ABES Engineering College, Ghaziabad

B. Tech(II Year) Even Semester Sessional Test-1

SOLUTION

Printed Pages:3 Session:2022-23

Course Code: KCS-403 Roll No.:

Course Name: Microprocessor Date of Exam:

Maximum Marks: 75 Time: 2 hrs

Instructions:

1. **Attempt All sections.**
2. **If require any missing data, then choose suitably.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Q. No.** | **Question** | **Marks** | **CO** | **KL** | **PI** |
| **Section-A Total Marks : 5\*2 =10** | | | | | |
| **1** | **Attempt ALL Parts** |  | | | |
| **(a)**  **ANS** | State the Function of ALE pin in 8085.  (Address Enable Latch) is the control signal which is nothing but a positive going pulse generated when a new operation is started by microprocessor. So when pulse goes high means ALE=1, it makes address bus enable and when ALE=0, means low pulse makes data bus enable | **2** | **CO1** | K1 | 1.3.1 |
|  |  |  |  |  |  |
| **(b)**  **ANS** | Define machine cycle. | **2** | **CO1** | K2 | 1.3.1 |
| **©**  **ANS** | Which instruction is used to multiply a 16-bit number by 2?  DAD instruction  DAD H  HL <- HL + HL | **2** | **CO4** | K2 | 1.4.1 |
| **(d)**  **ANS** | Name the instruction which uses bus idle machine cycle and explain with an example.  DAD instruction    It uses 1 Opcode Fetch and 2 Bus idle machine cycle  i.e 4+3+3=10 t-states | **2** | **CO2** | K2 | 1.4.1 |
| **(e)**  **ANS** | Explain LHLD instruction with an example. | **2** | **CO2** | K2 | 1.4.1 |
| **Section-B Total Marks: 3\*5 = 15** | | | | | |
| **2** | **Attempt ANY ONE part from the following** |  | | | |
| **(a)**  **ANS** | Show the output of Accumulator and Carry flag after the execution of the SUB A and ADD A instruction? If the contents of the A and Carry flag are A7H and 00H respectively. **(GATE 2016)**  **Accumulator will be 00H and CY will be 0** | **5** | **CO4** | K3 | 2.1.1 |
| **(b)**  **ANS** | An 8085-assembly language program is given below. Assume that the carry flag is initially unset. Identify the register contents and flag status as the following instructions are executed **(GATE 2011)**  MVI A,07H ADI F2H MOV B,A SUI C2H ADD B HLT | **5** | **CO4** | K3 | 2.1.1 |
| **3** | **Attempt ANY ONE part from the following** |  | | | |
| **(a)**  **ANS** | What happens when a CPU is interrupted in a microprocessor? Differentiate maskable and non-maskable interrupt of 8085 with example. **(GATE 1995)**  When an interrupt occurs, it causes the CPU to stop executing the current program. The control then passes to a special piece of code called an Interrupt Handler or Interrupt Service Routine. The interrupt handler will process the interrupt and resume the interrupted program.   **Difference Between Maskable and Non-Maskable Interrupt**  |  |  |  | | --- | --- | --- | | **Parameters** | **Maskable Interrupt** | **Non-Maskable Interrupt** | | Definition and Meaning | It is a hardware interrupt that the instructions of a system’s CPU can easily disable or ignore. | It is a hardware interrupt that the instructions of a system’s CPU cannot ignore or disable by any means. | | Interrupt Execution and Handling | When an interrupt of this type occurs, the system can handle it after it executes the current instructions. | When an interrupt of this type occurs, the system can store the current status and interrupts in the stack- so that the CPU gets to handle the interrupt. | | Priority of Tasks | It helps in handling the tasks that are of a lower priority. | It helps in handling the tasks that are of a higher priority- for example, a watchdog timer. | | Uses | One can make use of a Maskable Interrupt in the interfaces with the peripheral devices. | One can make use of a Non-Maskable Interrupt in case of emergencies, such as smoke detectors, power failures, and many more. | | Response Time | A Maskable Interrupt has a comparatively higher response time. | A Non-Maskable Interrupt has a very low response time as compared to the Maskable ones. | | Vectoring | This type of interrupt can be both vectored as well as non-vectored. | All of such interrupts of this type stay vectored in nature. | | Masked Operation | It can mask operations or make them pending. | It cannot mask operations or make them pending. | | Examples | A few examples of Maskable Interrupt are RST7.5, RST6.5 of 8085, etc. | A commendable example of Non-Maskable Interrupt is the Trap of 8085. | | **5** | **CO1** | K2 | 1.3.1 |
| **(b)** | Identify the Machine Cycles, T-States, size in bytes and addressing mode of the following instructions: **(GATE 2004)**  (i) LDA 3000H (ii) LXI D,F0F1H  Size : 3 bytes  Size : 3 bytes | **5** | **CO1** | K2 | 1.3.1 |
| **4** | **Attempt ANY ONE part from the following** |  | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **(a)** | State the function of RESET and READY pin in 8085.  What do you mean by wait state in microprocessor? **(GATE 2002)**  Wait State:  The WAIT state plays a significant role in preventing CPU speed incompatibilities.  - Many a times the processor is at a ready state to accept data from a device or location, but there might be no input available. This can lead to wastage of cpu time.  - So in such cases when the cpu is ready for an input but there is no such valid data then the system gets into WAIT state. In this scenario the pin 35 ( ready pin )is put into a low state.  - As soon as there is some valid data for the 8085 the system comes off the WAIT state and the low state of the READY pin is withdrawn. | **5** | **CO2** | K3 | 1.4.1 |
| **(b)** | Identify the register contents and flag status as the following instructions are executed:  A B S Z CY (**GATE-2018)**  SUB A MOV B, A DCR B INR B SUI 01H HLT | **5** | **CO2** | K3 | 1.4.1 |
| **Section-C Total Marks: 5\*10 =50** | | | | | |
| **5** | **Attempt ANY ONE part from the following** |  | | | |
| **(a)** | Sketch and Explain the Timing Diagram of MVI B, 31H. Consider that the instruction is stored on Memory Location 2000H. | **10** | **CO2** | K3 | 4.2.1 |
| **(b)** | Sketch and Explain the Timing Diagram of IN, 80H. Consider that the instruction is stored on Memory Location 2000H.  A picture containing text, diagram, line, plan  Description automatically generated | **10** | **CO2** | K3 | 4.2.1 |
| **6** | **Attempt ANY ONE part from the following** |  | | | |
| **(a)** | Sketch the internal architecture of 8085. Explain every block in  Detail.  Architecture of 8085 microprocessor  **Introduction :**  The architecture of the 8085 microprocessor consists of several key components, including the accumulator, registers, program counter, stack pointer, instruction register, flags register, data bus, address bus, and control bus.  The accumulator is an 8-bit register that is used to store arithmetic and logical results. It is the most commonly used register in the 8085 microprocessor and is used to perform arithmetic and logical operations such as addition, subtraction, and bitwise operations.  1.  The 8085 microprocessor has six general-purpose registers, including B, C, D, E, H, and L, which can be combined to form 16-bit register pairs. The B and C registers can be combined to form the BC register pair, the D and E registers can be combined to form the DE register pair, and the H and L registers can be combined to form the HL register pair. These register pairs are commonly used to store memory addresses and other data.  The program counter is a 16-bit register that contains the memory address of the next instruction to be executed. The program counter is incremented after each instruction is executed, which allows the microprocessor to execute instructions in sequence.  The stack pointer is a 16-bit register that is used to manage the stack. The stack is a section of memory that is used to store data temporarily, such as subroutine addresses and other data. The stack pointer is used to keep track of the top of the stack.  The instruction register is an 8-bit register that contains the current instruction being executed. The instruction register is used by the microprocessor to decode and execute instructions.  2.  The flags register is an 8-bit register that contains status flags that indicate the result of an arithmetic or logical operation. These flags include the carry flag, zero flag, sign flag, and parity flag. The carry flag is set when an arithmetic operation generates a carry, the zero flag is set when the result of an arithmetic or logical operation is zero, the sign flag is set when the result of an arithmetic or logical operation is negative, and the parity flag is set when the result of an arithmetic or logical operation has an even number of 1 bits.  3.  The data bus is an 8-bit bus that is used to transfer data between the microprocessor and memory or other devices. The data bus is bidirectional, which means that it can be used to read data from memory or write data to memory.  The address bus is a 16-bit bus that is used to address memory and other devices. The address bus is used to select the memory location or device that the microprocessor wants to access.  4.  The control bus is a set of signals that controls the operations of the microprocessor, including the read and write operations. The control bus includes signals such as the read signal, write signal, interrupt signal, and reset signal. The read signal is used to read data from memory or other devices, the write signal is used to write data to memory or other devices, the interrupt signal is used to signal the microprocessor that an interrupt has occurred, and the reset signal is used to reset the microprocessor to its initial state.  8085 is an 8-bit, general-purpose microprocessor. It consists of the following functional units: Arithmetic and Logic Unit (ALU) : It is used to perform mathematical operations like addition, multiplication, subtraction, division, decrement, increment, etc. Different operations are carried out in ALU: ***Logical operations, Bit-Shifting Operations, and Arithmetic Operations.*** Flag Register: It is an 8-bit register that stores either 0 or 1 depending upon which value is stored in the accumulator. Flag Register contains 8-bit out of which 5-bits are important and the rest of 3-bits are “don’t Care conditions”. The flag register is a dynamic register because after each operation to check whether the result is zero, positive or negative, whether there is any overflow occurred or not, or for comparison of two 8-bit numbers carry flag is checked. So for numerous operations to check the contents of the accumulator and from that contents if we want to check the behavior of given result then we can use Flag register to verify and check. So we can say that the **flag register is a status register and it is used to check the status of the current operation which is being carried out by ALU.**  *Different Fields of Flag Register:*   1. **Carry Flag** 2. **Parity Flag** 3. **Auxiliary Carry Flag** 4. **Zero Flag** 5. **Sign Flag**  Accumulator: Accumulator is used to perform I/O, arithmetic, and logical operations. It is connected to ALU and the internal data bus. The accumulator is the heart of the microprocessor because for all arithmetic operations Accumulator’s 8-bit pin will always there connected with ALU and in most-off times all the operations carried by different instructions will be stored in the accumulator after operation performance. General Purpose Registers: There are six general-purpose registers. These registers can hold 8-bit values. These 8-bit registers are B,C,D,E,H,L. These registers work as 16-bit registers when they work in pairs like B-C, D-E, and H-L. Here registers W and Z are reserved registers. We can’t use these registers in arithmetic operations. It is reserved for microprocessors for internal operations like swapping two 16-bit numbers. We know that to swap two numbers we need a third variable hence here W-Z register pair works as temporary registers and we can swap two 16-bit numbers using this pair. Program Counter : Program Counter holds the address value of the memory to the next instruction that is to be executed. It is a 16-bit register.  ***For Example:*** *Suppose current value of Program Counter : [PC] = 4000H*  *(It means that next executing instruction is at location 4000H.After fetching,program Counter(PC) always increments*  *by +1 for fetching of next instruction.)*   Stack Pointer : It works like a stack. In stack, the content of the register is stored that is later used in the program. It is a 16-bit special register. The stack pointer is part of memory but it is part of Stack operations, unlike random memory access. Stack pointer works in a continuous and contiguous part of the memory. whereas Program Counter(PC) works in random memory locations. This pointer is very useful in stack-related operations like ***PUSH, POP, and nested CALL requests*** initiated by Microprocessor. *It reserves the address of the most recent stack entry.* Temporary Register: It is an 8-bit register that holds data values during arithmetic and logical operations. Instruction register and decoder: It is an 8-bit register that holds the instruction code that is being decoded. The instruction is fetched from the memory. Timing and control unit: The timing and control unit comes under the CPU section, and it controls the flow of data from the CPU to other devices. It is also used to control the operations performed by the microprocessor and the devices connected to it. There are certain timing and control signals like Control signals, DMA Signals, RESET signals and Status signals. Interrupt control: Whenever a microprocessor is executing the main program and if suddenly an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program. There are 5 interrupt signals in 8085 microprocessors: INTR, TRAP, RST 7.5, RST 6.5, and RST 5.5.  **Priorities of Interrupts:** TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR Address bus and data bus: The data bus is bidirectional and carries the data which is to be stored. The address bus is unidirectional and carries the location where data is to be stored.  In the 8085 microprocessor, the address bus and data bus are two separate buses that are used for communication between the microprocessor and external devices.  The Address bus is used to transfer the memory address of the data that needs to be read or written. The address bus is a 16-bit bus, allowing the 8085 to access up to 65,536 memory locations.  The Data bus is used to transfer data between the microprocessor and external devices such as memory and I/O devices. The data bus is an 8-bit bus, allowing the 8085 to transfer 8-bit data at a time. The data bus can also be used for instruction fetch operations, where the microprocessor fetches the instruction code from memory and decodes it.  The combination of the address bus and data bus allows the 8085 to communicate with and control external devices, allowing it to execute its program and perform various operations. Serial Input/output control: It controls the serial data communication by using Serial input data and Serial output data.  Serial Input/Output control in the 8085 microprocessor refers to the communication of data between the microprocessor and external devices in a serial manner, i.e., one bit at a time. The 8085 has a serial I/O port (SID/SOD) for serial communication. The SID pin is used for serial input and the SOD pin is used for serial output. The timing and control of serial communication is managed by the 8085’s internal circuitry. The 8085 also has two special purpose registers, the Serial Control Register (SC) and the Serial Shift Register (SS), which are used to control and monitor the serial communication. The flow of an Instruction Cycle in 8085 Architecture :  1. Execution starts with Program Counter. It starts program execution with the next address field. it fetches an instruction from the memory location pointed by Program Counter. 2. For address fetching from the memory, multiplexed address/data bus acts as an address bus and after fetching instruction this address bus will now acts as a data bus and extract data from the specified memory location and send this data on an 8-bit internal bus. For multiplexed address/data bus Address Latch Enable(ALE) Pin is used. If ***ALE = 1 (Multiplexed bus is Address Bus otherwise it acts as Data Bus).*** 3. After data fetching data will go into the Instruction Register it will store data fetched from memory and now data is ready for decoding so for this Instruction decoder register is used. 4. After that timing and control signal circuit comes into the picture. *It sends control signals all over the microprocessor to tell the microprocessor whether the given instruction is for READ/WRITE and whether it is for MEMORY/I-O Device activity.* 5. Hence according to timing and control signal pins, logical and arithmetic operations are performed and according to that data fetching from the different registers is done by a microprocessor, and mathematical operation is carried out by ALU. And according to operations Flag register changes dynamically. 6. With the help of Serial I/O data pin(SID or SOD Pins) we can send or receive input/output to external devices .in this way execution cycle is carried out. 7. ***While execution is going on if there is any interrupt detected then it will stop execution of the current process and Invoke Interrupt Service Routine (ISR)*** Function. Which will stop the current execution and do execution of the current occurred interrupt after that normal execution will be performed. | **4+6** | **CO2** | K3 | 4.2.1 |
| **(b)** | Sketch a functional diagram of the 8085 microprocessor and explain its various Pins.  Pin diagram of 8085 microprocessor is as given below:      **1. Address Bus and Data Bus:**  The address bus is a group of sixteen lines i.e A0-A15. The address bus is unidirectional, i.e., bits flow in one direction from the microprocessor unit to the peripheral devices and uses the high order address bus.  **2. Control and Status Signals:**   * **ALE –** It is an Address Latch Enable signal. It goes high during first T state of a machine cycle and enables the lower 8-bits of the address, if its value is 1 otherwise data bus is activated. * **IO/M’ –** It is a status signal which determines whether the address is for input-output or memory. When it is high(1) the address on the address bus is for input-output devices. When it is low(0) the address on the address bus is for the memory. * **SO, S1 –** These are status signals. They distinguish the various types of operations such as halt, reading, instruction fetching or writing.  |  |  |  |  | | --- | --- | --- | --- | | **IO/M’** | **S1** | **S0** | **Data Bus Status** | | 0 | 1 | 1 | Opcode fetch | | 0 | 1 | 0 | Memory read | | 0 | 0 | 1 | Memory write | | 1 | 1 | 0 | I/O read | | 1 | 0 | 1 | I/O write | | 1 | 1 | 1 | Interrupt acknowledge | | 0 | 0 | 0 | Halt |  * **RD’ –** It is a signal to control READ operation. When it is low the selected memory or input-output device is read. * **WR’ –** It is a signal to control WRITE operation. When it goes low the data on the data bus is written into the selected memory or I/O location. * **READY –** It senses whether a peripheral is ready to transfer data or not. If READY is high(1) the peripheral is ready. If it is low(0) the microprocessor waits till it goes high. It is useful for interfacing low speed devices.   **3. Power Supply and Clock Frequency:**     * **Vcc –** +5v power supply * **Vss –** Ground Reference * **XI, X2 –** A crystal is connected at these two pins. The frequency is internally divided by two, therefore, to operate a system at 3MHZ the crystal should have frequency of 6MHZ. * **CLK (OUT) –** This signal can be used as the system clock for other devices. * **4. Interrupts and Peripheral Initiated Signals:**  The 8085 has five interrupt signals that can be used to interrupt a program execution. * (i) INTR  (ii) RST 7.5  (iii) RST 6.5  (iv) RST 5.5  (v) TRAP * The microprocessor acknowledges Interrupt Request by INTA’ signal. In addition to Interrupts, there are three externally initiated signals namely RESET, HOLD and READY. To respond to HOLD request, it has one signal called HLDA. * **INTR –** It is an interrupt request signal. * **INTA’ –** It is an interrupt acknowledgement sent by the microprocessor after INTR is received.   **5. Reset Signals:**     * **RESET IN’ –** When the signal on this pin is low(0), the program-counter is set to zero, the buses are tristated and the microprocessor unit is reset. * **RESET OUT –** This signal indicates that the MPU is being reset. The signal can be used to reset other devices.   **6. DMA Signals:**     * **HOLD –** It indicates that another device is requesting the use of the address and data bus. Having received HOLD request the microprocessor relinquishes the use of the buses as soon as the current machine cycle is completed. Internal processing may continue. After the removal of the HOLD signal the processor regains the bus. * **HLDA –** It is a signal which indicates that the hold request has been received after the removal of a HOLD request, the HLDA goes low.   **7. Serial I/O Ports:**  Serial transmission in 8085 is implemented by the two signals,     * **SID and SOD –** SID is a data line for serial input where as SOD is a data line for serial output. | **4+6** | **CO2** | K3 | 4.2.1 |
| **7** | **Attempt ANY ONE part from the following** |  | | | |
| **(a)** | Write an assembly-level program for the addition and subtraction to unsigned BCD numbers.    Subtraction  A piece of paper with writing on it  Description automatically generated with low confidence | **5+5** | **CO4** | K3 | 1.4.1 |
| **(b)** | 1. Write an assembly program using ADI instruction to add 2 hexadecimal number 3AH and 48H and display the result at an output port 80.      1. Write an assembly language program in 8085 to add two 16-bit   numbers.  (ii) | **5+5** | **CO4** | K3 | 1.4.1 |
| **8** | **Attempt ANY ONE part from the following** |  | | | |
| **(a)** | Define Addressing Modes. Demonstrate types of Addressing Modes available in 8085 microprocessors with the help of an example.  The way of specifying data to be operated by an instruction is called addressing mode.  **Why use addressing modes in 8085 microprocessor ?**   * Flexibility: Addressing modes provide a flexible way to access data and instructions in memory. Different addressing modes allow programmers to choose the most appropriate addressing technique for a particular task, depending on the type of data being accessed, the size of the data, and other factors. * Memory optimization: Addressing modes can help to optimize the use of memory resources by allowing data and instructions to be accessed in the most efficient way possible. For example, using indirect addressing modes can reduce the amount of memory needed to store addresses, while using indexed addressing modes can reduce the number of memory accesses needed to access a large array of data. * Performance optimization: Addressing modes can also help to optimize the performance of the microprocessor by reducing the number of memory accesses needed to fetch data or instructions. This can help to speed up the execution of programs and improve the overall efficiency of the microprocessor. * Reduced code size: Addressing modes can help to reduce the size of code needed to perform a particular task. By using addressing modes that allow data and instructions to be accessed using fewer instructions, programmers can write more compact and efficient code.   **Types of addressing modes –**  In 8085 microprocessor there are 5 types of addressing modes:   1. **Immediate Addressing Mode –**  In immediate addressing mode the source operand is always data. If the data is 8-bit, then the instruction will be of 2 bytes, if the data is of 16-bit then the instruction will be of 3 bytes. 2. **Examples:**  MVI B 45 (move the data 45H immediately to register B)  LXI H 3050 (load the H-L pair with the operand 3050H immediately)  JMP address (jump to the operand address immediately) 4. **Register Addressing Mode –**  In register addressing mode, the data to be operated is available inside the register(s) and register(s) is(are) operands. Therefore the operation is performed within various registers of the microprocessor. 5. **Examples:**  MOV A, B (move the contents of register B to register A)  ADD B (add contents of registers A and B and store the result in register A)  INR A (increment the contents of register A by one) 7. **Direct Addressing Mode –**  In direct addressing mode, the data to be operated is available inside a memory location and that memory location is directly specified as an operand. The operand is directly available in the instruction itself. 8. **Examples:**  LDA 2050 (load the contents of memory location into accumulator A)  LHLD address (load contents of 16-bit memory location into H-L register pair)  IN 35 (read the data from port whose address is 35) 10. **Register Indirect Addressing Mode –**  In register indirect addressing mode, the data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair. 11. **Examples:**  MOV A, M (move the contents of the memory location pointed by the H-L pair to the accumulator)  LDAX B (move contents of B-C register to the accumulator)  STAX B (store accumulator contents in memory pointed by register pair B-C) 13. **Implied/Implicit Addressing Mode –**  In implied/implicit addressing mode the operand is hidden and the data to be operated is available in the instruction itself. 14. **Examples:**  CMA (finds and stores the 1’s complement of the contents of accumulator A in A)  RRC (rotate accumulator A right by one bit)  RLC (rotate accumulator A left by one bit) | **2+8** | **CO1** | K2 | 1.4.1  . |
| **(b)** | Explain the concept of interrupts. Explain different types of interrupts in 8085 Microprocessors.  In the 8085 microprocessor, an interrupt is a signal that temporarily suspends the normal execution of a program and redirects the control to a specific interrupt service routine (ISR). Interrupts allow the microprocessor to respond to external events, such as user input, system events, or hardware signals, without the need for constant polling.  **There are five interrupt signals in the 8085 microprocessor:**   1. **TRAP:** The TRAP interrupt is a non-maskable interrupt that is generated by an external device, such as a power failure or a hardware malfunction. The TRAP interrupt has the highest priority and cannot be disabled. 2. **RST 7.5:** The RST 7.5 interrupt is a maskable interrupt that is generated by a software instruction. It has the second highest priority. 3. **RST 6.5:** The RST 6.5 interrupt is a maskable interrupt that is generated by a software instruction. It has the third highest priority. 4. **RST 5.5:** The RST 5.5 interrupt is a maskable interrupt that is generated by a software instruction. It has the fourth highest priority. 5. **INTR:** The INTR interrupt is a maskable interrupt that is generated by an external device, such as a keyboard or a mouse. It has the lowest priority and can be disabled.   When microprocessor receives any interrupt signal from peripheral(s) which are requesting its services, it stops its current execution and program control is transferred to a sub-routine by generating **CALL** signal and after executing sub-routine by generating **RET** signal again program control is transferred to main program from where it had stopped. When microprocessor receives interrupt signals, it sends an acknowledgement (INTA) to the peripheral which is requesting for its service. Interrupts can be classified into various categories based on different parameters:   1. **Hardware and Software Interrupts –** When microprocessors receive interrupt signals through pins (hardware) of microprocessor, they are known as *Hardware Interrupts*. There are 5 Hardware Interrupts in 8085 microprocessor. They are – *INTR, RST 7.5, RST 6.5, RST 5.5, TRAP* *Software Interrupts* are those which are inserted in between the program which means these are mnemonics of microprocessor. There are 8 software interrupts in 8085 microprocessor. They are – *RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6, RST 7*. 2. **Vectored and Non-Vectored Interrupts –** *Vectored Interrupts* are those which have fixed vector address (starting address of sub-routine) and after executing these, program control is transferred to that address. Vector Addresses are calculated by the formula 8 \* TYPE  |  |  | | --- | --- | | **INTERRUPT** | **VECTOR ADDRESS** | | TRAP (RST 4.5) | 24 H | | RST 5.5 | 2C H | | RST 6.5 | 34 H | | RST 7.5 | 3C H |  1. For Software interrupts vector addresses are given by:  |  |  | | --- | --- | | **INTERRUPT** | **VECTOR ADDRESS** | | RST 0 | 00 H | | RST 1 | 08 H | | RST 2 | 10 H | | RST 3 | 18 H | | RST 4 | 20 H | | RST 5 | 28 H | | RST 6 | 30 H | | RST 7 | 38 H |  1. *Non-Vectored Interrupts* are those in which vector address is not predefined. The interrupting device gives the address of sub-routine for these interrupts. *INTR* is the only non-vectored interrupt in 8085 microprocessor. 2. **Maskable and Non-Maskable Interrupts –** *Maskable Interrupts* are those which can be disabled or ignored by the microprocessor. These interrupts are either edge-triggered or level-triggered, so they can be disabled. *INTR, RST 7.5, RST 6.5, RST 5.5* are maskable interrupts in 8085 microprocessor. Non-Maskable Interrupts are those which cannot be disabled or ignored by microprocessor. *TRAP* is a non-maskable interrupt. It consists of both level as well as edge triggering and is used in critical power failure conditions. 3. **Priority of Interrupts –** When microprocessor receives multiple interrupt requests simultaneously, it will execute the interrupt service request (ISR) according to the priority of the interrupts. | **2+8** | **CO1** | K2 | 1.4.1 |
| **9** | **Attempt ANY ONE part from the following** |  | | | |
| **(a)**  **ans** | (i) Differentiate between assembly language and machine language. (ii)Explain the bus organization of 8085 microprocessor.  (i)    (II) BUS ORGANIZATION OF 8085  DATA BUS: 8085 Microprocessor has 8-bit data bus. So, it can be used to carry the 8 bit data starting from 00000000H (00H) to 11111111H (FFH). Here 'H' tells the Hexadecimal Number. It is bidirectional. These lines are used for data flowing in both direction means data can be transferred and can be received through these lines. The data bus also connects the I/O ports and CPU. It has 8 parallel lines of data bus. So, it can access up to 28 = 256 data bus lines.  ADDRESS BUS: The bus over which the CPU sends out the address of the memory location is known as Address bus. The address bus carries the address of memory location to be written or to be read from. The address bus is unidirectional. It means bits flowing occur only in one direction, only from microprocessor to peripheral devices. We can find that how much memory location it can use the formula 2N. Where N is the number of bits used for address lines. 8085 has 16 address lines. So, it can access up to 216 = 64KB memory locations (0000H-FFFFH).  CONTROL BUS: The control bus is used for sending control signals to the memory and I/O devices. The CPU sends control signal on the control bus to enable the outputs of addressed memory devices or I/O port devices. Some of the control bus signals are as follows: Memory read, Memory writes, I/O read, I/O write.  https://scanftree.com/microprocessor/8085-bus-structure.JPG  Fig.5 8085 Bus System | **5+5** | **CO1** | K2 | 1.4.1 |
| **(b)**  **ans** | * Explain why AD0-AD7 lines are multiplexed and A8 to A15 are not multiplexed. Show the process of demultiplexing. * Explain the following:   (a)HOLD (b) SID (c)SOD (d)TRAP  (e) IO/M  (i)The data bus and the low order address bus on the 8085 microprocessor are multiplexed with each other. This allows 8 pins to be used where 16 would normally be required. The hardware interface is required to demultiplex the bus by latching the low order address in the first T cycle, on the falling edge of ALE.  The address bus is multiplexed in 8085. The multiplexing is done with the help of ALE signal. ALE stands for address latch enable. When ALE is High (Logic 1) : Upper address lines (line 15-8) and Lower address lines (line 7-0) combinely holds the 16 bits of the address. When ALE is Low (Logic 0) : Upper address lines (line 15-8) holds the upper 8 bit address & Lower address lines (line 7-0) holds the "8 bit DATA"  Multiplexing is used to reduce the number of pins of 8085, which otherwise would have been a 48 pin chip. But because of multiplexing, external hardware is required to demultiplex the lower byte address cum data bus.  The Pin no= 30 of 8085 is the ALE pin which stands for ‘Address Latch Enable’. ALE signal is used to demultiplex the lower order address bus (AD0 – AD7).  Pins 12 to 19 of 8085 are AD0 – AD7 which is the multiplexed address-data bus. Multiplexing is done to reduce the number of pins of 8085.  Lower byte of address (A0 – A7) are available from AD0 – AD7 (pins 12 to 19) during T1 of machine cycle. But the lower byte of address (A0 – A7), along with the upper byte A8 – A15 (pins 21 to 28) must be available during T2 and rest of the machine cycle to access memory location or I/O ports.  Now ALE signal goes high at the beginning of T1 of each machine cycle and goes low at the end of T1 and remains low during the rest of the machine cycle. This high to low transition of ALE signal at the end of T1 is used to latch the lower order address byte (A0 – A7) by the latch IC 74LS373, so that the lower byte A0 – A7 is continued to be available till the end of the machine cycle. The situation is explained in the following  figure:    Fig: Lower byte of address latching achieved by the H to L transition of ALE signal, which occurs at the end of T1 of each machine cycle  Demultiplexing of AD0-AD7 using IC 74LS373  (ii)   * **HOLD** – This signal is generated at pin number 39. This pin generates a signal to notify the processor that more than one request is present to access the data and address bus.   When this signal gets enabled, the CPU frees the bus after completion of the recent operation. Once the hold signal gets disabled, the processor can access the bus again.   * **SID** – SID denotes serial input data pin and its pin is numbered as 5. With this pin, data is serially fed to the processor directly through the input devices. * **SOD** – SOD denotes serial output data pin and its pin number is 4, in the pin configuration of 8085. Once the data is processed in the microprocessor then this pin represents bit by bit results at the output devices. * **IO/M** – It is pin number 34 and indicates the selection of a memory address or input-output device. This shows whether the read/write operation is to be carried out at the memory location or at the I/O device.   The low signal at this pin shows that operation is performing over memory location. As against, a high signal at this pin represents the operation at I/O device.   * **TRAP:** The TRAP interrupt is a non-maskable interrupt that is generated by an external device, such as a power failure or a hardware malfunction. The TRAP interrupt has the highest priority and cannot be disabled. | **5+5** | **CO1** | K2 | 1.4.1 |

CO Course Outcomes mapped with respective question KL Bloom's knowledge Level (K1, K2, K3, K4, K5, K6)

K1- Remember, K2- Understand, K3-Apply, K4- Analyze, K5: Evaluate, K6- Create