VHDL Overview

- Introduction
- VHDL Components
  - Entity/Architecture
  - Data Objects
- Concurrency
- IEEE 1164 Library description

2

What is VHDL?

- Early ‘80s, US Dept. of Defense project
  - VHSIC Hardware Description Language
    - VHSIC = Very High Speed Integrated Circuit
  - For developing high-speed Digital Circuits
- 1987, IEEE-1076 standard adopted
- 1993, updated IEEE-1076
- Very popular in industry
- Main HDL competitor: Verilog, “simpler to learn”

3

VHDL: A programming Language?

- VHDL has similarities to programming languages:
  - Structures, statements, blocks, objects, libraries, operators, etc...
- Not a ‘Software’ language
  - Software runs sequentially
  - VHDL represents hardware (runs concurrently)

4

VHDL Design steps

- System specification
  - Description of functionality
- Coding
  - Libraries, modules, etc...
- Simulation
  - Test benches
  - Logical behavior
- Synthesis
  - Building the actual hardware (FPGA, ASICs, ...)
  - Simulation with the actual circuit behavior

5

VHDL Coding

- Behavioral
  - Logical description
  - Technology independent
- Structural
  - Blocks/Structures connection
  - Mostly technology dependent
- Mixed
  - Behavioral and structural, combined

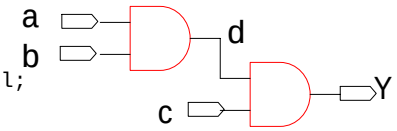
6

Simulation vs. Synthesis

- **Simulation**
  - uses a **VHDL Simulator**
  - requires a VHDL description + a Test bench
  - Verifies functionality and evaluates performance
- **Synthesis**
  - uses a **VHDL Compiler**
  - takes a VHDL description to generate a physical implementation of a circuit
  - The compiler must infer hardware structures necessary to implement the behavior described by the VHDL

7

First Example!



Library
- std_logic

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

Black box description

```
entity AND3 is
port(   a: in std_logic;
        b: in std_logic;
        c: in std_logic;
        y: out std_logic
   );
end AND3;
```

Functional description

```
architecture behav1 of AND3 is
Signal d: std_logic;
Begin
   -- output = f(inputs)
   d <= a and b;
   y <= c and d;
   -- OR
   y <= a and b and c;
end behav1;
```

Signal Assignment

8

VHDL Overview

- Introduction
- VHDL Components
  - Entity/Architecture
  - Data Objects
- Concurrency
- IEEE 1164 Library description

9

VHDL Components

- Interface
  - Entity
- Behavior (Implementation)
  - Architecture
  - Multiple architectures are allowed
- Data
  - Signal
  - Variable
  - Constants

10

Entity/Architecture

- The VHDL description of a circuit is called a **Design Entity** and consists of two main parts:
  - Entity Declaration
  - Architecture Definition
- The **entity declaration** describes the **interface** to the rest of the world; i.e., the inputs and outputs of the circuit
- The **architecture definition** describes one particular **implementation** of the circuit

11

Entity Declaration

- Entity Declarations have a specific syntax:

```
ENTITY entity_name IS
    PORT(
            SIGNAL signal_name : mode type ;
            SIGNAL signal_name : mode type ;
            ...
            SIGNAL signal_name : mode type ) ;
END entity_name;
```

- Entity Declarations have a name and a port. The port basically is where the inputs and outputs of the circuit are listed
- Notice the syntax of a singal declaration inside of the port:

```
SIGNAL signal_name : mode type ;
```

12

I/O Modes

- When declared inside of the port of an entity declaration, we must give the signal a **mode**
- The mode can be 1 of 4 values and basically tells us the direction of the signal
  - IN
    - Data flows along the signal into the circuit.
  - OUT
    - Data flows along the signal out of the circuit.
  - BUFFER
    - Data flows along the signal out of the circuit, but is used internally inside of the circuit.
  - INOUT
    - Data flows along the signal both into and out of the circuit.

13

Architecture Definition

- Architecture Definitions have a specific syntax:

```
ARCHITECTURE architecture_name OF entity_name IS
    -- declarative section
    [SIGNAL declarations]
    [CONSTANT declarations]
    [TYPE declarations]
    [COMPONENT declarations]
    [ATTRIBUTE declarations]
BEGIN
    -- implementation
    [COMPONENT instantiation statements]
    [CONCURRENT ASSIGNMENT statements]
    [PROCESS statements]
    [GENERATE statements]
END architecture_name ;
```

- We will worry about the different sections and possibilities as we need them

14

VHDL Overview

- Introduction
- VHDL Components
  - Entity/Architecture
  - Data Objects
- Concurrency
- IEEE 1164 Library description

15

Data Objects

- VHDL stores information via **Data Objects**
- There are three types of data objects:
  - Signals
  - Constants
  - Variables
- Signals => Hardware Wires
- Examples:
  - `c <= a and b  -- c is a signal`
  - `c := a and b  -- c is a variable (Temporary value)`
- **Signals are the most common**

16

Signals

- Signals have names and we must adhere to the VHDL naming convention
- Signal names can contain any alpha-numeric characters and the underscore
- Restrictions on signal names:
  - Must begin with a letter
  - Can't have two successive underscores
  - Can't end with an underscore
  - Can't be a VHDL reserved word
  - CASE INSENSITIVE
- Data objects have **types** (in our example all signals are type **std_logic**)
- **In our example, we have 5 signals:**
  - **a, b, c, d, and y**

17

Signal Declarations

- Signals need to be declared before assigning values to them
- Signals can be declared inside three places in a VHDL Description:
  - Port of an Entity Declarations
  - Declarations section of an Architecture Description
  - Declarations section of a Package
- **In our example:**
  - **4 signals are declared inside of the Entity Declaration**
  - **1 signal is declared inside the Architecture part**

18

VHDL Overview

- Introduction
- VHDL Components
  - Entity/Architecture
  - Data Objects
- Concurrency
- IEEE 1164 Library description

19

Concurrent Signal Assignment

- Concurrent signal assignment is used to update the value of a signal
- Concurrent signal assignments have the syntax:

  Signal_name < = expression ;

- **In our example, we have 2 concurrent signal assignments in order to assign the output to y**

20

Concept of "Concurrency"

- VHDL is intended to describe the behavior of **digital hardware systems**
- In digital hardware systems, logic circuits are always there and operate in **parallel**
- Whenever there is a change in inputs to a block, a reflection at the output should immediately occur (after a very small delay)
- Unlike a conventional programming language like C or Java, in VHDL the order of assignments is not important; i.e., we program with the notion of **concurrency**
- All concurrent signal assignments operate in parallel, signals are assigned to their new values and at time t+Δ t based on values at time **t**

21

Example of Concurrency

- The following examples are equivalent

```
architecture behav1 of AND2
is
Signal d: std_logic;
Begin
        -- output =
f(inputs)
        d <= a and b;
        y <= c and d;
end behav1;
```

```
architecture behav2 of AND2
is
Signal d: std_logic;
Begin
        -- output =
f(inputs)
        y <= c and d;
        d <= a and b;
end behav2;
```

22

Operators

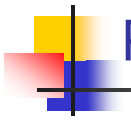
- Part of the library: IEEE.std_logic_1164
- VHDL Descriptions can use operators:
  - Boolean Operators
    - AND, OR, NOT, XOR, NAND, NOR, etc...
  - Relational Operators
    - = (equal),
    - /= (not equal),
    - < (less than),
    - > (greater than), etc...
  - Arithmetic Operators
    - +, -, &, etc.

23

VHDL Overview

- Introduction
- VHDL Components
  - Entity/Architecture
  - Data Objects
- Concurrency
- IEEE 1164 Library description

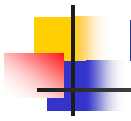
24



Signal and Signal Types Revisited

- Consider real wires in a digital circuit. The logical values 0 and 1 are represented by voltages, and are not sufficient:
  - What if a signal is not driven to a certain value because a wire is disconnected or temporarily disconnected?
  - What if accidentally a signal is concurrently driven to both 0 and 1... What is its correct value?
  - What if the initial value of a signal is not defined?
  - Since signals are implemented physically with wires, how can we represent the strength of a signal?

25



IEEE 1164 Standard

- A numeric standard that attempts to establish a common ground for signal values to enable sharing of VHDL descriptions.
- Approved a 9-valued system:

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

26

IEEE Standard 1164 Continued

- A signal type **std_ulogic** following this standard is defined as follows:

```
type std_logic is (
          'U',                -- Uninitialized
          'X',                -- Forcing Unknown
          '1',                -- Forcing 1
          '0',                -- Forcing 0
          'Z',                -- High impedance
          'W',                -- Weak Unknown
          'L',                -- Weak 0
          'H',                -- Weak 1
          '-'                 -- Don't care
        );
```

- This type is defined in the **IEEE library** in the **1164 package**

27

IEEE Standard 1164 Continued

- Because we can have multiple sources driving a wire, we need a **resolved type**.
  - **i.e., when multiple sources drive a wire (possibly with different values), we need to decide on one value for the driven wire**

	U	X	0	1	Z	W	L	H	-
U	U	U	U	U	U	U	U	U	U
X	U	X	X	X	X	X	X	X	X
0	U	X	0	X	0	0	0	0	X
1	U	X	X	1	1	1	1	1	X
Z	U	X	0	1	Z	W	L	H	X
W	U	X	0	1	W	W	W	W	X
L	U	X	0	1	L	W	L	W	X
H	U	X	0	1	H	W	W	H	X
-	U	X	X	X	X	X	X	X	X

- The IEEE 1164 Standard also defines the signal type **std_logic**

28

IEEE Standard 1164 (Cont.)

- We will see that sometimes we can have **vectors** of signals
  - **`std_logic` and `std_logic_vector`**
- These are defined as follows (for example):

```
signal signal_name : std_logic_vector(7 downto 0);
```

29

What is a Library?

- A library is a repository for frequently used design entities (consider a library to be much like an include file in a C program)
- The VHDL **library IEEE;** in our VHDL Descriptions simply identifies a library that we wish to access
- The library name is a **logical name** and in practice usually just **maps to a directory** on the computer in which various **design units** have been precompiled and stored

30

What is a Package?

- A package is a design unit that contains different types of useful stuff, like definitions of signal types, functions and procedures, etc… usable in our VHDL Descriptions.
- The VHDL **use ieee.std_logic_1164.all;** means that we want to use the **std_logic_1164** package which is stored inside the IEEE library.
- The "**.all**" simply means that we want access to everything stored inside of the package.

31

In Our VHDL Descriptions…

- We need to identify this library (IEEE) and this package (std_logic_1164) in order to use signal types **std_logic** and **std_logic_vector** in our VHDL Descriptions.
- So, we **always** place the following VHDL prior to **every entity declaration**.

```
-- following lines before every VHDL entity declaration.
Library ieee;
Use ieee.std_logic_1164.all;
```

32