Lecture-3

Chapter-2

Computer Architecture and Organization-
Jhon P. Hayes

Design Methodology

Design Methodology

System Design:

A computer is an example of a system, which is defined informally as a collection – often a large and complex one – of objects called components, that are connected to form a consistent entity with a specific function or purpose. The function of the system is determined by the functions of its components and how the components are connected.

If mathematical function f maps element a of A (input information items) onto element b of B (output information items), we write b = f (a).

Design Methodology

System Representation:

Figure 2 shows a block diagram representing a small gate-level logic circuit called an EXCLUIVE or **modulo-2 adder**. This circuit has the same general form as Figure 1 that exhibiting a graph, which is a useful way of representing a system with the help of vertices and edges.

Structure versus behaviors:

Figure 3 shows one kind of behavioral description for the logic circuit of Figure 2. This tabulation of all possible combinations of input output values is called a truth table. Another description of the same Exclusive OR behavior can be written in terms of mathematical equations as follows, noting that f (a) = $f(x_1, x_2)$

$$f(0,0) = 0$$
$$f(0,1) = 1$$
$$f(1,0) = 1$$
$$f(1,1) = 0$$

Design Methodology (Structure versus behaviors)

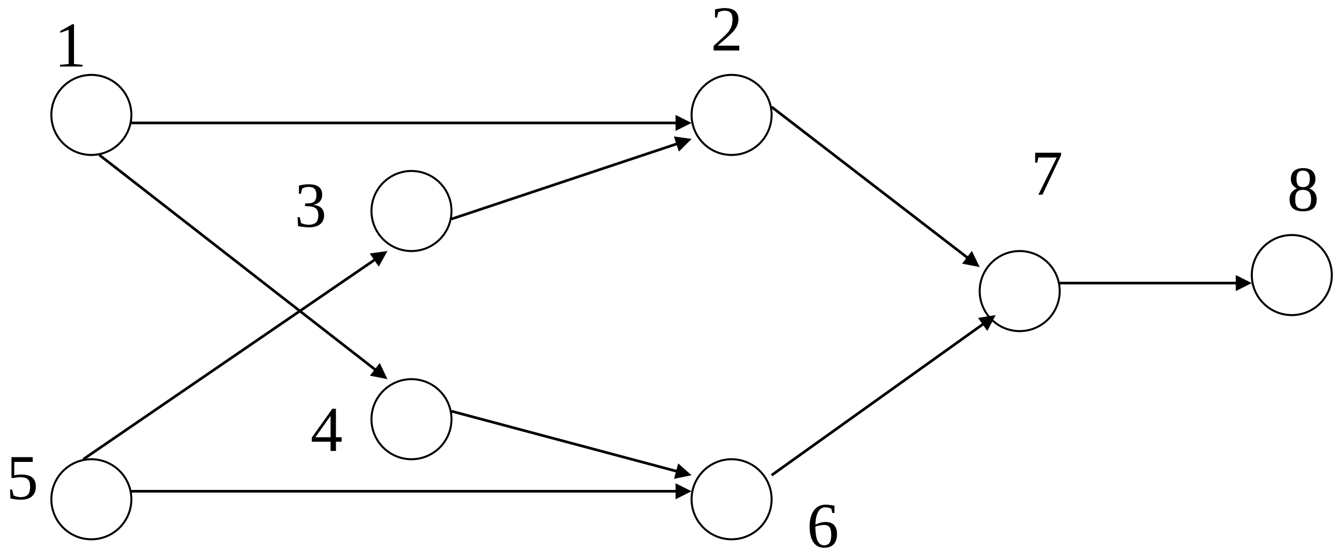


Figure 1: A graph with eight nodes and nine edges

Design Methodology (Structure versus behaviors)

Input a		Output
x_1	x_2	f(a)
0	0	0
0	1	1
1	0	1
1	1	0

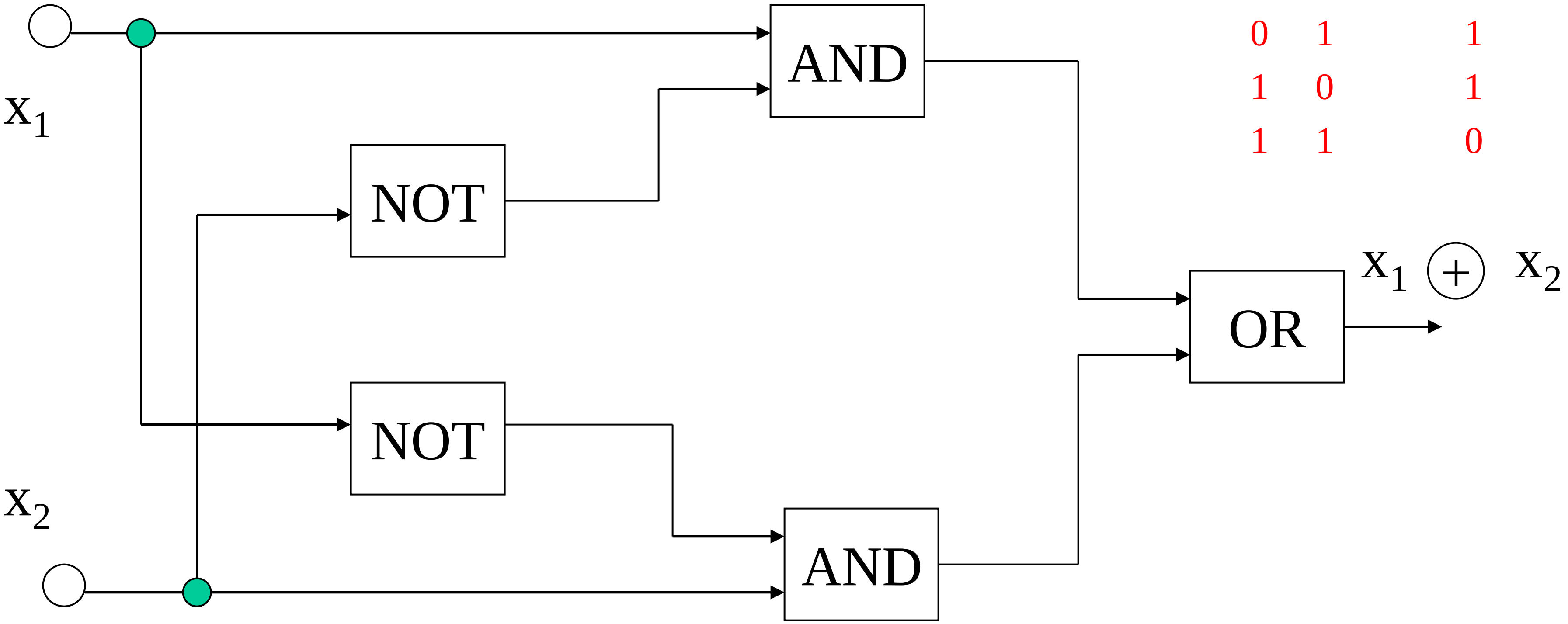


Figure 2: A block diagram representing an Exclusive–OR logic circuit

Design Methodology (Structure versus behaviors)

Input a		Output
x_1	x_2	f(a)
0	0	0
0	1	1
1	0	1
1	1	0

Figure 3: Truth table for the EXCLUSIVE-OR function

Design Methodology

Hardware description language: We can fully describe a system’s structure and behavior by means of a block diagram. The term schematic diagram is also used, in which it identify the functions of the components. We can convey the same detailed information by means of a *hardware description language* (HDL), a format that resembles a high-level programming language such as Ada or C.

Design Methodology

Design Process: Given a system's structure, the task of determining its function or behavior is termed analysis. The converse problem of determining a system structure that exhibits a given behavior is design or synthesis.

The complexity of computer systems is such that the design problem must be broken down into smaller, easier task involving various classes of components. Each smaller part can be solve independently by different designers and each major design step can be implemented by iterative process as shown in Figure 4

Design Methodology

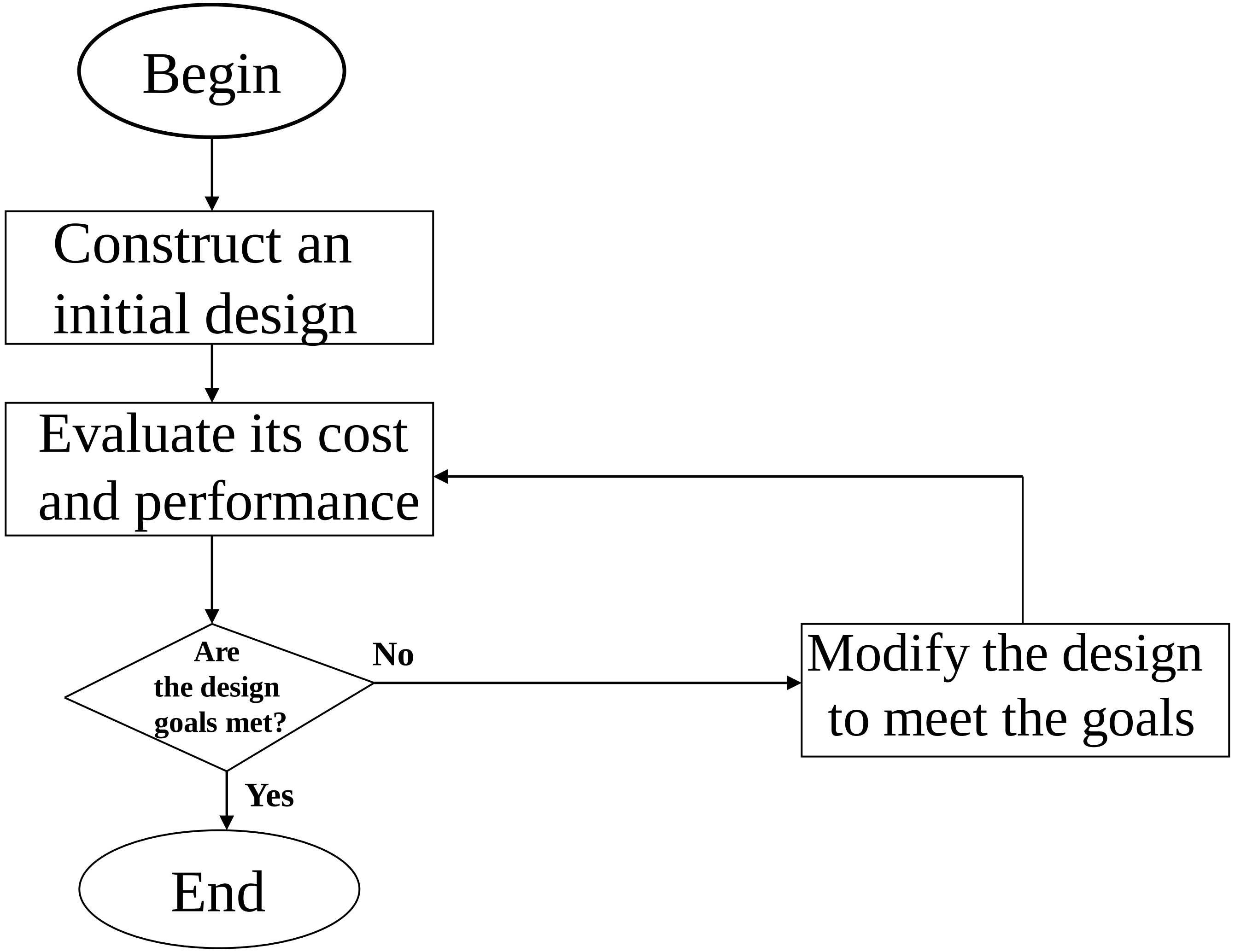
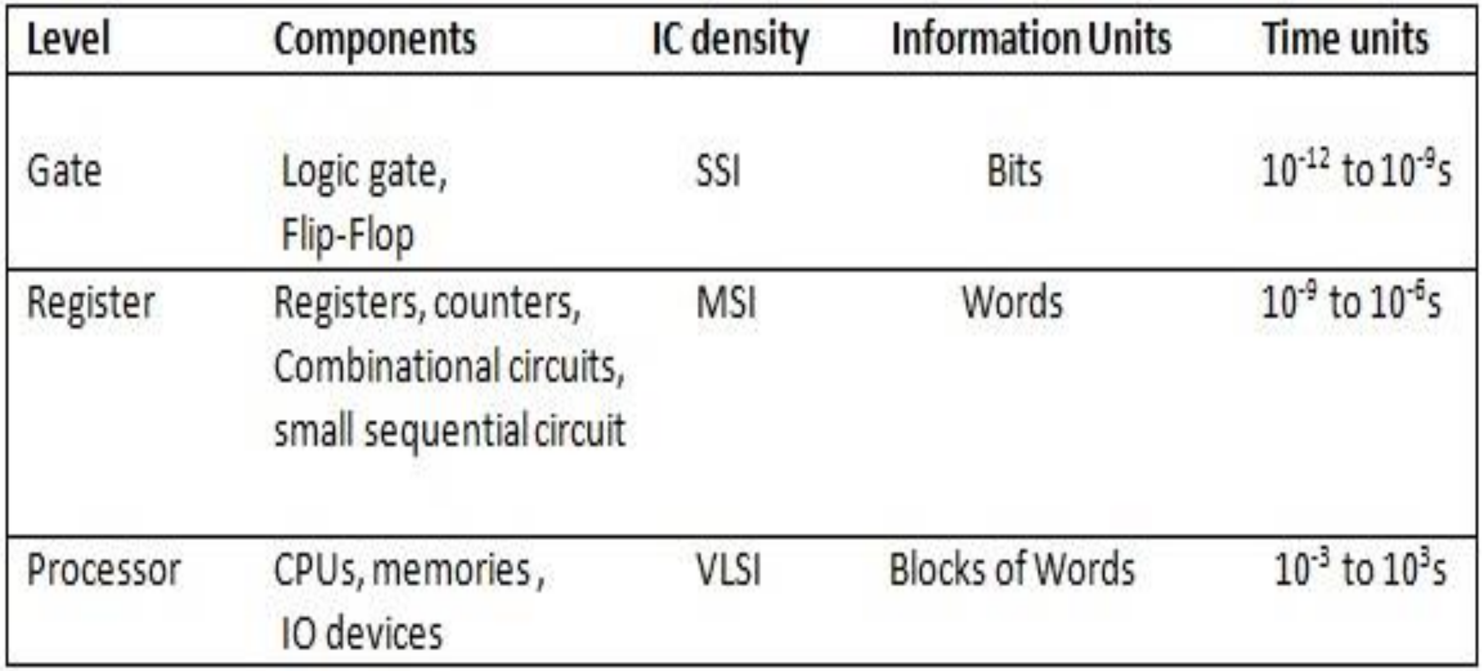

Begin
Construct an initial design
Evaluate its cost and performance
Are the design goals met?
No
Modify the design to meet the goals
Yes
End

Figure 4 Flowchart of an iterative design process

Design Methodology

Design levels:

1. The *processor level*, also called the architecture, behavior, or system level
2. The *register level*, also called the register-transfer level (RTL).
3. The *gate level*, also called the logic level



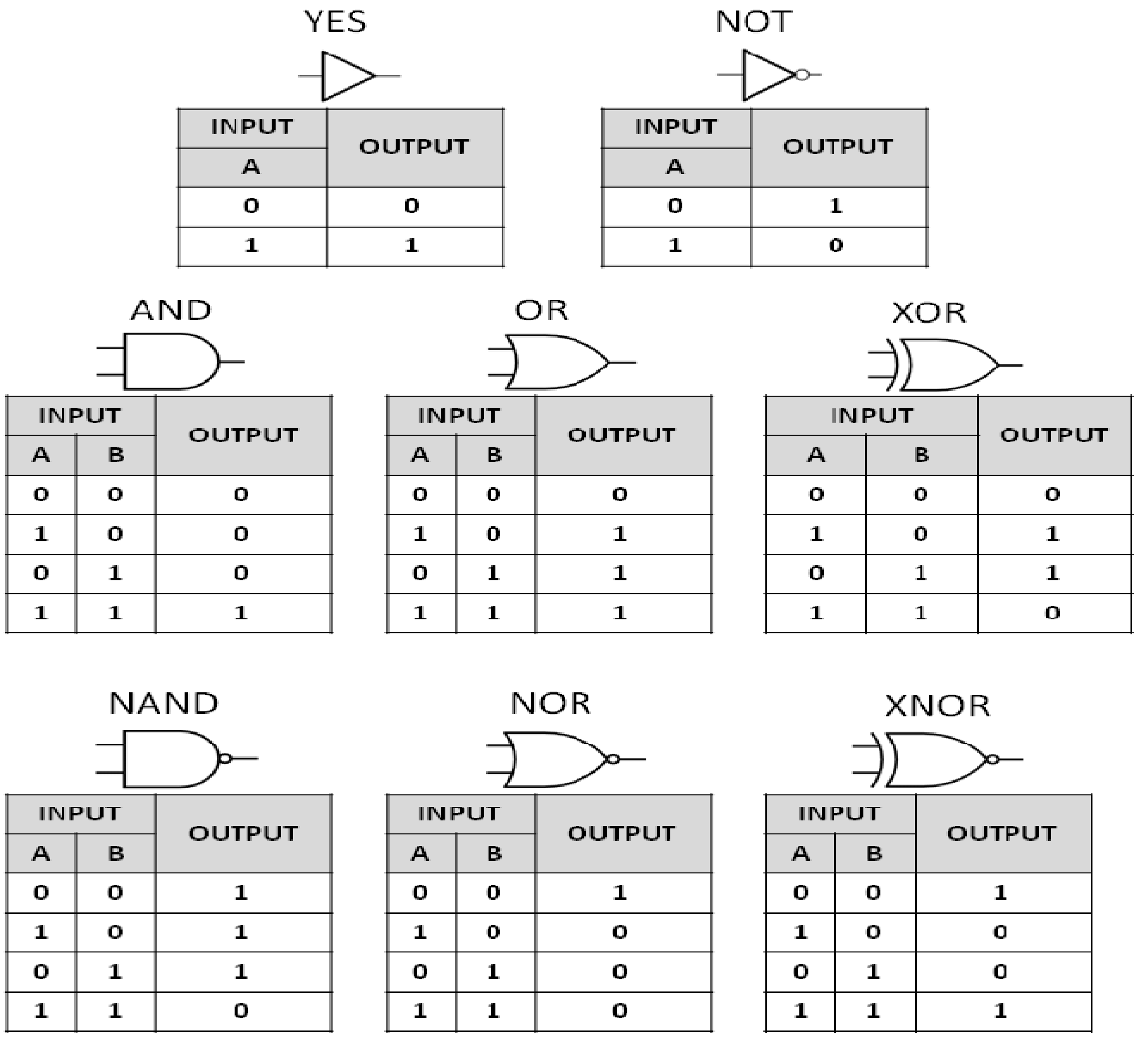
Level	Components	IC density	Information Units	Time units
Gate	Logic gate, Flip-Flop	SSI	Bits	10^{-12} to 10^{-9}s
Register	Registers, counters, Combinational circuits, small sequential circuit	MSI	Words	10^{-9} to 10^{-6}s
Processor	CPUs, memories , IO devices	VLSI	Blocks of Words	10^{-3} to 10^{3}s

The Gate Level

Gate level (logic) design is concerned with processing binary variables whose possible values are restricted to be the bits (binary digits) 0 and 1. The design components are logic gates, which are simple, memoryless processing elements and flip-flops which are bit-storage devices.

Combinational logic: A combinational function also referred to as a logic or a Boolean function, is to mapping from the set of 2^n input combinations of n binary variables onto the output values 0 and 1.

Common Logic Gates



YES

INPUT	OUTPUT
A	
0	0
1	1

NOT

INPUT	OUTPUT
A	
0	1
1	0

AND

INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

Fig 5: Common Boolean logic gates with symbols and truth tables

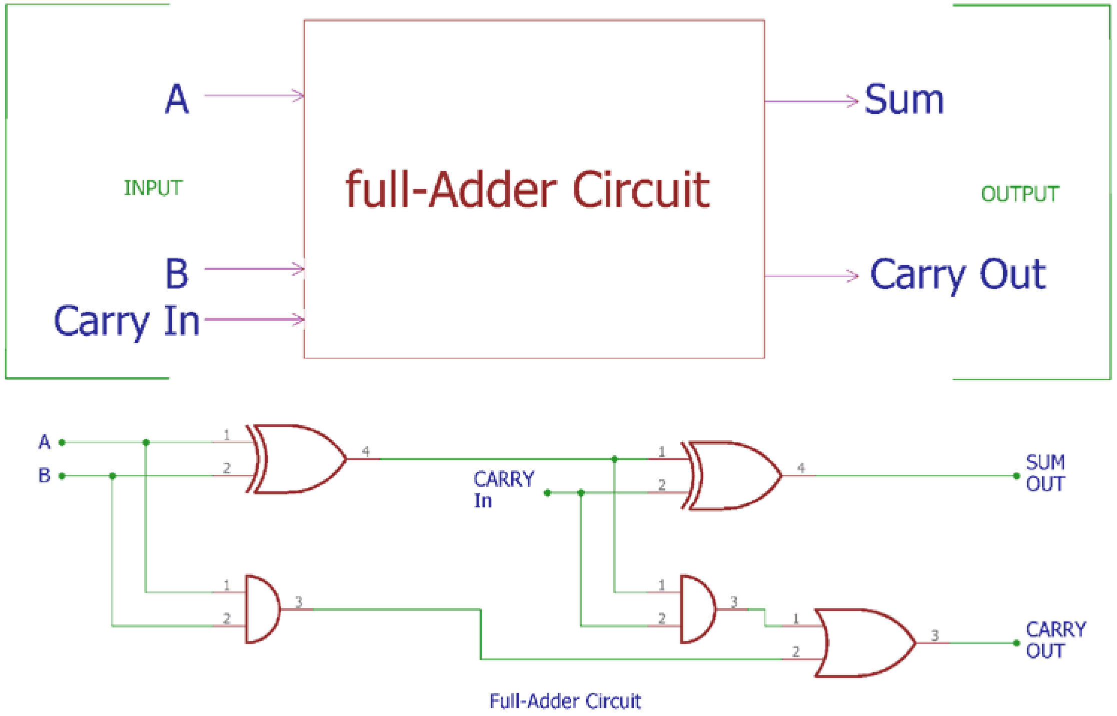
The Gate Level

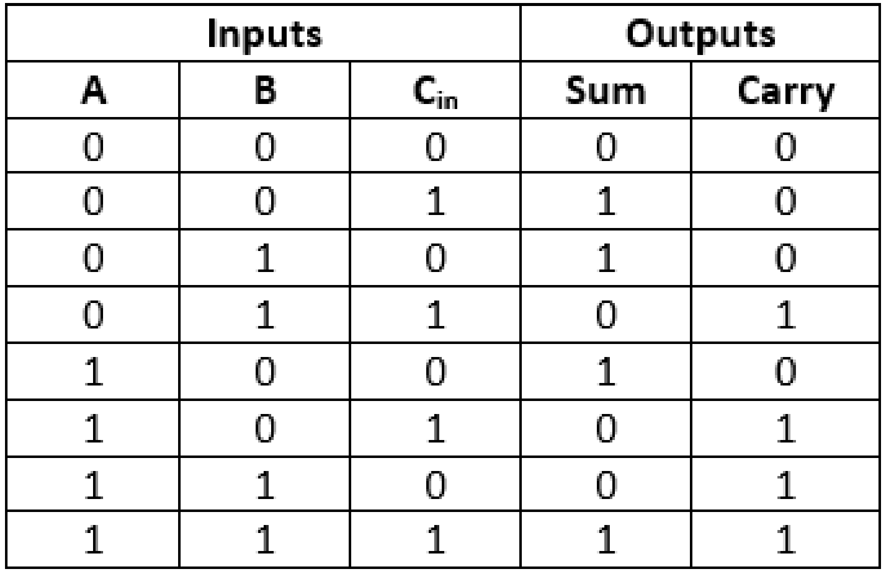
Figure 6 shows the truth table for a pair of three variable functions, $s_0(x_0, y_0, c_{-1})$ and $c_0(x_0, y_0, c_{-1})$, which are the sum and carry output, respectively, of a logic circuit called a ***full adder***. This useful logic circuit computes the numerical sum of its three input bits using binary (base 2) arithmetic:

$$c_0s_0 = x_0 \textbf{\textit{ plus }} y_0 \textbf{\textit{ plus }} c_{-1}$$

For example, the last row of the truth table of Figure 6 express the fact that the sum of three 1s is $c_0s_0 = 11_2$ that is the base-2 representation of the number three (3).

The Gate Level





Inputs			Outputs	
A	B	C_{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 6: Full Adder with circuit and truth table

The Gate Level

Sum of Product (SOP) and Product of Sum (POS)

$s_0 = x_0y_0c_{-1} + x_0y_0c_{-1} + x_0y_0c_{-1} + x_0y_0c_{-1}$ -------- (1)

Equation (1) represents ***sum-of-products (SOP)***.

$c_0 = (x_0 + c_{-1})(x_0 + y_0)(y_0 + c_{-1})$ -------- (2)

Equation (2) represents ***product-of-sum (POS)***.

Flip-flops

By adding memory to a combinational circuit in the form of 1-bit storage elements called flip-flops, we obtain a sequential circuit. Flip-flops rely on an external clock signal CK to synchronize the time at which they respond to change on their data lines. They are also designed to be unaffected by transient signal changes (noise) produced by the combinational logic that feeds them.

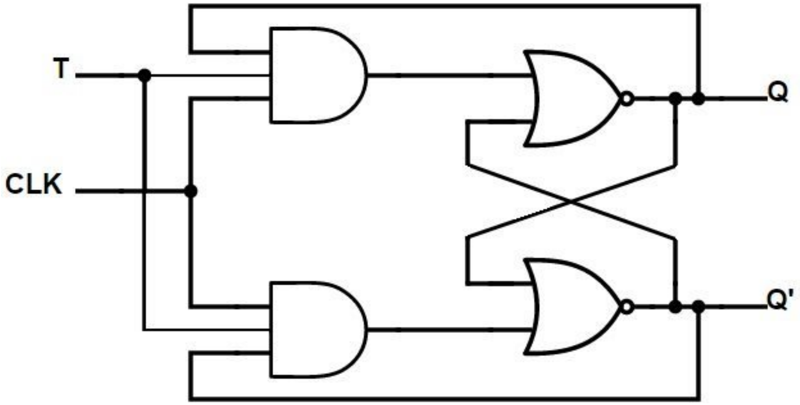


Fig 7: Flip-flops

Flip-flops

Four bit ripple carry: a) logic structure; b) high level symbol.

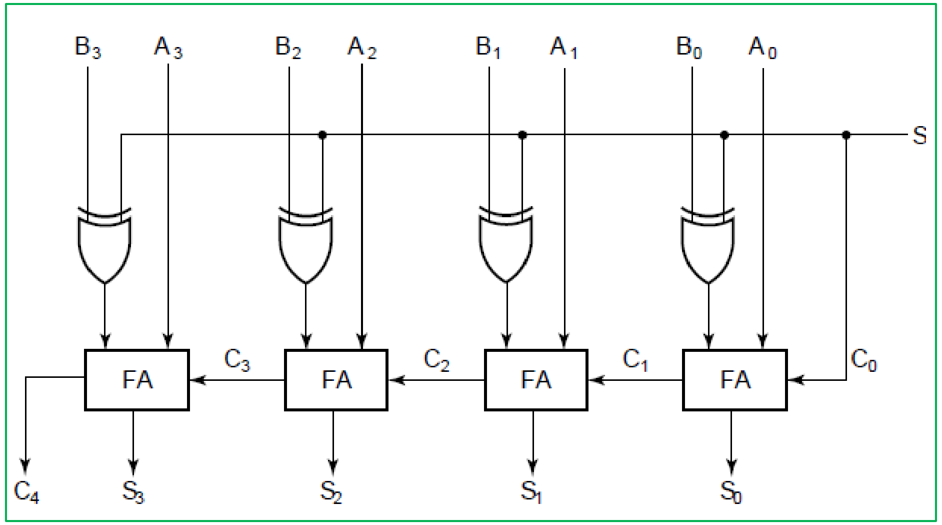
A_3 A_2 A_1 A_0

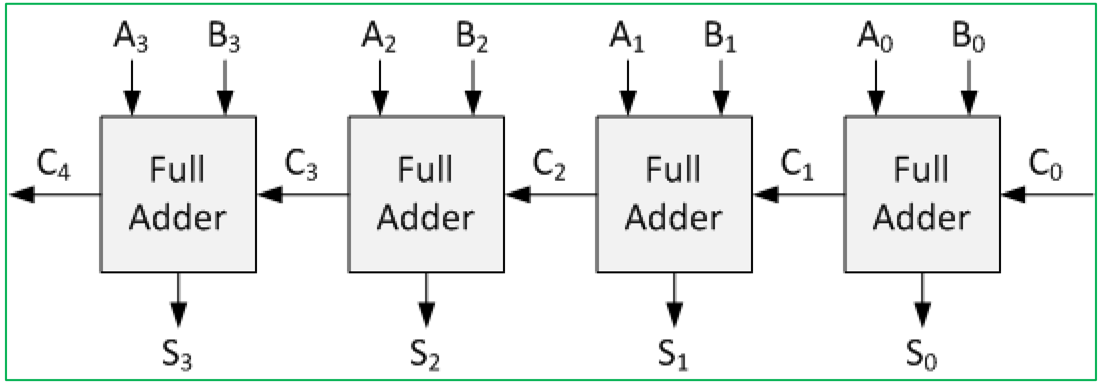
1 0 1 0 = 10

B_3 B_2 B_1 B_0

1 0 1 1 = 11

1 0 1 0 1 = 21





Use to add two n-bits binary numbers

Inputs			Outputs	
A	B	C_{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Flip-flops

Sequential circuits: A sequential circuit consists of a combinational circuit and a set of flip-flops. The combinational logic forms the computational or data processing part of the circuit. The flip-flops store information of the circuit’s past behavior; this stored information defines the circuits internal state. (output depends on present input and past input)

The Register level

At the register or register transfer level, related information bits are grouped into ordered sets called words or vectors. The primitive (Modern) components are small combinational or sequential circuits intended to process or store words.

Type	Component	Functions
Combinational	Word Gates	Logical (Boolean) Operation
	Multiplexers	Data routing; general, combinational function
	Decoder, encoder	Code checking and conversion
	Adders	Addition and subtraction
	Arithmetic Logic Unit	Numerical and logical operations
	Programmable logic devices	General sequential functions.
Sequential	(Parallel) registers, Shift registers	Information storage; serial-parallel conversion
	Counter, Programmable logic devices	Control/timing signal generation, General sequential functions.

Fig 8: Register level Components

The Processor level

There are four main groups of processor level components:

- Processors
- Memories
- IO devices
- Interconnection networks

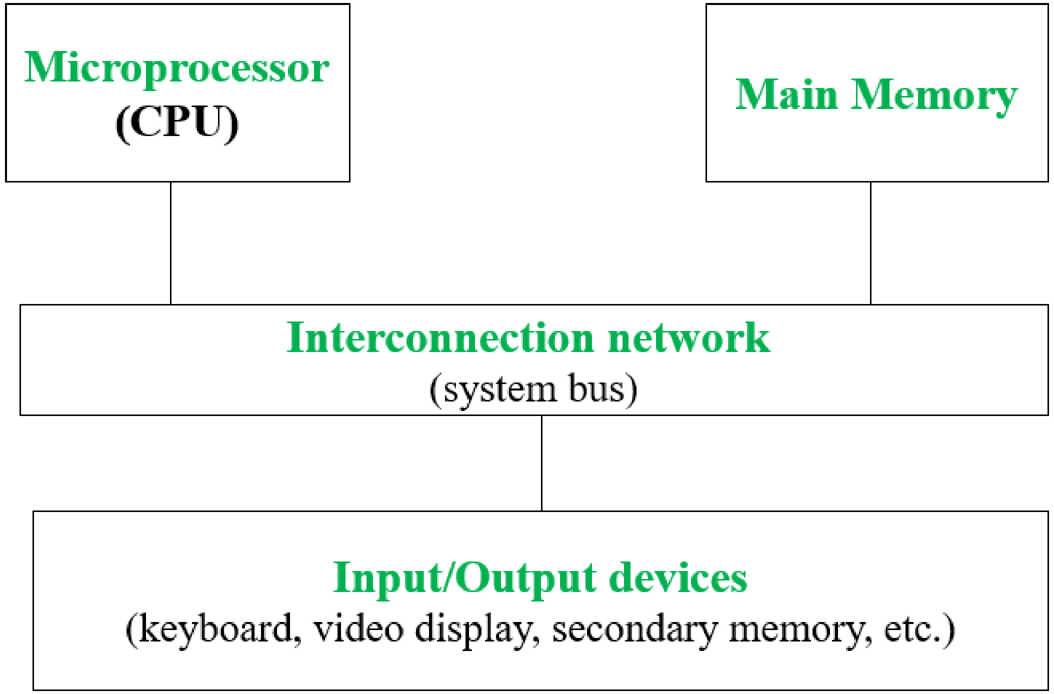


Figure: Major components of a computer system

The Processor level

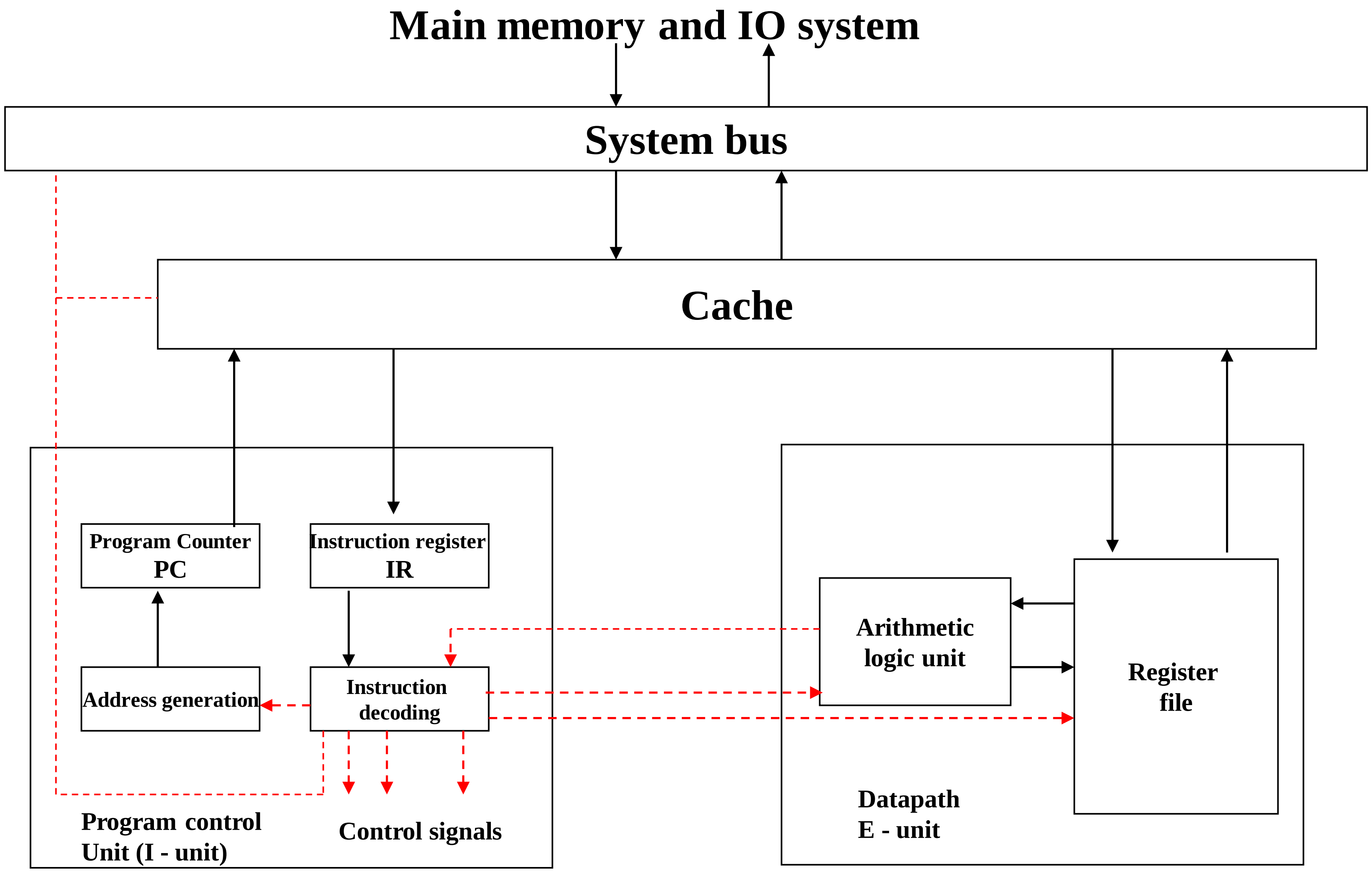


Figure 10: Internal organization of a CPU and cache memory

The Processor level

- Figure 10 shows the essential internal organization of a CPU at the register level. The CPU contains the logic needed to execute its particular instruction set and is divided into datapath and control unit.
- The control part (***I - unit***) generates the address of instructions and data stored in external memory. In this particular system a cache memory is interposed between the main memory M and the CPU.
- Each memory request generated by the CPU is first directed to the cache.

The Processor level

- If the required information is not currently assigned to the cache, the request is redirected to M and cache is automatically updated from M.
- The I – unit fetches instructions from the cache or M and decodes them to derive the control signals needed for their execution.
- The CPU's datapath (***E - unit***) has the arithmetic – logic circuits that execute most instructions; it has also a set of registers for temporary data storage.
- The CPU manages a ***system bus***, which is the main communication link among the CPU cache subsystem, main memory and the I/O devices.

The Processor level

- In one clock cycle the CPU can perform a register-transfer operation, such as fetching an instruction word from M via the system bus and loading it into the instruction register IR. This operation can be expressed formally by

  $$IR := M(PC);$$

  Where PC is the program counter used by the CPU to hold the expected address of the next instruction word. Once in the I-unit, an instruction is decoded to determine the actions needed for its execution.
- The entire process of fetching, decoding and executing an instruction constitutes the CPU's ***instruction cycle***.

The Processor level

Memories: The memory part of a computer can be divided into several major subsystem.

- *Main memory M,* consisting of relatively fast storage ICs connected directly to, and controlled by, the CPU.

- *Secondary memory,* Consist of less expensive devices that have very high storage capacity. These devices often involve mechanical motion and so are much slower than M. They are generally connected indirectly (via M) to the CPU and form part of the computer's I/O system.

- *Other memory,* many computer have a third type of memory called a cache, which is positioned between the CPU and main memory. Some or all of cache may be integrated on the same IC chip as the CPU itself.

The Processor Level (Cache memory)

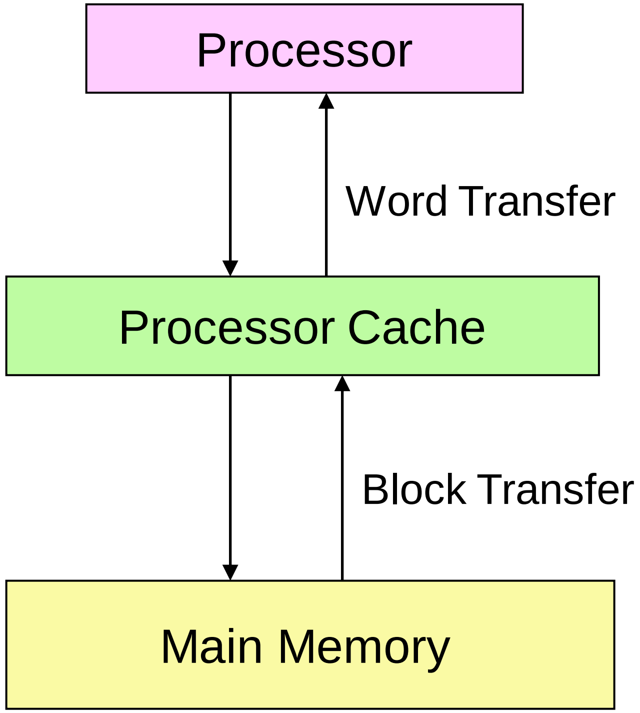


Fig 11: Processor Cache Memory

Cache memory is a small amount of fast but expensive memory placed between the processor and main memory (512 KByte on the P-III processor). In some PCs there may be extra cache attached directly to the motherboard. When the processor attempts to read a word of main memory, a check is made first to determine if the word is in the cache. If it is, a copy of the word is transferred to the processor. If not, a block of main memory, consisting of a fixed number of words, is transferred into the cache and then a copy of the required work is transferred to the processor.

There are two types of cache memory:

- External cache memory,
- Internal cache memory

The Processor level

I/O devices: Input-output devices are the means by which a computer communicates with the outside world.

- A primary function of I/O devices is to act as data transducers.
- I/O devices do not alter the information content or meaning of the data on which they act.
- Since data is transferred and processed within a computer system in the form of di gital electrical signals, input (output) devices transform other forms of informatio n to (from) digital electrical signal.

The Processor level

Interconnection networks: Processor level components communicate by word-oriented buses. In systems with many components, communication may be controlled by a subsystem called an *interconnection network*; terms such as *switching network*, *communications controller*, and *bus controller* are also used in this context.

Internal Communication:

The System BUS: A bus is a set of parallel wires connecting two or more components of the computer.

The number of lines is referred to as the width of the bus. Bus lines are classified into three functional groups:

–Control Bus

–Data Bus

–Address Bus

The Processor Level (System Bus)

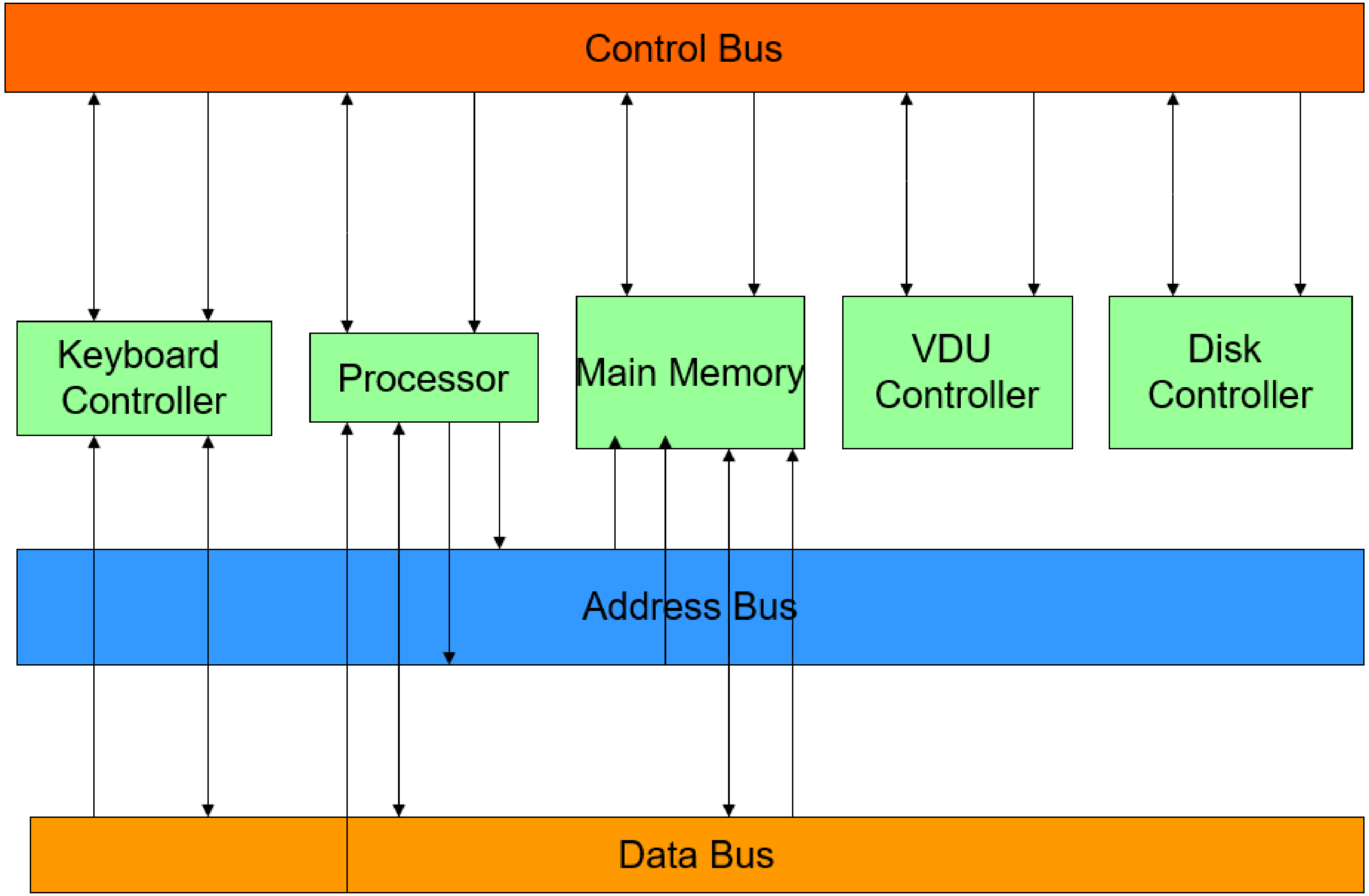


Fig 12: System Bus

The Processor Level (Internal Communication)

Interrupts: An interrupt is a signal from some device/source seeking the attenti on of the processor.

Types of Interrupts

- Program
- Timer
- Input / Output
- Hardware failure

That’s All
Thank You