

杭州电子科技大学

# 计算机组成原理（甲） 实 验 报 告

学 院	网络空间安全学院
专 业	网络工程
班 级	19272401
学 号	19061440
学生姓名	F001
教师姓名	袁理峰
完成日期	2020.11.27
成 绩	

## 实验二 超前进位加法器设计实验 （实验名称）

### 一、 实验目的

1. 掌握运用 Verilog HDL 进行数据流描述与建模的技巧和方法，掌握模块调用与实例引用的方法
2. 掌握超前进位加法器的原理与设计方法

### 二、 实验原理

超前进位加法器的原理

减少了进位链的存在，对加法进位器进行了优化

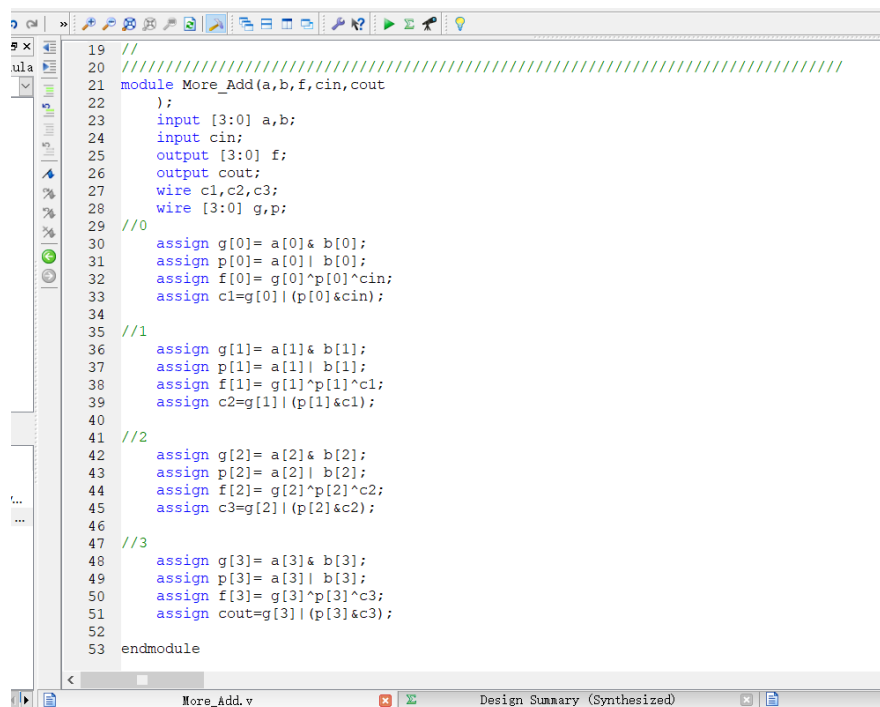
### 三、 实验环境

所用电脑的软硬件配置：4 教机房电脑、Windows 操作系统

实验所用的软件：ISE design suite

### 四、 主要操作步骤及实验结果记录

任务一：自 Xilinx ISE 中创建工程，编辑程序源代码，然后编译、综合

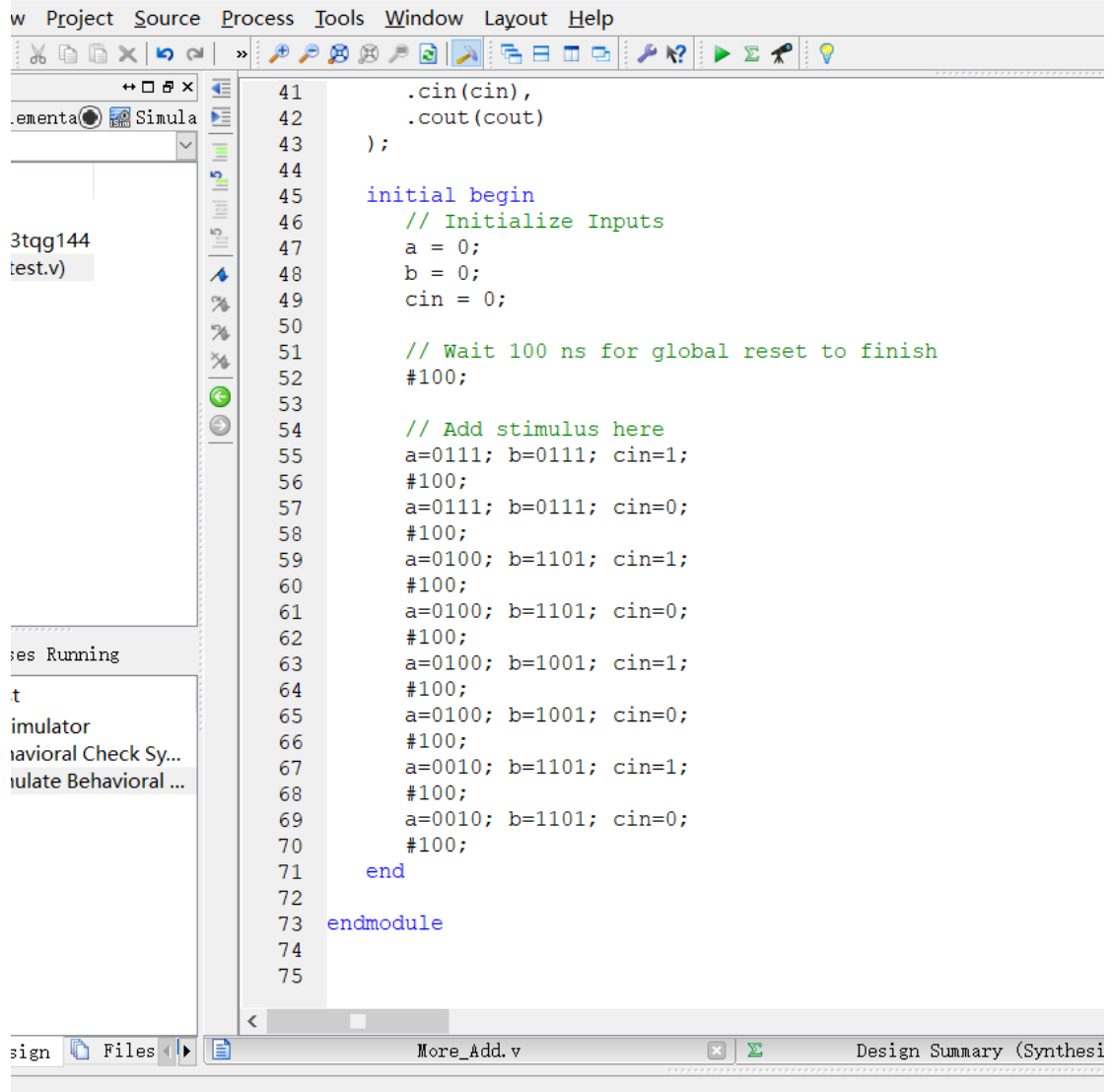


```
19 //
20 ///////////////////////////////////////////////////
21 module More_Add(a,b,f,cin,cout
22 );
23   input [3:0] a,b;
24   input cin;
25   output [3:0] f;
26   output cout;
27   wire c1,c2,c3;
28   wire [3:0] g,p;
29 //0
30   assign g[0]= a[0]& b[0];
31   assign p[0]= a[0]| b[0];
32   assign f[0]= g[0]^p[0]^cin;
33   assign c1=g[0]|(p[0]&cin);
34
35 //1
36   assign g[1]= a[1]& b[1];
37   assign p[1]= a[1]| b[1];
38   assign f[1]= g[1]^p[1]^c1;
39   assign c2=g[1]|(p[1]&c1);
40
41 //2
42   assign g[2]= a[2]& b[2];
43   assign p[2]= a[2]| b[2];
44   assign f[2]= g[2]^p[2]^c2;
45   assign c3=g[2]|(p[2]&c2);
46
47 //3
48   assign g[3]= a[3]& b[3];
49   assign p[3]= a[3]| b[3];
50   assign f[3]= g[3]^p[3]^c3;
51   assign cout=g[3]|(p[3]&c3);
52
53 endmodule
```

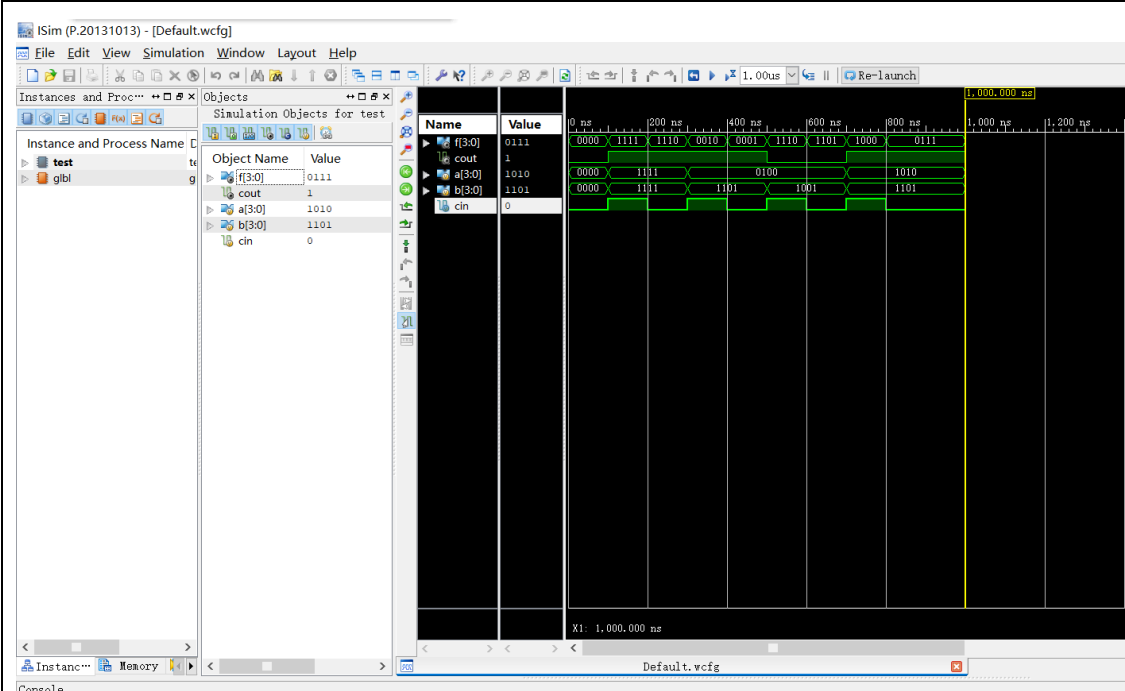
该段代码成功通过了语法检测

任务二：编写激励代码，观察仿真波形

rigator (P.20131013) - E:\Computer Orgnization\lab2\lab2.xise - [test.v]



```
41     .cin(cin),
42     .cout(cout)
43 );
44
45 initial begin
46     // Initialize Inputs
47     a = 0;
48     b = 0;
49     cin = 0;
50
51     // Wait 100 ns for global reset to finish
52     #100;
53
54     // Add stimulus here
55     a=0111; b=0111; cin=1;
56     #100;
57     a=0111; b=0111; cin=0;
58     #100;
59     a=0100; b=1101; cin=1;
60     #100;
61     a=0100; b=1101; cin=0;
62     #100;
63     a=0100; b=1001; cin=1;
64     #100;
65     a=0100; b=1001; cin=0;
66     #100;
67     a=0010; b=1101; cin=1;
68     #100;
69     a=0010; b=1101; cin=0;
70     #100;
71 end
72
73 endmodule
74
75
```



经过检验，激励代码与仿真波形一致，代码无误。

### 任务三：观察资源消耗情况与内部相关结构

?lab2.xise - [Design Summary (out of date)]

ayout Help

More_Add Project Status (11/27/2020 - 00:17:42)			
Project File:	lab2.xise	Parser Errors:	No Errors
Module Name:	More_Add	Implementation State:	Synthesized
Target Device:	xc6slx4-3tqg144	Errors:	No Errors
Product Version:	ISE 14.7	Warnings:	No Warnings
Design Goal:	Balanced	Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	Timing Constraints:	
Environment:	System Settings	Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	6	2400	
Number of fully used LUT-FF pairs	0	6	
Number of bonded IOBs	14	102	

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	周五 11月 27 00:17:41 2020	0	0	0
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports		
Report Name	Status	Generated
ISIM Simulator Log	Out of Date	周五 11月 27 00:25:43 2020

Date Generated: 11/27/2020 - 00:34:37

Design Summary (out of date) test.v

ccessfully

## 五、 实验分析总结及心得

（结合所学知识对实验过程中观察到的实验结果进行分析总结，以便加深对知识的理解，并总结通过实验学到的知识或技术）

此次实验过程中，由于有了实验一打基础，遇到的问题相对较少甚至没有，因此在此部分，我将重点谈谈对于此次实验的思考与更深层次的理解。

通过观察最终得到的电路图我发现，超前二进制加法器与二进制加法器尽管实现的是同样的功能，但是其内部组成部件大相径庭，我认为其原因是超前二进制加法器去除了进位链的概念，用更快的速度完成了相同的内容。可以说，原理不同，结构也就自然不同了。这启示我们在面对同样的问题时，眼界不能单纯局限于对于目的的实现，也要注重原理层面上是否有着更好的方式进行优化。

通过查阅资料，我了解到更多位数的加法器，例如 16 位，各位的仅为逻辑表达式将会十分可观，甚至难以通过自然语言加以描述，因此，更多得通过二级逻辑进行实现。例如 4 位 1 组，组内采用超前进位，组件也通过本组得进位产生函数和进位传递函数实现组间超前进位。这启示我们在面对类似的困难时，可以从自身已有条件出发，逐步逼近目标，从而实现局部乃至全局的最优解。这种全新的思维方式不仅仅适用于计算机上，也适用于我们其他学科乃至生活的方方面面。