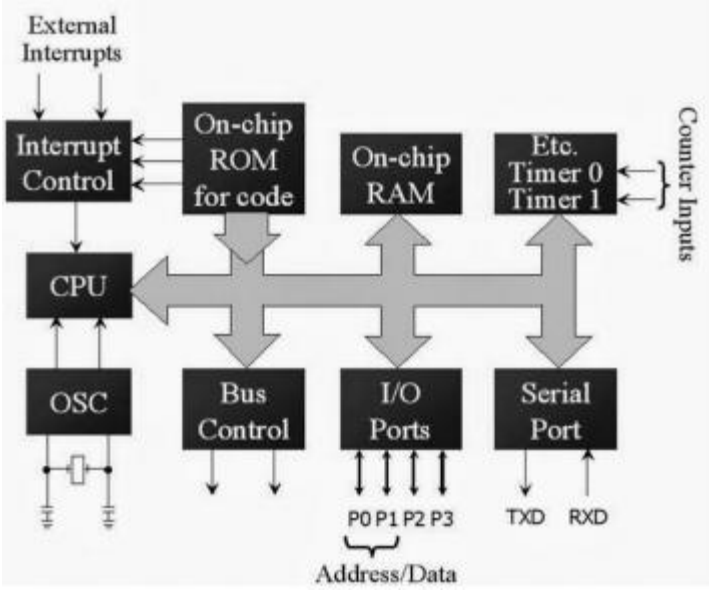
SUMMER– 19 EXAMINATION

Subject Name: Micro controller **Model Answer** Subject Code: 17534

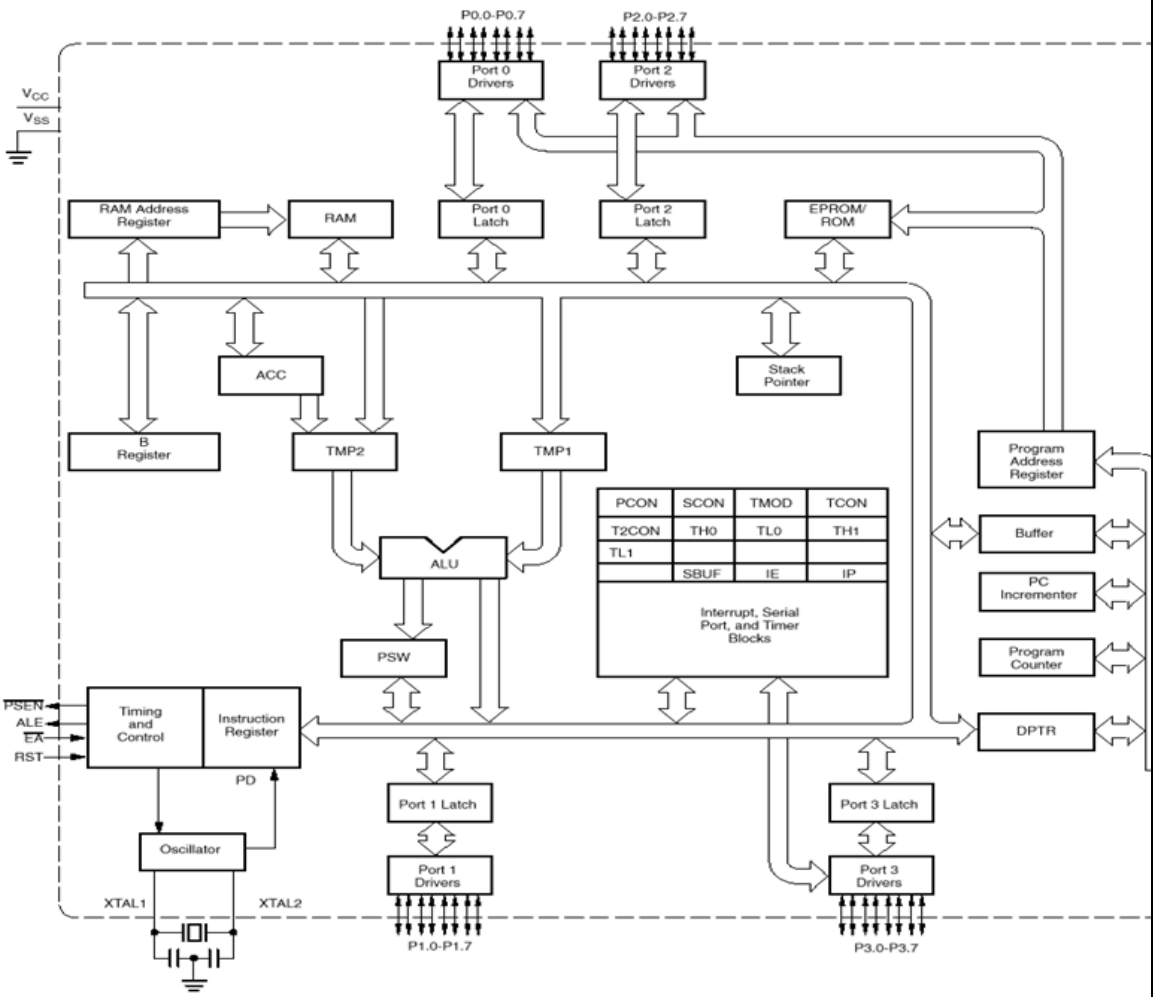
Important Instructions to examiners:

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. No.	Sub Q. N.	Answers	Marking Scheme
1	**a**	**Attempt any THREE:**	**12- Total Marks**
	i	**Draw block diagram of microcontroller.**	**4M**
	Ans:	**Block Diagram of Microcontroller:**	**Block Diagram: 4Marks**


External Interrupts
Interrupt Control
On-chip ROM for code
On-chip RAM
Etc. Timer 0 Timer 1
Counter Inputs
CPU
OSC
Bus Control
I/O Ports
Serial Port
P0 P1 P2 P3
TXD
RXD
Address/Data

OR


P0.0-P0.7
P2.0-P2.7
Port 0 Drivers
Port 2 Drivers
VCC
VSS
RAM Address Register
RAM
Port 0 Latch
Port 2 Latch
EPROM/ ROM
ACC
Stack Pointer
B Register
TMP2
TMP1
Program Address Register
PCON
SCON
TMOD
TCON
T2CON
TH0
TL0
TH1
TL1
SBUF
IE
IP
Interrupt, Serial Port, and Timer Blocks
Buffer
ALU
PC Incrementer
PSW
Program Counter
PSEN
ALE
EA
RST
Timing and Control
Instruction Register
DPTR
PD
Port 1 Latch
Port 3 Latch
Oscillator
Port 1 Drivers
Port 3 Drivers
XTAL1
XTAL2
P1.0-P1.7
P3.0-P3.7

	ii	**List different types of buses with their size in 8051 µc**	**4M**
	Ans:	<table><tr><th>BUSES</th><th>SIZE</th></tr><tr><td>Address Bus</td><td>16 Bits</td></tr><tr><td>Data Bus</td><td>8 Bits</td></tr><tr><td>Control Bus</td><td>1 Bit each control signal</td></tr></table>	**List of Bus: 1M** **Each Size: 2M**
	iii	**Why 8051 is known as Boolean processor?**	**4M**
	Ans:	**8051 is known as Boolean processor because:** • 8051 processor is a CPU that can perform some operation on a data and gives the output. • The 8051 processor contains a complete Boolean processor for single-bit operations. • The 8051 instruction set is optimized for the one bit operations. So often desired in real world, real time control applications. • The Boolean processor provides direct support for bit manipulation and testing of individual bit allows the use of single bit variable to perform logical operations therefore 8051 can be used to solve Boolean expression. • Eg: CLR C means clear the carry bit • SETB 20h means set the memory bit with bit address 20h	**Appropriate four points: 4Marks**
	iv	**Explain any four assembler directive with examples.**	**4M**
	Ans:	**1. 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. **2. 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	**Any four, Each directive: 1Mark**

		one space between DB & a byte. E.g. LOOKUP: DB 30h,31h,32h,33h,34h,35h **3. 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). **4. 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. 5. **CODE:** The CODE directive is used to assign a symbol to a program memory address. Since the maximum capacity of program memory is 64K, the address must be in the range of 0 to 65535. Syntax: Start Code **6. DW:** Define a word, puts a word (16 bit number) constant at this memory location. Syntax: Temp DW 4321H **7. DBIT:** Define a bit, defines a bit constant, which is stored in the bit addressable section of Internal RAM. **8. DATA:** By means of this directive an address with internal RAM is designated as a symbol (address must be in the range of 0-255). In other words, any selected register may change its name or be assigned a new one. Syntax: Name Data Data-address For e.g. TEMP 12 DATA 32:- register at address 32 is named as “TEMP 12”.	
V	**Compare linear and absolute decoding techniques.**		**4M**
Ans:			**Each Point: 1M. Consider Any other relevant point**

Sr. No.	Linear Decoding	Absolute Decoding
1	It is also called as partial decoding as all address lines are not used for decoding	It is also called as full decoding as all the address lines are used for decoding.
2	It is used in small systems	It is used in large memory systems
3	Hardware used for decoding logic is eliminated.	Hardware required for decoding logic is more

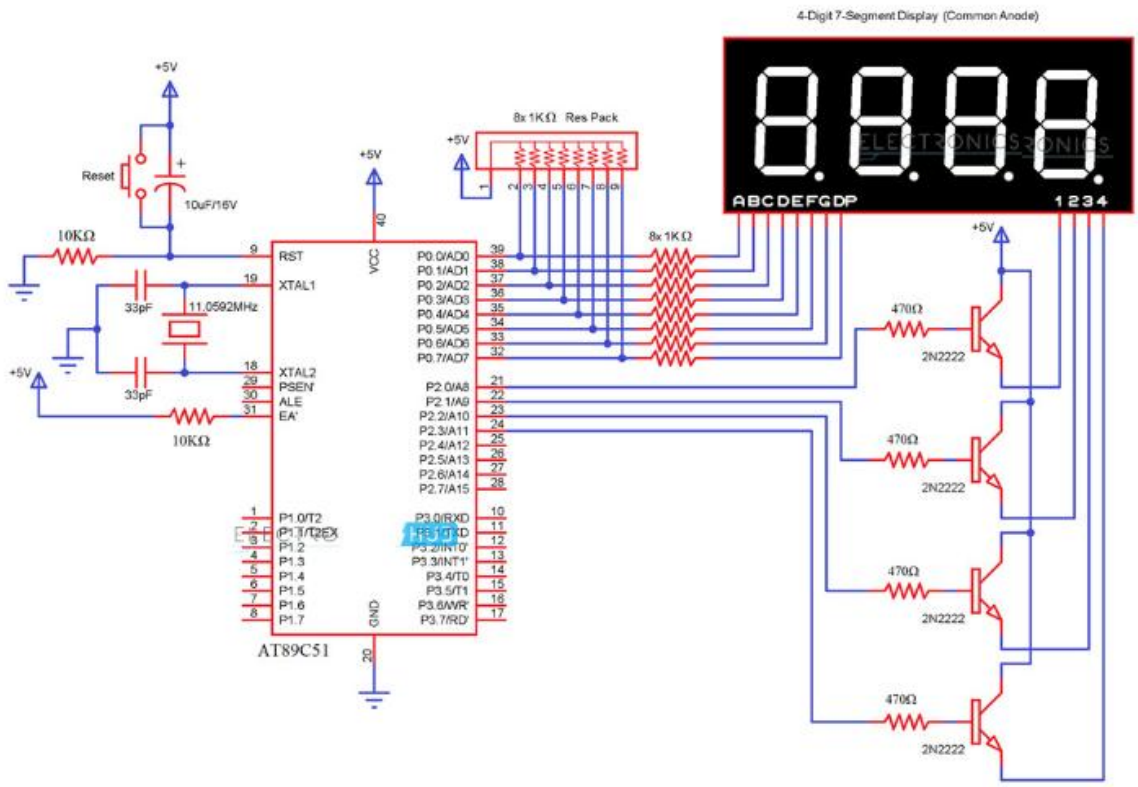
		4	Multiple addresses are generated	Multiple addresses are not generated	
	B	**Attempt any ONE:**		**6- Total Marks**
	I	**Write an ALP to find smallest number from given array of 10 bytes in internal RAM 40h onwards.**		**6M**

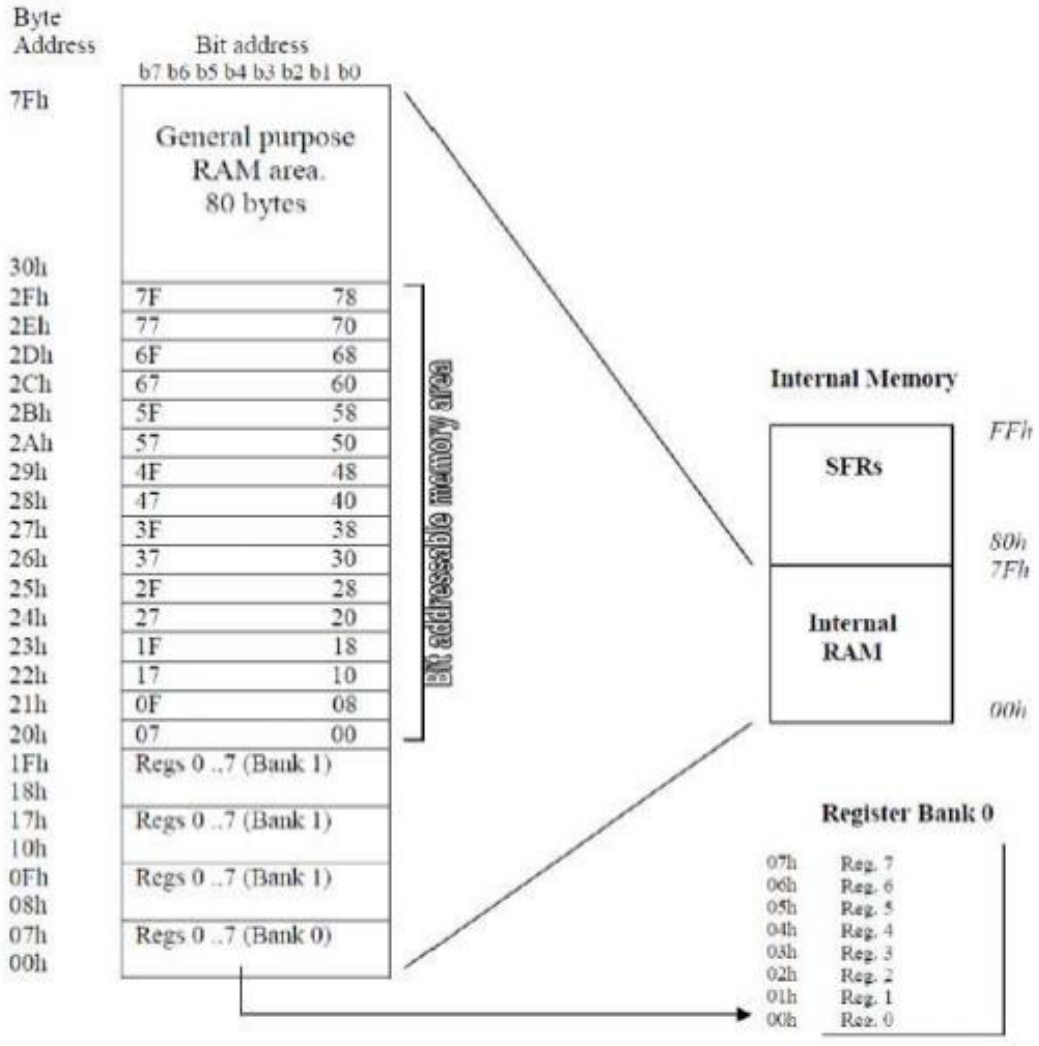
	Ans:	MOV R1, #0AH; Initialize Byte Counter
		MOV R0, #40H; Initialize source pointer R0 to 40H
		DEC R1; decrement counter by one
		MOV 60H, @R0 Read First Byte
		UP: INC R0; Increment the contents of R0
		MOV A, @R0; Read second number
		CJNE A, 60H, DN compare the first two numbers, if not equal go to DOWN
		AJMP SMALL else go to small
		DN: JNC SMALL check carry
		MOV 60H, A Store smallest number to 60H
		SMALL: DJNZ R1, UP decrement the counter by one, if count ≠ 0, then go to UP
		END
		Smallest No. is saved in memory 60H
				4M- correct prog, 2M- comments

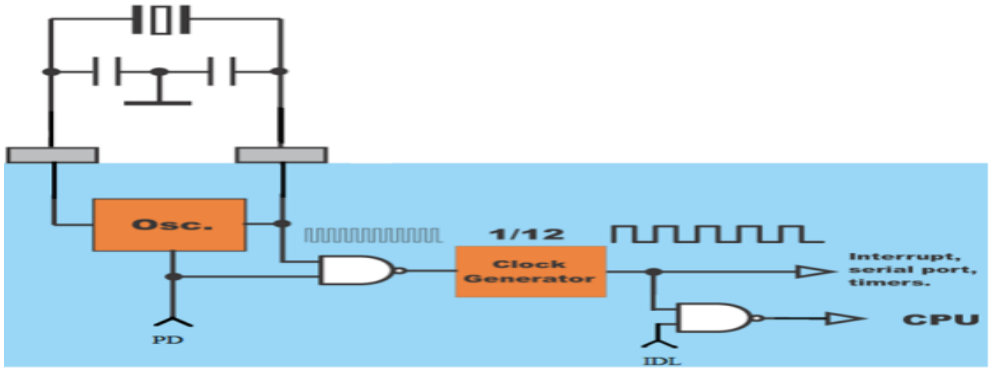
	Ii	**Interface 4- digit ,7-segment display (CA) with 8051 µc.**		**6M**

SUMMER– 19 EXAMINATION

Subject Name: Micro controller **Model Answer** Subject Code: 17534

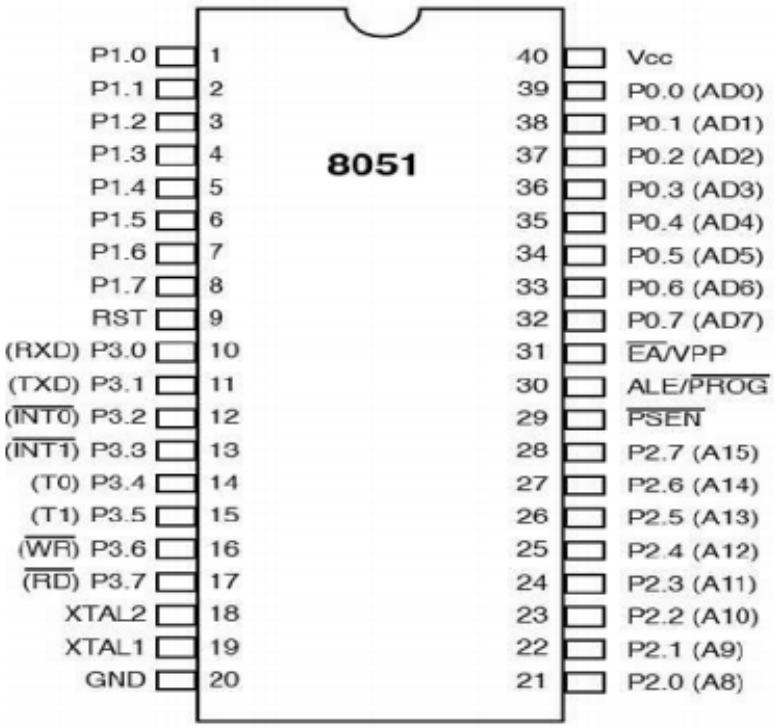
	Ans:		**6M**

Q. No.	Sub Q. N.	Answers	Marking Scheme
2		**Attempt any FOUR:**	**16- Total Marks**
	a	**Draw the structure of internal Ram of 8051 µc.**	**4M**
	Ans:		**1½M for Bank area** **1½M for Bit addressable area,** **1M for Gen. purpose RAM area**
	b	**List any eight features of 8051 µc.**	**4M**
	Ans:	Features of 8051 micro controller are as follows:- 1) 8- bit data bus and 8- bit ALU.	**Each correc**

		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	**t featur e: ½ Mark**
c	**Explain power saving modes of 8051 µc**		**4M**
Ans:	 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. IDLE MODE		2M For diagra m and PCON (Consid er the answer for full marks even if PCON is not written) 1Mark: Each 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	
	d	**Draw the pin diagram of 8051 µc.**	**4M**

Ans:



4M

e **Compare 8031,8032,8051,8052, 8751, 8752 and 89C51.** **4M**

Ans:

Parameter	8031	8032	8051	8052	8751	8752	89C51
ROM (bytes)	0	64K	4K (mask ROM)	8K (EPROM)	4K (UV-EPROM)	64K	4k
RAM (bytes)	128	256	128	256	128	256	128
Timers (16 bit)	2	3	2	3	2	3	2
I/O pins	32	32	32	32	32	32	32
Serial Port	1:UART	1:UART	1:UART	1:UART	1:UART		1:UART programmable
Interrupts	6	6	6	8	6	6	6
Watchdog timer	Yes	No	No	No	No		No
Max clock frequency (MHz.)	12	30	12	12	12		12

Any 4 parameters Each :1M

f **Compare microprocessor and microcontroller (any eight points).** **4M**

Ans:

Sr. No.	Parameter	Microprocessor	Microcontroller
1	No. of instructions 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	I/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	Single purpose(dedicated application), Automobile companies, embedded systems, remote control devices.

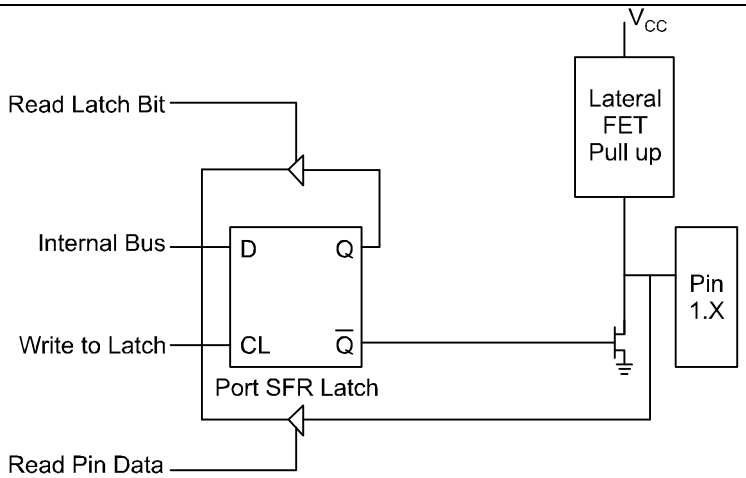
Each correct point: ½ Mark

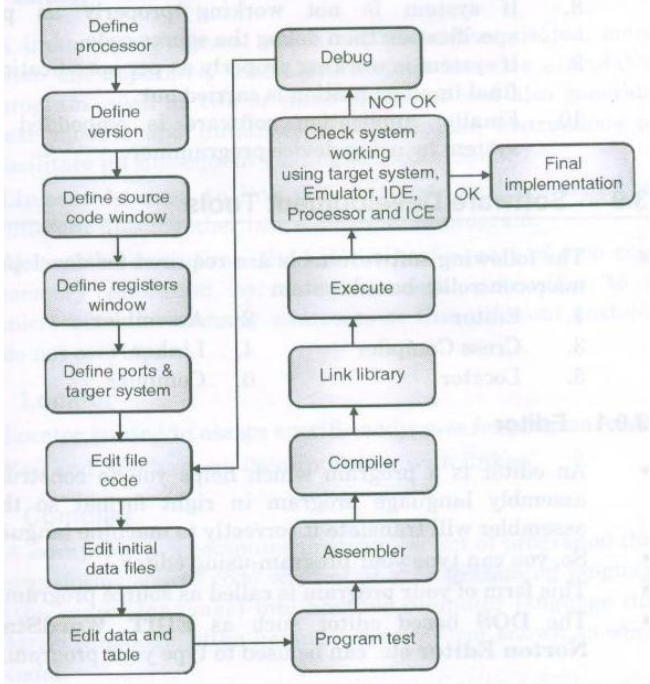
Q. No.	Sub Q. N.	Answers	Marking Scheme
3		**Attempt any FOUR:**	**16- Total Marks**
	a	**Write addressing modes for following instructions:** **(i) ADD A, R0** **(ii) ADD A, #20h** **(iii) ADD A, 50h** **(iv) ADD A, @R0**	**4M**
	Ans:	i) ADD A,R0 Register addressing mode ii) ADD A,#20h Immediate addressing mode iii) ADD A,50h Direct addressing mode iv) ADD A,@R0 Indirect addressing mode	**1 Mark for each correct answer**
	b	**Explain the following instructions:** **(i) MOVC A, @ A+DPTR** **(ii) RRC A** **(iii) SWAP A** **(iv) XCHD A, @RO**	**4M**
	Ans:	i) MOVC A, @A +DPTR Copy the contents of the external ROM address formed by adding A and the DPTR, to Accumulator(A). No flags are affected. ii) RRC A Rotate the byte right through the carry bit. Bit 0 moves into the carry flag; the original value of the carry flag moves into the 7^{th} position. No flags are affected. $D_n \rightarrow D_{n-1}$ $D_0 \rightarrow Cy$ $Cy \rightarrow D_7$ iii) SWAP A	**1 Mark for each correct explanation**

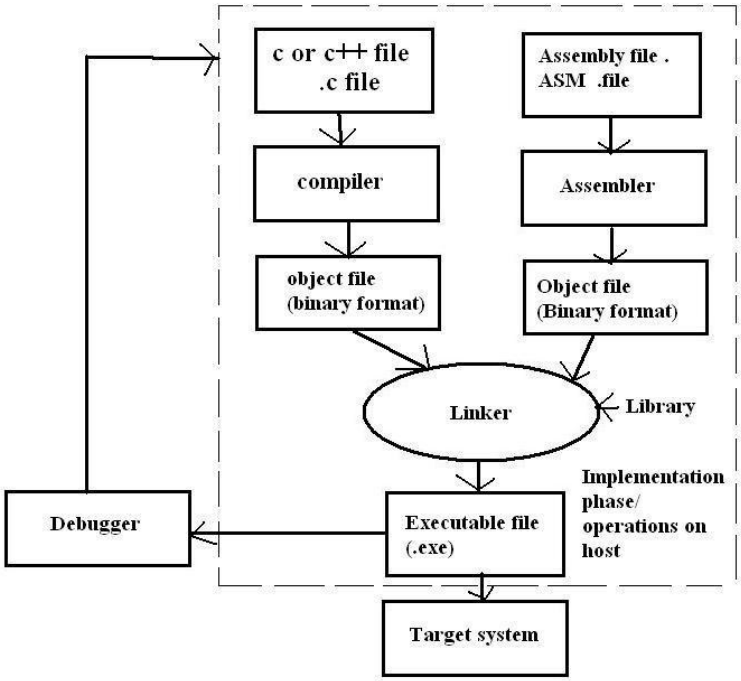
		SWAP A interchanges the low and high order nibbles of the Accumulator. No flags are affected. $[A_{low}] \leftrightarrow [A_{high}]$ iv) XCHD A,@R0 XCHD exchanges the lower order nibble of the Accumulator with that of internal RAM location indirectly addressed by the specified register. The higher order nibbles of each register are not affected, No flags are affected.	
c	**What is content of A after execution of following program:** **CLR A** **CPL A** **ANL A,# C0h** **RR A**		**4M**
Ans:	CLR A (A)=00 h CPL A (A)=FF h ANL A,#0Ch (A)=[A] AND C0 h = C0h RR A (A)=60h		**1 mark for each correct answer**
d	**Write an ALP to add array of 10 bytes stores at 50h onwards. Store result in external RAM location 2000h (LSB) and 2001h (MSB).**		**4M**

Ans:	ORG 0000h
	Start: MOV R0, #0AH ; Initialize byte counter
	MOV R1, #50H ; Initialize memory pointer
	MOV R7,#00H ; Initialize higher byte counter
	MOV DPTR, 2000H ;Initialize external memory pointer
	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)
			4M for correct programme Consider any other program which

		Next: INC R1 ; Increment memory pointer
		DJNZ R0, UP ; Decrement byte counter, if not zero add again
		MOVX @DPTR, A ; Store lower byte of result in External memory
		INC DPTR ; Increment DPTR
		MOV A, R7 ;Mov the higher byte to accumulator
		MOVX @DPTR, A ; Store higher byte of result in internal memory
		HERE: SJMP HERE ; Stop
		END
			is logically correct

	e	**Draw internal structure of port 1 pin.**	**4M**
	Ans:		**4 Marks for correct diagram**

Q. No.	Sub Q. N.	Answers	Marking Scheme
4	**a**	**Attempt any THREE:**	**12- Total Marks**
	i	**Draw software development cycle.**	**4M**
	Ans:	 **Software development cycle** **OR**	**4 Marks for correct diagram**



ii **List modes of serial communications with name and data size.** **4M**

Ans:

SM0	SM1	Mode	Description/Name	Data size	Baud Rate
0	0	Mode 0	8-bit shift register mode	8 bits	$F_{osc}/12$
0	1	Mode 1	8-bit UART	10 bits	Variable (Set by timer 1)
1	0	Mode 2	9-bit UART	11 bits	$F_{ose}/64$ or $F_{osc}/32$
1	1	Mode 3	9-bit UART	11 bits	Variable (Set by timer 1)

OR

Mode	Description/Name	Data size
Mode 0	8-bit shift register mode	8 bits
Mode 1	8-bit UART	10 bits
Mode 2	9-bit UART	11 bits
Mode 3	9-bit UART	11 bits

1M for each mode

iii **Write an ALP to get byte from P_0 and P_1 add it and send it to P_2.** **4M**

Ans:			**4 Marks for correct program**

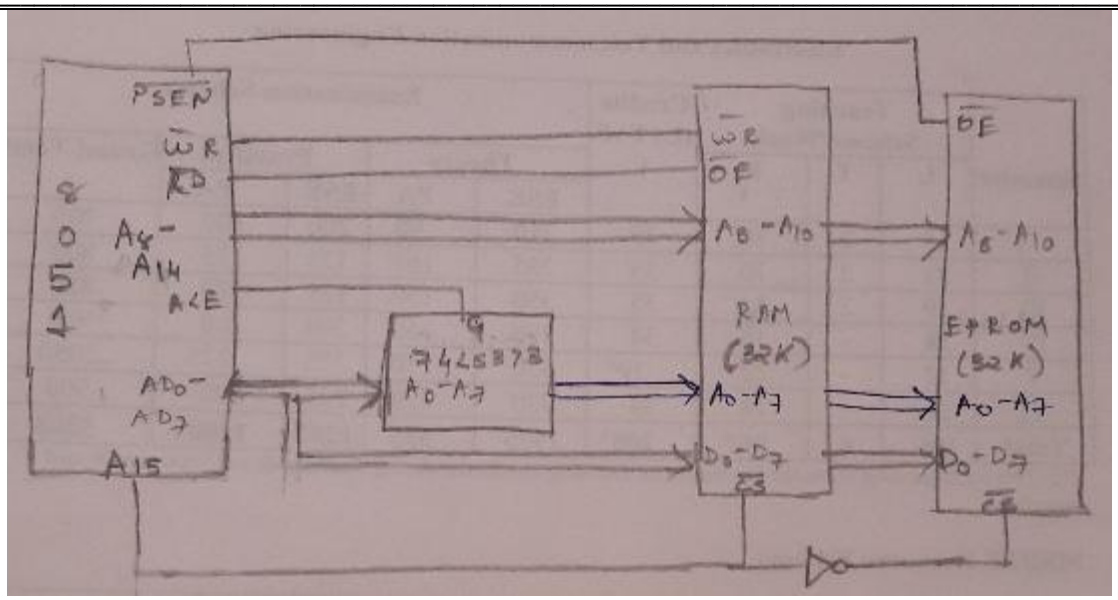
MOV A,P0	Get a byte from P0
MOV B,A	Transfer the byte to B register
MOV A,P1	Get the second byte from P1 to A
ADD A,B	Add both the bytes
MOV P2,A	Send the result to P2

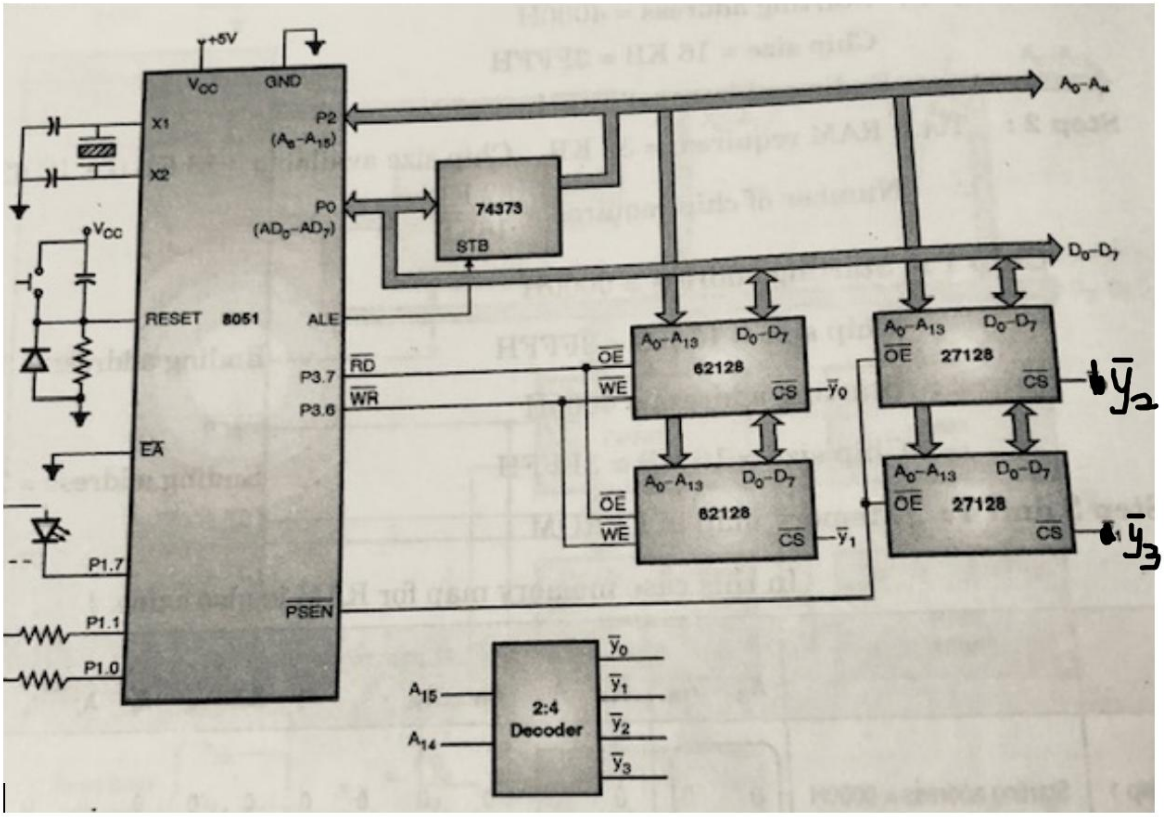
iv	**Write an ALP to send, MSBTE on TXD line with crystal frequency = 11.0592 MHz and to have 9600 baud rate.**		**4M**
Ans:	MOV TMOD, #20H	; timer 1, mode2	**3M- logically correct Prog,** **1M- comments**
	MOV TH1,#-3	; 9600 baud rate	
	MOV SCON, #50H	; 8-bit data,1 stop bit, REN enabled	
	SETB TR1	; Start timer 1	
	AGAIN: MOV A, #"M"	; transfer "M"	
	ACALL MESSAGE	; Some delay	
	MOV A, #"S"	; transfer "S"	
	ACALL MESSAGE		
	MOV A, #"B"	; transfer "B"	
	ACALL MESSAGE		
	MOV A, #"T"	; transfer "T"	
	ACALL MESSAGE		
	MOV A, #"E"	; transfer "E"	
	ACALL MESSAGE		
	SJMP AGAIN		
	MESSAGE: MOV SBUF, A;		
	JNB T1, HERE;		

		CLR T1; RET	
	b	Attempt any ONE:	**6- Total Marks**
	i	**Write an ALP to arrange 10 bytes starting from 50h onwards, in descending order.**	**6M**

	Ans:	ORG 0000h
		MOV R0,#50H ; Initialize memory pointer
		MOV R2, #0AH ;initialize counter 1
		UP2: MOV R1, #50H ;Initialize the memory pointer
		MOV R3,#0AH ; initialize counter 2
		UP1: MOV A, @R0 ;get the first byte from memory pointer R0
		MOV 0F0H, @ R1 ;get the first byte from memory pointer R1
		CJNE A,0F0H,DOWN ;compare the two bytes
		DOWN:JC A_SMALL ;if carry is 1, ie R0 data is smaller go to A_SMALL
		MOV @R0,0F0H ; if Carry =0, get smaller data to R0 from R1
		MOV @R1,A
		A_SMALL: INC R1 ; increment R1
		DJNZ R3,UP1 ; decrement R3 and repeat till it is 0
		INC R0 ; increment R0
		DJNZ R2,UP2 ; decrement R2 and repeat till it is 0
		SJMP $
			6M

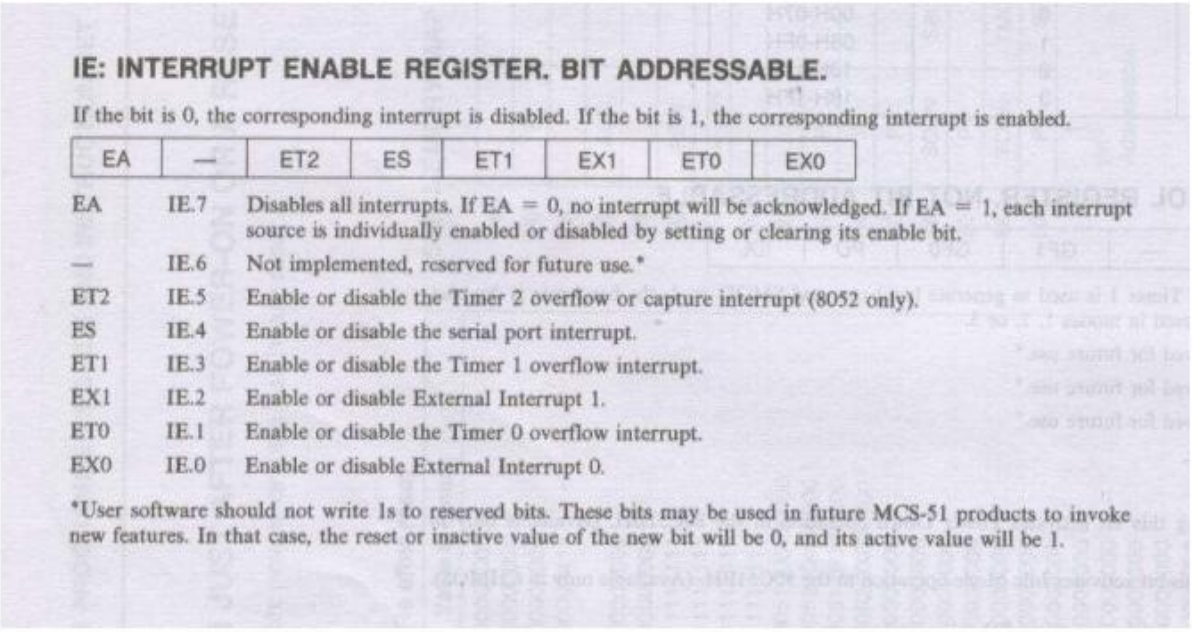
	ii	**Interface 32KB RAM and 32KB EPROM with 8051 microcontroller**	**6M**

	Ans:		Correct interfacing: 3M for RAM & 3M for EPROM
	iii	Draw address map table for 32KB RAM using 16 KB Chips and 32 KB EPROM using 16KB Chips . Draw decoder circuit.	6M
	Ans:		



Memory map for EPROM and RAM is same

	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Start address of 1st RAM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
End address of 1st RAM	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH
Start address of 2nd RAM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000H
End address of 2nd RAM	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFFH
Start address of 1st EPROM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H
End address of 1st EPROM	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	BFFFH
Start address of 2nd EPROM	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000H
End address of 2nd EPROM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFH

Q. No.	Sub Q. N.	Answers	Marking Scheme
5		**Attempt any FOUR:**	**16- Total Marks**
	a	**Draw format of IE and IP register.**	**4M**
	Ans :	**FORMAT OF IE:** 	**2M- IE, 2M-IP**

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.

FORMAT OF IP:

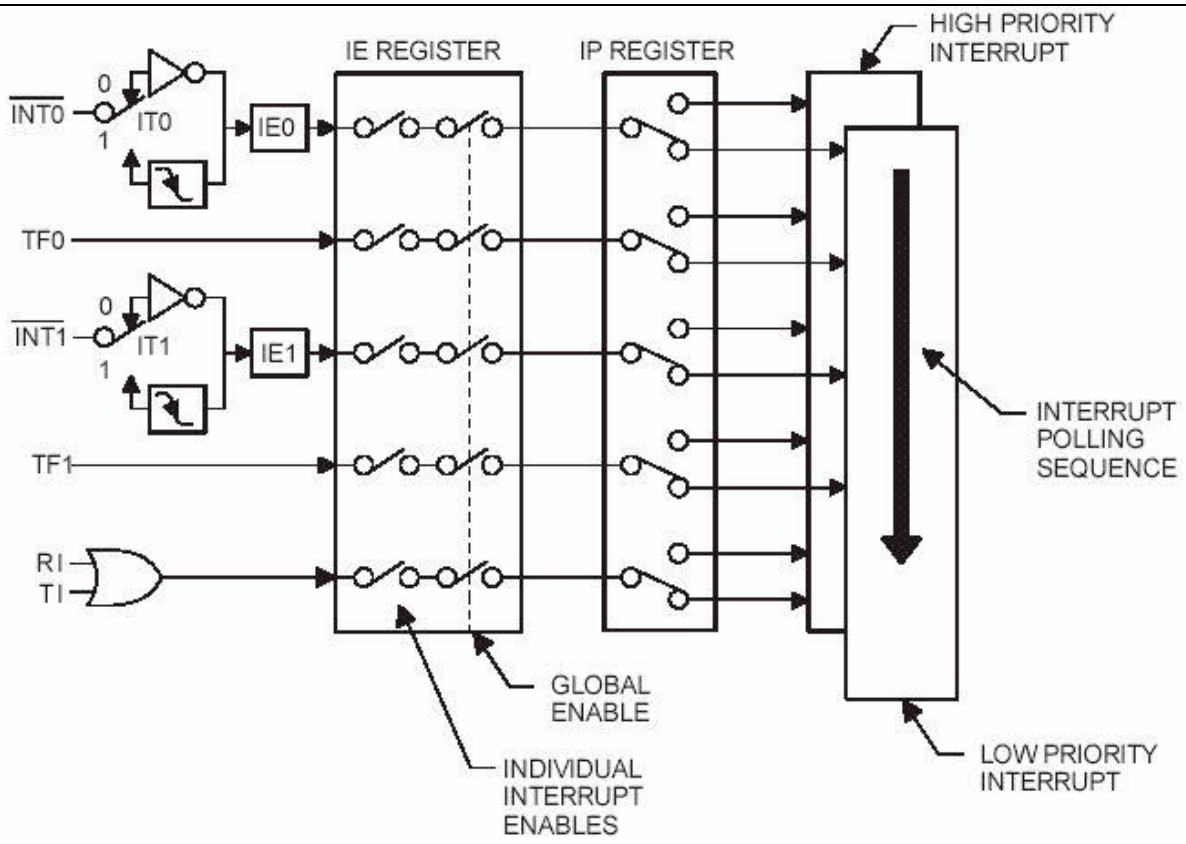
D7							D0
--	--	PT2	PS	PT1	PX1	PT0	PX0

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

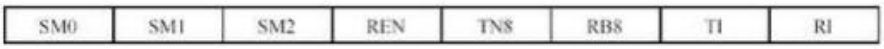
--	IP.7	Reserved
--	IP.6	Reserved
PT2	IP.5	Timer 2 interrupt priority bit (8052 only)
PS	IP.4	Serial port interrupt priority bit
PT1	IP.3	Timer 1 interrupt priority bit
PX1	IP.2	External interrupt 1 priority bit
PT0	IP.1	Timer 0 interrupt priority bit
PX0	IP.0	External interrupt 0 priority bit

b **Draw interrupt structure of 8051 µc.** **4M**

Ans : **4M**



c **Explain sequence of operations after arrival of interrupt.** **4M**

	Ans :	1. CPU finishes the instruction it is currently executing and stores the PC on the stack. 2. CPU saves the current status of all interrupts internally. 3. Fetches the ISR address for the interrupt from IVT and jumps to that address. 4. Executes the ISR until it reaches the RETI instruction. 5. Upon RETI, the CPU pops back the old PC from the stack and continues with whatever it was doing before the interrupt occurred.	**4M**
	d	**Draw the format of SCON register and explain it.**	**4M**
	Ans :	**SCON : Serial Port Control Register (Bit Addressable)**  SM0 SCON.7 Serial Port mode specifier (NOTE 1). SM1 SCON.6 Serial Port mode specifier (NOTE 1). SM2 SCON.5 Enables the multiprocessor communication feature in mode 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0 (See table 9). REN SCON.4 Set/Cleared by software to Enable/Disable reception. TB8 SCON.3 The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software. RB8 SCON.2 In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used. TI SCON.1 Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software. RI SCON.0 Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or half way through the stop bit time in the other modes (except see SM2). Must be cleared by software. **Note 1 :** SM0 SM1 MODE DESCRIPTION BAUD RATE 0 0 0 SHIFT REGISTER Fosc /12 0 1 1 8 bit UART Variable 1 0 2 8 bit UART Fosc /64 OR Fosc /32 1 1 3 8 bit UART Variable	**2M-FORMAT, 2M-EXPLANATION**
	e	**Draw the format of PCON register. How to double baud rate.**	**4M**

	Ans :	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. To double the baud rate the value of SMOD should be 1.	**3M- FORMAT, 1M- BAUDRATE**

Q. No.	Sub Q. N.	Answers	Marking Scheme
6		**Attempt any FOUR:**	**16- Total Marks**
	a	**Write an ALP for 500 msec delay, fosc = 12 MHz .**	**4M**
	Ans:	**CALCULATION:** Crystal freq=12MHz Timer frequency=12MHz/12=1MHz Timer Time Period=1/1MHz=1µs For 500msec delay, 50msec is repeated 10 times For delay of 50 ms, Count = 50ms/1µs=50000 Therefore count to be loaded in TH1 and TL1 can be calculated as 65536- 50000=15535D=3CB0H **PROGRAM:** MOV TMOD, #10H ;Timer1, mode 1	**1M - calculation, 3M- Program**

```
MOV R0,#0AH      ;Counter for 500ms(50*10)DELAY
BACK: MOV TL1, # B0H    ; load count value in TL1
MOV TH1, #3CH           ; load count value in TH1
SETB TR1                ;start Timer 1
AGAIN: JNB TF1, AGAIN   ; stay until timer rolls over
CLR TR1                 ; stop timer
CLR TF1                 ; clear timer flag
DJNZ R0, BACK           ;decrement R0,  if R0 ≠ 0, reload timer
HERE: SJMP HERE         ; repeat
```

b **Compare Timer and Counter operation.** **4M**

Ans: **Any 4 points-1M each**

SR.NO	TIMER	COUNTER
1	It is used for delay implementation	It is used to measure external event such as pulse width measurement
2	External pin T0/T1 is not used	External pin T0/T1 is monitored
3	Counter is incremented for 1/12th of Fosc	Counter is incremented for every pulse on T0 or T1 pin, irrespective of it's frequency
4	THx and TLx is loaded as per required Delay	THx and TLx must be initially cleared for counter
5	C/T=0	C/T=1

c **Draw format of CWR for BSR and I/O Mode of 8255.** **4M**

Ans: **2M- BSR Mode, 2M- I/O mode**

BSR MODE:

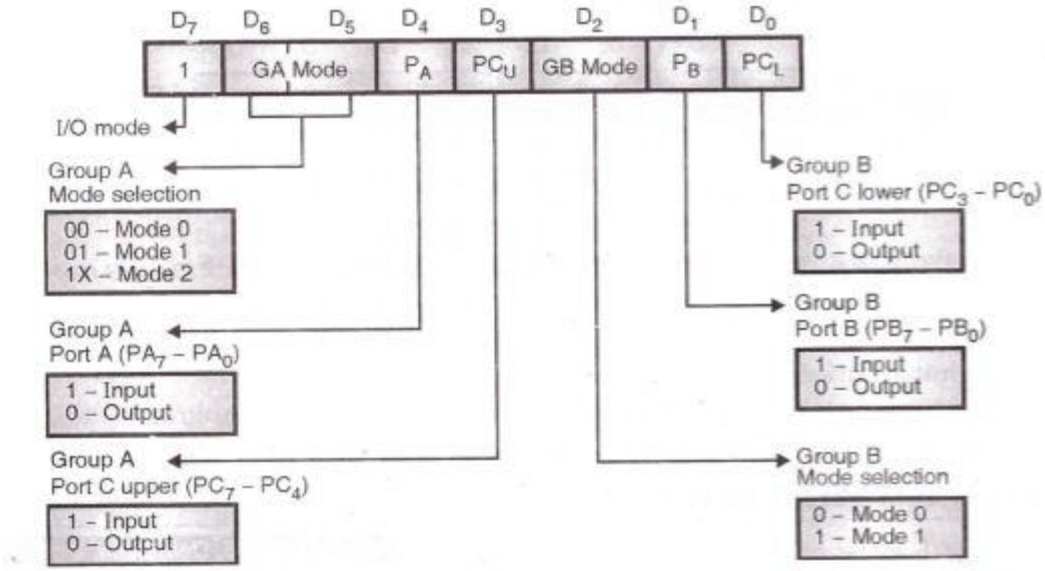
D7	D6	D5	D4	D3	D2	D1	D0
0 (0=BSR)	X	X	X	B2	B1	B0	S/R (1=S,0=R)

Bit select: (Taking Don't care's as 0)

B2	B1	B0	PC bit	Control word (Set)	Control word (reset)
0	0	0	PC0	0000 0001 = 01h	0000 0000 = 00h
0	0	1	PC1	0000 0011 = 03h	0000 0010 = 02h
0	1	0	PC2	0000 0101 = 05h	0000 0100 = 04h

0	1	1	PC3	0000 0111 = 07h	0000 0110 = 06h
1	0	0	PC4	0000 1001 = 09h	0000 1000 = 08h
1	0	1	PC5	0000 1011 = 0Bh	0000 1010 = 0Ah
1	1	0	PC6	0000 1101 = 0Dh	0000 1100 = 0Ch
1	1	1	PC7	0000 1111 = 0Fh	0000 1110 = 0Eh

I/O MODE:



I/O modes control word format

	d	**List interrupts with their priority and vector addresses.**	**4M**
	Ans:		**2M- list, 1M-vector address,1M priority**

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

	e	**Explain selection factor of microcontroller.**	**4M**
	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 correct selection of microcontroller.	**Any 4 points- 1M each**