FPGA第一次作业

第21组：**门钰涵(主要)**、谷世昌、张浩远、郭羿呈

18050100196

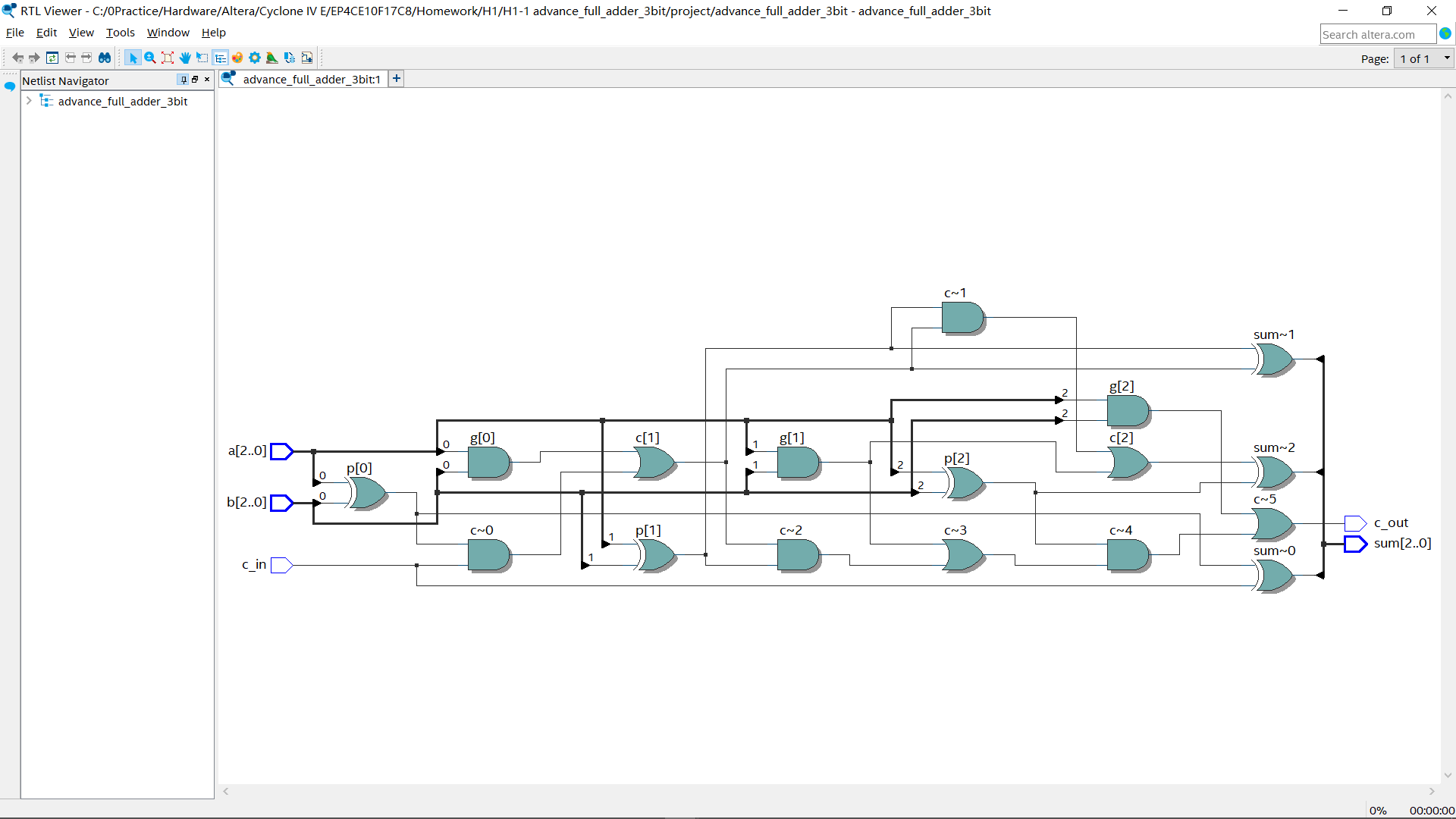
一、作业内容：构建一个3bit的超前进位全加器和一个串行进行全加器。

要求：

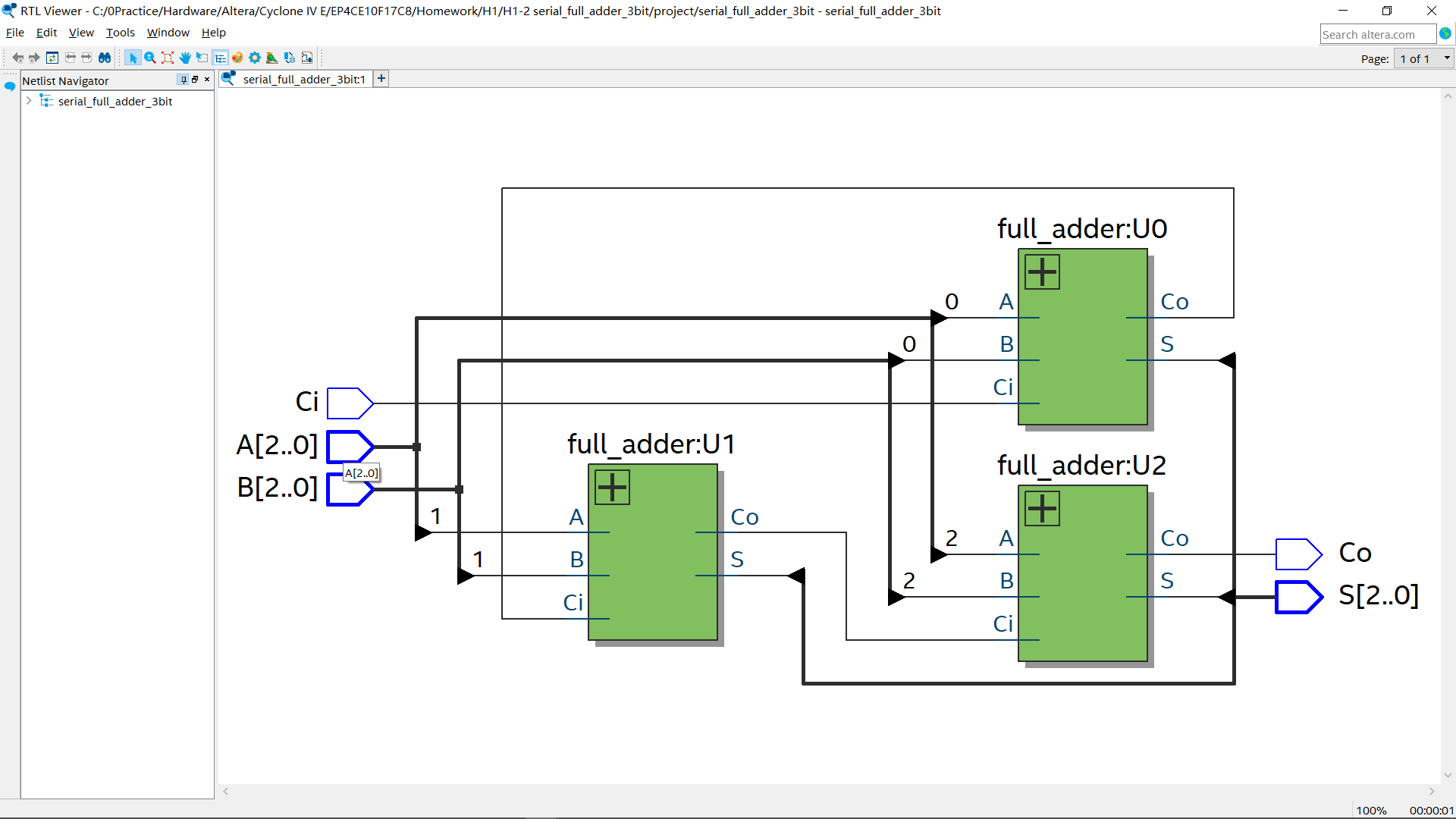
1. 比较二者的关键路径延迟时间，确定各自的最快运行速度。
2. 书写一个测试程序，测试两个全加器的性能。

二、电路图：

1. 3bit的超前进位全加器：



1. 3bit的串行进行全加器：



三、电路分析（要求1完成）：

1. 3bit的超前进位全加器：

延迟时间&最快运行速度：c\_out=8T，Sum=9T；

1. 3bit的串行进行全加器：

延迟时间&最快运行速度：c\_out=9T，Sum=10T；

四、代码：

1. 3bit的超前进位全加器：

模块代码：

module advance\_full\_adder\_3bit(sum,c\_out,a,b,c\_in);

input [2:0] a,b;

input c\_in;

output [2:0] sum;

output c\_out;

wire [3:0] g,p,c;

assign c[0]=c\_in;

assign p=a^b;

assign g=a&b;

assign c[1]=g[0]|(p[0]&c[0]);

assign c[2]=g[1]|(p[1]&(g[0]|(p[0]&c[0])));

assign c[3]=g[2]|(p[2]&(g[1]|(p[1]&(g[0]|(p[0]&c[0])))));

assign sum=p^c[2:0];

assign c\_out=c[3];

endmodule

功能测试代码：

`timescale 1ns/1ns // 定义仿真时间单位1ns和仿真时间精度为1ns

module advance\_full\_adder\_3bit\_tb; // 测试模块

reg [2:0] a,b;

reg c\_in;

wire [2:0] sum;

wire c\_out;

advance\_full\_adder\_3bit U0(.a(a),.b(b),.c\_in(c\_in),.sum(sum),.c\_out(c\_out));

initial

begin

a=3'b001;b=3'b001;c\_in=1'b0;

#20 a=3'b010;b=3'b010;

#20 a=3'b111;b=3'b001;

#20 a=3'b011;b=3'b001;

#20 a=3'b001;b=3'b000;c\_in=1'b1;

end

endmodule

1. 3bit的串行进行全加器：

模块代码：

module full\_adder

(

input A,B,Ci,

output reg S,Co

);

/\*参数说明：

输入：

A,B为一位二进制被加数，Ci为来自低位的进位

输出：

S为和，Co为进位\*/

always@(\*)

begin

case({A,B,Ci})

3'b000:begin S=0;Co=0; end

3'b001:begin S=1;Co=0; end

3'b010:begin S=1;Co=0; end

3'b011:begin S=0;Co=1; end

3'b100:begin S=1;Co=0; end

3'b101:begin S=0;Co=1; end

3'b110:begin S=0;Co=1; end

3'b111:begin S=1;Co=1; end

endcase

end

endmodule

module serial\_full\_adder\_3bit(

input [2:0] A,B,

input Ci,

output [2:0] S,

output Co);

/\*参数说明：

输入：

A,B为3位二进制被加数，Ci为来自低位的进位

输出：

S为和，Co为进位\*/

wire C0,C1,C2;

full\_adder U0(A[0],B[0],Ci,S[0],C0);

full\_adder U1(A[1],B[1],C0,S[1],C1);

full\_adder U2(A[2],B[2],C1,S[2],Co);

endmodule

功能测试代码：

`timescale 1ns/1ns

module serial\_full\_adder\_3bit\_tb();

reg [2:0] A,B;

reg Ci;

wire [2:0] S;

wire Co;

/\*参数说明：

输入：

A,B为一位二进制被加数，Ci为来自低位的进位

输出：

S为和，Co为进位\*/

serial\_full\_adder\_3bit U0(A,B,Ci,S,Co);

initial

$monitor($time,"\tA=%b,B=%b,Ci=%b,S=%b,Co=%b",A,B,Ci,S,Co);

initial

begin

A=3'b001;B=3'b001;Ci=0;

#20;A=3'b111;B=3'b001;

#20;A=3'b101;B=3'b001;

#20;A=3'b001;B=3'b000;Ci=1;

#20;$stop;

end

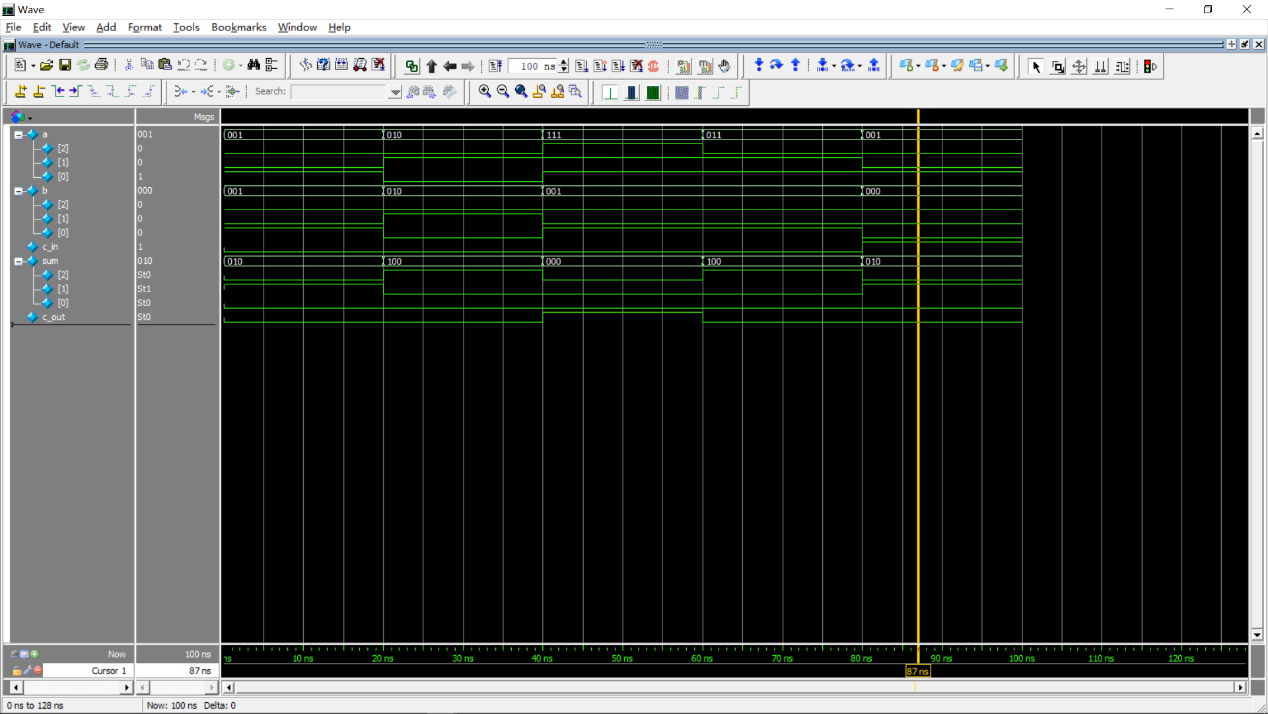
endmodule

**所有工程文件均已上传至Github:**

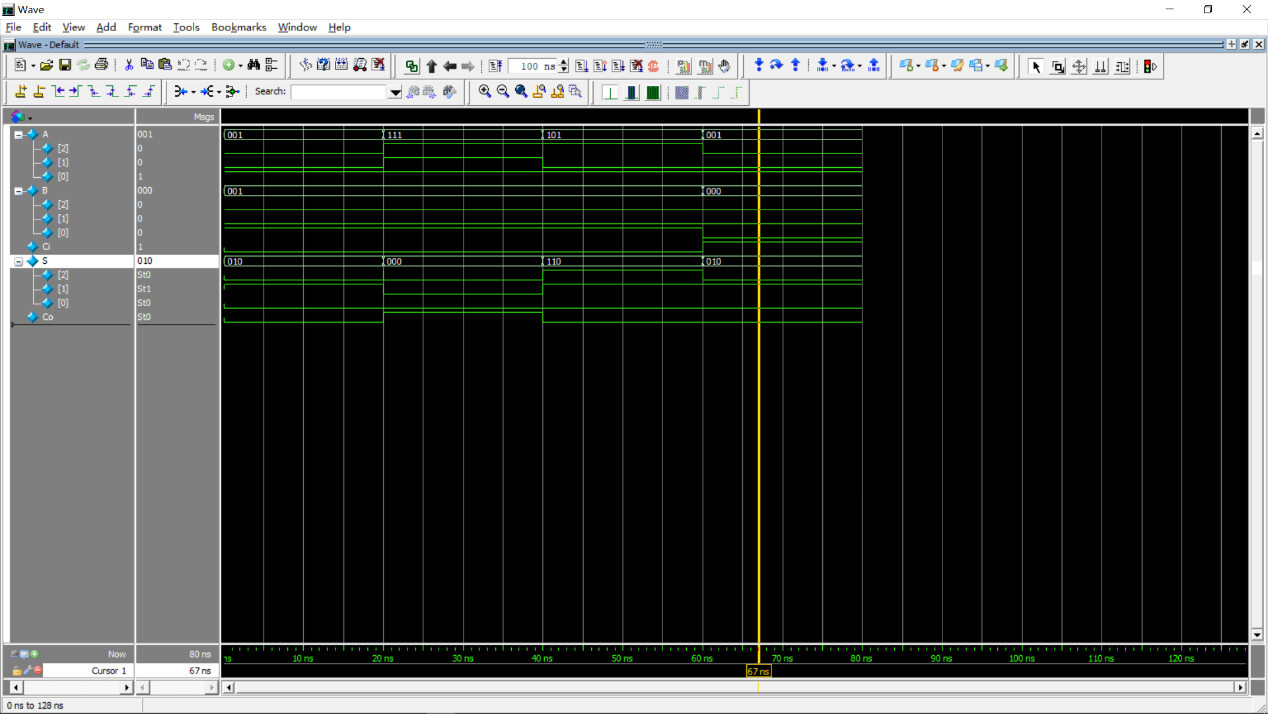
<https://github.com/MYH0/Altera/tree/main/Cyclone%20IV%20E/EP4CE10F17C8/Homework/H1>

五、功能仿真（使用Modelsim-Altera进行仿真）：

1. 3bit的超前进位全加器：

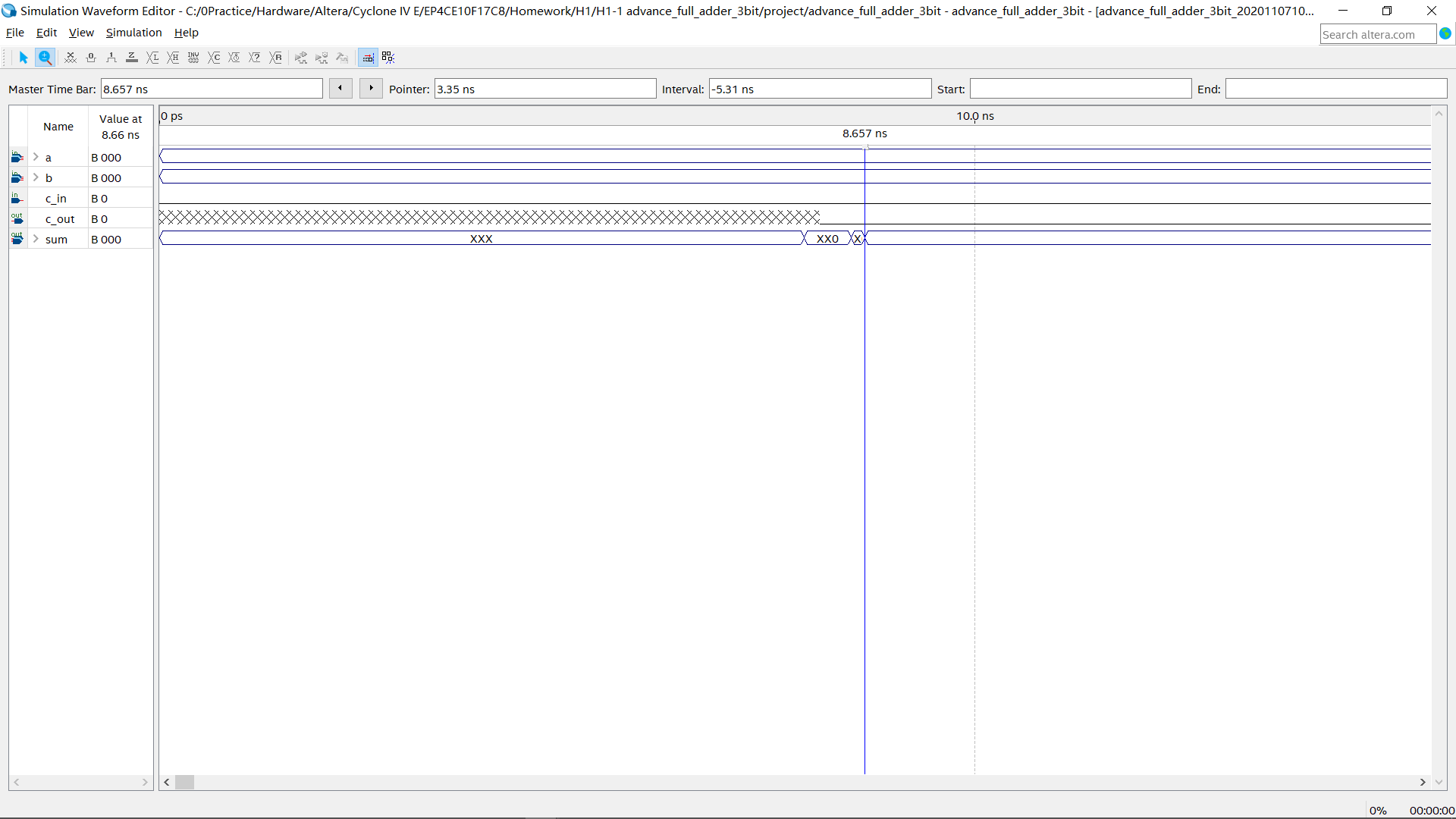


1. 3bit的串行进行全加器：



六、时序仿真（建立VWF文件进行仿真）：

1. 3bit的超前进位全加器：8.657ns



1. 3bit的串行进行全加器：8.952ns

