SUMMER– 15 EXAMINATION

Important Instructions to examiners:

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

2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.

3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.

4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.

5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.

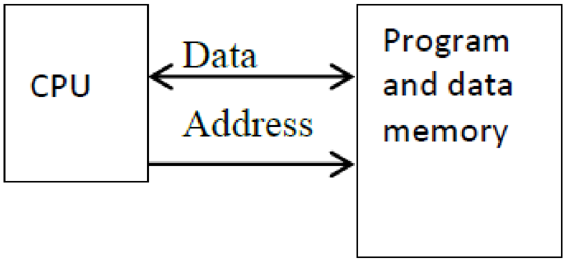
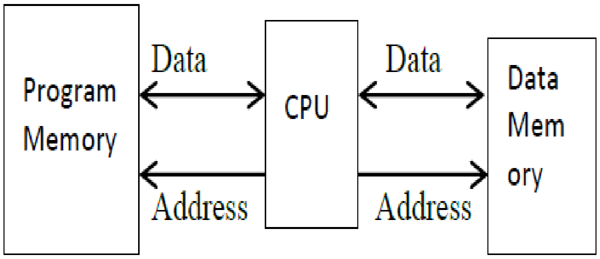
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.

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

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

a) Compare Von – Neumann and Harvard Architecture. Give Examples.

Ans: **(any Two relevant correct points – 1 mark each, example of each– 1 marks)**

Sr.No	Von Neumann architecture	Harvard architecture
1		 Program Memory Data Address CPU Data Address Data Mem ory
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.

- Examples of Von – Neumann Architecture: ARM 7 and Pentium Processors etc.
- Examples of Harvard Architecture: 8051, ARM 9, AVR by Atmel Corporation and PIC microcontrollers by microchip Technology etc.

b) State important selection factors of microcontroller for microcontroller based system.

Ans: **(any four relevant correct factors – 1 mark each)**

The selection of microcontroller depends upon the type of application. The following factors must be considered while selecting the microcontroller.

1. **Speed:** What is the highest speed that microcontroller supports?
2. The amount of RAM and ROM on chip.
3. **Power Consumption**
4. **Packaging**
5. **Easy availability of IC's.**
6. **Word length:** The word length of microcontroller is 8, 16 or 32 bit. As the word length increases, the cost, power dissipation and speed of the microcontroller increases.
7. **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.

8. **Clock frequency:** The speed of an embedded system depends upon the clock frequency. The clock frequency depends upon the application.
9. **Instruction Set:** On the basis of instructions microcontrollers are classified into two categories
   i. CISC
   ii. RISC.

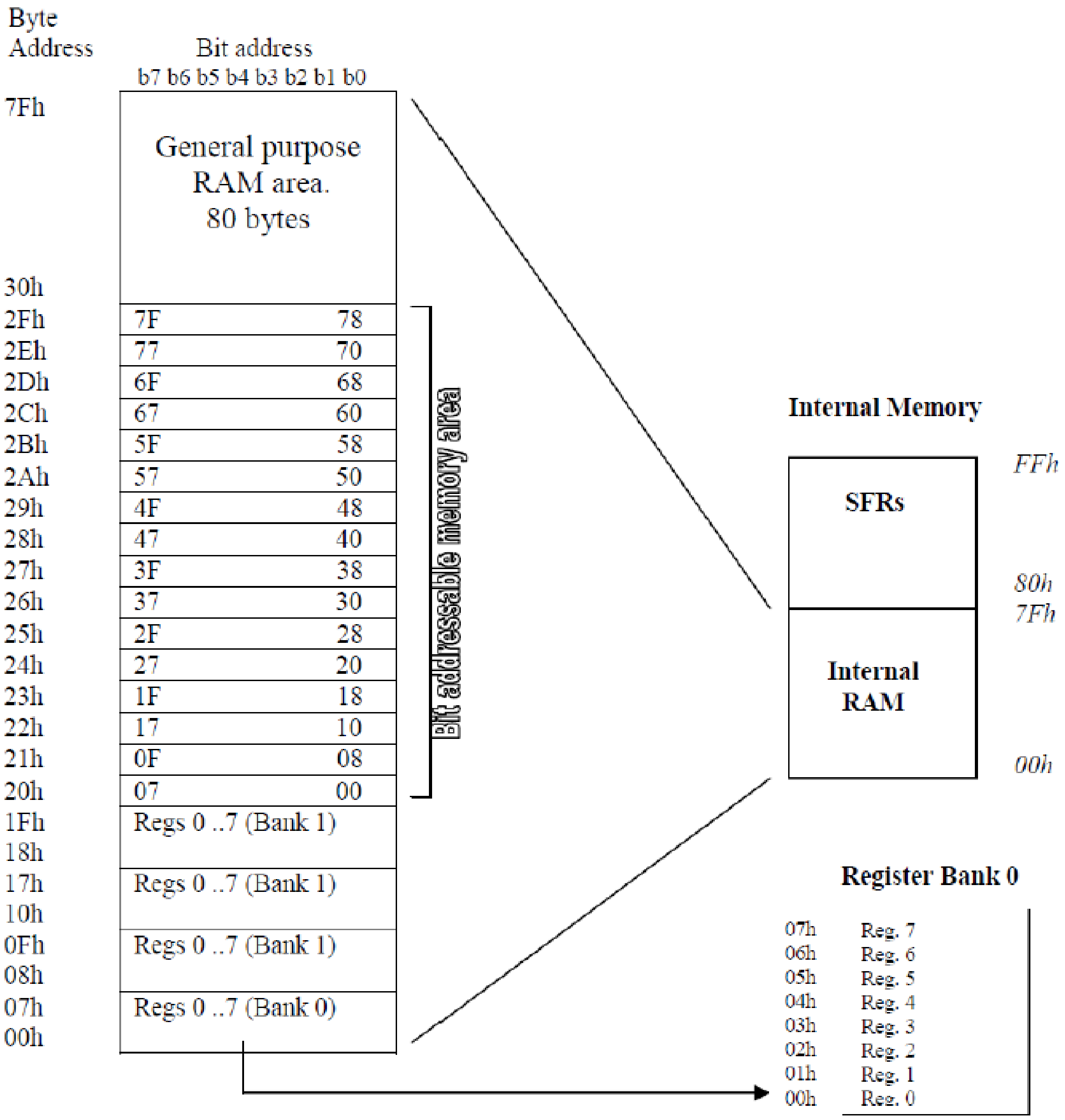
CISC system improves software flexibility. Hence it is used in general purpose systems.

RISC improves speed of the system for the particular applications.

10. **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.
11. **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.

c) Draw internal RAM organization of 8051 microcontroller.

Ans: **(any relevant correct diagram – 4 marks)**



d) Distinguish between assembler, cross compiler, compiler.

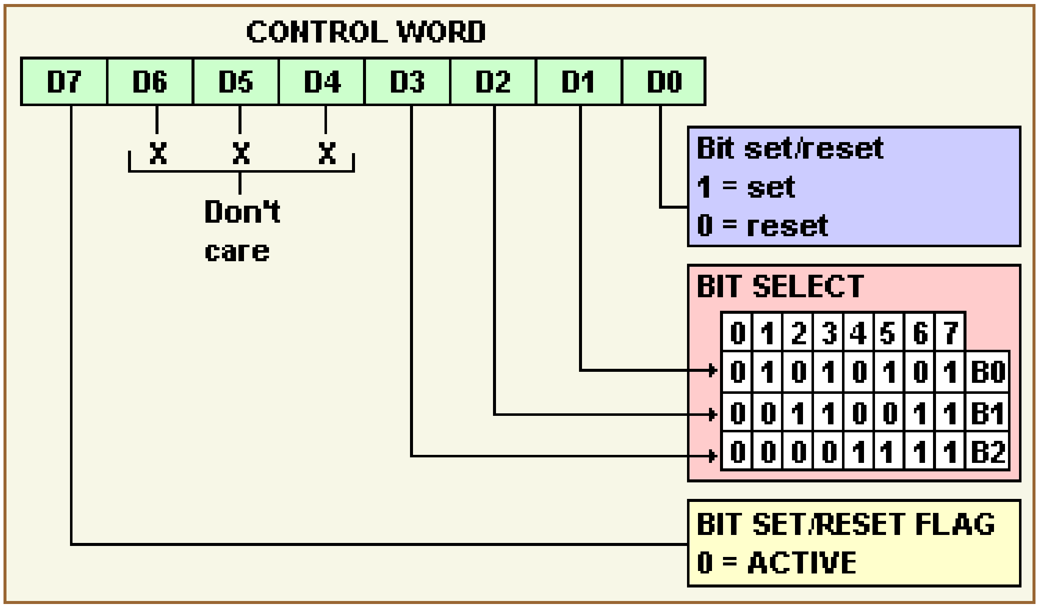
Ans: **(Any four relevant correct points – 1 mark each)**

Sr.No	Assembler	Cross Compiler	Compiler
1	An assembler is program that translate assembly language program to the correct binary code from each instruction.	A cross compiler is used to create executable code for a platform other than the one on which the compiler is run.	A compiler is a computer program that transforms source code written in a programming language into another computer language often a binary form.
2	An assembler is program when you assemble for the same architecture you're	Cross compiling is compiling something for different CPU type than the one you are	compiling is when you compile for the same architecture you're running

	running under, which is the normal situation.	running on.	under , which is the normal situation.
3	Assembler is used to generate object file with extension .obj	You use a cross compiler to produce executables (or objects) for a platform other than the local host.	The compiler only produces native binaries.
4	Assembler for 8051 are ASEM51, A51 etc.	Keil and SPJ have its own cross compiler	Compilers available are Keil and SPJ etc.

e) Draw control word format of 8255 for BSR mode and write control word to set bit PC7.

Ans: **(Control word format – 2 marks, Control word to set PC7 – 2 marks)**



The Control word to set bit PC7 is

D7 D6 D5 D4 D3 D2 D1 D0

0 0 0 0 1 1 1 1 = 0F H

(B) Attempt any ONE of the following 6 marks

a) Write an ALP for 8051 microcontroller to copy block of ten bytes of data from RAM location starting from 35H to RAM location starting at 60H.

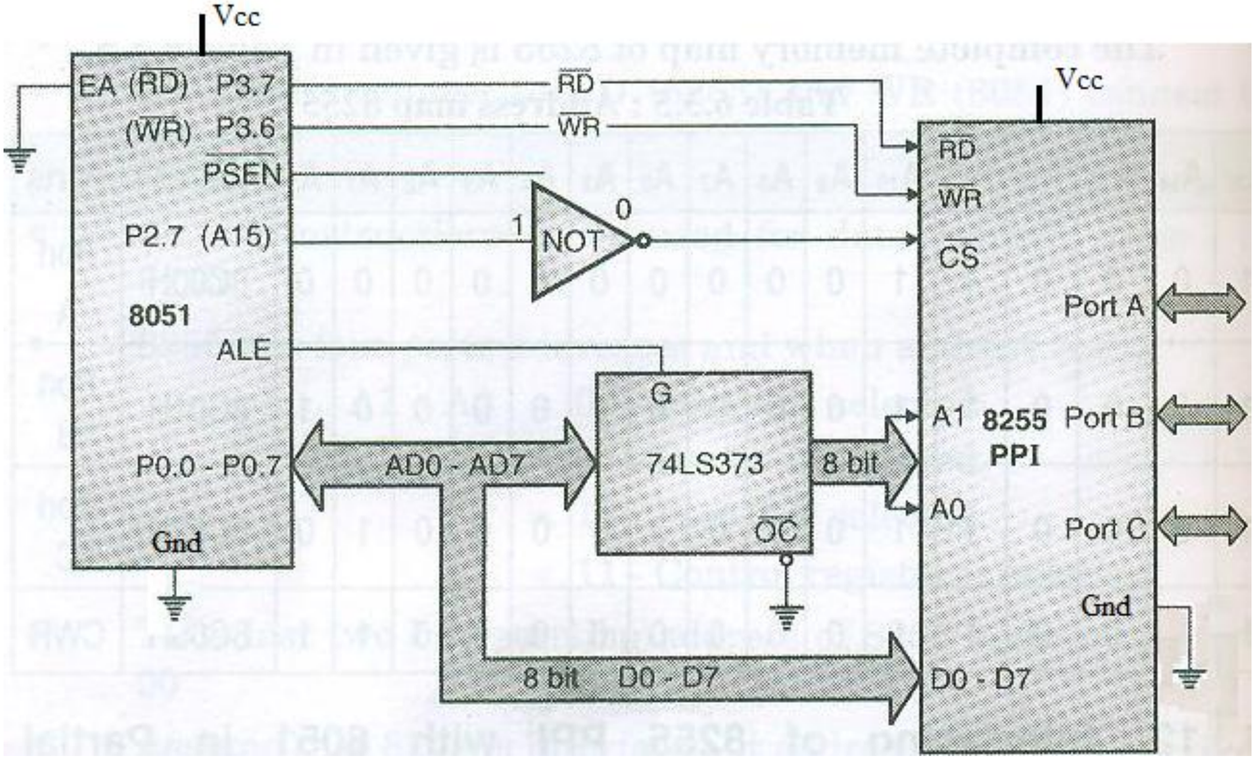
Ans: **(any relevant correct program with comments – 6 marks)**

```
ORG 0000H               ;Program from 0000H
MOV R3, #10             ;Initialize Byte counter
MOV R0, #35H            ;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 number from source array
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
```

b) Draw the interfacing diagram of 8051 microcontroller with 8255. State I/O port address and control word register address.

Ans: **(Interfacing Diagram – 4 marks, I/O Port & Control word address mapping – 2 marks)**

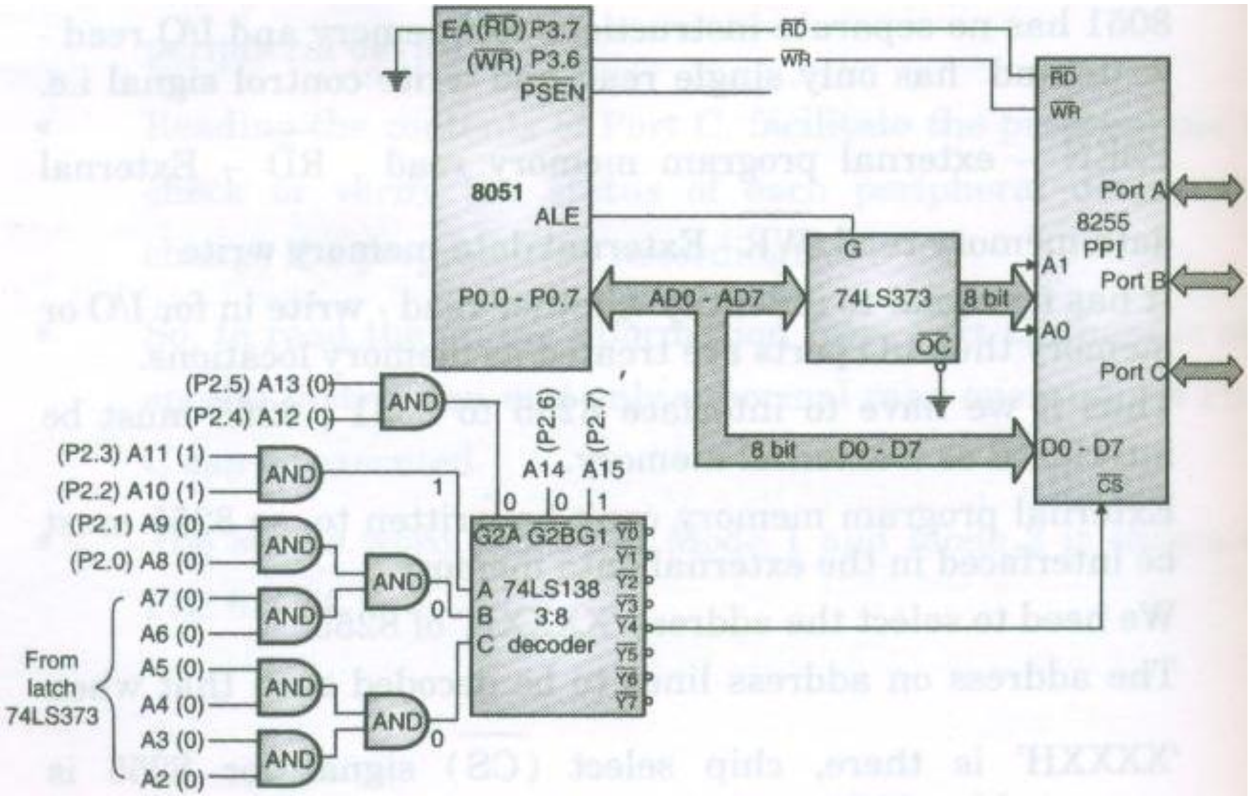
(Note: Interfacing and address mapping using absolute decoding should also be considered)



Address Mapping

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Add	Register
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H	Port A
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8001H	Port B
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	8002H	Port C
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	8003H	Control Word

(OR)



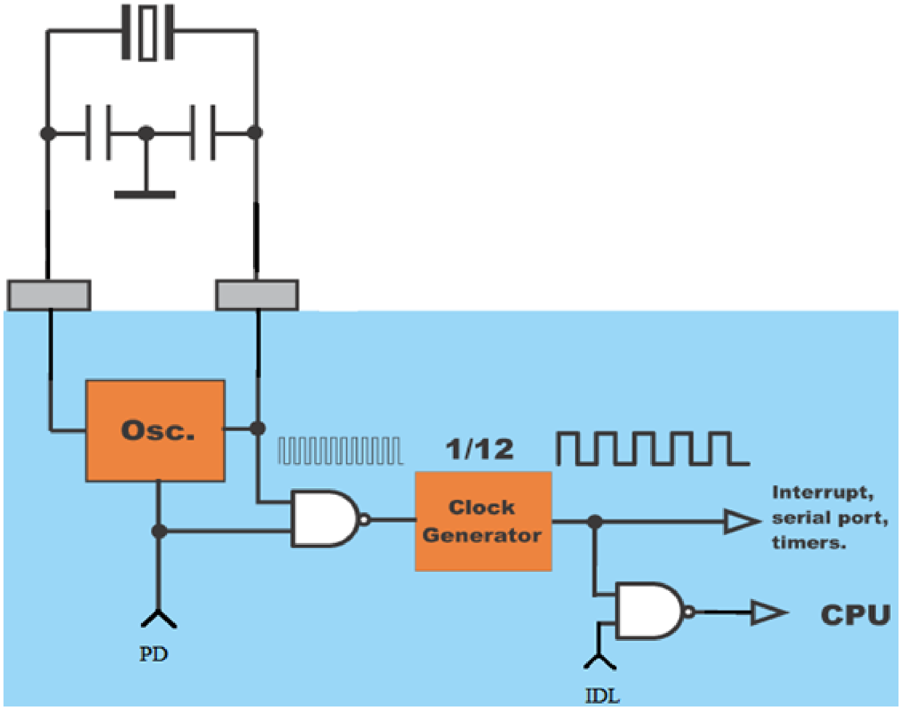
Address Mapping

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Add	Register
1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	8C00H	Port A
1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	8C01H	Port B
1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	8C02H	Port C
1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	8C03H	Control Word

Q.2 Attempt any FOUR **16 marks**

a) Describe the power saving options of 8051 microcontroller.

Ans: **(Diagram – 1 mark, PCON format – 1 mark, Power down mode & Idle Mode explanation – 1mark)**



SMOD	--	--	--	GF1	GF0	PD	IDL

Fig: Format of PCON

SMOD - Double Baud rate bit. If timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubles 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 8051.

IDL - Idle Mode bit. Setting this bit activates Idle Mode operation in 8051.

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.

b) Describe functions of following pins of 8051 microcontroller.

i) $\overline{\text{PSEN}}$

ii) ALE

iii) $\overline{\text{EA}}$

iv) RST

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

Function of $\overline{\text{PSEN}}$:

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 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.

Function of $\overline{\text{EA}}$:

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 V_{cc} 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 V_{cc}, 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:

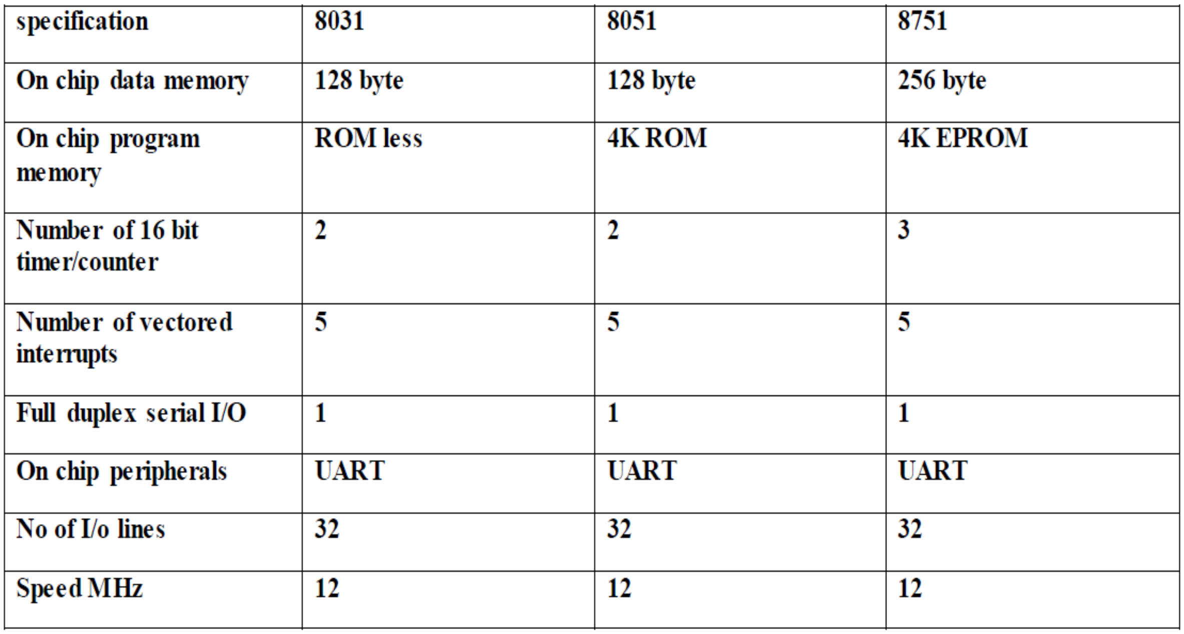
1. Pin 9 is the RESET pin. It is an input and is active high (normally low). Upon applying a high pulse to this pin, the microcontroller will reset and terminate all activities.
2. This is often referred to as a power-on reset. Activating a power-on reset will cause all values in the registers to be 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.

c) State any three derivatives of 8051 and compare. (Any THREE points)

Ans: **(List of any two derivatives – 1 mark, comparison any three correct points – 1 mark each)**

The derivatives of 8051 are:

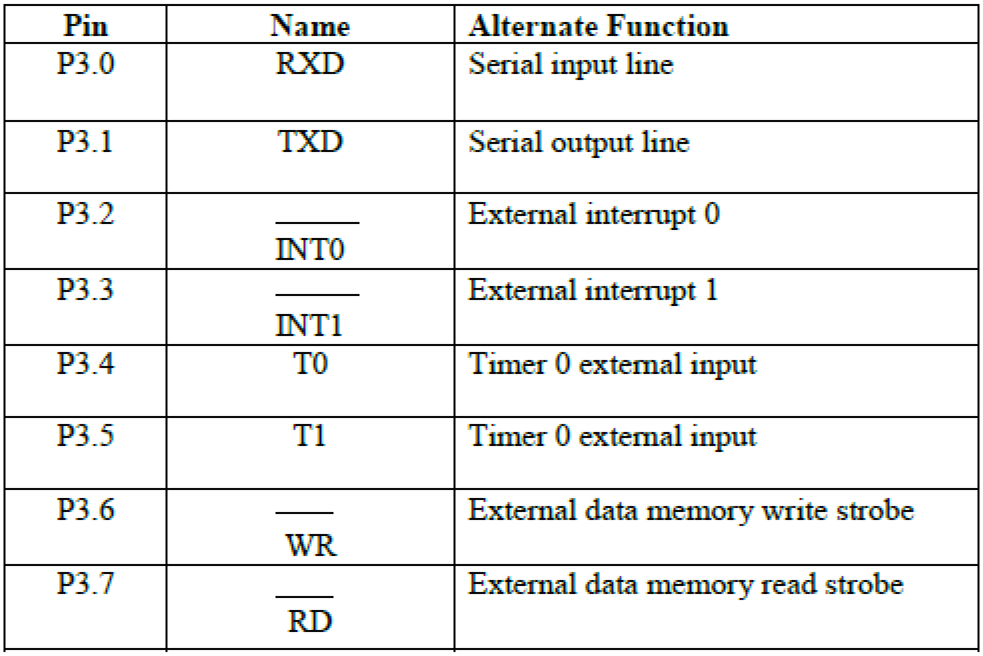
Sr.No	Manufacturer	Version
1	Intel	8031, 8051, 8751, 8052, 8752 etc.
2	Atmel	89C51, 89LV51, 89C52, 89LV52 etc.
3	Dallas	80CH11, 89C420, 89C430, 89C440 etc.
4	Phillips	89C51RA2, 89V51RD2, 89C51RD2, etc.



specification	8031	8051	8751
On chip data memory	128 byte	128 byte	256 byte
On chip program memory	ROM less	4K ROM	4K EPROM
Number of 16 bit timer/counter	2	2	3
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

d) State the alternate functions of Port 3 of 8051 microcontroller.

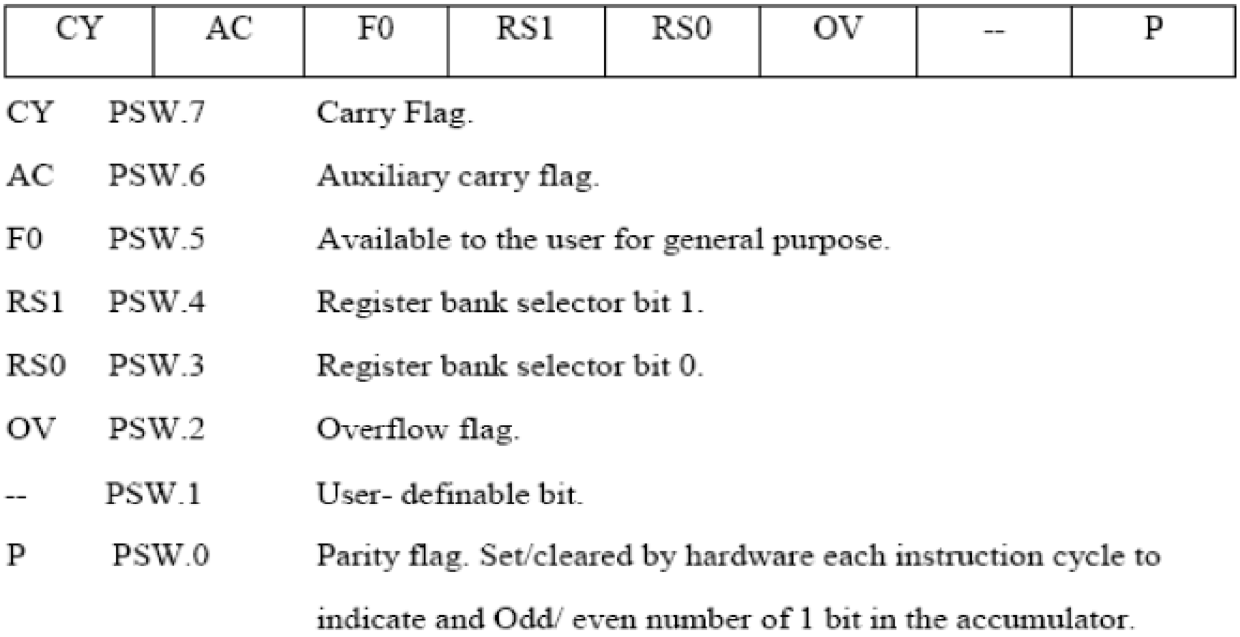
Ans: **(All eight correct pin functions – ½ mark each)**



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

e) Draw format of PSW register of 8051 microcomputer and state function of each flag.

Ans: **(PSW format – 2 marks, function of each bit – 2 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.

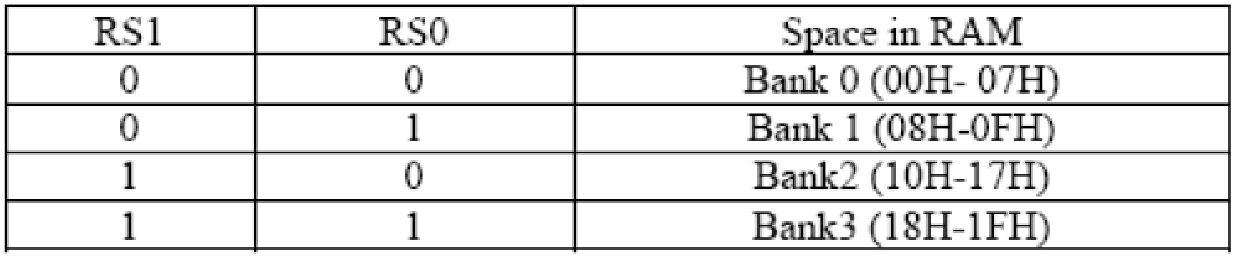
2. AC: the auxiliary carry flag

- 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.

f) Differentiate between microcomputer and microcontroller. (Any FOUR points)

Ans: **(any four relevant correct points – 1 mark each)**

Sr.No	Microcomputer	Microcontroller
1	Boolean operation is possible directly only if microcontroller is used.	Boolean operation is possible directly.
2	Microcontroller is a single chip microcomputer with on-board program ROM and I/O that can be programmed for various control functions.	Memory, I/O ports, timers, interrupts are integrated inside the microcontroller chip.
3	Widely used in large control system such as automation, CNC machine etc.	Widely used in small control system like washing machine, modem etc.
4	Mobile phone, Tabs, Laptops etc. are various microcomputers.	INTEL 8051, Phillips 89V51RD2, PIC 16F877 etc. are various microcontrollers
5	System is bulky, costly.	System is simple, cheaper.

Q.3 Attempt any FOUR 16

a) Draw and explain software development cycle.

Ans: (2M –diagram, 2M- explanation)

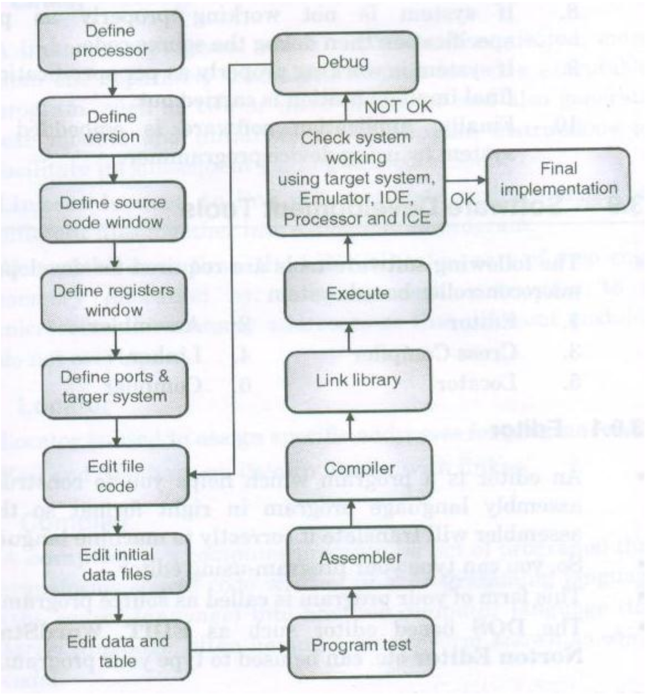


Fig: software development cycle.

OR

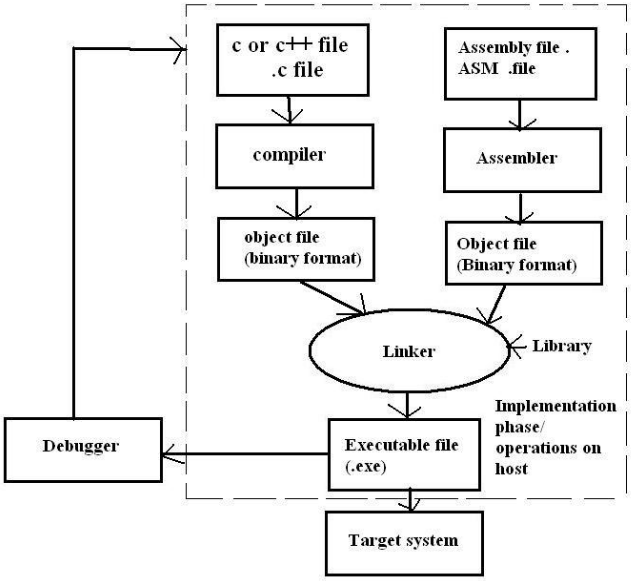
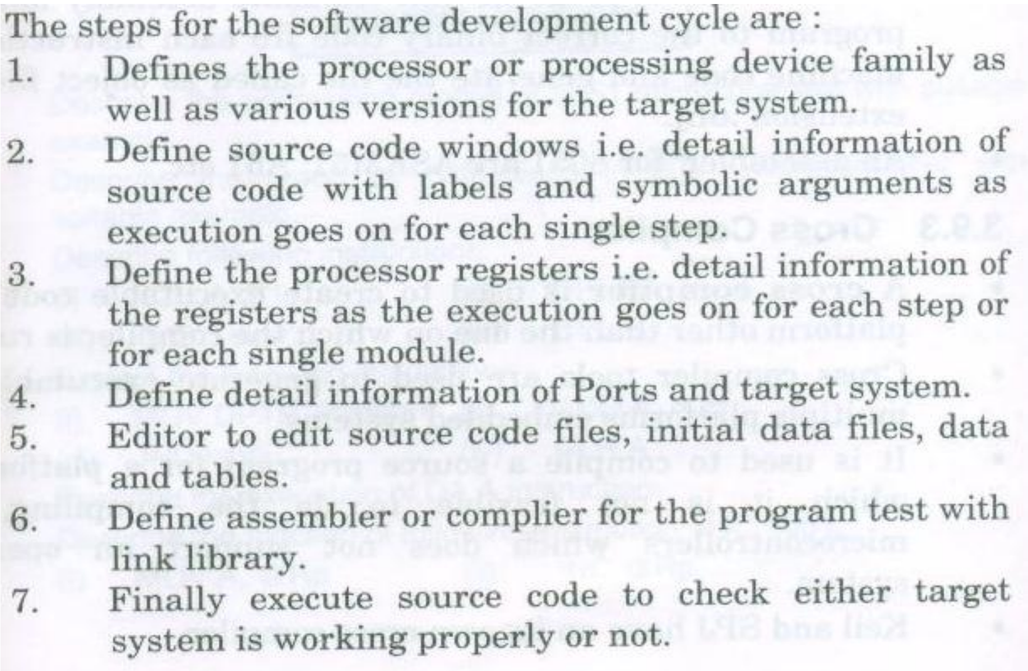
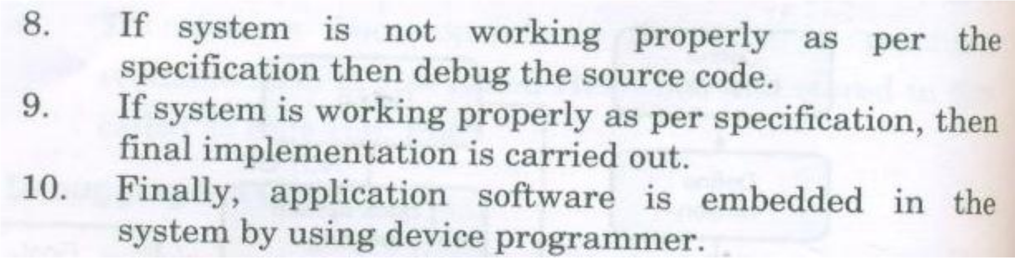


Fig: software development cycle



The steps for the software development cycle are :

1. Defines the processor or processing device family as well as various versions for the target system.
2. Define source code windows i.e. detail information of source code with labels and symbolic arguments as execution goes on for each single step.
3. Define the processor registers i.e. detail information of the registers as the execution goes on for each step or for each single module.
4. Define detail information of Ports and target system.
5. Editor to edit source code files, initial data files, data and tables.
6. Define assembler or complier for the program test with link library.
7. Finally execute source code to check either target system is working properly or not.



8. If system is not working properly as per the specification then debug the source code.
9. If system is working properly as per specification, then final implementation is carried out.
10. Finally, application software is embedded in the system by using device programmer.

b) State any four addressing modes of 8051 microcontroller and explain with one example each.
(any four addressing modes -1M each)

Ans: 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. Absolute addressing mode
7. Long Addressing mode
8. Indexed Addressing mode

1) **Immediate Addressing mode:**

Immediate addressing simply means that the operand (which immediately follows the Instruction op. code) is the data value to be used.

For example the instruction:

MOV A, #25H ; Load 25H into A

Moves the value 25H into the accumulator. The # symbol tells the assembler that the immediate addressing mode is to be used.

2) Register Addressing Mode:

One of the eight general-registers, R0 to R7, can be specified as the instruction Operand. The assembly language documentation refers to a register generically as Rn.
An example instruction using register addressing is :
ADD A, R5 ; Add the contents of register R5 to contents of A (accumulator)
Here the contents of R5 are added to the accumulator. One advantage of register addressing is that the instructions tend to be short, single byte instructions.

3) Direct Addressing Mode:

Direct addressing means that the data value is obtained directly from the memory location specified in the operand.
For example consider the instruction:
MOV R0, 40H; Save contents of RAM location 40H in R0.
The instruction reads the data from Internal RAM address 40H and stores this in theR0. Direct addressing can be used to access Internal RAM, including the SFR registers.

4) Register Indirect Addressing Mode:

Indirect addressing provides a powerful addressing capability, which needs to be appreciated.
An example instruction, which uses indirect addressing, is as follows:
MOV A, @R0; move contents of RAM location whose address is held by R0 into A
Note the @ symbol indicated that the indirect addressing mode is used. If the data is inside the CPU, only registers R0 & R1 are used for this purpose.

5) Relative Addressing Mode:

This is a special addressing mode used with certain jump 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.

c) State necessity of Assembler directives. List any four assembler directives and describe with example.

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

Ans: i) ORG:-ORG stands for Origin

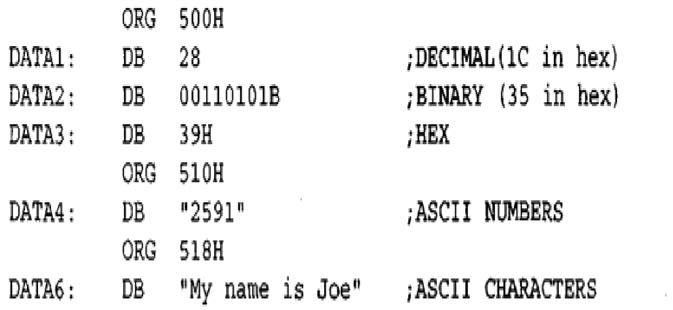
Syntax: | ORG | Address |

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.

ii) DB:- (Data 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. 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
```

iii)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).

iv) **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(asm).

Once it encounters this directive, the assembler will stop interpreting program into machine code.

e.g. END ; End of the program.

d) Write a program to find the sum of data stored at five consecutive memory locations starting from 40H.Store lower byte in A and higher byte in R7.

Assume data 40H=7D 41H=EB

42H= C5 43H= 5B 44H= 32

Ans: (Any relevant program-)

```
         CLR PSW.3                     ; Select register Bank 0
         CLR PSW.4;
         MOV Ro, #05H                  ; Initialize byte counter
         MOV R1, #40H                  ; Initialize memory pointer
         MOV R7, #00H                  ; Initialize MSB COUNTER
         MOV A, # 00H                  ; Clear Accumulator
UP:      ADD A @R1                     ; Add accumulator with number from array
         INC R1                        ; Increment memory pointer
         DJNZ R0, UP                   ; Decrement byte counter,
                                       ; if byte counter ≠ 0
                                       ; Then go to UP
         JC Down                       ; If result greater than 8 bit then go to Down
Down:    INC R7                        ; higher byte in R7
         MOV30H, A                     ; lower byte
                                         Is Available in A, or Store result in internal
                                         memory)
HERE: SJMP HERE                        ; Stop
```

e) Draw the format of SFR SCON and explain each bit of same.

Ans: (2M- format, 2M 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.

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 the 8051. Make SM2= 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 the 8051 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=0 can be achieved by the instructions “SETB SCON.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. We make TB8=0 since it is not used in our applications.

RB8: RB8 (receive bit 8) is bit D2 of the SCON register. In serial mode 1, this bit gets copy of the stop bit when an 8 bit data is received. This bit (as is the case for TB8) is rarely used anymore. In all our applications we will make RB8=0. Like TB8, the RB8 bit is also used in serial modes 2 and 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 T1 flag to 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 the D0 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.

Q4 .A) Attempt any THREE **12**

a) Describe following microcontroller instructions

i)LJMP addr ii) SJMP addr iii) RL A iv) RR A

Ans: (1M each)

i) LJMP addr: Long Address

The program counter is loaded with the 16-bit long address given in 2nd and 3rd byte of the instruction. Jump can take place anywhere in the 64KB memory space.

PC ⟸ addr

Example: LJMP 3245H

PC ⟸ 3245

The next instruction executes from the memory location 3245H.

ii) SJSJMP addr: Short address

PC_{new} ⟸ PC_{old} +addr

The program control jumps max 128 bytes backwards or max 127 bytes forward with respect to the current PC. Next instruction is executed from new PC.

iii) RL A: Rotate accumulator left.

It stands for rotate accumulator left. The contents of the Accumulator are rotated left bit-wise.

Example:

Before execution

A= 05H

0 ←	0 ←	0 ←	0 ←	0 ←	1 ←	0 ←	1

After execution

A= 0AH

0	0	0	0	1	0	1	0

iv) RR A: Rotate accumulator right.
It stands for rotate accumulator right. The contents of the Accumulator are rotated right bit-wise.
Example:
Before execution

A= 0AH

0 →	0 →	0 →	0 →	1 →	0 →	1 →	0

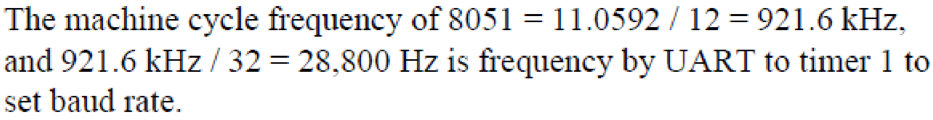
After execution
A= 05H

0	0	0	0	0	1	0	1

b)Write ALP for 8051 microcontroller to receive bytes of data serially and put them in port 1.Assume baud rate 4800,8 bit data , 1 stop bit.

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

Calculation :



The machine cycle frequency of 8051 = 11.0592 / 12 = 921.6 kHz, and 921.6 kHz / 32 = 28,800 Hz is frequency by UART to timer 1 to set baud rate.

28800/6=4800 where -6=FAH is loaded into TH1.

```
        MOV TMOD, #20H              ; Timer 1 , Mode 2
        MOV TH1, #-6                ; 4800 Baud
        MOV SCON, #50H              ; 8Bit ,1 Stop Bit
        SETB TR1                    ; Start Timer1
HERE: JNB RI, HERE                  ; Wait for char to come in
        MOV A, SUBF                 ; Save incoming byte in A
        MOV P1, A                   ; Send to port 1
        CLR RI                      ; Get ready to receive next byte
        SJMP HERE                   ; Keep getting data
```

c) Explain modes of the serial communication modes in 8051 microcontroller. (1M each mode)

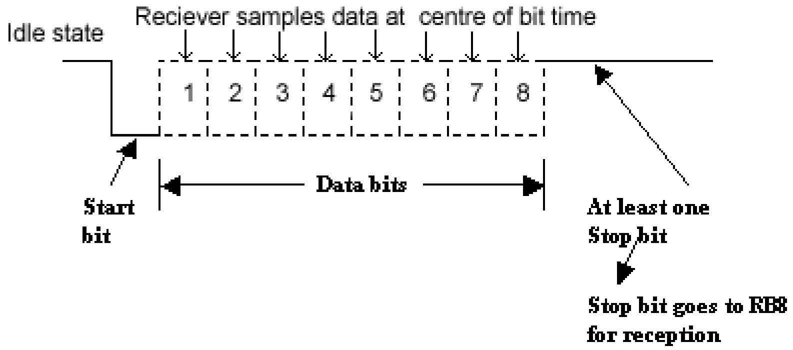
Ans: 8051 micro controller communicate with another peripheral device through RXD and TXD pin of port3.controller have four mode of serial communication.

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**.


Idle state
Reciever samples data at centre of bit time
1
2
3
4
5
6
7
8
Start bit
Data bits
At least one Stop bit
Stop bit goes to RB8 for reception

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

3. 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 9 th (TB8 or RB8)bit and a stop bit (usually '1'). While transmitting, the 9 th data bit (TB8 in SCON) can be assigned the value '0' or '1'. For example, if the information of parity is to be transmitted, the parity bit (P) in PSW could be moved into TB8.On reception of the data, the 9 th bit goes into RB8 in 'SCON', while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

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

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).

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

d) **What value should be loaded into TH1 of 8051 microcontroller to obtin 4800 baud rate? Assume crystal frequency = 11.0592 MHz, Give the answer in both decimal and hex.**

Ans:

The count to be loaded in TH1 to have a baud rate of 4800 can be calculated as follows

TH1=256d- [(K*oscillator frequency)/(384d*baud rate)]

Where,

SMOD = 0 then K=1 and SMOD = 1 then K=2

Therefore,

$TH1=256-[(1*11.0592*10^6)/(384*4800)]$

TH1=250d

TH1=0FAH

B) Attempt any ONE **6**

a) Assume internal RAM memory contains the following data. Write 8051 ALP to search for value equal to 65.If value 65 exists in the table then store it in R4.If value does not exists in the table then make R4= 0.

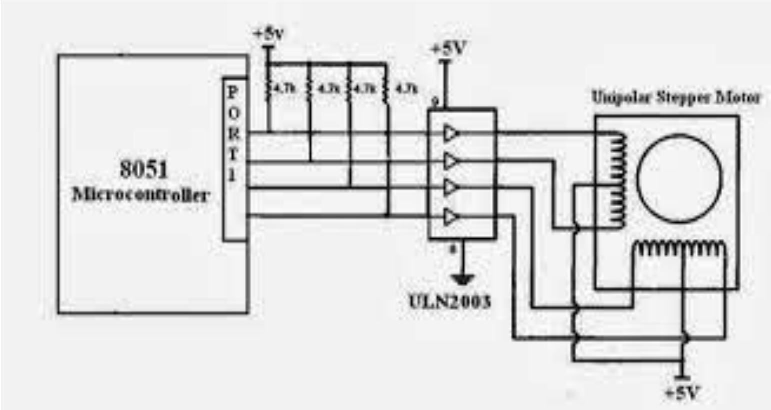
Data: 40H=76,41H=79,42H=rg,43H=65, 44H=62

Ans: (5M- program, 1M -comments)

```
        MOV R2, # 05H               ;Initialize the counter for RAM locations
        MOV R4,#00H                 ;load 00H in R4
        MOV R1,# 40H                ;initialize pointer
BACK: CJNE @R1, #65, NEXT           ; compare 65 with  contents if not equal then go
                                    NEXT
        MOV R4, #65                 ;if equal store 65 in R4
NEXT: INC R1                        ;increment pointer to go to next number in RAM
        DJNZ R2, BACK               ;check if compared with all numbers in RAM
HERE: SJMP HERE
```

b) **Draw the interfacing diagram of stepper motor with 8051 microcontroller and write ALP to rotate stepper motor in anticlockwise direction through 180^0. Asumme step angle of 1.8^0.**

Ans: (2M- diag ; 1M- step sequene, 1M – calculation of no. steps required;2 M - program)



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	

Calculation for no. of steps:

$$\textbf{Count} = \frac{\textbf{Required Rotaion Angle}}{\textbf{1.8 degree*no. of steps}}$$

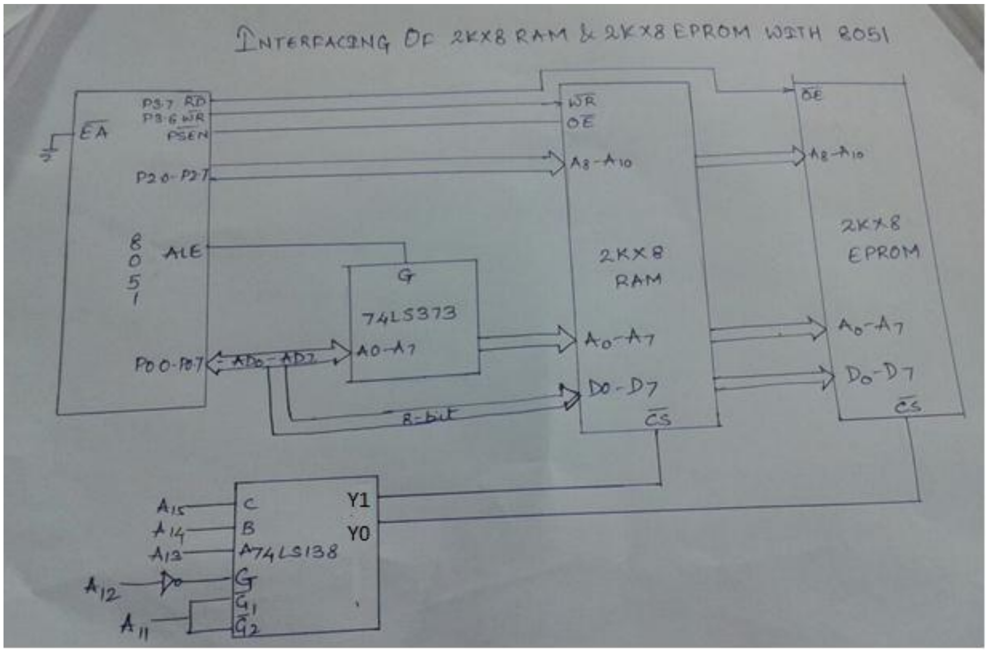
=180/1.8*4 steps=100/4=25

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 anticlockwise
        ACALL DELAY               ;wait
        DJNZ R0, AGAIN
        SJMP BACK                 ;keep going

DELAY                             ; delay subroutine.
        MOV R2, #100
H1:     MOV R3, #255
H2:     DJNZ R3, H2
        DJNZ R2, H1
        RET
```

c) Draw interfacing diagram of 2Kbyte EEPROM and 2Kbyte RAM to 8051 microcontroller. Draw memory map.



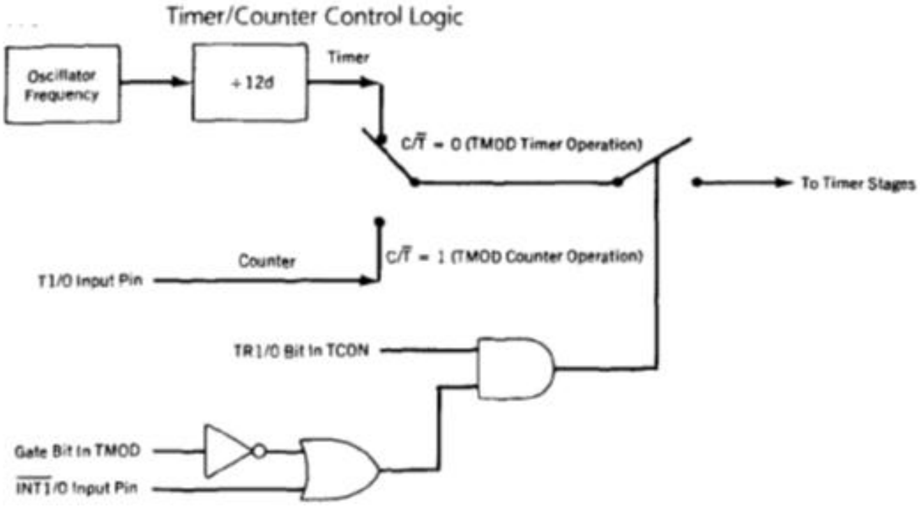
Memory map

	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 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.5Attempt any FOUR

a) Draw timer/counter control logic diagram of 8051 microcontroller.

(Correct diagram-4M)


Timer/Counter Control Logic
Oscillator Frequency
÷12d
Timer
C/$\overline{T}$ = 0 (TMOD Timer Operation)
To Timer Stages
Counter
C/$\overline{T}$ = 1 (TMOD Counter Operation)
T1/0 Input Pin
TR1/0 Bit In TCON
Gate Bit In TMOD
$\overline{INT1}$/0 Input Pin

b) Write an ALP for 8051 microcontroller to generate delay of 1 second by using Timer1.Use crystal frequency= 12MHz.

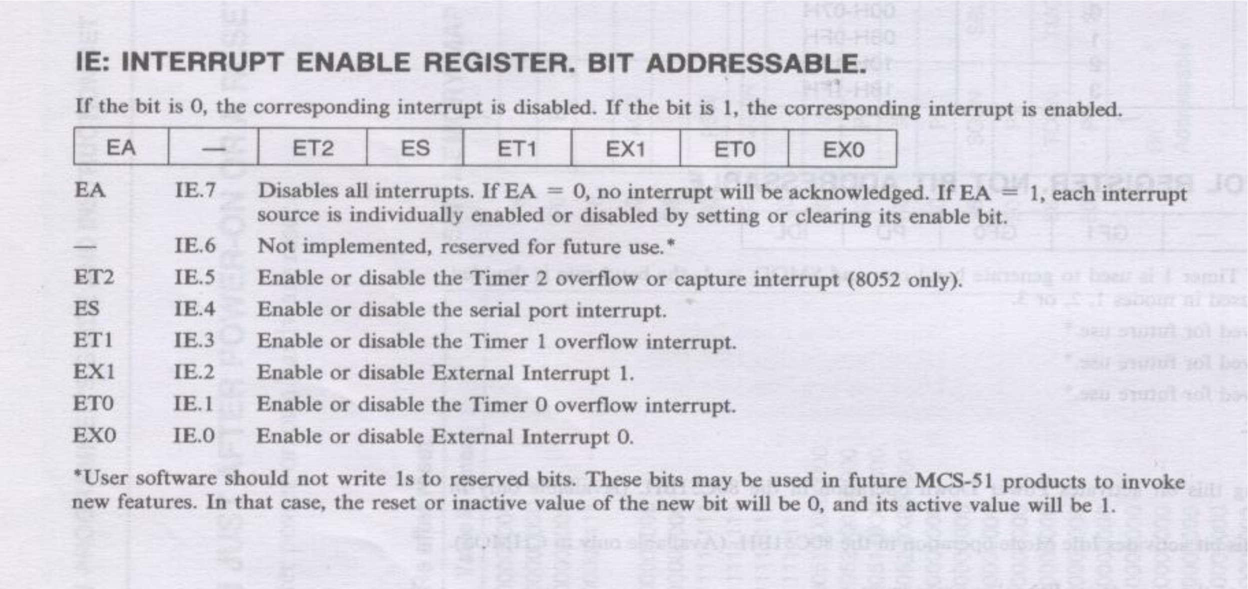
Ans: (1M- calculation, 2M- Program,1m- comments)

Crystal freq = 12MHz
Timer frequency= 12MHz/12= 1MHz
Time= 1/1MHz= 1us
For delay of 50ms
50ms/1us= 50000
Therefore count to be loaded in TH1 and TL1 can be calculated as 65536-50000= 15536d = 3CB0H
Program:

```
        MOV TMOD, #10H              ; Timer 1 , mode 1
HERE: MOV R0, #20                   ; counter for 1s delay (50ms*20=1sec)
         CPL P2.0                   ; complement P2.0
BACK: MOV TL1, #B0H                 ; load count value in TL1
         MOV TH1, #3CH              ; load count value in TH1
         SETB TR1                   ; start Timer1
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
         SJMP HERE                               ; repeat
```

c) Draw format of IE SFR and describe each bit.

Ans: **(2M- format, 2M –explaination)**



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, reserved 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 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.

d) Write ALP for 8051 microcontroller to transfer letter ‘ A’ serially at 4800 baud continuously.

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

Calculation :

The machine cycle frequency of 8051 = 11.0592 / 12 = 921.6 kHz, and 921.6 kHz / 32 = 28,800 Hz is frequency by UART to timer 1 to set baud rate.

28800/6=4800 where -6=FAH is loaded into TH1.

Program:

```
MOV TMOD, #20H                  ; Timer 1 , Mode 2
MOV TH1, #-6                    ; 4800 Baud
MOV SCON, #50H                  ; 8Bit ,1 Stop Bit
SETB TR1                        ; Start Timer1
AGAIN: MOV SBUF, #”A”                   ; Letter A to be transferred
HERE: JNB TI, HERE                      ; Wait for the last bit
```

```
        CLR T1                          ; Clear T1 for next char
        SJMP AGAIN                      ; Keep sending A
```

e) Describe the function of SBUF register for serial communication in 8051 microcontroller.

Ans: **(4M –explanation)**

SBUF is a 8 bit register used in serial communication of 8051.Serial data sending is done by writing to the register SBUF while data receiving is done by reading the same register. The SBUF is as shown below:

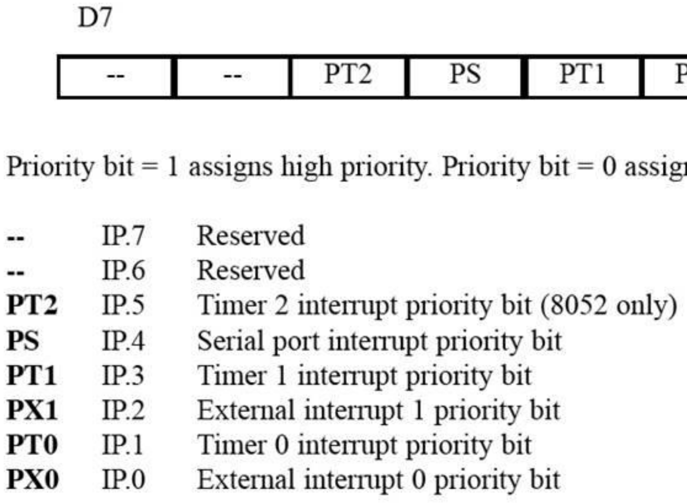
Bit 7							Bit 0

SBUF has physically two registers, one write only and other is read only. Both registers use one address 99H

Q6. Attempt any FOUR **16**

a) Draw format of IP SFR and describe each bit.

Ans: (2M- format, 2M –explanation)



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

Priority bit = 1 assigns high priority. 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

b) Write an ALP to generate continuous square wave of 2KHz frequency on pin P1.5,using Timer 0.Assume crystal freq= 11.0592MHz.

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

Crystal frequency= 12 MHz
I/P clock = (11.059 X 10^6)/12= 1000000 = 921.58KHz
T_{in} = 1.085µ sec
For 2 kHz square wave
F_{out} = 2 KHz
T_{out} = 1/ 2X 10^3
T_{out} = 0.5msec =500us
Consider half of it = T_{out} = 250µ sec
N = T_{out} / T_{in} = 250/1.085 = 230.41
65536-231= $(65305)_{10}$ = $(FF19)_{16}$

Program:-

```
            MOV TMOD, # 01H       ; Set timer 0 in Mode 1, i.e., 16 bit timer
  L2:      MOV TLO, # 19H        ; Load TL register with LSB of count
            MOV THO, # FFH        ; load TH register with MSB of count
            SETB TRO              ; start timer 0
L1:         JNB TFO, L1           ; poll till timer roll over
             CLR TRO              ; stop timer 0
             CPL P1.4             ; complement port 1.4 line to get high or
                                    low
             CLR TFO              ; clear timer flag 0
            SJMP L2               ; re-load timer with count as mode 1 is not
                                    auto reload
```

c) Differentiate between polling and interrupt approach to generate time delay by using 8051 microcontroller. (any two points).

Ans: (2M each)

Sr. no	Polling approach	Interrupt approach
1.	The timer overflow bit(TF) is continuously check and test for events as the timer runs. This testing is called as polling.	In this timer interrupts are used.Interrupts are generated when the overflow flag bit is set by the timer.
2.	It is a slower approach as compare to Interrupts	It is faster.

d) Differentiate between liner and absolute decoding technique.(any four points)

Ans: (1M each)

Absolute Decoding	Linear Decoding
1. It is also called as full decoding as all the address lines are used for decoding.	It is also called as partial decoding as all address lines are not used for decoding
2. It is used in large memory systems.	It is used in small systems
3. Hardware required for decoding logic is more	Hardware used for decoding logic is eliminated.
4. Multiple addresses are not generated	Multiple addresses are generated

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

(1.5 M- 1st and 2nd point and 1M - 3rd point)

Ans: **(1) Address Bus:** 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: 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 :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.