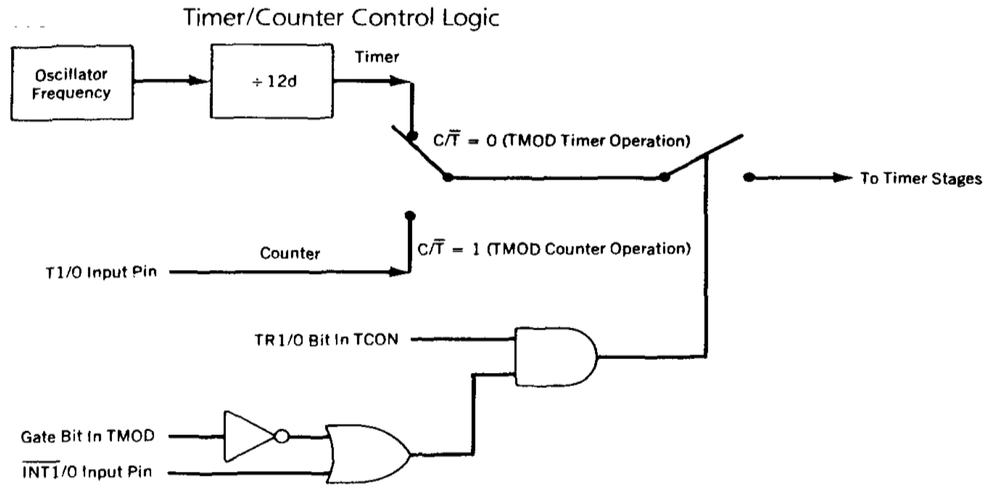
a) **Write the function of editor , assembler.**
Ans: Function of Editor : 2 marks, Function of Assembler: 2 marks.

1) Editor: An editor is a program which helps you to construct your assembly language program in right format so that the assembler will translate it correctly to machine language. So, you can type your program using editor. This form of your program is called as source program and extension of program must be .asm or .src depending on which assembler is used.
The DOS based editor such as EDIT , Wordstar, and Norton Editor etc. can be used to type your program.

2) Assembler: An assembler is programs that translate assembly language program to the correct binary code for each instruction i.e. machine code and generate the file called as Object file with extension .obj and list file with extension .lst extension. Some examples of assembler are ASEM-51, Keil‟s A51, AX 51 and C51, Intel PL/M-51 etc.

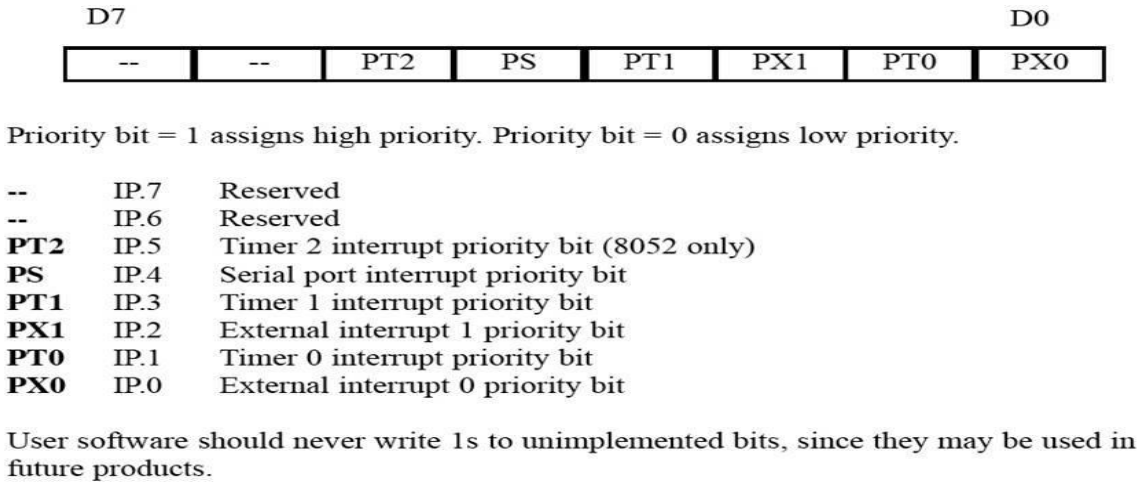
b) **Describe the timer and counter mode of 8051.**
Ans: Diagram : 2 marks, Description : 2 marks.

8051 has two 16-bit programmable UP timers/counters. They can be configured to operate either as timers or as event counters. The names of the two counters are T0 and T1 respectively. The timer/ content is available in four 8-bit special function registers, viz. TL0,TH0, TL1 and TH1 respectively. In the "timer" function mode, the counter is incremented in every machine cycle. Thus, one can think of it as counting machine cycles. Hence the clock rate is 1/12th of the oscillator frequency. In the "counter" function mode, the register is incremented in response to a 1 to 0 transition at its corresponding external input pin (T0 or T1). It requires 2 machine cycles to detect a high to low transition. Hence maximum count rate is 1/24th of oscillator frequency. The operation of the timers/counters is controlled by two special function registers, TMOD and TCON respectively.



c) Draw and explain IE and IP register formats in detail.

Ans: **IP register format –2Marks**



D7 D0

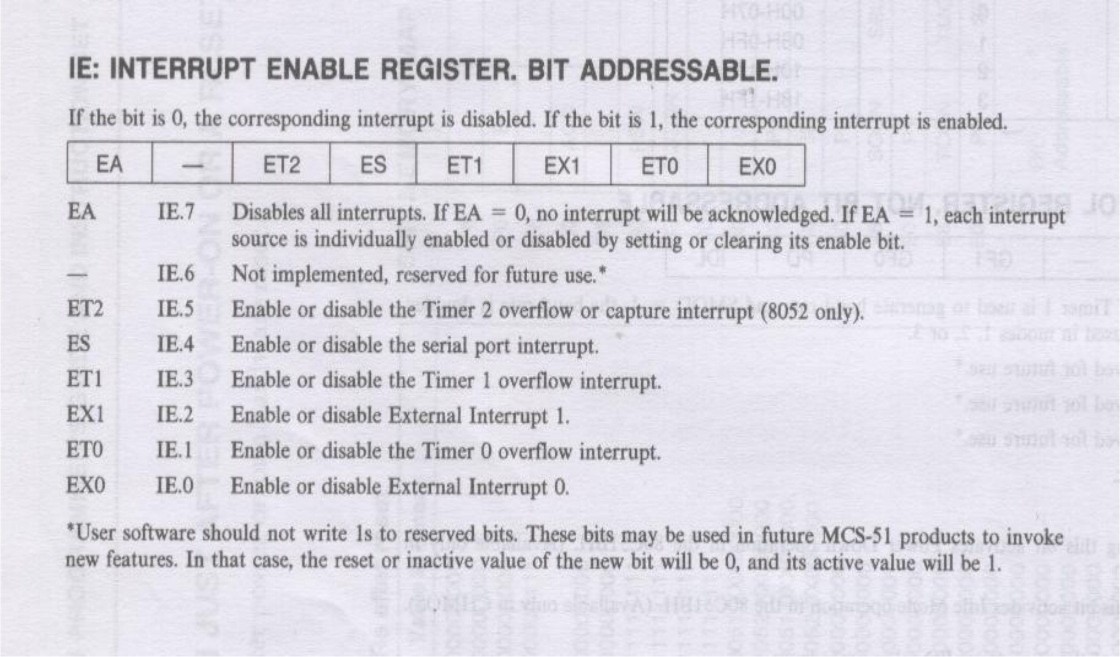
--	--	PT2	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority. Priority bit = 0 assigns low priority.

--	IP.7	Reserved
--	IP.6	Reserved
PT2	IP.5	Timer 2 interrupt priority bit (8052 only)
PS	IP.4	Serial port interrupt priority bit
PT1	IP.3	Timer 1 interrupt priority bit
PX1	IP.2	External interrupt 1 priority bit
PT0	IP.1	Timer 0 interrupt priority bit
PX0	IP.0	External interrupt 0 priority bit

User software should never write 1s to unimplemented bits, since they may be used in future products.

IE Register Format- 2 Marks.



IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	—	ET2	ES	ET1	EX1	ET0	EX0

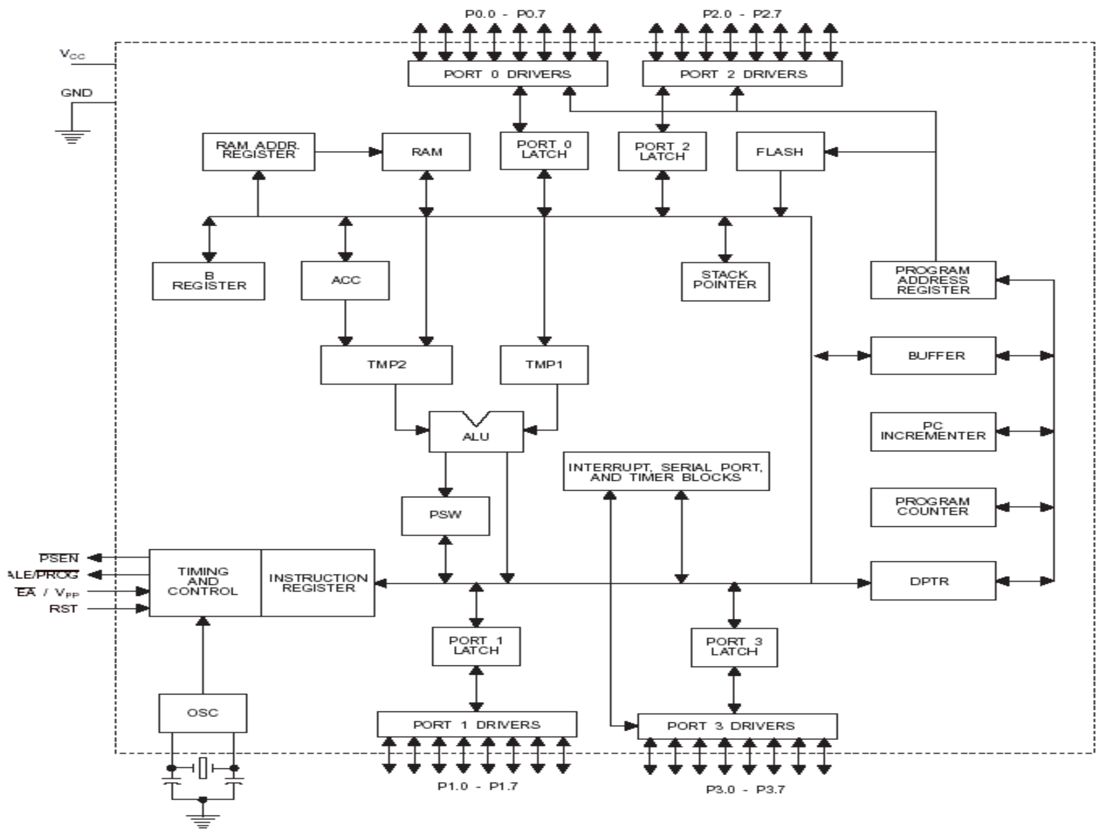
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Not implemented, reserved for future use.*
ET2	IE.5	Enable or disable the Timer 2 overflow or capture interrupt (8052 only).
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

d) Draw architecture diagram of 8051 microcontroller.

Ans: Diagram : 4 marks

Block Diagram



e) List the alternate functions of port 3 of 8051.

Ans : ½ mark each.

Pin	Name	Alternate Function
P3.0	RXD	Serial input line
P3.1	TXD	Serial output line
P3.2	$\overline{\text{INT0}}$	External interrupt 0
P3.3	$\overline{\text{INT1}}$	External interrupt 1
P3.4	T0	Timer 0 external input
P3.5	T1	Timer 0 external input
P3.6	$\overline{\text{WR}}$	External data memory write strobe
P3.7	$\overline{\text{RD}}$	External data memory read strobe