ECGR 4146 Introduction to VHDL
Lab 4 - 2-bit Comparator
Jonathon Nguyen
ID: 801093003

Objective of the lab:

The objective of the lab is to create a 2-bit comparator design file and a test bench to test the comparator. The 2-bit comparator will be designed with three types of modeling. The three types are dataflow, structural, and behavioral.

VHDL code for the 2-bit Comparator:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity comparator2Bit is
    port( a, b : in std_logic_vector(1 downto 0);
        eq : out std_logic
    );
end comparator2Bit;

architecture Dataflow of comparator2Bit is
    signal s0, s1, s2, s3: std_logic;
begin

    s0 <= (not a(0)) and (not a(1)) and (not b(0)) and (not b(1));
    s1 <= a(0) and (not a(1)) and b(0) and (not b(1));
    s2 <= (not a(0)) and a(1) and (not b(0)) and b(1);
    s3 <= a(0) and a(1) and b(0) and b(1);

    eq <= s0 or s1 or s2 or s3;

end dataflow;

architecture Behavioral of comparator2Bit is
begin

    process(a, b)
    begin

        if (a = b) then
            eq <= '1';
        else
            eq <= '0';
        end if;

    end process;

end Behavioral;
```

```
architecture Structural of comparator2Bit is

    component comparator1Bit
        port( x, y : in std_logic;
        eq : out std_logic );
    end component;

    signal eq0, eq1: std_logic;

begin

    comparatorbit0: comparator1Bit
        port map (x => a(0), y => b(0), eq => eq0);

    comparatorbit1: comparator1Bit
        port map (x => a(1), y => b(1), eq => eq1);

    eq <= eq0 and eq1;

end Structural;
```

The entity was made with two 2-bit inputs, a and b, and an output port of eq. If those inputs are equal, the output eq will be set to 1.

In the design file, there is three architecture. The data flow was made by looking at the circuit of the 2-bit comparator and using a logical operator to make the output.

The behavioral uses a process, and in the process, it checks if the two inputs are equal and sets the output accordingly.

The structural uses two 1-bit comparators and instance them. Then the output of the two comparators is anded together to produce the output bit.

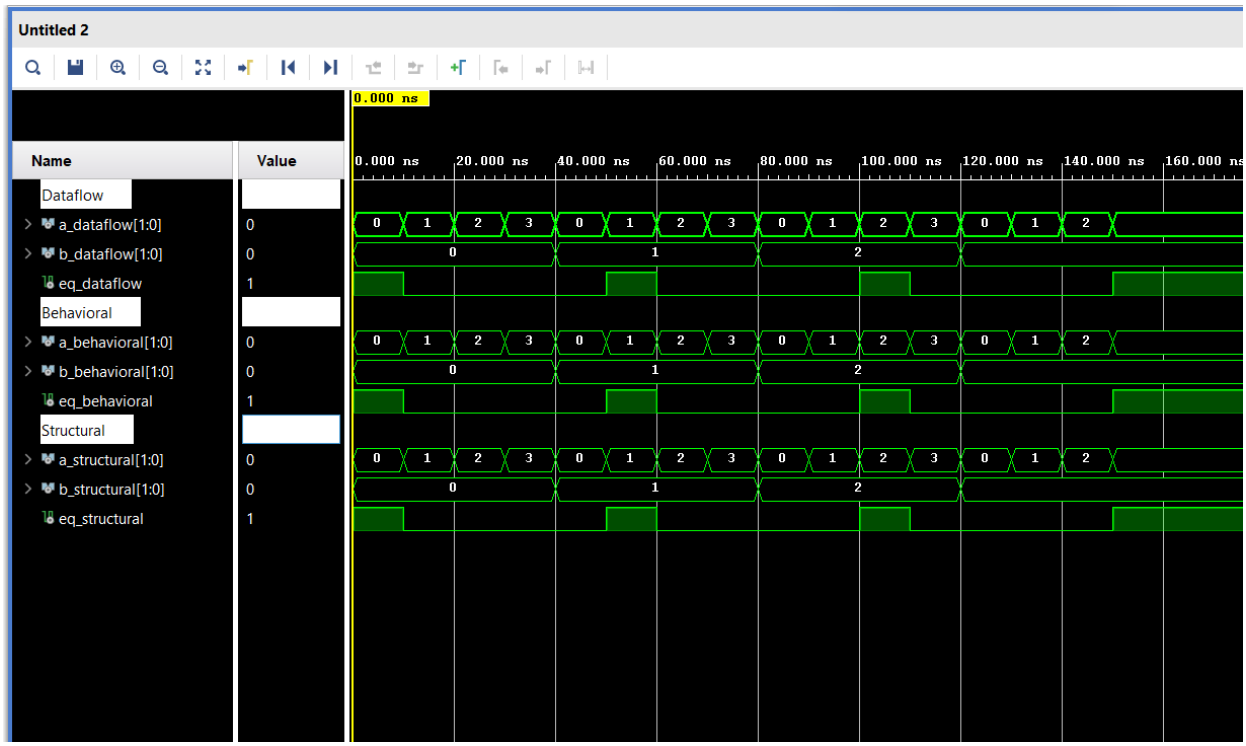
Testbench code for the 2-bit Comparator(Dataflow, Behavioral, Structural):

```
library IEEE;
library work;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity comparator2Bit_TB is
--  Port ( );
end comparator2Bit_TB;

architecture Behavioral of comparator2Bit_TB is

    component comparator2Bit is
        port( a, b : in std_logic_vector(1 downto 0);
            eq : out std_logic
        );
    end component;

    signal a_dataflow, b_dataflow: std_logic_vector(1 downto 0);
    signal eq_dataflow: std_logic;

    signal a_behavioral, b_behavioral: std_logic_vector(1 downto 0);
    signal eq_behavioral: std_logic;

    signal a_structural, b_structural: std_logic_vector(1 downto 0);
    signal eq_structural: std_logic;

begin

    dataflow_comparator2Bit: entity work.comparator2Bit(Dataflow)
    port map(a => a_dataflow, b => b_dataflow, eq => eq_dataflow);

    behavioral_comparator2Bit: entity work.comparator2Bit(Behavioral)
    port map(a => a_behavioral, b => b_behavioral, eq => eq_behavioral);

    structural_comparator2Bit: entity work.comparator2Bit(Structural)
    port map(a => a_structural, b => b_structural, eq => eq_structural);

    process
        variable temp : std_logic_vector(3 downto 0);
    begin
    for i in 0 to 15 loop
            temp := std_logic_vector(to_unsigned(i, 4));
            a_dataflow(0) <= temp(0);
            a_dataflow(1) <= temp(1);
            b_dataflow(0) <= temp(2);
            b_dataflow(1) <= temp(3);

            a_behavioral(0) <= temp(0);
            a_behavioral(1) <= temp(1);
            b_behavioral(0) <= temp(2);
            b_behavioral(1) <= temp(3);

            a_structural(0) <= temp(0);
            a_structural(1) <= temp(1);
            b_structural(0) <= temp(2);
            b_structural(1) <= temp(3);
            wait for 10 ns;
    end loop;

    wait;
    end process;

end Behavioral;
```

In the architecture, the 2-bit comparator was defined as a component with all the ports it needs. Nine different signals was made to connect to the comparators.

The comparator was instancated while also connecting the signals to it. A process was defined with no senetivity list. In the process, A for loop was used to loop through all 16 input states. The integer I is casted to a unsigned integer and then converted to a standard vector. Then the vector was split up for the different input. The dataflow, behavioral and the structural all have the same input.

Simulation Waveform for the 2-bit Comparator (Three architectures):



Conclusion

The simulation was ran on the testbench code and produce this waveform. Looking at the waveform, all three architectures was designed correctly and behavied correctly. The two bit comparator was correcty coded.