**实验报告**

2021年05月11日 成绩：---------

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | 刘爱兵 | 学号 | 19114801 | 班级 | 19计科2班 |
| 专业 | 计算机科学与技术 | 课程名称 | 计算机组成原理 实验 | 任课教师 | 吴云志 |
| 实验序号 | 02 | 实验名称 | 全加器设计、多功能ALU设计 | | |
| 实验时间 | 2021.05.11 | 机位号 | 18 | 实验设备号 | 21 |

1. **实验目的与要求**
2. 实验目的

（1）学习Vivado工具软件的使用及仿真方法；

（2）学习FPGA程序下载方法；

（3）熟悉HDU-XL-01教学开发板；

（4）掌握运用Verilog HDL语言进行结构描述与建模的技巧和方法；

（5）掌握二进制全加器的原理与设计方法；

1. 实验要求

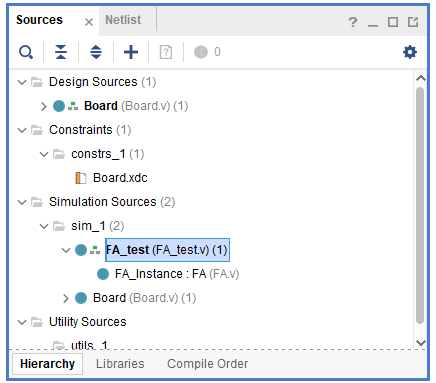
使用结构描述方式，编程实现1位二进制全加器模块

课前任务：编程、仿真、验证，确保逻辑正确性；

1. **实验设计与程序代码**

1、模块设计说明

（描述这个实验的设计方案，分几个模块，各模块的功能，各模块之间的连接关系，可附图）

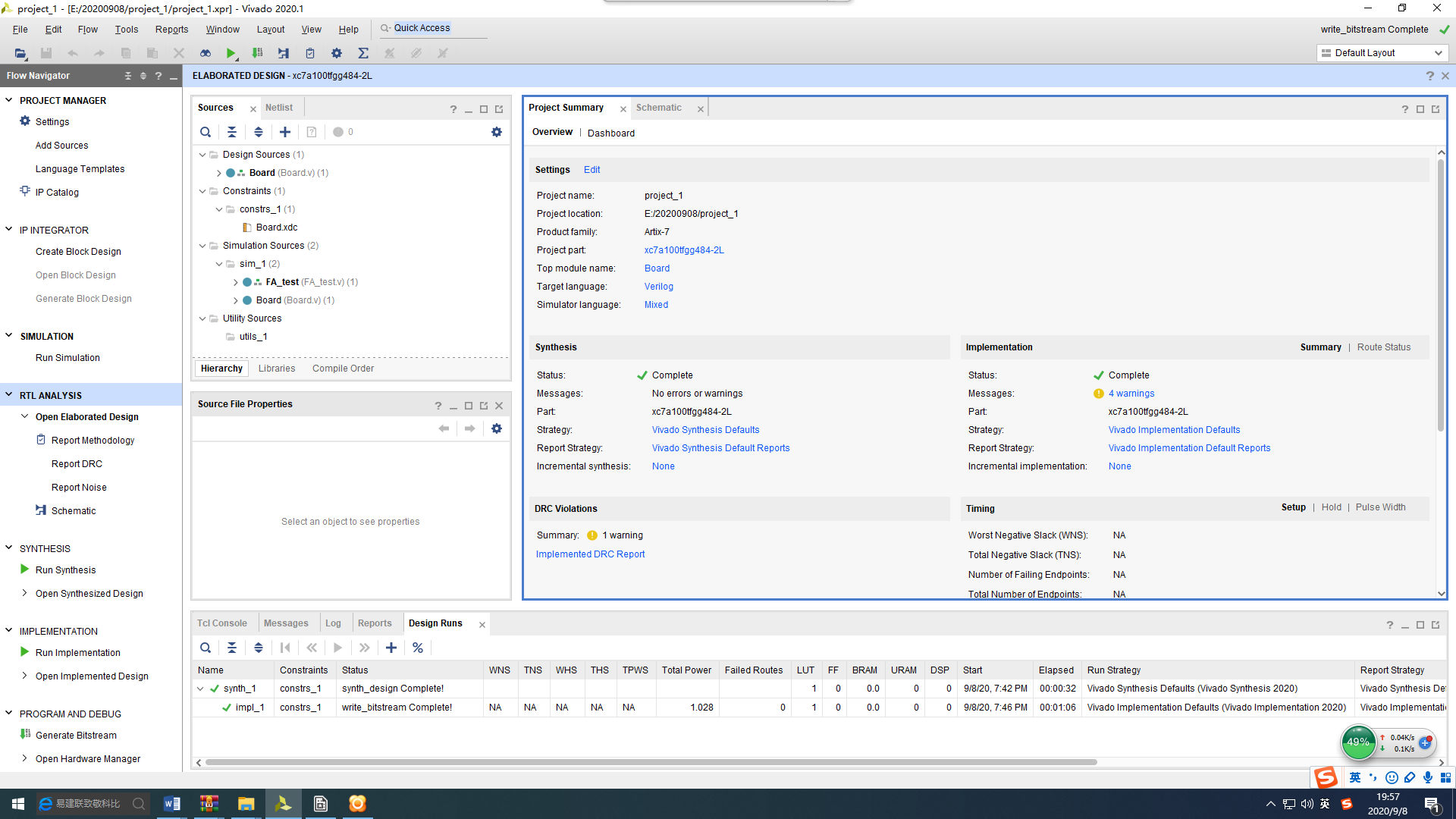


2、实验程序源代码及注释等

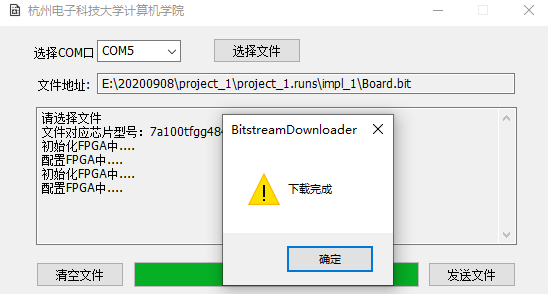
（实验各个模块的代码，包含功能注释）

|  |
| --- |
| (1) **Board.v**  `timescale 1ns / 1ps  //一位加法产生和位以及进位位  // LED的低两位分别为进位位和加法和位，拨号开关的后三位为3个加数  module Board(sw, led);  input [1:32] sw;  output [1:32] led;    FA FA\_Instance(led[32],led[31],sw[30],sw[31],sw[32]);  endmodule // Board  （2）**FA\_test.v**  `timescale 1ns / 1ps  //一位加法产生和位以及进位位  // LED的低两位分别为进位位和加法和位，拨号开关的后三位为3个加数  module FA\_test;  wire f,cc;  reg a,b,c;      FA FA\_Instance(f,cc,a,b,c);  initial begin  a = 1'b0; b = 1'b0; c = 1'b0; #50;  a = 1'b0; b = 1'b0; c = 1'b1; #50;  a = 1'b0; b = 1'b1; c = 1'b0; #50;  a = 1'b0; b = 1'b1; c = 1'b1; #50;  a = 1'b1; b = 1'b0; c = 1'b0; #50;  a = 1'b1; b = 1'b0; c = 1'b1; #50;  a = 1'b1; b = 1'b1; c = 1'b0; #50;  a = 1'b1; b = 1'b1; c = 1'b1; #50;  end  endmodule // FA\_test |

在Xilinx ISE/Vavido中创建工程，编源码，然后编译、综合



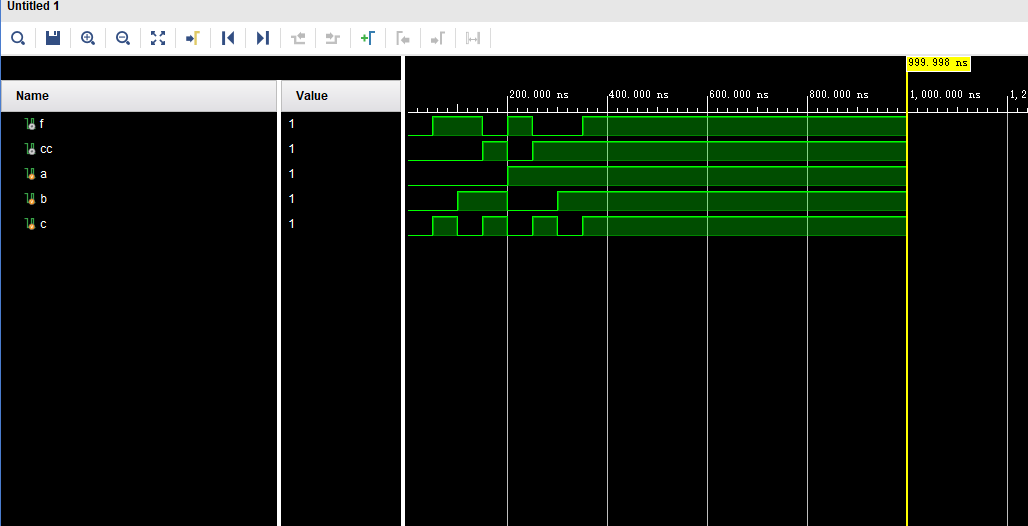
1. 使用本地板进行调试。观察拨动3位数据开关作为Ai、Bi和Ci，观察相应的LED灯，验证是否符合设计的1位二进制全加器的逻辑功能。



1. **实验仿真**
2. 仿真代码（仿真源代码）

|  |
| --- |
| module FA(  output Fi,  output Cout,  input Ai,  input Bi,  input Cin  );    //Method1  /\*  wire T1,T2,T3,T4;  xor xor1(Fi,Ai,Bi,Cin);  and and1(T2,Ai,Bi);  or or1(T3,Ai,Bi);  and and2(T4,T3,Cin);  or or2(Cout,T2,T4);  \*/    //Method2  assign {Cout,Fi}=Ai+Bi+Cin;  endmodule |

1. 仿真波形（运行仿真时波形截图）



1. 仿真结果分析（对仿真波形进行分析）

经过试验，所设计的全加器的输出的结果正确符合逻辑。

每隔50.ns Ci变动一次，即ABC每隔50ns加二进制1。

Ai Bi Ci Fi Ci+1

0 0 0 0 0

0 0 1 1 0

0 1 0 1 0

0 1 1 0 1

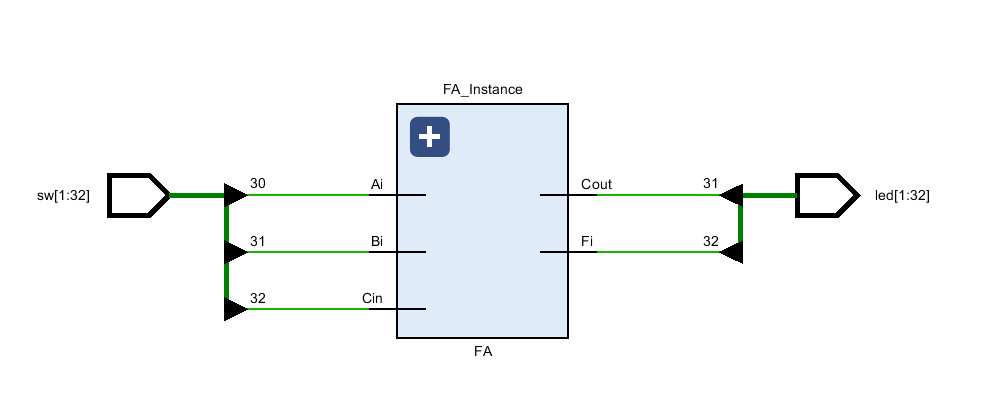
1 0 0 1 0

1 0 1 0 1

1 1 0 0 1

1 1 1 1 1

1. **电路图**



1. **引脚配置**

|  |  |  |
| --- | --- | --- |
|  | **信号** | **配置设备管脚** |
| **输入信号** | **Ai** | **逻辑开关** |
| **Bi** | **逻辑开关** |
| **Ci** | **逻辑开关** |
| **输出信号** | **Fi** | **LED** |
| **Ci+1** | **LED** |

1. **思考与探索**
2. 实验结果记录：
3. 实验结论

设计的全加器符合逻辑。其中Ai、Bi对应数据开关，拨动后观察相应的LED灯，发现Ci的结果符合逻辑。

1. 问题与解决方案：

代码理解有困难，经过老师讲解后有了更深刻的理解。

4、思考与探索

（1）观察经过逻辑综合后产生的电路原理图，和你使用门级描述时预期的电路一致吗？如果不一致，分析可能的原因。

答：一致。

（2）尝试使用数据流描述方法实现该实验，并观察它所综合产生的电路，和结构建模所产生的电路有何不同？

答：数据流描述方法把运算生成一个逻辑元件，结构建模生成的是复杂的逻辑结构。

（3）调用本次实验所设计的FA模块，尝试编程实现4位串行进位加法器。

|  |
| --- |
| Module FourBitFA(X,Y,Cin,Sum,Cour);  Parameter SIZE = 4;  Input[SIZE-1:0] X,Y;  Output[SIZE-1:0] Sum;  Input Cin;  Output Court;  Wire [SIZE-1:1]C;  Full\_Adder FA0(Sum[0],C(1),X(0),Y(0),Cin),  FA1(Sum[1],C(2),X(1),Y(1),C[1]),  FA2(  .X(X[2]),  .Y(Y[2]),  .Cin(C[2]),  .Sum(Sum[2]),  .Court(C[3])),  FA3(  .Sum(Sum[3]),  .Court(Court),  .X(X[3]),  .Y(Y[3]),  .Cin(C[3]))  endmodule |

（4）谈谈你在实验中碰到了哪些问题？又是如何解决的？

代码理解有困难，经过老师讲解后有了更深刻的理解。