**7. 4:1 Multiplexer using Data flow modeling.**

**AIM : -S**imulation of 4:1 mux using data flow modeling.

**OBJECTIVE: -** T**o** learn data flow modeling style. Its uses and different types of declarations with some different types of circuits. Structure of VHDL program is well

discussed with this modeling style.

**THEORY :-**

**Data flow style:**-

In this type of design, the view of data as flowing from input to output through a design.

An operation is defined in terms of a collection of data transformation expressed as concurrent statement. Each of the statement can be activated when any of its input signal changes its value. While these statements describe the behavior of the circuit, a lot of information about its structure can be extracted from the description as well.

Data flow modeling has a set of concurrent assignment statements. In the data flow level of abstraction we describe how information is passed in the circuit. The built in operators in VHDL are used in expression such as AND, OR, XOR, NOT, etc.

**Functional block diagram of 4:1 Mux:**

Y

S0

S1

b

c

d

a

ENABLE

**7. 4:1 Multiplexer using Data flow modeling**

**Function table of 4:1 Mux:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Enable | S1 | S0 | A | B | C | D | Y |
| 1 | X | X | X | X | X | X | 0 |
| 0 | 0 | 0 | A | X | X | X | A |
| 0 | 0 | 1 | X | B | X | X | B |
| 0 | 1 | 0 | X | X | C | X | C |
| 0 | 1 | 1 | X | X | X | D | D |

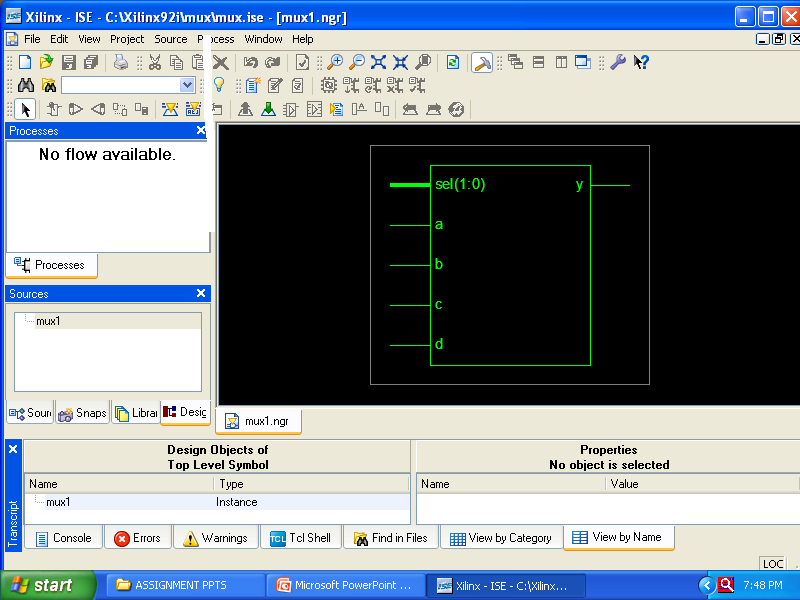
**Design steps:**

* Click on Xilinx ISE 9.2i.
* Create New project from file menu. Ensure top level source is HDL.
* Select family of devices (usually spartan2E or 3)
* Ensure preferred language is VHDL.
* Click new source which shows you project details, device details

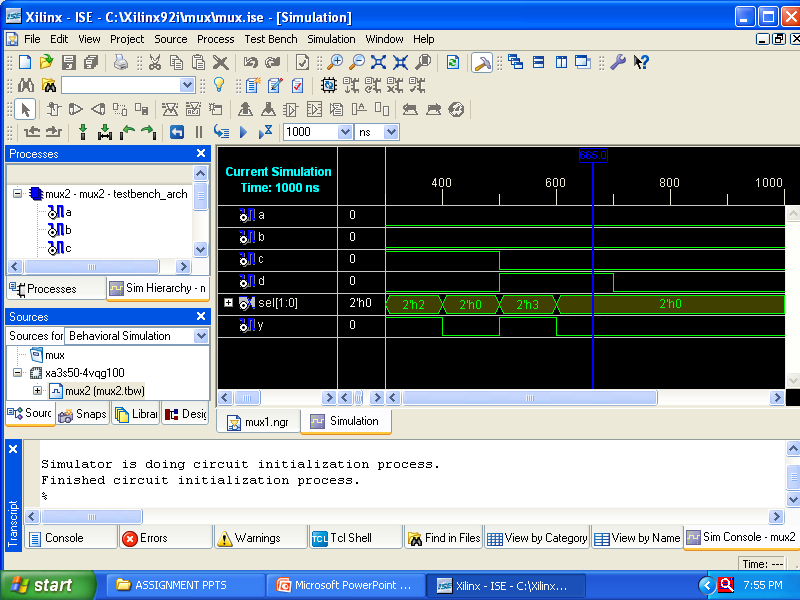
and Synthesis and simulator tools.

* After finishing project create new source by right clicking in project name with VHDL module.
* Complete ports name, directions and bus. Ensure architecture name is behavioral.
* After that we will get design summary and detailed reports.
* Close design summary.
* Create your code with given modeling style.
* Go to process window and synthesis to check if any error is there in code. Check syntax and view RTL schematic.
* Create new source to simulate the code.
* Right click on source name and create test bench waveform with proper name. Ensure the project is same.
* Click on combinational circuit in initial timing wizard.
* Select test bench wave in source window. Apply inputs to wave diagram. Ensure you are in behavioral simulation.
* Go to process box. Click on Xilinx ISE simulator and simulate the model. See results.

**RTL Schematic:**



**Timing Diagram:**



**INPUT:**

* Two select lines, S1,S0
* Four data lines, D0, D1, D2, D3

**OUTPUT:**

* One output line, Y

**FAQ’s:**

1. What are the different kinds of data objects in VHDL code?

**Ans:** data object may be any value or number; still some signal data objects and bit and bit\_vector types of these are available in VHDL. STD\_LOGIC and STD\_LOGIC\_VECTOR types are used widely in programming.

1. What do you mean by signal?

**Ans**: signal is a data object represents logic signals or wires in a circuit. There are three places in which signals can be declared in VHDL code: in an entity declaration, in the declarative section of architecture, and in declarative section of package.

1. What are different signal types?

**Ans**: BIT, BIT\_VECTOR, STD\_LOGIC, STD\_LOGIC\_VECTOR, STD\_ULOGIC, SIGNED, UNSIGNED, INTEGER, ENUMERATION, and BOOLEAN.

1. What is an *entity*?

**Ans:** a circuit or sub circuit described with VHDL code is called a design entity.

1. Explain structure of an *entity.*

**Ans:** it has two main parts: the entity declaration, which specifies the input and output signals for entity, and the architecture, which gives circuit details.

Entity

Architecture

Declaration

1. How will you declare a package with component? Explain structure?

The general form of package declaration is as shown in diagram.

**7. 4:1 Multiplexer using Data flow modeling**

PACKAGE package\_name IS

[TYPE declarations]

[SIGNAL declarations]

[COMPONENT declarations]

END package\_name;

**PRACTICE ASSIGNMENTS:**

1. Implement 8:1 Mux using data flow modeling.

2. Implement 8;1 mux using 4:1 mux.

3. Implement 4:16 decoder using data flow style.

4. Implement active high 3:8 decoder.

5. Write VHDL code for decimal to BCD Encoder.