Why VHDL?

- Shorter development time for electronic design
- Simpler maintenance
- Traditional: schematic design

VHDL Background

- VHDL - VHSIC Hardware Description Language.
- Industry Standard Language.
- ‘70s and ‘80s by U.S Department of Defense.
- Sophisticated electronics components.
- Standardized by the IEEE in 1987.

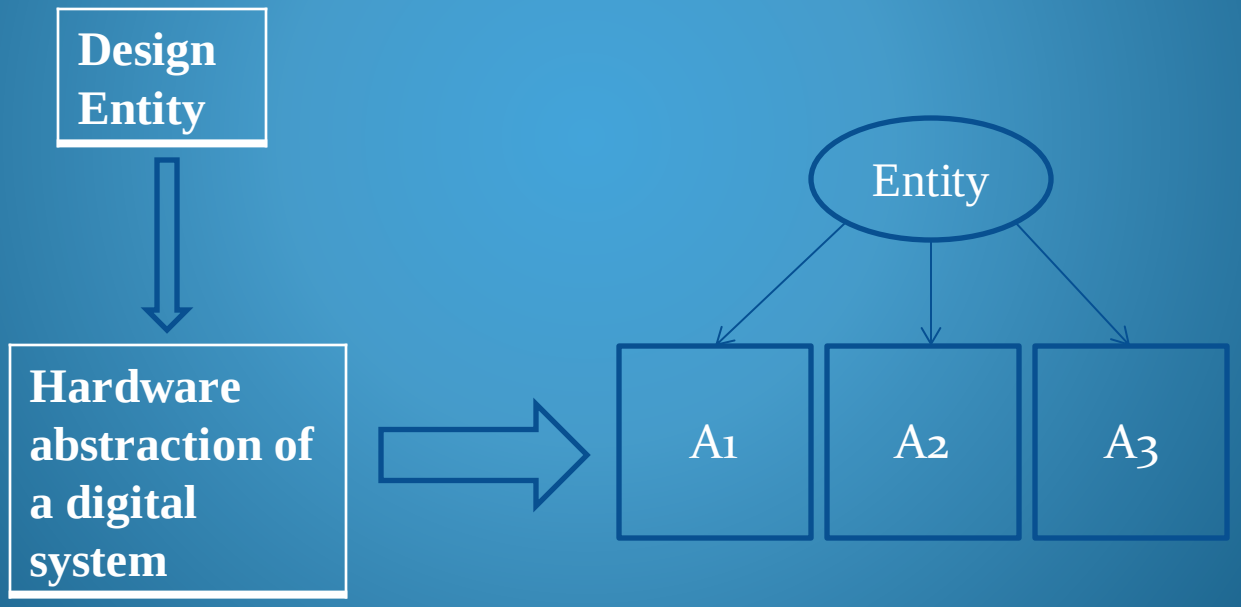
VHDL Background Contd…

- Enables hardware modeling from the gate level to the system level.
- Easy to move code between different commercial platform(tools).
- Supports Design Reuse.
- Supports Design Simulation and Synthesis.

VHDL Usage

- High-tech companies like texas instruments, Intel and most European companies use VHDL.
- Universities use VHDL.
- VHDL groups to support new users.

VHDL Concepts



VHDL Program file structure

- Entity Declaration
- Architecture Body
- Package declaration
- Package body
- Configuration declaration

Libraries in VHDL

library IEEE;

use IEEE.std_logic_1164.**all**;

http://www.csee.umbc.edu/portal/help/VHDL/packages/std_logic_1164.vhd

- Logical operators like and, or , xor etc…
- Conversion functions like To_bit, To_bitvector etc…

Libraries in VHDL

use IEEE.std_logic_textio.**all**;

http://www.csee.umbc.edu/portal/help/VHDL/packages/std_logic_textio.vhd

For File operations like read, write.

use IEEE.std_logic_arith.**all**;

http://www.csee.umbc.edu/portal/help/VHDL/packages/std_logic_arith_syn.vhd

For arithmetic operations like +,_, ABS, *, <,>, <=,>=,/, SHR, SHL, conv_integer, conv_unsigned, to_integer, etc.

Libraries in VHDL

use IEEE.numeric_std.**all**;

http://www.csee.umbc.edu/portal/help/VHDL/packages/numeric_std.vhd

abs, +,-,*,/, rem, mod, >,<,>=,<=,=, rotate left(rol) , rotate _right(ror), shift left logical(sll), shift right logical(srl),To_integer,

_logical operators, edge detection functions, std_match etc…

Libraries in VHDL

use IEEE.math_real.**all**;

http://www.csee.umbc.edu/portal/help/VHDL/packages/mathpack.vhd

Functions like SQRT, EXP, LOG, LOG10, SIN, COS, TAN, SINH, COSH, ARCSIN, ARCCOS, ARCTAN, TANH, ARCSINH, ARCCOSH, ARCTANH etc…

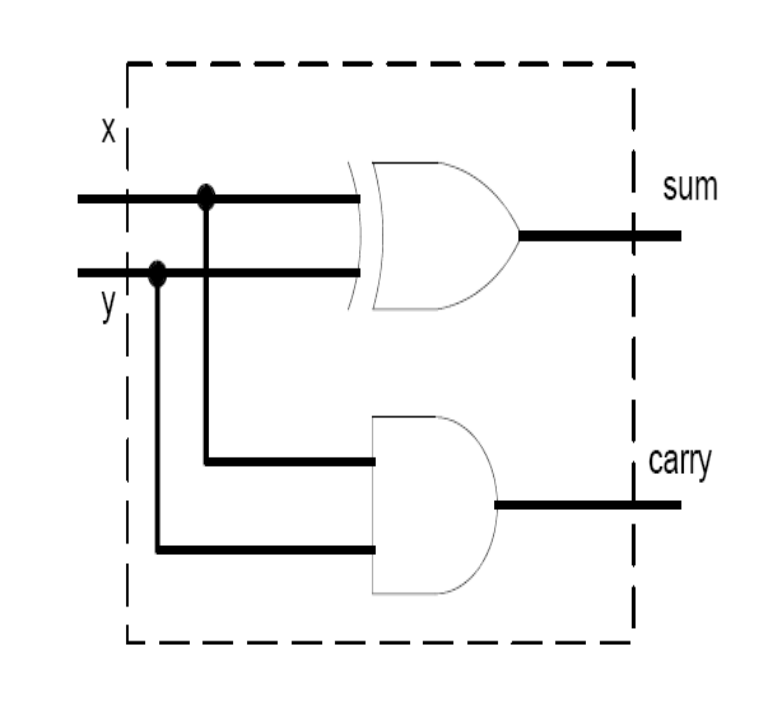
use IEEE.math_complex.**all**;

http://www.csee.umbc.edu/portal/help/VHDL/packages/mathpack.vhd

Functions like COMPLEX_TO_POLAR, POLAR_TO_COMPLEX, ABS, SQRT, EXP, LOG, SIN, COS, TAN, SINH, COSH, TANH, +,-,*,/ etc…

Entity Declaration Syntax

entity NAME_OF_ENTITY **is**
port (*signal_names*: **mode** *type*;
signal_names: **mode** *type*;
:
signal_names: **mode** *type*);
end [NAME_OF_ENTITY] ;



NAME_OF_ENTITY: user defined
Signal_names: list of signals (both input and output)
Mode: in, out, buffer, inout
type: boolean, integer, character, std_logic

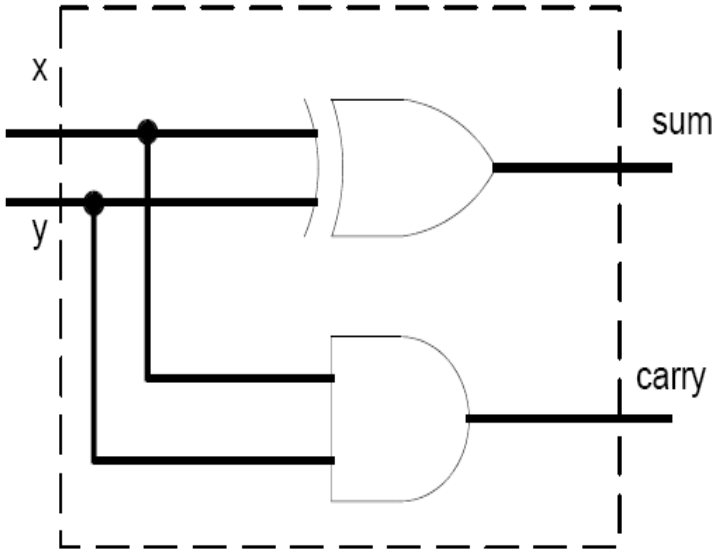
Architecture Declaration Syntax

architecture architecture_name **of** NAME_OF_ENTITY **is**
    -- Declarations
begin
    -- VHDL Statements
End architecture_name;

Internal details of Architecture can be modeled in 4 different styles:

i. Behavioral - sequential assignment statements.
ii. Dataflow – concurrent assignments.
iii. Structural - set of interconnected components
iv. Mixed Mode

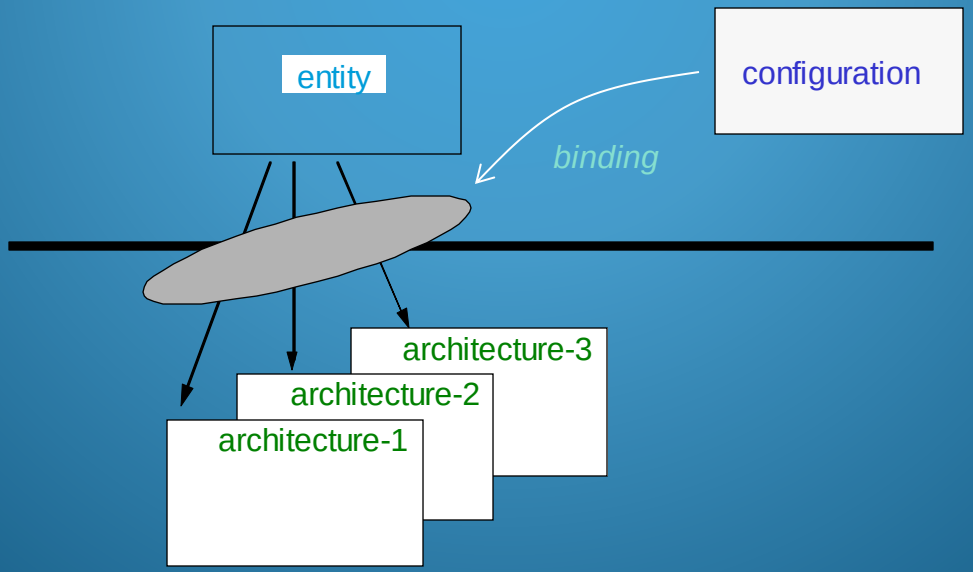
Example of Half Adder



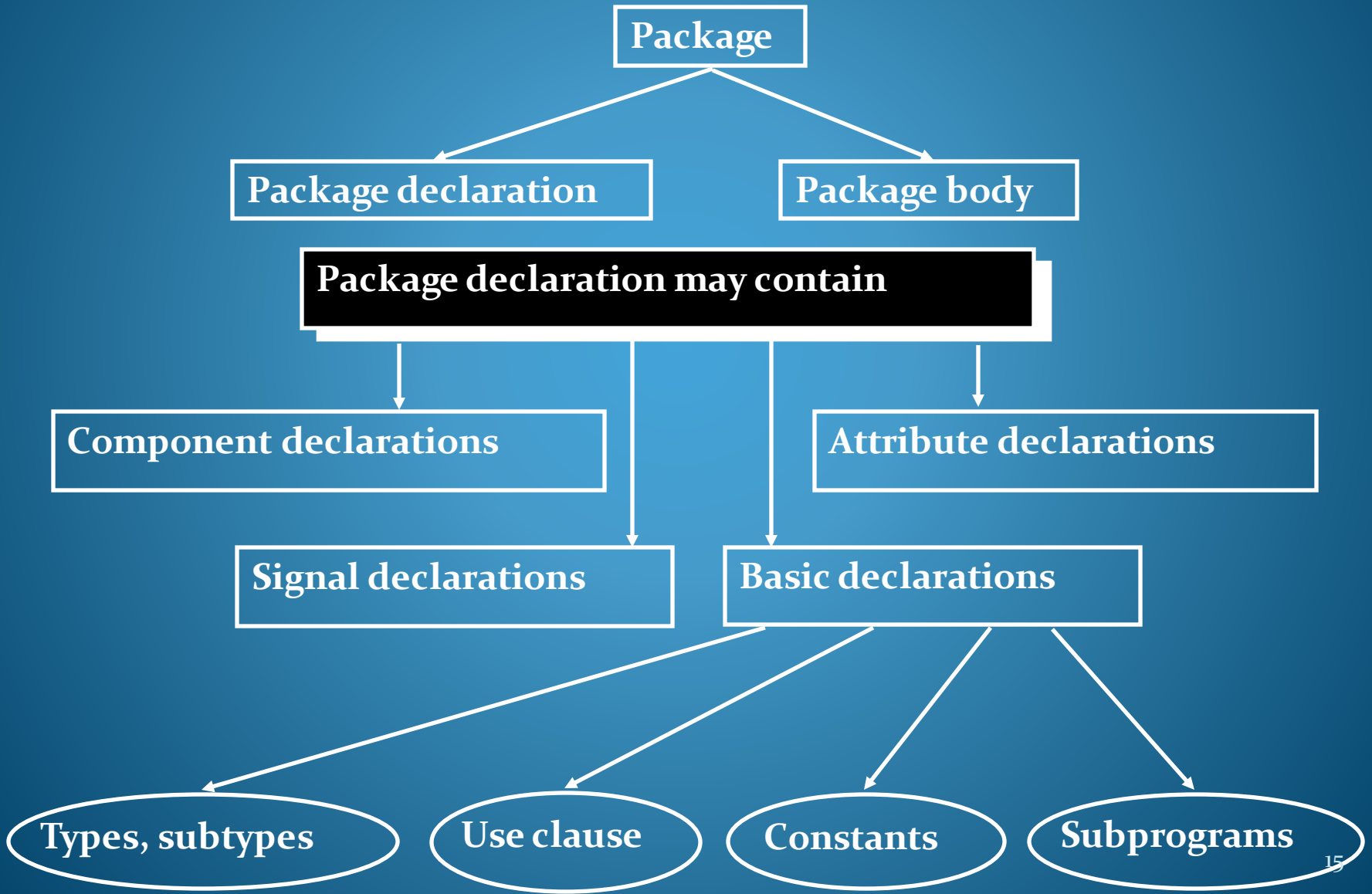

```
library ieee;
use ieee.std_logic_1164.all;

entity half_adder is
port(x,y: in std_logic;
          sum, carry: out std_logic);
end half_adder;
architecture myadd of half_adder
  is
        begin
                sum <= x xor y;
                carry <= x and y;
end myadd;
```

Configuration

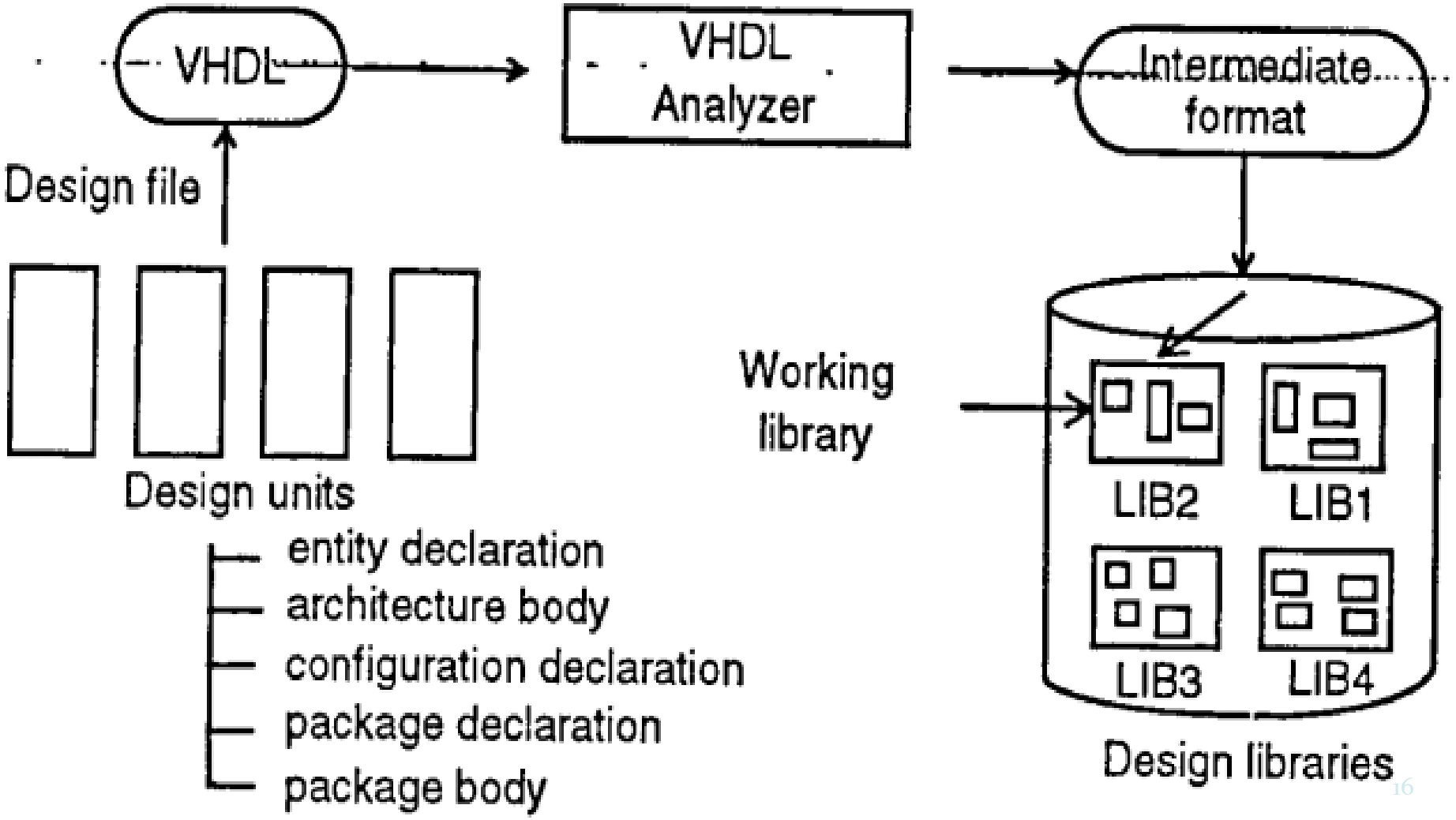
- A design entity can have multiple alternative architectures.
- A configuration specifies the architecture that is to be used to implement a design entity.



Package


Package
Package declaration
Package body
Package declaration may contain
Component declarations
Attribute declarations
Signal declarations
Basic declarations
Types, subtypes
Use clause
Constants
Subprograms

Compilation Process



Simulation Process

- Elaboration Phase
  - Hierarchy of entity is expanded and Linked.
  - Components are bound to the entities in library.
  - Top –level entity is built as network of behavioral models.
  - Storage is allocated for all data objects(signals, variables and co constants) declared in design.
  - Initial values are assigned.

- Initialization Phase
  - Effective values for explicitly declared signals are computed.
  - Processes are executed once until they suspend.
  - Simulation time is reset to 0 ns.

VHDL Language Elements

- Data Objects – stores value of given type
  - Constant
    - holds a single value of a given type.
    - value assigned can't be changed during simulation.
    - declared at the start of an architecture can be used anywhere in the architecture.
    - declared in the process can only be used inside the specific process.
    - Example: constant ADDRESS_WIDTH:INTEGER:=10;
  - Variable
    - Local storage of data.
    - are not available to multiple components or processes.
    - different values can be assigned at different time using variable assignment statement.
    - More convenient than signals for the storage of temporary data.
    - Example: variable SUM: INTEGER range 0 to 100:=10;

VHDL Language Elements

Signal

- communication between components.
- has past values, current values and future values.
- future values can be assigned using a signal assignment statement.
- declared outside the process.
- can be seen as real, physical signals.
- Some delay must be incurred in a signal assignment.
- Example: signal GATE_DELAY: TIME:=10NS;

- Literal – represents constant values.

- Operators – operate on data objects.

VHDL Language Elements

- Identifier
  - Sequence of 1 or more characters.
  - A legal character is an upper case or lower case or a digit or underscore.
  - 1st character must be a letter and the last character may not be underscore.
  - Lower-case and Upper-case letters are identical when used in an identifier. E.g., Count, COUNT, CounT.
  - Two underscore characters cannot appear consecutively.

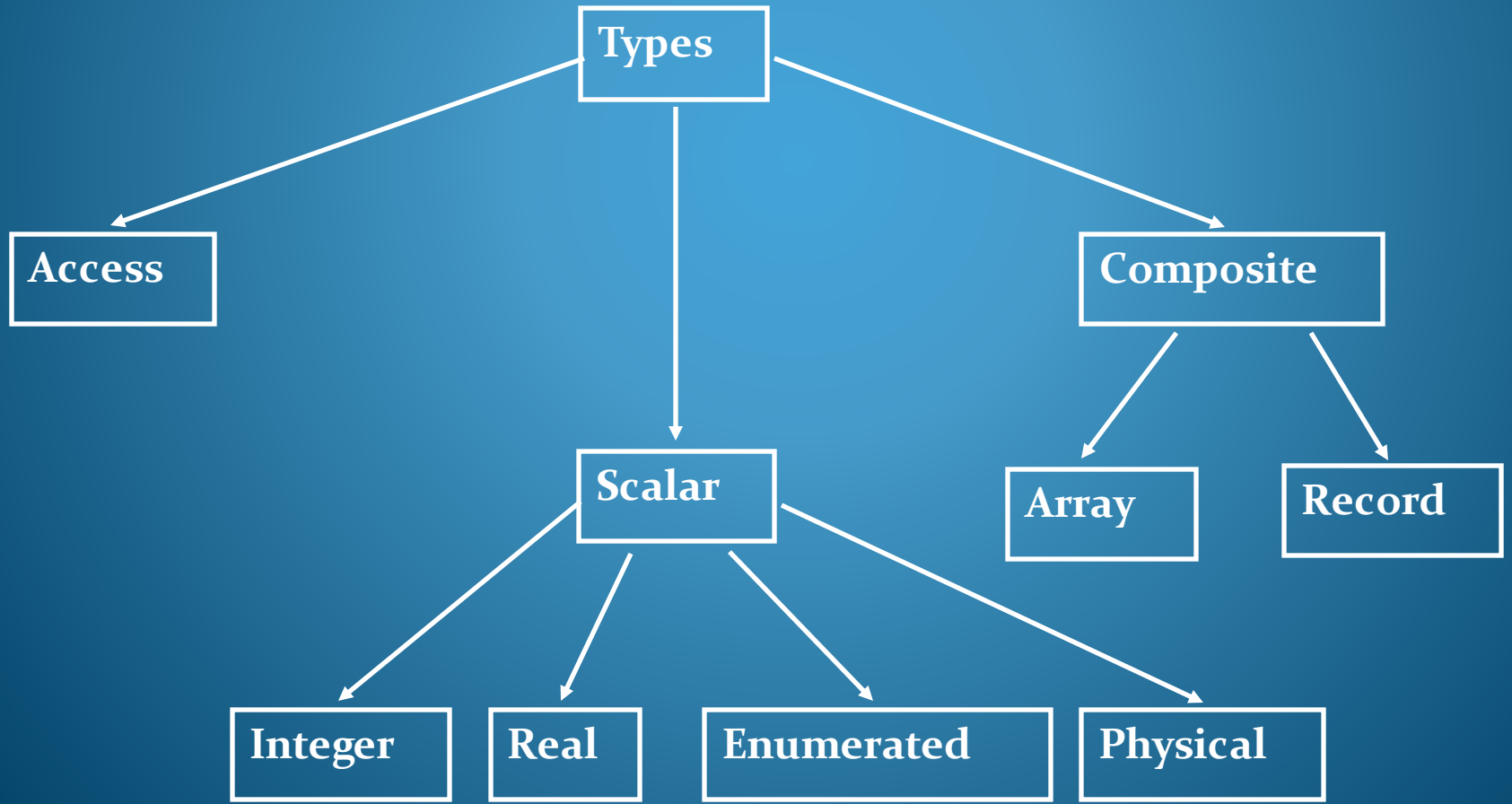
VHDL Language Elements

- Key Words or Reserved Words

  abs, all, assert, begin, bus, case, access, and, attribute, after, architecture, alias, array, block, body, buffer, case, component, configuration, constant, disconnect, downto, else, exit, elsif, end, entity, file, for, function, generate, generic, guarded, if, in , inout, is , label, library, linkage, loop, map, mod, nand, new, next, nor, not, null, of, on, open, or, others, out, package, port, procedure, process, range, record, register, rem, report, return, select, severity, signal, subtype, then, to, transport, type, units, until, use, variable, wait, when, while, with, xor.

Data Types

Data Types



Data Types

- **bit** values: '0', '1'
- **boolean** values: TRUE, FALSE
- **integer** values: -(231) to +(231 - 1)

- **std_logic** values: 'U','X','1','0','Z','W','H','L','-'
  - U' = uninitialized
  - 'X' = unknown
  - 'W' = weak 'X‘
  - 'Z' = floating
  - 'H'/'L' = weak '1'/'0‘
  - '-' = don't care

- **Std_logic_vector** (n downto 0);
- **Std_logic_vector** (0 upto n);