

## **Chapter 2**

### **VHDL Background**

#### **2.1 VHDL INITIATION**

#### **2.2 EXISTING LANGUAGES**

##### **2.2.1 AHPL**

##### **2.2.2 CDL**

##### **2.2.3 CONLAN**

##### **2.2.4 IDL**

##### **2.2.5 ISPS**

##### **2.2.6 TEGAS**

##### **2.2.7 TI-HDL**

##### **2.2.8 ZEUS**

#### **2.3 VHDL REQUIREMENTS**

##### **2.3.1 General Features**

##### **2.3.2 Support for Design Hierarchy**

##### **2.3.3 Library Support**

##### **2.3.4 Sequential Statement**

##### **2.3.5 Generic Design**

##### **2.3.6 Type Declaration and Usage**

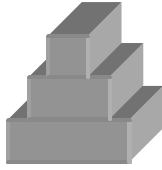
##### **2.3.7 Use of Subprograms**

##### **2.3.8 Timing Control**

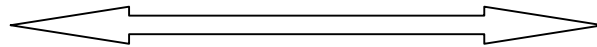
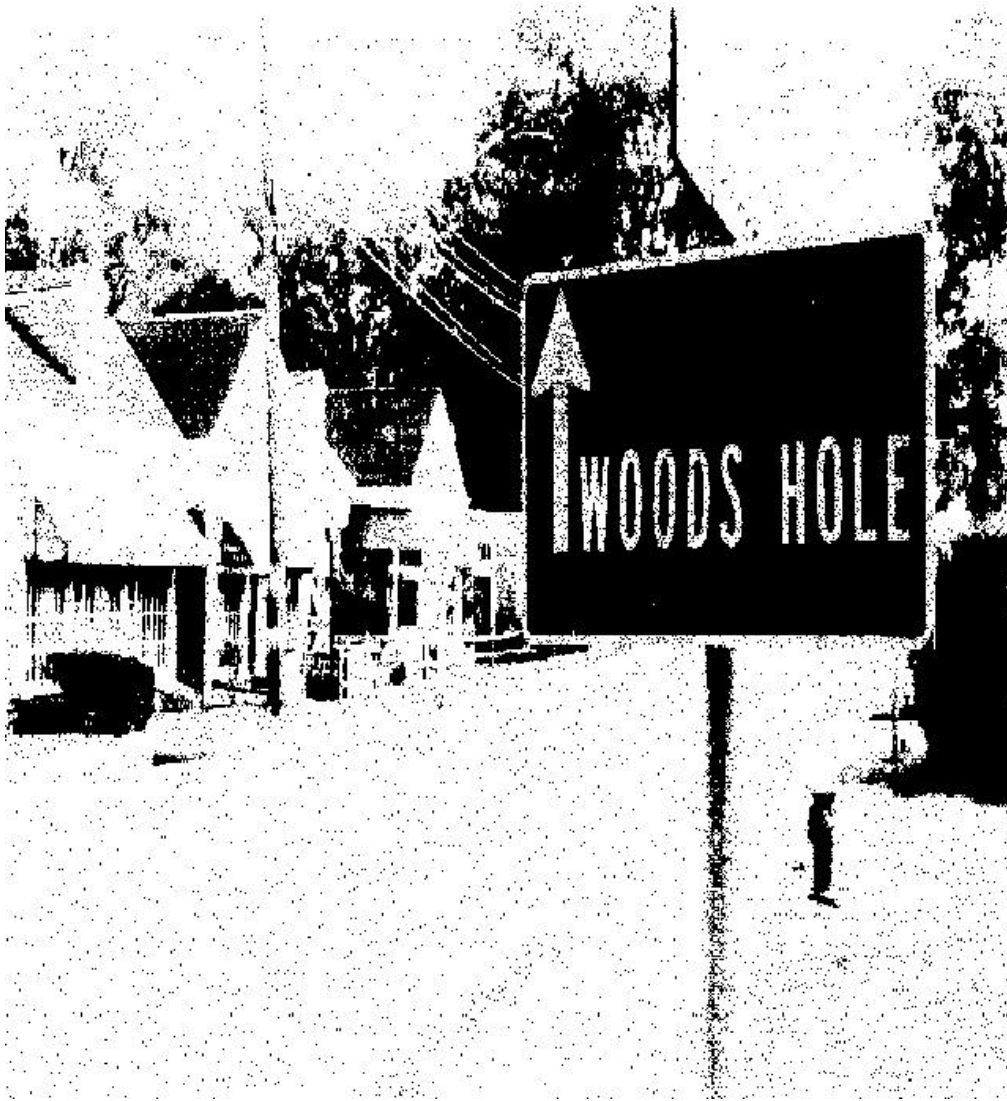
##### **2.3.9 Structural Specification**

#### **2.4 THE VHDL LANGUAGE**

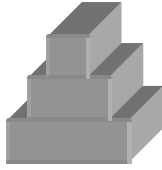
#### **2.5 SUMMARY**



## VHDL Initiation

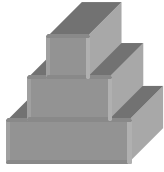


- **1981 DoD Woods Hole MA : Workshop on HDLs**
- Part of VHSIC program



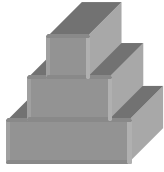
## VHDL Initiation

- **1981 DoD Woods Hole MA** : Workshop on HDLs ITAR restrictions
- **1983 DoD** : Requirements were established Contract was 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



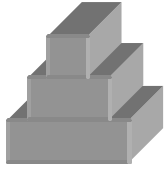
## Languages reviewed

- **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** : TI Hardware Description Language
- **ZEUS** : An HDL by GE corpration



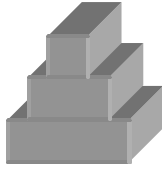
## VHDL Requirements

- **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

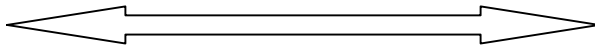
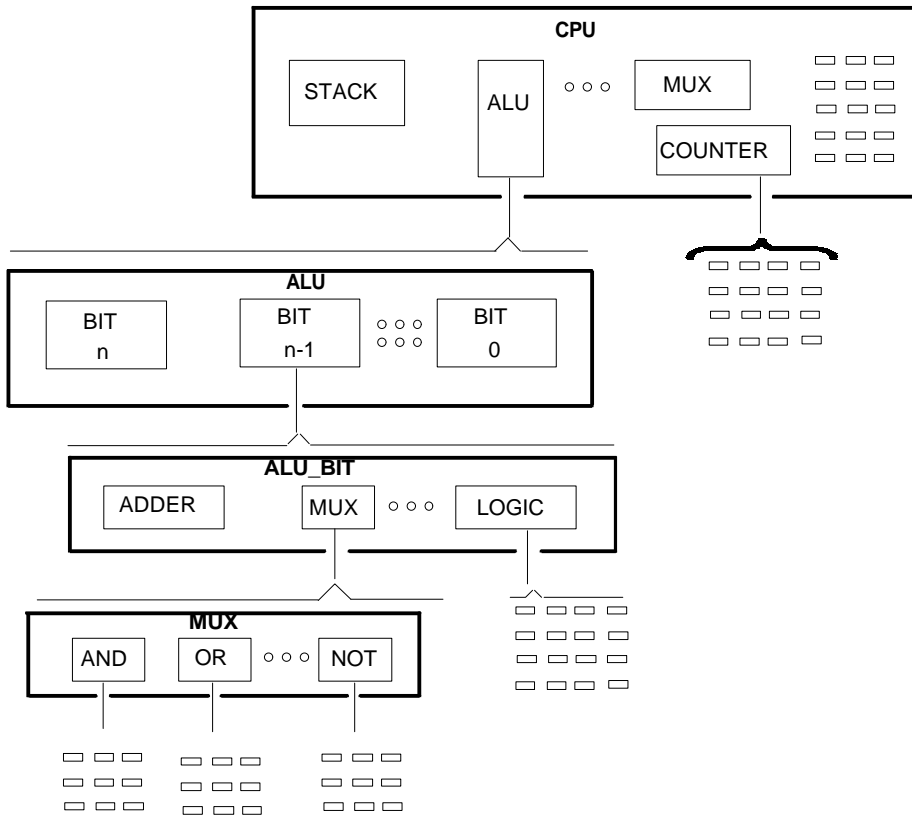


## VHDL Requirements

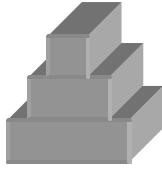
- **Generic Design**  
Binding to specific libraries
- **Type Declaration**  
Strongly typed language
- **Subprograms**
- **Timing**  
Delays, concurrency
- **Structural Specification**  
Wiring components



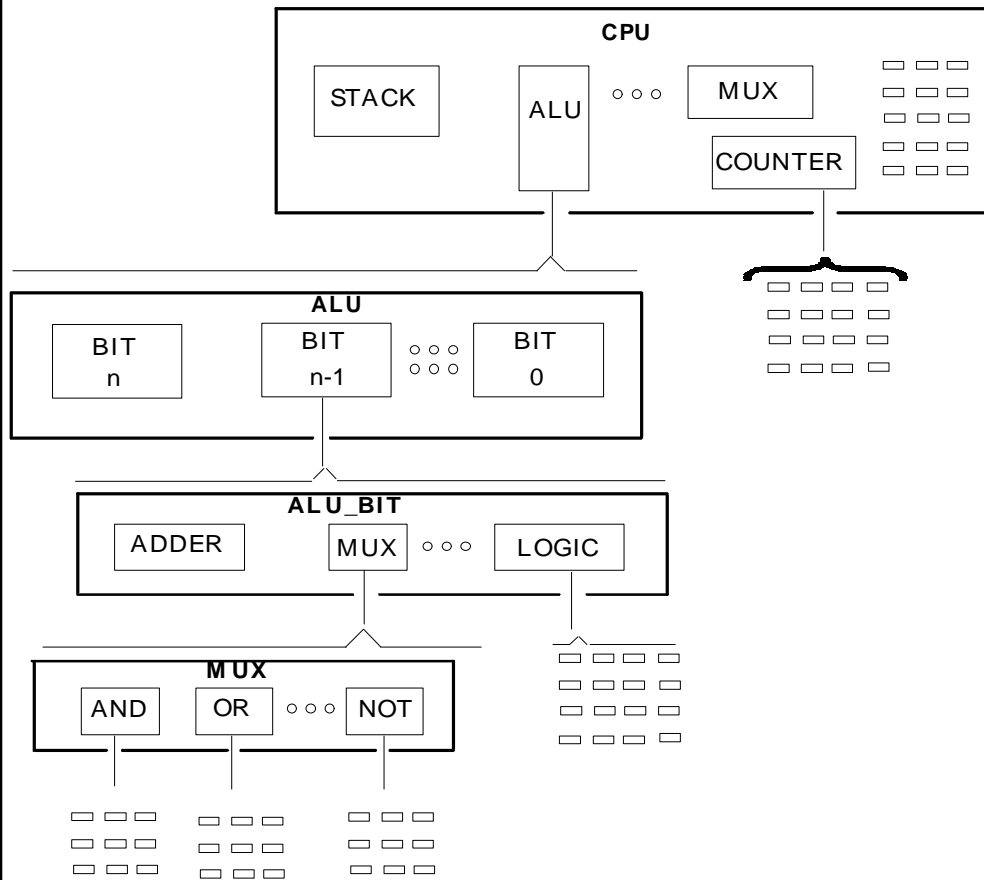
# VHDL Requirements



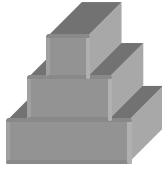
- Use various levels of abstraction for defining a system
- Upper level systems are partitioned into lower



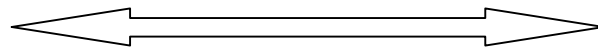
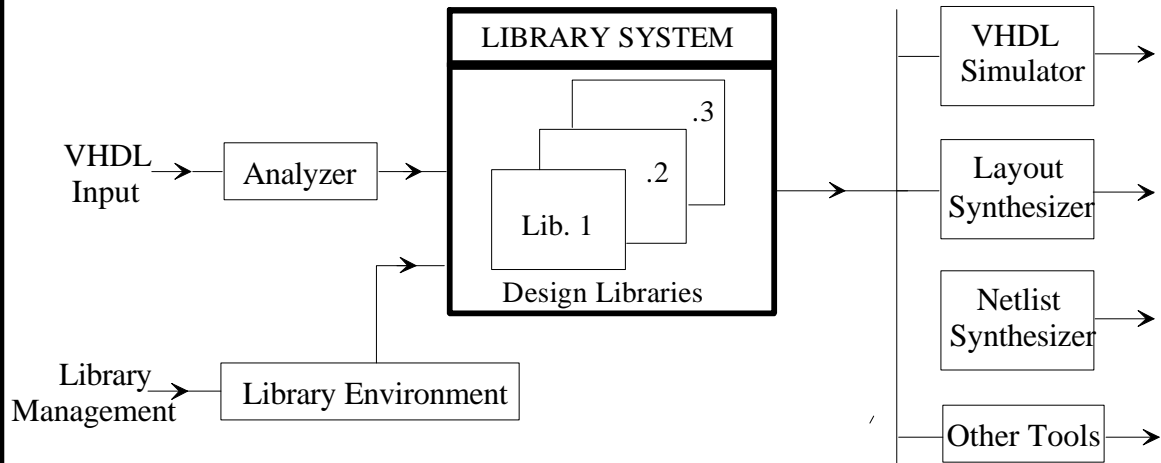
## Example for hierarchical partitioning.



- Recursive partitioning
- Simple components as terminals



## An example VHDL environment.



- VHDL defines library usage
- Tools define library management