

中国海洋大学计算机科学与技术系

实验报告

姓名：岳宇轩

年级：2019

专业：19 慧与

科目：计算机组成原理

题目：加法器

实验时间：2021 年 4 月 12 日

实验教师：张巍

一、实验结果及截图分析：

(※代码挖空的部分必须截图或复制)

1.添加 adder.v 文件，并补全代码

```
C:/Users/yyx/Desktop/adder.v

1  `timescale 1ns / 1ps
2  //*****
3  // > 文件名: adder.v
4  // > 描述 : 加法器, 直接使用"+", 会自动调用库里的加法器
5  // > 作者 : LOONGSON
6  // > 日期 : 2016-04-14
7  //*****
8  module adder(
9      input [31:0] operand1,
10     input [31:0] operand2,
11     input      cin,
12     output [31:0] result,
13     output      cout
14 );
15     //请于此处添加代码
16     //有 2 个 32 位数的输入和 1 个进位输入, 产生 1 个 32 位的加法 和结果和 1 个向高位的进位。
17     assign {cout,result} = operand1 + operand2 + cin;
18 endmodule
19
```

添加代码为：

```
assign {cout,result} = operand1 + operand2 + cin;
```

其作用为：

有 2 个 32 位数的输入和 1 个进位输入，产生 1 个 32 位的加法结果和 1 个向高位的进位。

（无论右边表达式操作数何时发生变化，右边表达式都会重新计算，并且在指定的延迟后给左边表达式赋值）

2.添加 testbench.v 文件

```
`timescale 1ns / 1ps    //仿真单位时间为 1ns，精度为 1ps
module testbench;
```

```
    // 输入
```

```
    reg [31:0] operand1;
```

```
    reg [31:0] operand2;
```

```
    reg cin;
```

```
    // 输出
```

```
    wire [31:0] result;
```

```
    wire cout;
```

```
    // Instantiate the Unit Under Test (UUT)
```

```
    adder uut (
```

```
        .operand1(operand1),
```

```
        .operand2(operand2),
```

```
        .cin(cin),
```

```
        .result(result),
```

```
        .cout(cout)
```

```
    );
```

```
    initial begin
```

```
        // 初始化输入
```

```
        operand1 = 0;
```

```
        operand2 = 0;
```

```
        cin = 0;
```

```
        // Wait 100 ns for global reset to finish
```

```
        #100;
```

```
        // Add stimulus here
```

```
    end
```

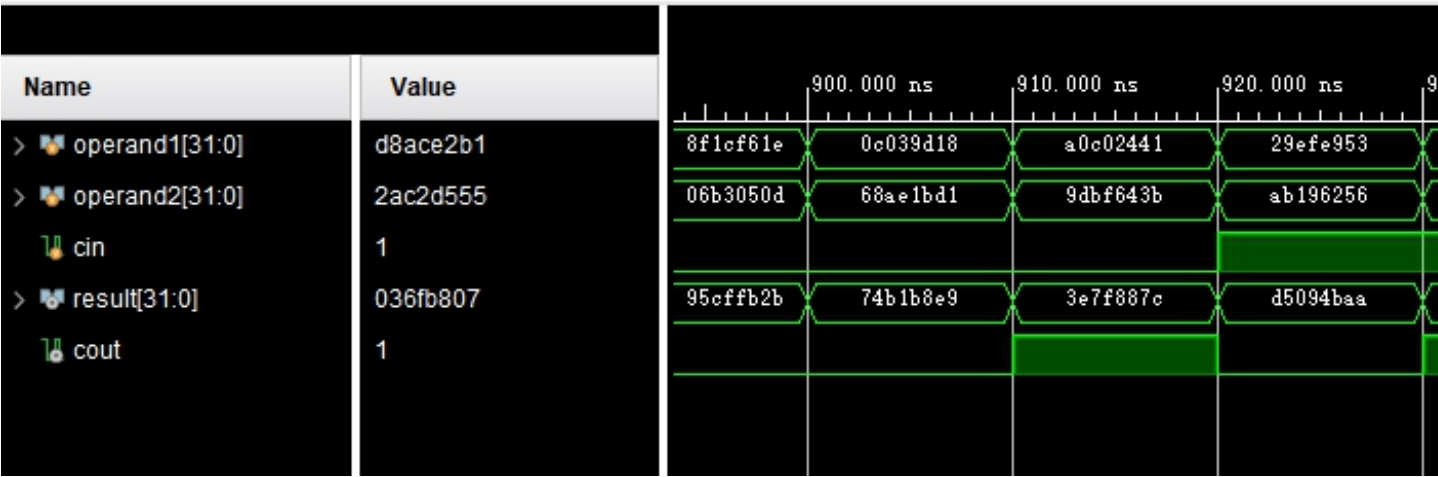
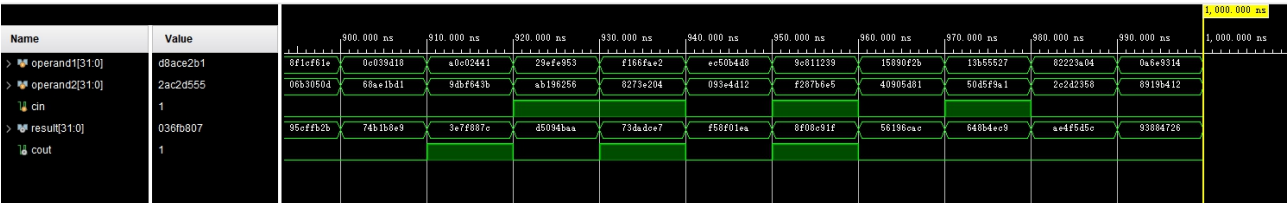
```
    always #10 operand1 = $random; // $random 为系统任务，产生一个随机的 32 位数
```

```
    always #10 operand2 = $random; // #10 表示等待 10 个单位时间(10ns)，即每过 10ns，赋值一个随机的 32 位数
```

```
    always #10 cin = {$random} % 2; //加了拼接符，{$random}产生一个非负数，除 2 取余得到 0 或 1
```

endmodule

3. 完成仿真，生成仿真波形图像



分析该图像

两个加数分别为 d8ace2b1(operand1)和 2ac2d555(operand2)。低位有 1 位进位(cin)。

$$\begin{aligned} &D8ACE2B1 + 2AC2D555 + 1 = \\ &\quad \quad \quad \mathbf{1\ 036F\ B807} \end{aligned}$$

实验预期的计算结果为

32 位结果为 036fb807 (result))。实验结果与预期结果一致，并且有 1 位的进位 (cout) 。

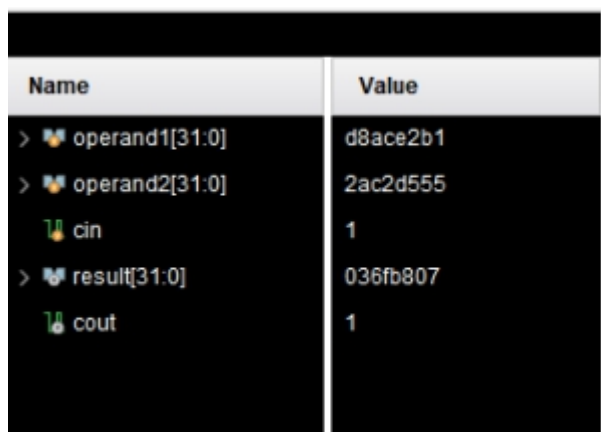
二、实验总结

首先，这次实验主要学习了 verilog 的安装。安装过程并没有遇到什么问题，按照实验指导一步一步进行即可完成安装。

然后，我学习了一些 Verilog 的基本语法，但还是掌握的不够全面，今后还需要继续学习。

在编写加法器的过程中，我学会了如何在 Verilog 中添加文件，以及将某个文件设置为 top。添加代码的过程中，我对于加法器的实现过程又有了更深刻的印象：共有三个输入，两个 operand 为加数，一个 cin 为低位向本位的进位；共有两个输出，一个 result 为加和后的 32 位结果，另一个 cout 为本位向高位的进位。根据这个理解，可以在 adder.v 中添加如下代码：`assign {cout,result} = operand1 + operand2 + cin;`

最后，我学会了如何生成仿真波形图像，如何进行调试以及如何进



Name	Value
> operand1[31:0]	d8ace2b1
> operand2[31:0]	2ac2d555
cin	1
> result[31:0]	036fb807
cout	1

行观察。Name 列为名称，Value 为对应的值（输入或者输出）

此外，有关 Verilog 的语法我还需要更好的去掌握。