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

Winter – 15 EXAMINATIONS

Subject Code: **17534** **Model Answer**

Important Instructions to examiners:

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

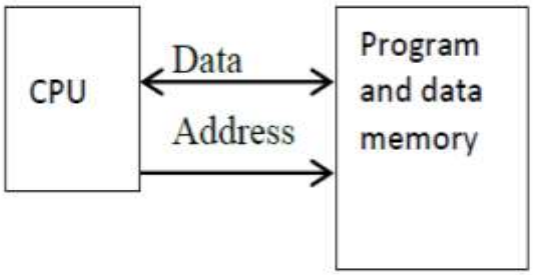
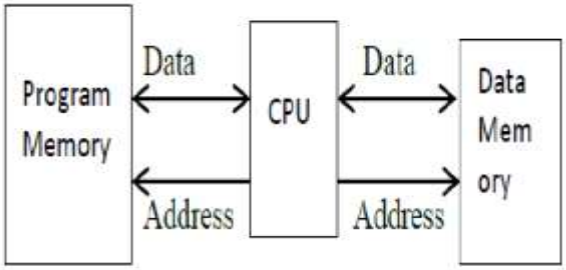
6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.

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

Q.1 a) Attempt any THREE of the following **12 marks**

a) State the difference between Harvard and Von – Neumann Architecture with suitable diagram.

Ans: **(diagram – 2 marks, any two points – 1 mark each)**

Sr.No	Von Neumann architecture	Harvard architecture
1		
2	The Van Neumann architecture uses single memory for their instructions and data.	The Harvard architecture uses physically separate memories for their instructions and data.
3	Requires single bus for instructions and data	Requires separate & dedicated buses for memories for instructions and data.
4	Its design is simpler	Its design is complicated
5	Instructions and data have to be fetched in sequential order limiting the operation bandwidth.	Instructions and data can be fetched simultaneously as there is separate buses for instruction and data which increasing operation bandwidth.
6	Program segments & memory blocks for data & stacks have separate sets of addresses.	Vectors & pointers, variables program segments & memory blocks for data & stacks have different addresses in the program.

b) List important any eight features of 8051 microcontroller.

Ans: **(any correct eight points – ½ mark each)**

Features of 8051 micro controller are as follows:-

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 external hardware interrupts- INT0 and INT1, Serial communication interrupt for both receive and transmit.

c) Explain the following 8051 microcontroller instructions:

(i) XCH A, @Ri

(ii) CJNE A, direct, rel

Ans: **(each correct instruction explanation – 2 marks each)**

(i) XCH A, @Ri

Exchange the contents of the accumulator with the contents of the memory location pointed by the 8 bit register in the instruction.

Example: XCH A, @R1

Exchange the contents of the accumulator with the contents of the memory location pointed by the R1 register.

(ii) CJNE A, direct, rel

Compare the contents of the accumulator with the 8 bit data directly 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 04H mentioned in the instruction and if they are not equal then jump to the line of instruction where UP label is mentioned.

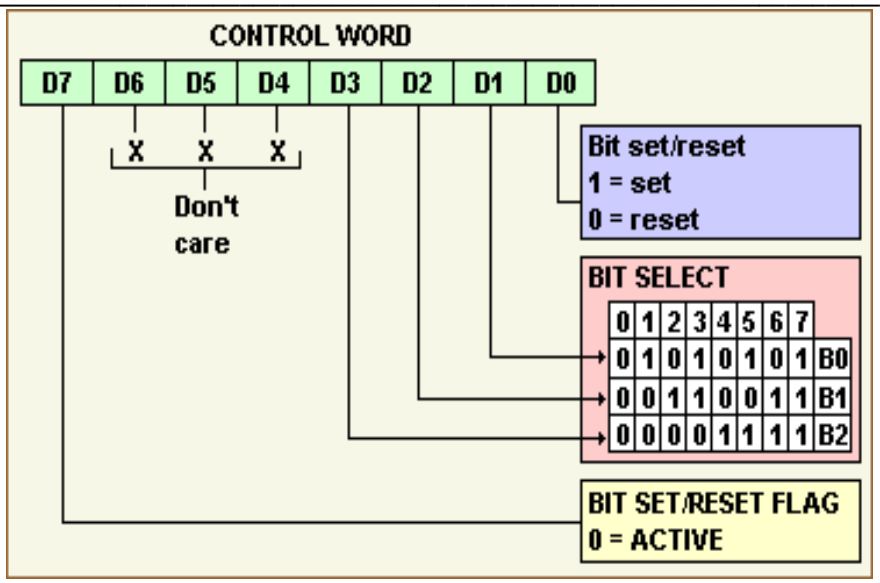
d) State any two differences between microcontroller and microprocessor.

Ans: **(any two correct differences – 2 mark each)**

Sr. No	Parameters	Microprocessor	Microcontroller
1	Number of Instructions	More (CISC)	Less (RISC)
2	Internal Memory	Absent	Present
3	Internal Timer	Absent	Present
4	Internal Timer	Absent	Present
5	I/O Ports	Not available	Available

e) Explain BSR mode of 8255. Write the control word in BSR mode to set, reset PC4 bit of Port C.

Ans: **(BSR format & explanation – 2 marks, control word for set/reset of PC4 – 1 mark each)**



The Control word to set bit PC4 is

D7 D6 D5 D4 D3 D2 D1 D0

0 0 0 0 1 0 0 1 = 09 H

The Control word to reset bit PC4 is

D7 D6 D5 D4 D3 D2 D1 D0

0 0 0 0 1 0 0 0 = 08 H

Q. 1 b) Attempt any ONE of the following **6 marks**

a) Write an assembly language program to exchange ten bytes of data from source location 40H to destination location 60H, for 8051 microcontroller.

NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

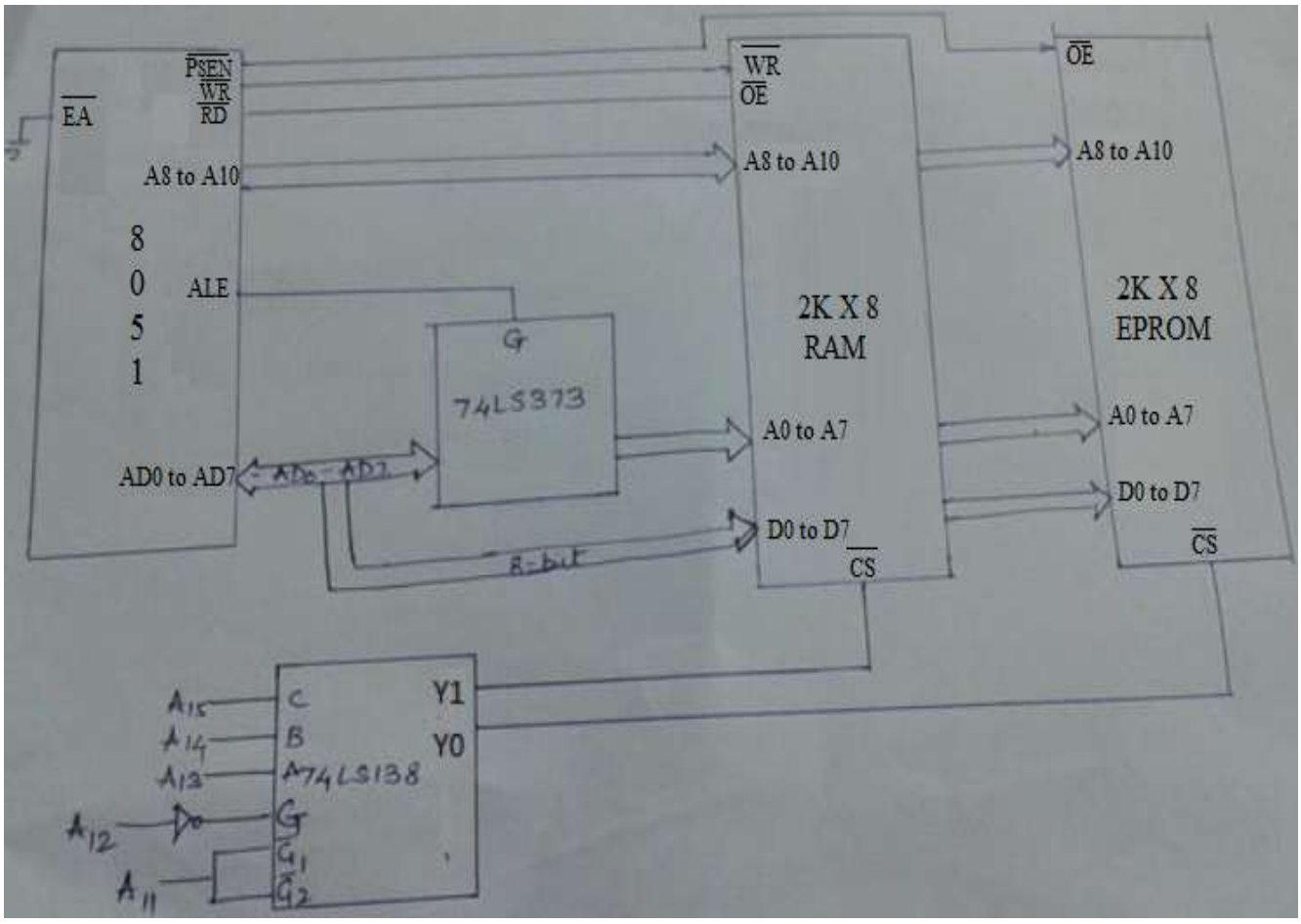
Ans: **(any correct logic of program – 6 marks)**

```
ORG 0000H          ; Program from 0000H
MOV R3, #10        ; Initialize Byte counter
MOV R0, #40H       ; Initialize memory pointer for source array
MOV R1, #60H       ; Initialize memory pointer for destination array
                   ; Therefore R0 ---> Source pointer R1 ---> destination pointer
UP: MOV A, @R0     ; Read data from SOURCE array
MOV B, @R1         ; Read data from DESTINATION array
MOV @R1, A         ; Write data of source array into destination array
MOV @R0, B         ; Write data of destination array into source 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
```

b) Draw the interfacing of 2 Kbyte EPROM and 2 Kbyte RAM to 8051 microcontroller. Draw the memory map.

Ans: **(Memory mapping – 2 marks, interfacing diagram – 4 marks)**

	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 ROM	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	**2000H**
End addr of ROM	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	**27FFH**



Q. 2 Attempt any FOUR of the following **16 marks**

a) Describe the function of address, data and control bus.

Ans:

(1) Address Bus (1 mark)

- The address bus is unidirectional over which the microcontroller sends an address code to the memory or input/output. The size of the address bus is specified by the number of bits it can handle.
- The more bits there are in the address bus, the more memory locations a microcontroller can access. A 16-bit address bus is capable of addressing (64k) addresses.

(2) Data Bus (1 mark)

- The data bus is bi—directional on which data or instruction codes are transferred into the microcontroller or on which the result of on operation or computation is sent out from the microcontroller to the memory or input/output.
- Depending on the particular microcontroller, the data bus can handle 8-bit or 16-bit data.

(3) Control Bus: **(2 marks)**

- The control bus is used by the microcontroller to send out or receive timing and control signals like read and write in order to co- ordinate and regulate its operation and to communicate with other devices i.e. memory or input/output.

b) Draw the format of PSW register of 8051 microcontroller and state the function of each bit.

Ans: **(format – 1 mark, explanation of each bit – 3 marks)**

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: the carry flag.

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

2. AC: the 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 n internal RAM in the table. By writing zeroes and ones to these bits, a group of registers R0- R7 can be used out of four registers banks in internal RAM.

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: the overflow flag

- This flag is set whenever the result of a signed number operation is too large, causing the high-order bit to overflow into the sign bit. In general, the carry flag is used to detect errors in unsigned arithmetic operations. 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 1's, then P=1, P=0 if A has an even number of 1's.

c) Describe the function of following pins of 8051 microcontroller.

(i) PSEN (iii) RST

(ii) EA (iv) ALE

Ans: **(each correct function – 1 mark each)**

Function of PSEN:

1. PSEN stands for "program store enable." The read strobe for external Program Memory is the signal PSEN (Program Store Enable). In an 8031-based system in which an external ROM holds the program code, this pin is connected to the OE pin of the ROM.
2. In other words, to access external ROM containing program code, the 8031/51 uses the PSEN signal. This read strobe is used for all external program fetches. PSEN is not activated for internal fetches.

Function of EA:

1. EA which stands for external access is pin number 31 in the DIP packages. It is an input pin and must be connected to either Vcc or GND. In other words, it cannot be left unconnected.
2. The lowest 4K (or SK or 16K) bytes of Program Memory can be either in the on-chip ROM or in an external ROM. This selection is made by strapping the EA (External Access) pin to either VCC or Vss.
3. In the 4K byte ROM devices, if the pin is strapped to Vcc, then program fetches to addresses 0000H through OFFFH are directed to the internal ROM. Program fetches to addresses 1000H through FFFFH are directed to external ROM.
4. If the pin is strapped to Vss, then all program fetches are directed to external ROM. The ROMless parts must have this pin externally strapped to VSS to enable them to execute properly.

Function of RESET:

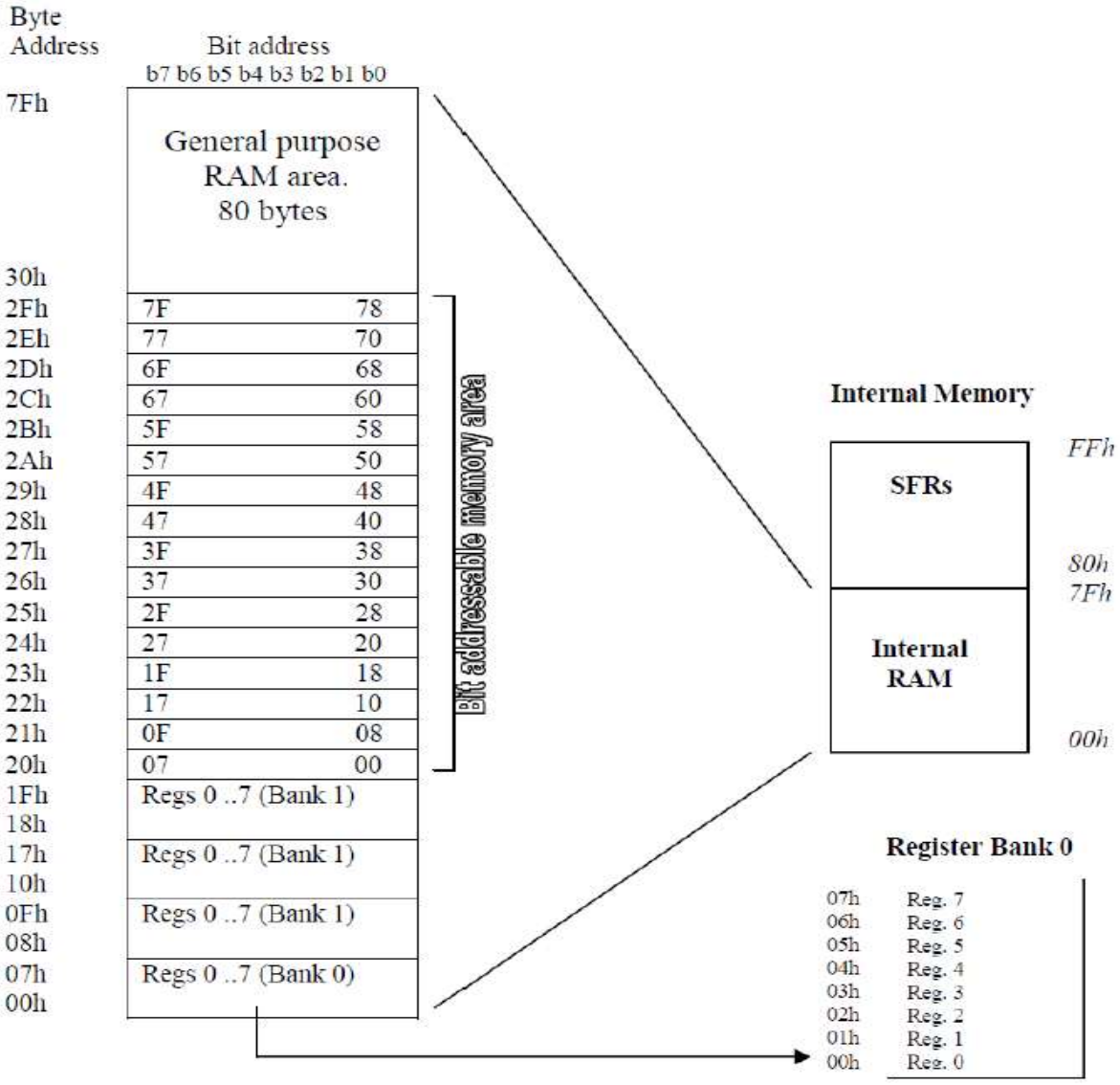
1. Pin 9 is the RESET pin. It is an input and is active high (normally low). Upon applying a high pulse to this pin, the microcontroller will reset and terminate all activities.
2. This is often referred to as a power-on reset. Activating a power-on reset will cause all values in the registers to be lost. It will set program counter to all 0s.
3. In order for the RESET input to be effective, it must have a minimum duration of two machine cycles. In other words, the high pulse must be high for a minimum of two machine cycles before it is allowed to go low.

Function of ALE:

1. ALE stands for address latch enable. It is an output pin and is active high for latching the low byte of address during accesses to external memory.
2. The ALE pin is used for demultiplexing the address and data by connecting to the G pin of the 74LS373 chip.

d) Draw Internal RAM memory organization of 8051 and explain.

Ans: **(diagram – 2 marks, explanation – 2 marks)**



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 in 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 you to set or reset any single bit in this section of RAM. With the general purpose RAM from 30H to 7FH and the register banks from 00H to 1FH, you may only read or write a full byte (8 bits) at these locations. However, with bit-addressable RAM (20H to 2FH) you can read or write any single bit in this region by using the unique address for that bit.

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.

e) List addressing modes of 8051 microcontroller. Explain any four with one example each.

Ans: **(list of modes – 2 marks, any four modes explanation – 2 marks)**

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

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

1) **Immediate Addressing mode:**

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

Move 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 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.

f) Explain the following directives with example:

(i) ORG (iii) EQU

(ii) DB (iv) CODE

Ans: **(each correct directive explanation – 1 mark)**

ORG: - ORG stands for Origin

Syntax:

ORG	Address

The ORG directive is used to indicate the beginning of the address. The number that comes after ORG can be either in hex or in decimal. If the number is not followed by H, it is decimal and the assembler will convert it to hex. Some assemblers use ―.ORG‖ (notice the dot) instead of ―ORG‖ for the origin directive.

DB: - (Data Byte)

Syntax:

Label:	DB	Byte

The byte is an 8-bit number represented in either binary, Hex, decimal or ASCII form. There should be at least one space between label & DB. The colon (:) must present after label. This directive can be used at the beginning of program. The label will be used in program instead of actual byte. There should be at least one space between DB & a byte. Following are some DB examples:

```
          ORG  500H
DATA1:    DB   28                ;DECIMAL(1C in hex)
DATA2:    DB   00110101B         ;BINARY (35 in hex)
DATA3:    DB   39H               ;HEX
          ORG  510H
DATA4:    DB   "2591"            ;ASCII NUMBERS
          ORG  518H
DATA6:    DB   "My name is Joe"  ;ASCII CHARACTERS
```

EQU: - Equate

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

Syntax:

Name	EQU	Constant

By means of this directive, a numeric value is replaced by a symbol.

For e.g. MAXIMUM EQU 99 After this directive every appearance of the label ―MAXIMUM‖ in the program, the assembler will interpret as number 99 (MAXIMUM=99).

CODE: -

Using code directive, an address in the code memory is assigned as a symbol.

As the maximum size of the code memory that can be accessed by 8051 is 64 Kbyte, the address should be within the range of 0 to 65535.

For example: START CODE 0 ; Memory location 0000H called START

Q3. Attempt any Four of the following:.

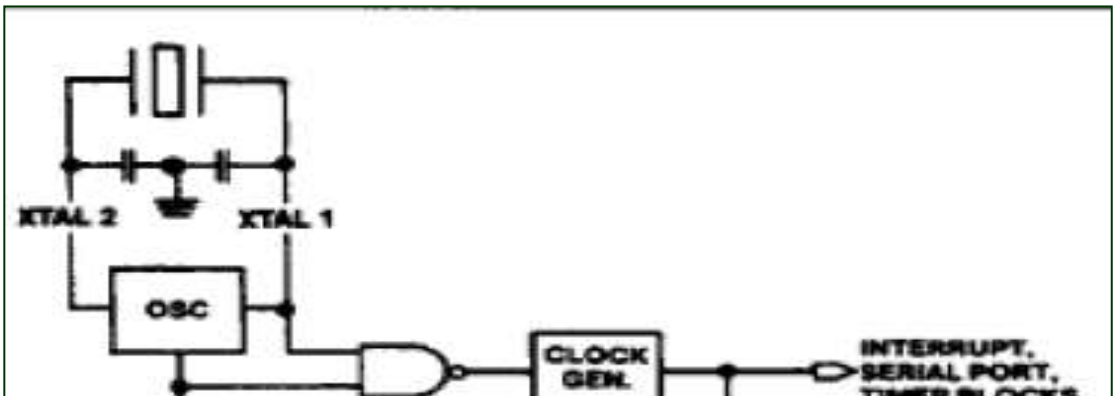
a. Distinguish between Microprocessor and Microcontroller on the basis of following points:

i. Architecture used
ii. Memory organization
iii. Ports
iv. Clock frequency.

Ans: (1M-each point)

Parameters	**Microprocessor**	**Microcontroller**
i. Architecture used	Von-nuemann Architecture is used in microprocessors.	Harvard Architecture is used in Microcontroller,
ii. Memory organization	It does not have inbuilt memory(RAM and ROM). And also it uses same memory for data and program.	It has inbuilt memory (RAM and ROM).It uses separate memory for data and program.
iii. Ports	It does not have I/O ports .	It has inbuilt I/O orts.
iv. Clock frequency.	It works on lower frequency as compared to microcontroller ie. 3MHz-6MHz.	It usually works on higher frequency ie.12MHz-26MHz.(8051 osc freq - 11.0592MHz)

b. Explain power saving options with diagram.
Ans: (1M- diagram ,2M-explanation,1M-PCON format)



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, 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.0 to be cleared and idle mode is terminated.

ii) Hardware reset: that is signal at RST pin clears IDLE 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.

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 function of Editor, Assembler, Compiler and Linker.
Ans: (1M- each)

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

2) Assembler: An assembler is programs that translate assembly language program to the correct binary code for each instruction i.e. machine code and generate the file called as Object file with extension .obj and list file with extension .lst extension.

Some examples of assembler are ASEM-51, Kiel's A51, AX 51 and C51, Intel PL/M-51 etc.

3) Compiler: Instructions in assembly language are represented in the form of meaningful abbreviations, and the process of their compiling into executable code is left over to a special program on a PC called compiler.

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. Describe selection factors of microcontroller.
Ans : (1M-each factor)

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, 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.

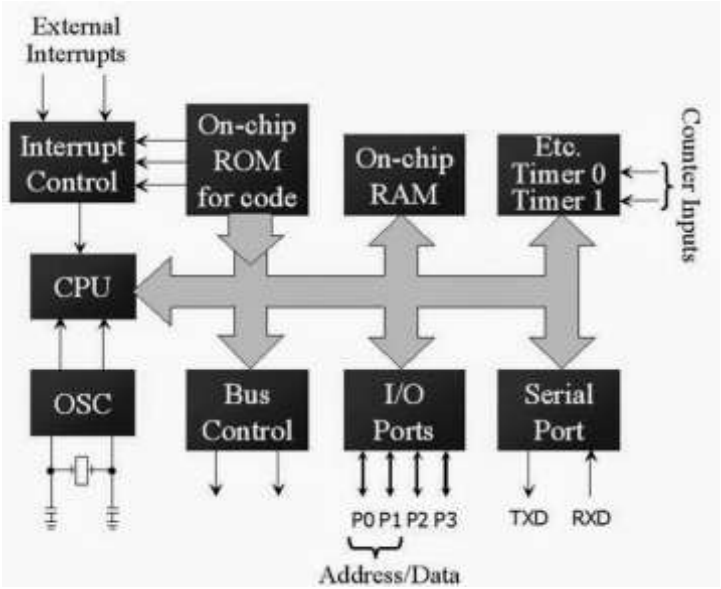
RISC improves speed of the system for the particular applications.

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.

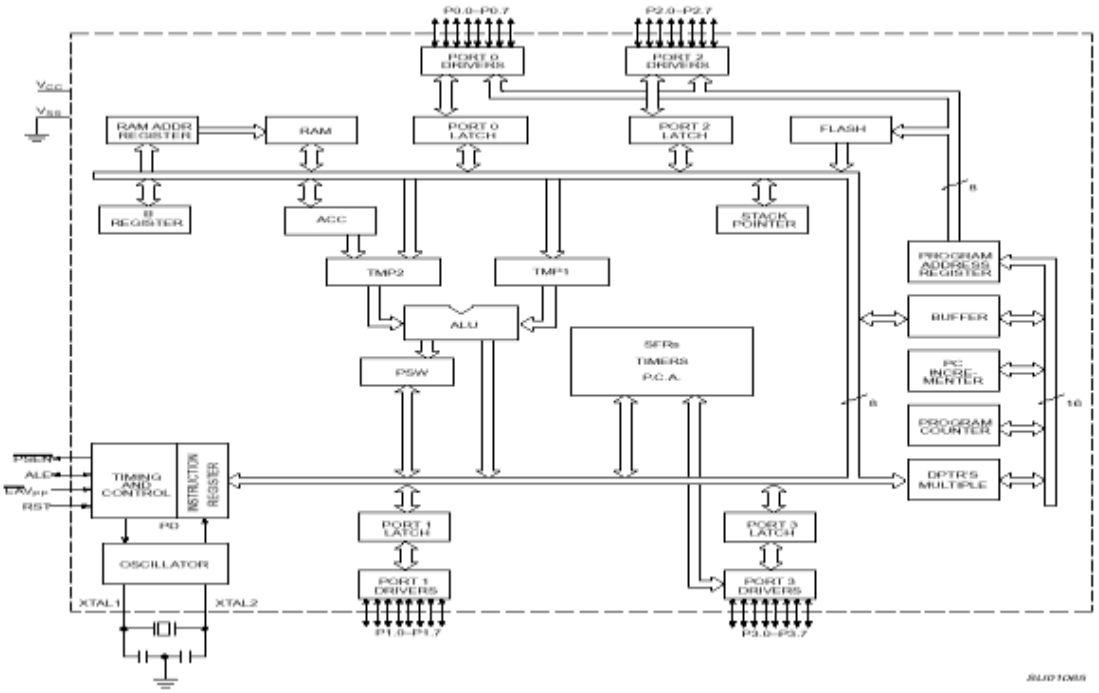
6. I/O capabilities: The number of I/O ports, size and characteristics of each I/O port, speed of operation of the I/O port, serial port or parallel ports. These are the considerations needed to ascertain.

e. Draw Architecture of 8051 microcontroller.

Ans: (4M-diagram)



OR



Q.4 (A) Attempt any THREE of the following:

a. With the help of ANL instructution explain :

i. Direct addressing mode

ii. Indirect addressing mode

iii. Register addressing mode

iv .Immediate addressing mode

Ans: (1M- each)

1) Direct Addressing mode
ANL A, add (8-bit Address)
A ←A ∧ (add)
Eg. ANL A, 12H
The contents of memory location specified by 8 bit direct address will be logically anded bit by bit with the contents of accumulator and result is stored in accumulator.

2) Indirect addressing mode
ANL A, @Ri
A ←A ∧ (Ri)
Eg. ANL A,@R0
The content of memory location whose address is specified by Ri (R0/R1) will be logically anded bit by bit with contents of accumulator. Result is stored in accumulator. Only R0 or R1 can be used.

3) Register addressing mode

ANL A, Rn

A ←A ∧ Rn

Eg. ANL A, R2

The contents of specified register Rn (R0-R7) will be logically anded bit by bit with the contents of accumulator and result is stored in Accumulator.

4) Immediate addressing mode

ANL A, #data(8-bit)

A ←A ∧ #data

ANL A, #23H

Immediate data is logically anded bit by bit with contents of accumulator and result is stored in accumulator.

b. Draw the format of SCON register. Explain any two modes of serial communication.

Ans: (2M- format , 1M- each mode explanation)

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.

RI SCON.0 Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1.Must be cleared by software.

8051 micro controller communicate with another peripheral device through RXD and TXD pin of port3.Controller have four mode of serial communication.

NOTE: (Any two modes)

1. Serial Data Mode-0 (Baud Rate Fixed)

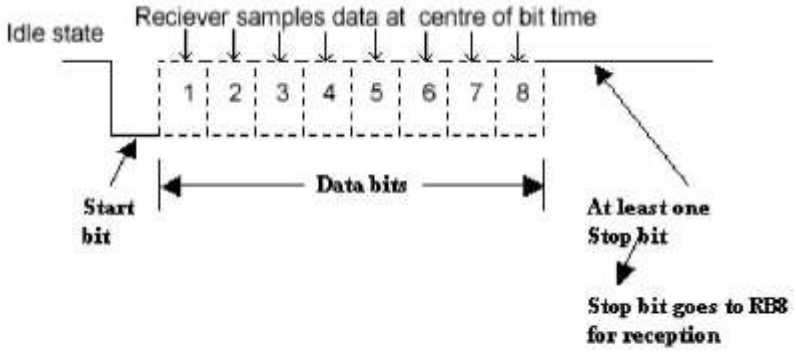
In this mode, the serial port works like a shift register and the data transmission works synchronously with a clock frequency of f_{osc} /12. Serial data is received and transmitted through RXD. 8 bits are transmitted/ received aty a time. Pin TXD outputs the shift clock pulses of frequency f_{osc} /12, which is connected to the external circuitry for synchronization.The shift frequency or baud rate is always 1/12 of the oscillator frequency.

2. Serial Data Mode-1 (standard UART mode)(baud rate is variable)

In mode-1, the serial port functions as a standard Universal Asynchronous Receiver

Transmitter

(UART) mode. 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**.



3. Seri

In th RXD.The vario first), a programmable 9 (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. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

$$f_{baud} = \frac{2^{SMOD}}{32} \times \frac{fosc}{12 \times [256-(TH1)]}$$

$f_{baud} = (2^{SMOD}/64) f_{osc}$

4. Serial Data Mode-3 - Multi processor mode(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 9 th 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). **$f_{baud} = (2^{SMOD}/32) * (f_{osc} / 12 (256-TH1))$**

c. Write an Assembly Langauge Program to transfer the message "MIC" serially at 4800 baud, 8 bit data , 1 stop bit. Do this continuously.

Ans: (3M- program, 1M-comments)

```
        ; Initialization
        MOV TMOD, #20H          ; timer1, mode2
        MOV TH1, #-6            ; 4800 baud
        MOV SCON, #50H          ; 8 bit data , 1 stop bit
        ; Begin to transmit
        SETB TR1                ; start Timer 1
AGAIN:  MOV A, # 'M'            ; transfer 'M'
        CALL TRANS
        MOV A, #'I'             ; transfer 'I'
        CALL TRANS
        MOV A, #'C'             ; transfer 'C'
        CALL TRANS
```

```
        SJMP   AGAIN                ; keep doing it
        ; Serial transmitting subroutine
TRANS:  MOV SBUF, A                 ; load SBUF
HERE:   JNB TI, HERE                ; wait for the last bit to transfer
        CLR TI                      ; get ready for the next byte
        RET
```

NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

d. Write an assembly language program to find two's complement of a number 55H, and store the result in the memory location 3000H.

Ans: (3M-prog, 1M-comments)

```
MOV DPTR, #3000H        ; initialize pointer to store result
MOV A, #55H             ; load number in Acc
CPL  A                  ; 1's complement of no.
ADD A, #01H             ; 2's complement of no.
MOV @DPTR, A            ; store the result in ML 3000H
SJMP $
```

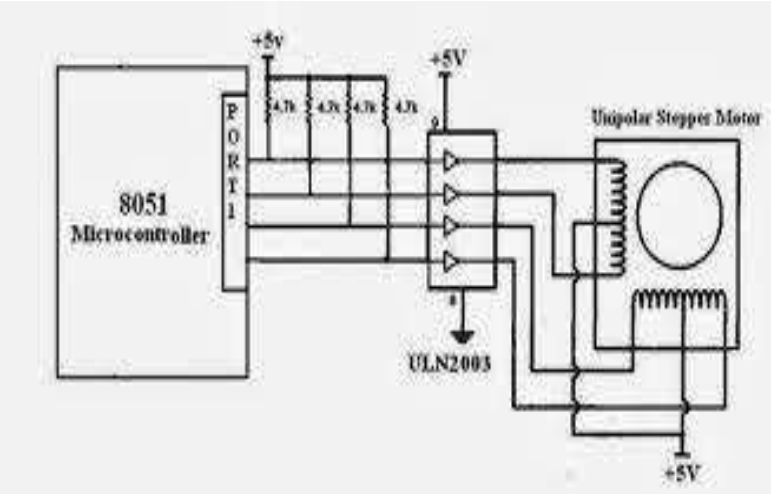
NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

B. Attempt any ONE of the following:

a. Draw the interfacing diagram of stepper motor with 8051 microcontroller and write assembly language program to rotate stepper motor in counter clockwise.

Ans: (1M-calculation,1M-diagram,3M-prog,1M-comments)

NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)



Step no	Winding A	Winding B	Winding C	Winding D	anticlockwise
1	1	0	0	1	
2	1	1	0	0	
3	0	1	1	0	
4	0	0	1	1	

```
PROGRAM:
        MOV A, #66H              ; load step sequence
BACK:MOV P1, A                   ; issue sequence to motor
AGAIN:RL A                       ; rotate left anti clockwise
        ACALL DELAY              ; wait
        SJMP BACK                ; keep going

DELAY
        MOV R2, #100
 H1:    MOV R3, #255
 H2:    DJNZ R3, H2
        DJNZ R2, H1
        RET
```

Or

Calculation for no. of steps:
Assume step angle of 1.8

Count = (Required Rotaion Angle)
1.8 degree x no. of steps
=180/1.8*4 steps=100/4=25

PROGRAM:

```
PROGRAM:
        MOV A, #66H              ; load step sequence
        MOV R0, # 25             ; count for 4 steps
BACK:   MOV P1, A                ; issue sequence to motor
AGAIN:  RL A                     ; rotate left anti clockwise
        ACALL DELAY              ; wait
        DJNZ R0, AGAIN           ; check for steps
```

```
        SJMP BACK                ; keep going
```

Delay subroutine.

```
DELAY
        MOV R2, #100
H1:     MOV R3, #255
H2:     DJNZ R3, H2
        DJNZ R2, H1
        RET
```

b. Write an assembly language program to arrange ten numbers in an ascending order.

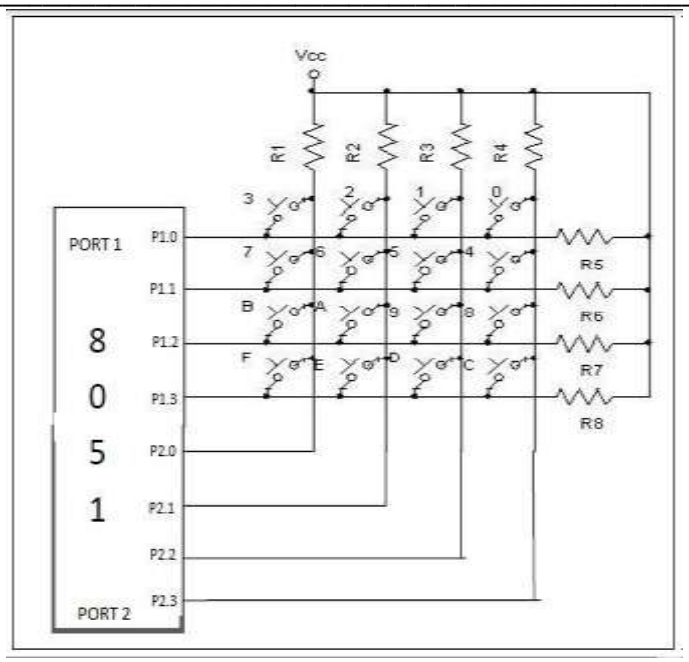
Ans: (4M-prog)

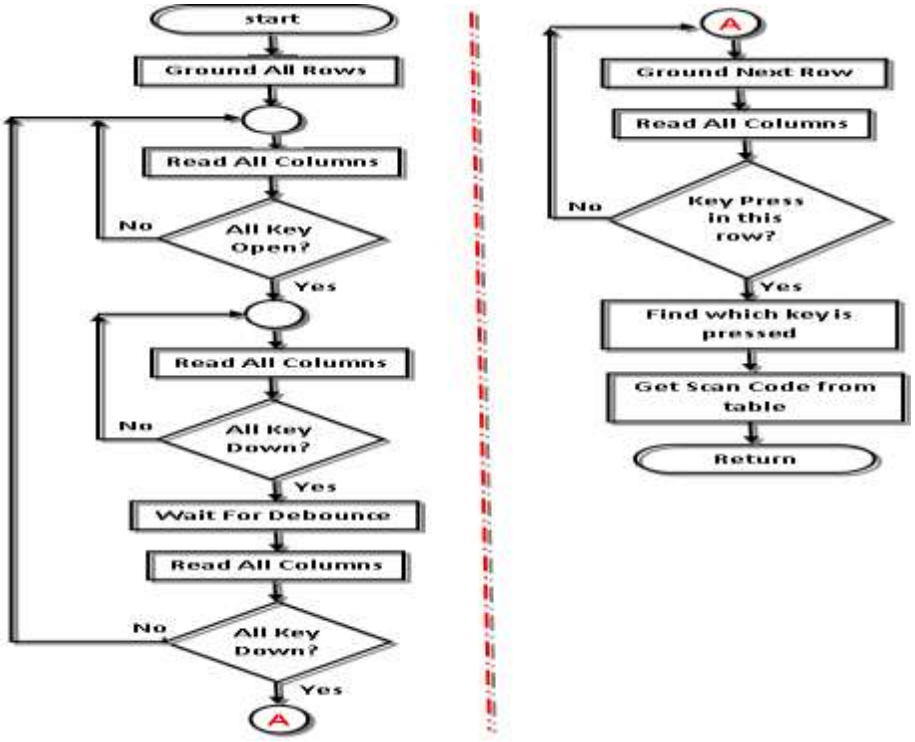
NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

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

c. Draw interfacing diagram showing 4x4 matrix keyboard connections to Port 2 and Port 1 of 8051 microcontroller. Draw flow chart to detect a pressed key.

Ans: (2M-diagram, 2M-flowchart)


Vcc
R1
R2
R3
R4
R5
R6
R7
R8
PORT 1
P1.0
P1.1
P1.2
P1.3
P2.0
P2.1
P2.2
P2.3
PORT 2
8051


start
Ground All Rows
Read All Columns
No
All Key Open?
Yes
Read All Columns
No
All Key Down?
Yes
Wait For Debounce
Read All Columns
No
All Key Down?
Yes
A
A
Ground Next Row
Read All Columns
No
Key Press in this row?
Yes
Find which key is pressed
Get Scan Code from table
Return

Q.5 Attempt any FOUR of the following:

a) List the various interrupts in 8051 microcontroller along with their priorities and vector locations.

Ans:(vector locations-2M,Priority-2M)

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

b) A switch is connected to pin P1.0 and LED to pin P2.7.Write a program to get the status of the switch and send it to the LED.

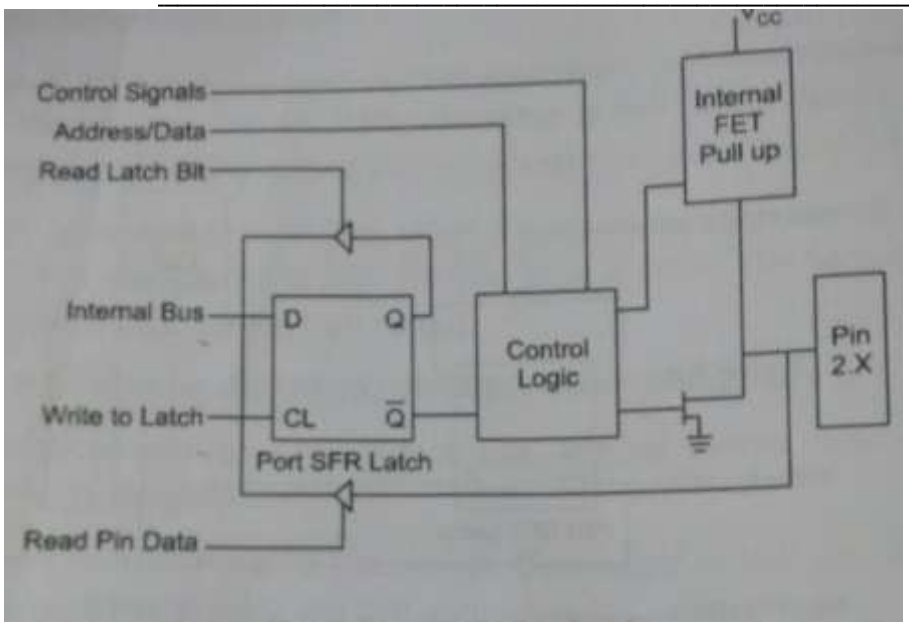
Ans: (Any correct program-3M,comments-1M)

NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

```
NEXT: MOV A,P1        ;Read key status from P1
ANL A,#01H            ;check P1.0 bit of port 1
CJNEA,#01H,DN         ;If not pressed then NEXT
MOV P2,A              ;Make LED ON
ACALL Delay           ; Wait for sometime
SJMP NEXT
DN: MOV P2,A          ;Make LED OFF
ACALL Delay           ;Wait for sometime
SJMP NEXT
```

c) Draw the circuit diagram of port 2 of 8051 and describe its function.

Ans: (Circuit diagram-3M,Function-1M)



Port 2: It can be used as
a) Simple input/output port
b) the alternative use is to supply a higher order address byte in conjunction with the port 0 lower order byte to address external memory.

d) Draw the format of TCON register and state the function of each bit.

Ans :(format : 2 Marks Explanation : 2 Marks)

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

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

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

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

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

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

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

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

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

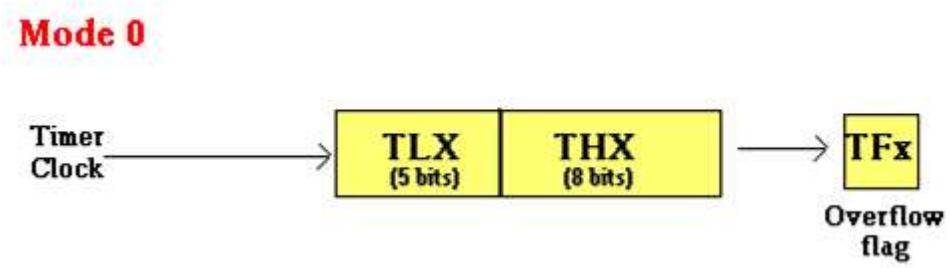
e) List the timer modes of 8051 microcontroller. Describe any two timer modes with a suitable diagram.

Ans :(List-1M,Description of any two timers-3M)

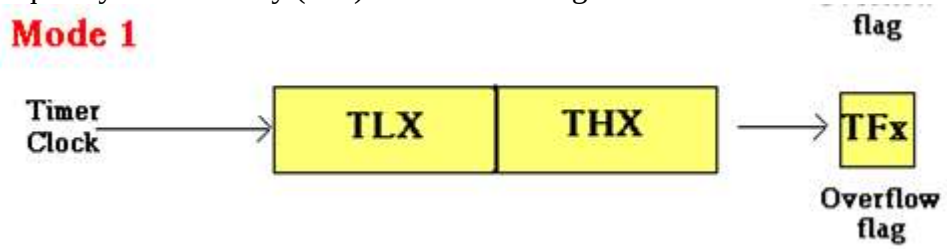
M1	M0	MODE	DESCRIPTION
0	0	0	13-bit timer
0	1	1	16-bit timer
1	0	2	8-bit auto-reload
1	1	3	Split mode

Operating modes of Timer: The timer may operate in any of the four modes that are determined byM1 and M0 bit in TMOD register.

Mode 0:

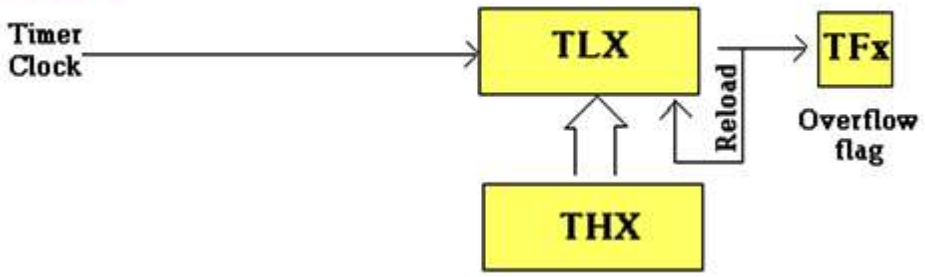


In mode0 the register THX is used as 8 bit counter and TLX is used as
5 bit counter. The pulse i/p is divided by (32)10so that TH counts. Hence original oscillator frequency is divided by (384)10.The timer flag is set when THX rolls over from FF to 00H.



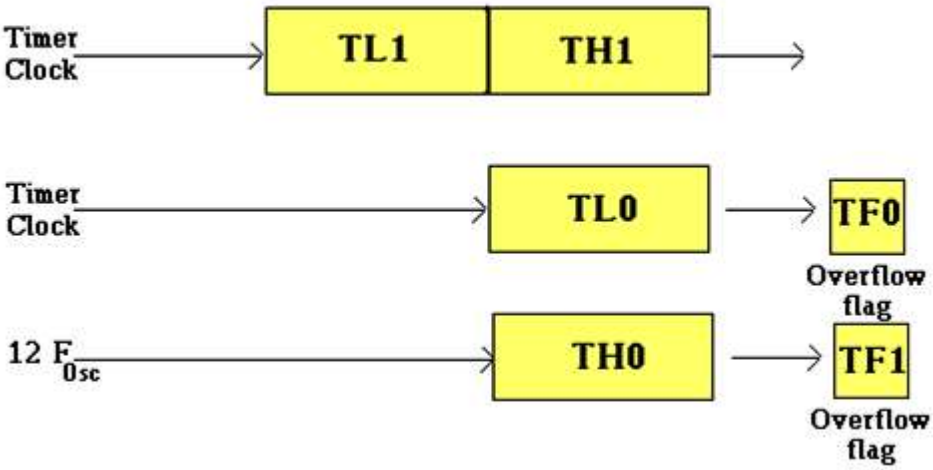
It is similar to Mode 0 except TLX is configured as a full 8-bit counter. Hence pulse input is divided by 25610 so that TH counts the timer flag is set when THX rolls over from FF to 00H.

Mode 2



In this mode only TLX is used as 8-bit counter. THX is used to hold the value which is loaded in TLX initially. Everytime TLX overflows from FFH to 00H the timer flag is set and the value from THX is automatically reloaded in TLX register.

Mode 3



In this mode, timer 0 becomes two completed separate 8-bit timers. TL0 is controlled by gate arrangement of timer 0 and sets timer 0 flag when it overflows. TH0 receives the timer clock under the control of TR1 bit and sets TF1 flag when it overflows. Timer 1 may be used in mode 0, 1 and 2 with one important exception that no interrupt will be generated by the timer when the timer 0 is using TF1 overflow flag.

Q.6 Attempt any FOUR of the following:

a) Write a program to generate a square wave of 50Hz frequency on pin P1.2,using an interrupt for Timer 0.Assume that XTAL=11.0592MHz.

Ans: (1M- calculation, 2M- Program,1M- comments)

NOTE: Program may change. Student can also use the other logic.
Please check the logic and understanding of students.)

Look at the following steps for 50Hz frequency calculations with 11.0592 MHz.
The period of the square wave = 1 / 50 Hz

= 20 ms.
The high or low portion of the square wave = Time period / 2

= 20 ms / 2 =10 ms =10,00uSec.
Timer clock Frequency is = XTAL / 12

= 11.0592MHz / 12 = 921.6 KHz
Timer clock period is = 1/ Timer Frequency

= 1 / 921.6 KHz
= 1.085 uSec
Counter = Delay / timer clock period

=10,000 uSec / 1.085 uSec =
Timer Reload value = Maximum Count – Counter

= 65536 – 9217 = (56319)d
Timer Reload value in HEX = (56319)d

= (DBFF) h .
TL1 = FF h and TH1 =DB h.

Assembly program to generate square wave over Port Pin P1.2 using timer 0

```
ORG 0000H ;Start the program

REPEAT: CPL P1.2 ;compliment the P1.4

 LCALL DELAY ;delay of 20msec for 50Hz

 SJMP REPEAT ;repeat the process

 DELAY: MOV TMOD ,#01H ;timer 0, mode 1

MOV TL1, #FFH ;Timer value = DBFFH

 MOV TH1 ,#DBH

 SETB TR0 ;start the timer

 BACK: JNB TF0,BACK ;wait until TF0 sets

 CLR TR0 ;stop timer CLR TF0 ;clear timer flag 0

 RET
```

b) Describe the steps for programming the 8051 to receive data serially.

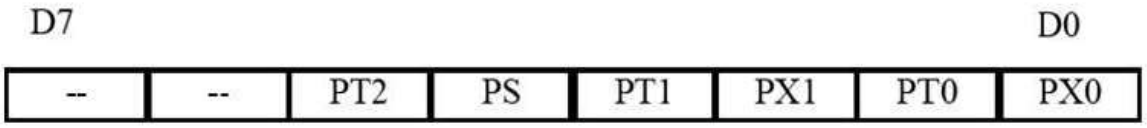
(Program steps to receive data serially - 4 Marks)

1. TMOD register is loaded with the value 20H
2. TH1 is loaded with value set the baud rate
3. SCON register is loaded with the value 50H
4. TR1 is set to 1 to start Timer 1
5. RI is cleared with the "CLR RI" instruction
6. RI flag bit is monitored (JNB RI)to see if an entire character has been received yet
7. RI=1 SBUF has the byte, its contents are moved into a safe place
8. To receive the next character, go to Step 5 .

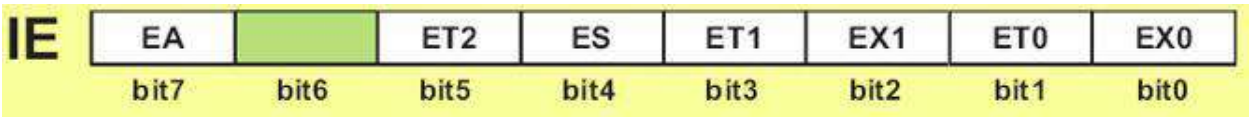
c) Draw the format of IE and IP register.

Ans : (Each format-2M)

Format of IP register

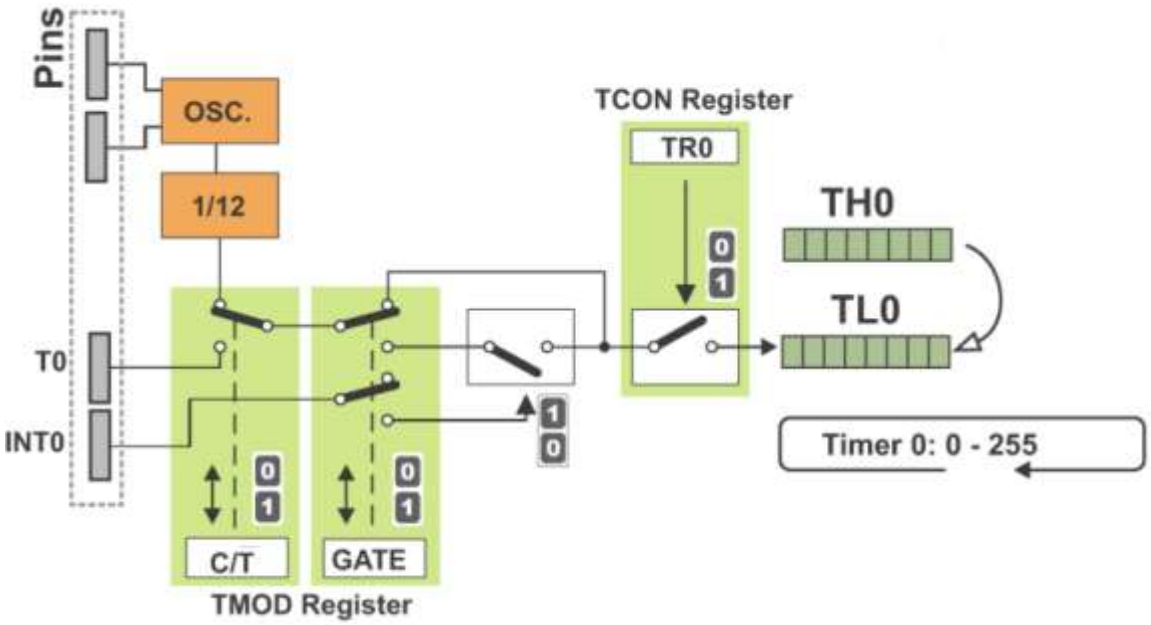


Format of IE register

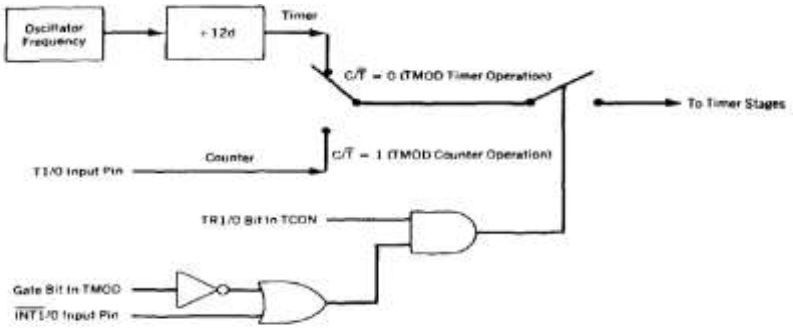


d) Explain the timer/counter logic with diagram.

Ans: (Diagram -2M,Explaination-2M)



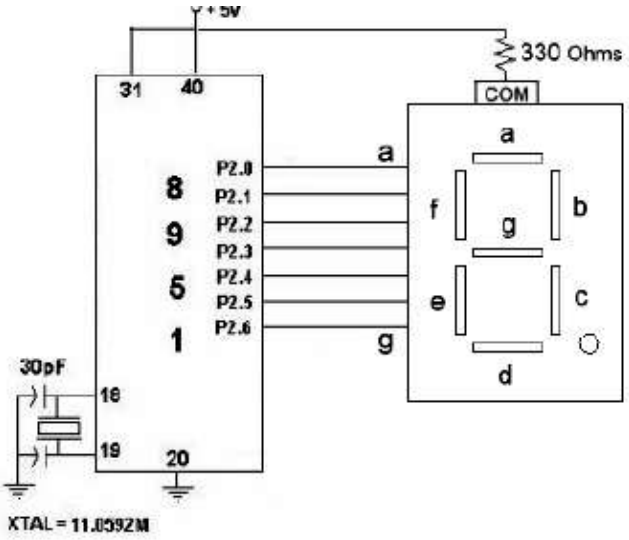
OR



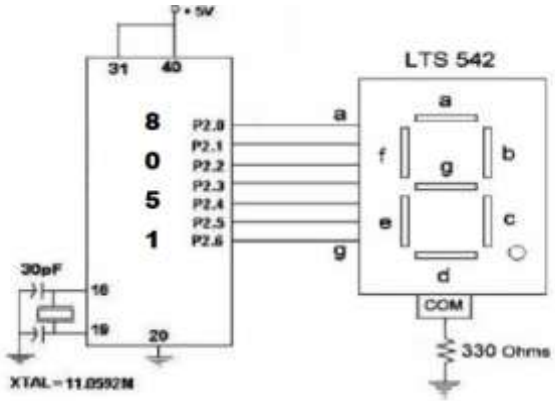
8051 has two 16-bit programmable UP timers/counters. They can be configured to operate either as timers or as event counters. The names of the two counters are T0 and T1 respectively. The timer/ content is available in four 8-bit special function registers, viz. TL0,TH0, TL1 and TH1 respectively. In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

e) Draw the interfacing diagram of seven segment display with 8051 microcontroller.

Ans**: (Any correct diagram: 4M))**

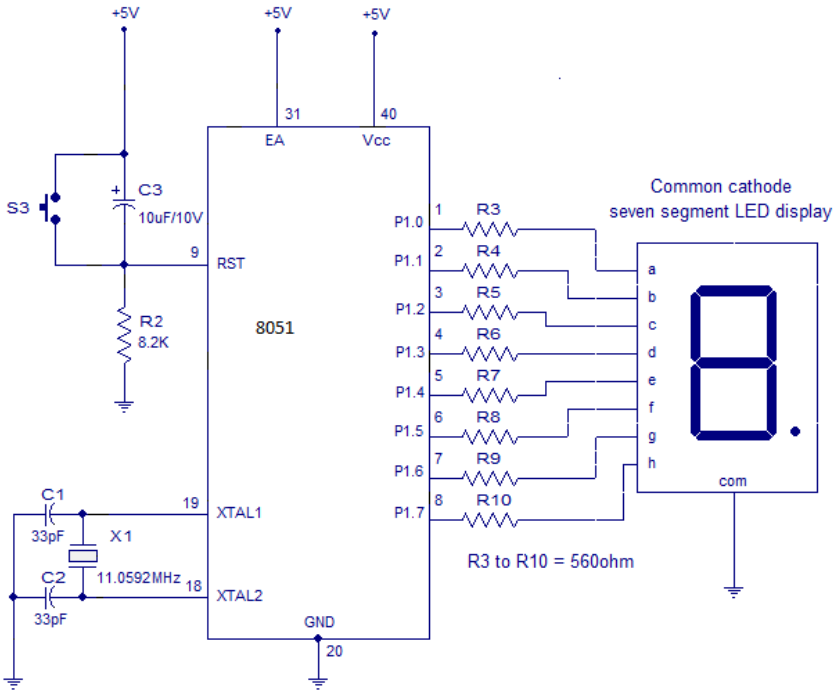


For Common anode display



For Common cathode display

(Any other relevant diagrams should also be considered correct.)


+5V
+5V
+5V
31
EA
40
Vcc
S3
C3
10uF/10V
9
RST
R2
8.2K
8051
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
1
R3
2
R4
3
R5
4
R6
5
R7
6
R8
7
R9
8
R10
Common cathode
seven segment LED display
a
b
c
d
e
f
g
h
com
C1
33pF
19
XTAL1
X1
C2
11.0592MHz
33pF
18
XTAL2
GND
20
R3 to R10 = 560ohm