

Lab 2 Exercise 1: Full Adder Simulation Report

姓名：王宇 学号：12112725

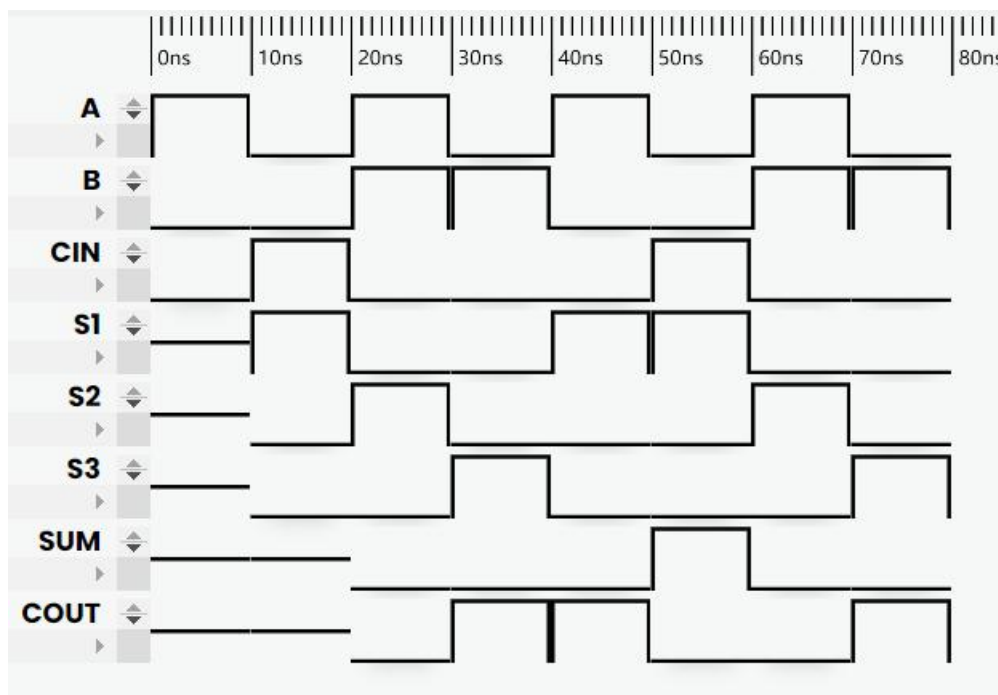
Introduction:

本次实验的目的在于对全加法器进行仿真，并比较三种仿真结果的差异，观察仿真得出波形与自己根据逻辑手动推导的结果是否一致，从中学习理论推导与实际仿真的差异。

实验方法为仿照课件代码写出 design source，然后再自行写出 testbench 进行仿真测试。

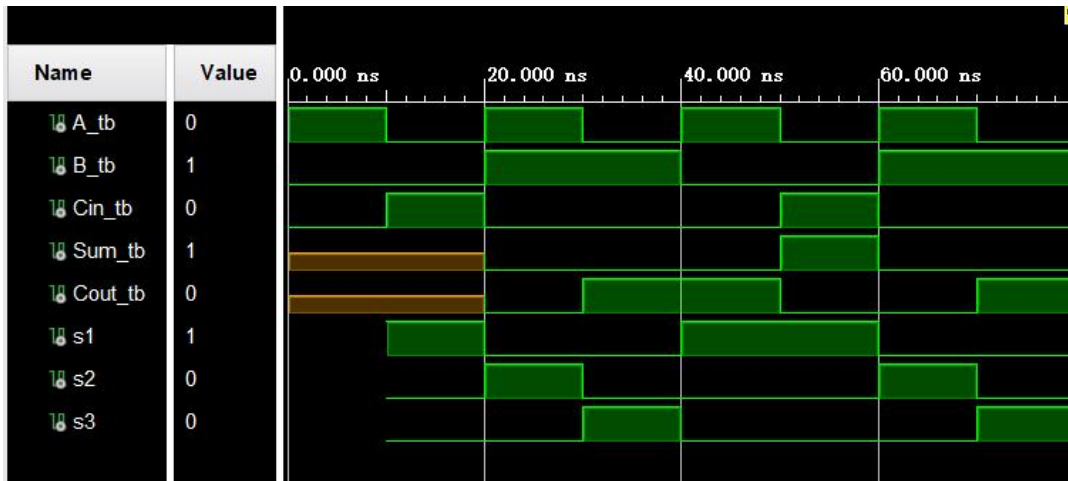
Pre-lab:

根据输入波形，输出波形和中间波形应该如下图所示(采用 TIAGRAM 绘制)，其中 s1, s2, s3 因为是在一个 gate_delay 之后才有定义的，所以应该从 10ns 后开始绘制，而 SUM 和 COUT 的图像取决于 s1, s2, s3，所以要从 20ns 后开始绘制：



Result:

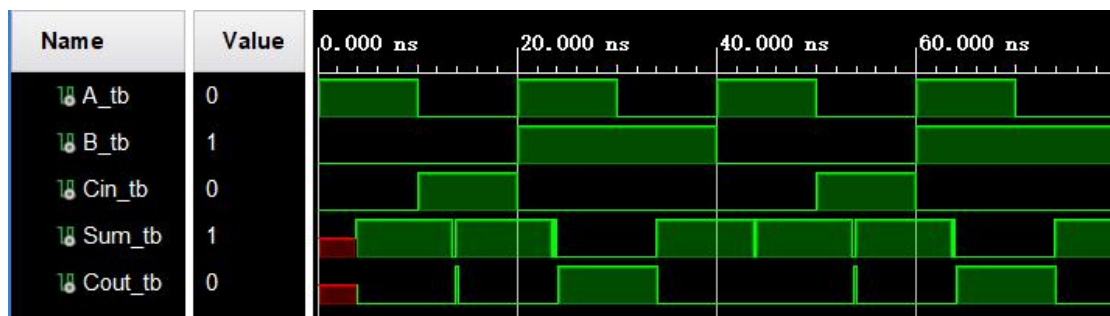
Behavior Simulation 仿真图：



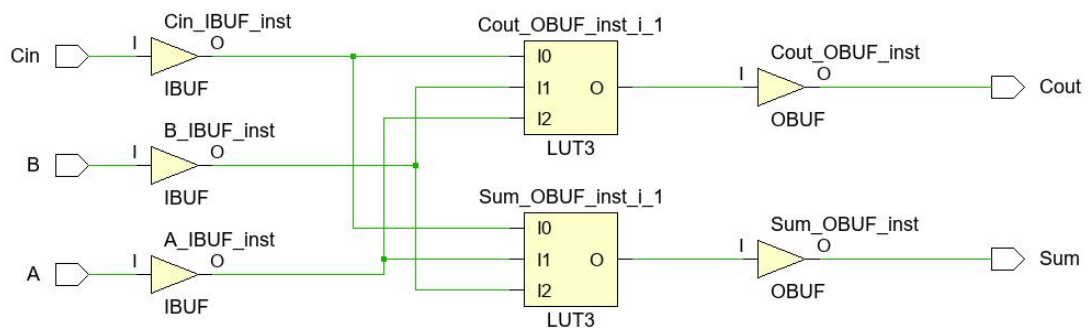
post synthesis simulations 仿真图:



post implementation simulation 仿真图:



模拟电路图如下:



观察结果可以发现，第一个仿真图和我们理想情况得到的结果是一致的。而第二个和第三个图与我们的预想结果明显不一致，第三个图与第二个图相比每隔一段时间会多出一段相反的图像。

Conclusion:

本次实验我学习到了三种仿真的区别。第一个行为仿真仅仅与逻辑和代码有关，与具体的硬件无关，可以用来检查代码有无错误。第二个综合时序仿真可以生成网表，并且已经与具体硬件有了一定的对应关系。第三个后仿真加入了物理层面的延时，最接近真实情况。

Codes:

VHDL code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity full_adder is
port(A,B,Cin: in std_logic; --输入信号
      Sum,Cout: out std_logic); --输出信号
end entity full_adder;

architecture dataflow of full_adder is
    signal s1,s2,s3: std_logic;
    constant gate_delay: time := 10ns;--设置 10ns 的延时
begin
    L1: s1 <= (A xor B) after gate_delay;
    L2: s2 <= (Cin and s1) after gate_delay;
    L3: s3 <= (A and B) after gate_delay;
    L4: sum <= (s1 xor Cin) after gate_delay;
    L5: cout <= (s2 or s3) after gate_delay;
end architecture dataflow;
```

Test bench code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity tb_adder is
-- Port ();
end entity tb_adder;

architecture test of tb_adder is
    component full_adder is
        port(A,B,Cin : in std_logic;
              Sum,Cout: out std_logic);
    end component full_adder;
```

```
signal A_tb,B_tb,Cin_tb: std_logic;
signal Sum_tb,Cout_tb: std_logic;

begin
    UUT:full_adder port map(A=> A_tb, B=> B_tb, Cin=> Cin_tb,
        Sum=> Sum_tb, Cout=> Cout_tb);
    A_tb <= '1','0' after 10ns, '1' after 20ns, '0' after 30ns, '1' after 40ns, '0' after
50ns, '1' after 60ns, '0' after 70ns;--设置 A 信号
    B_tb <= '0','1' after 20ns, '0' after 40ns, '1' after 60ns;--设置 B 信号
    Cin_tb <= '0', '1' after 10ns, '0' after 20ns, '1' after 50ns, '0' after 60ns;--设置
C 信号
end architecture test;
```