**MICROPROCESSORS AND MICROCONTROLLERS SHORT ANSWER QUESTIONS**

**UNIT:1**

1. What is a Microprocessor?

 Microprocessor is a CPU fabricated on a single chip, program-controlled device, which fetches the instructions from memory, decodes and executes the instructions.

1. What are the basic units of a microprocessor ?

 The basic units or blocks of a microprocessor are ALU, an array of registers and control unit.

1. What is Software and Hardware?

The Software is a set of instructions or commands needed for performing a specific task by a programmable device or a computing machine. • The Hardware refers to the components or devices used to form computing machine in which the software can be run and tested. Without software the Hardware is an idle machine.

1. What is the disadvantage of microprocessor?

 It has limitations on size of data. Most Microprocessor does not support floating-point operations.

1. What is the difference between microprocessor and microcontroller?

 In Microprocessor more op-codes, few bit handling instructions. But in Microcontroller: fewer op-codes, more bit handling Instructions, and also it is defined as a device that includes micro processor, memory, & input output signal lines on a single chip.

1. What is Instruction Set?

• It is the set of the instructions that the Microprocessor can execute

1. What is an instruction

 • An instruction is an order given to a computer processor by a computer program. At the lowest level, each instruction is a sequence of 0s and 1s that describes a physical operation the computer is to perform (such as "Add") and, depending on the particular instruction type, the specification of special storage areas called registers that may contain data to be used in carrying out the instruction, or the location in computer memory of data.

1. What is Instruction cycle?

 The sequence of operations that a processor has to carry out while executing the instruction is called Instruction cycle. Each instruction cycle of a processor indium consists of a number of machine cycles.

1. Explain the function of CPU in Microprocessor

• A microprocessor controls all functions of the CPU, or central processing unit, of a computer or other digital device. The microprocessor is programmed to give and receive instructions from other components of the device. The system can control everything from small devices such as calculators and mobile phones, to large automobiles.

1. Distinguish between Microprocessor & Microcontroller.

• The microprocessor is a digital integrated circuit device that can be programmed with a series of instructions to perform specified functions on data. • But micro controller is a computer on a chip which has memory, input,ouput on the chip itself. • So, micro processor can perform only few functions but micro controller can perform so many functions.

1. What is a register •

In computer architecture, a processor register (or general purpose register) is a small amount of storage available on the CPU whose contents can be accessed more quickly than storage available elsewhere. Typically, this specialized storage is not considered part of the normal memory range for the machine. Processor registers are at the top of the memory hierarchy, and provide the fastest way for a CPU to access data.

1. What is nibble, byte and word

• NIBBLE:-In computing, a nibble is a four-bit aggregation or half an octet. As a nibble contains 4 bits.

• BYTE:-The byte is a unit of digital information in computing and telecommunications. The size of a byte is typically hardware dependent, but the modern de facto standard is 8 bits, as this is a convenient power of 2.

• WORD:- The most common microprocessors used in personal computers are an example of this; their IA-32 architecture is an extension of the original Intel 8086 design which had a word size of 16 bits.

1. Which microprocessor accepts the program written for 8086 without any changes?

• 8088

1. What is cache memory?

 • Cache memory is a small high-speed memory. It is used for temporary storage of data & information between the main memory and the CPU (center processing unit). The cache memory is RAM.

1. What is flag?

Flag is a flip-flop used to store the information about the status of a processor and the status of the instruction executed most recently.

1. How many instructions can be executed per second in 8086/88?

 • 2.5 million

1. What is Logical Address:?

 • A memory address on the 8086 consists of two numbers, usually written in hexadecimal and separated by a colon, representing the segment and the offset. This combination of segment and offset is referred to as a logical address • Logical address=segment: offset

1. What is a bus?

 • Bus is a group of conducting lines that carries data, address and control signals.

1. Why data bus is bi-directional?

 • The microprocessor has to fetch (read) the data from memory or input device for processing and after processing, it has to store (write) the data to memory or output device. Hence the data bus is bi-directional.

1. Why address bus is unidirectional?

 • The address is an identification number used by the microprocessor to identify or access a memory location or I / O device. It is an output signal from the processor. Hence the address bus is unidirectional.

1. What is the data and address size in 8086?

 • The 8086 can operate on either 8-bit or 16-bit data. The 8086 uses 20 bit address to access memory and 16-bit address to access 1/0 devices.

1. Write the flags of 8086.

 • The 8086 has nine flags and they are 1. Carry Flag (CF) 6. Overflow Flag (OF) 2. Parity Flag (PF) 7. Trace Flag (TF) 3. Auxiliary carry Flag (AF) 8. Interrupt Flag (IF) 4. Zero Flag (ZF) 9. Direction Flag (DF) 5. Sign Flag (SF)

1. Explain the function of M/IO in 8086.

The signal M/IO is used to differentiate memory address and 1/0 address When the processor is accessing memory locations MI 10 is asserted high and when it is accessing 1/0 mapped devices it is asserted low.

1. What are the functions of BIU?

 • The BIU contains the circuit for physical address calculations and a pre - coding instruction byte queue & it makes the bus signals available for external interfacing of the devices

1. What are the functions of EU?

 • The EU contains the register set of 8086 except segment registers and IP. It has a 16-bit ALU, able to perform arithmetic and Logic operations.

1. What is the size of instruction queue in 8086?

 The queue length depends on the fetching speed and the execution speed. Sometime queue may be restricted due to the space available on the CPU chip.

1. What is called .Scratch pad of computer.?

 Cache Memory is scratch pad of computer.

1. Why does microprocessor contain ROM chips?

• Microprocessor contain ROM chip because it contain instructions to execute data.

1. What is meant by LATCH?

• Latch is a D- type flip-flop used as a temporary storage device controlled by a timing signal, which can store 0 or 1. The primary function of a Latch is data storage. It is used in output devices such as LED, to hold the data for display.

1. Difference between static and dynamic RAM?

 Static RAM: No refreshing, 6 to 8 MOS transistors are required to form one memory cell, Information stored as voltage level in a flip flop. Dynamic RAM: Refreshed periodically, 3 to 4 transistors are required to form one memory cell, Information is stored as a charge in the gate to substrate capacitance.

1. What are the interrupts of 8086?

 • The interrupts of 8085 are INTR and NMI. The INTR is general maskable interrupt and NMI is non-maskable interrupt.

1. What is Software interrupts?

• The Software interrupts are program instructions. These instructionsare inserted at desired locations in a program. While running a program, if software interrupt instruction is encountered then the processor executes an interrupt service routine.

1. What is Hardware interrupt?

• If an interrupt is initiated in a processor by an appropriate signal at theinterrupt pin, then the interrupt is called Hardware interrupt.

1. What is the position of the Stack Pointer after the POP instruction?

• The address is 02 greater than the earlier value.

1. Logic calculations are done in which type of registers?

• Accumulator is the register in which Arithmetic and Logic calculations are done.

1. Why crystal is a preferred clock source?

 • Because of high stability, large Q (Quality Factor) & the frequency that doesn’t drift with aging. Crystal is used as a clock source most of the times.

1. What is Program counter?

 • Program counter holds the address of either the first byte of the next instruction to be fetched for execution or the address of the next byte of a multi byte instruction, which has not been completely fetched. In both the cases it gets incremented automatically one by one as the instruction bytes get fetched. Also Program register keeps the address of the next instruction.

1. What is Tri-state logic?

• Three Logic Levels are used and they are High, Low, High impedance state. The high and low are normal logic levels & high impedance state is electrical open circuit conditions. • Tri-state logic has a third line called enable line.

1. How clock signal is generated in 8086?

What is the maximum internal clock frequency of 8086? • The 8086 does not have on-chip clock generation circuit. Hence the clock generator chip, 8284 is connected to the CLK pin of8086. The clock signal supplied by 8284 is divided by three for internal use. The maximum internal clock frequency of8086 is 5MHz.

1. What is PSW.

 Give its structure The current state of the processor is stored in a register called ProcessorStatus Word(PSW).The PSW contains bits which indicate such things as whether the previous arithmetic operations produced a positive,negative or zero result. If a subtract instruction is followed by a "branch on zero" instruction ,then the branch will be taken if the PSW indicates that the subtraction resulted in a zero. Most loops,such as DO-WHILE,FOR,etc involve incrementing or decrementing a counter and repeating the loop until the counter reaches the limit.Each time the counter is changed,the result is compared with the limit,the PSW is set accordingly,and branch is taken or not depending on the contents of the PSW. The 8086 microprocessor has a 16-bit PSW.

1. What is the use of HLDA

• HLDA is the acknowledgment signal for HOLD. It indicates whether the HOLD signal is received or not. • HOLD and HLDA are used as the control signals for DMA operations.

1. Define Pipelining?

 • In 8086,to speedup the execution program, the instructions fetching and execution of instructions are overlapped each other this is known as Pipelining.

of the register. • Where rotate command makes "wrap around" at the end of the register.

1. What is the main use of ready pin?

• READY is used by the microprocessor to check whether a peripheral is ready to accept or transfer data. • A peripheral may be a LCD display or analog to digital converter or any other. • These peripherals are connected to microprocessor using the READY pin. • If READY is high then the periphery is ready for data transfer. If not the microprocessor waits until READY goes high.

1. What is the need for timing diagram?

• The timing diagram provides information regarding the status of various signals, when a machine cycle is executed. The knowledge of timing diagram is essential for system designer to select matched peripheral devices like memories, latches, ports, etc., to form a microprocessor system.

1. Why status signals are provided in microprocessor?

• The status signals can be used by the system designer to track the internal operations of the processor. Also, it can be used for memory expansion (by providing separate memory banks for program & data and selecting the bank using status signals).

1. Define machine cycle.

Machine cycle is defined as the time required to complete one operation of accessing memory, I/O, or acknowledging an external request. This cycle may consist of three to six T-states.

1. Define T-State •

T-State is defined as one subdivision of the operation performed in one clock period. These subdivisions are internal states synchronized with the system clock, and each T-State is precisely equal to one clock period.

1. What is the difference between CPU bus and system bus?

 • The CPU bus has multiplexed lines but the system bus has separate lines for each signal. (The multiplexed CPU lines are demultiplexed by the CPU interface circuit to form system bus).

1. What is fetch and execute cycle?

 • In general, the instruction cycle of an instruction can be divided intofetch and execute cycles. The fetch cycle is executed to fetch the opcode from memory. The execute cycle is executed to decode the instruction and to perform the work instructed by the instruction.

1. Name the special registers available in 8086?

• IP, CS, DS, SS, ES

1. Name the flags available in 8086?

 • Control flags . Direction, Interrupt, Trap ; • Condition flags . CY, AC, S, Z, P, OV

1. How many types memory mgt can divided? • they are two types., 1:even bank 2:odd bank. •

**53. What is an instruction queue? Explain?**

**Ans.**  This is introduced in 8086 processor.This queue is in the BIU and is used for storing the predecoded instructions. This will overlap the fetching and execution cycle.

The E U will take the instructions from the queue for decoding and execution.

**54. What is stack? Explain the use and operation of stack and stack pointer?**

**Ans.** A stack is a portion of the memory used for the temporary storage. A stack is a last

 In first Out memory. A stack grows in the decreasing order. A stack will hold the

 temporary information’s push and pop are the instructions used for storing and

 accessing data from the stack. Contents can be moved as 16 bit only using push and

 pop instructions.

**55. What are the flags in 8086?**

**Ans.** In 8086 Carry flag, Parity flag, Auxiliary carry flag, Zero flag, Overflow flag,

 Trap flag, Interrupt flag, Direction flag, and Sign flag.

**56. What are the various interrupts in 8086? Explain.**

**Ans.** Maskable interrupts, Non-Maskable interrupts.

i) An interrupt that can be turned off by the programmer is known as Maskable

 interrupt.

ii) An interrupt which can be never be turned off (ie.disabled) by the programmer

 is known as Non-Maskable interrupt.

**57. Which interrupts are generally used for critical events?**

**Ans.** Non-Maskable interrupts are used in critical events. Such as Power failure, Emergency, Shut off etc.

**58. Explain different types of registers in 8086 microprocessor arch.**

**Ans.** Most of the registers contain data/instruction offsets within 64 KB memory segment. There are four different 64 KB segments for instructions, stack, data and extra data. To specify where in 1 MB of processor memory these 4 segments are located the processor uses four segment registers:

**Code segment** (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during far jump, far call and far return instructions.

**Stack segment** (SS) is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP) registers is located in the stack segment. SS register can be changed directly using POP instruction.

**Data segment** (DS) is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general registers (AX, BX, CX, DX) and index register (SI, DI) is located in the data segment. DS register can be changed directly using POP and LDS instructions.

**Extra segment** (ES) is a 16-bit register containing address of 64KB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions.

It is possible to change default segments used by general and index registers by prefixing instructions with a CS, SS, DS or ES prefix.

All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. The general registers are:

**Accumulator** register consists of 2 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX. AL in this case contains the low-order byte of the word, and AH contains the high-order byte. Accumulator can be used for I/O operations and string manipulation.

**Base** register consists of 2 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX. BL in this case contains the low-order byte of the word, and BH contains the high-order byte. BX register usually contains a data pointer used for based, based indexed or register indirect addressing.

**Count** register consists of 2 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. When combined, CL register contains the low-order byte of the word, and CH contains the high-order byte. Count register can be used as a counter in string manipulation and shift/rotate instructions.

**Data** register consists of 2 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX. When combined, DL register contains the low-order byte of the word, and DH contains the high-order byte. Data register can be used as a port number in I/O operations. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

The following registers are both general and index registers:

**Stack Pointer** (SP) is a 16-bit register pointing to program stack.

**Base Pointer** (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.

**Source Index** (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data address in string manipulation instructions.

**Destination Index** (DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data address in string manipulation instructions.

**Unit:2**

**SHORT QUESTIONS & ANSWERS**

**1. Write an ALP to find factorial of number for 8086.**

MOV AX, 05H

MOV CX, AX

Back: DEC CX

MUL CX

LOOP back

; results stored in AX

; to store the result at D000H

MOV [D000], AX

HLT

**2. The 8 data bytes are stored from memory location E000H to E007H. Write 8086 ALP to**

**transfer the block of data to new location B001H to B008H.**

MOV BL, 08H

MOV CX, E000H

MOV EX, B001H

Loop: MOV DL, [CX]

MOV [EX], DL

DEC BL

JNZ loop

HLT

**3. Write a program to reverse the given string for 8086.**

Title reverse the given string

Dosseg

.model small

.stack 100h

.data

String1 db .assembly language program., $

Length dw $-String1-1

.code

Main proc

MOV AX, data

MOV DS, AX

MOV SI, offset String1

MOV CX, Length

ADD SI, CX

Back: MOV DL, [SI]

MOV AH, 02H

INT 21H

DEC SI

LOOP Back

MOV AH, 4CH

INT 21H

Main endp

End Main

4. **Write a program to multiply 2 numbers (16-bit data) for 8086.**

Title multiply two numbers

.stack 100h

.data

Multiplier dw 1234H

Multiplicant dw 3456H

Product dw ?

.code

MULT proc

MOV AX, @data

MOV DS, AX

MOV AX, Multiplicant

MUL Multiplier

MOV Product, AX

MOV Product+2, DX

MOV AH, 4CH

INT 21H

MULT endp

End MULT

5. **Sum of series of 10 numbers and store result in memory location total.**

Title Sum of series

Dosseg

.model small

.stack 100h

.data

List db 12,34,56,78,98,01,13,78,18,36

Total dw ?

.code

Main proc

MOV AX, @data

MOV DS, AX

MOV AX, 0000H

MOV CX, 0AH ; counter

MOV BL, 00H ; to count carry

MOV SI, offset List

Back: ADD AL, [SI]

JC Label

Back1: INC SI

LOOP Back

MOV Total, AX

MOV Total+2, BL

MOV AH, 4CH

INT 21H

Label: INC BL

JMP Back1

Main endp

End Main

6. **Write a program to find Largest No. in a block of data. Length of block is 0A. Store the**

**maximum in location result.**

Title maximum in given series

Dosseg

.model small

.stack 100h

.data

List db 80, 81, 78, 65, 23, 45, 89, 90, 10, 99

Result db ?

.code

Main proc

MOV AX, @data

MOV DS, AX

MOV SI, offset List

MOV AL, 00H

MOV CX, 0AH

Back: CMP AL, [SI]

JNC Ahead

MOV AL, [SI]

Ahead: INC SI

LOOP Back

MOV Result, AL

MOV AH, 4CH

INT 21H

Main endp

End Main

7. **Find number of times letter .e. exist in the string .exercise., Store the count at memory**

**ans.**

Title string operation

Dosseg

.model small

.stack 100h

.data

String db .exercise., $

Ans db ?

Length db $-String

.code

Main proc

MOV AX, @data

MOV DS, AX

MOV AL,00H

MOV SI, offset String

MOV CX, Length

Back: MOV BH, [SI]

CMP BH, .e.

JNZ Label

INC AL

Label: INC SI

LOOP Back

MOV Ans, AL

MOV AH, 4CH

INT 21H

Main endp

End Main

**8. Write an ALP to generate square wave with period of 200µs and address of output**

**device is 55H for 8086 microprocessor.**

START: MOV AX, 01H

OUT 30H, AX

; to generate loop for 200 µs using system frequency 5MHz

MOV BX, Count ;7T

Label: DEC BX ;4T

JNZ Label ;10T/7T

MOV AX, 00H

OUT 30H, AX

MOV BX, Count

Label1: DEC BX

JNZ Label1

JMP START

**9.Addressing modes**

The different ways in which a source operand is denoted in an instruction is known as **addressing modes**. There are 8 different addressing modes in 8086 programming −

**Immediate addressing mode**

The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode.

**Example**

MOV CX, 4929 H, ADD AX, 2387 H, MOV AL, FFH

**Register addressing mode**

It means that the register is the source of an operand for an instruction.

**Example**

MOV CX, AX ; copies the contents of the 16-bit AX register into

 ; the 16-bit CX register),

ADD BX, AX

**Direct addressing mode**

The addressing mode in which the effective address of the memory location is written directly in the instruction.

**Example**

MOV AX, [1592H], MOV AL, [0300H]

**Register indirect addressing mode**

This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI.

**Example**

MOV AX, [BX] ; Suppose the register BX contains 4895H, then the contents

 ; 4895H are moved to AX

ADD CX, {BX}

**Based addressing mode**

In this addressing mode, the offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement.

**Example**

MOV DX, [BX+04], ADD CL, [BX+08]

**Indexed addressing mode**

In this addressing mode, the operands offset address is found by adding the contents of SI or DI register and 8-bit/16-bit displacements.

**Example**

MOV BX, [SI+16], ADD AL, [DI+16]

**Based-index addressing mode**

In this addressing mode, the offset address of the operand is computed by summing the base register to the contents of an Index register.

**Example**

ADD CX, [AX+SI], MOV AX, [AX+DI]

**Based indexed with displacement mode**

In this addressing mode, the operands offset is computed by adding the base register contents. An Index registers contents and 8 or 16-bit displacement.

**Example**

MOV AX, [BX+DI+08], ADD CX, [BX+SI+16]

**Intra Segment Direct Mode**
If the location to which control is to transferred is in the same segment, it is called intra segment mode. If address to which the control is to be transferred appears directly in the instruction as a displacement value, it is what called as intra segment direct mode.

For example,
JMP 2000h;
is a control transfer instruction following intra segment direct mode. Here, SHORT LABEL represents a signed displacement.

**Intra Segment Indirect Mode**
If the location to which control is to transferred is in the same segment, it is called intra segment mode. If address to which the control is to be transferred appears indirectly in the instruction, it is what called as intra segment indirect mode. For example,
JMP [AX];
is a control transfer instruction following intra segment indirect mode.

**Inter Segment Direct Mode**
If the location to which control is to transferred is not in the same segment, it is called inter segment mode. If address of segment to which the control is to be transferred and location in the segment appears directly in the instruction, it is what called as inter segment direct mode. For example,
JMP 2000H : 3000H;
is a control transfer instruction following inter segment direct mode. Here, Jump is to effective address 3000H in segment at 2000H.

**Inter Segment Indirect Mode**
If the location to which control is to transferred is not in the same segment, it is called inter segment mode. If address of segment to which the control is to be transferred and location in the segment appears indirectly in the instruction, it is what called as inter segment indirect mode. For example,
JMP [CS]:[SI]
is a control transfer instruction following inter segment indirect mode.

10. Explain ASSUME Directive

- The ASSUME directive is used to tell the assembler that the name of the logical segment should be used for a specified segment. The 8086 works directly with only 4 physical segments: a Code segment, a data segment, a stack segment, and an extra segment.

Example:

ASUME CS:CODE

This tells the assembler that the logical segment named CODE contains the instruction statements for the program and should be treated as a code segment.

ASUME DS:DATA

;This tells the assembler that for any instruction which refers to a data in the data segment, data will found in the logical segment DATA.

**11. What is REP prefix? How it functions for string instructions?**

**Ans.**  This REP prefix is used for repeating. The instruction with REP prefix will execute repeatedly till the count in the cx register will be zero. This can be used in with some of the string handling instructions.

**12. Explain the instructions (i) LDS (ii) PUSHF (iii) TEST (iv) CLD**

**Ans.**

* 1. LDS : load pointer to DS

 Move a 32 bit content from the memory given as source to 16

 bit destination register specified and to DS register.

* 1. PUSHF : push the flag

 After the execution the content of the flag register will be

 pushed to the stack.The higher byte to sp-1 and lower to

 sp-2

* 1. TEST : logical comparison

 This will compare the source and the destination specified.

 The result will be reflected only in the flag registers.

* 1. CLD : this will clear the direction flag.

 13.What is the position of the Stack Pointer after the PUSH instruction?

• - The address is 02 less than the earlier value.

1. What happens when HLT instruction is executed in processor?

• The Micro Processor enters into Halt-State and the buses are tri-stated.

1. Explain the difference between a JMP and CALL instruction?

• A JMP instruction permantely changes the program counter. • A CALL instruction leaves information on the stack so that the original program execution sequence can be resumed.

1. Explain CALL and RETURN?

 • It calls 16-bit memory address of a subroutine. It is a 3-byte instruction that transfers the program sequence to a subroutine Saves the content of the PC (Program Counter16-bit register), the address of the next instruction , on the stack. Decrements the stack pointer register by 2. Jumps unconditionally to the memory location specified by the 2nd and 3rd bytes. This instruction is accompanied by a return instruction in the subroutine. The return instruction is used either to return a function value or to terminate the execution of a function. The exit may be from anywhere within the function body, including loops or nested blocks. If the function returns a value, the return instruction is required

1. Explain about "LEA"?

• LEA(Load Effective Address) is used for initializing a register with an offset address. • A common use for LEA is to intialize an offset in BX, DI or SI for indexing an address in memory. • An equivalent operation to LEA is MOV with the OFFSET operator, which generates slightly shorter machine code

1. Difference between "Shift" and "Rotate".

• Shift and Rotate commands are used to convert a number to another form where some bits are shifted or rotated. • A rotate instruction is a closed loop instruction. That is, the data moved out at one end is put back in at the other end. • The shift instruction loses the data that is moved out of the last bit locations. • Basic difference between shift and rotate is shift command makes "fall of " bits at the end.

1. Explain about Direction Flag?

 • This is used by string manipulation instructions. • If this flag bit is 0 , the string is processed beginning from the lowest to the highest address,i.e.,.Autoincrement mode. • Otherwise,the string is processed from the highest towards the lowest address,i.e.,.Autodecrementing mode.

1. What is the difference between instructions MUL & IMUL?

 • MUL: the instruction is used for unsigned multiplication. This inst multiplies bytes or words. • IMUL (integers multiply): the inst is used for signed multiplication.

1. What is linking and relocation

 • A linker is a program used to join together several object files into one large object file. The linker produces a link file which contains the binary codes for all the combined modules. It also produces a link map which contains the address information about the link files. The linker does not assign absolute addresses but only relative address starting from zero, so the programs are relocatable & can be put anywhere in memory to be run.

1. What are the advantages of modular programming

• Modular programming can be used to break up a large program into manageable units, or to create code that can be easily re-used. Reduce problem to smaller, simpler, humanly comprehensible problems

1. Define assembly process

• A microprocessor executes a collection of machine instructions that tell the processor what to do is known as assembly process.

1. What are directives

 • Assembler directives are instructions to the assembler concerning the program being assembled.

1. What do you mean by assembler directives?

• There are some instructions in the assembly language program which r not a part of processor instruction set. These instructions are instructions to the assembler, linker and loader. They are referred as assembler directives. The assembler directives enable us to control the way in which a program assembles and lists. They acts during the assembly of program and do not generate any executable machine • code. E.g .CODE, .data

1. What are Mnemonics?

• Mnemonics are instructions or commands to perform a particular operation given by user to microprocessor. E.g. MOV, ADD, SUB, • etc.

1. What is a compiler?

• Compiler is used to translate the high-level language program into machine code.

1. What is meant by cross-compiler?

 • A program runs on one machine and executes on another is called as cross-compiler

1. What is a stack

• A stack is a last in, first out (LIFO) abstract data type and data structure. A stack can have any abstract data type as an element, but is characterized by only two fundamental operations: push and pop.

Unit 3

1. What is a programmable peripheral device?

• If the functions performed by a peripheral device can be altered or changed by a program instruction then the peripheral device is called programmable device. Usually the programmable devices will have control registers. The device can be programmed by sending control word in the prescribed format to the control register.

1. Explain the working of a handshake output port

• In handshake output operation, the processor will load a data to port. When the port receives the data, it will inform the output device to collect the data. Once the output device accepts the data, the port will inform the processor that it is empty. Now the processor can load another data to port and the above process is repeated.

1. What are the internal devices of 8255

• The internal devices of 8255 are port-A, portB and port-C. The ports can be programmed for either input or output function in different operating modes.

1. What is baud rate?

• The baud rate is the rate at which the serial data are transmitted. Baudrate is defined as l (The time for a bit cell). In some systems one bit cell has one data bit, then the baud rate and bits/sec are same.

1. What is USART?

• The device which can be programmed to perform Synchronous orAsynchronous serial communication is called USART (Universal Synchronous Asynchronous Receiver Transmitter). The INTEL 8251Ais an example of USART.

1. What are the tasks involved in keyboard interface?

• The task involved in keyboard interfacing are sensing a key actuation, Debouncing the key and Generating key codes (Decoding the key). These task are performed software if the keyboard is interfaced through ports and they are performed by hardware if the keyboard is interfaced through 8279.

1. List the components of microprocessor (single board microcomputer) based system

 The microprocessor based system consist of microprocessor as CPU, semiconductor memories like EPROM and RAM, input device, output device and interfacing devices.

1. Why interfacing is needed for 1/0 devices?

• Generally I/O devices are slow devices. Therefore the speed of I/O devices does not match with the speed of microprocessor. And so an interface is provided between system bus and I/O devices.

1. What does memory-mapping mean?

• The memory mapping is the process of interfacing memories to microprocessor and allocating addresses to each memory locations.

1. What is the drawback in memory mapped I/0?

 • When I/O devices are memory mapped, some of the addresses are allotted to I/O devices and so the full address space cannot be used for addressing memory (i.e., physical memory address space will be reduced). Hence memory mapping is useful only for small systems, where the memory requirement is less.

1. What is the need for Port?

• The I/O devices are generally slow devices and their timing characteristics do not match with processor timings. Hence the I/O devices are connected to system bus through the ports.

1. What is DMA?

• The direct data transfer between I/O device and memory is called DMA.

1. How DMA is initiated?

• When the I/O device needs a DMA transfer, it will send a DMA request signal to DMA controller. The DMA controller in turn sends a HOLD request to the processor. When the processor receives a HOLD request, it will drive its tri-stated pins to high impedance state at the end of current instruction execution and send an acknowledge signal to DMA controller. Now the DMA controller will perform DMA transfer.