



西安电子科技大学
XIDIAN UNIVERSITY

基于 FPGA 的数字系统设计

1 位比较器实验报告

实验名称：1 位比较器数据流描述和行为/结构描述
实现

任课教师：沈沛意老师

学号姓名：

提交日期：

第一部分：数据流描述实现

一、 实验目的

验证使用数据流描述方式实现 1 位比较器的正确性, 通过功能仿真测试比较器在不同输入组合下的输出结果。

二、 实验要求

- 采用数据流描述方式实现 1 位比较器
- 比较器输出在输入位相等时为 1
- 提供功能仿真结果

三、 实验代码注释

comparator_dataflow.vhd:

```
library IEEE;
```

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

```
entity comparator is
```

```
    Port ( A : in  STD_LOGIC;
```

```
          B : in  STD_LOGIC;
```

```
          AB_GT : out STD_LOGIC;  -- A 大于 B 时为 1
```

```
          AB_EQ : out STD_LOGIC;  -- A 等于 B 时为 1
```

```
          AB_LT : out STD_LOGIC); -- A 小于 B 时为 1
```

```
end comparator;
```

```
architecture Behavioral of comparator is
```

begin

AB_GT <= A and (not B); -- A=1 且 B=0 时输出 1

AB_EQ <= A xnor B; -- A 和 B 相等时输出 1

AB_LT <= (not A) and B; -- A=0 且 B=1 时输出 1

end Behavioral;

四、 实验仿真结果

——功能仿真

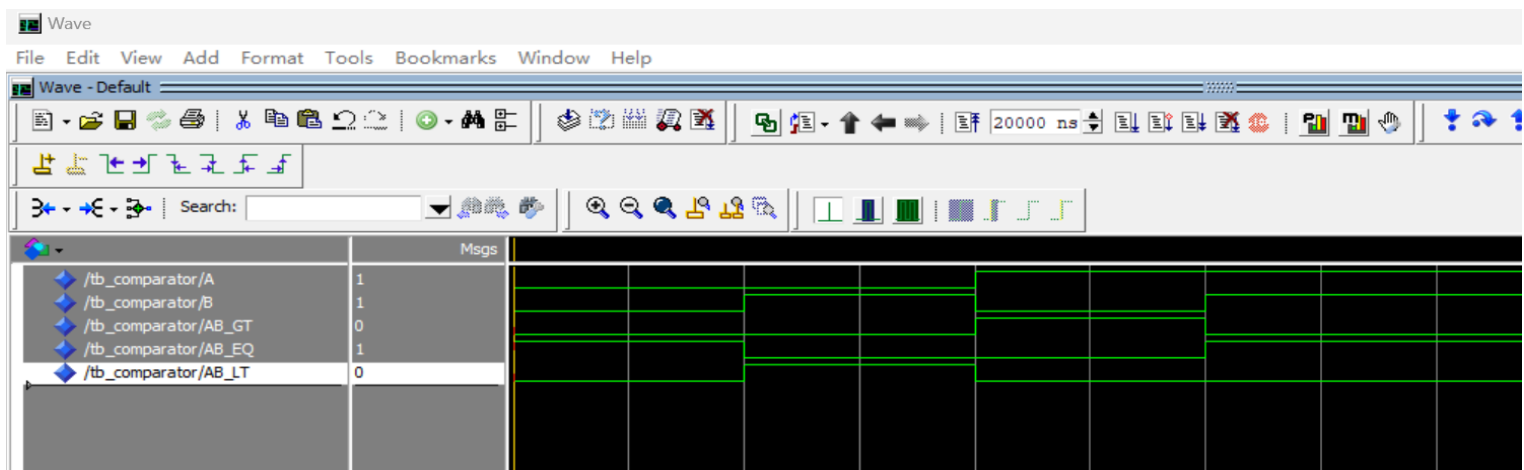
测试文件验证了四种输入组合：

1. A=0, B=0 → AB_EQ=1 (通过)

2. A=0, B=1 → AB_LT=1 (通过)

3. A=1, B=0 → AB_GT=1 (通过)

4. A=1, B=1 → AB_EQ=1 (通过)



所有测试用例均通过，验证了数据流描述的正确性。

五、 实验总结

数据流描述方式简洁明了，直接使用逻辑运算符实现比较功能。XNOR 运

算完美实现了相等比较，而 AND 运算实现了大小比较。这种实现方式在综合后会生成简单的组合逻辑电路。

第二部分：行为/结构描述实现

一、 实验目的

验证使用行为描述和结构描述方式实现 1 位比较器的正确性，分析其综合后的 RTL 原理图和布线后仿真结果。

二、 实验要求

- 采用行为描述和结构描述两种方式实现 1 位比较器
- 比较器输出在输入位相等时为 1
- 提供布线后仿真结果和 RTL 原理图

三、 实验代码注释

comparator_behavioral.vhd:

-- 行为描述（顺序语句）

architecture Behavioral of comparator is

begin

 process(A, B)

 begin

 -- 默认输出为 0

 AB_GT <= '0';

 AB_EQ <= '0';

 AB_LT <= '0';

```

-- 顺序判断逻辑

if A > B then

    AB_GT <= '1';

elsif A = B then

    AB_EQ <= '1';

else

    AB_LT <= '1';

end if;

end process;

end Behavioral;


-- 结构描述（并发语句）

architecture Structural of comparator is

begin

    -- 三个并发条件赋值语句

    AB_EQ <= '1' when (A = B) else '0';  -- 相等比较

    AB_GT <= '1' when (A > B) else '0';  -- 大于比较

    AB_LT <= '1' when (A < B) else '0';  -- 小于比较

end Structural;


tb_comparator.vhd:

-- 测试平台实体声明

```

```
ENTITY tb_comparator IS
```

```
END tb_comparator;
```

```
ARCHITECTURE behavior OF tb_comparator IS
```

```
-- 被测组件声明
```

```
COMPONENT comparator PORT(...);
```

```
END COMPONENT;
```

```
-- 测试信号定义
```

```
signal A, B : std_logic := '0';
```

```
signal AB_GT, AB_EQ, AB_LT : std_logic;
```

```
BEGIN
```

```
-- 实例化被测单元
```

```
uut: comparator PORT MAP(...);
```

```
-- 测试过程
```

```
stim_proc: process
```

```
begin
```

```
-- 测试用例 1: A=0,B=0 → 应输出 AB_EQ=1
```

```
A <= '0'; B <= '0';
```

```
wait for 20 ns;
```

```
assert (AB_EQ='1') report "Case 1 Failed";
```

```
-- 其他测试用例...
```

```
wait; -- 结束仿真
```

```
end process;
```

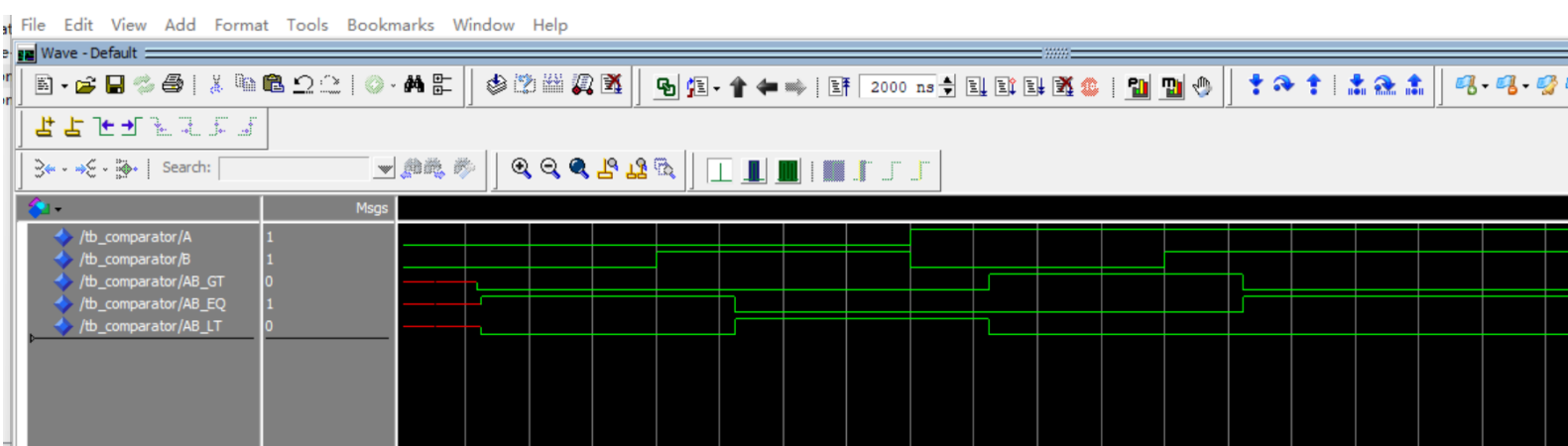
```
END;
```

四、 实验仿真结果

——后仿与 RTL

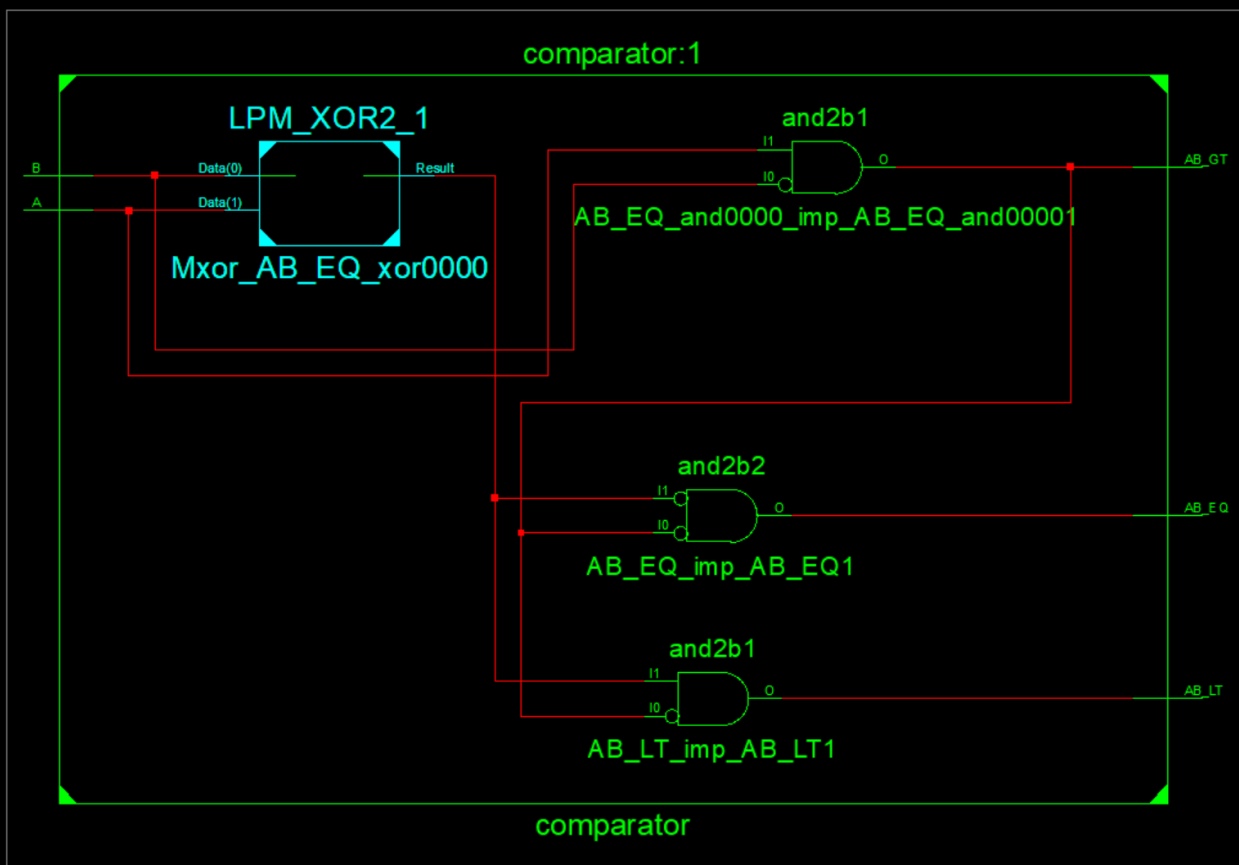
测试文件验证了四种输入组合：

1. A=0, B=0 → AB_EQ=1 (通过)
2. A=0, B=1 → AB_LT=1 (通过)
3. A=1, B=0 → AB_GT=1 (通过)
4. A=1, B=1 → AB_EQ=1 (通过)



行为/结构描述的所有测试用例均通过。

行为/结构描述的 RTL 原理图：



- 两种方式最终功能等效

五、实验总结

行为描述使用顺序语句更适合复杂的状态机实现，而结构描述使用并发语句更直观地反映了硬件并行特性。布线后仿真显示两种实现方式功能相同，但资源占用和时序特性有所不同。而结构描述通常会产生更优化的硬件实现。