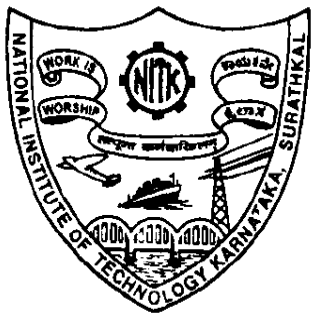
EC806 DSD using FPGA

LAB - 3



Report Submission

By

Inbasekaran Perumal

201EC226

Pranav Koundinya

201EC247

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

SRINIVASNAGAR 575025 KARNATAKA INDIA

27 September 2023

1. Behavioral verilog code for (a) 2 to 4 decoder (b) M to N decoder

Verilog code

a) 2 to 4 decoder

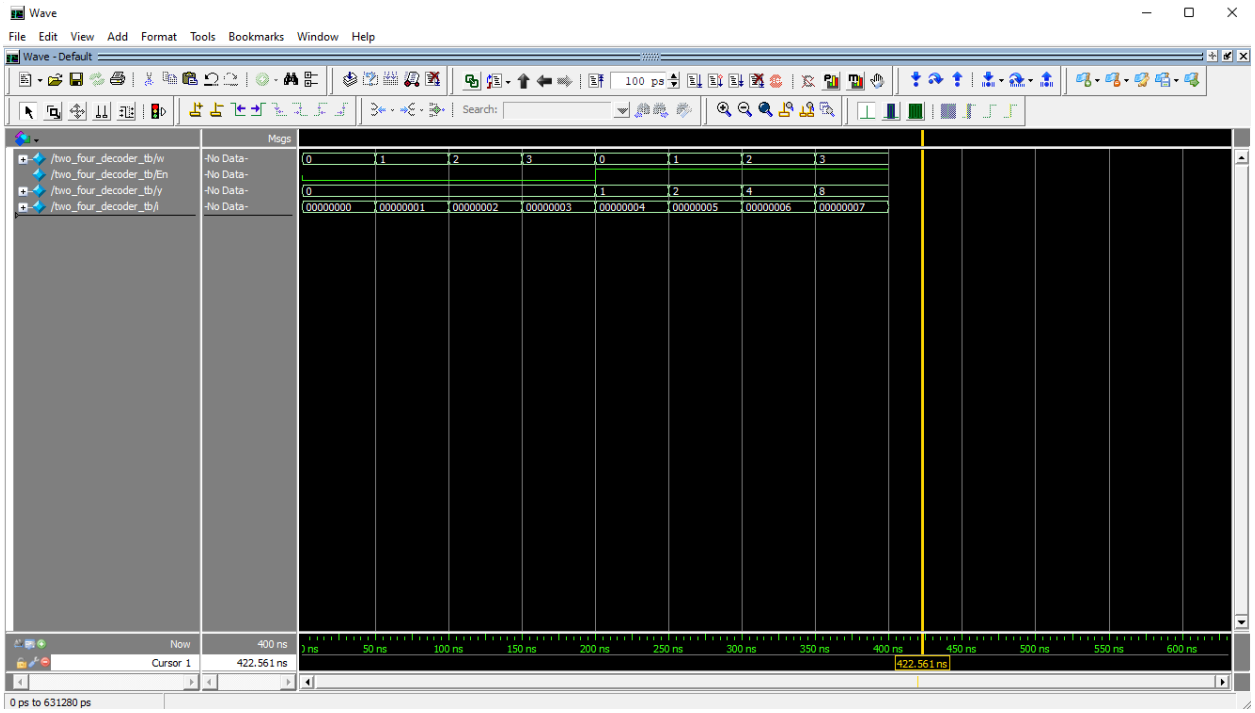
```
module two_four_decoder(
input [1:0]w,
input En,
output reg [3:0]y
);

always @*
begin
if(~En)
y = 4'b0000;
else
y = 1 << w;
end
endmodule


`timescale 1ns/1ps

module two_four_decoder_tb();
reg [1:0]w;
reg En;
wire [3:0]y;
integer i;
two_four_decoder dut (w, En, y);
initial
begin
for(i = 0; i < 8; i = i + 1)
begin
{En, w} <= i;
#50;
end
$stop;
end
endmodule
```

b) M to N decoder

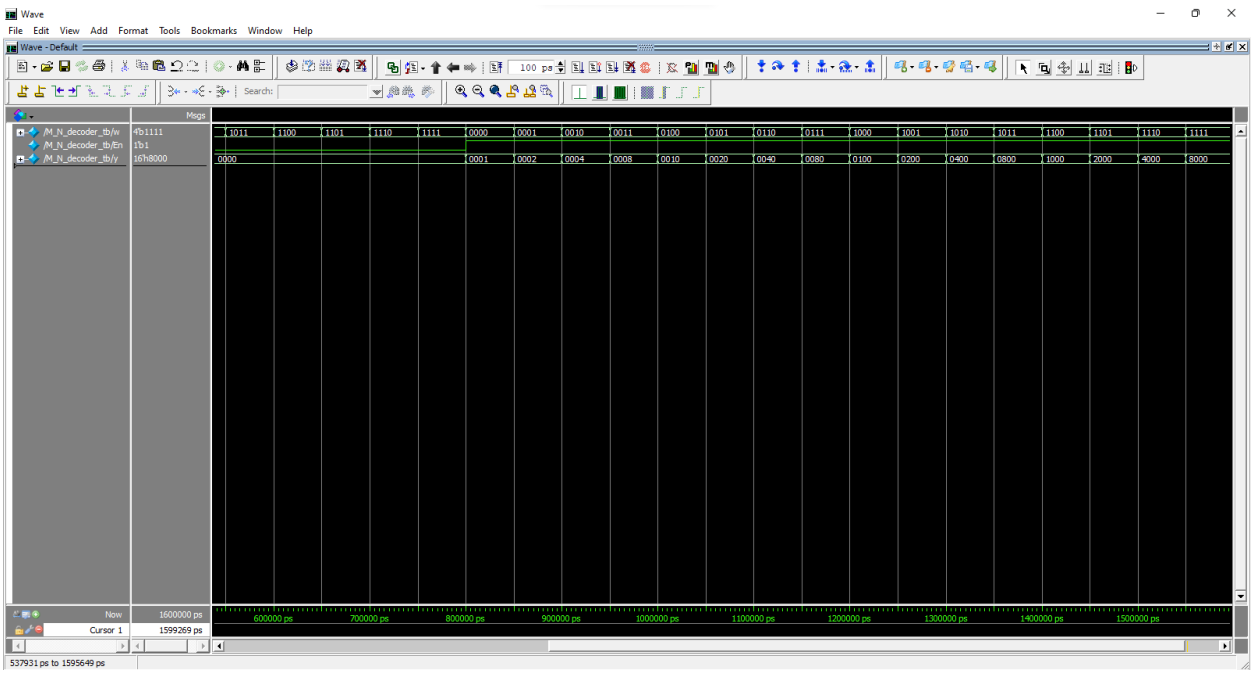
```
module M_N_decoder(
w, En, y
);
parameter M = 4;
parameter N = 2**M;
input [M-1:0]w;
input En;
output reg [N-1:0]y;
always @*
begin
if(~En)
y = {N{1'b0}};
else
y = 1 << w;
end
endmodule

`timescale 1ns/1ps

module M_N_decoder_tb();
reg [3:0]w;
reg En;
wire [15:0]y;
integer i;
M_N_decoder dut (w, En, y);
initial
begin
for(i = 0; i < 32; i = i + 1)
begin
{En, w} <= i;
#50;
end
$stop;
end
endmodule
```

Simulation results

a) 2 to 4 decoder

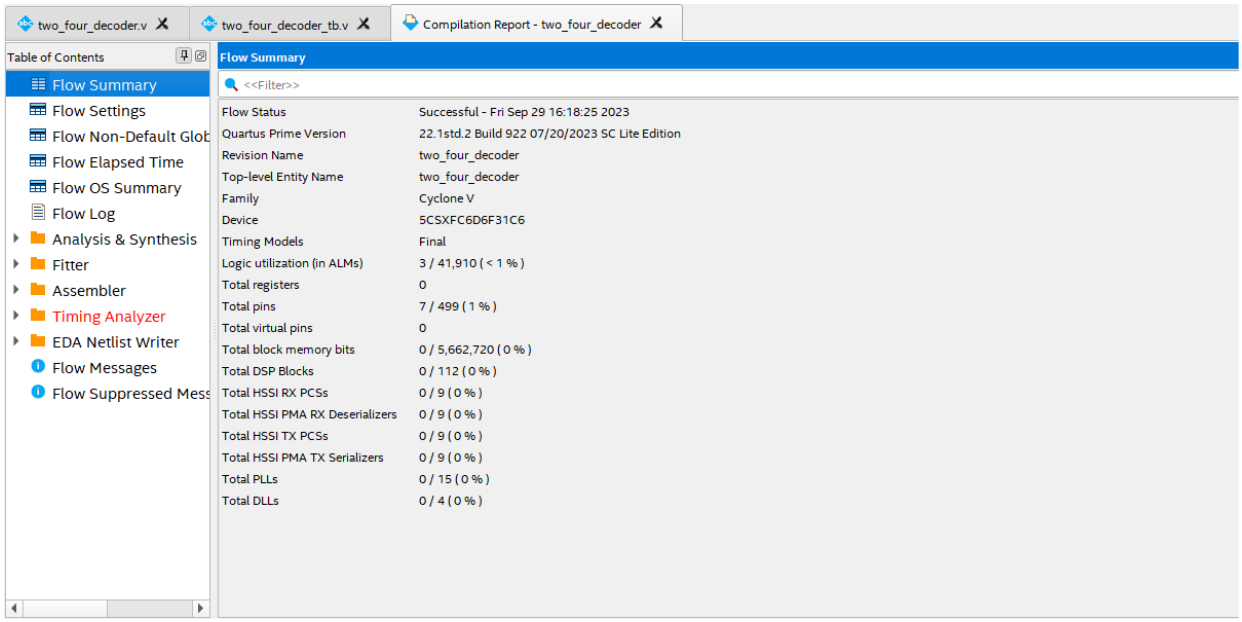


b) M to N decoder

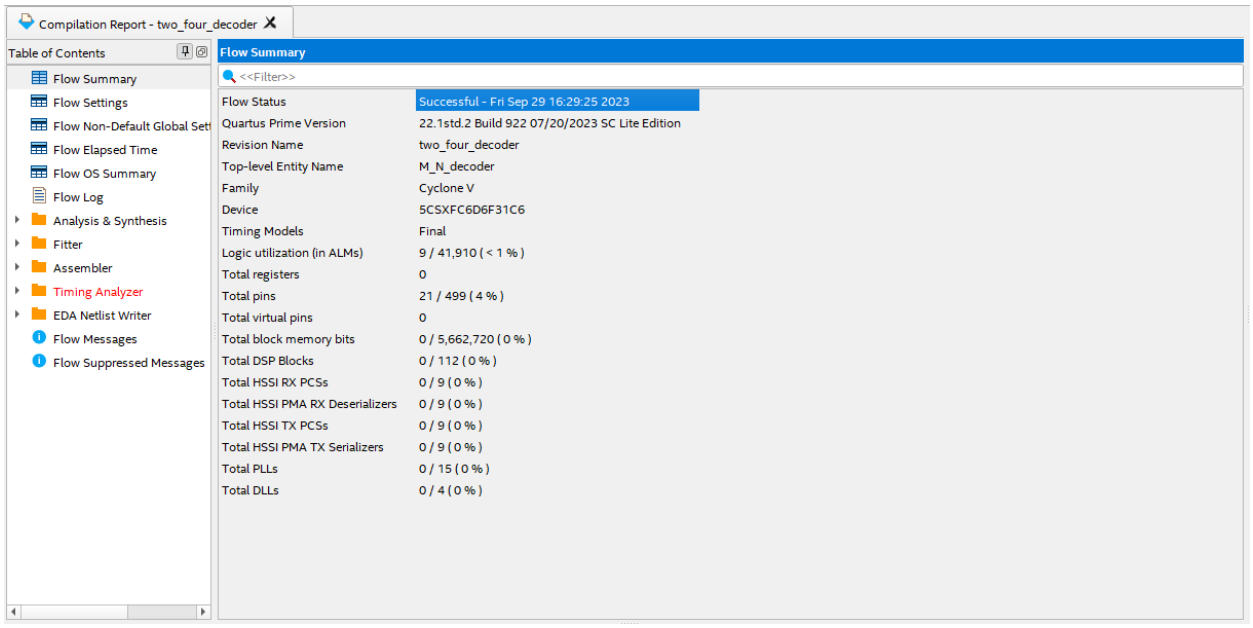


Resource utilization

a) 2 to 4 decoder



b) M to N decoder



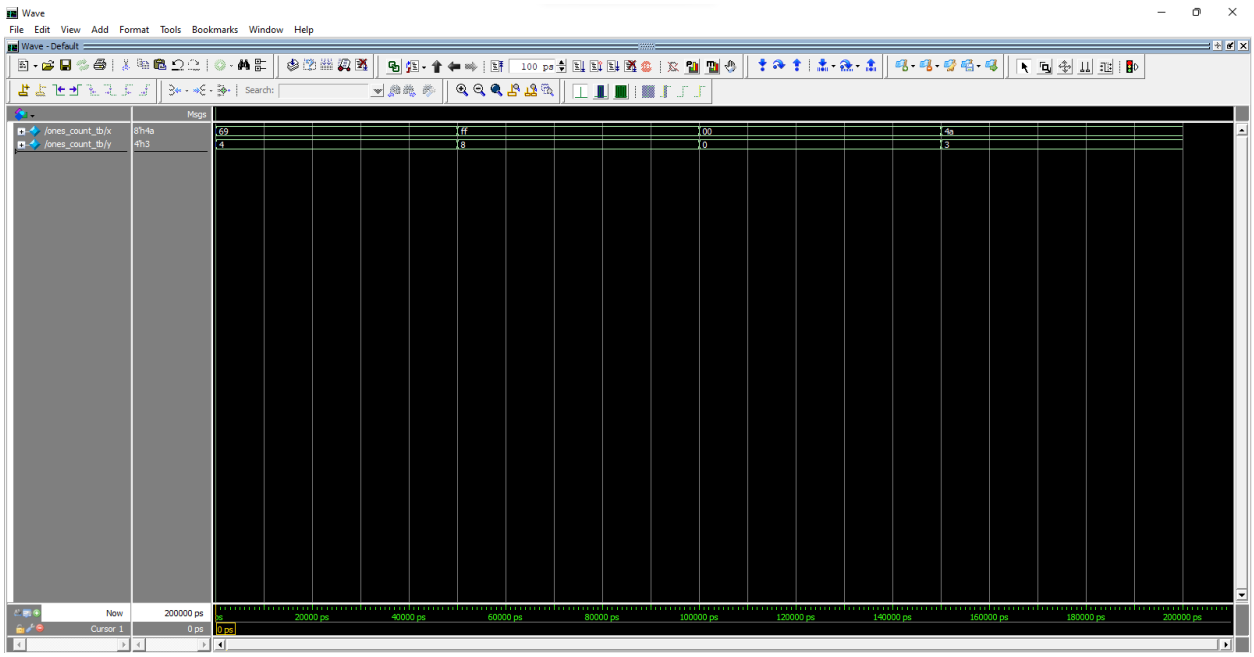
2. Verilog code to Count the number of 1s in a 8 bit number

Verilog code

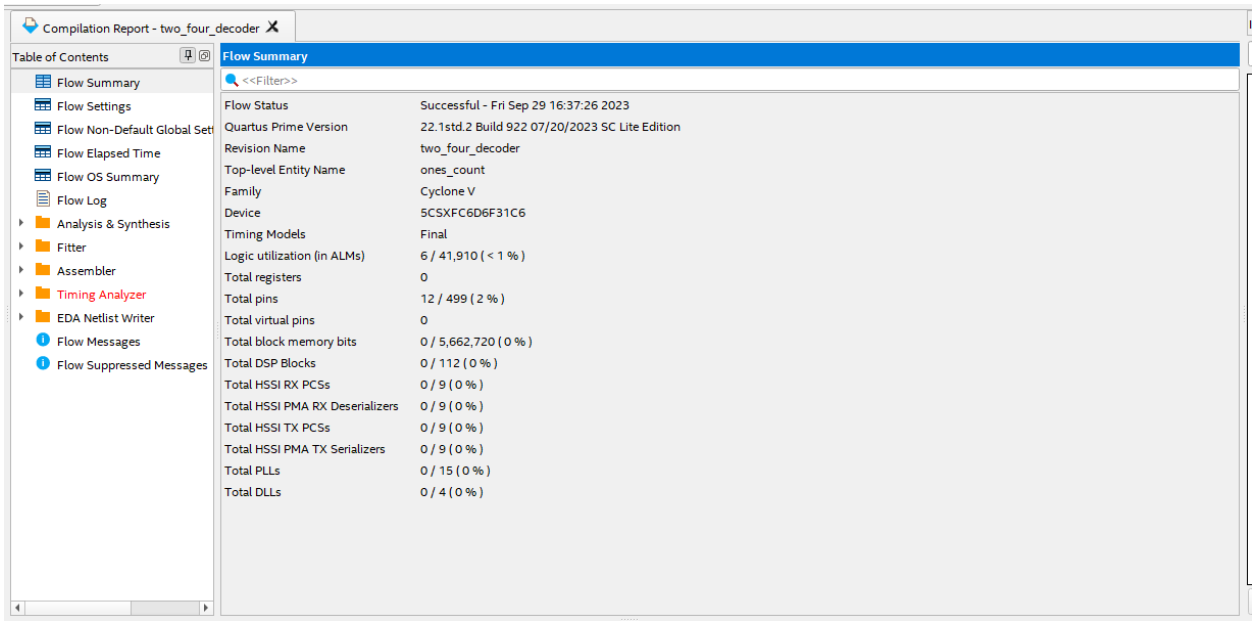
```
module ones_count(
```

```
input [7:0]x,
output reg [3:0]y
);
integer i;
always @(x)
begin
y = 'b0;
for(i = 0; i < 8;i = i + 1)
y = y + x[i];
end
endmodule


`timescale 1ns/1ps

module ones_count_tb();

reg [7:0]x;
wire [3:0]y;
ones_count dut(x, y);
initial
begin
x <= 8'h69; #50
x <= 8'hff; #50
x <= 8'h00; #50
x <= 8'h4a; #50;
$stop;
end
endmodule
```

Simulation results



Resource utilization



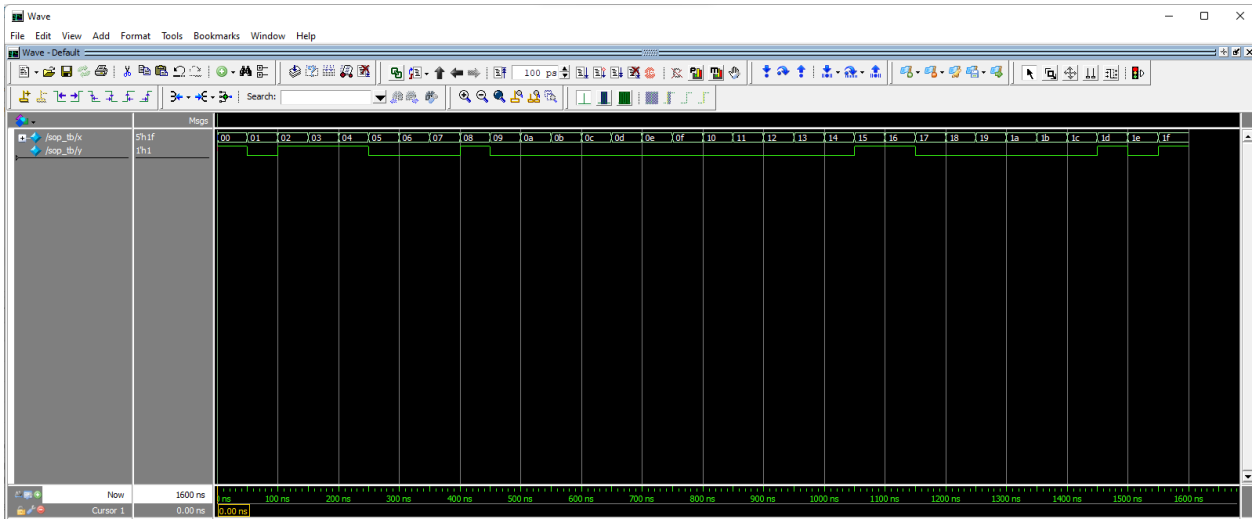
3. Verilog code to Implement F(v,w,x,y,z) = (0, 2, 3, 4, 8, 21, 22, 29, 31)

Verilog code

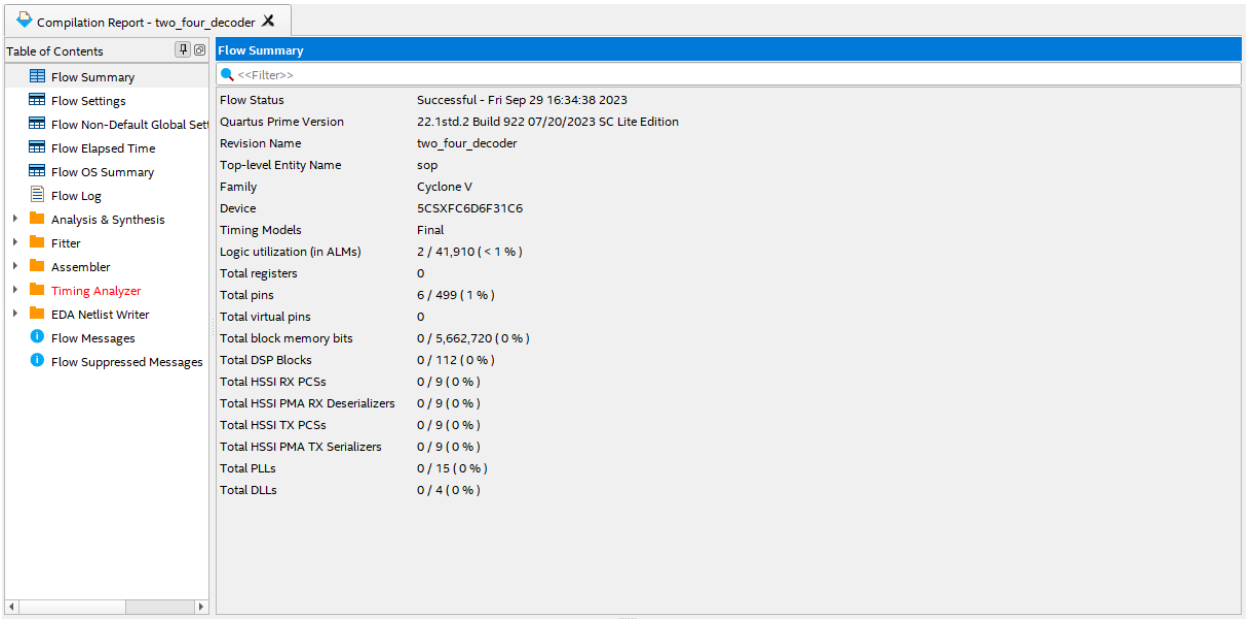
```
module sop(input [4:0]x, output reg f);
always @(*) begin
```

```
if(x == 0 || x == 2 || x == 3 || x == 4 || x == 8 || x == 21 || x == 22 ||
x == 29 || x == 31)
f = 1;
else
f = 0;
end
endmodule

`timescale 1ns/1ps
module sop_tb();
reg [4:0]x;
wire y;
integer i = 0;
sop dut (x,y);
initial
begin
for(i = 0; i < 32; i = i + 1)
begin
x <= i;
#50;
end
//$stop;
end
Endmodule
```

Simulation results



Resource utilization



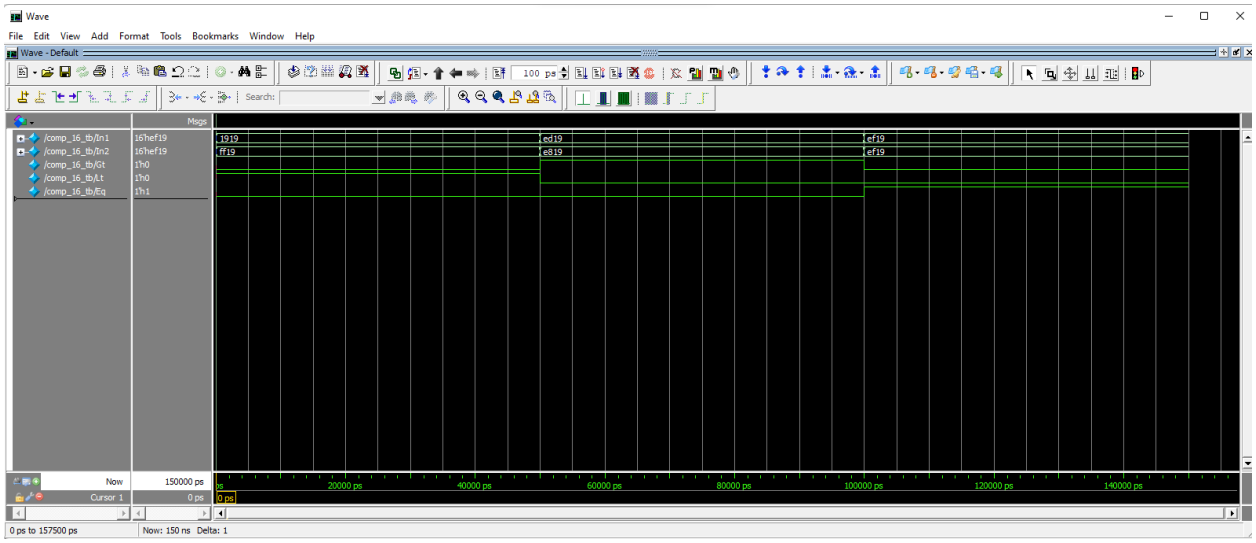
4. Verilog code to model a 16 bit magnitude comparator

Verilog code

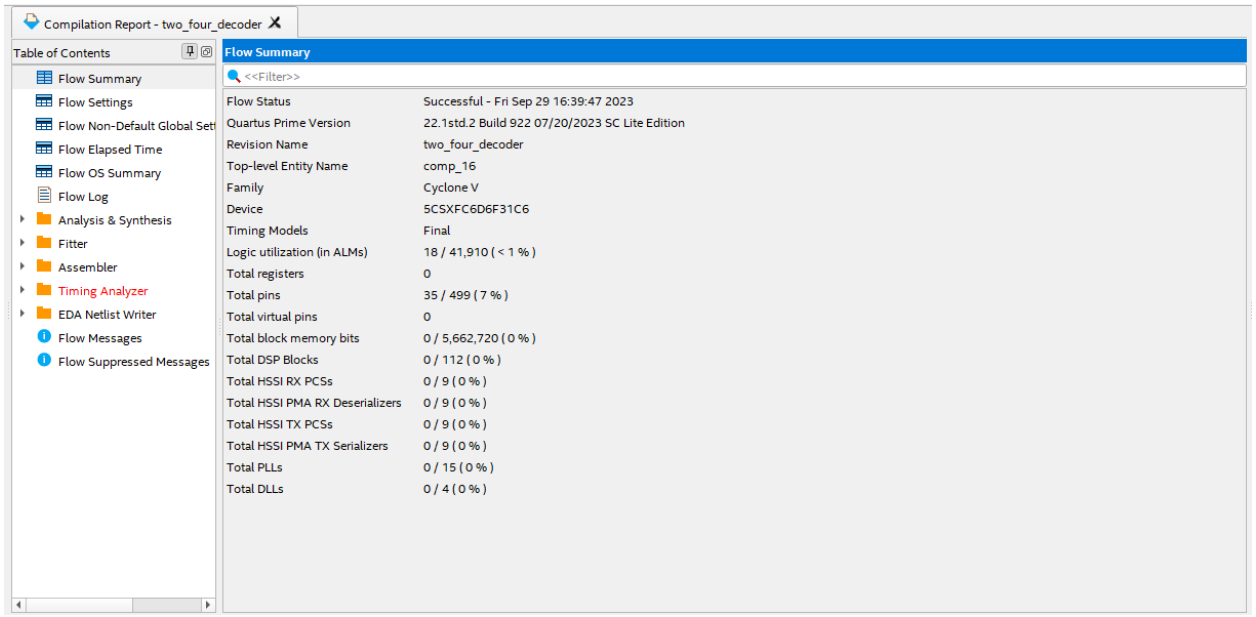
```
module comp_16(In1, In2, Gt,Eq,Lt);
input [15:0] In1, In2;
output Gt,Lt,Eq;
assign Gt = (In1 > In2) ? 1'b1 : 1'b0;
assign Lt = (In1 < In2) ? 1'b1 : 1'b0;
assign Eq = (In1 == In2) ? 1'b1 : 1'b0;
endmodule

`timescale 1ns/1ps
module comp_16_tb();
reg [15:0] In1, In2;
wire Gt,Lt,Eq;
comp_16 dut(In1,In2,Gt,Eq,Lt);
initial
begin
In1 = 16'b0001_1001_0001_1001;
In2 = 16'b1111_1111_0001_1001;
#50
In2 = 16'b1110_1000_0001_1001;
```

```
In1 = 16'b1110_1101_0001_1001;
#50
In1 = 16'b1110_1111_0001_1001;
In2 = 16'b1110_1111_0001_1001;
#50;
end
endmodule
```

Simulation results



Resource utilization

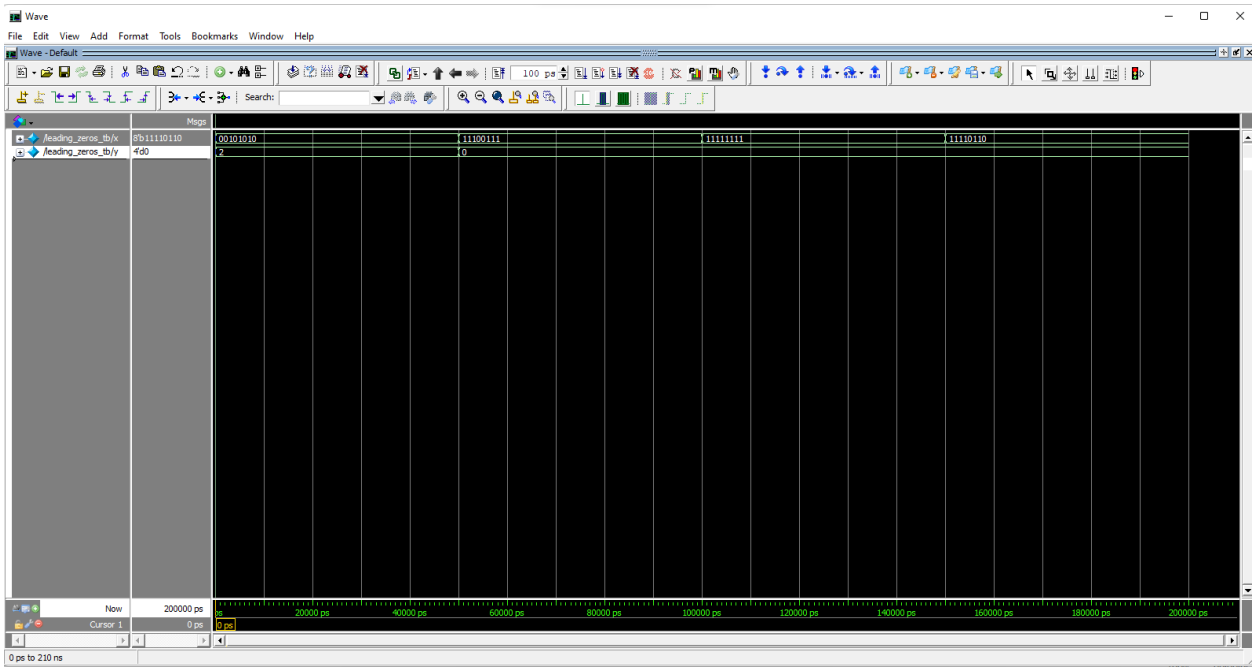


5. Verilog code to Count the number of leading 0s in an 8 bit number

Verilog code

```
module leading_zeros(
input [7:0]x,
output reg [3:0]y
);
always @(x)
begin
casex(x)
8'b00000000 : y = 4'b1000;
8'b00000001 : y = 4'b0111;
8'b0000001x : y = 4'b0110;
8'b000001xx : y = 4'b0101;
8'b00001xxx : y = 4'b0100;
8'b0001xxxx : y = 4'b0011;
8'b001xxxxx : y = 4'b0010;
8'b01xxxxxx : y = 4'b0001;
8'b1xxxxxxx : y = 4'b0000;
endcase
end
endmodule


`timescale 1ns/1ps

module leading_zeros_tb();
reg [7:0]x;
wire [3:0]y;
leading_zeros dut (x, y);
initial
begin
x = 8'b0010_1010; #50
x = 8'b1110_0111; #50
x = 8'b1111_1111; #50
x = 8'b1111_0110; #50;
end
endmodule
```

Simulation results



Resource utilization