



# Introduction to VHDL

Key Words and Basic Concepts

**By Benjamin Abramov**

© All rights reserved to Abramov B.

All Rights reserved ©

No duplication copying and distribution of this document is allowed without the proper authorization of the author

# Abbreviations

- VHDL – VHASIC Hardware Description Language
- VHSIC – Very High Speed Integrated Circuit
- RTL – Register Transfer Level
- LSB – Least significant Bit
- MSB – Most Significant Bit
- DOD – Department of Defense
- DARPA - Defense Advanced Research Projects Agency
- 
-

# Preface

**V**ery High Speed Integrated Circuit (VHSIC)  
**H**ardware  
**D**escription  
**L**anguage

The VHDL language was developed during 1970s and 1980s as part of the VHSIC program funded by DARPA and aimed at developing the next generation of integrated circuits at the time. Originally the aim of the vhdl development was to be used for documentation, description and simulation of digital electronic designs.

# HDL languages

**System Verilog**



**VHDL**

**Verilog**

**AHDL**

**Abel**

**System C**

# Preface

VHDL was subsequently developed further under the auspices of the - IEEE (Institute of Electrical and Electronic Engineers) and adopted in the form of the - IEEE 1076 standard .



# Preface

## VHDL History

- The American Department Of Defense (DOD) signs a development contract with IBM, TI & Intermetrics for a standard HDL.
- 1987 – The **IEEE-1076-87** VHDL Standard is signed.
- 1988 – The first simulation programs in VHDL go out to market.
- 1990 – The Israeli Ministry Of Defense initiates projects using VHDL in some military companies.
- 1993 – Next version of the language, giving us **VHDL-93**
- 1995 – VITAL'95 standard is signed to include detailed timing data in VHDL models.
- 2001 – The current version of the language **VHDL-2001**

# Preface

## Initial Motivation for VHDL

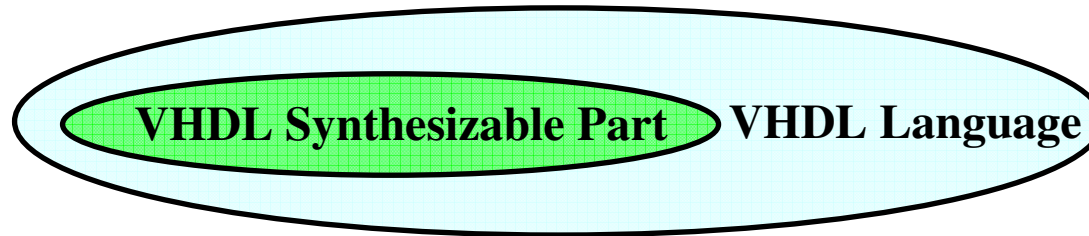
- Hardware documentation
- Hardware simulation
- Clarified standard to DOD contractors
- Increased ability of working with massive hardware design
- Higher level of abstraction
- Design Reuse
- Hierarchical Design methodology

# Preface

VHDL has evolved into a synthesizable language. Has a result it is a much more versatile and descriptive language than standard structural languages that were available at the time of its initiation. Languages such as AHDL and ABLE can be described as netlist format languages and are mainly supporting structural description.

# Preface

VHDL is a *fully simulatable* language,  
but **not a *fully synthesizable*** one.

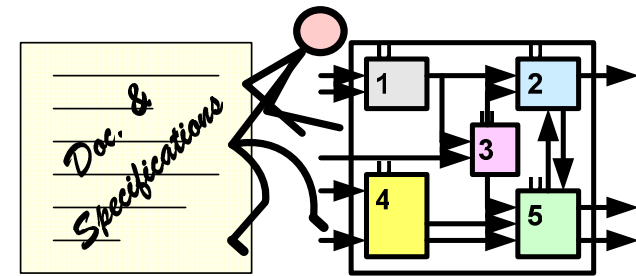


For example the language supports delay time modeling which is very useful for many modeling structures but has no meaning when it come to synthesis.

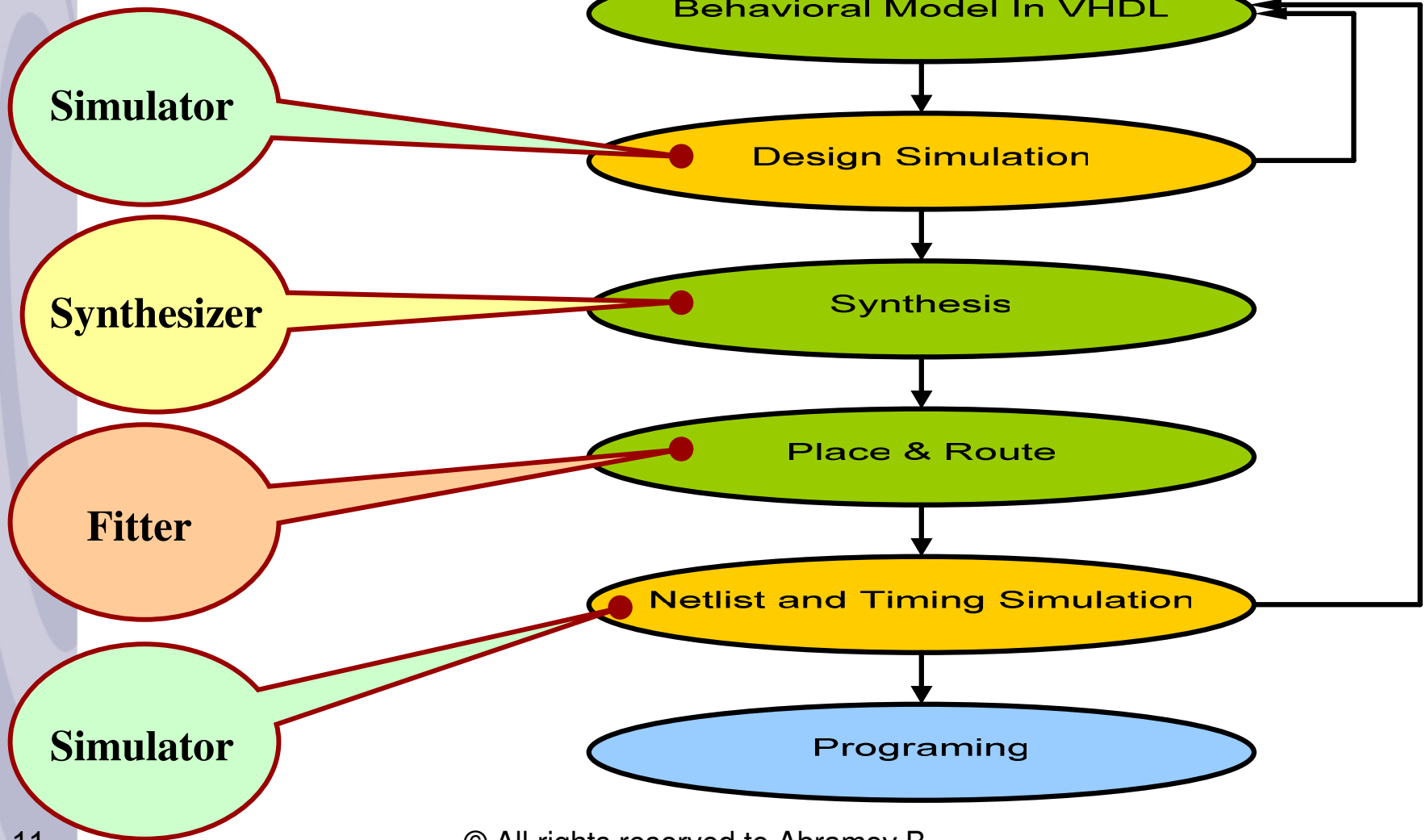
# Design flow

- Design Description and definition.
- Code writing of the design in VHDL.
- Functional Simulation - Simulating the design for logic verification.
- Synthesizing the design. (translation of the VHDL design code to a netlist.)
- Place and Route (Fitting the design to a specific technology).
- Timing Verification - Post fit Simulation (Gate level simulation).
- Programming the design to the selected device.

**Note** – Fault injection should be applied to all simulation levels.



# Design flow



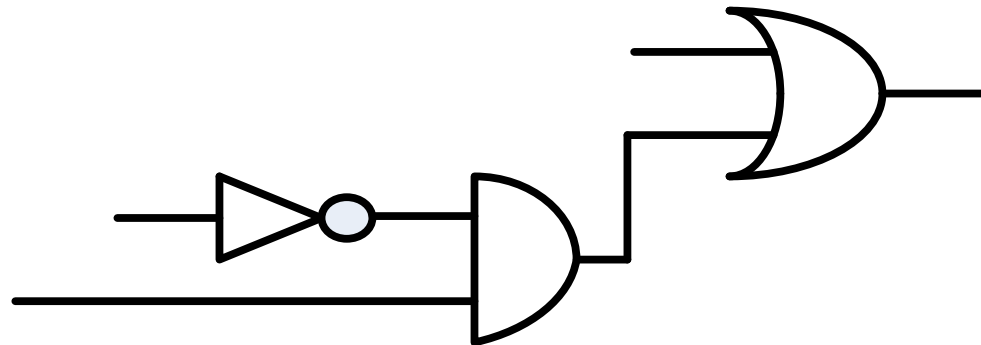
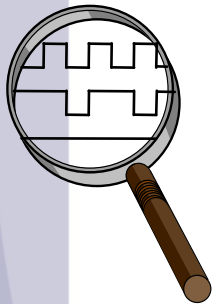
# Design flow

- **Simulator** : The simulation process is done in two phases:
  - Functional simulation, testing the design behavior for proper operation as defined in the specifications.
  - Timing and functional simulation of the gate level netlist.
  - Simulation is mainly done using Test bench.
  - Verification usually done using higher level simulation tools such as Vera and Specman.
- **Synthesizer** : Translates and optimizes the high level design to a gate level netlist.
- **Fitter (Place & Route)** : Mapping of the netlist into the desired device. This is a propriety process dependent on the maker of the device and on the device technology.

# Data flow in VHDL

## Concurrency –

All logical structures are acting concurrently.

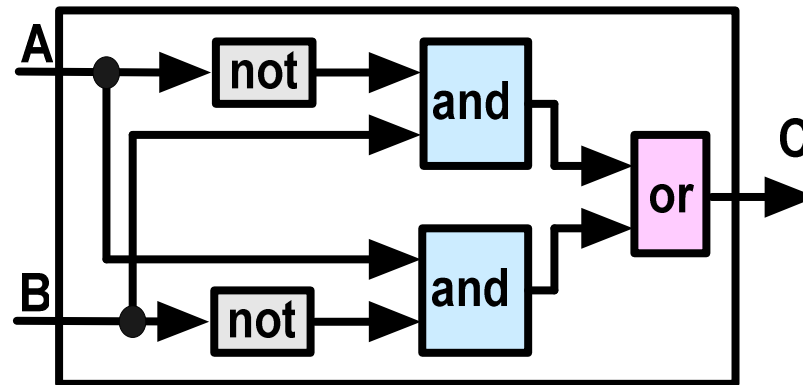


## Sequentially –

Sequential models are result of logical delays and enabled barriers ( i.e. Usually – FFs and Registers – memory modules).

# Level of Abstraction

- Structural Modeling describes the design by means of components interconnections.



- Behavioral Modeling describes the design by means of logical behavior.

$$C \leq (\text{not } A \text{ and } B) \text{ or } (A \text{ and not } B)$$

- The two different modeling approaches can produce a physical implementation. The main advantage is the ease of use and grasp of the design.
  - Ease of use code and not gate connections.

# Level of Abstraction

- **Layout (circuit) Level** – Circuit Description in silicon including all necessary physical data (Mainly applicable to ASICs).
- **Gate Level (netlist)** – Gate level description relates to technology and architecture.
- **RTL Level** – A description of every Register in the design and the Logic in between. Does not include timing and does not relate to technology.
- **Behavioral Level** – describes functionality without specifying the architecture of registers.

# HDL Design Methodology

## *Top Down Design.*

In a top-down design methodology, the design starts with the function of the root (or the top-level block).

The design is then partitioned into a set of lower-level Primitives, Hierarchies or Blocks until the leaf-nodes (or the bottom) of the design is reached.



# HDL Design Methodology

## **Bottom UP Design.**

Bottom-up design methodologies are used when creating a team design or when working with extremely large design.

When using a bottom-up design methodology, the design begins with knowledge of the root and is then partitioned based on the primitives that are available as leaf-nodes.

