DIGITAL ELECTRONICS

VHDL

Objectives

- At the end of this lecture the student should be able:
- To understand why we need HDLs,
- To understand the application of HDLs

Introduction

- Today design of digital systems is done by HDLs such as VHDL or Verilog unlike the traditional way of manually connecting discrete components on printed circuit boards.
- VHDL is a hardware description language (HDL) and not a programming language, though its format is similar to popular programming languages.
- To be able to manage complex designs quickly and without errors, the use of VHDL software packages becomes inevitable.

CAD for Hardware Description

- There are many CAD packages for describing hardware but the most popular and widely used are:

➢ VHDL translated as very high speed integrated circuits hardware description language.

➢ Verilog

HDLs

- A hardware description language is inherently parallel i.e. its commands, which correspond to logic gates are executed in parallel as soon as a new input arrives.
- It also allows the incorporation of timing specifications (gate delays).
- It also describes the system as an interconnection of different components but all done inside a chip.

Why do we need HDLs?

- To automate much of the traditional design and debugging of a digital system methodology.
- For specifying, modeling, designing, synthesis, documenting and simulating digital systems.

VHDL

- What is VHDL?

*V*HSIC → *Very High Speed Integrated Circuit*

*H*ardware

*D*escription

*L*anguage

History of VHDL

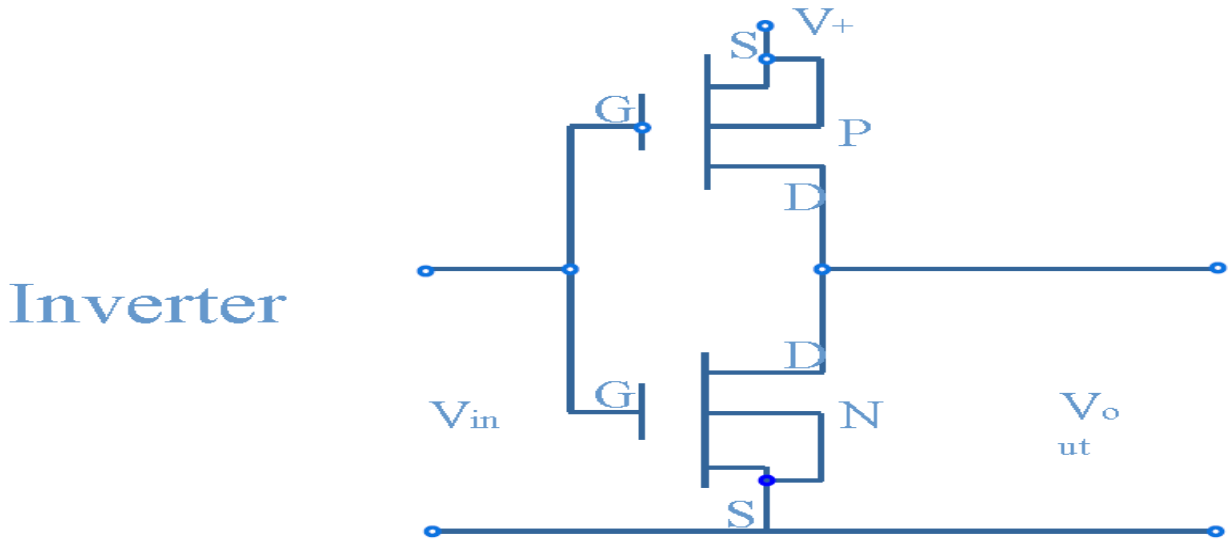
- Designed by IBM, Texas Instruments, and Intermetrics and was funded by American Department of Defense (DoD).
- Standardized by the IEEE in 1987: IEEE 1076-1987
- Enhanced version of the language defined in 1993: IEEE 1076-1993
- Additional standardized packages provide definitions of data types and expressions of timing data
  - IEEE 1164 (data types)
  - IEEE 1076.3 (numeric)
  - IEEE 1076.4 (timing)

Traditional vs. Hardware Description Languages

- Procedural programming languages provide the *how* or recipes
  - for computation
  - for data manipulation
  - for execution on a specific hardware model
- Hardware description languages *describe* a system
  - Systems can be described from many different points of view
    - Behavior: what does it do?
    - Structure: what is it composed of?
    - Functional properties: how do I interface to it?
    - Physical properties: how fast is it?

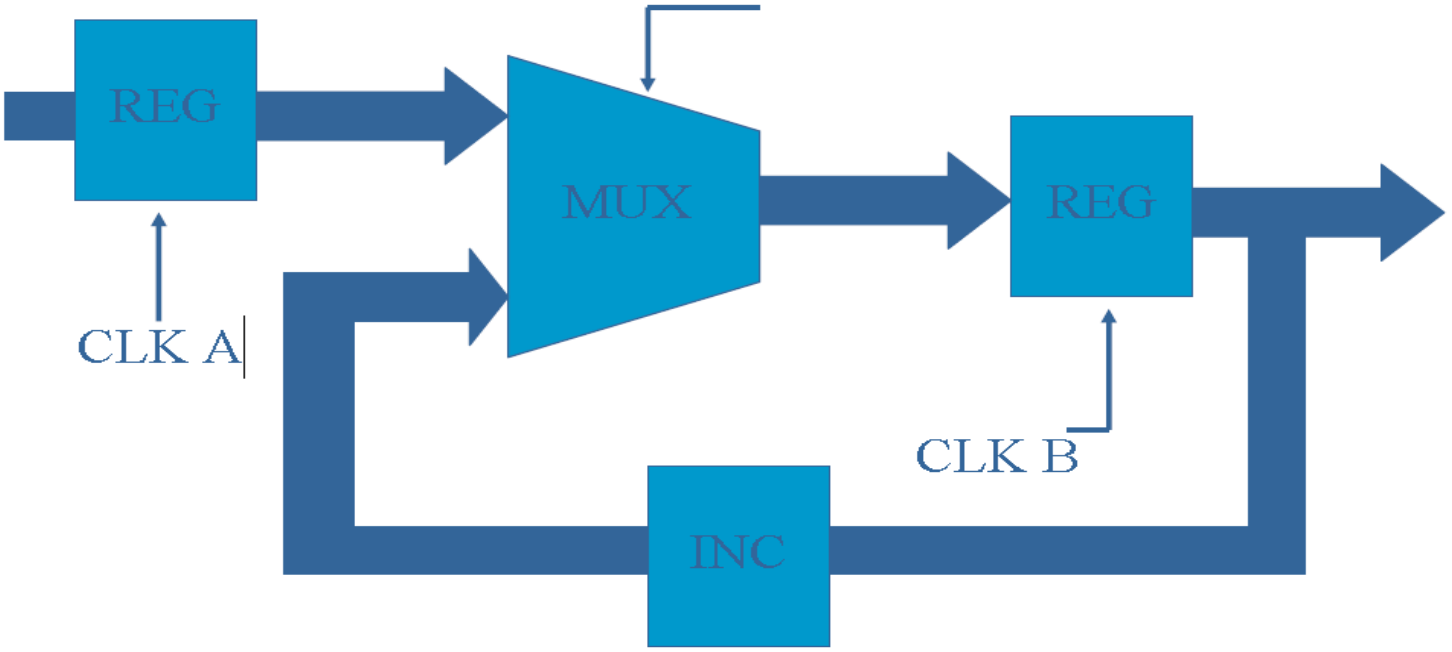
Physical Level

- Descriptions can be done at different levels of abstraction
  - physical level: model switching behavior of transistors



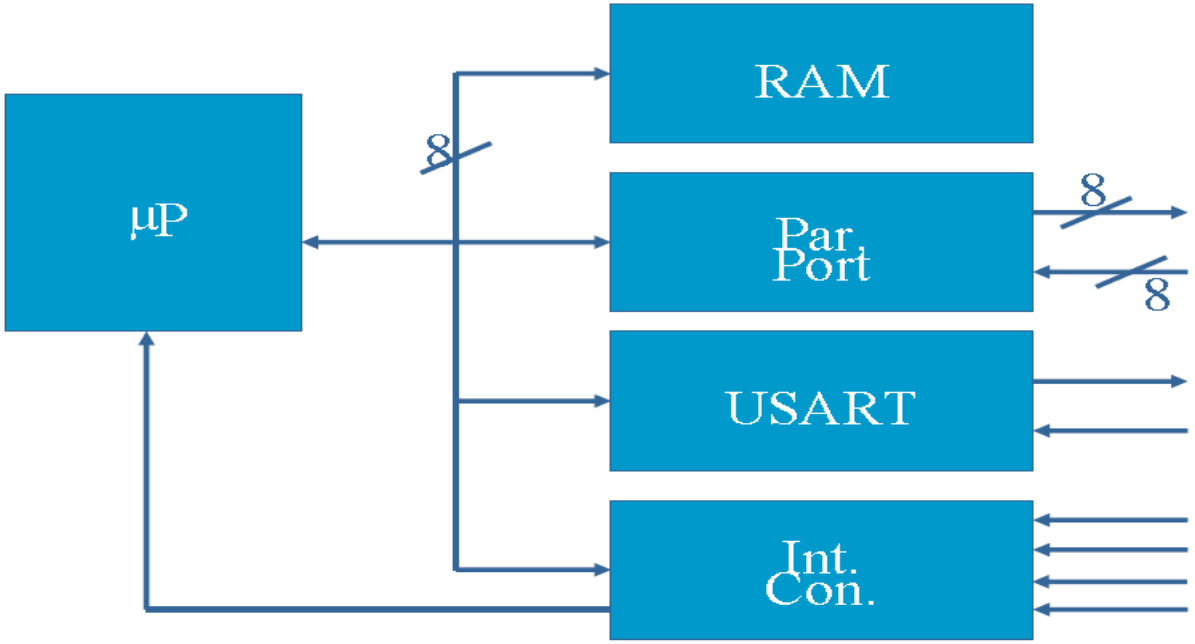
Register Transfer Level

- Register transfer level (RTL): model combinational and sequential logic components



System Level

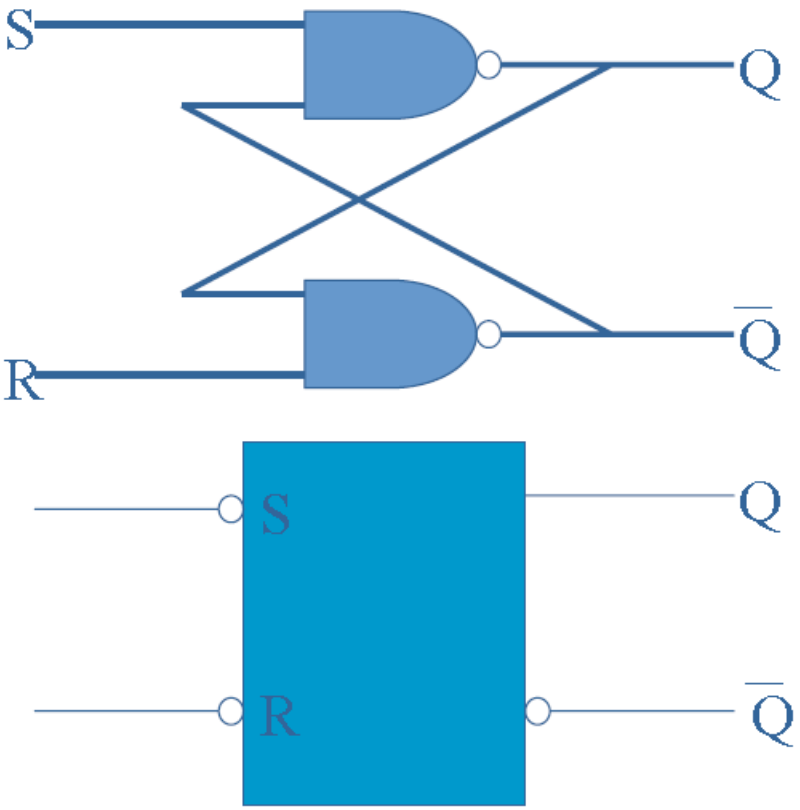
- Instruction set architecture level: functional behavior of a microprocessor



Gate Level

- Synthesis –

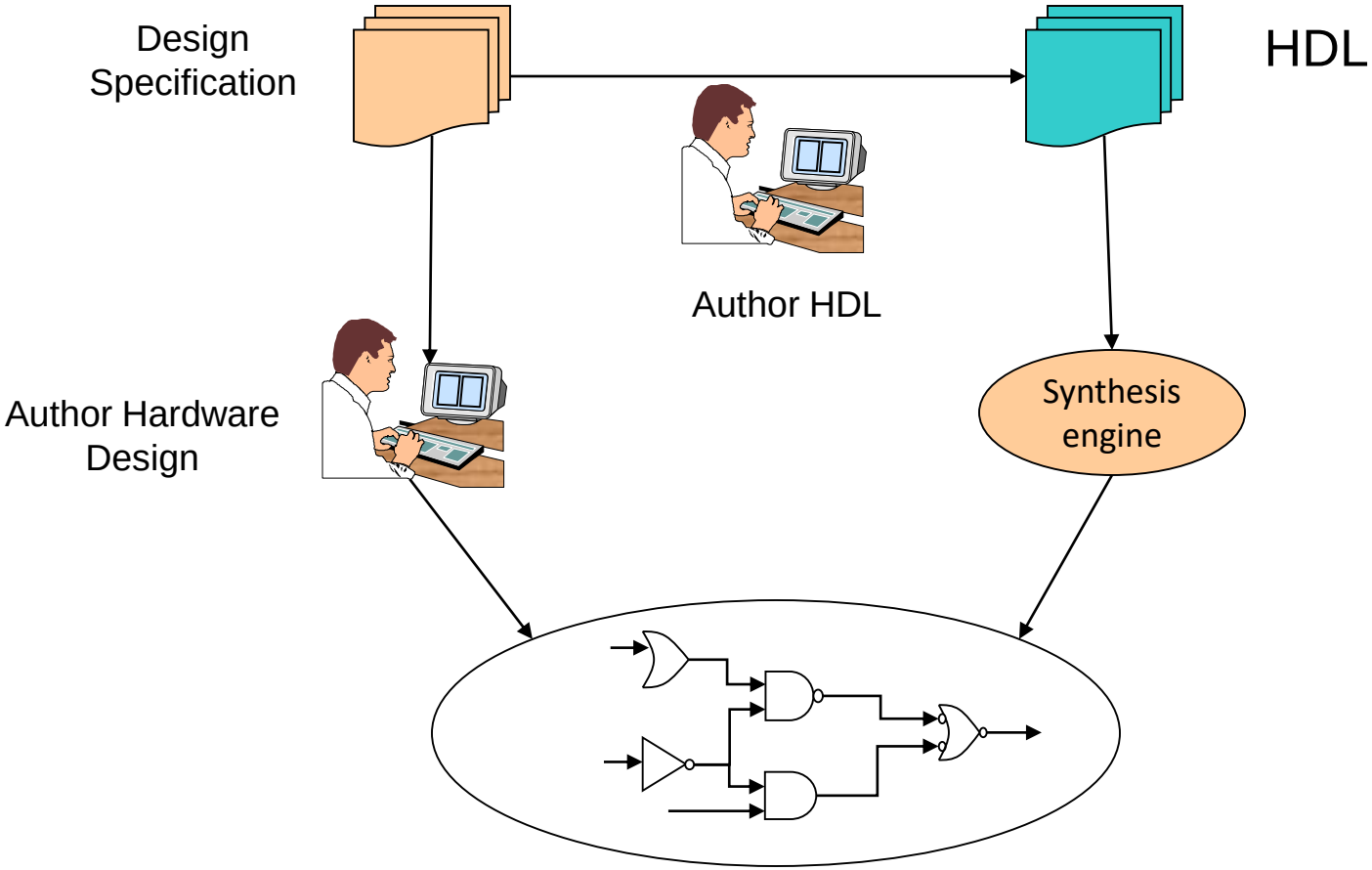
SR Flip Flop



Why do we Describe Systems?

- Design Simulation
  - verify system/subsystem/chip performance prior to design implementation
- Design Synthesis
  - automated generation of a hardware design

Synthesis and Hardware Inference



- Both processes can produce very different results!

Alternatives

- The Verilog hardware description language
  - Finding increasing use in the commercial world
    - System Verilog gaining prominence
  - VHDL dominates the aerospace and defense worlds

- Programming language based design flows
  - SystemC
    - C++ with additional hardware-based language elements
  - C-based design flows
    - C + extensions as well as ANSI C based
  - Other
    - Java, MATLAB, and specialized languages

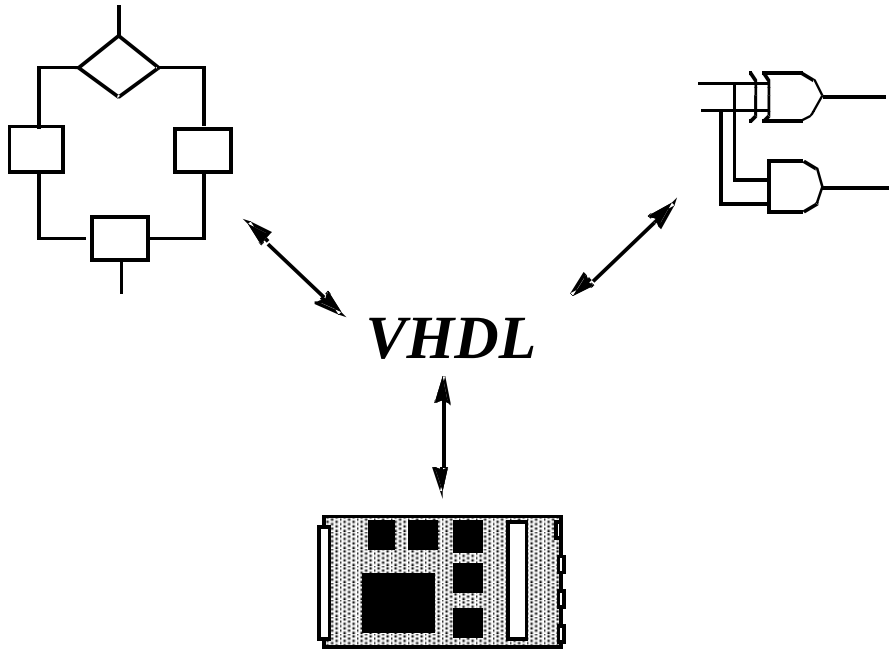
Role of VHDL

V ***Very High Speed Integrated Circuit***

H ***Hardware***

D ***Description***

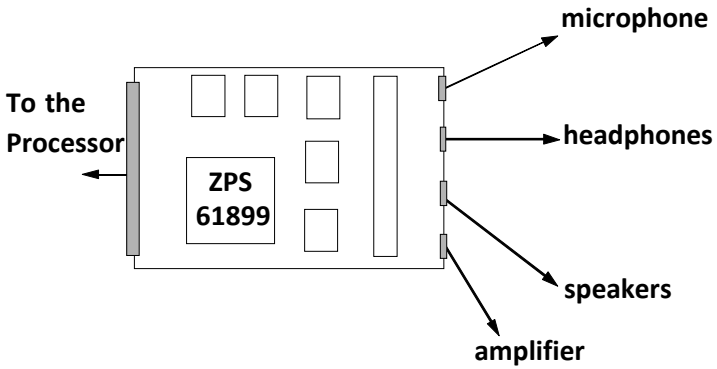
L ***Language***



- System description and documentation
- System simulation
- System synthesis

Modeling Digital Systems

Describing Systems



- From Webster's Dictionary:
  - **System**: "An assemblage of objects united by some form of regular interaction or dependence"

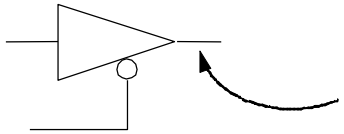
- What aspects of a digital system do we want to describe?
  - Interface
  - Function: behavioral and structural

What Elements Should be in a Description?

- Descriptions should be at multiple levels of abstraction
  - The descriptive elements must be common to multiple levels of hierarchy
- The elements should enable meaningful and accurate simulation of hardware described using the elements
  - Elements should have attributes of time as well as function
- The elements should enable the generation of hardware elements that realize a correct physical implementation
  - Existence of a mapping from elements to VLSI devices

Attributes of Digital Systems: Signal Values

- We associate logical values with the state of a signal



possible signal values?

- Signal Values: IEEE 1164 Value System

Value	Interpretation
U	Uninitialized
X	Forcing Unknown
0	Forcing 0
1	Forcing 1
Z	High Impedance
W	Weak Unknown
L	Weak 0
H	Weak 1
-	Don’t Care

Modeling Digital Systems

- We seek to describe attributes of digital systems common to multiple levels of abstraction
  - events, propagation delays, concurrency
  - waveforms and timing
  - signal values
  - shared signals

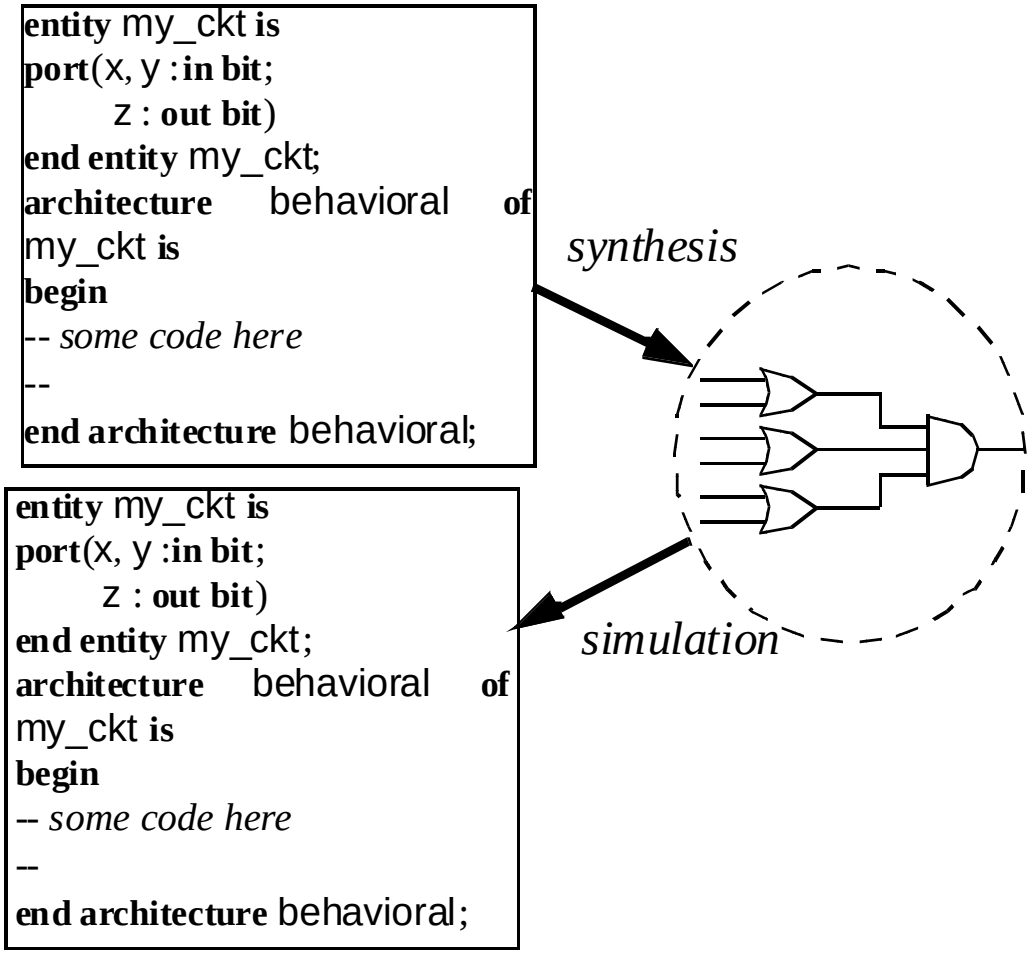
- Hardware description languages must provide constructs for naturally describing these attributes of a specific design
  - simulators use such descriptions for “mimicking” the physical system
  - synthesis compilers use such descriptions for synthesizing manufacturable hardware specifications that conform to this description

Execution Models for VHDL Programs

- Two classes of execution models govern the application of VHDL programs

- For Simulation
  - Discrete event simulation
- For Synthesis
  - Hardware inference
  - The resulting circuit is a function of the building blocks used for implementation
    - Primitives: NAND vs. NOR
    - Cost/performance

Simulation vs. Synthesis



- Simulation and synthesis are complementary processes

Assignment

- Download quartus 2 VHDL software from atera website
- The software is important as it shall be a tool for the study of digital electronics from now onwards.
- I propose you download version 12 together with modelsim.

End