جامعة بيرزيت

BIRZEIT UNIVERSITY

Digital project : BCD adder_subtractor.

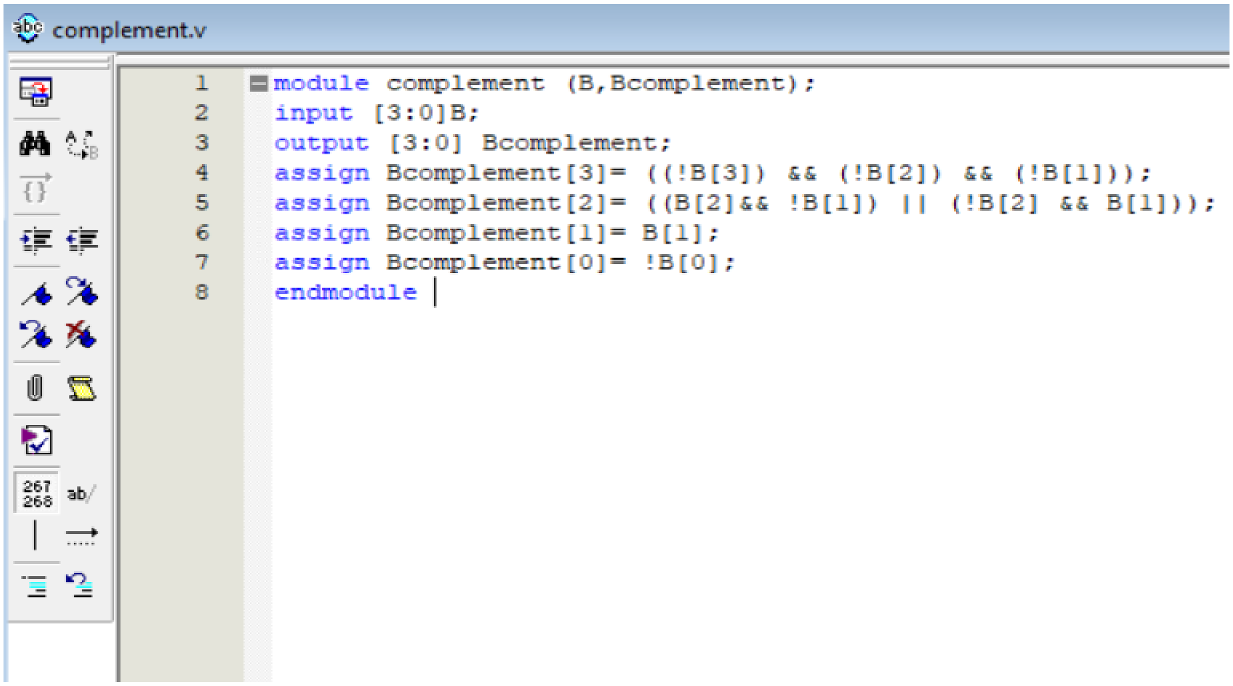
Name : Rahaf Mohammed Naser.

ID : 1201319.

The aim of this project is : to implement the BCD adder_subtractor circuit .

9’s complementor :

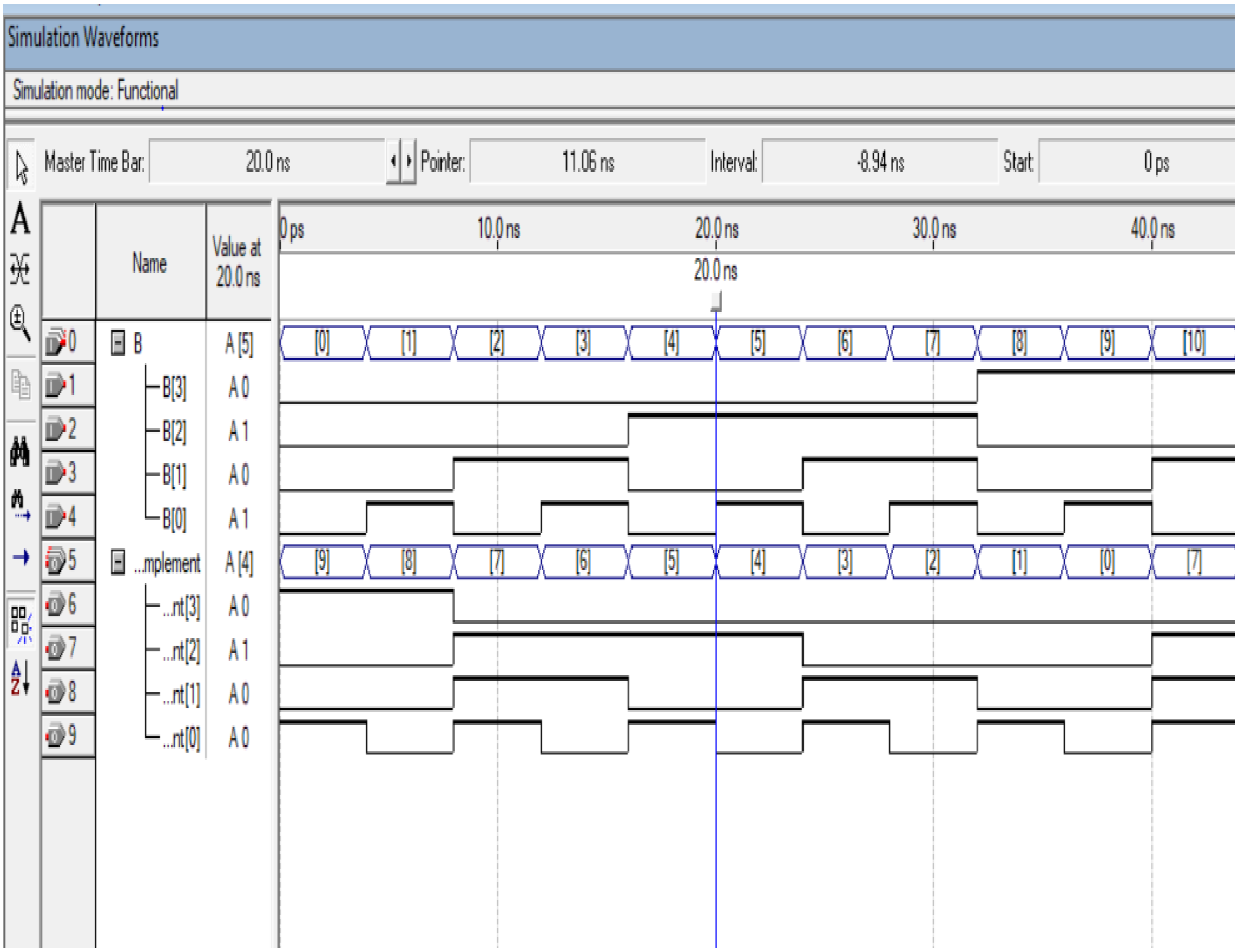
(9’s complementor code):



```
module complement (B,Bcomplement);
input [3:0]B;
output [3:0] Bcomplement;
assign Bcomplement[3]= ((!B[3]) && (!B[2]) && (!B[1]));
assign Bcomplement[2]= ((B[2]&& !B[1]) || (!B[2] && B[1]));
assign Bcomplement[1]= B[1];
assign Bcomplement[0]= !B[0];
endmodule
```

```
module complement (B,Bcomplement);

input [3:0]B;

output [3:0] Bcomplement;

assign Bcomplement[3]= ((!B[3]) && (!B[2]) && (!B[1]));

assign Bcomplement[2]= ((B[2]&& !B[1]) | | (!B[2] && B[1]));

assign Bcomplement[1]= B[1];

assign Bcomplement[0]= !B[0];

endmodule
```

(waveform of the 9’s complementor):

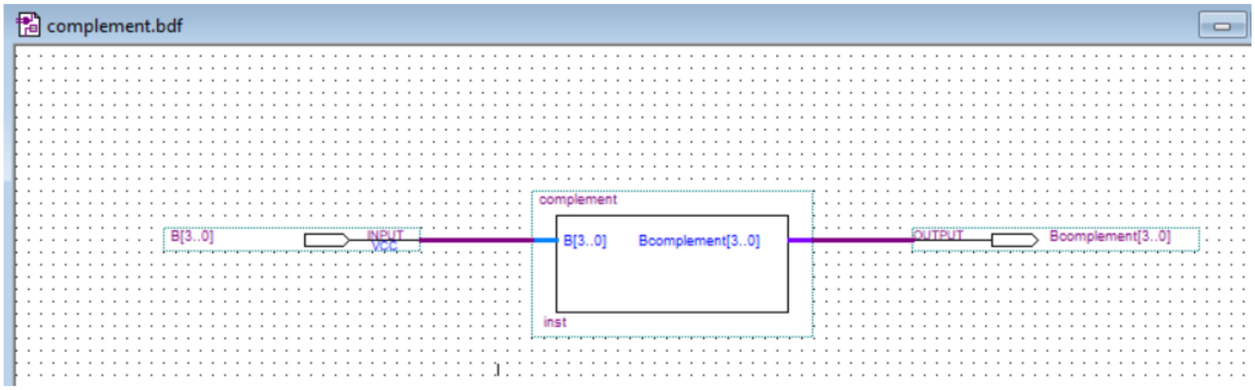

Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 20.0 ns Pointer: 11.06 ns Interval: -8.94 ns Start: 0 ps

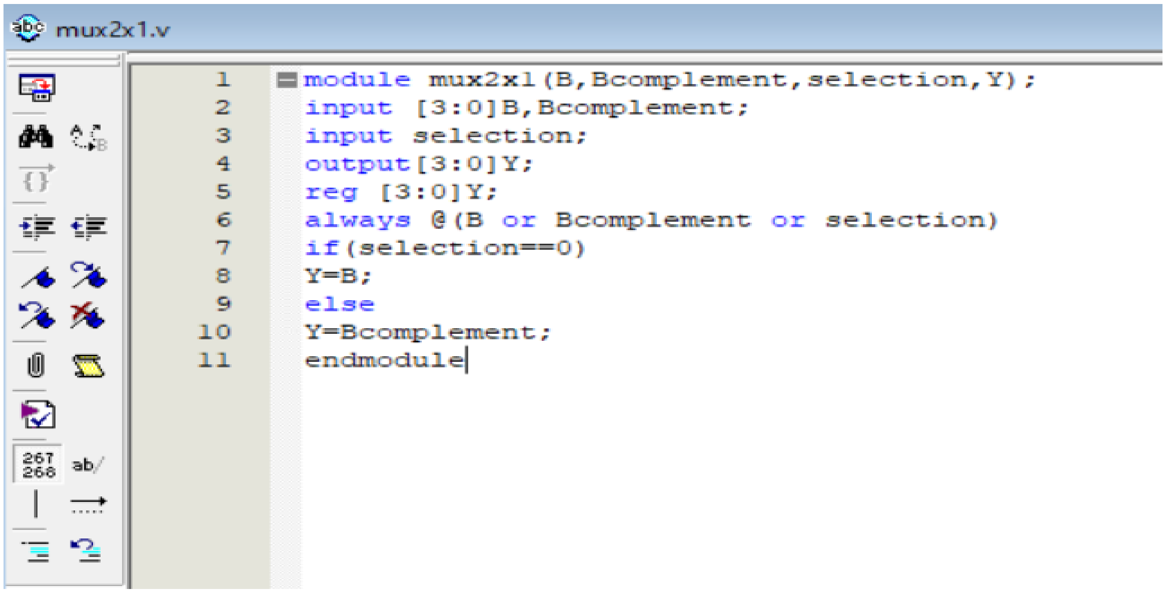
	Name	Value at 20.0 ns
0	B	A [5]
1	B[3]	A 0
2	B[2]	A 1
3	B[1]	A 0
4	B[0]	A 1
5	...mplement	A [4]
6	...nt[3]	A 0
7	...nt[2]	A 1
8	...nt[1]	A 0
9	...nt[0]	A 0

(9’s complement diagram):



Quadruple 2x1 mux:

(quadruple 2x1 mux code):



```
module mux2x1(B,Bcomplement,selection,Y);
input [3:0]B,Bcomplement;
input selection;
output[3:0]Y;
reg [3:0]Y;
always @(B or Bcomplement or selection)
if(selection==0)
Y=B;
else
Y=Bcomplement;
endmodule
```

module mux2x1(B,Bcomplement,selection,Y);

input [3:0]B,Bcomplement;

input selection;

output[3:0]Y;

reg [3:0]Y;

always @(B or Bcomplement or selection)

if(selection==0)

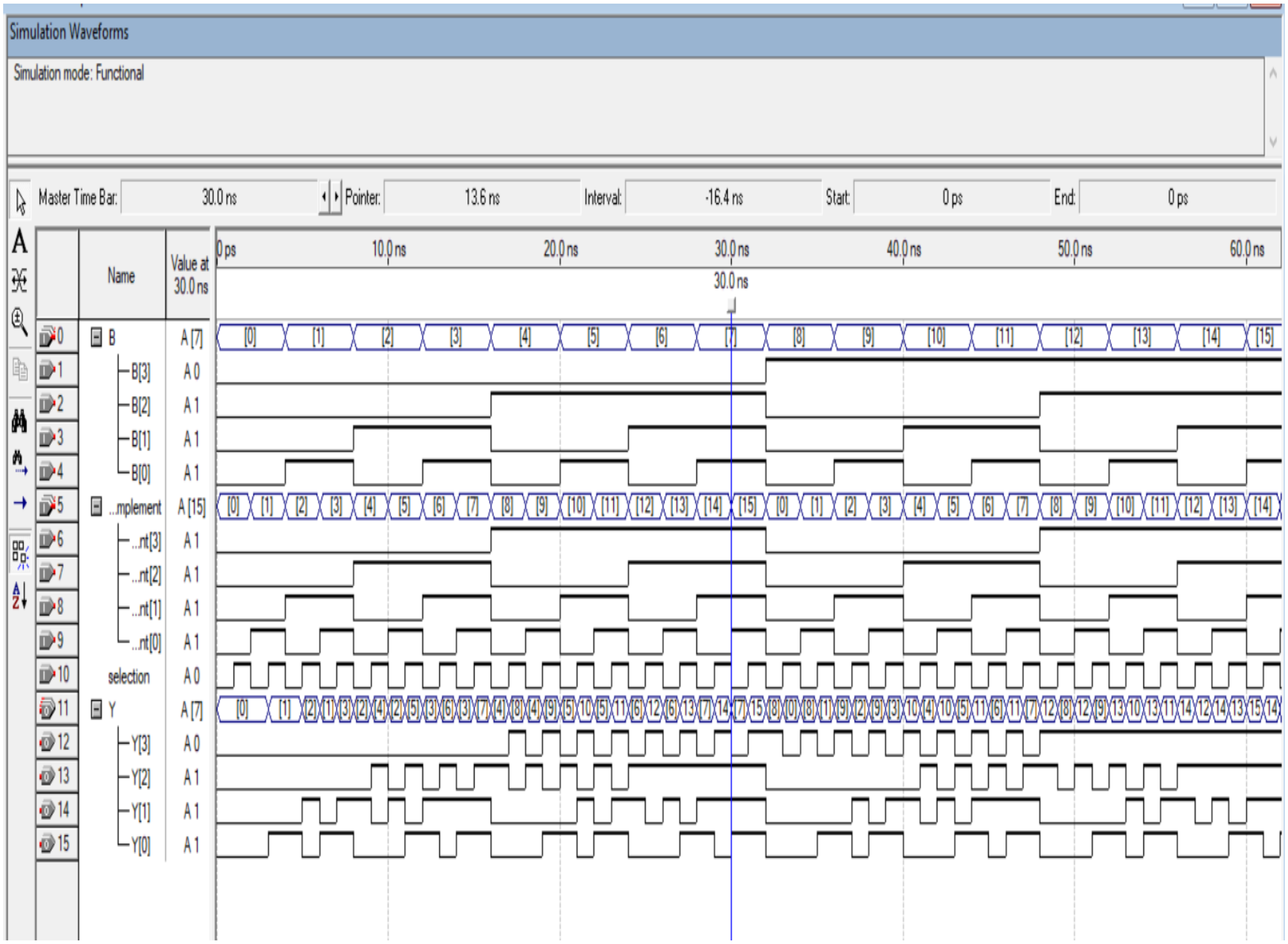
Y=B;

else

Y=Bcomplement;

endmodule

(waveform of quadruple 2x1 mux):



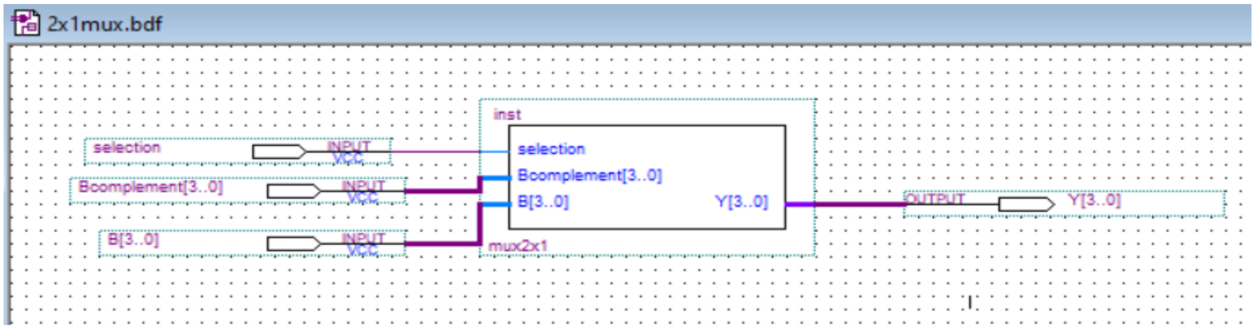
Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 30.0 ns | Pointer: 13.6 ns | Interval: -16.4 ns | Start: 0 ps | End: 0 ps

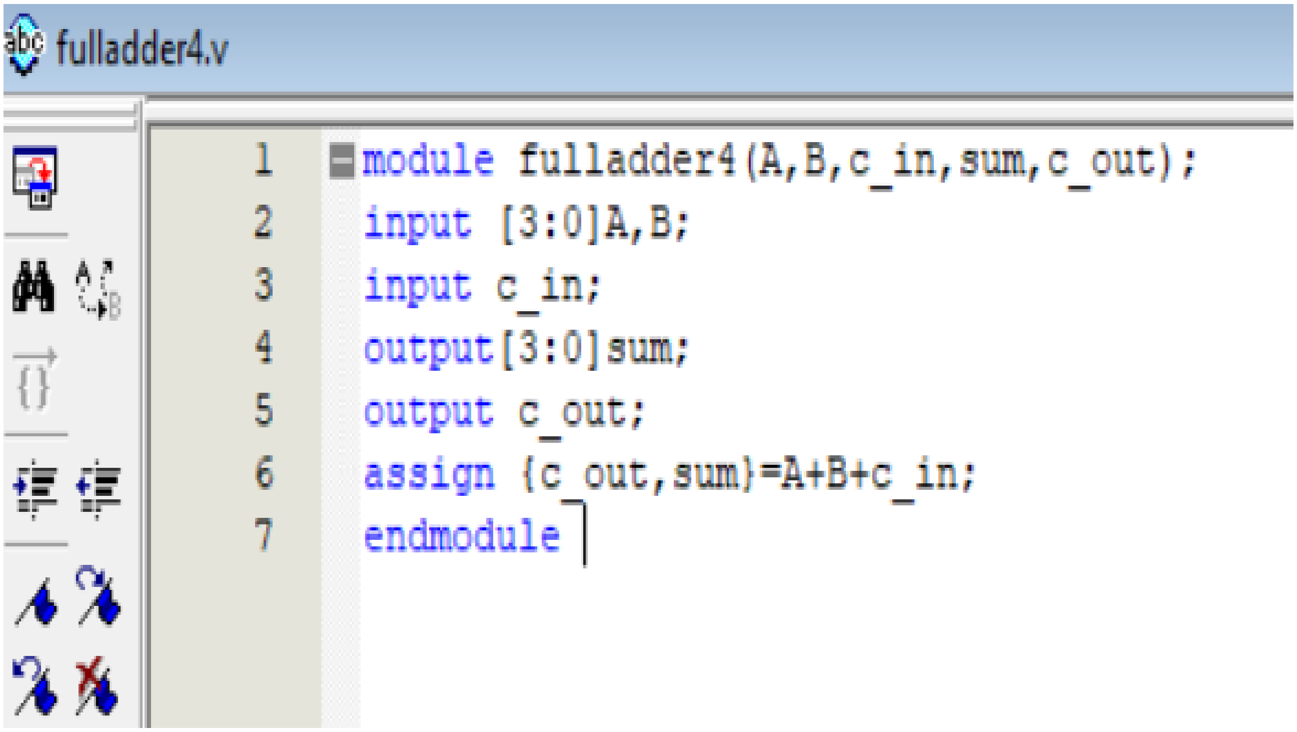
	Name	Value at 30.0 ns
0	B	A [7]
1	B[3]	A 0
2	B[2]	A 1
3	B[1]	A 1
4	B[0]	A 1
5	...mplement	A [15]
6	...nt[3]	A 1
7	...nt[2]	A 1
8	...nt[1]	A 1
9	...nt[0]	A 1
10	selection	A 0
11	Y	A [7]
12	Y[3]	A 0
13	Y[2]	A 1
14	Y[1]	A 1
15	Y[0]	A 1

(2x1mux diagram):



4 bit binary adder:

(4 bit binary adder code):



```
module fulladder4(A,B,c_in,sum,c_out);
 input [3:0]A,B;
 input c_in;
 output[3:0]sum;
 output c_out;
 assign {c_out,sum}=A+B+c_in;
 endmodule
```

module fulladder4(A,B,c_in,sum,c_out);

input [3:0]A,B;

input c_in;

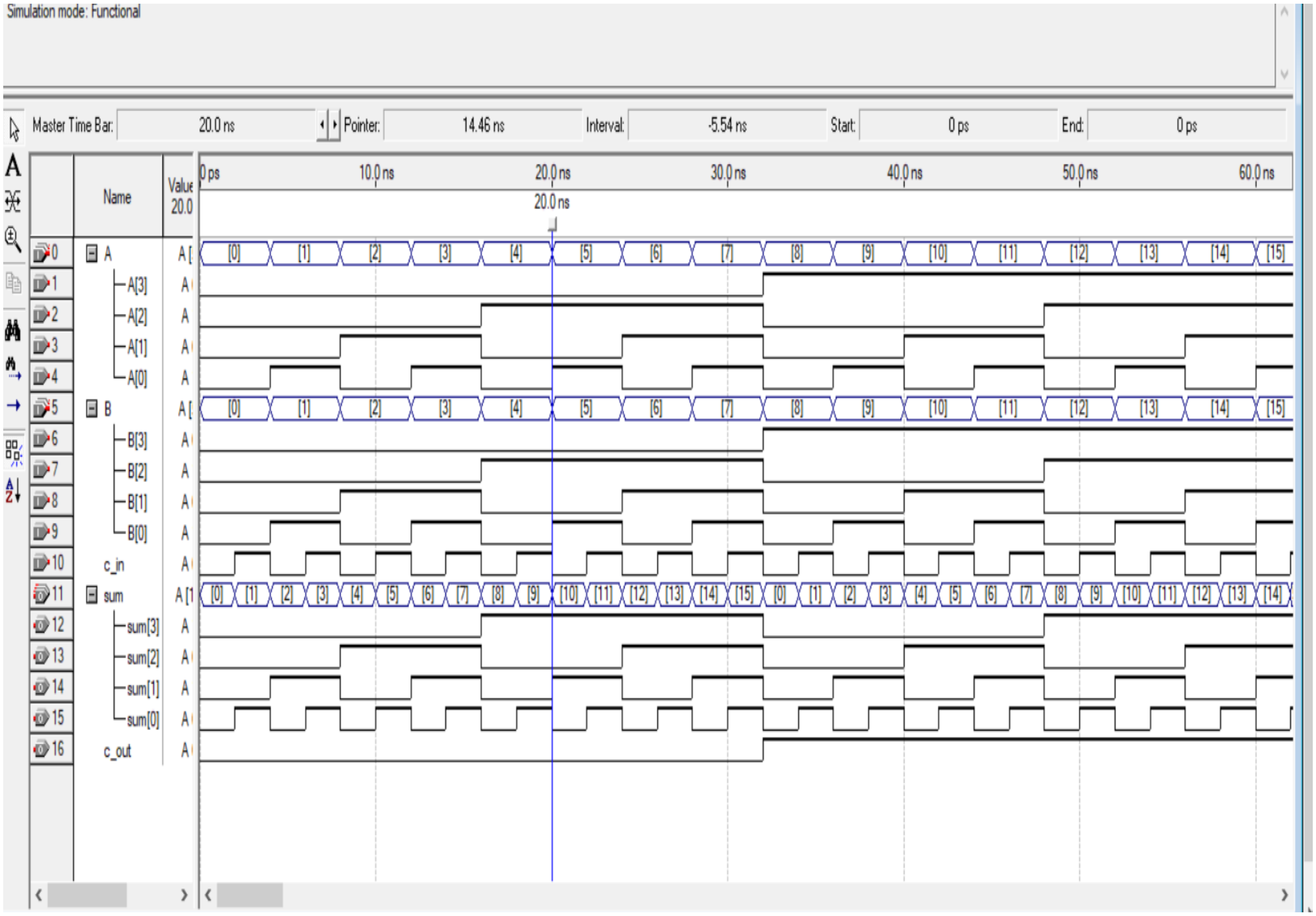
output[3:0]sum;

output c_out;

assign {c_out,sum}=A+B+c_in;

endmodule

(waveform of 4bit binary adder):



Simulation mode: Functional

Master Time Bar: 20.0 ns

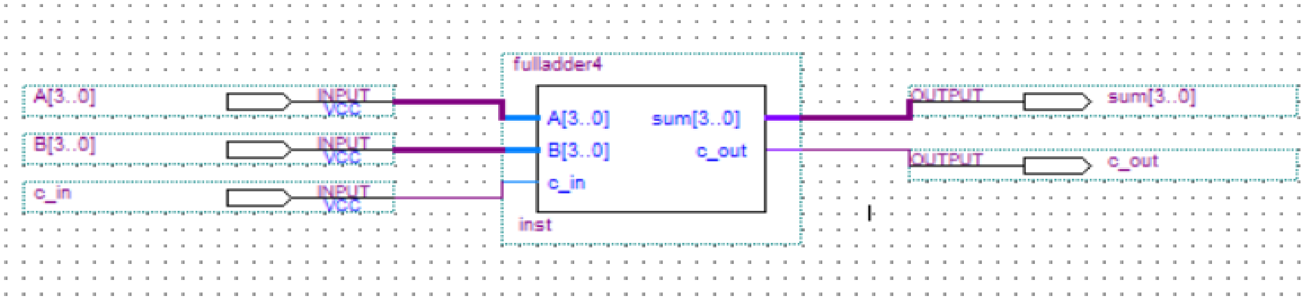
Pointer: 14.46 ns

Interval: -5.54 ns

Start: 0 ps

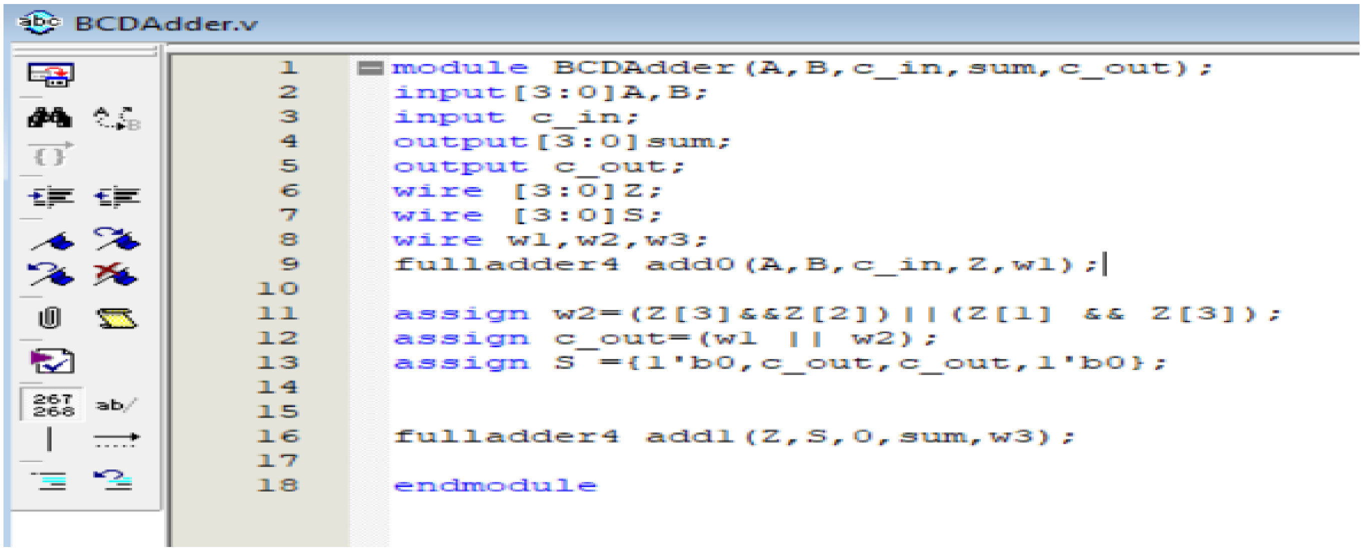
End: 0 ps

(4bit binary adder diagram):



BCD Adder:

(BCD Adder code):

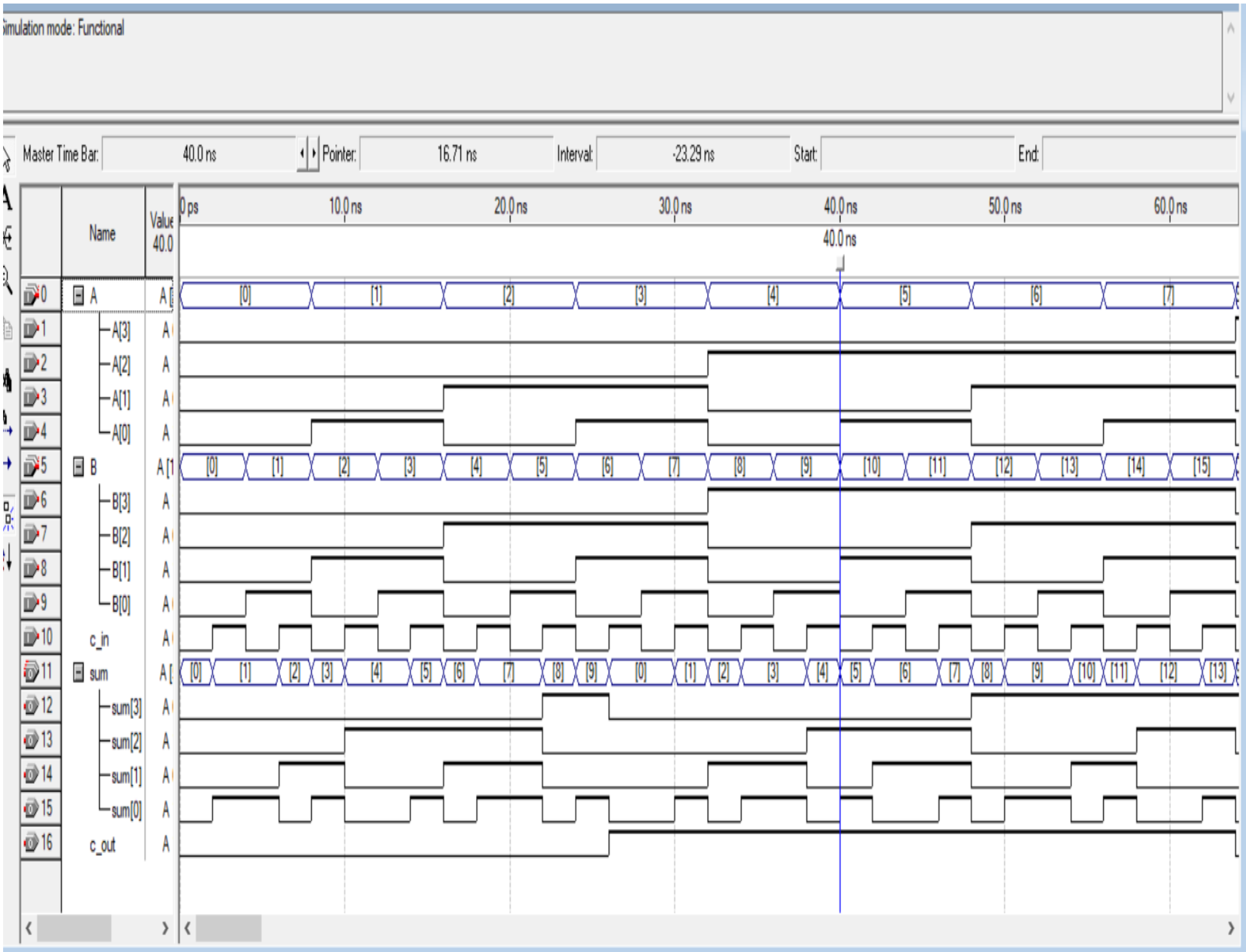


BCDAdder.v

```
module BCDAdder(A,B,c_in,sum,c_out);
input[3:0]A,B;
input c_in;
output[3:0]sum;
output c_out;
wire [3:0]Z;
wire [3:0]S;
wire w1,w2,w3;
fulladder4 add0(A,B,c_in,Z,w1);

assign w2=(Z[3]&&Z[2])||(Z[1] && Z[3]);
assign c_out=(w1 || w2);
assign S ={1'b0,c_out,c_out,1'b0};


fulladder4 add1(Z,S,0,sum,w3);

endmodule
```

```
module BCDAdder (A,B,c_in,sum,c_out);

input [3:0] A, B;

input c_in;

output[3:0]sum;

output  c_out;

wire  [3:0] Z;

wire [3:0]S;

wire w1,w2,w3;

fulladder4 add0(A,B,c_in,Z,w1);

assign w2= ( Z[3]  &&  Z[2] )  ||   (  Z[1]   &&   Z[3] );

assign c_out=(w1 || w2);

assign S ={1'b0 ,c_out, c_out, 1'b0};

fulladder4 add1(Z,S,0,sum,w3);

endmodule
```

(Waveform of BCD Adder):



Simulation mode: Functional

Master Time Bar: 40.0 ns

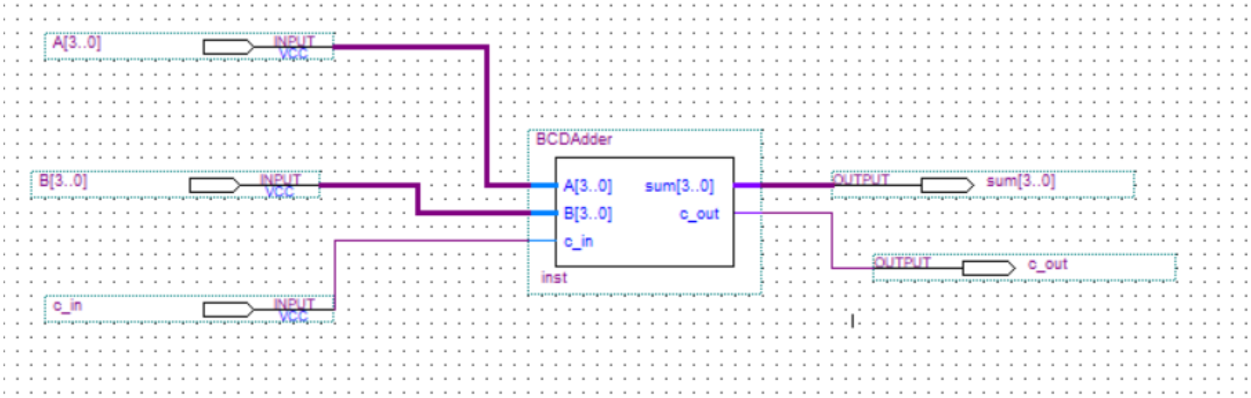
Pointer: 16.71 ns

Interval: -23.29 ns

Start:

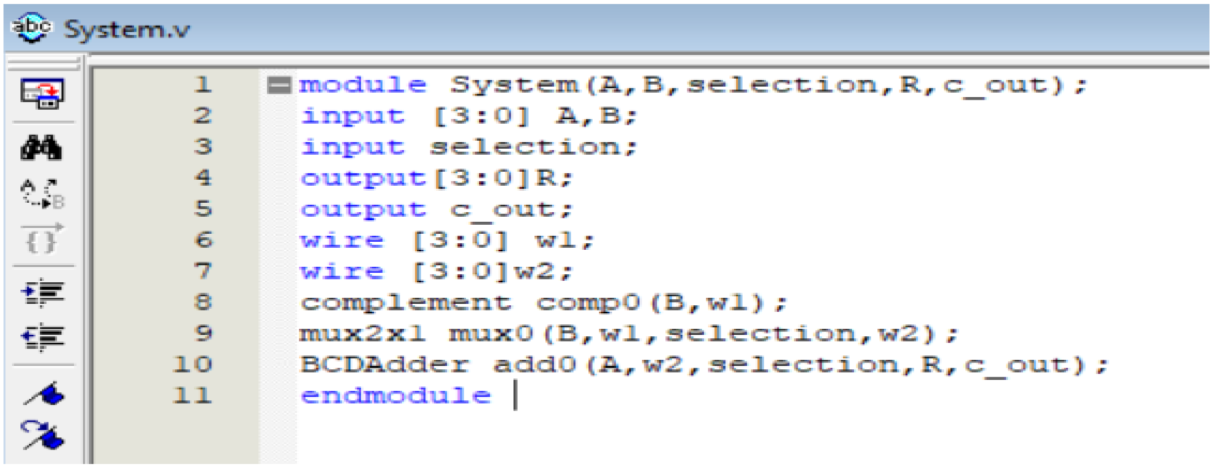
End:

(BCD Adder diagram):



The whole system:

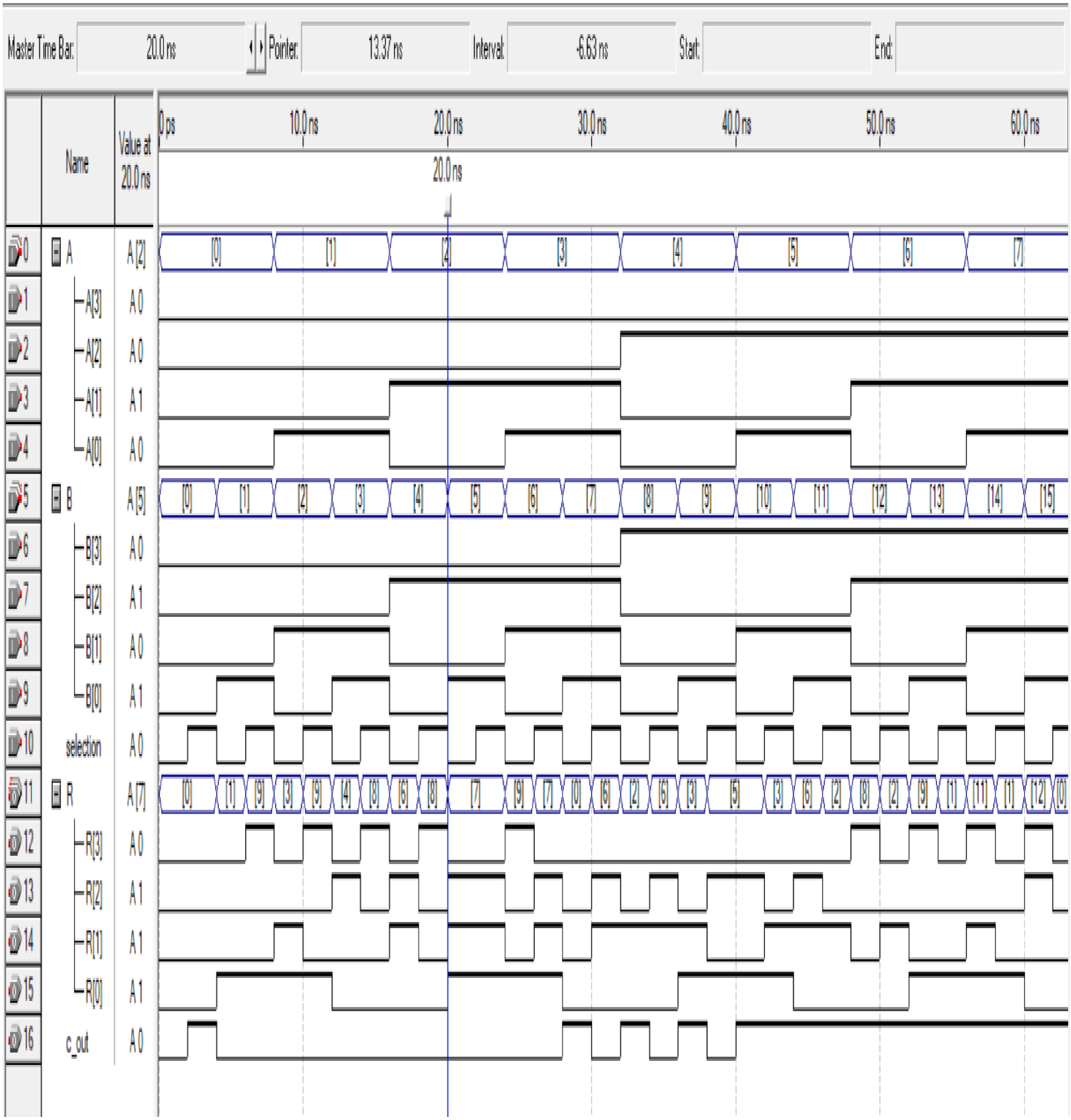
(The code of system):



```
module System(A,B,selection,R,c_out);
input [3:0] A,B;
input selection;
output[3:0]R;
output c_out;
wire [3:0] w1;
wire [3:0]w2;
complement comp0(B,w1);
mux2x1 mux0(B,w1,selection,w2);
BCDAdder add0(A,w2,selection,R,c_out);
endmodule
```

```
module Adder_Subtractor(A,B,selection,R,c_out);

input [3:0] A,B;

input selection;

output[3:0]R;

output c_out;

wire [3:0] w1;

wire [3:0]w2;

complement comp0(B,w1);

mux2x1 mux0(B,w1,selection,w2);

BCDAdder add0(A,w2,selection,R,c_out);

endmodule
```

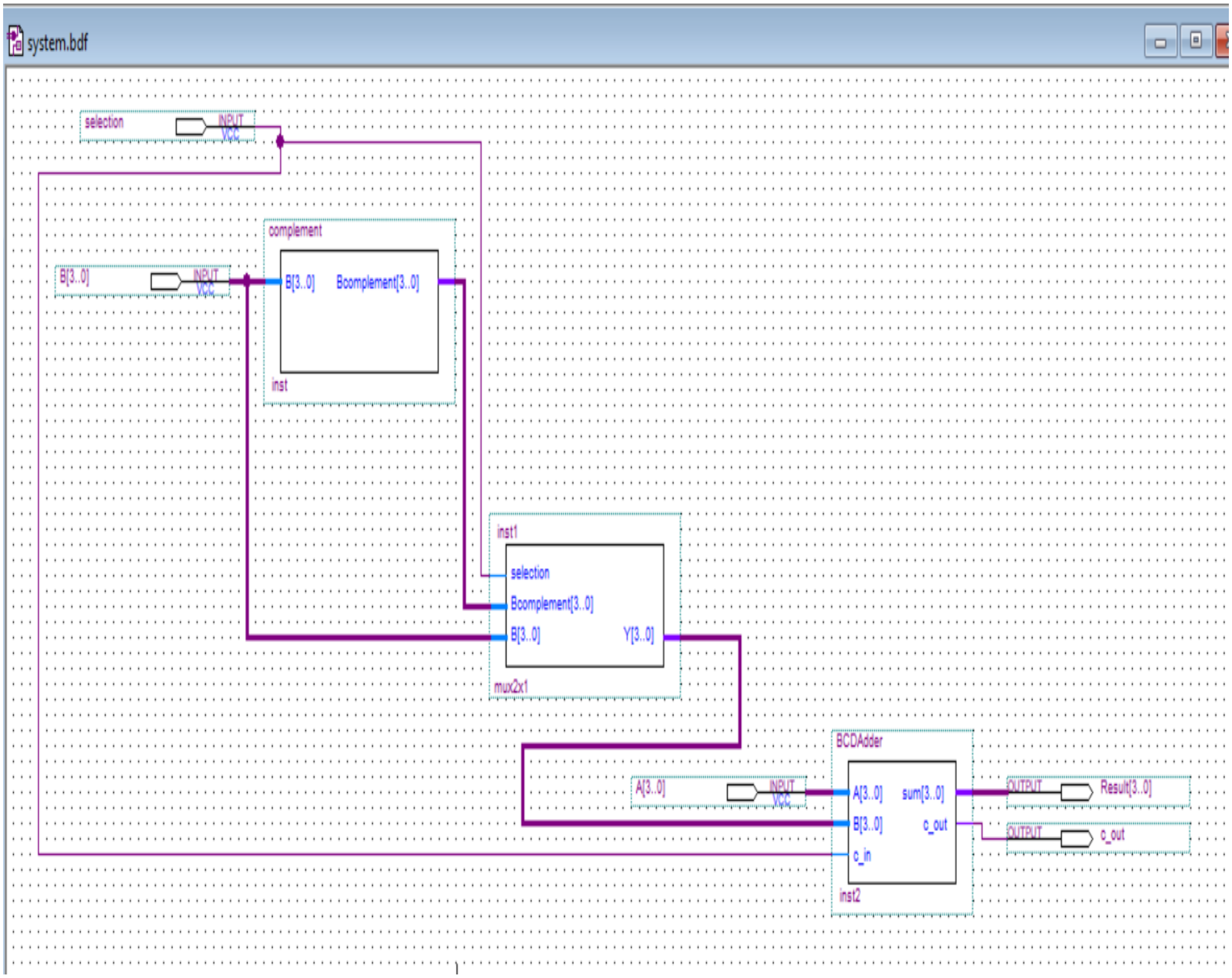
(The waveform of whole system):



Master Time Bar: 20.0 ns Pointer: 13.37 ns Interval: -6.63 ns Start: End:

	Name	Value at 20.0 ns
0	A	A [2]
1	A[3]	A 0
2	A[2]	A 0
3	A[1]	A 1
4	A[0]	A 0
5	B	A [5]
6	B[3]	A 0
7	B[2]	A 1
8	B[1]	A 0
9	B[0]	A 1
10	selection	A 0
11	R	A [7]
12	R[3]	A 0
13	R[2]	A 1
14	R[1]	A 1
15	R[0]	A 1
16	c_out	A 0

(The diagram of whole system):


system.bdf
selection
INPUT
VCC
B[3..0]
INPUT
VCC
complement
B[3..0]
Bcomplement[3..0]
inst
inst1
selection
Bcomplement[3..0]
B[3..0]
Y[3..0]
mux2x1
BCDAdder
A[3..0]
INPUT
VCC
A[3..0]
sum[3..0]
B[3..0]
c_out
c_in
inst2
OUTPUT
Result[3..0]
OUTPUT
c_out