SUMMER– 18 EXAMINATION

Subject Name: Microprocessor & Applications **Model Answer** Subject Code: 17509

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.	Answers	Marking Scheme
1	**a**	**Attempt any THREE:**	**12- Total Marks**
	i	**Draw the format of TCON and state the function of each bit in it.**	**4M**
	Ans:	TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE. | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | Bit Symbol TCON Bit Function 7 TF1 Timer 1 Overflow flag. Set when timer rolls from all 1's to 0. Cleared when processor vectors to execute interrupt service routine located at program address 001Bh. 6 TR1 Timer 1 run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. 5 TF0 Timer 0 Overflow flag. Set when timer rolls from all 1's to 0. Cleared when processor vectors to execute interrupt service routine located at	Format-2 marks Function-2marks

		program address 000Bh. 4 TR0 Timer 0 run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. 3 IE1 External interrupt 1 Edge flag. Set to 1 when a high-to-low edge signal is received on port 3.3 (INT1). Cleared when processor vectors to interrupt service routine at program address 0013h. Not related to timer operations. 2 IT1 External interrupt 1 signal type control bit. Set to 1 by program to enable external interrupt 1 to be triggered by a falling edge signal. Set to 0 by program to enable a low-level signal on external interrupt 1 to generate an interrupt. 1 IE0 External interrupt 0 Edge flag. Set to 1 when a high-to-low edge signal is received on port 3.2 (INT0). Cleared when processor vectors to interrupt service routine at program address 0003h. Not related to timer operations. 0 IT0 External interrupt 0 signal type control bit. Set to 1 by program to enable external interrupt 1 to be triggered by a falling edge signal. Set to 0 by program to enable a low-level signal on external interrupt 0 to generate an interrupt.	
	ii	**Compare between RISC and CISC machines (any four points).**	**4M**
	Ans:		Any 4 points – 1mark each

RISC	CISC
Reduced instruction set computer	Complex instruction set computer.
Instruction set is simple and limited.	Instruction set is very large.
Instruction set is not flexible, hence the program is long.	Instruction set is flexible, hence the program is short
Provides few addressing modes normally register.	Provides many addressing modes
Large memory is required	Less memory required
Provides large number of registers	No. of registers are less.
Process architecture & control unit is simple	Processor architecture and control unit is complicated.
Instruction are shorter, hence execution speed is fast.	Instructions are lengthy hence execution speed is slow.
External memory is accessed rarely	External memory is accessed frequently
Each instruction required few bus cycle.	Each instruction requires many bus cycle.
e. g. ARM, ATMEL, AVR, MIPS, PIC, POWER PC, etc.	Interx86, Motorola, 68000 series.

iii **Write a program in 'C' to read a byte of data from Port 1 if it is greater than 99 send it to Port 0 otherwise send it to Port 2.** **4M**

Ans : Program:

Correct program-4 marks

```
#include <reg51.h>
void main(void)
{
unsigned char x;
P1=0xFF;                    //make P1 input port
while (1)
{
x=P1;                        //get a byte from P1
if (x>99)                   //check if a byte is greater than 99h
{
P0=x;
}
else
```

		P2=x; //send it to P0
		}
		}	

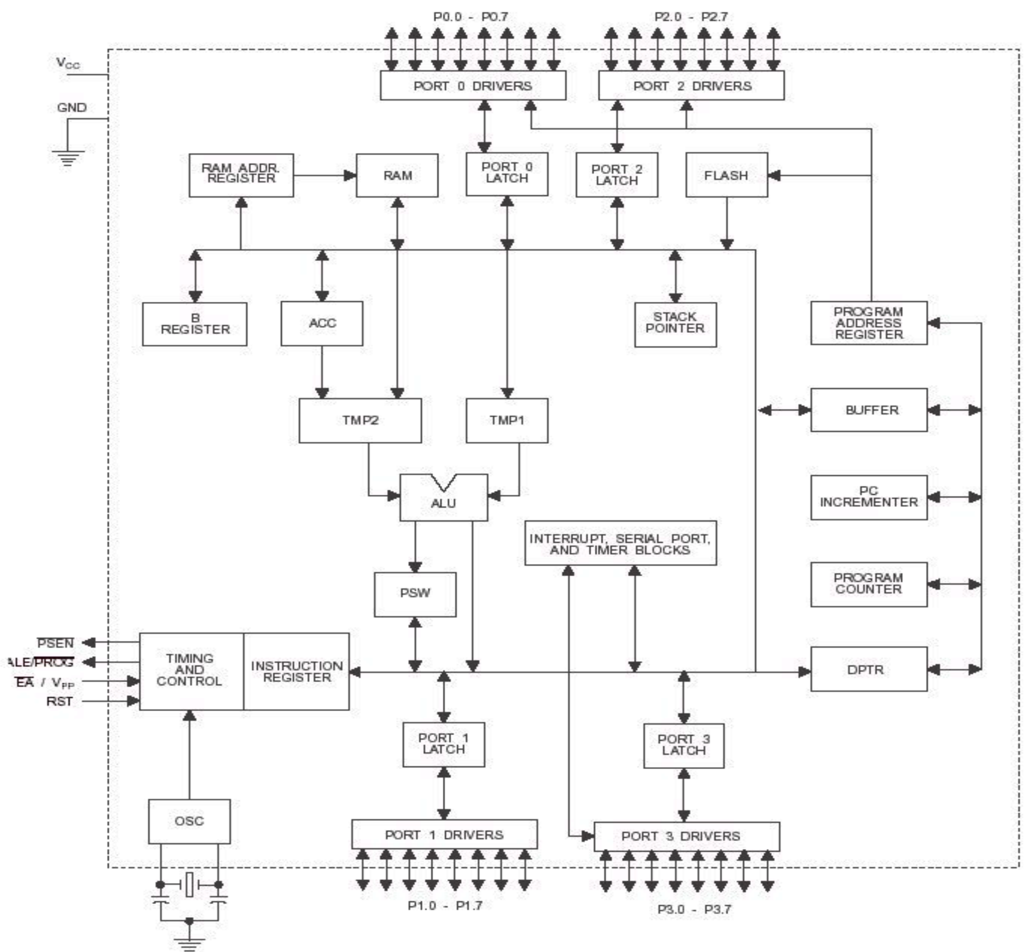
	iv	**Define Baud rate. State the Baud rate of each mode in UART.**	**4M**
	Ans:	**Baud rate :**Baud rate is defined as the number of bits transmitted per sec. OR Baud rate is defined as the number of signal changes per second. (see table below) Where f is oscillator frequency	
	b	Attempt any ONE:	**6- Total Marks**
	I	**Describe the function of following instructions in terms of length of bytes and operation.** a) **RRC A** b) **DIV AB** c) **JNB P1.3, DOWN**	**6M**
	Ans:	a) RRCA The **RRC A** instruction rotates the eight bits in the accumulator and the one bit in the carry flag right one bit position. Bit 0 of the accumulator is rotated into the carry flag while the original value of the carry flag is rotated in to bit 7 of the accumulator. Bit 7 of the accumulator is rotated into bit 6, bit 6 into bit 5, and so on. RRC $A_n = A_{n+1}$ where n = 0 to 6	1 mark for operation 1mark for no. of bytes of each instruction

Mode	Baud Rate
Mode 0	f/12
Mode 1	Variable determined by Timer 1
Mode 2	f/32 or f/64
Mode 3	Variable determined by Timer 1

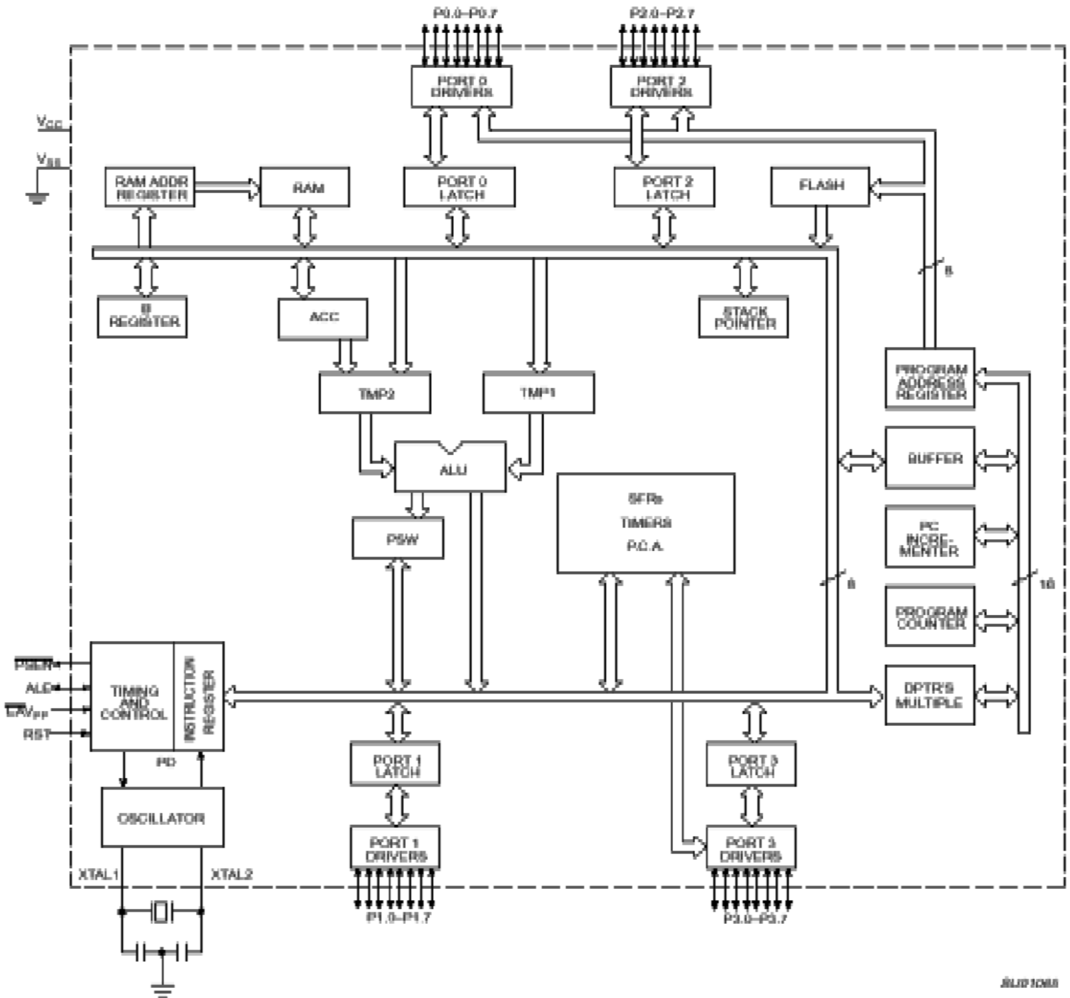
		$A_7 = C$
		
		$C = A_0$
		
		Bytes:1
		
		.
		
		b)DIV AB
		
		The **DIV** instruction divides the unsigned 8-bit integer in the accumulator by the unsigned 8-bit integer in register B. After the division, the quotient is stored in the accumulator and the remainder is stored in the B register. The carry and OV flags are cleared.
		
		`AB = A / B`
		
		Bytes:1
		
		c)JNB P1.3, DOWN
		
		The **JNB** instruction branches to the specified address (DOWN) if the bit P1.3 has a value of 0. Otherwise, execution continues with the next instruction. No flags are affected by this instruction.
		
		`JNB`
		`PC = PC + 3`
		`IF (bit) = 0`
		` PC = PC + offset`
		
		Bytes:3	

	ii	**Draw the architecture of 8051 microcontroller.**	**6M**
	Ans :		Correct labelled diagram-4 marks

Block Diagram



(OR)


P0.0–P0.7
P2.0–P2.7
PORT 0 DRIVERS
PORT 2 DRIVERS
RAM ADDR REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
TMP2
TMP1
ALU
PSW
SFRs
TIMERS
P.C.A.
PROGRAM ADDRESS REGISTER
BUFFER
PC INCRE-MENTER
PROGRAM COUNTER
DPTR'S MULTIPLE
TIMING AND CONTROL
INSTRUCTION REGISTER
OSCILLATOR
PORT 1 LATCH
PORT 1 DRIVERS
PORT 3 LATCH
PORT 3 DRIVERS
XTAL1
XTAL2
P1.0–P1.7
P3.0–P3.7
PSEN
ALE
EA/VPP
RST
PD
8
16

SUMMER– 18 EXAMINATION

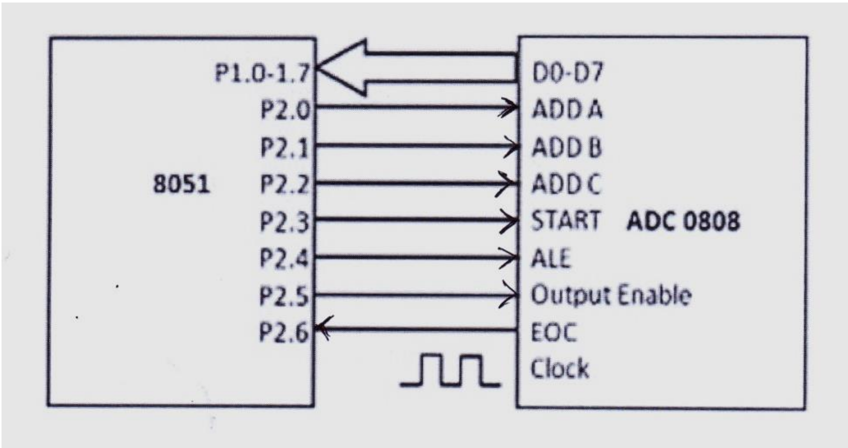
Subject Name: Microprocessor & Applications **Model Answer** Subject Code: 17509

Q. No.	Sub Q. N.	Answers	Marking Scheme
2		**Attempt any TWO:**	**16- Total Marks**
	a	**Write assembly language program to find largest number from the array of ten numbers stored in external memory RAM. Assume suitable data.**	**8M**

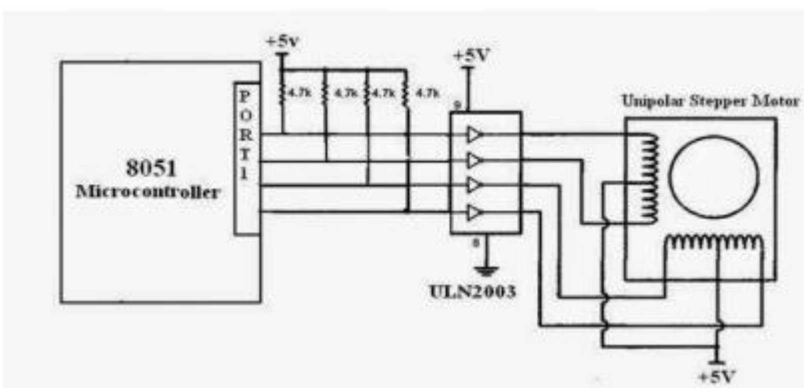
	Ans:	Assumptions: Array of ten numbers are stored from memory location 3000H onwards and result is stored in memory location 6000H
		Program:
		CLR PSW.3 ; Select Bank 0 PSW.3
		MOV R1, 0AH ; Initialize byte counter
		MOV DPTR, # 3000H ; Initialize memory pointer
		DEC R1 ; Decrement byte counter by 1
		MOV X A, @DPTR ; Load number in accumulator
		MOV 40 H, A ; Store number in memory location
		UP: INC DPTR ; Increment memory pointer by 1
		MOVXA, @DTPR ; Read next number
		CJNE A, 40 H, DN ; if number≠ next number, and then go to NEXT
		DN: JNC NEXT ; If next number < number then go to NEXT
		MOV 40H, A ; Else replace NEXT number with number
		NEXT: DJNZ R1, UP ; Decrement byte counter by 1, if byte counter≠ 0 then go to UP
		MOV DPTR,#6000H ; Increment memory pointer by 1
		MOV A, 40H
		MOVX@ DPTR, A ; Store result t external memory location
		LOOP: AJMP LOOP ; Stop
			Correct program-8 marks

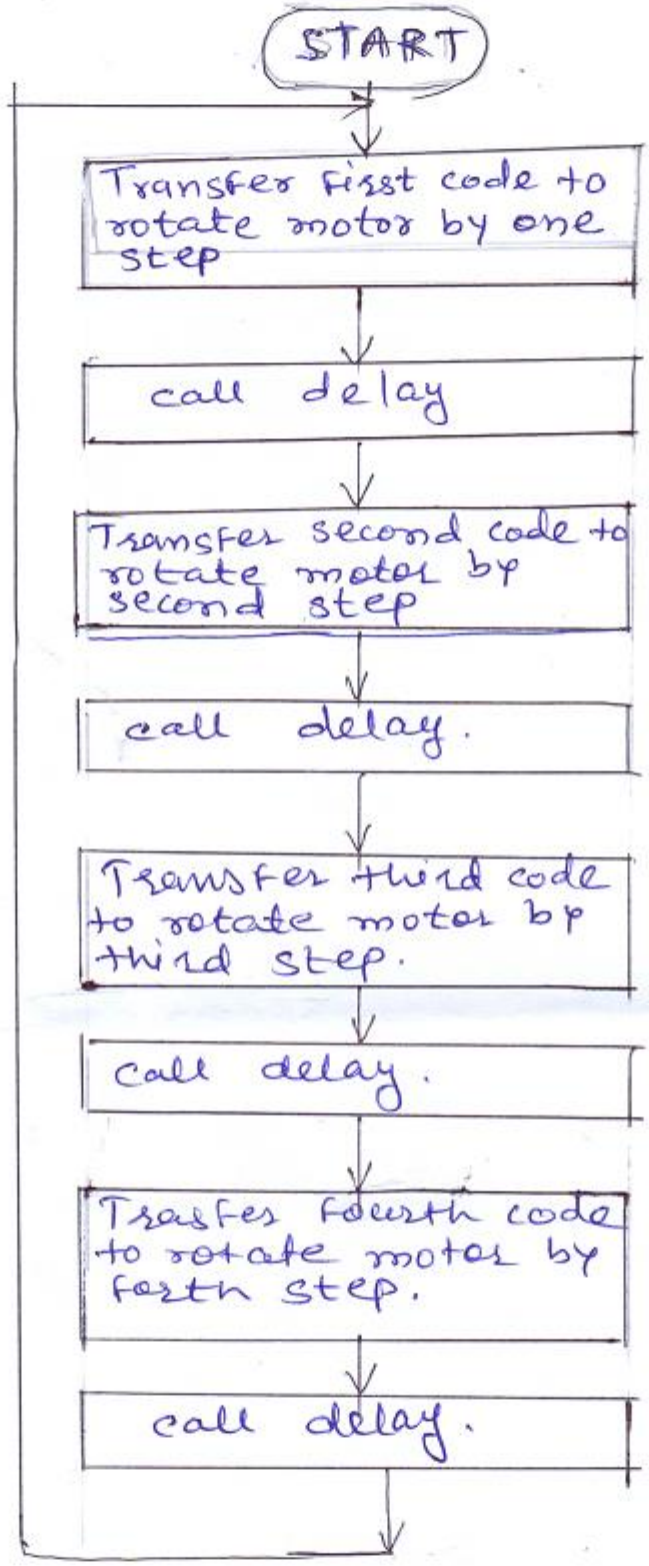
	b	**Draw interface diagram of ADC 0809 with 8051. Write 'C' language program to generate**	**8M**

		50 Hz sq. wave with crystal freq = 12 MHz.	

	Ans:	
		
		'C' language program to generate 50 Hz sq. wave with crystal freq = 12 MHz.:
		Square wave Frequency = 50Hz
		Therefore Time period T = 1 / 50HZ = 0.02s
		Therefore TON = TOFF = 0.02s / 2 = 0.01s
		(65535-count+1)1MHZ=0.01s
		65536-count=0.01s/1µs
		Count=65536-10000
		Count=55536
		Count=D8F0H
		Program:
		#include<reg51.h>
		void delay(void);
		sbit sq=P1^5;
		void main (void)
		{
		while (1)
		{
		sq=~sq;
		delay();
			Diagram-4 marks
			Program-4 marks

```
}
}
void delay()
 {
 TMOD=0X01;        //set timer 0 in mode 1 i.e. 16 bit number
TL0=0X0F0;          //load TL register with LSB of count
TH0=0X0D8 ;        //Load TH register with MSB of count
TR0 =1;              //Start timer 0
While(TF0==0);     //wait until timer rolls over
TR0=0;               //Stop timer 0
TF0=0;               //Clear timer flag 0 }
}
```

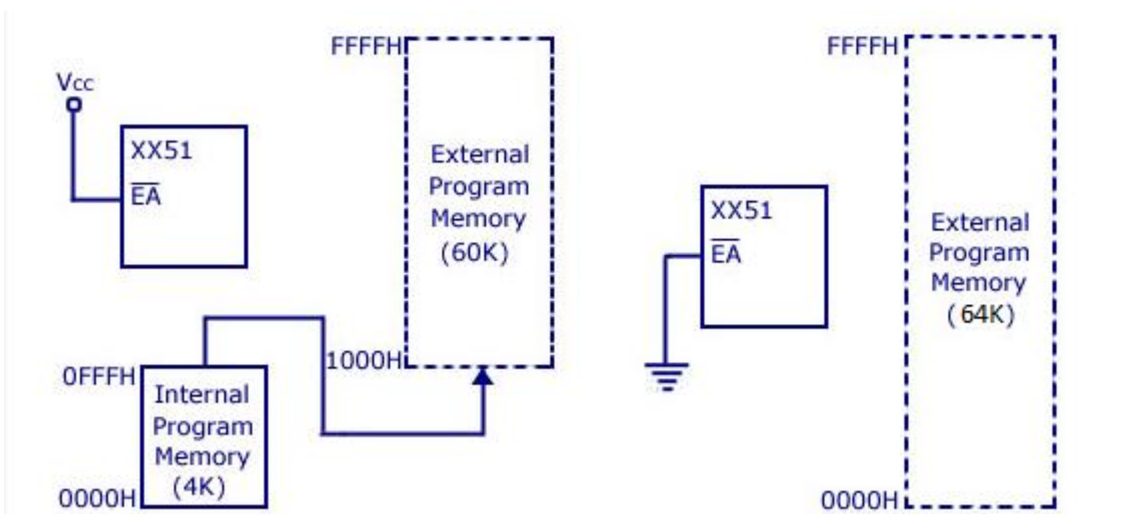
c	**Draw interfacing diagram for stepper motor with 8051. Draw flow chart for rotating stepper motor in clockwise direction.**	**8M**
Ans:		Diagram 4 marks Flowchart- 4 marks


START
Transfer first code to rotate motor by one step
call delay
Transfer second code to rotate motor by second step
call delay.
Transfer third code to rotate motor by third step.
call delay.
Trasfer fourth code to rotate motor by forth step.
call delay.

SUMMER– 18 EXAMINATION

Subject Name: Microprocessor & Applications **Model Answer** Subject Code: 17509

Q. No.	Sub Q. N.	Answers	Marking Scheme
3		**Attempt any FOUR:**	**16- Total Marks**
	a	**i) Convert $(1011101)_2$ to $(\quad)_{10}$.** **ii) Subtract $(1001)_2$ from $(1100)_2$ by using 2's complement method.**	**4M**
	Ans:	Q3a). (i) Convert $(1011101)_2$ to $()_{10}$ $(1011101)_2$ $= (1\times2^6)+(0\times2^5)+(1\times2^4)+(1\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$ $= 64+0+16+8+4+0+1$ $= (93)_{10}$	2M EACH

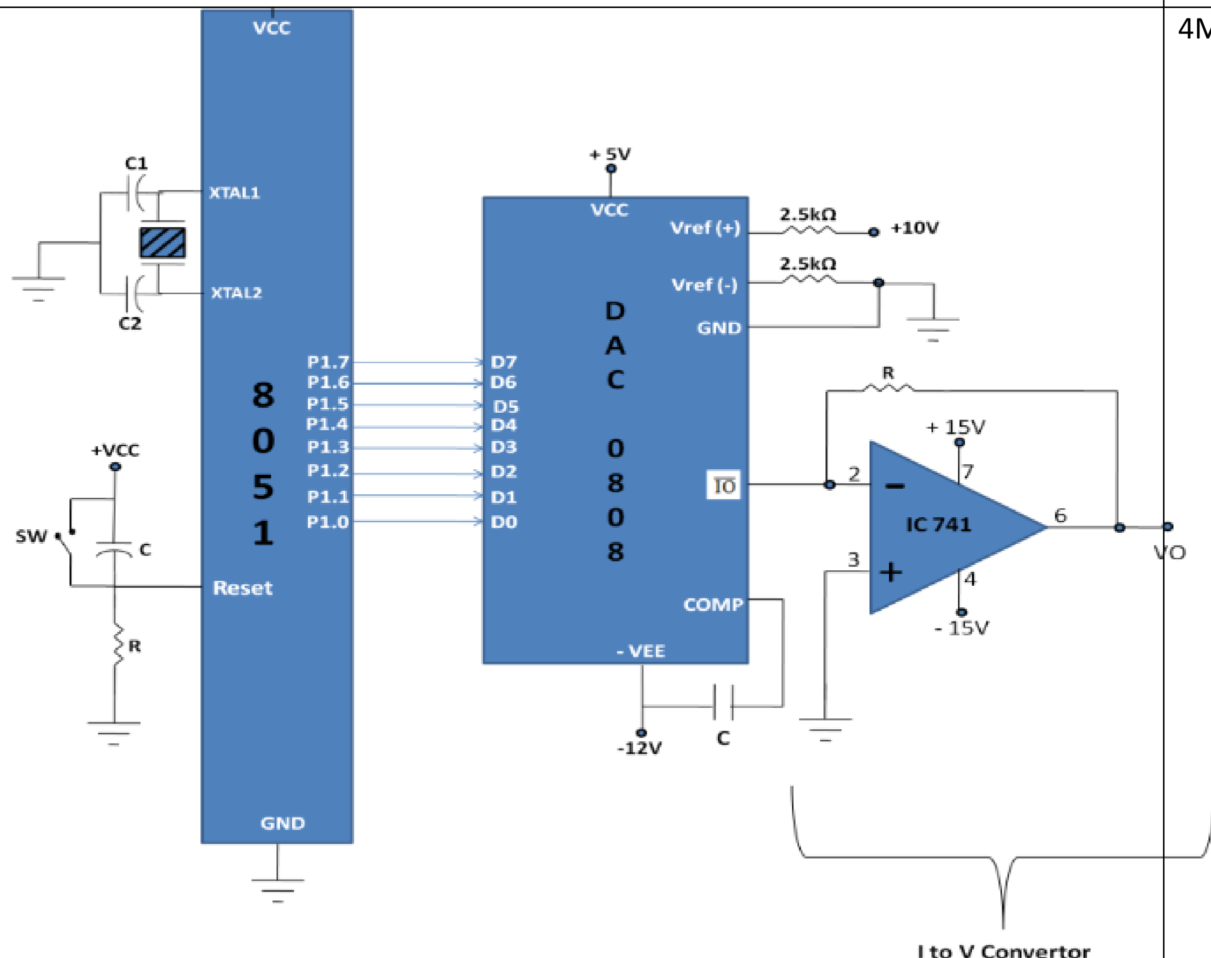
		(ii) $(1100)_2 - (1001)_2$ 1's complement of $(1001) = 0110$ 2's comp $\Rightarrow$ $\begin{array}{r} 0110 \\ +\quad 1 \\ \hline 0111 \end{array}$ Now add, $1100 + 0111$ $\begin{array}{r} 1100 \\ +\ 0111 \\ \hline \boxed{1}\ 0011 \end{array}$ Neglect the carry, So the result is $(0011)_2 \rightarrow (3)_{10}$	
	b	**Draw program memory organization for i) $E\overline{A}$= 0 ii) E $\overline{A}$= 1.**	**4M**
	Ans :	Program memory organization	2M each

c	**Compare between 8051 and 8052 microcontroller.**	**4M**
Ans:	(table below)	1M each

Feature	8051	8052
ROM(bytes)	4K	8K
RAM(bytes)	128	256
Timers	2	3
I/O pins	32	32
Serial Port	1	1
Interrupts	6	8
Watchdog timer	No	No

d	**Describe the standard data types in 'C' for 8051 with suitable example.**	**4M**
Ans:	**(4M for correct answer)**	Any four 1 M each

DATA TYPES	SIZE IN BITS	DATA RANGE	EXAMPLE
Unsigned char	8-bit	0 to 255	Unsigned char x;
(signed) char	8-bit	-128 to +127	Signed char x;
Unsigned int	16-bit	0 to 65535	Unsigned int x;
(signed) int	16-bit	-32768 to +32767	Signed int x;
Sbit	1-bit	SFR bit addressable only	Sbit LED=P1^2;
Bit	1-bit	RAM bit addressable only	Bit x;
sfr	8-bit	RAM addresses 80-FFH only	Sfr P1= 0x90;

	e	**Draw interlacing diagram of DAC 0808 with 8051 microcontroller.**	**4M**
	Ans :		4M
Q. No.	Sub Q. N.	Answers	Marking Scheme
4	**a**	**Attempt any THREE:**	**12- Total Marks**
	i	**State the alternate function of Port 3 pins of microcontroller 8051.**	**4M**
	Ans :		½ M each point

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

ii	**Compare EPROM and EEPROM (any four points).**	**4M**
Ans :		1M each

BASIS FOR COMPARISON	EPROM	EEPROM
Basic	Ultraviolet Light is used to erase the content of EPROM.	EEPROM contents are erased using electronic signal.
Appearance	EPROM has a transparent quartz crystal window at the top.	EEPROM are totally encased in an opaque plastic case.
Erased and Reprogrammed	EPROM chip has to be removed from the computer circuit to erase and reprogram the computer BIOS.	EEPROM chip can be erased and reprogrammed in the computer circuit to erase and reprogram the content of computer BIOS.
Technology	EPROM is an older technology.	EEPROM is a modern version over EPROM.

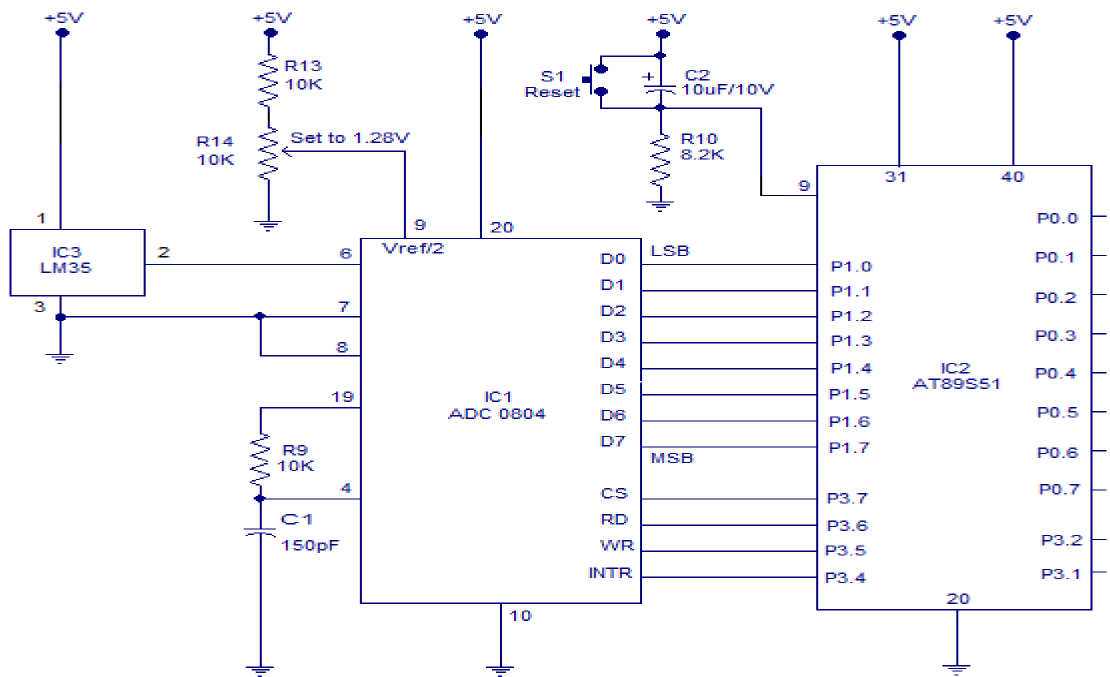
iii Write 'C' program to toggle only bit P1.2 continuously with 200 ms delay. **4M**

Ans:

(4M for correct program)

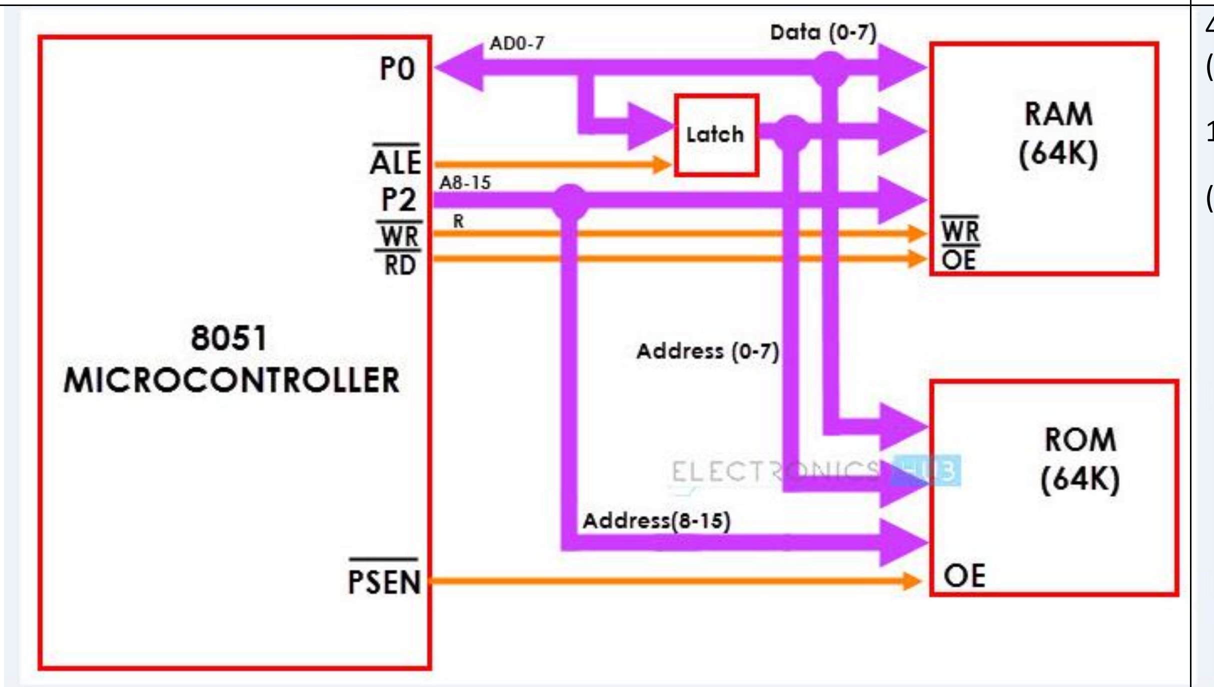
```
#include <reg51.h>
sbit LED= P1^2;
void delay(unsigned int);
void main()
{
while(1)
{
LED=0;
delay(200);
```

		LED=1;
		delay (200);
		}
		}
		void delay(unsigned int t)
		{
		unsigned int i,j;
		for(i=0;i<=t;i++)
		for(j=0;j<=1275;j++);
		}	

	iv	**Draw interfacing of LM 35 temperature sensor with 8051 microcontroller.**	**4M**
	Ans :	 Note:(Diagrams drawn using any other ADC (0808,0848 etc.) also to be considered)	4M

b	**Attempt any ONE:**		**6- Total Marks**
i	**Write assembly language program to count time period of square wave using 8051 counter. Assume suitable data.**		**6M**
Ans :	Assumptions : This program will display the time period in ms Oscillator frequency = 12 MHz, Timer0 as counter in Mode1, Timer1 for delay generation in Mode1, Square wave whose time period to be found is applied to P3.4 (T0 input for Timer0) , Higher byte of result is displayed in Port2 and lower byte in Port1. Explanation: If we count the number of pulses applied to P3.4 in 1 ms, we will directly get the time period in milliseconds. Timer0 is used to count the pulses and Timer1 is used to generate a delay of 1ms. The value to be given to TMOD register is 15H The count value to generate time delay of 1ms is FC18H Program:		Correct program : 6M

```
        MOV TMOD,#15H      //LOAD 15H TO TOMOD REGISTER
        SETB P3.4          //P3.4 AS INPUT
        MOV TL0,#00H
        MOV TH0,#00H       //LOAD INITIAL VALUE FOR COUNTING
        SETB TR0           //START COUNTER
        MOV TL1,#18H
        MOV TH1,#0FCH      //LOAD COUNT VALUE FOR 1ms DELAY
SAME:   JB P3.4,SAME       //WAIT FOR FIRST FALLING EDGE OF SQUARE WAVE
        SETB TR1           //START TIMER1
BACK:   JNB TF1,BACK       //CHECK FOR 1ms DELAY
        CLR TF1            //CLEAR TF1
        CLR TR1            //STOP TIMER1
        CLR TR0            //STOP COUNTER0
        MOV A,TL0
```

		MOV P1,A //LOWER BYTE OF COUNT IN P1 MOV A,TH0 //HIGHER BYTE OF COUNT IN P2 MOV P2,A SJMP $	
	ii	**Draw the diagram to interface external RAM and ROM with 8051 microcontroller and explain the function of ALE and $\overline{PSEN}$ pins of 8051.**	**6M**
	Ans :	 PSEN pin stands for Program Store Enable. It is used to read a signal from the external program memory. ALE/PROG: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during EPROM programming.	4M (drawing) 1M each (function)
Q. No.	Sub Q. N.	Answers	Marking Scheme
5		**Attempt any TWO:**	**16- Total Marks**
	a	**Describe the addressing modes of 8051 with suitable example.**	**8M**
	Ans :	There are a number of addressing modes available to the 8051 instruction set, as follows:	Any four, 2 marks

		1. Immediate Addressing mode 2. Register Addressing mode 3. Direct Addressing mode 4 Register Indirect addressing mode 5. Relative Addressing mode 6. Absolute addressing mode 7. Long Addressing mode 8. Indexed Addressing mode 1) Immediate Addressing mode: Immediate addressing simply means that the operand (which immediately follows the Instruction op. code) is the data value to be used. For example the instruction: MOV A, #25H ; Load 25H into A Moves the value 25H into the accumulator. The # symbol tells the assembler that the immediate addressing mode is to be used. 2) Register Addressing Mode: One of the eight general-registers, R0 to R7, can be specified as the instruction Operand. The assembly language documentation refers to a register generically as Rn. An example instruction using register addressing is : ADD A, R5 ; Add the contents of register R5 to contents of A (accumulator) Here the contents of R5 are added to the accumulator. One advantage of register addressing is that the instructions tend to be short, single byte instructions. 3) Direct Addressing Mode: Direct addressing means that the data value is obtained directly from the memory location specified in the operand. For example consider the instruction: MOV R0, 40H; Save contents of RAM location 40H in R0. The instruction reads the data from Internal RAM address 40H and stores this in theR0. Direct addressing can be used to access Internal RAM, including the SFR registers. 4) Register Indirect Addressing Mode: Indirect addressing provides a powerful addressing capability, which needs to be appreciated. An example instruction, which uses indirect addressing, is as follows: MOV A, @R0; move contents of RAM location whose address is held by R0 into A Note the @ symbol indicated that the indirect addressing mode is used. If the data is inside the CPU, only registers R0 & R1 are used for this purpose. 5) Relative Addressing Mode: This is a special addressing mode used with certain jump	each (1 mark for explanation and 1 mark for example)

		instructions. The relative address, often referred to as an offset, is an 8-bit signed number, which is automatically added to the PC to make the address of the next instruction. The 8-bitsigned offset value gives an address range of + 127 to –128 locations. 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) Absolute addressing Mode: Absolute addressing within the 8051 is used only by the AJMP (Absolute Jump) and ACALL (Absolute Call) instructions. 7) Long Addressing Mode: The long addressing mode within the 8051 is used with the instructions LJMP and LCALL. The address specifies a full 16 bit destination address so that a jump or a call can be made to a location within a 64KByte code memory space (216 = 64K). An example instruction is: LJMP 5000h; full 16 bit address is specified in operand 8) Indexed Addressing Mode: With indexed addressing a separate register, either the program counter, PC, or the data pointer DTPR, is used as a base address and the accumulator is used as an offset address. The effective address is formed by adding the value from the base address to the value from the offset address. Indexed addressing in the 8051 is used with the JMP or MOVC instructions. Look up tables are easy to implement with the help of index addressing. 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.	
	b	**Write 'C' language program to transfer the message "AICTE" serially at baud rate 9600. Assume crystal frequency 11.0592 MHz.**	**8M**
	Ans :	Normally Serial communication MODE 1 is used (8 bit UART with variable baud rate). For setting baud rate, TIMER 1 to be programmed in Mode 2 – Auto reload mode. Following formula is used for calculation of Count to be given to Timer 1 register, TH1 to set standard baud rate. $f_{baud} = \frac{2^{smod}}{32} X \frac{f_{osc}}{12\ X\ (256 - TH1)}$	Calculation of count – 1M Program :- 7M

Hence for 9600 baud, $9600 = \frac{2^0}{32} X \frac{11059200}{12 X (256-TH1)}$

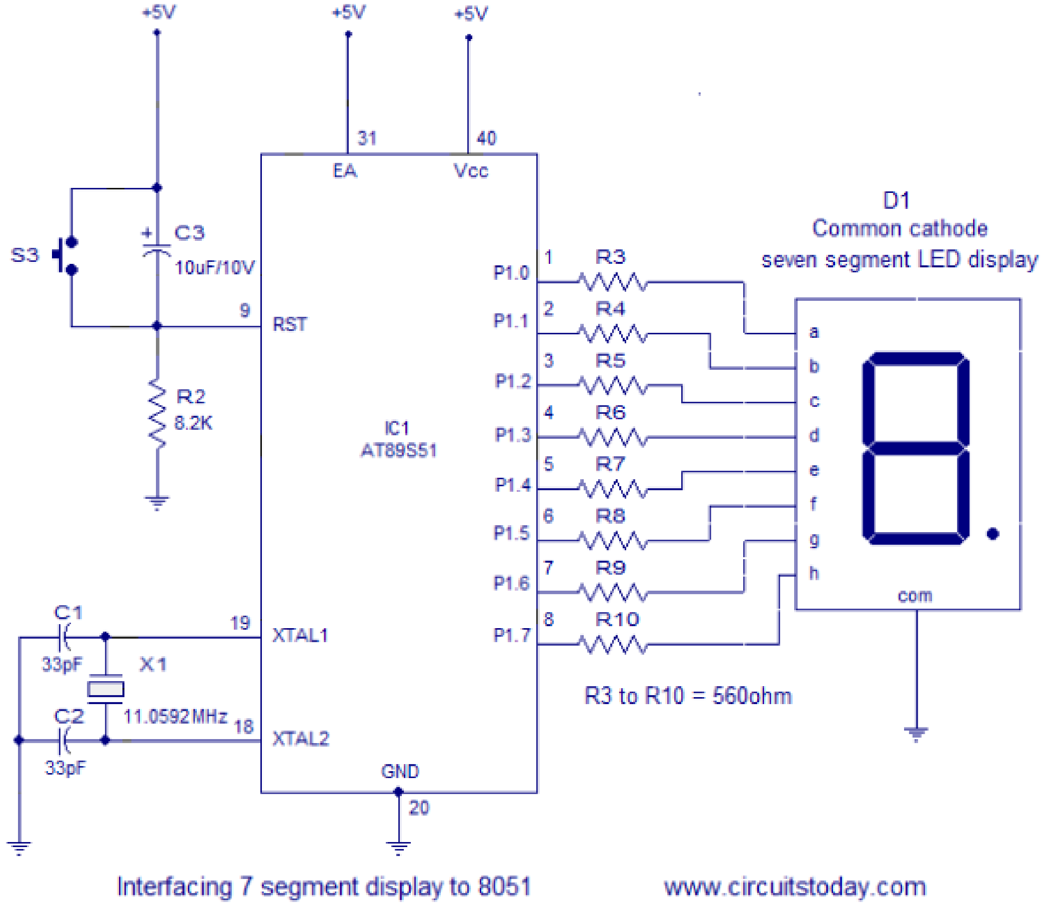
From this formula we get value to be loaded to TH1 as FD H

(Any other logically similar method may also be used for calculation)

Program:

```
#include <REG51.H>
void main(void)
{
    unsigned char text[] = "AICTE";
    unsigned char i;
    TMOD = 0x20;
    TH1 = 0xFD;
    SCON = 0x50;
    TR1 = 1;
    while(1)
    {
        for(i =0;i<5;i++)
        {
            SBUF= text[i];
            while(TI ==0);
            TI = 0;
        }
    }
}
```

	c	**Draw and explain the interfacing of seven segment LED display in common cathode with 8051 microcontroller. Write 'C' language program to display digit 0 to 9.**	**8M**
	Ans :		4 marks diagram 4 marks program



A = P1.0 B = P1.1 C = P1.2 D = P1.3

E = P1.4 F = P1.5 G = P1.6 DP= P1.7

DB 3FH // digit drive pattern for 0
DB 06H // digit drive pattern for 1
DB 5BH // digit drive pattern for 2
DB 4FH // digit drive pattern for 3
DB 66H // digit drive pattern for 4
DB 6DH // digit drive pattern for 5
DB 7DH // digit drive pattern for 6
DB 07H // digit drive pattern for 7
DB 7FH // digit drive pattern for 8
DB 6FH // digit drive pattern for 9

void main ()

```
{
 P1 = 0xFF;  //DISPLAY OFF
 while(1)
 {
P1 = 0x3F;  //DISPLAY 0
  delay_ms(1000);
  P1 = 0x06;  //DISPLAY 1
  delay_ms(1000);
  P1 = 0x5B;  //DISPLAY 2
  delay_ms(1000);
  P1 = 0x4F;  //DISPLAY 3
  delay_ms(1000);
  P1 = 0x66;  //DISPLAY 4
  delay_ms(1000);
  P1 = 0x6D;  //DISPLAY 5
  delay_ms(1000);
  P1 = 0x7D;  //DISPLAY 6
  delay_ms(1000);
  P1 = 0x07;  //DISPLAY 7
  delay_ms(1000);
  P1 = 0x7F;  //DISPLAY 8
  delay_ms(1000);
  P1 = 0x6F;  //DISPLAY 9
  delay_ms(1000);
 }
```

		} NOTE: Program may change. Student can also use the other logic. Please check the logic and understanding of students.	
Q. No.	Sub Q. N.	Answers	Marking Scheme
6		**Attempt any FOUR:**	**16- Total Marks**
	a	**Draw the format of IE register of 8051 and state the function of each bit.**	**4M**
	Ans :	**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 | — | — | 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.* — IE.5 Not implemented, reserved for future use.* 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.	2 mark format 2 marks function
	b	**Describe the following assembler directives with one example.** i) **ORG** ii) **DB** iii) **EQU** iv) **END**	**4M**
	Ans :	(Each directive explanation with example : 1 mark each) **1. ORG** : ORIGIN The ORG directive is used to indicate the beginning of the address. The	1 mark each

number that comes after ORG is the address from where program will begin. The number can be either in hex and decimal

e.g. ORG 1000H

It indicates that program shall start from memory address 1000h

2. DB : Define Byte It is used to define the 8-bit data. It is used to write the value after DB , into the program memory. When DB is used to define data, the numbers can be in decimal, binary, hex, ASCII formats

e.g. ORG 1000H

MYDATA: DB 20,21

After execution of this , location 1000h=20 & 1001h = 21

3. EQU: EQUATE It is used to define a constant without occupying a memory location The EQU directive assigns a constant value to a label. When the label appears in the program, its constant value will be substituted for the label.

e.g. COUNT EQU 10

MOV R2, #COUNT

When the instruction is executed , register R2 is loaded with value 10.

4. END: This indicates to the assembler the end of the source (asm) file . The END directive is the last line of an 8051 program. Means that in the program anything after the END directive is ignored by the assembler

e.g. MOV A , #20

ADD A , #10

END

c	**Write down the function of 16 pin connector of LCD module.**	**4M**
Ans :		4 marks

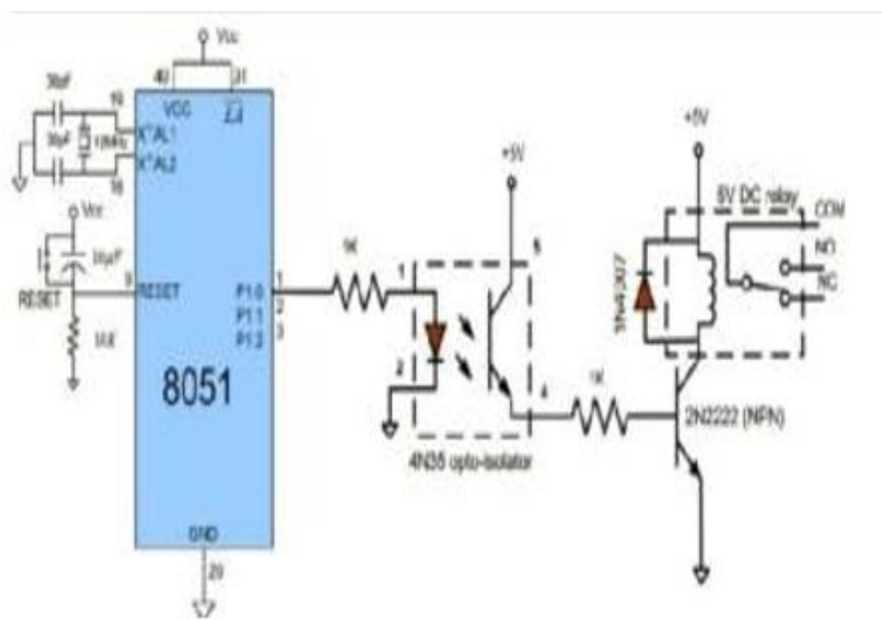
Pin	Pin Name:	Description

		No:			
		1	Vss (Ground)	Ground pin connected to system ground	
		2	Vdd (+5 Volt)	Powers the LCD with +5V (4.7V – 5.3V)	
		3	VE (Contrast V)	Decides the contrast level of display. Grounded to get maximum contrast.	
		4	Register Select	Connected to Microcontroller to shift between command/data register	
		5	Read/Write	Used to read or write data. Normally grounded to write data to LCD	
		6	Enable	Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement	
		7	Data Pin 0	Data pins 0 to 7 forms a 8-bit data line. They can be connected to Microcontroller to send 8-bit data. These LCD's can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free.	
		8	Data Pin 1		
		9	Data Pin 2		
		10	Data Pin 3		
		11	Data Pin 4		

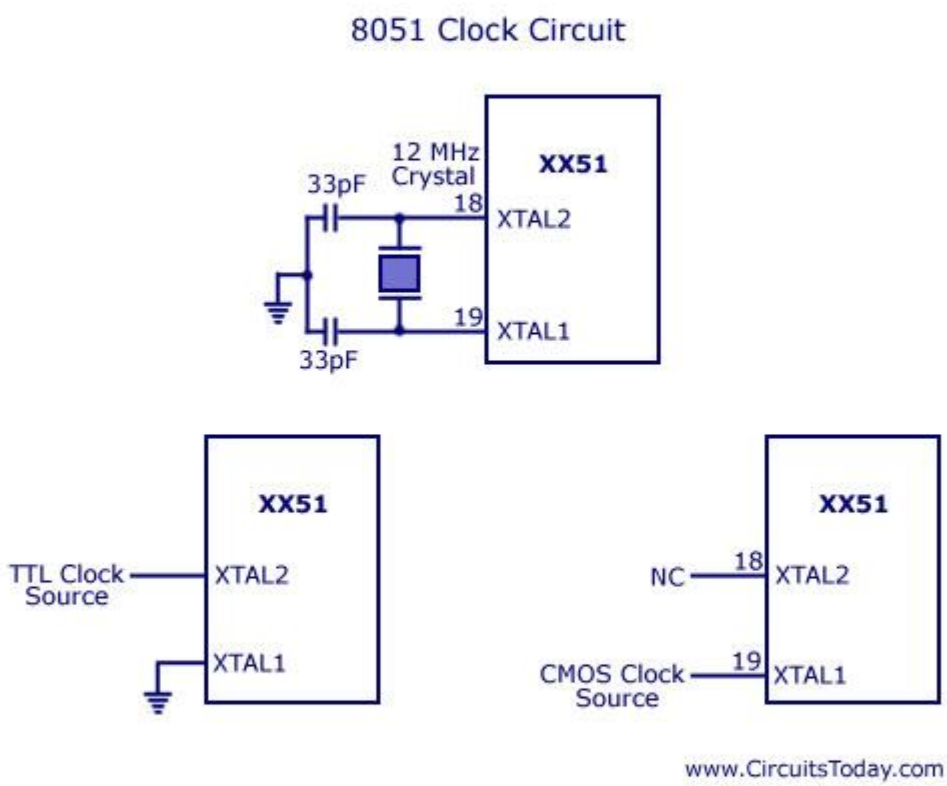
12	Data Pin 5		
13	Data Pin 6		
14	Data Pin 7		
15	LED Positive	Backlight LED pin positive terminal	
16	LED Negative	Backlight LED pin negative terminal	

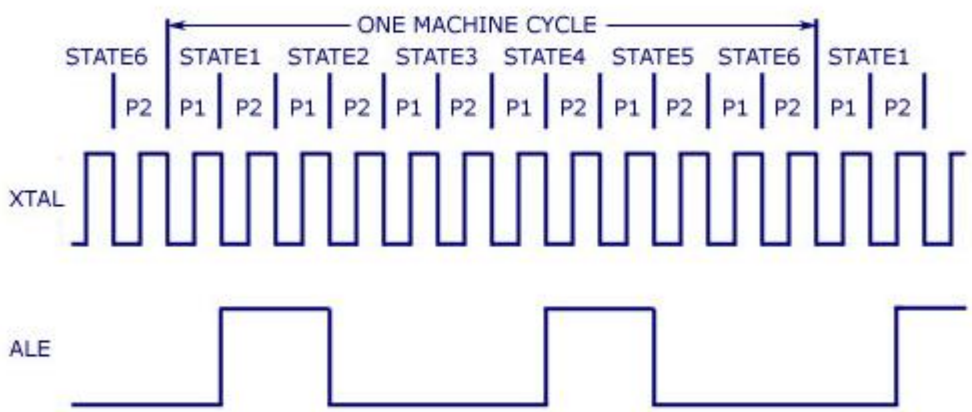
d	**Draw and describe the interfacing of opto-isolator with 8051 microcontroller.**	**4M**

Ans: 2 marks diagram 2 marks explaination



Optocoupler is also called optoisolator, photocoupler and optical isolator. It is used to provide isolation between two electrical circuits. It is a electrical component which is used to transmit input signal by using light energy signals. It provides electrical coupling between input and output through light waves. Its main purpose is to avoid changing in voltages at output side to appearing at input side. Higher voltage fluctuations may

		damage input side electrical components. The output of opto coupler is given to relay which further operates the output device	
	e	**Explain system clock and machine cycle of 8051 microcontroller.**	**4M**
	Ans :	 An 8051 clock circuit is shown above. In general cases, a quartz crystal is used to make the clock circuit. The connection is shown in figure (a) and note the connections to XTAL 1 and XTAL 2. In some cases external clock sources are used and you can see the various connections above. Clock frequency limits (maximum and minimum) may change from device to device. Standard practice is to use 12MHz frequency. If serial communications are involved then its best to use 11.0592 MHz frequency.	2 marks each



One machine cycle has 6 states. One state is 2 T-states. Therefore one machine cycle is 12 T-states. Time to execute an instruction is found by multiplying C by 12 and dividing product by Crystal frequency.

T=(C*12d)/crystal frequency