22bec121

Experiment – 5 Date:-15/2/2024

Lab Work

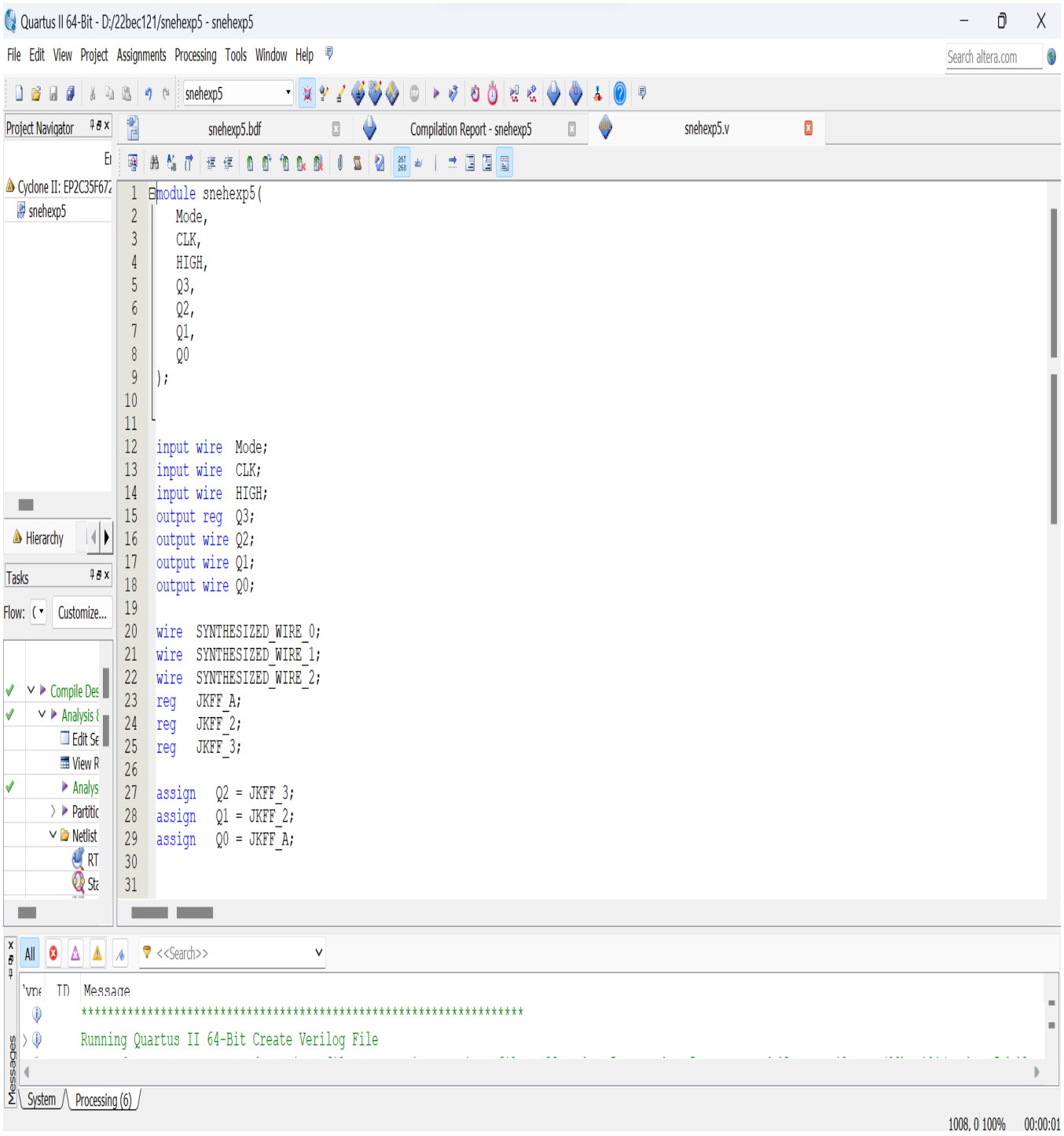
Q1 .Post Lab Exercises

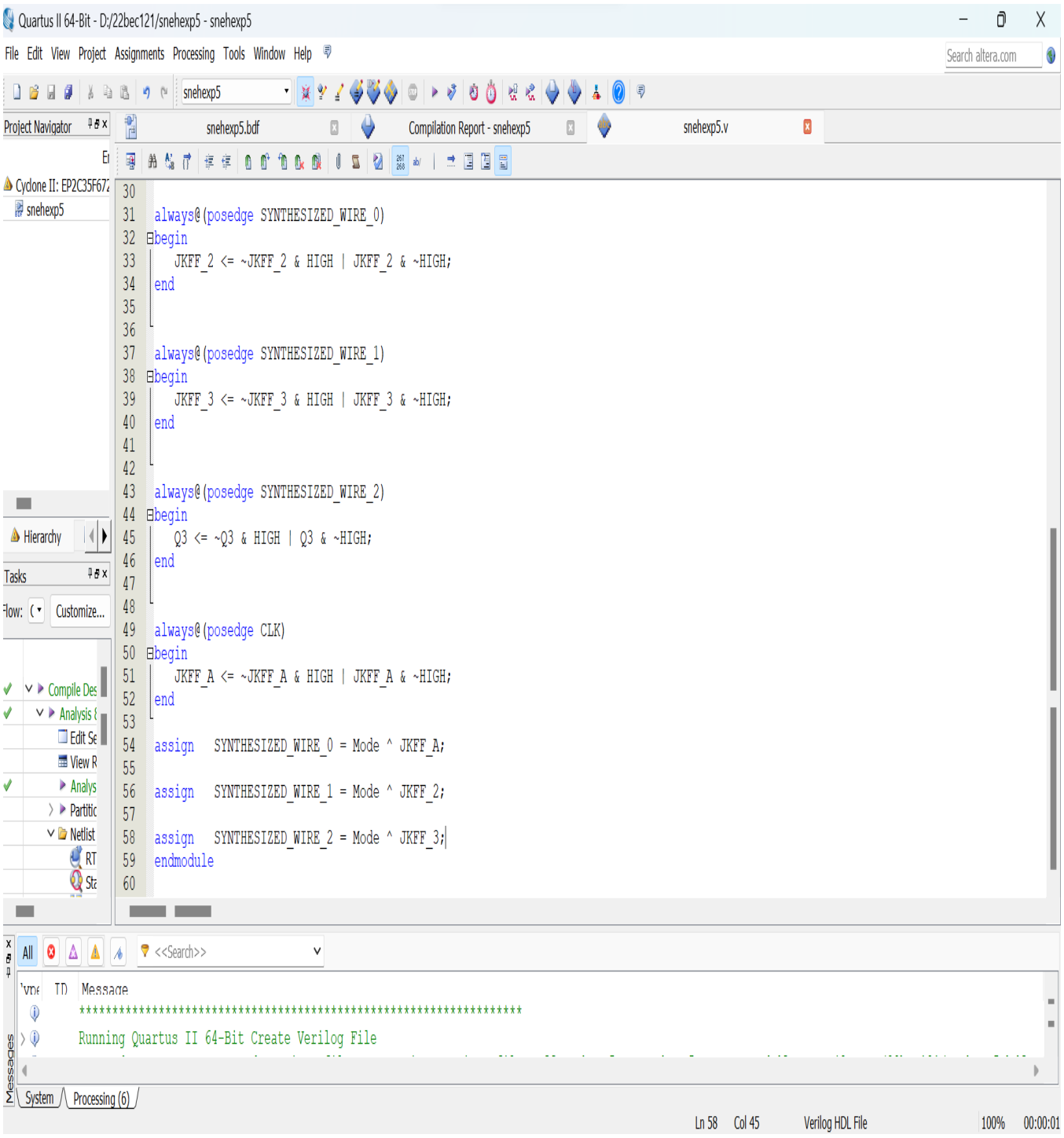
Code (Generated)

```
module snehexp5(
	Mode,
	CLK,
	HIGH,
	Q3,
	Q2,
	Q1,
	Q0
);


input wire	Mode;
input wire	CLK;
input wire	HIGH;
output reg	Q3;
output wire	Q2;
output wire	Q1;
output wire	Q0;

wire	SYNTHESIZED_WIRE_0;
wire	SYNTHESIZED_WIRE_1;
```

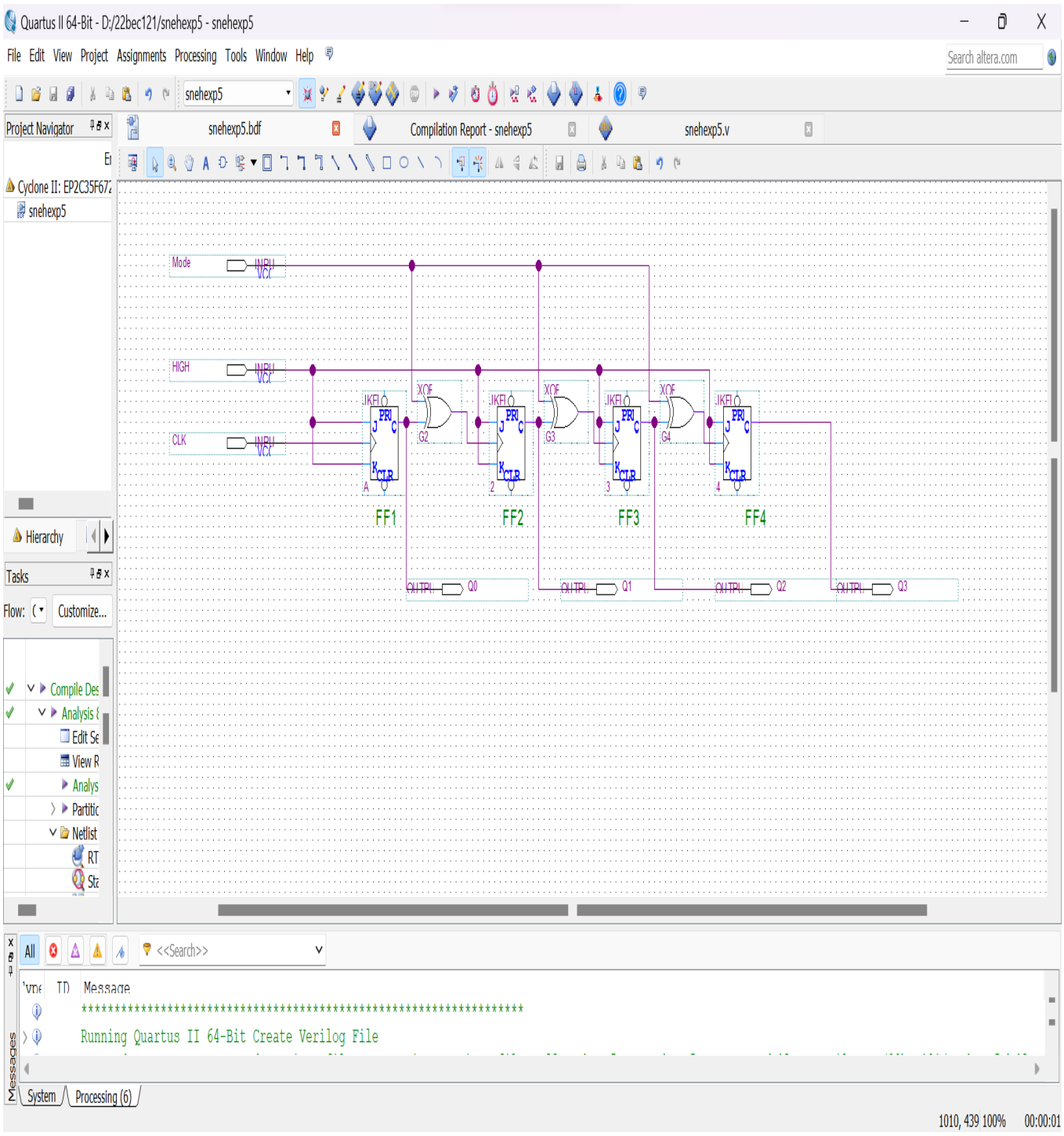
```
wire	SYNTHESIZED_WIRE_2;
reg	JKFF_A;
reg	JKFF_2;
reg	JKFF_3;

assign	Q2 = JKFF_3;
assign	Q1 = JKFF_2;
assign	Q0 = JKFF_A;

always@(posedge SYNTHESIZED_WIRE_0)
begin
	JKFF_2 <= ~JKFF_2 & HIGH | JKFF_2 & ~HIGH;
end


always@(posedge SYNTHESIZED_WIRE_1)
begin
	JKFF_3 <= ~JKFF_3 & HIGH | JKFF_3 & ~HIGH;
end


always@(posedge SYNTHESIZED_WIRE_2)
begin
	Q3 <= ~Q3 & HIGH | Q3 & ~HIGH;
end


always@(posedge CLK)
```

```
begin
        JKFF_A <= ~JKFF_A & HIGH | JKFF_A & ~HIGH;
end

assign   SYNTHESIZED_WIRE_0 = Mode ^ JKFF_A;

assign   SYNTHESIZED_WIRE_1 = Mode ^ JKFF_2;

assign   SYNTHESIZED_WIRE_2 = Mode ^ JKFF_3;
endmodule
```

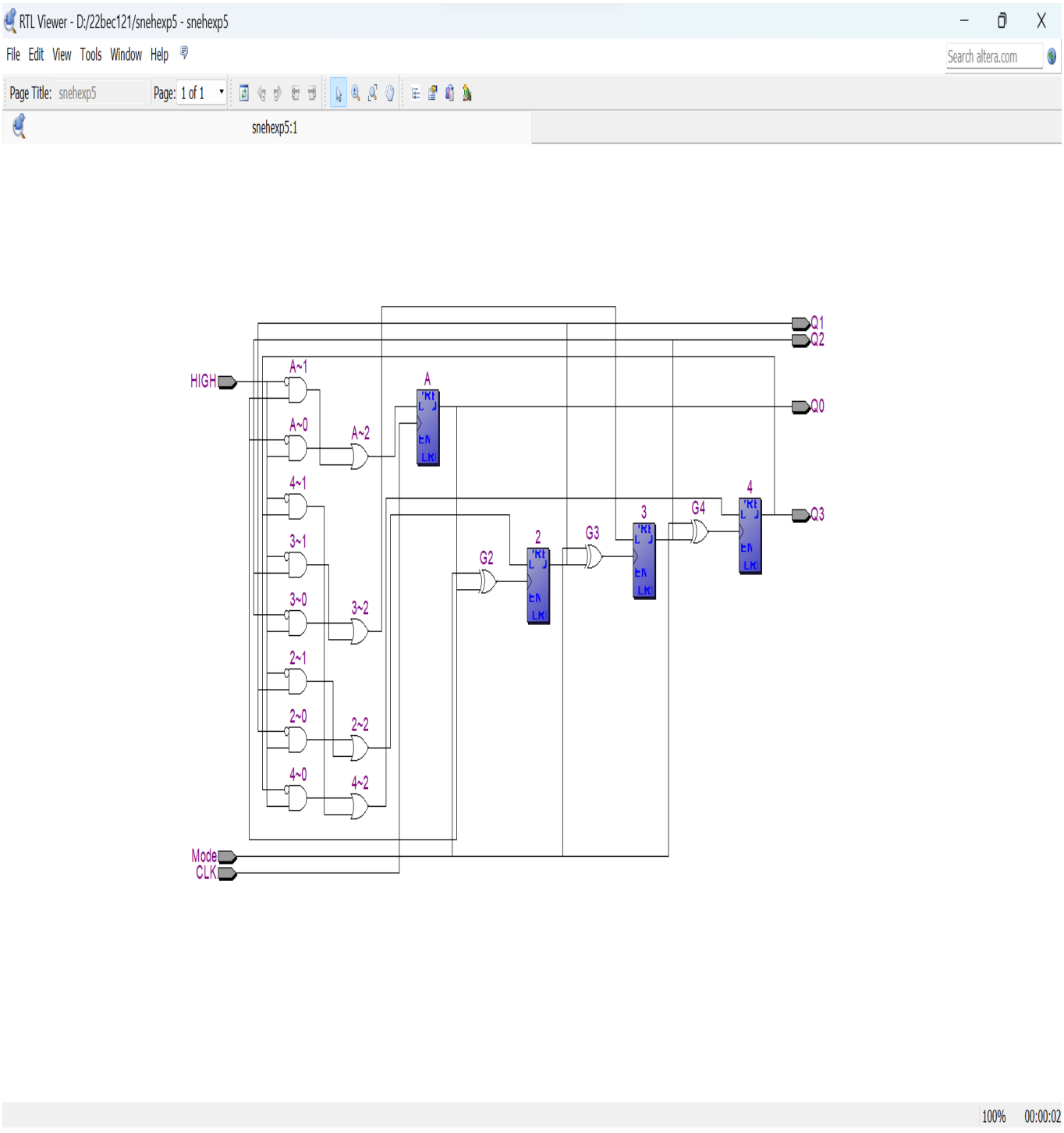

Quartus II 64-Bit - D:/22bec121/snehexp5 - snehexp5
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
snehexp5
Project Navigator
Cyclone II: EP2C35F67
snehexp5
Hierarchy
Tasks
Flow:
Customize...
Compile Des
Analysis
Edit Se
View R
Analys
Partitic
Netlist
RT
Sta
snehexp5.bdf
Compilation Report - snehexp5
snehexp5.v
1 module snehexp5(
2 Mode,
3 CLK,
4 HIGH,
5 Q3,
6 Q2,
7 Q1,
8 Q0
9);
10
11
12 input wire Mode;
13 input wire CLK;
14 input wire HIGH;
15 output reg Q3;
16 output wire Q2;
17 output wire Q1;
18 output wire Q0;
19
20 wire SYNTHESIZED_WIRE_0;
21 wire SYNTHESIZED_WIRE_1;
22 wire SYNTHESIZED_WIRE_2;
23 reg JKFF_A;
24 reg JKFF_2;
25 reg JKFF_3;
26
27 assign Q2 = JKFF_3;
28 assign Q1 = JKFF_2;
29 assign Q0 = JKFF_A;
30
31
All
<<Search>>
Type ID Message
Running Quartus II 64-Bit Create Verilog File
Messages
System
Processing (6)
1008, 0 100% 00:00:01



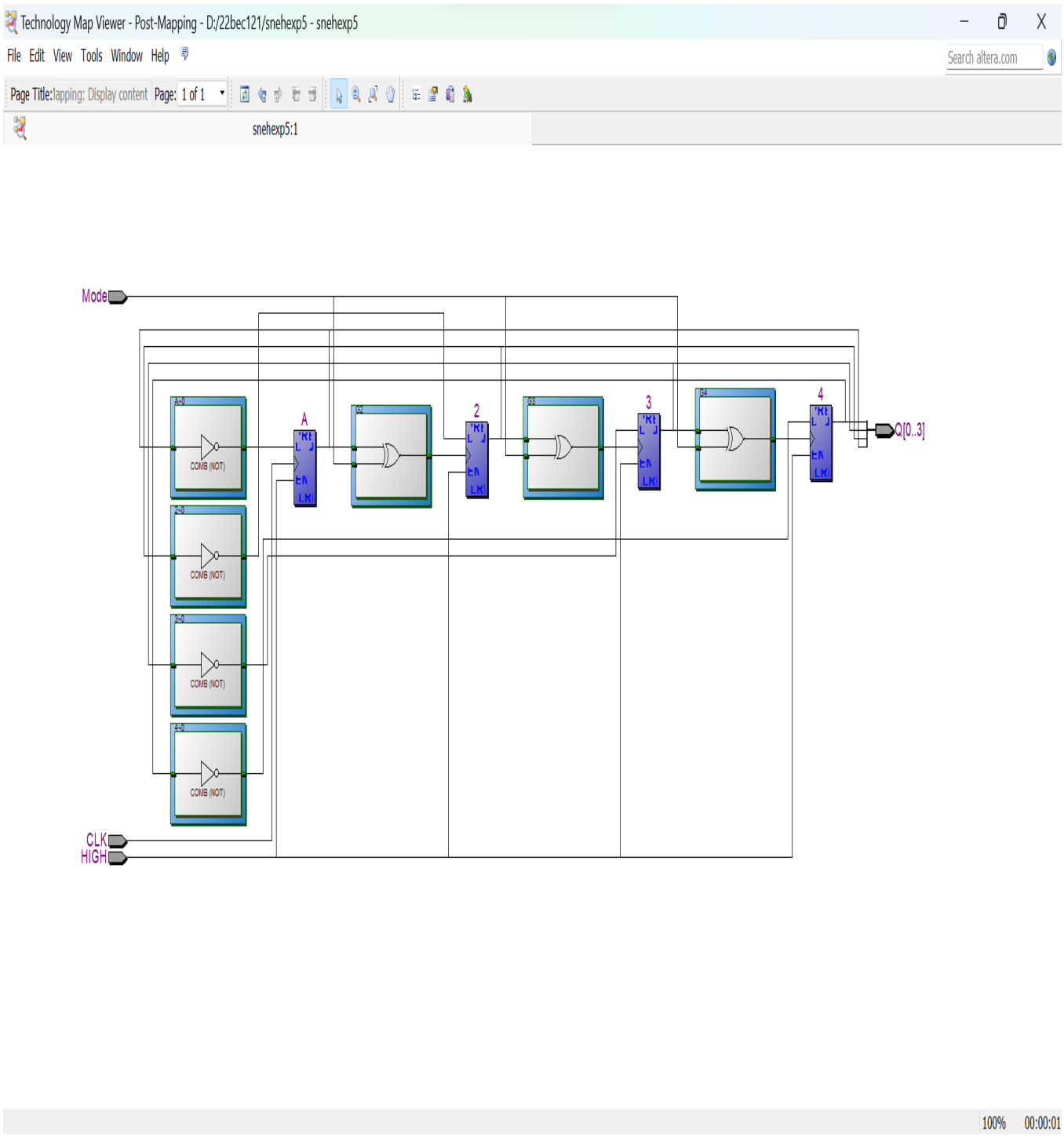
Block Design



Output 1)Cyclone II (RTL View)



2)Cyclone II (TTL View)



Conclusion :- In this experiment we learnt that how we can implement any circuit using Block Diagrams .

This is a tedious way to implement a circuit but it is a way to develop small circuits with ease .

We also generated Verilog Code from the Block Diagram to implement a 4 bit Up/Dow Counter .