**Name:Niraj Naik**

**Roll no: 21CO34**

**Class:COMP-A**

1)What is computer organization?

Ans:Design of the components and functional blocks using which computer systems are built.

example : In your computer some functional blocks like your processor unit; inside processor unit we will be seeing that we have many other components like registers, ALU and other units.Take a small example let say I will have an adder, but what kind of adder I will be having that is to the discretion of the computer organization, whether we will have a carry save adder or a carry look ahead adder or anything else. So, these are the two different aspects of computer organization and computer architecture that we will be seeing in this particular course.

2)What is computer architecture?

Ans:How to integrate the components to build a computer system to achieve a desired level of performance is called computer architecture. Computer architecture consists of those attributes of the system that are visible to the programmer. By this what we mean is how the various components of a computer system are integrated to achieve the desired level of performance.

In an analogy, I can say that you think of an architect who does who plans the entire design of your house, but it is ultimately the civil engineers who actually does the exact building like what kind of construction will be taken care of, how the construction will be taken care of, how much percentage of cement, bricks will be there , will be taken care by a civil engineer.

3)What are structure and fuction?

Ans: Structures (also called structs) are a way to group several related variables into one place. Each variable in the structure is known as a member of the structure. Unlike an array, a structure can contain many different data types (int, float, char, etc.).A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions.You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.The C standard library provides numerous built-in functions that your program can call. For example, strcat() to concatenate two strings, memcpy() to copy one memory location to another location, and many more functions.A function can also be referred to as a method or a subroutine or a procedure, etc.

4. What are functions of computers?

Input: The transferring of information into a computer system; data entry. In addition to the more commonly found methods of input (keyboard, scanner, flash drive, CD), data can also be entered via vocal dictation and even body movements. For our purposes, input will be accomplished when you, the programmer, type source code at the keyboard or open a previously typed program.

Processing: The manipulation and control of information (data) within the computer system. The Central Processing Unit (CPU) is the "brain" of a computing device.Manipulations are handled by the Control Unit, the Arithmetic Logic Unit and Temporary Storage. The Control Unit is the computer's "policeman" that directs the traffic of the computer's "thinking". The Arithmetic Logic Unit performs mathematical and logical computations, and the Temporary Storage (RAM) holds the "thoughts" of the CPU.

Storage: The means by which information can be "permanently" saved (until such time as you wish to delete it). This storage may be a computer's internal or external storage. Storage usually occurs on a hard drive, flash drive, a CD, or a cloud account.

Output:The displaying of information. Output may occur via a display monitor, audio file, hard-copy printing on paper, or printing as 3-D models. For our purposes, output will be accomplished when your program displays information on the monitor

5)Distinguish between computer organisation and computer architecture

Ans:Computer Organisation:

1)Design of the components and functional blocks using which computer systems are built.

2)Computer organization refers to the operational units and their interconnections that realize the architectural specifications.

3)example : In your computer some functional blocks like your processor unit; inside processor unit we will be seeing that we have many other components like registers, ALU and other units.Take a small example let say I will have an adder, but what kind of adder I will be having that is to the discretion of the computer organization, whether we will have a carry save adder or a carry look ahead adder or anything else. So, these are the two different aspects of computer organization and computer architecture that we will be seeing in this particular course.

Computer Architecture:

1)How to integrate the components to build a computer system to achieve a desired level of performance is called computer architecture.

2)Computer architecture refers to those parameters of a computer system that are visible to a programmer or those parameters that have a direct impact on the logical execution of a program.

3)In an analogy, I can say that you think of an architect who does who plans the entire design of your house, but it is ultimately the civil engineers who actually does the exact building like what kind of construction will be taken care of, how the construction will be taken care of, how much percentage of cement, bricks will be there , will be taken care by a civil engineer.

6)List and explain different components of a computer.(structural component)

Ans:A. Central Processor Unit (CPU) :

Central processor unit consists of two basic blocks :

The program control unit has a set of registers and control circuit to generate control signals.

The execution unit or data processing unit contains a set of registers for storing data and an Arithmatic and Logic Unit (ALU) for execution of arithmatic and logical operations.

In addition, CPU may have some additional registers for temporary storage of data.

B. Input Unit :

With the help of input unit data from outside can be supplied to the computer. Program or data is read into main storage from input device or secondary storage under the control of CPU input instruction.

Example of input devices: Keyboard, Mouse, Hard disk, Floppy disk, CD-ROM drive etc.

C. Output Unit :

With the help of output unit computer results can be provided to the user or it can be stored in stograge device permanently for future use. Output data from main storage go to output device under the control of CPU output instructions.

Example of output devices: Printer, Monitor, Plotter, Hard Disk, Floppy Disk etc.

D. Memory Unit :

Memory unit is used to store the data and program. CPU can work with the information stored in memory unit. This memory unit is termed as primary memory or main memory module. These are basically semi conductor memories.

There ate two types of semiconductor memories -

Volatile Memory : RAM (Random Access Memory).

Non-Volatile Memory : ROM (Read only Memory), PROM (Programmable ROM)

EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM).

Secondary Memory :

There is another kind of storage device, apart from primary or main memory, which is known as secondary memory. Secondary memories are non volatile memory and it is used for permanent storage of data and program.

Example of secondary memories:

Hard Disk, Floppy Disk, Magenetic Tape ------ These are magnetic devices,

CD-ROM ------ is optical device

Thumb drive (or pen drive) ------ is semiconductor memory.

7)List and explain different structural components of a processor(CPU)

Ans:The CPU can work with the information available in main memory only.To access the data from memory, we need two special registers one is known as Memory Data Register (MDR) and the second one is Memory Address Register (MAR). CPU brings instruction and data from main memory, performs the tasks as per the instuction fetch from the memory. After completion of operation, CPU stores the result back into the memory.

We need two operation to work with memory.

READ Operation:This operation is to retrive the data from memory and bring it to CPU register

WRITE Operation:This operation is to store the data to a memory location from CPU register

To transfer the data from CPU to memory module and vice-versa, we need some connection. This is termed as DATA BUS.

8)What general categories of function are specified by computer engineer?

The categories are processor-memory, processor-I/O, data processing, and c ontrol.

Processor-memory: Data may be transferred from processor to memory or from memory to processor. It is used for storing programs and data. The memory locations of memory unit is uniquely specified by the memory address of the location. M(X) is used to indicate the location of the memory unit M with address X.The data transfer between memory unit and CPU takes place with the help of data register DR.

Processor-I/O: Data may be transferred to or from a peripheral device by transferring between the processor and an I/O module.Input devies are used to put the information into computer. With the help of input devices we can store information in memory so that CPU can use it. Program or data is read into main memory from input device or secondary storage under the control of CPU input instruction.Output devices are used to output the information from computer. If some results are evaluated by computer and it is stored in computer, then with the help of output devices, we can present it to the user. Output data from the main memory go to output device under the control of CPU output instruction.

Data processing: The processor may perform some arithmetic or logic operation on data.The processor may perform some arithmetic or logic operation on data.The data processing unit contains a high speed registers intended for temporary storage of instructions, memory addresses and data.The main action specified by instructions are performed by the arithmatic-logic circuits of the data processing unit.

Control:An instruction may specify that the sequence of execution be altered.Two instructions are fetch simultaneously from M and transferred to the program control unit.In the decoding phase, the control circuits generate the required control signals to perform the specified operation in the instruction.

9)List and briefly define the possible state that define instruction execution.

Ans: Instruction address calculation (iac): Determine the address of the next instruction to be executed.Determine the address of the next instruction to be executed. Usually, this involves adding a fixed number to the address of the previous instruction.

Instruction operation decoding (iod): Analyze instruction to determine type of operation to be performed and operand(s) to be used.

Operand address calculation (oac): If the operation involves reference to an operand in memory or available via I/O, then determine the address of the operand.

Operand fetch (of): Fetch the operand from memory or read it in from I/O.

Data operation (do): Perform the operation indicated in the instruction.

Operand store (os): Write the result into memory or out to I/O.

10)List and briefly define 2 approachesto delaing with multiple interrupts.

Ans:CPU can read this word and can check bits one by one which of them is 1. It can identify which of the bits is 1. This process is sometimes called polling. Polling means scanning one by one to find out which device has sent the interrupt.

Each device can have a status bit indicating whether it has interrupted, and CPU can go on polling those bits one by one to find out that which device had interrupted. A better alternative will be to use interrupt vector concept that we talked about earlier, because when the CPU is doing polling, some additional time will be required because there will be program which will be running which will be shifting the bits one by one and will be checking which bit is 1, which bit is 0.

11)What types of transfers must a computer interconnections support?

Ans:Memory to processor: The processor reads an instruction or a unit of data from memory.

Processor to memory: The processor writes a unit of data to memory.

I/O to processor: The processor reads data from an I/O device via an I/O module.

Processor to I/O: The processor sends data to the I/O device.

I/O to or from memory: For these two cases, an I/O module is allowed to exchange data directly with memory, without going through the processor, using direct memory access (DMA).

12)What is the benefits of using multiple bus architecture compared to single bus architecture

modern processors use this multi-bus architecture. Here, within your processor to communicate between various registers you have multiple-bus. The advantage we get is that more operations can be performed and we get results much faster. So, there will be a overall improvement in instruction execution time. Some smaller parasitic capacitance can be there and hence smaller delay.

13)Briefly explain sign magnitude,2’s complement,biased

Ans:In signed-magnitude form, one particular bit is used to indicate the sign of the number, whether it is a positive number or a negative number. Other bits are used to represent the magnitude of the number.

For an n-bit number, one bit is used to indicate the signed information and remaining *(n-1) bits* are used to represent the magnitude. Therefore, the range is from .

Generally, Most Significant Bit (MSB) is used to indicate the sign and it is termed as signed bit. 0 in signed bit indicates positive numvber and 1 in signed bit indicates negative number.

Consider the eight bit number 01011100, 2's complements of this number is 10100100. If we perform the follwoing addition:

| **0 1 0 1 1 1 0 0** |
| --- |
| **1 0 1 0 0 0 1 1** |
| -------------------------------- |
| **1 0 0 0 0 0 0 0 0** |

Since we are considering an eight bit number, so the 9th bit (MSB) of the result can not be stored. Therefore, the final result is 00000000.Since the addition of two number is 0, so one can be treated as the negative of the other number. So, 2's complement can be used to represent negative number.

**Biased representation:** A fixed value, called the bias, is added to the integer.

14) explain how to determine if a number is negative in following representation:sign magnitude,2’s complement,biased

Ans:In signed-magnitude form, one particular bit is used to indicate the sign of the number, whether it is a positive number or a negative number.Generally, Most Significant Bit (MSB) is used to indicate the sign and it is termed as signed bit. 0 in signed bit indicates positive numvber and 1 in signed bit indicates negative number.

Since the addition of two number is 0, so one can be treated as the negative of the other number. So, 2's complement can be used to represent negative number.

In biased representation, a number is negative if the value of the representation is less than the bias.

15)What is sign extension rule for 2’s complement number

Ans: To increase the number of bits in a representation of an integer in two's complement, add copies of the leftmost bit (the sign bit) to the left until you have the desired number of bits. This is called **sign extension**.

16)How can you perform negation of an integer in 2’s complement representation?

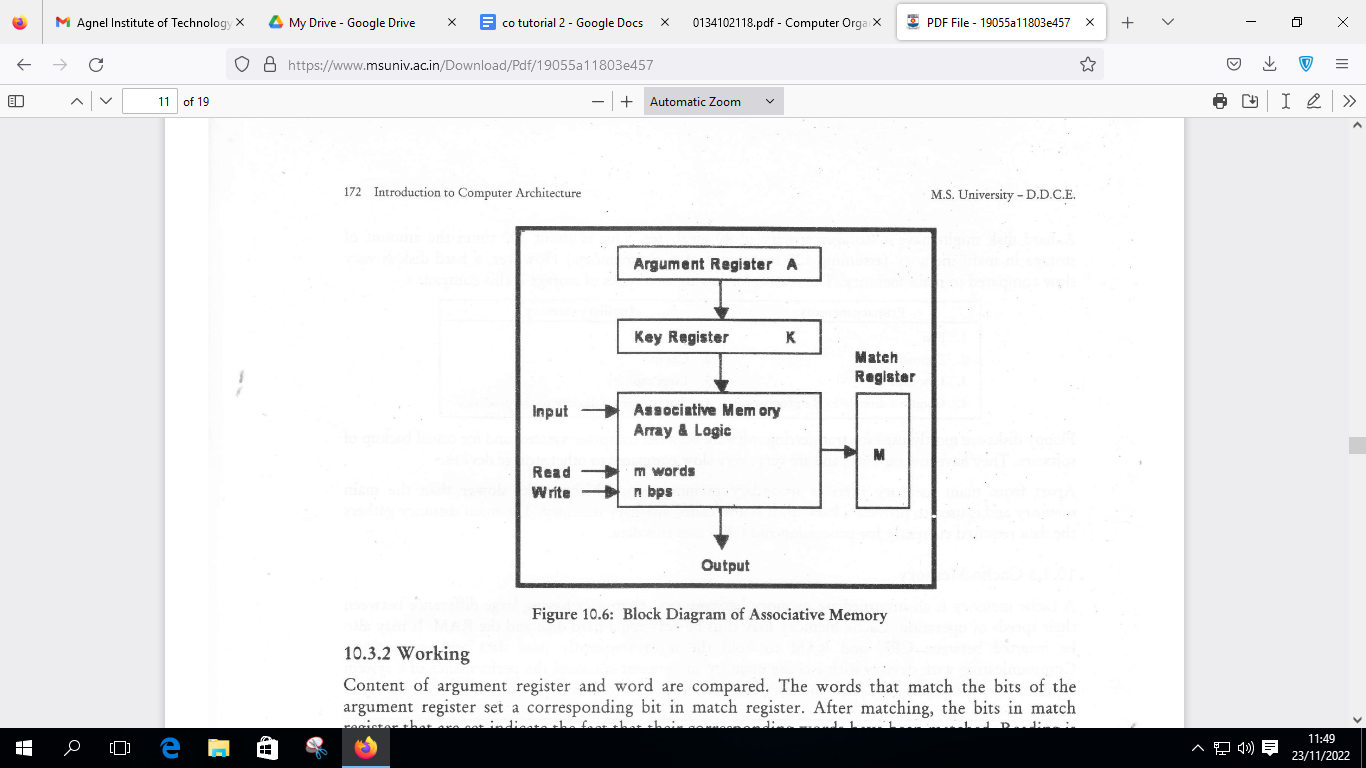
| Ans:In Twos Complement Notation is also used to represent a negative number. In this notation 1 is added to the Ones Complement Notation of a negative number. For example, since 11011001 represents -38 in Ones Complement Notation 11011010 used represent -38 in Twos Complement Notation.  Negating a number in two’s complement representation can be defined as a two-step operation:   1. Invert all the bits 2. Add 1 to the result, treating the bit pattern as an unsigned value   17)When does the 2’s complement operation on an n-bit integer produce the same integer?  Ans:There are two cases where you get back the same value:   * When the number is zero * When the number is the smallest negative value representable in n bits.   18)Explain Associative Memory  -Associative memory is also known as content addressable memory (CAM) / associative storage / associative array. It is a special type of memory that is optimized for performing searches through data, as opposed to providing a simple direct access to the data based on the address.  -Associative memory of conventional semiconductor memory (usually RAM) with added comparison circuity that enables a search operation to complete in a single clock cycle. It is a hardware search engine, a special type of computer memory used in certain very high search application.    19. Draw and explain the Intel 8085 processor.  - Several key components of the 8085 are:  - Incrementer / decrementer address latch: Logic that can add 1 to or subtract 1 from the contents of the stack pointer or program counter. This saves time by avoiding the use of the ALU for this purpose.  - Interrupt control: This module handles multiple levels of interrupt signals.  - Serial I/O control: This module interfaces to devices that communicate 1 bit at a time.  - The external signals into and out of the 8085 are linked to the external system bus. These signals are the interface between the 8085 processor and the rest of the system. For e.g:  - High Address (A15–A8): The high-order 8 bits of a 16-bit address.  - Address/Data (AD7–AD0): The lower-order 8 bits of a 16-bit address or 8 bits of data.  - Serial Input Data (SID): A single-bit input to accommodate devices that transmit serially  - Serial Output Data (SOD): A single-bit output to accommodate devices that receive serially.  - Timing and Control Signals CLK (OUT): The system clock. The CLK signal goes to peripheral chips and synchronizes their timing.  - X1, X2: These signals come from an external crystal or other device to drive the internal clock generator.  - Address Latch Enabled (ALE): Occurs during the first clock state of a machine cycle and causes peripheral chips to store the address lines.  - Status (S0, S1): Control signals used to indicate whether a read or write operation is taking place.  - IO/M: Used to enable either I/O or memory modules for read and write operations.  - Read Control (RD): Indicates that the selected memory or I/O module is to be read and that the data bus is available for data transfer.  - Write Control (WR): Indicates that data on the data bus is to be written into the selected memory or I/O location.  - Hold: Requests the CPU to relinquish control and use of the external system bus. The CPU will complete execution of the instruction presently in the IR and then enter a hold state, during which no signals are inserted by the CPU to the control, address, or data buses.  - Hold Acknowledge (HOLDA): This control unit output signal acknowledges the HOLD signal and indicates that the bus is now available.  - READY: Used to synchronize the CPU with slower memory or I/O devices. When an addressed device asserts READY, the CPU may proceed with an input (DBIN) or output (WR) operation.  - Interrupt Request (INTR) These five lines are used by an external device to interrupt the CPU. The CPU will not honor the request if it is in the hold state or if the interrupt is disabled. An interrupt is honored only at the completion of an instruction. The interrupts are in descending order of priority. Interrupt Acknowledge Acknowledges an interrupt  - Voltage and Ground:  VCC: +5-volt power supply  VSS: Electrical ground          20. Explain READ and WRITE operation of associative memory.  - When a write operation is performed on associative memory, no address or memory location is given to the word. The memory itself is capable of finding an empty unused location to store the word.  - On the other hand, when the word is to be read from an associative memory, the content of the word, or part of the word, is specified. The words which match the specified content are located by the memory and are marked for reading. | | |
| --- | --- | --- |

1)explain associative memory

: This is a random access type of memory that enables one to make a comparison of desired bit locations within a word for a specified match, and to do this for all words simultaneously. Thus, a word is retrieved based on a portion of its contents rather than its address. As with ordinary random-access memory, each location has its own addressing mechanism, and retrieval time is

constant independent of location or prior access patterns. Cache memories may employ associative access Associative memory is often referred to as Content Addressable Memory (CAM).

2)draw and explain block diagram of associative memory



The block diagram of associative memory is shown in the figure. It includes a memory array and logic for m words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word.

The match register M has m bits, one for each memory word. Each word in memory is related in parallel with the content of the argument register.

The words that connect the bits of the argument register set an equivalent bit in the match register. After the matching process, those bits in the match register that have been set denote the fact that their equivalent words have been connected.

Reading is proficient through sequential access to memory for those words whose equivalent bits in the match register have been set.

The key register supports a mask for selecting a specific field or key in the argument word. The whole argument is distinguished with each memory word if the key register includes all 1's. Hence, there are only those bits in the argument that have 1's in their equivalent position of the key register are compared. Therefore, the key gives a mask or recognizing a piece of data that determines how the reference to memory is created.

3)explain read and write operation of associative memory

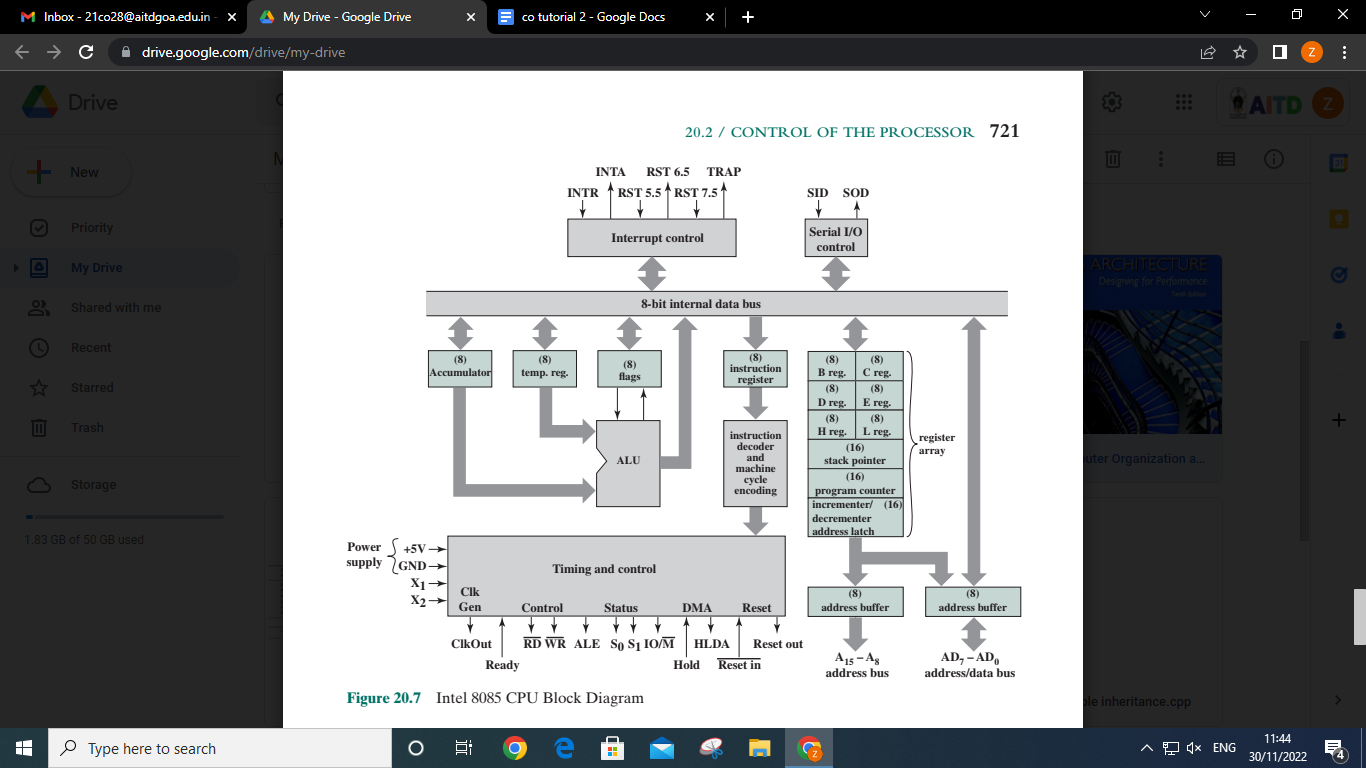
READ

Memory read operation transfers the desired word to address lines and activates the read control line. MDR

can contain any garbage value and MAR is containing memory address. After the execution of read instruction, the data of memory location will be read and the MDR will get updated by the value of the memory location

WRITE

Memory write operation transfers the address of the desired word to the address lines, transfers the data bits to be stored in memory to the data input lines. Then it activates the write control line.

4) draw and explain block diagram of 8085

let us consider

the Intel 8085. Its organization is shown in Figure 20.7. Several key components that

may not be self-explanatory are:

■ Incrementer/decrementer address latch: Logic that can add 1 to or subtract 1

from the contents of the stack pointer or program counter. This saves time by

avoiding the use of the ALU for this purpose.

■ Interrupt control: This module handles multiple levels of interrupt signals.

■ Serial I/O control: This module interfaces to devices that communicate 1 bit

at a time.

The control unit is identified as having two components labeled (1) instruc-

tion decoder and machine cycle encoding and (2) timing and control. A discussion

of the first component is deferred until the next section. The essence of the control

unit is the timing and control module. This module includes a clock and accepts as

inputs the current instruction and some external control signals. Its output consists

of control signals to the other components of the processor plus control signals to

the external system bus.

The timing of processor operations is synchronized by the clock and controlled

by the control unit with control signals. Each instruction cycle is divided into from

one to five machine cycles; each machine cycle is in turn divided into from three to

five states. Each state lasts one clock cycle. During a state, the processor performs

one or a set of simultaneous micro-operations as determined by the control signals.

The number of machine cycles is fixed for a given instruction but varies

from one instruction to another. Machine cycles are defined to be equivalent to

bus accesses. Thus, the number of machine cycles for an instruction depends on

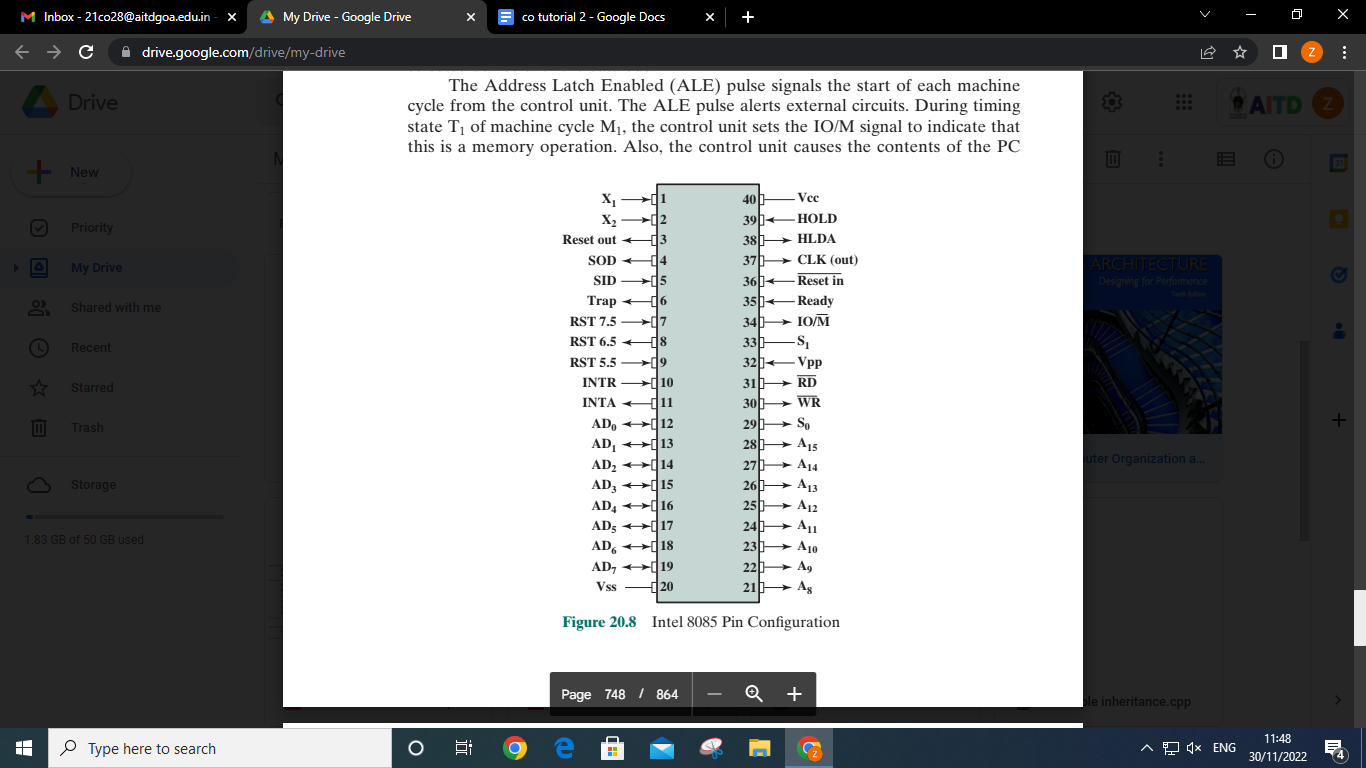
the number of times the processor must communicate with external devices. For

example, if an instruction consists of two 8-bit portions, then two machine cycles are

required to fetch the instruction. If that instruction involves a 1-byte memory or I/O

operation, then a third machine cycle is required for execution.

5)draw and explain pin diagram of 8085



Address and Data Signals

High Address (A15–A8)

The high-order 8 bits of a 16-bit address.

Address/Data (AD7–AD0)

The lower-order 8 bits of a 16-bit address or 8 bits of data. This multiplexing saves on pins.

Serial Input Data (SID)

A single-bit input to accommodate devices that transmit serially (one bit at a time).

Serial Output Data (SOD)

A single-bit output to accommodate devices that receive serially.

Timing and Control Signals

CLK (OUT)

The system clock. The CLK signal goes to peripheral chips and synchronizes their timing.

X1, X2

These signals come from an external crystal or other device to drive the internal clock generator.

Address Latch Enabled (ALE)

Occurs during the first clock state of a machine cycle and causes peripheral chips to store the address lines.

This allows the address module (e.g., memory, I/O) to recognize that it is being addressed.

Status (S0, S1)

Control signals used to indicate whether a read or write operation is taking place.

IO/M

Used to enable either I/O or memory modules for read and write operations.

Read Control (RD)

Indicates that the selected memory or I/O module is to be read and that the data bus is available for data

transfer.

Write Control (WR)

Indicates that data on the data bus is to be written into the selected memory or I/O location.

Memory and I/O Initiated Symbols

Hold

Requests the CPU to relinquish control and use of the external system bus. The CPU will complete execution

of the instruction presently in the IR and then enter a hold state, during which no signals are inserted by the

CPU to the control, address, or data buses. During the hold state, the bus may be used for DMA operations.

Hold Acknowledge (HOLDA)

This control unit output signal acknowledges the HOLD signal and indicates that the bus is now available.

READY

Used to synchronize the CPU with slower memory or I/O devices. When an addressed device asserts

READY, the CPU may proceed with an input (DBIN) or output (WR) operation. Otherwise, the CPU

enters a wait state until the device is ready.

Interrupt-Related Signals

TRAP

Restart Interrupts (RST 7.5, 6.5, 5.5)

Interrupt Request (INTR)

These five lines are used by an external device to interrupt the CPU. The CPU will not honor the request

if it is in the hold state or if the interrupt is disabled. An interrupt is honored only at the completion of an

instruction. The interrupts are in descending order of priority.

Interrupt Acknowledge

Acknowledges an interrupt.

CPU Initialization

RESET IN

Causes the contents of the PC to be set to zero. The CPU resumes execution at location zero.

RESET OUT

Acknowledges that the CPU has been reset. The signal can be used to reset the rest of the system.

Voltage and Ground

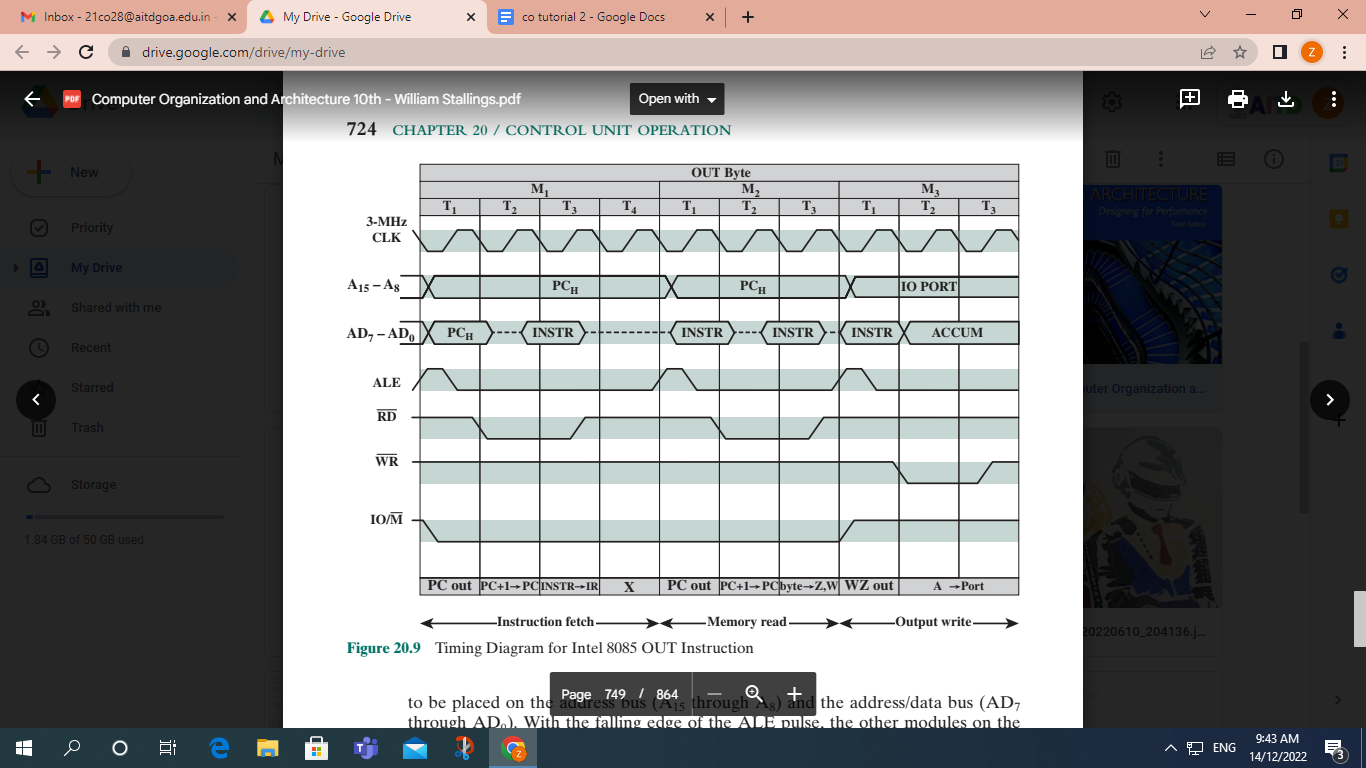
VCC

+5-volt power supply

VSS

Electrical ground

6)timing diagram of 8085 out instruction



gives an example of 8085 timing, showing the value of external

control signals. Of course, at the same time, the control unit generates internal con-

trol signals that control internal data transfers. The diagram shows the instruction

cycle for an OUT instruction. Three machine cycles (M1, M2, M3) are needed. Dur-

ing the first, the OUT instruction is fetched. The second machine cycle fetches the

second half of the instruction, which contains the number of the I/O device selected

for output. During the third cycle, the contents of the AC are written out to the

selected device over the data bus.

The Address Latch Enabled (ALE) pulse signals the start of each machine

cycle from the control unit. The ALE pulse alerts external circuits. During timing

state T1 of machine cycle M1, the control unit sets the IO/M signal to indicate that

this is a memory operation. Also, the control unit causes the contents of the PC

to be placed on the address bus (A15 through A8) and the address/data bus (AD7

through AD0). With the falling edge of the ALE pulse, the other modules on the

bus store the address.

During timing state T2, the addressed memory module places the contents of the

addressed memory location on the address/data bus. The control unit sets the Read

Control (RD) signal to indicate a read, but it waits until T3 to copy the data from the

bus. This gives the memory module time to put the data on the bus and for the signal

levels to stabilize. The final state, T4, is a bus idle state during which the processor

decodes the instruction. The remaining machine cycles proceed in a similar fashion.