UNIT-2: THE COMPUTER HARDWARE

Contents:

- Introduction, to computer system, Central Processing Unit, Components of CPU, Instruction Format, Instruction Set ,Instruction Cycle, Microprocessor, and Computer bus.
- Components of computer Cabinet(Power supply, motherboard, memory chips, expansion slots, ports and interfaces, processor, cables and storage devices)
- Memory unit: memory and its functions, Primary memory and Secondary memory

HARDWARE

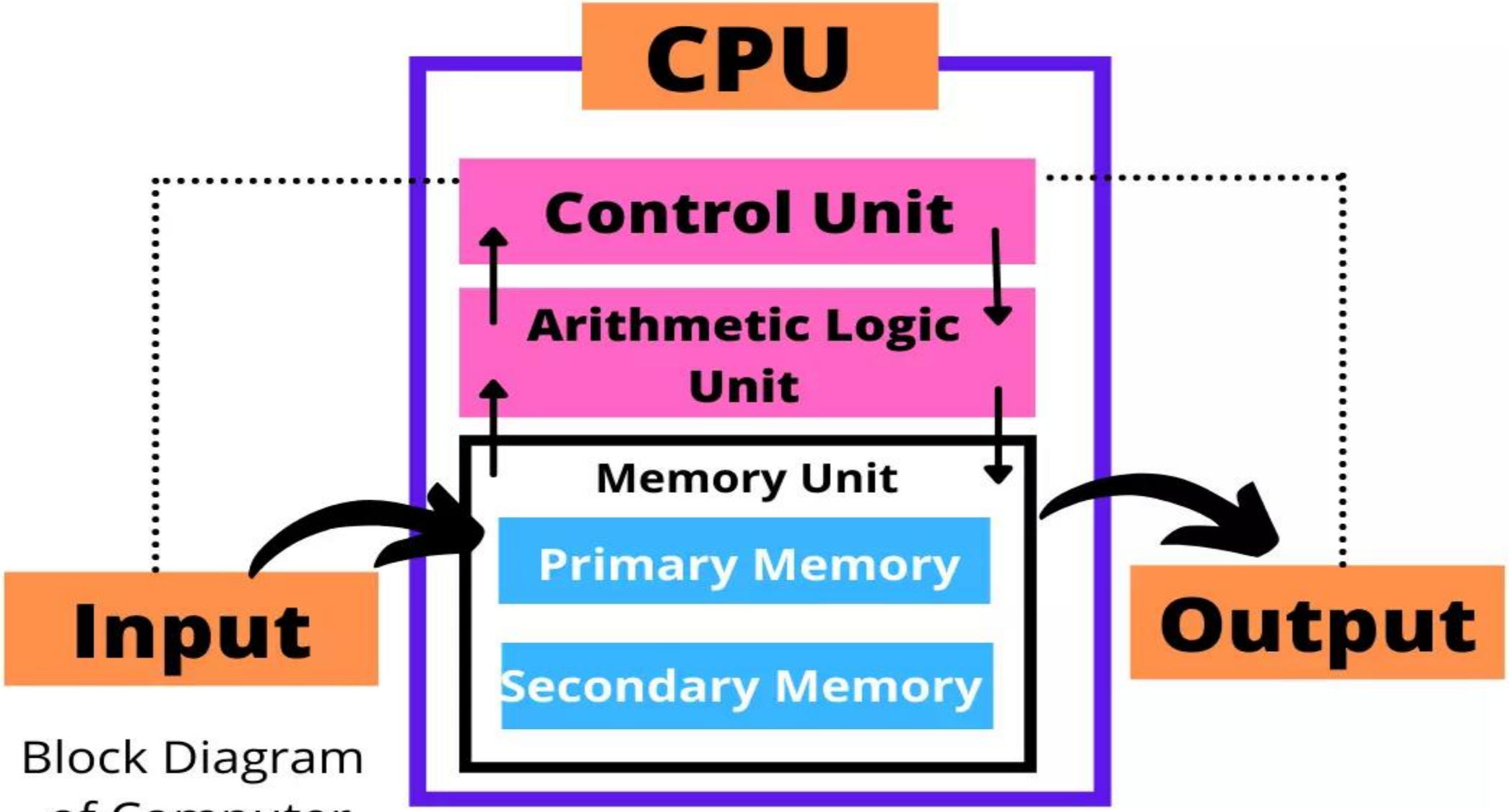
- Computer hardware includes the physical parts of a computer, such as the case, central processing unit, random access memory, monitor, mouse, keyboard, computer data storage, graphics card, sound card, speakers and motherboard.
- Quite simply, computer hardware is **the physical components that a computer system requires to function**.
- It encompasses everything with a circuit board that operates within a PC or laptop; including the motherboard, graphics card, CPU (Central Processing Unit), ventilation fans, webcam, power supply, and so on

COMPUTER SYSTEMS

- **A computer along with additional hardware and software together** is called a computer system.
- A computer system primarily comprises a central processing unit (CPU), memory, input/output devices and storage devices.
- All these components function together as a single unit to deliver the desired output.
- Computer systems are a combination of both hardware and software working together.
- Hardware is the physical components of a computer and software is the programs that run on a computer.

WHAT IS CPU ?

- A central processing unit, also called a central processor, main processor or just processor, is the electronic circuitry that executes instructions comprising a computer program.
- The CPU performs basic arithmetic, logic, controlling, and input/output operations specified by the instructions in the program.
- It is the brain of a computer where all the calculations and process takes place

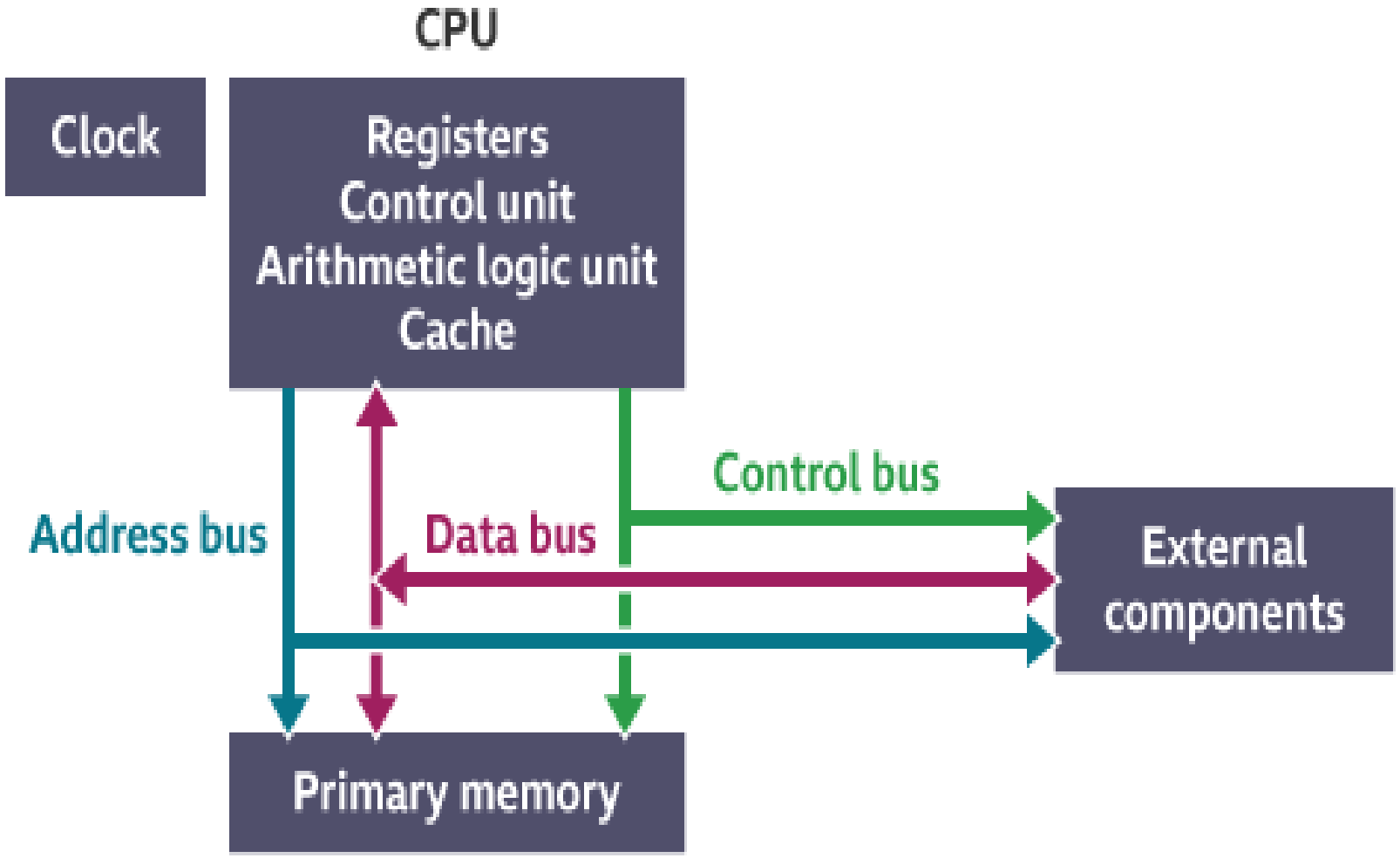


Block Diagram of Computer

COMMON CPU COMPONENTS

- The **central processing unit (CPU)** consists of six main components:
  - **control unit (CU)**
  - **arithmetic logic unit (ALU)**
  - **registers**
  - **cache**
  - **buses**
  - **clock**
- All components work together to allow processing and system control.

COMMON CPU COMPONENTS CONT..



CONTROL UNIT

- The CU provides several functions:
- it **fetches**, **decodes** and **executes** instructions
- it issues control signals that control **hardware**
- it moves **data** around the system
- It directs the operation of the processor
- It tells the computer's memory, arithmetic/logic unit and I/O devices on how to respond to a program instruction
- It directs the operation of the other units by providing timing and control signals

ARITHMETIC LOGIC UNIT

- The ALU has two main functions:
- It performs arithmetic and logical operations (decisions).
- The ALU is where calculations are done and where decisions are made.
- It acts as a gateway between primary **memory** and **secondary storage** . Data transferred between them passes through the ALU.
- **The ALU performs calculations and makes logical decisions**

REGISTERS

- Registers are small amounts of high-speed memory contained within the CPU. They are used by the processor to store small amounts of data that are needed during processing, such as:
- the address of the next instruction to be executed
- the current instruction being decoded
- the results of calculations
- Different processors have different numbers of registers for different purposes, but most have some, or all, of the following:
- program counter
- memory address register (MAR)
- memory data register (MDR)
- current instruction register (CIR)
- accumulator (ACC)

CACHE

- Cache is a small amount of high-speed random access memory (RAM) built directly within the processor.
- It is used to temporarily hold data and instructions that the processor is likely to reuse.
- This allows for faster processing as the processor does not have to wait for the data and instructions to be fetched from the RAM.

BUSES

- A bus is a high-speed internal connection. Buses are used to send control signals and data between the processor and other components.
- Three types of bus are used:
- **Address bus** - carries memory addresses from the processor to other components such as primary memory and input/output devices.
- **Data bus** - carries the actual data between the processor and other components.
- **Control bus** - carries control signals from the processor to other components. The control bus also carries the clock's pulses.

BUSES CONT..

- Bus is a communication system that transfer data between components inside a computer.
- Wires and electronic pathways joins the various components of a computer system
- Network of such electronic pathways and various components that create a communication among them is known as bus.
- **Internal Bus**
  - A BUS or set of wires which connects the various components inside a computer, is known as Internal Bus.
- **External Bus**
  - A Bus or set of wires which is used to connect outer peripherals or components to computer , is known as External Bus

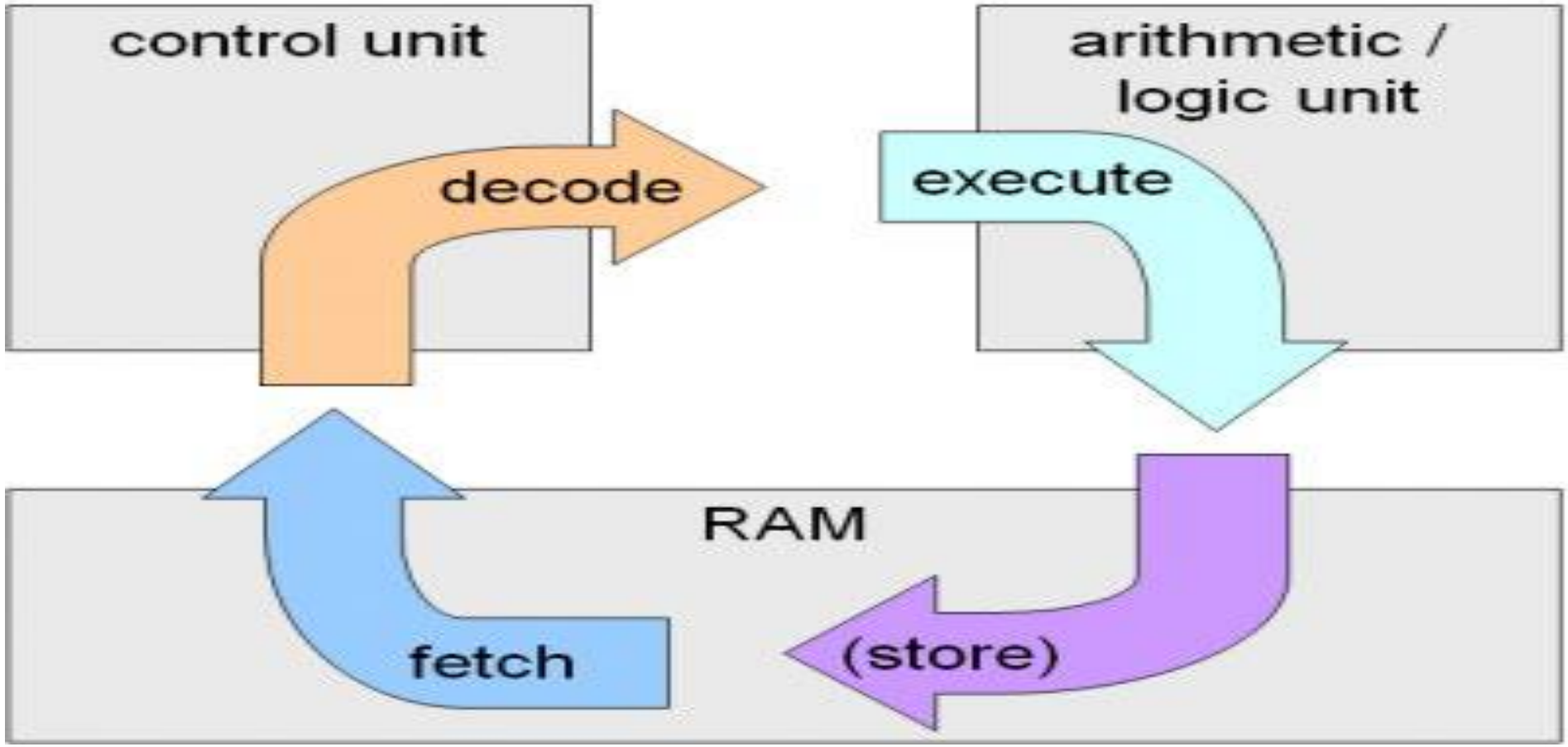
CLOCK

- The CPU contains a clock which is used to coordinate all of the computer's components.
- The clock sends out a regular electrical pulse which synchronises (keeps in time) all the components.
- The frequency of the pulses is known as the **clock speed**. Clock speed is measured in **hertz**. The higher the frequency, the more instructions can be performed in any given moment of time.

FUNCTIONS OF CPU

- **Fetch**
  - Each instruction is stored in memory and has its own address. The processor takes this address number from the program counter, which is responsible for tracking which instructions the CPU should execute next.
- **Decode**
- All programs to be executed are translated to into Assembly instructions. Assembly code must be decoded into binary instructions, which are understandable to your CPU. This step is called decoding.
- **Execute**
  - While executing instructions the CPU can do one of three things: Do calculations with its ALU, move data from one memory location to another, or jump to a different address.
- **Store**
  - The CPU must give feedback after executing an instruction and the output data is written to the memory.

CPU CYCLE


control unit
arithmetic / logic unit
decode
execute
RAM
fetch
(store)

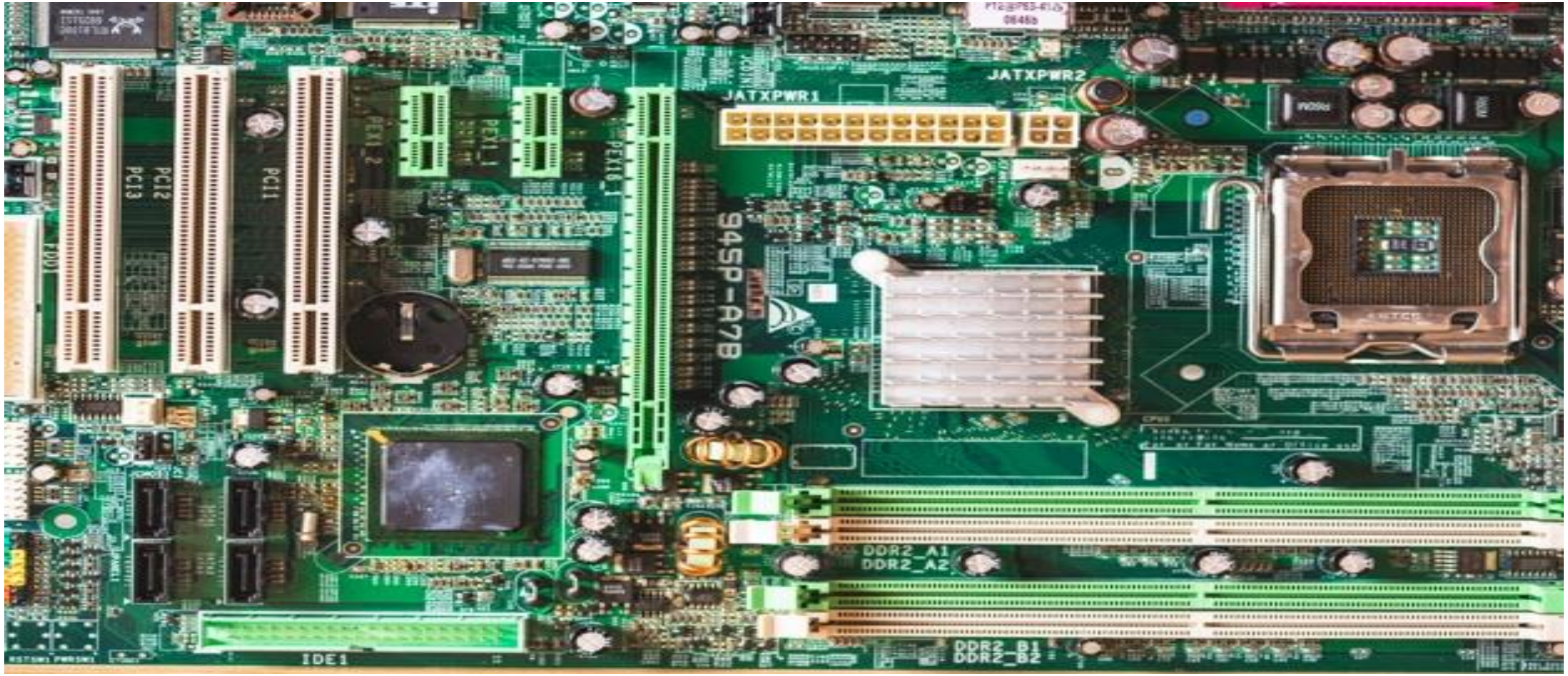
INSIDE A COMPUTER CABINET

- The computer cabinet consists of the components that are required for running the computer system effectively with fewer errors.
- There are various elements in the cabinet to which some of them are Motherboard, memory chips, cables, processors, ports, etc.

Motherboard

- The **motherboard** is the computer's **main circuit board**.
- It's a thin plate that holds the CPU, memory, connectors for the hard drive and optical drives, expansion cards to control the video and audio, and connections to your computer's ports (such as USB ports).The motherboard connects directly or indirectly to every part of the computer.

INSIDE A COMPUTER CABINET CONT..
MOTHERBOARD



INSIDE A COMPUTER CABINET CONT..

- **Ports and interfaces Cont..**
- Motherboard has a certain number of I/O sockets that are connected to the ports and interfaces found on the rear side of a computer.
- We can connect external devices to the ports and interfaces, which get connected to the computer's motherboard.
- · Serial Port— to connect old peripherals.
- · Parallel Port— to connect old printers.
- · USB Ports—to connect newer peripherals like cameras, scanners and printers to the computer. It uses a thin wire to connect to the devices, and many devices can share that wire simultaneously.
- · Fire wire is another bus, used today mostly for video cameras and external hard drives.
- · J45 connector (called LAN or Ethernet port) is used to connect the computer to a network. It corresponds to a network card integrated into the motherboard.

INSIDE A COMPUTER CABINET CONT..

Ports and interfaces Cont..

- J45 connector (called LAN or Ethernet port) is used to connect the computer to a network. It corresponds to a network card integrated into the motherboard.
- · VGA connector for connecting a monitor. This connector interfaces with the built-in graphics card.
- · Audio plugs (line-in, line-out and microphone), for connecting sound speakers and the microphone. This connector interfaces with the built-in sound card.
- · PS/2 port to connect mouse and keyboard into PC.
- · SCSI port for connecting the hard disk drives and network connectors.

INSIDE A COMPUTER CABINET CONT..

- **Expansion Slots**
- The expansion slots are located on the motherboard. The expansion cards are inserted in the expansion slots. These cards give the computer new features or increased performance.
- There are several types of slots:
- · ISA (Industry Standard Architecture) slot—To connect modem and input devices.
- · PCI (Peripheral Component Interconnect) slot—To connect audio, video and graphics. They are much faster than ISA cards.
- · AGP (Accelerated Graphic Port) slot—A fast port for a graphics card.
- · (Peripheral Component Interconnect) Express slot—Faster bus architecture than AGP and PCI buses.
- · PC Card—It is used in laptop computers. It includes Wi-Fi card, network card and external modem.

INSIDE A COMPUTER CABINET CONT..

Memory Chips

- The RAM consists of chips on a small circuit board.
- Two types of memory chips—— Single In-line Memory Module (SIMM) and Dual In-line Memory Module (DIMM) are used in desktop computers.
- The CPU can retrieve information from DIMM chip at 64 bits compared to 32 bits or 16 bits transfer with SIMM chips.
- DIMM chips are used in Pentium 4 onwards to increase the access speed.

Processor

- The processor or the CPU is the main component of the computer. Select a processor based on factors like its speed, performance, reliability and motherboard support.
- Pentium Pro, Pentium 2 and Pentium 4 are some of the processors.

INSIDE A COMPUTER CABINET CONT..

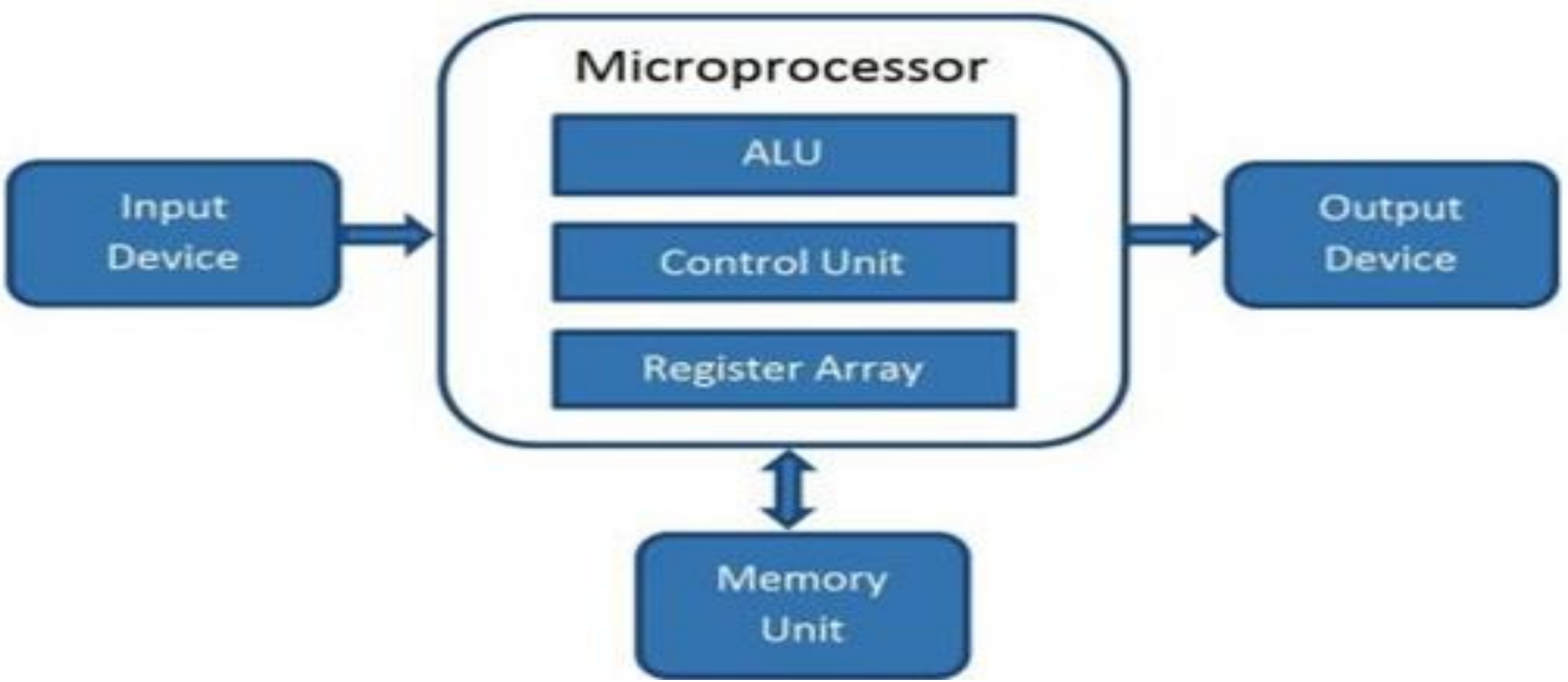
Storage Devices

- The disk drives are present inside the machine.
- The common disk drives in a machine are hard disk drive, floppy drive and CD drive or DVD drive.
- High-storage devices like hard disk, floppy disk and CDs are inserted into the hard disk drive, floppy drive and CD drive, respectively.
- These storage devices can store large amounts of data, permanently.

MICROPROCESSOR

- A **Microprocessor** is an important part of a **computer architecture** without which you will not be able to perform anything on your **computer**. ... 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.
- A **microprocessor** is an electronic component that is used by a computer to do its work. It is a central processing unit on a single integrated circuit chip containing millions of very small components including transistors, resistors, and diodes that work together.
- The **microprocessor** is the central unit of a computer system that performs arithmetic and logic operations, which generally include adding, subtracting, transferring numbers from one area to another, and comparing two numbers. It's often known simply as a processor, a central processing unit, or as a logic chip.

MICROPROCESSOR CONT..



MICROPROCESSOR CONT..

- *Microprocessors can be classified in different categories, as follows:*

Based on Word Length

- Microprocessors can be based on the number of bits the processor's internal data bus or the number of bits that it can process at a time (which is known as the word length). Based on its word length, a microprocessor can be classified as 8-bit, 16-bit, 32-bit, and 64-bit.

Reduced Instruction Set Computer (RISC)

- RISC microprocessors are more general use than those that have a more specific set of instructions.
- The execution of instructions in a processor requires a special circuit to load and process data. Because RISC microprocessors have fewer instructions, they have simpler circuits, which means they operate faster.
- Additionally, RISC microprocessors have more registers, use more RAM, and use a fixed number of clock cycles to execute one instruction.

MICROPROCESSOR CONT..

Complex Instruction Set Computer

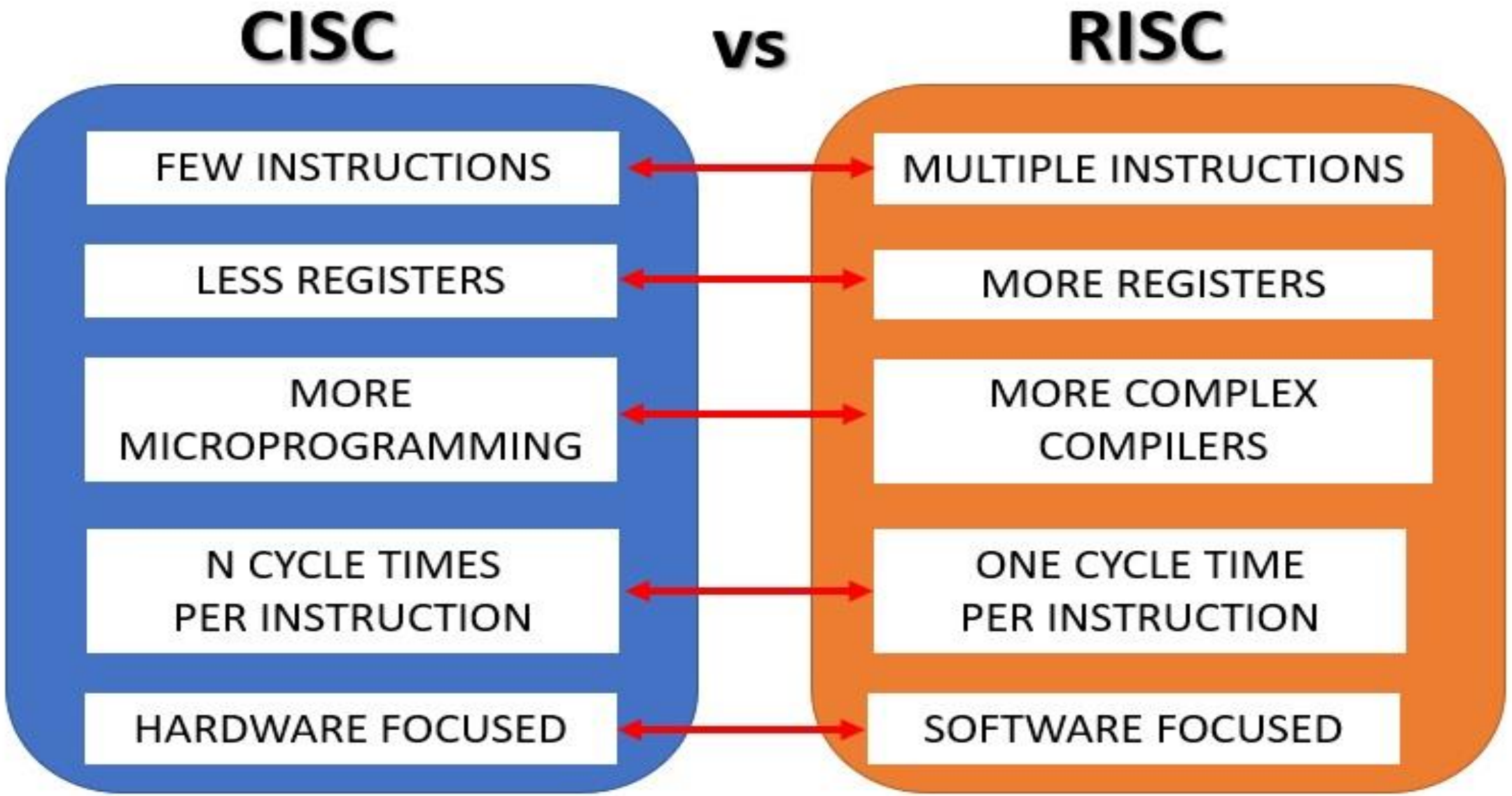
- CISC microprocessors are the opposite of RISC microprocessors. Their purpose is to reduce the number of instructions for each program.
- The number of cycles per instruction is ignored. Because complex instructions are made directly into the hardware, CISC microprocessors are more complex and slower.
- CISC microprocessors use little RAM, have more transistors, have fewer registers, have numerous clock cycles for each instruction, and have a variety of addressing modes.

Special Purpose Processors

- Some microprocessors are built to perform specific functions. For example, coprocessors are used in combination with a main processor, while a transporter is a transistor computer: a microprocessor that has its own local memory.

RISC AND CISC ARCHITECTURE

- **RISC** stands for 'Reduced Instruction Set Computer whereas, **CISC** stands for Complex Instruction Set Computer. The **RISC** processors have a smaller set of instructions with few addressing nodes.
- It is a type of microprocessor that has a limited number of instructions. They can execute their instructions very fast because instructions are very small and simple.
- **RISC** chips require fewer transistors which make them cheaper to design and produce.
- The **CISC** processors have a larger set of instructions with many addressing nodes.
- A complex instruction set **computer** (**CISC**) is a **computer** in which single instructions can execute several low-level operations (such as a load from memory, an arithmetic operation, and a memory store) or are capable of multi-step operations or addressing modes within single instructions.


CISC
vs
RISC
FEW INSTRUCTIONS
LESS REGISTERS
MORE MICROPROGRAMMING
N CYCLE TIMES PER INSTRUCTION
HARDWARE FOCUSED
MULTIPLE INSTRUCTIONS
MORE REGISTERS
MORE COMPLEX COMPILERS
ONE CYCLE TIME PER INSTRUCTION
SOFTWARE FOCUSED

RISC VS. CISC

- **RISC – Reduced Instruction Set Computer**
  - Advocates fewer and simpler instructions
  - CPU can be simpler, means each instruction can be executed quickly
  - Benchmarks: indicate that most programs spend the majority of time doing these simple instructions, so make the common case go fast!
  - Downside: uncommon case goes slow (e.g., instead of a single SORT instruction, need lots of simple instructions to implement a sort)
  - Sparc, Motorola, Alpha
- **CISC – Complex Instruction Set Computer**
  - Advocates many instructions that can perform complex tasks
  - E.g. SORT instruction
  - Additional complexity in the CPU
    - This complexity typically makes ALL instructions slower to execute, not just the complex ones
  - Fewer instructions needed to write a program using CISC due to richness of instructions available
  - Intel x86

INSTRUCTION SET

- An **instruction set** is a group of commands for a CPU in machine language. ... All CPUs have **instruction sets** that enable commands to the processor directing the CPU to switch the relevant transistors.
- Some **instructions** are simple read, write and move commands that direct data to different hardware.

WHAT IS AN INSTRUCTION SET?

- The complete collection of instructions that are understood by a CPU
  - The physical hardware that is controlled by the instructions is referred to as the Instruction Set Architecture (ISA)
- The instruction set is ultimately represented in binary **machine code** also referred to as **object code**
  - Usually represented by assembly codes to human programmer

INSTRUCTION FORMATS

Instruction sets are differentiated by the following:

- Number of bits per instruction.
- Stack-based or register-based.
- Number of explicit operands per instruction.
- Operand location.
- Types of operations.
- Type and size of operands.

INSTRUCTION FORMATS CONT..

Instruction set architectures are measured according to:

- Main memory space occupied by a program.
- Instruction complexity.
- Instruction length (in bits).
- Total number of instruction in the instruction set.

INSTRUCTION FORMATS CONT..

In designing an instruction set, consideration is given to:

- Instruction length.
  - Whether short, long, or variable.
- Number of operands.
- Number of addressable registers.
- Memory organization.
  - Whether byte- or word addressable.
- Addressing modes.
  - Choose any or all: direct, indirect or indexed.

INSTRUCTION FORMATS CONT..

- Byte ordering, or *endianness*, is another major architectural consideration.
- If we have a two-byte integer, the integer may be stored so that the least significant byte is followed by the most significant byte or vice versa.
  - In *little endian* machines, the least significant byte is followed by the most significant byte.
  - *Big endian* machines store the most significant byte first (at the lower address).

INSTRUCTION FORMATS CONT..

- As an example, suppose we have the hexadecimal number 12345678.
- The big endian and small endian arrangements of the bytes are shown below.

Address →	00	01	10	11
Big Endian	12	34	56	78
Little Endian	78	56	34	12

Note: This is the internal storage format, usually invisible to the user

INSTRUCTION FORMATS CONT..

- **Big endian:**
  - Is more natural.
  - The sign of the number can be determined by looking at the byte at address offset 0.
  - Strings and integers are stored in the same order.
- **Little endian:**
  - Makes it easier to place values on non-word boundaries, e.g. odd or even addresses
  - Conversion from a 32-bit integer to a 16-bit integer does not require any arithmetic.

STANDARD…WHAT STANDARD?

- Intel (80x86), VAX are little-endian
- IBM 370, Motorola 680x0 (Mac), and most RISC systems are big-endian
- Makes it problematic to translate data back and forth between say a Mac/PC
- Internet is big-endian
  - Why? Useful control bits in the Most Significant Byte can be processed as the data streams in to avoid processing the rest of the data
  - Makes writing Internet programs on PC more awkward!
  - Must convert back and forth

ELEMENTS OF AN INSTRUCTION

- Operation code (Op code)
  - Do this
- Source Operand reference(s)
  - To this
- Result Operand reference(s)
  - Put the answer here
- Next Instruction Reference
  - When you are done, do this instruction next

WHERE ARE THE OPERANDS?

- Main memory
- CPU register
- I/O device
- In instruction itself
- To specify which register, which memory location, or which I/O device, we'll need some addressing scheme for each

INSTRUCTION SET DESIGN

- One important design factor is the number of operands contained in each instruction
  - Has a significant impact on the word size and complexity of the CPU
  - E.g. lots of operands generally implies longer word size needed for an instruction
- Consider how many operands we need for an ADD instruction
  - If we want to add the contents of two memory locations together, then we need to be able to handle at least **two** memory addresses
  - Where does the result of the add go? We need a **third** operand to specify the destination
  - What instruction should be executed next?
    - Usually the next instruction, but sometimes we might want to jump or branch somewhere else
    - To specify the next instruction to execute we need a **fourth** operand
- If all of these operands are memory addresses, we need a really long instruction!

NUMBER OF OPERANDS

- In practice, we won't really see a four-address instruction.
  - Too much additional complexity in the CPU
  - Long instruction word
  - All operands won't be used very frequently
- Most instructions have one, two, or three operand addresses
  - The next instruction is obtained by incrementing the program counter, with the exception of branch instructions
- Let's describe a hypothetical set of instructions to carry out the computation for:

$$Y = (A-B) / (C + (D * E))$$

THREE OPERAND INSTRUCTION

- If we had a three operand instruction, we could specify two source operands and a destination to store the result.
- Here is a possible sequence of instructions for our equation:

$$Y = (A-B) / (C + (D * E))$$

  - SUB R1, A, B ; Register R1 ← A-B
  - MUL R2, D, E ; Register R2 ← D * E
  - ADD R2, R2, C ; Register R2 ← R2 + C
  - DIV R1, R1, R2 ; Register R1 ← R1 / R2

- The three address format is fairly convenient because we have the flexibility to dictate where the result of computations should go. Note that after this calculation is done, we haven't changed the contents of any of our original locations A,B,C,D, or E.

TWO OPERAND INSTRUCTION

- How can we cut down the number of operands?
  - Might want to make the instruction shorter
- Typical method is to assign a **default** destination operand to hold the results of the computation
  - Result always goes into this operand
  - Overwrites and old data in that location
- Let's say that the default destination is the first operand in the instruction
  - First operand might be a register, memory, etc.

TWO OPERAND INSTRUCTIONS

- Here is a possible sequence of instructions for our equation (say the operands are registers):

  Y = (A-B) / (C + (D * E))

  - SUB A, B ; Register A ← A-B
  - MUL D, E ; Register D ← D*E
  - ADD D, C ; Register D ← D+C
  - DIV A, D ; Register A ← A / D
- Get same end result as before, but we changed the contents of registers A and D
- If we had some later processing that wanted to use the original contents of those registers, we must make a copy of them before performing the computation
  - MOV R1, A ; Copy A to register R1
  - MOV R2, D ; Copy D to register R2
  - SUB R1, B ; Register R1 ← R1-B
  - MUL R2, E ; Register R2 ← R2*E
  - ADD R2, C ; Register R2 ← R2+C
  - DIV R1, R2 ; Register R1 ← R1 / R2
- Now the original registers for A-E remain the same as before, but at the cost of some extra instructions to save the results.

ONE OPERAND INSTRUCTIONS

- Can use the same idea to get rid of the second operand, leaving only one operand
- The second operand is left implicit; e.g. could assume that the second operand will always be in a register such as the Accumulator:

  Y = (A-B) / (C + (D * E))

  - LDA D ; Load ACC with D
  - MUL E ; Acc ← Acc * E
  - ADD C ; Acc ← Acc + C
  - STO R1 ; Store Acc to R1
  - LDA A ; Acc ← A
  - SUB B ; Acc ← A-B
  - DIV R1 ; Acc ← Acc / R1
- Many early computers relied heavily on one-address based instructions, as it makes the CPU much simpler to design. As you can see, it does become somewhat more unwieldy to program.

ZERO OPERAND INSTRUCTIONS

- In some cases we can have zero operand instructions
- Uses the **Stack**
  - Section of memory where we can add and remove items in LIFO order
  - Last In, First Out
  - Envision a stack of trays in a cafeteria; the last tray placed on the stack is the first one someone takes out
  - The stack in the computer behaves the same way, but with data values
    - PUSH A ; Places A on top of stack
    - POP A ; Removes value on top of stack and puts result in A
    - ADD ; Pops top two values off stack, pushes result back on

STACK-BASED INSTRUCTIONS

Y = (A-B) / (C + (D * E))

Instruction	Stack Contents (top to left)
PUSH B	; B
PUSH A	; B, A
SUB	; (A-B)
PUSH E	; (A-B), E
PUSH D	; (A-B), E, D
MUL	; (A-B), (E*D)
PUSH C	; (A-B), (E*D), C
ADD	; (A-B), (E*D+C)
DIV	; (A-B) / (E*D+C)

HOW MANY OPERANDS IS BEST?

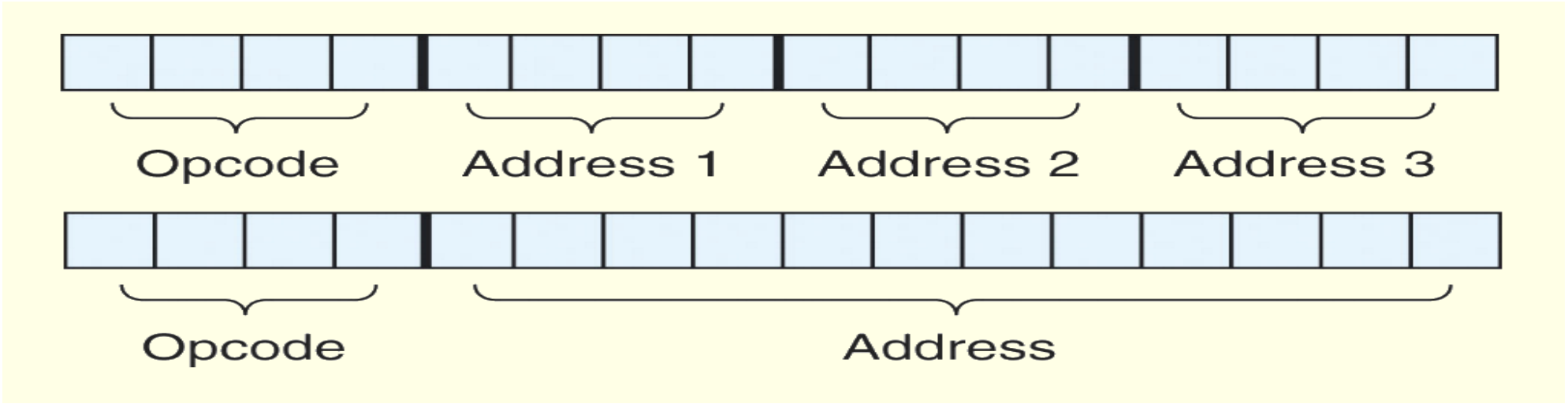
- More operands
  - More complex (powerful?) instructions
  - Fewer instructions per program
- More registers
  - Inter-register operations are quicker
- Fewer operands
  - Less complex (powerful?) instructions
  - More instructions per program
  - Faster fetch/execution of instructions

DESIGN TRADEOFF DECISIONS

- Operation repertoire
  - How many ops?
  - What can they do?
  - How complex are they?
- Data types
  - What types of data should ops perform on?
- Registers
  - Number of registers, what ops on what registers?
- Addressing
  - Mode by which an address is specified (more on this later)

Instruction Formats

- A system has 16 registers and 4K of memory.
- We need 4 bits to access one of the registers. We also need 10 bits for a memory address.
- If the system is to have 16-bit instructions, we have two choices for our instructions:



INSTRUCTION TYPES

Instructions fall into several broad categories that you should be familiar with:

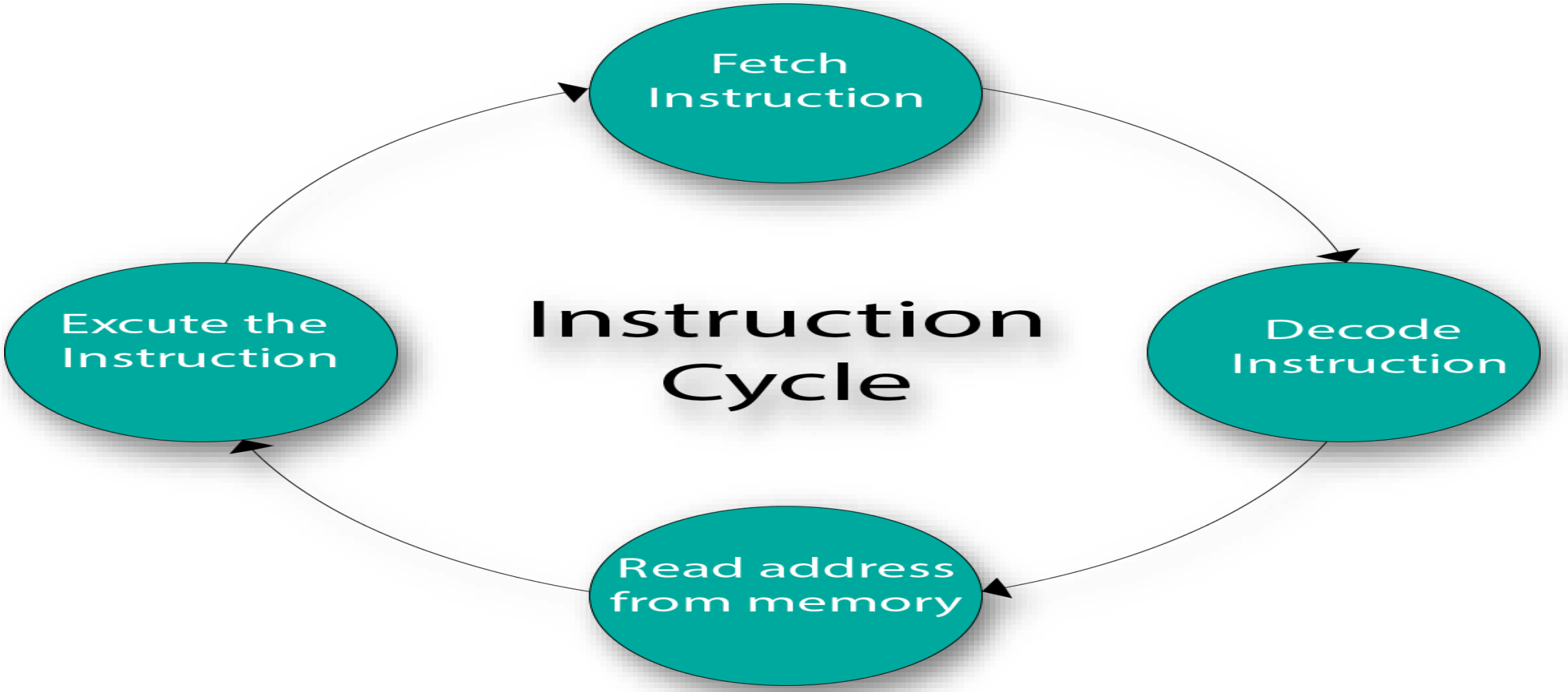
- Data movement
- Arithmetic
- Boolean
- Bit manipulation
- I/O
- Control transfer
- Special purpose

Can you think of some examples of each of these?

INSTRUCTION CYCLE

- A program residing in the memory unit of a computer consists of a sequence of instructions. These instructions are executed by the processor by going through a cycle for each instruction.
- In a basic computer, each instruction cycle consists of the following phases:
- Fetch instruction from memory.
- Decode the instruction.
- Read the effective address from memory.
- Execute the instruction.

INSTRUCTION CYCLE CONT..



INSTRUCTION-LEVEL PIPELINING

- Some CPUs divide the fetch-decode-execute cycle into smaller steps.
- These smaller steps can often be executed in parallel to increase throughput.
- Such parallel execution is called *instruction-level pipelining*.
- This term is sometimes abbreviated *ILP* in the literature.

INSTRUCTION PRE FETCH

- Simple version of Pipelining – treating the instruction cycle like an assembly line
- Fetch accessing main memory
- Execution usually does not access main memory
- Can fetch next instruction during execution of current instruction
- Called instruction prefetch

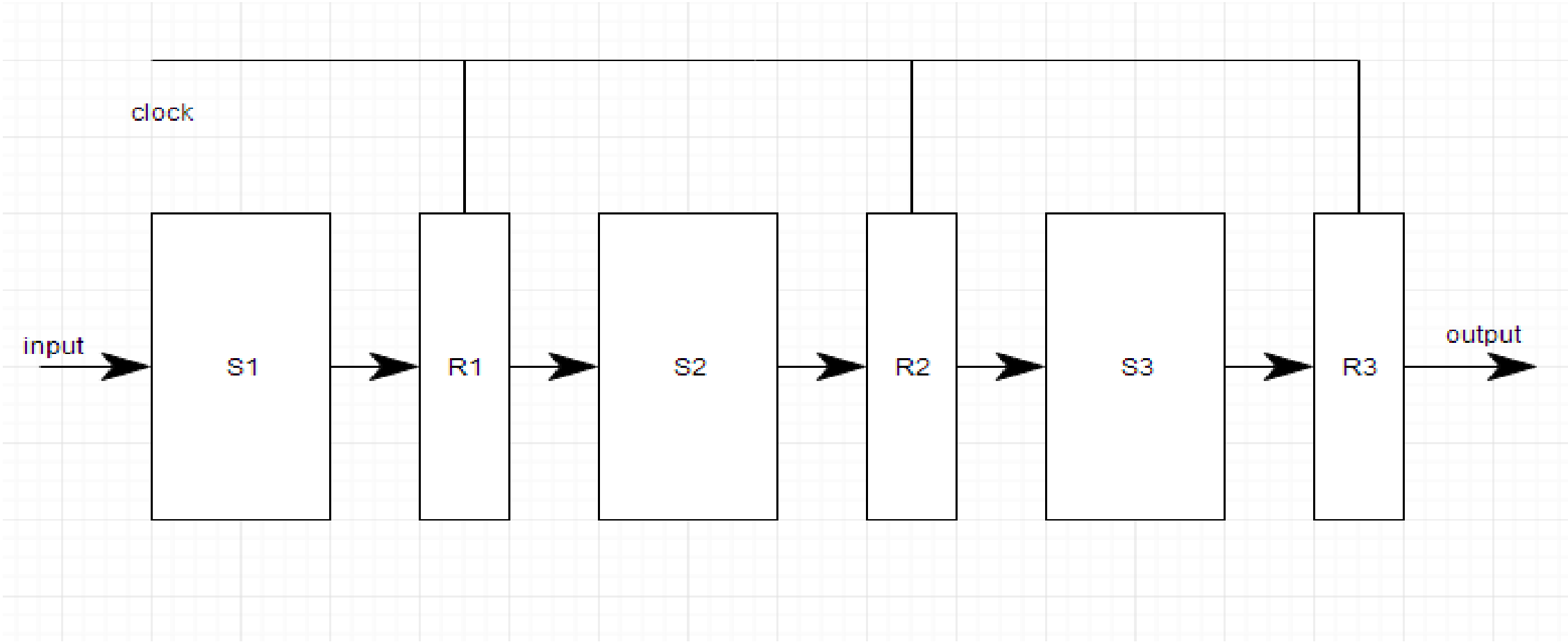
PIPELINING

- Pipelining is the process of accumulating instruction from the processor through a pipeline.
- It allows storing and executing instructions in an orderly process. It is also known as **pipeline processing**.
- Pipelining is a technique where multiple instructions are overlapped during execution.
- Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.
- Pipelining increases the overall instruction throughput.

PIPELINING CONT..

- In pipeline system, each segment consists of an input register followed by a combinational circuit. The register is used to hold data and combinational circuit performs operations on it.
- The output of combinational circuit is applied to the input register of the next segment.
- Pipeline system is like the modern day assembly line setup in factories.
- For example in a car manufacturing industry, huge assembly lines are setup and at each point, there are robotic arms to perform a certain task, and then the car moves on ahead to the next arm.
- **Pipelining** is an implementation technique where multiple instructions are overlapped in execution.
- The computer **pipeline** is divided in stages. Each stage completes a part of an instruction in **parallel**.
- **Pipelining** does not decrease the time for individual instruction execution. .

PIPELINING CONT..



INSTRUCTION PIPELINE

- In this a stream of instructions can be executed by overlapping *fetch*, *decode* and *execute* phases of an instruction cycle.
- This type of technique is used to increase the throughput of the computer system.
- An instruction pipeline reads instruction from the memory while previous instructions are being executed in other segments of the pipeline.
- Thus we can execute multiple instructions simultaneously.
- The pipeline will be more efficient if the instruction cycle is divided into segments of equal duration.

WHAT IS COMPUTER MEMORY?

- **Computer memory** is also known as "**Computer Storage Device**" help to store or saves of all important data such as songs, movies, pictures, software, and more.
- Those all data are saved in two different modes it can either temporary or permanent nature.
- All data are stored in **computer memory (computer storage device)** in the digital form such as binary form like as 0 and 1.
- Users can retrieves of saved instruction or information anytime when they are needed

TYPES OF COMPUTER MEMORY

- **Computer memory** plays vital role in the computer industry because without **computer memory** entire system like as plastic box.
- There are two types
- **Primary Memory (Storage Device)**
- **Secondary Memory (Storage Device)**

WHAT IS PRIMARY MEMORY(MAIN MEMORY)?

- **Primary memory is known as "Main Memory"** or **"Internal Memory"** or **"Primary Storage Device"** or **"Internal Storage Device"** as well as they play vital role in **computer**, because those memories are capable to access all data directly from **CPU** with the help of various buses.
- These memories have limited capacity for storage and made by integrated circuits (IC) or semiconductor components.
- **Primary storage devices** are available in two variance such as volatile and non volatile.
- Volatile memory is called temporary memory because all data deleted when power get turn off mode but its access time and response time much fine to secondary memory.
- Non volatile memory is permanent memory in which nothing data erase when system is turn off.

WHAT IS PRIMARY MEMORY(MAIN MEMORY)? CONT..

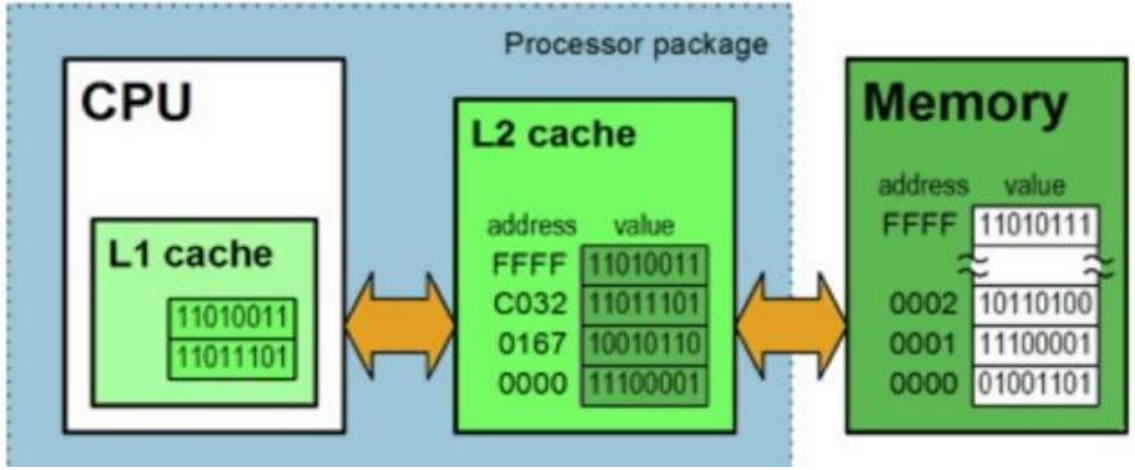
- The operating system and launched all application are loaded into **primary storage device (memory)** while turn on the computer because firstly CPU search all data in **primary memory (storage device)**.
- In this process, data transfer rate is very faster from CPU to RAM compare to transfer rate between CPU to Hard drive.
- So **Primary storage devices (memory)** are more costly compare to secondary memory.

WHAT IS SECONDARY MEMORY

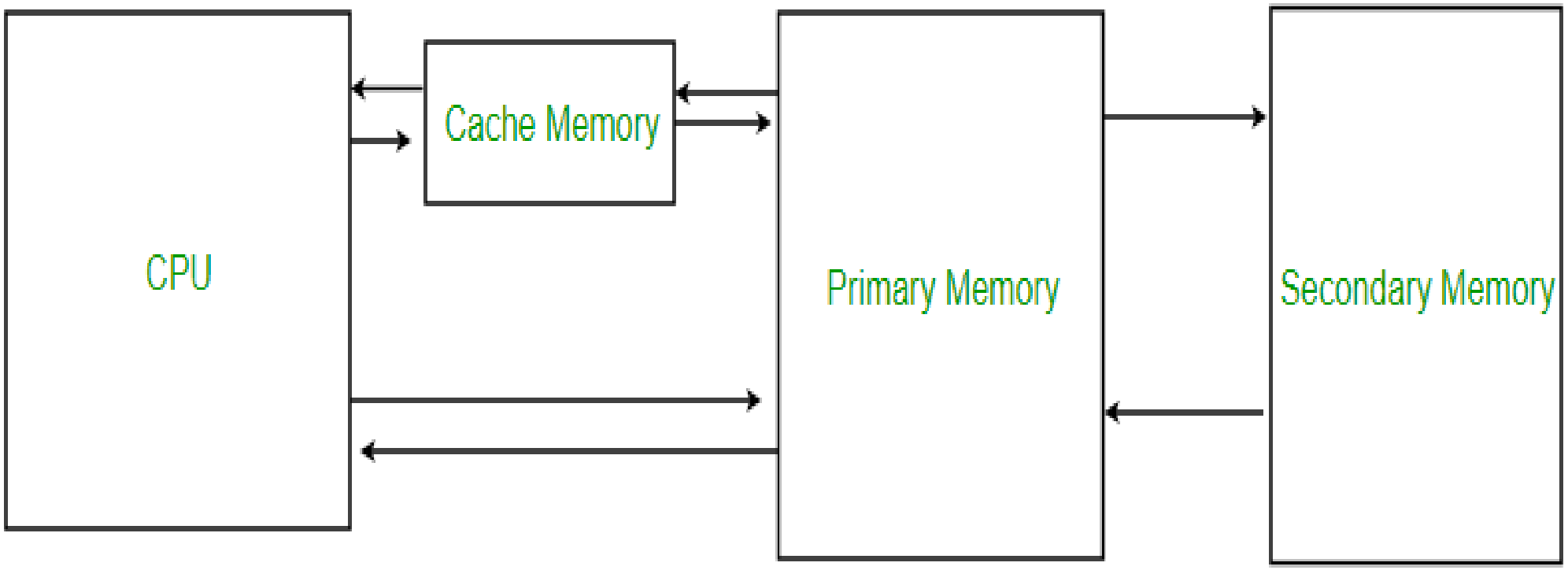
- **Secondary Memory** is also called "**storage device**" and "**auxiliary memory**","**external memory**".
- **Secondary storage devices** are non-volatile in nature, it means that data does not discard while power turn-off, in which all data store for long time.
- **Secondary memory** has the speed of access of data is very slow compare to **primary memory**, and cheaper as well.
- Without primary memory, those **secondary storage devices** are useless because for processing the secondary memory must be needed the **primary memory**, first of all data are transferred into **primary memory** then these data make for executable

Cache memory

Cache memory, also called CPU memory, is high-speed memory that a CPU can access more quickly than it can access regular random access memory (RAM). This memory is typically integrated directly into the CPU chip. The purpose of cache memory is to store program instructions and data that are used repeatedly in the operation of programs or information that the CPU is likely to need next.



CACHE MEMORY CONT..



Primary and Secondary Memory Comparison

Primary memory	Secondary memory
Fast	Slow
Expensive	Cheap
Low capacity	Large capacity
Connects directly to the processor	Not connected directly to the processor


Thank you!