**基于fpga的数字系统设计 实验报告**

|  |  |  |
| --- | --- | --- |
| 评 语 | **成绩** |  |
| 教 师：  年 月 日 | | |

**学院班级： 171301201**

**学生学号： 17130120116**

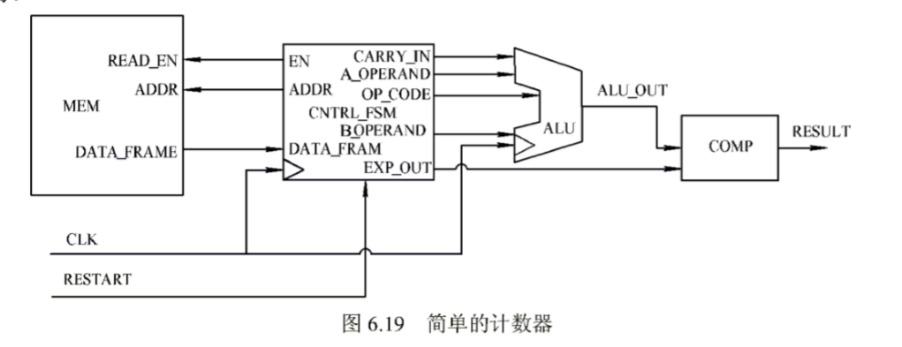
**学生姓名： 李云水**

**实验日期： 2020/5/23**

**实验题目： 比较器实验**

**实验环境：联想小新潮7000 i7 8G ，win10，ISE Design Suite 14.7**

**一、实验介绍**

 **本次实验将完成图6.19中比较器COMP的两种描述方式下的独立描述，一种为行为描述方式，应用Assert语句和条件信号赋值语句；另一种为RTL描述方式，用if/else语句。**

**二、实验目标**

**1.学习使用Assert语句**

**2.学习使用VHDL语言中的基本条件描述结构if/else；**

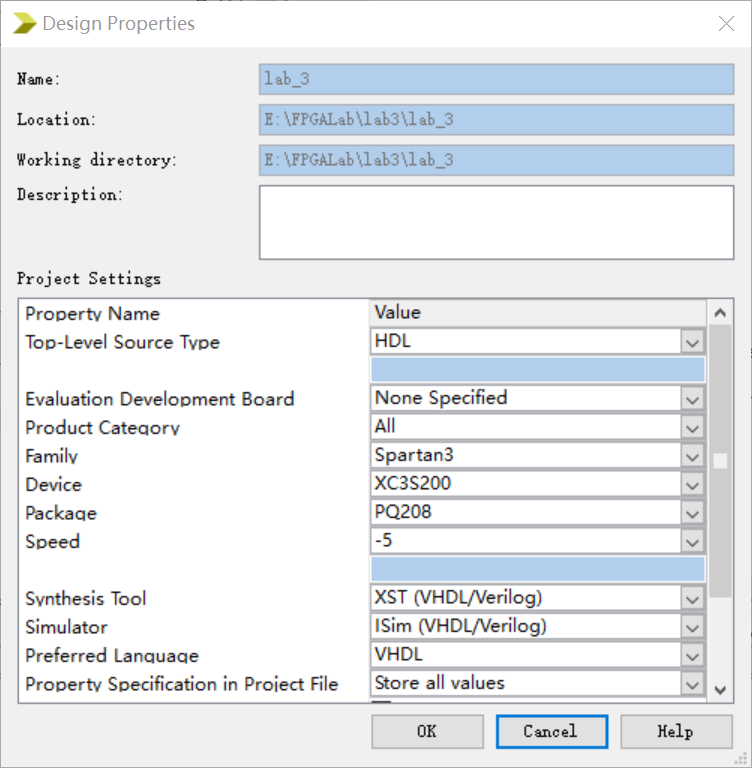
**3.用行为描述方式和RTL描述方式分别给定的模块进行描述；**

**4.验证设计文件的正确性。**

**三、实验过程**

**1.创建一个新的工程**

**配置属性**

****

**2.创建COMP的行为描述**

1）创建comp的行为描述



2）完成模块的结构体描述

  if(EXPECTED=ALU\_OUT) then

          RESULT <= '1';

        else

          RESULT <= '0';

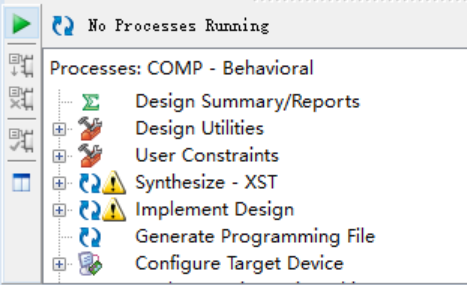
        end if;

        assert(EXPECTED=ALU\_OUT)

          report "Waring simulation mismatch has occurred"

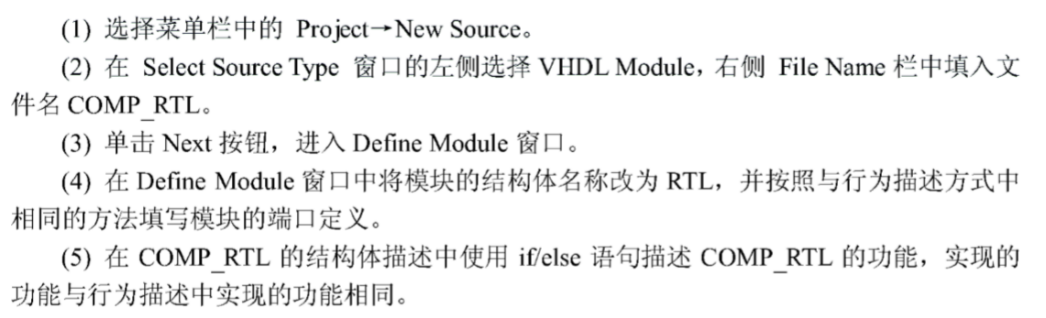
          severity warning;

3）语法检查

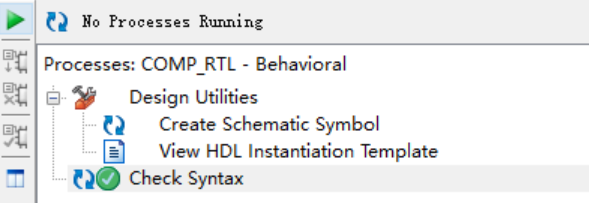


出现了一些版本的警告。

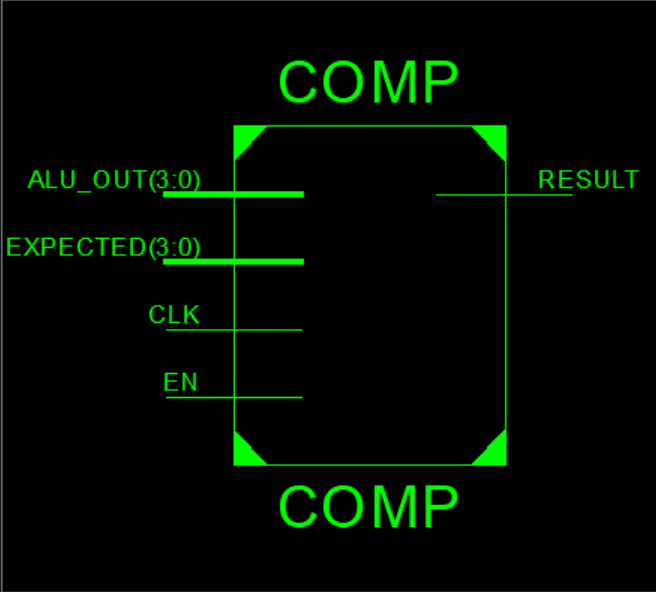
**3.创建COMP的RTL描述**



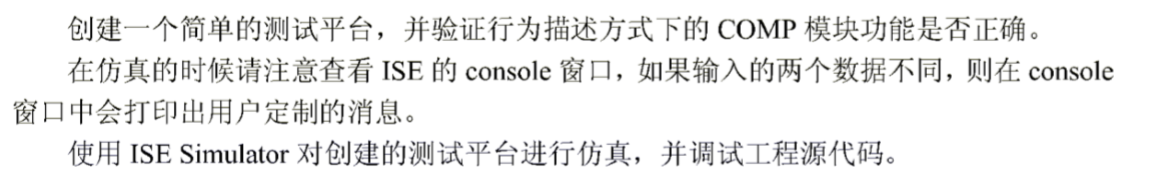
**检查语法是否正确：**

****

**RTL图：**

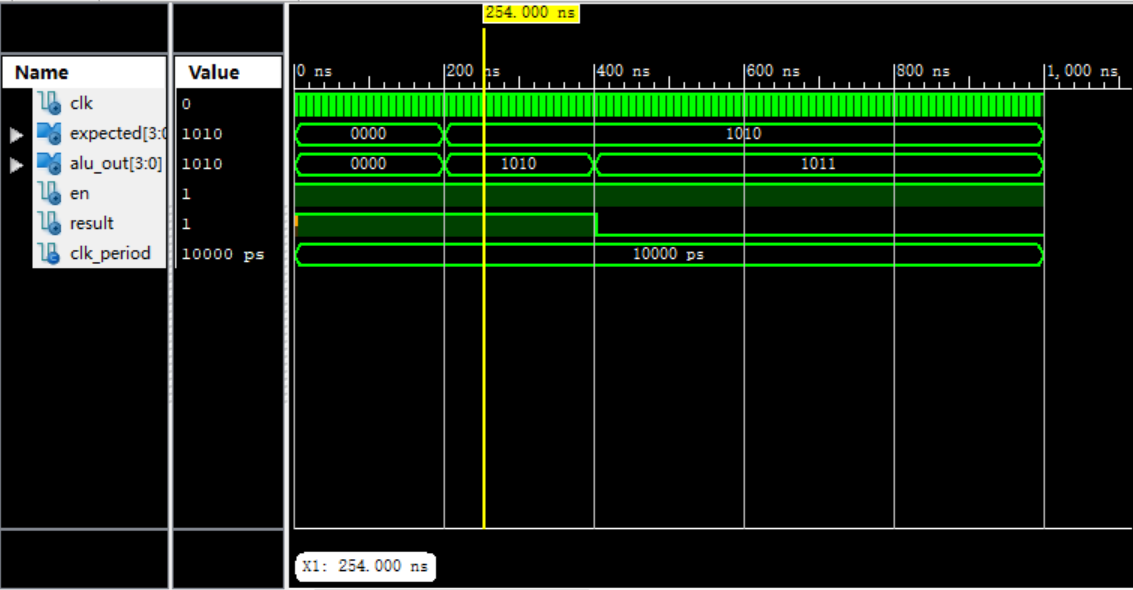
****

**4.创建测试平台文件**

****

**a.将模式设置为behavioral打开仿真工具**

**仿真波形如下：**

****

**四、实验总结**

在本实验中，我们首先创建了COMP模块的两种描述方式下的代码，第一种为行为级描述方式，主要用于仿真，使用的语句为assert语句和report语句，第二种为RTL描述，主要用于综合生成最总的电路，使用的描述语句为if/else语句。

在实验中，用户需要使用VHDL中的generate语句在两种结构体描述中做出选择。

**五、实验代码**

**COMP.VHD：**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity COMP is

--端口定义

port(

     CLK: in std\_logic;  --时钟输入

     EXPECTED: in std\_logic\_vector(3 downto 0);  --待比较数1

     ALU\_OUT:  in std\_logic\_vector(3 downto 0);  --待比较数2

     EN:       in std\_logic;                     --使能信号

     RESULT : out std\_logic                      --输出结果

     );

end COMP;

architecture Behavioral of COMP is

begin

  process(CLK)

  begin

  if rising\_edge(CLK) then   --时钟上升沿

    if(EN='1') then            --使能

       if(EXPECTED=ALU\_OUT) then  --判断是否相等

          RESULT <= '1';          --相等输出1

        else

          RESULT <= '0';           --反之

        end if;

        assert(EXPECTED=ALU\_OUT)    --输入信号不等在控制台打印自定义信息

          report "Waring simulation mismatch has occurred"

          severity warning;

    end if;

  end if;

end process;

end Behavioral;

**COMP\_RTL：**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity COMP\_RTL is

port(                               --端口定义

     CLK: in std\_logic;              --时钟

     EXPECTED: in std\_logic\_vector(3 downto 0);  --两个待比较的数

     ALU\_OUT:  in std\_logic\_vector(3 downto 0);  --两个待比较的数

     EN:       in std\_logic;                     --使能信号

     RESULT : out std\_logic                      --输出比较结果

     );

end COMP\_RTL;

architecture Behavioral of COMP\_RTL is

begin

  process(CLK)

  begin

  if rising\_edge(CLK) then       --判断是否是时钟上升沿

    if(EN='1') then              --判断使能信号是否为1

       if(EXPECTED=ALU\_OUT) then   --判断两个输入信号是否相等

          RESULT <= '1';           --相等输出赋值为1

        else

          RESULT <= '0';           --不等输出赋值为0

        end if;

    end if;

  end if;

end process;

end Behavioral;

**COMP\_TB:**

Library IEEE;

USE ieee.std\_logic\_1164.ALL;

--定义实体

ENTITY COMP\_TB IS

END COMP\_TB;

ARCHITECTURE behavior OF COMP\_TB IS

    -- Component Declaration for the Unit Under Test (UUT)

    --映射组件

    COMPONENT COMP

    PORT(

         CLK : IN  std\_logic;

         EXPECTED : IN  std\_logic\_vector(3 downto 0);

         ALU\_OUT : IN  std\_logic\_vector(3 downto 0);

         EN : IN  std\_logic;

         RESULT : OUT  std\_logic

        );

    END COMPONENT;

   --Inputs

   signal CLK : std\_logic := '0';

   signal EXPECTED : std\_logic\_vector(3 downto 0) := (others => '0');

   signal ALU\_OUT : std\_logic\_vector(3 downto 0) := (others => '0');

   signal EN : std\_logic := '0';

    --Outputs

   signal RESULT : std\_logic;

   -- Clock period definitions

   constant CLK\_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test (UUT)

    --映射端口

   uut: COMP PORT MAP (

          CLK => CLK,

          EXPECTED => EXPECTED,

          ALU\_OUT => ALU\_OUT,

          EN => EN,

          RESULT => RESULT

        );

   -- Clock process definitions

   CLK\_process :process

   begin

        CLK <= '0';

        wait for CLK\_period/2;

        CLK <= '1';

        wait for CLK\_period/2;

   end process;

   -- Stimulus process

    TB: process

   begin

      EN <='1';

      -- hold reset state for 100 ns.

      wait for 100 ns;

      wait for CLK\_period\*10;

        EXPECTED <=X"A";   --16进制A

        ALU\_OUT <=X"A";

        wait for 200 ns;

        ALU\_OUT <=X"B";

      -- insert stimulus here

      wait;

   end process;

END;

**流程图：**

**开始**

N

Y

E=A?

Result=0

Result=1

N

N

Y

E=A？

Y

Y

N

**结束**

Clk上升？

**输入**

**结束**

Report

Error

输出Result

EN=1？