Digital Systems

Universidad de Guanajuato

(FIMEE)

Master of Electrical Engineering

D.Phil. Eduardo CABAL-YEPEZ

1 Introduction

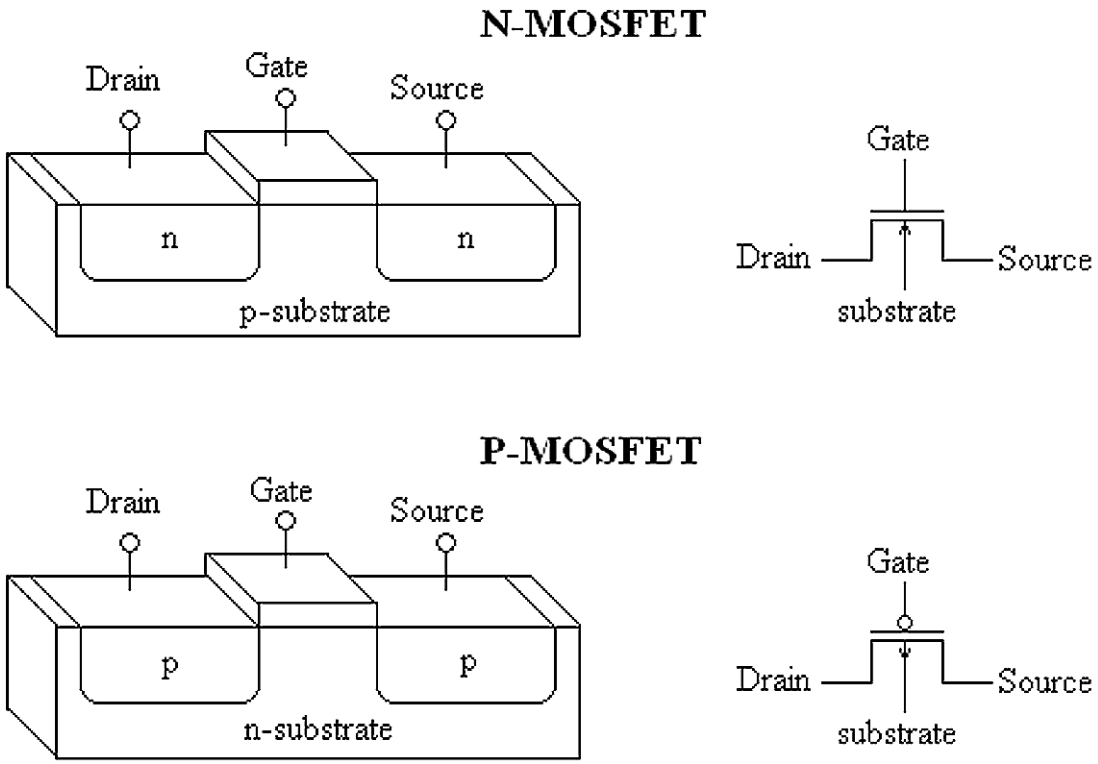
Universidad de Guanajuato

(FIMEE)

Master of Electrical Engineering

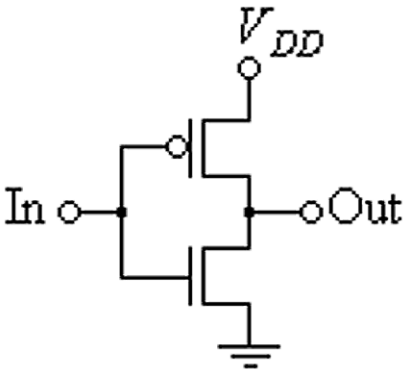
Abstraction Levels

- Transistor Level

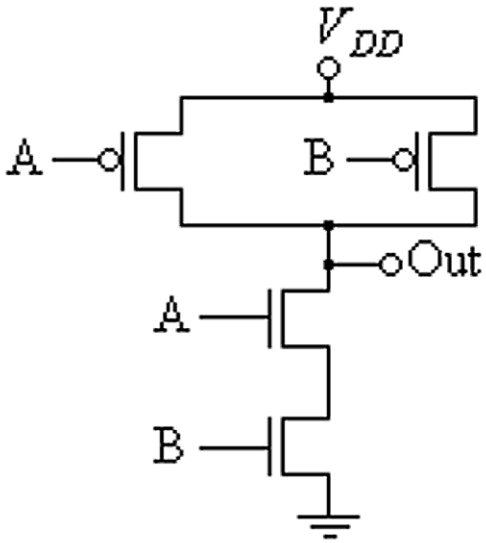


Abstraction Levels

- Circuit Level



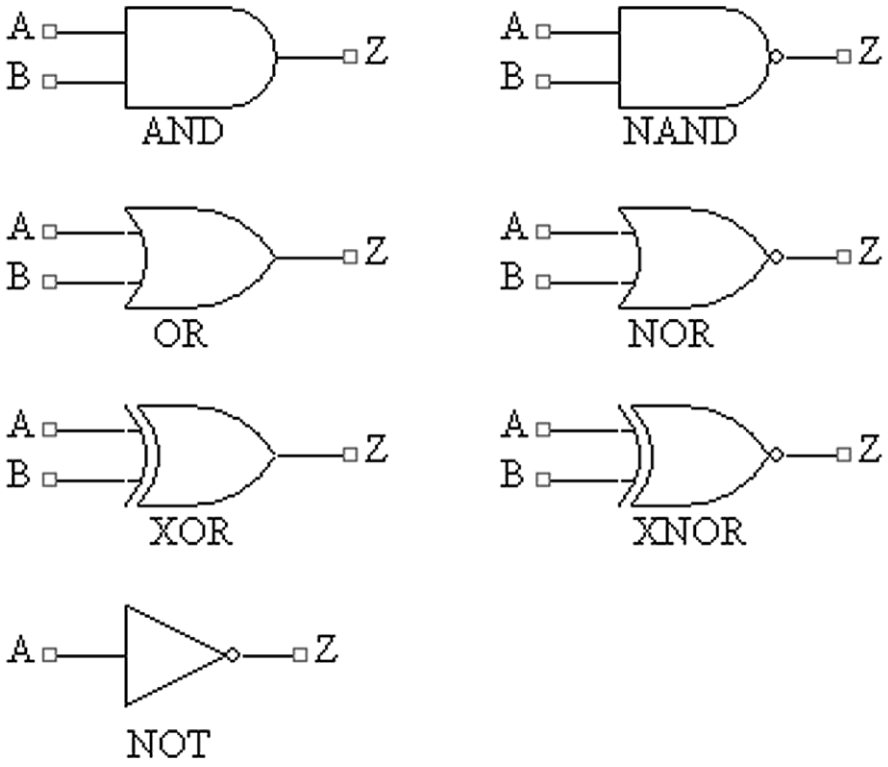
CMOS-Inverter



CMOS-NAND

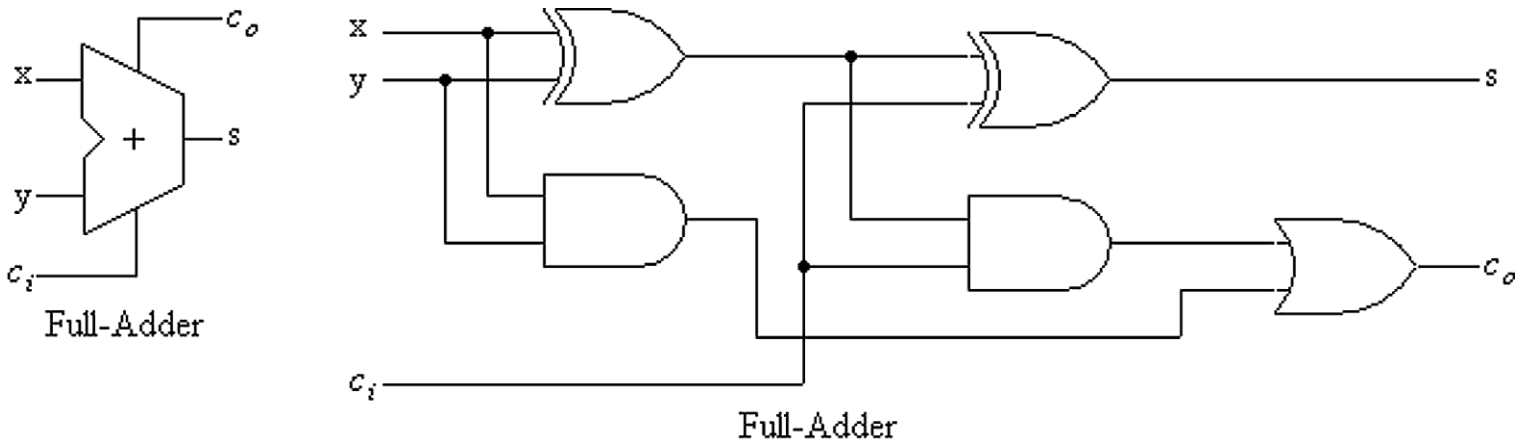
Abstraction Levels

- Gate Level



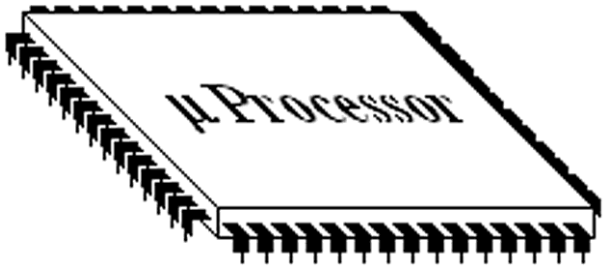
Abstraction Levels

- Block or Module Level



Abstraction Levels

- Digital System



Digital Systems

- A digital system uses discrete values to represent information for input, processing, transmission, storage, etc.
- It has several combinational and sequential blocks working together for carrying out all these tasks.

Synchronous Digital System

- Purely synchronous
- One Trigger Edge Synchronization
- Derived Synchronization.

Purely Synchronous Digital System

- In a purely-synchronous digital system, all of its building sequential machines are synchronous in a Moore architecture, with a common clock for all of them, and all its memory elements are identical.

One Trigger Edge Synchronization

- A one trigger edge synchronous system has a Moore architecture, where the memory elements respond to the different trigger edges of a common clock.

Derived Synchronization

- A digital system has *derived synchronization* when its building states machines are synchronous in a Moore architecture, and has multiple clocks derived synchronously from one master clock.

Asynchronous Digital System

- An asynchronous digital system has multiple clock signals not synchronized by a global clock.

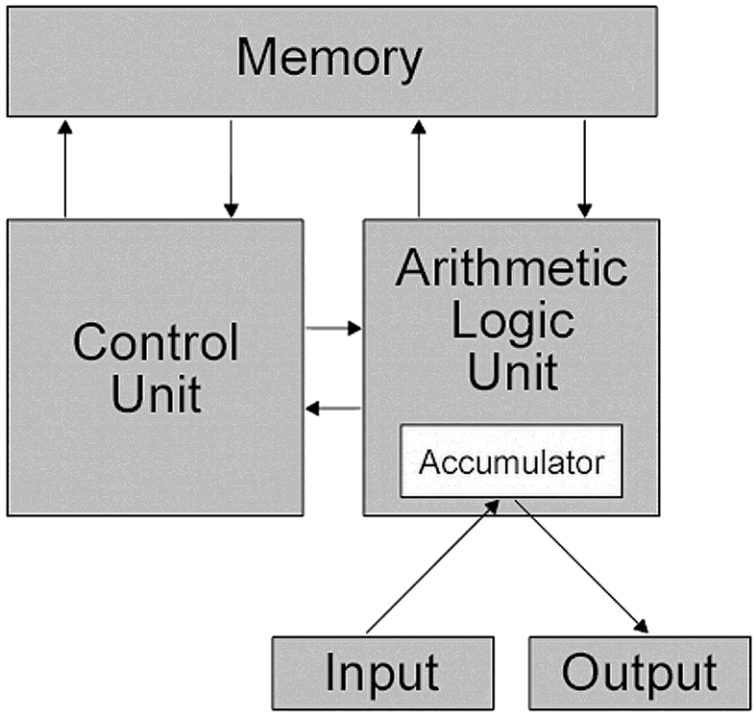
Microprocessor Architectures

- Von Neumann Architecture
- Harvard Architecture

Von Neumann Architecture

- Data and programs are stored together in memory
- Follows the following pattern
  - ***Fetch***: Instructions and necessary data are obtained from memory.
  - ***Decode***: Instructions and data are separated.
  - ***Execute***: Instructions are performed, data is manipulated, results are stored

Von Neumann Architecture



Harvard Architecture

- Independent memory for data and instructions.
- Modified Hardware Architecture:
  - Allows words from the instruction memory to be treated as read-only data.

Instruction Set Architecture (ISA)

- Part of the computer architecture related to programming
  - CISC (Complex Instruction Set Computing)
  - RISC (Reduced Instruction Set Computing)

CISC

- Each instruction can execute several low-level operations in a single instruction.
- Instructions might take long time to execute.
  - Several memory cycles.

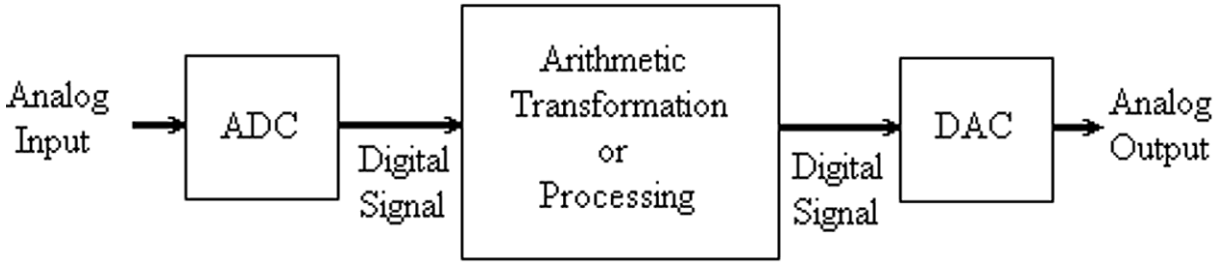
RISC

- Execute small instructions in fast way.
- A RISC chip has:
  - Fewer transistors dedicated to the core logic.
  - General purpose registers.
  - Uniform instruction format.

Other ISAs

- Very Long Instruction Word (VLW)
  - A processor that executes every instruction one after the other.
- Minimal Instruction Set Computing (MISC)
  - Very small number of basic operations.
- Zero Instruction Set Computing (ZISC)
  - Based on pattern matching.
  - Massively hardwired parallel processing.

Digital Signal Processing in Hardware



1. Analog signal to digital domain.
2. Arithmetic Transformation.
3. Back to the analog domain.

Digital Signal Processor (DSP)

- Vehicle for implementing digital-signal-processing functions.
  - Real-time processing.
  - Separate program and data memories.
  - Special Instructions for SIMD.
  - Process digital signals converted from analog signals.
  - Provide an analog output from the processed digital signal.

Application Specific Integrated Circuits (ASICs)

- Hard-wired arithmetic logic blocks and state machines.
  - Inflexible.
  - Hardware design at transistor level.
  - High cost.
  - Long time to market.

Multi core DSPs

- Very-long-instruction-word engines.
- Great deal of parallelism.
- Gain for each added engine, less 100%.
- Difficult to program.
- Application specific DSP.
- More expensive than ASICs

Programmable Logic Devices (PLDs)

- Infinitely customizable.
- Silicon physical-design ready for customization.

Field Programmable Gate Arrays (FPGAs)

- Array of uncommitted circuits elements.
- Spatial computations.
- End-user programmable.

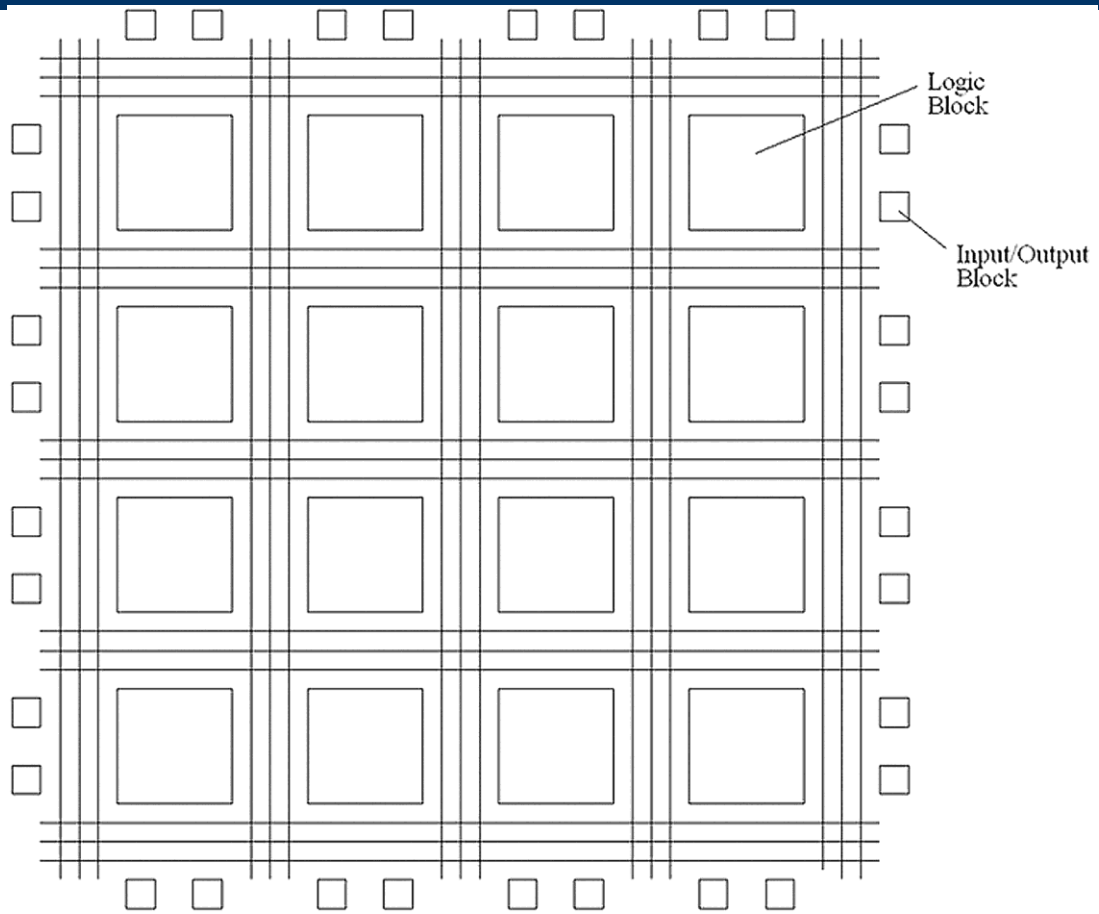
FPGAs vs ASICs & DSPs

FPGAs vs ASICs	FPGAs vs Processors
●More power consumption. ●Slower. ●Looser. ●Commercially available. ●Fast time to market.	●Inflexible. ●Slower. ●Lower power consumption. ●Faster. ●Tighter.

FPGA architecture

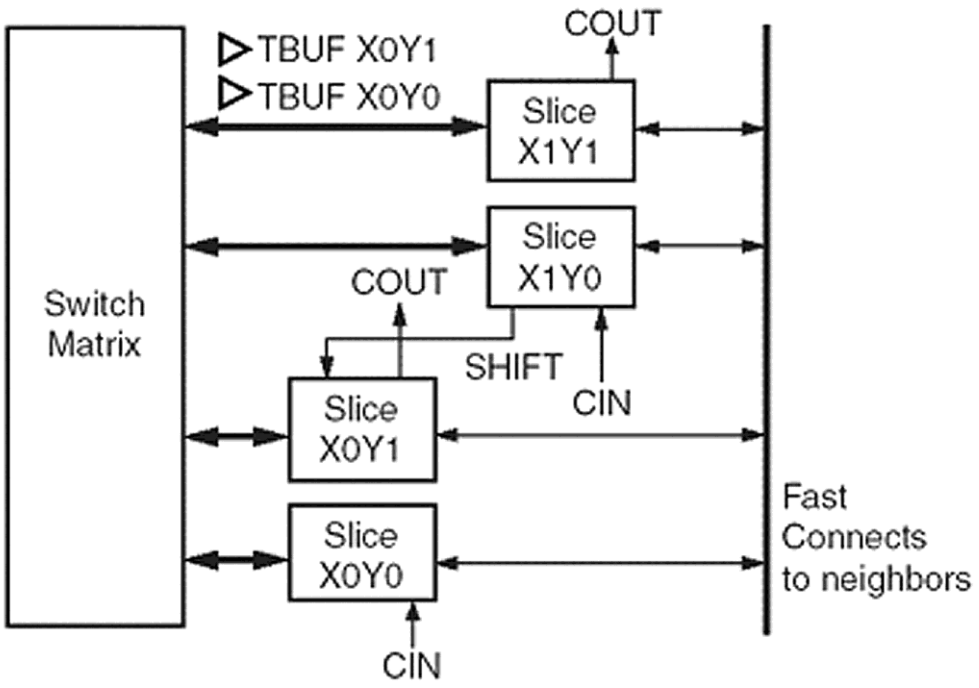
- Flexible, regular programmable architecture surrounded by a perimeter of programmable Input/Output blocks, which are interconnected by a hierarchy of routing channels.

FPGA internal structure

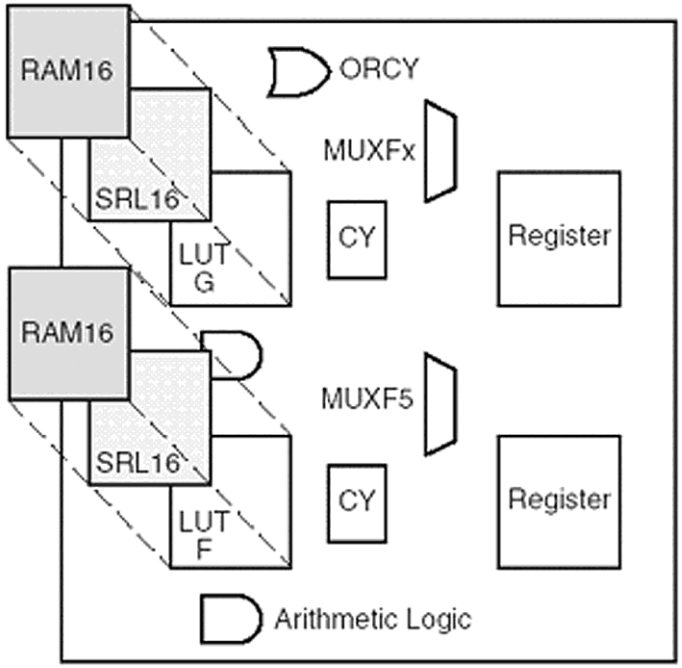


FPGA Configurable Logic Blocks (CLBs)

CLB



Slice



FPGA technology

- Antifuse:
  - Configured by burning a set of fuses.
  - Once configured, it can not be altered any more.
- Flash:
  - Can be re-programmed several times.
  - Non-volatile.
- SRAM:
  - Unlimited reprogramming.
  - Very fast configuration
  - Partial configuration
  - Volatile.

Hardware Description Languages (HDLs)

- Design tool for describing digital systems.
  - CUPL (Universal compiler for programmable logic, Logical Devices).
  - ABEL (Advanced Boolean Expression Language, Data I/O corporation).
  - Verilog (VERIfy LOGic, IEEE standard 1364-2001).
  - VHDL (Very high speed integrated circuit Hardware Description Language, IEEE standard 1076-1993)

VHDL

- Initiated in 1981 by the US DoD.
- Language with wide range capability
- Independent of technology or design methodology.
- Development of baseline language (1983-85).
- All rights transferred to IEEE (1986).
- Publication of the IEEE standard (1987)
- Revised Standard named VHDL 1076-1993 (1994)

VHDL features

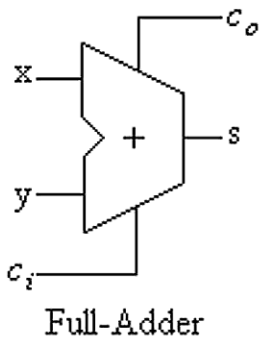
- Is concurrent i.e. all operations are executed simultaneously.

C-code segment
x=2;
y=3*x+2;
z=x+y;
	x=2;
	y=8;
	z=10;

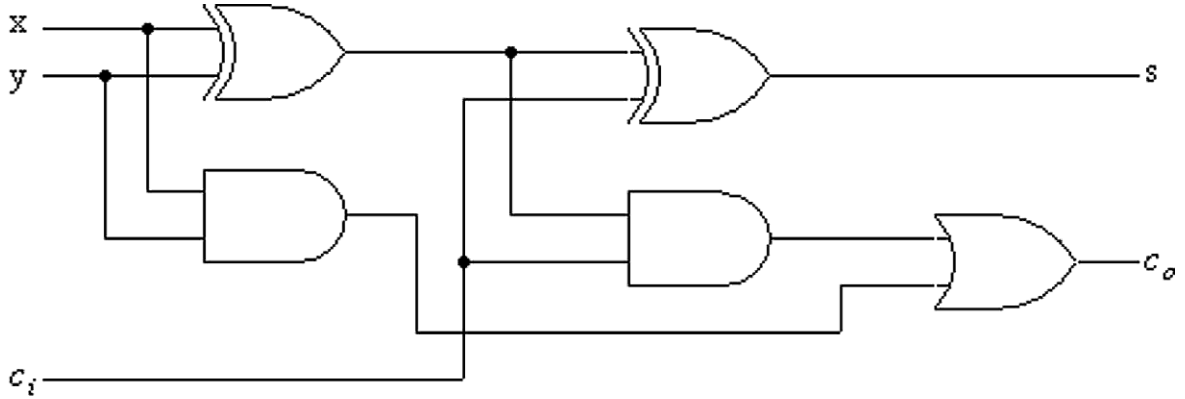
VHDL-description segment x<=z; y<=‘1’; z<=Y XOR ‘0’;	x<=‘1’; y<=‘1’; z<=‘1’;

VHDL features (cont.)

- Synthesis depends on the used structures. i.e. the final implementation depends on the inferred structure.



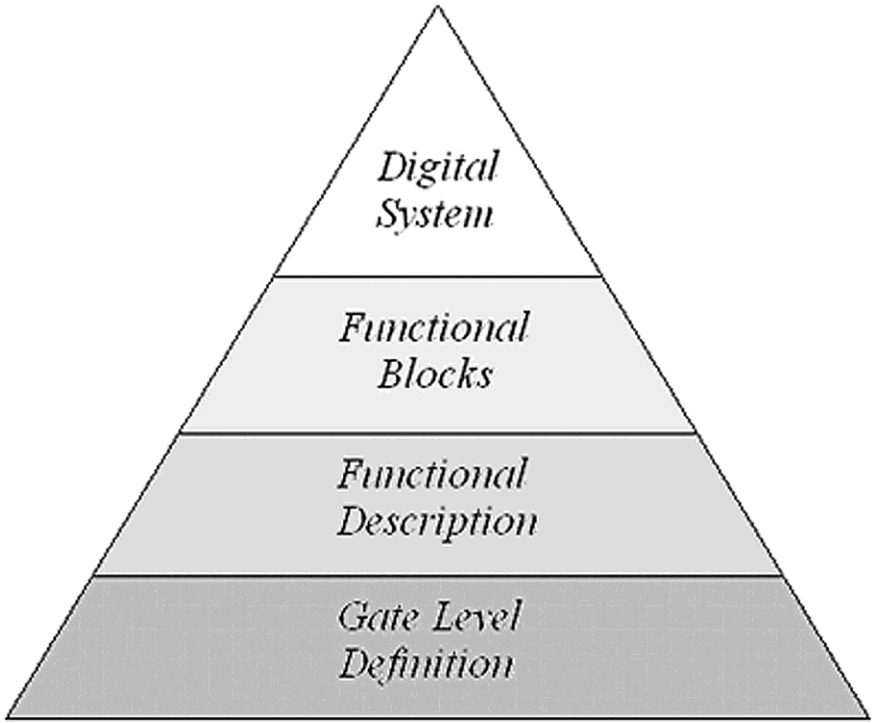
Full-Adder



Full-Adder

VHDL features (cont.)

- Highly hierarchical.



VHDL elements

1. Library declarations.
2. Port declarations.
3. Architecture description.
4. Test Bench.

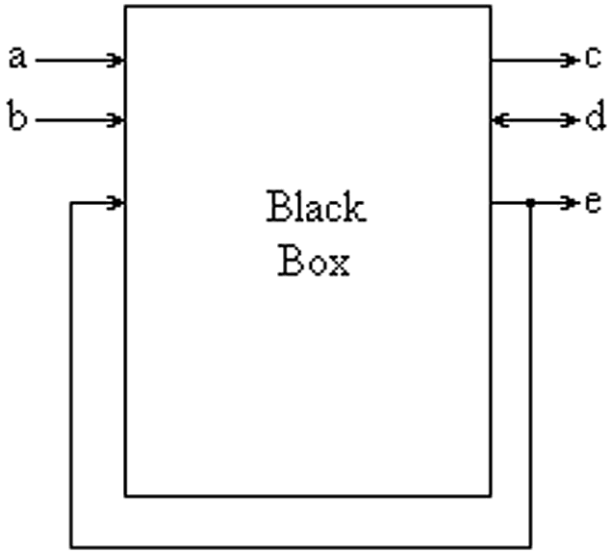
VHDL library declaration.

```
Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
```

VHDL port declaration

```
entity Black_box is
port(
   a, b : in  std_logic; --Single input
   c     : out std_logic; --Single output
   d     : inout std_logic; -- bidirectional port
   e     : buffer std_logic -- fedback output
);
end Black_box;
```

VHDL port declaration (cont.)



VHDL declaration

Valid	Invalid
A	8
B5	b%
Counter_10	R__5
AND_gate_1	Gate_
Logic_System	XOR
S32C5	S32 C5

Simple Circuit Description

```
Library IEEE;
use IEEE.std_logic_1164.all;

entity AND_gate is
port(
  A, B : in  std_logic;
  F    : out std_logic
);
end AND_gate;

architecture behavioral of AND_gate is
begin
  F <= A AND B;
end behavioral;
```

Simple Gates

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity simple_gates is
port(
  A : in  std_logic;
  B : in  std_logic;
  F : out std_logic_vector(1 to 7)
);
end simple_gates;

architecture Behavioral of simple_gates is
begin
  F(1) <= NOT A;
  F(2) <= NOT B;
  F(3) <= A NAND B;
  F(4) <= A NOR B;
  F(5) <= A AND B;
  F(6) <= A OR B;
  F(7) <= A XOR B;
end Behavioral;
```