

# Digital Design Diploma

## Assignment 5

### FSM & Memories

<b>Name</b>	Kareem Hassan Atif
<b>Group</b>	G2

**Submitted to: Eng. Kareem Waseem**

# 1. Tesla Car:

```

1 module Tesla_Car (input clk,rst, input [7:0] speed_limit,car_speed, input [6:0] leading_distance,
2 output reg unlock_doors, accelerate_car);
3
4 parameter MIN_DISTANCE = 7'd40;
5 localparam STOP = 2'b00, ACCELERATE = 2'b01, DECELERATE = 2'b10;
6 (*fsm_encoding = "gray"*)
7 reg [1:0] cs,ns;
8
9 /*--State Memory--*/
10 always @(posedge clk or posedge rst) begin
11     if (rst)        cs <= STOP;
12     else            cs <= ns;
13 end
14
15 /*--Next State and Output Logic--*/
16 always @(*) begin
17     case (cs)
18         STOP: begin
19             {unlock_doors,accelerate_car} = {1'b1,1'b0};
20             if (leading_distance >= MIN_DISTANCE) ns = ACCELERATE;
21             else ns = STOP;
22         end
23
24         ACCELERATE: begin
25             {unlock_doors,accelerate_car} = {1'b0,1'b1};
26             if (leading_distance >= MIN_DISTANCE && car_speed < speed_limit) ns = ACCELERATE;
27             else ns = DECELERATE;
28         end
29
30         DECELERATE: begin
31             {unlock_doors,accelerate_car} = {1'b0,1'b0};
32             if (!car_speed) ns = STOP;
33             else if (leading_distance >= MIN_DISTANCE && car_speed < speed_limit) ns = ACCELERATE;
34             else ns = DECELERATE;
35         end
36
37         default: {unlock_doors,accelerate_car, ns} = {1'b1,1'b0, STOP};
38     endcase
39 end
40 endmodule

```

Figure 1: Q1 Code

The screenshot shows the Q1 linting tool interface. The main window displays the Verilog code from Figure 1. The left pane shows the project structure with 'Tesla\_Car' selected. The right pane shows the 'Lint Summary' table.

Name	Count
Resolved(verified, fixed, waived)	3
Info	3
● async_reset_active_high	1
● logical_not_on_multi_bit	1
● multi_ports_in_single_line	1

Below the lint summary, there is a 'Lint Checks' section with a filter and a table of results.

Severity	Status	Check	Alias	Message	Module	Category
Total: 0 Selected: 0						

Figure 2: Q1 Linting

```

1  module Tesla_Car_tb ();
2  reg clk,rst;
3  reg [7:0] speed_limit,car_speed;
4  reg [6:0] leading_distance;
5  wire unlock_doors, accelerate_car;
6
7  /*module Tesla_Car (input clk,rst, input [7:0] speed_limit,car_speed, input [6:0] leading_distance,
8  output reg unlock_doors, accelerate_car);*/
9  Tesla_Car DUT(clk,rst,speed_limit,car_speed,leading_distance,unlock_doors, accelerate_car);
10
11 initial begin
12     clk=0;
13     forever #5 clk=~clk;
14 end
15
16 initial begin
17     rst=1; speed_limit=$random; car_speed=$random; leading_distance=$random;
18     #2;
19     if (DUT.cs != DUT.STOP || !unlock_doors || accelerate_car) begin
20         $display ("Error in reset!!");
21         $stop;
22     end
23
24     rst=0; speed_limit=$random; car_speed=$random; leading_distance=41;
25     @(negedge clk);
26     if (DUT.cs != DUT.ACCELERATE || unlock_doors || !accelerate_car) begin
27         $display ("Error in ACCELERATE!!");
28         $stop;
29     end
30
31     speed_limit=80; car_speed=70; leading_distance=41;
32     @(negedge clk);
33     if (DUT.cs != DUT.ACCELERATE || unlock_doors || !accelerate_car) begin
34         $display ("Error in ACCELERATE!!");
35         $stop;
36     end
37
38     speed_limit=80; car_speed=90; leading_distance=41;
39     @(negedge clk);
40     if (DUT.cs != DUT.DECELERATE || unlock_doors || accelerate_car) begin
41         $display ("Error in DECELERATE!!");
42         $stop;
43     end
44
45     car_speed=0;
46     @(negedge clk);
47     if (DUT.cs != DUT.STOP || !unlock_doors || accelerate_car) begin
48         $display ("Error in STOP!!");
49         $stop;
50     end
51
52     repeat (10) begin
53         speed_limit=$random; car_speed=$random; leading_distance=$random;
54         @(negedge clk);
55     end
56     $stop;
57 end
58 endmodule

```

Figure 3: Q1 Testbench

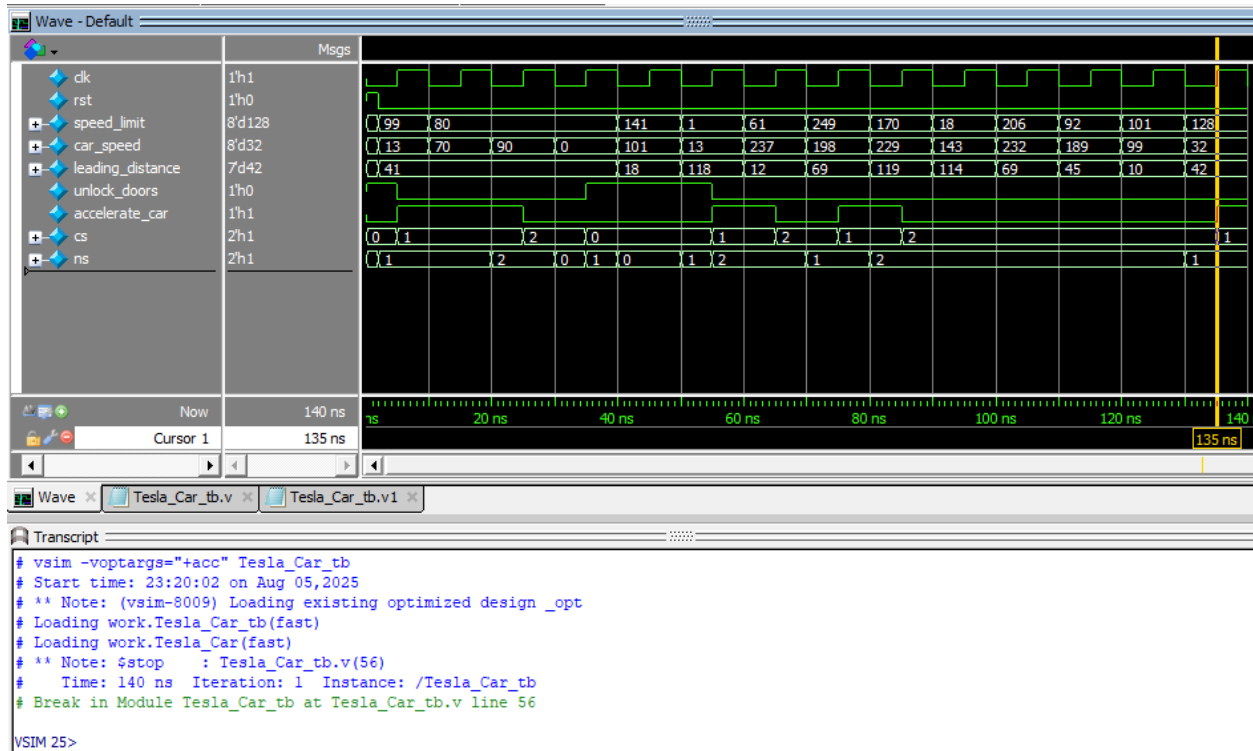


Figure 4: Q1 Wave

```
1 vlib work
2 vlog Tesla_Car.v Tesla_Car_tb.v
3 vsim -voptargs="+acc" Tesla_Car_tb
4 add wave *
5 add wave -position insertpoint \
6 sim:/Tesla_Car_tb/DUT/cs \
7 sim:/Tesla_Car_tb/DUT/ns
8 run -all
9 #quit -sim
```

Figure 5: Q1 DO File

```
1 ## This file is a general .xdc for the Basys3 rev B board
2 ## To use it in a project:
3 ## - uncomment the lines corresponding to used pins
4 ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6 ## Clock signal
7 set_property -dict { PACKAGE_PIN W5 IOSTANDARD LVCMOS33 } [get_ports clk]
8 create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
9
10 ## Configuration options, can be used for all designs
11 set_property CONFIG_VOLTAGE 3.3 [current_design]
12 set_property CFGBVS VCC0 [current_design]
13
14 ## SPI configuration mode options for QSPI boot, can be used for all designs
15 set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
16 set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
17 set_property CONFIG_MODE SPIx4 [current_design]
```

Figure 6: Q1 Constraint File

## 1.1 Gray Encoding:

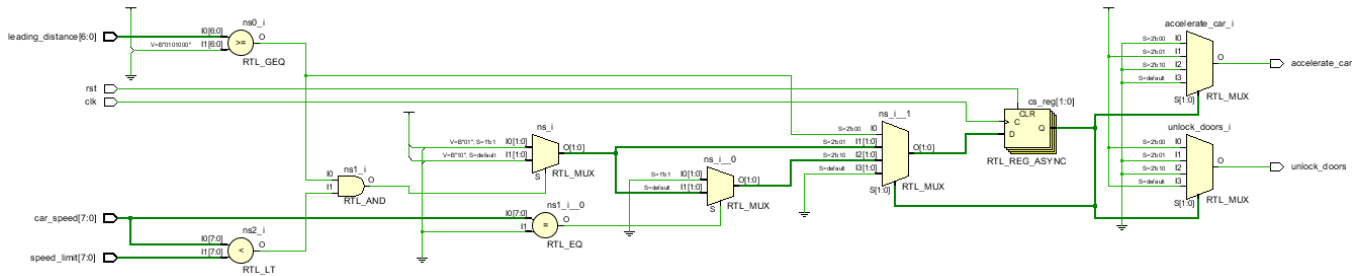


Figure 7: Q1 RTL (Gray)

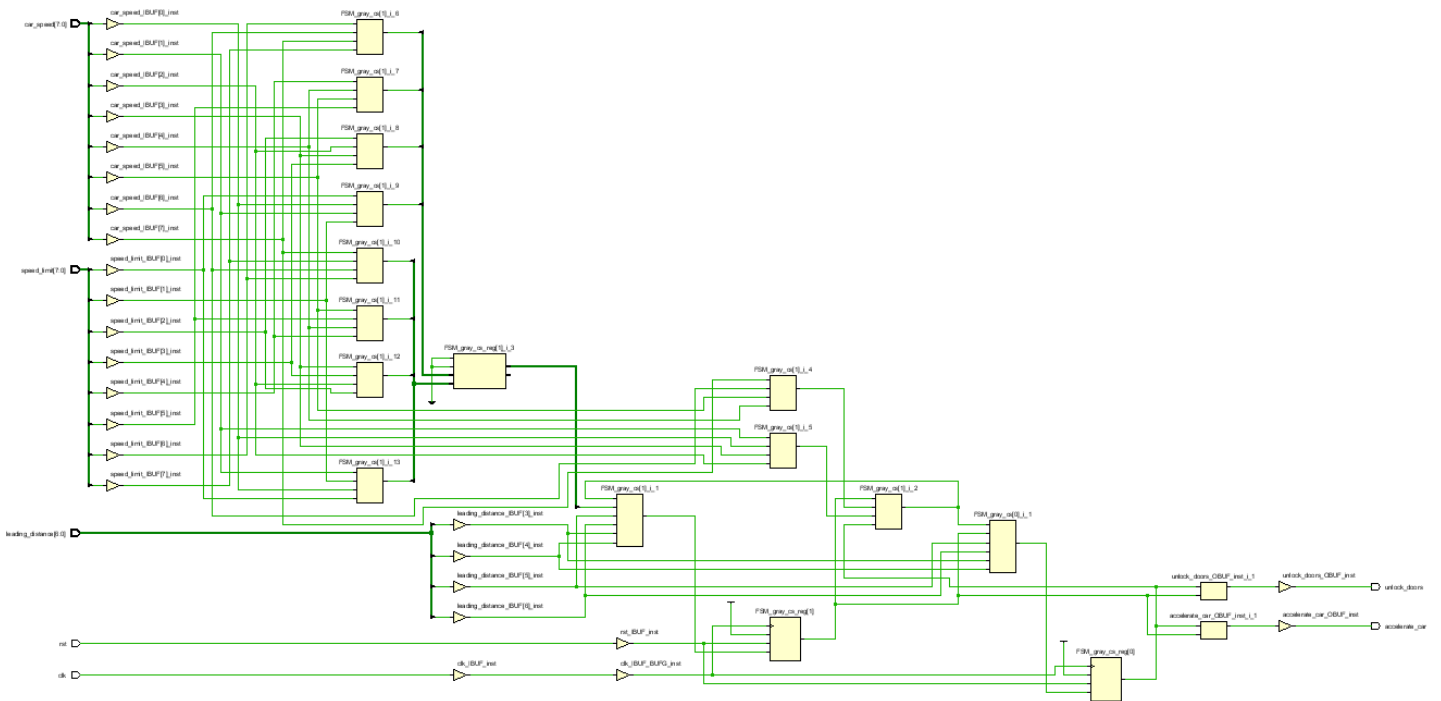


Figure 8: Q1 Synthesis (Gray)

State	New Encoding	Previous Encoding
STOP	00	00
ACCELERATE	01	01
DECELERATE	11	10

INFO: [Synth 8-3354] encoded FSM with state register 'cs\_reg' using encoding 'gray' in module 'Tesla\_Car'

Figure 9: Q1 Encoding Report (Gray)

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Wed Aug  6 00:06:56 2025
5 // Host       : DESKTOP-DTNSA8B running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q1/Tesla_Car_Gray.v}
7 // Design     : Tesla_Car
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35ticpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* ACCELERATE = "2'b01" *) (* DECELERATE = "2'b10" *) (* MIN_DISTANCE = "7'b0101000" *)
16 (* STOP = "2'b00" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Tesla_Car
19 (
20     clk,
21     rst,
22     speed_limit,
23     car_speed,
24     leading_distance,
25     unlock_doors,
26     accelerate_car);
27 input clk;
28 input rst;
29 input [7:0]speed_limit;
30 input [7:0]car_speed;
31 input [6:0]leading_distance;
32 output unlock_doors;
33 output accelerate_car;

```

Figure 10: Q1 Netlist (Gray)

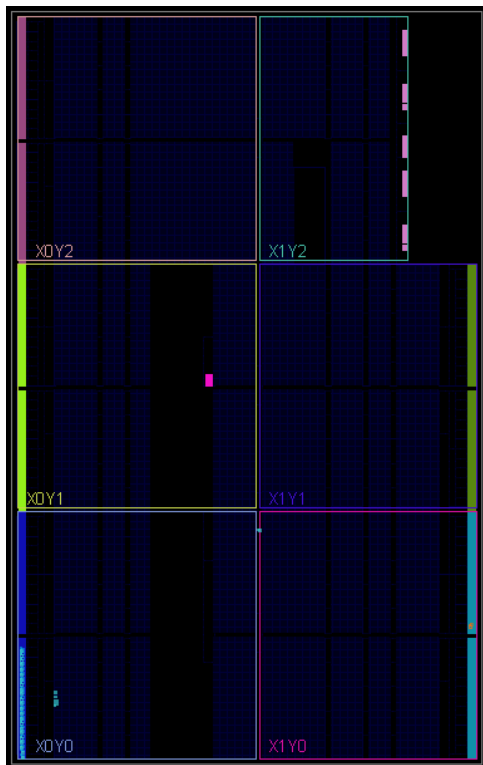


Figure 11: Q1 Device (Gray)

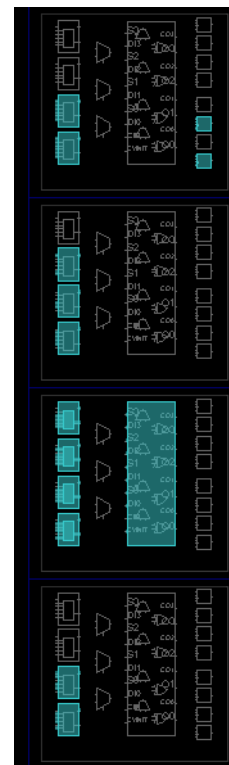


Figure 12: Q1 Zoomed Device (Gray)

### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 7.755 ns	Worst Hold Slack (WHS): 0.420 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 2	Total Number of Endpoints: 2	Total Number of Endpoints: 3

All user specified timing constraints are met.

Figure 13: Q1 Synthesis Timing Report (Gray)

### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.181 ns	Worst Hold Slack (WHS): 0.521 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 2	Total Number of Endpoints: 2	Total Number of Endpoints: 3

All user specified timing constraints are met.

Figure 14: Q1 Implementation Timing Report (Gray)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
N Tesla_Car		11	2	24	1

Figure 15: Q1 Synthesis Utilization Report (Gray)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
N Tesla_Car		11	2	4	11	2	24	1

Figure 16: Q1 Implementation Utilization Report (Gray)

## 1.2 Sequential Encoding:

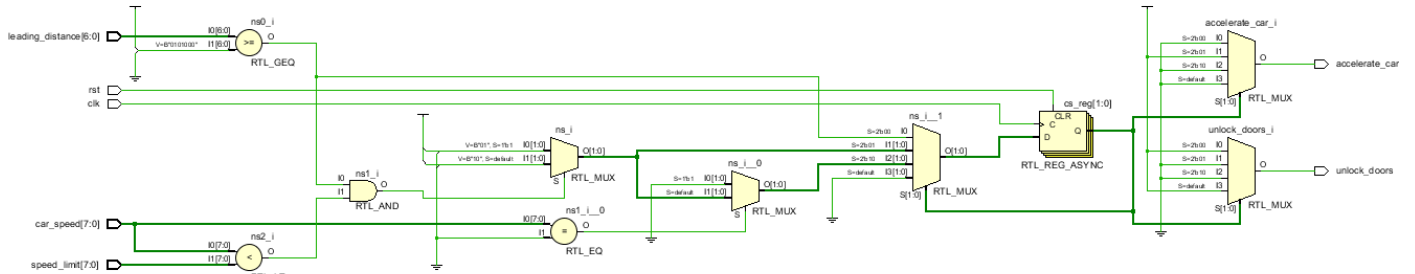


Figure 17: Q1 RTL (Sequential)

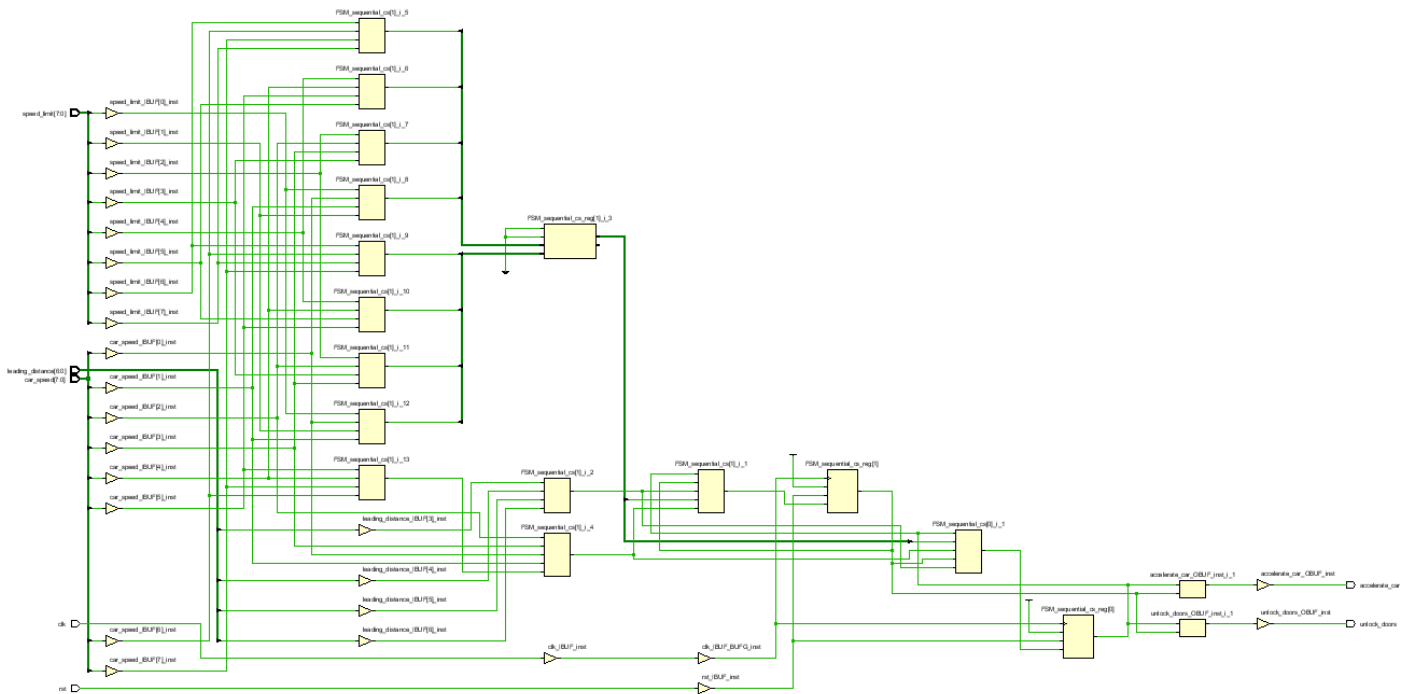


Figure 18: Q1 Synthesis (Sequential)

State	New Encoding	Previous Encoding
STOP	00	00
ACCELERATE	01	01
DECELERATE	10	10

INFO: [Synth 8-3354] encoded FSM with state register 'cs\_reg' using encoding 'sequential' in module 'Tesla\_Car'

Figure 19: Q1 Encoding Report (Sequential)



```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Wed Aug  6 00:57:56 2025
5 // Host       : DESKTOP-DTNSAHB running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q1/Tesla_Car_Seq.v}
7 // Design     : Tesla_Car
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is a
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35ticpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* ACCELERATE = "2'b01" *) (* DECELERATE = "2'b10" *) (* MIN_DISTANCE = "7'b0101000" *)
16 (* STOP = "2'b00" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Tesla_Car
19   (clk,
20    rst,
21    speed_limit,
22    car_speed,
23    leading_distance,
24    unlock_doors,
25    accelerate_car);
26   input clk;
27   input rst;
28   input [7:0]speed_limit;
29   input [7:0]car_speed;
30   input [6:0]leading_distance;
31   output unlock_doors;
32   output accelerate_car;

```

Figure 20: Q1 Netlist (Sequential)

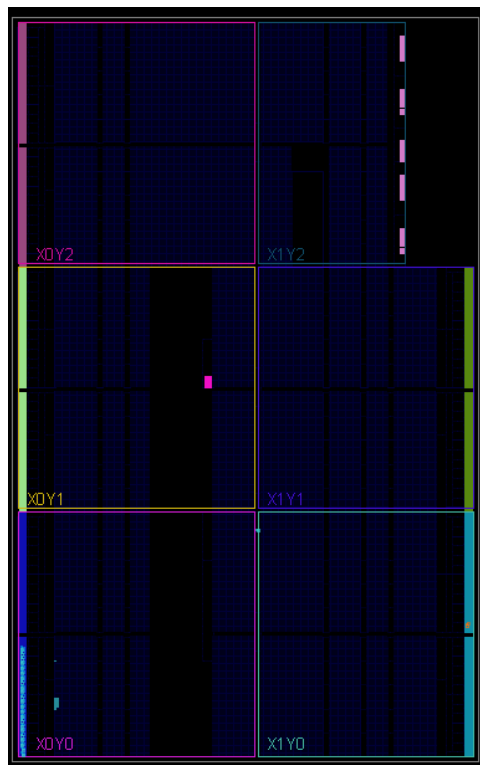


Figure 21: Q1 Device (Sequential)

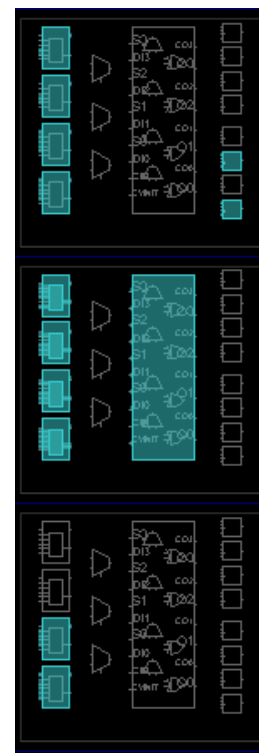


Figure 22: Q1 Zoomed Device (Sequential)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.268 ns	Worst Hold Slack (WHS): 0.294 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 2	Total Number of Endpoints: 2	Total Number of Endpoints: 3

All user specified timing constraints are met.

Figure 23: Q1 Synthesis Timing Report (Sequential)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.745 ns	Worst Hold Slack (WHS): 0.248 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 2	Total Number of Endpoints: 2	Total Number of Endpoints: 3

All user specified timing constraints are met.

Figure 24: Q1 Implementation Timing Report (Sequential)

Name	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
<b>N</b> Tesla_Car	11	2	24	1

Figure 25: Q1 Synthesis Utilization Report (Sequential)

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
<b>N</b> Tesla_Car	11	2	4	11	2	24	1

Figure 26: Q1 Implementation Utilization Report (Sequential)

### 1.3 One-Hot Encoding:

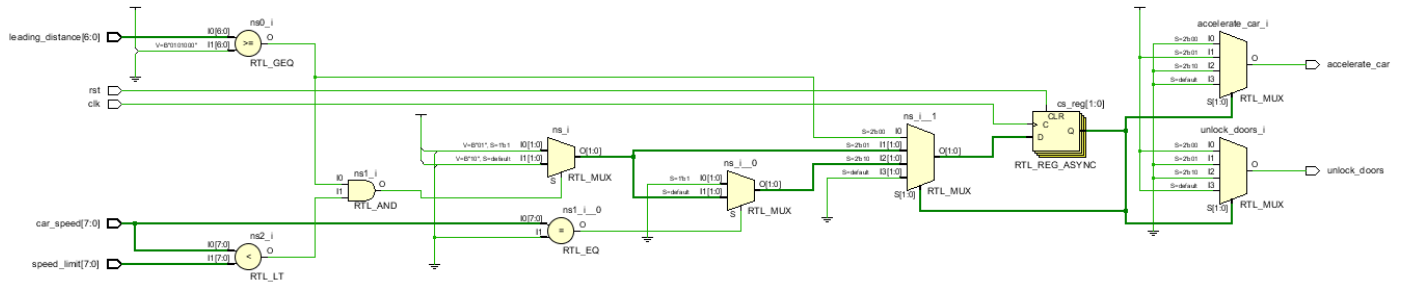


Figure 27: Q1 RTL (One-Hot)

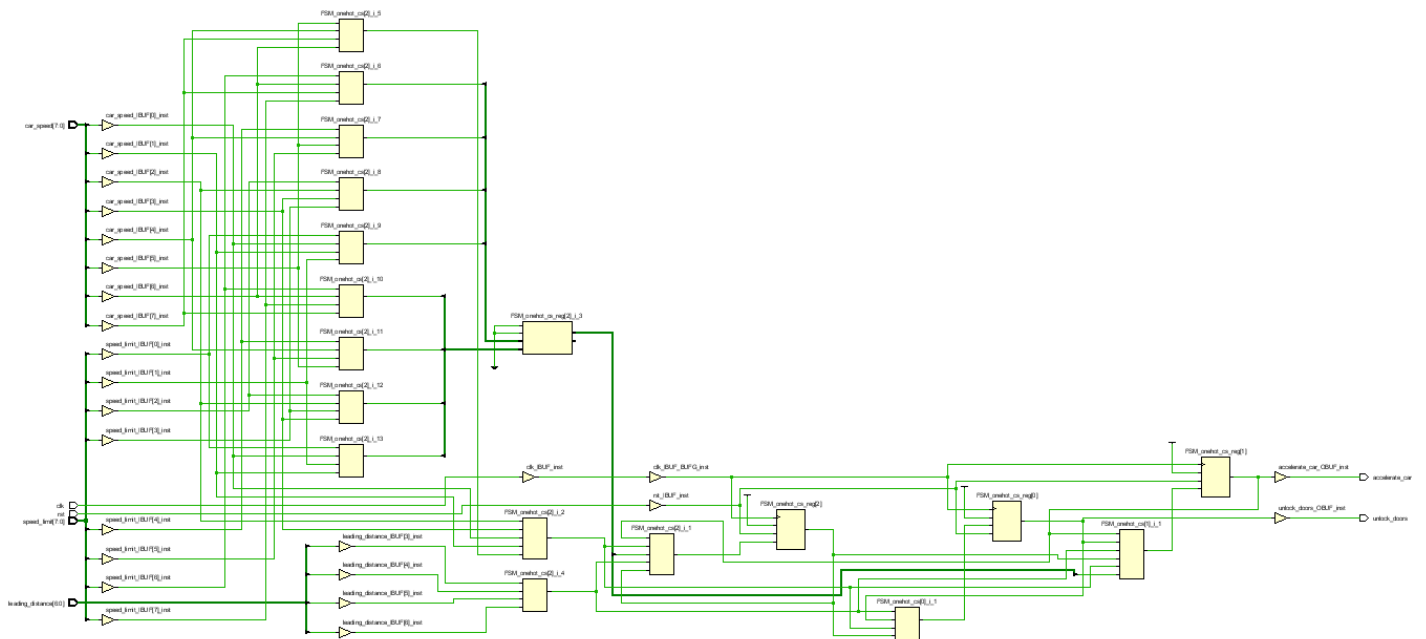


Figure 28: Q1 Synthesis (One-Hot)

State	New Encoding	Previous Encoding
STOP	001	00
ACCELERATE	010	01
DECELERATE	100	10

INFO: [Synth 8-3354] encoded FSM with state register 'cs\_reg' using encoding 'one-hot' in module 'Tesla\_Car'

Figure 29: Q1 Encoding Report (One-Hot)

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Wed Aug  6 00:35:30 2025
5 // Host       : DESKTOP-DTNSAHB running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q1/Tesla_Car_One_Hot.v}
7 // Design     : Tesla_Car
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35ticpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* ACCELERATE = "2'b01" *) (* DECELERATE = "2'b10" *) (* MIN_DISTANCE = "7'b0101000" *)
16 (* STOP = "2'b00" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Tesla_Car
19 (clk,
20  rst,
21  speed_limit,
22  car_speed,
23  leading_distance,
24  unlock_doors,
25  accelerate_car);
26 input clk;
27 input rst;
28 input [7:0]speed_limit;
29 input [7:0]car_speed;
30 input [6:0]leading_distance;
31 output unlock_doors;
32 output accelerate_car;

```

Figure 30: Q1 Netlist (One-Hot)

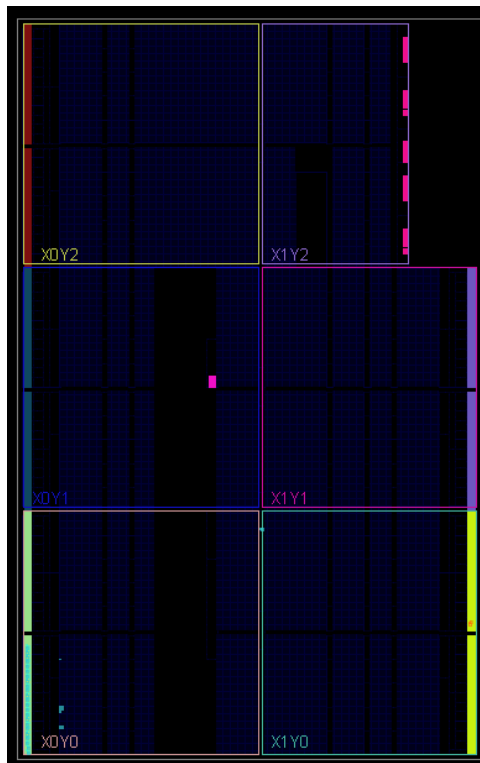


Figure 31: Q1 Device (One-Hot)

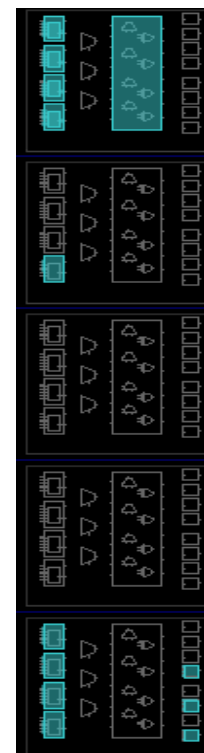


Figure 32: Q1 Zoomed Device (One-Hot)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.115 ns	Worst Hold Slack (WHS): 0.291 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 3	Total Number of Endpoints: 3	Total Number of Endpoints: 4

All user specified timing constraints are met.

Figure 33: Q1 Synthesis Timing Report (One-Hot)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.697 ns	Worst Hold Slack (WHS): 0.281 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 3	Total Number of Endpoints: 3	Total Number of Endpoints: 4

All user specified timing constraints are met.

Figure 34: Q1 Implementation Timing Report (One-Hot)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
N Tesla_Car		10	3	24	1

Figure 35: Q1 Synthesis Utilization Report (One-Hot)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
N Tesla_Car		10	3	4	10	3	24	1

Figure 36: Q1 Implementation Utilization Report (One-Hot)

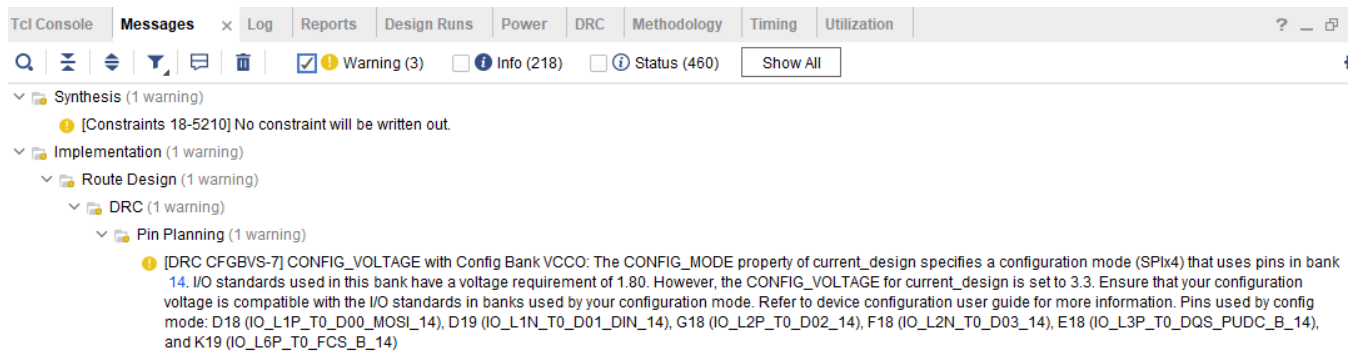


Figure 37: Q1 Messages

Sequential and Gray encodings use two flip flops, while one-hot encoding uses three flip flops. Sequential encoding has the highest setup slack. So, Sequential is the best encoding in this case, as it has the highest slack which means higher frequency clock, with the least number of registers used.

## 2. Gray Counter FSM:

```
1  module Gray_Counter_FSM (input clk,rst, output reg [1:0] y);
2  localparam A=2'b00,  B=2'b01,  C=2'b10,  D=2'b11;
3  reg [1:0] cs,ns;
4
5  /*--State Memory--*/
6  always @(posedge clk or posedge rst) begin
7      if (rst)      cs <= A;
8      else          cs <= ns;
9  end
10
11 /*--Next State and Output Logic--*/
12 always @(*) begin
13     case (cs)
14         A:      {ns, y} = {B, 2'b00};
15         B:      {ns, y} = {C, 2'b01};
16         C:      {ns, y} = {D, 2'b11};
17         D:      {ns, y} = {A, 2'b10};
18         default: {ns, y} = {A, 2'b00};
19     endcase
20 end
21 endmodule
```

Figure 38: Q2 Code

```
1  module Gray_Counter (input clk,rst, output [1:0] gray_out);
2  reg [1:0] binary_out;
3
4  always @(posedge clk or posedge rst) begin
5      if (rst)  binary_out <= 0;
6      else      binary_out <= binary_out + 1;
7  end
8
9  assign gray_out = {binary_out[1] , ^binary_out};
10
11 endmodule
```

Figure 39: Q2 Golden Model Code

```

1  module Gray_Counter_FSM_tb ();
2  reg clk,rst;
3  wire [1:0] y,y_Golden;
4
5  //module Gray_Counter_FSM (input clk,rst, output reg [1:0] y);
6  Gray_Counter_FSM DUT(clk,rst,y);
7  //module Gray_Counter (input clk,rst, output reg [1:0] gray_out);
8  Gray_Counter DUT_GOLDEN(clk,rst,y_Golden);
9
10 initial begin
11     clk=0;
12     forever #5 clk=~clk;
13 end
14
15 initial begin
16     rst=1;
17     #2;
18     if (DUT.cs != DUT.A || y !=y_Golden) begin
19         $display("Error in Reset!!");
20         $stop;
21     end
22     rst=0;
23     @(negedge clk);
24     if (DUT.cs != DUT.B || y !=y_Golden) begin
25         $display("Error!!");
26         $stop;
27     end
28     @(negedge clk);
29     if (DUT.cs != DUT.C || y !=y_Golden) begin
30         $display("Error!!");
31         $stop;
32     end
33     @(negedge clk);
34     if (DUT.cs != DUT.D || y !=y_Golden) begin
35         $display("Error!!");
36         $stop;
37     end
38     @(negedge clk);
39     if (DUT.cs != DUT.A || y !=y_Golden) begin
40         $display("Error!!");
41         $stop;
42     end
43     repeat (5) begin
44         @(negedge clk);
45         if (y !=y_Golden) begin
46             $display("Error!!");
47             $stop;
48         end
49     end
50     $stop;
51 end
52 endmodule

```

Figure 40: Q2 Testbench



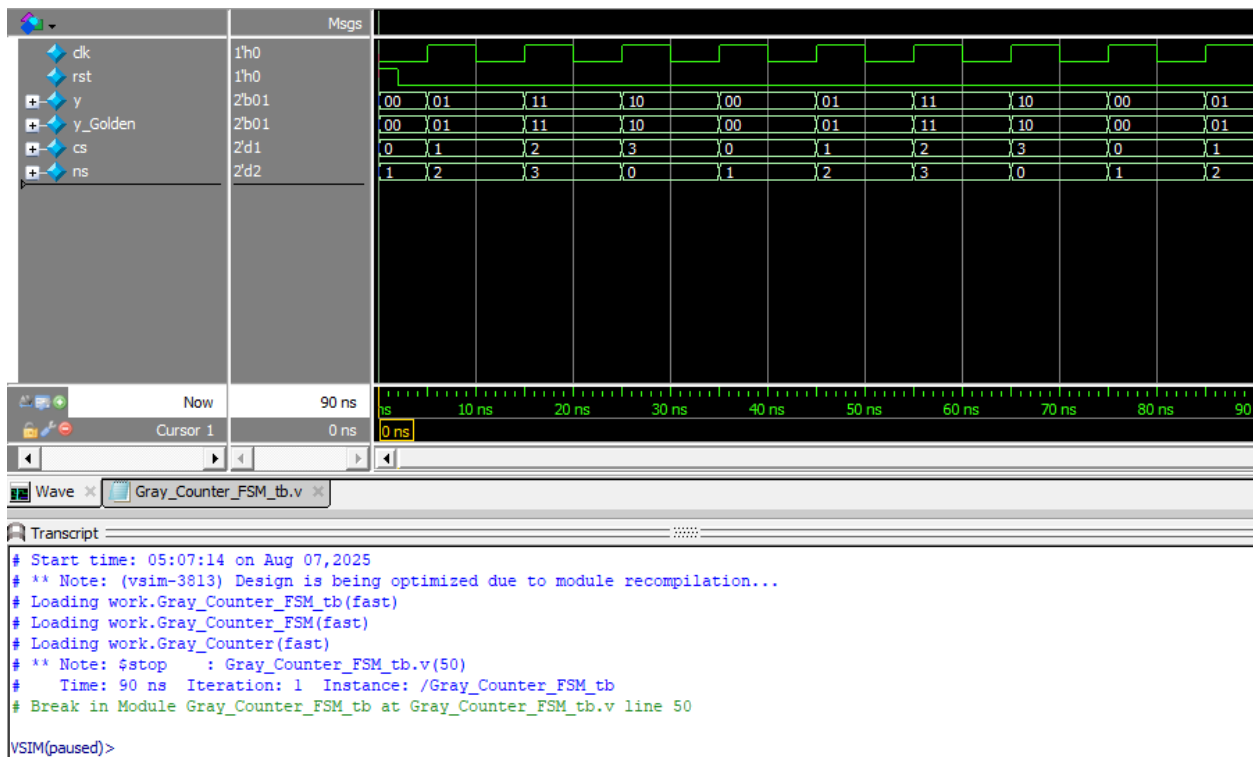


Figure 41: Q2 Wave

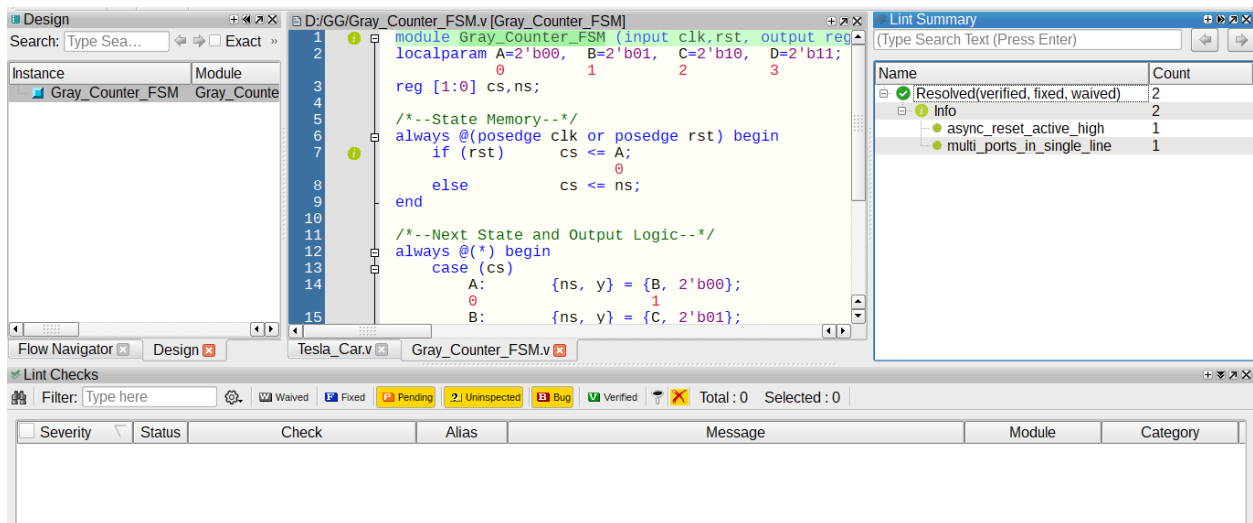


Figure 42: Q2 Linting

```

1  vlib work
2  vlog Gray_Counter_FSM.v Gray_Counter_FSM_tb.v
3  vsim -voptargs=+acc Gray_Counter_FSM_tb
4  add wave *
5  add wave -position insertpoint \
6  sim:/Gray_Counter_FSM_tb/DUT/cs \
7  sim:/Gray_Counter_FSM_tb/DUT/ns
8  run -all
9  #quit -sim

```

Figure 43: Q2 DO File

```

1  ## This file is a general .xdc for the Basys3 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  ## Clock signal
7  set_property -dict { PACKAGE_PIN W5    IOSTANDARD LVCMOS33 } [get_ports clk]
8  create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
9
10 ## LEDs
11 set_property -dict { PACKAGE_PIN U16    IOSTANDARD LVCMOS33 } [get_ports {y[0]}]
12 set_property -dict { PACKAGE_PIN E19    IOSTANDARD LVCMOS33 } [get_ports {y[1]}]
13
14 ##Buttons
15 set_property -dict { PACKAGE_PIN U18    IOSTANDARD LVCMOS33 } [get_ports rst]
16
17 ## Configuration options, can be used for all designs
18 set_property CONFIG_VOLTAGE 3.3 [current_design]
19 set_property CFGBVS VCC0 [current_design]
20
21 ## SPI configuration mode options for QSPI boot, can be used for all designs
22 set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
23 set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
24 set_property CONFIG_MODE SPIx4 [current_design]

```

Figure 44: Q2 Constraint File

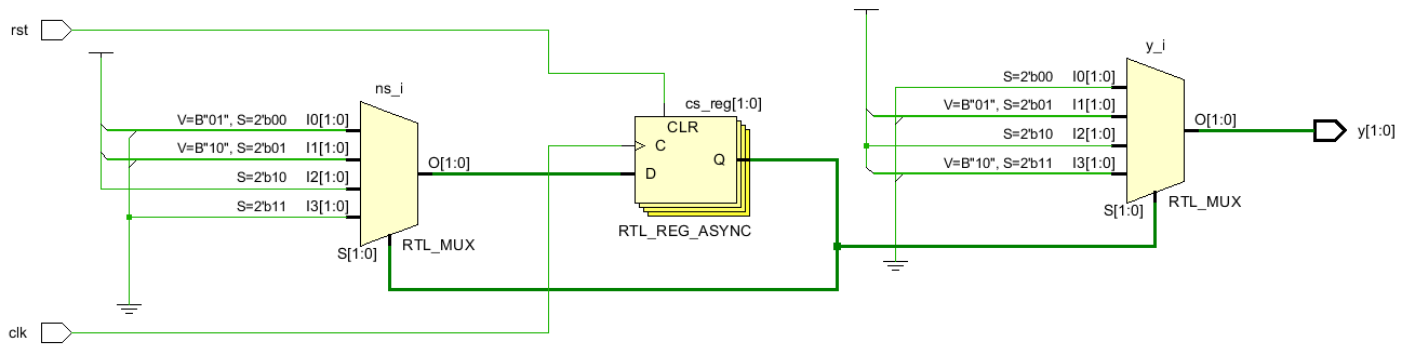


Figure 45: Q2 RTL

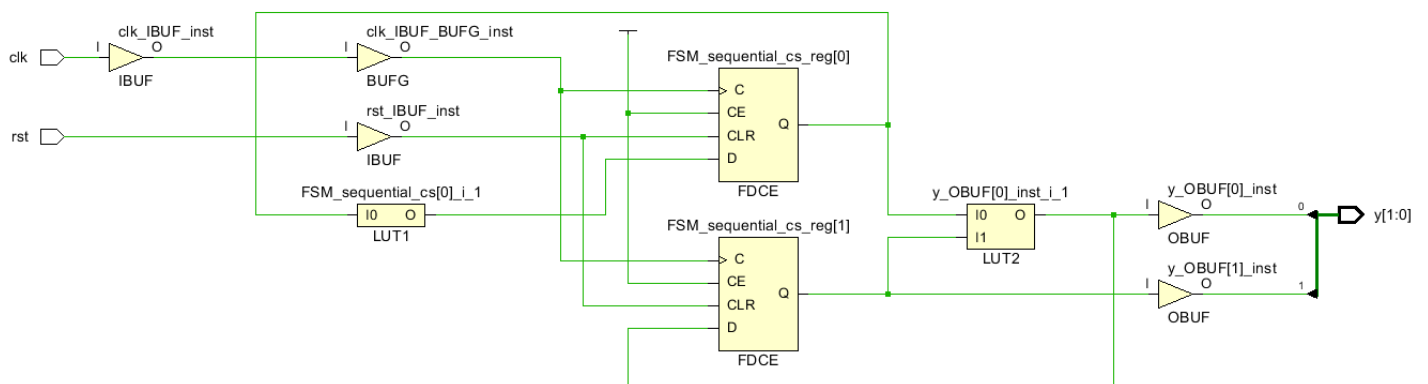


Figure 46: Q2 Synthesis

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 //
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date : Wed Aug 6 03:22:40 2025
5 // Host : DESKTOP-DTNSAHB running 64-bit major release (build 9200)
6 // Command : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q2/Gray_Counter_FSM_Netlist.v}
7 // Design : Gray_Counter_FSM
8 // Purpose : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 // IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 // design files.
11 // Device : xc7a35ticpg236-1L
12 //
13 `timescale 1 ps / 1 ps
14
15 (* A = "2'b00" *) (* B = "2'b01" *) (* C = "2'b10" *)
16 (* D = "2'b11" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Gray_Counter_FSM
19 (
20 clk,
21 rst,
22 y);
23 input clk;
24 input rst;
25 output [1:0]y;
26
27 wire \<const1> ;
28 wire clk;
29 wire clk_IBUF;
30 wire clk_IBUF_BUFG;
31 (* RTL_KEEP = "yes" *) wire [0:0]cs;
32 wire [0:0]ns;
33 wire rst;

```

Figure 47: Q2 Netlist

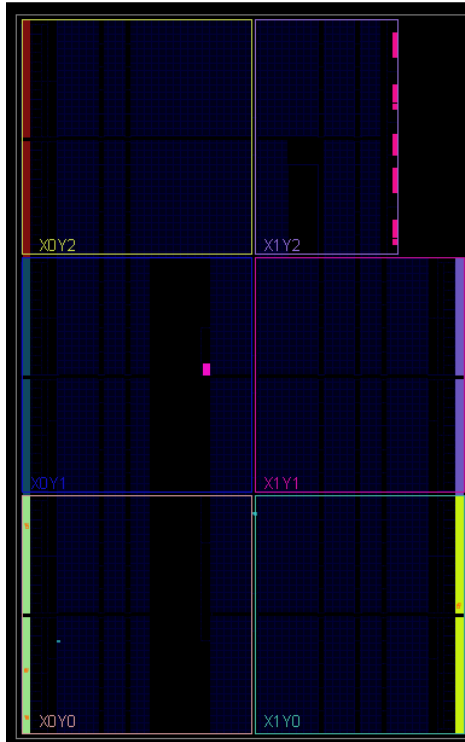


Figure 48: Q2 Device

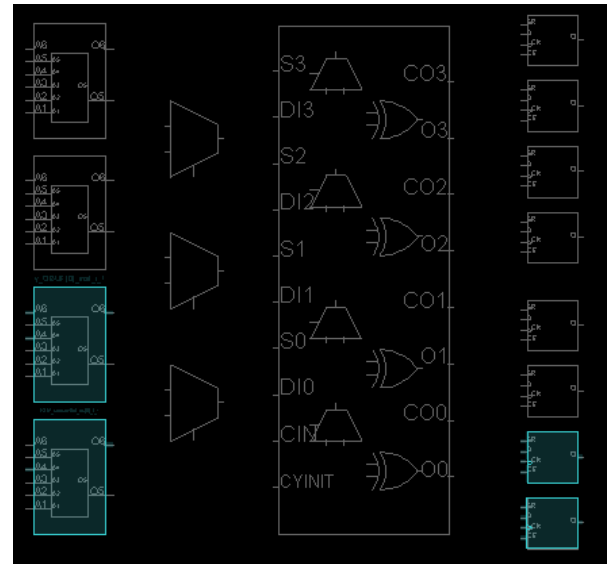


Figure 49: Q2 Zoomed Device

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.617 ns	Worst Hold Slack (WHS): 0.139 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 2	Total Number of Endpoints: 2	Total Number of Endpoints: 3

All user specified timing constraints are met.

Figure 50: Q2 Synthesis Timing Report

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.019 ns	Worst Hold Slack (WHS): 0.367 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 2	Total Number of Endpoints: 2	Total Number of Endpoints: 3

All user specified timing constraints are met.

Figure 51: Q2 Implementation Timing Report

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
Gray_Counter_FSM		2	2	4	1

Figure 52: Q2 Synthesis Utilization Report

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
Gray_Counter_FSM		2	2	1	2	1	4	1

Figure 53: Q2 Implementation Utilization Report

Bitstream Generation Completed

Bitstream Generation successfully completed.

Next

☒

View Reports

☐

Open Hardware Manager

☐

Generate Memory Configuration File

☐

Don't show this dialog again

OK

Cancel

Figure 54: Q2 Bitstream

Tcl ConsoleMessagesLogReportsDesign Run

☒

Warning (1)

Synthesis (1 warning)

[Constraints 18-5210] No constraint will be written out.

Figure 55: Q2 Messages

### 3. Sequence Detector (111):

```

1 module Sequence_Detector_111 (input clk,rst,Din, output ERR);
2   localparam START = 3'b000, D0_IS_1 = 3'b001, D1_IS_1 = 3'b010, D0_NOT_1 = 3'b011, D1_NOT_1 = 3'b100;
3   (*fsm_encoding = "gray"*)
4   reg [2:0] cs,ns;
5
6   /*--State Memory--*/
7   always @(posedge clk or posedge rst) begin
8     if (rst) cs <= START;
9     else cs <= ns;
10  end
11
12  /*--Next State Logic--*/
13  always @(*) begin
14    case (cs)
15      START: begin
16        if (Din) ns = D0_IS_1;
17        else ns = D0_NOT_1;
18      end
19      D0_IS_1: begin
20        if (Din) ns = D1_IS_1;
21        else ns = D1_NOT_1;
22      end
23      D1_IS_1: ns = START;
24      D0_NOT_1: ns = D1_NOT_1;
25      D1_NOT_1: ns = START;
26      default: ns = START;
27    endcase
28  end
29  assign ERR = (cs == D1_IS_1 && Din)? 1: 0;
30
31 endmodule

```

Figure 56: Q3 Code

The screenshot displays the Q3 Linting tool interface. The main window shows the Verilog code from Figure 56. On the right, the 'Lint Summary' window is open, displaying a table with the following data:

Name	Count
Resolved(verified, fixed, waived)	2
Info	2
● async_reset_active_high	1
● multi_ports_in_single_line	1

At the bottom, the 'Lint Checks' window shows a summary of linting results:

Severity	Status	Check	Alias	Message	Module	Category
Total: 0 Selected: 0						

Figure 57: Q3 Linting

```

1  module Sequence_Detector_111_tb ();
2  reg clk,rst,Din;
3  wire ERR;
4
5  //module Sequence_Detector_111 (input clk,rst,Din, output reg ERR);
6  Sequence_Detector_111 DUT(clk,rst,Din,ERR);
7
8  initial begin
9      clk=0;
10     forever #5 clk=~clk;
11 end
12
13 initial begin
14     rst=1; Din=$random;
15     @(negedge clk);
16     if (DUT.cs != DUT.START || ERR) begin
17         $display("Error in Reset!!");
18         $stop;
19     end
20     rst=0;
21     repeat (30) begin
22         Din=$random;
23         @(negedge clk);
24     end
25     $stop;
26 end
27 endmodule

```

Figure 58: Q3 Testbench

```

1  vlib work
2  vlog Sequence_Detector_111.v Sequence_Detector_111_tb.v
3  vsim -voptargs=+acc Sequence_Detector_111_tb
4  add wave *
5  add wave -position insertpoint \
6  sim:/Sequence_Detector_111_tb/DUT/cs \
7  sim:/Sequence_Detector_111_tb/DUT/ns
8  run -all
9  #quit -sim

```

Figure 59: Q3 DO File

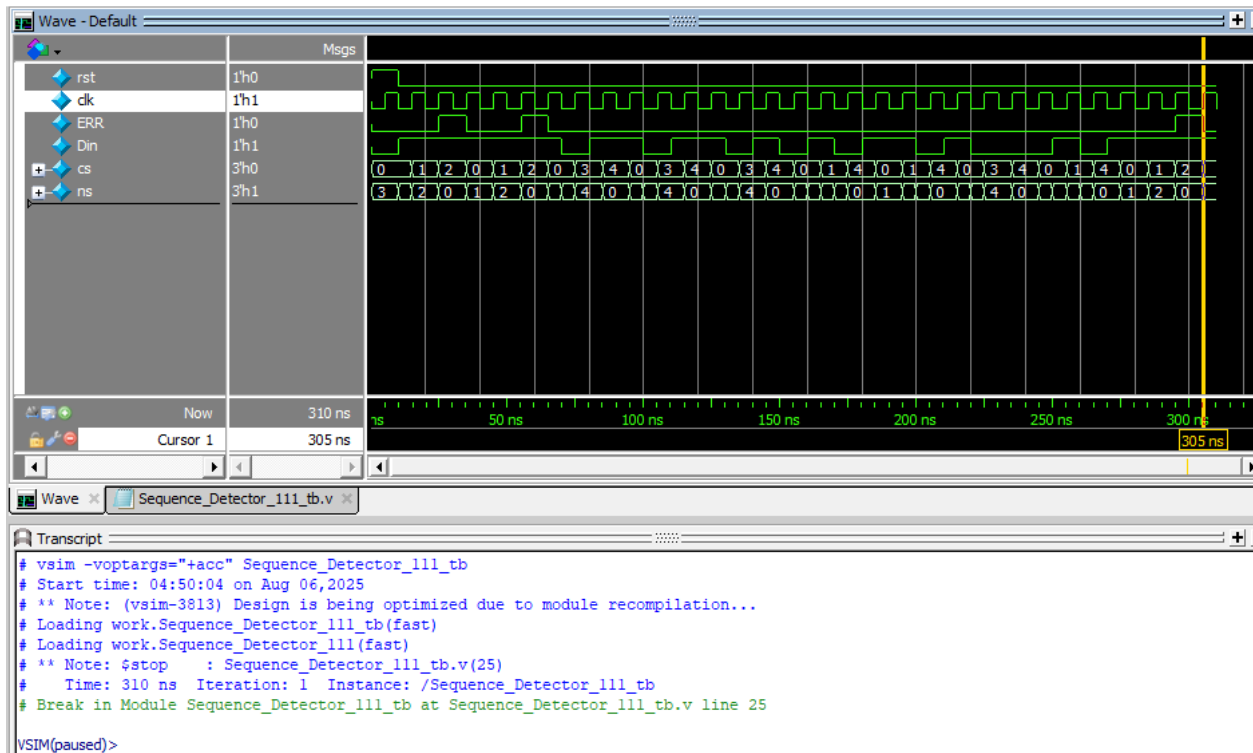


Figure 60: Q3 Wave

```
1  ## This file is a general .xdc for the Basys3 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  ## Clock signal
7  set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
8  create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
9
10
11  ## Switches
12  set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {Din}]
13
14  ## LEDs
15  set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {ERR}]
16
17  ##Buttons
18  set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports rst]
19
20
21  ## Configuration options, can be used for all designs
22  set_property CONFIG_VOLTAGE 3.3 [current_design]
23  set_property CFGBVS VCC0 [current_design]
24
25  ## SPI configuration mode options for QSPI boot, can be used for all designs
26  set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
27  set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
28  set_property CONFIG_MODE SPIx4 [current_design]
```

Figure 61: Q3 Constraint File



### 3.1 Gray Encoding:

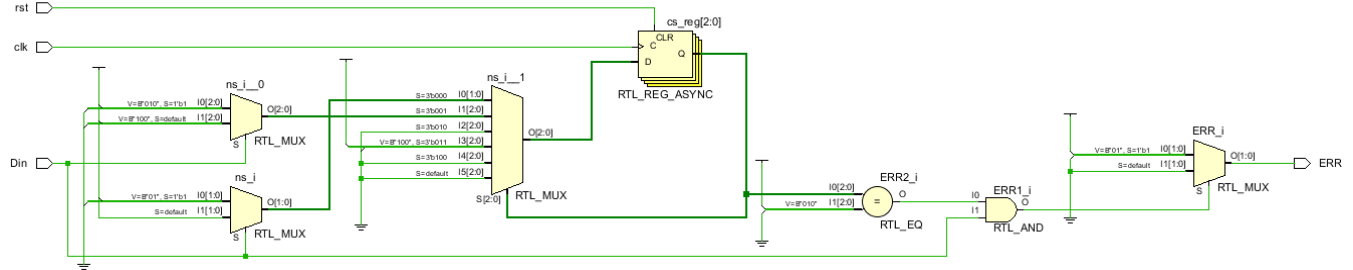


Figure 62: Q3 RTL (Gray)

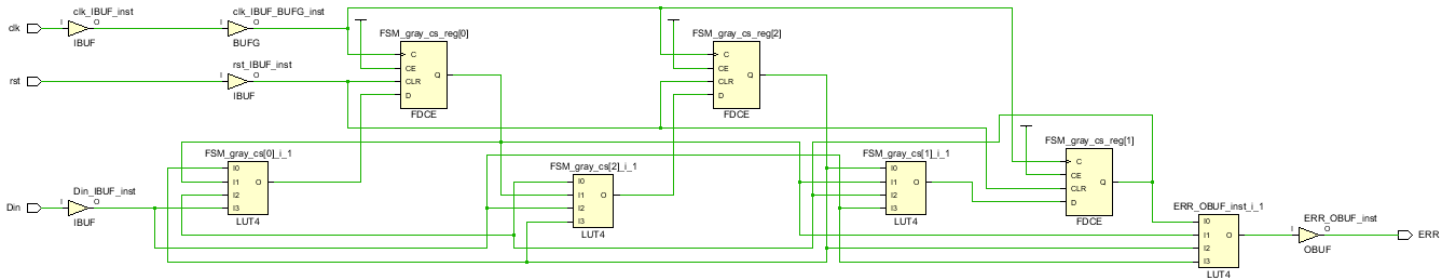


Figure 63: Q3 Synthesis (Gray)

State	New Encoding	Previous Encoding
START	000	000
D0_IS_1	001	001
D1_IS_1	011	010
D0_NOT_1	010	011
D1_NOT_1	111	100

INFO: [Synth 8-3354] encoded FSM with state register 'cs\_reg' using encoding 'gray' in module 'Sequence\_Detector\_111'

Figure 64: Q3 Encoding Report (Gray)

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Wed Aug  6 05:22:32 2025
5 // Host       : DESKTOP-DTNSAHB running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q3/Sequence_Detector_111_Gray.v}
7 // Design     : Sequence_Detector_111
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35t1cp236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* D0_IS_1 = "3'b001" *) (* D0_NOT_1 = "3'b011" *) (* D1_IS_1 = "3'b010" *)
16 (* D1_NOT_1 = "3'b100" *) (* START = "3'b000" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Sequence_Detector_111
19   (
20     clk,
21     rst,
22     Din,
23     ERR);
24   input clk;
25   input rst;
26   input Din;
27   output ERR;
28
29   wire \<const1> ;
30   wire Din;
31   wire Din_IBUF;
32   wire ERR;
33   wire ERR_OBUF;

```

Figure 65: Q3 Netlist (Gray)

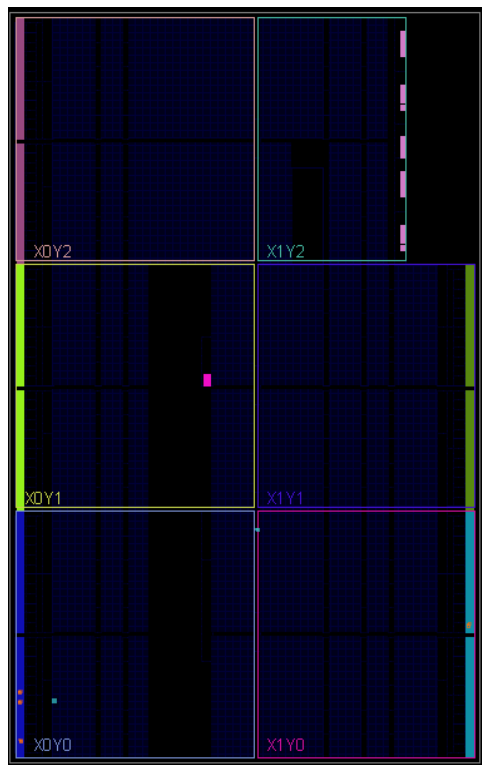


Figure 66: Q3 Device (Gray)

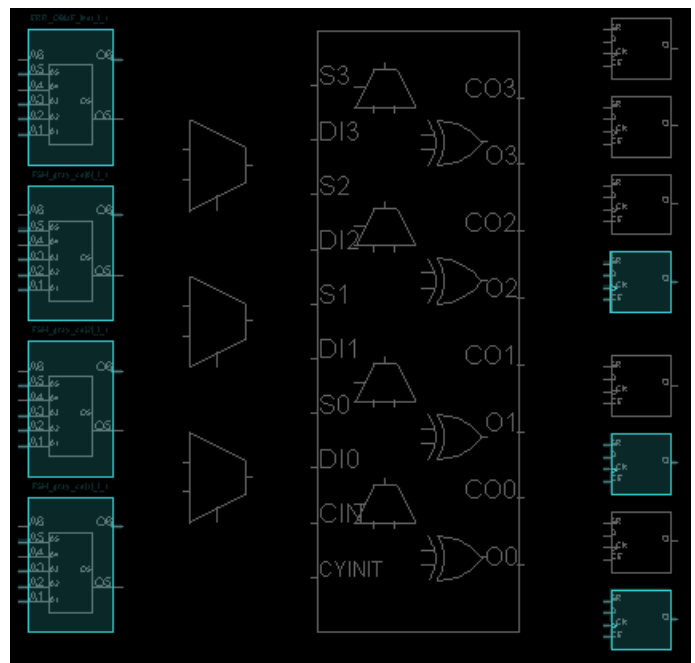


Figure 67: Q3 Zoomed Device (Gray)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.333 ns	Worst Hold Slack (WHS): 0.144 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 3	Total Number of Endpoints: 3	Total Number of Endpoints: 4

All user specified timing constraints are met.

Figure 68: Q3 Synthesis Timing Report (Gray)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.513 ns	Worst Hold Slack (WHS): 0.279 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 3	Total Number of Endpoints: 3	Total Number of Endpoints: 4

All user specified timing constraints are met.

Figure 69: Q3 Implementation Timing Report (Gray)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
N Sequence_Detector_111		4	3	4	1

Figure 70: Q3 Synthesis Utilization Report (Gray)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
N Sequence_Detector_111		4	3	1	4	3	4	1

Figure 71: Q3 Implementation Utilization Report (Gray)

## 3.2 Sequential Encoding:

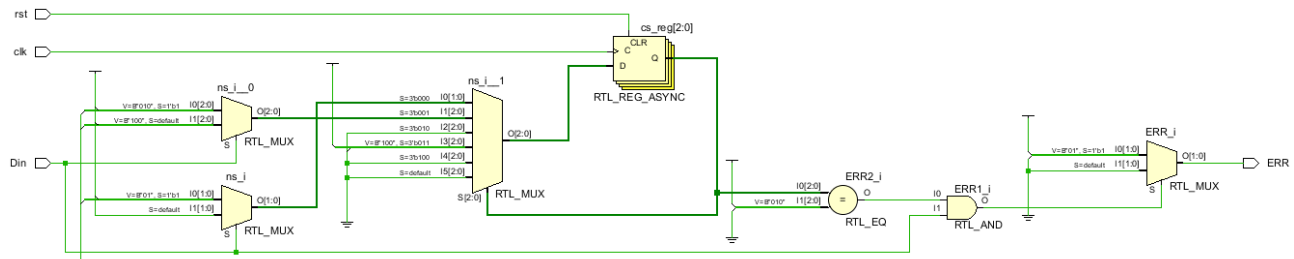


Figure 72: Q3 RTL (Sequential)

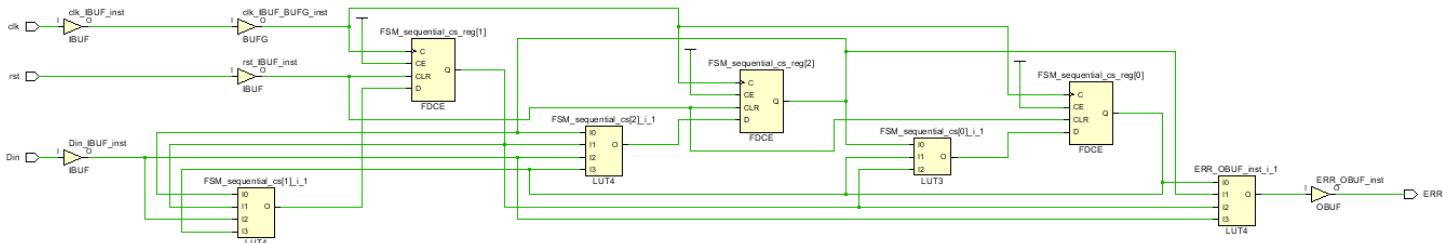


Figure 73: Q3 Synthesis (Sequential)

State	New Encoding	Previous Encoding
START	000	000
D0_IS_1	001	001
D1_IS_1	010	010
D0_NOT_1	011	011
D1_NOT_1	100	100

INFO: [Synth 8-3354] encoded FSM with state register 'cs\_reg' using encoding 'sequential' in module 'Sequence\_Detector\_111'

Figure 74: Q3 Encoding Report (Sequential)

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Wed Aug  6 06:07:30 2025
5 // Host       : DESKTOP-DTNSA8B running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Sequence_Detector_111_Seq.v}
7 // Design     : Sequence_Detector_111
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35ticpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* D0_IS_1 = "3'b001" *) (* D0_NOT_1 = "3'b011" *) (* D1_IS_1 = "3'b010" *)
16 (* D1_NOT_1 = "3'b100" *) (* START = "3'b000" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Sequence_Detector_111
19   (
20     clk,
21     rst,
22     Din,
23     ERR);
24   input clk;
25   input rst;
26   input Din;
27   output ERR;
28
29   wire \<const1> ;
30   wire Din;
31   wire Din_IBUF;
32   wire ERR;
33   wire ERR_OBUF;

```

Figure 75: Q3 Netlist (Sequential)

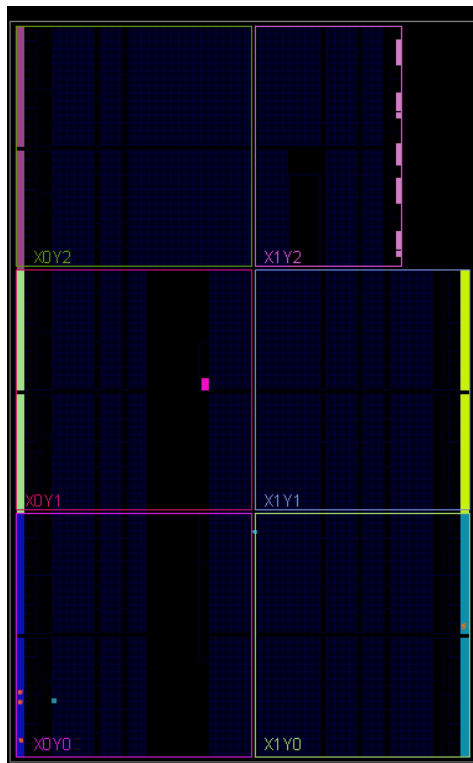


Figure 76: Q3 Device (Sequential)

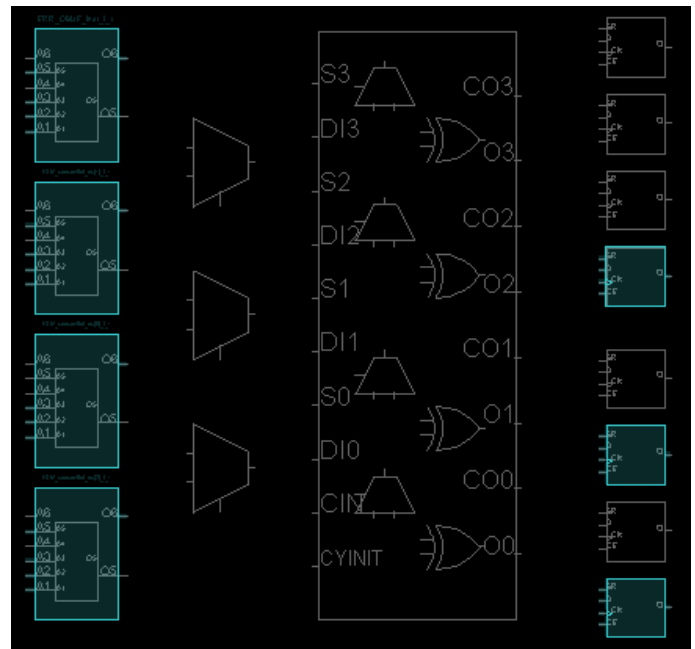


Figure 77: Q3 Zoomed Device (Sequential)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.333 ns	Worst Hold Slack (WHS): 0.144 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 3	Total Number of Endpoints: 3	Total Number of Endpoints: 4

All user specified timing constraints are met.

Figure 78: Q3 Synthesis Timing Report (Sequential)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.513 ns	Worst Hold Slack (WHS): 0.279 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 3	Total Number of Endpoints: 3	Total Number of Endpoints: 4

All user specified timing constraints are met.

Figure 79: Q3 Implementation Timing Report (Sequential)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
N Sequence_Detector_111		4	3	4	1

Figure 80: Q3 Synthesis Utilization Report (Sequential)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
N Sequence_Detector_111		4	3	1	4	3	4	1

Figure 81: Q3 Implementation Utilization Report (Sequential)

### 3.3 One-Hot Encoding:

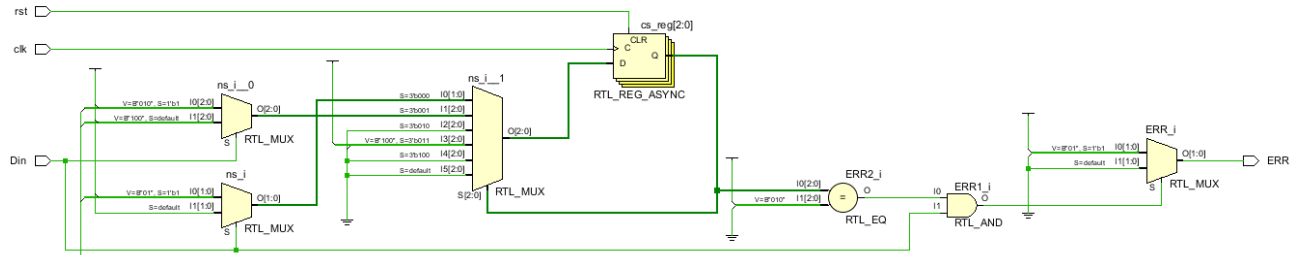


Figure 82: Q3 RTL (One-Hot)

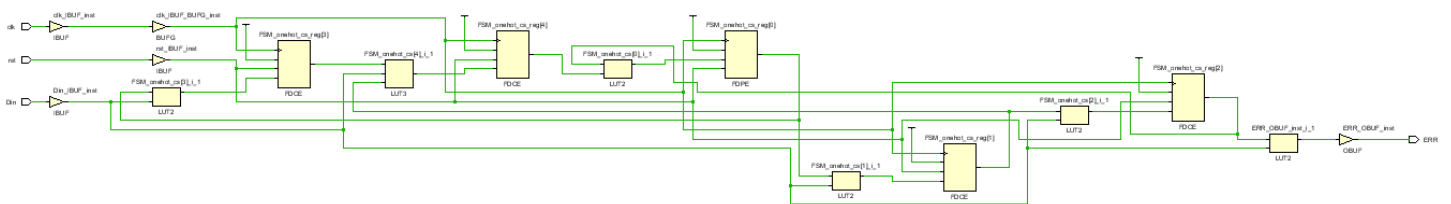


Figure 83: Q3 Synthesis (One-Hot)

State	New Encoding	Previous Encoding
START	00001	000
D0_IS_1	00010	001
D1_IS_1	00100	010
D0_NOT_1	01000	011
D1_NOT_1	10000	100

INFO: [Synth 8-3354] encoded FSM with state register 'cs\_reg' using encoding 'one-hot' in module 'Sequence\_Detector\_111'

Figure 84: Q3 Encoding Report (One-Hot)

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Wed Aug  6 05:43:27 2025
5 // Host       : DESKTOP-DTNSA8B running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q3/Sequence_Detector_111_One_Hot.v}
7 // Design     : Sequence_Detector_111
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35ticpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* D0_IS_1 = "3'b001" *) (* D0_NOT_1 = "3'b011" *) (* D1_IS_1 = "3'b010" *)
16 (* D1_NOT_1 = "3'b100" *) (* START = "3'b000" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Sequence_Detector_111
19   (clk,
20    rst,
21    Din,
22    ERR);
23   input clk;
24   input rst;
25   input Din;
26   output ERR;
27
28   wire \<const1> ;
29   wire Din;
30   wire Din_IBUF;
31   wire ERR;
32   wire ERR_OBUF;

```

Figure 85: Q3 Netlist (One-Hot)

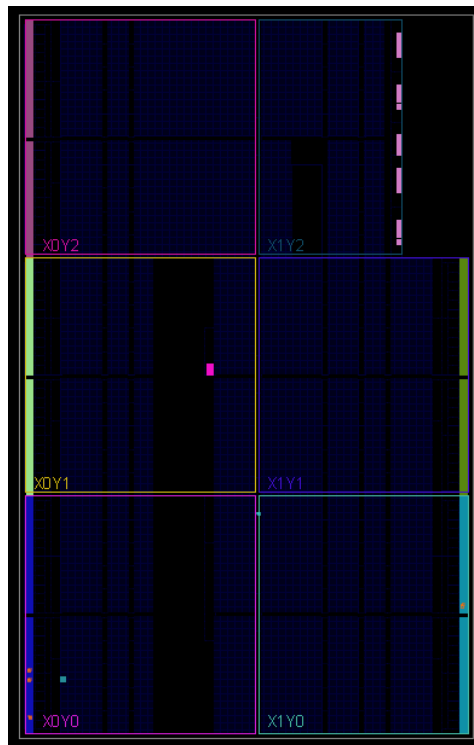


Figure 86: Q3 Device (One-Hot)

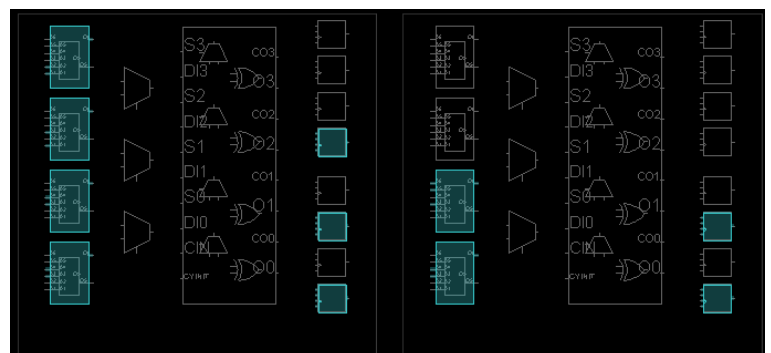


Figure 87: Q3 Zoomed Device (One-Hot)



#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.357 ns	Worst Hold Slack (WHS): 0.134 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 5	Total Number of Endpoints: 5	Total Number of Endpoints: 6

All user specified timing constraints are met.

Figure 88: Q3 Synthesis Timing Report (One-Hot)

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 8.711 ns	Worst Hold Slack (WHS): 0.145 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 5	Total Number of Endpoints: 5	Total Number of Endpoints: 6

All user specified timing constraints are met.

Figure 89: Q3 Implementation Timing Report (One-Hot)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
N Sequence_Detector_111		6	5	4	1

Figure 90: Q3 Synthesis Utilization Report (One-Hot)

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
N Sequence_Detector_111		6	5	2	6	5	4	1

Figure 91: Q3 Implementation Utilization Report (One-Hot)

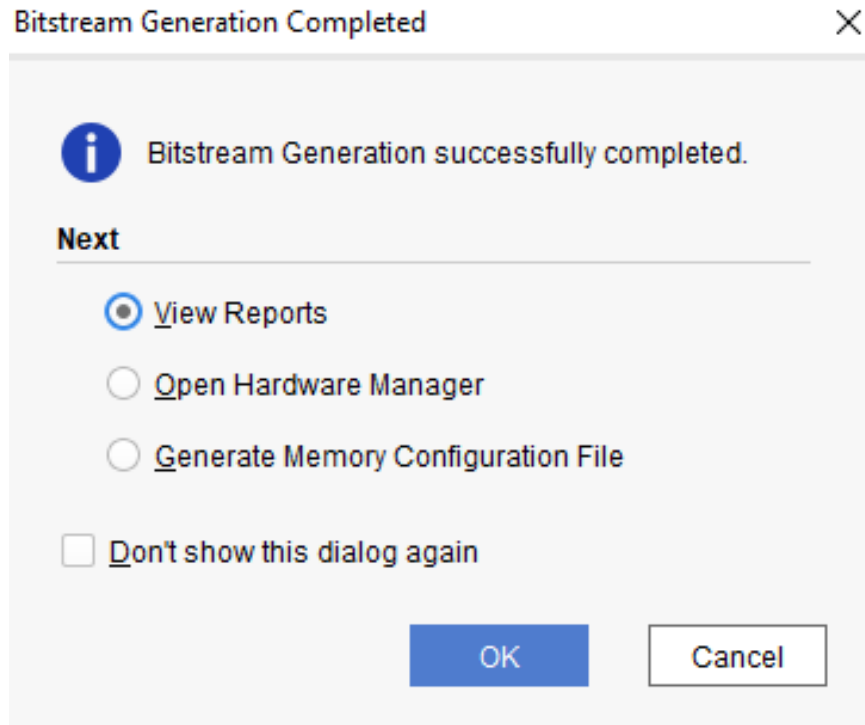


Figure 92: Q3 Bitstream



⚠ [Constraints 18-5210] No constraint will be written out.

Figure 93: Q3 Messages

- If we want to use the highest frequency, then we choose **One Hot** encoding, as it has the highest setup time slack.
- If we want minimum area, we choose **Sequential** or **Gray** encoding, as they use less components than One Hot encoding.

## 4. Pipelined RAM:

```
1  module Reg_MUX #(parameter WIDTH=18, parameter REG= "TRUE")
2  (input clk,rst,enable, input [WIDTH-1:0] D, output [WIDTH-1:0] out);
3  reg [WIDTH-1:0] D_reg;
4
5  always @(posedge clk) begin
6      if (rst)          D_reg <= 0;
7      else if (enable)  D_reg <= D;
8  end
9
10 assign out = (REG == "TRUE")? D_reg : (REG == "FALSE")? D : 0;
11
12 endmodule
```

Figure 94: Q4 Reg with MUX

```
1  module Pipelined_RAM #(parameter MEM_WIDTH = 16, MEM_DEPTH = 1024, ADDR_SIZE = 10,
2  ADDR_PIPELINE = "FALSE", DOUT_PIPELINE = "TRUE", PARITY_ENABLE = 1)
3  (input clk,rst,blk_select,wr_en,rd_en,addr_en,dout_en, input [ADDR_SIZE-1:0] addr, input [MEM_WIDTH-1:0] din,
4  output [MEM_WIDTH-1:0] dout, output parity_out);
5
6  reg [MEM_WIDTH-1:0] dout_mux;
7  wire [ADDR_SIZE-1:0] addr_mux;
8  reg [MEM_WIDTH-1:0] mem [MEM_DEPTH-1:0];
9
10 /*module Reg_MUX #(parameter WIDTH=18, parameter REG= "TRUE")
11 (input clk,rst,enable, input [WIDTH-1:0] D, output [WIDTH-1:0] out);*/
12 Reg_MUX #(ADDR_SIZE,ADDR_PIPELINE) addr_REG(clk,rst,addr_en,addr, addr_mux);
13 Reg_MUX #(MEM_WIDTH,DOUT_PIPELINE) dout_REG(clk,rst,dout_en,dout_mux, dout);
14
15 always @(posedge clk) begin
16     if (rst)          dout_mux <= 0;
17     else if (blk_select) begin
18         if (wr_en)      mem [addr_mux] <= din;
19         if (rd_en)      dout_mux <= mem [addr_mux];
20     end
21 end
22
23 assign parity_out = PARITY_ENABLE? ^dout : 0;
24
25 endmodule
```

Figure 95: Q4 Code

```

1 module Pipelined_RAM_tb ();
2 parameter MEM_WIDTH = 16, MEM_DEPTH = 1024, ADDR_SIZE = 10,
3 ADDR_PIPELINE = "FALSE", DOUT_PIPELINE = "TRUE", PARITY_ENABLE = 1;
4 reg clk,rst,blk_select,wr_en,rd_en,addr_en,dout_en;
5 reg [ADDR_SIZE-1:0] addr;
6 reg [MEM_WIDTH-1:0] din;
7 wire [MEM_WIDTH-1:0] dout;
8 wire parity_out;
9
10 /*module Pipelined_RAM #(parameter MEM_WIDTH = 16, MEM_DEPTH = 1024, ADDR_SIZE = 10,
11 ADDR_PIPELINE = "FALSE", DOUT_PIPELINE = "TRUE", PARITY_ENABLE = 1)
12 (input clk,rst,blk_select,wr_en,rd_en,addr_en,dout_en, input [ADDR_SIZE-1:0] addr, input [MEM_WIDTH-1:0] din,
13 output [MEM_WIDTH-1:0] dout, output parity_out);*/
14 Pipelined_RAM #(MEM_WIDTH,MEM_DEPTH,ADDR_SIZE,ADDR_PIPELINE,DOUT_PIPELINE,PARITY_ENABLE)
15 DUT(clk,rst,blk_select,wr_en,rd_en,addr_en,dout_en,addr,din,dout,parity_out);
16
17 initial begin
18     clk=0;
19     forever #5 clk=~clk;
20 end
21
22 initial begin
23     $readmemh ("mem.dat",DUT.mem);
24     rst = 1; blk_select=$random; wr_en=$random; rd_en=$random; addr_en=$random; dout_en=$random; addr=$random; din=$random;
25     @(negedge clk);
26     if (dout || parity_out) begin
27         $display("Error in Reset");
28         $stop;
29     end
30
31     rst = 0; blk_select=1; addr_en=0; dout_en=1;
32     repeat (20) begin
33         wr_en=$random; rd_en=$random; addr=$urandom_range(1008,1023); din=$random;
34         @(negedge clk);
35         @(negedge clk);
36     end
37
38     dout_en=0; wr_en=$random; rd_en=1; addr=$urandom_range(1008,1023); din=$random;
39     @(negedge clk);
40
41     blk_select=0; wr_en=1; rd_en=1; addr=$urandom_range(1008,1023); din=5;
42     @(negedge clk);
43     $stop;
44 end
45 endmodule

```

Figure 96: Q4 Testbench

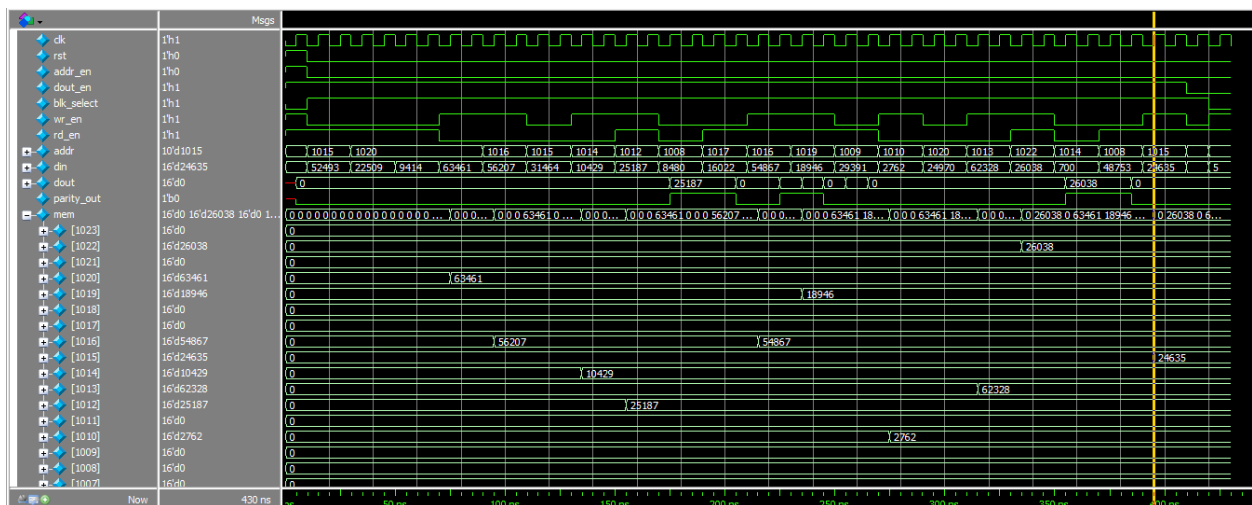


Figure 97: Q4 Wave

```

1  vlib work
2  vlog Pipelined_RAM.v Pipelined_RAM_tb.v
3  vsim -voptargs=+acc Pipelined_RAM_tb
4  add wave *
5  add wave -position insertpoint \
6  sim:/Pipelined_RAM_tb/DUT/mem
7  run -all
8  #quit -sim

```

Figure 98: Q4 DO File

The screenshot shows the Xilinx IDE interface. The main editor displays the Verilog code for the module `Pipelined_RAM`. The code includes parameters for `MEM_WIDTH` and `MEM_DEPTH`, and defines signals for `ADDR_PIPELINE`, `DOUT_PIPELINE`, `clk`, `rst`, `blk_select`, `wr_en`, `rd_en`, `addr_en`, `dout`, `output`, `parity_out`, `dout_mux`, `addr_mux`, and `mem`. The right-hand pane shows the 'Lint Summary' table, which lists the linting results for the module.

Name	Count
Resolved (verified, fixed, waived)	4
Info	4
condition_const	2
multi_ports_in_single_line	2

The bottom of the IDE shows the 'Lint Checks' section, which includes a filter and a table of linting results. The table has columns for Severity, Status, Check, Alias, Message, Module, and Category. The current status is 'Total: 0 Selected: 0'.

Figure 99: Q4 Linting

```

1  ## This file is a general .xdc for the Basys3 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  ## Clock signal
7  set_property -dict { PACKAGE_PIN W5 IOSTANDARD LVCMOS33 } [get_ports clk]
8  create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
9
10
11  ## Configuration options, can be used for all designs
12  set_property CONFIG_VOLTAGE 3.3 [current_design]
13  set_property CFGBVS VCC0 [current_design]
14
15  ## SPI configuration mode options for QSPI boot, can be used for all designs
16  set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
17  set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
18  set_property CONFIG_MODE SPIx4 [current_design]

```

Figure 100: Q4 Constraint File

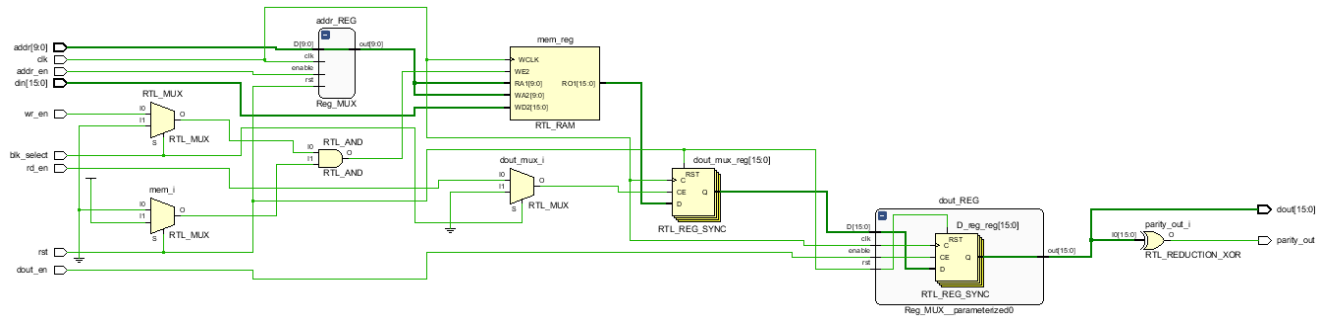


Figure 101: Q4 RTL

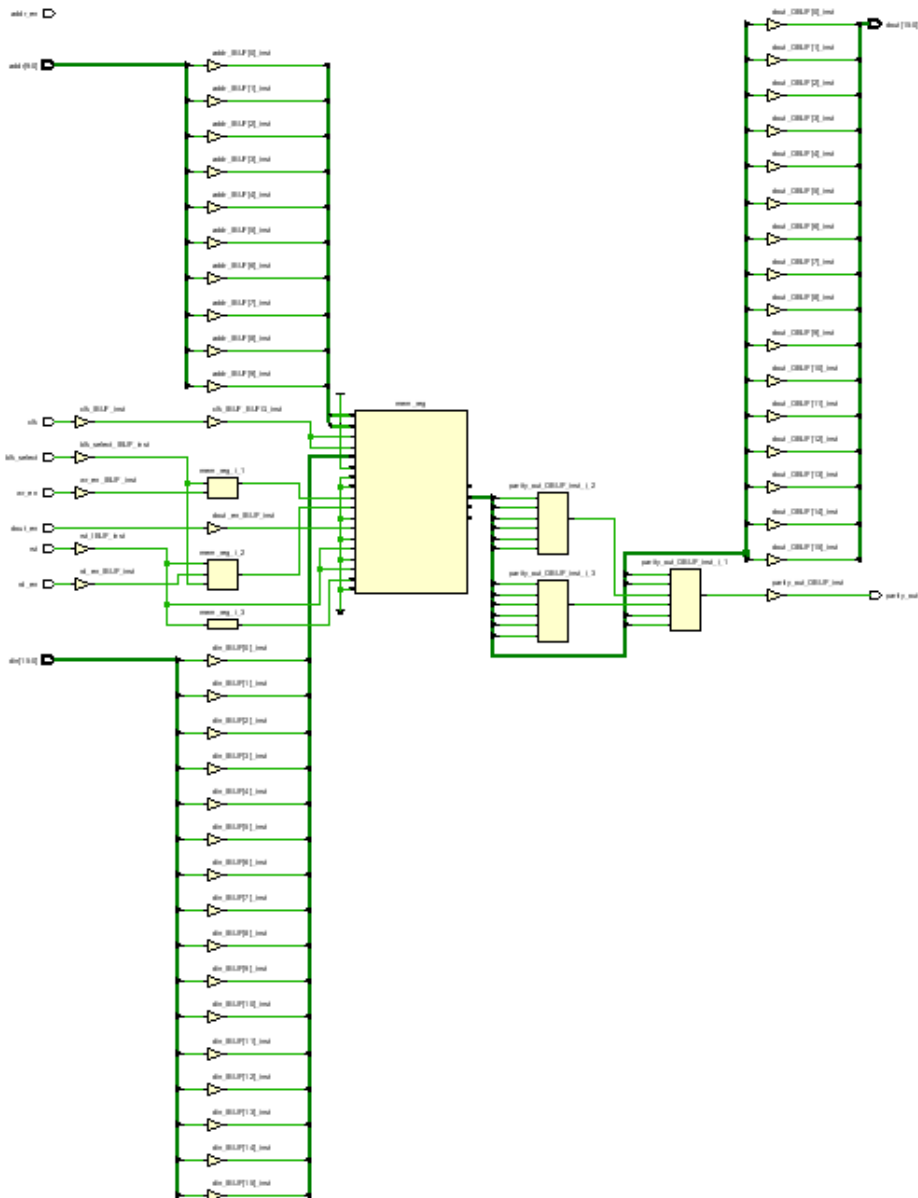


Figure 102: Q4 Synthesis

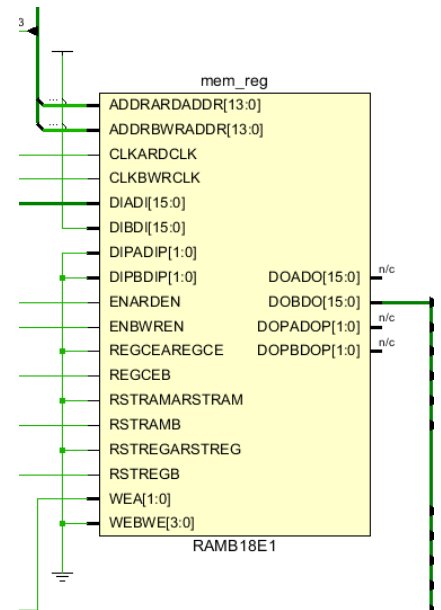


Figure 103: Q4 RAM Synthesis

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Wed Aug  6 10:10:45 2025
5 // Host       : DESKTOP-DTNSAHB running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q4/Pipelined_RAM_Netlist.v}
7 // Design     : Pipelined_RAM
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35ticpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* ADDR_PIPELINE = "FALSE" *) (* ADDR_SIZE = "10" *) (* DOUT_PIPELINE = "TRUE" *)
16 (* MEM_DEPTH = "1024" *) (* MEM_WIDTH = "16" *) (* PARITY_ENABLE = "1" *)
17 (* STRUCTURAL_NETLIST = "yes" *)
18 module Pipelined_RAM
19   (clk,
20    rst,
21    blk_select,
22    wr_en,
23    rd_en,
24    addr_en,
25    dout_en,
26    addr,
27    din,
28    dout,
29    parity_out);
30   input clk;
31   input rst;
32   input blk_select;

```

Figure 104: Q4 Netlist

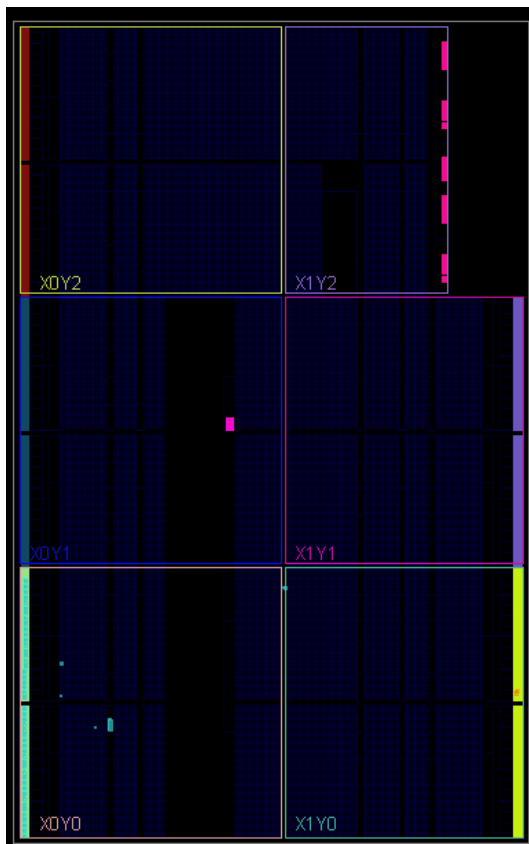


Figure 105: Q4 Device

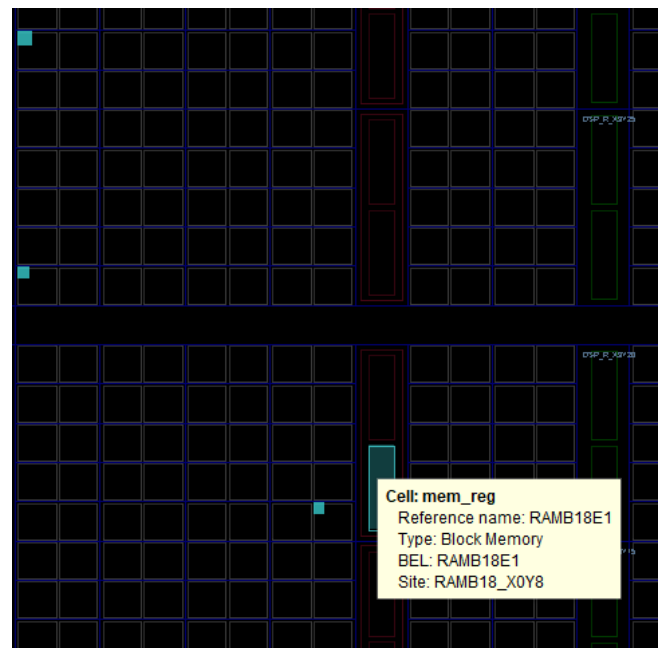


Figure 106: Q4 Zoomed Device

Name	1	Slice LUTs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
<b>N</b> Pipelined_RAM		6	0.5	49	1

Figure 107: Q4 Synthesis Utilization Report

Name	1	Slice LUTs (20800)	Slice (8150)	LUT as Logic (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
<b>N</b> Pipelined_RAM		7	3	7	0.5	49	1

Figure 108: Q4 Implementation Utilization Report

Tcl Console
Messages
x
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Utilization
?
\_

Q
H
A
F
M
W
Warning (9)
Info (207)
Status (444)
Show All

Synthesis (5 warnings)

[Synth 8-3331] design Reg\_MUX has unconnected port clk (3 more like this)
[Constraints 18-5210] No constraint will be written out.

Implementation (2 warnings)

Place Design (1 warning)

[Place 46-29] place\_design is not in timing mode. Skip physical synthesis in placer

Route Design (1 warning)

DRC (1 warning)

Pin Planning (1 warning)

[DRC CFGBVS-7] CONFIG\_VOLTAGE with Config Bank VCCO: The CONFIG\_MODE property of current\_design specifies a configuration mode (SPIx4) that uses pins in bank 14. I/O standards used in this bank have a voltage requirement of 1.80. However, the CONFIG\_VOLTAGE for current\_design is set to 3.3. Ensure that your configuration voltage is compatible with the I/O standards in banks used by your configuration mode. Refer to device configuration user guide for more information. Pins used by config mode: D18 (IO\_L1P\_T0\_D00\_MOSI\_14), D19 (IO\_L1N\_T0\_D01\_DIN\_14), G18 (IO\_L2P\_T0\_D02\_14), F18 (IO\_L2N\_T0\_D03\_14), E18 (IO\_L3P\_T0\_DQS\_PUDC\_B\_14), and K19 (IO\_L6P\_T0\_FCS\_B\_14)

Figure 109: Q4 Messages



## 5. FIFO:

```
1 module FIFO #(parameter FIFO_WIDTH = 16, FIFO_DEPTH = 512 /* 2^9 */)
2 (input clk_a,clk_b,rst,wen_a,ren_b, input [FIFO_WIDTH-1:0] din_a, output reg [FIFO_WIDTH-1:0] dout_b, output full,empty);
3 reg [8:0] addr_wr,          //9 bit Write Counter (Pointer) (It's the address in the memory where we'll be writing)
4     addr_rd,              //9 bit Read Counter (Pointer) (It's the address in the memory where we'll be reading)
5     size;                 //9 bit counter to Keep track of the number of data in memory to know if it's full or empty
6
7 reg [FIFO_WIDTH-1:0] mem [FIFO_DEPTH-1:0];
8
9 //Write Logic
10 always @(posedge clk_a) begin
11     if (rst)                {addr_wr, size} <= 0;
12     else if (!full && wen_a) begin
13         mem [addr_wr] <= din_a;
14         addr_wr <= addr_wr + 1;
15         size <= size + 1;
16     end
17 end
18
19 //Read Logic
20 always @(posedge clk_b) begin
21     if (rst)                {dout_b, addr_rd, size} <= 0;
22     else if (!empty && ren_b) begin
23         dout_b <= mem [addr_rd];
24         addr_rd <= addr_rd + 1;
25         size <= size - 1;
26     end
27 end
28
29 assign full = size == FIFO_DEPTH? 1 : 0;
30 assign empty = !size? 1 : 0;
31
32 endmodule
```

Figure 110: Q5 Code

```
1 module FIFO_tb ();
2 parameter FIFO_WIDTH = 16, FIFO_DEPTH = 512;
3 reg clk_a,clk_b,rst,wen_a,ren_b;
4 reg [FIFO_WIDTH-1:0] din_a;
5 wire [FIFO_WIDTH-1:0] dout_b;
6 wire full,empty;
7
8 /*module FIFO #(parameter FIFO_WIDTH = 16, FIFO_DEPTH = 512)
9 (input clk_a,clk_b,rst,wen_a,ren_b, input [FIFO_WIDTH-1:0] din_a, output reg [FIFO_WIDTH-1:0] dout_b, output full,empty);*/
10 FIFO #(FIFO_WIDTH,FIFO_DEPTH) DUT(clk_a,clk_b,rst,wen_a,ren_b,din_a,dout_b,full,empty);
11
12 initial begin
13     clk_a = 0;
14     clk_b = 0;
15     forever begin
16         #5 clk_a = ~clk_a;
17         #7 clk_b = ~clk_b;
18     end
19 end
20
21 initial begin
22     $readmemh ("mem.dat", DUT.mem);
23     rst = 1; wen_a=$random; ren_b=$random; din_a=$random;
24     @(negedge clk_b);
25     if (dout_b || full || !empty) begin
26         $display("Error in Reset!!");
27         $stop;
28     end
29
30     rst = 0;
31     repeat (30) begin
32         wen_a=$random; ren_b=$random; din_a=$random_range(1,100);
33         @(negedge clk_b);
34     end
35     $stop;
36 end
37 endmodule
```

Figure 111: Q5 Testbench

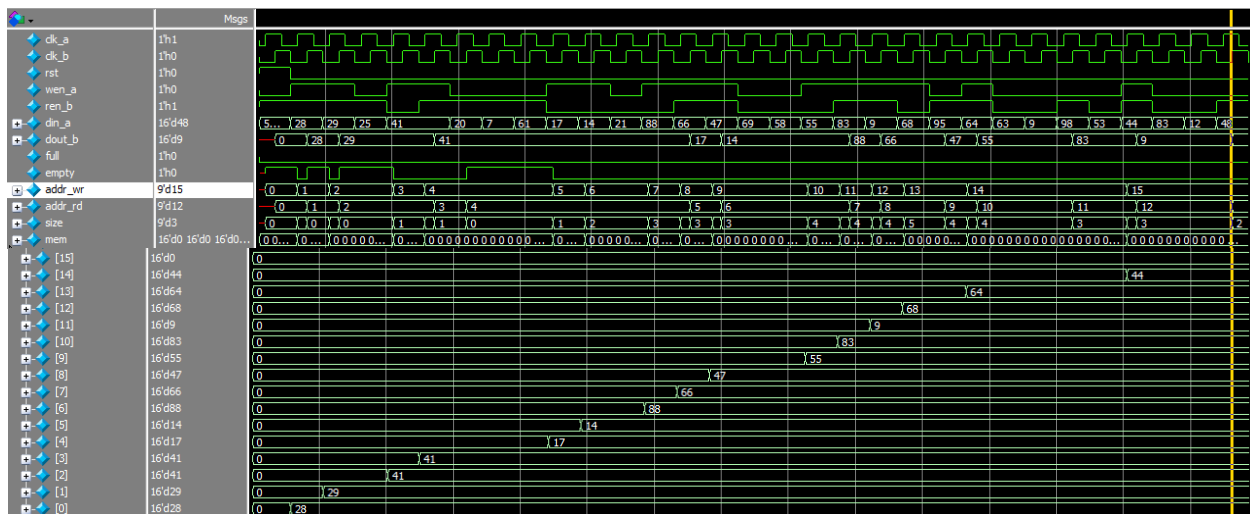


Figure 112: Q5 Wave

Lint Summary

Name	Count
Open (uninspected, pending, bug)	1
Warning	1
Resolved (verified, fixed, waived)	3
Info	3

Lint Checks

Severity	Status	Check	Alias	Message	Module	Category
Warning	?	multi_driven_signal		Net has multiple drivers. Signal size, Module FIFO, File D:/GG/FIFO.v... FIFO	FIFO	Simulation

Figure 113: Q5 Linting

```

1 vlib work
2 vlog FIFO.v FIFO_tb.v
3 vsim -voptargs=+acc FIFO_tb
4 add wave *
5 add wave -position insertpoint \
6 sim:/FIFO_tb/DUT/addr_wr \
7 sim:/FIFO_tb/DUT/addr_rd \
8 sim:/FIFO_tb/DUT/size \
9 sim:/FIFO_tb/DUT/mem
10 run -all
11 #quit -sim

```

Figure 114: Q5 DO File

```

1  ## This file is a general .xdc for the Basys3 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  ## Clock signal
7  set_property -dict { PACKAGE_PIN W5 IOSTANDARD LVCMOS33 } [get_ports clk_a]
8  create_clock -add -name sys_clk_a_pin -period 10.00 -waveform {0 5} [get_ports clk_a]
9  create_clock -add -name sys_clk_b_pin -period 10.00 -waveform {0 5} [get_ports clk_b]
10
11
12  ## Configuration options, can be used for all designs
13  set_property CONFIG_VOLTAGE 3.3 [current_design]
14  set_property CFGBVS VCC0 [current_design]
15
16  ## SPI configuration mode options for QSPI boot, can be used for all designs
17  set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
18  set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
19  set_property CONFIG_MODE SPIx4 [current_design]

```

Figure 115: Q5 Constraint File

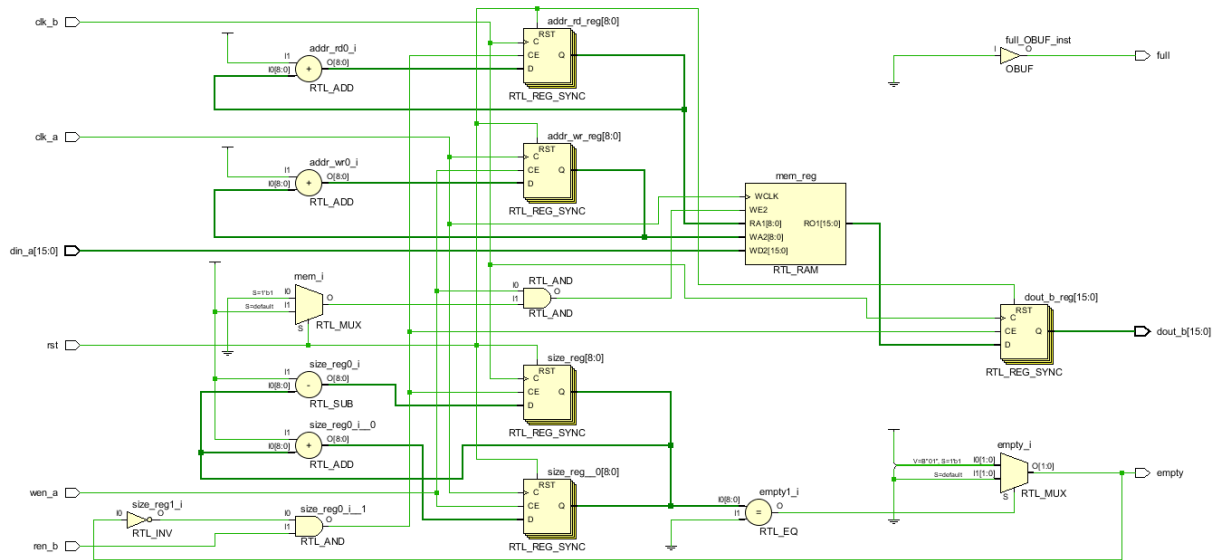


Figure 116: Q5 RTL

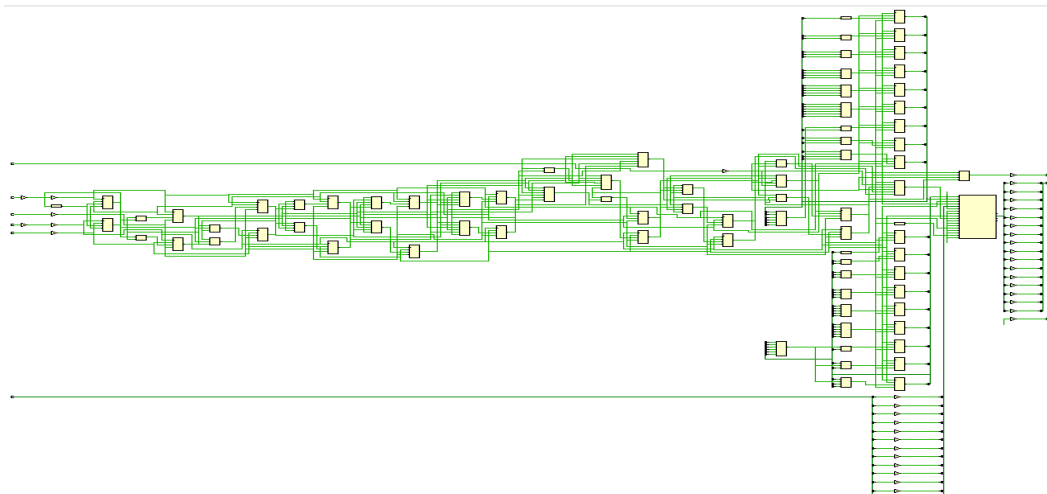


Figure 117: Q5 Synthesis

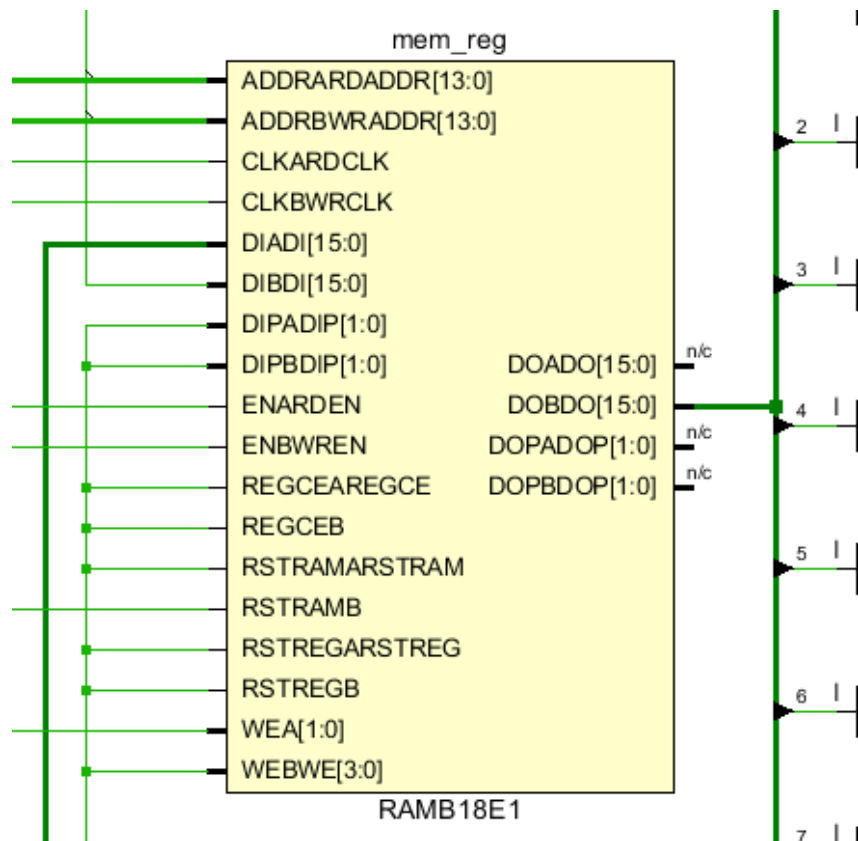


Figure 118: Q5 RAM Synthesis

```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Thu Aug  7 03:11:50 2025
5 // Host       : DESKTOP-DTNSAHB running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q5/FIFO_Netlist.v}
7 // Design     : FIFO
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35ticpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* FIFO_DEPTH = "512" *) (* FIFO_WIDTH = "16" *)
16 (* STRUCTURAL_NETLIST = "yes" *)
17 module FIFO
18   (
19     clk_a,
20     clk_b,
21     rst,
22     wen_a,
23     ren_b,
24     din_a,
25     dout_b,
26     full,
27     empty);
28   input clk_a;
29   input clk_b;
30   input rst;
31   input wen_a;
32   input ren_b;
33   input [15:0]din_a;

```

Figure 119: Q5 Netlist

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.504 ns	Worst Hold Slack (WHS): 0.232- ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 1.724- ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 15	Number of Failing Endpoints: 0
Total Number of Endpoints: 73	Total Number of Endpoints: 73	Total Number of Endpoints: 40
Timing constraints are not met.		

Figure 121: Q5 Synthesis Utilization Report

Figure 122: Q5 Messages

44 | Page

## 6. IP ALSU:

What changed in the RTL code due to using IPs:

```
8  wire [3 : 0] adder_out;
9  generate
10     if (FULL_ADDER == "ON")
11         c_addsub_1 IP_Full_Adder(
12             .A(A_reg),          // input wire [2 : 0] A
13             .B(B_reg),          // input wire [2 : 0] B
14             .C_IN(cin_reg),     // input wire C_IN
15             .S(adder_out)       // output wire [3 : 0] S
16         );
17     else if (FULL_ADDER == "OFF")
18         c_addsub_1 IP_Half_Adder(
19             .A(A_reg),          // input wire [2 : 0] A
20             .B(B_reg),          // input wire [2 : 0] B
21             .C_IN(0),           // input wire C_IN
22             .S(adder_out)       // output wire [3 : 0] S
23         );
24 endgenerate
25
26 wire [5 : 0] multiply_out;
27 mult_gen_0 IP_multiplier (
28     .A(A_reg), // input wire [2 : 0] A
29     .B(B_reg), // input wire [2 : 0] B
30     .P(multiply_out) // output wire [5 : 0] P
31 );
```

Figure 123: Q6 Code (1)

```
74 3'b010: begin //Addition
75     if (red_op_A_reg || red_op_B_reg) {out, leds} <= {6'b0, ~leds}; //Invalid
76     else {out, leds} <= {2'b0, adder_out, 16'b0};
77 end
78
79 3'b011: begin //Multiplication
80     if (red_op_A_reg || red_op_B_reg) {out, leds} <= {6'b0, ~leds}; //Invalid
81     else {out, leds} <= {multiply_out, 16'b0};
82 end
```

Figure 124: Q6 Code (2)

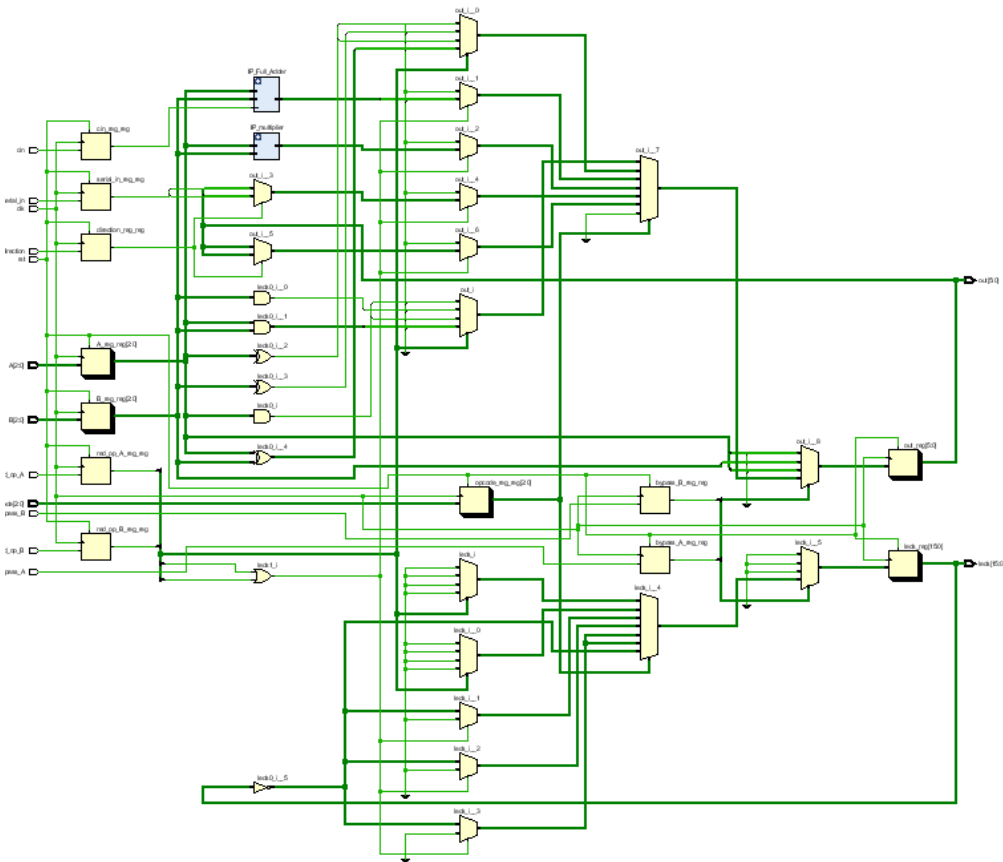


Figure 125: Q6 RTL

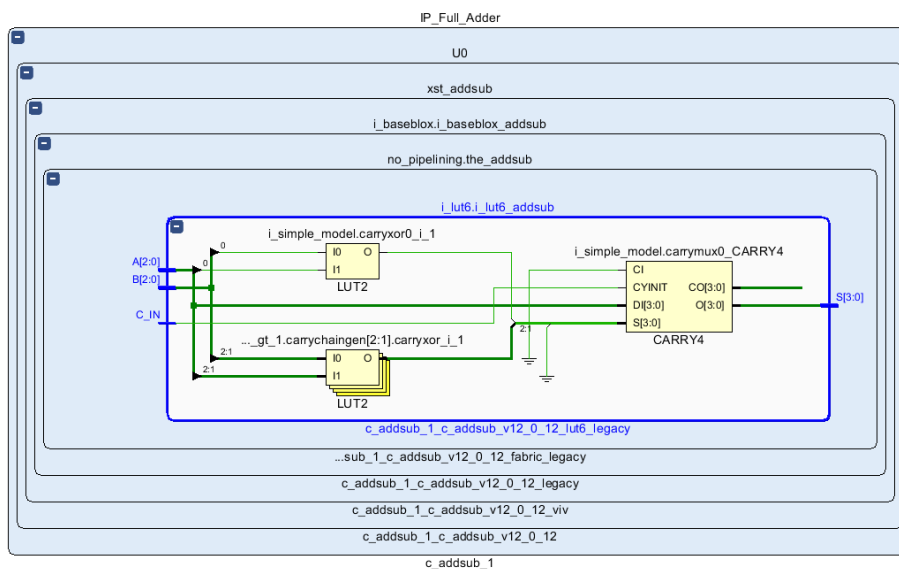


Figure 126: Q6 IP Adder RTL

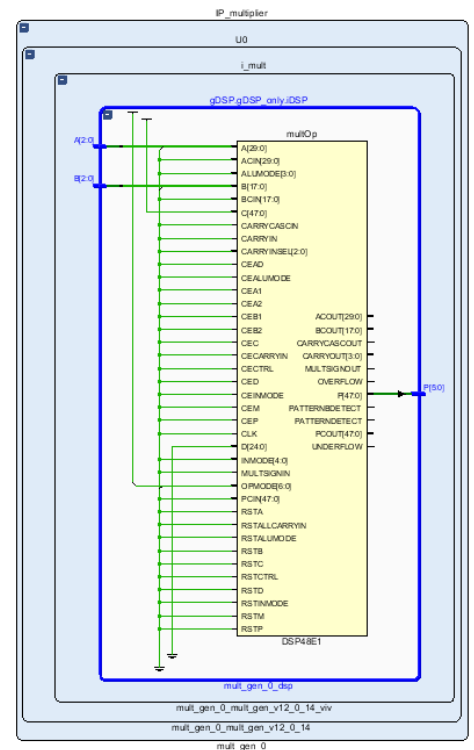


Figure 127: Q6 IP Multiplier RTL

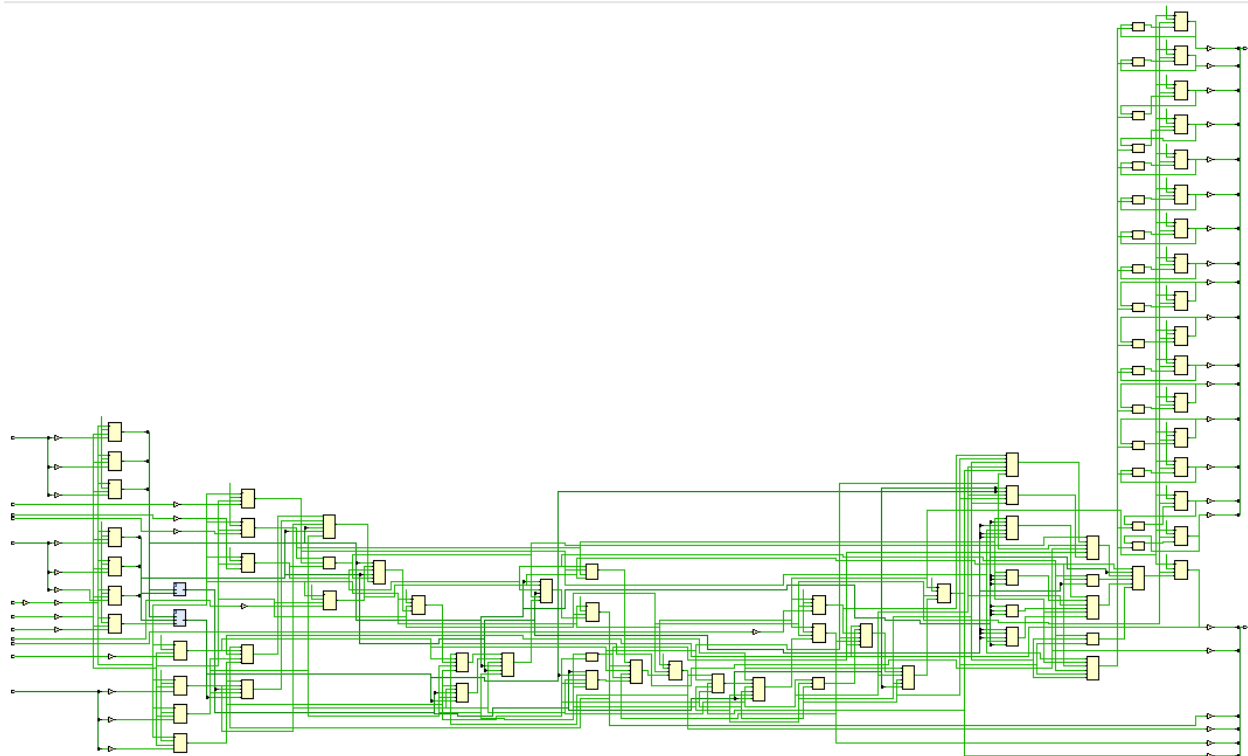


Figure 128: Q6 Synthesis

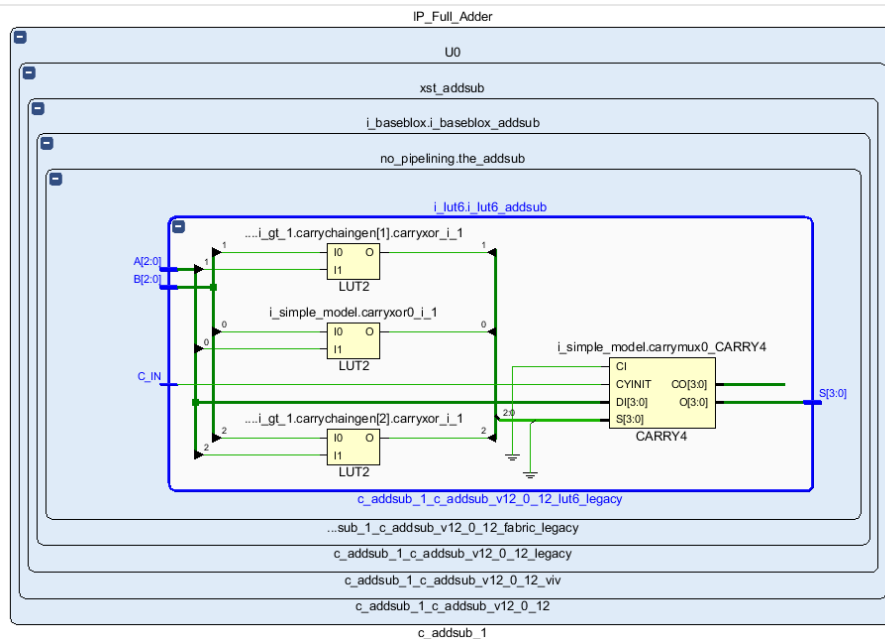


Figure 129: Q6 IP Adder Synthesis

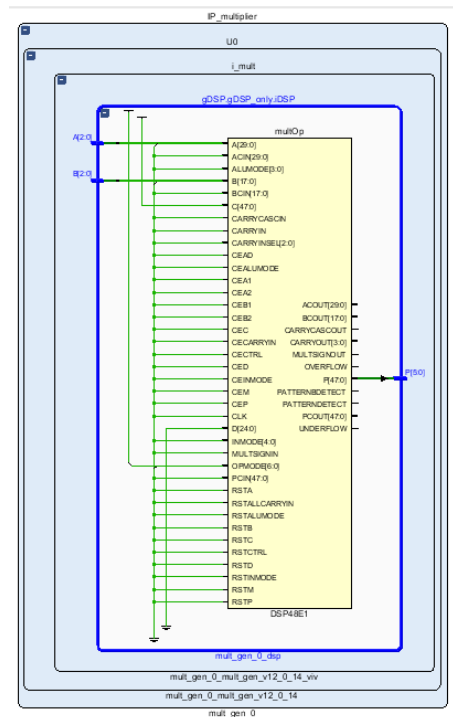


Figure 130: Q6 IP Multiplier Synthesis



```

1 // Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2 // -----
3 // Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
4 // Date       : Thu Aug  7 04:24:24 2025
5 // Host       : DESKTOP-DTNSAHB running 64-bit major release (build 9200)
6 // Command    : write_verilog {D:/College Projects/Digital_Course_Waseem/Vv/Q6/ALSY_IP_Netlist.v}
7 // Design     : ALSU_IP
8 // Purpose    : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
9 //              IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
10 //              design files.
11 // Device     : xc7a35t1cpg236-1L
12 // -----
13 `timescale 1 ps / 1 ps
14
15 (* FULL_ADDER = "ON" *) (* INPUT_PRIORITY = "A" *)
16 (* STRUCTURAL_NETLIST = "yes" *)
17 module ALSU_IP
18   (clk,
19    rst,
20    cin,
21    serial_in,
22    red_op_A,
23    red_op_B,
24    bypass_A,
25    bypass_B,
26    direction,
27    A,
28    B,
29    opcode,
30    out,
31    leds);
32   input clk;

```

Figure 131: Q6 Netlist

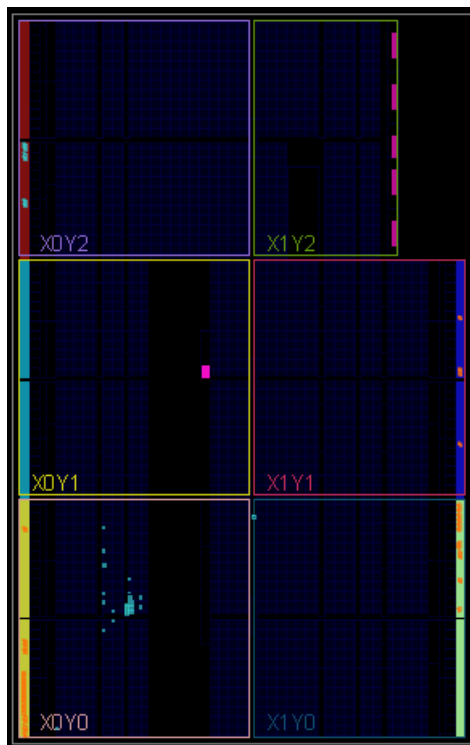


Figure 132: Q6 Device

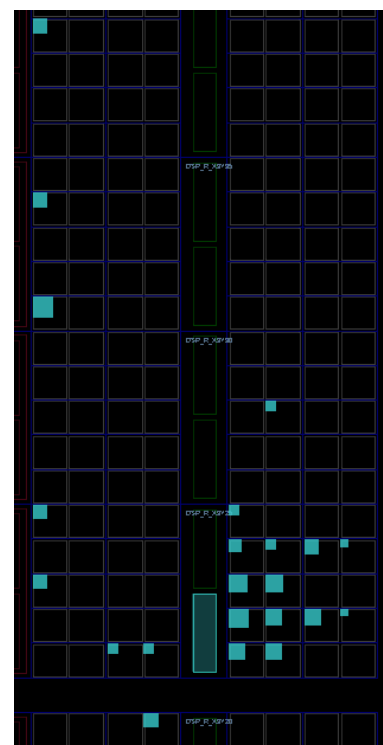


Figure 133: Q6 Zoomed Device

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 2.481 ns	Worst Hold Slack (WHS): 0.206 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 22	Total Number of Endpoints: 22	Total Number of Endpoints: 39

All user specified timing constraints are met.

Figure 134: Q6 Synthesis Timing Report

#### Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 2.123 ns	Worst Hold Slack (WHS): 0.283 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 28	Total Number of Endpoints: 28	Total Number of Endpoints: 45

All user specified timing constraints are met.

Figure 135: Q6 Implementation Timing Report

Name	1	Slice LUTs (20800)	Slice Registers (41600)	DSP s (90)	Bonded IOB (106)	BUFGCTRL (32)
▼ N ALSU_IP		36	38	1	40	1
> IP_Full_Adder (c_adds...		3	0	0	0	0
> IP_multiplier (mult_ge...		0	0	1	0	0

Figure 136: Q6 Synthesis Utilization Report

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	DSP s (90)	Bonded IOB (106)	BUFGCTRL (32)
▼ N ALSU_IP		36	44	24	36	12	1	40	1
> IP_Full_Adder (c_adds...		3	0	1	3	0	0	0	0
> IP_multiplier (mult_ge...		0	0	0	0	0	1	0	0

Figure 137: Q6 Implementation Utilization Report

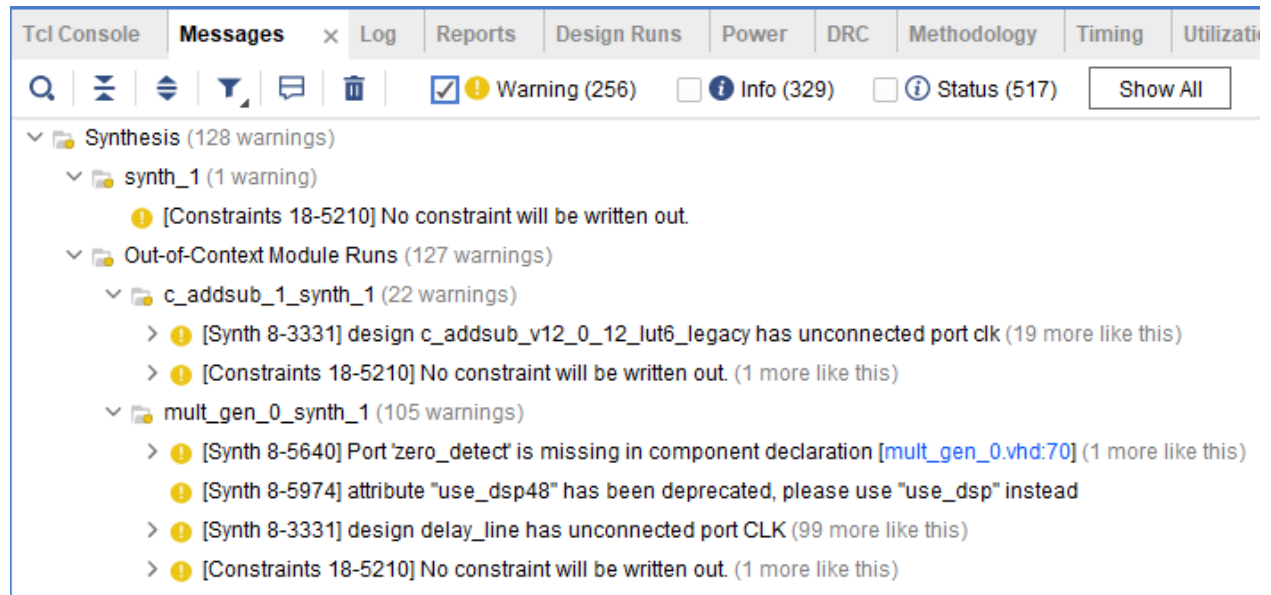


Figure 138: Q6 Messages