

**数 字 电 路**

**实验报告**

**班级：**教221

**组号：----**

**姓名：**唐嘉良

**学号：**2020K8009907032

**实验名称：**熟悉Verilog环境

**2021年10月16日**

**一、实验目的**

1.熟悉vivado设计流程。

2.掌握利用vivado创建设计的方法。

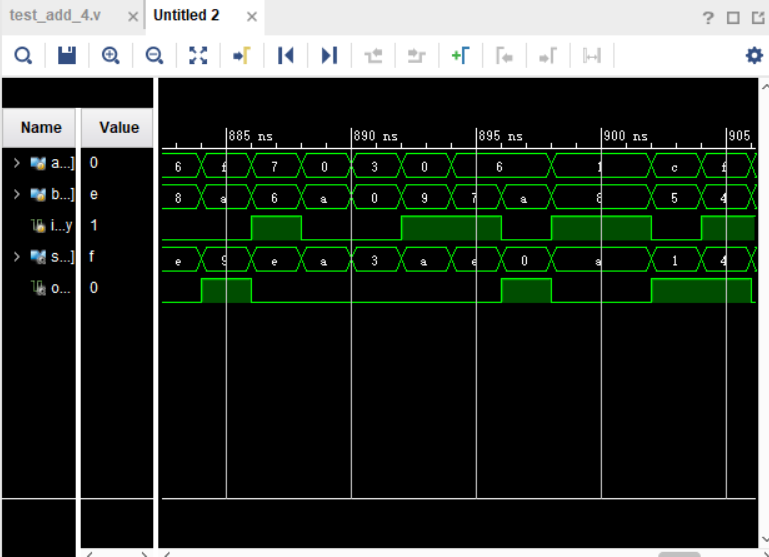
3.掌握编写testbench的方法，以及行为仿真方法。

1. **实验环境**

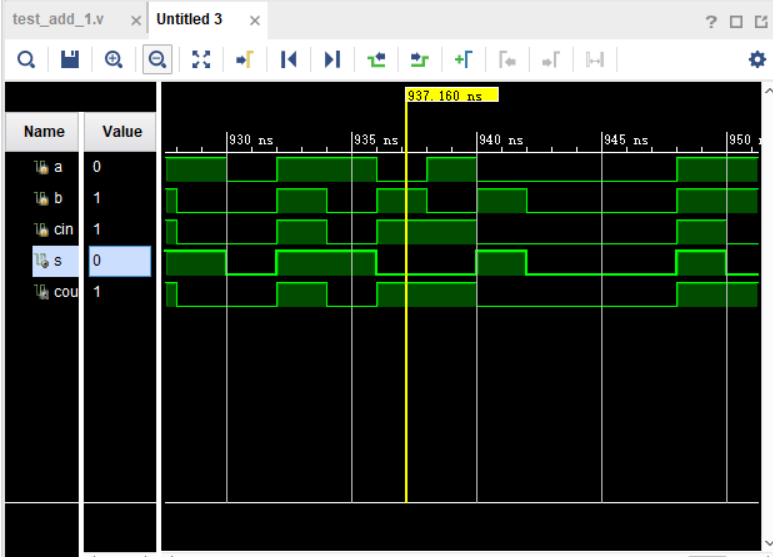
本次实验我采用的是vivado 2017.4版本。

1. **调试过程**

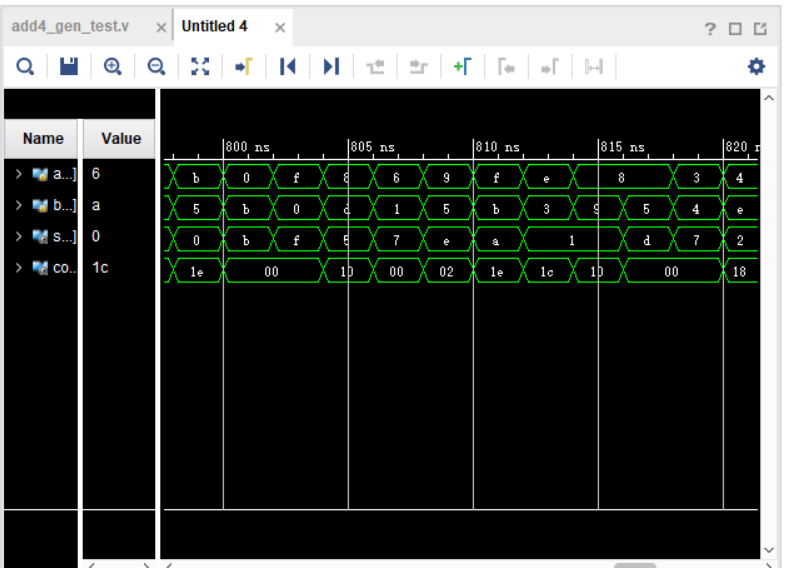
**4bit全加器：**



**1bit全加器：**

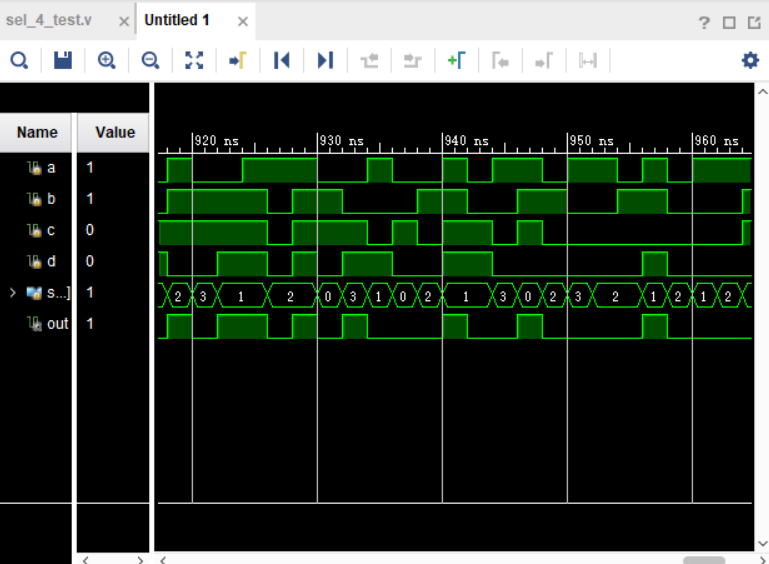


**4bit全加器用generate实现：**



1. **实验总结**

在此次实验中，我首先熟悉了vivado平台的操作流程，了解如何创建激励文件，并学会了初始化、随机赋值等激励操作。同时，通过使用常规方法及generate代码构建4bit全加器、行为建模和结构建模构建1bit全加器，我对Verilog语言的掌握程度大大提升，能够更加熟练自如地构建并调试某些常见组合电路模块。例如，在实验课后，我自己尝试构建了4选1数据选择器并得到了正确的波形图如下。



1. **源代码**

**4bit全加器：**

module add\_4(

input [3:0] in\_0,

input [3:0] in\_1,

input cin,

output [3:0] out,

output cout

);

assign {cout,out}=in\_0 + in\_1 + cin;

endmodule

**1bit全加器：**

（行为建模）

module add\_1(

input a,

input b,

input cin,

output s,

output cout

);

assign s = a ^ b ^ cin;

assign cout = (a&b) | (cin&(a|b));

Endmodule

（结构建模）

module add\_1\_structure(

input a,

input b,

input cin,

output s,

output cout

);

wire x,c0,c1,c2;

xor(x,a,b);

xor(s,x,cin);

and(c0,a,b);

and(c1,a,cin);

and(c2,b,cin);

or(cout,c0,c1,c2);

Endmodule

**4bit全加器用generate实现：**

module add4\_gen

#(parameter N = 4)

(

input [3:0] a,

input [3:0] b,

output [3:0] s,

output [4:0] cout

);

assign cout[0] = 0;

genvar i;

generate

for(i=0;i<N;i=i+1) begin

add\_1 u0(

.a(a[i]),

.b(b[i]),

.cin(cout[i]),

.s(s[i]),

.cout(cout[i+1]));

end

endgenerate

endmodule

**课后自主尝试：4选1数据选择器**

module sel\_4(

input a,

input b,

input c,

input d,

input [1:0] sel,

output reg out

);

always @(\*)

case(sel)

2'h0: out = a;

2'h1: out = b;

2'h2: out = c;

2'h3: out = d;

endcase

endmodule

波形图见“实验总结”。