

Section A

1. It is a programmable device that takes in input perform some arithmetic and logical operations over it and produce desired output. In simple words, a Microprocessor is a digital device on a chip which can fetch instruction from memory, decode and execute them and give results.

2.

Address bus - Address bus is a unidirectional bus which carries address only.

Data bus – Data bus is a bidirectional bus which carries only Data only.

Control bus - Control bus is used to generate timing and control signals to control all the associated peripherals.

3.

1. Sign Flag
2. Zero Flag
3. Auxiliary Carry Flag
4. Parity Flag
5. Carry Flag

4.

1. MVI A, 00H
2. ANI 00H
3. XRA A
4. SUB A

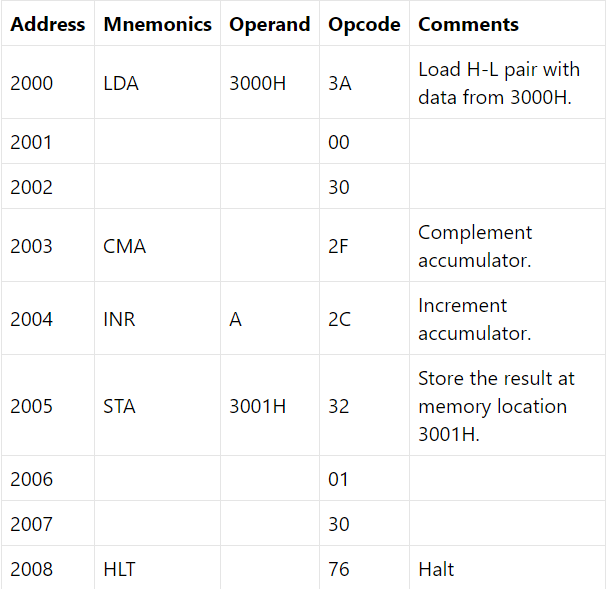
5.



6.

|  |  |
| --- | --- |
| CMP | SUB |
| Compare the numerical value of the destination with the source and set flags appropriately. | Subtracts the source value from the destination. |
| Does not modify the destination field | Modifies the destination field. |

7. 2’s Complement of an bit number:



8. The machine cycle (or **instruction cycle)** is the main activity performed of the CPU to execute the program instruction. The CPU continuously performs the machine cycle necessary for program execution. It consists of 4 steps: Fetch, Decode, Execute and Store which are continuously and repetitively performed sequentially in the cycle by the processor.

9. Counter: A counter is designed simply by loading appropriate number into one of the registers and using INR or DNR instructions. Loop is established to update the count. Each count is checked to determine whether it has reached final number if not, the loop is repeated.

Time Delay: Procedure used to design a specific delay. A register is loaded with a number, depending on the time delay required and then the register is decremented until it reaches zero by setting up a loop with conditional jump instruction.

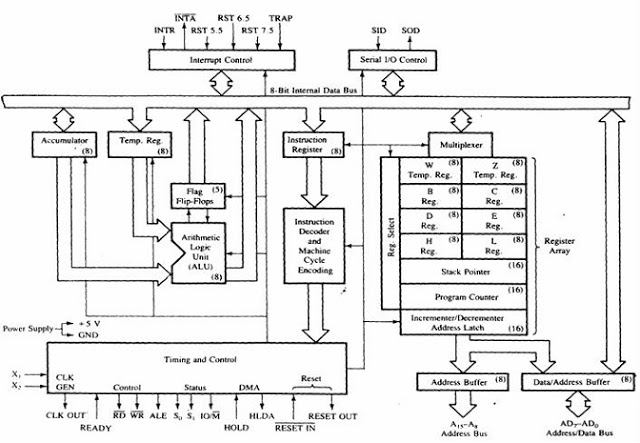
10. Interrupt: mechanism by which modules like I/O or memory may interrupt the normal processing by CPU. It may be either clicking a mouse, dragging a cursor, printing a document etc. the case where interrupt is getting generated.

11. Didn’t get specific answer

12. Didn’t get asnwer

Section – B

13.



The following are the functional blocks in the 8085 Microprocessor.  
1. Accumulator – 8bit register which hold one of the operands during arithmetic/logic operations  
2. Temporary register – 8bit register which holds data temporarily during arithmetic/logical operations  
3. Arithmetic and Logic Unit (ALU) – includes Accumulator, Temporary register, arithmetic and logic circuits and flag register. The ALU can perform arithmetic and logical operations  
4. Flag register – 8bit register which holds flags that affected by operations of ALU  
5. Instruction Register – 8bit register which stores instructions fetched from the memory.  
6. Instruction Decoder and Machine cycle encoder - This unit decodes the instruction stored in the Instruction register. It determines the nature of the instruction and establishes the sequence of events to be followed by the Timing and control unit.  
7. General purpose registers - There are six 8-bit general purpose registers namely B, C, D, E, H and L registers. B and C registers are combined together as BC register pair for 16-bit operations. Similarly, D and E registers can be used as DE resister pair and H and L as HL register pair.  
8. Stack Pointer - Stack is a portion of memory (RAM) used as FILO (First In Last Out) buffer. This is mainly used during subroutine operations. Stack Pointer is a 16-bit register used as a memory pointer (16-bit address) for denoting the stack position in memory.  
9. Program Counter - The Program Counter (PC) is a 16-bit register. It is used to point the address of the next instruction to be fetched from the memory. When one instruction is fetched from memory, PC is automatically incremented to point out the next instruction.  
10. Incrementer / Decrementer - This unit is used to increment or decrement the contents of the 16-bit registers.  
11. Timing and Control unit - The internal clock generator is available in this unit. This unit has the micro programs for all the instructions to carry out the micro steps required in completing the instructions.  
12. Interrupt control - interrupts are handled by the Interrupt control unit. INT A signal is generated by the Interrupt control unit as an acknowledgement for an interrupting device. If two or more interrupts occur at the same time, service is given according to the priority basis.  
13. Serial I/O control - Serial data is transmitted to the peripherals through SOD pin and received through the SID pin. The SOD and SID pins are handled by the Serial I/O control unit using the SIM and RIM instructions.  
14. Address buffer and Address / Data buffer - The Address buffer is an 8-bit unidirectional buffer from which the higher order address bits A8 – A15 leaves the microprocessor to the memory and peripherals. The Address / Data buffer is an 8-bit bidirectional buffer used for sending the lower order address bits A0 – A7 and sending and receiving the data bits D0 – D7 to the memory and peripherals.

14 A. Addressing Modes– The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

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. 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.
3. 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.
4. 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.
5. 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. B The 8085-instruction set is classified into 3 categories by considering the length of the instructions. In 8085, the length is measured in terms of “byte” rather than “word” because 8085 microprocessor has 8-bit data bus. Three types of instruction are: 1-byte instruction, 2-byte instruction, and 3-byte instruction.

* One-byte instructions –  
  In 1-byte instruction, the opcode and the operand of an instruction are represented in one byte.
* Two-byte instructions –  
  Two-byte instruction is the type of instruction in which the first 8 bits indicates the opcode and the next 8 bits indicates the operand.
* Three-byte instructions –  
  Three-byte instruction is the type of instruction in which the first 8 bits indicates the opcode and the next two bytes specify the 16-bit address. The low-order address is represented in second byte and the high-order address is represented in the third byte.

15. **Write a program for subtraction of two 16-bit HEX numbers.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Memory Address** | **Label** | **Mnemonics** | **Operand** |
| 2000 |  | MVI | C,00H |
| 2002 |  | LHLD | 2500H |
| 2005 |  | XCHG |  |
| 2006 |  | LHLD | 2502H |
| 2009 |  | MOV | A, E |
| 200A |  | SUB | L |
| 200B |  | JNC | LOOP1 |
| 200E |  | DCR | D |
| 200F | LOOP1 | STA | 2504H |
| 2012 |  | MOV | A, D |
| 2013 |  | SUB | H |
| 2014 |  | JNC | LOOP2 |
| 2017 |  | INR | C |
| 2018 | LOOP2 | STA | 2505H |
| 201B |  | MOV | A, C |
| 201C |  | STA | 2506H |
| 201F |  | HLT |  |

15 B

DAA -

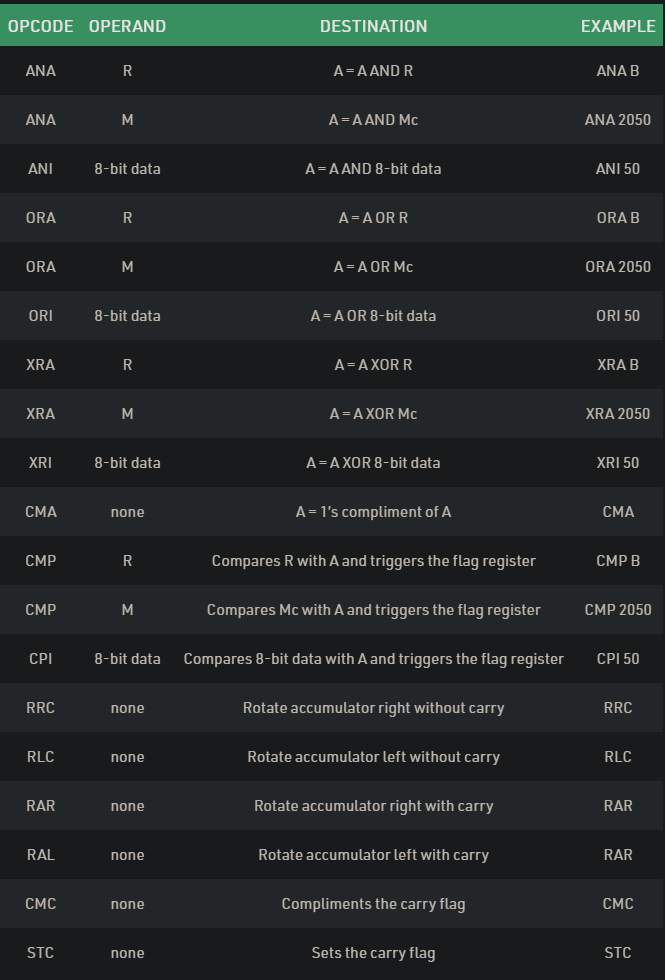
The DAA (Decimal Adjust after Addition) instruction allows addition of numbers represented in 8-bit packed BCD code. It is used immediately after normal addition instruction operating on BCD codes. This instruction assumes the AL register as the source and the destination, and hence it requires no operand.

DAD rp –

DAD is a mnemonic, which stands for Double ADd and also rp stands for any one of the following register pairs as mentioned below.

rp = BC, DE, or HL

As rp can have any of the three values, there are three opcodes for this type of instruction. It occupies only 1-Byte in memory.

16 A. Logical instructions are the instructions which perform basic logical operations such as AND, OR, etc. In 8085 microprocessors, the destination operand is always the accumulator. Here logical operation works on a bitwise level. 

16 B idk bois

17 A Mainly the following three basic operations are performed in the stack:

* Push: Adds an item in the stack. If the stack is full, then it is said to be an Overflow condition.
* Pop: Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.
* Peek or Top: Returns top element of stack.
* isEmpty: Returns true if stack is empty, else false.

17 B.

**conditional Call instruction –**  
CALL address is the format for call instruction. After execution of this

In these instructions program control is transferred to subroutine and value of PC(Program Counter) is pushed into SP(Stack Pointer) only if condition is satisfied.

| INSTRUCTION | PARAMETER | COMMENT |
| --- | --- | --- |
| CC | 16-bit address | Call at address if cy (carry flag) = 1 |
| CNC | 16-bit address | Call at address if cy (carry flag) = 0 |
| CZ | 16-bit address | Call at address if ZF (zero flag) = 1 |

**conditional Return instruction –**  
RET is the instruction used to mark the end of sub-routine. It has no parameter. After execution By these instructions program control is transferred back to main program and value of PC(Program Counter) is popped from SP(Stack Pointer).

only if condition is satisfied. There is no parameter for return instruction.

| INSTRUCTION | COMMENT |
| --- | --- |
| RC | Return from subroutine if cy (carry flag) = 1 |
| RNC | Return from subroutine if cy (carry flag) = 0 |
| RZ | Return from subroutine if ZF (zero flag) = 1 |

18 A

|  |  |
| --- | --- |
| ***Memory Mapped I/O*** | ***Peripheral MappedI/O*** |
| 16-bit device address | 8-bit device address |
| Data transfer between any general-purpose register and I/O port. | Data is transfer only between accumulator and I.O port |
| The memory map (64K) is shared between I/O device and system memory. | The I/O map is independent of the memory map; 256 input device and 256. output device can be connected |
| More hardware is required to decode 16-bit address | Less hardware is required to decode 8-bit address |
| Arithmetic or logic operation can be directly performed with I/O data | Arithmetic or logical operation cannot be directly performed with I/O data |

18 B.

|  |  |  |
| --- | --- | --- |
| ADDRESS | LABEL | MNEMONIC |
| 2000H |  | LDA 201FH |
| 2001H |  |  |
| 2002H |  |  |
| 2003H |  | MOV B, A |
| 2004H |  | ANI 0FH |
| 2005H |  |  |
| 2006H |  | MOV C, A |
| 2007H |  | MOV A, B |
| 2008H |  | ANI F0H |
| 2009H |  |  |
| 200AH |  | JZ SKIPMULTIPLY |
| 200BH |  |  |
| 200CH |  |  |
| 200DH |  | RRC |
| 200EH |  | RRC |
| 200FH |  | RRC |
| 2010H |  | RRC |
| 2011H |  | MOV D, A |
| 2012H |  | XRA A |
| 2013H |  | MVI E, 0AH |
| 2014H |  |  |
| 2015H | SUM | ADD D |
| 2016H |  | DCR E |
| 2017H |  | JNZ SUM |
| 2018H |  |  |
| 2019H |  |  |
| 201AH | SKIPMULTIPLY | ADD C |
| 201BH |  | STA 2020H |
| 201CH |  |  |
| 201DH |  |  |
| 201EH |  | HLT |

19. A **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 |

19. B. **SIM and RIM instructions in 8085 microprocessor:**

| Sr. No. | Sim Instruction | Rim Instruction |
| --- | --- | --- |
| 1 | SIM stands for Set Interrupt Mask. | RIM stands for Read Interrupt Mask. |
| 2 | It is responsible for masking/unmasking of RST 7.5, RST 6.5 and RST 5.5. | It checks whether RST 7.5, RST 6.5, RST 5.5 are masked or not. |
| 3 | It resets to 0 RST 7.5 flip flop. | It checks whether interrupts are enabled or not and to check whether RST 7.5, RST 6.5 or RST 5.5 interrupts are pending or not. |
| 4 | The content of the Accumulator decides the action to be taken. So before executing the SIM instruction, it is mandatory to initialize Accumulator with the required value. | The contents of the Accumulator after the execution of the RIM instruction provide this information.Thus, it is essential to look into the Accumulator contents after the RIM instruction is executed. |
| 5 | SIM instruction can be used for serial output of data. | RIM instruction can be used for serial input of data. |
| 6 | Its opcode(in Hex) is 30. | Its opcode(in Hex) is 20. |

20 A. Direct Access Media (DMA) :

DMA Controller is a hardware device that allows I/O devices to directly access memory with less participation of the processor. DMA controller needs the same old circuits of an interface to communicate with the CPU and Input/Output devices.

The DMA controller has three registers as follows.

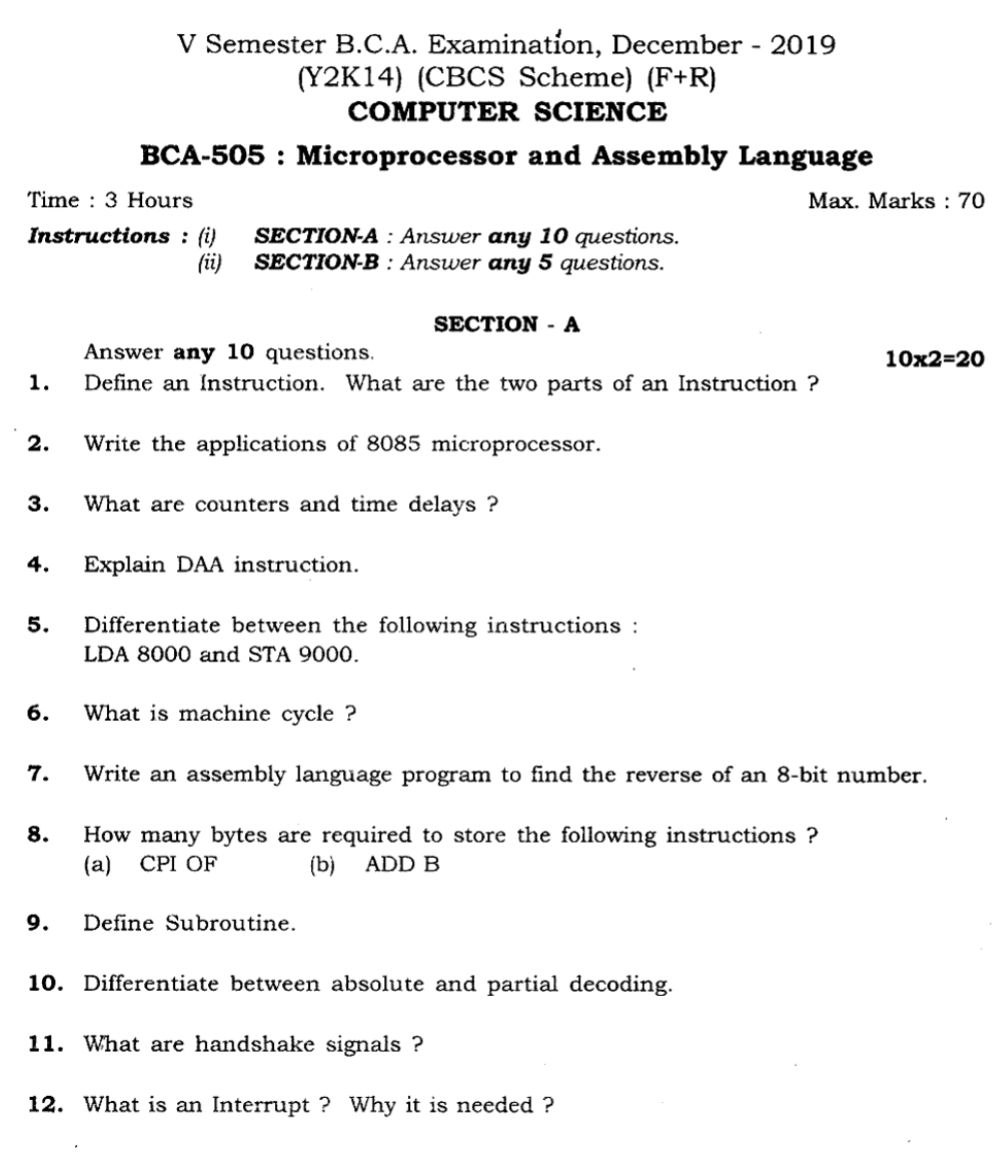
* Address register –It contains the address to specify the desired location in memory.
* Word count register –It contains the number of words to be transferred.
* Control register –It specifies the transfer mode.

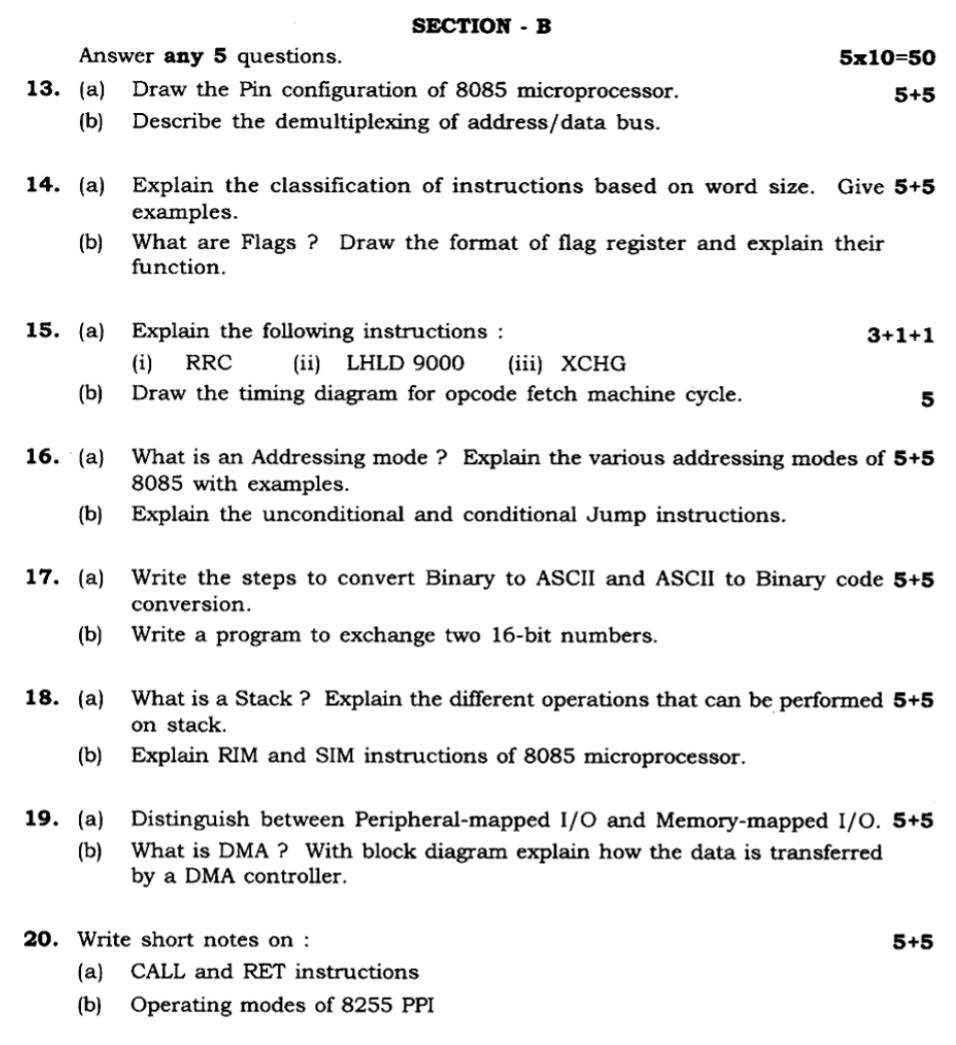
B.

The address bus has 8 signal lines A8 - A15 which are unidirectional.

The other 8 address bits are multiplexed(time shared) with the 8 data bits. So, the bits ADO -AD7are bi-directional and serve as A0 -A7and DO -D7at the same time.

it becomes obvious that the AD-7 to AD-0 lines are serving a dual purpose and that they need to be demultiplexed to get all the information. The high order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period and they would be lost if they are not saved externally. Also, notice that the low order bits of the address disappear when they are needed most. To make sure we have the entire address for the full three clock cycles, we will use an external latch to save the value of AD7-ADO when it is carrying the address bits. We use the ALE signal to enable this latch. the address is saved and the AD7-ADO lines can be used for their purpose as the bi-directional data lines.





1. An instruction is a binary pattern designed inside a microprocessor to perform a specific function.

* The entire group of instructions that a microprocessor supports is called Instruction Set.
* 8085 has 246 instructions.
* Each instruction is represented by an 8-bit binary value.
* These 8-bits of binary value is called Op-Code or Instruction Byte.
* Instruction in memory has two parts: opcode and operands.
* The opcode identifies the operation that instruction does
* The operands are subjects of the operation, such as data values, registers, or memory addresses.
* Due to variety of opcodes and operands, instructions may occupy different sizes of bytes in memory.
* Instructions with no operands can have *implied operands*, those that are not explicitly shown.

1. the extensive use of 8085 in various applications, the microprocessor is provided with an instruction set consisting of various instructions such as MOV, ADD, SUB, JMP, etc. These instructions are written in the form of a program which is used to perform various operations such as branching, addition, subtraction, [bitwise logical](https://en.wikipedia.org/wiki/Bitwise_operation), and [bit shift](https://en.wikipedia.org/wiki/Bit_shifting) operations. More complex operations and other arithmetic operations must be implemented in software. For example, multiplication is implemented using a [multiplication algorithm](https://en.wikipedia.org/wiki/Multiplication_algorithm).
2. **counter**

A counter is designed simply by loading appropriate number into one of the registers and using INR or DNR instructions. Loop is established to update the count.

**time delay**

Procedure used to design a specific delay. A register is loaded with a number , depending on the time delay required and then the register is decremented until it reaches zero by setting up a loop with conditional jump instruction.

1. DAA -

The DAA (Decimal Adjust after Addition) instruction allows addition of numbers represented in 8-bit packed BCD code. It is used immediately after normal addition instruction operating on BCD codes. This instruction assumes the AL register as the source and the destination, and hence it requires no operand.

1. LDA

Load the accumulator with a data from the the memory in a particular address. It's is a three byte instruction.First byte is the opcode, second & third byte together will give the address from where data is to be moved into accumulator.

STA ,

Store the current contents of the Accumulator in to the memory, whose address will be given as part of instruction. Here also first byte is opcode and the second & third byte together give the address. Both are examples of Direct Addressing method.

"LDA 8000" is Load the data in accumulator from the memory location 8000. 8000 can be anything from 0000H to FFFFH.

"STA 9000" is Store the data from accumulator in the memory location 9000.

1. The machine cycle (or **instruction cycle)** is the main activity performed of the CPU to execute the program instruction. The CPU continuously performs the machine cycle necessary for program execution. It consists of 4 steps: Fetch, Decode, Execute and Store which are continuously and repetitively performed sequentially in the cycle by the processor

|  |  |  |
| --- | --- | --- |
| MEMORY ADDRESS | MNEMONICS |  |
| 2000 | LDA 2050 |  |
| 2003 | RLC |  |
| 2004 | RLC |  |
| 2005 | RLC |  |
| 2006 | RLC |  |
| 2007 | STA 3050 |  |
| 200A | HLT |  |

1. I AM NOT SURE

instruction **CPI OF** requires 2-Bytes,

instruction **ADD B** requires 4-Bytes,

1. a subroutine is a sequence of program instructions that perform a specific task, packaged as a unit. This unit can then be used in programs wherever that particular task have to be performed. A subroutine is often coded so that it can be started (called) several times and from several places during one execution of the program, including from other subroutines, and then branch back (return) to the next instruction after the call, once the subroutine’s task is done
2. **ABSOLUTE DECODING**

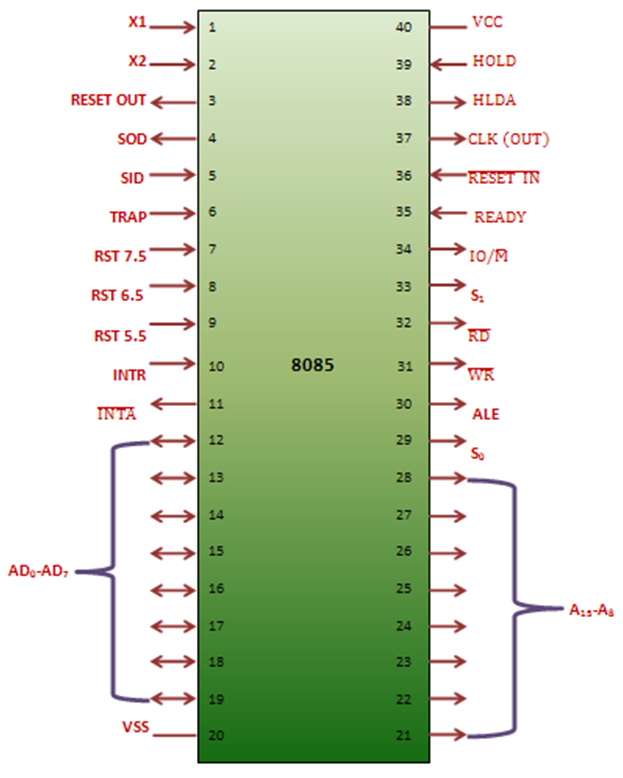
The decoding in which all available address line (16 lines in memory mapped and 8 lines in peripheral mapping) are used for decoding to generate a unquie address is called absolute decoding

PARTIAL DECODING  
The decoding in which all available address line(16 lines in memory mapping and 8 lines in peripheral mapping) are not used for decoding resulting in multiple address for same port is called partial decoding.

1. Handshaking is a I/O control method to synchronize I/O devices with the microprocessor. As many I/O devices accepts or release information at a much slower rate than the microprocessor, this method is used to control the microprocessor to work with a I/O device at the I/O devices data transfer rate.
2. 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.

PART-B

1. A. Pin diagram of 8085



B.

The address bus has 8 signal lines A8 - A15 which are unidirectional.

The other 8 address bits are multiplexed(time shared) with the 8 data bits. So, the bits ADO -AD7are bi-directional and serve as A0 -A7and DO -D7at the same time.

it becomes obvious that the AD-7 to AD-0 lines are serving a dual purpose and that they need to be demultiplexed to get all the information. The high order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period and they would be lost if they are not saved externally. Also, notice that the low order bits of the address disappear when they are needed most. To make sure we have the entire address for the full three clock cycles, we will use an external latch to save the value of AD7-ADO when it is carrying the address bits. We use the ALE signal to enable this latch. the address is saved and the AD7-ADO lines can be used for their purpose as the bi-directional data lines.

1. A

The 8085 instruction set is classified into 3 categories by considering the length of the instructions. In 8085, the length is measured in terms of “byte” rather then “word” because 8085 microprocessor has 8-bit data bus. Three types of instruction are: 1-byte instruction, 2-byte instruction, and 3-byte instruction.

**1. One-byte instructions –**  
In 1-byte instruction, the opcode and the operand of an instruction are represented in one byte.

* **Example-1:**  
  Task- Copy the contents of accumulator in register B.
* **Mnemonic- MOV B, A**
* Opcode- MOV
* Operand- B, A
* Hex Code- 47H

Binary code- 0100 0111

**2. Two-byte instructions –**  
Two-byte instruction is the type of instruction in which the first 8 bits indicates the opcode and the next 8 bits indicates the operand.

* **Example-1:**  
  Task- Load the hexadecimal data 32H in the accumulator.
* **Mnemonic- MVI A, 32H**
* Opcode- MVI
* Operand- A, 32H
* Hex Code- 3E
* 32
* Binary code- 0011 1110

0011 0010

**3. Three-byte instructions –**  
Three-byte instruction is the type of instruction in which the first 8 bits indicates the opcode and the next two bytes specify the 16-bit address. The low-order address is represented in second byte and the high-order address is represented in the third byte.

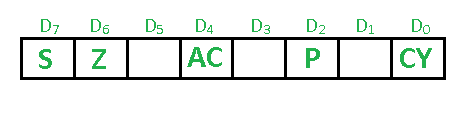
* **Example-1:**  
  Task- Load contents of memory 2050H in the accumulator.
* **Mnemonic- LDA 2050H**
* Opcode- LDA
* Operand- 2050H
* Hex Code- 3A
* 50
* 20
* Binary code- 0011 1010
* 0101 0000

0010 0000

B.

The **Flag register** is a Special Purpose Register. Depending upon the value of result after any arithmetic and logical operation the flag bits become set (1) or reset (0). In 8085 microprocessor, flag register consists of 8 bits and only 5 of them are useful.

The 5 flags are:



**Sign Flag (S) –** After any operation if the MSB (B(7)) of the result is 1, it indicates the number is negative and the sign flag becomes set, i.e. 1. If the MSB is 0, it indicates the number is positive and the sign flag becomes reset i.e. 0.

**Zero Flag (Z) –** After any arithmetical or logical operation if the result is 0 (00)H, the zero flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.  
00H zero flag is 1.  
from 01H to FFH zero flag is 0

**Auxiliary Carry Flag (AC) –** This flag is used in BCD number system(0-9). If after any arithmetic or logical operation D(3) generates any carry and passes on to B(4) this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0. This is the only flag register which is not accessible by the programmer

**Parity Flag (P) –** If after any arithmetic or logical operation the result has even parity, an even number of 1 bits, the parity register becomes set i.e. 1, otherwise it becomes reset i.e. 0.

**Carry Flag (CY) –** Carry is generated when performing n bit operations and the result is more than n bits, then this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.

1. A

**RRC**

This instruction does not require any operand after the opcode. It operates the content of accumulator and the result is also stored in the accumulator. The Rotate instruction is used to rotating the bits of accumulator.

**RRC** stands for “Rotate Right Accumulator”. With the help of this instruction, we can rotate the Accumulator current content to the right by 1-bit position.

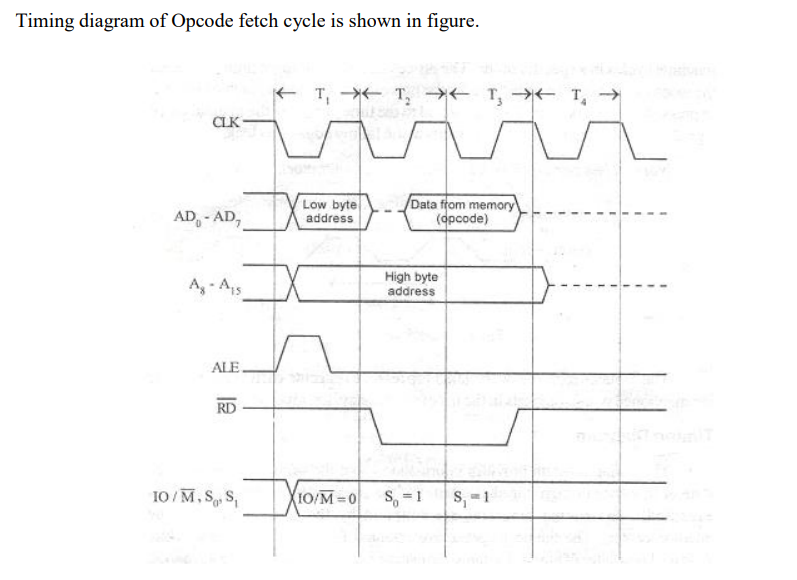
**LHLD**

**LHLD** is a mnemonic that stands for Load HL pair using Direct addressing from memory location whose 16-bit address is denoted as a16. So the previous content of HL register pair will get updated with the new 16-bits value. As HL pair has to be updated, so data comes from two consecutive memory locations starting at the address a16 and also from next address location. This instruction uses absolute addressing mode for specifying the data in the instruction. It occupies 3-Bytes in the memory.

**XCHG**

**XCHG**, which stands for eXCHanGe. This is an instruction to exchange contents of HL register pair with DE register pair. This instruction uses implied addressing mode. As it is1-Byte instruction, so It occupies only 1-Byte in the memory. After execution of this instruction, the content between H and D registers and L and E registers will get swapped respectively.

B.



A

Addressing Modes– The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

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. 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.
3. 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.
4. 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.
5. 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.

B. there are a set of jump instructions, which can transfer program control to a certain memory location. So after these branching mnemonics we shall have to mention 16-bit target address of the location. These jump instructions can be divided into two categories–

* Unconditional jump instructions and
* Conditional jump instructions

Under unconditional jump instructions there is only one mnemonic i.e. JUMP. But under conditional Jump instructions we are having 8 different mnemonics. We know that there are 5 flag bits in 8085 Flag register. They are S, Z,P, Cy, AC. Out of them only on AC flag bit, there is no jump instruction. But for rest 4 flag bits, we are having 8 conditional jump instructions depending upon their 1 or 0 i.e. TRUE and FALSE values respectively.

| **Mnemonics, Operand** | **Opcode(in HEX)** | **Bytes** |
| --- | --- | --- |
| JC Label | DA | 3 |
| JM Label | FA | 3 |
| JMP Label | C3 | 3 |

17.

A

Binary to ascii

| ADDRESS | MNEMONICS | |
| --- | --- | --- |
| 2000 | LDA 2050 | |
|  |  | |
| 2003 | CALL 2500 | |
| 2006 | STA 3050 | |
| 2009 | LDA 2050 | |
| 200C | RLC | |
| 200D | RLC | |
| 200E | RLC | |
| 200F | RLC | |
| 2010 | CALL 2500 | |
| 2013 | STA 3051 | |
| 2016 | HLT | |
|  | |

Ascii to Binary

| **Address** | **Labels** | **Mnemonics** |
| --- | --- | --- |
| **F000** |  | LXI H, 8000H |
| **F003** |  | MOV A,M |
| **F004** |  | CPI 58H |
| **F006** |  | JNC NUM |
| **F009** |  | SUI 37H |
| **F00B** |  | JMP STORE |
| **F00E** | NUM | SUI 30H |
| **F010** | STORE | INX H |
| **F011** |  | MOV M,A |
| **F012** |  | HLT |

B.

| **Address** | **Mnemonics** |
| --- | --- |
| 2000 | LHLD 2500H |
| 2003 | XCHG |
| 2004 | LHLD 2600H |
| 2007 | SHLD 2500H |
| 200A | XCHG |
| 200B | SHLD 2600H |
| 200E | HLT |

18. A

The stack is a LIFO (last in, first out) data structure implemented in the RAM area and is used to store addresses and data when the microprocessor branches to a subroutine. Then the return address used to get pushed on this stack. Also to swap values of two registers and register pairs we use the stack as well.

SP is a special purpose 16-bit register. It contains a memory address. Suppose SP contents are FC78H, then the 8085 interprets it as follows.

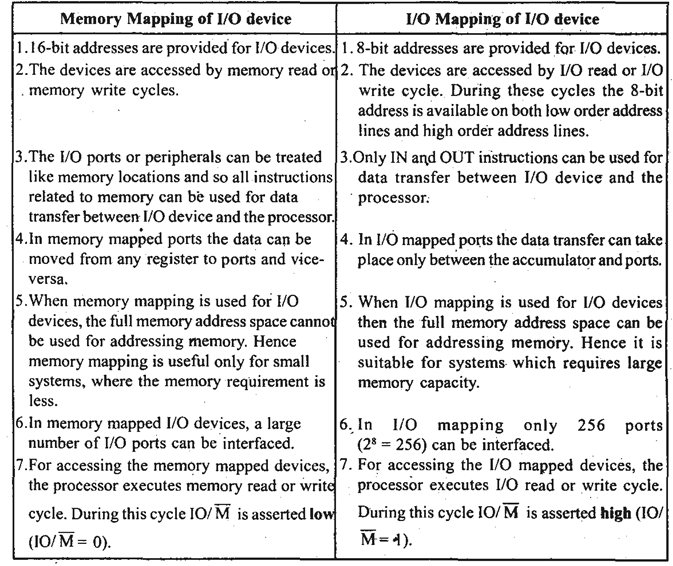
On a stack, we can perform two operations. PUSH and POP. In case of PUSH operation, the SP register gets decreased by 2 and new data item used to insert on to the top of the stack. On the other hand, in case of POP operation, the data item will have to be deleted from the top of the stack and the SP register will get increased by the value of 2.

B.

**SIM and RIM instructions in 8085 microprocessor:**

| Sr. No. | Sim Instruction | Rim Instruction |
| --- | --- | --- |
| 1 | SIM stands for Set Interrupt Mask. | RIM stands for Read Interrupt Mask. |
| 2 | It is responsible for masking/unmasking of RST 7.5, RST 6.5 and RST 5.5. | It checks whether RST 7.5, RST 6.5, RST 5.5 are masked or not. |
| 3 | It resets to 0 RST 7.5 flip flop. | It checks whether interrupts are enabled or not and to check whether RST 7.5, RST 6.5 or RST 5.5 interrupts are pending or not. |
| 4 | The content of the Accumulator decides the action to be taken. So before executing the SIM instruction, it is mandatory to initialize Accumulator with the required value. | The contents of the Accumulator after the execution of the RIM instruction provide this information.Thus, it is essential to look into the Accumulator contents after the RIM instruction is executed. |
| 5 | SIM instruction can be used for serial output of data. | RIM instruction can be used for serial input of data. |
| 6 | Its opcode(in Hex) is 30. | Its opcode(in Hex) is 20. |

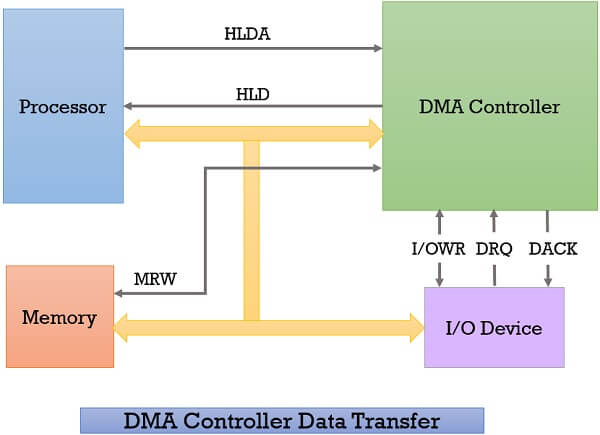
19.A



B.

Direct Access Media (DMA) :

DMA Controller is a hardware device that allows I/O devices to directly access memory with less participation of the processor. DMA controller needs the same old circuits of an interface to communicate with the CPU and Input/Output devices.



1. Whenever an I/O device wants to transfer the data to or from memory, it sends the DMA request (**DRQ**) to the DMA controller. DMA controller accepts this DRQ and asks the CPU to hold for a few clock cycles by sending it the Hold request (**HLD**).
2. CPU receives the Hold request (HLD) from DMA controller and relinquishes the bus and sends the Hold acknowledgement (**HLDA**) to DMA controller.
3. After receiving the Hold acknowledgement (HLDA), DMA controller acknowledges I/O device **(DACK)** that the data transfer can be performed and DMA controller takes the charge of the system bus and transfers the data to or from memory.
4. When the data transfer is accomplished, the DMA raise an **interrupt** to let know the processor that the task of data transfer is finished and the processor can take control over the bus again and start processing where it has left.

20.

A.

**Call Instructions –** The call instruction transfers the program sequence to the memory address given in the operand. Before transferring, the address of the next instruction after CALL is pushed onto the stack. Call instructions are 2 types: Unconditional Call Instructions and Conditional Call Instructions.

**(a) Unconditional Call Instructions:** It transfers the program sequence to the memory address given in the operand.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| CALL | address | Unconditionally calls | CALL 2050 |

**(b) Conditional Call Instructions:** Only if the condition is satisfied, the instructions executes.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| CC | address | Call if carry flag is 1 | CC 2050 |
| CNC | address | Call if carry flag is 0 | CNC 2050 |

**Return Instructions –** The return instruction transfers the program sequence from the subroutine to the calling program. Return instructions are 2 types: Unconditional Jump Instructions and Conditional Jump Instructions.

**(a) Unconditional Return Instruction:** The program sequence is transferred unconditionally from the subroutine to the calling program.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| RET | none | Return from the subroutine unconditionally | RET |

**(b) Conditional Return Instruction:** The program sequence is transferred unconditionally from the subroutine to the calling program only is the condition is satisfied.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| RC | none | Return from the subroutine if carry flag is 1 | RC |

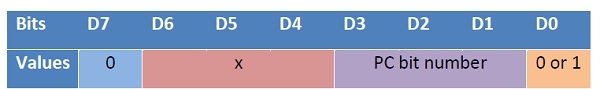
B.

There are two different modes of 8255. These modes are:

* Bit Set Reset (BSR) Mode
* Input/ Output Mode

**Bit Set Reset (BSR) Mode**

This mode is used to set or reset the bits of the Port-C only. For BSR mode always D7 will be 0. The control register is looking like this:



The (D3, D2, D1) will be 000 to 111. In this mode it affects only one bit of Port C at a time. When user set the bit, it remains set until user unset it. The user needs to load the bit pattern in control register to change the bit.

**Input/ Output Mode**

This mode is selected when the D7 bit of the control register is 1.

This mode has also three different modes. These modes are Mode 0 and Mode 1 and Mode 3.