Digital System Design Lab Report, Lab3, Homework2

Author: Ying Yiwen
Number: 12210159

Abstract

This lab report presents the design, simulation, and analysis of a digital system using VHDL. The primary objective is to explore the differences between signal and variable types in VHDL through practical implementation. The report includes the VHDL code for a digital circuit, the corresponding testbench code for verification, theoretical analysis of the circuit's behavior, and simulation results obtained from behavioral simulation, schematic simulation, post-synthesis timing simulation, and post-implementation timing simulation. The findings confirm the correctness of the VHDL code and provide insights into the distinct characteristics and applications of signals and variables in digital system design.

Contents

1 Introduction

In digital system design, particularly when working with hardware description languages like VHDL, understanding the distinction between signals and variables is crucial. This experiment delves into the practical application of VHDL, focusing on the differences between signal and variable types through the implementation and simulation of a digital circuit.

Signals in VHDL are used for communication between different parts of a design. They maintain their values until explicitly updated and are suitable for representing hardware connections. Signals are updated at the end of a simulation cycle, making them ideal for modeling hardware components that maintain their state until new values are assigned.

Variables, on the other hand, are used within processes and retain their values only during the execution of the process. They are updated immediately within the process and are ideal for internal calculations and temporary storage within a process. Variables are not visible outside the process they are declared in, making them suitable for local computations.

This lab aims to clarify these distinctions by implementing a digital circuit using both signal and variable types. The VHDL code for the circuit, along with a testbench for verification, is provided. Theoretical analysis of the circuit's behavior is conducted, and simulation results from various stages—behavioral simulation, schematic simulation, post-synthesis timing simulation, and post-implementation timing simulation—are presented and compared. This comprehensive approach not only reinforces theoretical knowledge but also enhances practical skills in VHDL coding and digital system simulation.

2 VHDL Code

The VHDL code is as follows:

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

-- entity declaration
entity sig_var is
Port ( x : in STD_LOGIC;
       y : in STD_LOGIC;
       z : in STD_LOGIC;
       res1 : out STD_LOGIC;
       res2 : out STD_LOGIC);
end sig_var;

-- architecture declaration
architecture Behavioral of sig_var is

-- signal declaration
signal sig_s1, sig_s2 : std_logic;

-- logic function
begin

-- process declaration, variable type
proc1: process (x, y, z) is
variable var_s1, var_s2 : std_logic;
```

```
begin
    var_s1 := x and y;
    var_s2 := var_s1 xor z;
    res1 <= var_s1 nand var_s2;
end process proc1;

-- process declaration, signal type
proc2: process (x, y, z) is
begin
    sig_s1 <= x and y;
    sig_s2 <= sig_s1 xor z;
    res2 <= sig_s1 nand sig_s2;
end process proc2;

end Behavioral;
```

3 Testbench Code

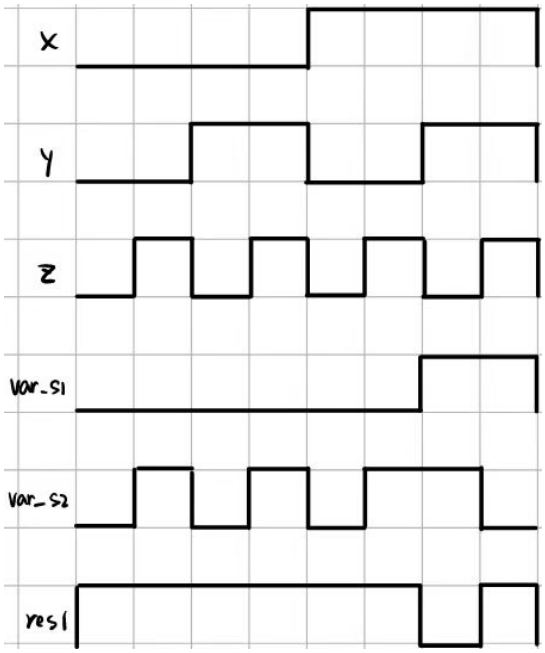
The testbench code is as follows, and it can test the correctness of the VHDL code.

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

-- entity declaration
entity testbench is
--  Port ( );
end testbench;

-- architecture declaration
architecture tb of testbench is
-- component declaration
component sig_var is
Port ( x : in STD_LOGIC;
        y : in STD_LOGIC;
        z : in STD_LOGIC;
        res1 : out STD_LOGIC;
        res2 : out STD_LOGIC);
end component;

-- signal and constant declaration
signal x, y, z, r1, r2 : std_logic ;
constant period: time := 10ns;

-- logic function
begin
-- instantiation
uut: sig_var port
    map( x => x,
            y => y,
            z => z,
            res1 => r1,
            res2 => r2);
```

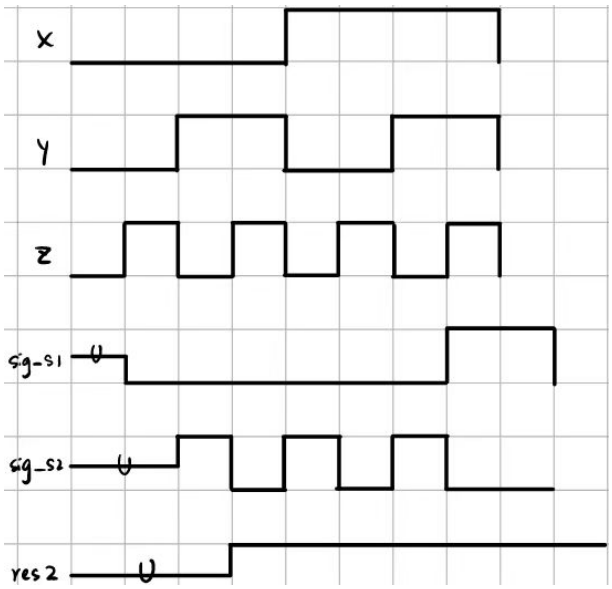
```
-- test case
x <= '0' after period * 0, '1' after period * 4;
y <= '0' after period * 0, '1' after period * 2, '0' after period * 4, '1' after
    period * 6;
z <= '0' after period * 0, '1' after period * 1, '0' after period * 2, '1' after
    period * 3,
         '0' after period * 4, '1' after period * 5, '0' after period * 6, '1' after
            period * 7;

end tb;
```

4 Therotical Analysis

The theoretical analysis is as follows:



(a) register1



(b) register2

Figure 1: Theoretical Analysis

5 Simulation Result

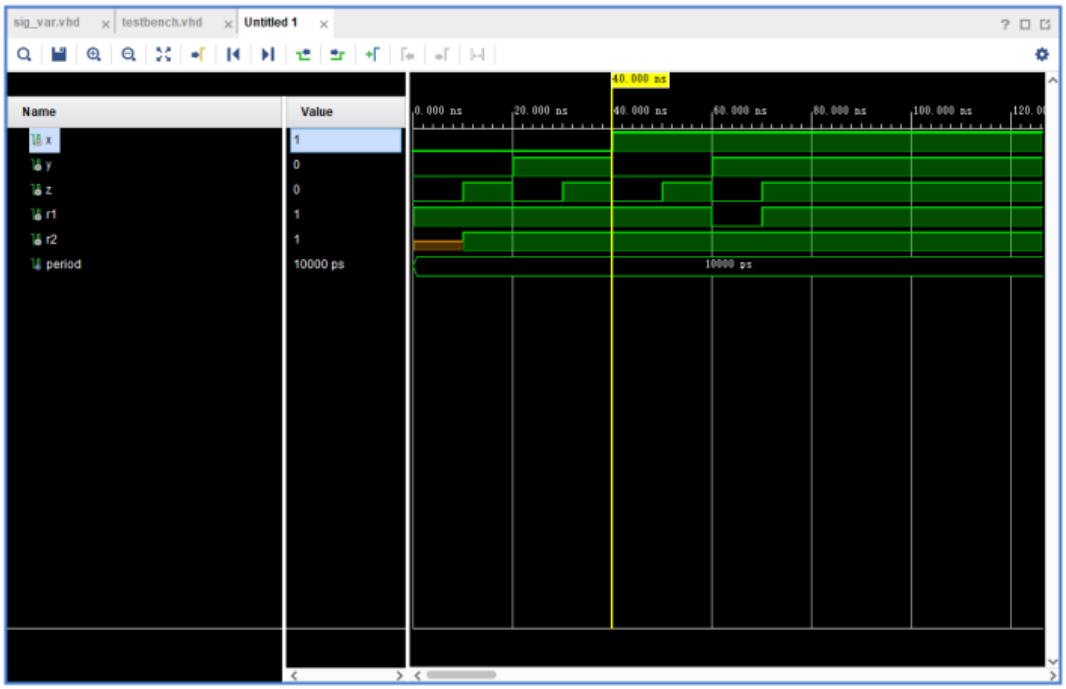


Figure 2: Behavior Simulation Result

The behavior simulation is the same as the theoretical analysis.

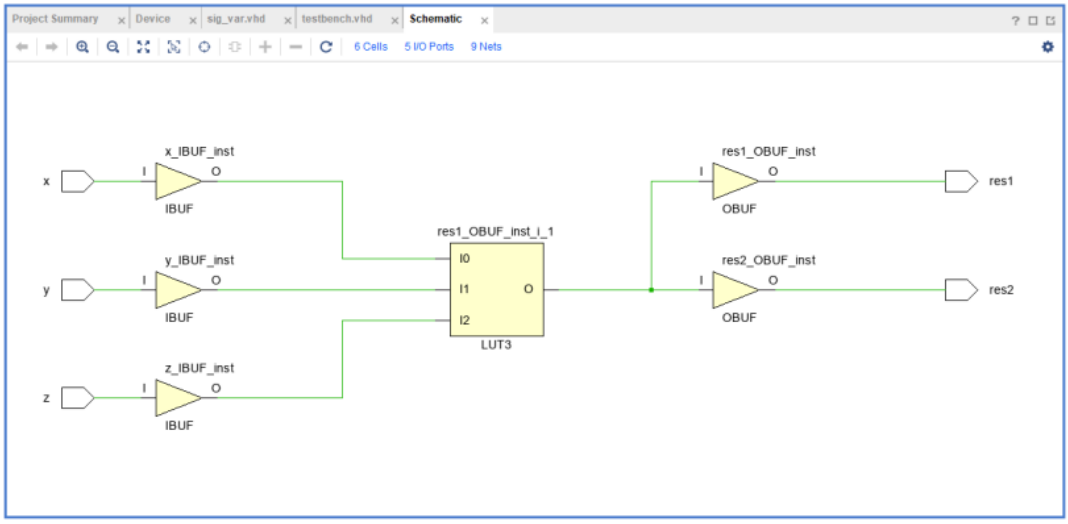


Figure 3: Schematic Simulation Result

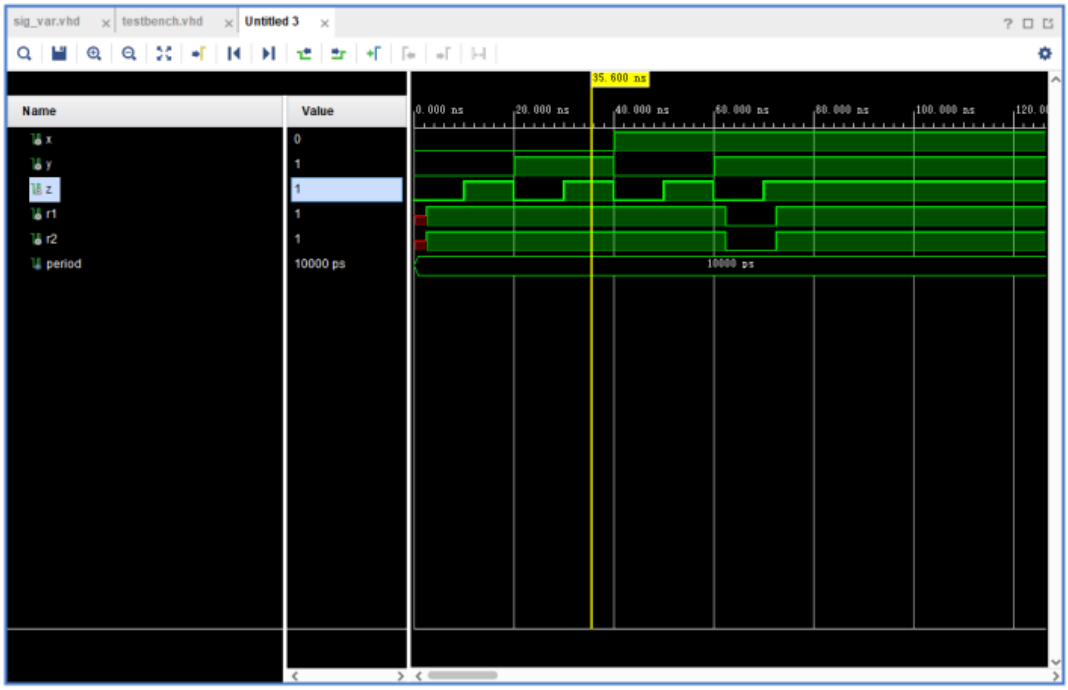


Figure 4: Post Synthesis Timing Simulation Result

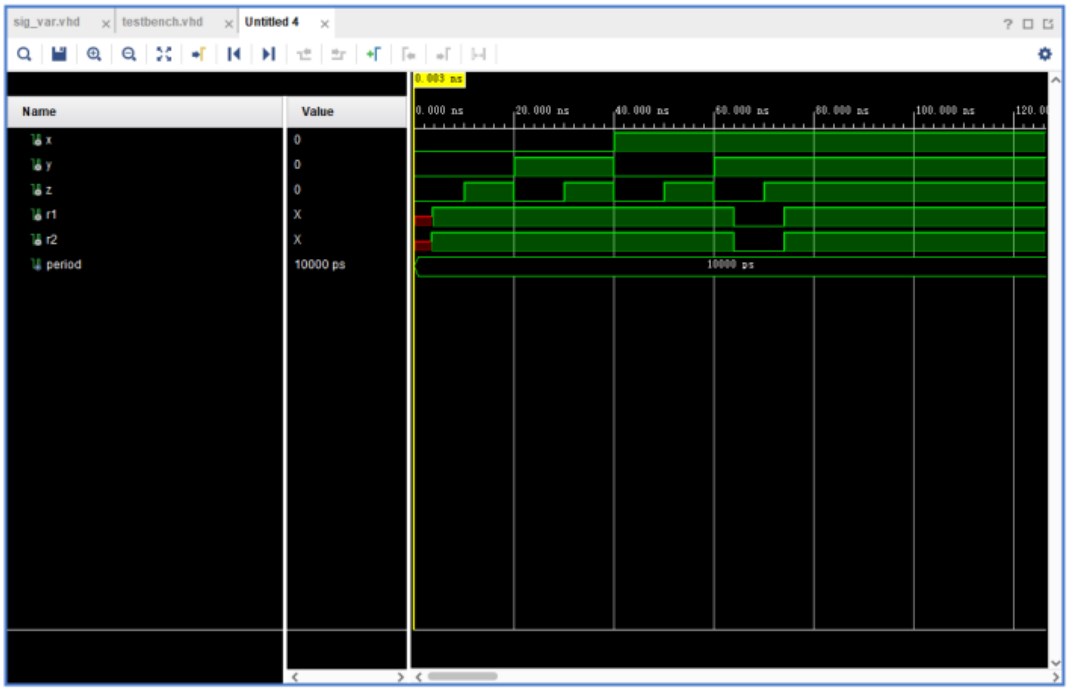


Figure 5: Post Implementation Timing Simulation Result

6 Conclusion

This experiment successfully demonstrated the differences between signal and variable types in VHDL through the design and simulation of a digital circuit. The VHDL code was correctly implemented, and the testbench effectively verified its functionality. The theoretical analysis aligned with the simulation results, confirming the proper operation of the circuit. Through this lab, the importance of understanding the distinct behaviors of signals and variables in digital system design was highlighted. Signals are suitable for inter-component communication and maintaining state information, while variables are ideal for internal calculations within processes.