**8. Full Adder using VHDL**

**AIM : -S**imulation of Full Adder using structural modeling.

**OBJECTIVE:** T**o** learn structural 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 :- Structural style:**-

A digital electronic system can be described as a module with inputs and/or outputs. The electrical values on the outputs are same functions of the values of inputs. The example of it is as shown. The NAND gate has 2i/ps, A&B, & an output y. Using VHDL terminology, we call the NAND2 design entity & the inputs & outputs are called ports. 1 way of describing the function of NAND2 is to describe how it is composed of sub modules AND & INVERTER Each of the Sub modules is an instance of some entity & ports of the instances are connected using signals. Structural modeling has a set of interconnected component. Structure can be used to create a very low level description of a circuit or a very high level description.

In a gate level description of a circuit, for example components such as basic logic gates & F/Fs might be connected in some logical stricture to create the circuit. This is what is often called a net list.

**8. Full Adder using Structural modeling.**

**Structure of structural modeling:**

**architecture Netlist of Half\_Adder is**

**-- component with locals**

**component MyXor port (A\_Xor,B\_Xor : in BIT; Z\_Xor : out BIT);**

**end component;**

**-- component with locals**

**component MyAnd port (A\_And,B\_And : in BIT; Z\_And : out BIT);**

**end component;**

**begin**

**Xor1: MyXor port map (X, Y, Sum);**

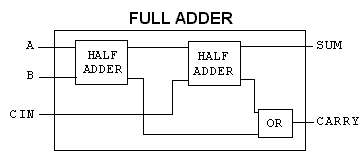
**-- instance with actuals**

**And1 : MyAnd port map (X, Y, Cout);**

**-- instance with actuals**

**end;**

**Functional block diagram of Full Adder:**

****

**Function table of Full Adder:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | SUM | CARRY |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Sum = Σm(1, 2, 4, 7)**

**Carry= Σm(3, 5, 6, 7)**

**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

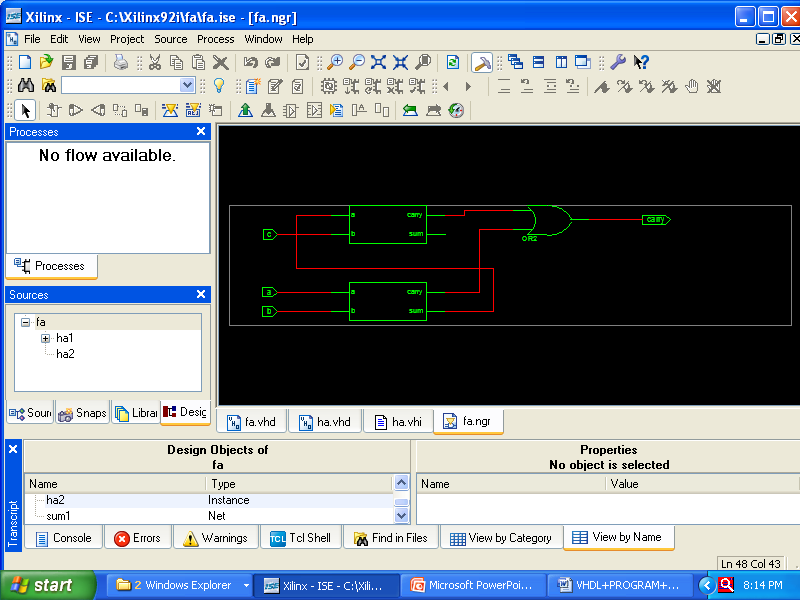
and Synthesis and simulator tools.

* After finishing project create new source(full adder) 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.
* Click new source which shows you project details, device details

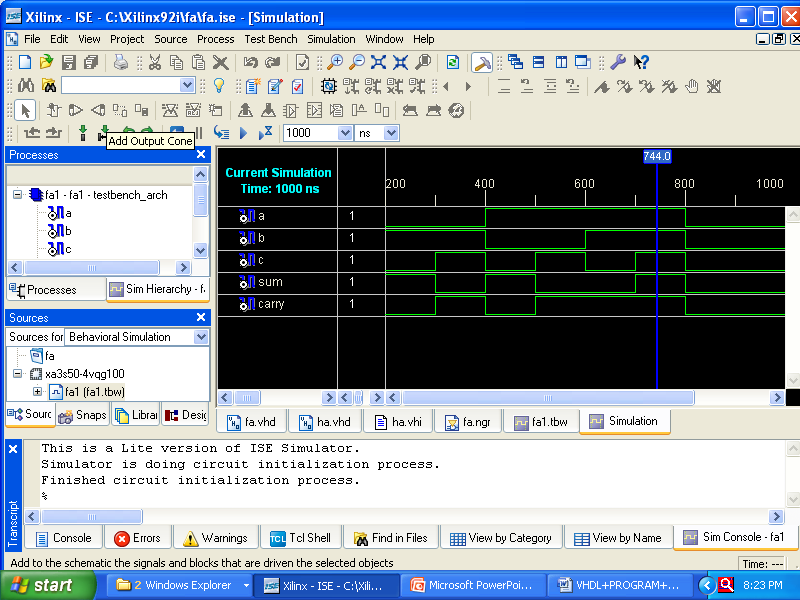
and Synthesis and simulator tools.

* After finishing project create new source(half adder) by right clicking in project name with VHDL module.
* Complete ports name, directions and bus. Ensure architecture name is behavioral. Names should be same as previous inputs.
* After that we will get design summary and detailed reports.
* Close design summary
* Create your code for half adder with given modeling style.
* After that go to process window. Click on design utilities.
* View HDL instantiation code of half adder.
* Copy component part of half adder in architecture block of full adder. Copy instantiation template twice after begin as we require two half adders to construct full adder (structural modeling). Name them differently (e.g. ha1 or ha2).
* Create signals as per the logic. Complete code of full adder.
* 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:**

* Three inputs A, B, Carry (previous)

**OUTPUT:**

* Two outputs Sum, Carry

**FAQ’s:**

1. What do you mean by structural style?

**Ans :** in this type of code a circuit is described in hierarchical fashion, by connecting together sub circuits.

1. Explain IF statement with example.

**Ans:** The general form of an IF statement is as follws:

**IF Sel=’0’ THEN**

**F<=x1;**

**ELSE**

**F<=x2;**

**END IF;**

1. Explain declaration and instantiation process in VHDL.

**Ans:** first declare a small vhdl code in formal way. E.g. half adder. Define project name is full adder. And simply copy instantiation template in main source code. Define how many times you want to declare this template. Give separate names for those**.**

1. What is STD\_LOGIC\_VECTOR?

**Ans:** It is standard data object added to VHDL standard in IEEE 1164**.** It provides more flexibility than the Bit type.

1. Explain difference between concurrent and sequential statements?

**Ans:** Concurrent statements define interconnected processes and blocks that together describe a design’s overall behavior or structure. They can be grouped using block s**13.**

Groups of blocks can also be partitioned into other blocks. At this same

level, a VHDL component can be connected to define signals within the blocks.  
 It is a reference to an entity. A process can be a single signal assignment statement or a series of sequential statements (SS). Within a process, procedures and functions can partition the sequential statements

**PRACTICE ASSIGNMENTS:**

1. Implement half adder using data flow modeling.

2. Implement Full Adder using behavioral modeling.

3. Implement 3 bit magnitude comparator using a 3 bit adder.

4. Implement half sub tractor using data flow modeling.

5. Write VHDL code for Full Subtractor.