USN | | | | | | | | | | |

INTERNAL ASSESSMENT TEST – I

Sub:	Microcontroller							Code:	BEC405A
Date:	05/06/24	Duration:	90 mins	Max Marks:	50	Sem:	V	Branch:	ECE

Answer any 5 full questions

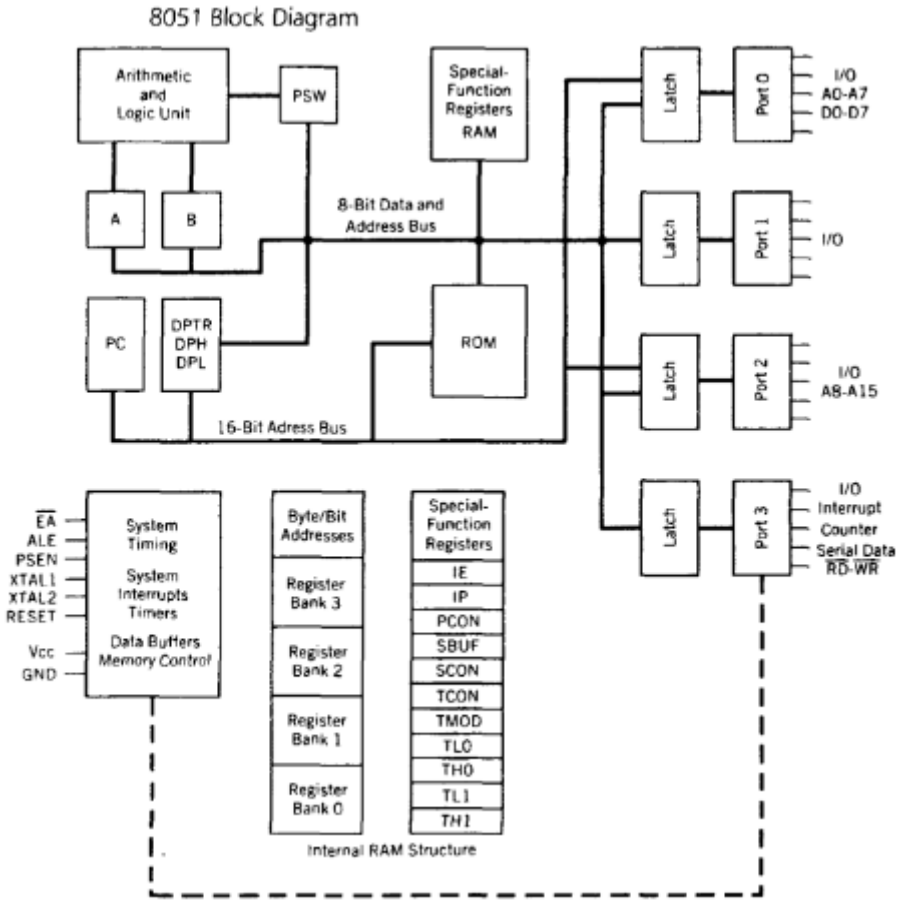
		Marks	CO	RBT
1	With neat diagram, explain the internal architecture of 8051. Explain the CPU registers.	[10]	CO1	L2
2	Name the bit addressable SFRs present in 8051 with addresses.	[10]	CO1	L2
3	(a) Compare microprocessor with microcontrollers. (b) List out the differences between CISC and RISC.	[05] [05]	CO1	L2
4	Explain with functional block diagram 'Port 0' and 'Port 1'of 8051.	[10]	CO1	L2

		Marks	CO	RBT
5	List bit level logical instructions and their operations in 8051	[10]	CO2	L2
6	Explain the following instructions i) MOVX A,@dptr ii) RRC A iii) RLC A iv) DIV AB	[10]	CO2	L2
7	With a neat diagram, explain the steps to interface 8KB of program RAM and 4 KB of ROM to 8031 based systems.	[10]	CO1	L2
8	Write an ALP to find the number of positive and negative numbers in a given array of ten bytes of data, the number is available from memory location 8000H	[10]	CO2	L3

CI CCI HoD/ECE

Solution

1. With neat diagram, explain the internal architecture of 8051. Explain the CPU registers.



PSW Program Status Word Register

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	—	P

THE PROGRAM STATUS WORD (PSW) SPECIAL FUNCTION REGISTER

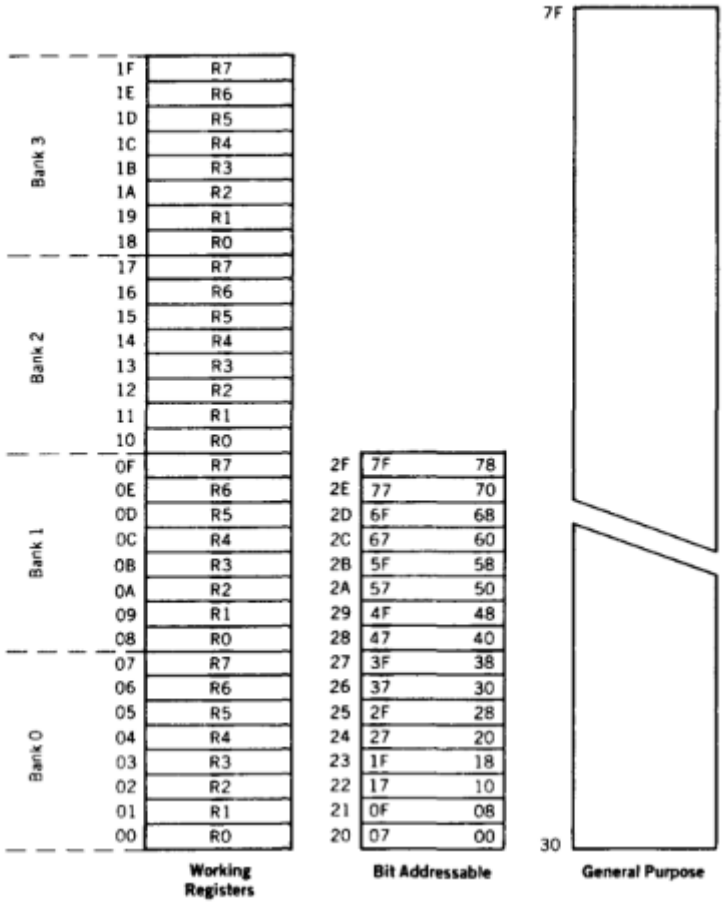
Bit	Symbol	Function
7	CY	Carry flag; used in arithmetic, JUMP, ROTATE, and BOOLEAN instructions
6	AC	Auxilliary carry flag; used for BCD arithmetic
5	F0	User flag 0
4	RS1	Register bank select bit 1
3	RS0	Register bank select bit 0

RS1	RS0	
0	0	Select register bank 0
0	1	Select register bank 1
1	0	Select register bank 2
1	1	Select register bank 3

2	OV	Overflow flag; used in arithmetic instructions
1	—	Reserved for future use
0	P	Parity flag; shows parity of register A: 1 = Odd Parity

Bit addressable as PSW.0 to PSW.7

Internal RAM Organization



The figure also shows the usual CPU components: program counter, ALU, working registers, and clock circuits.[1]

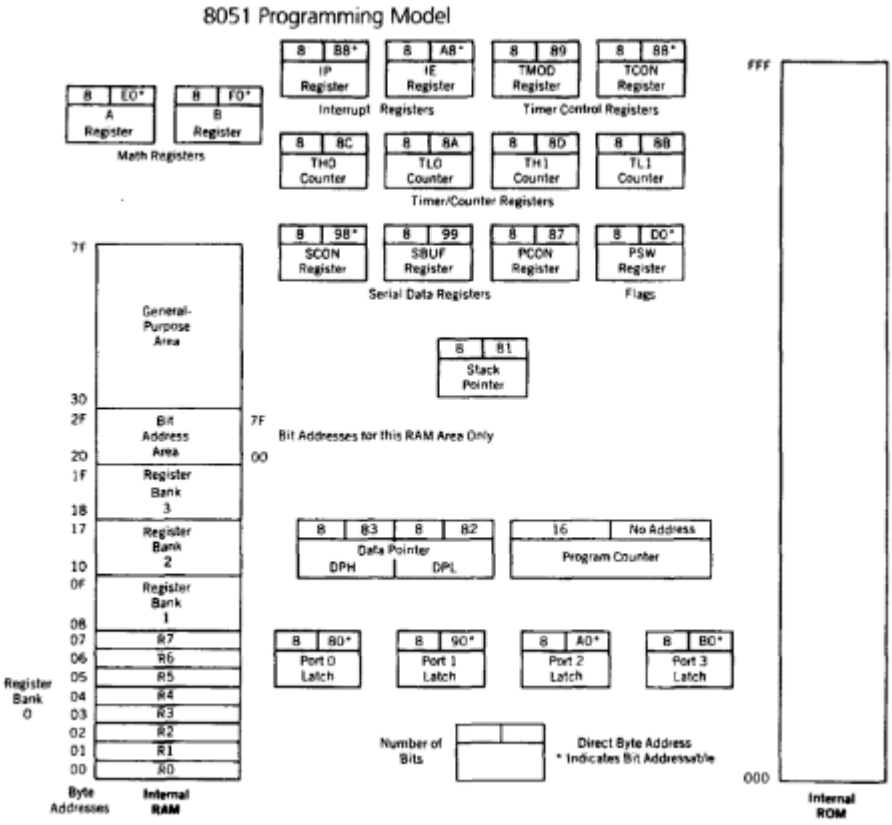
The 8051 architecture consists of these specific features:

- Eight-bit CPU with registers A (the accumulator) and B
- Sixteen-bit program counter (PC) and data pointer (DPTR)
- Eight-bit program status word (PSW)
- Eight-bit stack pointer (SP)
- Internal ROM or EPROM (8751) of 0 (8031) to 4K (8051)
- Internal RAM of 128 bytes:
  - Four register banks, each containing eight registers
  - Sixteen bytes, which may be addressed at the bit level
  - Eighty bytes of general-purpose data memory
- Thirty-two input/output pins arranged as four 8-bit ports: P0–P3
- Two 16-bit timer/counters: T0 and T1
- Full duplex serial data receiver/transmitter: SBUF
- Control registers: TCON, TMOD, SCON, PCON, IP, and IE
- Two external and three internal interrupt sources
- Oscillator and clock circuits

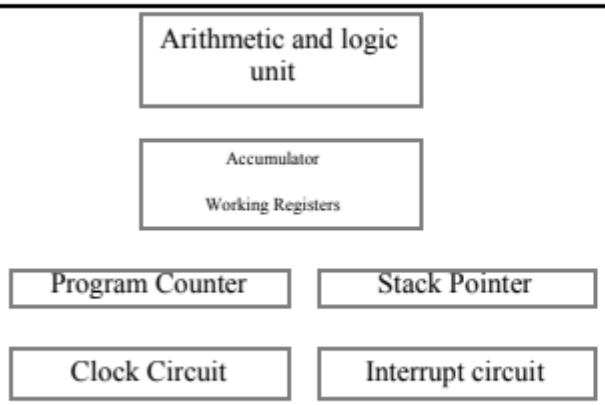
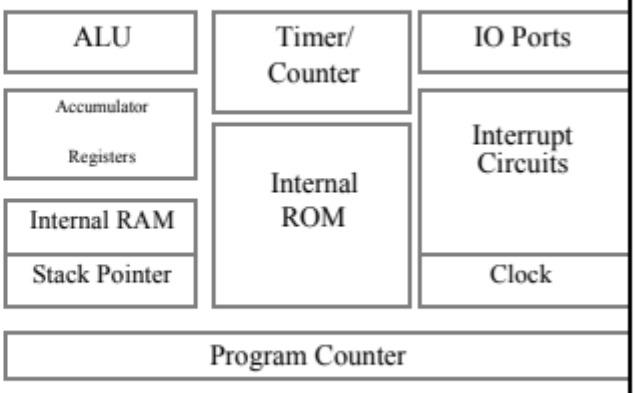
2. Name the bit addressable SFRs present in 8051 with addresses.

Special Function Registers

NAME	FUNCTION	INTERNAL RAM ADDRESS (HEX)
A	Accumulator	0E0
B	Arithmetic	0F0
DPH	Addressing external memory	83
DPL	Addressing external memory	82
IE	Interrupt enable control	0A8
IP	Interrupt priority	0B8
P0	Input/output port latch	80
P1	Input/output port latch	90
P2	Input/output port latch	A0
P3	Input/output port latch	0B0
PCON	Power control	87
PSW	Program status word	0D0
SCON	Serial port control	98
SBUF	Serial port data buffer	99



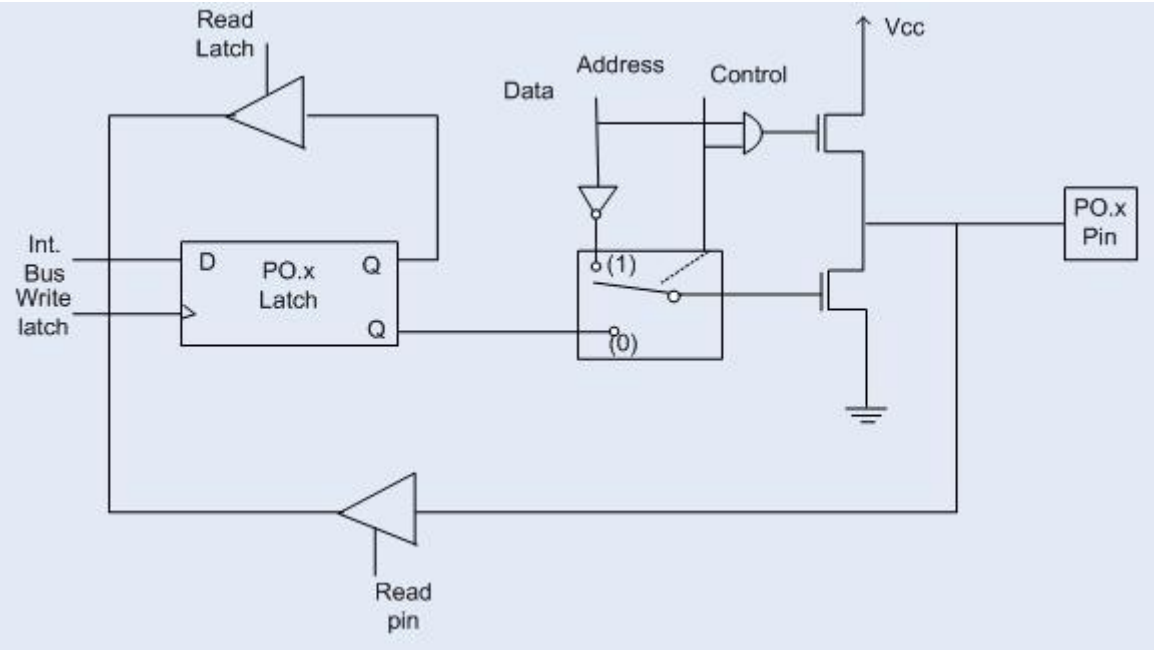
3.(a) Compare microprocessor with microcontroller.

Microprocessor	***Microcontroller***
	
Block diagram of microprocessor	*Block diagram of microcontroller*
Microprocessor contains ALU, General purpose registers, stack pointer, program counter, clock timing circuit, interrupt circuit	Microcontroller contains the circuitry of microprocessor, and in addition it has built in ROM, RAM, I/O Devices, Timers/Counters etc.
It has many instructions to move data between memory and CPU	It has few instructions to move data between memory and CPU
Few bit handling instruction	It has many bit handling instructions
Less number of pins are multifunctional	More number of pins are multifunctional
Single memory map for data and code (program)	Separate memory map for data and code (program)
Access time for memory and IO are more	Less access time for built in memory and IO.
Microprocessor based system requires additional hardware	It requires less additional hardwares
More flexible in the design point of view	Less flexible since the additional circuits which is residing inside the microcontroller is fixed for a particular microcontroller
Large number of instructions with flexible addressing modes	Limited number of instructions with few addressing modes

3.b. List out the differences between CISC and RISC.

RISC	CISC
Instruction takes one or two cycles	Instruction takes multiple cycles
Only load/store instructions are used to access memory	In additions to load and store instructions, memory access is possible with other instructions also.
Instructions executed by hardware	Instructions executed by the micro program
Fixed format instruction	Variable format instructions
Few addressing modes	Many addressing modes
Few instructions	Complex instruction set
Most of the have multiple register banks	Single register bank
Highly pipelined	Less pipelined
Complexity is in the compiler	Complexity in the microprogram

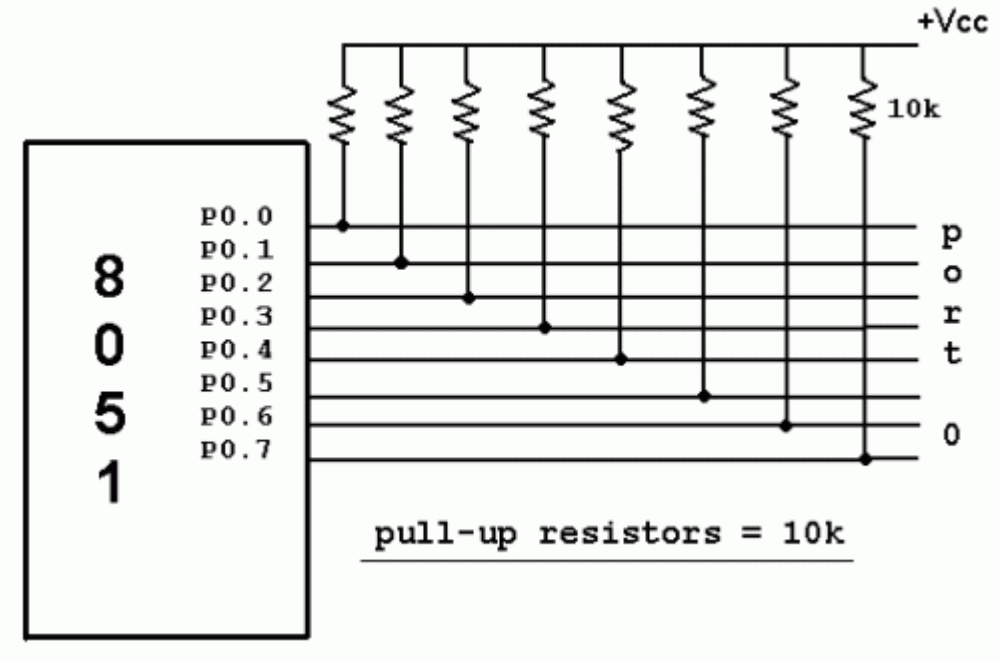
4.Explain with functional block diagram 'Port 0' and 'Port 1'of 8051.



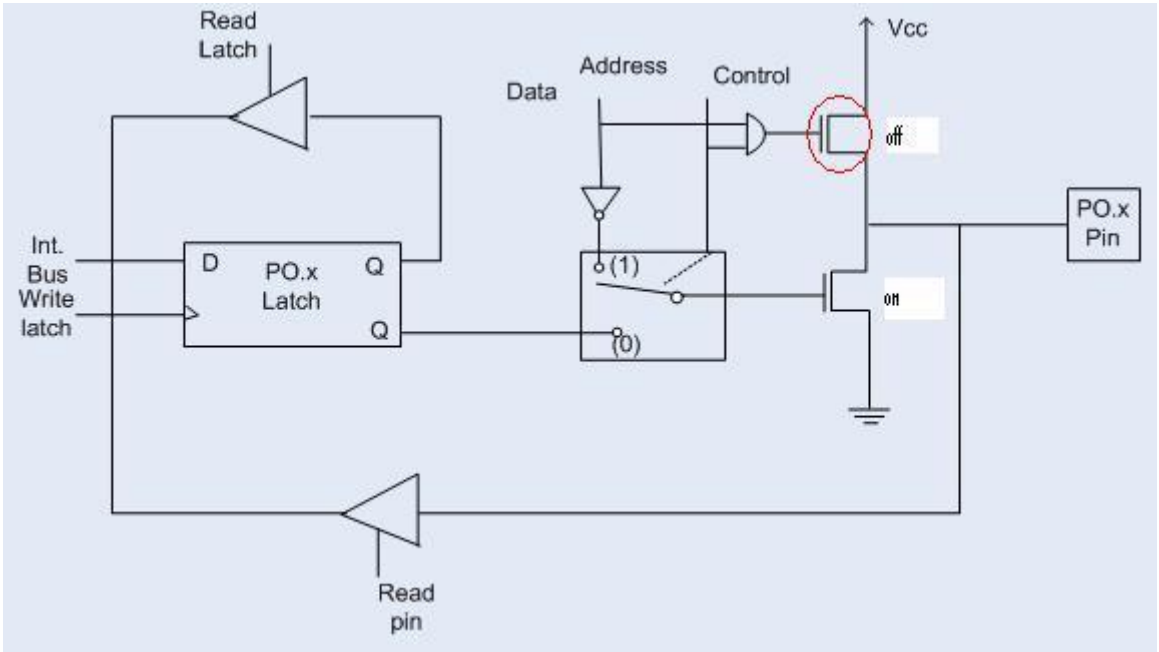
PORT 0

PORT 0 as an Output Port

Suppose we want to write 1 on pin of Port 0, a '1' written to the latch which turns 'off' the lower FET while due to '0' control signal upper FET also turns off as shown in fig. above. Here we wants logic '1' on pin but we getting floating value so to convert that floating value into logic '1' we need to connect the pull up resistor parallel to upper FET . This is the reason **why we needed to connect pull up resistor to port 0 when we want to initialize port 0 as an output port** .



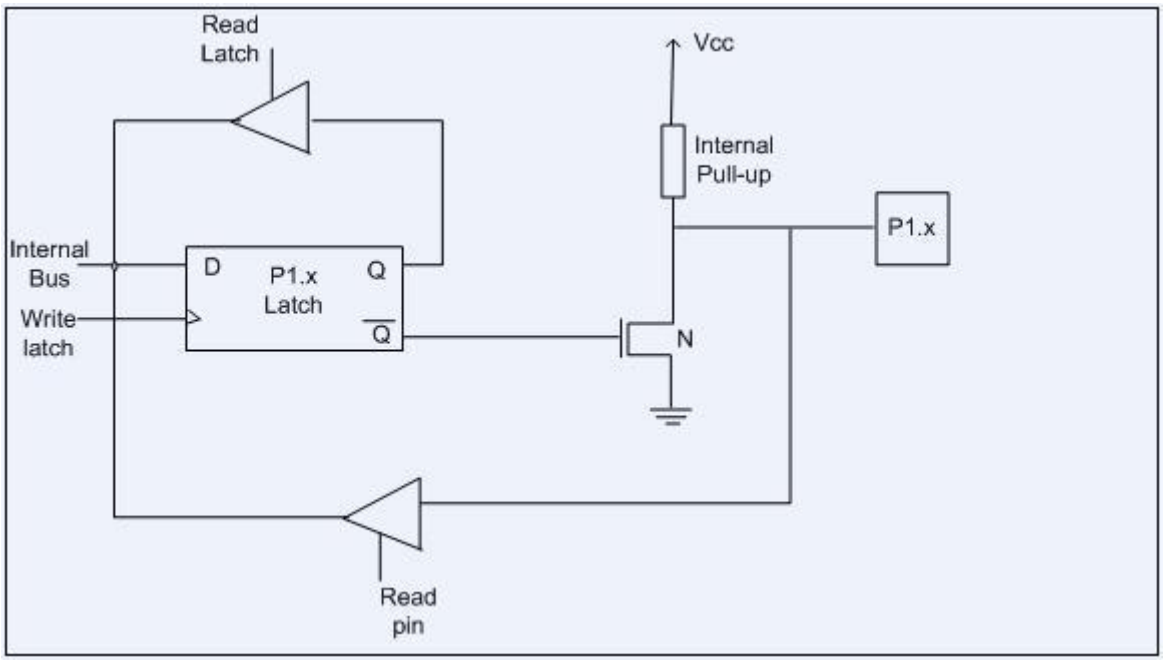
If we want to write '0' on pin of port 0 , when '0' is written to the latch, the pin is pulled down by the lower FET. Hence the output becomes zero.



When the control is '1', address/data bus controls the output driver FETs. If the address/data bus (internal) is '0', the upper FET is 'off' and the lower FET is 'on'. The output becomes '0'. If the address/data bus is '1', the upper FET is 'on' and the lower FET is 'off'. Hence the output is '1'. Hence for normal address/data interfacing (for external memory access) no pull-up resistors are required.Port-0 latch is written to with 1's when used for external memory access.

PORT 1:

The structure of a port-1 pin is shown in fig below.It has 8 pins (P1.1-P1.7) .



Port-1 dedicated only for I/O interfacing. When used as output port, not needed to connect additional pull-up resistor like port 0. It have provided internally pull-up resistor as shown in fig. below. The pin is pulled up or down through internal pull-up when we want to initialize as an output port. To use port-1 as input port, '1' has to be written to the latch. In this input mode when '1' is written to the pin by the external device then it read fine. But when '0' is written to the pin by the external device then the external source must sink current due to internal pull-up. If the external device is not able to sink the current the pin voltage may rise, leading to a possible wrong reading.

5. List bit level logical instructions and their operations in 8051

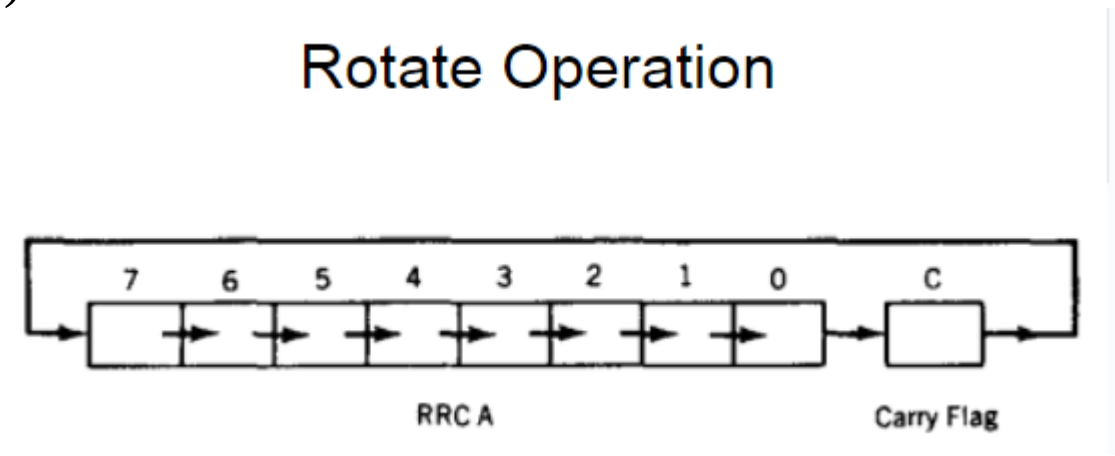
Mnemonic	Operation
ANL C,b	AND C and the addressed bit; put the result in C
ANL C,/b	AND C and the complement of the addressed bit; put the result in C; the addressed bit is not altered
ORL C,b	OR C and the addressed bit; put the result in C
ORL C,/b	OR C and the complement of the addressed bit; put the result in C; the addressed bit is not altered
CPL C	Complement the C flag
CPL b	Complement the addressed bit
CLR C	Clear the C flag to zero
CLR b	Clear the addressed bit to zero
MOV C,b	Copy the addressed bit to the C flag
MOV b,C	Copy the C flag to the addressed bit
SETB C	Set the flag to one
SETB b	Set the addressed bit to one

6. Explain the following instructions i) MOVX A,@dptr ii) RRC A iii) RLC A iv) DIV AB

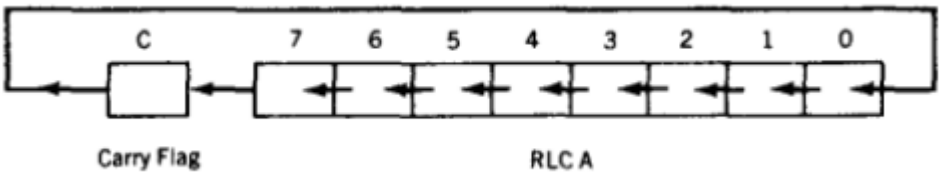
i) MOVX A,@dptr

MOVX A,@DPTR	Copy the contents of the external address in DPTR to A

ii) RRC A



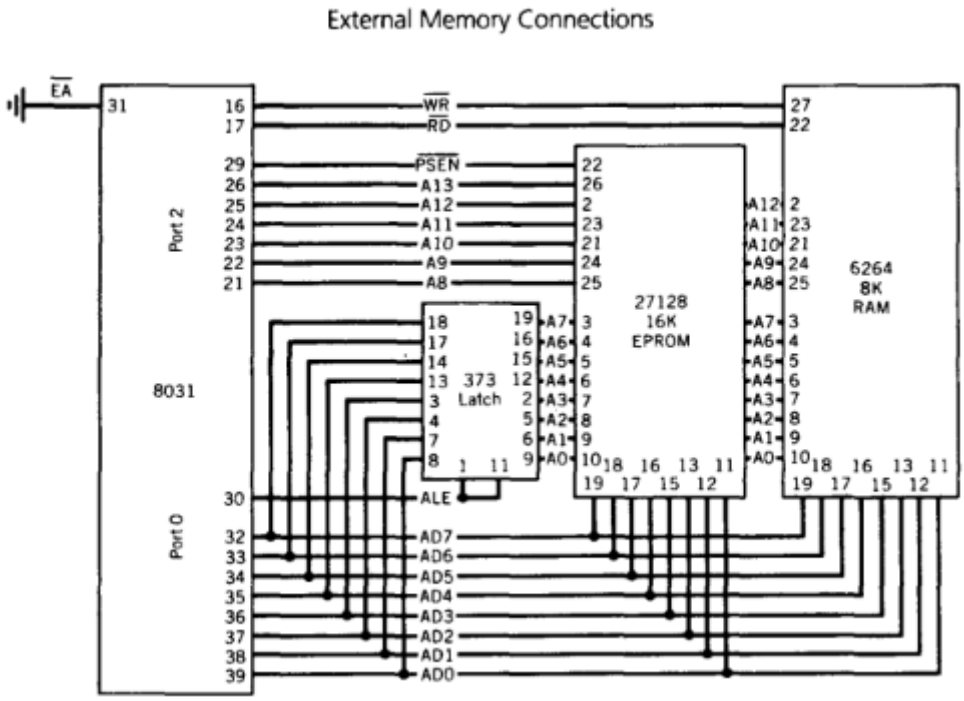
iii) RLC A



iv) DIV AB

Mnemonic	Operation
DIV AB	Divide A by B; put the integer part of quotient in register A and the integer part of the remainder in B

7. With a neat diagram, explain the steps to interface 8KB of program RAM and 4 KB of ROM to 8031 based systems.



8. Write an ALP to find the number of positive and negative numbers in a given array of ten bytes of data, the number is available from memory location 8000H

```
ORG 0000H
MOV DPTR, #8000H  ;Input array location starting address
MOV R0, #10H              ;Negative numbers stored at address 10H onwards
MOV R1, #20H              ;Positive numbers stored at address 20H onwards
MOV R2, #5                ;Number of elements in input array

REPEAT:MOVX A, @DPTR  ;Move the element from location pointed by DPTR into Accumulator
JB ACC.7, NEG             ;Check if MSB bit of accumulator is set, then jump to label NEG
MOV @R1, A                ;else if the number is positive, then store at address pointed by R1
INC R1                            ;increment R1 to point to next location
SJMP LOOP1                ; jump to LOOP1

NEG: MOV @R0, A
INC R0

LOOP1:INC DPTR            ; increment DPTR to point to next location of input array
DJNZ R2, REPEAT           ; decrement the count value and check if zero, else jump to label REPEAT
SJMP $                            ;stop the program here
END
```