CE221-L Digital Logic Design Lab

Lab report # 08

Submitted by: **Shayan Rizwan**

Registration Number: **2024585**

Submitted to: Engr. Irfanullah

Semester: 03

Faculty of Computer Science and Engineering

GIK Institute of Engineering Sciences and Technology

Task Statement:

Design a full adder using Behavioral modeling in Verilog, write test benches to verify its functionality, and simulate to visualize the waveforms and show its RTL.

Solution

CODE:

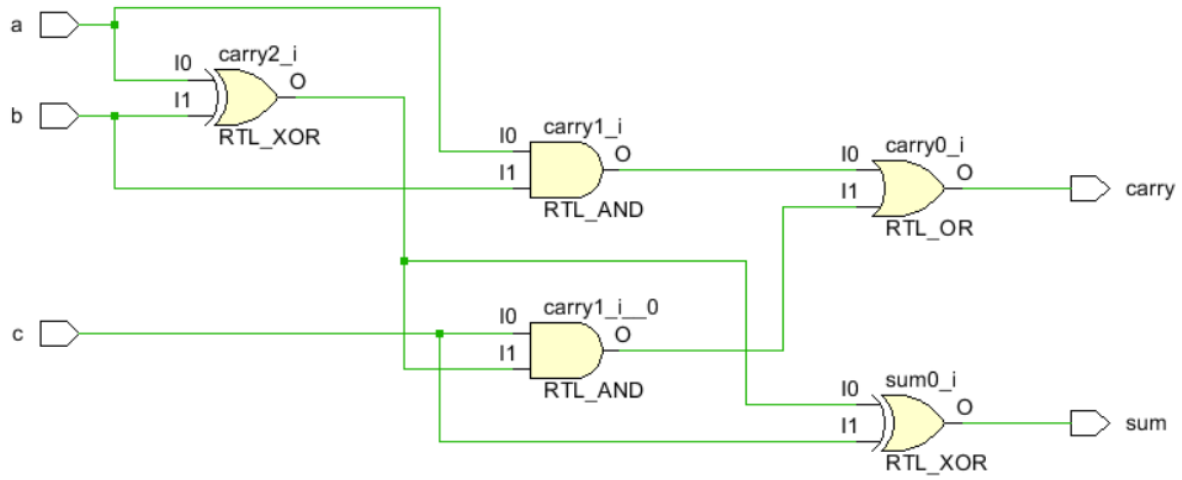
```
module
full_adder(a,b,c,carry,sum );

input a,b,c;

output sum,carry;

reg sum,carry;

always@( a or b or c)

begin

sum = a^b^c;

carry = (a&b) | c & (a^b);

end

endmodule
```

Test Bench:

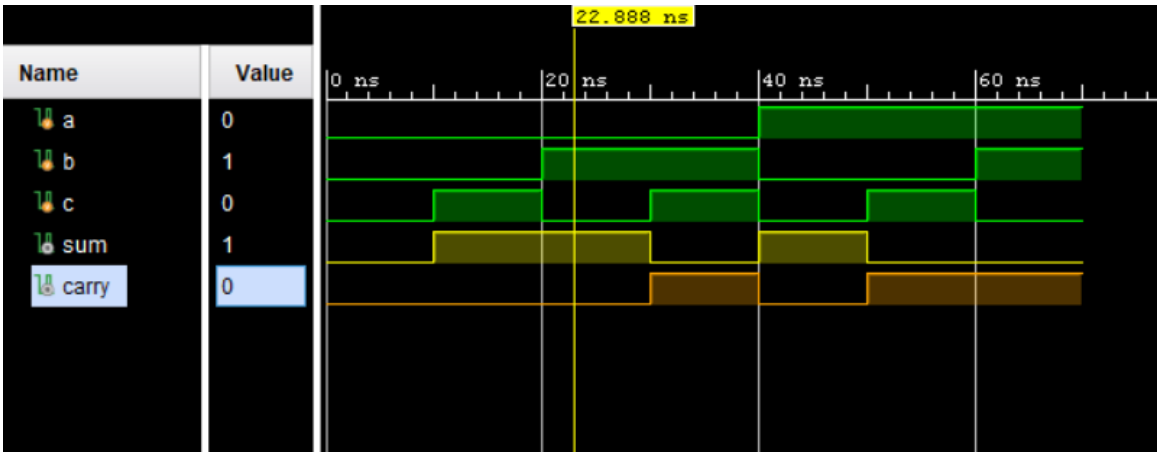
```
` module full_adder_tb();

// Inputs

reg a;

reg b;

reg c;

// Outputs

wire sum;

wire carry;

full_adder
uut(.sum(sum),.carry(carry),.a(a
),.b(b),.c(c));
```

```
initial
begin
    a = 0; b = 0; c=0;
#10 a = 0; b = 0; c=1;
#10 a = 0; b = 1; c=0;
#10 a = 0; b = 1; c=1;
#10 a = 1; b = 0; c=0;
#10 a = 1; b = 0; c=1;
#10 a = 1; b = 1; c=0;
#10 a = 1; b = 1; c=1;
 $finish;
end
endmodule
```

RTL:



Output:



Task Statement:

Design a 4x1 Multiplexer using Behavioral modeling in Verilog, write test benches to verify its functionality, and simulate to visualize the waveforms and show its RTL.

Solution

Code

```
module mux_4x1(y,i,s);
output y;
input [3:0]i;
input [1:0]s;
reg y;
// Behavioural
always@(i or s)
// Using case statement
case(s)
3'b00:y = i[0];
3'b01:y = i[1];
3'b10:y = i[2];
3'b11:y = i[3];
 default: y = 1'b0;
endcase
```

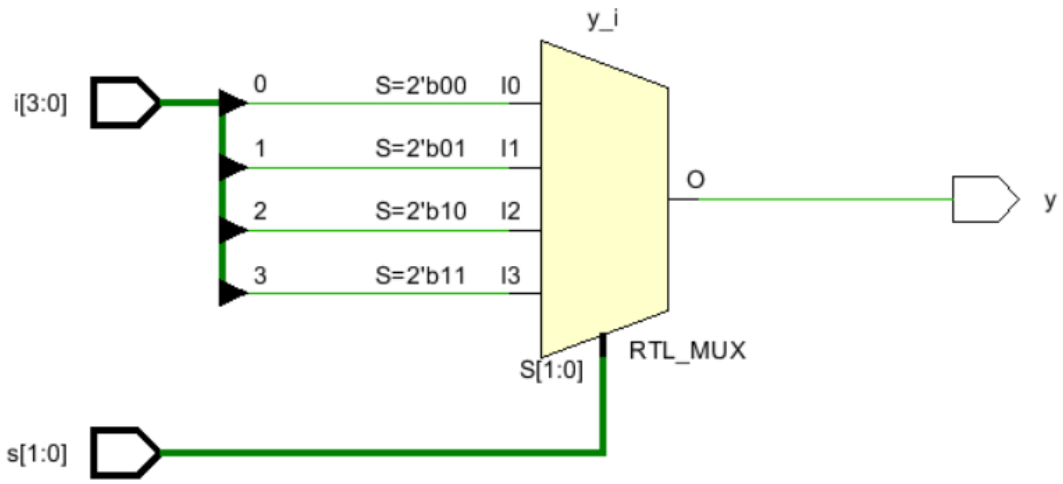
```
endmodule
```

Test Bench

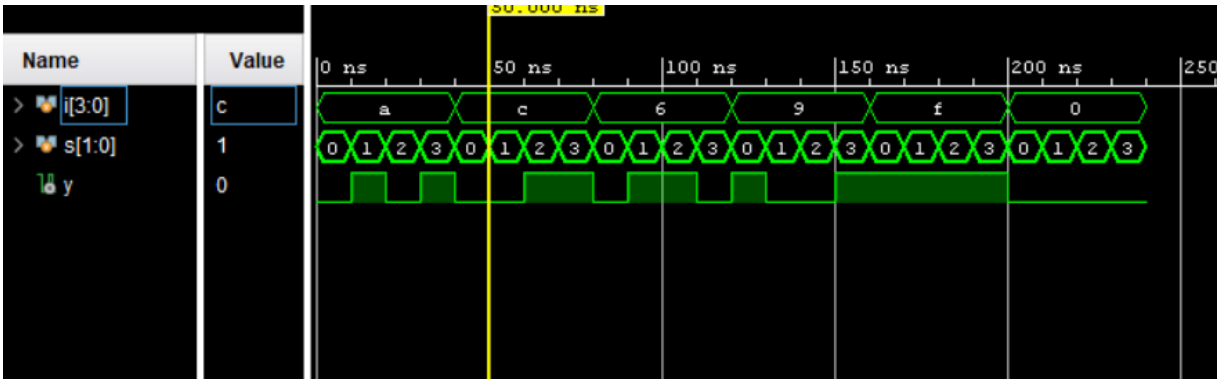
```
module mux_4x1_tb();
// Inputs
reg [3:0] i;
reg [1:0] s;
// Outputs
wire y;
// Instantiate the Unit Under Test (UUT)
mux_8x1 uut ( .y(y),.i(i),.s(s));
initial
begin
// Initialize Inputs

      i = 4'b1010;
      s = 2'b00; #10;
      s = 2'b01; #10;
      s = 2'b10; #10;
      s = 2'b11; #10;

     i= 4'b1100;
     s = 2'b00; #10;
     s = 2'b01; #10;
     s = 2'b10; #10;
     s = 2'b11; #10;

     i = 4'b0110;
     s = 2'b00; #10;
     s = 2'b01; #10;
     s = 2'b10; #10;
     s = 2'b11; #10;

     i = 4'b1001;
     s = 2'b00; #10;
     s = 2'b01; #10;
     s = 2'b10; #10;
```

```
        s = 2'b11; #10;

        i = 4'b1111;
        s = 2'b00; #10;
        s = 2'b01; #10;
        s = 2'b10; #10;
        s = 2'b11; #10;

        i = 4'b0000;
        s = 2'b00; #10;
        s = 2'b01; #10;
        s = 2'b10; #10;
        s = 2'b11; #10;
        $finish;
end
endmodule
```

RTL



Output



Task Statement:

Design a 3x8 Decoder using Behavioral modeling in Verilog, write test benches to verify its functionality, and simulate to visualize the waveforms and show its RTL.

Solution

CODE:

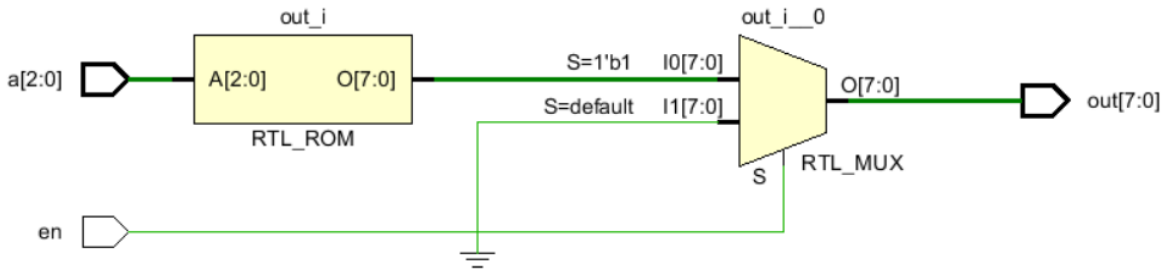
```
module decoder_3x8(input [2:0] a,output reg [7:0] out,input en);
always @(*)
begin
if (en)
  case(a)
  3'b000: out = 8'b00000001;
  3'b001: out = 8'b00000010;
  3'b010: out = 8'b00000100;
  3'b011: out = 8'b00001000;
  3'b100: out = 8'b00010000;
  3'b101: out = 8'b00100000;
  3'b110: out = 8'b01000000;
  3'b111: out = 8'b10000000;
  default: out = 8'b00000000;
  endcase
else
   out = 8'b00000000;
```

```
end
endmodule
```

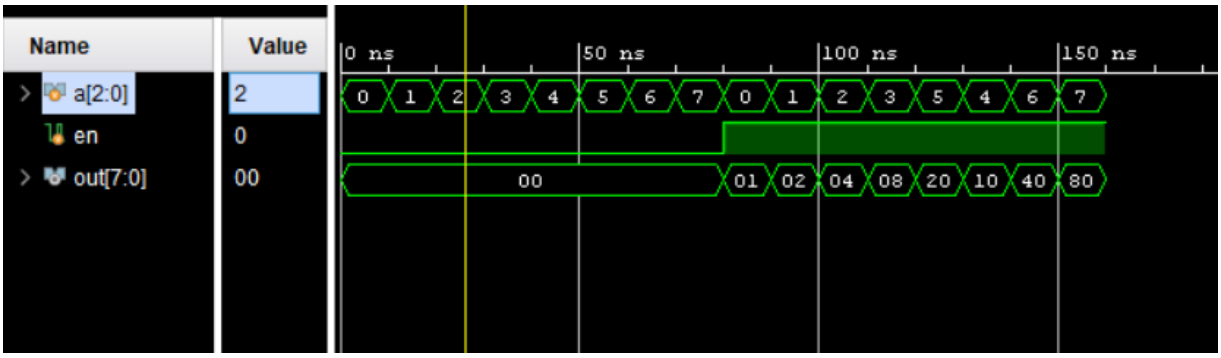
Test Bench:

```
module deocoder_3x8_tb();
reg [2:0] a;
reg en;
wire [7:0] out;
decoder_3x8 uut(.a(a),.en(en),.out(out));
initial
begin
$monitor($time, " a=%b, en=%b, out=%b", a, en, out);
en = 0;
a = 3'b000; #10;
a = 3'b001; #10;
a = 3'b010; #10;
a = 3'b011; #10;
a = 3'b100; #10;
a = 3'b101; #10;
a = 3'b110; #10;
a = 3'b111; #10;

en = 1;
a = 3'b000; #10;
a = 3'b001; #10;
a = 3'b010; #10;
      a = 3'b011; #10;
      a = 3'b101; #10;
      a = 3'b100; #10;
      a = 3'b110; #10;
      a = 3'b111; #10;
      $finish;
   end
endmodule
```

RTL:



Output:



Task Statement:

Design an ALU using Behavioral modeling in Verilog, write test benches to verify its functionality, and simulate to visualize the waveforms and show its RTL.

Solution

Code:

```
module ALU(input [8:0] a,input [8:0] b,input [3:0] sel,output reg [8:0] c);
always @(*)
begin
case(sel)
```

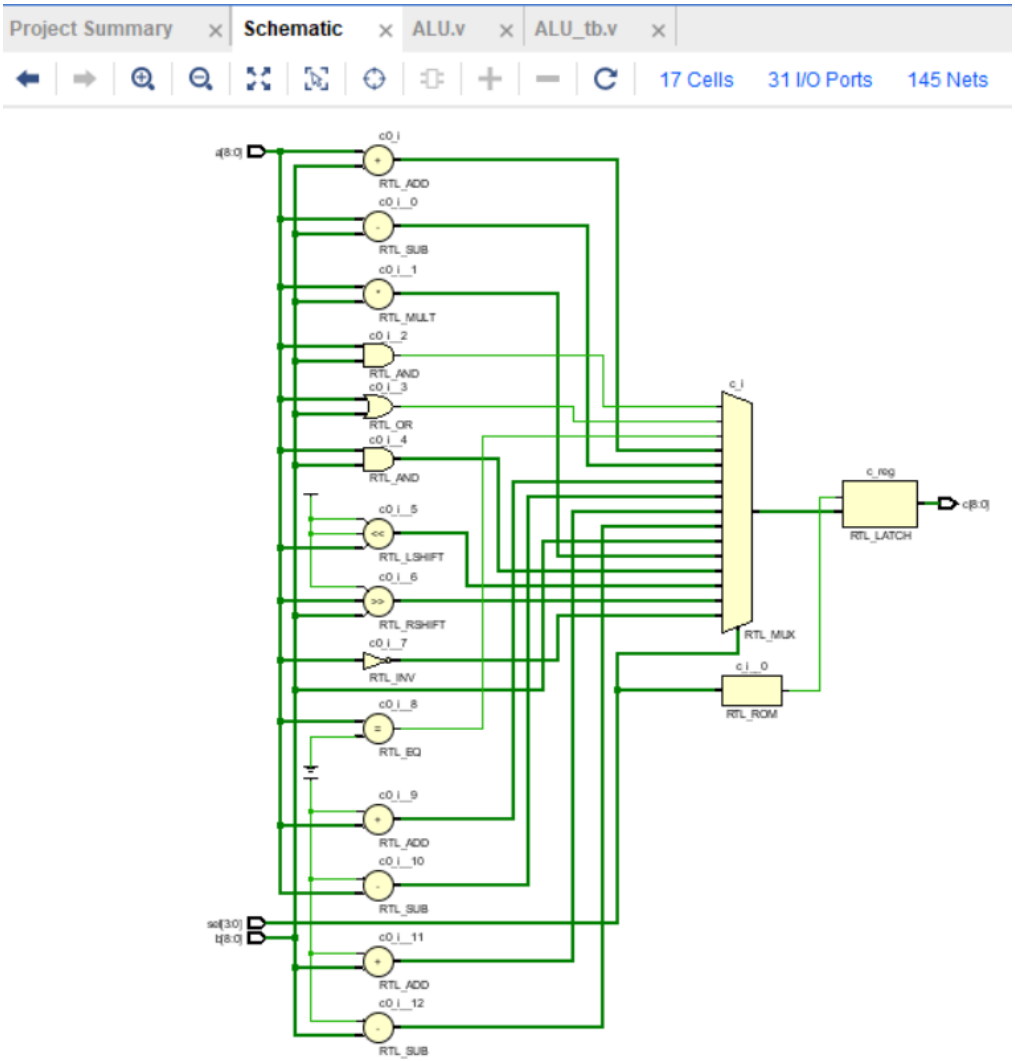
```
4'b0000: c = a+b;
4'b0001: c = a-b;
4'b0010: c = a*b;
4'b0011: c = a&&b;
4'b0100: c = a||b;
4'b0101: c = a&b;
4'b0100: c = a|b;
4'b0111: c = a<<1;
4'b1000: c = a>>b;
4'b1001: c = ~a;
4'b1010: c = !a;
4'b1011: c = a+1;
4'b1100: c = a-1;
4'b1101: c = b+1;
4'b1110: c = b-1;
4'b1111: c = b;
endcase
end
endmodule
```

Test Bench:

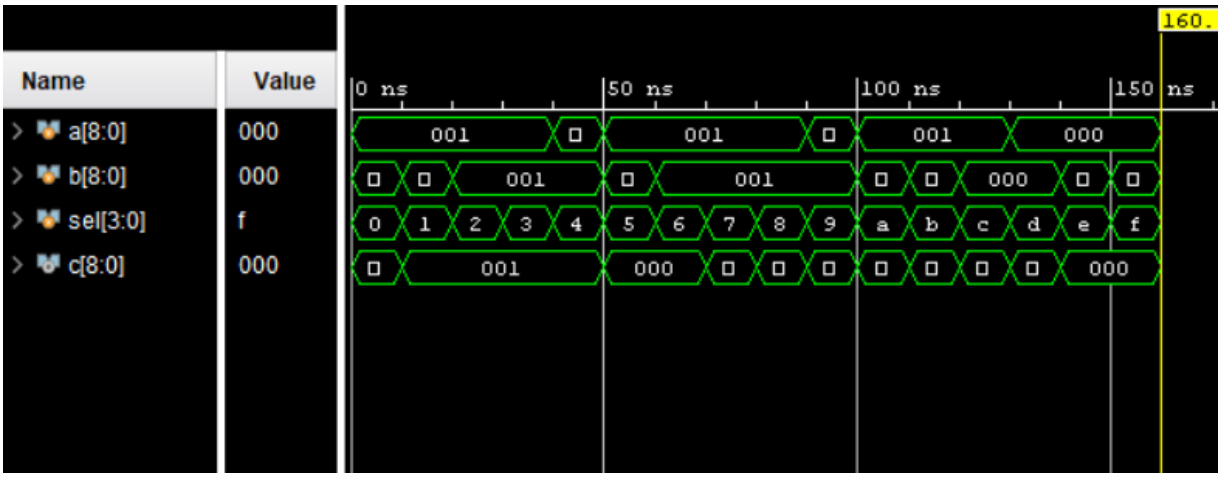
```
module ALU_tb();
reg [8:0] a;
reg [8:0] b;
reg [3:0] sel;
wire [8:0] c;
ALU uut (.c(c),.a(a),.b(b),.sel(sel));
initial
begin
   a = 1;b = 1; sel = 4'b0000;
   #10    a = 1;b = 0; sel = 4'b0001;
   #10    a = 1;b = 1; sel = 4'b0010;
   #10    a = 1;b = 1; sel = 4'b0011;
   #10    a = 0;b = 1; sel = 4'b0100;
   #10    a = 1;b = 0; sel = 4'b0101;
   #10    a = 1;b = 1; sel = 4'b0110;
```

```
    #10    a = 1;b = 1; sel = 4'b0111;
    #10    a = 1;b = 1; sel = 4'b1000;
    #10    a = 0;b = 1; sel = 4'b1001;
    #10    a = 1;b = 0; sel = 4'b1010;
    #10    a = 1;b = 1; sel = 4'b1011;
    #10    a = 1;b = 0; sel = 4'b1100;
    #10    a = 0;b = 0; sel = 4'b1101;
    #10    a = 0;b = 1; sel = 4'b1110;
    #10    a = 0;b = 0; sel = 4'b1111;
    #10
    $finish;
    end
endmodule
```

RTL


Project Summary
Schematic
ALU.v
ALU_tb.v
17 Cells
31 I/O Ports
145 Nets
RTL_ADD
RTL_SUB
RTL_MULT
RTL_AND
RTL_OR
RTL_AND
RTL_LSHIFT
RTL_RSHIFT
RTL_INV
RTL_EQ
RTL_ADD
RTL_SUB
RTL_ADD
RTL_SUB
RTL_MUX
RTL_ROM
RTL_LATCH

Output


Name
Value
a[8:0]
000
b[8:0]
000
sel[3:0]
f
c[8:0]
000
0 ns
50 ns
100 ns
150 ns
160.
001
001
001
000
001
001
000
0
1
2
3
4
5
6
7
8
9
a
b
c
d
e
f
001
000
000