Computer Organization and Architecture

UNIT 1

Introduction

Outline

- Digital Computer Generation
- Computer Types and Classification
- Functional Units and their Interconnections

Suggested Books

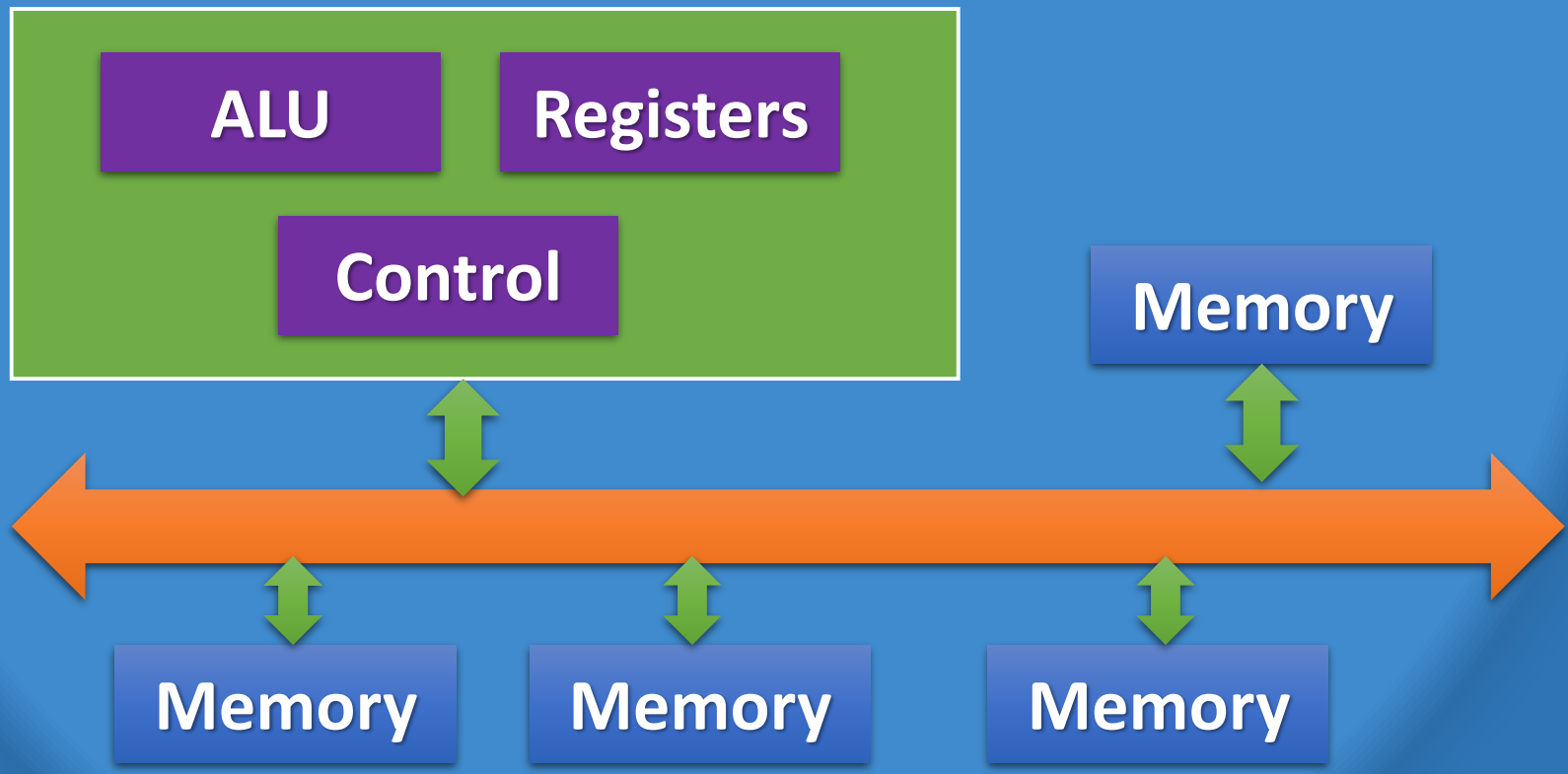
Patterson, Computer Organization and Design, Elsevier Pub,2009

Morris Mano, Computer System Architecture, PHI

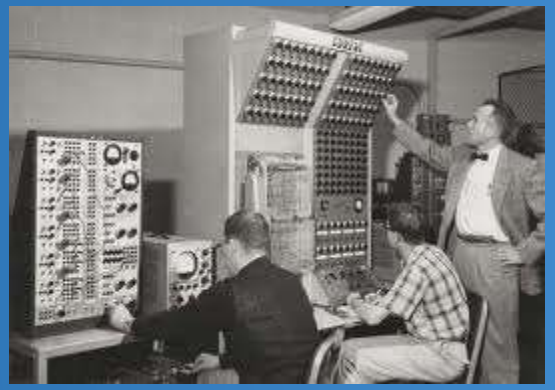
William Stalling, Computer Organization, PHI

BASIC COMPUTER ORGANISATION

Computer organisation refers to logical structure of a computer how its components are connected to one another



The Five Generations of Computers



Generations of Computer

- The computer has evolved from a large-sized simple calculating machine to a smaller but much more powerful machine.

- The evolution of computer to the current state is defined in terms of the generations of computer.

- Each generation of computer is designed based on a new technological development, resulting in better, cheaper and smaller computers that are more powerful, faster and efficient than their predecessors.

Generations of Computer

- Currently, there are five generations of computer. In the following subsections, we will discuss the generations of computer in terms of the technology used by them (hardware and software), computing characteristics (speed, i.e., number of instructions executed per second), physical appearance, and their applications.

First Generation Computers (1940-1956)

- The first computers used vacuum tubes(a sealed glass tube containing a near-vacuum which allows the free passage of electric current.) for circuitry and magnetic drums for memory.
- They were often enormous and taking up entire room.
- First generation computers relied on machine language.
- They were very expensive to operate and in addition to using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions(defect or breakdown).
- The UNIVAC and ENIAC computers are examples of first-generation computing devices.

First Generation Computers

Advantages :

- It was only electronic device
- First device to hold memory

Disadvantages :

- Too bulky i.e large in size
- Vacuum tubes burn frequently
- They were producing heat
- Maintenance problems

Some computers of this generation were:

- UNIVersal Automatic Computer (UNIVAC),
- Electronic NumericalIntegrator And Calculator (ENIAC),
- Electronic Discrete Variable Automatic Computer (EDVAC)
- IBM-701
- IBM-750

Second Generation Computers (1956-1963)

- Transistors replaced vacuum tubes and used in the second generation of computers.
- Second-generation computers moved from cryptic binary machine language to symbolic.
- High-level programming languages were also being developed at this time, such as early versions of COBOL and FORTRAN.
- These were also the first computers that stored their instructions in their memory.
- Magnetic core technology for primary memory.
- magnetic tapes and magnetic disks for secondary storage.

Second Generation Computers

Advantages :

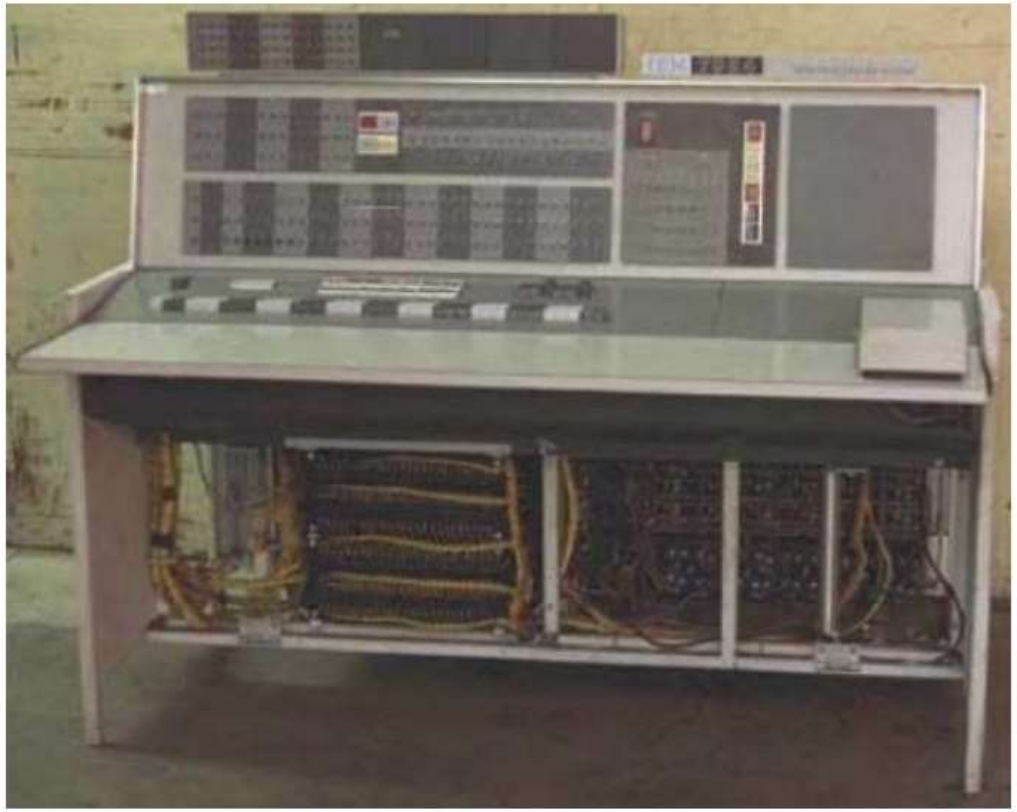
- Size reduced considerably
- The very fast
- Very much reliable

Disadvantages :

- They over heated quickly
- Maintenance problems

Some computers of this generation were:

- PDP-8 :12bit minicomputer, 22 march, 1965
- IBM 1401 : 6 bit plus work mark and parity, 1959
- CDC 1604: 48 bit , 1960



Third Generation Computers (1964-1971)

- The development of the integrated circuit was the hallmark of the third generation of computers.
- Transistors were miniaturized and placed on siliconchips, called semiconductors.
- Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system.
- Allowed the device to run many different applications at one time.

Third generation computers

Advantages :

- ICs are very small in size
- Improved performance
- Production cost cheap

Disadvantages :

- ICs are sophisticated

- **Examples**

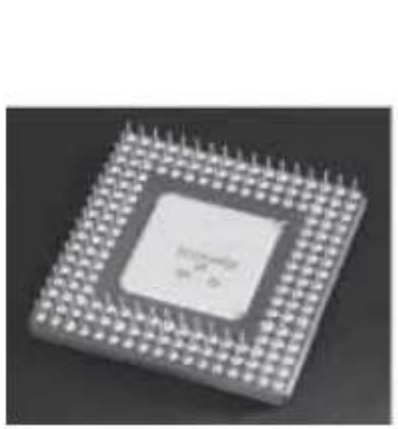
- IBM 370, PDP 11.

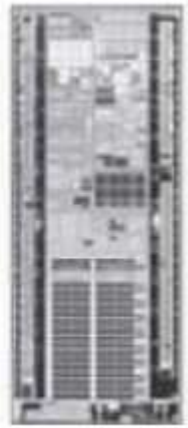
Fourth Generation Computers (1971-1981)

- The VLSI-microprocessor brought the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip.
- The Intel 4004 chip, developed in 1971, located all the components of the computer.
- From the central processing unit and memory to input/output controls—on a single chip.
- . Fourth generation computers also saw the development of GUIs, the mouse and handheld devices.

Fourth Generation Computers

Microprocessor





Examples

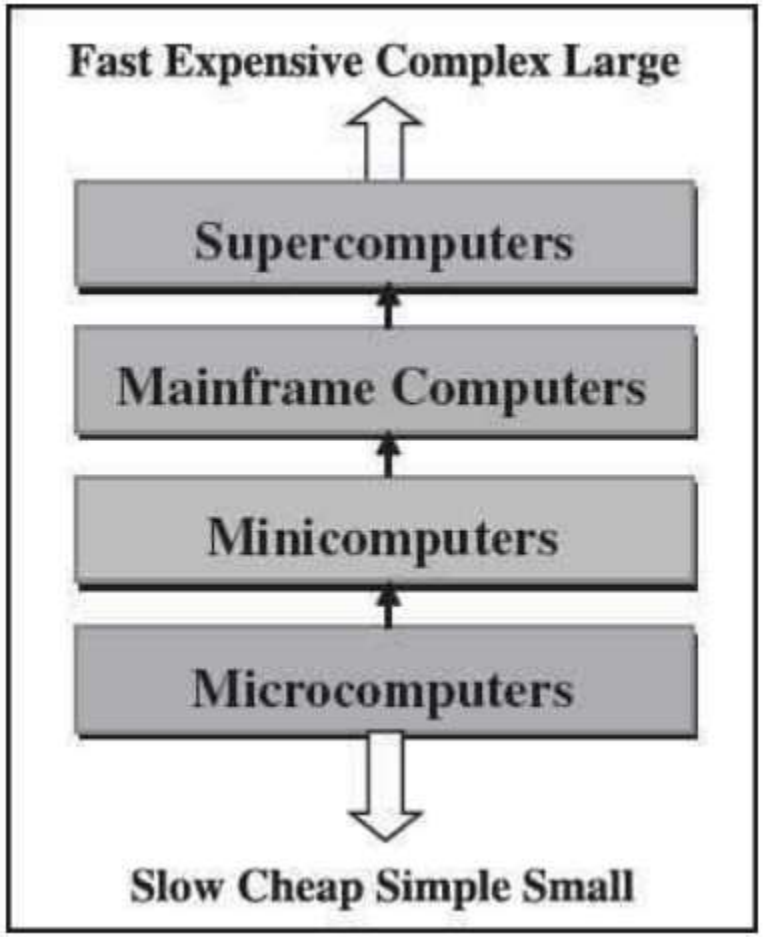
- The Intel 4004 chip was the first microprocessor.
- The components of the computer like Central Processing Unit (CPU) and memory were located on a single chip.
- In 1981, IBM introduced the first computer for home use.
- In 1984, Apple introduced the Macintosh.

Fifth Generation Computers (1980 - onwards)

- Fifth generation computing devices, based on artificial intelligence.
- ULSI microprocessor based
- Are still in development, though there are some applications, such as voice recognition.
- The use of parallel processing and superconductors is helping to make artificial intelligence a reality.
- The goal of fifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

Computer Types and Classifications

Classification of computers based on size and type



Computer Types and Classifications

According to size, cost computational power and application computers are classified as:

- Microcomputers
- Minicomputers
- Desktop computers
- Personal computers
- Portable notebook computers
- Workstations
- Mainframes or enterprise system
- Serves
- Super computers

Microcomputers

Smaller computers which contain only one Central Processing Unit

Also known as personal computers and are the ones mostly found in big and small office

They are normally standalone computers known PC, or Desktop Computers.

Micro Computers are small and expensive designed for individual use.

It contains two types of memories RAM and ROM.

Minicomputers

Midsized computers capable of supporting from 4 – 200 users simultaneously

Moderate speed and storage capacity

Mainly used as departmental computers for data processing in large organization or governmental institutions like hospitals

Scientific calculations, research, data processing application

Desktop computers

Usually found on a home or office desk.

Which consist of processing unit, storage unit, visual display and audio as output units, and keyboard and mouse as input units.

Usually storage unit of such computer consists of hard disks, CD-ROMs etc

Personal computers

The personal computers are the most common form of desktop computers.

They found wide use in homes, schools and business offices.

Portable Notebook Computers

Portable notebook computers are the compact version of personal computers.

The laptop computers are the good example of portable notebook computer

Workstations

Workstations have higher computation power than personal computers.

They have high resolution graphics terminals and improved input/output capabilities.

Workstations are used in engineering applications and in interactive graphics applications.

Mainframes or Enterprise System

Implemented using two or more central processing units (CPU).

These are designed to work at very high speeds with large data word lengths, typically 64 bits or greater.

The data storage capacity of these computers is very high.

This type of computers are used for complex scientific calculations, large data processing

Example: For creating walkthroughs with the help of animation softwares

Servers

These computers have large storage unit and faster communication links.

The large storage unit allows to store sizable database and fast communication links allow faster communication of data blocks with computers connected in the network.

These computers serve major role in internet communication

Supercomputers

Super Computers are the fastest computer which are very expensive and requires a lot of mathematical calculations.

These computers are basically multiprocessor computers

Used for the large-scale numerical calculations required in applications such as weather forecasting, robotic engineering, aircraft design simulation.

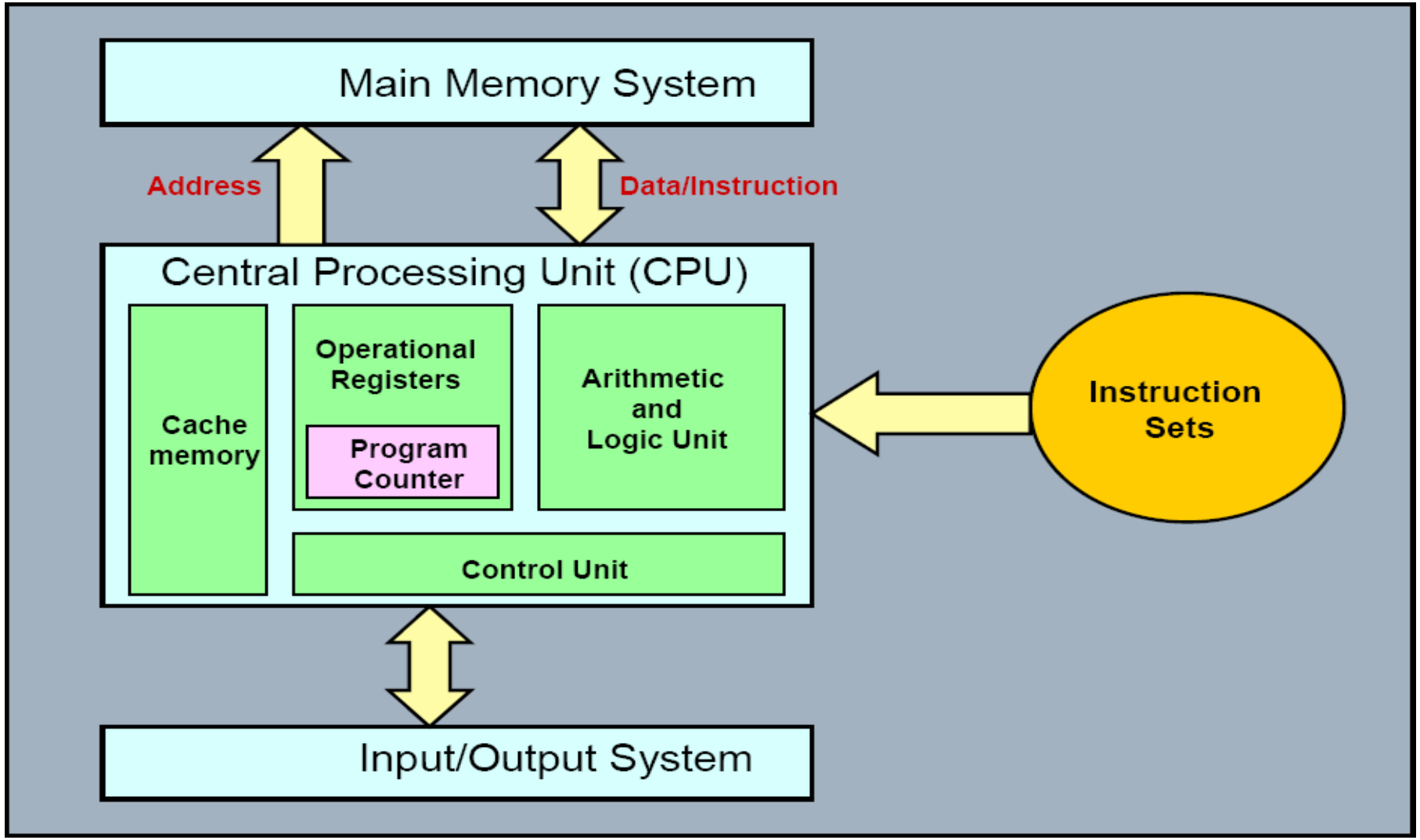
The speed of a supercomputer is generally measured in FLOPS (FLoating point Operations Per Second).

Some of the faster supercomputers can perform trillions of calculations per second.

Supercomputers are built by interconnecting thousands of processors that can work in parallel.

Supercomputer:

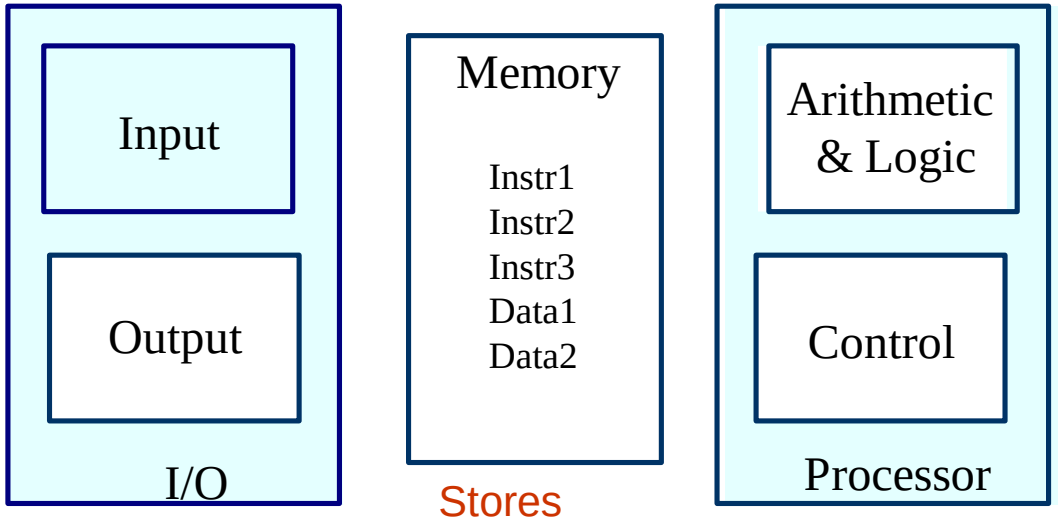
Functional Units and Their Interconnections



Functional units of a computer

Input unit accepts information:
- Human operators,
- Electromechanical devices (keyboard)
- Other computers

Arithmetic and logic unit(ALU):
- Performs the desired operations on the input information as determined by instructions in the memory



Output unit sends results of processing:
- To a monitor display,
- To a printer

Stores information:
- Instructions,
- Data

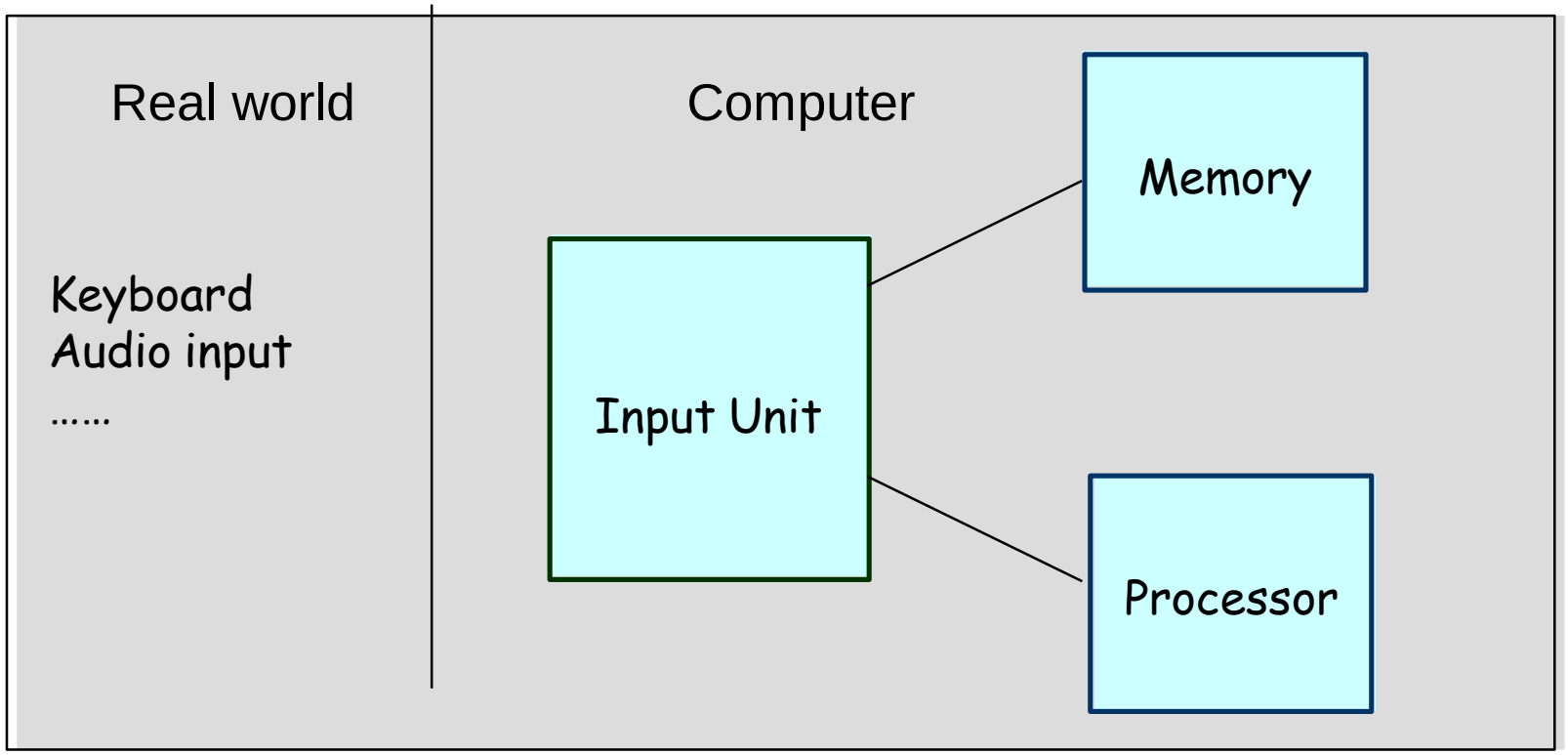
Control unit coordinates various actions
- Input,
- Output
- Processing

- Instructions specify commands to:
  - Transfer information within a computer (e.g., from memory to ALU)
  - Transfer of information between the computer and I/O devices (e.g., from keyboard to computer, or computer to printer)
  - Perform arithmetic and logic operations (e.g., Add two numbers, Perform a logical AND).
- A sequence of instructions to perform a task is called a program, which is stored in the memory.
- Processor fetches instructions that make up a program from the memory and performs the operations stated in those instructions.
- What do the instructions operate upon?
- Data are the "operands" upon which instructions operate.
- Data could be:
  - Numbers,
  - Encoded characters.
- Data, in a broad sense means any digital information.
- Computers use data that is encoded as a string of binary digits called bits.

Input unit

Binary information must be presented to a computer in a specific format. This task is performed by the input unit:

- Interfaces with input devices.
- Accepts binary information from the input devices.
- Presents this binary information in a format expected by the computer.
- Transfers this information to the memory or processor.



Memory Unit

- Memory unit stores instructions and data.
  - Recall, data is represented as a series of bits.
  - To store data, memory unit thus stores bits.
- Processor reads instructions and reads/writes data from/to the memory during the execution of a program.
  - In theory, instructions and data could be fetched one bit at a time.
  - In practice, a group of bits is fetched at a time.
  - Group of bits stored or retrieved at a time is termed as “word”
  - Number of bits in a word is termed as the “word length” of a computer.
- In order to read/write to and from memory, a processor should know where to look:
  - “Address” is associated with each word location.

Memory unit (contd..)

- Processor reads/writes to/from memory based on the memory address:
  - Access any word location in a short and fixed amount of time based on the address.
  - Random Access Memory (RAM) provides fixed access time independent of the location of the word.
  - Access time is known as "Memory Access Time".
- Memory and processor have to "communicate" with each other in order to read/write information.
  - In order to reduce "communication time", a small amount of RAM (known as Cache) is tightly coupled with the processor.
- Modern computers have three to four levels of RAM units with different speeds and sizes:
  - Fastest, smallest known as Cache
  - Slowest, largest known as Main memory.

Memory unit (contd..)

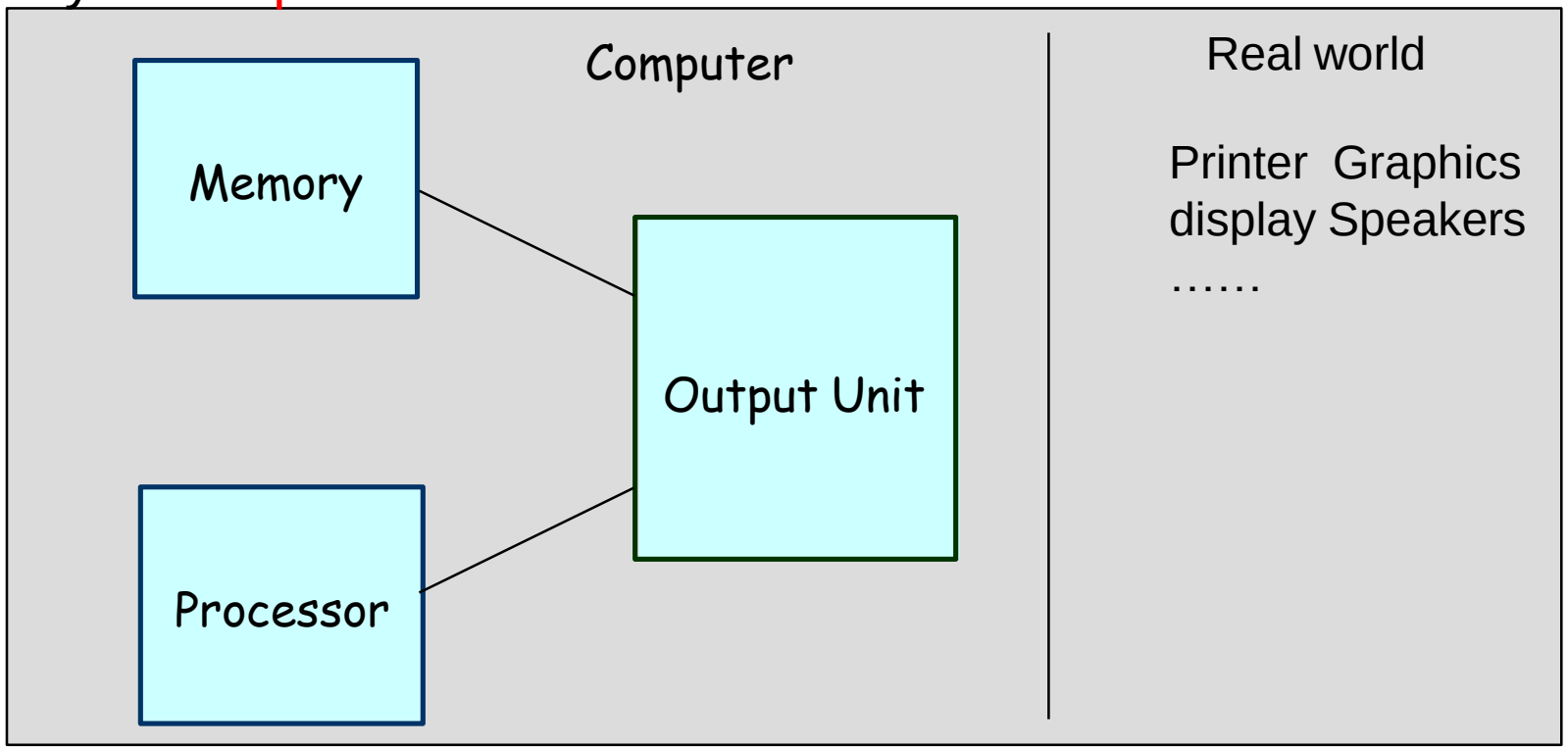
- Primary storage of the computer consists of RAM units.
  - Fastest, smallest unit is Cache.
  - Slowest, largest unit is Main Memory.
- Primary storage is insufficient to store large amounts of data and programs.
  - Primary storage can be added, but it is expensive.
- Store large amounts of data on secondary storage devices:
  - Magnetic disks and tapes,
  - Optical disks (CD-ROMS).
  - Access to the data stored in secondary storage in slower, but take advantage of the fact that some information may be accessed infrequently.
- Cost of a memory unit depends on its access time, lesser access time implies higher cost

Arithmetic and logic unit (ALU)

- Operations are executed in the Arithmetic and Logic Unit (ALU).
  - Arithmetic operations such as addition, subtraction.
  - Logic operations such as comparison of numbers.
- In order to execute an instruction, operands need to be brought into the ALU from the memory.
  - Operands are stored in general purpose registers available in the ALU.
  - Access times of general purpose registers are faster than the cache.
- Results of the operations are stored back in the memory or retained in the processor for immediate use.

Output Unit

- Computers represent information in a specific binary form.
- Output units:
  - Interface with output devices.
  - Accept processed results provided by the computer in specific binary form.
  - Convert the information in binary form to a form understood by an output device.



Control unit

- Operation of a computer can be summarized as:
  - Accepts information from the input units (Input unit).
  - Stores the information (Memory).
  - Processes the information (ALU).
  - Provides processed results through the output units (Output unit).
- Operations of Input unit, Memory, ALU and Output unit are coordinated by Control unit.
- Instructions control "what" operations take place (e.g. data transfer, processing).
- Control unit generates timing signals which determines "when" a particular operation takes place.

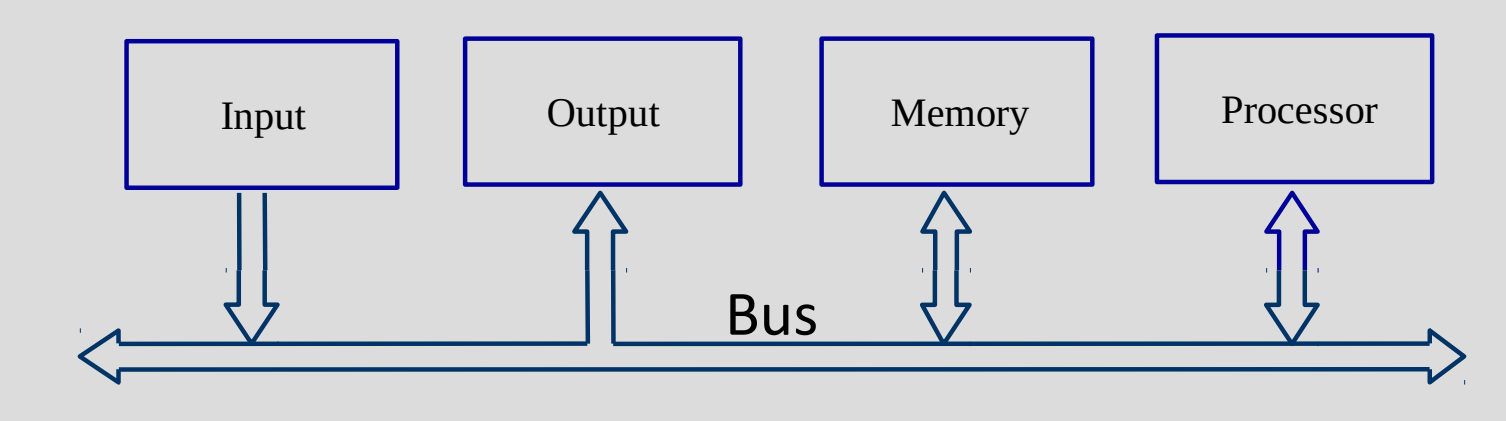
A Typical Instruction

❑Add LOCA, R0

- ❖ Add the operand at memory location LOCA to the operand in a register R0 in the processor.
- ❖ Place the sum into register R0.
- ❖ The original contents of LOCA are preserved.
- ❖ The original contents of R0 is overwritten.
- ❖ Instruction is fetched from the memory into the processor
  – the operand at LOCA is fetched and added to the contents of R0 – the resulting sum is stored in register R0.

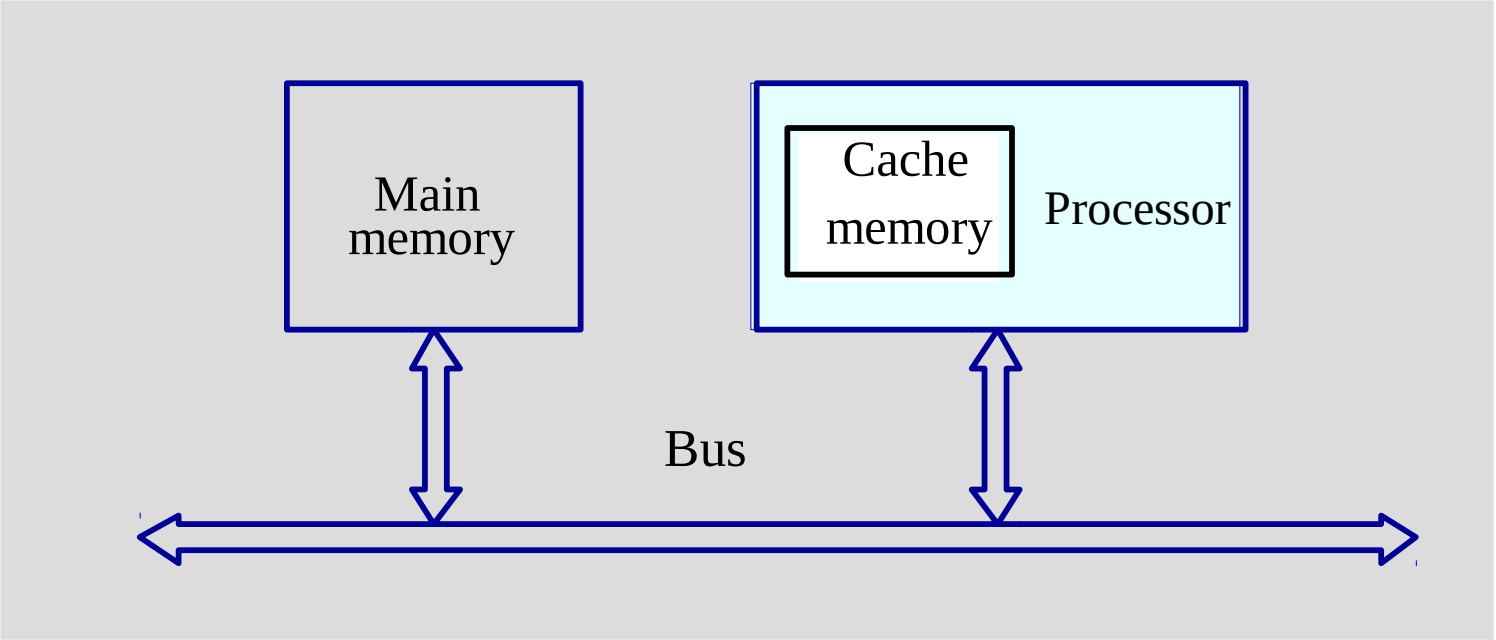
How are the functional units connected?

- For a computer to achieve its operation, the functional units need to communicate with each other.
- In order to communicate, they need to be connected.



- Functional units may be connected by a group of parallel wires.
- The group of parallel wires is called a ***bus***.
- Each wire in a bus can transfer one bit of information.
- The number of parallel wires in a bus is equal to the word length of a computer

Organization of cache and main memory



Why is the access time of the cache memory lesser than the access time of the main memory?

Registers

- **In addition to the ALU and the control circuit, the processor contains number of registers used for several different purposes.**

Instruction register (IR)

- It holds the instruction that is currently being executed.
- Its output is available to the control circuits which generates the timings signals that control the various processing elements involved in executing the instruction.

Program counter (PC)

- PC is another specialized register.
- It keeps track of the execution of a program. It contains the memory address of the next instruction to be fetched and executed.
- During the execution of an instruction, the contents of the PC updated to correspond to the address of the next instruction to executed.
- PC *points* to the next instruction that is to be fetched from the memory.

Registers (Cont'd)

- Two registers facilitate communication with the memory

❖ Memory address register (MAR)

- Holds the address of the memory location to be accessed.

❖ Memory data register (MDR)

- MDR contains the data to be written into or read out of the addressed location.

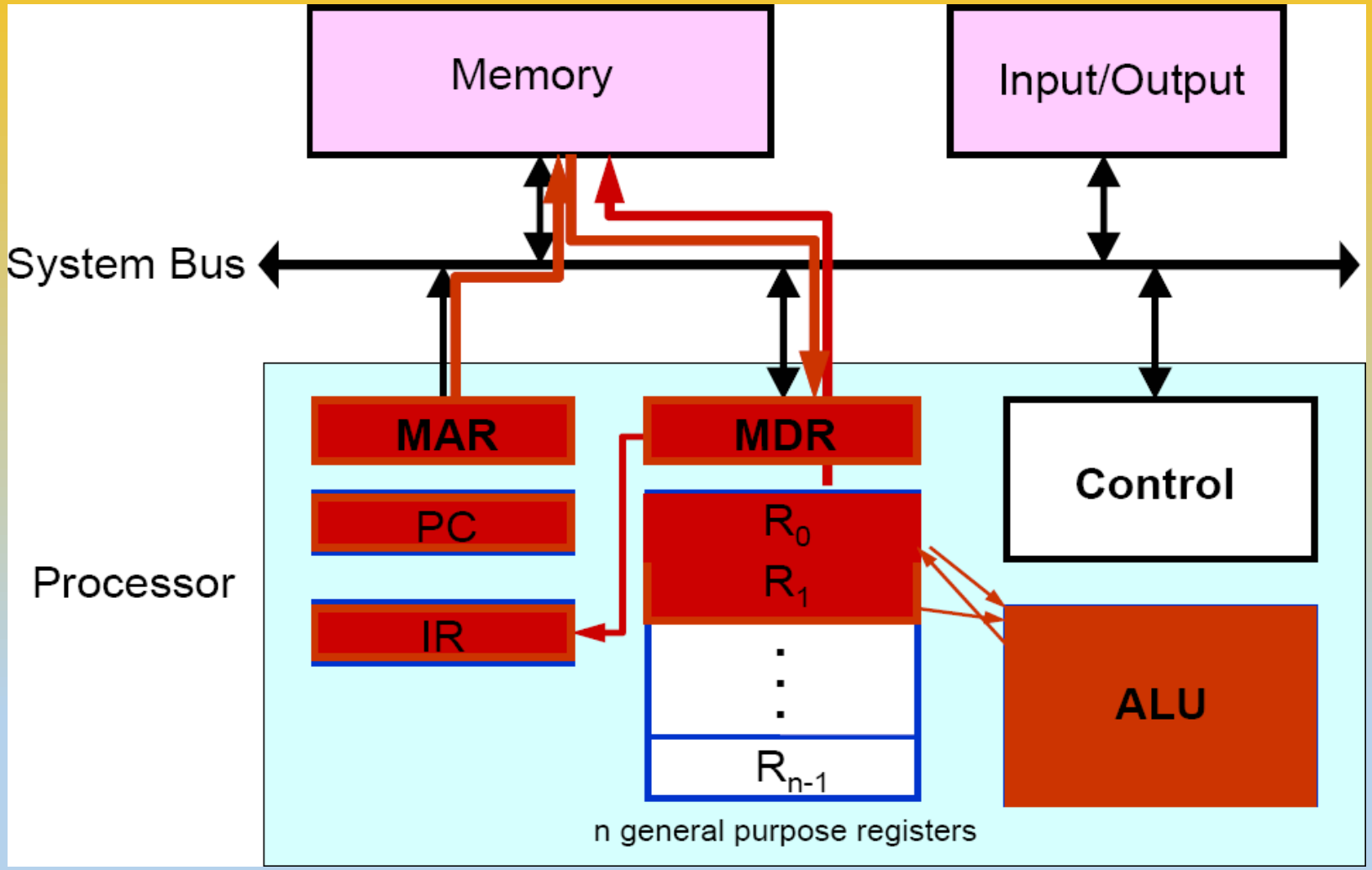
❖ General-purpose register ($R_0 – R_{n-1}$)

❖ Memory Buffer Register (MBR) :

- Stores information that is being sent to, or received from, the memory along the bidirectional data bus.

❖ Accumulator (AC): AC is used to store data that is being worked on by the ALU, and is the key register in the data section of the cpu. Notice that the memory can't access the AC directly. The MBR is an intermediary.

Basic Operational Concepts

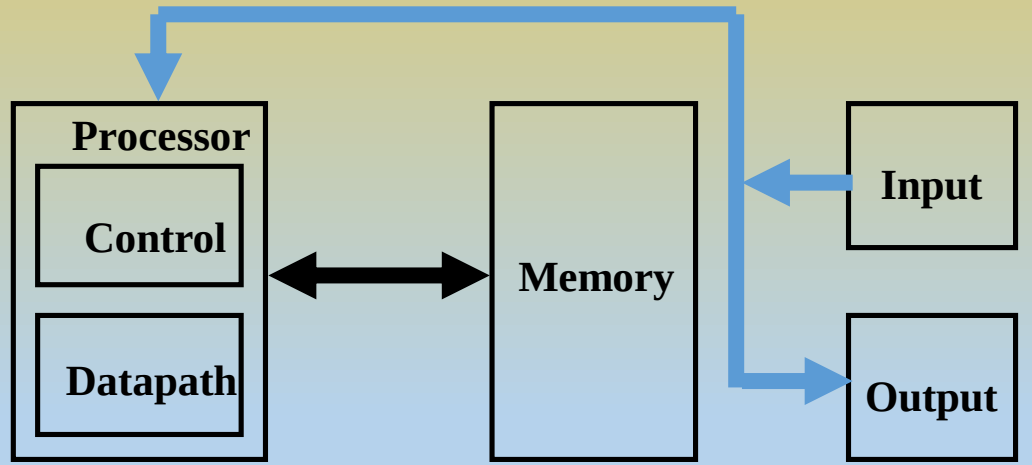


Typical Operating Steps

- Programs reside in the memory through input devices
- PC is set to point to the first instruction
- The contents of PC are transferred to MAR
- A Read signal is sent to the memory
- The first instruction is read out and loaded into MDR
- The contents of MDR are transferred to IR
- Decode and execute the instruction
  - Get operands for ALU
    - General-purpose register
    - Memory (address to MAR – Read – MDR to ALU)
- Perform operation in ALU
- Store the result back
    - To general-purpose register
    - To memory (address to MAR, result to MDR – Write)
- During the execution, PC is incremented to the next instruction

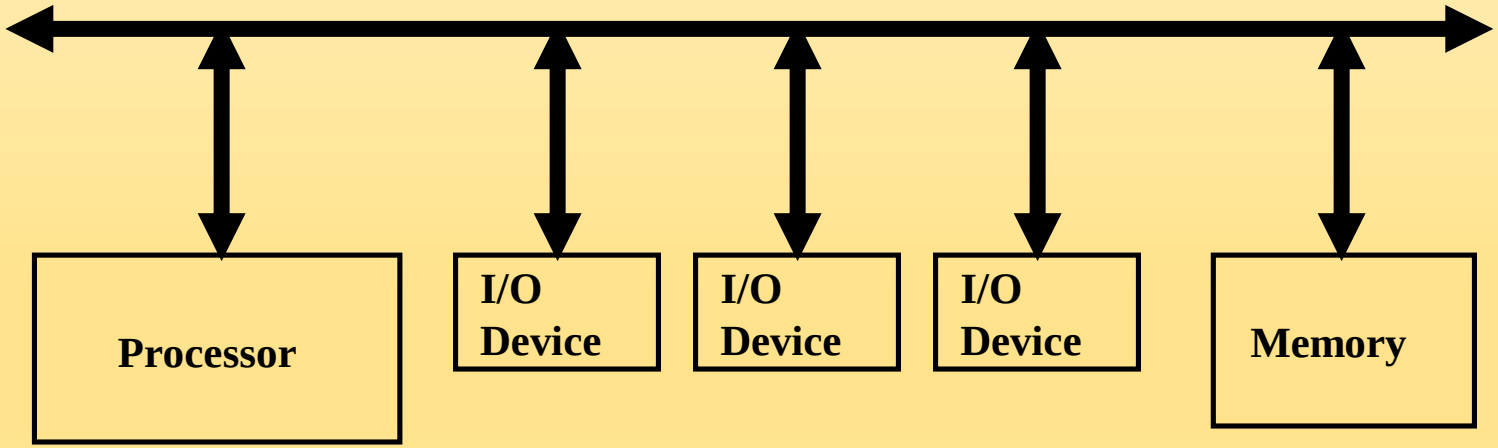
What is a Bus?

- A shared communication link
- A single set of wires used to connect two or more components
  - Unlike a typical wire in a processor, a bus can communicate in different directions at different times.



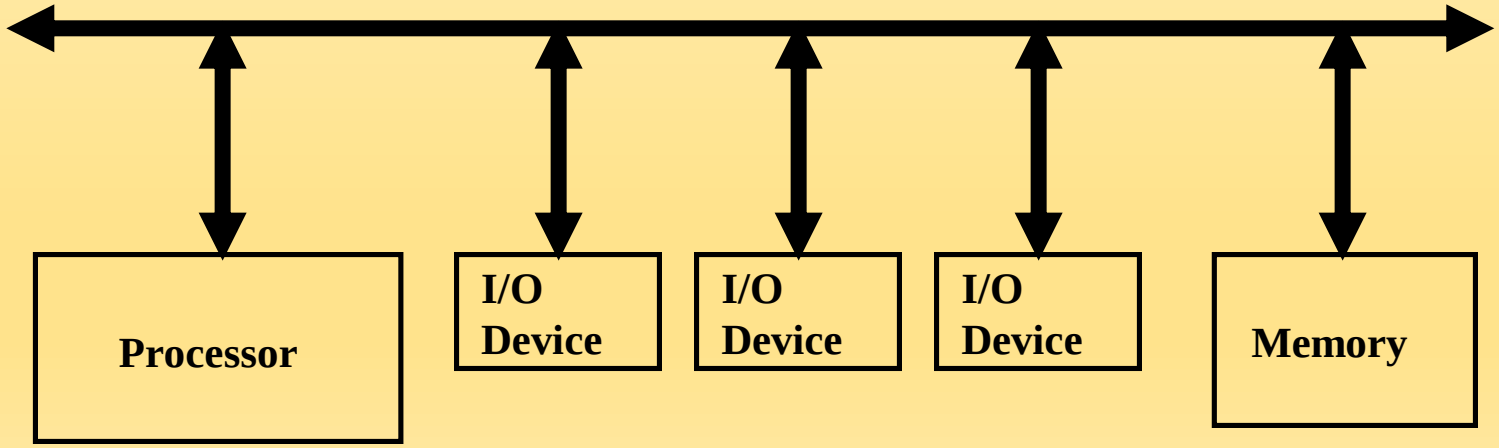
Advantages of Buses

- Versatility:
  - New devices can be added easily
  - Peripheral devices can be moved between computer systems that use the same bus standard
- Low Cost:
  - A single set of wires is shared in multiple ways
- Provides a way to manage the complexity of design
  - Device only has to implement the bus standard.

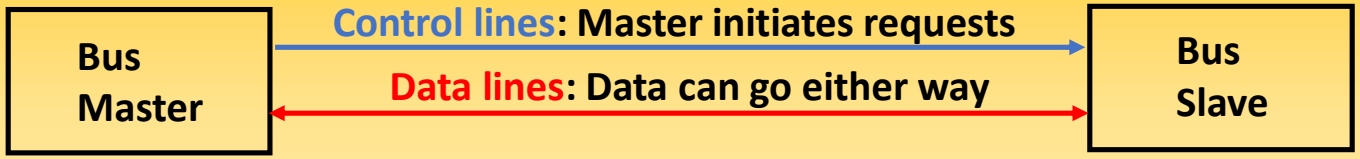


Disadvantage of Buses

- It creates a communication bottleneck
  - Bus bandwidth can limit the maximum I/O throughput
- The maximum bus speed is largely limited by:
  - The length of the bus
  - The number of devices on the bus
  - The need to support a range of devices with:
    - Widely varying latencies
    - Widely varying data transfer rates



Bus Characteristics



- **Control lines**
  - Signal requests and acknowledgments
  - Indicate what type of information is on the data lines
- **Data lines:** Carry information between the source and the destination
  - Data, addresses, and complex commands
- Bus transaction consists of
  - Master issuing the command (and address) – request
  - Slave receiving (or sending) the data – action
  - Defined by what the transaction does to memory
    - Input – inputs data from the I/O device to the memory
    - Output – outputs data from the memory to the I/O device

Bus Structures

- A group of wires, called **bus** is used to provide necessary signals for communication between modules.
- A bus that connects major computer components/ modules (CPU, memory I/O) is called a **system bus.**
- The system bus is a set of conductors that connects the CPU, memory and I/O modules.
- The system bus is separated into three functional groups:
- Data Bus
- Address Bus
- Control Bus

Data Bus

- The data bus consists of 8, 16, 32 or more parallel signal lines.
- Carry data from one component to another component in a computer.
- The data bus lines are bi-directional i.e. CPU can read data on these lines from memory or from a ports, as well as send data out on these lines to a memory location or to a port.
- The data bus is connected in parallel to all peripherals.
- The communication between peripheral and CPU is activated by giving output enable pulse to the peripheral.
- Outputs of peripherals are floated when they are not in use.

Address Bus

- It is an unidirectional bus .
- The address bus consists of 16, 20, 24 or more parallel signal lines.
- On these lines the CPU sends out the address of the memory location or I/O port that is to be written to or read from.
- It will help to know about the memory location of different components in the computer.

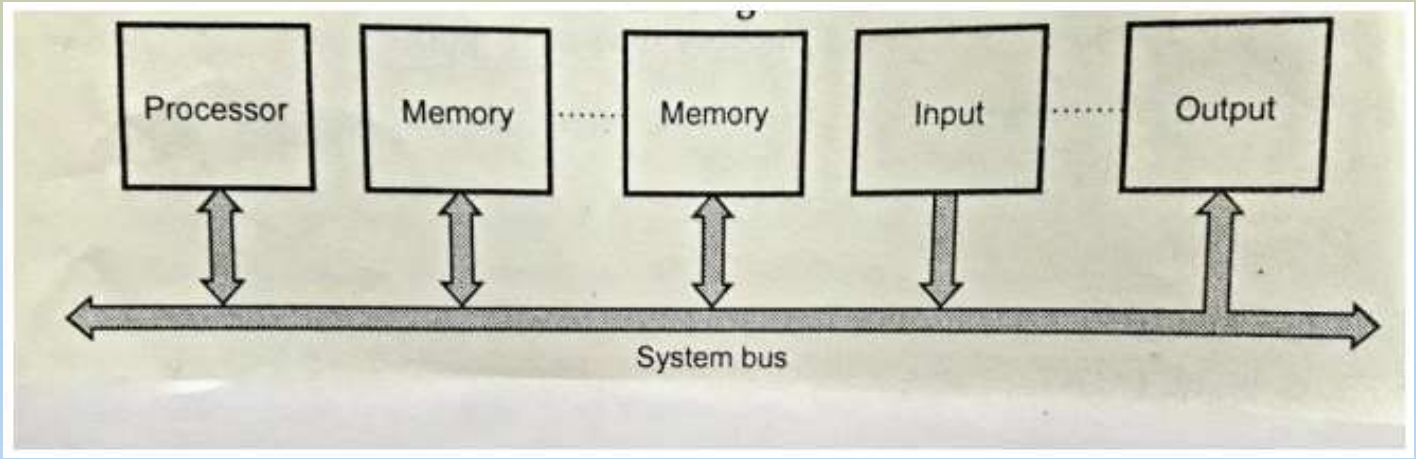
Control Bus

It is also known as **command bus** which **transmits control signals**. CPU consist of a control unit which this unidirectional bus to control other components of the computer. Typical control bus signals are:

- Memory Read (MEMR)
- Memory Write (MEMW)
- I/O Read (IOR)
- I/O Write (IOW)
- Bus Request (BR)
- Bus Grant (BG)
- Interrupt Request (INTR)
- Interrupt Acknowledge (INTA)
- Clock (CLK)
- Reset
- Ready
- Hold
- Hold Acknowledge (HLDA)

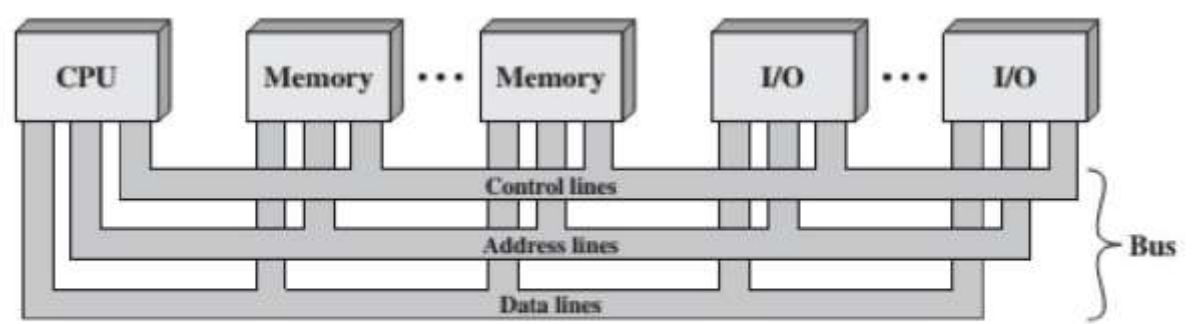
Single Bus Structure

- All units are connected to common bus called system bus.
- With single bus only two units can communicate with each other at a time.
- There is no messaging engine on Single bus structure.
- The complexity of bus control logic depends on the amount of translation needed between the system bus and CPU, the timing requirements



Multiple Bus Structures

- The performance of computer system suffers when large numbers of devices are connected to the bus.
- The sharing mechanism coordinates the use of bus to different devices which requires finite time called propagation delay.
- When control of the bus passes from one device to another frequently, these propagation delays are noticeable and affect the performance of computer system

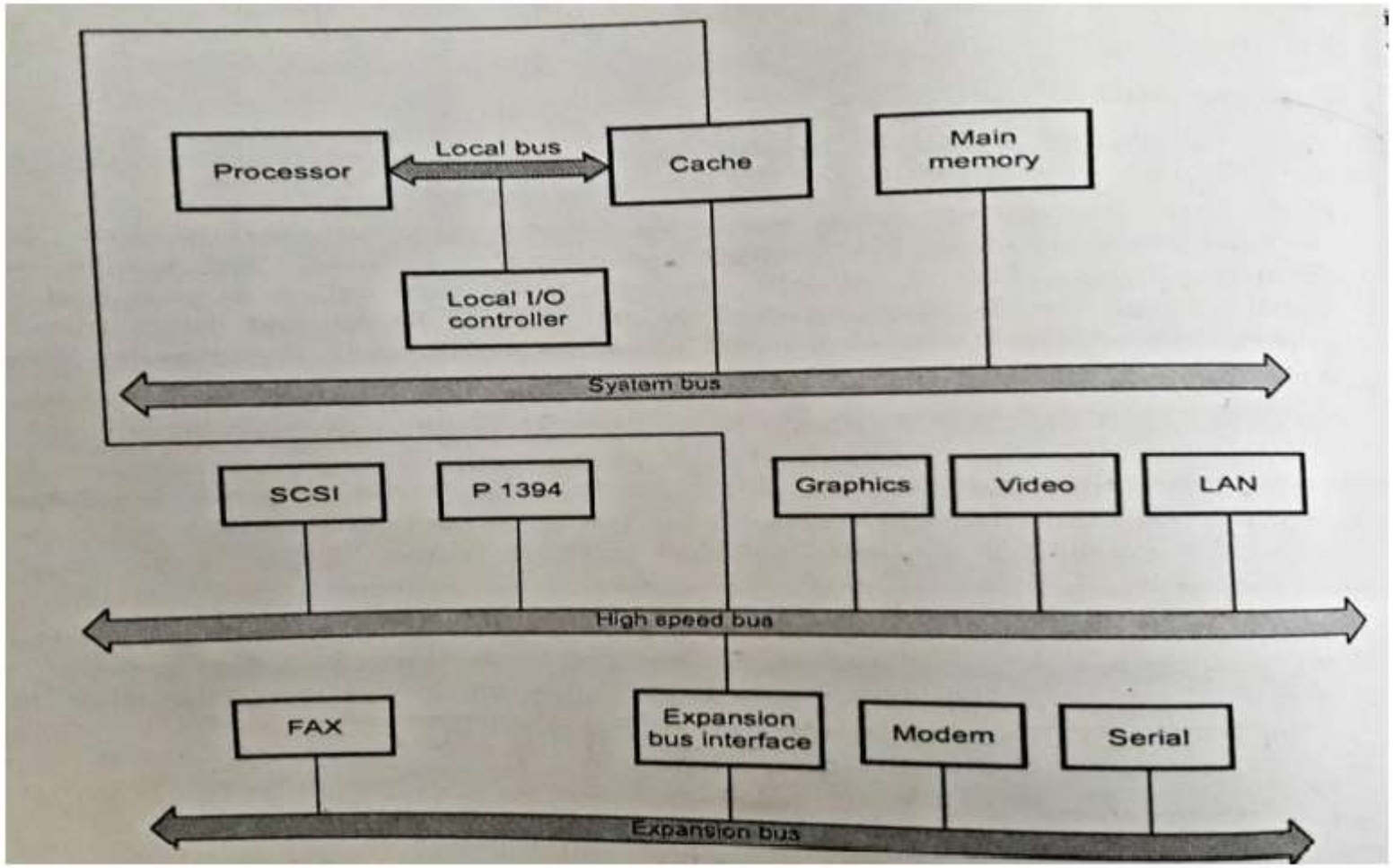


Multi-Bus Interconnection

High-speed Bus Configuration

- The high speed bus configuration uses high-speed bus along with the three buses used in the traditional bus connection i.e. local bus, system bus and expanded bus.
- Cache controller is connected to high-speed bus.
- This bus supports connection high-speed LANs, such as Fiber Distributed Data Interface (FDDI), video and graphics workstation controllers, as well as interface controllers to local peripheral buses including SCSI (Small Computer System Interface) and P1394 (IEEE standard, high speed, low cost serial bus)/Fire Wire.

High-speed Bus Configuration

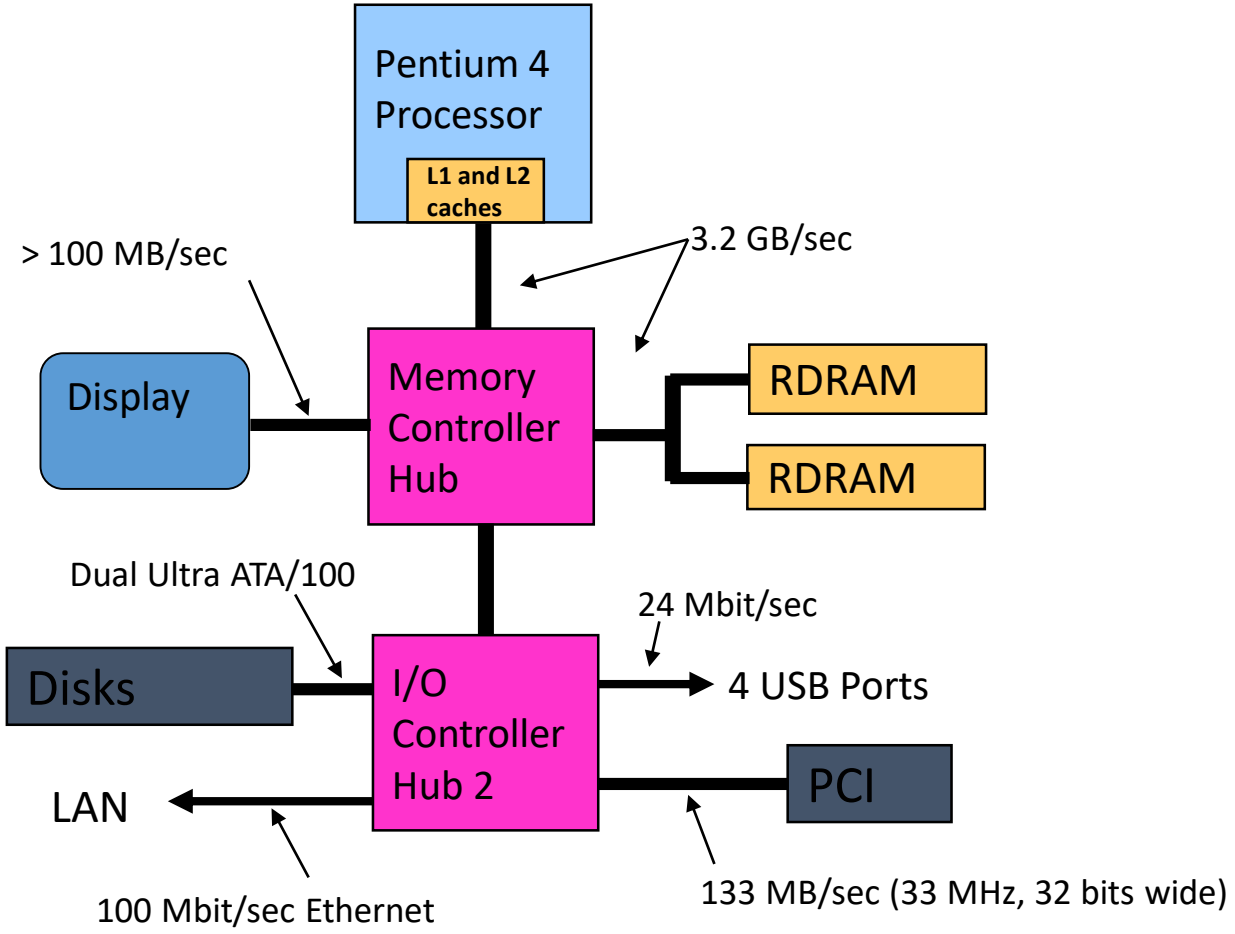


High-speed Bus Configuration

Bus Design Parameters

- Type of Bus : Dedicated (permanently assigned to one function) or multiplexed
- Method of Arbitration : Centralized or distributed
- Timing : Synchronous or asynchronous
- Bus Width : Address or Data
- Data Transfer Type: Read, write, read modify write, read-after write or block

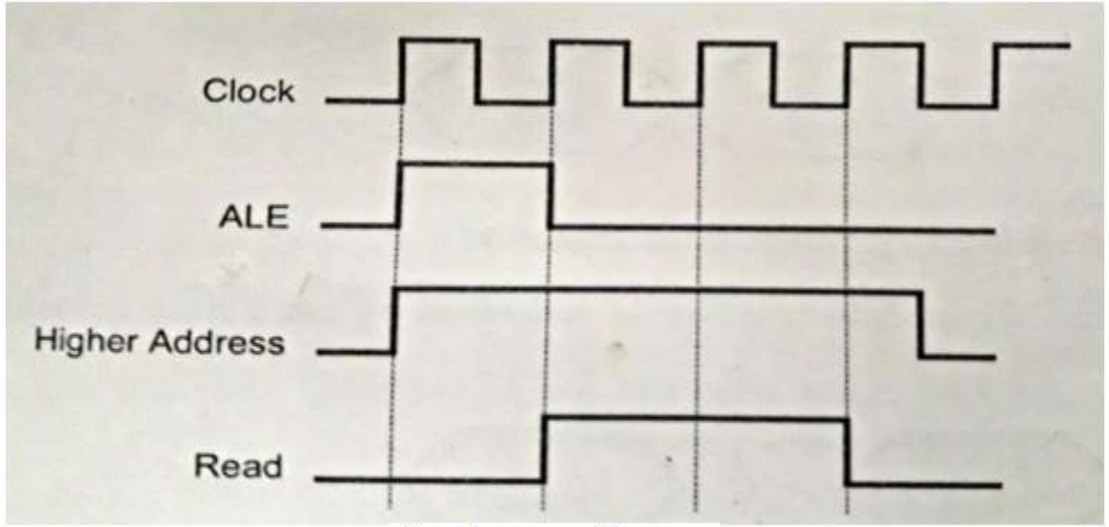
Example: Pentium 4 system



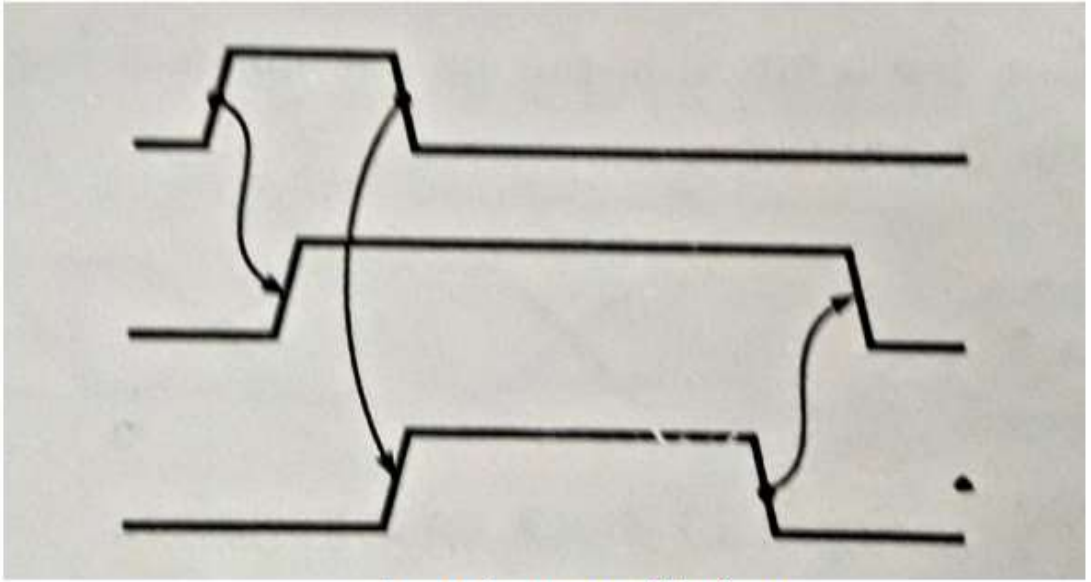
Synchronous vs Asynchronous Bus

- Synchronous Bus:
  - Control and data signals move with respect to a common clock (usually one of the bus control lines)
  - Advantage: low latency & high bandwidth.
  - Disadvantages:
    - Every device on the bus must run at the same clock rate
    - To avoid clock skew, bus cannot be long if it is fast
- Asynchronous Bus:
  - It is not clocked
  - It can accommodate a wide range of devices
  - It can be lengthened without worrying about clock skew
  - Communication protocol is more complicated

In synchronous timing, the occurrence of events on the bus is determined by the clock whereas in the asynchronous timing, the occurrence of events on the bus is determined by the previous events on the bus.



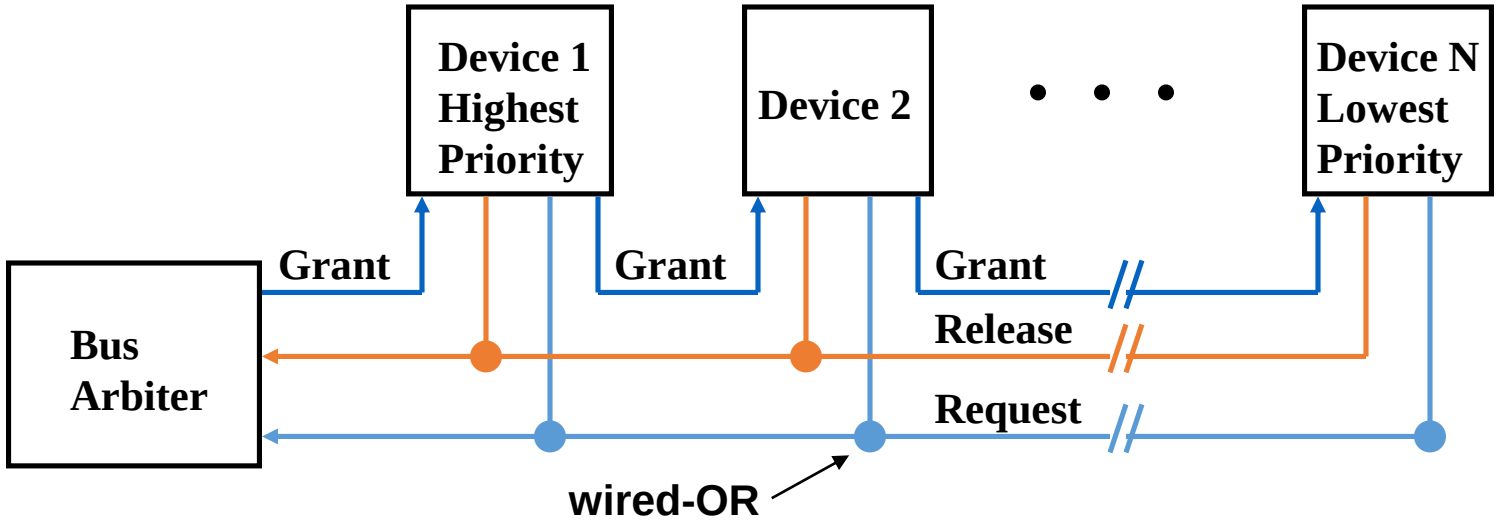
Synchronous Timings


Asynchronous Timings

Bus Arbitration: choosing a master

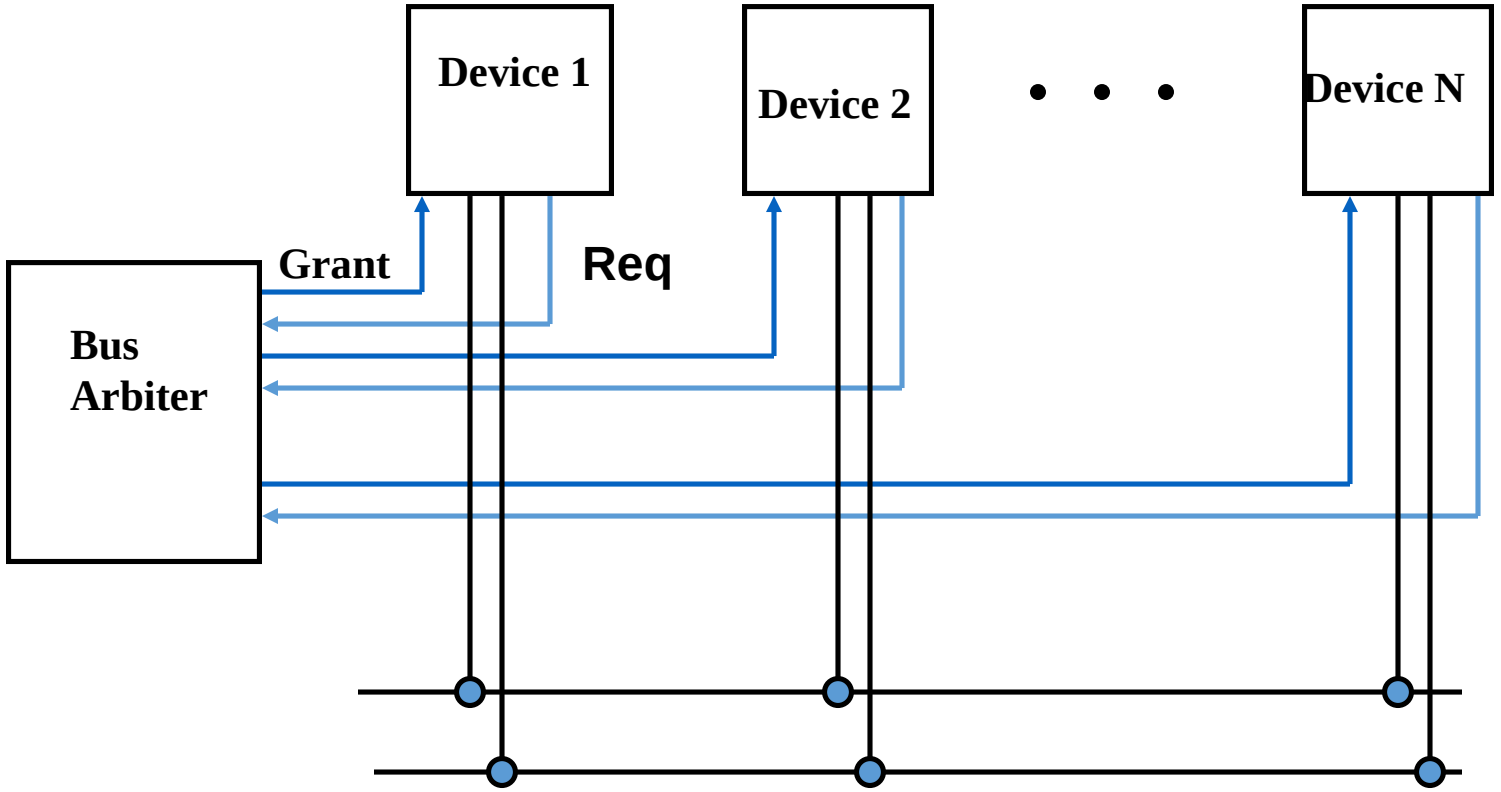
- Bus arbitration schemes must to balance 2 factors:
  - Bus priority: highest priority device should be serviced first
  - Fairness: even the lowest priority device should never be completely locked out from the bus
- Bus arbitration schemes can be divided into four broad classes:
  - Daisy chain arbitration
  - Centralized arbitration
  - Distributed arbitration by self-selection:
    - each device wanting the bus places an id code on the bus.
    - they all follow the same protocol to choose master
  - Distributed arbitration by collision detection (e.g. Ethernet)

Daisy Chain Bus Arbitration



- Advantage: simple
- Disadvantages:
  - Doesn't assure fairness:
    A low-priority device may be locked out indefinitely
  - The daisy chain grant signal also limits the bus speed

Centralized Arbitration



- Used in essentially all processor-memory busses and in high-speed I/O busses

Register Transfer Language

BASIC DEFINITIONS

- A digital system is an interconnection of digital hardware modules i.e. registers, decoders, arithmetic elements, and control logic.
- The various modules are interconnected with common data and control paths to form a digital computer system.
- Digital modules are best defined by the registers they contain and the operations that are performed on the data stored in them.
- The operations executed on data stored in registers are called microoperations.
- A microoperation is an elementary operation performed on the information stored in one or more registers.
- The result of the operation may replace the previous binary information of a register or may be transferred to another register.
- Examples of microoperations are shift, count, clear, and load.

Register Transfer Language

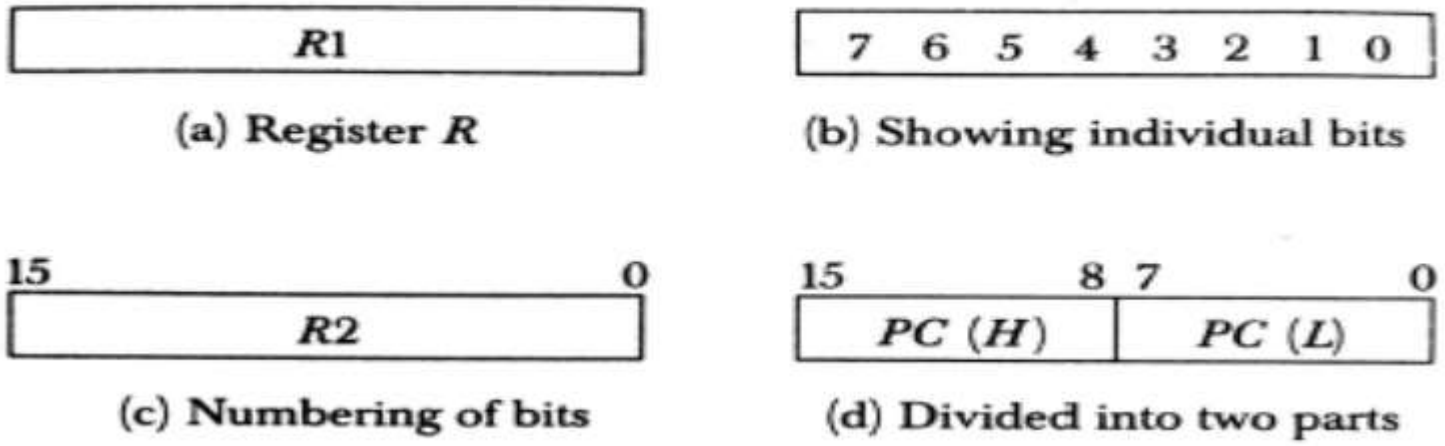
- The symbolic notation used to describe the micro-operation transfer among registers is called RTL (Register Transfer Language).
- The use of symbols instead of a narrative explanation provides an organized and concise manner for listing the micro-operation sequences in registers and the control functions that initiate them.
- A register transfer language is a system for expressing in symbolic form the microoperation sequences among the registers of a digital module.
- It is a convenient tool for describing the internal organization of digital computers in concise and precise manner

Registers

- Computer registers are designated by upper case letters (and optionally followed by digits or letters) to denote the function of the register.
- For example, the register that holds an address for the memory unit is usually called a memory address register and is designated by the name MAR.
- Other designations for registers are PC (for program counter), IR (for instruction register, and R1 (for processor register).
- The individual flip-flops in an n-bit register are numbered in sequence from 0 through n-1, starting from 0 in the rightmost position and increasing the numbers toward the left.

- The most common way to represent a register is by a rectangular box with the name of the register inside, as in Fig.(a).
- The individual bits can be distinguished as in (b).
- The numbering of bits in a 16-bit register can be marked on top of the box as shown in (c).
- 16-bit register is partitioned into two parts in (d). Bits 0 through 7 are assigned the symbol L (for low byte) and bits 8 through 15 are assigned the symbol H (for high byte).
- The name of the 16-bit register is PC. The symbol PC (0-7) or PC (L) refers to the low-order byte and PC (8-15) or PC (H) to the high-order byte.

Block diagram of register.



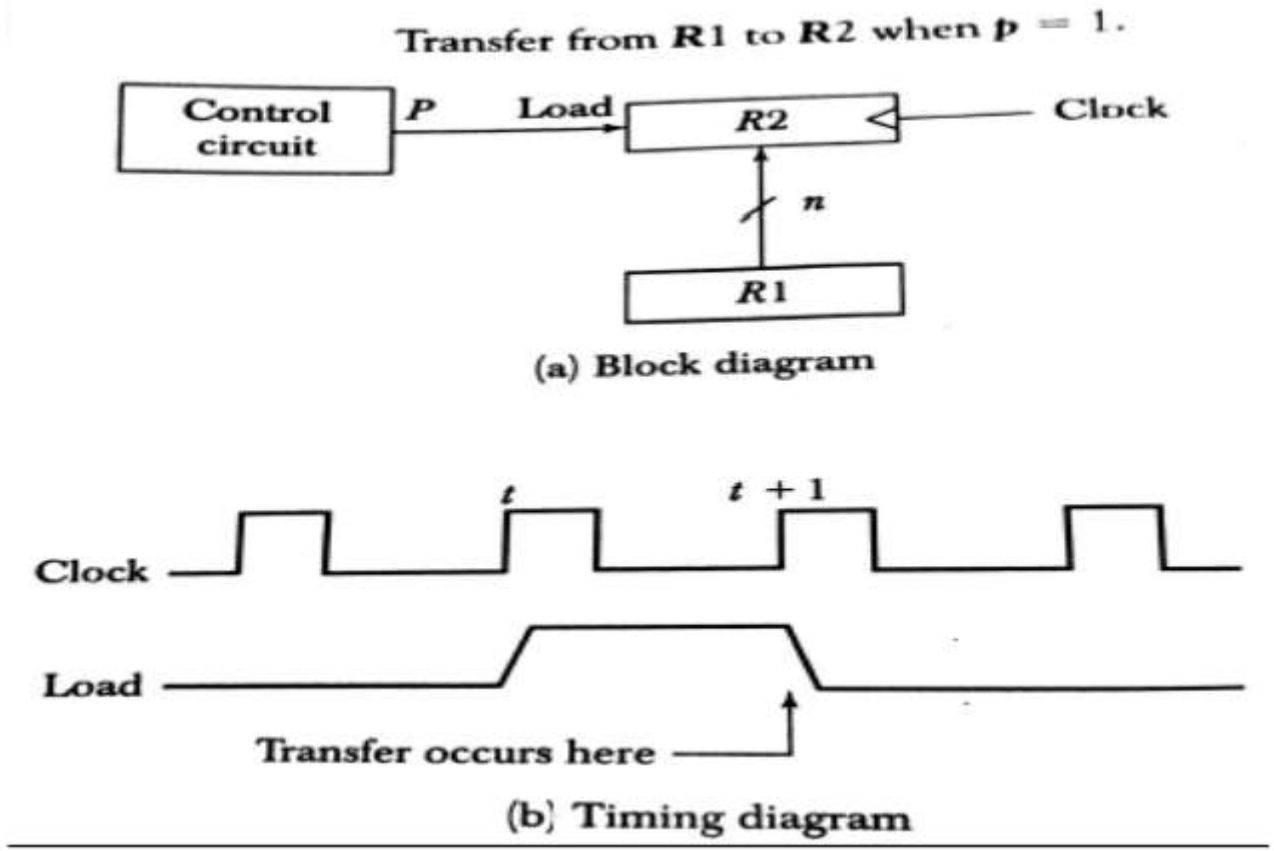
Register Transfer

- Information transfer from one register to another is designated in symbolic form by means of a replacement operator.
- The statement R2← R1 denotes a transfer of the content of register R1 into register R2.
- It designates a replacement of the content of R2 by the content of R1. By definition, the content of the source register R 1 does not change after the transfer.
- If we want the transfer to occur only under a predetermined control condition then it can be shown by an if-then statement.
    - if (P=1) then R2← R1

Contd…

- P is the control signal generated by a control section.
- We can separate the control variables from the register transfer operation by specifying a Control Function.
- Control function is a Boolean variable that is equal to 0 or 1.
- control function is included in the statement as P: R2← R1
- Control condition is terminated by a colon implies transfer operation be executed by the hardware only if P=1.
- Every statement written in a register transfer notation implies a hardware construction for implementing the transfer.

Block diagram that depicts the transfer from R1 to R2



Contd….

- The n outputs of register R1 are connected to the n inputs of register R2.
- The letter n will be used to indicate any number of bits for the register. It will be replaced by an actual number when the length of the register is known.
- Register R2 has a load input that is activated by the control variable P.
- It is assumed that the control variable is synchronized with the same clock as the one applied to the register.
- As shown in the timing diagram, P is activated in the control section by the rising edge of a clock pulse at time t.
- The next positive transition of the clock at time t + 1 finds the load input active and the data inputs of R2 are then loaded into the register in parallel.

Contd..

- P may go back to 0 at time t+1; otherwise, the transfer will occur with every clock pulse transition while P remains active.
- Even though the control condition such as P becomes active just after time t, the actual transfer does not occur until the register is triggered by the next positive transition of the clock at time t +1.
- A comma is used to separate two or more operations that are executed at the same time.
- The statement T : R2← R1, R1← R2 (exchange operation) denotes an operation that exchanges the contents of two registers during one common clock pulse provided that T=1.

The basic symbols of the register transfer notation are listed in below table

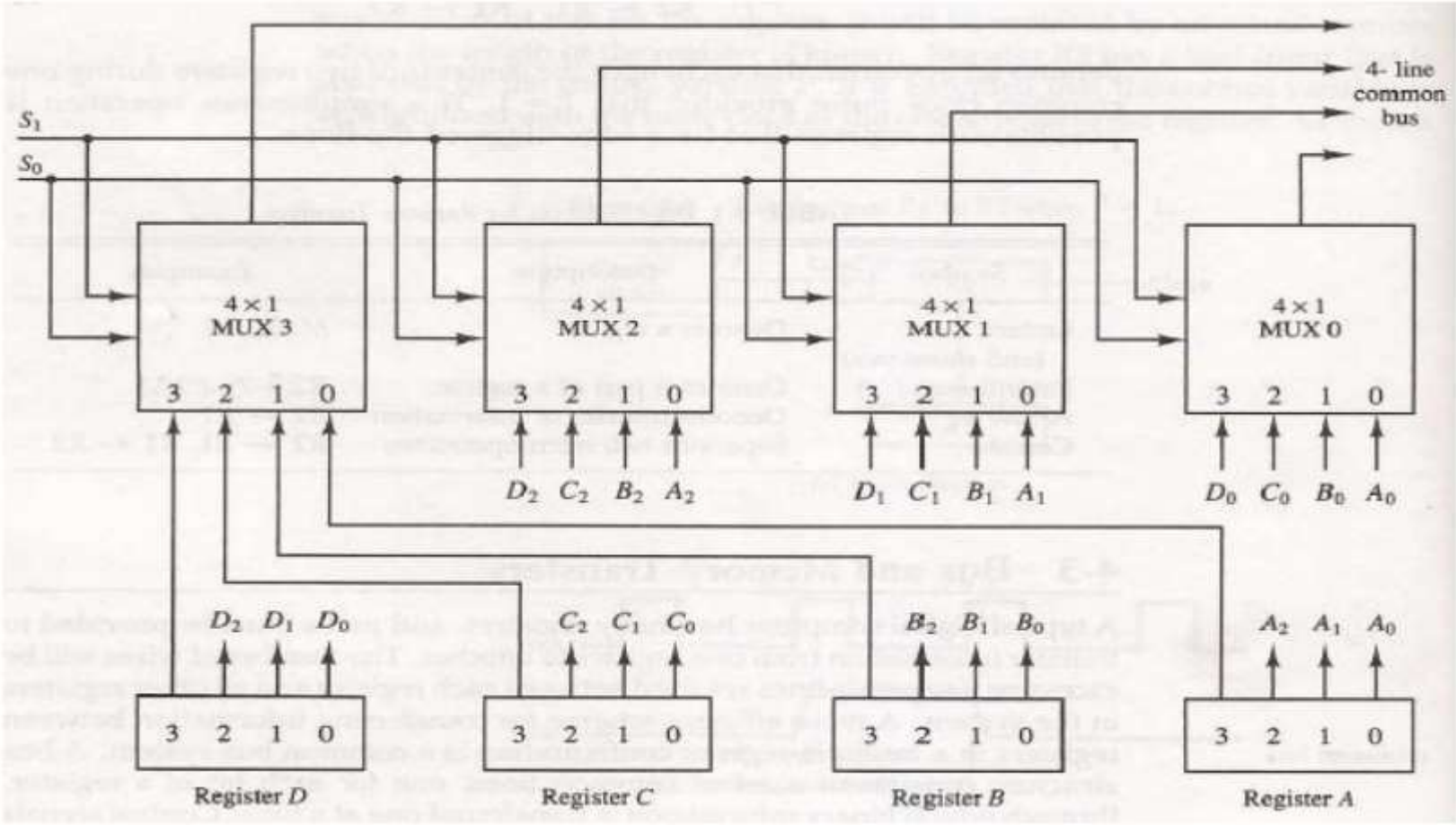
Symbol	Description	Examples
Letters(and numerals)	Denotes a register	MAR, R2
Parentheses ( )	Denotes a part of a register	R2(0-7), R2(L)
Arrow <--	Denotes transfer of information	R2 <-- R1
Comma ,	Separates two microoperations	R2 <-- R1, R1 <-- R2

Bus and Memory Transfers

- A more efficient scheme for transferring information between registers in a multiple-register configuration is a Common Bus System.
- A common bus consists of a set of common lines, one for each bit of a register.
- Control signals determine which register is selected by the bus during each particular register transfer.
- Different ways of constructing a Common Bus System
    1. Using Multiplexers
    2. Using Tri-state Buffers

Common bus system is with multiplexers:

- The multiplexers select the source register whose binary information is then placed on the bus.



Contd..

- The bus consists of four 4 x 1 multiplexers each having four data inputs, 0 through 3, and two selection inputs, S1 and S0.
- For example, output 1 of register A is connected to input 0 of MUX 1 because this input is labelled A1.
- The diagram shows that the bits in the same significant position in each register are connected to the data inputs of one multiplexer to form one line of the bus.
- Thus MUX 0 multiplexes the four 0 bits of the registers, MUX 1 multiplexes the four 1 bits of the registers, and similarly for the other two bits.
- The two selection lines S1 and S0 are connected to the selection inputs of all four multiplexers.
- The selection lines choose the four bits of one register and transfer them into the four-line common bus.

Contd..

- When S1S0 = 00, the 0 data inputs of all four multiplexers are selected and applied to the outputs that form the bus.
- This causes the bus lines to receive the content of register A since the outputs of this register are connected to the 0 data inputs of the multiplexers.
- Similarly, register B is selected if S1S0 = 01, and so on.
- Table shows the register that is selected by the bus for each of the four possible binary value of the selection lines.

S_1	S_0	Register selected
0	0	*A*
0	1	*B*
1	0	*C*
1	1	*D*

Contd….

- In general a bus system has
  - ✓multiplex “k” Registers
  - ✓each register of “n” bits
  - ✓to produce “n-line bus”
  - ✓no. of multiplexers required = n
  - ✓size of each multiplexer = k x 1
  - ➢When the bus is includes in the statement, the register transfer is symbolized as follows: BUS← C, R1← BUS
  - ➢The content of register C is placed on the bus, and the content of the bus is loaded into register R1 by activating its load control input. If the bus is known to exist in the system, it may be convenient just to show the direct transfer. R1← C

Memory Transfer

- The transfer of information from a memory word to the outside environment is called a read operation.
- The transfer of new information to be stored into the memory is called a write operation.
- A memory word will be symbolized by the letter M.
- The particular memory word among the many available is selected by the memory address during the transfer.
- It is necessary to specify the address of M when writing memory transfer operations.
- This will be done by enclosing the address in square brackets following the letter M.
- Consider a memory unit that receives the address from a register, called the address register, symbolized by AR.
- The data are transferred to another register, called the data register, symbolized by DR.
- The read operation can be stated as follows:

Read: DR<- M [AR]

This causes a transfer of information into DR from the memory word M selected by the address in AR

Contd….

- The write operation transfers the content of a data register to a memory word M selected by the address. Assume that the input data are in register R1 and the address is in AR.
- The write operation can be stated as follows:

Write: M [AR] <- R1

Types of Micro-operations

- Register Transfer Micro-operations: Transfer binary information from one register to another.
- Arithmetic Micro-operations: Perform arithmetic operation on numeric data stored in registers.
- Logical Micro-operations: Perform bit manipulation operations on data stored in registers.
- Shift Micro-operations: Perform shift operations on data stored in registers
- Note: Register Transfer Micro-operation doesn't change the information content when the binary information moves from source register to destination register. Other three types of micro-operations change the information change the information content during the transfer.

Arithmetic Micro-operations

- The basic arithmetic micro-operations are
  - ✓Addition
  - ✓Subtraction
  - ✓Increment
  - ✓Decrement
  - ✓Shift
- The arithmetic Micro-operation defined by the statement below specifies the add microoperation. $R3 \leftarrow R1 + R2$
- It states that the contents of R1 are added to contents of R2 and sum is transferred to R3.
- To implement this statement hardware requires 3 registers and digital component that performs addition

Contd….

- Subtraction is most often implemented through complementation and addition.
- The subtract operation is specified by the following statement
- $R3 \leftarrow R1 + \overline{R2} + 1$
- instead of minus operator, we can write as
- $\overline{R2}$ is the symbol for the 1’s complement of R2
- Adding 1 to 1’s complement produces 2’s complement
- Adding the contents of R1 to the 2's complement of R2 is equivalent to R1-R2.