Experiment - 9: Single and Dual Port RAM and ROM Using Quartus Prime

NAME: Jayakrishnan Menon

REG NO: 22BEC1205

DATE: 12/03/2025

Aim:
Write a Verilog RTL code for Simulation and Implementation of Single and Dual Port RAM and ROM. Also, perform the simulation of the circuit using Quartus Prime and Model Sim. Finally, implement these circuits on the FPGA kit, "5CSXFC6D6F31C6N"

Software Required: Quartus Prime**,** ModelSim

Haraware Required: Altera Cyclone V 5CSXFC6D6F31C6N

Procedure:

- Open Quartus Prime 21.1.
- Go to File -> New Project wizard -> select Source folder & type file name -> Next.
- Select Empty Project in Project type -> Next.
- Click on Next in Add Files dialog box.
- Select "Cyclone V SX Extended Features" in Device Family.
- In available devices, select the one ending with "31C6" -> Next.
- Select the tool name as "ModelSim" and Format as "VerilogHDL" in simulation.
- Click on Finish. The project is now created.
- In the Task window, select RTL simulation and run, this would open the ModelSim window.
- Simulate the full adder as you would do using ModelSim by forcing the input values.
- In Compilation -> select compile design.
- Go to Assignments tab -> Pin planner -> Give the location for each i/o pin.
- Go to Hardware Setup -> Select USB.
- Change the file to Fulladder.v in the program/configure option and select Start.

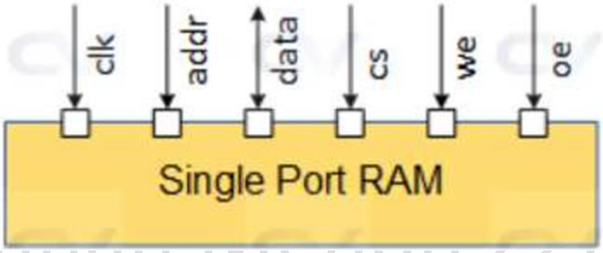
Theory

• A single port memory is a type of memory that can be accessed by only one device or process at a time. In this type of memory, data can be written and read from the same port.

• It is generally used in applications where only one processor is used, and the memory is no required to be accessed by multiple processors simultaneously.

• A single-port RAM (Random Access Memory) is a type of digital memory component that allows data to be read from and written to a single memory location (address) at a time. It is a simple form of memory that provides a basic storage mechanism for digital systems.

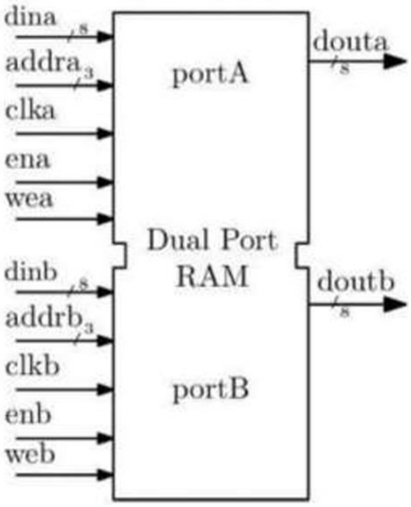
• Each memory location in a single-port RAM can store a fixed number of bits (usually a power of 2, such as 8, 16, 32, etc.).



• The recent technology has developed dual port memories. Now it is possible to access the same address locations through two ports.

• Dual port memories have simplified many problems in designing digital systems.

• Both ROM and RAM can be of dual port. The block diagram of a true dual port RAM is shown below.

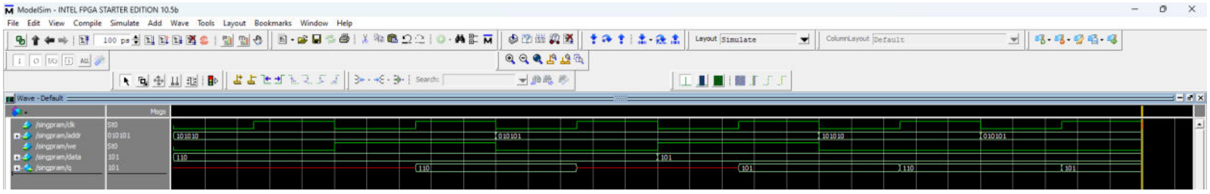


Source code and Outputs:

Single Port RAM:

```
module singpram(
input [2:0] data,
input [5:0] addr,
input we, clk,
output [2:0] q
);

reg [2:0] ram[63:0];
reg [5:0] addr_reg;

always @ (posedge clk)begin
// Write
if (we)
ram[addr] <= data;
addr_reg <= addr;
end

assign q = ram[addr_reg];

endmodule
```

Output:



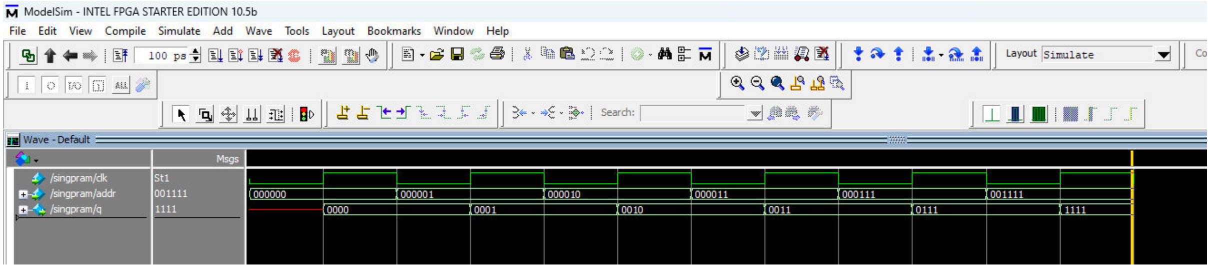
ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
100 ps
Search:
Layout Simulate
ColumnLayout Default
Wave - Default
Msgs
101010
010101
101010
010101
110
101
110
101
110
101

/singpram/clk St0
/singpram/addr 010101
/singpram/we St0
/singpram/data 101
/singpram/q 101

Single Port ROM:

```
module singpram(
input [5:0] addr,
input clk,
output [3:0] q
);
reg [3:0] rom[63:0];
reg [5:0] addr_reg;

initial begin
rom[0]=4'b0000;
rom[1]=4'b0001;
rom[2]=4'b0010;
rom[3]=4'b0011;
rom[4]=4'b0100;
rom[5]=4'b0101;
rom[6]=4'b0110;
rom[7]=4'b0111;
rom[8]=4'b1000;
rom[9]=4'b1001;
rom[10]=4'b1010;
rom[11]=4'b1011;
rom[12]=4'b1100;
rom[13]=4'b1101;
rom[14]=4'b1110;
rom[15]=4'b1111;
end

always @ (posedge clk)begin
addr_reg <= addr;
end

assign q = rom[addr_reg];
endmodule
```

Output:

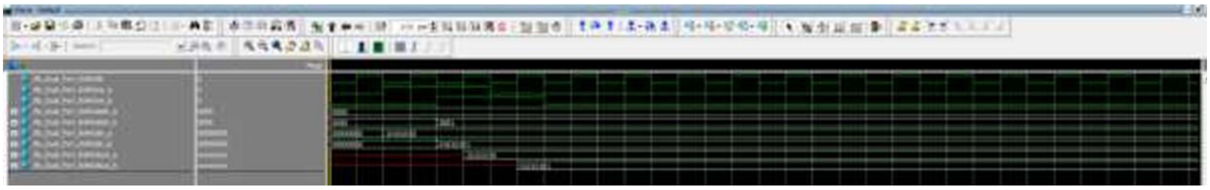

ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
100 ps
Search:
Layout Simulate
Wave - Default
Msgs

```
/singpram/clk    St1
/singpram/addr   001111   000000  000001  000010  000011  000111  001111
/singpram/q      1111     0000  0001  0010  0011  0111  1111
```

Dual Port RAM:

```
module Dual_Port_RAM (
input clk,
input we_a,
input we_b,
input [3:0] addr_a,
input [3:0] addr_b,
input [7:0] din_a,
input [7:0] din_b,
output reg [7:0] dout_a,
output reg [7:0] dout_b
);

 reg [7:0] ram [15:0];

 always @(posedge clk) begin
 // Port A operations
if (we_a)
ram[addr_a] <= din_a;
dout_a <= ram[addr_a]; // Ensure read after write is handled correctly

 // Port B operations
if (we_b)
ram[addr_b] <= din_b;
dout_b <= ram[addr_b]; // Ensure read after write is handled correctly

end

endmodule
```

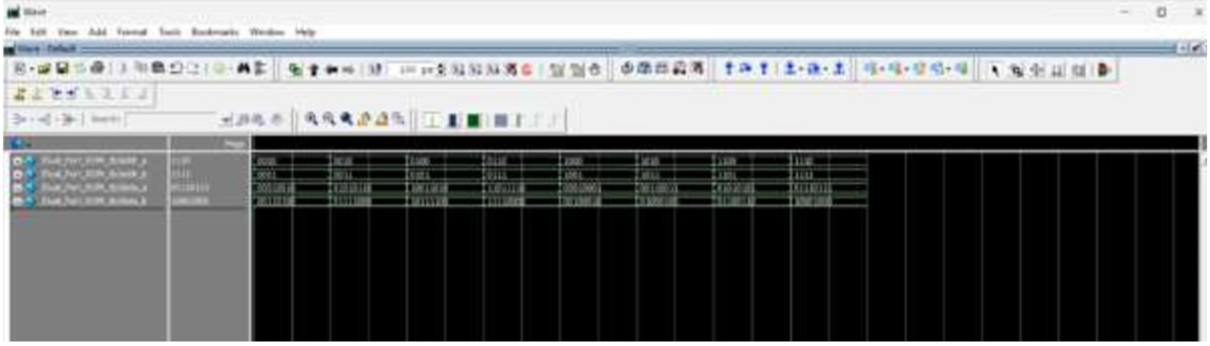
Output:



Dual Port ROM:

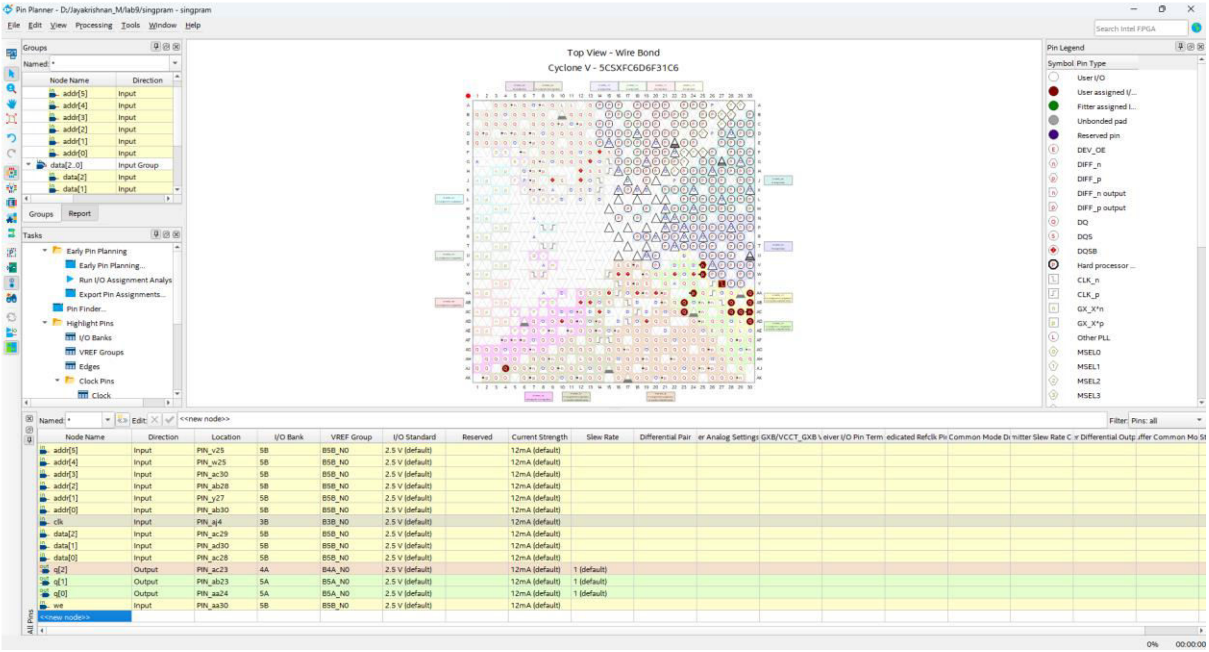
```
module Dual_Port_ROM (
 input [3:0] addr_a, addr_b, // Two 4-bit address inputs
 output reg [7:0] data_a, data_b // Two 8-bit data outputs
);

// ROM Memory Initialization (Preloaded values)
 reg [7:0] rom [15:0];
 initial begin
 rom[0] = 8'h12; rom[1] = 8'h34; rom[2] = 8'h56; rom[3] = 8'h78;
 rom[4] = 8'h9A; rom[5] = 8'hBC; rom[6] = 8'hDE; rom[7] = 8'hF0;
 rom[8] = 8'h11; rom[9] = 8'h22; rom[10] = 8'h33; rom[11] = 8'h44;
 rom[12] = 8'h55; rom[13] = 8'h66; rom[14] = 8'h77; rom[15] = 8'h88;
 end

 always @(*) begin
 data_a = rom[addr_a]; // Read data from address A
 data_b = rom[addr_b]; // Read data from address B
 end

endmodule
```

Output:



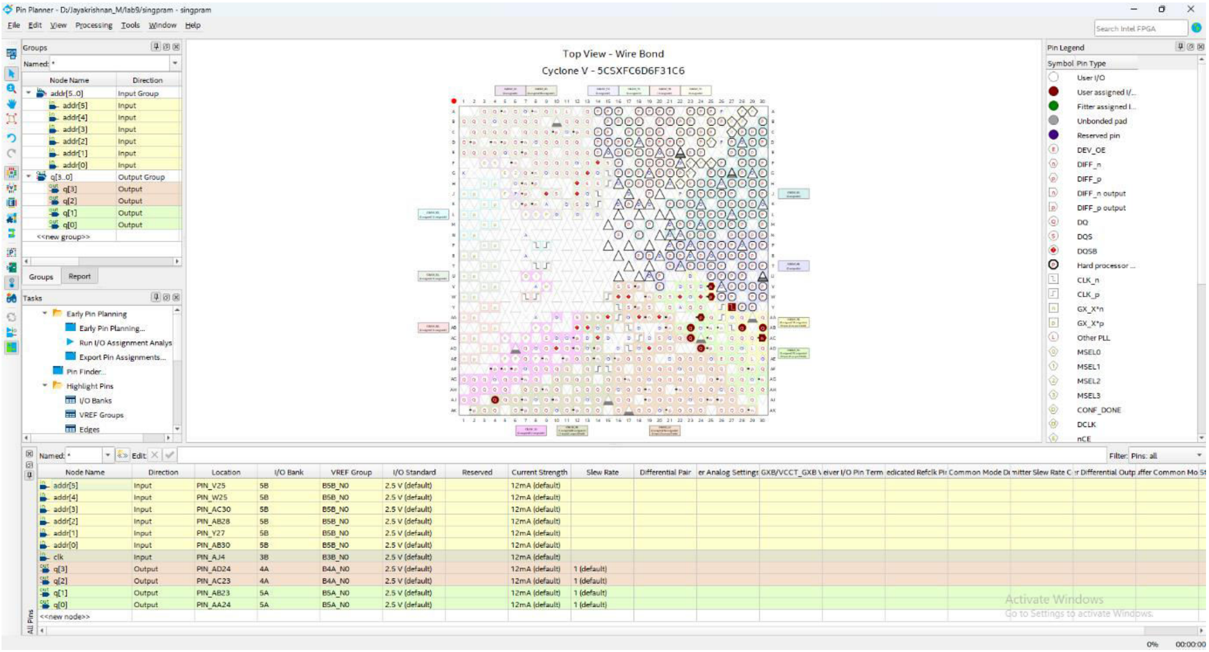
Pin Planning:

RAM (Single Port):



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
addr[5]	Input	PIN_v25	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[4]	Input	PIN_w25	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[3]	Input	PIN_ac30	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[2]	Input	PIN_ab28	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[1]	Input	PIN_y27	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[0]	Input	PIN_ab30	5B	B5B_N0	2.5 V (default)		12mA (default)	
clk	Input	PIN_aj4	3B	B3B_N0	2.5 V (default)		12mA (default)	
data[2]	Input	PIN_ac29	5B	B5B_N0	2.5 V (default)		12mA (default)	
data[1]	Input	PIN_ad30	5B	B5B_N0	2.5 V (default)		12mA (default)	
data[0]	Input	PIN_ac28	5B	B5B_N0	2.5 V (default)		12mA (default)	
q[2]	Output	PIN_ac23	4A	B4A_N0	2.5 V (default)		12mA (default)	1 (default)
q[1]	Output	PIN_ab23	5A	B5A_N0	2.5 V (default)		12mA (default)	1 (default)
q[0]	Output	PIN_aa24	5A	B5A_N0	2.5 V (default)		12mA (default)	1 (default)
we	Input	PIN_aa30	5B	B5B_N0	2.5 V (default)		12mA (default)	
<<new node>>								

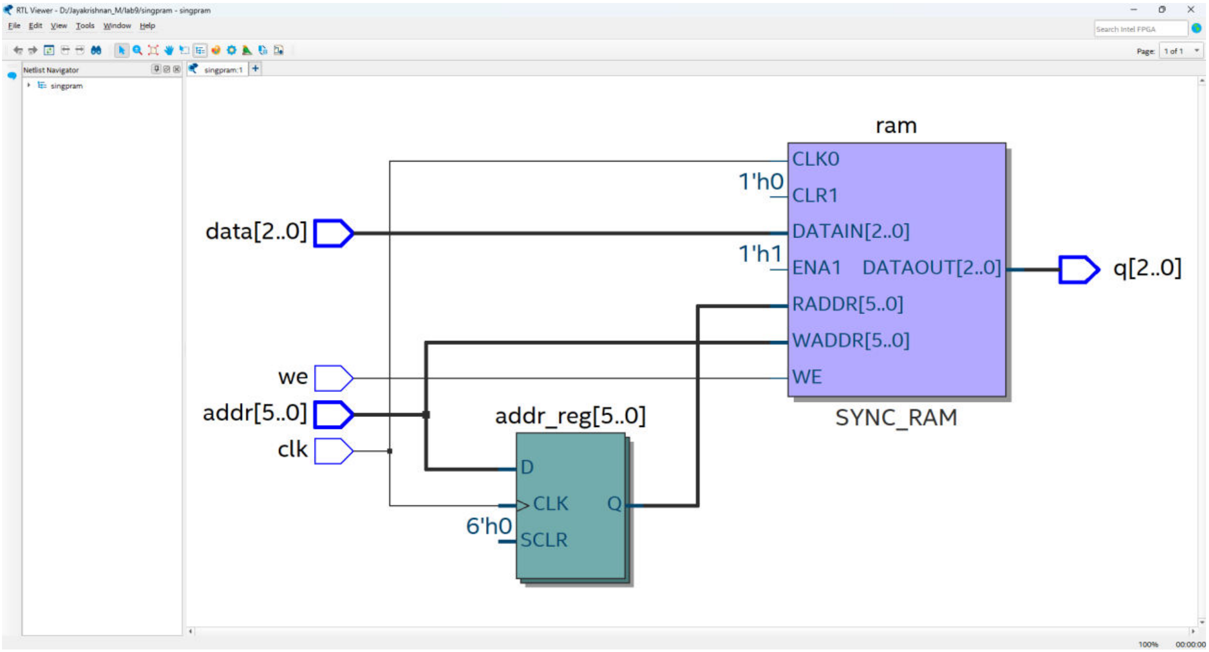
ROM (Single Port):



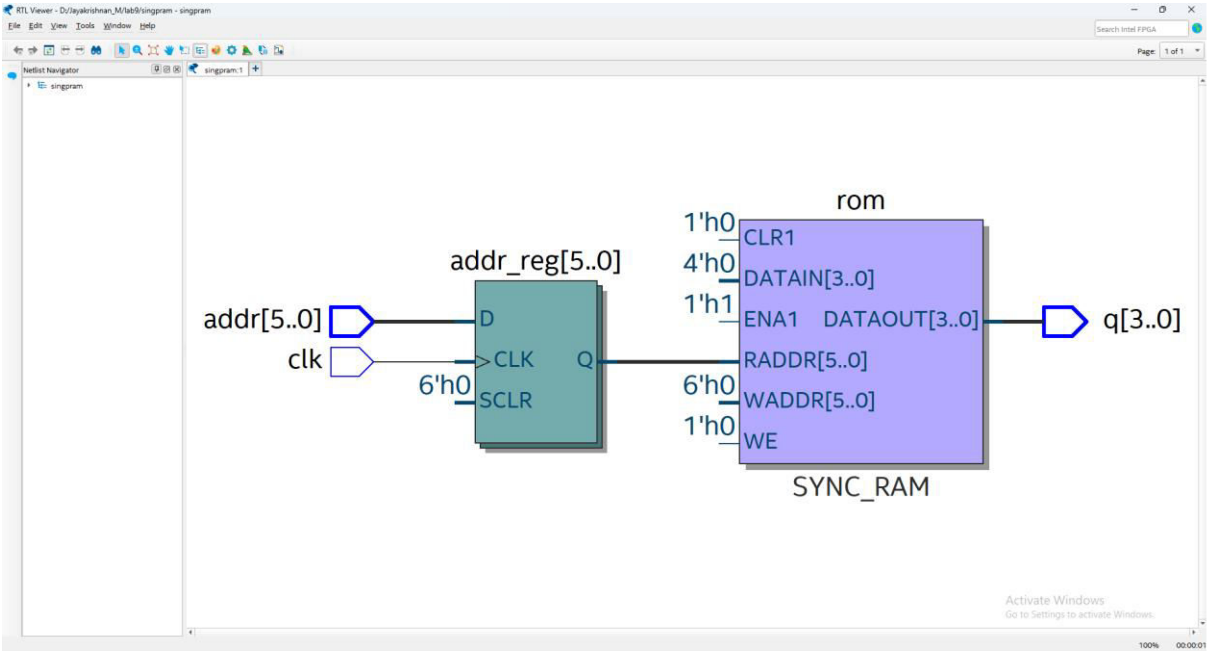
Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
addr[5]	Input	PIN_V25	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[4]	Input	PIN_W25	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[3]	Input	PIN_AC30	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[2]	Input	PIN_AB28	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[1]	Input	PIN_Y27	5B	B5B_N0	2.5 V (default)		12mA (default)	
addr[0]	Input	PIN_AB30	5B	B5B_N0	2.5 V (default)		12mA (default)	
clk	Input	PIN_AJ4	3B	B3B_N0	2.5 V (default)		12mA (default)	
q[3]	Output	PIN_AD24	4A	B4A_N0	2.5 V (default)		12mA (default)	1 (default)
q[2]	Output	PIN_AC23	4A	B4A_N0	2.5 V (default)		12mA (default)	1 (default)
q[1]	Output	PIN_AB23	5A	B5A_N0	2.5 V (default)		12mA (default)	1 (default)
q[0]	Output	PIN_AA24	5A	B5A_N0	2.5 V (default)		12mA (default)	1 (default)
<<new node>>								

RTL Viewer:

RAM (Single Port):

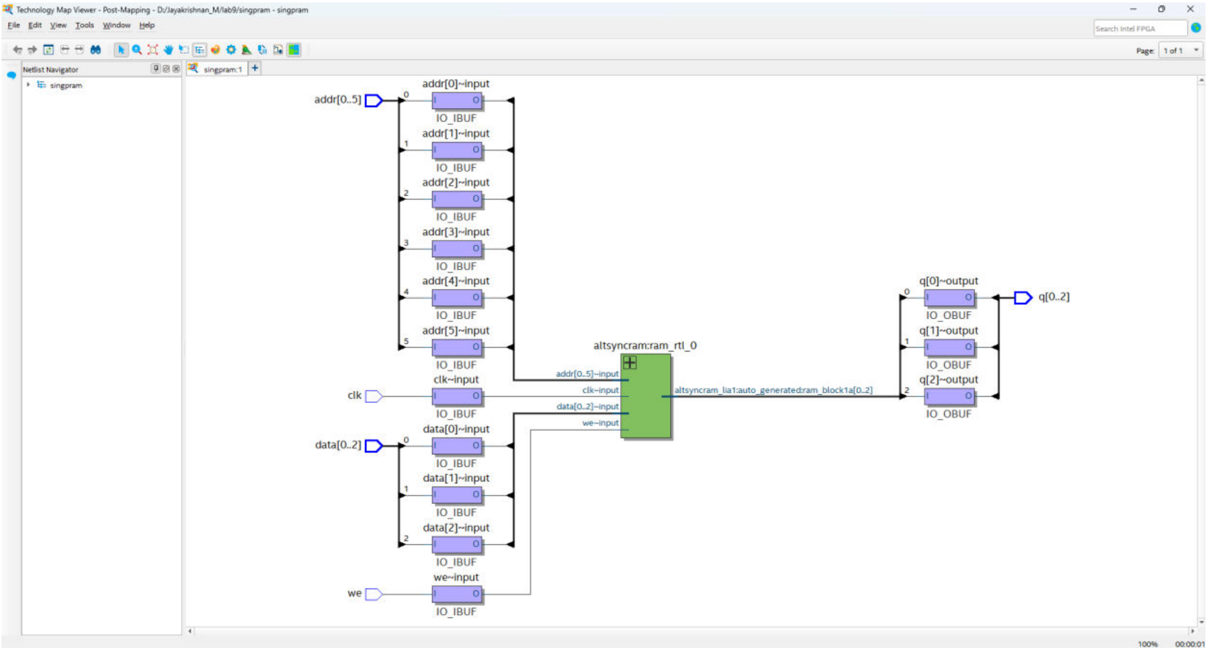


ROM (Single Port):

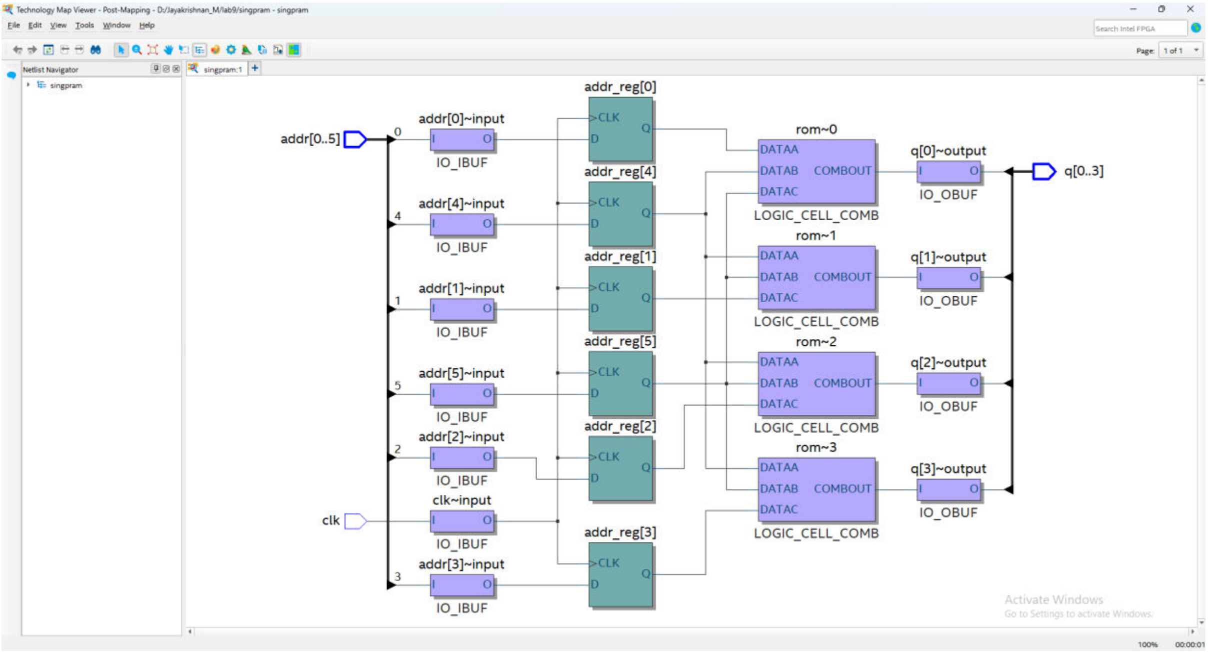


Technology Map Viewer:

RAM (Single Port):

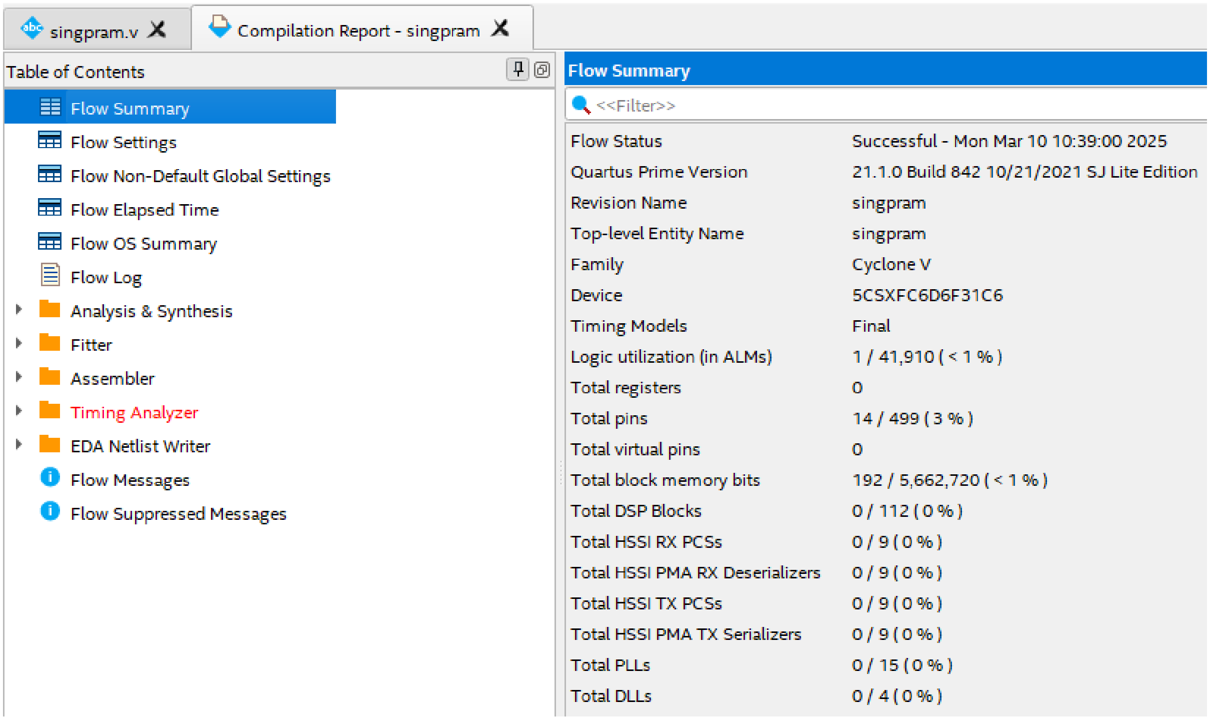


ROM (Single Port):



Logic Utilization:

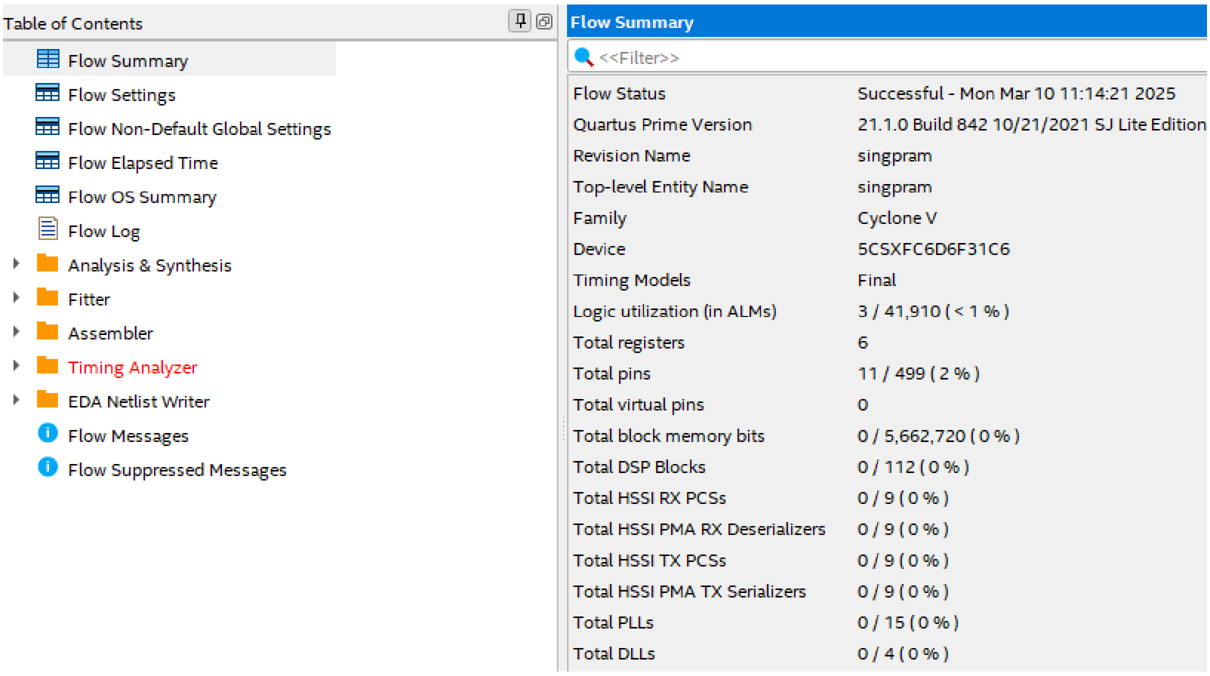
RAM (Single Port):



Flow Summary

Flow Status	Successful - Mon Mar 10 10:39:00 2025
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	singpram
Top-level Entity Name	singpram
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 41,910 (< 1 %)
Total registers	0
Total pins	14 / 499 (3 %)
Total virtual pins	0
Total block memory bits	192 / 5,662,720 (< 1 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

ROM (Single Port):



Flow Summary

Flow Status	Successful - Mon Mar 10 11:14:21 2025
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	singpram
Top-level Entity Name	singpram
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	3 / 41,910 (< 1 %)
Total registers	6
Total pins	11 / 499 (2 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

Timing Analysis:

RAM (Single Port):

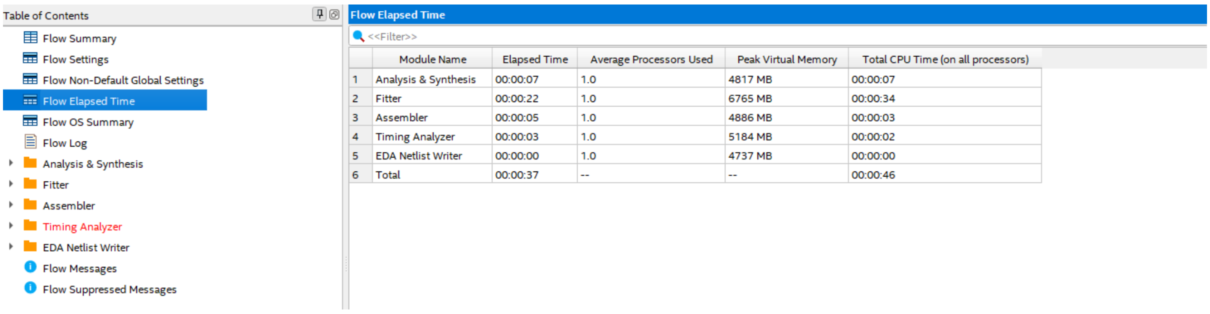


Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- Timing Analyzer
- EDA Netlist Writer
- Flow Messages
- Flow Suppressed Messages

Flow Elapsed Time

<<Filter>>

	Module Name	Elapsed Time	Average Processors Used	Peak Virtual Memory	Total CPU Time (on all processors)
1	Analysis & Synthesis	00:00:07	1.0	4817 MB	00:00:07
2	Fitter	00:00:22	1.0	6765 MB	00:00:34
3	Assembler	00:00:05	1.0	4886 MB	00:00:03
4	Timing Analyzer	00:00:03	1.0	5184 MB	00:00:02
5	EDA Netlist Writer	00:00:00	1.0	4737 MB	00:00:00
6	Total	00:00:37	--	--	00:00:46

ROM (Single Port):

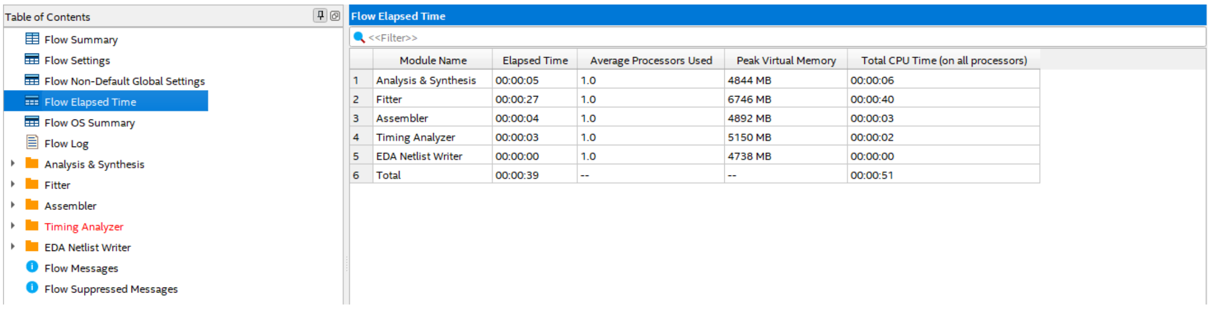


Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- Timing Analyzer
- EDA Netlist Writer
- Flow Messages
- Flow Suppressed Messages

Flow Elapsed Time

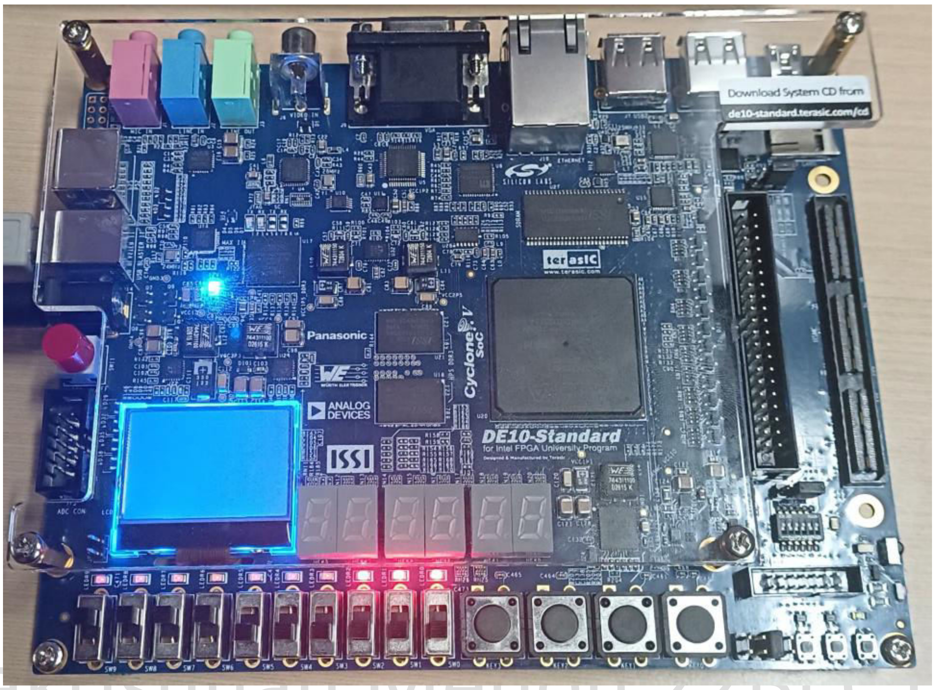
<<Filter>>

	Module Name	Elapsed Time	Average Processors Used	Peak Virtual Memory	Total CPU Time (on all processors)
1	Analysis & Synthesis	00:00:05	1.0	4844 MB	00:00:06
2	Fitter	00:00:27	1.0	6746 MB	00:00:40
3	Assembler	00:00:04	1.0	4892 MB	00:00:03
4	Timing Analyzer	00:00:03	1.0	5150 MB	00:00:02
5	EDA Netlist Writer	00:00:00	1.0	4738 MB	00:00:00
6	Total	00:00:39	--	--	00:00:51

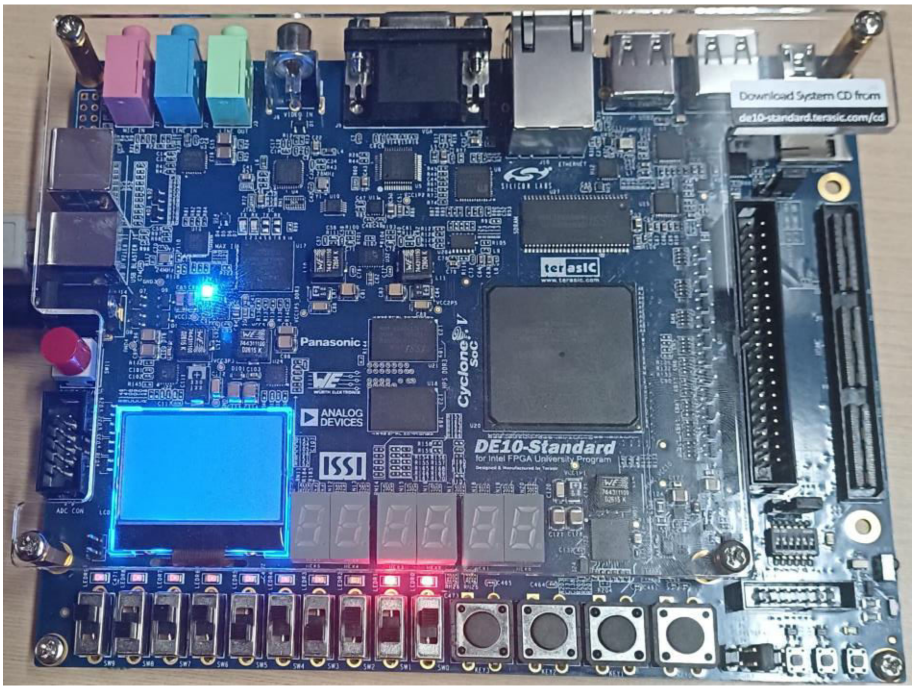
Hardware Implementation:

RAM (Single Port):

Location:000000 Data:111

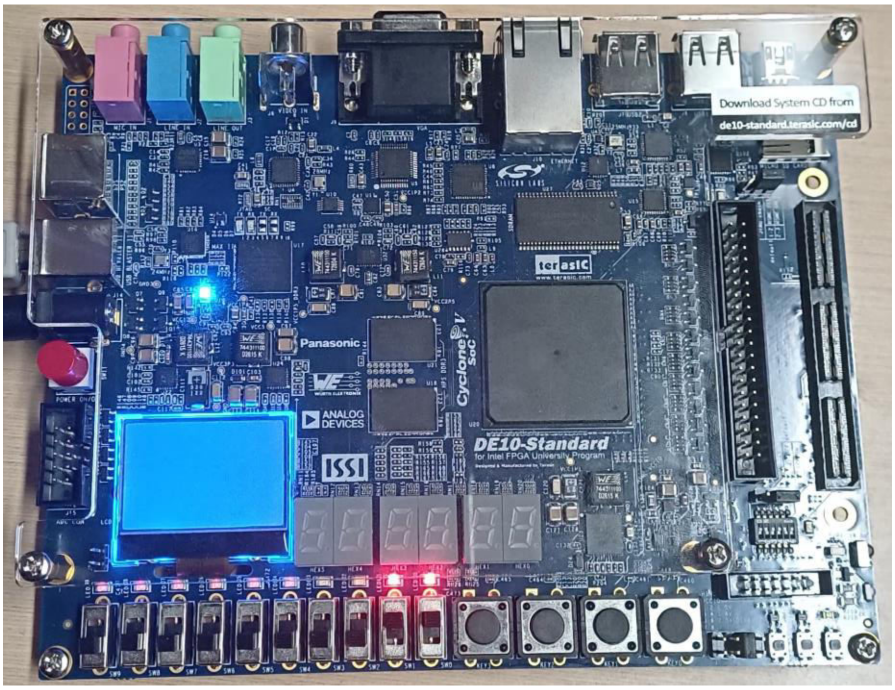


Location:010101 Data:011

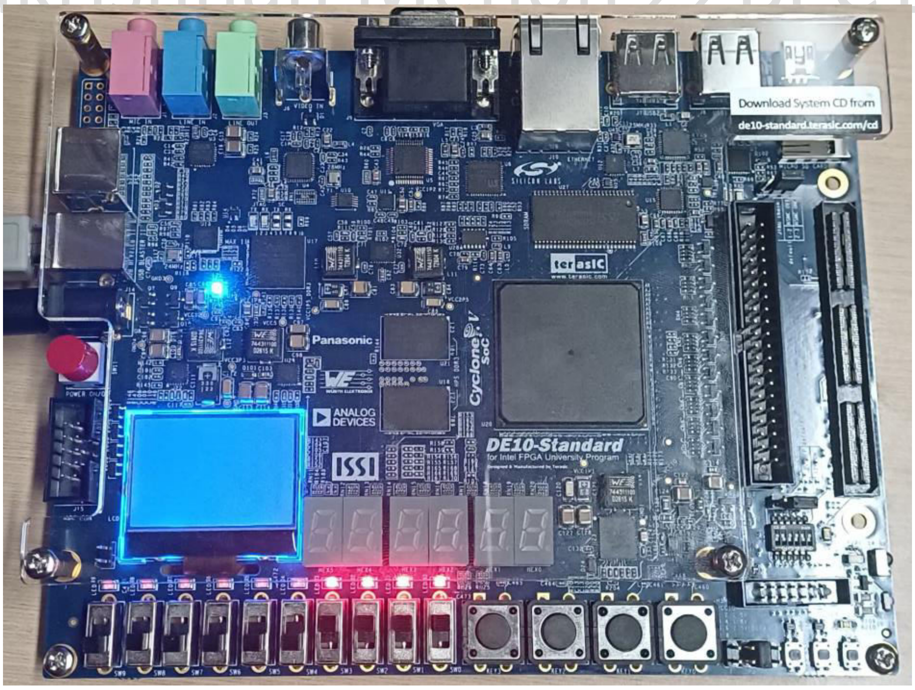


ROM (Single Port):

Location:000011 Data:0011



Location:001111 Data:1111



Result:

Hence, Verilog RTL codes were successfully verified and implemented using Quartus Prime. The given task was completed successfully