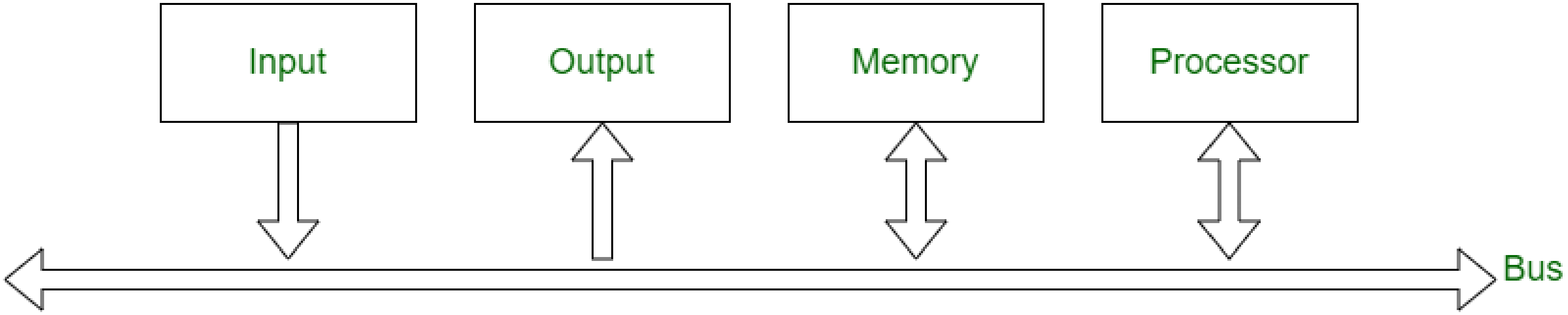
Bus Structure

- **Bus:**
  - ➢A shared transmission medium that connects the components of a computer.
  - ➢It is a collection of wires/cables/lines.
  - ➢It transfers data between the components of a computer.
  - ➢A group of data (word) is transferred between the units of a computer; all bits are transferred simultaneously, over the wires (one bit per wire/line) of a bus.
- **Single Bus Structure:** The simplest way to interconnect the units is to use a single system bus.
  - ➢In this case, only two devices/components/units can actively use the bus at a given time.
  - ➢It is low cost and flexible to add peripheral devices.

Single Bus Structure



Single Bus Structure

Bus Structure (contd.)

- A system bus typically consists of fifty to hundred separate lines.
- The lines are classified into three main groups:
  - ➢**Data lines/data bus:** Data lines provide a path for moving data among the units; these lines are collectively called data bus. The number of lines refer to as the width of the data bus.
  - ➢**Address lines/Address bus:** Address lines are used to specify the source or destination of the data on the data bus. For example, if the processor wants to read a word of data from the memory, it puts the address of the data on the address bus. Address bus is also used to address the I/O ports.
  - ➢**Control lines/control bus:** The CPU transmits various control signals to the units through the control bus. Control lines control access to and the use of the data and address bus as these buses are shared by all units.

Bus Structure (contd.)

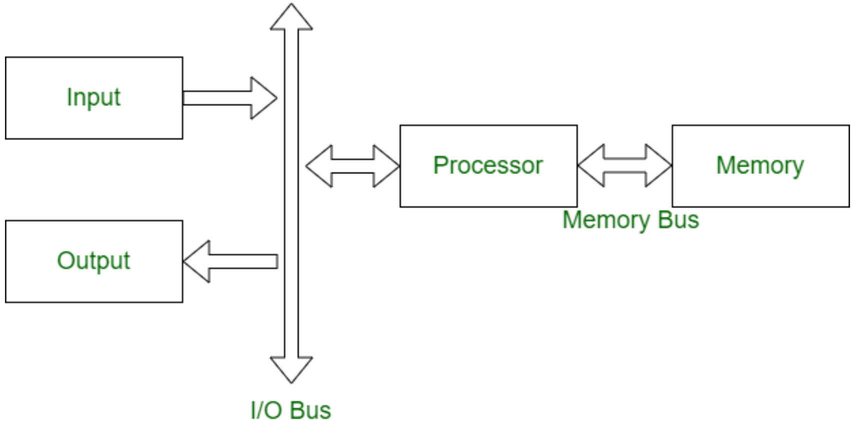
Control bus transmits both commands and timing information. Typically the control bus includes the following commands:

- ✓ Memory write: data on the data bus will be written on the addressed location.
- ✓ Memory read: data will be placed on the data bus from the addressed location.
- ✓ I/O write: data (output) on the data bus will be moved to the addressed I/O port.
- ✓ I/O read: data (input) from the addressed I/O port will be placed on the bus.
- ✓ Transfer ACK: indicates that data have been transferred to or from the bus.
- ✓ Bus request: indicates that a unit needs to get the control of a bus.
- ✓ Bus grant: indicates that a requested unit has got the control of the bus.
- ✓ Interrupt request: Indicates that an interrupt is pending.
- ✓ Interrupt ACK: Indicates that a pending interrupt has been recognized.
- ✓ Clock: being used to synchronize operations.
- ✓ Reset: initializes all the units.

❖ A bus may have additional lines to supply power to the attached units.

Double Bus Structure

- In this structure, one bus is used to fetch instruction while other is used to carry data. This overcomes the limitation of single bus structure.



Double Bus Structure

Performance

- How quickly a computer can execute programs is the most important measure of performance.
- It depends on the design of hardware and its machine language instruction sets.
- Depends on compiler also, as it converts the programs (written in high level languages) into machine language.
- Basic performance equation:

$$T = \frac{NxS}{R}$$

Here, T is the processor time required to execute a program

N is the number of machine language instruction executes that the program requires

S is the average number of steps required to execute a machine language instruction (each step is completed in one clock cycle)

Clock rate is R cycles per second

- Clock cycle: The speed of a computer processor, or CPU, is determined by the **Clock Cycle**, which is the amount of time between two pulses of an oscillator. The higher number of pulses per second, will faster the computer processor ability to process information. The clock speed is measured in Hz, typically either megahertz (MHz) or gigahertz (GHz). For example, a 4GHz processor performs 4,000,000,000 clock cycles per second. Computer processors can execute one or more instructions per clock cycle, depending on the type of processor.
- Clock rate: refers to the number of pulses per second generated by an oscillator that sets the rate of speed for the processor. Clock speed is usually measured in MHz (megahertz, or millions of pulses per second) or GHz (gigahertz, or billions of pulses per second).

Machine Instructions

- A computer program consists of a sequence of small steps such as adding two numbers, checking for a condition, read or write data etc.
- A computer must have the instructions capable of performing the tasks.
- Each family of processors has its own set of instructions for handling various operations. These set of instructions are called machine language instructions.
- A computer processor understands machine language instructions only which consist of strings of 0s and 1s.
- Machine language is too hard and complex to use. That is why, low level assembly language is designed for a specific family of processors that represents various machine instructions in symbolic code which is more understandable.

Assembly Language Notation and Instruction

- Memory location: LOC, PLACE, A, VAR.
- Processor register notation: R0, R1.
- I/O register: DATAIN, OUTSTATUS.
- Assembly language instructions: ADD, MOVE, STORE, LOAD, HALT.
- For example:

  ADD A,B adds the data stored in memory location A and B, and stores the result in B

  MOVE B,C moves the data from B to C (B source and C destination)

  ADD A adds the content of memory location A to the content of accumulator, and places the result back into the accumulator.

  Load A copies the content stored in A into the accumulator

  STORE A copies the content of the accumulator into A

Assembly Language Instruction

- Perform the following operation using one-address instruction: C=A+B.

Solution:
```
Solution: LOAD A
          ADD B
          STORE C
```

Instruction Execution

- The CPU executes instructions. The instruction execution takes place in CPU registers:
    - **Memory Address Register (MAR):** It specifies the address of memory location from which data or instruction is to be accessed (for read operation) or to which the data is to be stored (for write operation).
    - **Program Counter (PC):** holds the address of the instruction to be executed next, after the execution of an on-going instruction.
    - **Instruction Register (IR):** instructions are loaded/placed into IR before execution.

Steps to Execute an Instruction

- **Step 1: Fetch instruction**

  The instruction is fetched from the memory location whose address is in the PC. This instruction is placed in the IR.
- **Step 2: Decode instruction**

  Next, the encoded instruction is interpreted by the decoder to determine which operation is to be performed.
- **Step 3: Perform ALU operation**

  The specific operation is performed by the ALU.
- **Step 4: Access memory**

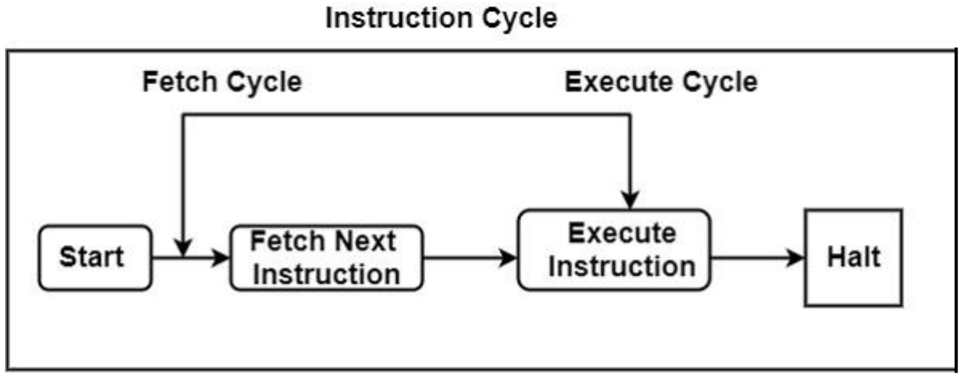
  During the operation, it requires to access the memory or the CPU register to get the operands.
- **Step 5: Store result**

  Stores the result in the destination location.
- **Step 6: Update the PC**

  At the end of the instruction execution, the PC is advanced/updated to hold the address of the next instruction to be executed.

Instruction Cycle

- The entire process of executing an instruction is known as instruction cycle.


Instruction Cycle
Fetch Cycle
Execute Cycle
Start
Fetch Next Instruction
Execute Instruction
Halt