T.Y. Diploma : Sem. V
[EJ/EN/ET/EX/DE/IS/IC/IE/EV/MU]

Microcontrollers

Time: 3 Hrs.] Prelim Question Paper Solution **[Marks : 100**

Q.1(a) Attempt any THREE of the following: [12]

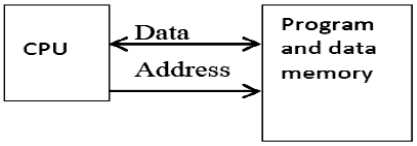
Q.1(a) (i) Distinguish between Microprocessor and Microcontroller (any four points). [4]

Ans.:

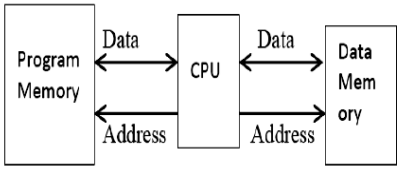
Sr. No.	Parameter	Microprocessor	Microcontroller
1.	Number of instruction used	Many instructions to read / write data to/ from external memory.	Few instruction to read/write data to/ from external memory
2.	Memory	Do not have inbuilt RAM or ROM.	Inbuilt RAM/or ROM
3.	Registers	Microprocessor contains general purpose registers, stack pointer register, program counter register,	Microcontroller contains general purpose registers, Stack pointer register, Program counter register additional to that it contains Special Function Registers (SFRs) for Timer, Interrupt and serial communication etc
4.	Timer	Do not have inbuilt Timer.	Inbuilt Timer
5.	T/O ports	I/O ports are not available requires extra device like 8155 or 8255.	I/O ports are available
6.	Serial port	Do not have inbuilt serial port, requires extra devices like 8250 or 8251.	Inbuilt serial port.
7.	Multifunction pins	Less Multifunction pins on IC.	Many multifunction pins on the IC
8.	Boolean Operation	Boolean operation is not possible directly.	Boolean operation i.e. operation on individual bit is possible directly.
9.	Applications	General purpose, Computers and personal Uses.	Automobile companies, embedded systems, remote control devices.

Q.1(a) (ii) Draw neat labelled block diagram of Von–neumann and Harvard architecture. [4]

Ans.: Von-Neumann Architecture

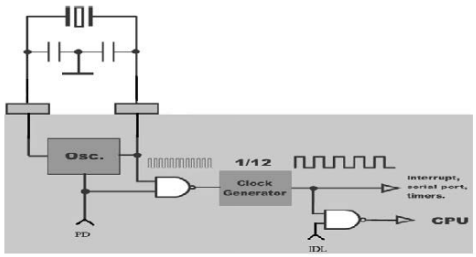


Harvard Architecture



Q.1(a) (iii) Describe power saving options of 8051 microcontroller **[4]**

Ans.:



Format of PCON:

PCON: POWER CONTROL REGISTER NOT BIT ADDRESSABLE

SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD Double band rate bit. If Timer 1 is used tc generate bund rate and SMOD = 1, the baud rate is dcuble when the Serial port is used in modes 1, 2, or3.

- Not implemented, reserved for inture use.*
- Not implemented, reserved for inture use.*
- Not implemented, reserved for inture use.*

GFI General purpose flag bit.

GFD General purpose flag bit.

PD Power Down bit.Seting this bit activates Power Down operation in the 80C51BH.

IDL Idle Mode bit. Seting this bit activates Icle Mode operation in the 80C51BH

Idlf Mode : In the Idlemode, 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

Q.1(a) (iv) State the function of : **[4]**

(1) Editor (2) Assembler (3) Compiler (4) Linker

Ans.: (1) Editor: An editor is 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.

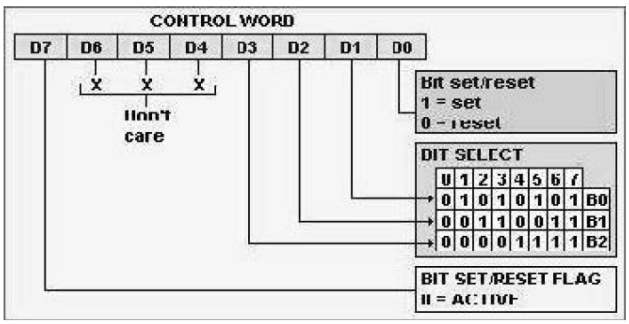
(ii) Assembler : An assembler is programs that translate assembly language program to the correct binary/hex code for each instruction i.e. machine code and generate the file called as Object file with extension .obj and list file with extension .lst extension.
Some examples of assembler are ASEM-51, Kiel's A51, AX 51 and C51, Intel PL/M-51 etc.

(iii) 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.

(iv) 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.

Q.1(a) (v) With control word register, explain Bit Set reset (BSR) mode of 8255 [4]

Ans.:



Explanation : The Bit Set/Reset (BSR) mode is available on port C only. Each line of port C (PC_7 - PC_0) can be set or reset by writing a suitable value to the control word register. BSR mode and I/O mode are independent and selection of BSR mode does not affect the operation of other ports in I/O mode.

- D_7 bit is always 0 for BSR mode.
- Bits D_6, D_5 and D_4 are don't care bits.
- Bits D_3, D_2 and D_1 are used to select the pin of Port C.

Selection of port C pin is determined as follows:

D3	D2	D1	Bit/pin of port C selected
0	0	0	PC_0
0	0	1	PC_1
0	1	0	PC_2
0	1	1	PC_3
1	0	0	PC_4
1	0	1	PC_5
1	1	0	PC_6
1	1	1	PC_7

- Bit D_0 is used to set/reset the selected pin of Port C.

As an example, if it is needed that PC_5 be set, then in the control word,
(i) Since it is BSR mode, D_7 = '0'.
(ii) Since D_4, D_5, D_6 are not used, assume them to be '0'.
(iii) PC_5 has to be selected, hence, D_3 = '1', D_2 = '0', D_1 = '1'.
(iv) PC_5 has to be set, hence, D0 = '1'.

Thus, as per the above values, 0B (Hex) will be loaded into the Control Word Register (CWR).

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	1	0	1	1

Q.1 (b) Attempt any ONE of the following : [6]

Q.1(b) (i) Write an assembly language program for 8051 microcontroller for finding the largest number in a given set of 05 number (Assume suitable data/memory addresses) [6]

Ans.: Program can be written using internal/external memory address. Any suitable address or register can be assumed

Program:

```
MOV DPTR, # 3000H ;      Initialize memory pointer
MOV R0,#00h              Assume R0=00h is largest no
MOV R1, #05              Initialize byte counter R0=5
UP: MOVX A, @DPTR        Load number in accumulator
CJNE A, R0, NEXT         Compare the no in A with Largest no in R0
NEXT: JC SKIP            If A<=R0 skip loading new no in R0
MOV R0, A                Store new larger number in R0
SKIP: INC DPTR           Increment memory pointer by 1
DJNZ R1, UP              Decrement byte counter and repeat if not 0
HERE: SJMP HERE          Stop the program, largest no is in r0
```

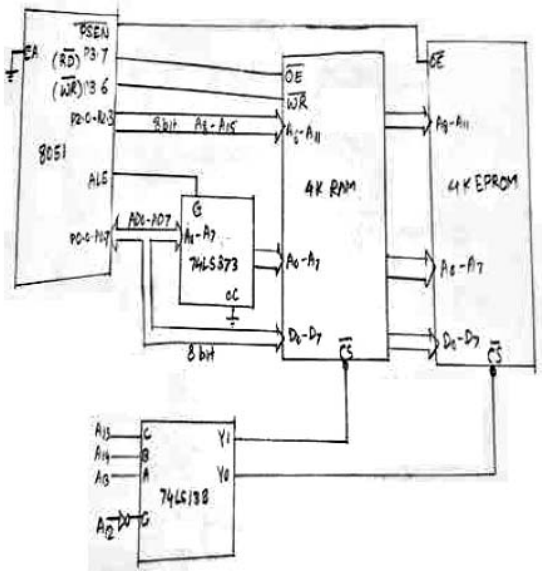
OR

```
MOV R0,#50h        ;Five nos are stored 59h onwards
MOV R1,#05         ;Counter for 5 nos
MOV R2,#00h        ;Assume 00 is the largest no
UP:   MOV A,@R0    ;Load the no in A
CJNE A, R2, NEXT   ;Compare the no in A and R2

NEXT:  JC SKIP       ; if A < = R2 skip loading new no the R2
       MOV R2, A     ; if A > R2 load no in A into R2
SKIP:  INC  R0       ; Increment pointer
       DJNZ R1, UP   ; Decrement counter and repeat the process till R1 = 0
HERE:  SJMP HERE     ; Stop here, the largest no is in R2
       END
```

Q.1(b) (ii) Draw a diagram to interface 4K byte EPROM and 4K Byte RAM to 8051 microcontroller. Draw the memory map. [6]

Ans.:



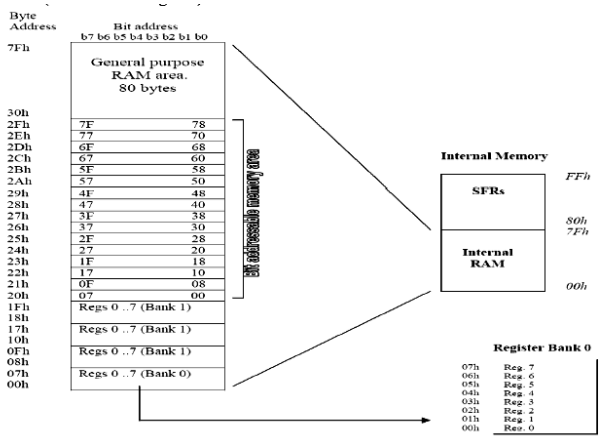
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	HEX ADDR
Start addr of EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
End addr of EPROM	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH
Start addr RAM	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H
End addr RAM	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	2FFFH

Starting address of RAM can be taken as 1000h and ending address 1FFFh but accordingly Decorder logic i.e. 74LS138 connections will change.

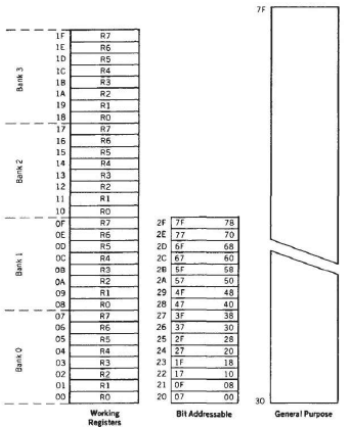
Q.2 Attempt any FOUR of the following : [16]

Q.2(a) Draw the internal RAM organization of 8051 with the address location. [4]

Ans.:



OR



Q.2(b) Compare 8031, 8051 and 8751 (four points) [4]

Ans.:

Specification	8031	8051	8751
On chip data memory	128 byte	128 byte	128 byte
On chip program memory	ROM less	4K ROM	4K EPROM
Number of 16 bit timer/counter	2	2	2
Number of vectored interrupts	5	5	5
Full duplex serial I/O	1	1	1
On chip peripherals	UART	UART	UART
No of I/o lines	32	32	32
Speed MHz	12	12	12

Q.2(c) Write an assembly language program to add two BCD numbers 66H and 95 H which are stored at external memory location 3000 H and 3001 H respectively. Store the result at memory location 3002 H. **[4]**

Ans.: Data:

3000H – 66H
3001H – 95H

```
MOV DPTR, #33000H      ; Initialize data pointer with external ML
MOVX A, @DPTR          ; load 1st no. in acc from the external ML
MOV R2, A              ; move the 1st no. in reg R2
INC DPTR               ; increment data pointer
MOVX A, @DPTR          ; load the 2nd no. in acc from external ML
ADD A, R2              ; add 1st no. with 2nd no.
DA A                   ; decimal adjust accumulator after addition
INC DPTR
MOV @DPTR, A           ; store result in internal memory
LOOP: AJMP LOOP        ; stop
```

Q.2(d) Draw the format of PSW register of 8051 μC and state the function of each flag. **[4]**

Ans.:

CY	AC	F0	RS1	RS0	OV	-	P

CY	PSW	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.

(i) CY: 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.

(ii) AC: Auxiliary carry flag

If there is a vary 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.

(iii) F0: Available to the user for general purposes.

(iv) 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	Bank 2 (10H-17H)
1	1	Bank 3 (18H-1FH)

(v) 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.

(vi) 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.

Q.2(e) The fit addressable feature in 8051 microcontroller makes it more powerful than microprocessor, justify your answer. **[4]**

Ans.: The 8051 instruction set is optimized for the one bit operations so often desired in real – world, real- time control applications.

(i) The Boolean processor is an internal part of 8051 architecture. It is an independent bit processor with its own accumulator and its own bit addressable RAM and I/O.

(ii) The Boolean processor provides direct support for bit manipulation .This leads to more efficient programs that needs to deal with binary input and output conditions inherent in digital control problems.

(iii) Bit addressing can be used for test pin monitoring or program control flags.

For example, instructions for Boolean function are as given below.

- SETB P1.0 ; Set P1.0
- JB P1.0,NEXT ; Jump to label if P1.0 is set
- ANL C, P1.4 ; AND the bit on P1.4 with carry.

Any other bit addressable instructions minimum 4 must be explained

Q.2(f) State and describe the alternate functions of port 3 pins of 8051. **[4]**

Ans.:

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

Q.3 Attempt any FOUR of the following : **[16]**

Q.3(a) Explain the following directives with example. **[4]**

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

Ans.: 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: (Define Byte)

Syntax: Label: DB Byte

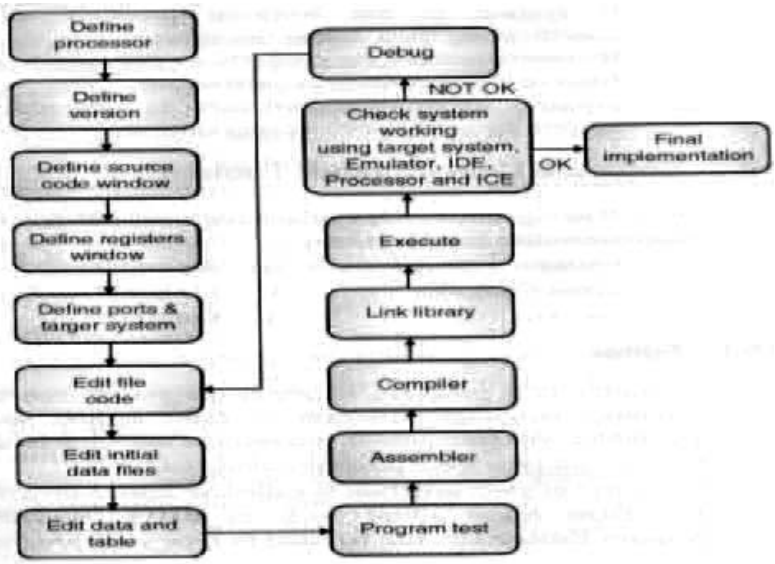
Where byte is an 8-bit number represented in either binary, Hex, decimal or ASCII form. There should be at least one space between label & DB. The colon (:) must present after label. This directive can be used at the beginning of program. The label will be used in program instead of actual byte. There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

EQU: Equate
It is used to define constant without occupying a memory location.
Syntax: Label EQU Numeric value
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).

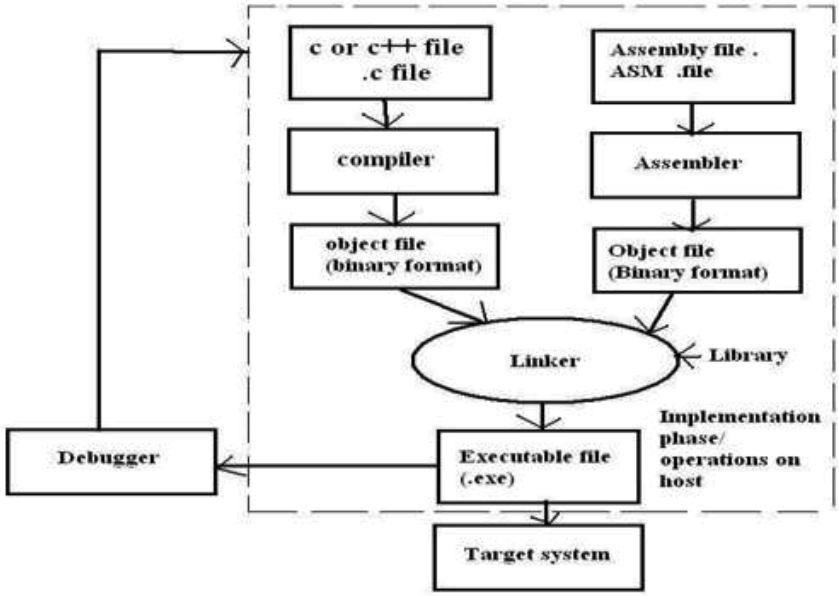
END:
This directive must be at the end of every program meaning that in the source code anything after the END directive is ignored by the assembler.
This indicates to the assembler the end of the source file.
Once it encounters this directive, the assembler will stop interpreting program into machine code.
e.g. END ; End of the program.

Q.3(b) Draw the software development cycle. State the function of editor, assembler and cross compiler. **[4]**

Ans.:



OR



(i) 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.

(ii) 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.

(iii) Cross Compiler: A cross compiler is used to create executable code for a platform other than the one on which the compiler is run. Cross compiling is compiling something for different CPU type than the one you are running on. You use a cross compiler to produce objects for a platform other than the local host. Keil and SPJ have its own cross compiler.

Q.3(c) Describe the function of following instruction of 8051 microcontroller. [4]

(i) RLC A **(ii) XRL A, 15h**
(iii) DIV A B **(iv) MOVX @ DPTR, A**

Ans.: (i) RLC A
Description: Rotate a byte and carry bit to the left: the carry becomes the LSB and the MSB becomes the carry.

(ii) XRL A, 15h
Description: XOR each bit of A with the some bit of memory location 15H; put the result in A

(iii) DIV A B
Description Divide A and B; put the interger part of quotient in A and the integer part of remainder in B

(v) MOVX @ DPTR, A
Description: Copy yhe data from A to the external address given by DPTR.

Q.3(d) State the function of compiler, linker, assembler and editor in S/W development. [4]

Ans.: Assembly language programming tools:
(i) Editor (ii) Assembler
(iii) Compiler (iv) Linker

(i) Editor
An editor is a grogram 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.

(ii) 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 1st extension.
Some examples of assembler are ASEM–51, Keil"s A51, AX 51 and C 51, Intel PL/M–51 etc.

(iii) 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.

(iv) 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 is subsequent loading the execution.

Some examples of linker are ASEM–51 BL–51, Keil u Vision Debugger, LX 51 Enhanced Linker etc.

Q.3(e) Write an ALP for 8051 microcontroller to multiply two 8-bit numbers 23H and 15H. (Assume suitable memory addresses to store the result) **[4]**

Ans.: Program:

```
MOV 50H, # 23H          ; store first 8-bit no. in 50H
    MOV 51H, #15H       ; store second 8-bit no. in 51H
    MOV A, 50H          ; move first number to A
    MOV B, 51H          ; move second number to B
    MUL AB              ; multiply the numbers
    MOV 52H, A          ; move LSB to 52H
    MOV 53H, B          ; move MSB to 53H
HERE: SJMP HERE
```

Q.4(a) Attempt any THREE of the following: **[12]**

Q.4(a) (i) Write an ALP to calculate the sum of five consecutive numbers stored from memory location starting at 20 H. Store the lower byte at memory location 25 H and higher byte at 26 H. **[4]**

Ans.: Program for addition of five 8 bit nos.

```
        Org 0000h
        Sjmp start
        Org 0030h
Start:  MOV R0, #05H        ; Initialize byte counter
        MOV R1, #20H        ; Initialize memory pointer
        MOV R7,#00H         ;Initialize higher byte counter
        MOV A, # 00H        ; Clear Accumulator
UP:     ADD A @R1           ; Add accumulator with number from array
        JNC Next            ;if cy-0,then go to next
        INC R7              ;increment R7(for carry)
Next:   INC R1              ; Increment memory pointer
        DJNZ R0, UP         ; Decrement byte counter, if not zero add again
        MOV25H, A           ; Store lower byte of result in internal memory
        MOV 26H,R7          ;Store higher byte of result in internal memory
HERE:   SJMP HERE           ; Stop
        end
```

Q.4(a) (ii) Explain the following addressing modes with the help of ADD instruction in **[4]**
(1) Direct addressing mode (2) Indirect addressing mode
(3) Register addressing mode (4) Immediate addressing mode

Ans.: (1) Direct Addressing mode

ADD A, add (8-bit Address)

A←A+ (add)

Eg. ADD A, 12H

The contents of memory location specified by 8 bit direct address will be logically added bit by bit with the contents of accumulator and result is stored in accumulator.

(2) Indirect addressing mode

ADD A,@Ri

A ← A+ (Ri)

ADD A,@R0

The content of memory location whose address is specified by Ri (R0/R1) will be logically added bit by bit with contents of accumulator. Result is stored in accumulator. Only R0 or R1 can be used.

(3) Register addressing mode

ADD A, Rn

A ← A+ Rn

ADD A, R2

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

(4) Immediate addressing mode

ADD A, #data(8-bit)

A ← A+ #data

ADD A, #23H

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

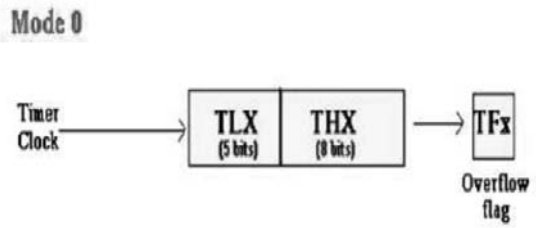
Q.4(a) (iii) With the help of neat diagram, describe the timer modes of 8051 μC. [4]

Ans.:

M1	M0	MODE	DESCRIPTION
0	0	0	13-bit time
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:

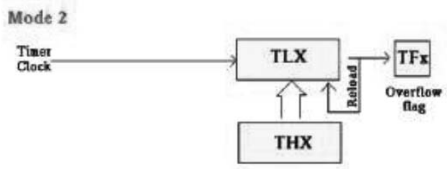


It is similar to Mode 0 except TLX is configured as a full 8-bit counter. Hence pulse input is divided by 256_{10} so that TH counts the timer flag is set when THX rolls over from FF to 00H.

Mode 1:

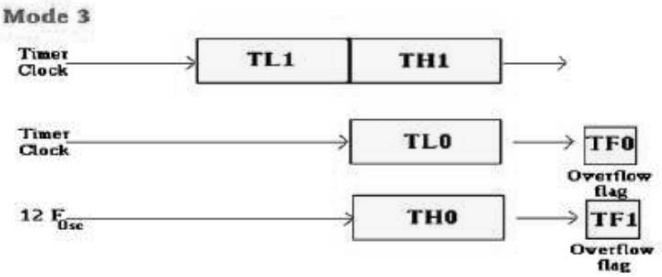
It is similar to Mode 0 except TLX is configured as a full 8-bit counter. Hence pulse input is divided by 256_{10} so that TH counts the timer flag is set when THX rolls over from FF to 00H.

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. Every time 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.4(a) (iv) Explain the four operating modes of serial communication of 8051 microcontroller. [4]

Ans.: 8051 micro controller communicate with another peripheral device through RXD and TXD pin of port3. Controller have four mode of serial communication. This four mdoe fo serial communication are below.

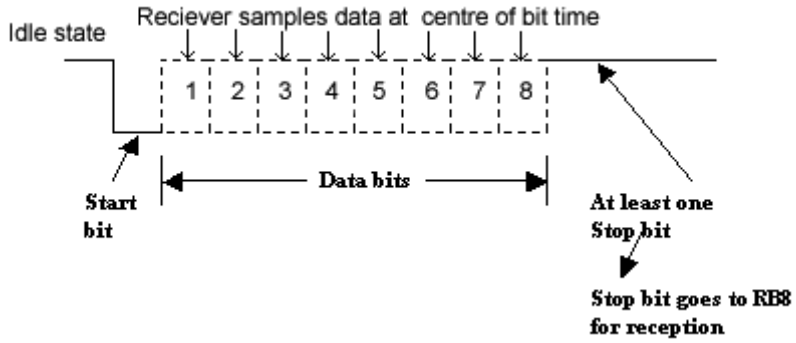
(i) Serial data mode 0–fixed baud rate.
(ii) Serial data mode 1-variable baud rate
(iii) Serial data mode 2-fixed baud rate
(vi) Serial Data mode 3-variable baud rate.

(i) 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.

(ii) 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**.



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

(iii) Serial Data Mode-2 Multiprocessor (baud rate is fixed)

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 first), a programmable 9th (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 9th 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.

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

(iv) 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 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).

$f\ baud = (2^{SMOD} /32) * (f_{osc} / 12 (256-TH1))$

Q.4(b) Attempt any ONE of the following : [6]

Q.4(b) (i) Write an assembly language program to send message "HELLO" serially at 4800 band rate continuously (Crystal frequency = 11.0592 MHz) [6]

Ans.:

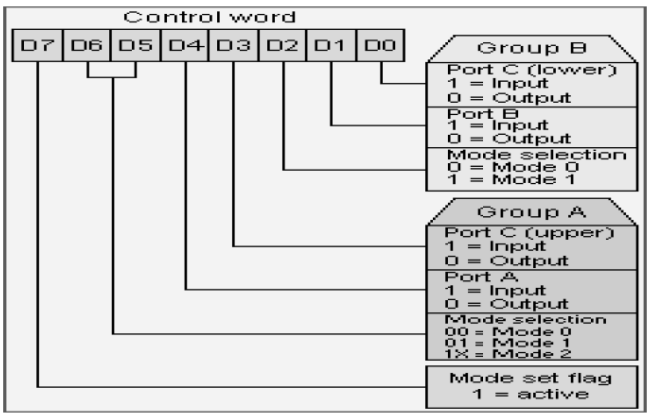
```
MOV TMOD, #20H                  ; timer 1, mode2
MOV TH1,#-6 or MOV TH1,#0FAh    ; 4800 baud rate
MOV SCON, #50H                  ; 8-bit data,1 stop bit, REN enabled
SETB TR1                        ; Start timer 1
AGAIN: MOV A, # "H"             ; transfer "H"
ACALL MESSAGE                   ; Some delay
    MOV A, # "E"                ; transfer "E"
ACALL MESSAGE
MOV A, # "L"                    ; transfer "L"
ACALL MESSAGE
    MOV A, # "L"                ; transfer "L"
ACALL MESSAGE
    MOV A, # "O"                ; transfer "O"
ACALL MESSAGE
    SJMP AGAIN

MESSAGE: MOV SBUF, A;

JNB TI, HERE;
CLR TI;
RET
```

Q.4(b) (ii) Draw the interfacing diagram of 8 LEDs to port 2 of 8051 microcontroller. Write an ALP to turn these LEDs ON and OFF after a certain delay. [6]

Ans.:

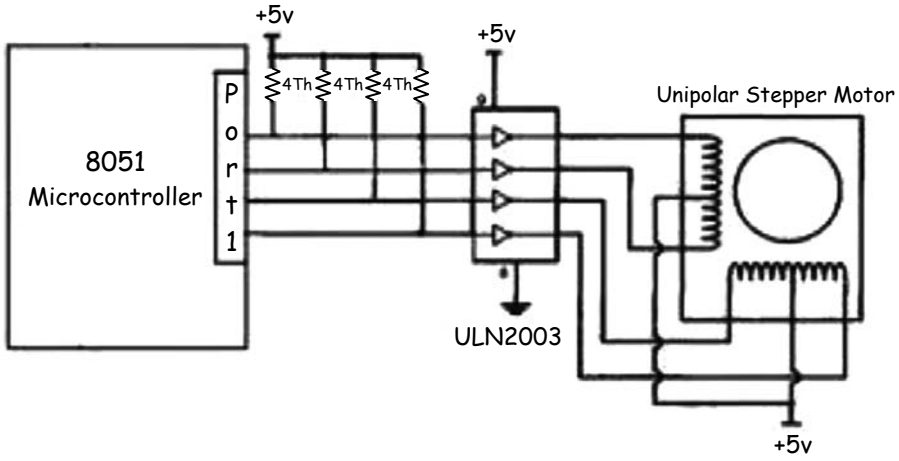


Program

```
        Org 0000h
        Sjmp start
        Org 0030h
Start:  MOV SP,#30h
        MOV A, # 0FFH          ; Store FFH in A
BACK:   MOV P2, A              ; move FF to P2 to glow all the LEDs
        ACALL DELAY            : wait for some time
        CPL A                  ; turn off the LEDs
        SJMP BACK
DELAY:  MOV R3, #255
AGAIN:      DJNZ R3, AGAIN
        RET
        End
```

Q.4(b) (iii) Draw the interfacing diagram of stepper motor with 8051 microcontroller. [6]
Write an ALP to motor continuously in clockwise direction.

Ans.:



Program

```
        Org 0000h
        Sjmp start
        Org 0030h
Start   MOV SP,#30h
        MOV A,#66H      ;load step sequence
BACK:   MOV P1,A        ;issue sequence to motor
        RRA             ;rotate right clockwise
        ACALL DELAY     ;wait
        SJMP BACK       ;keep going
DELAY:  MOV R2,#100
H1:     MOV R3,#H255
H2:     DJNZ R3, H2
        DJNZ R2, H1
        RET
        End
```

Or

```
        Org 0000h
        Sjmp start
        Org 0030h
Start:  MOV SP,#30h
AGAIN:  MOV R1,#4
BACK:   MOV A,#00H
```

```
MOV DPTR,#2000h
MOVC A,@A+DPTR
MOV P1,A                ;issue sequence to motor
ACALL DELAY             ;wait
        INC DPTR
        DJNZ R1,BACK
        SJMP AGAIN      ;keep going
DELAY:  MOV R2,#100
H1:     MOV R3,#255
H2:     DJNZ R3, H2
        DJNZ R2,H1
        RET
EXCIT:  db 03h, 06h, 0ch, 09h
        End
```

Q.5 Attempt any FOUR of the following : [16]

Q.5 (a) List the interrupts used in 8051. Give their priorities and addresses. [4]

Ans.:

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.5 (b) Write a program to generate a square wave of 50% duty cycle on P1. 5 bit. [4]
Timer 0 is used to generate the time delay.

Ans.: 50% duty cycle so on time and off time is same. Assume square wave of 1khz so T_{on} and T_{off} will be 500µ sec.

I/P clock = $(11.059 \times 10^6)/12 = 1000000 = 921.58$KHz

$T_{in} = 1.085$µ sec

For 1 kHz square wave

$F_{out} = 1$ KHz

$T_{on} = 1/ 1 \times 10^3$

$T_{on} = 1000$µ sec

Consider half of it = T_{on} = 500µ sec

$N = T_{on} / T_{in} = 500/1.085 = 460.82$

$65536 - 461 = (65075)_{10} = (FE33)_{16}$

NOTE: Students can consider any frequency with 50% duty cycle. Accordingly TH0 and TL0 will change. They can consider even timer 1.

```
        ORG 0000
MOV     TMOD,# 01H  ; Mode 1,timer 0
HERE :  MOV         TL0,# 33H           ; Lower byte of timer 0
        MOV         TH0, # 0FF          ; Higher byte of timer 0
        CPL         P1.5                ; toggle P 1.5
        ACALL       DELAY
        SJMP        HERE
delay using timer 0
DELAY : SETB        TR0                 ; Start time 0
AGAIN : JNB         TF0, AGAIN
        CLR         TR0                 ; Stop timer 0
        CLR         TF0
        RET
```

Q.5 (c) Draw the format of IP register of 8051 microcontroller. Describe the function of each bit in it. **[4]**

Ans.:

D7							D0
--	--	PT2	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority bit = 0 assigns low priority.

--	IP.7	Reserved
--	IP.6	Reserved
PT2	IP.5	Timer 2 interrupt priority bit (8052 only)
PS	IP.4	Serial port interrupt priority bit
PT1	IP.3	Timer 1 interrupt priority bit
PX1	IP.2	External interrupt 1 priority bit
PT0	IP.1	Timer 0 interrupt priority bit
PX0	IP.0	External interrupt 0 priority bit

Q.5 (d) Write an ALP for 8051 microcontroller to generate square wave on port pin P2.1 using delay subroutine. **[4]**

Ans.:

```
ORG     0000
MOV     TMOD,# 01H                  ; Mode 1
HERE :  MOV     TL0,# 0F2H          ; Lower byte of timer 0
        MOVE    TH0, # 0FF          ;Higher byte of timer 0
        CPL     P2.1                    ; toggle P 2.1
        ACALL   DELAY
        SJMP    HERE
                                    ; delay using timer 0
DELAY : SETB    TR0                 ; Start time 0
AGAIN : JNB     TF0, AGAIN
        CLR     TR0                 ; Stop timer 0
        CLR     TF0
        RET
Or
        ORG 0000h
        SJMP START
        ORG 0030h
START:      P2.1
        ACALL DELAY
        SJMP START
        SJMP $

DELAY:  MOV R2,#250
XX:     DJNZ R2,XX
        RET
END
```

Q.5(e) Draw the circuit diagram of port 2 and describe its function. [4]

Ans.:

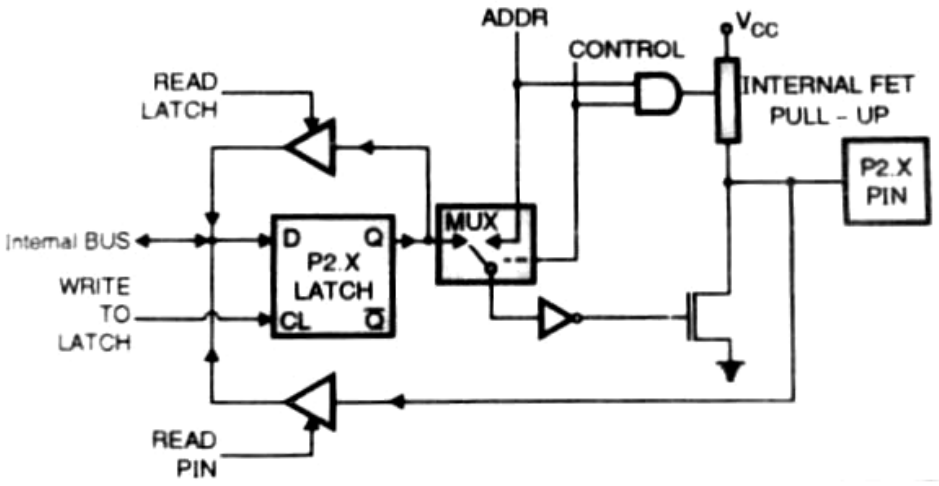


Fig.: Port 2 circuit

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.

Q.6 Attempt any FOUR of the following : [16]

Q.6 (a) Draw the format of TCON register and describe the function of each bit. [4]

Ans.: 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.

Q.6 (b) Draw the format of IE register of μc 8051 and describe function of each bit in it. [4]

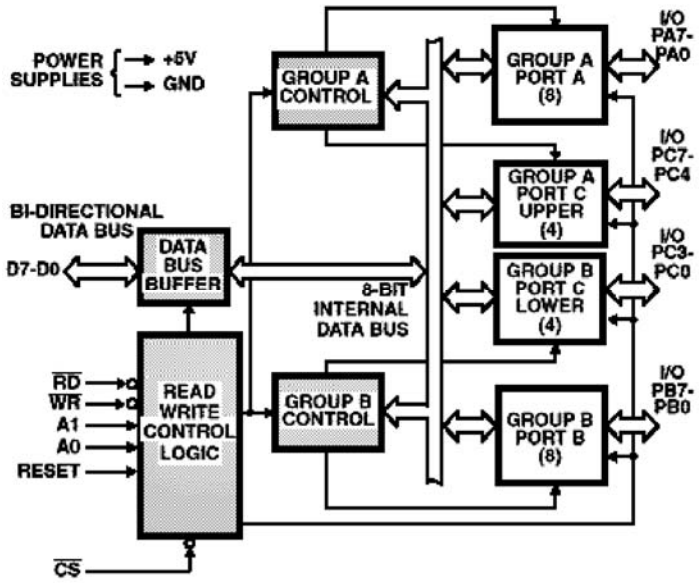
Ans.: IE: INTERRUPT ENABLE REGISTER, BIT ADDRESSABLE

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	-	ET2	ES	ET1	EX1	ET0	EX0

EA IE.7 Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
– IE.6 Not implemented, reversed for future use.
ET2 IE.5 Enable or disable the Timer 2 overflow or capture interrupt (8052 only)
ES IE.4 Enable or disable the serial port interrupt.
ET1 IE.3 Enable or disable the Timer 1 overflow interrupt.
EX1 IE.2 Enable or disable the External Interrupt 1.
ET0 IE.1 Enable or disable the Timer 0 overflow interrupt.
EX0 IE.0 Enable or disable External Interrupt 0.
*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

Q.6 (c) Draw the block diagram of IC 8255 and describe its operating mode. [4]
Ans.:



Q.6 (d) Draw format of SFR SCON and explain each bit of same. [4]
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.
RI SCON.0 Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.
Note: Make SM2, TB8 and RB8=0.

SM0	SM1	
0	0	Serial Mode 0
0	1	Serial Mode 1, 8-bit data, 1 stop bit, 1 start bit
1	0	Serial Mode 2
1	1	Serial Mode 3

SM2: SM2 is the D5 bit of the SCON register.
This bit enables the multiprocessing capability of the8051. MakeSM2=0 since we are not using the 8051 in a multiprocessor environment.

REN: The REN (receive enable)bit is D4 of the SCON register. The REN bit is also referred to as SCON.4 since SCON is a bit addressable register.
When the REN =1, it allows the 8051 to receive data on the RxD pin of the 8051. As a result, if we want the8051 to both transfer and receive data, REN must be set to 1.

By making REN=0, the receiver is disabled. Making REN=1 or REN=0can be achieved by the instructions "SETBSCON.4" and "CLR SCON.4", respectively.
This bit can be used to block any serial data reception and is an extremely important bit in the SCON register.

TB8: TB8 (transfer bit 8) is bit D3 of SCON. It is used for serial modes 2 and 3.
WemakeTB8=0 since it is not used in our applications.

RB8: RB8 (receive bit 8) is bit D2 of the SCON register. In serial mode1, this bit gets copy of the stop bit when an 8-bitdata is received. This bit (as is the case for TB8) is rarely used any more. In all our applications wewillmakeRB8=0.Like TB8, the RB8 bit is also used in serial modes 2and 3.

TI: TI (transmit interrupt)is bit D1 of the SCON register. This is an extremely important flag bit in the SCON register.

When the 8051 finishes the transfer of the 8-bit character, it raises the T1flagto indicate that it is ready to transfer another byte. The TI bit is raised at the beginning of the stop bit.

RI: RI (receive interrupt)is theD0 bit of the SCON register. This is another extremely important flag in the SCON register.

When the 8051 receives data serially via RxD, it gets rid of the start and stop bits and places the byte in the SBUF register.

Then it raises the RI flag bit to indicate that a byte has been received and picked up before it is lost. RI is raised halfway through the stop bit.

Q.6 (e) Describe selection factors of microcontroller. [4]

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, 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

❑ ❑ ❑ ❑ ❑