Lab 2

ENGR 2323

Digital Design Lab

Class Outline

- Hardware description languages
- VHDL.
- VHDL Modeling Styles.
- 2 to 4 Decoder VHDL Design.
- Lab 2 deliverables.

Objectives

- Be able to explain the difference between behavioral, dataflow, and structural VHDL architectures.
- Be able to design combinational circuits using a hardware description language such as VHDL.
- Be able to verify combinational circuit design operation using simulations and testing on a FPGA development board.

Important Terms/Concepts

- Hardware description languages such as VHDL allow one to describe a digital design using a text description similar to a program. Complex designs are often much more manageable using HDLs than a schematic design.
- Important terms/concepts include:
  - HDL, VHDL
  - Entity, architecture
  - Structural, dataflow, behavioral
  - library

Hardware Description Languages

- Hardware description languages (HDL) are text-based high-level languages resembling programming languages that allow one to specify a circuit design in a form that resembles a computer program. HDLs differ from most programing languages in that they support concurrent operation.
- HDLs support design, modeling, simulating/testing, and implementation of designs.
- HDLs support designs at various levels of abstraction from high-level behavioral descriptions to low-level primitive components.
- Any design that could be described using a schematic can be described using HDLs and vice versa.
- The most commonly used HDLs are VHDL and Verilog. We will focus on VHDL in this course.

VHDL

- VHDL stands for VHSIC HDL
  - VHSIC stands for very high-speed integrated circuit
  - HDL stands for hardware description language
- VHDL is non-proprietary and an IEEE standard (IEEE 1076 – 2019).
- VHDL designs consist of two parts:
  - Entity: the device interface, specifies inputs and outputs via a port map
  - Architecture: implementation, specifies device operation
- The architecture for a design is not unique. A designer may use one architecture to test their logic but use another for the actual implementation and verify timing.

VHDL Syntax

- VHDL is strongly typed. VHDL is case insensitive.
- Semicolons indicate end of statement. Double hyphen (--) indicates a comment.
- VHDL keywords include:
  - entity, architecture
  - port, in, out
  - signal
  - begin, end

VHDL Libraries

- Designs typically use one or more libraries that define data types and operations. The library and use clauses specify the library and which packages of the library that will be used in the design (VHDL compiler needs this information).
- For example, the following two statements import all of the standard logic data type from the IEEE library.

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

- Some common packages used in digital designs:

```
ieee.std_logic_1164
ieee.std_logic_signed
ieee.std_logic_unsigned
ieee.numeric_std
```

Typical VHDL Description

- For a single file VHDL design:
  - Specify the libraries and packages
  - Entity specifies the device name and the interface (inputs and outputs).
  - Architecture describes the device operation. Architecture may use internal signals (signal statement).

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

ENTITY name_of_device IS
  PORT
  ( X, Y    : IN  type;
    Z       : OUT type );
END name_of_device;

ARCHITECTURE arch_name OF name_of_device IS
  SIGNAL  W      : type;
BEGIN
  <statements of the design>
END arch_name;
```

VHDL Modeling Styles

- Designs cane be described using four styles of modeling:
  - Structural: operation described by interconnections of components similar to a schematic design.
  - Dataflow: operation described by how signals flow through the design using concurrent assignment statements. Description looks like logic expressions.
  - Behavioral: sequential assignment statements describing the behavior. This is closest to a programming language description from an algorithm.
  - Mixed: combination of styles.

VHDL Design of 2 to 4 Decoder

- A 2 to 4 decoder has two inputs, B and A, and four outputs Y0, Y1, Y2, and Y3. Most decoders also have an enable input.

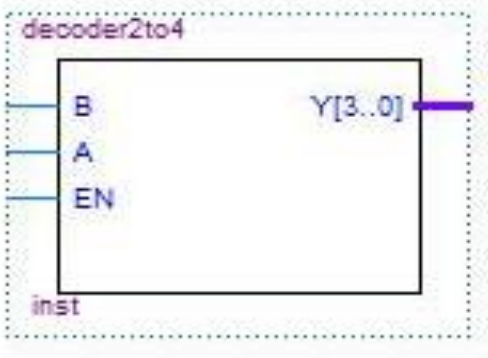
EN B A	Y0	Y1	Y2	Y3
0 X X	0	0	0	0
1 0 0	1	0	0	0
1 0 1	0	1	0	0
1 1 0	0	0	1	0
1 1 1	0	0	0	1

$$Y0 = EN \cdot \overline{B} \cdot \overline{A}$$

$$Y1 = EN \cdot \overline{B} \cdot A$$

$$Y2 = EN \cdot B \cdot \overline{A}$$

$$Y3 = EN \cdot B \cdot A$$



Entity for 2 to 4 Decoder

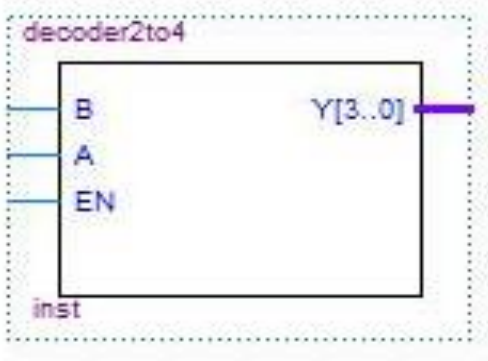
- Entity describes interface of component. Decoder interface has:
  - Three inputs, EN, B, and A
  - Four outputs, Y0, Y1, Y2, and Y3
- The entity name and .vhd filename must be the same. For Quartus designs, if the design is the top-level file the file name must be the same as the project name.

Libraries and what packages to use

Entity name

Port map

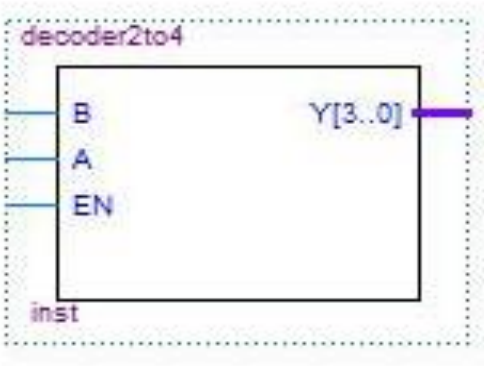
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY decoder2to4 IS
PORT(B, A : IN STD_LOGIC;
       EN : IN STD_LOGIC;
       Y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END decoder2to4;
```



PORT Declaration

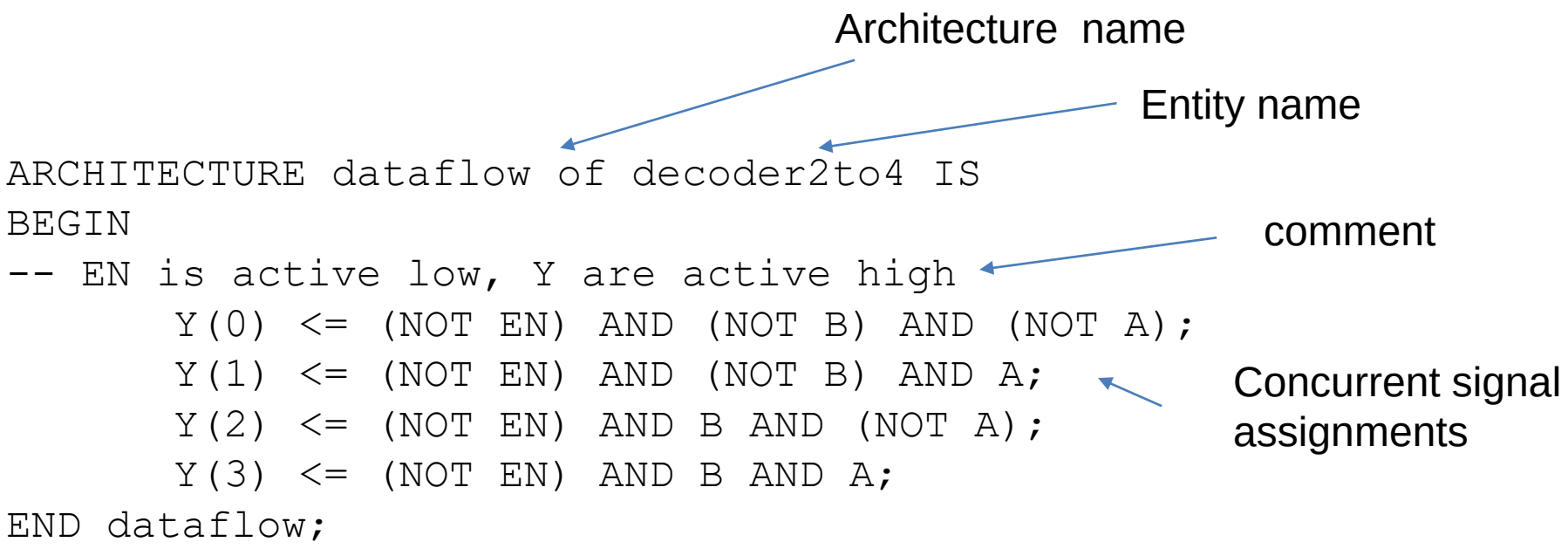
- The PORT declaration establishes the interface of the object to the outside world. The PORT statement includes the entity name, the names of the inputs and output signals, and the data type and mode of input and output the signals.
- Commonly used modes are: in, out, inout, and buffer.
- Commonly used signal data types are std_logic (single bit signal) or std_logic_vector (bus). The width of the bus can be defined in either ascending or descending order. For example, a 4-bit bus can be written as (0 upto 3) or (3 downto 0).

```
PORT(B, A : IN STD_LOGIC;
       EN : IN STD_LOGIC;
       Y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
```



Architecture for 2 to 4 Decoder

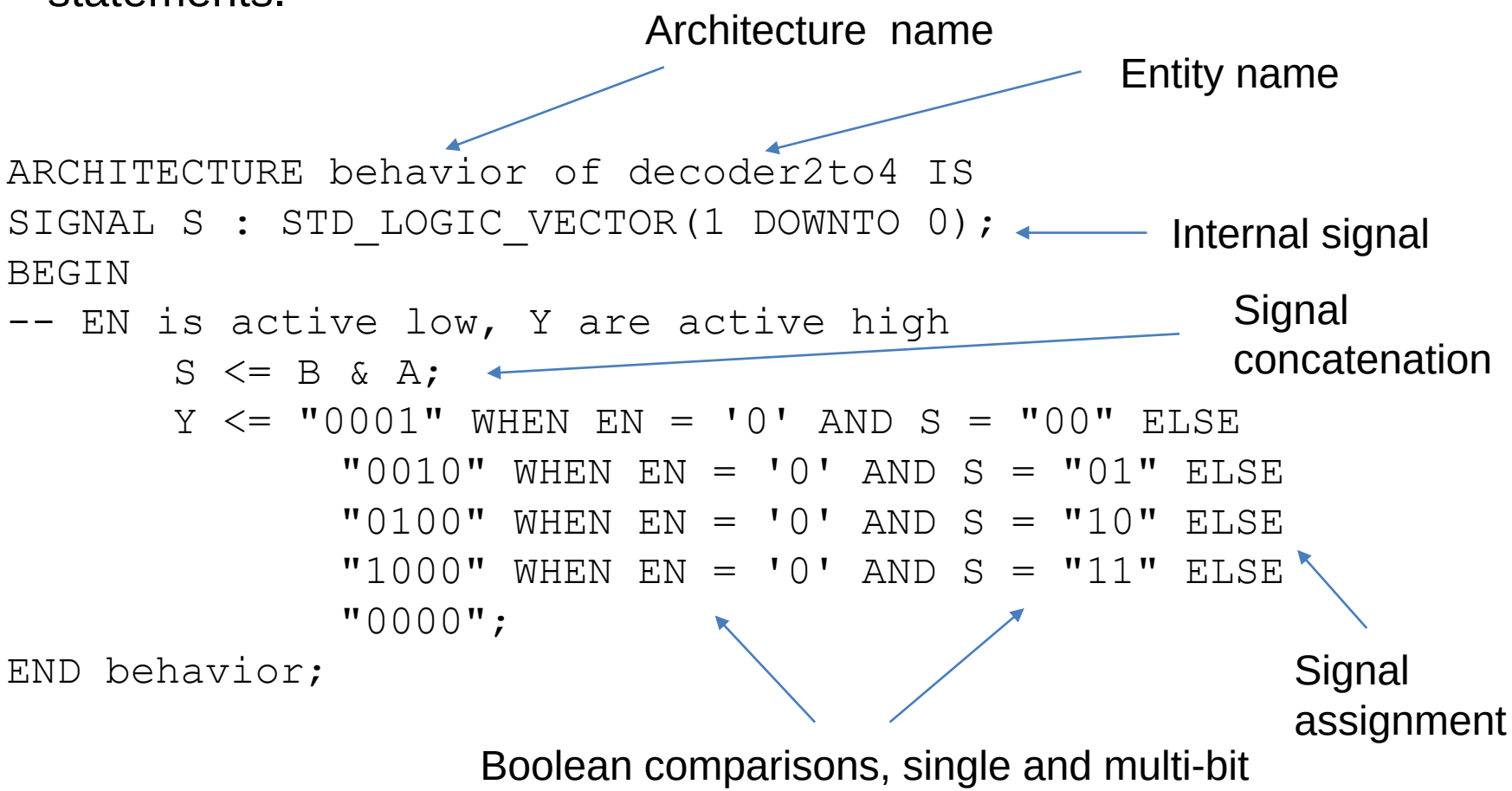
- Dataflow architectures describe the operation using concurrent signal assignment statements (logic expressions).




```
ARCHITECTURE dataflow of decoder2to4 IS
BEGIN
-- EN is active low, Y are active high
       Y(0) <= (NOT EN) AND (NOT B) AND (NOT A);
       Y(1) <= (NOT EN) AND (NOT B) AND A;
       Y(2) <= (NOT EN) AND B AND (NOT A);
       Y(3) <= (NOT EN) AND B AND A;
END dataflow;
```

Architecture for 2 to 4 Decoder

- Behavioral architectures describe the operation similar to a high-level algorithm often using concurrent or sequential conditional statements.



Architecture for 2 to 4 Decoder

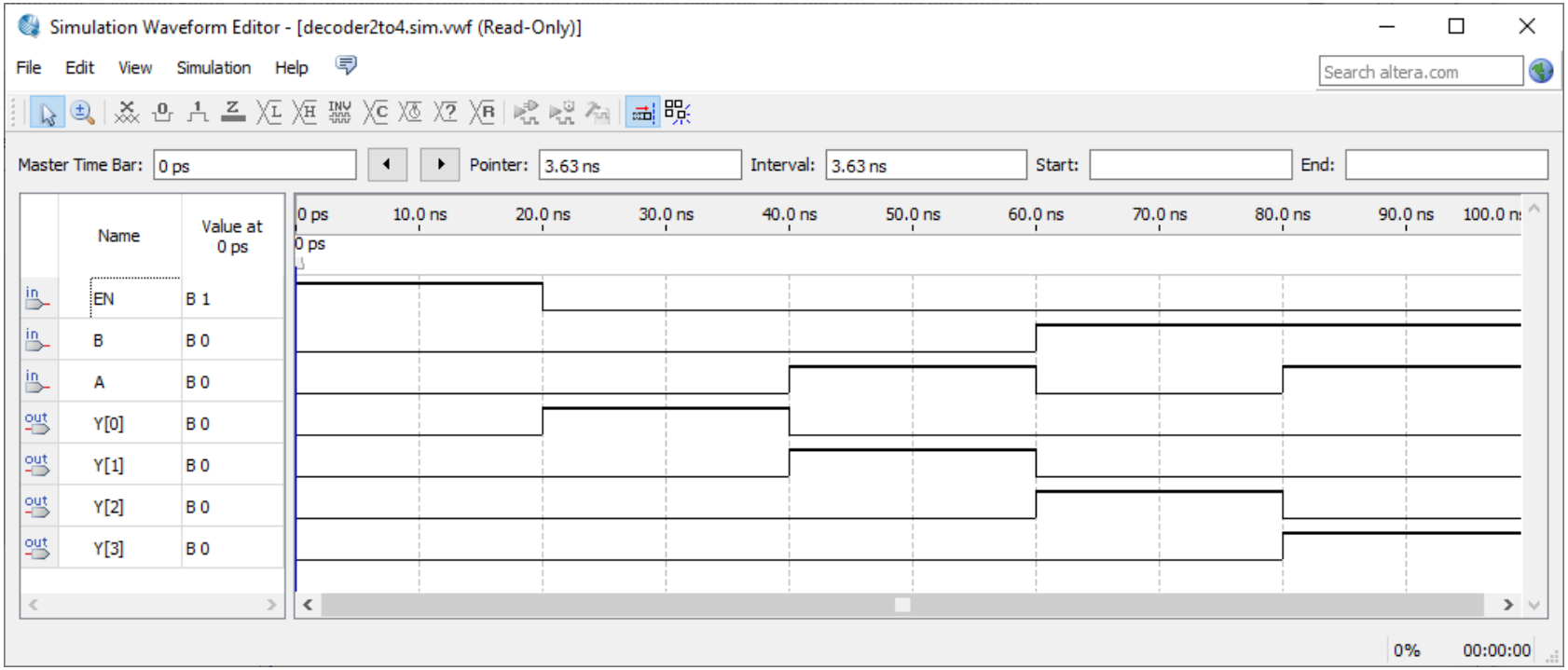
- An alternative behavioral architecture that uses a with select instead of a when else construct.

```
ARCHITECTURE behavior of decoder2to4 IS
SIGNAL S : STD_LOGIC_VECTOR(2 DOWNTO 0);
BEGIN
-- EN is active low, Y are active high
       S <= EN & B & A;
       WITH S SELECT
       Y <= "0001" WHEN "000",
              "0010" WHEN "001",
              "0100" WHEN "010",
              "1000" WHEN "011",
              "0000" WHEN OTHERS;
END behavior;
```

2 to 4 Decoder Design (Dataflow)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY decoder2to4 IS
PORT(B, A : IN STD_LOGIC;
       EN : IN STD_LOGIC;
       Y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END decoder2to4;

ARCHITECTURE dataflow of decoder2to4 IS
BEGIN
-- EN is active low, Y are active high
       Y(0) <= (NOT EN) AND (NOT B) AND (NOT A);
       Y(1) <= (NOT EN) AND (NOT B) AND A;
       Y(2) <= (NOT EN) AND B AND (NOT A);
       Y(3) <= (NOT EN) AND B AND A;
END dataflow;
```

2 to 4 Decoder Design



Lab 2

- You will need to determine the logic expression(s) for the 4-1 multiplexer.
- Remember that the project name and top-level design file name in Quartus designs must be the same. For VHDL designs the top-level entity must also have the same name as top-level design file.
- Make sure to assign the correct device for your project. Designs are compiled for targeted FPGA. The DE10-standard boards have a Cyclone V 5CSXFC6D6F31C6 FPLD. Make sure all unused FPGA pins are set as inputs tri-stated.

Lab 2 Deliverables

- Lab 2 Prelab
  - Dataflow VHDL design for 4-1 multiplexer.
  - Functional simulation results for the design
- Lab 2 Work
  - Program the DE10-standard FPLD with your dataflow design of the 4-1 multiplexer and verify the design operation.
  - VHDL design of 4-1 multiplexer using behavioral architecture.
  - Functional simulation results of behavioral design.
  - Program the DE10-standard FPLD with your behavioral design of the 4-1 multiplexer and verify the design operation.
- Lab 2 Results
  - Dataflow VHDL design for 4-1 multiplexer.
  - Functional simulation results for the dataflow design.
  - Behavioral VHDL design for 4-1 multiplexer.
  - Functional simulation results for the behavioral design.
  - Images of the DE10-standard board showing the output Y for four of the test combinations.
  - Brief explanation of how the design operation was verified on the DE10-Standard board.