FACULTY OF ENGINEERING

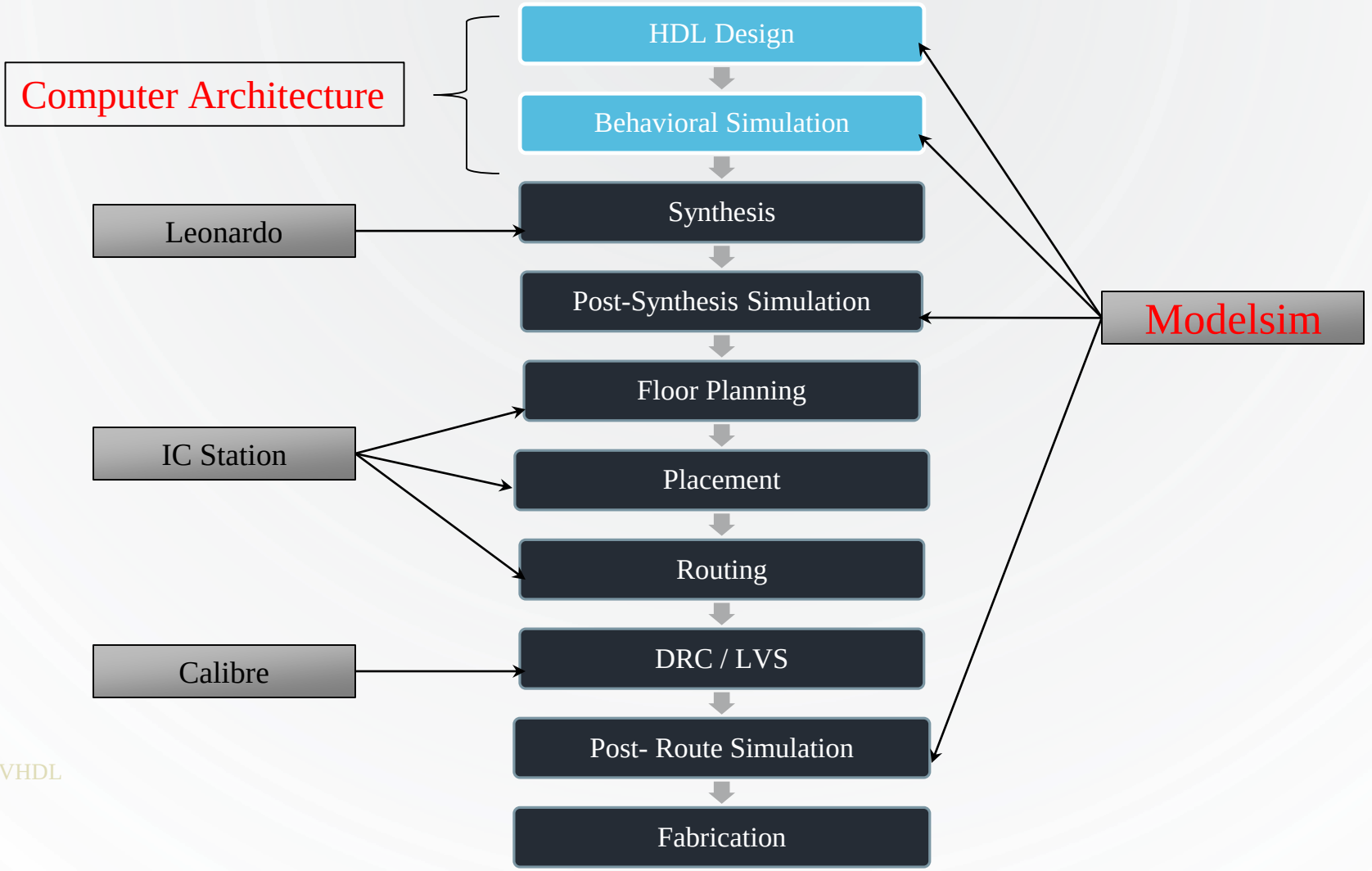
CAIRO UNIVERSITY

LAB 1
INTRODUCTION TO VHDL

CMPN301: COMPUTER ARCHITECTURE COURSE

EXERTED FROM DR. ADNAN SHAOUT- *THE UNIVERSITY OF MICHIGAN-DEARBORN*

DESIGN FLOW


Computer Architecture
HDL Design
Behavioral Simulation
Leonardo
Synthesis
Post-Synthesis Simulation
Modelsim
Floor Planning
IC Station
Placement
Routing
Calibre
DRC / LVS
Post- Route Simulation
Fabrication

OBJECTIVES

- Define VHDL and its usage
- Understand different modeling techniques in designing digital circuits (Dataflow , structural)
- Understand the Hardware created from the VHDL code
- Understand Concurrent statement
- Reuse previously created entity (component instantiation a.k.a port mapping)
- Practice designing with VHDL

WHAT IS VHDL?

- VHSIC Hardware Description Language
- VHDL is a programming language that allows one to model and develop complex digital systems.

WHY VHDL?

- Requirements specification
- Documentation
- Testing using simulation
- Formal verification
- Synthesis
- Class assignments ☺

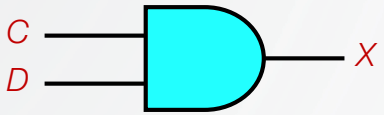
HW DESIGNER GOAL

- Most ‘reliable’ design process, with minimum cost and time.
- Avoid design errors!

*VHDL is **Describing Hardware** so every line counts and cost a lot in real world*

VHDL LANGUAGE: VARIABLES

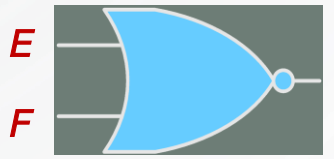
- Variables are Wires
- Assignment are “<=” or “:=” instead of “=”



AND

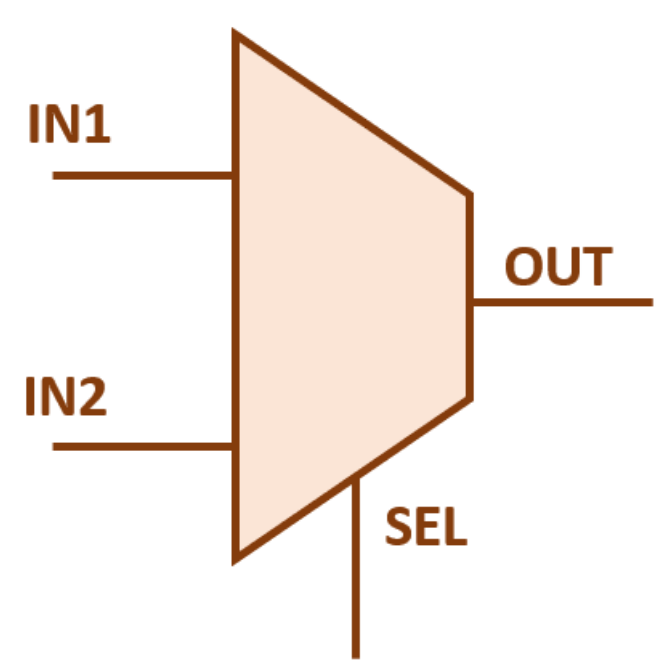
X <= C and D;

W <= E nor F;



NOR

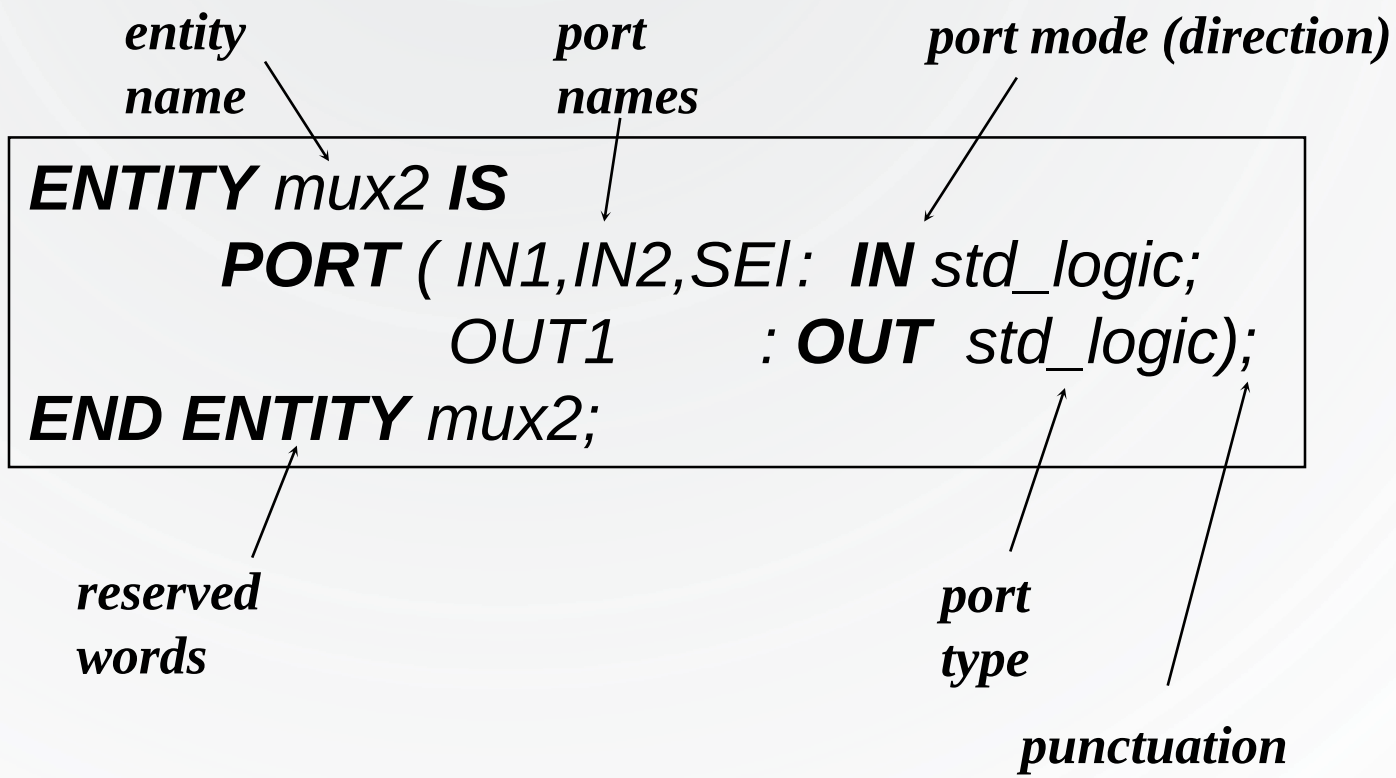
VHDL LANGUAGE: MODEL DEFINITION



Images are from http://vlsiuniverse.blogspot.com.eg/2016/07/multiplexer.html

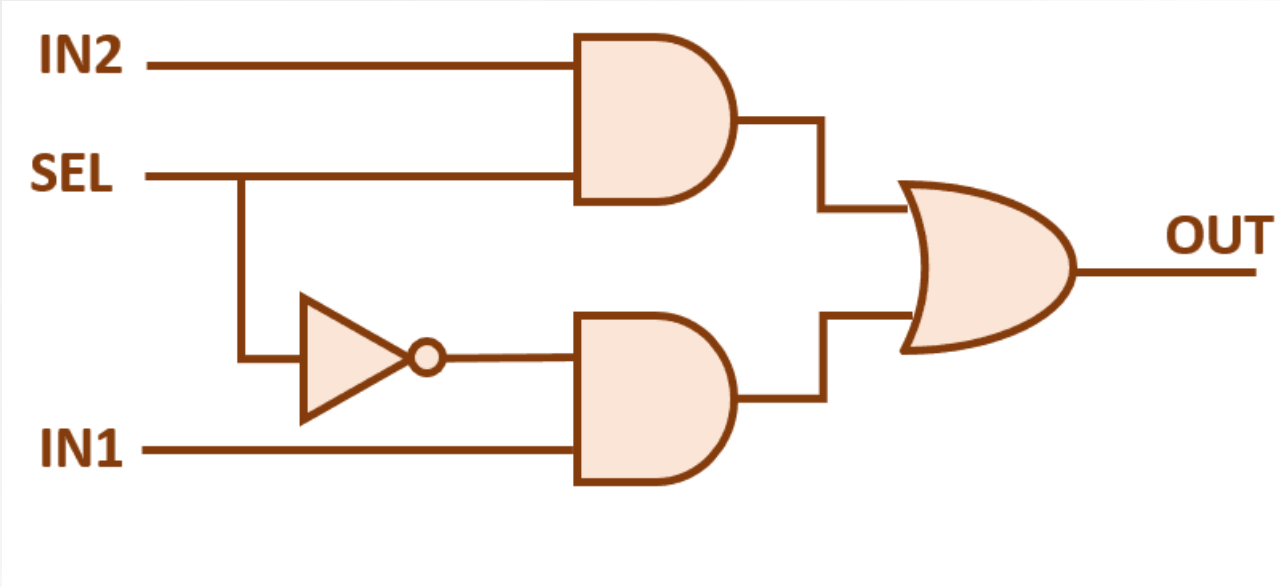
VHDL LANGUAGE: MODEL DEFINITION …(CONT)

- Modules
- Models
- Entities
- Components
- System

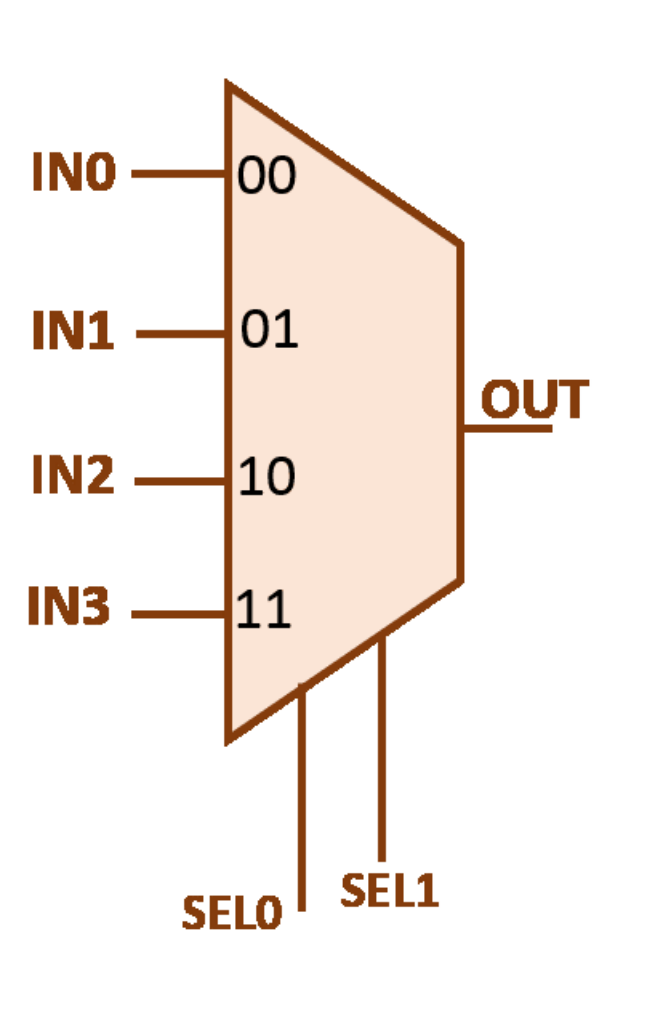


VHDL LANGUAGE: MODEL IMPLEMENTATION

```
ARCHITECTURE  arch1 OF mux2 IS
BEGIN

    out1 <= (in1 and (not Sel))
                or
                (in2 and Sel);

END arch1;
```

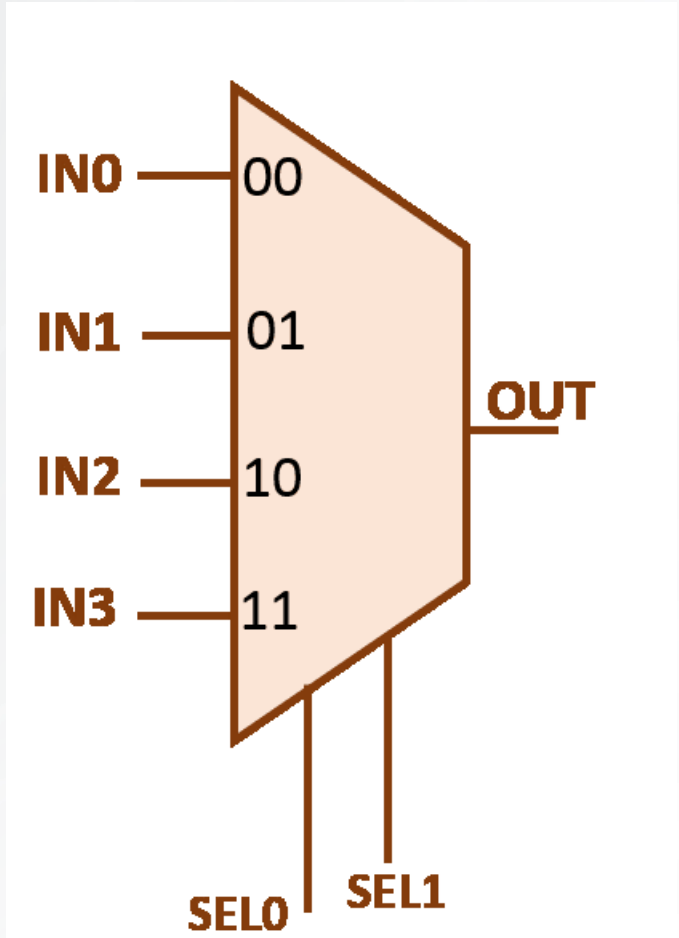


LET'S MAKE A MUX 4X1



MUX 4X1

```
ENTITY mux4 IS
PORT( in0,in1,in2,in3: IN std_logic;
        sel : IN std_logic_vector (1 DOWNTO 0);
        out1: OUT std_logic);
END mux4;
```

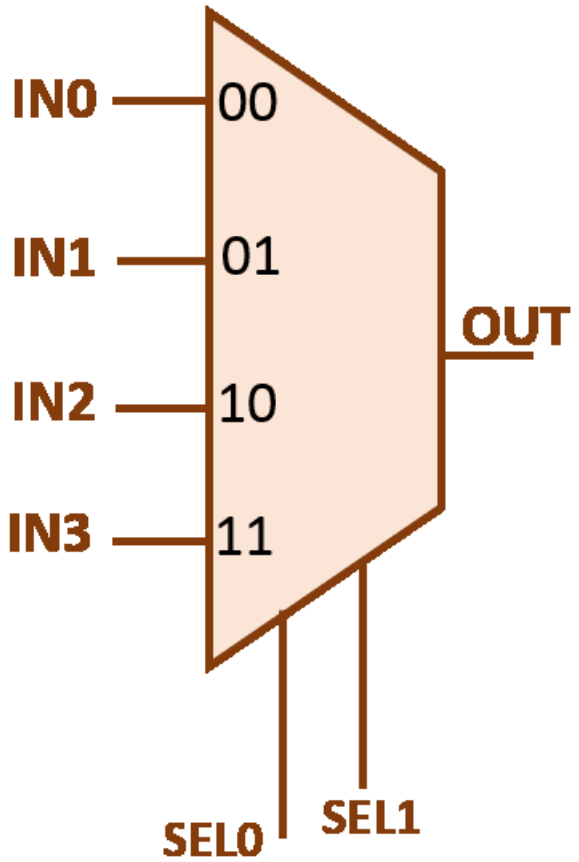


VHDL LANGUAGE: MODEL IMPLEMENTATION

- MODEL Implementation is called Architecture
- Types of Modeling:
  - Dataflow
    - Describes the flow of data within a component/system.
  - Structural
    - Describes the component by the interconnection of lower level components/primitives
  - Behavioral (next time)
    - Describes the functionality of a component/system usually using a process
  - Mixed of the above ways

VHDL LANGUAGE: MODEL IMPLEMENTATION
DATA FLOW

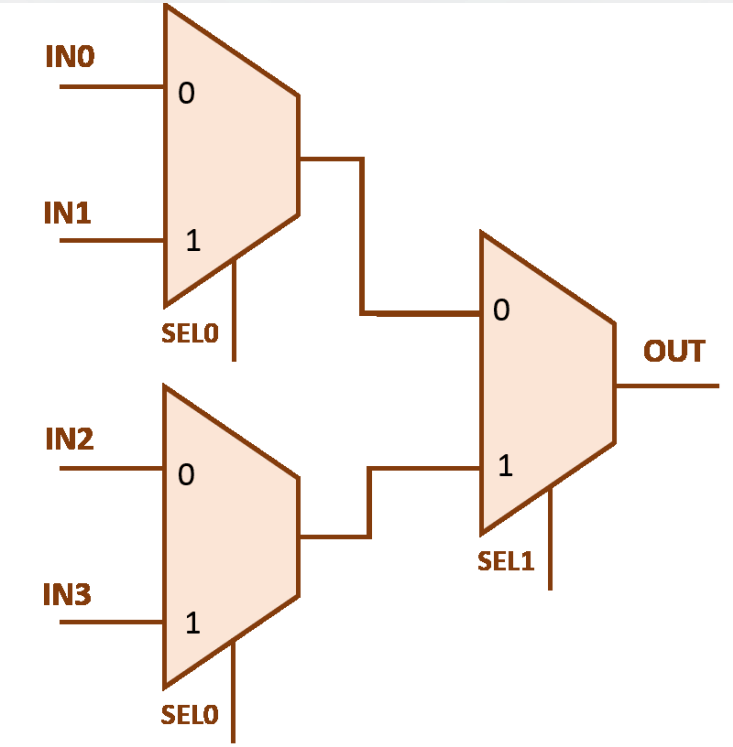
```
ARCHITECTURE  arch1 OF mux4 IS
BEGIN

    out1 <=   in0 WHEN sel(0) = '0' AND sel(1) ='0'
        ELSE in1 WHEN  sel(0) = '0' AND sel(1) ='1'
        ELSE in2 WHEN  sel(0) = '1' AND sel(1) ='0'
        ELSE in3;

END arch1;
```



VHDL LANGUAGE: MODEL IMPLEMENTATION STRUCTURAL

- Can we Make Mux 4x1 from Mux 2x1 ?

VHDL LANGUAGE: MODEL IMPLEMENTATION STRUCTURAL



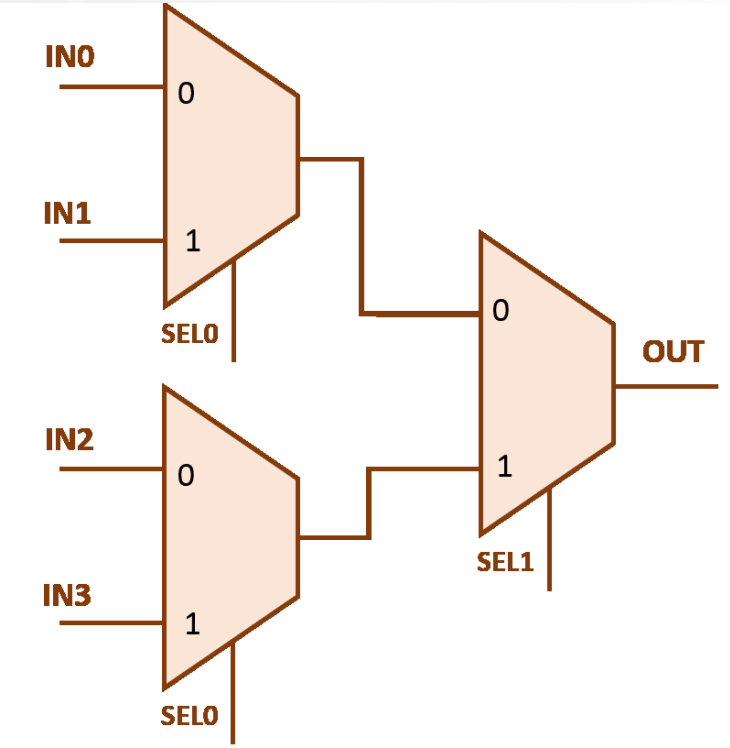
VHDL LANGUAGE: MODEL IMPLEMENTATION
STRUCTURAL

When we use another entity inside our architecture we call it component

```
ARCHITECTURE struct OF mux4 IS
COMPONENT mux2 IS
        PORT (IN1,IN2,SEl:  IN std_logic;
                OUT1  : OUT  std_logic);
    END COMPONENT ;
SIGNAL  x1,x2 : std_logic;
Begin
```

Signal are considered as wires

Signal type
Notice there is no direction for signals



VHDL LANGUAGE: MODEL IMPLEMENTATION STRUCTURAL

When we use another entity inside our architecture we call it component

```
ARCHITECTURE struct OF mux4 IS
COMPONENT mux2 IS
        PORT (IN1,IN2,SEl:  IN std_logic;
              OUT1  : OUT  std_logic);
    END COMPONENT ;
SIGNAL  x1,x2 : std_logic;
Begin
```

Signal are considered as wires

Signal type
Notice there is no direction for signals

- *Till here we told our circuit that there is a component called mux2 and this is how to connect to it but we didn't use it yet*

VHDL LANGUAGE: MODEL IMPLEMENTATION
STRUCTURAL (CONT)

Keyword , whenever I port map a component I put it in the circuit (as hardware IC)

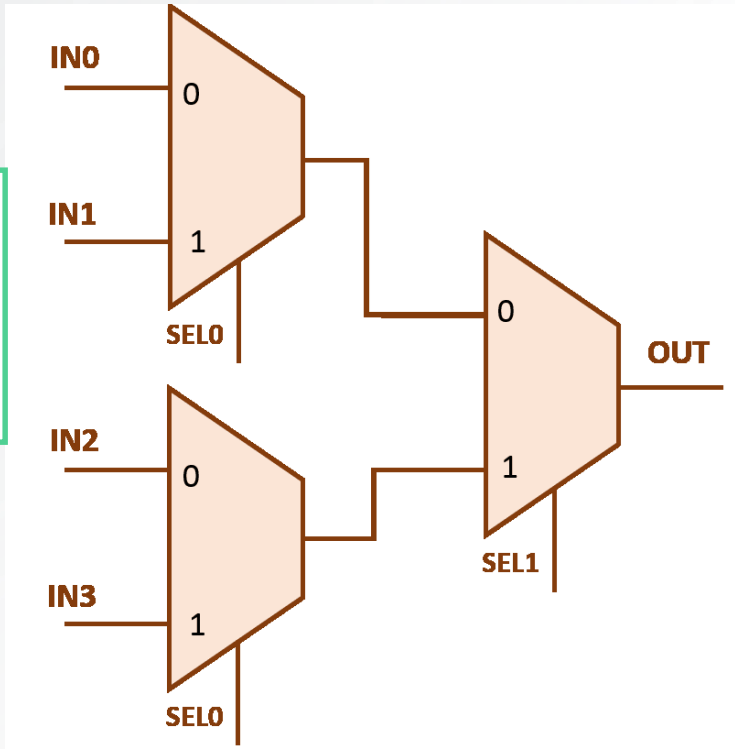
```
BEGIN
    u0: mux2 PORT MAP (in0,in1,s(0),x1);
    u1: mux2 PORT MAP (in2,in3,s(0),x2);
    u2: mux2 PORT MAP (x1,x2,s(1),out1);
END struct;
```

Label to identify component with while tracing

Concurrent statements All executes at the same time

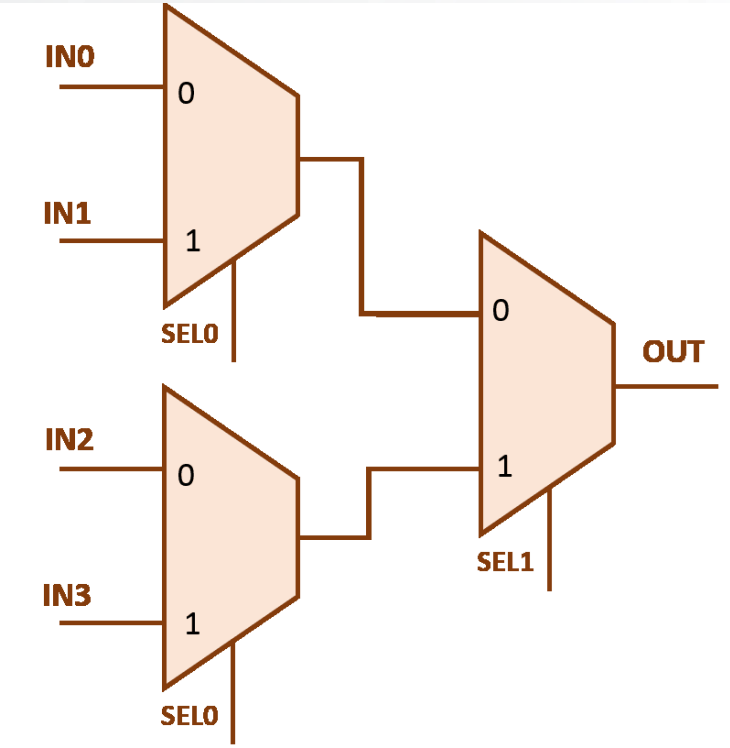
Component name

Parameters sent to the component in the same order they are mentioned in the declaration



VHDL LANGUAGE: MODEL IMPLEMENTATION STRUCTURAL (CONT)

```
BEGIN
    u0: mux2 PORT MAP (in0,in1,s(0),x1);
    u1: mux2 PORT MAP (in2,in3,s(0),x2);
    u2: mux2 PORT MAP (x1,x2,s(1),out1);
END struct;
```



DATA TYPE (STD_LOGIC)

- Include
  - Library ieee;
  - Use ieee.std_logic_1164.all
- Std_logic Possible values
  - 1 or 0
  - X → for conflict
  - Z → high impedance (remember tristate buffer?)
  - U → undefined
  - others
- Think: What do you think possible values of Data Type (bit) ?
- Which is better to use bit or std_logic?

DATA TYPE (STD_LOGIC_VECTOR)

- Std_logic_vector
  - **a,b : in std_logic_vector(2 downto 0)**
  - Can be accessed like b(2) , b(1)
- To access all vector
  - a <= b;
- To access a range of the vector
  - b(1 downto 0) <= a(2 downto 1)
- ‘&’ operator to concatenate two vectors or vector and single element
  - b(1 downto 0) <= a(2) & ‘0’

Still confused !!

- Consider the module as a function or class
- The entity declaration is like a header for the function (interface to our module)
- The architecture is the implementation of the function (how this module behaves inside)
- Whenever I want to use function in another , I have to include that function header (that’s when I introduce the module between architecture and begin)
- The actual usage of the function happens when we call it – function call- this is like port mapping a component But …

Important Consideration

- Whenever you use port map , **you add a component to the circuit** , it is not like a function that you can call it and re-call it. so in our previous example we had 3 mux2 components (i.e. 3 ICs on the breadboard).
- Since port mapping is physical insertion of hardware **it can't be made in a condition** (i.e. x< = mux2 port map(…) when A='0' ; is totally **incorrect**)
- Unlike software , VHDL statement are **concurrent** , they all execute in the same time not after each other (unless in certain cases to be mentioned later)

OBJECTIVES

- Define VHDL and its usage
- Understand different modeling techniques in designing digital circuits (Dataflow , structural)
- Understand the Hardware created from the VHDL code
- Understand Concurrent statement
- Reuse previously created entity (component instantiation a.k.a port mapping)
- Practice designing with VHDL

TIPS

1. Always Save & Compile before simulation.
2. Read the Error/Warning messages in “Transcript” tap.
3. Change the “Radix” to make the simulation easier (right click on signal name in simulation).
4. Re-writing your code all over again will **NOT** solve your problems.
5. For any Error, check the few lines before the line with error message.
6. Always use Do files instead of Changing the inputs every time.

DEMO#1

- Create Project
- Add files to project (and.vhd)
- Compile (Fix Errors)
- Simulate (force, clock, do)