© Zainalabedin Navabi, 1994



# PART 3

## VHDL_ Background

- Initiation
- Existing Languages
- VHDL Requirements
- VHDL Language
- VHDL Based Design Process
- Levels of abstraction

© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.initiation

- 1981 DoD Woods Hole MA
  Workshop on HDLs
  ITAR restrictions

- 1983 DoD
  Requirements were established
  Contract awarded to IBM, TI, Intermetrics
  ITAR restrictions removed from language

- 1984 IBM, TI, Intermetrics
  VHDL 2.0 was defined

- December 1984
  VHDL 6.0 was released
  Software development started

- 1985
  VHDL 7.2 was released to IEEE
  ITAR removed from software

- May 1985
  Standard VHDL 1076/A

- December 1987
  VHDL 1076-1987 became IEEE standard

- 1993
  VHDL 1076-1993 was approved



© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.existing_languages

- **AHPL**

A Hardware Programming Language

- **CDL**

Computer Design Language

- **CONLAN**

CONsensus LANguage

- **IDL**

Interactive Design Language

- **ISPS**

Instruction Set Processor Specification

- **TEGAS**

TEst Generation And Simulation

- **TI-HDL**

Texas Instruments Hardware Description Language

- **ZEUS**

An HDL by GE corporation



© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.VHDL_requirements

- General features

Documentation, High level design, Simulation, Synthesis, Test, Automatic hardware

- Design Hierarchy

Multi_level description
Partitioning

- Library Support

Standard Packages
Cell based design

- Sequential Statements

Behavioral software-like constructs

- Generic Design

Binding to specific libraries

- Type declaration

Stongly typed language

- Subprograms

- Timing

Delays, concurrency

- Structural specification

Wiring components

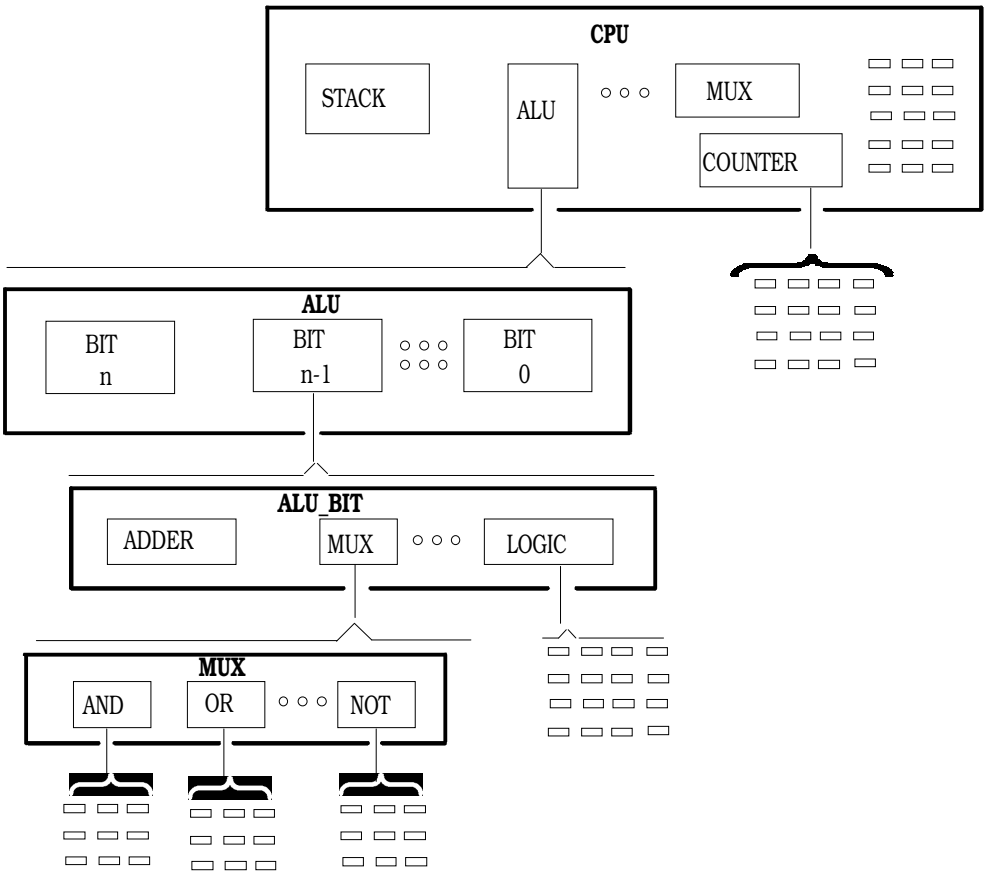


© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.VHDL_requirements.hierarchy




- Use various levels of abstraction for defining a system
- Upper level systems are partitioned into lower



© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.VHDL_requirements.language

- **Entity**
- **Architecture**
- **Package**
- **Configurations**
- **Strong Timing Support**

- A concurrent language for hardware description
- Allows sequential bodies

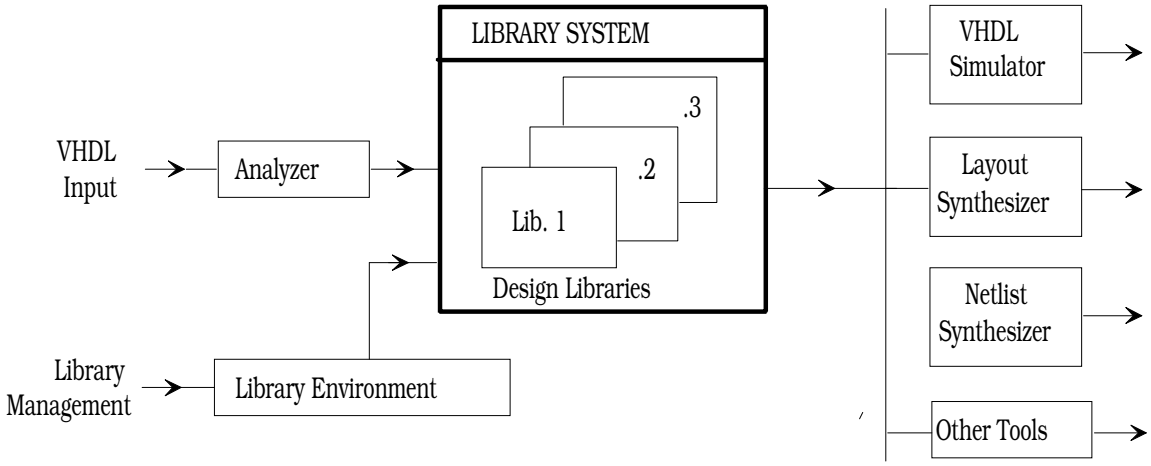


© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.VHDL_requirements.environment




- An environment for simulation, synthesis, test, ...

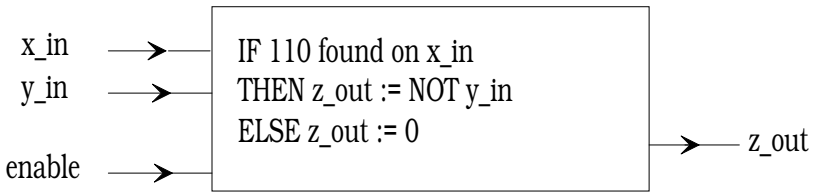


© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.design_process




- A simple example illustrates design process
- Top level system description

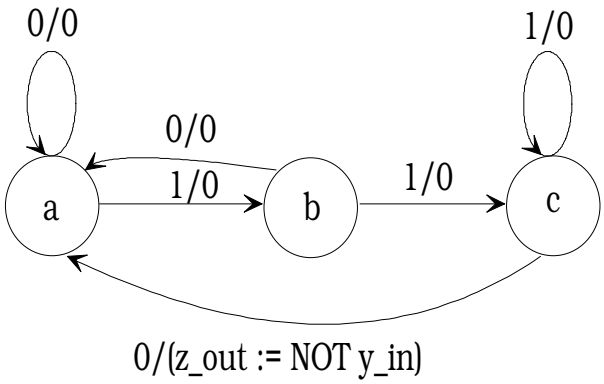


© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.design_process




- A behavioral description of the system
- The first step of the design
- Design can write a behavioral VHDL description



© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.design_process

| | | | x_in | |
|---|---|---|---|---|
| STATE | V0 | V1 | 0 | 1 |
| a | 0 | 0 | 00, 0 | 01, 0 |
| b | 0 | 1 | 00,0 | 10,0 |
| c | 1 | 0 | 00, y | 10, 0 |
| - | 1 | 1 | -- - | -- - |

V0 $^{+}$, V1 , z_out $^{+}$

- Data flow description of the design
- Clock level timing details are specified
- Can generate corresponding VHDL



© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.design_process

V0 += x_in . V1 + x_in . V0

V1 += x_in . V1' . V0'

z_out = x_in' . V0 . y_in'

- Detailed gate level description can be extracted
- Corresponding VHDL structural describes this level of design



© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994



# VHDL_Background.conclusions

**1. Outline:** Introduction, Organization, Outline

**2. Design_Environments:** Digital system design process, Hardware description languages, Hardware simulation, Hardware synthesis

**3. VHDL_Background: VHDL initiation, Existing languages, VHDL requirements, The VHDL language, VHDL based design process, Levels of abstraction**

**4. VHDL_Overview:** Behavioral description, Using process statements, Top-down design, Using available components, Wiring predefined components, Wiring from bottom to top, Generation of testbench data, Using procedures



© Zainalabedin Navabi, 1994