EXPERIMENT- 3
Verilog Structural Modeling

Objective:

To understand the concepts related to structural modeling style and write Verilog programs using the same.

Theory: Structures can be described in Verilog HDL using

i. Built in gate primitives (at the gate level) [Note-will be used in this Lab]
ii. Switch level primitives (at the transistor level)
iii. User defined primitives (at the gate level)
iv. Module primitives (to create hierarchy) [Note-will be used in this Lab]

A module can be instantiated in another module, thus creating hierarchy. A module instantiation statement is of the form:

```
Module_name instance_name ( port_association);
```

Port association can be by position or by name, however associations cannot be mixed. A port association is of the form.

```
port_expr
.portname(port_expr)
```

In positional association, the port expressions connect to the ports of the module in the specified order. In association by name, the connection between the module port and the port expression is explicitly specified and thus the order of port associations is not important.

Exercise Problems

1. **Write structural Verilog code for mod-14 ripple counter and verify the design by simulation.**

Source Code

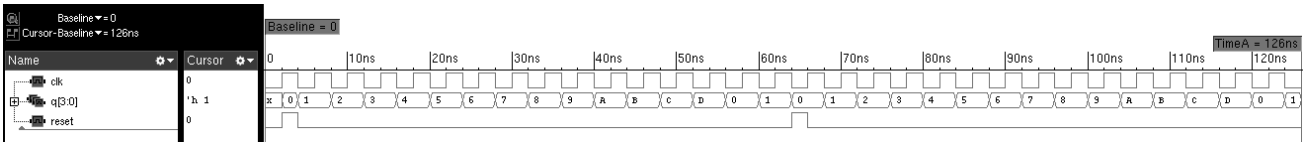
```
module mod14ripplecounter(input clk,reset,output wire [3:0]q);
TFF tff0(.t(1),.clk(clk),.clear(clr),.reset(reset),.q(q[0]));
TFF tff1(.t(1),.clk(q[0]),.clear(clr),.reset(reset),.q(q[1]));
TFF tff2(.t(1),.clk(q[1]),.clear(clr),.reset(reset),.q(q[2]));
TFF tff3(.t(1),.clk(q[2]),.clear(clr),.reset(reset),.q(q[3]));
and (clr,q[3],q[2],q[1],~q[0]);
endmodule
module TFF(input t,clk,clear,reset,output reg q);
always@(clear)
q<=0;
always @(negedge clk or posedge reset) begin
if (reset)
q <= 0;
else if (t)
q <= ~q;
else
q <= q;
end
endmodule
```

Testbench

```
module tb();
reg clk,reset;
wire [3:0]q;
mod14ripplecounter uut(.clk(clk),.reset(reset),.q(q));
initial begin
clk = 0;
forever #2 clk = ~clk;
end
initial begin
reset = 0;
#2 reset =1;
```

```
#2 reset =0;
#60 reset =1;
#2 reset=0;
#60 $finish;
end
initial begin
$monitor($time, " clk=%b, q=%b",clk,q);
end
endmodule
```

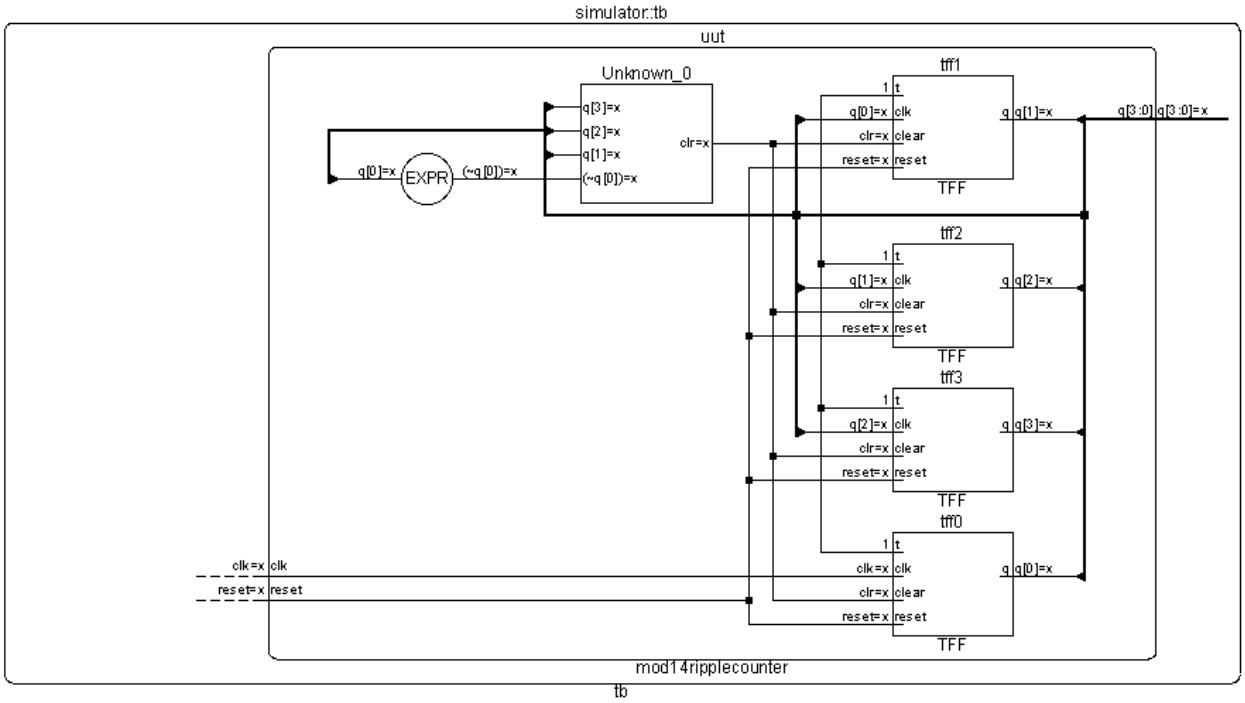
Waveform



Console

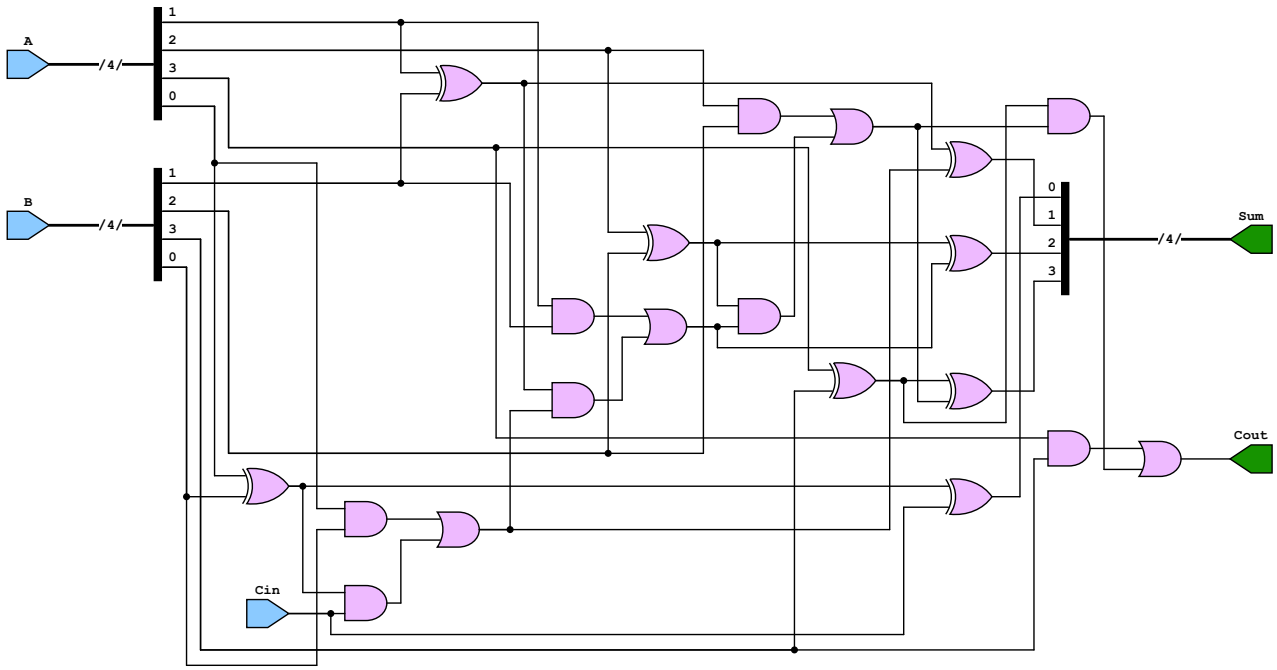
```
ncsim> run
                   0 clk=0, q=xxxx
                   2 clk=1, q=0000
                   4 clk=0, q=0001
                   6 clk=1, q=0001
                   8 clk=0, q=0010
                  10 clk=1, q=0010
                  12 clk=0, q=0011
                  14 clk=1, q=0011
                  16 clk=0, q=0100
                  18 clk=1, q=0100
                  20 clk=0, q=0101
                  22 clk=1, q=0101
                  24 clk=0, q=0110
                  26 clk=1, q=0110
                  28 clk=0, q=0111
                  30 clk=1, q=0111
                  32 clk=0, q=1000
                  34 clk=1, q=1000
                  36 clk=0, q=1001
                  38 clk=1, q=1001
                  40 clk=0, q=1010
                  42 clk=1, q=1010
                  44 clk=0, q=1011
                  46 clk=1, q=1011
                  48 clk=0, q=1100
                  50 clk=1, q=1100
                  52 clk=0, q=1101
                  54 clk=1, q=1101
                  56 clk=0, q=0000
                  58 clk=1, q=0000
                  60 clk=0, q=0001
                  62 clk=1, q=0001
                  64 clk=0, q=0000
                  66 clk=1, q=0000
                  68 clk=0, q=0001
                  70 clk=1, q=0001
                  72 clk=0, q=0010
                  74 clk=1, q=0010
                  76 clk=0, q=0011
                  78 clk=1, q=0011
                  80 clk=0, q=0100
                  82 clk=1, q=0100
                  84 clk=0, q=0101
                  86 clk=1, q=0101
                  88 clk=0, q=0110
                  90 clk=1, q=0110
                  92 clk=0, q=0111
                  94 clk=1, q=0111
                  96 clk=0, q=1000
                  98 clk=1, q=1000
                 100 clk=0, q=1001
                 102 clk=1, q=1001
                 104 clk=0, q=1010
                 106 clk=1, q=1010
                 108 clk=0, q=1011
                 110 clk=1, q=1011
                 112 clk=0, q=1100
                 114 clk=1, q=1100
                 116 clk=0, q=1101
                 118 clk=1, q=1101
                 120 clk=0, q=0000
                 122 clk=1, q=0000
                 124 clk=0, q=0001
Simulation complete via $finish(1) at time 126 NS + 0
./mod14ripplecounter_tb.v:15 #60 $finish;
ncsim>
```

Schematic



2. Write structural Verilog code for 4-bit carry look ahead adder and verify the design by simulation.

Circuit



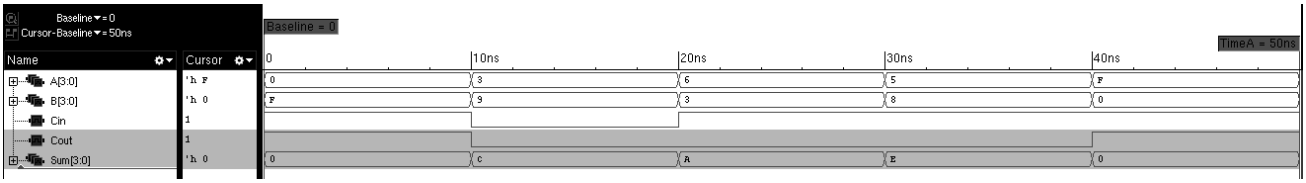
Source Code

```
module claa(input [3:0]A,B,input Cin,output [3:0]Sum,output Cout);
wire [3:0] P, G, C;
wire pp1,pp2,pp3,pp4;
xor (P[0],A[0],B[0]);
xor (P[1],A[1],B[1]);
xor (P[2],A[2],B[2]);
xor (P[3],A[3],B[3]);

and (G[0],A[0],B[0]);
and (G[1],A[1],B[1]);
and (G[2],A[2],B[2]);
and (G[3],A[3],B[3]);

assign C[0] = Cin;

and a1(pp1,P[0],C[0]);
or o1(C[1],G[0],pp1);

and a2(pp2,P[1],C[1]);
or o2(C[2],G[1],pp2);

and a3(pp3,P[2],C[2]);
or o3(C[3],G[2],pp3);

and a4(pp4,P[3],C[3]);
or o4(Cout,G[3],pp4);

xor (Sum[0],P[0],C[0]);
xor (Sum[1],P[1],C[1]);
xor (Sum[2],P[2],C[2]);
xor (Sum[3],P[3],C[3]);

endmodule
```

Testbench

```
module tb();
reg [3:0] A, B;
reg Cin;
wire [3:0] Sum;
wire Cout;
claa uut (.A(A),.B(B),.Cin(Cin),.Sum(Sum),.Cout(Cout));
initial begin
$monitor($time," A:%b,B:%b,Cin:%b,Sum:%b,Cout:%b ",A,B,Cin,Sum,Cout);
A=4'b0000; B=4'b1111; Cin=1;
#10;
A=4'b0011; B=4'b1001; Cin=0;
#10;
A=4'b0110; B=4'b0011; Cin=1;
#10;
A=4'b0101; B=4'b1000; Cin=1;
#10;
A=4'b1111; B=4'b0000; Cin=1;
#10;
$finish;
end
endmodule
```

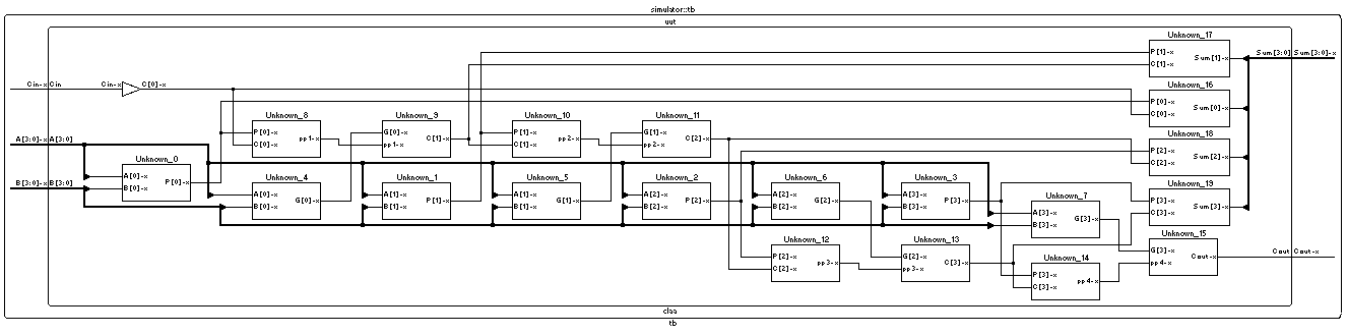
Waveform



Console

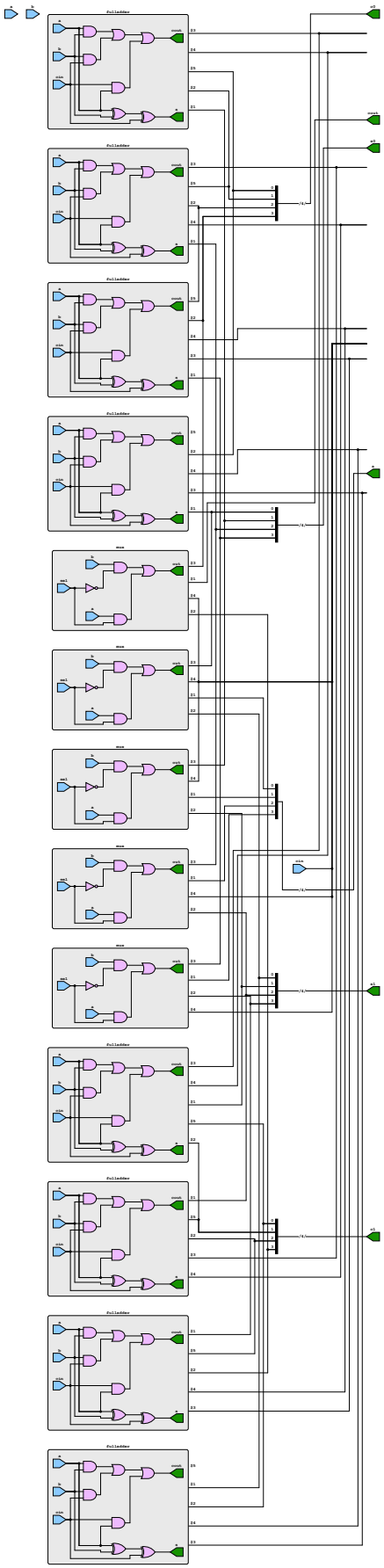
```
ncsim>
ncsim> source /home/install/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
database -open waves -into waves.shm -default
probe -create -shm tb.A tb.B tb.Cin tb.Cout tb.Sum
run
Created default SHM database waves
ncsim> Created probe 1
ncsim>                      0 A:0000,B:1111,Cin:1,Sum:0000,Cout:1
                   10 A:0011,B:1001,Cin:0,Sum:1100,Cout:0
                   20 A:0110,B:0011,Cin:1,Sum:1010,Cout:0
                   30 A:0101,B:1000,Cin:1,Sum:1110,Cout:0
                   40 A:1111,B:0000,Cin:1,Sum:0000,Cout:1
Simulation complete via $finish(1) at time 50 NS + 0
./cla4bit_tb.v:19 $finish;
ncsim>
```

Schematic



3. Write structural Verilog code for 4-bit carry select adder and verify the design by simulation.

Circuit



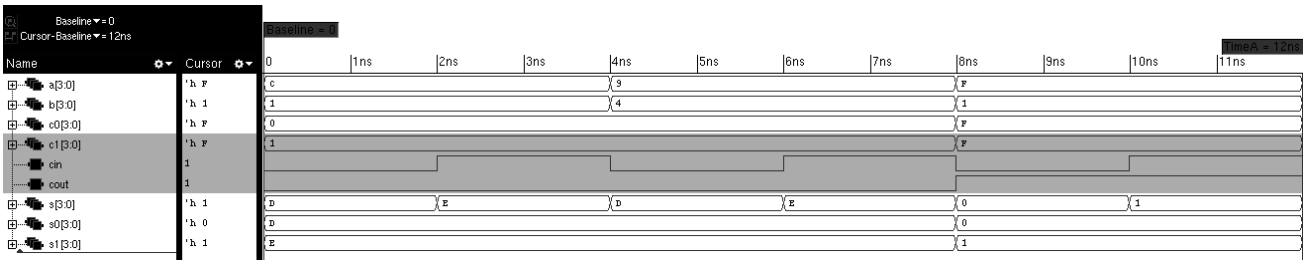
Source Code

```
module carryselect(output [3:0]s,s0,s1,c0,c1,output cout,input
[3:0]a,b,input cin);

fulladder f0(s0[0],c0[0],a[0],b[0],1'b0);
fulladder f1(s0[1],c0[1],a[1],b[1],c0[0]);
fulladder f2(s0[2],c0[2],a[2],b[2],c0[1]);
fulladder f3(s0[3],c0[3],a[3],b[3],c0[2]);

fulladder f4(s1[0],c1[0],a[0],b[0],1'b1);
fulladder f5(s1[1],c1[1],a[1],b[1],c1[0]);
fulladder f6(s1[2],c1[2],a[2],b[2],c1[1]);
fulladder f7(s1[3],c1[3],a[3],b[3],c1[2]);

mux m0(s[0],s1[0],s0[0],cin);
mux m1(s[1],s1[1],s0[1],cin);
mux m2(s[2],s1[2],s0[2],cin);
mux m3(s[3],s1[3],s0[3],cin);
mux m4(cout,c1[3],c0[3],cin);

endmodule

module mux(output out,input a,b,sel);
not n1(sel_n,sel);
and a1(sel_b,b,sel_n); //sel=0 out=b
and a2(sel_a,a,sel); //sel=1 out=a
or o1(out,sel_b,sel_a);
endmodule

module fulladder(output s,cout,input a,b,cin);
xor sum(s,a,b,cin);
and a1(c1,a,b);
and a2(c2,b,cin);
and a3(c3,cin,a);
or carry(cout,c1,c2,c3);
endmodule
```

Testbench

```
module tb();
reg [3:0]a,b;
reg cin;
wire [3:0]s,s0,s1,c0,c1;
wire cout;
carryselect
uut(.s(s),.s0(s0),.s1(s1),.c0(c0),.c1(c1),.cout(cout),.a(a),.b(b),.cin(ci
n));
initial begin
$monitor($time," a=%b b=%b cin=%b s=%b cout=%b ",a,b,cin,s,cout);
a = 4'b1100; b = 4'b0001; cin = 1'b0;#2;
a = 4'b1100; b = 4'b0001; cin = 1'b1;#2;
a = 4'b1001; b = 4'b0100; cin = 1'b0;#2;
a = 4'b1001; b = 4'b0100; cin = 1'b1;#2;
a = 4'b1111; b = 4'b0001; cin = 1'b0;#2;
a = 4'b1111; b = 4'b0001; cin = 1'b1;#2;
$finish;
end
endmodule
```

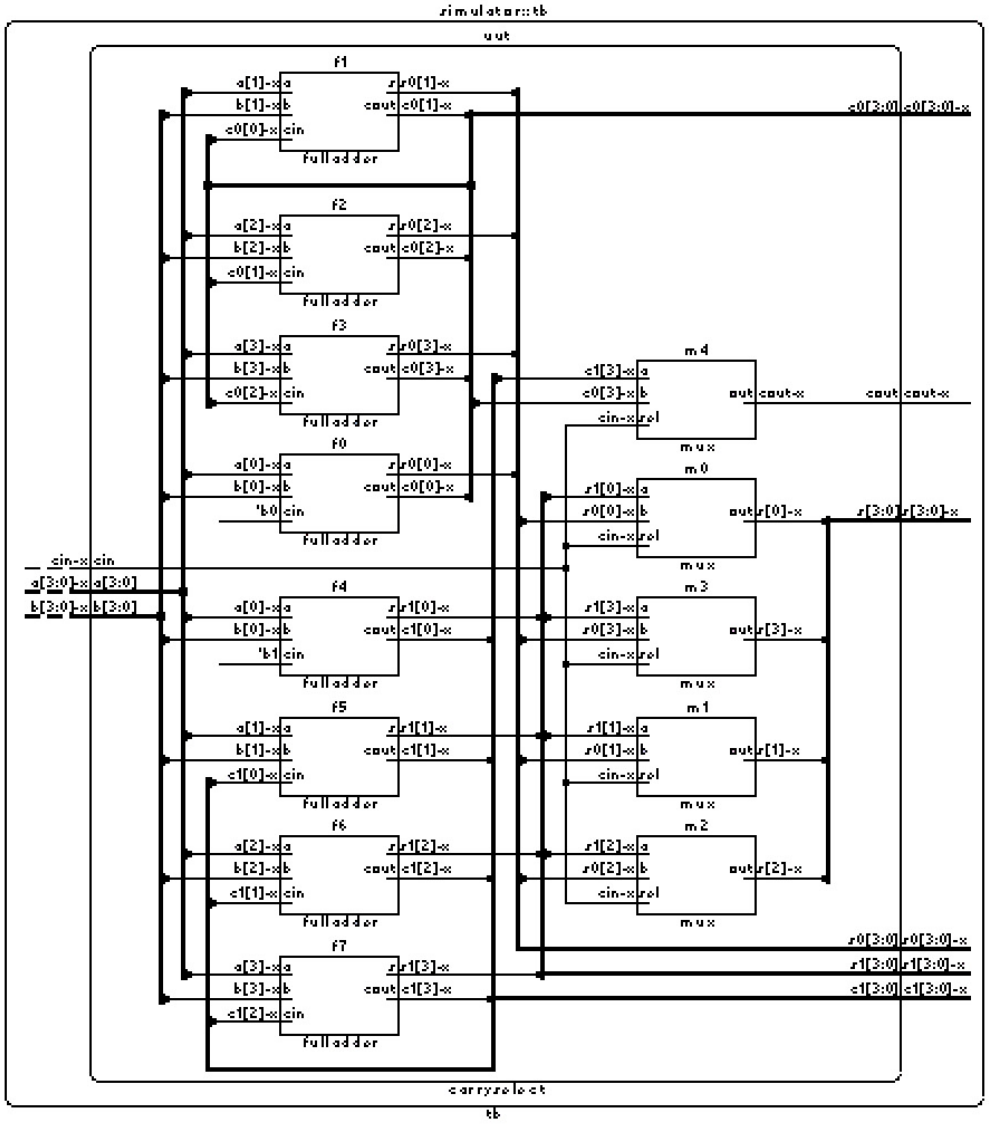
Waveform



Console

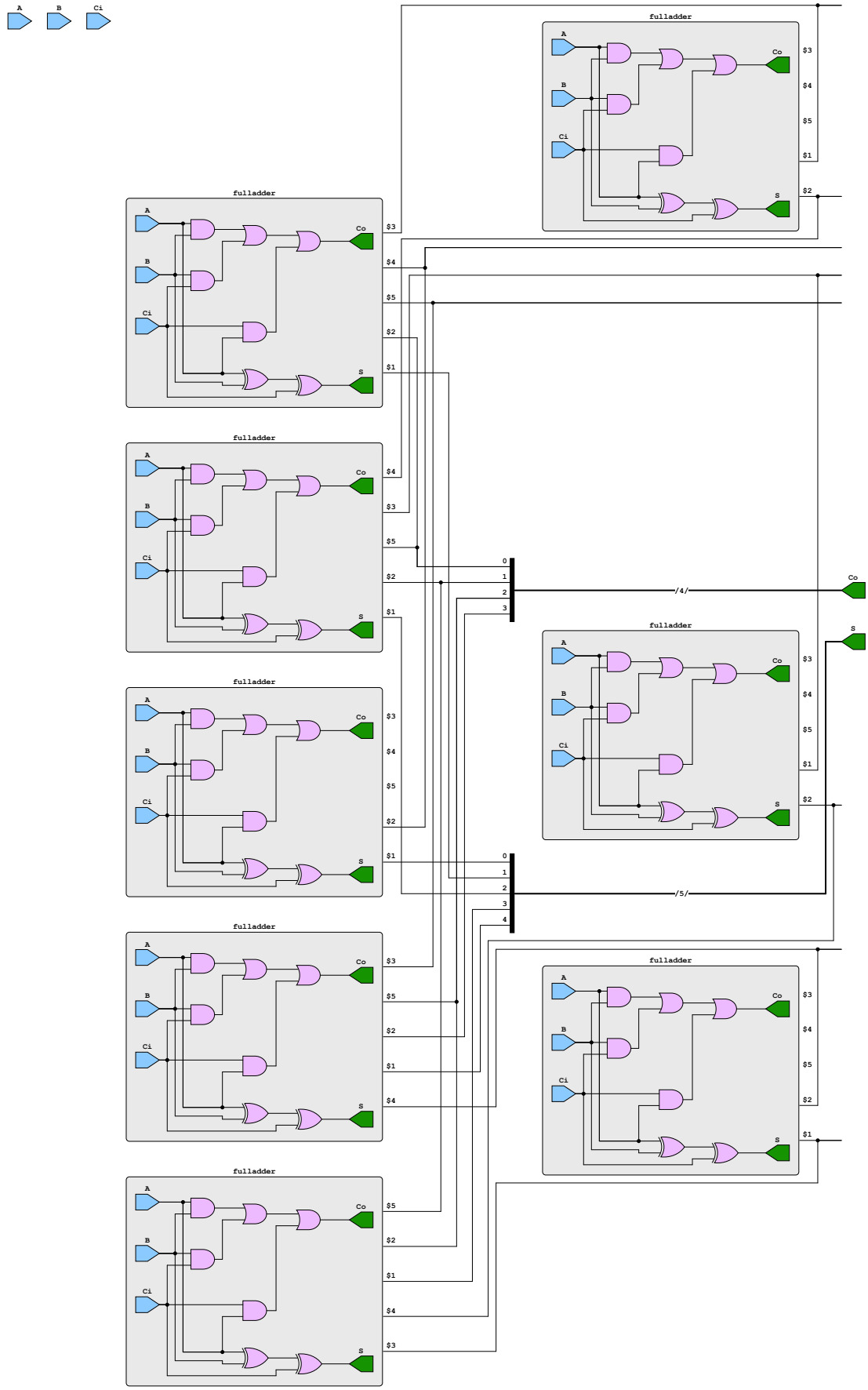
```
ncsim>
ncsim> source /home/install/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
                   0 a=1100 b=0001 cin=0 s=1101 cout=0
                   2 a=1100 b=0001 cin=1 s=1110 cout=0
                   4 a=1001 b=0100 cin=0 s=1101 cout=0
                   6 a=1001 b=0100 cin=1 s=1110 cout=0
                   8 a=1111 b=0001 cin=0 s=0000 cout=1
                  10 a=1111 b=0001 cin=1 s=0001 cout=1
Simulation complete via $finish(1) at time 12 NS + 0
./carryselect_tb.v:15 $finish;
ncsim>
```

Schematic



4. **Write structural Verilog code for 4-bit carry save adder and verify the design by simulation.**

Circuit



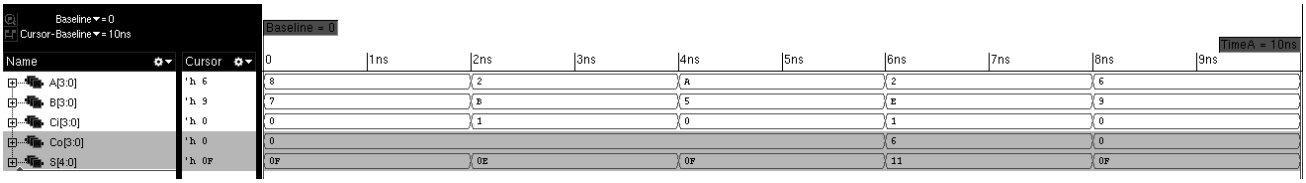
Source Code

```
module carrysave(output [4:0]S,output [3:0]Co,input [3:0]A,B,Ci);
wire [2:0]Si;
wire [3:0]Cio;
fulladder f0(S[0],Cio[0],A[0],B[0],Ci[0]);
fulladder f1(Si[0],Cio[1],A[1],B[1],Ci[1]);
fulladder f2(Si[1],Cio[2],A[2],B[2],Ci[2]);
fulladder f3(Si[2],Cio[3],A[3],B[3],Ci[3]);
fulladder f4(S[1],Co[0],Si[0],Cio[0],1'b0);
fulladder f5(S[2],Co[1],Si[1],Cio[1],Co[0]);
fulladder f6(S[3],Co[2],Si[2],Cio[2],Co[1]);
fulladder f7(S[4],Co[3],1'b0,Cio[3],Co[2]);
endmodule
module fulladder(output S,Co,input A,B,Ci);
xor sum(S,A,B,Ci);
and a1(c1,A,B);
and a2(c2,B,Ci);
and a3(c3,Ci,A);
or carry(Co,c1,c2,c3);
endmodule
```

Testbench

```
module tb();
wire [4:0]S;
wire [3:0]Co;
reg [3:0]A,B,Ci;
carrysave uut(.S(S),.Co(Co),.A(A),.B(B),.Ci(Ci));
initial begin
$monitor ($time,"a=%b , b=%b , cin=%b ,s=%b ,cout=%b ",A,B,Ci,S,Co);
A=4'b1000; B=4'b0111; Ci=4'b0000; #2
A=4'b0010; B=4'b1011; Ci=4'b0001; #2
A=4'b1010; B=4'b0101; Ci=4'b0000; #2
A=4'b0010; B=4'b1110; Ci=4'b0001; #2
A=4'b0110; B=4'b1001; Ci=4'b0000; #2
$finish;
end
endmodule
```

Waveform



Console

```
ncsim>
ncsim> run
database -open waves -into waves.shm -default
probe -create -shm tb.A tb.B tb.Ci tb.Co tb.S
run
Created default SHM database waves
ncsim> Created probe 1
ncsim>                    0a=1000 , b=0111 , cin=0000 ,s=01111 ,cout=0000
                  2a=0010 , b=1011 , cin=0001 ,s=01110 ,cout=0000
                  4a=1010 , b=0101 , cin=0000 ,s=01111 ,cout=0000
                  6a=0010 , b=1110 , cin=0001 ,s=10001 ,cout=0110
                  8a=0110 , b=1001 , cin=0000 ,s=01111 ,cout=0000
Simulation complete via $finish(1) at time 10 NS + 0
./carrysave_tb.v:13 $finish;
ncsim>
```

Schematic