**Institute of Engineering & Management**

**Department of Computer Science & Engineering**

**Computer Architecture Laboratory for 2nd year 4th semester 2018**

**Code: CS 493**

**Date:** 20/01/18

**WEEK-1**

**Assignment-1:** Implementation of AND, OR, NOT, XOR, NAND, NOR gates using Xilinx ISE.

**Objective:** To implement OR gate

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value c for both data flow and behavioral model using if and else statement with required condition.
3. Make test bench waveform and check for the given inputs.

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **Y** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Behavioral Model:**

**Data flow Model:**

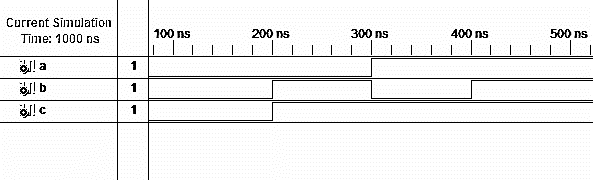
**Code:**

**1. Behavioral Model Code:**

entity orgate is: port( a: in std\_logic;  
 b: in std\_logic;   
 c: out std\_logic);  
 end orgate;   
architectural behaviour of orgate is:   
begin Process(a,b)   
begin   
 if(a=’0’ and b=’0’)  
 then c<=’0’;  
 else c<=’1’;   
end if;  
 end process;  
 end behavioral;

**2. Data flow Model Code:**

entity orgate is:   
port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);   
end orgate;  
 architectural Dataflow of orgate is:   
begin   
c<=a or b;   
end Dataflow;

**Output:**

**Objective:** To implement AND gate

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value c for both data flow and behavioral model using if and else statement with required condition.
3. Make test bench waveform and check for the given inputs.

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **Y** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Behavioral Model:**

**Data flow Model:**

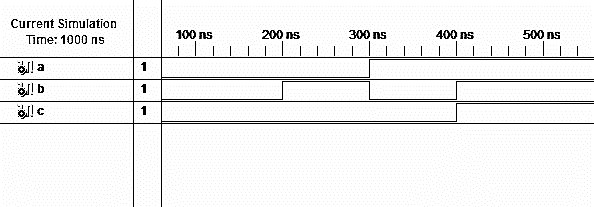
**Code:**

**1. Behavioral Model Code:**

entity andgate is:  
 port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);  
 end andgate;  
 architectural behaviour of andgate is:  
 begin Process(a,b)   
begin   
if(a=’1’ and b=’1’) then  
 c<=’1’;  
 else c<=’0’;  
 end if;   
end process;   
end behavioral;

**2. Data flow Model Code:**

entity andgate is:   
port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);   
end andgate;   
architectural Dataflow of andgate is:  
 begin   
c<=a and b;   
end Dataflow;

**Output:**

**Objective:** To implement NOT gate

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value c for both data flow and behavioral model using if and else statement with required condition.
3. Make test bench waveform and check for the given inputs.

**Truth Table:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| **A** | **B** |
| 0 | 1 |
| 1 | 0 |

**Behavioral Model:**

**Data flow Model:**

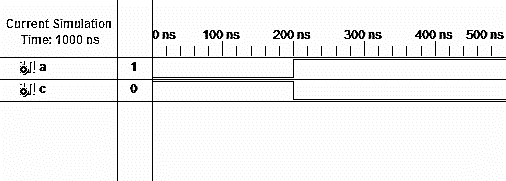
**Code:**

**1. Behavioral Model Code:**

entity notgate is:  
 port( a: in std\_logic;   
 c: out std\_logic);   
end notgate;   
architectural behaviour of notgate is:   
begin   
Process(a)   
begin   
if(a=’1’)   
then   
c<=’0’;   
else c<=’1’;  
end if;   
end process;   
end behavioral;

**2. Data flow Model Code:**

entity notgate is:  
 port( a: in std\_logic;   
 c: out std\_logic);  
 end notgate;   
architectural Dataflow of notgate is:   
begin   
c<=not a;   
end Dataflow;

**Output:**

**Objective:** To implement NOR gate

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value c for both data flow and behavioral model using if and else statement with required condition.
3. Make test bench waveform and check for the given inputs.

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **Y** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**Behavioral Model:**

**Data flow Model:**

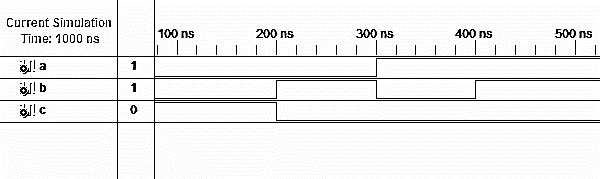
**Code:**

**1. Behavioral Model Code:**

entity norgate is:  
 port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);   
end norgate;   
architectural behaviour of norgate is:  
 begin   
Process(a,b)   
begin   
 if(a=’0’ and b=’0’)  
 then  
 c<=’1’;  
 else c<=’0’;  
 end if;   
end process;   
end behavioral;

**2. Data flow Model Code:**

entity norgate is:  
 port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);   
end norgate;   
architectural Dataflow of norgate is:  
 begin   
c<=a nor b;   
end Dataflow;

**Output:**

**Objective:** To implement NAND gate

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value c for both data flow and behavioral model using if and else statement with required condition.
3. Make test bench waveform and check for the given inputs.

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **Y** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Behavioral Model:**

**Data flow Model:**

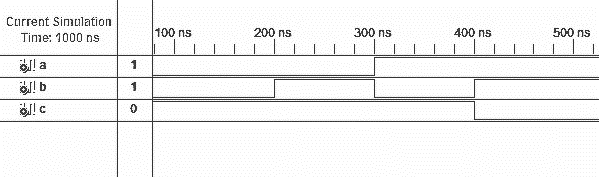
**Code:**

**1. Behavioral Model Code:**

entity nandgate is:   
port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);   
end nandgate;  
 architectural behaviour of nandgate is:  
 begin   
Process(a,b)  
 begin   
if(a=’1’ and b=’1’)  
 then   
c<=’0’;  
 else c<=’1’;  
 end if;  
end process;   
end behavioral;

**2. Data flow Model Code:**

entity nandgate is:  
 port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);  
 end nandgate;  
 architectural Dataflow of nandgate is:  
 begin   
c<=a nand b;   
end Dataflow;

**Output:**

**Objective:** To implement XOR gate

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value c for both data flow and behavioral model using if and else statement with required condition.
3. Make test bench waveform and check for the given inputs.

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **Y** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Behavioral Model:**

**Data flow Model:**

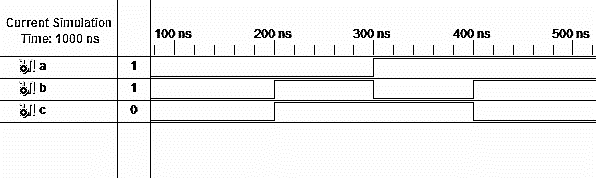
**Code:**

**1. Behavioral Model Code:**

entity xorgate is:  
 port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);   
end xorgate;  
 architectural behaviour of xorgate is:   
begin   
Process(a,b)  
 Begin  
 if(a= b)  
 then   
c<=’0’;   
else c<=’1’;   
end if;   
end process;   
end behavioral;

**2. Data flow Model Code:**

entity xorgate is:   
port( a: in std\_logic;   
 b: in std\_logic;   
 c: out std\_logic);   
end xorgate;   
architectural Dataflow of xorgate is:  
 begin   
c<=a xor b;   
end Dataflow;

**Output:**

**Assignment-2:** Implementation of Half Adder and Full Adder using Xilinx ISE.

**Objective:** To implement Half Adder

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value of ‘c’ (c<= a and b) and ‘s’ (s<= a xor b) for dataflow and behavioral model using if and else statements with required conditions.
3. Make test bench waveform and check for the given inputs.

**Truth Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | | **OUTPUT** | |
| **A** | **B** | **S** | **Cout** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Data flow Model:**

**Behavioral Model:**

**Code:**

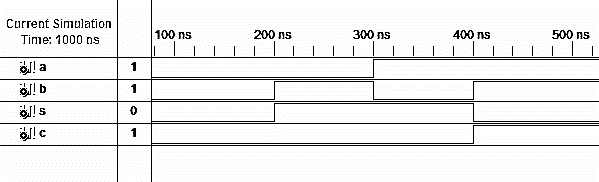
**1. Behavioral Model Code:**

entity halfadder is:

port( a: in std\_logic;  
 b: in std\_logic;   
 s: out std\_logic;   
 c: out std\_logic);   
end halfadder;   
architectural behaviour of halfadder is:   
begin   
Process(a,b)  
 begin if(a=b) then   
s<=a;   
c<=a;   
else s<=’1’;   
c<=’0’;   
end if;   
end process;   
end behavioral;

**2. Data flow Model Code:**

entity halfadder is:  
port( a: in std\_logic;   
 b: in std\_logic;   
 s: out std\_logic;   
 c: out std\_logic);   
end halfadder;   
architectural Dataflow of halfadder is:   
begin  
 s<=a xor b;   
c<=a and b;   
end Dataflow;

**Output:**

**Objective:** To implement Full Adder

**Software used:**

|  |  |
| --- | --- |
| Property Name | Value |
| Device family | Spartan 3 |
| Device | XC3S50 |
| Package | PQ208 |
| Speed | -5 |
| Top-level source type | HDL |
| Synthesis Tool | XST(VHDL/Verilog) |
| Simulator | ISE Simulator |
| Preferred Language | VHDL |

**Theory:**

1. Make VHDL with required port specification.
2. Calculate the value of s (s<= a xor b xor c) and C0(c<=(a and ) or ( b and c) or( c and a)) for both behavioral and dataflow model using if and else statement.
3. Make test bench and check for the given input.

**Truth Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INPUT** | | | **OUTPUT** | |
| **A** | **B** | **Cin** | **S** | **Cout** |
| 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 |

**Data flow Model:**

**Behavioral Model:**

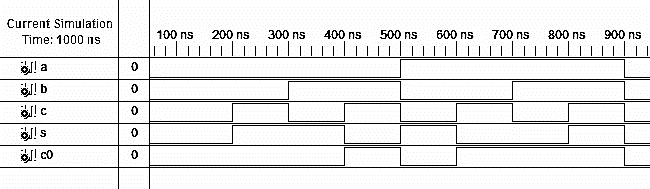
**Code:**

**1. Behavioral Model Code:**

entity fulladder is:   
port( a: in std\_logic;   
 b: in std\_logic;   
 cin: in std\_logic;   
 s: out std\_logic;   
 c: out std\_logic);   
end fulladder;   
architectural behaviour of fulladder is:   
begin   
Process(a,b,cin)   
begin   
if(a=b) then   
s<=cin; c<=a;   
else   
s<=(not cin);   
c<=cin;   
end if;   
end process;   
end behavioral;

**2. Data flow Model Code:**

entity fulladder is:   
port( a: in std\_logic;   
 b: in std\_logic;   
 cin: in std\_logic;   
 s: out std\_logic;   
 c: out std\_logic);   
end fulladder;  
 architectural Dataflow of fulladder is:  
 begin   
s<=(a xor b) xor cin; c<=(a and b) or (b and c) or (a and c);   
end Dataflow;

**Output:**