MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

17509

WINTER– 16 EXAMINATION **(Subject Code: 17509)**

Model Answer

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.

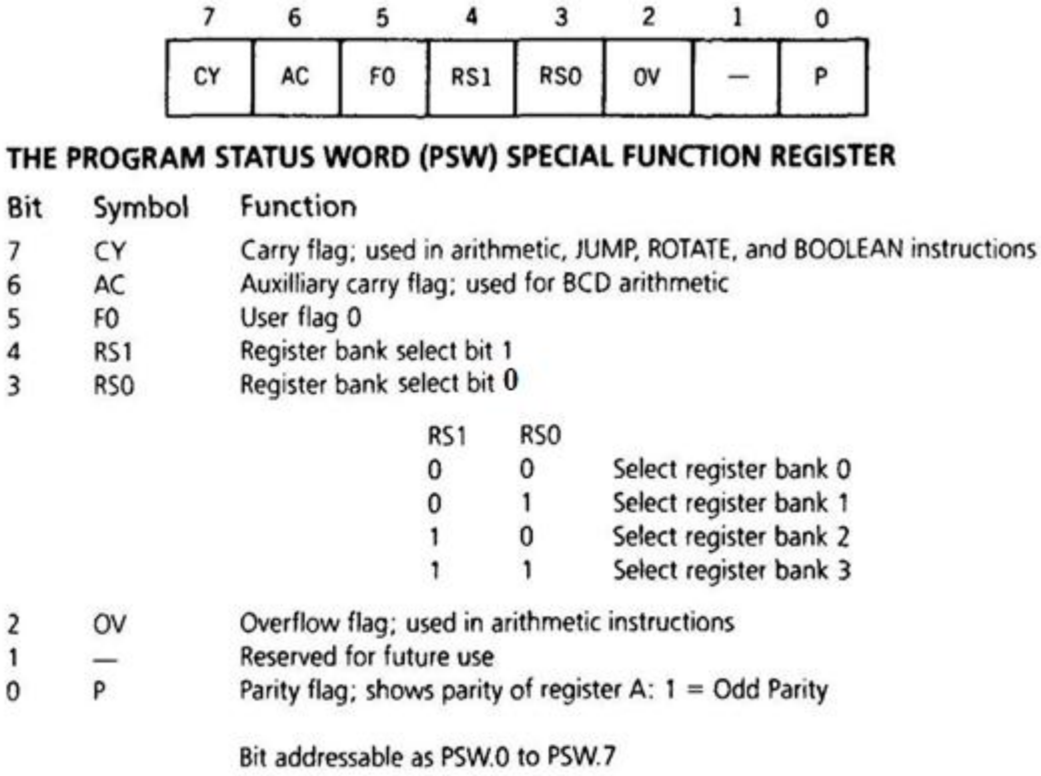
Q. No.	Sub Q.N.	Answer	Marking Scheme
Q.1	**A**	**Attempt any three:**	**12-Total Marks**
1	**a)**	**State any eight features of 8051 microcontroller.**	**4 M**
	Ans:	• It is an 8bit microcontroller. • 8bit accumulator, 8bit Register and 8bit ALU. • On chip RAM 128 bites (data memory). • On chip ROM 4 Kbytes (program memory). • Two 16bit counter/ timer. • A 16 bit DPTR(data pointer) • Two levels of interrupt priority. • 4 byte bi-directional input/ output port. • Power saving mode (on some derivatives). • 16bit address bus:-it can access 2^16 memory locations:-64kb (65536) each of RAM and ROM. • It is an inclusion of Boolean processing system, have an ability to allow logic operations to be carried out on registers and RAM. • 8bit data bus:-it can access 8bit of data in one operation. • It also consist of 3 internal and two external interrupts • UART (this serial communication port makes chip to use simply as a serial communication interface) • It has four separate Register set. (Each contains 8 Registers (R0 to R7)).	**½ M Each**

	b)	**Draw format of PSW register in 8051 microcontroller and state significance of each bit.**	**4 M**

Ans:

Format: 2M

Significance:2M



7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	—	P

THE PROGRAM STATUS WORD (PSW) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7	CY	Carry flag; used in arithmetic, JUMP, ROTATE, and BOOLEAN instructions
6	AC	Auxilliary carry flag; used for BCD arithmetic
5	F0	User flag 0
4	RS1	Register bank select bit 1
3	RS0	Register bank select bit 0

RS1	RS0	
0	0	Select register bank 0
0	1	Select register bank 1
1	0	Select register bank 2
1	1	Select register bank 3

Bit	Symbol	Function
2	OV	Overflow flag; used in arithmetic instructions
1	—	Reserved for future use
0	P	Parity flag; shows parity of register A: 1 = Odd Parity

Bit addressable as PSW.0 to PSW.7

PSW register is one of the most important SFRs. It contains several status bits that reflect the current state of the CPU. Besides, this register contains Carry bit, Auxiliary Carry, two register bank select bits, Overflow flag, parity bit and user-definable status flag.

P - Parity bit. If a number stored in the accumulator is even then this bit will be automatically set (1), otherwise it will be cleared (0). It is mainly used during data transmit and receive via serial communication.

- Bit 1. This bit is intended to be used in the future versions of microcontrollers.

OV Overflow 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. The overflow flag is only used to detect errors in signed arithmetic operations

RS0, RS1 - Register bank select bits. These two bits are used to select one of four register banks of RAM. By setting and clearing these bits, registers R0-R7 are stored in one of four banks of RAM.

RS1	RS0	SPACE IN RAM
0	0	Bank0 00h-07h
0	1	Bank1 08h-0Fh
1	0	Bank2 10h-17h
1	1	Bank3 18h-1Fh

F0 - Flag 0. This is a general-purpose bit available for user.

AC - Auxiliary Carry Flag is used for BCD operations only. If there is a carry from D3 and D4 during an ADD or SUB operation, this bit is set.

CY - Carry Flag is set whenever there is a carry out from D7 bit. It is affected after all arithmetical operations and shift instructions.

It also can be set to 1 or 0 using instructions SETB c and CLR c

c) **Explain the interfacing of 3x 3 key matrix with 8051 microcontroller.** **4 M**

Ans:

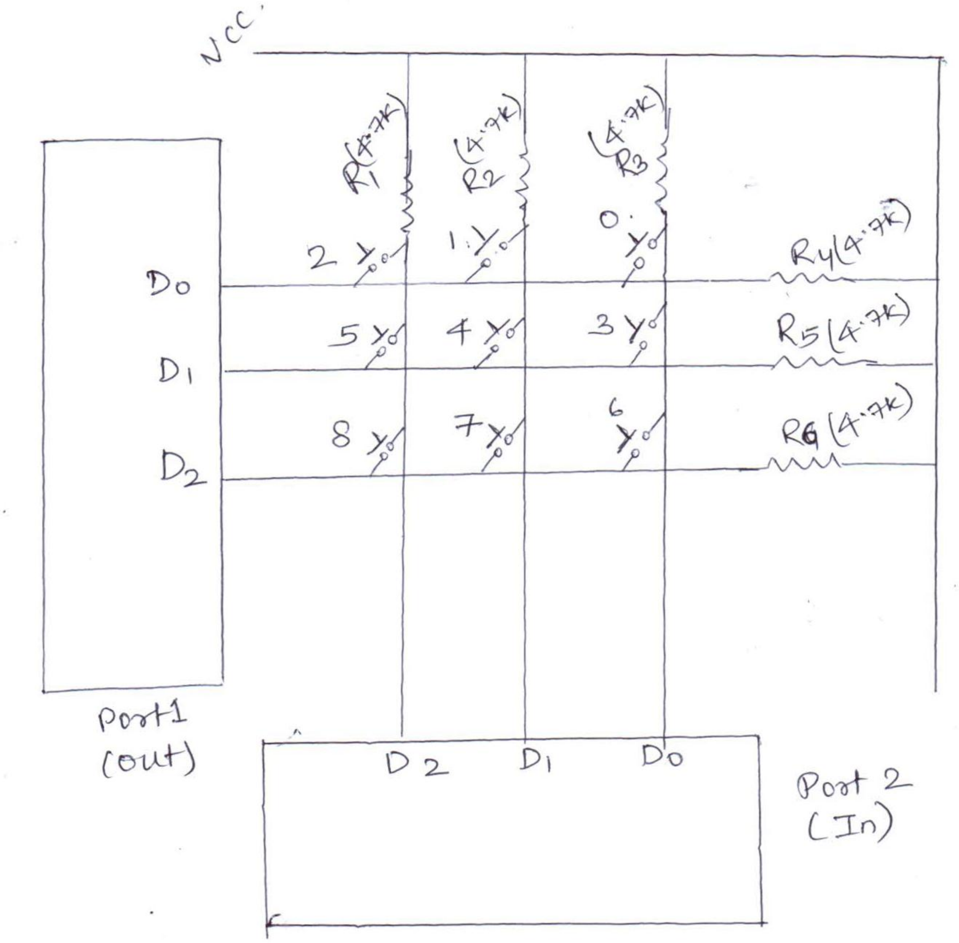


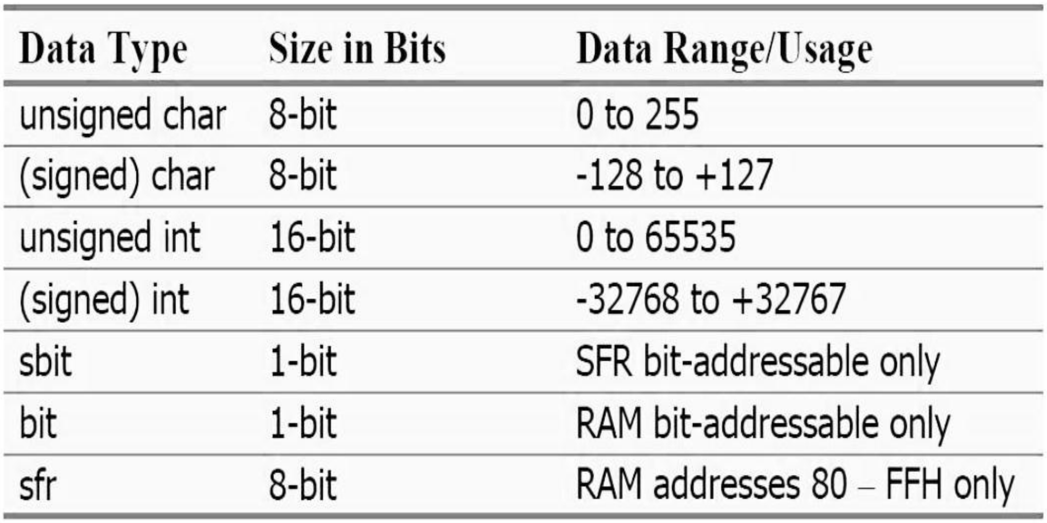
Diagram :2M Explanation:2M

Explanation:

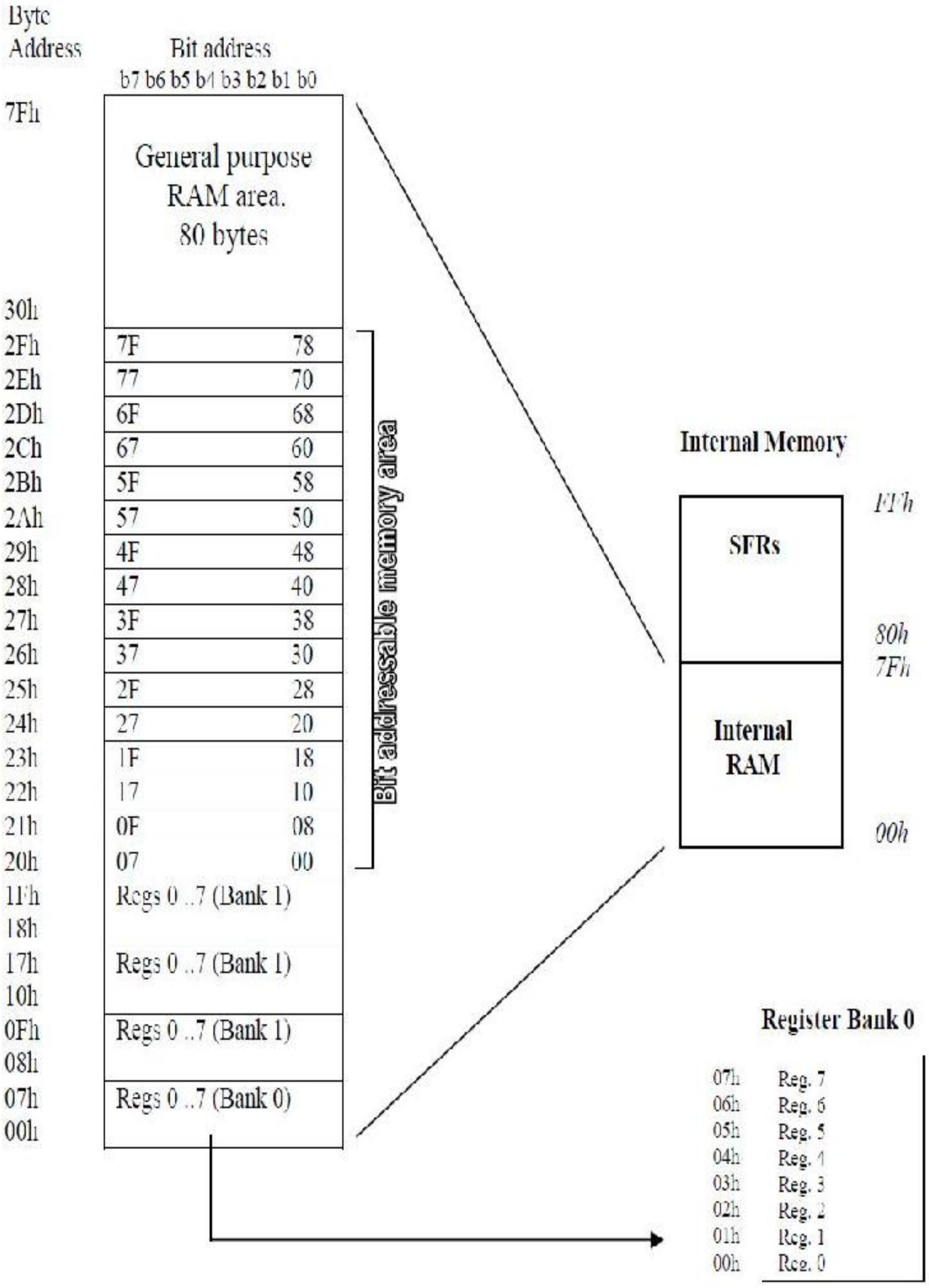
- It is the function of the microcontroller to scan the keyboard continuously to detect and identify the key pressed
- First microcontroller checks whether all keys are open before the start of operation, by grounding all the rows and ensuring that the input port has 111.
- To detect a pressed key, the microcontroller again grounds all rows by providing 0 to the output latch, then it reads the columns
- If the data read from columns is D2 – D0 = 111, no key has been pressed and the process continues till key press is detected
- If one of the column bits has a zero, this means that a key press has occurred
- Then the program waits for 20ms (debounce time) and check for the key pressed again. If still the key pressed is detected, it proceeds.
- After detecting a key press, microcontroller will go through the process of identifying the key
- Starting with the top row, the microcontroller grounds it by providing a low to row D0 only
- It reads the columns, if the data read is all 1s, no key in that row is activated and the process is moved to the next row
- It grounds the next row, reads the columns, and checks for any zero
- This process continues until the row is identified
- After identification of the row in which the key has been pressed Find out which column the pressed key belongs to
- Upon finding the row that the key press belongs to, it sets up the starting address for the look-up table holding the scan codes (or ASCII) for that row
- To identify the key press, it rotates the column bits, one bit at a time, into the carry flag and checks to see if it is low
- Upon finding the zero, it pulls out the ASCII code for that key from the look-up table
- Otherwise, it increments the pointer to point to the next element of the look-up table

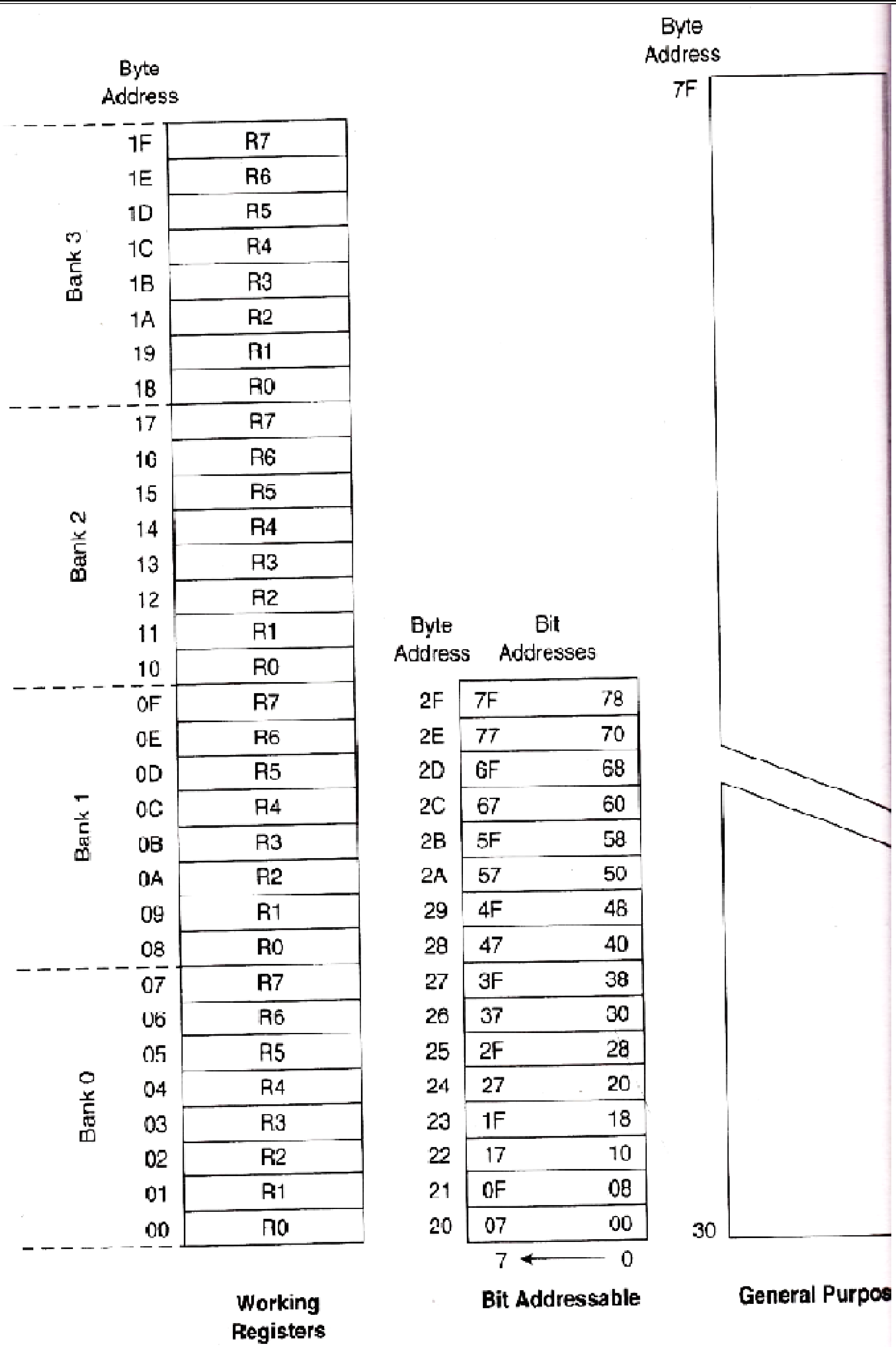
d) **State any four C data types with their value range.** **4 M**

1 M Each

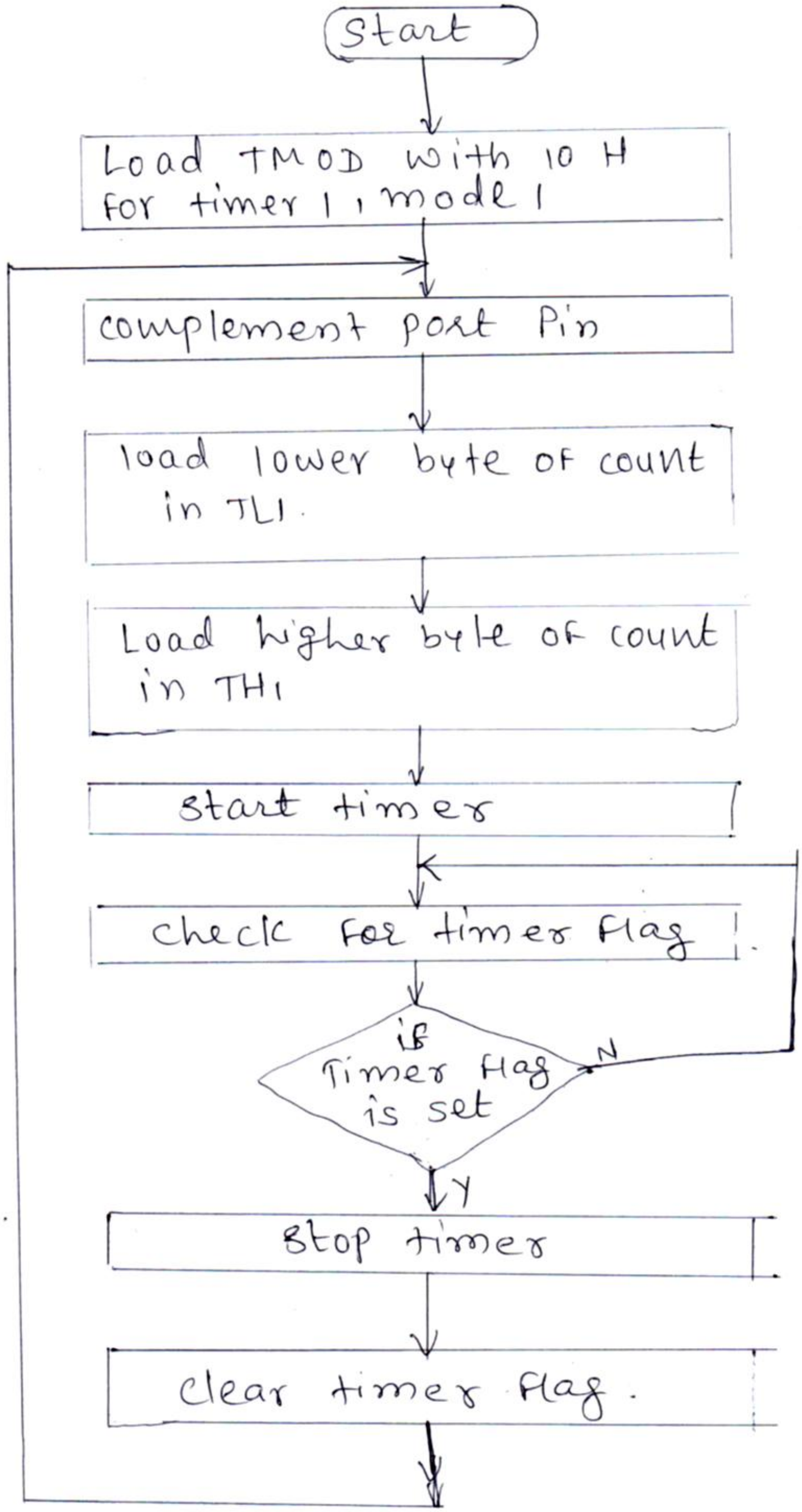


Data Type	Size in Bits	Data Range/Usage
unsigned char	8-bit	0 to 255
(signed) char	8-bit	-128 to +127
unsigned int	16-bit	0 to 65535
(signed) int	16-bit	-32768 to +32767
sbit	1-bit	SFR bit-addressable only
bit	1-bit	RAM bit-addressable only
sfr	8-bit	RAM addresses 80 – FFH only

B)	**Attempt any one:**	**6 M**
a)	**Draw internal RAM organization of microcontroller 8051 and show address areas for each Section.**	**6 M**
Ans:	 Byte Address Bit address b7 b6 b5 b4 b3 b2 b1 b0 7Fh General purpose RAM area. 80 bytes 30h 2Fh 7F 78 2Eh 77 70 2Dh 6F 68 2Ch 67 60 2Bh 5F 58 2Ah 57 50 29h 4F 48 28h 47 40 27h 3F 38 26h 37 30 25h 2F 28 24h 27 20 23h 1F 18 22h 17 10 21h 0F 08 20h 07 00 Bit addressable memory area 1Fh Regs 0 ..7 (Bank 1) 18h 17h Regs 0 ..7 (Bank 1) 10h 0Fh Regs 0 ..7 (Bank 1) 08h 07h Regs 0 ..7 (Bank 0) 00h Internal Memory SFRs FFh 80h 7Fh Internal RAM 00h Register Bank 0 07h Reg. 7 06h Reg. 6 05h Reg. 5 04h Reg. 4 03h Reg. 3 02h Reg. 2 01h Reg. 1 00h Reg. 0	**6M**


Byte Address
Bank 3
1F R7
1E R6
1D R5
1C R4
1B R3
1A R2
19 R1
18 R0
Bank 2
17 R7
16 R6
15 R5
14 R4
13 R3
12 R2
11 R1
10 R0
Bank 1
0F R7
0E R6
0D R5
0C R4
0B R3
0A R2
09 R1
08 R0
Bank 0
07 R7
06 R6
05 R5
04 R4
03 R3
02 R2
01 R1
00 R0
Working Registers
Byte Address
Bit Addresses
2F 7F 78
2E 77 70
2D 6F 68
2C 67 60
2B 5F 58
2A 57 50
29 4F 48
28 47 40
27 3F 38
26 37 30
25 2F 28
24 27 20
23 1F 18
22 17 10
21 0F 08
20 07 00
7 ← 0
Bit Addressable
Byte Address
7F
30
General Purpos

	b)	**Write an ALP to find the largest number in an array of 10 numbers stored in internal RAM**	**6 M**
	Ans:	ORG 0000H MOV R1,#0AH MOV R0,#40H DEC R1 MOV 60H, @R0 UP: INC R0 MOV A, @R0 CJNE A,60H,DN AJMP LARGE DN: JC LARGE MOV 60H, A LARGE: DJNZ R1, UP END **(Any other relevant correct logic can be given full marks)**	**6M**
Q 2		**Attempt any two:**	**16M**
	a)	**Write a assembly language program for 8051 microcontroller to generate a delay of 1 second. Use timerl.Assume crystal frequency =12 MHz. Draw flowchart.**	**8M**
	Ans:	**Calculations:** Crystal freq = 12MHz Timer frequency= 12MHz/12= 1MHz Input Time= 1/1MHz= 1us For delay of 50ms 50ms/1us= 50000 Therefore count to be loaded in TH1 and TL1 can be calculated as 65536-50000= 15536d = 3CB0H **Program:** MOV TMOD, #10H ; Timer 1 , mode 1 HERE: MOV R0, #20 ; counter for 1s delay (50ms*20=1sec) CPL P2.0 ; complement P2.0 BACK: MOV TL1, #0B0H ; load count value in TL1 MOV TH1, #3CH ; load count value in TH1 SETB TR1 ; start Timer1 AGAIN: JNB TF1, AGAIN ; stay until timer rolls over CLR TR1 ; stop timer CLR TF1 ; clear timer flag DJNZ R0, BACK ; if R0 is not equal to 0,reload timer SJMP HERE ; repeat	**calculation:1m, Program: 3M, comments:1M flowchart:3M**

		Start Load TMOD with 10 H for timer 1, mode 1 complement port Pin load lower byte of count in TL1. Load higher byte of count in TH1 start timer check for timer Flag if Timer Flag is set N Y Stop timer Clear timer Flag.	

b) **Draw interfacing diagram of 8 bit serialADC, MAX112 with 8051 microcontroller.**

Write aC language program to read data.

8M

Ans:

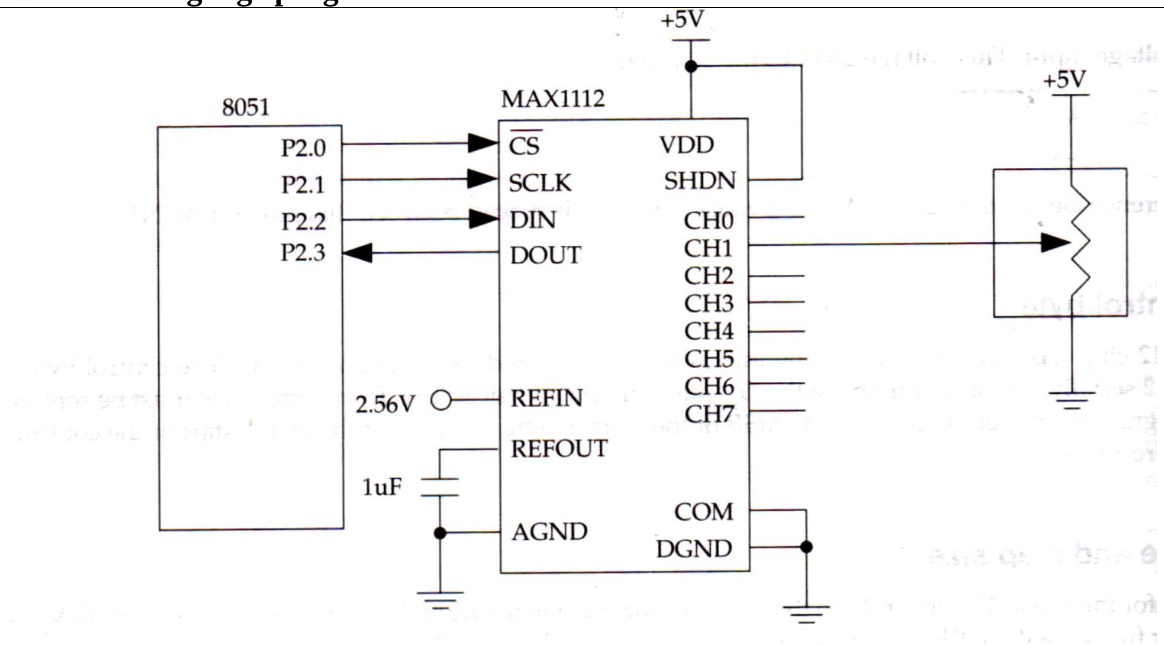
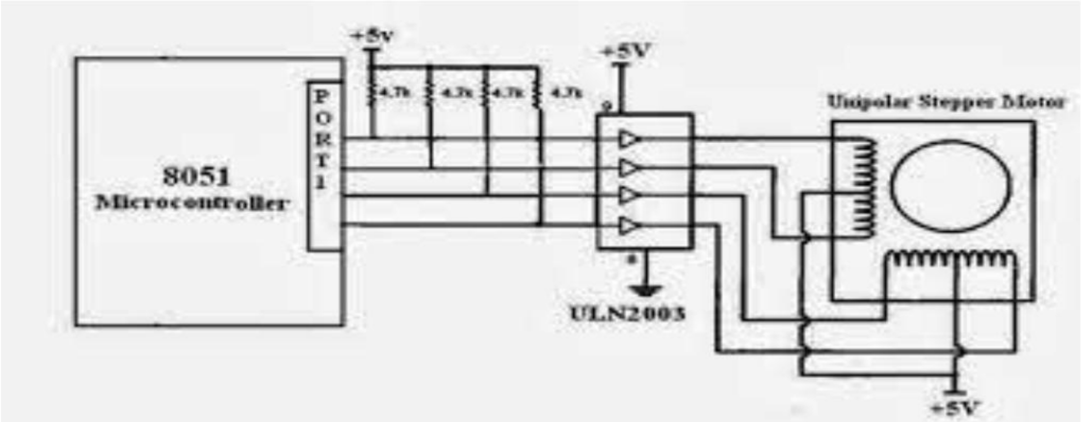
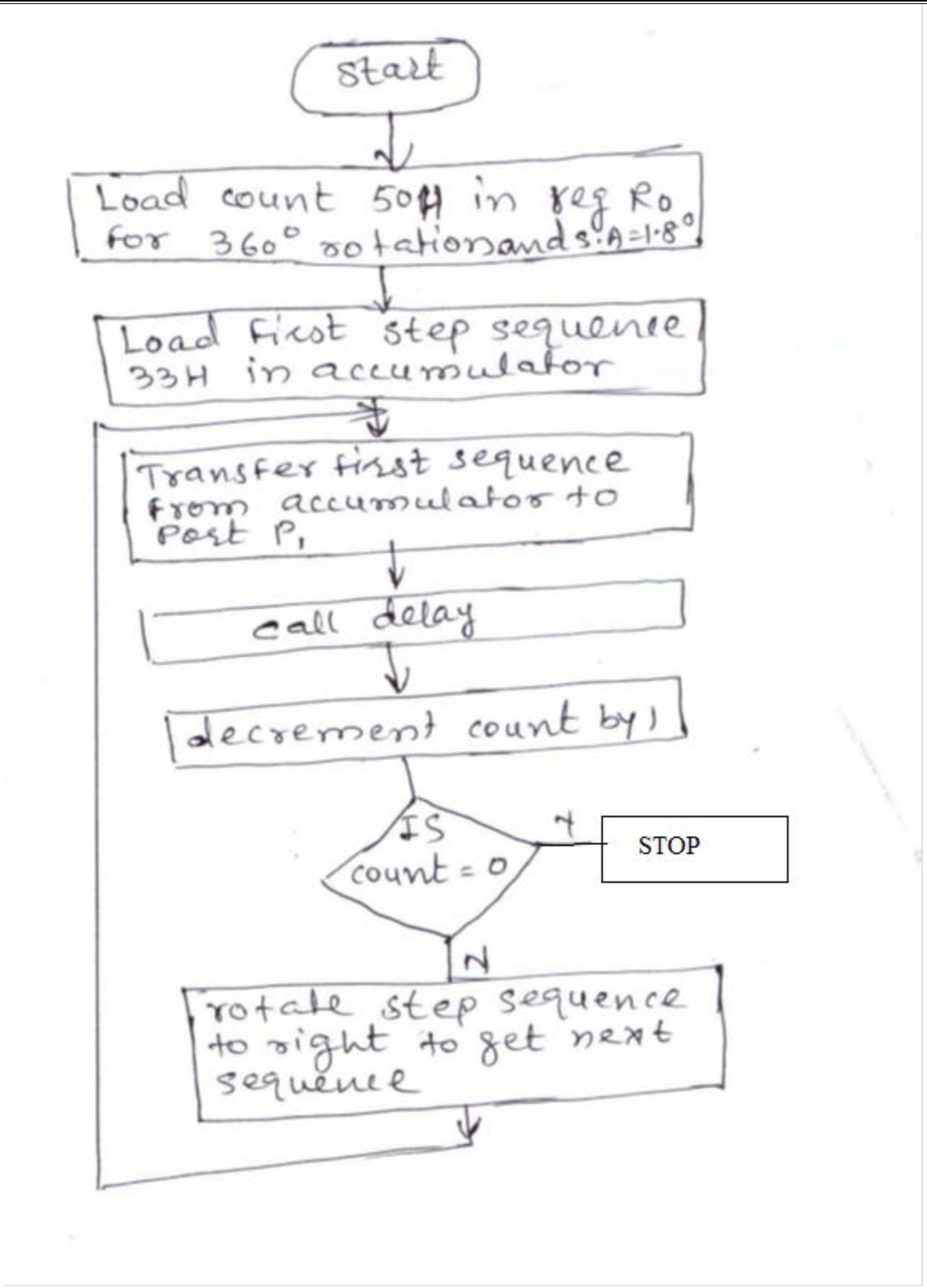


Diagram :4M
Program :4M

<u>Program:</u>

```
# include <reg51.h>

sbit CS =P2^0;

sbit SCLK =P2^1;

sbit DIN =P2^2;

sbit DOUT=P2^3;

sbit LSBRA = ACC^0;

void main( )

{

Unsigned char x;

CS=0; //select max1112

SCLK =1; //an extra H to L pulse

Delay( );

SCLK=0;
Delay( );
for (x=0; x<8; x++) //get all 8 bits
{
SCLK=1
Delay( );
SCLK=0;
Delay ( )
LSBRA=DOUT; // bring in bit from DOUT
//pin to DO of Reg A
ACC=ACC <<1; //Keep shifting data
// for all 8 bits
}
CS=1; //deselect ADC
P1=ACC; //display data on P1
}
```

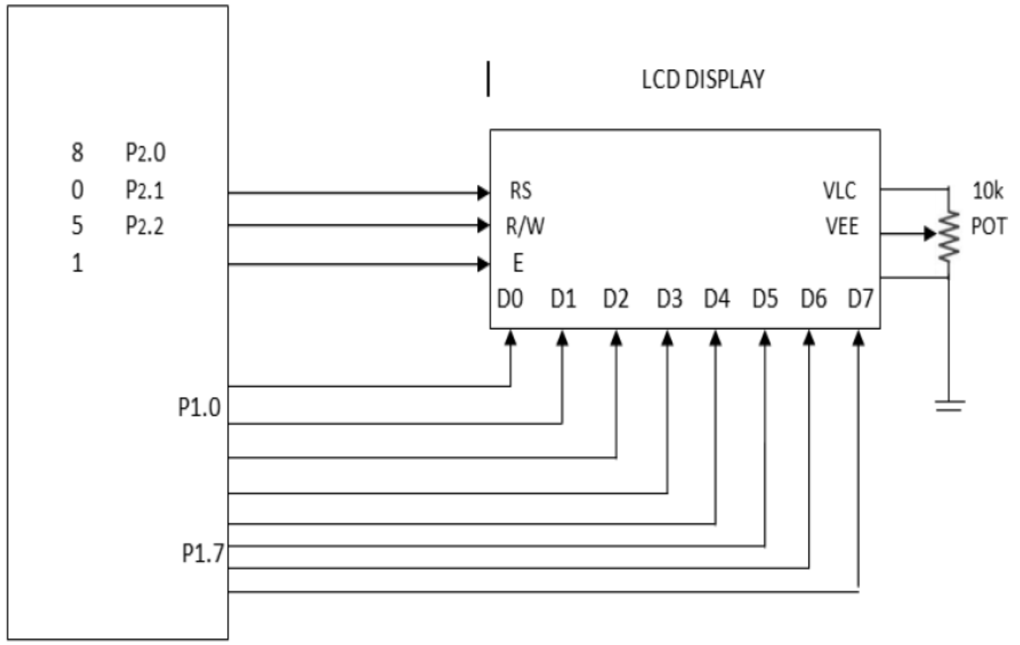
c)	**Draw interfacing diagram of stepper motor control with 8051 microcontroller. Draw flow chart to rotate a stepper motor clockwise through 360°.Assume step angle of 1.8 ° .**	**8M**
Ans:	 +5v 4.7k 4.7k 4.7k 4.7k +5V 9 P O R T 1 8051 Microcontroller Unipolar Stepper Motor 8 ULN2003 +5V	**Diagram :4M, Flowchart:4M**


Start
Load count 50H in reg R0 for 360° rotation and S.A=1.8°
Load first step sequence 33H in accumulator
Transfer first sequence from accumulator to Port P1
call delay
decrement count by 1
IS count = 0
Y
STOP
N
rotate step sequence to right to get next sequence

Q. 3 **Attempt any four:** **16M**

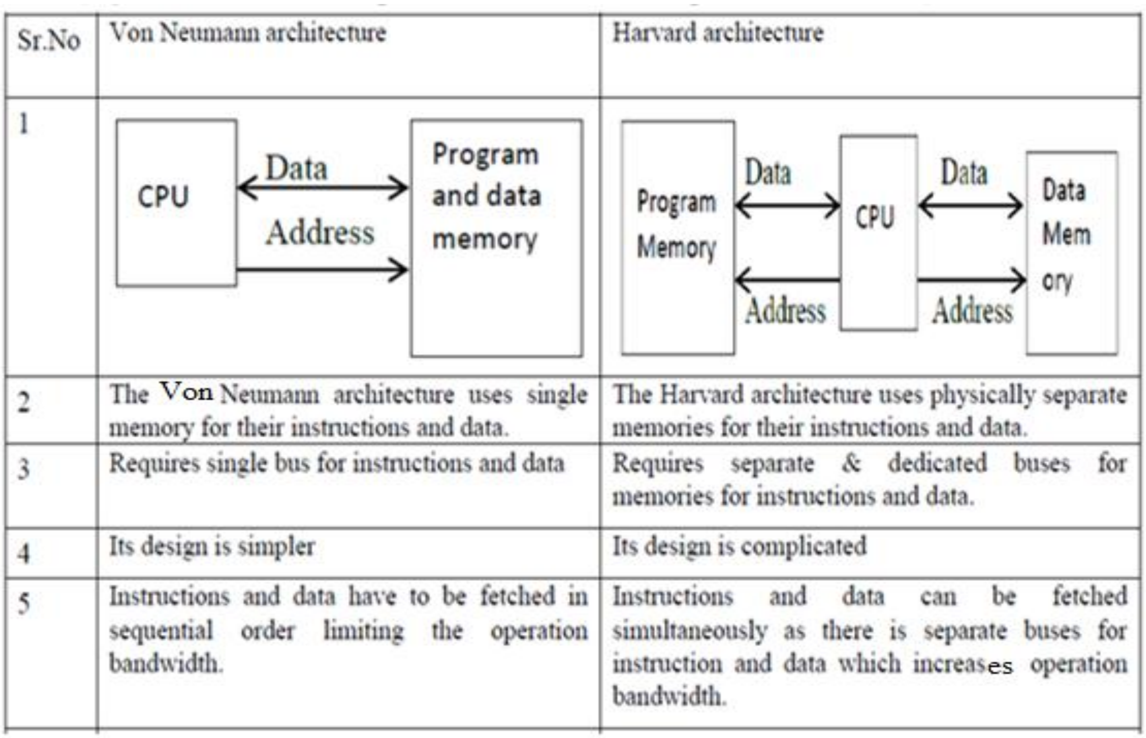
a) **Draw neat labeled interfacing diagram of LCD with 8051 microcontroller.** **4M**

Ans: **Diagram :4M**



b) **Compare Von-Neumann and HarwardArchitecture. Give examples.** **4M**

Ans: **Any four points:4 M**



Sr.No	Von Neumann architecture	Harvard architecture
1	CPU Data Address Program and data memory	Program Memory Data Address CPU Data Address Data Mem ory
2	The Von 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 increases operation bandwidth.

c) Write down instructions to READ input port and send hex data to output port using "C"operators. **4M**

Ans: **4M**

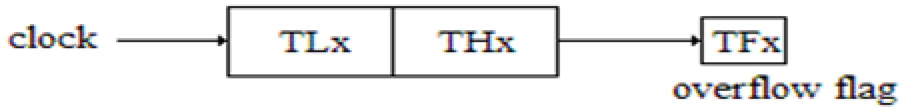
```
#include <reg51.h>
void main(void)
{
P1=0xFF; //make P1 input port
P3=0x00; //make P0 output port
while (1)
{
P3=P1;
}
}
```

Note: Any other correct program logic can be given marks.

d) Describe timer operations of 8051 microcontroller in mode 1 and mode 2 with respect to application and advantages. **4M**

Ans: **Mode1:2 M**
Mode2:2 M

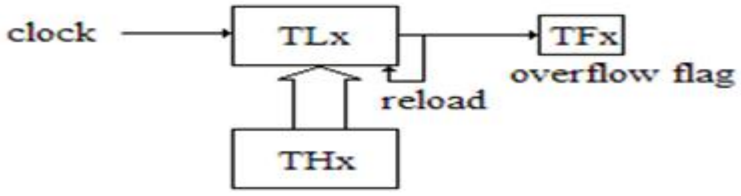
Mode 1 - 16-bit mode



mode 1

The high byte (THx) is cascaded with the low byte (TLx) to produce a 16-bit timer. This timer counts from 0000H to FFFFH - it has 2^{16} (65,536) states. An overflow occurs during the FFFFH to 0000H transition, setting the overflow flag.This is a very commonly used mode to generate time delay using 16bit count.

Mode 2- 8-bit auto-reload mode



mode 2

The timer low byte (TLx) operates as an 8-bit timer (counting to FFH) while the high-byte (Tl value. When the timer overflows from FFH, rather than starting again from 00H, the value in T TLx and the count continues from there. The auto-reload mode is very commonly used for ger .which is used for Serial Communication.

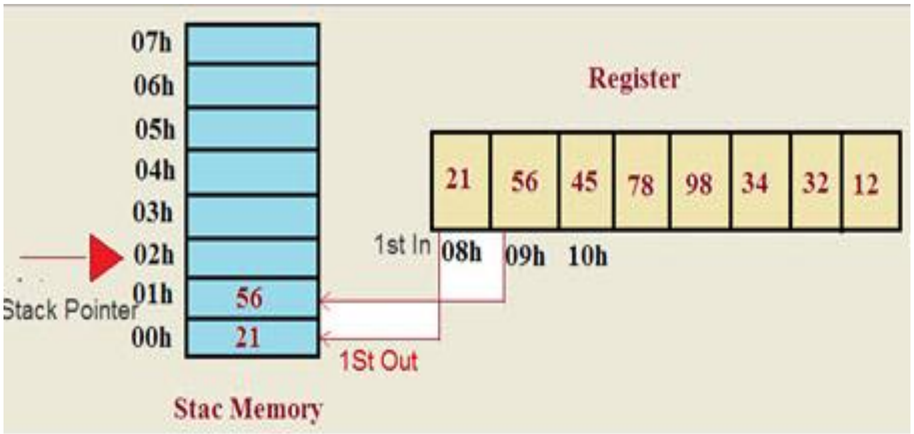
e) **Describe stack operations in 8051 microcontroller with suitable examples.** **4M**

Ans: Stack Memory is a part of internal data memory which is used for temporary data storage in 8051.Stack pointer is used to point to stack memory. Data is stored in stack memory using PUSH instruction and is retrieved using POP instruction.

PUSH operation:2M
POP operation :2M

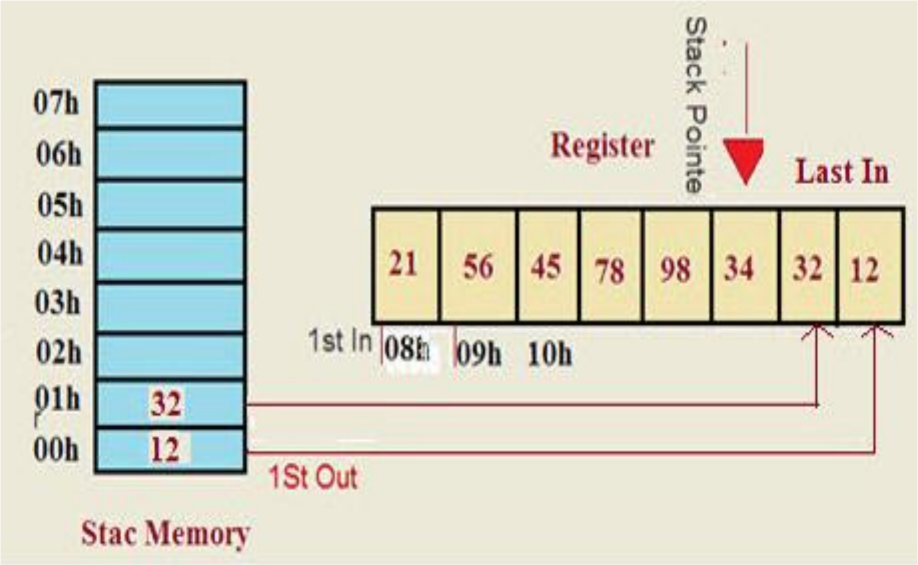
PUSH operation

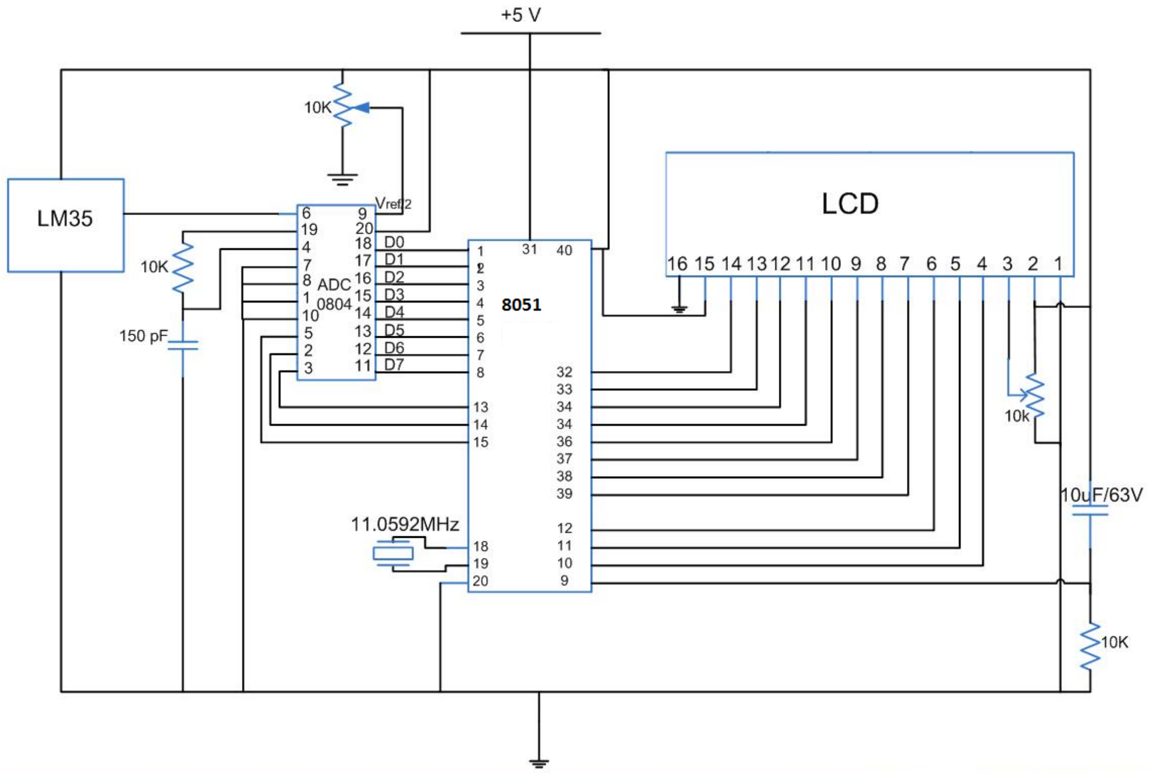
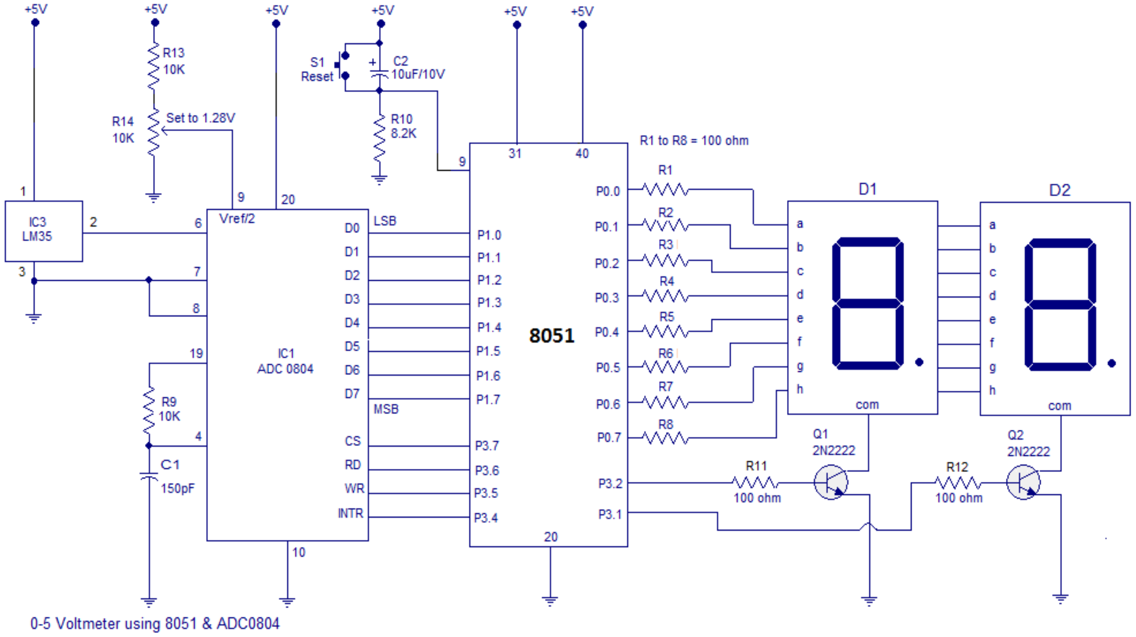
The **PUSH** instruction increments the stack pointer and stores the value of the specified byte operand at the internal RAM address indirectly referenced by the stack pointer.



POP Operation

Popping the contents of the stack back into a given register is the opposite process of pushing. With every pop, the top byte of the stack is copied to the register specified by the instruction and the stack pointer is decremented once.



Q. 4	A)	Attempt any three:	12M
	a)	**Draw interfacing diagram for temperature measurement using LM35 with 8051 Microcontroller.**	**4 M**
	Ans:	 **OR** 	**Labeled Diagram : 4 M**

b) **Write C language program to toggle a bit of P 1.5 continuously with 250 msec. delay. Use timer 0, mode 2. Assume crystal freqn. 11.0592 MHz.** **4 M**

Ans:

Calculation:2m, Program :2m

Solution:

CALCULATIONS
Delay of 250ms=1000 X 250us
For 250us, calculation of count
Crystal frequency= 11.0592 MHz
I/P clock = (11.059 X 10^6)/12= 1000000 = 921.58KHz
T-XTAL= 1/f-XTAL
=1/921.58 x 10^(3)
=**1.085us**
256-count = 250us/1.085us=230.42
The count which is to be loaded in timer register=256-**230=26 decimal=1AH.**
Load TL and TH with 1AH.

PROGRAM:

```
#INCLUDE <REG51.H>
SBIT TOGGLE = P1^5;
UNSIGNED INT X;
TMOD=0X02;
WHILE(1)
{
TL0=0XIA;
TH0=0X1A;
TOGGLE=~TOGGLE
TR0=1;
FOR (X=0;X<1000;X++)
{
WHILE(TF0==0);
TF0=0;
}
TR0=0;
}
```

c) **Compare RISC and CISC machines with examples.** **4 M**

Ans:

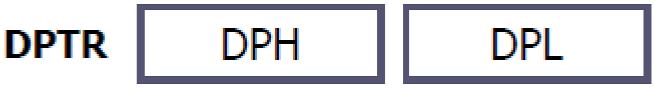
Example :1M,Any three points:3 M

RISC	CISC
Emphasis on software. Due to few instructions hardware implementation is less	Emphasis on hardware.
Single-clock, reduced instruction only	Includes multi-clock complex instructions.
Register to register: “LOAD” and “STORE” are independent instructions.	Memory to memory: “LOAD” and “STORE” incorporated in instructions.
Low cycles per second, large code sizes.	Small code sizes, high cycles per second.
More Instructions to write the code for the same task	Code can be written effectively in few lines Instruction code becomes short, saving the program memory space
Fewer, Faster and Simpler Instruction Set	Larger instruction set Hence flexibility in writing programmers.
Spends more transistors on memory registers.	Transistors used for storing complex instructions.
e.g. PIC microcontrollers by Microchip	e.g. MCS-51 (Note the MULTIPLY instruction)

d) Describe operations of DPTR register in 8051, along with related instructions. **4 M**

Ans:

- The SFRs. DPL and DPH work together to represent a 16- bit value called Data Pointer.



- DPTR can also be accessed as two 8-bit registers, the high byte DPH and low byte DPL.
- The Data Pointer, DPTR, is a special 16-bit register used to address the external code or external data memory.
- Since the SFR registers are just 8-bits wide the DPTR is stored in two SFR registers, where DPL (82h) holds the low byte of the DPTR and DPH (83h) holds the high byte of the DPTR.
- For example, if you wanted to write the value 46h to external data memory location 2500h, you might use the following instructions:

```
MOV A, #46h          ;Move immediate 8 bit data 46h to A (accumulator)
MOV DPTR, #2500h ; Move immediate 16 bit address value 2504h to A.
                 ; Now DPL holds 04h and DPH holds25h.
MOVX @DPTR, A      ; Move the value in A to external RAM location
                   2500h.Uses indirect addressing.
```

- Note the MOVX (Move X) instruction is used to access external memory

- **MOV DPTR, # 16 bit data ; 16 bit data is transferred to DPTR register.**
  (DPTR) ⟵ (DPTR) + 1
- **MOVC A,@A+DPTR**
  Operation: MOVC
  (A) ⟵ ((A) + (DPTR))
- **MOVX A,@DPTR**
  Operation: MOVX
  (A) ⟵ ((DPTR))

Operations:2m, related instructions:2m

B) Attempt any <u>one</u>: **6M**

a) Describe any four assembler directives used in 8051 programing. **6M**

Ans: **i)DB ii) ORG iii) EQU iv) END**

i) <u>DB:</u>

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.

ii) **<u>EQU: Equate</u>**
It is used to define constant without occupying a memory location.
Syntax:

Name EQU Constant

Each directive : 1$^{1/2}$ Marks

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).

iii) **ORG: Origin**
It is used to indicate the beginning of address.
Syntax:
ORG Address

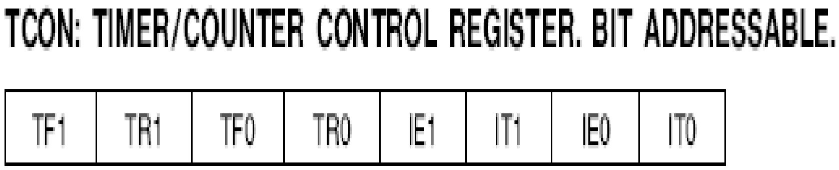
The address can be given in either hex or decimal there should be a space of at least one character between ORG & address fields. Some assemblers use ORG should not begin in label field.

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

b) Draw the format of TCON register in 8051 and describe in short. **6M**

Ans:

Format 2 M
Description: 4 M



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 set baud rate.

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

Q.5		**Attempt any two:**	**16 M**
	a)	**Write a assembly language program for 8051 microcontroller to generate a square wave of 2 KHz frequency on Pin P 1.5. Assume crystal freqn. = 11.0592 MHz.**	**8 M**

Ans:

Crystal frequency= 11.0592 MHz
I/P clock = (11.059 X 10^6)/12= 1000000 = 921.58KHz
T_{in} = 1.085µ sec
For 2 kHz square wave
F_{out} = 2 KHz
T_{out} = 1/ 2X 10^3
T_{out} = 0.5msec =500us
Consider half of it = T_{out} = 250µ sec
N = T_{out} / T_{in} = 250/1.085 = 230.41
65536-231= $(65305)_{10}$ = $(FF19)_{16}$

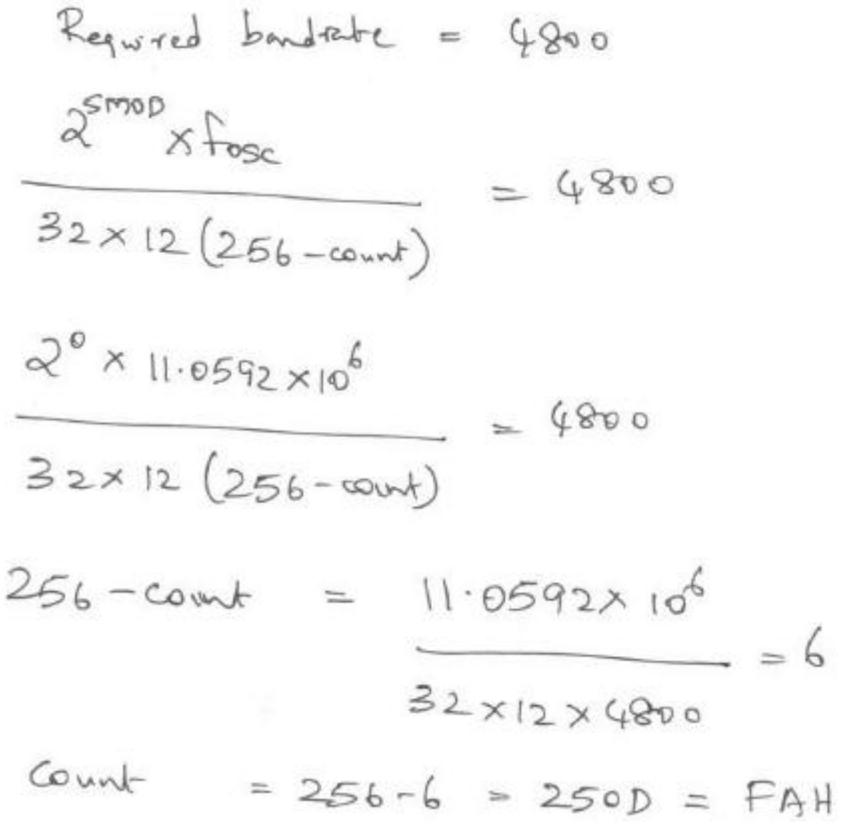
Calculation:3M, Program: 4M, comment :1M

Program:-

```
MOV TMOD, # 01H           ; SET TIMER 0 IN MODE 1, I.E., 16 BIT
                            TIMER
L2:    MOV TL0, # 19H     ; LOAD TL REGISTER WITH LSB
                            OF COUNT
MOV  TH0, # 0FFH          ; LOAD TH REGISTER WITH MSB
                            OF COUNT
SETB TR0                  ; START TIMER 0
L1:    JNB TFO, L1        ; POLL TILL TIMER ROLL OVER
CLR TR0                   ; STOP TIMER 0
CPL P1.5                  ; COMPLEMENT PORT 1.4 LINE TO GET
                            HIGH OR LOW
CLR TF0                   ; CLEAR TIMER FLAG 0
SJMP L2                   ; RE-LOAD TIMER WITH COUNT AS MODE 1
                            IS NOT AUTO RELOAD
```

b) **Write C language program to receive bytes of data serially and put them in port P1. Set baud rate of 4800, 8 bit data and 1 stop bit. Assume crystal freqn. = 11.0592 MHz.** **8 M**

Ans:



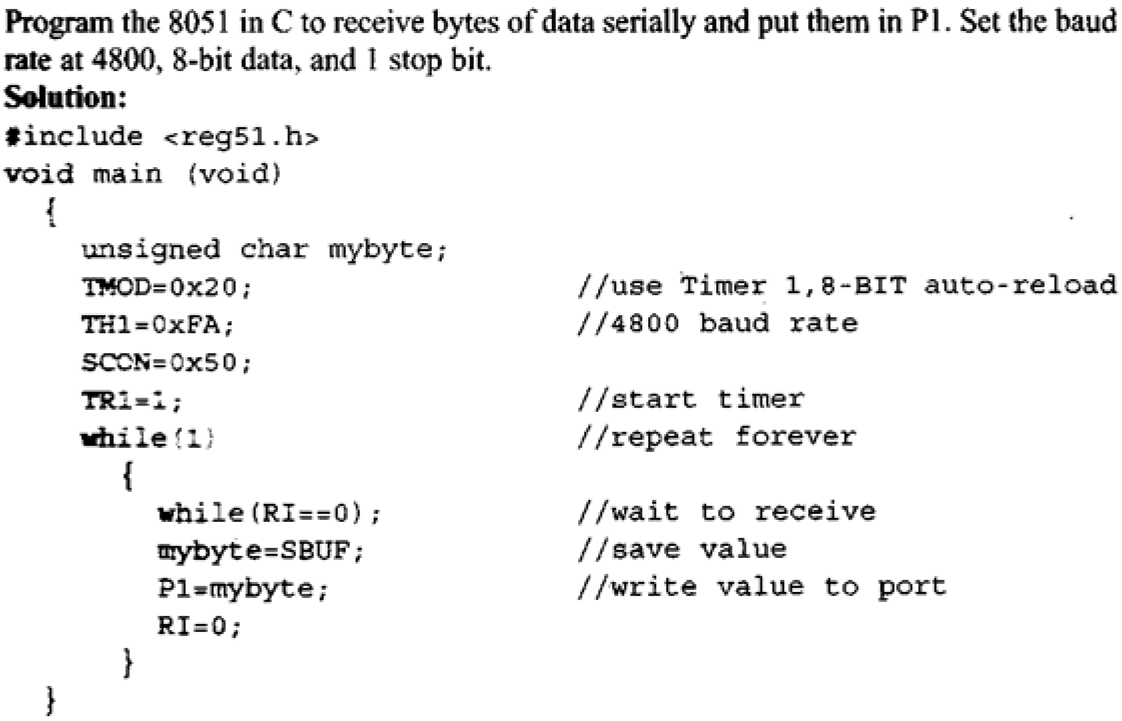
Required baudrate = 4800

$$\frac{2^{SMOD} \times f_{osc}}{32 \times 12 (256 - count)} = 4800$$

$$\frac{2^{0} \times 11.0592 \times 10^{6}}{32 \times 12 (256 - count)} = 4800$$

$$256 - count = \frac{11.0592 \times 10^{6}}{32 \times 12 \times 4800} = 6$$

$$Count = 256 - 6 = 250D = FAH$$



Program the 8051 in C to receive bytes of data serially and put them in P1. Set the baud rate at 4800, 8-bit data, and 1 stop bit.

Solution:

```
#include <reg51.h>
void main (void)
  {
    unsigned char mybyte;
    TMOD=0x20;                  //use Timer 1,8-BIT auto-reload
    TH1=0xFA;                   //4800 baud rate
    SCON=0x50;
    TR1=1;                      //start timer
    while(1)                    //repeat forever
      {
        while(RI==0);           //wait to receive
        mybyte=SBUF;            //save value
        P1=mybyte;              //write value to port
        RI=0;
      }
  }
```

Correct Program : 5M
Calculation: 2 M
comment :1M

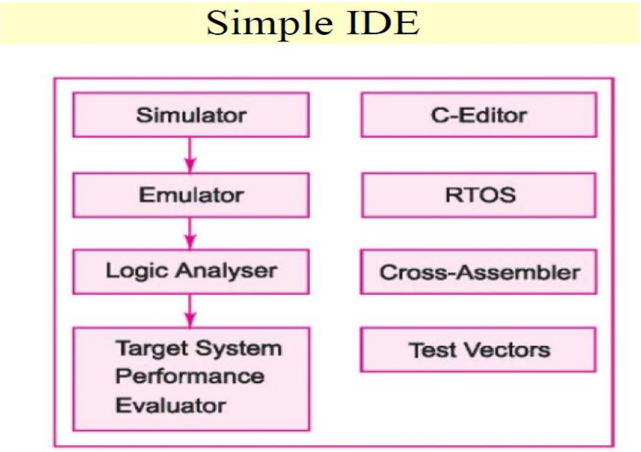
c) **What is integrated development environment for microcontroller based systems? Describe at least four features of Keil µ -vision.** **8 M**

Ans: IDE (Integrated Development Environment) is used to develop the fully simulated, tested and debugged sophisticated embedded systems with simpler efforts.IDE consists of

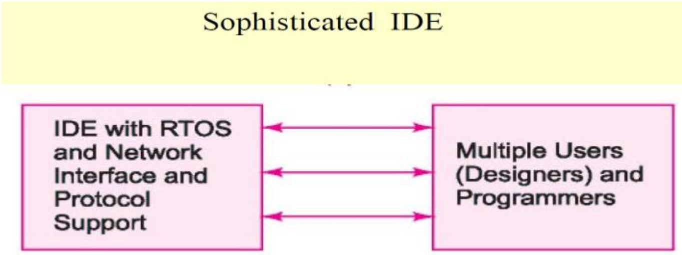
IDE-4M
Keil-4M

Simulators, Editors, compilers, Assemblers etc. Some of the functions of IDE are:

- Provides Windows on the screen for the detailed information of source code part with labels, registers as the execution continues, status of peripheral devices, status of RAM and ports etc.
- Verifies the performance of a target system that an emulator built into the development system, which remains independent of a particular targeted system
- Includes a logic analyzer for up to 256 or 512 transactions on the address and data buses after triggering
- Simulates on a host system (PC), the hardware unit like emulator, peripherals, and I/O devices.
- Debug by single stepping.



IDE for Various types and Versions of Microcontroller with Upgradability of IDE for future Versions.


Sophisticated IDE
IDE with RTOS and Network Interface and Protocol Support
Multiple Users (Designers) and Programmers

Features of Keil:

- The µVision IDE combines project management, run-time environment, build facilities, source code editing, and program debugging in a single powerful environment.
- µVision is easy-to-use and accelerates your embedded software development.
- µVision supports multiple screens and allows you to create individual window layouts anywhere on the visual surface.
- The µVision Debugger provides a single environment in which you may test, verify, and optimize your application code.
- The debugger includes traditional features like simple and complex breakpoints, watch windows, and execution control and provides full visibility to device

peripherals.

Q.6		**Attempt any FOUR of following:**	**16 M**
	a)	**Draw format of IE register in 8051 microcontroller and explain each bit.**	**4 M**
	Ans:	(MSB) (LSB)	**Format: 2M Explanation:2m**

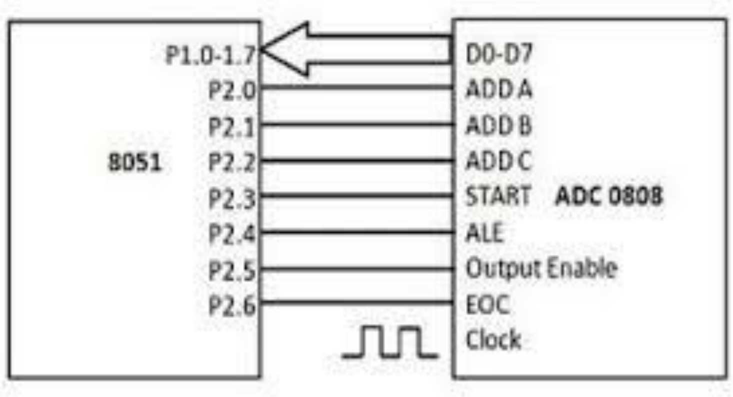
EA	-	-	ES	ET1	EX1	ET0	EX0

Symbol	Position	Name and Significance
EA	IE.7	EA = 1, interrupts are enabled and will be responded to if their corresponding bits in IE are high. If EA = 0, no interrupt will be responded to, even if the associated bit in the IE register is high.
-	IE.6	(Reserved)
-	IE.5	(Reserved)
ES	IE.4	Enable Serial port control bit Set/cleared by software to enable/disable interrupts from Tl or Rl flags.
ET1	IE.3	Enable Timer 1 control bit. Set/cleared by software to enable/disable interrupts from timer/counter 1
EX1	IE.2	Enable External interrupt 1 control bit. Set/cleared by software to enable/ disable interrupts from INT1.
ET0	IE.1	Enable Timer 0 control bit Set/cleared by software to enable/disable interrupts from timer/counter 0.
EX0	IE 0	Enable external interrupt 0 control bit. Set/cleared by software to enable/disable interrupts from INTO.

	b)	**Draw and explain the interfacing of ADC with 8051 microcontroller.**	**4 M**

Ans:

Diagram :2M
Explanation:2M



ADC808 Chip with 8 analog channels. This means this kind of chip allows to monitor 8 different transducers.

- ALE: Latch in the address
- Start : Start of conversion
- OE: output enable
- EOC: End of Conversion

Select an analog channel by providing bits to A, B, and C addresses.
2. Activate the ALE pin. It needs an Low-to-High pulse to latch in the address.
3. Activate SC (start conversion) by an High-to-Low pulse to initiate conversion.
4. Monitor EOC (end of conversion) to see whether conversion is finished.
5. Activate OE (output enable) to read data out of the ADC chip. A High-to-Low pulse to the OE pin will bring digital data out of the chip.

c) **Write an ALP for 16 bit multiplication. Assume numbers to be stored in internal RAM.** **4 M**

Ans: **(Assume first 16-bit number is stored in 40H and 41H and second 16-bit number is stored in 42H and 43H.;Result is in 20H,21H,22H & 23H)** **4M**

```
MOV R1,41H
MOV R2,43H
MOV R3,40H
MOV R4,42H

MOV A,R3
MOV B,R4
MUL AB
MOV 20H,A
MOV 21H,B

MOV A,R3
MOV B R2
MUL AB
MOV 22H,B
ADDC A,21H
MOV 21H,A

MOV A,R1
MOV B,R4
MUL AB
```

```
ADDC A,21H
MOV 21H,A
MOV A,B
ADDC A,22H
MOV 22H,A

MOV A,R1
MOV B,R2
MUL AB
ADDC A,22H
MO 22H,A
MOV A,B
ADDC A,#00H
MOV 23H,A
END
```

(Program with any other relevant logic can be given full marks)

d) **State and explain the interrupts used in 8051 microcontroller.** **4 M**

Ans: The 8051 provides five interrupt sources. These are listed below as per their default priority are: **4M**

1. Timer 0 (TF0)
2. External hardware interrupt, INT0
3. Timer 1 (TF1)
4. External hardware interrupt, INT1
5. Serial communication interrupt TI and RI

TIMER FLAG INTERRUPT:

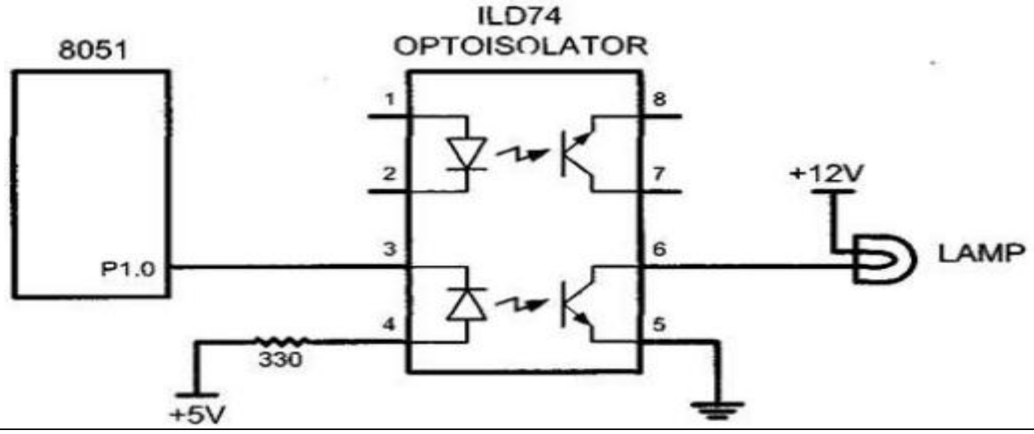
When a timer/counter overflows, the corresponding timer flag, TF0 or TF1, is set to 1. This can cause Timer 0 or Timer 1 interrupts. The flag is cleared to 0 when the resulting interrupt generates a program call to the appropriate timer subroutine in memory.

EXTERNAL INTERRUPTS:

Pins INT0 and INT1are used by external circuitry. Inputs on these pins can set the Interrupt flags IE0 and IE1 in TCON register to 1 by two different method. The IEX flags may be set when the INTX pin signal reaches a low level, or the flags may be set when a high-to-low transition takes place on the INTX pin. Bits IT0 and IT1 in TCON program the INTX pin for low level interrupt when set to 0 and program the INTX pins for transition interrupt when set to 1.

SERIAL PORT INTERRUPT:

If a data byte is received serially an interrupt bit ,RI, is set to 1 in the SCON register. When a data byte has been transmitted serially an interrupt bit , TI, is set in SCON.

	e)	**Draw neat labeled interfacing diagram to control a lamp at pin P1.0, by using Optoisolator with 8051 microcontroller.**	**4 M**
	Ans:	 ILD74 OPTOISOLATOR 8051 P1.0 1 2 3 4 8 7 6 5 +12V LAMP 330 +5V	**4M**