



中国科学院大学
University of Chinese Academy of Sciences

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

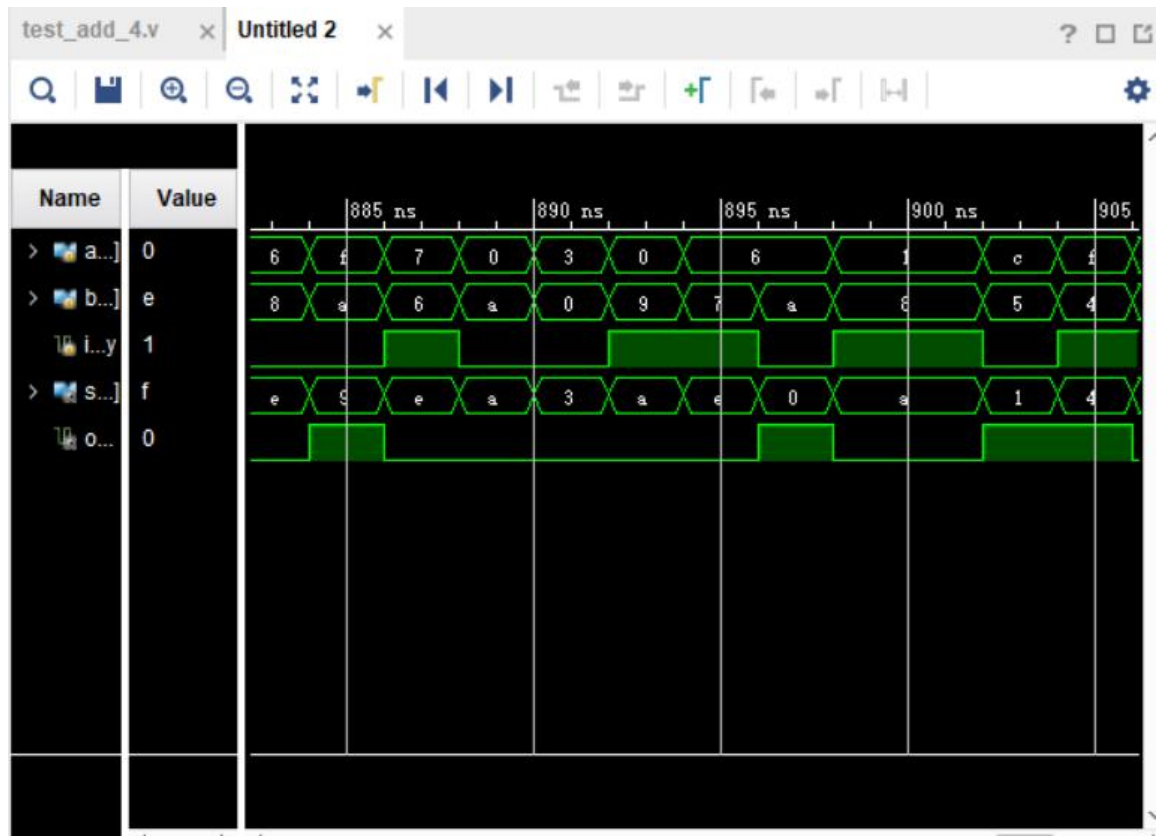
1. 熟悉 vivado 设计流程。
2. 掌握利用 vivado 创建设计的方法。
3. 掌握编写 testbench 的方法，以及行为仿真方法。

二、实验环境

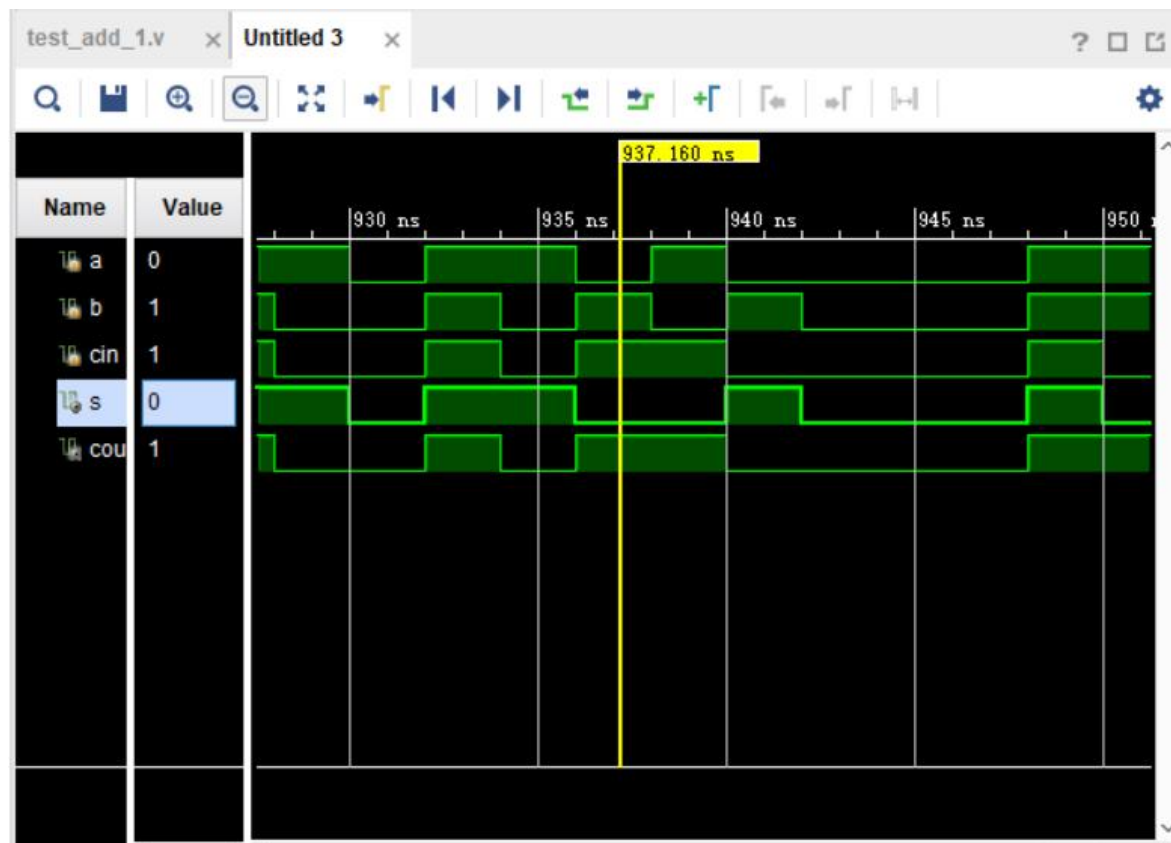
本次实验我采用的是 vivado 2017.4 版本。

三、调试过程

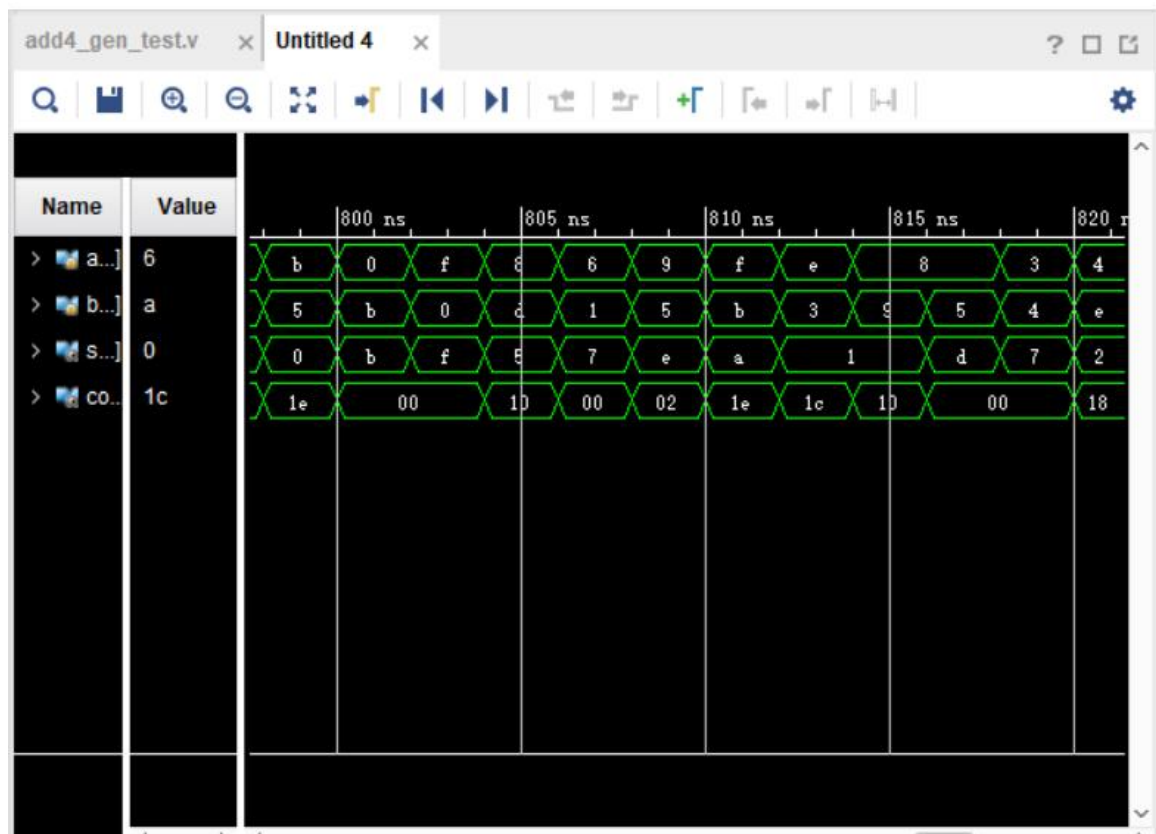
4bit 全加器：



1bit 全加器：

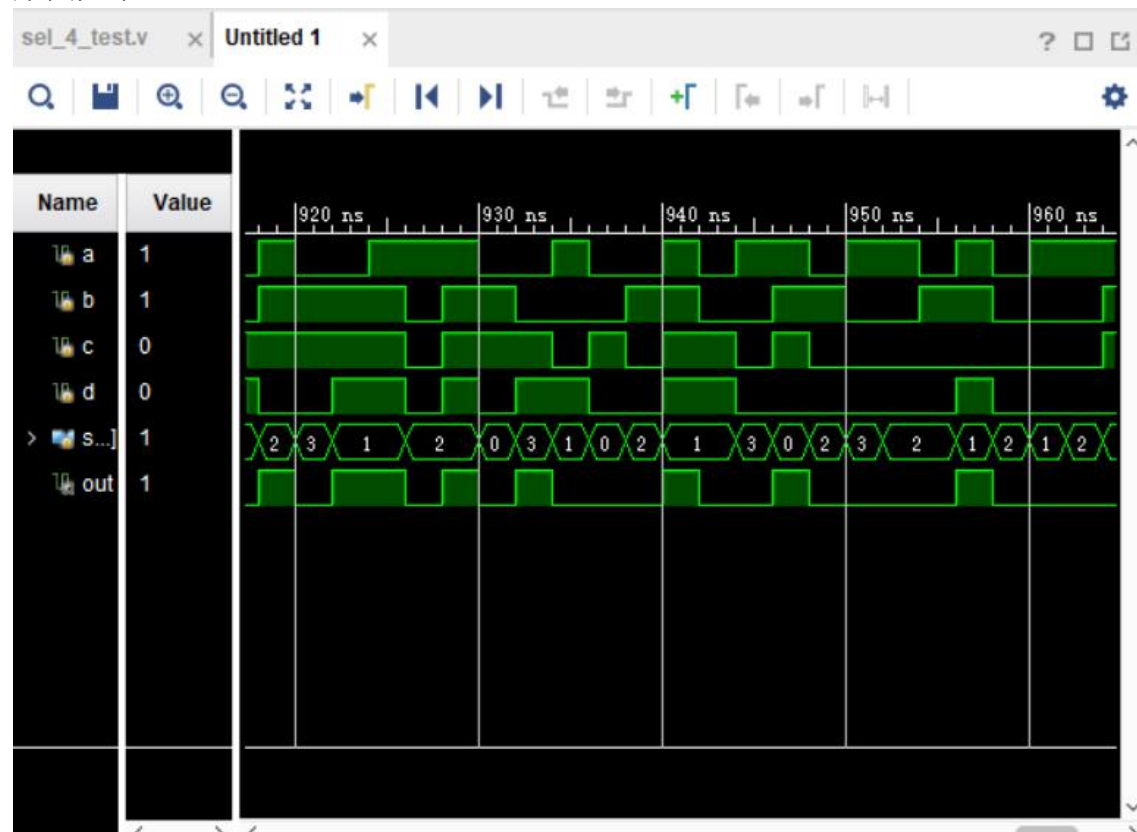


4bit 全加器用 generate 实现:



四、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。



五、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”。