FIRST TERM EXAMINATION [SEPT. 2018]
FIFTH SEMESTER [B.TECH]
MICROPROCESSOR AND MICROCONTROLLERS
[ETEC–305]

Time : 1.30 hrs. **M.M. : 30**

***Note:** Attempt Q. No. 1 which is compulsory and any two more questions from remaining.*

Q.1.Diferentiate between

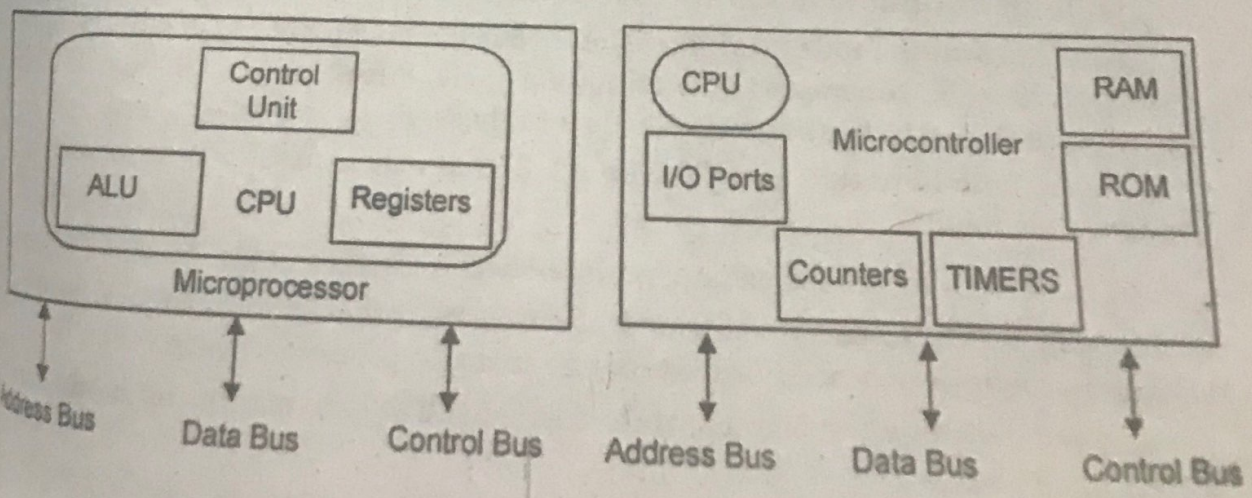
(a) I/O mapped I/O and Memory mapped I/O **(2)**

Ans.

Memory Mapped I/O	I/O Mapped I/O
In Memory Mapped I/O Address width is 16-bit. A0 to A15 are used to generate address of the device.	In I/O Mapped I/O Address width is 8-bit. A0 to A7 lines are used to generate address of the device.
MEMR and MEMW control signals are used to control read and write I/O operations respectively.	IOR and IOW control signals are used to control read and write I/O operations respectively.
Instructions available are STA addr, LDA addr, LDAX rp, STAX rp, ADD M, CMP M,MOVr,M,etc.	IN and OUT are the only available instructions.
Data transfer takes place between any register and I/O device.	Data transfer takes place between accumulator and I/O device.
Maximum number of I/O devices that can be addressed is 65536 (theoretically).	Maximum number of I/O devices that can be addressed is 256.
Execution speed using STA addr, LDA addr is 13 T-state and for MOV M, r, etc.,it is 7-T states.	Execution speed is 10 T-states.
It requires more hardware circuitry because it decodes 16-bit address.	It requires less hardware circuitry because it decodes 8-bit address.

Q. 1. (b) Microprocessors and Microcontroller. **(2)**

Ans. The following table shows some of the differences between microprocessors and microcontrollers.



Microprocessor	Microcontroller
Microprocessor assimilates the function of a central processing unit (CPU) on to a single integrated circuit (IC).	Microcontroller can be considered as a small computer which has a processor and some other components in order to make it a computer.
Microprocessors are mainly used in designing general purpose systems from small to large and complex systems like super computers.	Microcontrollers are used in automatically controlled devices.
Microprocessors are basic components of personal computers.	Microcontrollers are generally used in embedded systems
Computational capacity of microprocessor is very high. Hence can perform complex tasks.	Less computational capacity when compared to microprocessors . Usually used for simpler tasks.

Q. 1. (c) 8085 and 8086 Microprocessors. (2)

Ans.

Difference Between 8085 & 8086 Microprocessor

8085 Microprocessor	8086 Microprocessor
• Is an 8 Bit Miroprocessor	• Is a 16 Bit Microprocessor
• Has 8 bit data bus	• Has 16 bit data bus
• Has is bit address line	• Has 20 bit address line
• Only 64kB of memory can be used (2^{16})	• IMB of memory can be used (2^{20})
• Has 5 Flags (Carry, Parity, Sign, Zero, Axillary Carry)	• Has 9 Flags (Carry, Parity, Sign, Zero, Auxillary Carry, Direction, Trap, Interrupt, Overflow)
• It is Accumulator based processor	• It is General Purpose Register Based Processor
• It has no MIN mode or MAX mode	• It can operate in any one of MIN or MAX Mode
• Does not support Pipelining	• Supports Pipelining
• Does not support Memory Segmentation	• Supports Memory Segmentation
• Has 6500 transistors	• Has 29000 transistors

Q. 1. (d) Minimum mode and Maximum Mode of 8086 Microprocessor. (2)

Ans. Minimum mode configuration of 8086: If pin number 33 of 8086 is connected to logic 1 then 8086 operates in the minimum mode. Pins 24 to 31 of 8086 will have the function as shown in the parenthesis, next to these pins.

Please note here that pins 24 through 31 have dual functions depending on the mode of operation.

In minimum mode, for example, pin 29 acts as write ($\overline{WR}$). This pin will go low when 8086 wants to carry out a operation on memory or ports. Also note that the minimum mode of operation is suitable for a single processor systems.

Let us now study the signals which correspond only to **minimum mode** shown in dotted lines above.

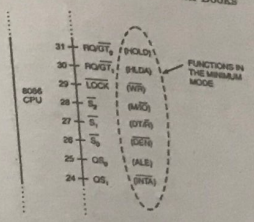


Fig.

1. M/$\overline{IO}$ (Memory/$\overline{IO}$): In minimum mode, pin number 28 acts as Memory / $\overline{IO}$ output line. The status of this pin indicates whether memory is being accessed or I/O port is being accessed by CPU.

If $M/\overline{IO}$ = 1 then CPU is ac ic memory.

else if $M/\overline{IO}$ = 0 then CPU is accessing I/O devices.

2. Write $\overline{WR}$: It is an output an number 29. If processor wants to perform a write operation on memory and I / O devices then the $\overline{WR}$ pin goves low.

3. $\overline{INTA}$ (Interrupt Acknowledge): It is an active low signal which is sent out to the interrupting device to tell it that its interrupt requiest has been accepted.

4. ALE (Address Latch Enable): It is an active high output line (Pin no. 24). If ALE = 1, it indicates that address is present on the multiplexed address/data bus. This ALE signal is connected to the strobe (STB) inputs of the external latches to strobe in or latch this address into them.

5. DT/$\overline{R}$ (Data Transmit/Receive): An output pin of 8086 in minimum mode that decides the direction of flow of data on the data line.

If DT/$\overline{R}$ = 1 then data flows out from CPU.

else if DT/$\overline{R}$ = 0 then data flows into the CPU.

6. $\overline{DEN}$ (Data Enable): It is an output pin number 26 in the minimum mode $\overline{DEN}$ is connected to $\overline{OE}$ (output enable) input of transceiver. When $\overline{DEN}$ = 0, the transceiver output is enabled. When $\overline{DEN}$ = 1, the transceiver output floats in tristate. So, $\overline{DEN}$ is used to enable the transceiver. This output is active low during each memory and I/O access.

7. HOLD and HLDA: Both HOLD (pin 31) and HOLD acknowledge (HLDA) i.e., pin 30 are used for DMA (Direct Memory Access).

Maximum Mode Configuration of 8086: If pin number 33 of 8086 is connected to logic 0 then the processor operates in the maximum modes. The configuration is quite complex and is used for the multiprocessor systems. Here in, pins 24 to 31 have functions described outside the dotted line in Fig. Like now pins 26,27 and 28 works as $\overline{S_0}$,$\overline{S_1}$ and $\overline{S_2}$ respectively. These are control bus signals.

To achieve maximum mode for use with external co-processor, the MN/$\overline{MX}$ Pin must be connected to ground. Let us see the pins of this mode now

1. $\overline{S_2}$, $\overline{S_1}$ and $\overline{S_0}$: Status bits show the function of the current bus cycle. Such signals are normally decoded by 8086 bus controller. Let us see in a tabular form, their functions.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Functions
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Opcode Fetch
1	0	1	Memory Read
1	1	0	Memory Write
1	1		
1	Passive		

2. $\overline{RQ}$/GT1 and $\overline{RQ}$/$\overline{GT0}$: The request/grant pins request DMA during maximum mode operation. Both these lines are bidirectional and are needed to request and grant a DMA operation.

3. $\overline{LOCK}$: The lock output is used to lock peripherals off the system. This pin is activated by using the clock prefix on any instruction.

4. QS_1 and QS_0: The queue status bits show the states of the internal instruction queue. These pins are provided to access numeric math co-processor (8087). The operation of the queue status bits is in truth-table below:

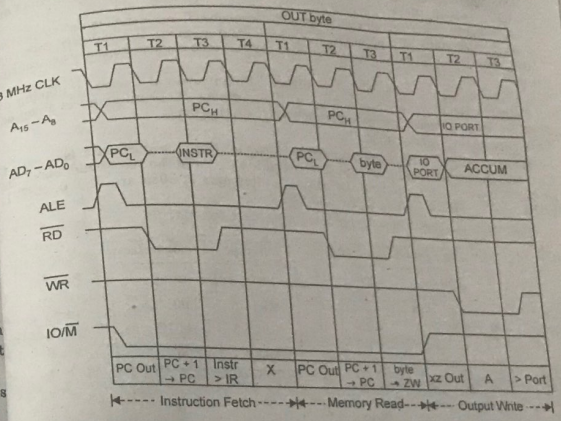
QS_1	QS_0	Function
0	0	Queue is idle (NOP)
0	1	First byte is opcodes
1	0	Queue is empty
1	1	Subsequent byte of opcode

Q. 1. (e) Non maskable and Maskable Interrupts (2)

Ans. The interrupt which can be ignored by the processor, while performing its operations are called maskable interrupts, generally maskable interrupts are the interrupts that comes from the pheripheral devices, where as the non maskable interrupts are the interrupts which cannot be ignored, generally these type of interrupts are specified to be software interrupts. The examples of maskable are: mouseclick.memory read etc the examples of non maskable are powerfailure,software corrupted etc.

Q. 2.(a) Draw timing diagram for OUT 20H for 8085 Microprocessors. (5)

Ans. The above figure gives an example of 8085 timing, showing the value of external control signals. Of course, at the same time internal control signals are being generated by the control unit to control internal data transfers. Three machine cycles (M1, M2, M3) are needed. During the first, the OUT instruction is fetched. The second machine cycle fetches the second half of the instruction, which contains the number of the I/O device selected for output. During the third cycle, the contents of the AC are written out to the selected device over the data bus.



The start of each machine cycle is signaled by the Address Latch Enabled (ALE) pulse from the control unit. The ALE pulse alerts external circuits. During timing state T1, of machine cycle M1, the control unit sets the IO/M signal to indicate that this is a memory operation. Also, the control unit causes the contents of the PC to be placed on the address bus (A_{15} - A_8) and address/data bus (AD_7 - AD_0). With the falling edge of the ALE pulse, the other modules on the bus store the address.

During timing state T2, the addressed memory module places the contents of the addressed memory location on the address / data bus. The control unit sets the Real Control (RD signal to indicate a read, but it waits until T3 to copy the data from the bus. This gives the memory module time to put the data on the bus and for the signal levels to stabilize. The final state, T4, is a *bus idle* state during which the CPU decodes the instruction. The remaining machine cycle proceed in a similar fashion.

Q. 2. (b) What do you mean by interrupts of 8085 Microprocessors also draw its vector diagram. (5)

Ans. Interrupts of Intel 8085

Need For Interrupts: Interrupt is a signal send by an external device to the processor, to the processor to perform a particular task or work. Mainly in the microprocessor based system the interrupts are used for data transfer between the peripheral and the microprocessor.

When a peripheral is ready for data transfer, it interrupts the processor by sending an appropriate signal to the interrupt pin of the processor. If the processor accepts the interrupt then the processor suspends its current activity and executes an interrupt service subroutine to complete the data transfer between the peripheral and processor. After executing the interrupt service routine the processor resumes its current activity. This type of data transfer scheme is called interrupt driven data transfer scheme.

Types of interrupts: The interrupts are classified into software interrupts and hardware interrupts.

- The software interrupts are program instructions. These instructions are inserted at desired locations in a program. While running a program, If a software interrupt instruction is encountered, then the processor executes an interrupt service routine (ISR).
- The hardware interrupts are initiated by an external device by placing an appropriate signal at the interrupt pin of the processor. If the interrupt is accepted, then the processor executes an interrupt service routine (ISR).

Software Interrupts of 8085

The software interrupts are program instructions. When the instruction is executed, the processor executes an interrupt service routine stored in the vector address of the software interrupt instruction. The software interrupts of 8085 are RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6 and RST 7.

The vector addresses of software interrupts are given in table.

Interrupt	Vector address
RST 0 RST 1	0000_H 0008_H
RST 2 RST 3	0010_H 0018_H
RST 4 RST 5	0020_H 0028_H
RST 6 RST 7	0030_H 0038_H

Interrupt	Vector address
RST 7.5	$003C_H$
RST 6.5	0034_H
RST 5.5	$002C_H$
TRAP	0024_H

When the processor encounters the software instruction, it pushes the content of PC (Program Counter) to stack. Then loads the Vector address in PC and starts executing the Interrupt Service Routine (ISR) stored in this vector address. At the end of ISR, a return instruction - RET will be placed. When the RET instruction is executed, the processor POP the content of stack to PC. Hence the processor control returns to the main program after servicing the interrupt. Execution of ISR is referred to as servicing of interrupt.

All software interrupts of 8085 are vectored interrupts. The software interrupts cannot be masked and they cannot be disabled. The software interrupts are RST 0, RST1, ...RST 7 (8 Nos).

Hardware Interrupts of 8085

An external device, initiates the hardware interrupts of 8085 by placing an appropriate signal at the interrupt pin of the processor. The processor keeps on checking the interrupt pins at the second T -state of last machine cycle of every instruction. If the processor finds a valid interrupt signal and if the interrupt is unmasked and enabled, then the processor accepts the interrupt. The acceptance of the interrupt is acknowledged by sending an $\overline{INTA}$ signal to the interrupted device.

The processor saves the content of PC (program Counter) in stack and then loads the vector address of the interrupt in PC. (If the interrupt is non-vectored, then the interrupting device has to supply the address of ISR when it receives INTA signal). It starts executing ISR in this address. At the end of ISR, a return instruction, RET will be placed. When the processor executes the RET instruction, it POP the content of top of stack to PC. Thus the processor control returns to main program after servicing interrupt. The hardware interrupts of 8085 are TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR.

Q. 3.(a) Explain the 8086 Microprocessor in Minimum Mode?
Ans. Refer Q. 4. (a) of End Term Exam 2018. (5)

Q. 3. (b) Explain the addressing modes of 8086 Microprocessors? (5)

Ans. Addressing modes: The method by which address of source of data is given alongwith, instruction is called as addressing mode of source.

1. Immediate addressing mode (IAM): If 8/16 bit data required for executing the instruction is given alongwith the instruction, then it is called immediate addressing mode.

Example:

1. MOV AL, 75H; 78H → *AL*

75 06 *BX*

2. MOV BX, 7506 H; 7506 H → BH BL

2. Direct addressing mode (DAM): If 8/16 bit data is present in memory and 16 bit E.A. of this memory location is given along with the instruction, then it is called direct addressing mode instructions.

Example: MOV AL, [9106H]

3. Register direct addressing mode (RDAM): If 8/16 bit data required for executing the instruction, is present in register and the name of register is given alongwith the instruction, then it is called *RDAM* instruction.

Example: *MOV CX, BX*

4. Register indirect addressing mode (RIAM): If the data is present in memory and the E.A. is present in a register, then it is called RIAM instruction.

EA = [BX]/[SI]/[DI]

Example: MOV AL, [SI]

5. Register relative addressing mode (RRAM): Data is present in memory location and the EA = [BX]/[BP]/[SI]/[DI] + 8/6 bit displacement.

Example: MOV CX, 97H [BP]

6. Base index addressing mode (BIAM): Data is present in memory location and the EA = [BX]/[BP] + [SI]/[DI]

Example: MOV CX, [BX] [SI]

7. Relative base index addressing mode (RBIAM): EA. = [BX]/[BP] + [SI]/[DI] + 8/16 bit displacement

Example: MOV AH, 1907H [BX][DI]

8. Implicit addressing mode (IPAM): If address of source of data as well as address of destination of result, are fixed then no operand is given alongwith the instruction.

Example: CLD, STD

MOV DX, 8AH[BP][SI] is used the relative base index addressing mode.

Q.4.(a) Write an assembly language program for Fibonacci series for 8085 Microprocessors? (5)

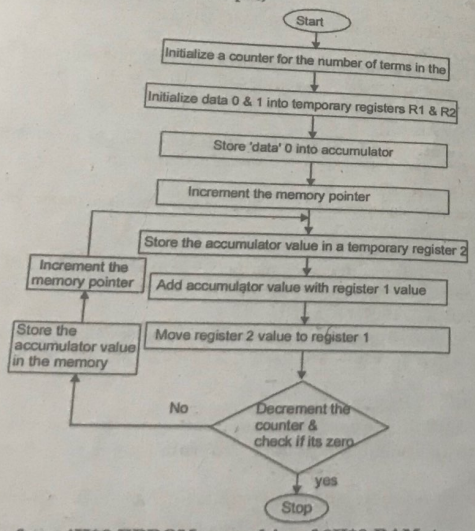
Ans. To write an assembly language program to generate the Fibonacci series with given first two terms.

```
C100 LXI H C300 21 ; Load the HL register pair immediately
C103 MOV B M 46 ; Move the memory content in to B register
C104 INX H 23 ; Increment the HL register pair
C105 MOV C M 4E ; Move the memory content in to C register
C106 DCR B 05 ; Decrement the B register content
C107 INX H 23 ; Increment the HL register pair
```

C108 MOV A M 7E ; Move the memory content in to accumulator
C109 DCR B 05 ; Decrement the B register content
C10A MOV D A 57 ; Move the accumulator content to D register
C10B ADD C 81 ; Add the C register content with accumulator
C10C INX H 23 ; Increment the HL register pair
C10D MOV M A 77 ; Move the accumulator content to memory
C10E MOV C D 4A ; Move the D register content to C register
C10F DCR B 05 ; Decrement the B register content
C110 JNZ C10A C2 ; Jump if no zero to C10AH
C113 HLT 76 ; Halt the execution

EXECUTION

C300 08 ; Number of input data
C301 00 ; 1st data (Input)
C302 01 ; 2nd data (Input)
C303 01 ; Series of Fibonacci (Output)
C304 02 ; Series of Fibonacci (Output)
C305 03 ; Series of Fibonacci (Output)
C306 05 ; Series of Fibonacci (Output)
C307 08 ; Series of Fibonacci (Output)
C308 0D ; Series of Fibonacci (Output)



Q.4.(b) Interface 4K*8 EPROM, one chip of 8K*8 RAM, two chips of 8K*4 RAM with 8086 Microprocessor. (5)

Ans.

4K × 8	EPROM	=	1000 Memory location
8K × 8	RAM	=	2000 Memory location
Two 8K × 4	RAM	=	2000 Memory location

Address lines 4K × 8 EPROM = $2^2 \times 2^{10} = A_0 - A_{11}$
8K × 8 RAM = $2^3 \times 2^{10} = A_0 - A_{12}$

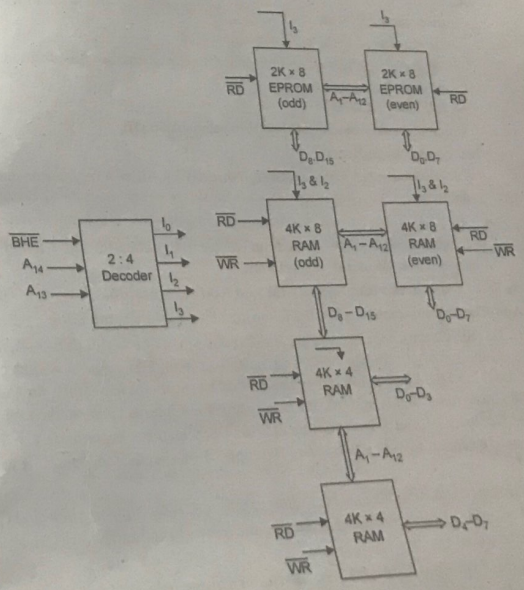
We will take maximum address lines

Memory Map

Memory	Hexa Add	A_{19} A_{18} A_{17} A_{16} A_{15} A_{14} A_{13} A_{12}	A_{11} A_{10} A_9 A_8 ..... A_3 A_2 A_1 A_0
4K × 8 EPROM	FFFFF	1 1 1 1 1 1 1 1	1 1 1 1 ..... 1 1 1 1
	FF000	1 1 1 1 1 1 1 1	1 0 0 0 0 ..... 0 0 0 0
8K × 8 RAM	FEFFF	1 1 1 1 1 1 1 0	1 1 1 1 ..... 1 1 1 1
	FD000	1 1 1 1 1 1 0 1	1 0 0 0 0 ..... 0 0 0 0
Two 8Kx4 RAM	FCFFF	1 1 1 1 1 1 0 0	1 1 1 1 ..... 1 1 1 1
	FB000	1 1 1 1 1 1 0 1	1 0 0 0 0 ..... 0 0 0 0

Chip select Lines | I/P Address Lines

2:4 Decoder used to select Memory A_{13} and A_{14} as selection lines of decoder and four output to select RAM/EPROM



END TERM EXAMINATION [NOV-DEC. 2018]
FIFTH SEMESTER [B.TECH]
MICROPROCESSOR AND MICROCONTROLLERS
[ETEC–305]

Time : 3 hrs. **M.M. : 75**

Note: *Attempt any five questions including Q. no. 1 which is compulsory. Attempt one question from each unit.*

Q.1. (a) What determines whether a microprocessor is considered a 4-bit, a 16-bit and 32-bit device? (2.5)

Ans. The size of the data bus and all the internal register of the microprocessor decides whether a microprocessor is a 4-bit as Intel 4004, 8-bit as Intel 8085 or 16-bit as Intel 8086.

Q. 1. (b) Explain intra segment and inter segment jump. (2.5)

Ans.

- **Inter-segment Direct:**
  - → Destination is in the different segment.
  - → New segment as well as Immediate displacement value is defined in the instruction
  - → JMP 5555H:2222H
  - → Jump to effective address 2222H in segment 5555H
- **Inter-segment Indirect**
  - → Here the displacement is passed indirectly by either a register or memory location
  - → JMP [2000]
  - → It seeks 2 consecutive words from the location defined in the DS and that becomes new CS and IP respectively.

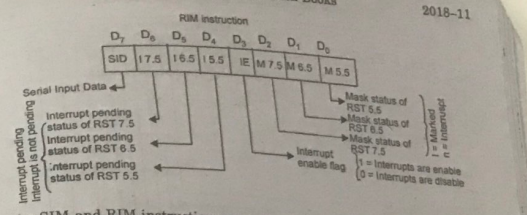
Q. 1. (c) Explain the function of SIM and RIM instruction. (2.5)

Ans. The RIM instruction reads the following bits into the accumulator:

Bit D7 (SID-Serial Input Data) This is the input pin of the serial data interface which is connected to pin 5 of the 8085, and indicates the high/low status of that pin. Bits D6-D4 (I 7.5,I 6.5,I 5.5) These bits indicates that an interrupt is pending for these three 8085 interrupts 7.5,6.5, and 5.5. If interrupts 5.5 or 6.5 have been masked off by bits D0 or D1, bits, D4 and D5 will not be set. Bit D6, which corresponds to the 7.5 interrupt, will be set on to indicate that an interrupt 7.5 was requested, even if it was masked off.

Bit D3 (IE-Interrupt Enable) This bit indicates whether interrupts are enabled (1) using the EI (Enable Interrupts) instruction, or disabled (0) using the DI (Disable Interrupts) instruction.

Bits 2-D0 (M 7.5, M6.5, M5.5) Mask status of interrupts 7.5, 6.5, and 5.5. Corresponds to bits D2-D0 of the SIM instruction. 1 if masked, 0 if enabled.



So the SIM and RIM instructions are typically used to either output to or input from 8085 serial interface, or enable/disable/read the interrupt masks for interrupts 7.5,6.5,5.5, but usually not at the same time.

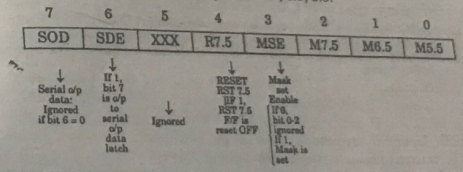
RIM = Read Interrupt Mask

It is uses to find the Pending Interrupt of the 8085.

RIM

SIM (Set Interrupt Mask)

Used to Mask any one of the Interrupt RST 7.5, 6.5, 5.5.



Q. 1. (d) Explain the assembler directives: EQU and DB used in 8051. (2.5)

Ans. Refer to Q.5.(a)(i)&(iii) of page number 14-2017.

Q. 1. (e) Explain the flag register of microprocessor 8085 and 8086. (2.5)

Ans.

8085 FLAG REGISTER

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY
SIGN FLAG	ZERO FLAG		AUXILIARY CARRY		PARITY FLAG		CARRY FLAG

- **CY-** The carry flag is used for carrying and borrowing in case of addition and subtraction operations, it set when carry generated otherwise it reset
- **P-** The parity flag is used for results containing an even number of one's, it set for even parity and reset for odd parity
- **AC-** In an arithmetic operation, when a carry is generated by digit D3 and passed on to digit D4, the AC flag is set. The auxiliary carry flag is used for BCD operations, not free to the programmer
- **Z-** The zero flag is set if the result of an instruction is zero otherwise it reset
- **S-** Sign flag is-ve if S = 1 and Sign flag is +ve if S = 0

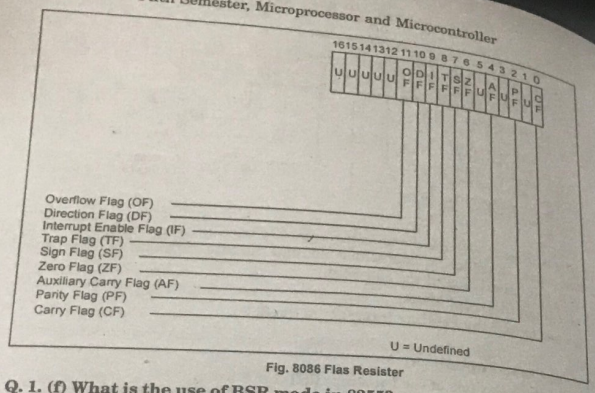


Fig. 8086 Flas Resister

Q. 1. (f) What is the use of BSR mode in 8255? (2.5)

Ans. The Bit Set/Reset (BSR) mode is applicable to port C only. Each line of port C ($PC_0 - PC_7$)can be set/reset by suitably loading the control word register. BSR mode and I/O mode are independent and selection of BSR mode does not affect the operation of other ports in I/O mode.

8255 BSR mode

- D_7 bit is always 0 for BSR mode.
- Bits D_6, D_5 and D_4 are don't care bits.
- Bits D_3, D_2 and D_1 are used to select the pin of-Port C.
- Bit D_0 is used to set/reset the selected pin of Port C.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	B_2	B_1	B_0	S/R

Always 0 for BSR mode — Don't care — Port C bit select — Set/Reset

8255 Control Register formet for BSR Mode

Selection of port C pin is determined as follows:

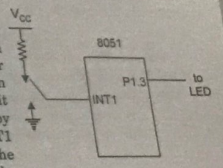
B3	B2	B1	Bit/pin of port C selected
0	0	0	PC_0
0	0	1	PC_1
0	1	0	PC_2
0	1	1	PC_3
1	0	0	PC_4
1	0	1	PC_5
1	1	0	PC_6
1	1	1	PC_7

Q. 1. (g) Write a program to toggle P3.2 using Timer interrupt in 8051. (3.5)

Ans. Assume that the INT1 pin is connected to a switch that is normally high. Whenever it goes low, it should turn on an LED. The LED is connected to PI .3 and is normally off. When it is turned on it should stay on for a fraction of a second. As long as the switch is pressed low, the LED should stay on.

```
        ORG     0000H
        LJMP    MAIN                ;bypass interrupt vector table
ISR for hardware interrupt INT1 to turn on the LED
        ORG     0013H               ; INT1 ISR
        SETB    P1.3                ; turn on LED
        MOV     R3,#255             ; load counter
Back:   DJNZ    R3, Back            ; keep LED on for a while
        CLR     P1.3                ; turn off the LED
        RETI                        ; return from ISR
;---MAIN program for initialization
        ORG     30H
MAIN:   MOV     IE,#10000100B       ;enable external INT1
HERE:   SJMP    HERE                ;stay here until interrupted
        END
```

Pressing the switch will turn the LED on. If it is kept activated, the LED stays on.



In this program, the microcontroller,is looping continuously in the HERE loop. Whenever the switch on INT1 (pin P3.3) is activated, the microcontroller gets out of the loop and jumps to vector location 0013H. The ISR for INT1 turns on the LED, keeps it on for a while, and turns it off before it returns. If by the time it executes the RETI instruction, the INT1 pin is still low, the microcontroller initiates the interrupt again. Therefore, to end this problem, the INT1 pin must be brought back to high by the time RETI is executed.

Q. 1. (h) Explain the function of TMOD register in 8051. (2.5)

Ans. Refer to Q. 8. (b) of page number 22-2017

Q. 1. (i) Write a program to toggle pin P1.2 every second. (2.5)

Ans. I/O ports and bit-addressability

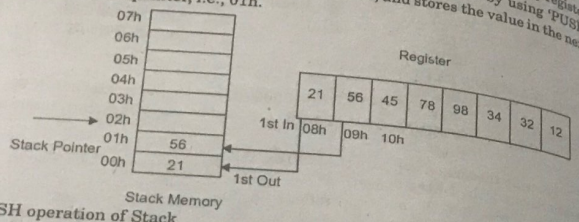
Sometimes we need to access only 1 or 2 bits of the port instead of the entire 8 bits. A powerful feature of 8051 I/O ports is their capability to access individual bits of the port without altering the rest of the bits in that port. Of the four 8051 ports, we can access either the entire 8 bits or any single bit without altering the rest. When accessing a port in single-bit manner, we use the syntax "SETB X. Y" where X is the port number 0, 1,2. or 3, and Y is the desired bit number from 0 to 7 for data bits D0 to D7. For example, "SETB PI. 5" sets high bit 5 of port 1. Remember that D0 is the LSB and D7 is the MSB. For example, the following code toggles bit PI .2 continuously.

```
BACK:   CPL     P1.2        ;complement P1.2 only
        ACALL   DELAY
        SJMP    BACK
        ; another variation of the above program follows
```

```
AGAIN:  SETB   P1.2
        ACALL  DELAY   ;change only P1.2 = high
        CLR    P1.2
        ACALL  DELAY   ;change only P1.2 = low
        SMP    AGAIN
```

Q.1.(j) Explain PUSH and POP instruction in 8051. (2.5)

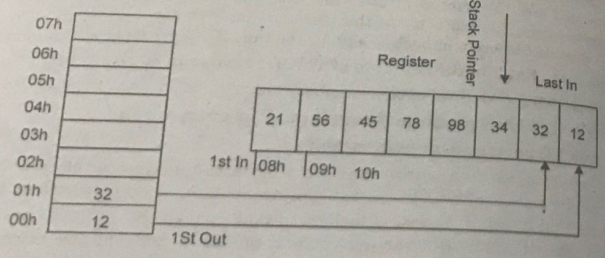
Ans. PUSH operation: The 'PUSH' is used for taking the values from any register and storing in the starting address of the stack pointer, i.e., 00h by using 'PUSH' operation. And, for the next 'PUSH', it increments +1, and stores the value in the next address of the stack pointer, i.e., 01h.



PUSH operation of Stack

PUSH operation means (First in First out)

POP Operation: It is used for placing the values from the stack pointer's maximum address to any other register's address. If we use this 'POP' again, then it decrements by 1, and the value stored in any register is given as 'POP'.



POP Operation in Stack

POP operation means 'Last in First out'.

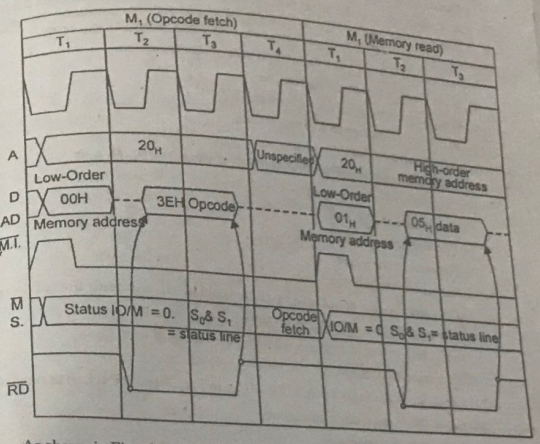
UNIT-I

Q.2.(a) Draw and explain the architecture of 8085. (6)

Ans. *Refer to Q.4.(a) of page number 4-2017*

Q. 2. (b) Draw and explain timing diagram for the instruction MVIB, 05h. (6.5)

Ans. Timing diagram for MVIB, 05 H. 2018–15



As shown in Fig., the instruction needs two machine cycles. The first machine cycle is OpCode fetch. It takes 4 T-states (4 clock periods). The second machine cycle is a memory read which takes 3 T-states. The instruction cycle takes 7 T-states.

Here is description of what happens in the system bus until the instruction MVI B, 05_H is executed.

- The Program Counter loads the memory address 2000H into the address bus during T_1 of the Op Code machine cycle.
- The address decoding system locates and identifies the memory location 2000H.
- At T_2, the Timing and Control unit produces the MEMR-signal (Read) which lasts during T_2 and T_3. During this window of time the memory places the Op Code $3E_H$ from location 2000H into the data bus.
- The operating system places the Op Code in the Instruction Register then into the Instruction Decoder. When the Instruction Decoder decodes the Op Code it feeds the decoded signal into the Timing and Control Unit. The fetch operation is completed in 1 T_4.
- The Program Counter is incremented to 2001H.
- During T_4, the Timing and Control unit finds out that a second byte which contains the data needs to be read (i.e. address 2001H).
- The second machine cycle is Memory Read cycle.
- At the T_1 of the 2nd machine cycle the Program Counter loads the address 2001H into the address bus.

• The address decoding system locates and identifies the memory location 2001H.

• At T_2 the timing and control unit produces MEMR signal which lasts during T_2 and T_3. During the rest of T_3, the data 05_H is stored into Accumulator. During this window of time the data 05_H is placed into the data bus then into the MPU.

• During the rest of T_3, the data 05_H is stored into B register.

Q. 3. (a) Write a program in 8085 to

(i) Find largest number in a given string.

(ii) Find even number in a given string.

Ans. (i) ALGORITHM: (6)

(1) Load the address of the first element of the array in HL pair

(2) Move the count to B – reg.

(3) Increment the pointer

(4) Get the first data in A – reg.

(5) Decrement the count.

(6) Increment the pointer

(7) Compare the content of memory addressed by HL pair with that of A - reg.

(8) If Carry = 0, go to step 10 or if Carry = 1 go to step 9

(9) Move the content of memory addressed by HL to A – reg.

(10) Decrement the count

(11) Check for Zero of the count. If ZF = 0, go to step 6, or if ZF = 1 go to next step.

(12) Store the largest data in memory.

(13) Terminate the program.

PROGRAM:

```
      LXI H,4200 Set pointer for array
      MOV B,M    Load the Count
      INX H      Set 1st element as largest data
      MOV A,M
      DCR B      Decrements the count
LOOP: INX H
      CMP M      F A- reg > M go to AHEAD
      JNC AHEAD
      MOVA,M     Set the new value as largest
AHEAD:DCR B
      JNZ LOOP   Repeat comparisons till count = 0
      STA 4300   Store the largest value at 4300
      HLT
```

OBSERVATION:

Input: 05 (4200) ------ Array Size

Output: 0A (4201)

F1 (4202)
1F (4203)
26 (4204)
FE (4205)
FE (4300)

RESULT: Thus the program to find the largest number in an array of data executed.

(ii) To find whether an 8 bits number is even or odd using 8085 Microprocesso

Given number is EVEN number if its lower bit is 0 i.e. low otherwise number is O

To check whether the number is odd or even, we basically perform AND opera with 01 by using ANI instruction. If number is even then we will get 00 otherwise accumulator.

We use 11 to represent odd number and 22 to represent even number.

Algorithm:

1. Load the accumulator with the first data.
2. Perform AND operation with 01 on first data using ANI Instruction.
3. Check if zero flag is set then set the value of accumulator to 22 otherwise accumulator.
4. Now load the result value in memory location.

Program:

```
LDA     2050H
ANI     01
JZ **
MVI     A, 11H
JMP ***
MVI     A, 22H
STA     3050H
HLT
```

Observation:

INPUT:
2050:05
3050:11
OUTPUT:
3052:02
3053:00

Hence we successfully find whether 8 bits number is even or odd.

Q.3.(b) Write a program in 8085 to generate Fibonacci series.

Ans. Refer Q.4. (a) of First Term Examination 2018.

UNIT-II

Q. 4. (a) Explain minimum mode and maximum mode configuration of

Ans. Minimum and Maximum Modes:

• The minimum mode is selected by applying logic 1 to the MN/$\overline{MX}$ input pin. is a single microprocessor configuration.

• The maximum mode is selected by applying logic 0 to the MN/$\overline{MX}$ input pin. This is a multi micro processors configuration.

Minimum Mode 8086 System

• In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/$\overline{MX}$ pin to logic 1.

• In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.

• The remaining components in the system are latches, transceivers, clock generator, memory and I/O devices, Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.

• Latches are generally buffered output D-type flip-flops like 74LS373 or 8282. The are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.

• Transceivers are the bidirectional buffers and some times they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signals.

• They are controlled by two signals namely, $\overline{DEN}$ and DT/$\overline{R}$.

• The $\overline{DEN}$ signals indicates the direction of data, i.e. from or to the processor. The system contains memory for the monitor and users program storage.

• Usually, EPROM is used for monitor storage, while RAM for users program storage. A system may contain I/O devices.

Write Cycle Timing Diagram for Minimum Mode

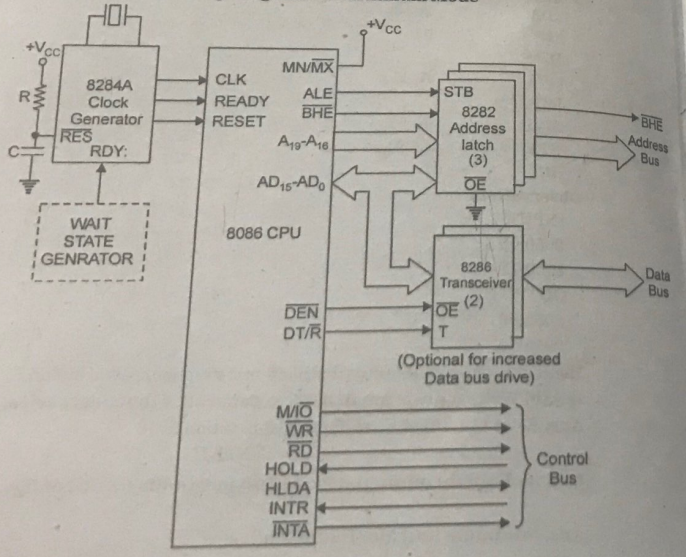


Fig. Typical minimum mode configuration

Maximum Mode 8086 System

• In the maximum mode, the 8086 is operated by strapping the MN/MX pin to ground.

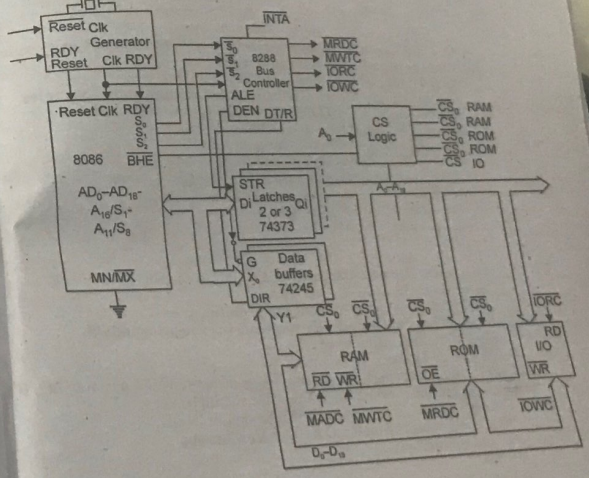
• In this mode, the processor derives the status signal S2, S1, S0. Another chip called bus controller derives the control signal using this status information.

• In the maximum mode, there may be more than one microprocessor in the system configuration.

• The components in the system are same as the minimum mode system.

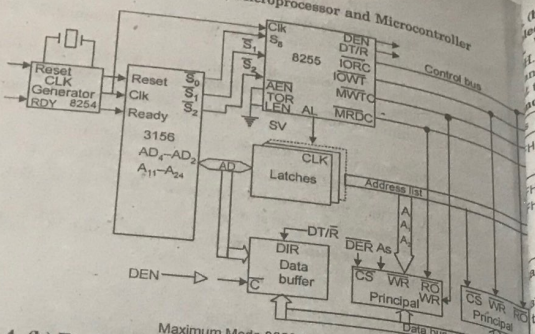
• The basic function of the bus controller chip IC8288 is to derive control signals like $\overline{RD}$ and $\overline{WR}$ (for memory and I/O devices), $\overline{DEN}$ DT/$\overline{R}$, ALE etc. using the information by the processor on the status lines.

• The bus controller chip has input lines S2, S1, S0 and CLK. These inputs to 8288 are driven by CPU.



• It derives the outputs ALE, $\overline{DEN}$, DT/$\overline{R}$, $\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$ and $\overline{AIOWC}$. The AEN, IOB and CEN pins are especially useful for multiprocessor systems.

• AEN and IOB are generally grounded. CEN pin is usually tied to +5V. The significance of the MCE/PDEN output depends upon the status of the IOB pin.



Maximum Mode 8086 System.

Q. 4. (b) Explain various interrupts of 8086 in detail and explain calculate vectored address of interrupts.

Ans. Refer to Q. 3. (b) of page number 3-2017

Q.5.(a) Explain function of instruction with suitable examples.

(i) LEA (ii) DAA (iii) LOOP (iv) LDS (v) IN (vi) PUSH

Ans. (i) LEA Load Effective Address.

Algorithm: REG = address of memory (offset)

(ii) DAA No operands Decimal adjust After Addition.

Corrects the result of addition of two packed BCD values.

Algorithm:

if low nibble of AL > 9 or AF = 1

then: AL = AL + 6 z AF = 1

if AL > 9Fh or CF = 1

then: AL = AL + 60h z CF = 1

(iii) LOOP Decrease CX, jump to label if CX not zero.

Algorithm: CX = CX - 1

if CX <> 0 then { jump else { no jump, continue

(iv) LDS Load memory double word into word register and DS.

Algorithm: REG = first word

DS = second word

(v) IN Input from port into AL or AX. Second operand is a port number. If re
to access port number over 255 - DX register should be used.

Example: IN AX, 4 ; get status of traffic lights. IN AL, 7 ;
get status of stepper-motor. C Z S O P A unchanged
AL, immediate byte
AL, DX
AX, im.byte
AX, DX

(vi) PUSH Store 16 bit value in the stack.

Note: PUSH immediate works only on 80186 CPU and later!

Algorithm: SP = SP - 2
SS:[SP] (top of the stack) = operand

Example: MOV AX, 1234h
PUSH AX
POP DX ; DX = 1234h
RET

(b) Intereface two 4kx8 EPROM chips and two 4kx8 RAM chips with lect suitable memory maps. **(6.5)**

We know that, after reset, the IP and CS are initialised to form address ff. Hence, this address must lie in the EPROM. The address of RAM may be ny where in the 1MB address space of 8086, but we will select the RAM address the address map of the system is continuous.

mory Map

	A_{19}	A_{18}	A_{17}	A_{16}	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	
FH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	EPROM 8K × 8
FH	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	RAM 8K × 8
	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

al 8K bytes of EPROM need 13 address lines A_0–A_{12} (since 2^{13} = 8K). Address –A_{19} are used for decoding to generate the chip select. The $\overline{BHE}$ signal goes low transfer is at odd address or higher byte of data is to be accessed. Let us assume latched address, $\overline{BHE}$ and demultiplexed data lines are modily available for ing. Figure shows the interfacing diagram for the memory system.

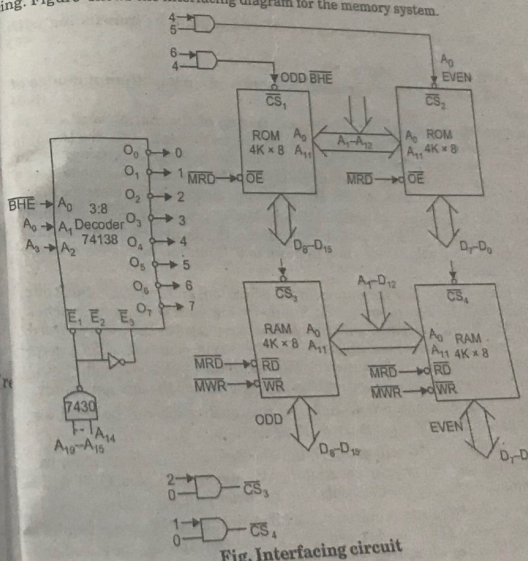


Fig. Interfacing circuit

The memory system in this example contains in total four 4K × 8 memory chips. The two 4K × 8 chips of RAM and ROM are arranged in parallel to obtain 16-bit data bus width. If A_0 is 0 i.e. the address is even and is in RAM, then the lower RAM chip is selected indicating 8-bit transfer at an even address. If A_0 is 1 i.e. the address is odd and is in RAM, the $\overline{BHE}$ goes low, the upper RAM chip is selected, further indicating that the 8-bit transfer is at an odd address. If the selected addresses are in ROM, then respective ROM chips are selected. If at a time A_0 and $\overline{BHE}$ both are 0, both the RAM or ROM chips are selected, i.e. the data transfer is of 16 bits. The selection of chips here takes place as shown in Table.

Table Memory Chip selection for Problem

Decoder I/P →	A_2	A_1	A_0	
Address/$\overline{BHE}$ →	A_3	A_0	$\overline{BHE}$	Selection/ Comment
Word transfer on $D_0 - D_{15}$	0	0	0	Even and odd addresses in RAM
Byte transfer on $D_7 - D_0$	0	0	1	Only even address in RAM
Byte transfer on $D_8 - D_{15}$	0	1	0	Only odd address in RAM
Word transfer on $D_0 - D_{15}$	1	0	0	Even and odd addresses in ROM
Byte transfer on $D_0 - D_7$	1	0	1	Only even address in ROM
Byte transfer on $D_8 - D_{15}$	1	1	0	Only odd address in ROM

UNIT-III

Q. 6. (a) Draw the block diagram of 8255 and explain its interfacing with 8086. (6)

Ans. Refer to Q. 6. (a) of page number 15-2017

Q. 6. (b) Draw the block diagram of 8253 and explain its different modes of operations. (6.5)

Ans. Operation Modes: The D3, D2, and D1 bits of the control word set the operating mode of the timer. These are 6 modes in total; for modes 2 and 3, the D3 bit is ignored, so the missing modes 6 and 7 are aliases for modes 2 and 3. Notice that, for modes 0, 2, 3 and 4, GATE must be set to HIGH to enable counting. For mode 5, the rising edge of GATE starts the count. For details on each mode, see the reference links.

Mode 0 (000): Interrupt on Terminal Count: Mode 0 is used for the generation of accurate time delay under software control. In this mode, the counter will start counting from the initial COUNT value loaded into it, down to 0. Counting rate is equal to the input clock frequency.

The OUT pin is set low after the Control Word is written, and counting starts one clock cycle after the COUNT programmed. OUT remains low until the counter reaches 0, at which point OUT will be set high until the counter is reloaded or the Control Word is written. The Gate signal should remain active high for normal counting. If Gate goes low counting gets terminated and current count is latched till Gate pulse goes high again, the first byte of the new count when loaded in the count register.stop the previous count.

Mode 1 (001): Programmable One Shot: In this mode 8253 can be used as Monostable Multivibrator. GATE input is used as trigger input.

OUT will be initially high. OUT will go low on the Clock pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration.

The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT. If a new count is written to the Counter during a oneshot pulse, the current one-shot is not affected unless the counter is retriggered. In that case, the Counter is loaded with the new count and the oneshot pulse continues until the new count expires.

Mode 2 (X10): Rate Generator: In this mode, the device acts as a divide-by-n counter, which is commonly used to generate a real-time clock interrupt.

Like other modes, counting process will start the next clock cycle after COUNT is sent. OUT will then remain high until the counter reaches 1, and will go low for one clock pulse. OUT will then go high again, and the whole process repeats itself.

The time between the high pulses depends on the preset count in the counter's register, and is calculated using the following formula:

$$\text{Value to be loaded into counter} = \frac{f_{\text{input}}}{f_{\text{output}}}$$

Note that the values in the COUNT register range from n to 1; the register never reaches zero.

Mode 3 (X11): Square Wave Generator: This mode is similar to mode 2. However, the duration of the high and low clock pulses of the output will be different from mode 2.

Suppose is the number loaded into the counter (the COUNT message), the output will be

- high for $\frac{n}{2}$ counts, and low for $\frac{n}{2}$ counts, if n is even.
- high for $\frac{n+1}{2}$ counts, and low for $\frac{n-1}{2}$ counts, if n is odd.

Mode 4 (100): Software Triggered Strobe

After Control Word and COUNT is loaded, the output will remain high until the counter reaches zero. The counter will then generate a low pulse for 1 clock cycle (a strobe) - after that the output will become high again.

Mode 5 (101): Hardware Triggered Strobe

This mode is similar to mode 4. However, the counting process is triggered by the GATE input.

After receiving the Control Word and COUNT, the output will be set high. Once the device detects a rising edge on the GATE input, it will start counting. When the counter reaches 0, the output will go low for one clock cycle - after that it will become high again, to repeatable and cycle on the next rising edge of GATE.

Q. 7. (a) Interface 8 bit Analog to Digital converter with microprocessor using 8255 IC. Draw the block diagram of interfacing; decode hex address for the ports and write and ALP for its functioning. (10)

Ans. Interfacing Analog to Digital Data Converters

- Interfacing ADC 0808 with 8086 using 8255 ports. Use port A of 8255 for transferring digital data output of ADC to the CPU and port C for control signals.

Assume that an analog input is present at I/P_2 of the ADC and a clock input of suitable frequency is available for ADC.

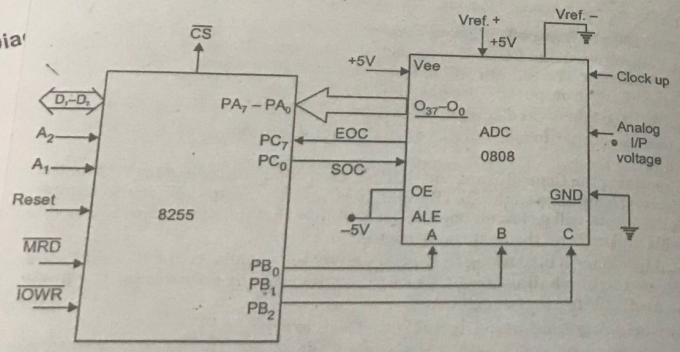
The analog input I/P_2 is used and therefore address pins A,B,C should be 0, 1, 0 respectively to select I/P_2. The OE and ALE pins are already kept at +5V to side the ADC and enable the outputs. Port C upper acts as the input port to receive the DC signal while port C lower acts as the output port to send SOC to the ADC.

Port A acts as a 8-bit input data port to receive the digital data output from the . The 8255 control word is written as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1
1	0	1	1	0	0	0

The required ALP is as follows:

```
MOV     AL,98h      ;initialise 8255 as
OUT     CWR,AL      ;discussed above.
MOV     AL,02h      ;Select I/P2 as analog
OUT     Port B,AL   ; input.
MOV     AL,00h      ; Give start of conversion
OUT     Port C, AL  ; pulse to the ADC
MOV     Al,01h
OUT     Port C,AL
MOV     AL,00h
OUT     Port C, AL
WAIT: IN AL,Port C  ;Check for EOC by
                    ; reading port C upper and
RCR                 ; rotating through carry.
JNC     WAIT        ;If EOC, read digital equivalent;inAL
IN      AL, Port A
HLT;    Stop.
```



Interfacing 0808 with 8086

Q. 7. (b) Explain asynchronous mode of USART 8251? (2.5)

Ans. *The 8251 is a USART (Universal Synchronous Asynchronous Receiver smitter) for serial data communication. As a peripheral device of a microcomputer m, the 8251 receives parallel data from the CPU and transmits serial data after ersion. This device also receives serial data from the outside and transmits parallel to the CPU after conversion.*

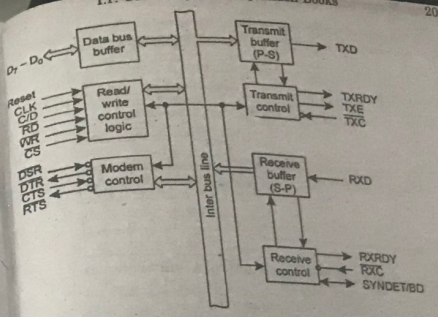


Fig. 1. Block diagram of the 8251 USART (Universal Synchronous Asynchrounous Receiver Transmitter)

8251 functional configuration is programed by software. Operation between 1 and a CPU is executed by program control. Table 1 shown the operation a CPU and the device.

Table 1: Operation between a CPU and 8251

	C/D	RD	WR	
	×	×	×	Data Bus 3-State
	×	1	1	Data Bus 3-State
	1	0	1	Status → CPU
	1	1	0	Control Word← CPU
	0	0	1	Data→CPU
	0	1	0	Data ← CPU

ntrol Words

ere are two types of control word.

Mode instruction (setting of function)

Command (setting of operation)

Mode Instruction: Mode instruction is used for setting the function of the 8251 nstruction will be in "wait for write" at either internal reset or external reset. s, the writing of a control word after resetting will be recognized as a "mode ction."

ems set by mode instruction are as follows:

Synchronous/asynchronous mode

Stop bit length (asynchronous mode)

Character length

Parity bit

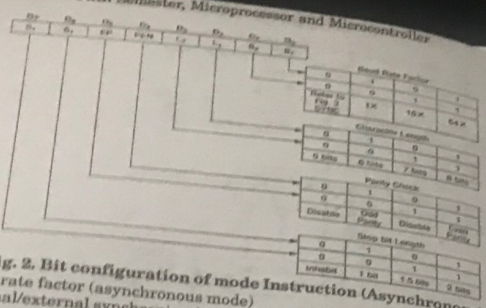


Fig. 2. Bit configuration of mode Instruction (Asynchronous)

- Baud rate factor (asynchronous mode)
- Internal/external synchronization (synchronous mode)
- Number of synchronous characters (Synchronous mode)

The bit configuration of mode instruction is shown in Figures 2 and 3. In the case of synchronous mode, it is necessary to write one-or two byte sync characters. If sync characters were written, a function will be set because the writing of sync character constitutes part of mode instruction.

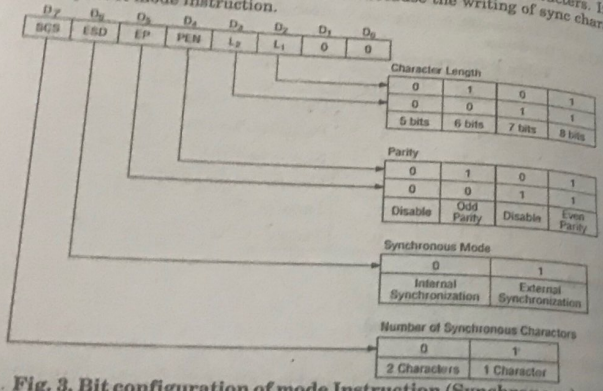


Fig. 3. Bit configuration of mode Instruction (Synchronous)

Command: Command is used for setting the operation of the 8251. It is possible to write command whenever necessary after writing a mode instruction and sync characters.

Items to be set by command are as follows:

- Transmit Enable/Bisable
- Receive Enable/UiBable
- DTK, RTS Output of data.
- Resetting of error flag.
- Sending to break characters

Q. 8. (a) Explain the 8051 microcontroller block diagram in detail. (6)

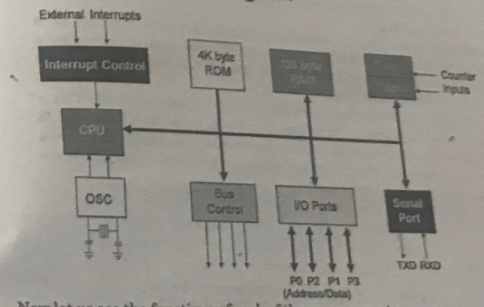
Ans. Major components of Intel 8051 microcontroller

The 8051 microcontroller is an 8-bit microcontroller. The major components of 8051 microcontroller and their functions.

An 8051 microcontroller has the following 12 major components:

1. ALU (Arithmetic and Logic Unit)
2. PC (Program Counter)
3. Registers
4. Timers and counters
5. Internal RAM and ROM
6. Four general purpose parallel input/output ports
7. Interrupt control logic with five sources of interrupt
8. Serial date communication
9. PSW (Program Status Word)
10. Data Pointer (DPTR)
11. Stack Pointer (SP)
12. Data and Address bus.

The 8051 Block Diagram



Now let us see the functions of each of these componenta.

1. ALU: All arithmetic and logical functions are carried out by the ALU. Addition, subtraction with carry, and multiplication come under arithmetic operations. Logical AND, OR and exclusive OR (XOR) come under logical operations.

2. Program counter (PC): A program counter is a 16-bit register and it has no internal address. The basic function of program counter is to fetch from memory the adress of the next instruction to be executed. The PC holds the address of the next instruction residing in memory and when a command is encountered, it produces that instruction. This way the PC increments automatically, holding the address of the next instruction.

3. Registers: Registers are usually known as data storage devices. 8051 microcontroller has 2 registers, namely Register A and Register B. Register A serves as an accumulator while Register B functions as a general purpose register. These registers are used to store the output of mathematical and logical instructions.

The operations of addition, subtraction, multiplication, and division are carried out by Register A. Register B is usually unused and comes into picture only when multiplication and division functions are carried out by Register A. Register A also involved in data transfers between the microcontroller and external memory.

8051 microcontroller also has 7 Special Function Registers (SFRs). They are:

1. Serial Port Data Buffer (SBUF)
2. Timer/Counter Control (TCON)

3. Timer/Counter Mode Control (TMOD) 4. Serial Port Control (SCON)
5. Power Control (PCON) 6. Interrupt Priority (IP)
7. Interrupt Enable Control (IE)

4. Timers and Counters: Synchronization among internal operations can be achieved h the help of clock circuits which are responsible for generating clock pulses. During h clock pulse a particular operation will be carried out, thereby, assuring synchronization ong operations. For the formation of an oscillator, we are provided with two pins AL1 and XTAL2 which are used for connecting a resonant network in 8051 rocontroller device. In addition to this, circuit also consists of four more pins. They are, ses. With each clock pulse, a particular function will be accomplished and hence chronization is achieved. There are two pins XTAL1 and XTAL2 which form an illator circuit which connect to a resonant network in the microcontroller. The circuit o has 4 additional pins .

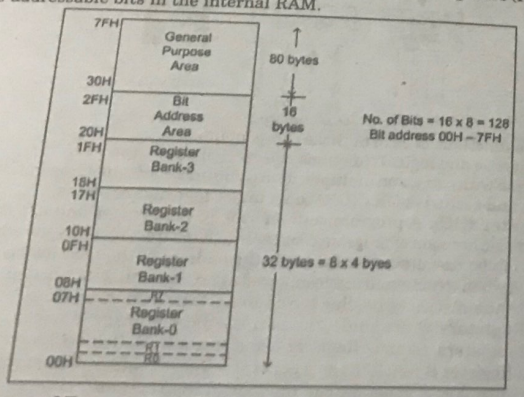
1. EA: External enable 2. ALE: Address latch enable
3. PSEN: Program store enable and 4. RST: Reset.

Quartz crystal is used to generate periodic clock pulses.

5. Internal RAM and ROM

ROM: A code of 4K memory is incorporated as on-chip ROM in 8051. The 8051 ROM non-volatile memory meaning that its contents cannot be altered and hence has a ilar range of data and program memory, i.e, they can address program memory as l as a 64K separate block of data memory.

RAM: The 8051 microcontroller is composed of 128 bytes of internal RAM. This is olatile memory since its contents will be lost if power is switched off. These 128 es of internal RAM are divided into 32 working registers which in turn constitute 4 ister banks (Bank 0-Bank 3) with each bank consisting of 8 registers (R0 - R7). There 128 addressable bits in the internal RAM.



6. Four General Purpose Parallel Input/Output Ports: The 8051 microcontroller four 8-bit input/output ports. These are: PORT PO: When there is no external memory sent, this port acts as a general purpose input/output port. In the presence of external mory, it functions as a multiplexed address and data bus. It performs a dual role.

PORT P1: This port is used for various interfacing activities. This 8-bit port normal I/O port i.e. it does not perform dual functions.

PORT P2: Similar to PORT PO, this port can be used as a general purpose port w there is no external memory but when external memory is present it works in conjunc with PORT PO as an address bus. This is an 8-bit port and performs dual functions

PORT P3: PORT P3 behaves as a dedicated I/O port

PSW (Program status word) is an 8-bit register. It is used to indicate some condit that result after an instruction is executed.

CY	AC	F0	RS1	RS0	OV	–	P

Cy →Carry Flag → shows carry or borrow after addition or subtraction.

AC →Auxilliary carry flag→ it is set if there is a carry from D_3 to D_4 during an or SUB operation.

P → Parity Flag → it reflects the number if 1's in Accumulator. P = 1, for odd n 1's & P = 0 for even no. of 1's

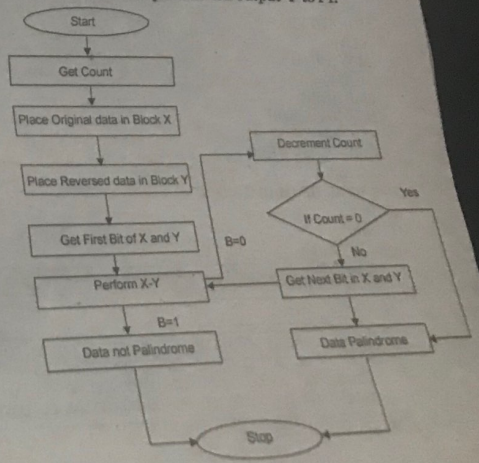
OV → overflow flag → this flag is set whenever the result of a signed no. opera is too large, causing the high-order bit to overflow into the sign bit

RS1, RS0 → Register bank select bits

RS1	RS0	Reg. bank
0	0	0
0	1	1
1	0	2
1	1	3

F0 → Available to the user for general purpose.

Q. 8. (b) Write an ALP to check if the character string of length 7, store RAM locations 40H onwards is a palindrome output 'Y' to P1.



Program:-

```
ORG 0000H
MOV DPTR,#MYDATA
MOV R2,#06H
MOV R3,#06H
MOV R1,#20H
MOV R0,#30H
LOOP: CLR A
MOVC A,@A+DPTR
MOV @R1,A
INC R1
INC DPTR
DJNZ R2,LOOP
DEC R1
LOOP 1: CLR A
MOV A,@R1
MOV @R0,A
DEC R1
INC R0
DJNZ R3,LOOP1
MOV R7,#06H
MOV R1,#20H
MOV R0,#30H
YES: MOV A,@R1
MOV  B,@R0
CJNE A,B,NO
INC R1
INC R0
DJNZ R7,YES
CLR A
MOV A,#11H
MOV 40H,A
JMP HERE
NO: MOV A,#00H
MOV 40H,A
HERE : SJMP HERE
ORG 250H
MY DATA: DB "MALAYALAM"
END
```

Output:-

Input Window:

Memory 1

Address: c.250h

C:0×0250:	4D	41	4C	41	59	41	4C	41	4D	00	00	00	00	00
C:0×025E:	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C:0×026C:	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C:0×027A:	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Output Window:

Memory 1

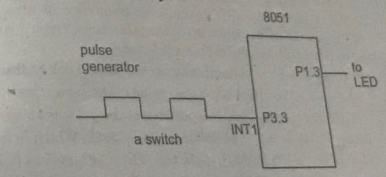
Address: 40h

I:0 × 40:	11	00	00	00	00	00	00	00	00	00	00	00	00	00
I:0 × 4E:	00	00	00	00	00	00	00	00	00	00	00	00	00	00
I:0 × 5C:	00	00	00	00	00	00	00	00	00	00	00	00	00	00
I:0 × 6A:	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Q.9.(a) Write a program to implement Interrupt INT0, INT1. GLow all LEDs of P0 when INT0 is activated and glow all LEDs of P2 when INT1 is activated. (6)

Ans. Assuming that pin 3.3 (INT1) is conneted to a pulse generator, write a program in which the falling edge of the pulse will send a high to P1.3 which is connected to an LED (or buzzer). In other words, the LED is turned on and off at the same rate as the pulse are applied to the INT1 pin.

This is an edge-triggered version of Example. But in this example, to turn on the LED again, the INT1 pulse must be brought back high and then forced low to create a falling edge to activate the interrupt.



Example:

```
        ORG     0000H
        LJMP    MAIN
;ISR for hardware interrpt INT1 to turn on the LED
        ORG     0013H
        SETB    P1.3
        MOV     R3, #255
BACK:   DJNZ    R3, HERE  ; keep the buzzer on for a while
        CLR     P1.3
        RETI
;--MAIN program for initialization
MAIN:   SETB    TCON.2    ;make INT1 edge-trigger interrupt
        MOVE    IE, #10000100B   ;enable External INT 1
HERE:   SJMP    HERE
        END
```

Q. 9. (b) Write a program to generate two square waves of frequency 5KHz at pin p1.3 and another of frequency 25 KHz at pin p2.3. Assume XTAL = 22 MHz.

Ans. (i) The timer works with a clock frequency of 1/12 of the XTAL frequency, therefore, we have 22 MHz / 12 = 1.833 MHz as the timer frequency. As a result, each clock has a period of T = 1 / 1.833 MHz = 0.545μs (is. In other words, Timer 1 counts up each 0.545μs resulting in delay = number of counts x 0.545μs.

we must toggle the bit to generate the square wave. Look at the following ste

1. T = 1 / f = 1 / 5 kHz = 200 µs the period of the square wave.
2. 1/2 of it for the high and low portions of the pulse is 100 µs.
3. 100 us / 0.545 µs = 184 and 65536 – 184 = 65352. Which in hex is FF48
4. TL = 48H and TH = FFH. all in hex. The program is as follows.

```
        MOV TMOD, #10H      ;Timer 1, mode 1(16 bit)
AGAIN:  MOV TL1, #48H       ;TL1=48H, Low byte
        MOV TH1, #FFH       ;TH1=FFH, High byte
        SETB TR1            ;Start Timer 1
BACK:   JNB TF1, BACK       ;Stay until timer rolls over
        CLR TR1             ;Stop Timer 1
        CPL P1.3            ;Complement P1.3 to get hi, lo
        CLR TF1             ;Clear timer since mode 1
        SJMP AGAIN          ;Reload timer since mode 1 is not auto r
```

(ii) The timer works with a clock frequency of 1/12 of the XTAL frequency; therefor have 22 MHz / 12 = 1.833 MHz as the timer frequency. As a result, each clock p period of T = 1 / 1.833 MHz = 0.545µs (is. In other words, Timer 1 counts up each 0.5 resulting in delay = number of counts x 0.545µs.

we must toggle the bit to generate the square wave. Look at the following steps

5. T = 1 / f = 1 / 25 kHz = 40 µs the period of the square wave.
6. 1/2 of it for the high and low portions of the pulse is 20 µs.
7. 20 us / 0.545 µs = 37 and 65536 – 184 = 65352. Which in hex is FFDBH.
8. TL = DBH and TH = FFH. all in hex. The program is as follows.

```
        MOV TMOD, #10H      ;Timer 1, mode 1(16 bit)
AGAIN:  MOV TL1, #DBH       ;TL1=48H, Low byte
        MOV TH1, #FFH       ;TH1=FFH, High byte
        SETB TR1            ;Start Timer 1
BACK:   JNB TF1, BACK       ;Stay until timer rolls over
        CLR TR1             ;Stop Timer 1
        CPL P2.3            ;Complement P2.3 to get hi, lo
        CLR TF1             ;Clear timer since mode 1
        SJMP AGAIN          ;Reload timer since mode 1 is not auto r
```