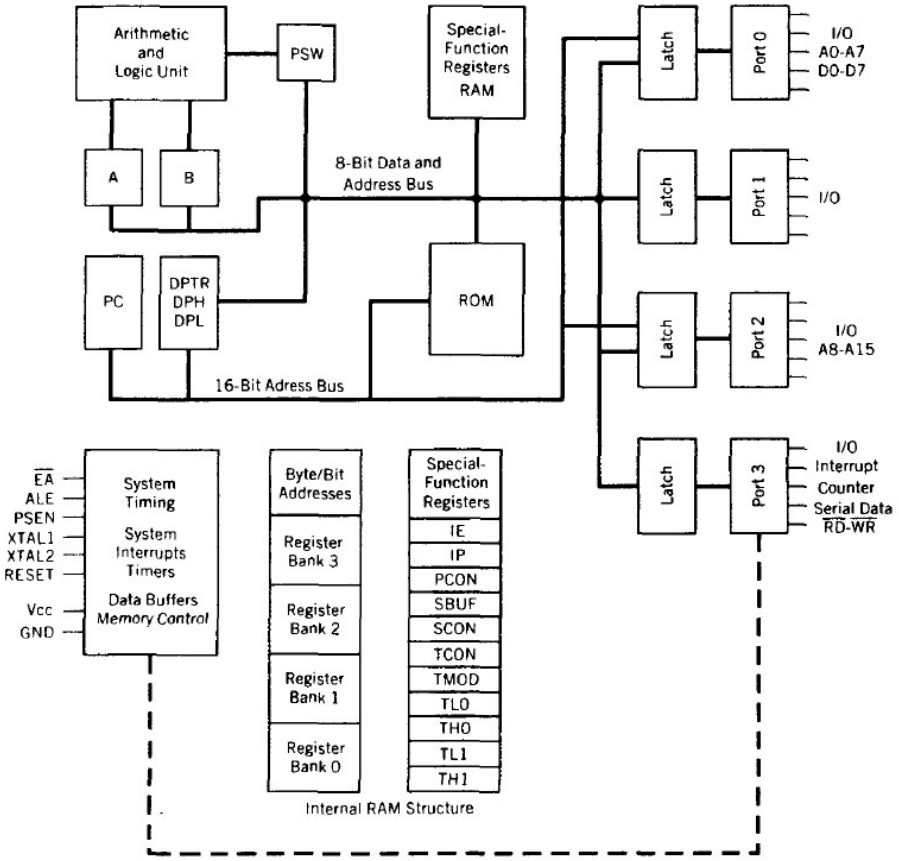
CMR
INSTITUTE OF
TECHNOLOGY

USN

Internal Assesment Test-I							
Sub:	Microcontroller					Code:	18EC46
Date:	12/07 /2022	Duration:	90 mins	Max Marks:	50	Sem: 4	Branch: ECE
Answer FIVE FULL Questions							

		OBE	
	Marks	CO	RBT
1. With a neat diagram, explain the architecture of 8051.	[10]	CO1	L2



Salient features of 8051 microcontroller are given below.

- Eight bit CPU
- On chip clock oscillator
- 4K bytes of internal program memory (code memory) *[ROM]*
- 128 bytes of internal data memory *[RAM]*
- 64 Kbytes of external program memory address space. 64 Kbytes of external data memory address space.
- 32 bi directional I/O lines (can be used as four 8 bit ports or 32 individually addressable I/O lines)
- Two 16 Bit Timer/Counter :T0, T1
- Full Duplex serial data receiver/transmitter

- Four Register banks with 8 registers in each bank.
- Sixteen bit Program counter (PC) and a data pointer (DPTR)
- 8 Bit Program Status Word (PSW)
- 8 Bit Stack Pointer
- Five vector interrupt structure (RESET not considered as an interrupt.)
- 8051 CPU consists of 8 bit ALU with associated registers like accumulator ‘A’ , B register, PSW, SP, 16 bit program counter, stack pointer.
- ALU can perform arithmetic and logic functions on 8 bit variables.
- 8051 has 128 bytes of internal RAM which is divided into
  - Working registers [00 – 1F]
  - Bit addressable memory area [20 – 2F]
  - General purpose memory area (Scratch pad memory) [30-7F]
- 8051 has 4 K Bytes of internal ROM. The address space is from 0000 to 0FFFh. If the program size is more than 4 K Bytes 8051 will fetch the code automatically from external memory.

- There are 128 bytes of RAM in the 8051 with assigned addresses 00 to 7FH
- The 128 bytes are divided into three different groups as follows:
  - A total of 32 bytes from locations 00 to 1F hex are set aside for register banks and the stack
  - A total of 16 bytes from locations 20H to 2FH are set aside for bit-addressable read/write memory
  - A total of 80 bytes from locations 30H to 7FH are used for read and write storage, called scratch pad

- Accumulator is an 8 bit register widely used for all arithmetic and logical operations. Accumulator is also used to transfer data between external memory. B register is used along with Accumulator for multiplication and division. A and B registers together is also called MATH registers.
- PSW (Program Status Word). This is an 8 bit register which contains the arithmetic status of ALU and the bank select bits of register banks.

CY	AC	F0	RS1	RS0	OV	-	P

CY - carry flag

AC - auxiliary carry flag

F0 - available to

the user for general purpose

RS1,RS0 - register bank

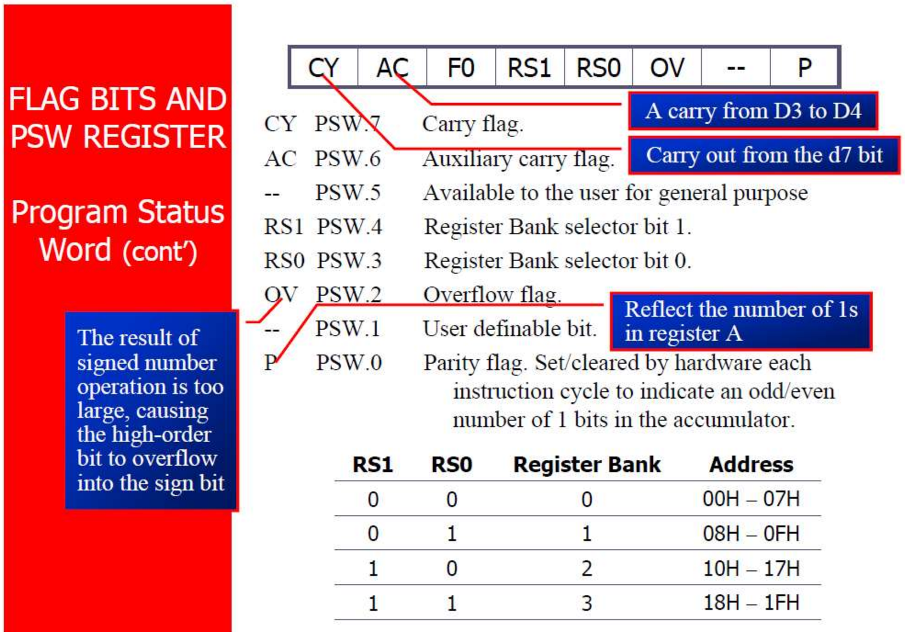
select bits

OV - overflow

P - parity

- Stack Pointer (SP) – it contains the address of the data item on the top of the stack. Stack may reside anywhere on the internal RAM. On reset, SP is initialized to 07 so that the default stack will start from address 08 onwards.
- Data Pointer (DPTR) – DPH (Data pointer higher byte), DPL (Data pointer lower byte). This is a 16 bit register which is used to furnish address information for internal and external program memory and for external data memory.
- Program Counter (PC) – 16 bit PC contains the address of next instruction to be executed. On reset PC will set to 0000. After fetching every instruction PC will increment by one.

2.(a) Explain the PSW register and also find the status of the conditional flags if number ABH and FFH is added in the accumulator. [07] CO1 L2



FLAG BITS AND PSW REGISTER

Program Status Word (cont')

CY	AC	F0	RS1	RS0	OV	--	P

A carry from D3 to D4

Carry out from the d7 bit

CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary carry flag.
--	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 an odd/even number of 1 bits in the accumulator.

Reflect the number of 1s in register A

The result of signed number operation is too large, causing the high-order bit to overflow into the sign bit

RS1	RS0	Register Bank	Address
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

CY - CARRY FLAG:

This flag is set whenever there is a carry out of MSB bit that is from D7th bit. This flag bit is affected after an 8-bit addition or subtraction.

If CY = 1 : There was a carry out of the MSB

If CY = 0 : There was no carry out of the MSB

This carry flag bit can be set 1 or 0 directly by an instruction such as "SETB C" (set
bit carry) and "CLR C" (clear carry).

AC – AUXILIARY CARRY FLAG:

It indicates the carry from lower nibble (4-bits) to higher nibble. If the 8 bits are numbered Bit 7 - Bit 0, this is the carry from Bit 3 to Bit 4.

If AC = 1 : There was an auxiliary carry

If AC = 0 : There was no auxiliary carry

OVR - OVERFLOW FLAG:

This flag is set whenever the result of signed number operation is too large, causing the higher-order bit to overflow into the sign bit. The overflow flag is used to detect errors in signed arithmetic operations.

An 8-bit signed number has the range -80H… ,-01H,00H,… +7FH (Decimal -128 …,-01,00,….+127). Any result, out of this range causes an overflow.

If OVR = 1 : There was an overflow in the result

If OVR = 0 : There was no overflow in the result

Overflow is determined by doing an Ex-Or between the 2nd last carry (C6) and the
last carry (C7)

P - PARITY FLAG:

The parity flag reflects the number of 1's in A (accumulator) register only. It indicates the Parity of the result.

If PF = 1 : The register 'A' has ODD number of 1's

If PF = 0 : The register 'A' has EVEN number of 1's

F0 – USER DEFINED FLAG:

This flag is available to the programmer.
It can be used by us to store any user defined information.

RS1, RS0 – REGISTER BANK SELECT:

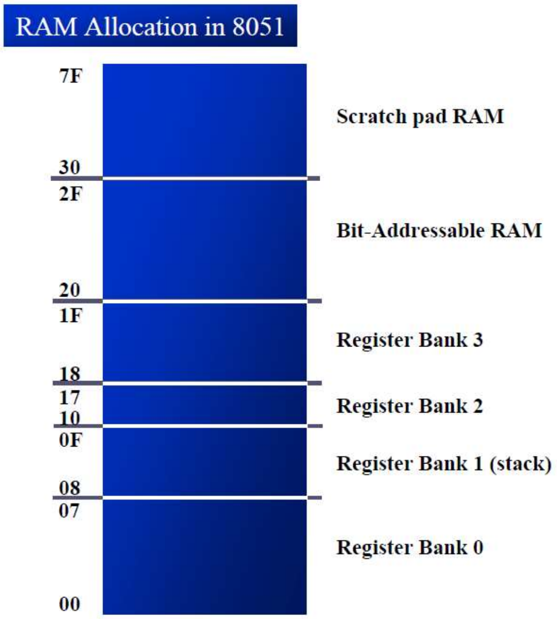
AB H + FF H = AA H; CY = 1, AC = 1, P = 0, OV = 0.

2.(b) Explain the following instructions and their addressing mode, (i)MOVC A,@A+DPTR, (ii) MOV A, @R0. [03] CO2 L2

(i) Copies the contents of memory location pointed to by the sum of the accumulator A and the DPTR into the accumulator.

(ii) R0 register is used to hold the actual address that will be used in data movement. Value in R0 is used as the address of the location from which data is copied to accumulator.

3.(a) Explain the organization of internal RAM of 8051. [05] CO1 L2



- There are 128 bytes of RAM in the 8051 with assigned addresses 00 to 7FH
- The 128 bytes are divided into three different groups as follows:

1. A total of 32 bytes from locations 00 to 1F hex are set aside for register banks

and the stack

2. A total of 16 bytes from locations 20H to 2FH are set aside for bit-addressable

read/write memory

3. A total of 80 bytes from locations 30H to 7FH are used for read and write

storage, called scratch pad

Register Banks: 00h to 1Fh.

- The 8051 uses 8 general-purpose registers R0 through R7 (R0, R1, R2, R3, R4, R5, R6, and R7).
- There are four such register banks.
- Selection of register bank can be done through RS1, RS0 bits of PSW.
- On reset, the default Register Bank 0 will be selected.

Bit Addressable RAM: 20h to 2Fh.

- The Bit Addressable area of the RAM is just 16 bytes of Internal RAM located between 20h and 2Fh.
- We can perform ordinary byte operations on these locations, as well as bit operations.
- As each location has 8-bits, we have a total of 16 × 8 = 128 Addressable Bits.
- These bits can be addressed using their individual address 00H … 7FH.

  SETB 00H: Will store a “1” on the LSB of location 20H

  CLR 07H: Will store a “0” on the MSB of location 20H
- MOV 20H, #00H ; Will store a “0” on all 8-bits of location 20H.
- Addressable bits are useful when the program need only remember a binary event (switch on, light off, etc.).

General Purpose RAM: 30h to 7Fh.

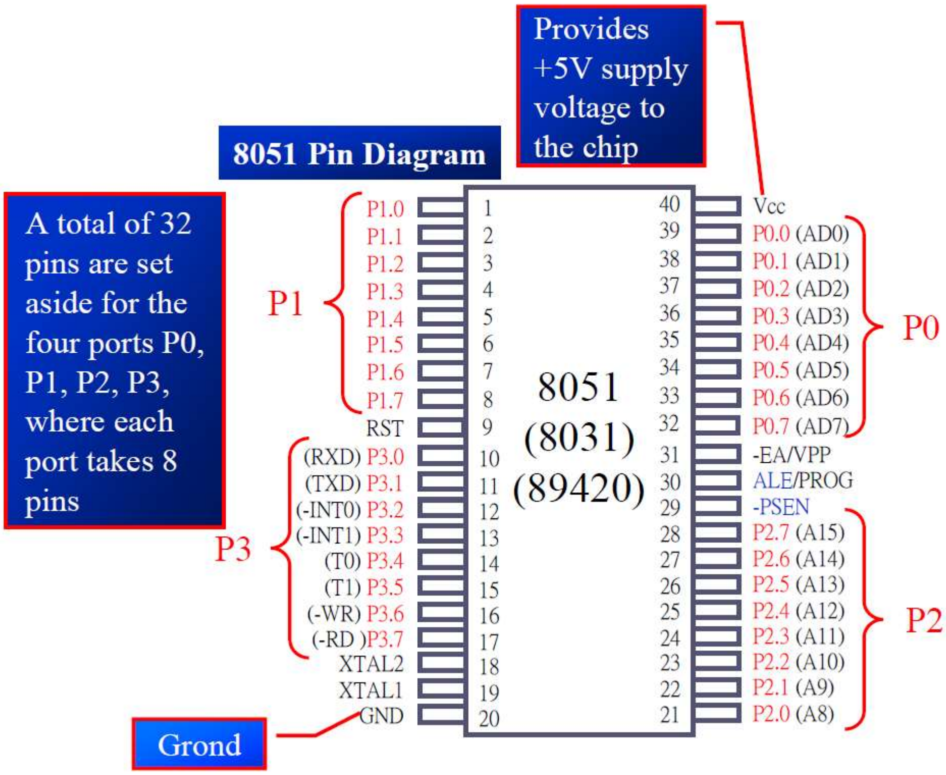
- 80 bytes of Internal RAM memory are available for general-purpose data storage
- It is a good practice to use general purpose memory from 30 – 7Fh
- The general-purpose RAM can be accessed using direct or indirect addressing modes.

3.(b) Write an ALP to copy a block of data byte from external RAM to internal RAM. [05] CO1 L3

```
         MOV DPTR, #9000H
         MOV R0, #40H
         MOV R2, #05
BACK:    MOVX A, @DPTR
         MOV @R0, A
         INC DPTR
         INC R0
         DJNZ R2, BACK
```

```
SJMP $
END
```

4. How many pins are present in 8051 microcontroller and list the function of each pin with a neat pin diagram. [10] CO1 L1



Pins 1-8 (PORT 1) . It has 8 pins It can be used as input or output. It already has internal pull-up resistors. Hence no external pull-up resistors are required. On Reset port1 is configured as I/P port by writing 1 to all its bits .

Pin 9 (RESET). RESET pin 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. 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. For the RESET input to be effective, it must have a minimum duration of 2 machine cycles.

Pins10-17 (PORT 3).

□ It has 8 pins from 10 to 17

□ It does not need any external pull-up resistors

□ On reset it is configured as input

Pin 20 GND. Ground.

Pin 21-28(Port 2). If there is no intention to use external memory then these port pins are Configured as general inputs/outputs. In case external memory is used, the higher address byte. It has 8 pins. It can be used as input or output

It already has internal pull-up resistors Hence no external pull-up resistors are required

In 8051-based systems with no external memory connection. Both P1 and P2 are used as simple I/O

In 8031/51-based systems with external memory connections:

Port 2 must be used along with P0 to provide the 16-bit address for the external memory

P0 provides the lower 8 bits via A0 – A7

P2 is used for the upper 8 bits of the 16-bit address, designated as A8 – A15, and it cannot be used for I/O

Pin 29 (PSEN)¯. If external ROM is used for storing program then a logic zero (0) appears on it

every time the microcontroller reads a byte from memory.

PSEN, "program store enable'', is an output pin

This pin is connected to the OE pin of the ROM

When the (EA)¯ pin is connected to GND, 8051 fetches opcode from external ROM

Pin 30 ALE: "address latch enable" is an output pin and is active high

Port 0 provides both address and data

The 8051 multiplexes address and data through port 0 to save pins

ALE indicates if P0 has address or data

When ALE=0, it provides data D0-D7

When ALE=1, it has address A0-A7

Prior to reading from external memory, the microcontroller puts the lower address byte (A0-A7) on P0 and activates the ALE output. After receiving signal from the ALE pin, the external latch latches the state of P0 and uses it as a memory chip address. Immediately after that, the ALE pin is returned its previous logic state and P0 Is now used as a Data Bus.

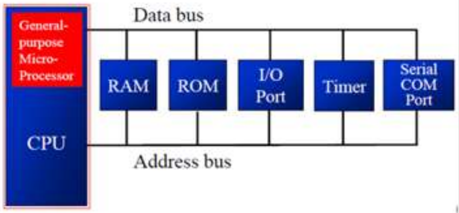
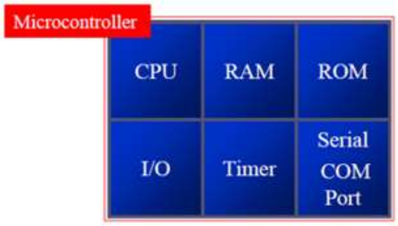
Pin 31 (EA)¯. By applying logic zero to this pin, P2 and P3 are used for data and address transmission with no regard to whether there is internal memory or not. It means that even there is a program written to the microcontroller, it will not be executed. Instead, the program written to external ROM will be executed.

(EA)¯pin must be connected to GND to indicate that the code is stored externally

Pin 32-39 (PORT 0): Similar to P2, if external memory is not used, these pins can be used as general inputs/outputs. Otherwise, P0 is configured as address output (A0-A7) when the ALE pin is driven high (1) or as data output (Data Bus) when the ALE pin is driven low (0). When ALE=0, it provides data D0-D7 When ALE=1, it has address A0-A7 Port 0 has 8 pins.

Pin 40 VCC. +5V power supply.

5.(a) Write the differences between microprocessor and microcontroller. [04] CO1 L1

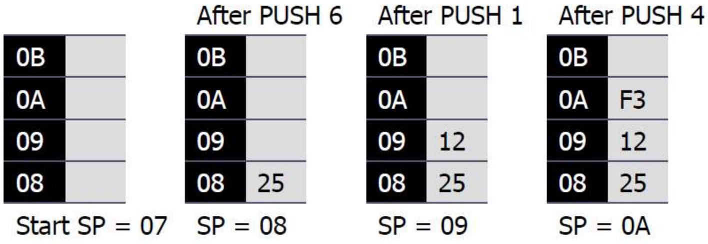
Microprocessor	***Microcontroller***
 General-purpose Micro-Processor CPU Data bus RAM ROM I/O Port Timer Serial COM Port Address bus	 Microcontroller CPU RAM ROM I/O Timer Serial COM Port
Block diagram of microprocessor	*Block diagram of microcontroller*
Microprocessor contains ALU, General purpose 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
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

5.(b) Discuss the need of stack memory in microcontroller. Explain PUSH and POP operation in detail. [06] CO2 L2

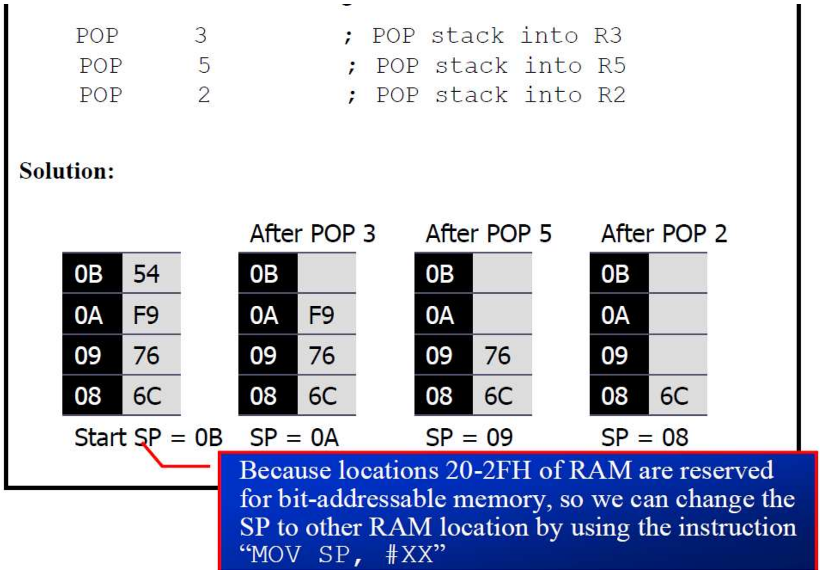
- SP is pointing to the last used location of the stack
- As we push data onto the stack, the SP is incremented by one
- This is different from many microprocessors
- Loading the contents of the stack back into a CPU register is called a POP
- With every pop, the top byte of the stack is copied to the register specified by the instruction and the stack pointer is decremented once

```
MOV R6, #25H
MOV R1, #12H
MOV R4, #0F3H
PUSH  6
PUSH  1
PUSH  4
```

Solution:



			After PUSH 6		After PUSH 1		After PUSH 4
0B		0B		0B		0B	
0A		0A		0A		0A	F3
09		09		09	12	09	12
08		08	25	08	25	08	25
Start SP = 07		SP = 08		SP = 09		SP = 0A	



```
POP     3          ; POP stack into R3
POP     5          ; POP stack into R5
POP     2          ; POP stack into R2
```

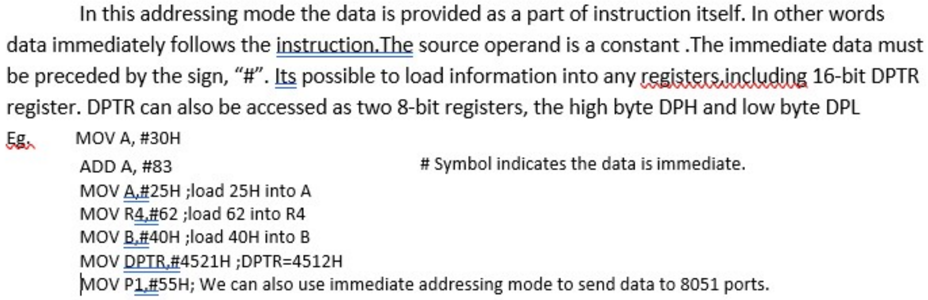
Solution:

			After POP 3		After POP 5		After POP 2
0B	54	0B		0B		0B	
0A	F9	0A	F9	0A		0A	
09	76	09	76	09	76	09	
08	6C	08	6C	08	6C	08	6C
Start SP = 0B		SP = 0A		SP = 09		SP = 08	

Because locations 20-2FH of RAM are reserved for bit-addressable memory, so we can change the SP to other RAM location by using the instruction "MOV SP, #XX"

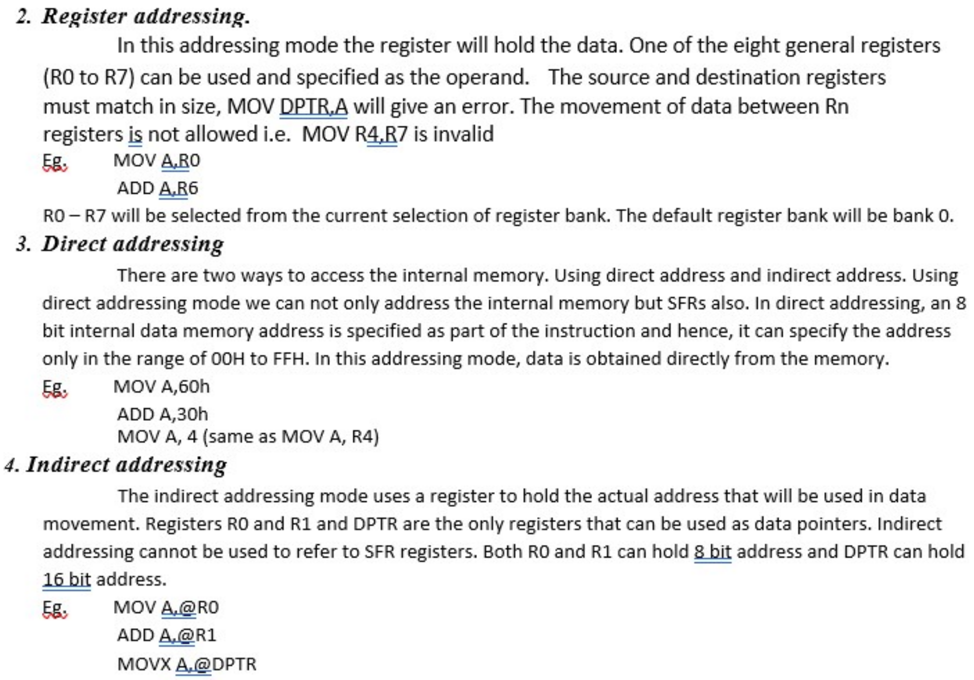
6.(a) Explain any four addressing modes used in microcontroller 8051 with suitable examples. [05] CO2 L2

1. Immediate addressing.



In this addressing mode the data is provided as a part of instruction itself. In other words data immediately follows the instruction.The source operand is a constant .The immediate data must be preceded by the sign, "#". Its possible to load information into any registers,including 16-bit DPTR register. DPTR can also be accessed as two 8-bit registers, the high byte DPH and low byte DPL

```
Eg.    MOV A, #30H
       ADD A, #83                       # Symbol indicates the data is immediate.
       MOV A,#25H ;load 25H into A
       MOV R4,#62 ;load 62 into R4
       MOV B,#40H ;load 40H into B
       MOV DPTR,#4521H ;DPTR=4512H
       MOV P1,#55H; We can also use immediate addressing mode to send data to 8051 ports.
```



2. Register addressing.

In this addressing mode the register will hold the data. One of the eight general registers (R0 to R7) can be used and specified as the operand. The source and destination registers must match in size, MOV DPTR,A will give an error. The movement of data between Rn registers is not allowed i.e. MOV R4,R7 is invalid

Eg. MOV A,R0
ADD A,R6

R0 – R7 will be selected from the current selection of register bank. The default register bank will be bank 0.

3. Direct addressing

There are two ways to access the internal memory. Using direct address and indirect address. Using direct addressing mode we can not only address the internal memory but SFRs also. In direct addressing, an 8 bit internal data memory address is specified as part of the instruction and hence, it can specify the address only in the range of 00H to FFH. In this addressing mode, data is obtained directly from the memory.

Eg. MOV A,60h
ADD A,30h
MOV A, 4 (same as MOV A, R4)

4. Indirect addressing

The indirect addressing mode uses a register to hold the actual address that will be used in data movement. Registers R0 and R1 and DPTR are the only registers that can be used as data pointers. Indirect addressing cannot be used to refer to SFR registers. Both R0 and R1 can hold 8 bit address and DPTR can hold 16 bit address.

Eg. MOV A,@R0
ADD A,@R1
MOVX A,@DPTR

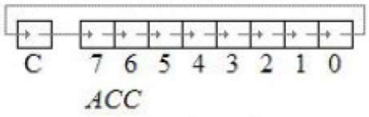
6. (b) Explain the following instructions: [05] CO2 L3

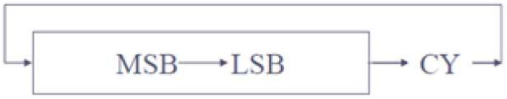
(i) RRC A, (ii) DA A, (iii) DIV AB.

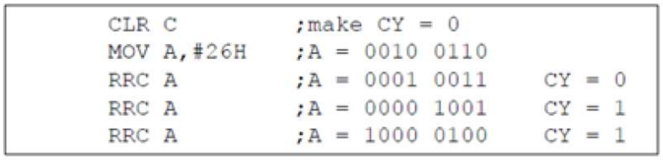
(i)

RRC A (Rotate Accumulator Right through Carry)

Rotate right through the carry. Each bit is shifted one location to the right, with bit 0 going into the carry bit in the PSW, while the carry was at goes into bit 7

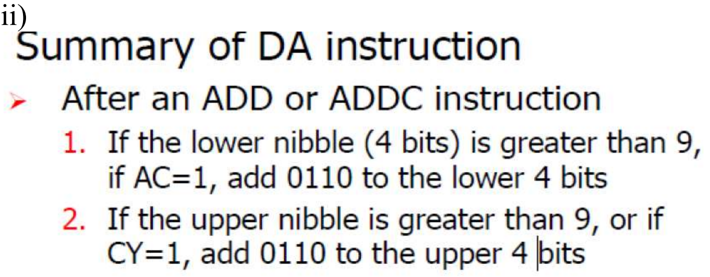






```
CLR C           ;make CY = 0
MOV A,#26H      ;A = 0010 0110
RRC A           ;A = 0001 0011     CY = 0
RRC A           ;A = 0000 1001     CY = 1
RRC A           ;A = 1000 0100     CY = 1
```

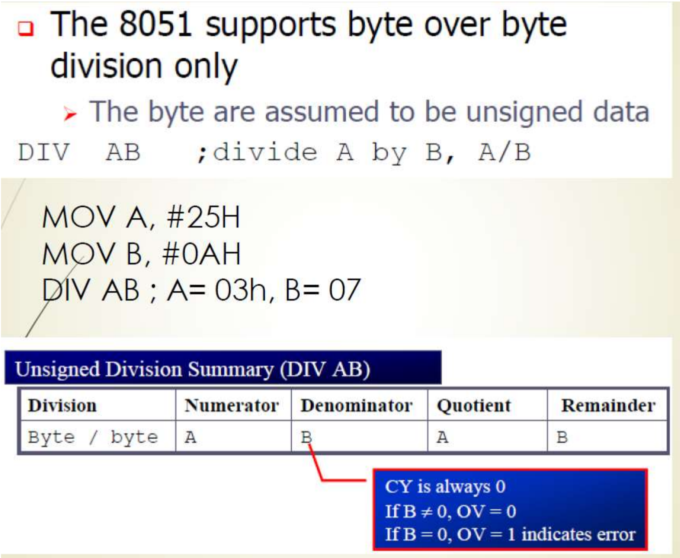
(ii)



Summary of DA instruction

- After an ADD or ADDC instruction
  1. If the lower nibble (4 bits) is greater than 9, if AC=1, add 0110 to the lower 4 bits
  2. If the upper nibble is greater than 9, or if CY=1, add 0110 to the upper 4 bits

(iii)



- The 8051 supports byte over byte division only
  - The byte are assumed to be unsigned data

```
DIV  AB   ;divide A by B, A/B
```

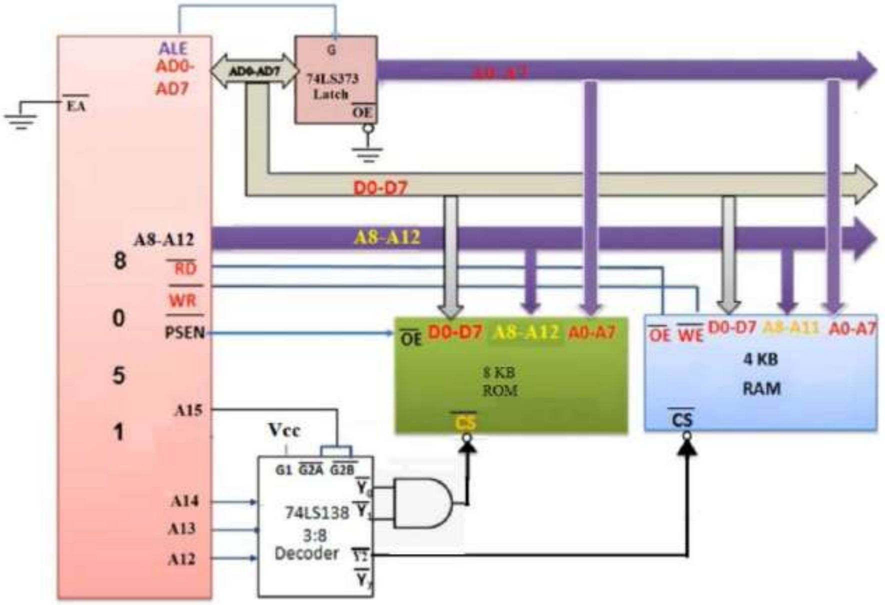
```
MOV A, #25H
MOV B, #0AH
DIV AB ; A= 03h, B= 07
```

Unsigned Division Summary (DIV AB)

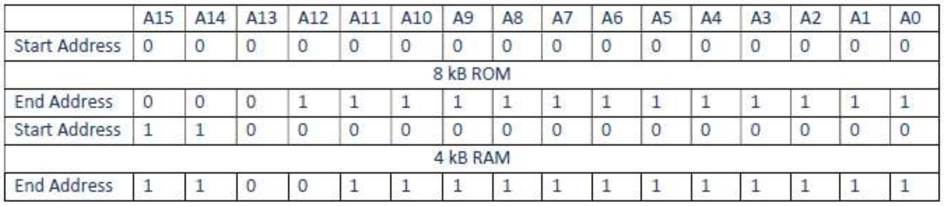
Division	Numerator	Denominator	Quotient	Remainder
Byte / byte	A	B	A	B

CY is always 0
If B ≠ 0, OV = 0
If B = 0, OV = 1 indicates error

7. Write the interfacing diagram of 4 K bytes of RAM and 8K bytes of ROM to 8051 microcontroller and explain. [10] CO1 L3



Interface 4k bytes of RAM and 8k bytes ROM to 8051 microcontrollers in such a way that starting address of RAM is C000H and ROM is 0000H.



	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Start Address	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8 kB ROM																
End Address	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
Start Address	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4 kB RAM																
End Address	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Address mapping