WINTER– 18 EXAMINATION

Subject Name: Microcontroller and 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 themodel answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given moreImportance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in thefigure. The figures drawn by candidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constantvalues 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**
	(a)	**Perform following subtraction using 2's compliment method** **1) $(69)_2 - (34)_2$** **2) $(18)_2 - (27)_2$**	**4M**

	Ans:	1) $(69)_{10} - (34)_{10}$ $(69)_{10} = (1000101)_2$ $(34)_{10} = (100010)_2$ $\therefore (1000101)_2 - (100010)_2$ 1's complement of $(100010) = (011101)_2$ 2's complement — 011101 + 1 = 011110 Now add, 1000101 + 011110 = 1 0100011 Neglect carry, $\therefore$ result is $(0100011)_2 = (35)_{10}$. 2). $(18)_{10} - (27)_{10}$ $(18)_{10} = (10010)_2$ $(27)_{10} = (11011)_2$ $\therefore (10010)_2 - (11011)_2$ 1's complement of $(11011)_2 = (00100)_2$ 2's complement = 00100 + 1 = 00101 Now add, 10010 + 00101 = 0 10111 $\therefore$ No carry, the answer is –ve. Now take 2's of $(10111)_2$ i.e 01000 + 1 = 01001 $\therefore$ Result is $-(01001)_2 = (-9)_{10}$.	**2M EACH**
	(b)	**Describe function of following pins of 8051 microcontroller** **i) $\overline{PSEN}$** **ii) $\overline{EA}$** **iii) RST** **iv) ALE**	**4M**

	Ans:	1. $\overline{PSEN}$ – It is active low output control signal used to activate enable signal of external ROM/ EPROM .it is activated every six oscillator periods while reading the external memory. 2. $\overline{EA}$ – It is active low output control signal. When EA = 1, µc accesses internal and external program memory when EA =0, µc accesses only external program memory. 3. **RST:** It is a RESET pin, which is used to reset the microcontroller to its initial values. 4. **ALE:** There are two ALE pulses per machine cycle. The ALE pulse, which is primarily used as a timing pulse for external memory access, indicates when every instruction byte is fetched.	**1M each**
	(c)	**With the help of ADD instruction, explain** **1) Direct addressing mode** **2) Indirect addressing mode** **3) Register addressing mode** **4) Immediate addressing mode**	**4M**
	Ans:	**1.) Direct Addressing Mode:** Direct addressing means that the data value is obtained directly from the memory location specified in the operand. For example: ADD A, 40H- ADD the content of Acumulator and the content of memory location 40H and store the result in A. **2.) Indirect Addressing Mode:** Indirect addressing means that the data value is obtained from the memory location pointed by the register specified in the operand. Eg: ADD A, @R0 - ADD the content of A and the content of memory location pointed by R_0 and store the result in A. Note the @ symbol indicated that the indirect addressing mode is used. If the data is inside the CPU, only registers R_0 & R_1 are used for this purpose. **3.) Register Addressing Mode:** One of the eight general-registers, R0 to R7, can be	**1M each**

		specified as the instruction Operand. Example: ADD A, R5; Add the contents of register R5 to contents of A (accumulator) and store the result in A. **4.) Immediate Addressing mode:** Immediate addressing simply means that the operand (which immediately follows the Instruction op. code) is the data value to be used. Example: ADD A, #25H - ADD the content of A and the immediate data 25H and store the result in A. The # symbol tells the assembler that the immediate addressing mode is to be used.	
	(d)	**Explain various modes of serial communication operation.**	**4M**
	Ans:	**Mode 0 – 8 bit shift Register**- In this mode, the serial port works like a shift register and the data transmission works synchronously with a clock frequency of f_{osc} /12. Serial data is received and transmitted through RXD. 8 bits are transmitted/ received at a time. Pin TXD outputs the shift clock pulses of frequency fosc /12, which is connected to the external circuitry for synchronization In Mode 0, the baud rate is fixed at Fosc. / 12. **Mode 1 : 8-bit UART** 10 bits are transmitted through TXD or received through RXD. The 10 bits consist of one start bit (which is usually '0'), 8 data bits (LSB is sent first/received first), and a stop bit (which is usually '1'). Once received, the stop bit goes into RB8 in the special function register SCON. The baud rate is variable. $\text{BAUD RATE} = \frac{2^{SMOD}}{32} X \frac{\text{OSCILLATOR FREQ}}{12 \text{ X } [256-(TH1)]}$ **Mode 2 : 9-bit UART-** In this mode 11 bits are transmitted through TXD or received through RXD.The various bits are as follows: a start bit (usually '0'), 8 data bits (LSB	**1M each**

first), a programmable 9 th (TB8 or RB8)bit and a stop bit (usually '1'). While transmitting, the 9 th data bit (TB8 in SCON) can be assigned the value '0' or '1'. For example, if the information of parity is to be transmitted, the parity bit (P) in PSW could be moved into TB8.On reception of the data, the 9 th bit goes into RB8 in 'SCON', while the stop bit is ignored.

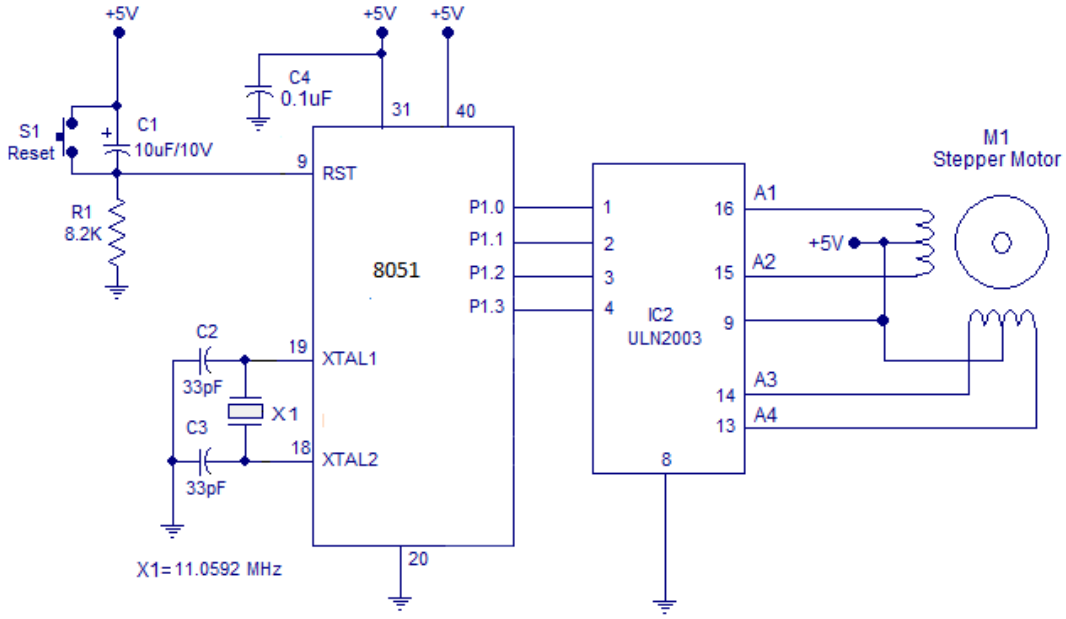
$$\text{BAUD RATE} = \frac{2^{SMOD}}{64} \times \text{OSCILLATOR FREQ}$$

Mode 3: 9-Bit UART with Variable Baud Rate- In this mode 11 bits are transmitted through TXD or received through RXD. The various bits are: a start bit (usually '0'), 8 data bits (LSB first), a programmable 9th bit and a stop bit (usually '1'). Mode-3 is same as mode-2, except the fact that the baud rate in mode-3 is variable (i.e., just as in mode-1).

$$\text{BAUD RATE} = \frac{2^{SMOD}}{32} \times \frac{\text{OSCILLATOR FREQ}}{12 \times [256-(TH1)]}$$

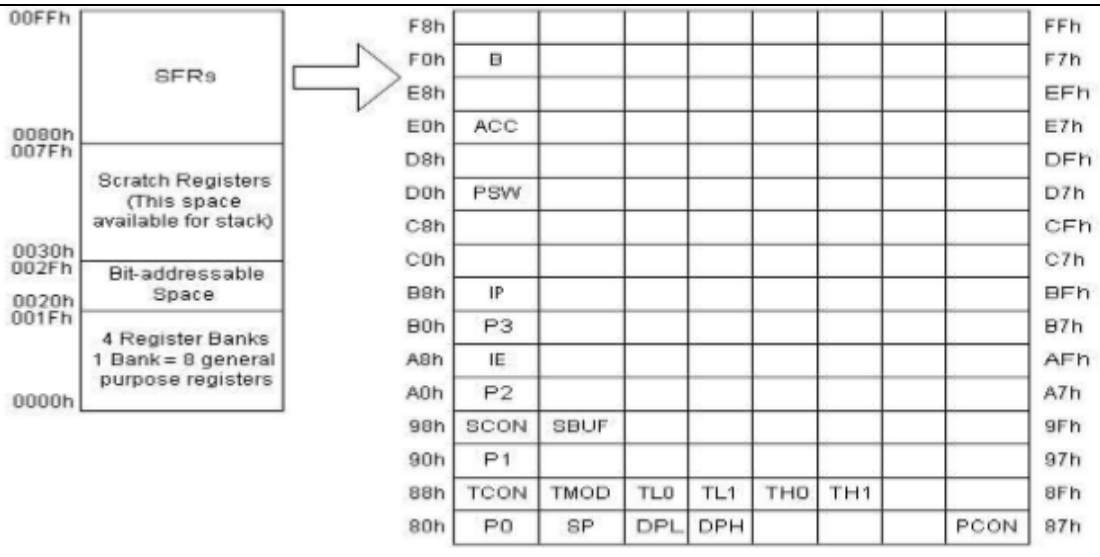
(B)	**Attempt any ONE :**	**06- Total Marks**
(a)	**Write ALP to find smallest number in an array of 10 numbers stored in Internal RAM.**	**6M**

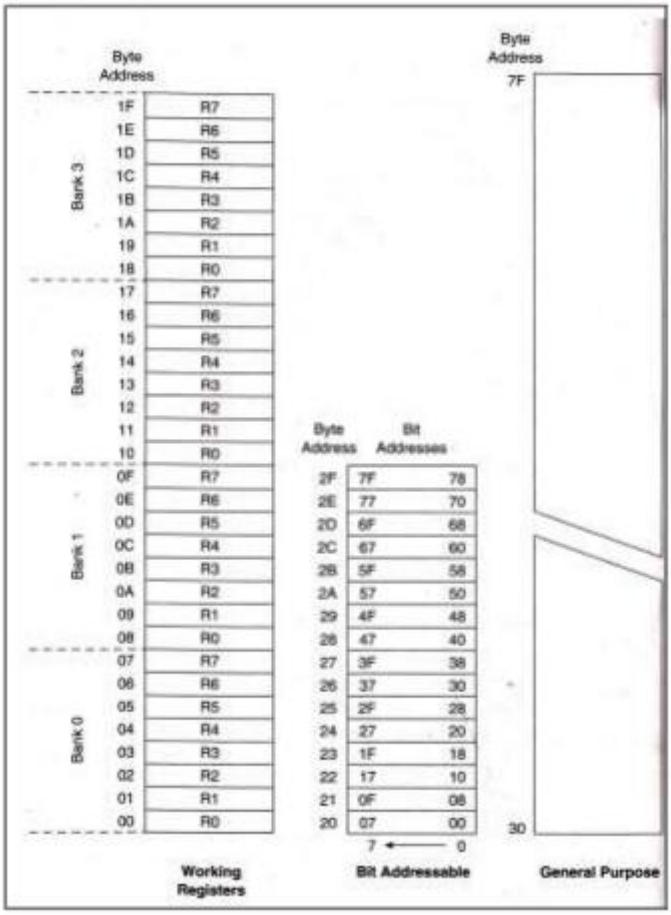
Ans:	ORG 0000H
	MOV R1, # 0AH ; Load R1 with 10 nos.
	MOV R0, # 40H ; load R0 with 40h
	DEC R1 ; decrement R1
	MOV 60H,@R0 ; move content of mem. loc^n pointed by R0 to 60h mem. location
	UP: INC R0 ; increment R0
	MOV A,@R0 ; move content of mem. location pointed by R0 to A
	CJNE A,60H,DOWN ; compare content of A and 60h and jump to down label if not
		6M for correct program

		equal AJMP SMALL ; if equal jump to small label DOWN: JNC SMALL ; jump to small label if no carry MOV 60H, A ; move the content of A to 60h SMALL: DJNZ R1,UP ; decrement R1 and jump to up label if not equal END **(For any other relevant logic marks can be given)**	
	(b)	**With the help of interfacing diagram, write ALP to rotate stepper motor clockwise.**	**6M**
	Ans:	 **PROGRAM:** ORG 0000H MOV A,#33H L1: MOV P1,A ACALL DELAY RR A SJMP L1	**3M diagram, 3M program**

```
DELAY:
MOV R1,#0FH
L2: MOV R2, # 0FFH
L: DJNZ R2, L
DJNZ R1, L2
RET
END
```

(For any other relevant logic marks can be given)

Q. No.	Sub Q. N.	Answers	Marking Scheme
2		**Attempt any TWO:**	**16- Total Marks**
	a)	**Draw and explain internal RAM organization of microcontroller 8051, and show address areas for each section.**	**8M**
	Ans:	 **Internal memory organization** **OR**	**4M diagram,4M explanation**



Internal ROM- The 8051 has 4K (4096 locations) of on-chip ROM. This is used for storing the system program. 2^{12} = 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. 2^8 = 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 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. Not all SFRs are bit-addressable. Those that are have a unique address for each bit.	
	b)	**Write ALP for 8051 microcontroller to generate a delay 0f 3µs using timer 1. Draw flow chart also. Assume crystal frequency 12MHz.**	**8M**

		Calculations:
		Formula to calculate count
		Crystal freq. = 12MHz
		Timer freq. – 12/12 =1MHz
		Timer Time period= 1 µs
		So (FFFF- COUNT+1)X 1µs= 3µs
		COUNT= FFFFH – 3 = FFFDH
		TH1= 0FFH
		TL1=0FDH
		Program:
		MOV TMOD, #10H ; Timer 1 , mode 1
		UP: MOV TL1, #0FDH; load count value in TL1
		MOV TH1, #0FFH; load count value in TH1
			1 M Calculation, 4M program, 3M flowchart

SETB TR1 ; start Timer1

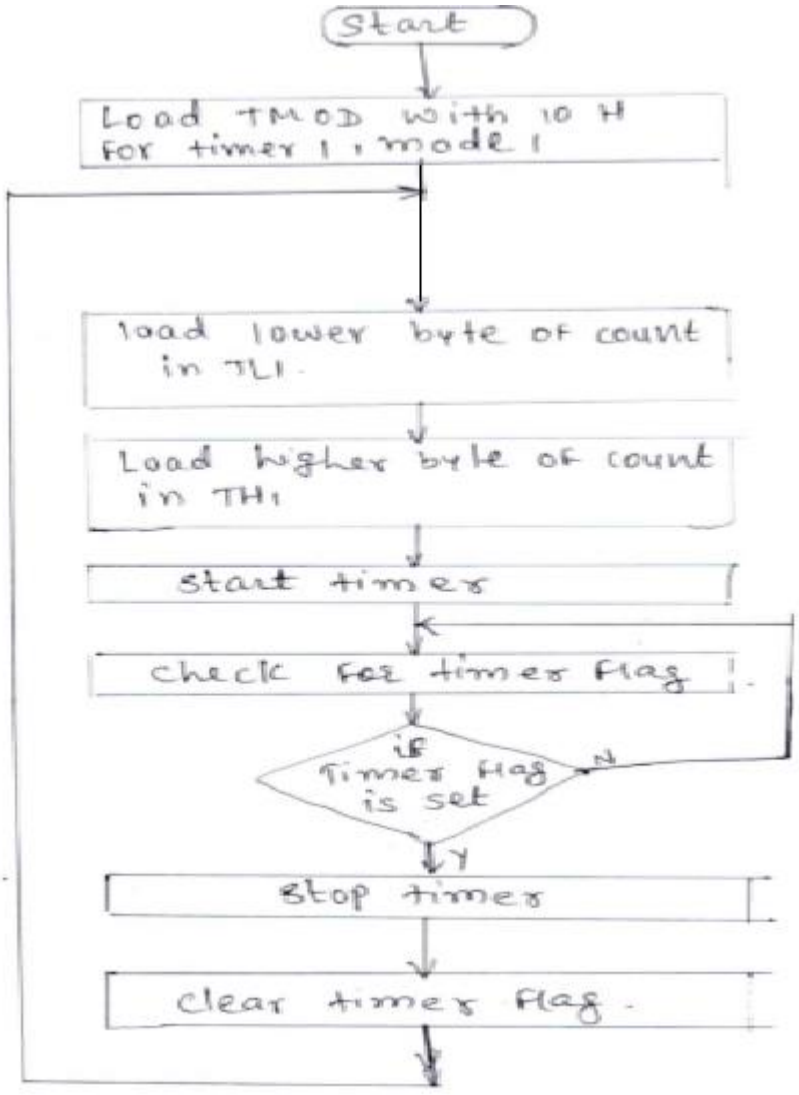
AGAIN: JNB TF1, AGAIN ; stay until timer rolls over

CLR TR1 ; stop timer

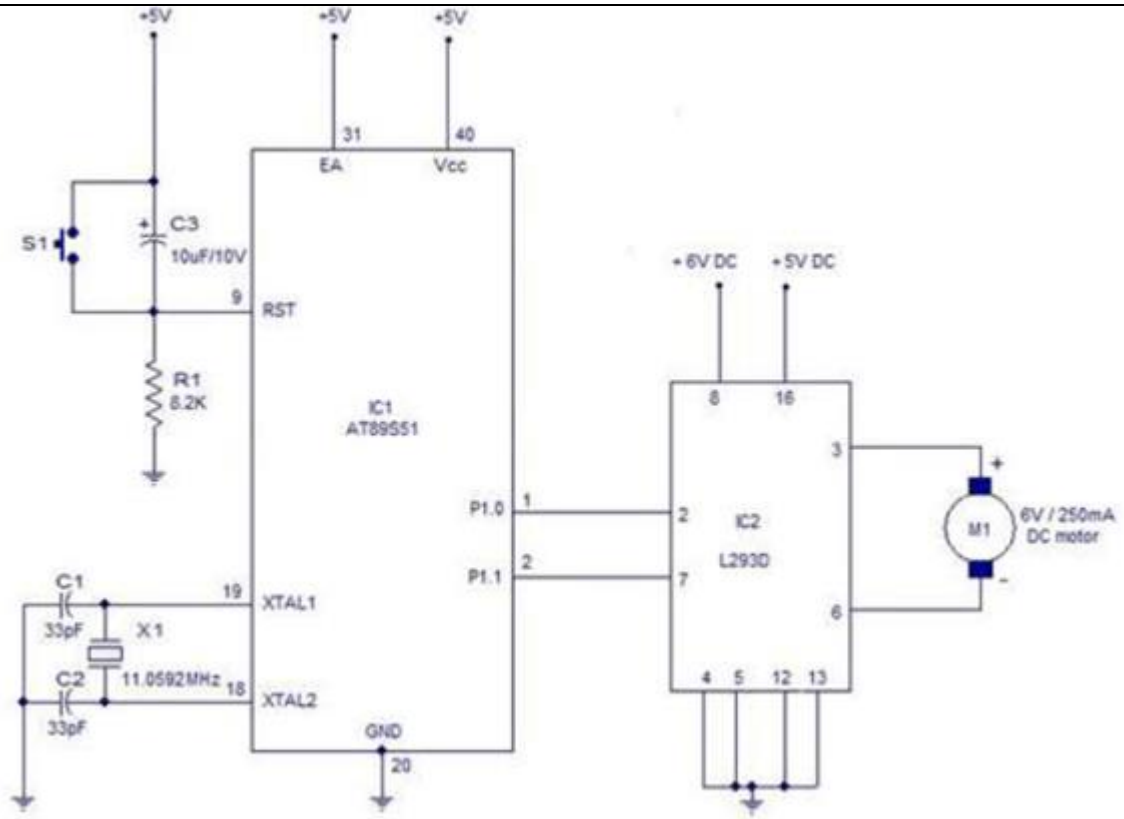
CLR TF1 ; clear timer flag

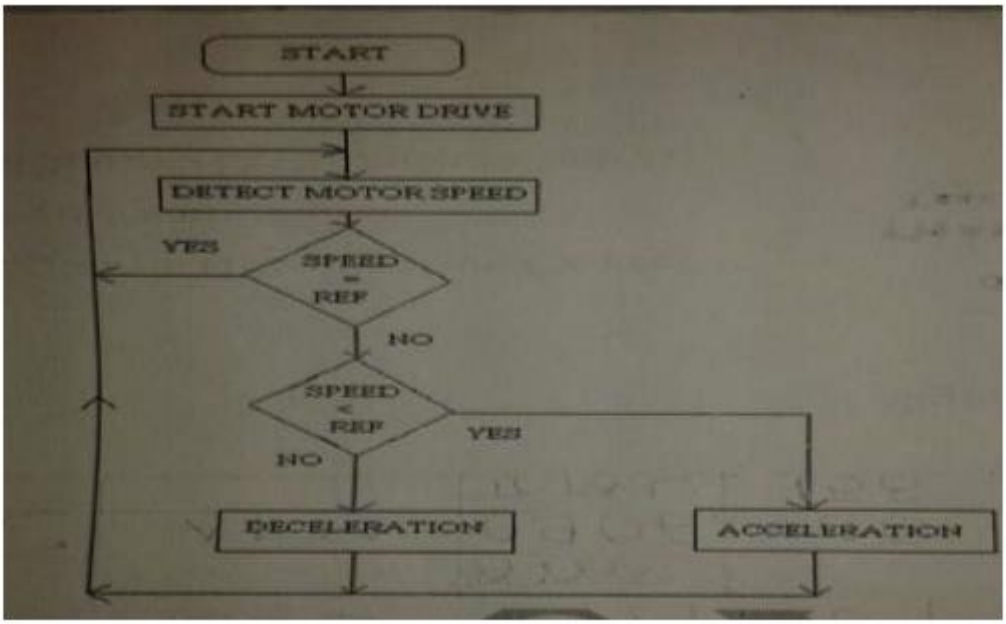
SJMP UP ; repeat

(For any other relevant logic marks can be given)



	c)	**Draw and explain interfacing diagram for DC motor speed control using 8051 µc. Also develop flow chart for same operation.**	**8M**

	Ans:	 **Explanation:** 1. The maximum current which can be sunk from 89C51 is upto 15mA at 5V. 2. But DC motor needs more current depending on its type. Hence DC motor cannot be connected directly to the microcontroller. 3. L293D motor driver stops the back emf produced by the motor from affecting the microcontroller. 4. Port pins P1.0 and P1.1 are connected to the input pins 2 and 7 of L293D motor driver. 5. The output pins of the L293D ,pin 3 and 6 are connected to the DC motor. 6. Direction of the rotation of the motor can be changed by inverting the output terminals	**3M diagram,** **3M Explaination,** **2M flowchart**



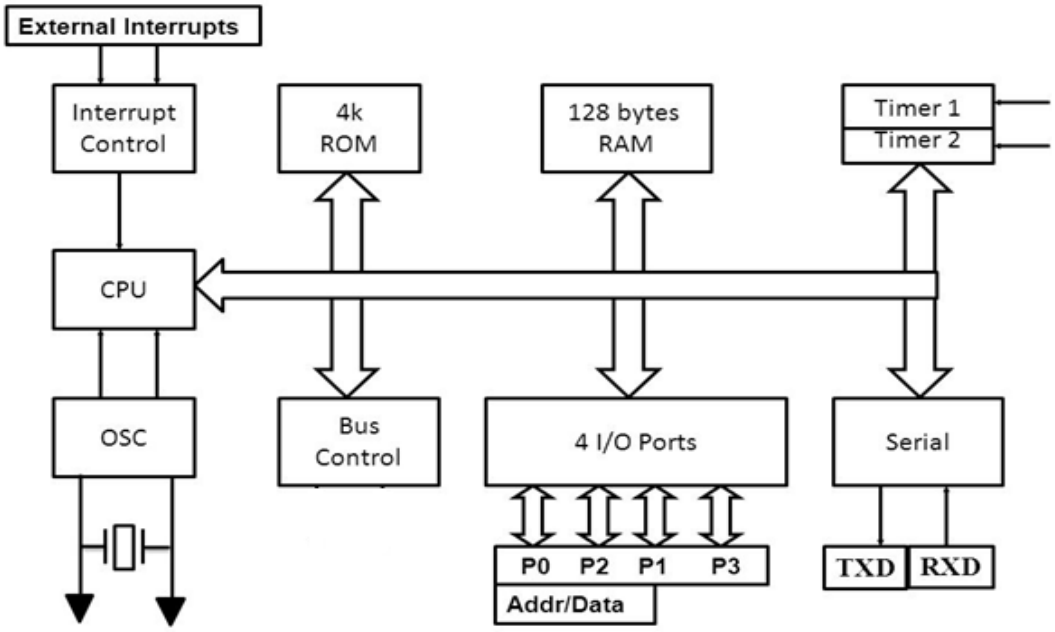
Q. No.	Sub Q. N.	Answers	Marking Scheme
3		**Attempt any FOUR :**	**16- Total Marks**
	a)	**Compare microprocessor and microcontroller.**	**4M**
	Ans:		**Any four points- 4M(Each 1M)**

Sr. No.	Microprocessor	Microcontroller
1	Microprocessor don't have inbuilt ROM and RAM	Microcontroller has inbuilt ROM and RAM
2	Microprocessor don't have inbuilt Timer and Counter	Microcontroller has inbuilt Timer and Counter
3	Microprocessor doesn't have inbuilt UART	Microcontroller has inbuilt UART
4	Microprocessor doesn't have inbuilt	Microcontroller has inbuilt I/O

	I/O PORTs	PORTs
5	It has many instructions to move data between memory and CPU.	It has one or two instructions to move data between memory and CPU.
6	It has one or two bit handling instructions.	It has many bit handling instructions.
7	Access times for memory and I/O devices are more.	Less access time for built-in memory and I/O devices.
8	Microprocessor based system requires more hardware.	Microcontroller based system requires less hardware reducing PCB size and increasing the reliability.

b) Draw block diagram of 8051 microcontroller. **4M**

Ans: **Diagram- 4M**



c) Explain following assembler directives: **4M**

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

	Ans:	**i) DB (define byte):** **Syntax-** LABEL: DB BYTE 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. Regardless of which type is used, the assembler will convert the numbers into hex. The DB directive is the only directive that can be used to define ASCII string larger than two characters. Example: DATA1: DB 28 ; (decimal data stored as 1C hex) DATA2: DB 01010101B ;(Binary data stored as 35 in hex) DATA3: DB 5FH ;(Hex) DATA4: DB "ABCDE" ; ASCII characters **ii) ORG (origin)** The ORG directive is used to indicate the beginning of the address. The 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. iii) **EQU: Equate** Syntax: **Name EQU Constant** **EQU**: This is used to define a constant without occupying a memory location. When the label appears in the program, constant value will be substituted for the label.	**Each directive- 1M**

Example:

NUMBER EQU 25H

MOV R3,#NUMBER ;

R3 = 25H as 25H will be substituted for NUMBER

iv) END

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

e.g. END ; End of the program

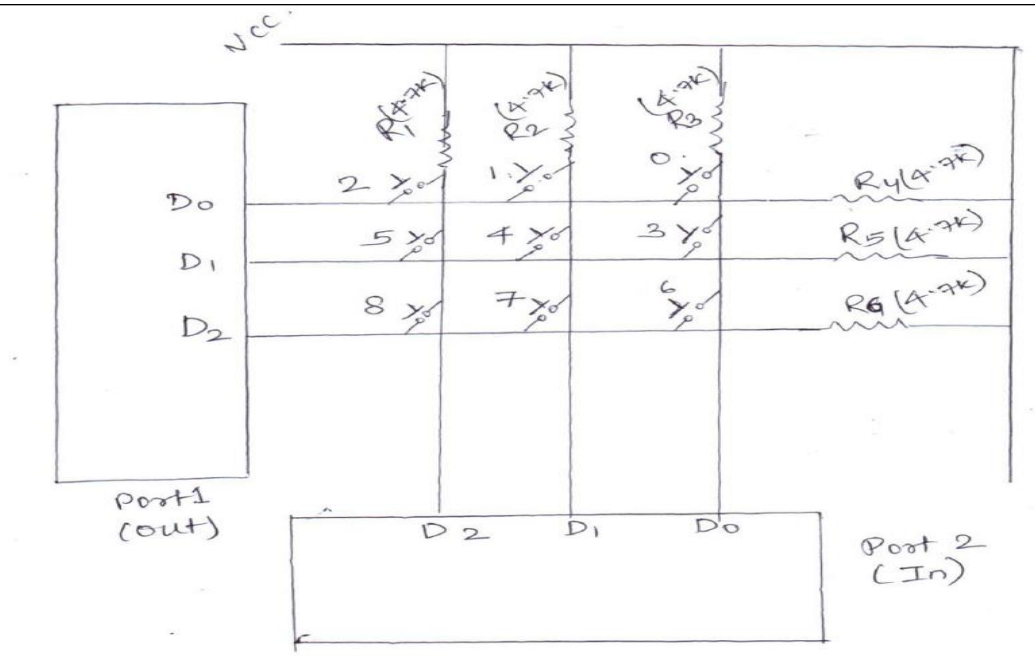
d) State any four 'C' data types with their range of values. **4M**

Ans:

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	0-65535
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

Any four data types- 4M(Each 1M)

e) Draw and explain interfacing of 3 x 3 key matrix with 8051 microcontroller. **4M**

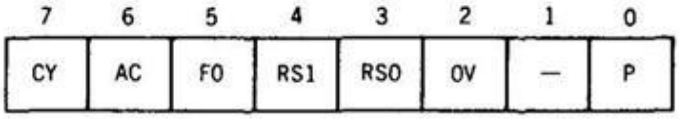
	Ans:	 **Explanation:** • It is the function of the microcontroller to scan the keyboard continuously to detect and identify the key pressed • First microcontroller checks whether all keys are open before the start of operation, by grounding all the rows and ensuring that the input port has 111. • To 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	**Interfacing diagram-2M** **& Explaination-2M**

		• 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	

Q. No.	Sub Q. N.	Answers	Marking Scheme
4	**A)**	**Attempt any THREE :**	**12- Total Marks**
	(a)	**Compare 8051 and 8052 microcontrollers.**	**4M**
	Ans:		**Any four points- 4M**

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**

(b)	**Draw format of PSW register of 8051 µc and explain function of any two flag.**	**4M**
Ans:	 **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 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 r**egister 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	**Format-2M** **&** **Function of any two flag-2M(1M each)**

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 directly by using instructions such as SETB C and CLR C.

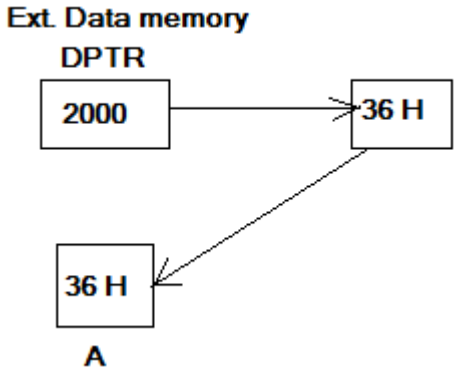
(c) **Describe function of following instructions of 8051 µc.** **4M**

1) **MOVX A,@DPTR**
2) **SWAP A**
3) **MUL AB**
4) **MOV A, R_0**

Ans: 1) **MOVX A,@DPTR** **Each descriptio**

MOVX instruction allows access to external data memory space. This instruction is used for general purpose access of data memory. This instruction will read the byte of data pointed to by register DPTR and store it in the accumulator.

Operation: A ← (DPTR)

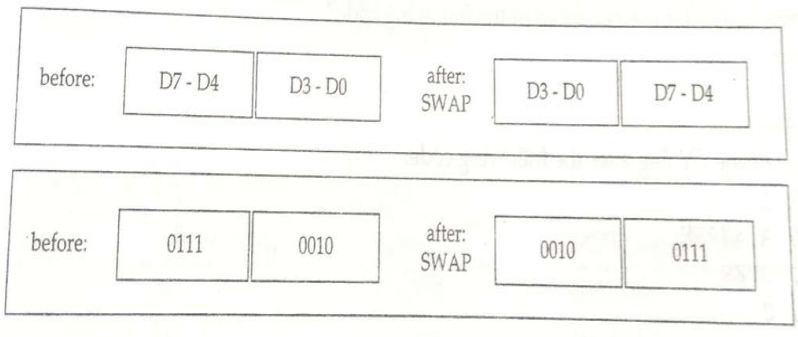


n-1M

2) SWAP A

Operation: A_{3-0} <- -> A_{7-4}

Swap the nibbles of the accumulator. It swaps the lower nibble and the higher nibble of the accumulator. The lower 4 bits are put into the higher 4 bits, and the higher 4 bits are put into the lower 4 bits**.**



3) MUL AB

Operation: BA ← A * B

In byte by byte multiplication, one of the operand must be in register A, and second

		operand must be in register B. after multiplication, the 16 bit result is held by A and B registers as lower byte is in A, and upper byte is in B. Addressing mode: Register addressing mode **4) MOV A,R_0** No. of bytes – 1byte Operation: A<--- R_0 Load the contents of R_0 in the accumulator. Addressing mode: Register addressing mode	
	(d)	**Write a program to send " WELCOME" serially on 8051 µc, use baud rate of 2400.**	**4M**
	Ans:	**Assume crystal frequency= 11.0592 MHz** The machine cycle frequency of the 8051 = 11. 0592 MHz / 12 = 921.6 KHz, and 921.6 KHz /32 = 28,800 Hz is the frequency provided by UART to Timer 1 to set baud rate. Therefore 28,800 / 2400 = -12 or F4 H is loaded into TH1. **(OR)** Required baud rate= 2^{SMOD} X f_{OSC} /32 X 12 (256 –COUNT) =2400 2^0 X 11.0592 X 10^6 /32 X 12 (256 – COUNT) =2400 256 – COUNT = 11.0592 X 10^6 /32 X 12 X 2400 = 12	**Baud rate calculation-1M** **&** **Program-3M**

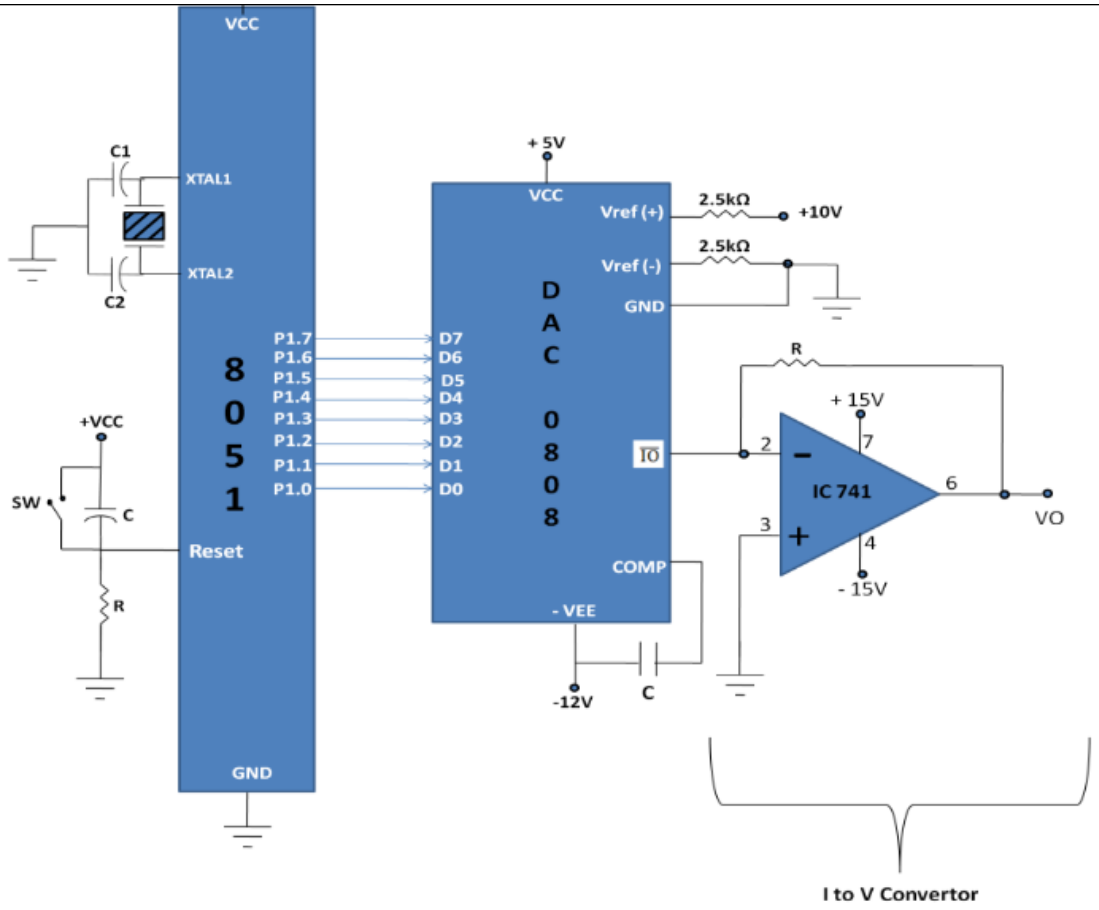
		Therefore COUNT = 256 – 12 =244D =F4 H.
		From this formula we get value to be loaded to TH1 as F4 H
		
		Program:
		#include <REG51.H>
		void main(void)
		{
		unsigned char text[] = “WELCOME”;
		unsigned char i;
		TMOD = 0x20;
		TH1 = 0XF4;
		SCON = 0x50;
		TR1 = 1;
		while(1)
		{
		for(i =0;i<7;i++)
		{
		SBUF= text[i];
		while(TI ==0);
		TI = 0;
		}
		}
		}	

B)	**Attempt any ONE :**		**06- Total Marks**
(a)	**List all I/O ports of 8051 µc and describe alternative functions of each ports.**		**6M**

	Ans:	**8051 Microcontroller provides 4 I/O ports as P0,P1, P2 & P3 each as 8 bit port.** **Alternate functions of each port:** i) **Port 0 (P0)-**Used as input or output. Port 0 also provides both address and data as AD0-AD7 when connected to external memory. The 8051 multiplexes address and data through port 0 to save pins. ALE indicates if P0 has address or data. When ALE = 1 ,it has lower 8 bit address A0-A7 and when ALE = 0 it provides data D0 – D7. ii) **Port 1 (P1)-**- Used only as simple input or output port. This port is called as dedicated I/O port. iii) **Port 2 (P2)-**- - This port can be used for higher address byte with addresses A8-A15.Used only as input or output port. However in external memory connections P2 must be used along with P0 to provide the 16 bit address for the external memory. P2 provides higher order bits A8- A15 of the address. iv) **Port 3 (P3)**- - Used as input or output port. Port 3 has additional function of providing some alternate functions like interrupts , RxD and TxD signals for serial communication. Following table shows alternate functions provided by port 3.	**List-1M** **Alternate functions of P0,P1 & P2- 1M each and** **P3 functions- 2M**

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

(b)	**With necessary interfacing diagram, write a 'C' program to generate a triangular wave using DAC.**	**6M**
Ans:		**Interfacing diagram- 3M & Program- 3M**




```
#include<reg51.h>
void main(void)
{
unsigned char d;
while(1)
{
for(d=0; d<=255; d++)
{
P1 = d;
}
for(d=255; d>=0; d--)
{
```

```
P1 = d;
}
}
}
```

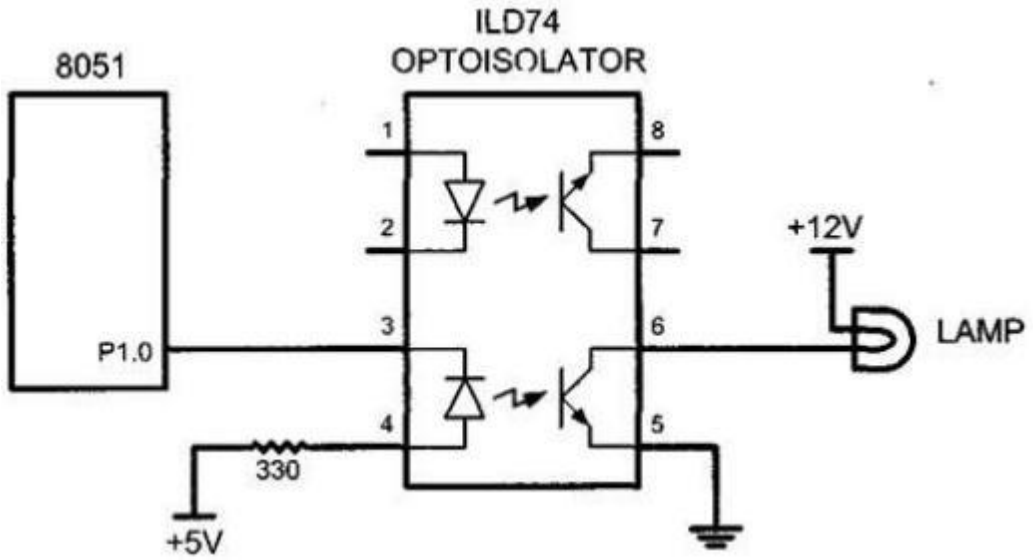
Q. No.	Sub Q. N.	Answers	Marking Scheme
5.		**Attempt any TWO:**	**16- Total Marks**
	a)	**Write ALP to exchange 10 bytes of data from source location 20H to destination location 60 H for 8051 µc.**	**8M**

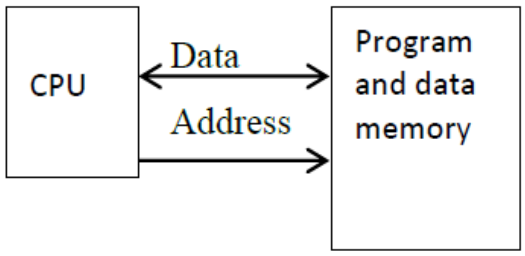
	Ans:	**(Program with any other relevant logic can be considered)**
		ORG 0000H
		MOV R0,#20H ; source pointer R0 = 20H
		MOV R1,#60H ; Destination pointer R1= 60H
		MOV R2,# 0AH ; count = 10
		UP: MOV A,@R0 ; Read first data byte
		MOV B,@R1 ; Read second data byte
		MOV @R0,B ; exchange numbers
		MOV @R1,A ;
		INC R0 ; increment source pointer
		INC R1 ; increment destination pointer
		DJNZ R2,UP ; decrement count and if count not equal to 0 jump UP
		END ; end of program
			(Correct program logic= 6 marks, Comments =2 marks)

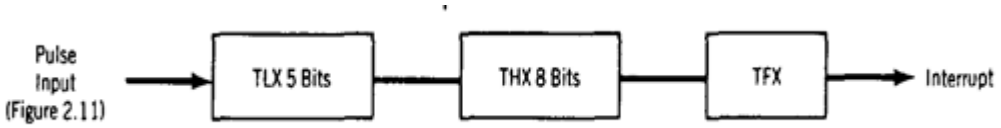
	b)	**Write 'C' language program to toggle all bits of port 1 of 8051 µc continuously with 5 ms delay.**	**8M**

	Ans:	**(Program with any other relevant logic can be considered)**
		
		#include < reg 51.h>
		void delay (unsigned int);
		void main (void)
		{
		while(1) //repeat loop
		{
		P1=0xff; //toggle all bits of port1
		delay (5); //add 5ms delay
		P1=0x00; //toggle all bits of port1
		delay (5); //add 5ms delay
		}
		}
		void delay (unsigned inti)
		{
		Unsigned int x,y;
		for(x=0; x<i ; x++)
		for(y =0 ; y <1275 ; y++);
		}
			(Correct program logic= 6 marks, Comments =2 marks)

	c)	**Draw neat labeled interfacing diagram to control lamp at pin 1.0, by using opto isolator with 8051 µc.**	**8M**

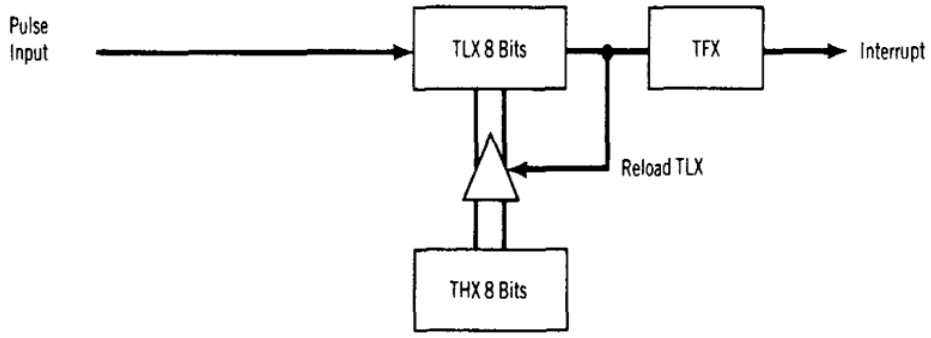
	Ans:		**(CORRECT DIAGRAM : 8M)**

Q. No.	Sub Q. N.	Answers	Marking Scheme
6.		**Attempt any FOUR :**	**16- Total Marks**
	a)	**Explain Von Neumann architecture with suitable diagram.**	**4M**
	Ans:	 • The Von Neumann architecture uses single memory for storing instructions and data. • It requires single bus for instructions and data. • Width of address and data bus for program and data memory are same • Instructions and data have to be fetched in sequential order limiting the	**(Diagram : 2 marks explainati on : 2 marks)**

		operation bandwidth • For e.g. general purpose microprocessors	
	b)	**With the help of neat diagram, describe timer modes of 8051 µc.**	**4M**
	Ans:	There are four timer modes: **1. Mode 0 :** 13 bit timer/counter 2. **Mode 1 :** 16 bit timer / counter **3. Mode 2 :** 8bit Auto Reload Mode **4. Mode 3 :** Split mode **1. MODE 0 - 13 bit Timer / Counter** In mode 0, the timer/counter is configured as a 13-bit timer/counter. The upper 8 bits of the count are in TH .The lower 5 bits are in the lower 5 bits of TL. The upper 3 bits of TL are not used. The TFx flag will be set when the timer /counter Overflows from all 1's to all 0's. The timer continues to count.  **2. MODE 1- 16 bit Timer / Counter** In mode 1, the timer/counter is configured as a 16-bit timer/counter. The upper 8 bits of the count are in TH & The lower 8 bits are in TL. The TFx flag will be set when the timer / counter overflows from all 1's to all 0's. The timer continues to count.	**Correct explanation:4M**

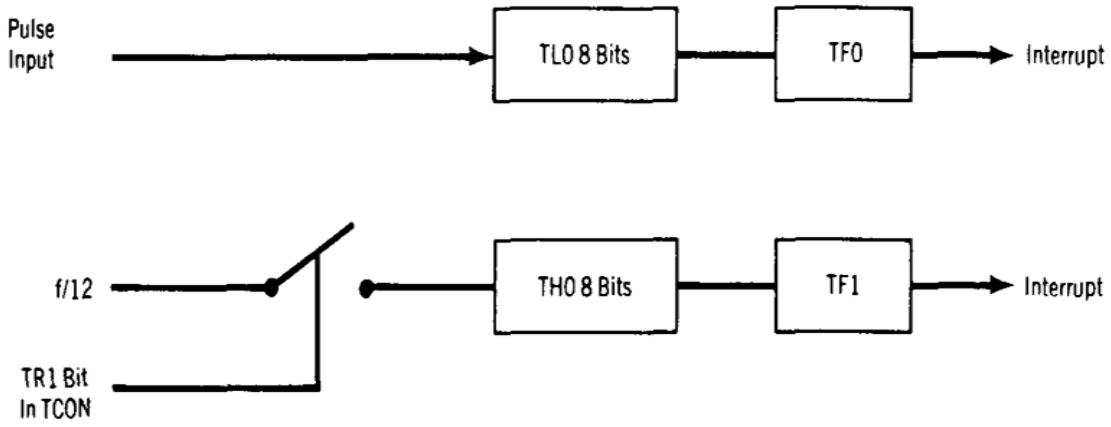
3. Mode 2 – 8 bit Auto Reload

TL operates as an 8-bit Timer / counter.TH holds a reload value. When TL overflows (reached FFH), the TFx flag is set, TL is reloaded from the value in TH and counting continues.



4. Mode 3- Split Mode

Timer 0 is split into two independent 8-bit timers.TL0 acts as 8 bit Timer / Counter When TL0 overflows, it sets the TF0 flag.TH0 acts as 8 bit Timer ,When TH0 overflows, it sets the TF1 flag. Timer 1 is stopped in mode 3. It can be switched independently to a different mode. However, when it overflows it will NOT set the TF1 flag.



c) **What is integrated development environment for microcontroller based system? Describe at least four features of Keil µ-version.** **4M**

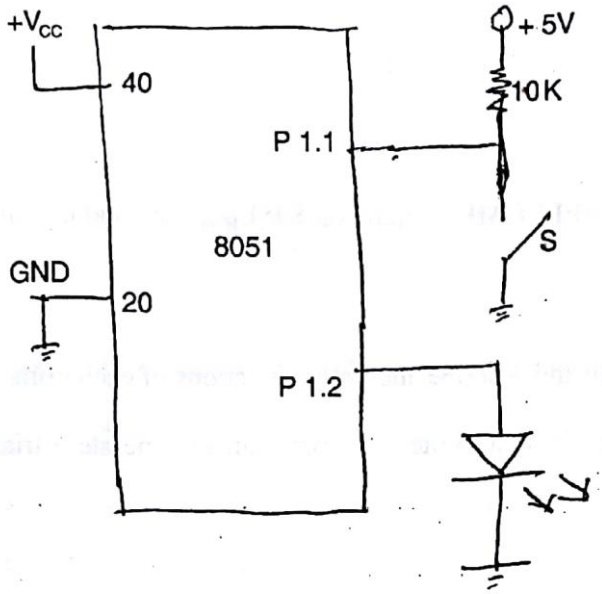
Ans:

Integrated Development Environment (IDE): **(IDE: 2M, Any 4 features of Keil: 2M)**

- An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
- An IDE normally consists of a project manager , source code editor, build automation tools and a debugger.
- Integrated development environments are designed to maximize programmer productivity by providing a single program in which all development is done.
- This program typically provides many features for authoring, modifying, compiling, deploying and debugging software.
- Some IDEs contain a compiler, interpreter, or both.
- Example : keil µVision IDE

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.

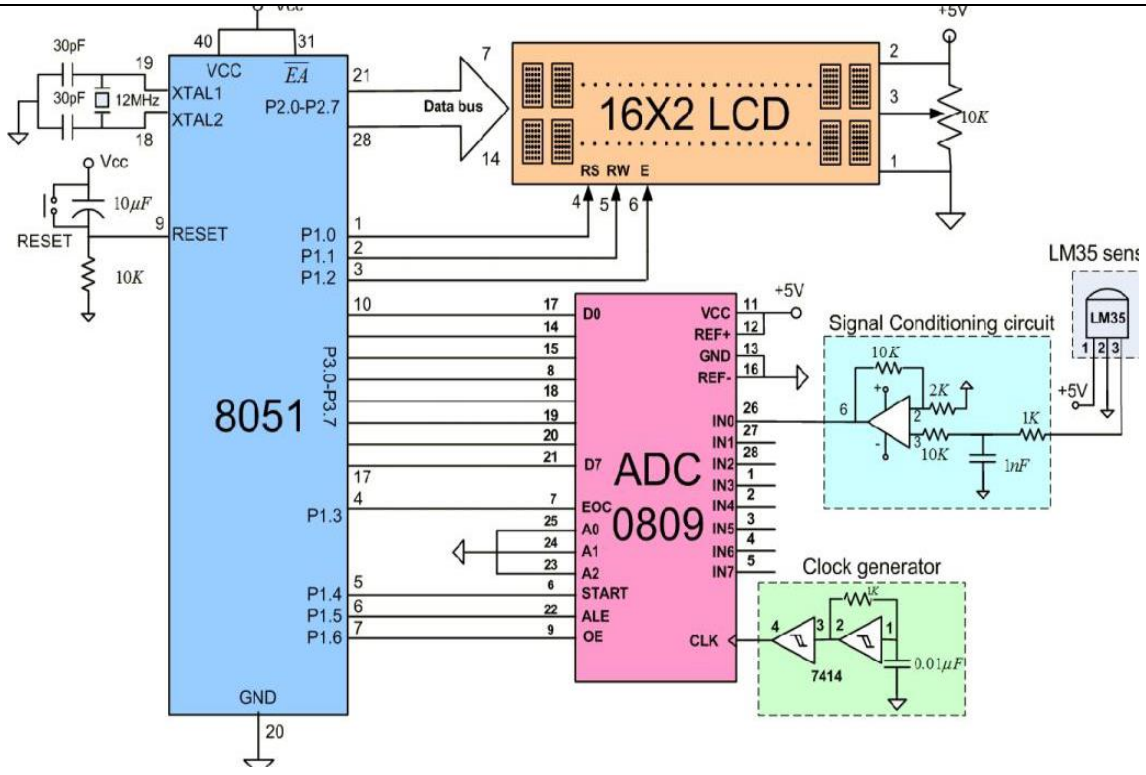
		• µVision is easy-to-use and accelerates your embedded software development. • µVision supports multiple screens and allows you to create individual window layouts anywhere on the visual surface. • The µVision Debugger provides a single environment in which you may test, verify, and optimize your application code. • The debugger includes traditional features like simple and complex breakpoints, watch windows, and execution control and provides full visibility to device	
	d)	**Write a program to read switch as shown below if switch is closed , turn ON the LED Else turn OFF the LED.**  	**4M**

	Ans:	`#include <reg51.h>`
		`Sbit SW = P1.1;`
		`Sbit LED = P1.2;`
		`void main(void)`
		`{`
		`SW =1 ;// make P1.1 as input pin`
		`LED =0;// make P1.2 as output pin`
		`While (1) // repeat forever`
			(Correct program logic= 4 marks)

```
{
 if ( SW == 0) // check if switch is closed
{
LED =1;// turn LED ON
}
Else
{
LED = 0;// turn LED OFF
}
}// End of while loop
}// End of program
```

e) **Draw interfacing diagram for temperature measurement using LM 35 sensor with 8051 µc.** **4M**

Ans:



(Correct diagram : 4 M)