CMR

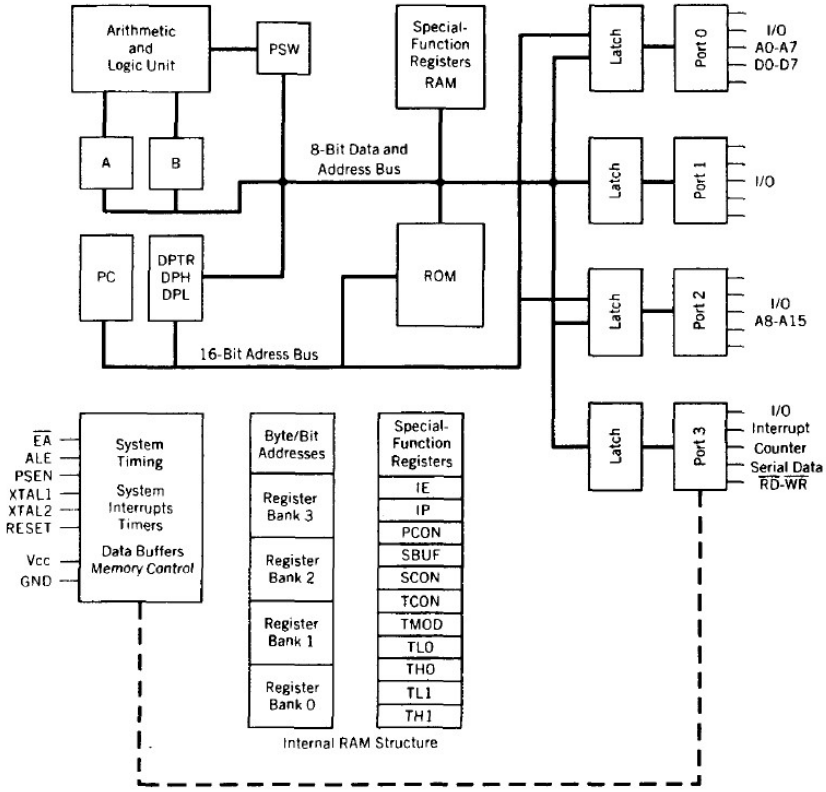
INSTITUTE OF

TECHNOLOGY

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

Internal Assesment Test - I

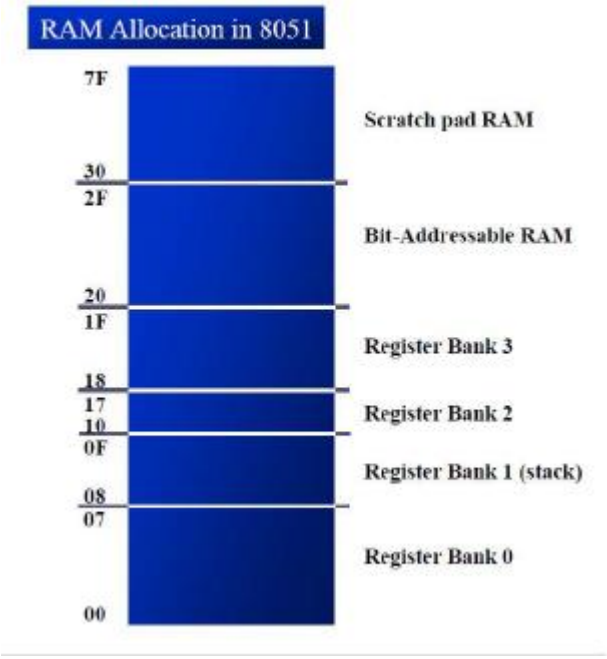
Sub:	Microcontroller							Code:	18EC46
Date:	21/05 / 2021	Duration:	60 mins	Max Marks:	50	Sem:	3rd	Branch:	ECE
Answer All Questions									

		Marks	OBE	
			CO	RBT
1.	• Draw the architecture of 8051 microcontroller and explain memory mapping of 8051. • Solution: • THE 8051 ARCHITECTURE  • 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.	[10]	CO1	L1

- 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.
- 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

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]



- 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.	Explain any five addressing modes related to 8051 microcontroller.
	
	Solution:
	
	Various methods of accessing the data are called addressing modes.
	8051 addressing modes are classified as follows.
	1. Immediate addressing.
	2. Register addressing.
	3. Direct addressing.
	4. Indirect addressing.
	5. Indexed addressing.
	
	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.
	E,g, MOV A,R0
	ADD A, R6
	
	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.
		[2x5]	CO2	L1

E,g, 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.

e.g. MOV A, @R0
ADD A, @R1

5. Indexed addressing

In indexed addressing, either the program counter (PC), or the data pointer (DTPR)—is used to hold the base address, and the A is used to hold the offset address. Adding the value of the base address to the value of the offset address forms the effective address. Indexed addressing is used with JMP or MOVC instructions. Look up tables are easily implemented with the help of index addressing.

E.g. MOVC A, @A+DPTR
MOVC A, @A+PC

3. Explain the operations of the following pins of 8051: ALE, RST, $\overline{RD}$, $\overline{WR}$, $\overline{EA}$, $\overline{PSEN}$, TxD and RxD.

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

RESET value of some 8051 registers

we must place the first line of source code in ROM location 0

Register	Reset Value
PC	0000
DPTR	0000
ACC	00
PSW	00
SP	07
B	00
P0-P3	FF

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.

RD. Read from external RAM, $\overline{RD}$ = 0, microcontroller reads the data from external RAM

[10] CO1 L1

WR. Pin 16 of 8051. Write to external (additional) RAM, When $\overline{WR}$ = 0: Microcontroller writes the data into external RAM.

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.

$\overline{\boldsymbol{PSEN}}$**.** If external ROM is used for storing program then a logic zero (0) appears on itevery time the microcontroller reads a byte from memory.

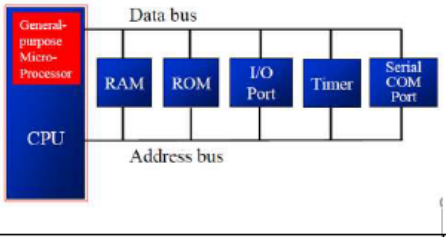
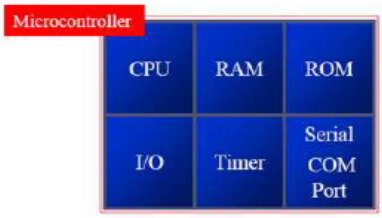
- PSEN, "program store enable'', is an output pin
- This pin is connected to the OE pin of the ROM
- When the $\overline{EA}$ pin is connected to GND, 8051 fetches opcode from external ROM

RxD : Pin 10. Data is received by 8051 through RxD pin.

TxD : Pin 11. Data is transmitted out of 8051 using TxD pin

4. Write the differences between microprocessor and microcontroller.

MICROPROCESSORS AND MICROCONTROLLERS

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
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

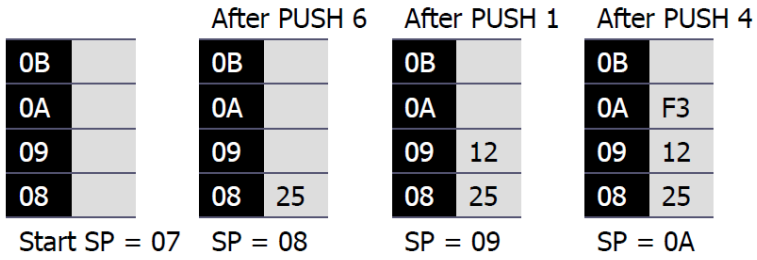
[10] CO1 L2

	Discuss the importance and operation of stack memory in 8051. ➡ A stack is a last in first out memory ➡ The stack is a section of RAM used by the CPU to store information temporarily ➡ This information could be data or an address ➡ The CPU also uses the stack to save the address of the instruction just below the CALL instruction ➡ This is how the CPU knows where to resume when it returns from the called subroutine ➡ The register used to access the stack is called the SP (stack pointer) register ➡ The stack pointer in the 8051 is only 8 bit wide, which means that it can take value of 00 to FFH ➡ When the 8051 is powered up, the SP register contains value 07 ➡ RAM location 08 is the first location used for the stack by the 8051 ➡ When a data is to be placed on the stack, the stack pointer increments before storing the data on the stack so that the stack grows up as data is stored (pre-increment). As the data is retrieved from the stack the byte is read from the stack, and then SP decrements to point the next available byte of stored data (post decrement). ➡ The storing of a CPU register in the stack is called a PUSH ➡ 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			

Show the stack and stack pointer from the following. Assume the default stack area.

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

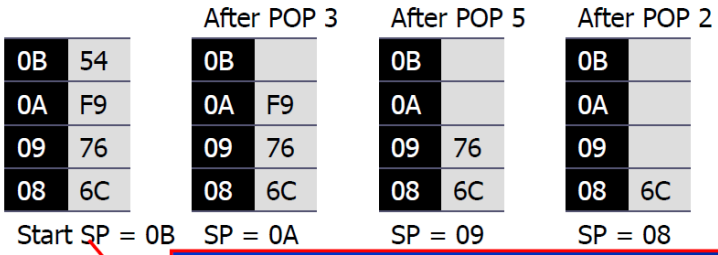
Solution:



Examining the stack, show the contents of the register and SP after execution of the following instructions. All value are in hex.

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

Solution:



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"

5. Explain the structure of PSW register with a neat diagram. Explain with examples the following instructions: exchange and data copy instructions.

i. The program status word (PSW) register, also referred to as the flag register, is an 8 bit register
ii. Only 6 bits are used
iii. These four are CY (carry), AC (auxiliary carry), P (parity), and OV (overflow)
iv. They are called conditional flags, meaning that they indicate some conditions that

[05+05] | CO1 | L2

resulted after an instruction was executed

v. The PSW3 and PSW4 are designed as RS0 and RS1, and are used to change the bank. The two unused bits are user-definable

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

PSW bank selection

	RS1(PSW.4)	RS0(PSW.3)
Bank 0	0	0
Bank 1	0	1
Bank 2	1	0
Bank 3	1	1

Data Transfer Instructions:

In this group, the instructions perform data transfer operations of the following types.

i. Move the contents of a register Rn to A
   1. MOV A, # data
   2. MOV A,Rn ;(n = 0,1.....7)
   3. MOV A, Direct
   4. MOV A, @Ri (i = 0,1)

ii. Register A is the source
   1. MOV Rn,A
   2. MOV Direct,A

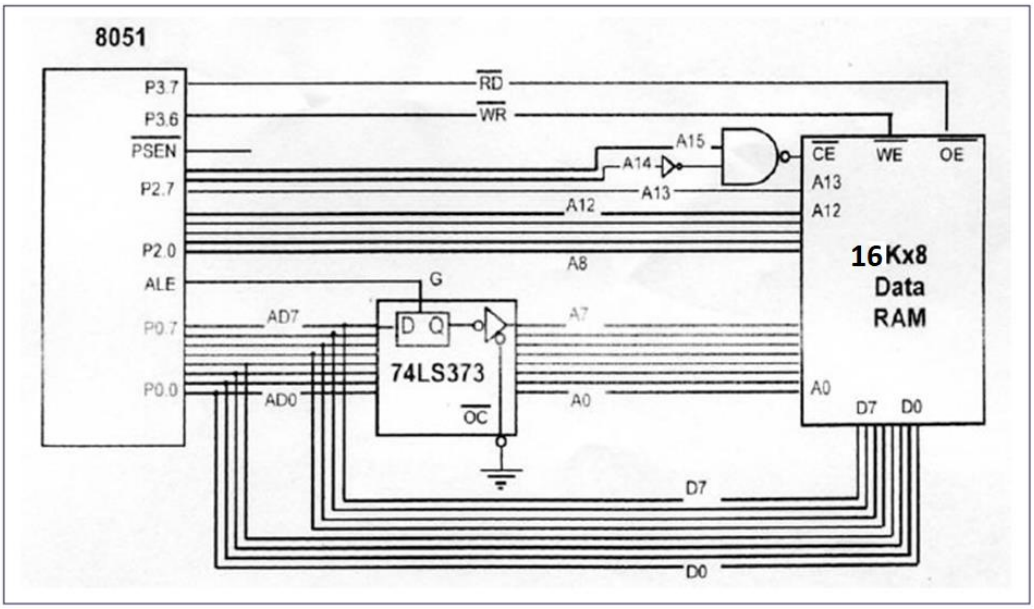
iii. Rn is the destination
   1. MOV Rn,immediate
   2. MOV Rn, A
   3. MOV Rn, Direct

iv. The destination is a direct address
   1. MOV Direct, #data
   2. MOV Direct ,@Ri
   3. MOV Direct, A
   4. MOV Direct, Rn

v. The destination is an indirect address held

	by R0 or R1 1. MOV @Ri,#data 2. MOV @Ri,A 3. MOV @Ri, Direct 4. MOV dest-bit, source-bit Push and Pop instructions The storing of a CPU register in the stack is called a **PUSH**. When a data is to be placed on the stack, the stack pointer increments before storing the data on the stack so that the stack grows up as data is stored (pre-increment). SP always points to the last used location of the stack. As we push data onto the stack, the SP is incremented by one 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. As the data is retrieved from the stack the byte is read from the stack, and then SP decrements to point the next available byte of stored data (post decrement). **Exchange instructions** The content of source ie., register, direct memory or indirect memory will be exchanged with the contents of destination i.e., accumulator. Exchange A with a byte variable *i.* XCH A,R3 *ii.* XCH A,@R1 *iii.* XCH A,54h **Exchange digit**. Exchange the lower order nibble of Accumulator (A0-A3) with lower order nibble of the internal RAM location which is indirectly addressed by the register. *iv.* XCHD A,@R1 *v.* XCHD A,@R0			

6.	Write an assembly language program in 8051 to transfer 10 bytes of data stored in external memory location, with starting address 2000H to internal memory location 30H. Draw algorithm and flow chart for the same.
	MOV R0, #30H
	MOV DPTR, #2000H
	MOV R2, #0AH
	BACK: MOVX A, @DPTR
	MOV @R0, A
	INC R0
		[10]	CO2	L3

	INC DPTR DJNZ R2, BACK SJMP $ END **Algorithm**: 1. Copy the starting address to R0 2. Copy the external address to DPTR register 3. Initialize the count in R2 register 4. Move the content of external memory to accumulator 5. Move the contents of accumulator to internal memory 6. Increment R0 and DPTR 7. Repeat the steps 4 to 6 till contents of counter register R2 becomes zero.			
7.	Draw an interfacing diagram showing the interfacing of 8051 with (16K x 8) external RAM. 	[10]	CO1	L3