V2V EDTECH LLP

Online Coaching at an Affordable Price.

OUR SERVICES:

- Diploma in All Branches, All Subjects
- Degree in All Branches, All Subjects
- BSCIT / CS
- Professional Courses

+91 93260 50669

v2vedtech.com

V2V EdTech LLP

v2vedtech

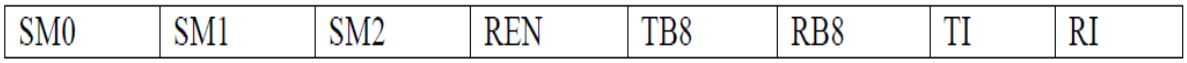
SUMMER-19 EXAMINATION

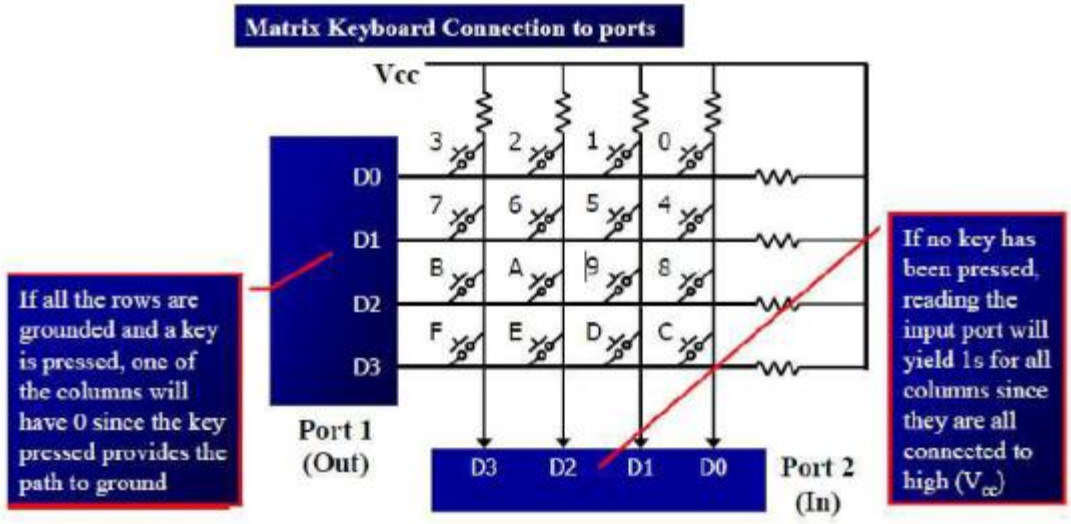
Subject Name: Microcontroller and applications Model Answer Subject Code: 22426

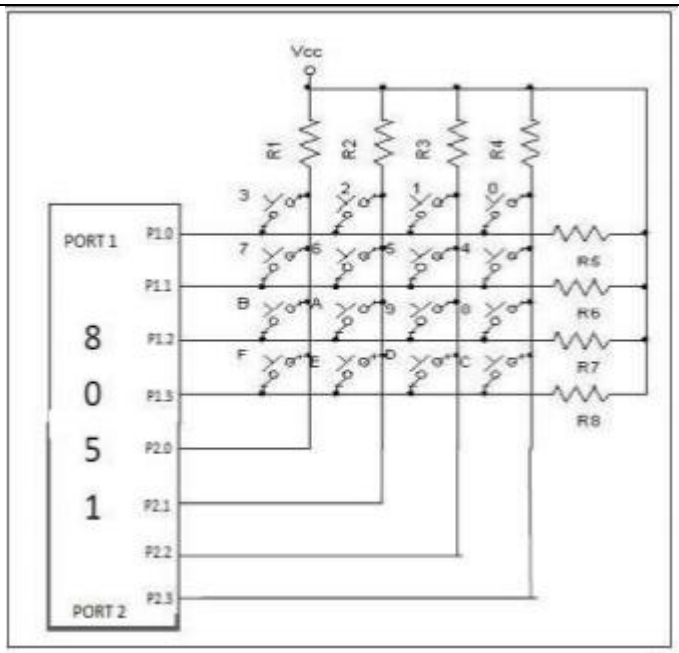
Important Instructions to examiners:

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.

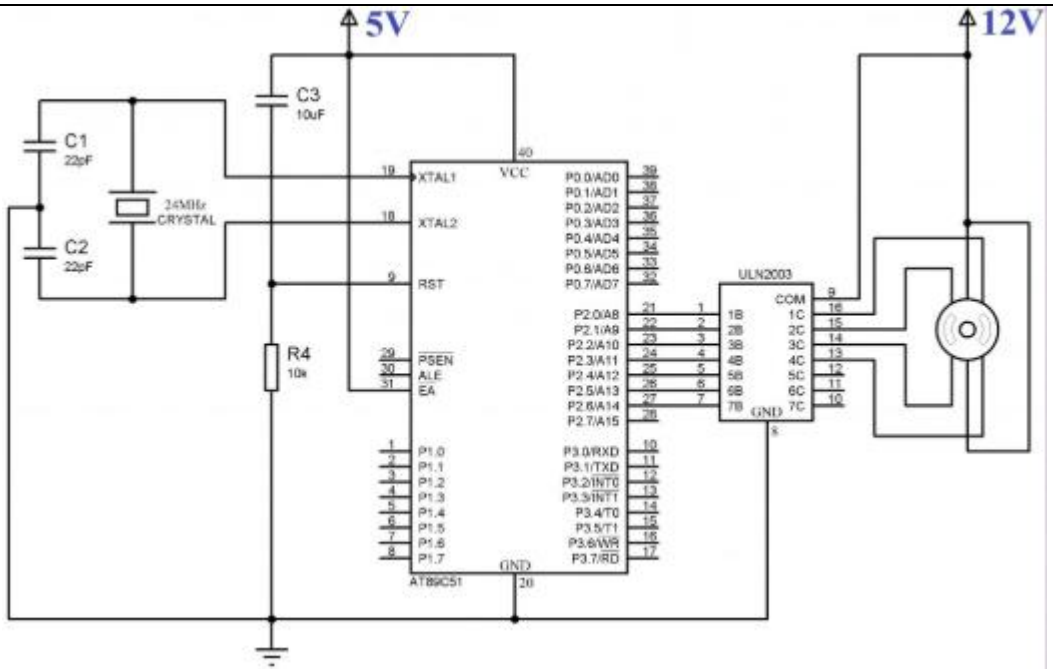
Q. No.	Sub Q. N.	Answers	Marking Scheme
1	**(A)**	**Attempt any FIVE of the following:**	**10- Total Marks**
	(a)	**State any four important features of 8051 microcontroller.**	**2M**
	Ans:	**Features of 8051 microcontroller: (Any Four)** 1) 8- bit data bus and 8- bit ALU. 2) 16- bit address bus – can access maximum 64KB of RAM and ROM. 3) On- chip RAM -128 bytes (Data Memory) 4) On- chip ROM – 4 KB (Program Memory) 5) Four 8-bit bi- directional input/output ports Four 8-bit bi- directional input/ output ports. 6) Programmable serial ports i.e. One UART (serial port) 7) Two 16- bit timers- Timer 0& Timer 1 8) Works on crystal frequency of 11.0592 MHz 9) Has power saving and idle mode in microcontroller when no operation is performed. 10) Six interrupts are available: Reset, Two interrupts Timers i.e. Timer 0 and Timer 1, two	**Each correct feature: ½ Mark**

		external hardware interrupts- INT0 and INT1, Serial communication interrupt for both receive and transmit.	
	(b)	**Find out the number of address lines required to access 4 KB of RAM**	**2M**
	Ans:	12 address lines required to access 4 KB of RAM as 2^{12} = 4KB	**Calculation:1M** **Answer: 1M**
	(c)	**List out any two instructions of following addressing modes:** **(i) Immediate addressing.** **(ii) Register addressing.**	**2M**
	Ans:	(i) Immediate addressing instructions: 1. MOV A, #36H 2. MOV DPTR, #27A2H (ii) Register addressing. 1. MOV A, R0 2. MOV R7, A (NOTE: Consider any relevant correct instructions)	**Each instruction ½ M**
	(d)	**Draw the format of SCON register.**	**2M**
	Ans:	 SM0 SM1 SM2 REN TB8 RB8 TI RI SM0 SCON.7 Serial port mode specifier SM1 SCON.6 Serial port mode specifier SM2 SCON.5 Used for multiprocessor communication (Make it 0.) REN SCON.4 Set/ cleared by software to enable/ disable reception. TB8 SCON.3 Not widely used. RB8 SCON.2 Not widely used TI SCON.1 Transmit interrupt flag. Set by hardware at the beginning of the stop Bit in mode 1. Must be cleared by software.	**2M for format** **Bitwise explaination optional**

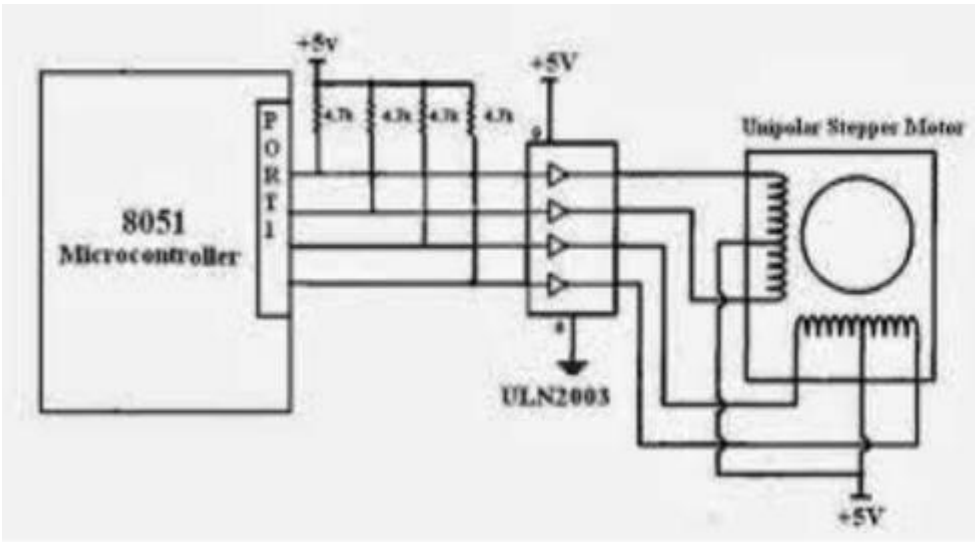
		RI SCON.0 Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.	
	e)	**Compare 8951 and 8031 derivatives of 8051 on the basis of :** **(i) RAM in bytes** **(ii) Timers used.**	**2M**
	Ans:	<table><tr><td>Parameter</td><td>8951</td><td>8031</td></tr><tr><td>RAM in bytes</td><td>128 Bytes</td><td>128 Bytes</td></tr><tr><td>Timers used</td><td>Two 16bit Timers</td><td>Two 16bit Timers</td></tr></table>	**Each Parameter : 1M**
	f)	**Draw interfacing diagram of 4×4 keyboard matrix with 8051 microcontroller.**	**2M**
	Ans:	 **OR**	**Diagram :2M**

			
	g)	**Define the term BUS related to microprocessor/controller and list different buses used in microcontroller.**	**2M**
	Ans:	**BUS:** A Bus is a set of physical connections used for communication between CPU and peripherals. **Different buses used in microcontroller are:** 1. Address Bus 2. Data Bus 3. Control Bus	**Define:1 M** **List:1M**

Q. No.	Sub Q. N.	Answers	Marking Scheme
2		**Attempt any THREE of the following:**	**12- Total Marks**
	a)	**Draw the interfacing of stepper motor and write an ALP to rotate in anticlockwise direction**	**4M**
	Ans:	**Interfacing diagram of stepper motor with 8051:**	**Diagram :2M**



OR



ALP to rotate motor in anticlockwise direction:

```
PROGRAM:
        MOV A, #66H ; load step sequence
BACK:   MOV P1, A ;   issue sequence to motor
AGAIN:  RL A ;        rotate left anticlockwise
        ACALL DELAY ; wait
        SJMP BACK ;   keep going
DELAY ;       delay subroutine.
MOV R2, #100
H1: MOV R3, #255
```

Program :2M

		H2: DJNZ R3, H2 DJNZ R2, H1 RET **(NOTE: Any other correct logic used for program should be considered)**	
	b)	**Describe power down mode and ideal mode of 8051 with circuit diagram . which SFR is used to set these modes and draw the same.**	**4M**
	Ans:	**IDLE MODE** In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer and Serial Port functions. The CPU status is preserved in its entirety, the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical state they had at the time idle mode was activated. ALE and PSEN hold at logic high levels. There are two ways to terminate the idle mode. i) Activation of any enabled interrupt will cause PCON.O to be cleared and idle mode is terminated. ii) Hard ware reset: that is signal at RST pin clears IDEAL bit IN PCON register directly. At this time, CPU resumes the program execution from where it left off. **POWER DOWN MODE** An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In the Power Down mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Special Function Register are maintained held. The port pins output the values held by their respective SFRS. ALE and PSEN are held low. Termination from power down mode: an exit from this mode is hardware reset. Reset defines all SFRs but doesn't change on chip RAM **PCON (Power Control Register) SFR is used to set these modes.**	**Power down mode:1 M** **Idle Mode:1 M** **Identification of PCON:1 M** **PCON Format: 1M**

Format of PCON:

PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	—	—	—	GF1	GF0	PD	IDL

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

c) State the alternative functions of port 3 of 8051 microcontroller. **4M**

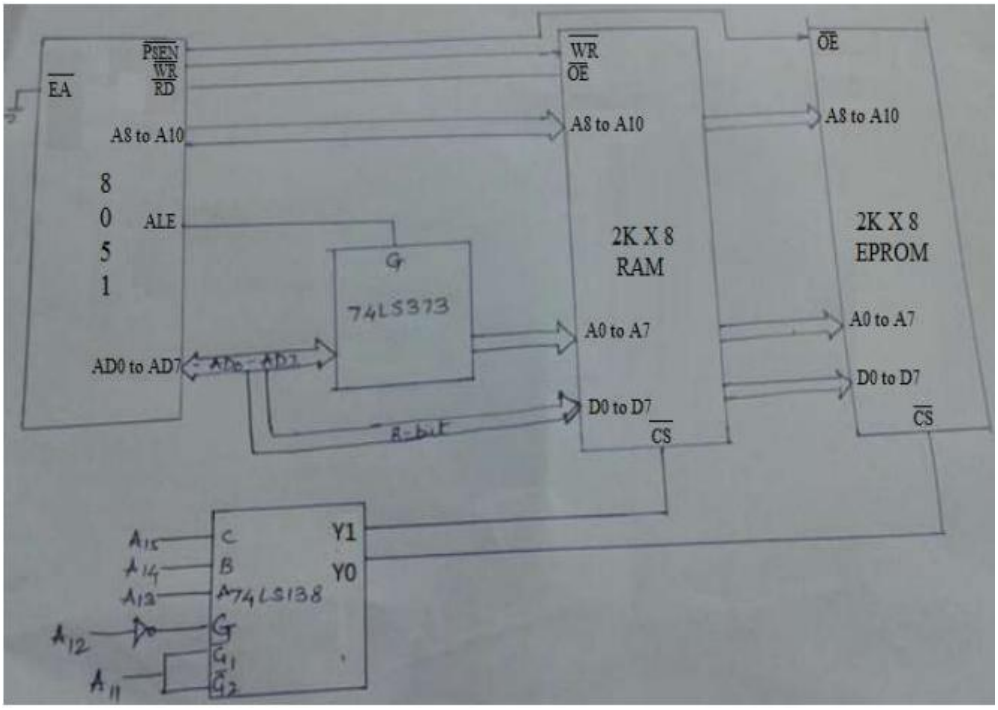
Ans: **Each pin function :1/2 M**

P3.0	RxD
P3.1	TxD
P3.2	$\overline{INT0}$
P3.3	$\overline{INT1}$
P3.4	T0
P3.5	T1
P3.6	$\overline{WR}$
P3.7	$\overline{RD}$

RXD it is used for serial input port
TXD it is used for serial output port
$\overline{INT0}$ used for external interrupt 0
$\overline{INT1}$ used for external interrupt 1
T0 Timer 0 external input
T1 Timer 1 external input
$\overline{WR}$ external data memory write strobe
$\overline{RD}$ external data memory Read strobe

d) Sketch interfacing diagram of 2 Kbyte RAM and 2Kbyte EPROM to 8051. Draw the memory map. **4M**

Ans:



Memory Map:

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	ADDR
Start addr of EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
End addr of EPROM	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FFH
Start addr of **RAM**	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H
End addr of **RAM**	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	27FFH

Q. No.	Sub Q. N.	Answers	Marking Scheme
3		**Attempt any THREE of the following :**	**12- Total Marks**
	a)	**Draw the format of PSW register of 8051 microcontroller and explain the function of each bit.**	**4M**
	Ans:		**2M format, 2M function**

CY	AC	F0	RS1	RS0	OV	--	P

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

1. CY: Carry flag.
This flag is set whenever there is a carry out from the D7 bit after an 8 bit addition or subtraction. It can also be set to 1 or 0 directly by instructions such as “SETB C” and CLR C” where “SETB C” stands for “set bit carry” and “CLR C” for “clear carry”.
2. AC: Auxiliary carry flag
If there is a carry from D3 and D4 during an ADD or SUB operation, this bit is set; it is cleared. This flag is used by instructions that perform BCD (binary coded decimal) arithmetic.
3. F0: Available to the user for general purposes.
4. RS0, RS1: Register bank selects bits
These two bits are used to select one of the four register banks from internal RAM as shown in given table. The user can use only one bank of register at one time. By default , bank 0 gets selected.

RS1	RS0	Space in RAM
0	0	Bank 0 (00H- 07H)
0	1	Bank 1 (08H-0FH)
1	0	Bank2 (10H-17H)
1	1	Bank3 (18H-1FH)

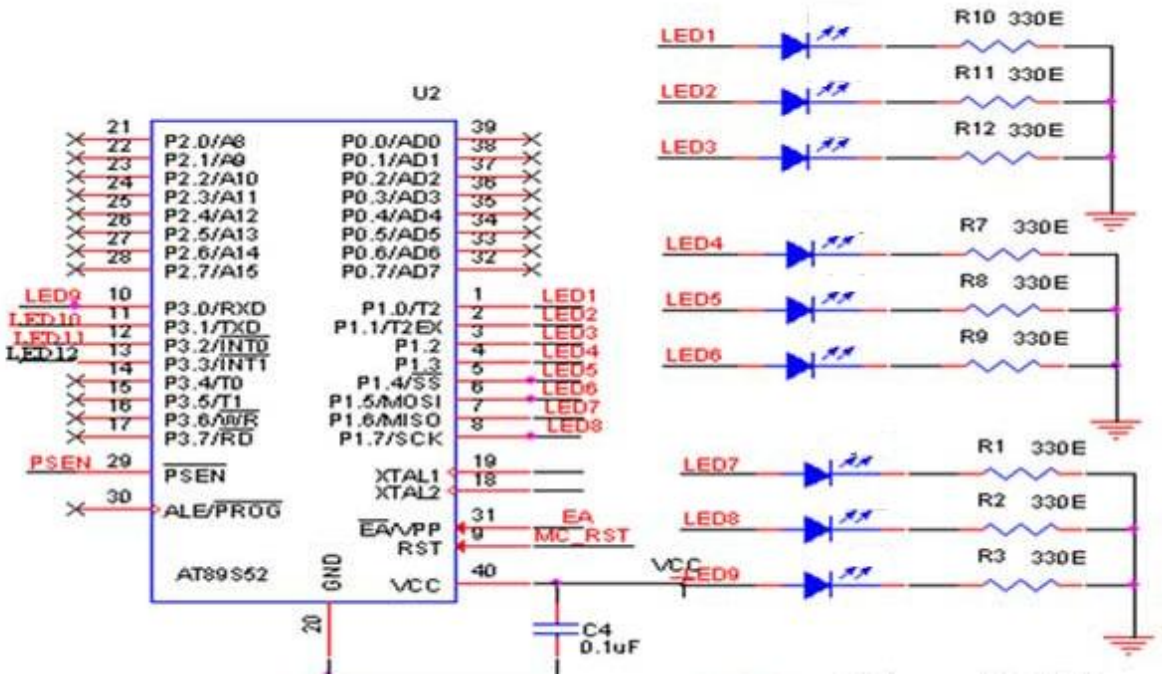
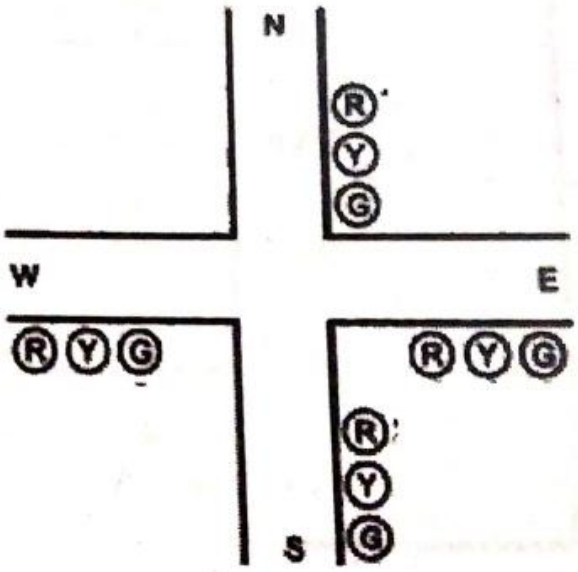
5. OV: Overflow flag

		This flag is set whenever the result of a signed number operation is too large, causing the high- order bit to overflow into the sign bit. In general, the carry flag is used to detect errors in unsigned arithmetic operations. The overflow flag is only used to detect errors in signed arithmetic operations. **6. P: Parity flag** The parity flag reflects the number of 1s in the A (accumulator) register only. If the A register contains an odd number of 1s, then P=1. P=0 if A has an even number of 1s.	
	b)	**Develop an ALP to generate square wave of 2 kHz on port pin P2.1 generate delay using timer 0 in mode 1. Assume crystal frequency of 11.0592 MHz.**	**4M**
	Ans:	**Calculation:** Crystal frequency= 11.0592 MHz I/P clock = (11.059 X 10^6)/12= 1000000 = 921.58KHz Tin = 1.085µ sec For 2 kHz square wave Fout = 2 KHz Tout = 1/ 2X 10^3 = 0.5msec =500µ sec So $T_{ON} = T_{OFF}$ = 250µ sec N = T_{ON} / Tin = 250/1.085 = 230.41 65535 – 231+1 = $(65305)_{10} = (FF19)_{16}$ **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 TF0, L1 ; poll till timer roll over CLR TR0 ; stop timer 0 CPL P2.1 ; complement port 2.1 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	**1M- Calculation, 2M program, 1M comments**
	c)	**State and explain the need of the following development tools microcontroller board:** **(i) Editor** **(ii) Assembler** **(iii) Compiler** **(iv) Linker**	**4M**
	Ans:	**1) Editor:** An editor is a program which helps you to construct your assembly language program in right format so that the assembler will translate it correctly to machine language. So, you can type your program using editor. This form of your program is called as source program and extension of program must be .asm or .src depending on which assembler is	**1M each**

		used. The DOS based editor such as EDIT, WordStar, and Norton Editor etc. can be used to type your program. **2) Assembler:** An assembler is programs that translate assembly language program to the correct binary/hex code for each instruction i.e. machine code and generate the file called as Object file with extension .obj and list file with extension .lst extension. It is used to find syntax error in the program. **3) Compiler:** Compiler is programs that translate C language program to the correct binary/hex code for each command i.e. machine code and generate the file called as Object file with extension .obj and list file with extension .lst extension. It is used to find syntax error in the program. **4) Linker:** A linker is a program, which combines, if requested, more than one separately assembled object files into one executable program, such as two or more programs and also generate .abs file and initializes it with special instructions to facilitate its subsequent loading the execution. Some examples of linker are ASEM-51 BL51, Keil u Vision Debugger, LX 51 Enhanced Linker etc.	
	d)	**List software and hardware interrupts used in 8051 with their vector addresses and priorities.**	**4M**
	Ans:		**2M-List,** **1M - Vector , 1M- priority**

Interrupt Source	Vector address	Interrupt priority
External Interrupt 0 –INT0	0003H	1
Timer 0 Interrupt	000BH	2
External Interrupt 1 –INT1	0013H	3
Timer 1 Interrupt	001BH	4
Serial Interrupt	0023H	5

Q. No.	Sub Q. N.	Answers	Marking Scheme
4		**Attempt any THREE of the following :**	**12- Total Marks**
	(a)	**Develop an 8051 based system for traffic light controlling .Draw interfacing diagram and write ALP for the same.**	**4M**

	Ans:	 Diagram shows four way traffic light control. 	**2M-DRAW,** **2M-PROGRAM**

LANE Direction	8051 LINES	TRAFFIC LIGHT
NORTH	P1.0(NR)	RED
	P1.1(NY)	YELLOW
	P1.2(NG)	GREEN
SOUTH	P1.3(SR)	RED
	P1.4(SY)	YELLOW
	P1.5(SG)	GREEN
EAST	P1.6(ER)	RED
	P1.7(EY)	YELLOW
	P3.0(EG)	GREEN
WEST	P3.1(WR)	RED
	P3.2(WY)	YELLOW
	P3.3(WG)	GREEN

Process:

1. **Allow traffic from W to E and E to W.**
2. **Yellow light ON.**
3. **Allow traffic from N to S and S to N**
4. **Yellow light ON.**
5. **Repeat Process**

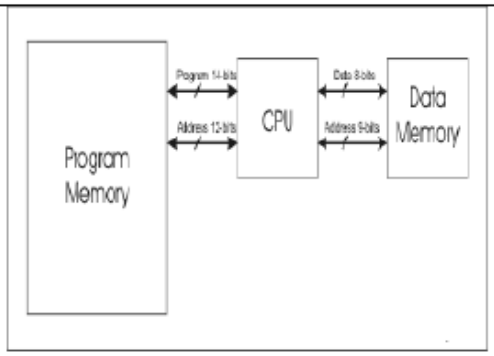
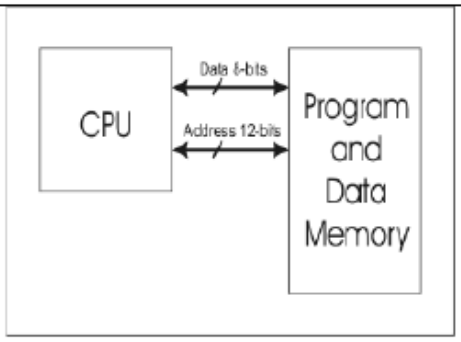
Program:

```
NR EQU P1.0

NY EQU P1.1

NG EQU P1.2
```

```
SR EQU P1.3
SY EQU P1.4
SG EQU P1.5


ER EQU P1.6
EY EQU P1.7
EG EQU P3.0


WR EQU P3.1
WY EQU P3.2
WG EQU P3.3


MOV  P1,#00H
MOV P3,#00H


AGAIN: SETB NR      ;North Red ON
SETB SR             ; South Red ON
SETB EG             ;East Green ON
SETB WG             ; West Green ON
ACALL DELAY
CLR EG              ;East Green OFF
CLR WG              ;West Green OFF
SETB EY             ; East Yellow ON
SETB WY             ; West Yellow ON
ACALL Y_DELAY     ; Small Delay for Yellow
```

```
CLR EY              ; East Yellow OFF
CLR WY              ; West Yellow OFF
SETB ER             ; East Red ON
SETB WR             ;West Red ON
CLR SR              ; South Red OFF
CLR NR              ; North Red OFF
SETB NG             ; North Green ON
SETB SG             ; South Green ON
ACALL DELAY
CLR NG              ; North Green OFF
CLR SG              ; South Green OFF
SETB NY             ; North Yellow ON
SETB SY             ; South Yellow ON
ACALL Y_DELAY
CLR NY              ; North Yellow OFF
CLR SY              ; South Yellow OFF
CLR ER              ; East Red OFF
CLR WR              ; West Red OFF
AJMP AGAIN


DELAY: MOV R0,#0FFH
L:MOV R1,#0FFH
DJNZ  R1,$
DJNZ R0,L
RET
```

		Y_DELAY: MOV R2,#0FFH **DJNZ R2,$** **RET** **END**	
	(b)	**Compare Von-Neumana and Harvard Architecture (any four points)**	**4M**
	Ans:		**1M Each**

Sr. No	Harvard Architecture	Van Neumann's Architecture
1.		
2.	The Harvard architecture uses physically separate memories for their instructions and data.	The Van Neumann's architecture uses single memory for their instructions and data.
3.	Requires separate & dedicated buses for memories for instructions and data	Requires single bus for instructions and data.
4.	Its design is complicated	Its design is simpler.
5.	Instructions and data can be fetched simultaneously as there is separate buses for instructions and data which increasing operation bandwidth.	Instructions and data have to be fetched in sequential order limiting the operation bandwidth.

(c) **List different timer modes of 8051 microcontroller and describe mode 2 with neat sketch.** **4M**

Ans:

1M- List,

1.5M- Diagram, 1.5M- describe

M1	M0	MODE	DESCRIPTION
0	0	0	13-bit timer
0	1	1	16-bit timer
1	0	2	8-bit auto-reload
1	1	3	Split mode

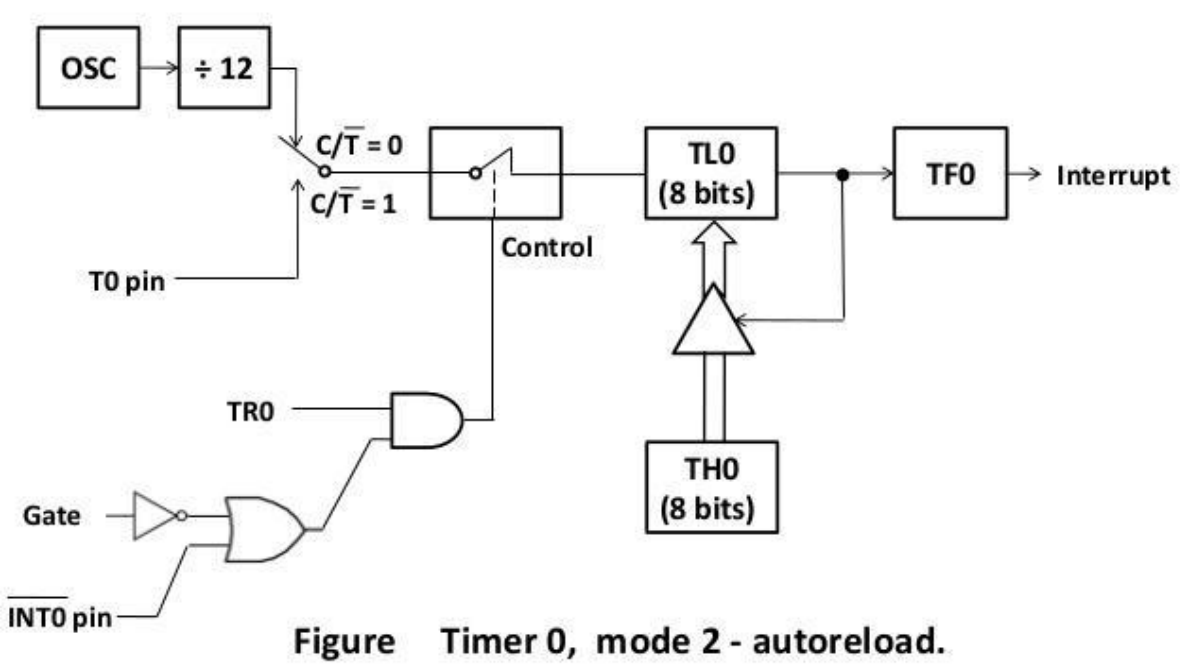
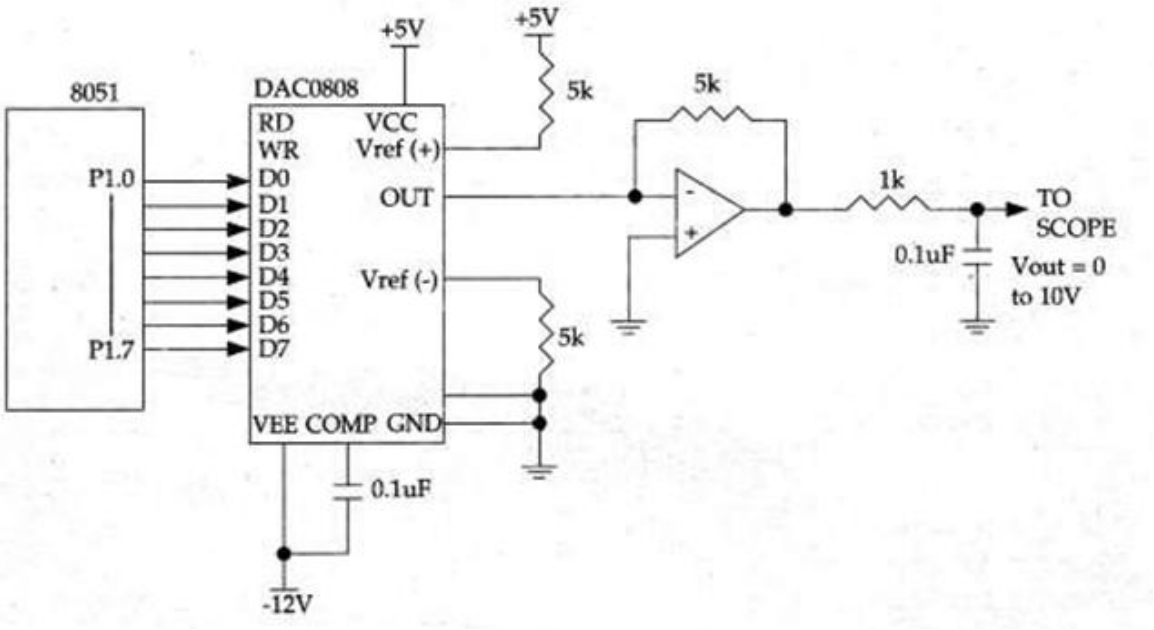


Figure Timer 0, mode 2 - autoreload.

To start the timer in mode 2, C/T= 0 and TR0=1 and the other input of AND gate is also 1.In this mode only TLX is used as 8-bit counter. THX is used to hold the value which is loaded in TLX initially. Every time TLX overflows from FFH to 00H the timer flag is set and the value from THX is automatically reloaded in TLX register.

	(d)	**Explain the interfacing diagram of DAC to 8051. Write an ALP to generate triangular waveform using DAC.**	**4M**
	Ans:	**(For any other relevant Program marks can be given)**	**2M diagram, 2M Program**



Program:

```
ORG 0000H
REPEAT: MOV A, #00H          ; Clear A
INCR: MOV P1, A              ; Send value to P1
INC A                        ; increment value
CJNE A, #0FFH, INCR          ;Compare with highest value
DECR: MOV P1, A
DEC A                        ; Decrement value
CJNE A,#00H, DECR            ;Compare with lowest value
SJMP REPEAT                  ; repeat
END
```

(e)	**Develop an ALP to transmit message "MSBTE" serially at baud rate 4800 8bit data , 1 stop bit. Assume crystal frequency of 11.0592 MHz .**	**4M**
Ans:	Org 0000h MOV TMOD, #20H ; timer 1, mode2	**3M program**

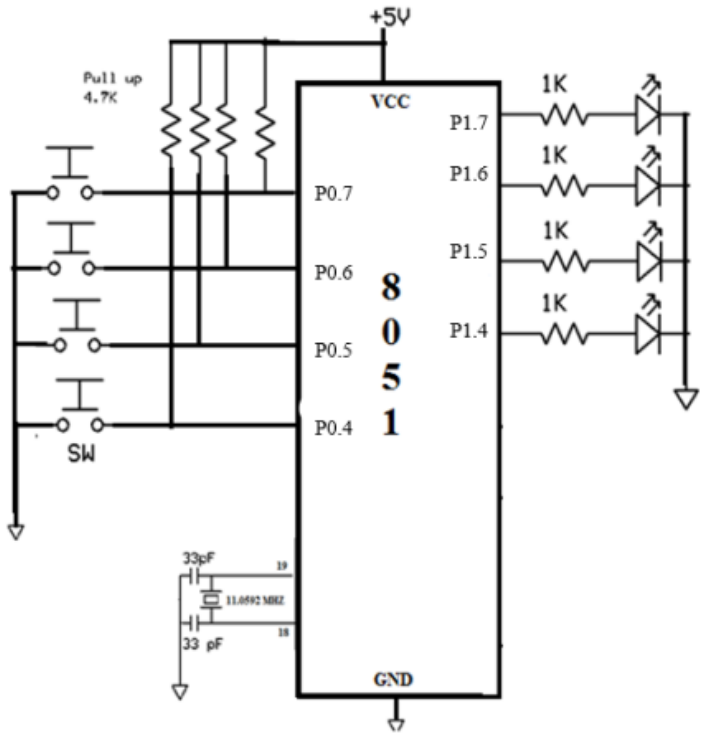
		MOV TH1,#-6 or MOV TH1,#0FAh ; 4800 baud rate
		MOV SCON, #50H ; 8-bit data,1 stop bit, REN enabled
		SETB TR1 ; Start timer 1
		AGAIN: MOV A, #"M" ; transfer "M"
		ACALL MESSAGE
		MOV A, #"S" ; transfer "S"
		ACALL MESSAGE
		MOV A, #"B" ; transfer "B"
		ACALL MESSAGE
		MOV A, #"T" ; transfer "T"
		ACALL MESSAGE
		MOV A, #"E" ; transfer "E"
		ACALL MESSAGE
		SJMP AGAIN
		
		MESSAGE: MOV SBUF, A
		JNB TI, $
		CLR TI
		RET
		END
			, 1M- Comments

Q. No.	Sub Q. N.	Answers	Marking Scheme
5.		**Attempt any TWO of the following:**	**12- Total Marks**
	a)	**Explain the various selection factors of microcontroller suitable for application.**	**6M**
	Ans:	The selection of microcontroller depends upon the type of application. The following factors must be considered while selecting the microcontroller. 1. Word length: The word length of microcontroller is either 8, 16 or 32 bit. As the word length increases, the cost, power dissipation and speed of the microcontroller increases. 2. Power dissipation: It depends upon various factors like clock frequency, speed,	**Any 6** **1 Mark— each factor**

		supply voltage, VLSI technology etc. For battery operated embedded systems, we must use low power microcontrollers. 3. Clock frequency: The speed of an embedded system depends upon the clock frequency. The clock frequency depends upon the application. 4. Instruction Set: On the basis of instructions microcontrollers are classified into two categories 1. CISC 2. RISC. CISC system improves software flexibility. Hence it is used in general purpose systems. RISC improves speed of the system for the particular applications. 5. Internal resources: The internal resources are ROM, RAM, EEPROM, FLASH ROM, UART, TIMER, watch dog timer, PWM, ADC, DAC, network interface, wireless interface etc. It depends upon the application for which microcontroller is going to be used. 6. I/O capabilities: The number of I/O ports, size and characteristics of each I/O port, speed of operation of the I/O port, serial port or parallel ports. These are the considerations needed to ascertain. 7.Memory: For mass production of microcontrollers ROM versions and for lesser production EPROM version or CPU version with external program memory is suitable	
b)	**Develop a program to transfer block of 05 numbers. From memory location 50H to 60H.**	**6M**

Ans:	**NOTE: Program may change. Please check the logic and understanding of students**
	
	ORG 0000H ; Program from 0000H
	CLR PSW.3 ; select bank 0
	CLR PSW.4 ;
	MOV R3, #05H ; Initialize Byte counter
	MOV R0, #50H ; Initialize memory pointer for source array
	MOVR1,#60H ; Initialize memory pointer for destination array
	; therefore R0 → Source pointer
	; R1 → destination pointer
	UP : MOV A, @R0 ; Read number from source array
		4 M— Correct Program ,2 M-comments

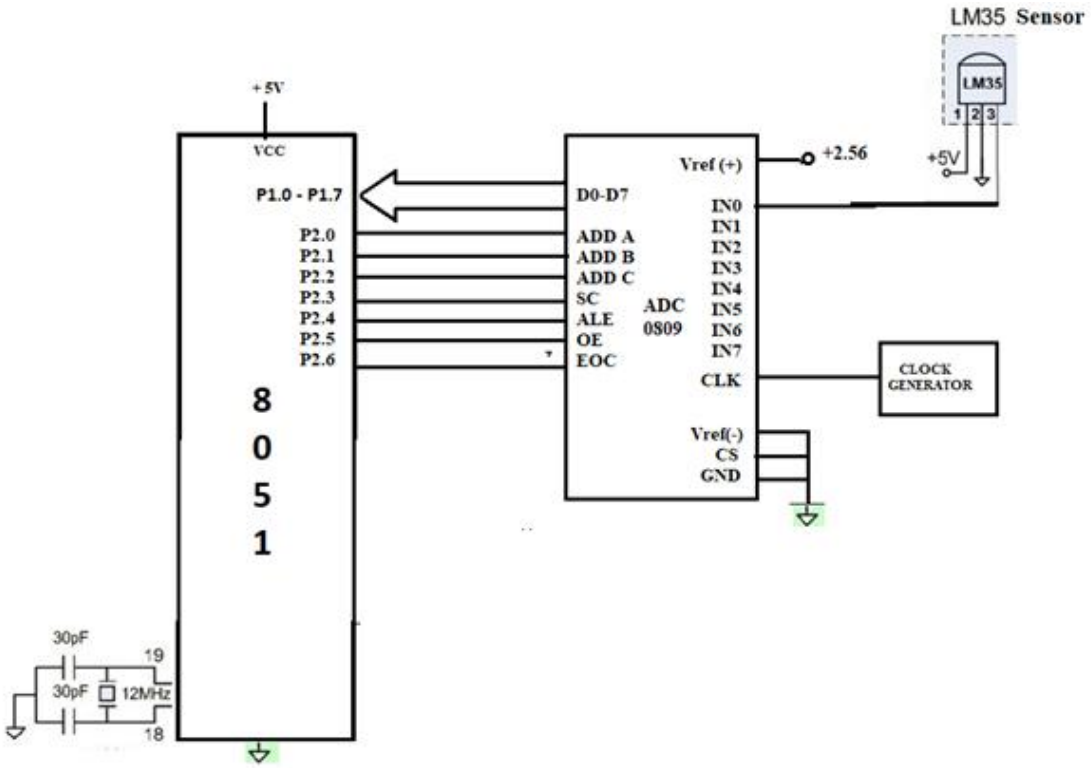
```
        MOV  @R1, A        ; Write number to destination array
        INC  R0            ; Increment source memory pointer by 1
        INC  R1            ; Increment destination memory pointer by 1
        DJNZ R3, UP        ; Decrement byte counter by 1
                           ; Is it zero? No, jump to UP
HERE :  SJMP HERE
        END    ; Stop
```

	c)	**Sketch 8051 interfacing diagram to interface 4 LED's and 4 switches. Interface switches to port 0 and LED to port 1 upper nibble. Develop an ALP to read status of switches and operate LED's as per switch status.**	**6M**
	Ans:	 **NOTE: Program may change. Please check the logic and understanding of students**	**3 M - correct interfacing diagram, 3 M - correct program**

PROGRAM TO DISPLAY STATUS OF SWITCHES ON LED:

```
        ORG 0000H
         MOV P0, #0F0H            ; Make P0 as input
START:   MOV A, P0                ; Read status of the key
         CJNE A, #0F0H, CHECK1    ; Key pressed branch from Port 0
         SJMP START               ; Jump to start
CHECK1: LCALL DELAY               ; Call Key debounce delay
         MOV A, P0                ; Read data from port 0
         CPL A                    ; Complement A
         MOV P1, A                ; Send data to LED
         SJMP START               ; Jump to start

DELAY:   MOV R1,#0FFH             ; Delay program
UP:      MOV R2, #0FFH;
HERE:    DJNZ R2, HERE
         DJNZ R1, UP
         RET
         END
```

Q. No.	Sub Q. N.	Answers	Marking Scheme
6.		**Attempt any TWO of the following :**	**12- Total**

			Marks
	a)	**Develop an ALP to read temperature from LM 35 sensor. Draw the interfacing diagram with 8051**	**6M**
	Ans:	NOTE: Program may change. Please check the logic and understanding of students  **Program:** ORG 0000H ADDR_A BIT P2.0 ADDR_B BIT P2.1 ADDR_C BIT P2.2 SC BIT P2.3 ALE BIT P2.4 OE BIT P2.5 EOC BIT P2.6	**3 M – Correct diagram, 3 M- Correct Program**

```
        MY_DATA EQU P1
        ORG 0000H
        MOV MY_DATA,#0FFH ; make P1 as input
        SETB EOC ; make EOC an input
        CLR ALE ; clear ALE
        CLR SC ; clear SC
        CLR OE ;clear OE
         CLR ADDR_C ; C=0
         CLR ADDR_B ; B=0
         CLR ADDR_A ; A=0(select channel 0)
         ACALL DELAY
         SETB ALE ;latch address
         ACALL DELAY
BACK:   SETB SC ;start conversion
        ACALL DELAY
        CLR ALE
        CLR SC
HERE:   JB EOC,HERE       ; wait
HERE1: JNB EOC,HERE1
        SETB OE
        ACALL DELAY
        MOV A, MY_DATA
        MOV P1, A
         CLR OE
        SJMP BACK

DELAY: MOV R3,#25 ;Delay Subroutine
    L3: MOV R4,#100
    L2: MOV R5,#100
    L1: DJNZ R5,L1
         DJNZ R4,L2
         DJNZ R3,L3
          RET
          END
```

	b)	**Develop a program to toggle the LED's after every 500m sec connected to P1.0 and P1.1 after receiving the external interrupt on INT0.**	**6M**

	Ans:	**NOTE: Program may change. Please check the logic and understanding of students**
		Solution :
		Crystal freq=11.0592MHz
		Timer frequency=11.0592MHz/12
		Time=12/11.0592MHz=1.085µs
		For delay of 50 ms,
		50ms/1.085µs=46082
		Therefore, count to be loaded in TH1 and TL1 can be calculated as
		65536 - 46082 =19454D=4BFEH
		
		Note: If crystal frequency is taken as 12MHz then count to be loaded in TH1 and TL1 will be 3CB0h.
		
		Program:
		ORG 00 H
		LJMP MAIN
		ORG 0003 H
		MOV TMOD, #10H ; Timer1, mode 1
		HERE : MOV R0, #0AH ; Counter for 500ms (50*10)delay
		BACK : MOV TL1, # B0H ; load count value in TL1
		MOV TH1, #3CH ; load count value in TH1
		
		SETB TR1 ; start Timer 1
		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
		CPL P1.0 ; Toggle P1.0
			4 M-correct program ,1 M-delay calculation,1M-comments

		CPL P1.1 ; Toggle P1.1 RETI ; repeat MAIN : MOV IE, #81H ; Enable the external interrupt 0 SETB P3.2 ; P3.2 as input pin HERE : SJMP HERE END	
	c)	**Explain the following instructions.** **SWAP A** **ADD C** **MUL AB** **CJNE A, add, radd** **MOV A, R_0** **MOVX A, @ A + DPTR.**	**6M**
	Ans:	**SWAP A** Description: This instruction exchanges bits 0-3 of the Accumulator with bits 4-7 of the Accumulator. This instruction is identical to executing "RR A" or "RL A four times Example: MOV A, #59H ; A= 59H SWAP A ; A= 95H **ADD C** Description: This instruction is used to perform addition of two eight-bit numbers along with carry. The result is stored in accumulator which is the default destination. Example: ADDC A, R0 : Add contents of accumulator, R0 and carry .The result is stored in accumulator. **MUL AB** Description: the multiplicand and the multiplier must be in A and B registers. After multiplication if the result is 8 bit it will be in the accumulator and if the result is larger than 8 bit ,lower byte of result will be in accumulator and higher byte will be in register B. Example :MOV A,#10H MOV B,#02 H MUL AB	**1 M – each instruction.**

After execution A=20H,B=0 H

CJNE A, add, radd

Description: Compare the contents of the accumulator with the 8 bit data in memory address mentioned in the instruction and if they are not equal then jump to the relative address mentioned in the instruction.
Example: CJNE A, 04H, UP: Compare the contents of the accumulator with the contents of 04H memory and if they are not equal then jump to the line of instruction where UP label is mention

MOV A,R_0
Description: this instruction copies the contents of source register R0 into accumulator. The register R0 remains unaffected.
Example: Before Execution A=43 H, R0=32 H
After execution A=32 H, R0-32H

MOVX A, @ A + DPTR. (Consider it as MOVC A,@A+DPTR)

Description: Copy the contents of code memory pointed by the sum of Accumulator and DPTR to the Accumulator

MOVC is a move instruction, which moves data from the 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 is referred to as the index address.

(NOTE : If student has attempted to solve considering as above or attempted to solve as given in question paper, give appropriate marks)
