Virtual Labs
EPM3064 CPLD board (Helium)

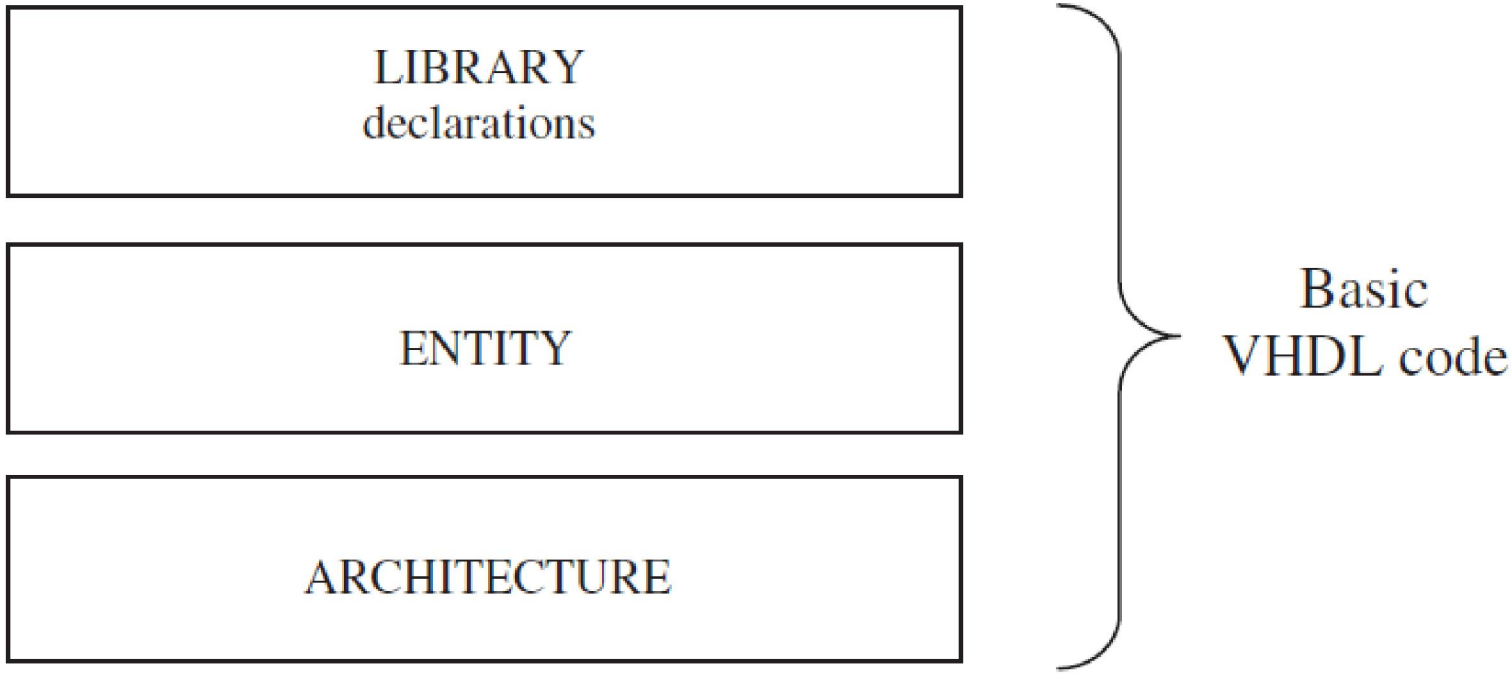
VHDL Basics

Jasveer Singh T. Jethra
Department of Electrical Engineering,
IIT Bombay

VHDL

V – VHSIC (Very High Speed Integrated Circuit)
H – Hardware
D – Description
L -Language

Code Structure



- LIBRARY declarations: Contains a list of all libraries to be used in the design. For eg.: ieee, std, work, etc.
- ENTITY: Specifies the I/O pins of the circuit.
- ARCHITECTURE: Contains the VHDL code proper, which describes how the circuit should behave (function).

Library

- Library Declartion:

LIBRARY library_name;

USE library_name.package_name.package_parts;

- Commonly used packages inside IEEE library:
  1. std_logic_1164
  2. std_logic_arith
  3. std_logic_unsigned

Entity

- Entity declaration:

  ENTITY entity_name IS

  PORT (

  port_name : signal_mode signal_type;

  port_name : signal_mode signal_type;

  ...);

  END entity_name;
- The mode of the signal can be IN, OUT, INOUT, or BUFFER.
- The type of the signal can be BIT, STD_LOGIC, INTEGER, etc. datatypes.

Architecture

- Architecture Declaration:

  ARCHITECTURE architecture_name OF entity_name IS

  [declarations]

  BEGIN

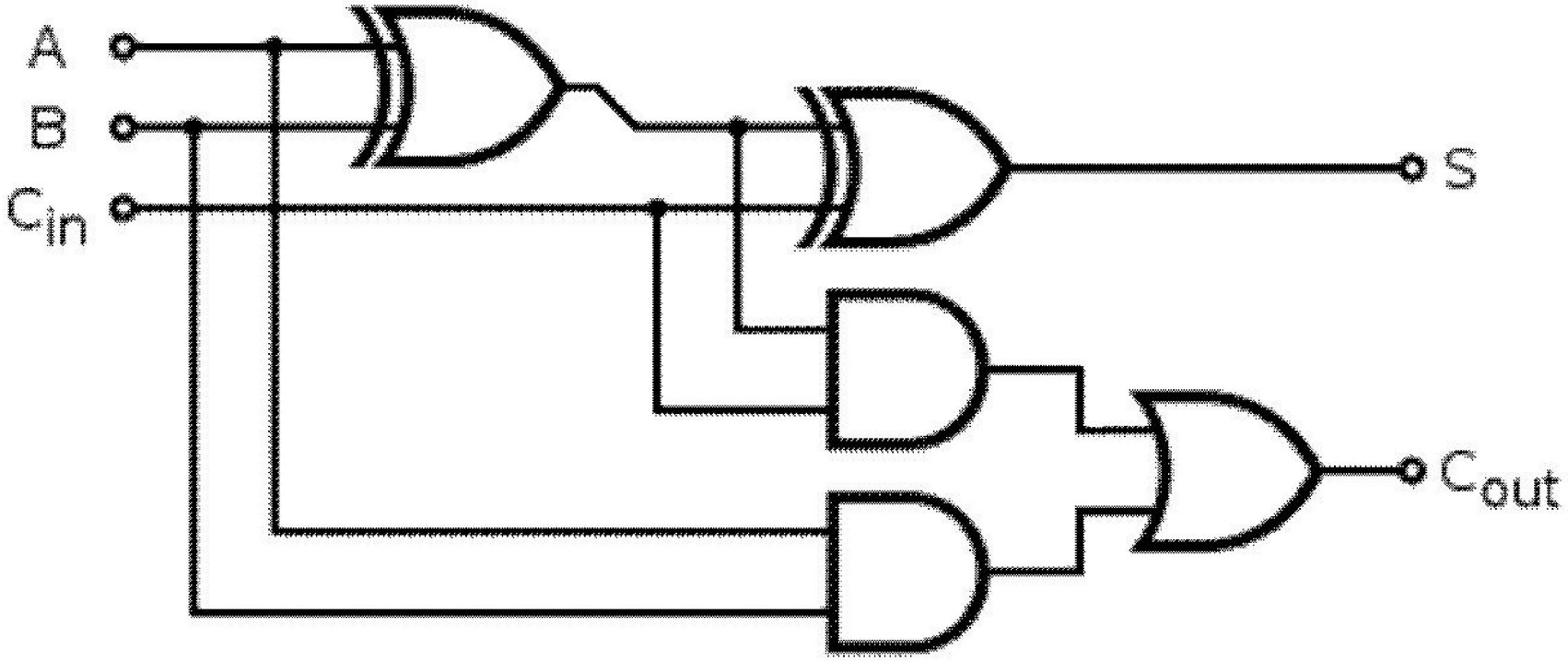
  (code)

  END architecture_name;

Few Important Points

- Every statement ends with a semicolon ‘ ; ’.
- All assignments are made using the symbol ‘ <= ’.
- The code written is concurrent.
- To include sequential statements, we need to write a process.

Example: Full Adder



Example: Full Adder

```
1.  LIBRARY ieee;
2.  USE ieee.std_logic_1164.all;
3.  ENTITY full_adder IS
4.  PORT (a, b, cin: IN BIT;
5.        s, cout: OUT BIT);
6.  END full_adder;
7.  ------------------------------------
8.  ARCHITECTURE dataflow OF
    full_adder IS
9.  BEGIN
10. s <= a XOR b XOR cin;
11. cout <= (a AND b) OR ((a XOR b)
    AND cin);
12. END dataflow;
```

Comments:
Lines 1–2: Library declaration (library name and library use clause).
Lines 3–6: Entity dff.
Line 4: Input ports (input mode can only be IN). In this example, all input signals are of type BIT.
Line 5: Output port (output mode can be OUT, INOUT, or BUFFER). Here, the output is also of type BIT.
Lines 8–12: Architecture behavior.
Lines 9–11: Code part of the architecture (from word BEGIN on).
Lines 10: Dataflow kind of assignment for sum.
Line 11: Dataflow kind of assignment for cout.
Lines 10 and 11: The ‘‘<=’’ operator is used to assign a value to a SIGNAL. All ports in an entity are signals by default.
Lines 7: ‘‘- -’’ indicates a comment. Used only to better organize the design.

Note: VHDL is not case sensitive.

Example: D-f/f

```
1 ------------------------------------
2 LIBRARY ieee;
3 USE ieee.std_logic_1164.all;
4 ------------------------------------
5 ENTITY dff IS
6 PORT ( d, clk, rst: IN STD_LOGIC;
7 q: OUT STD_LOGIC);
8 END dff;
9 ------------------------------------
10 ARCHITECTURE behavior OF dff IS
11 BEGIN
12 PROCESS (rst, clk)
13 BEGIN
14 IF (rst='1') THEN
15 q <= '0';
16 ELSIF (clk'EVENT AND clk='1') THEN
17 q <= d;
18 END IF;
19 END PROCESS;
20 END behavior;
21 ------------------------------------
```

Comments:

Lines 2–3: Library declaration (library name and library use clause).

Lines 5–8: Entity dff.

Line 6: Input ports (input mode can only be IN). In this example, all input signals are of type STD_LOGIC.

Line 7: Output port (output mode can be OUT, INOUT, or BUFFER). Here, the output is also of type STD_LOGIC.

Lines 10–20: Architecture behavior.

Lines 11–19: Code part of the architecture (from word BEGIN on).

Lines 12–19: A PROCESS (inside it the code is executed sequentially).

Line 12: The PROCESS is executed every time a signal declared in its sensitivity list changes. In this example, every time rst or clk changes the PROCESS is run.

Lines 14–15: Every time rst goes to ‘1’ the output is reset, regardless of clk (asynchronous reset).

Contd.

```
1 ------------------------------------
2 LIBRARY ieee;
3 USE ieee.std_logic_1164.all;
4 ------------------------------------
5 ENTITY dff IS
6 PORT ( d, clk, rst: IN STD_LOGIC;
7 q: OUT STD_LOGIC);
8 END dff;
9 ------------------------------------
10 ARCHITECTURE behavior OF dff IS
11 BEGIN
12 PROCESS (rst, clk)
13 BEGIN
14 IF (rst='1') THEN
15 q <= '0';
16 ELSIF (clk'EVENT AND clk='1') THEN
17 q <= d;
18 END IF;
19 END PROCESS;
20 END behavior;
21 ------------------------------------
```

Lines 16–17: If rst is not active, plus clk has changed (an EVENT occurred on clk), plus such event was a rising edge (clk = ‘1’), then the input signal (d) is stored in the flip-flop (q <= d).

Lines 15 and 17: The ‘‘<=’’ operator is used to assign a value to a SIGNAL. In contrast, ‘‘:=’’ would be used for a VARIABLE. All ports in an entity are signals by default.

Lines 1, 4, 9, and 21: ‘‘- -’’ indicates a comment. Used only to better organize the design.

Note: VHDL is not case sensitive.

Data Types

- BIT, BIT_VECTOR, BOOLEAN, INTEGER
- STD_LOGIC_1164 of library ieee has : STD_LOGIC, STD_LOGIC_VECTOR.
- STD_LOGIC_ARITH of library ieee: Defines SIGNED and UNSIGNED.