CENTER OF NANOELECTRONICS & DEVICES

Ministry of Communications and Information Technology

eme
egypt makes electronics

CND 211

Final Project

Complete ASIC Flow of I2C communication protocol

Submitted by:

Section 13 - CND211_Group 1

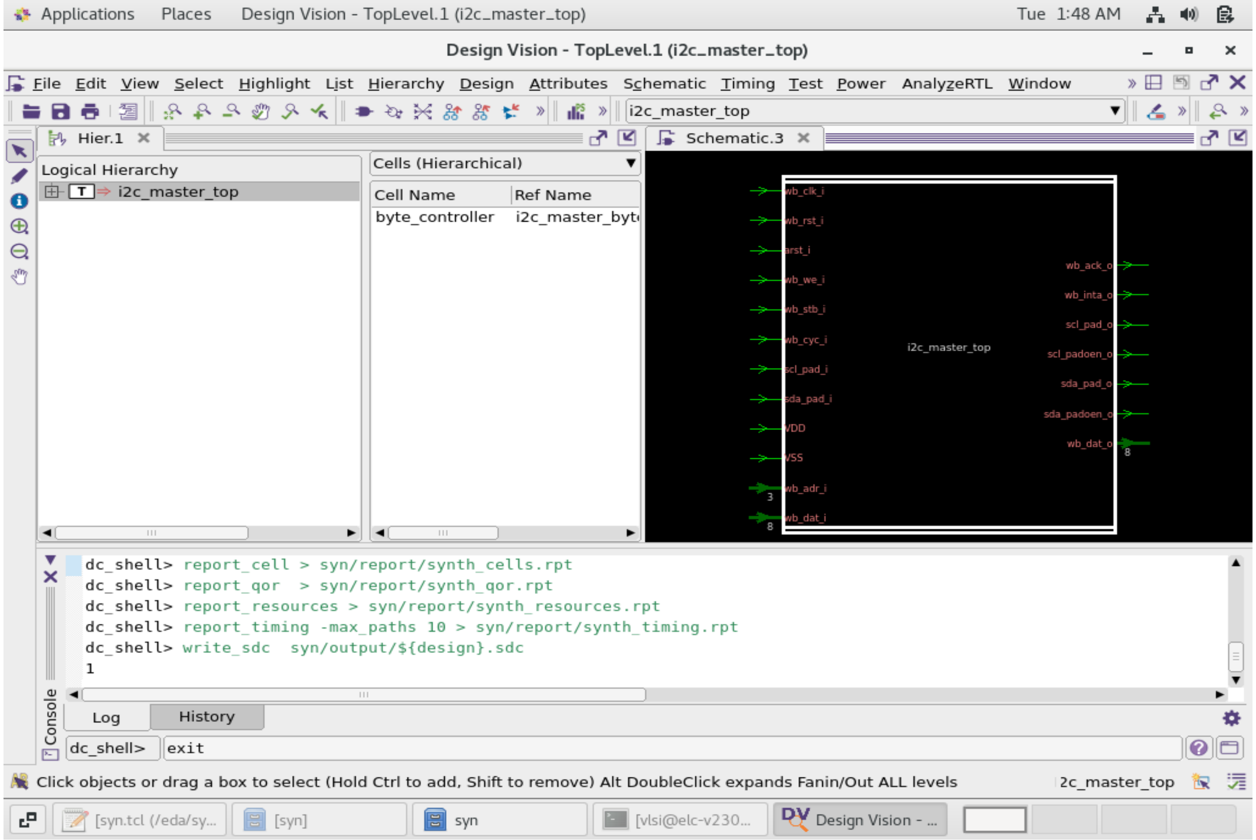
Name	ID
Ganna Hassan Helmi	v23010408
Yomnah Abdelaziz hassan	v23010327
Mohamed Waleed	v23010364
Zeyad Mohsen Abdelmenam	v23010303

Introduction

In this project we have pass by two flows firstly we have made Complete ASIC flow for the implementation of the I2C Master with wishbone interface , Concerning the ASIC flow we have made synthesis,STA, floorplanning, powerplanning , pnr , CTS&Prime time scribts & run them through different synopsys tool : dc_shell , icc2_shell , lm_shell & pm_shell . Also we have done formal verification for the design using fm_shell tool . We gonna show you the outputs of all of these steps in the rest of the documntation

Also we have passed by another flow which is the front end part & it’s an additional part in the project as we have designed a new implementation of I2C protocoal between master & slave of sending 8-bits of data with taking refrence from a paper **Design of I2C Protocol in Verilog-A New Approach[1] ,** where in these approach we take in our considration that we are going to implement our Design in RTL so we have changed some signals and we have done simulation of this part using questa sim tool

Synthesis output of the design:

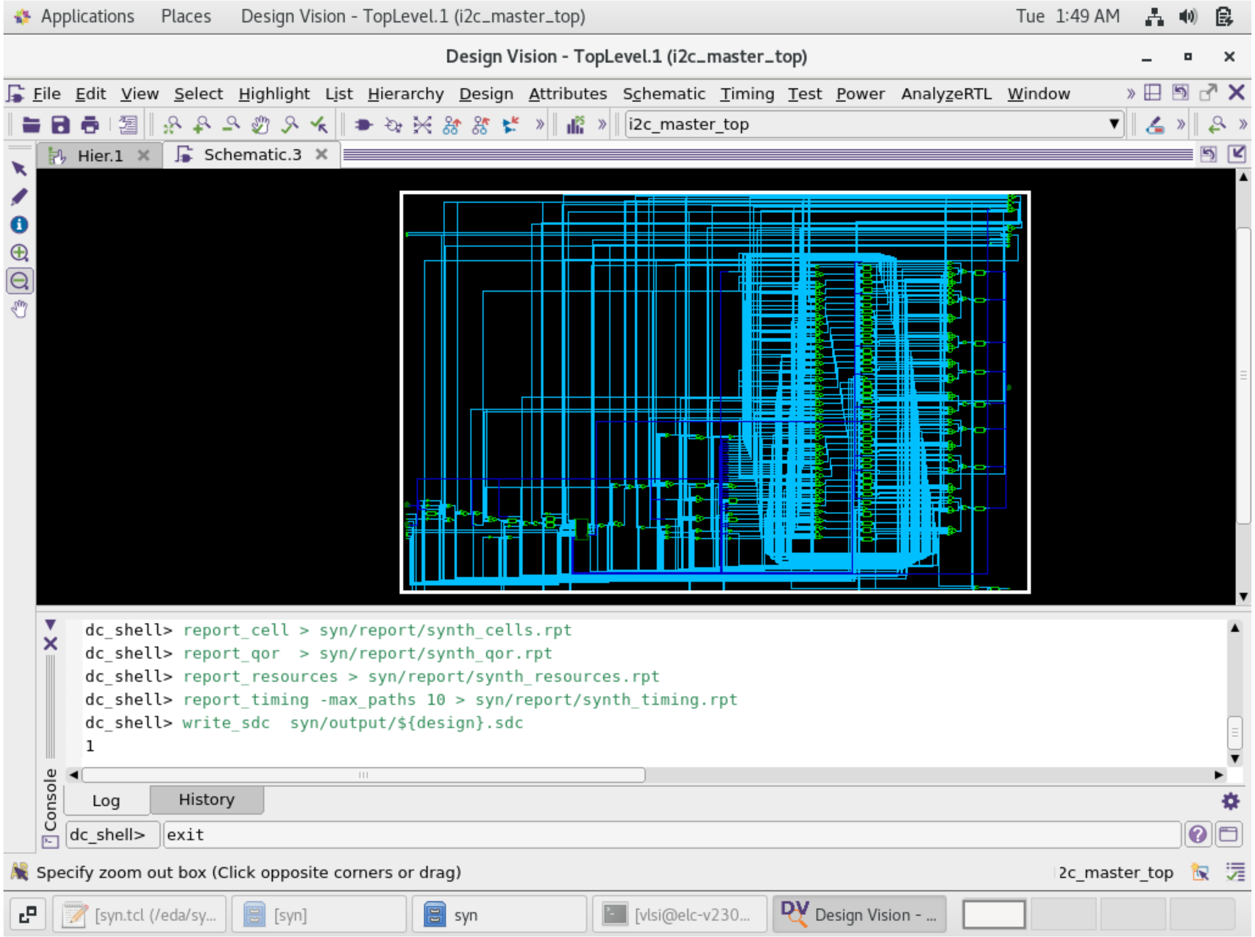


Cell Name	Ref Name
byte_controller	i2c_master_byt

```
dc_shell> report_cell > syn/report/synth_cells.rpt
dc_shell> report_qor  > syn/report/synth_qor.rpt
dc_shell> report_resources > syn/report/synth_resources.rpt
dc_shell> report_timing -max_paths 10 > syn/report/synth_timing.rpt
dc_shell> write_sdc  syn/output/${design}.sdc
1
```

```
dc_shell> exit
```

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove) Alt DoubleClick expands Fanin/Out ALL levels



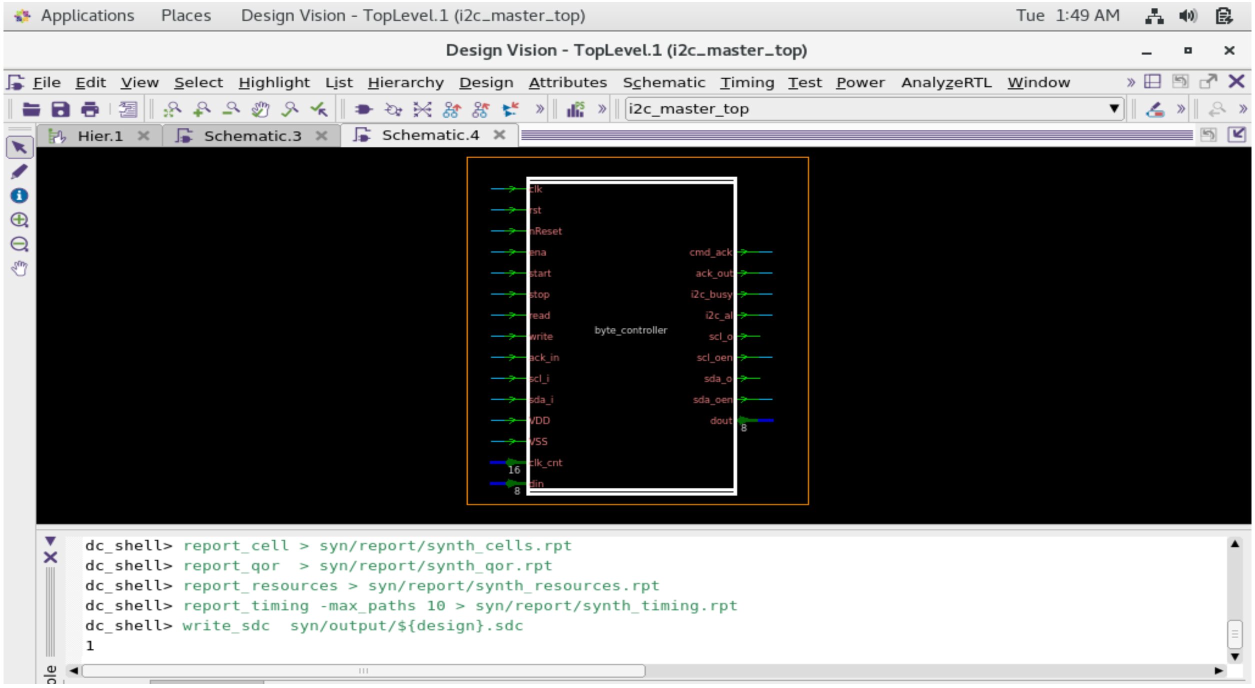
```
dc_shell> report_cell > syn/report/synth_cells.rpt
dc_shell> report_qor  > syn/report/synth_qor.rpt
dc_shell> report_resources > syn/report/synth_resources.rpt
dc_shell> report_timing -max_paths 10 > syn/report/synth_timing.rpt
dc_shell> write_sdc  syn/output/${design}.sdc
1
```

```
dc_shell> exit
```

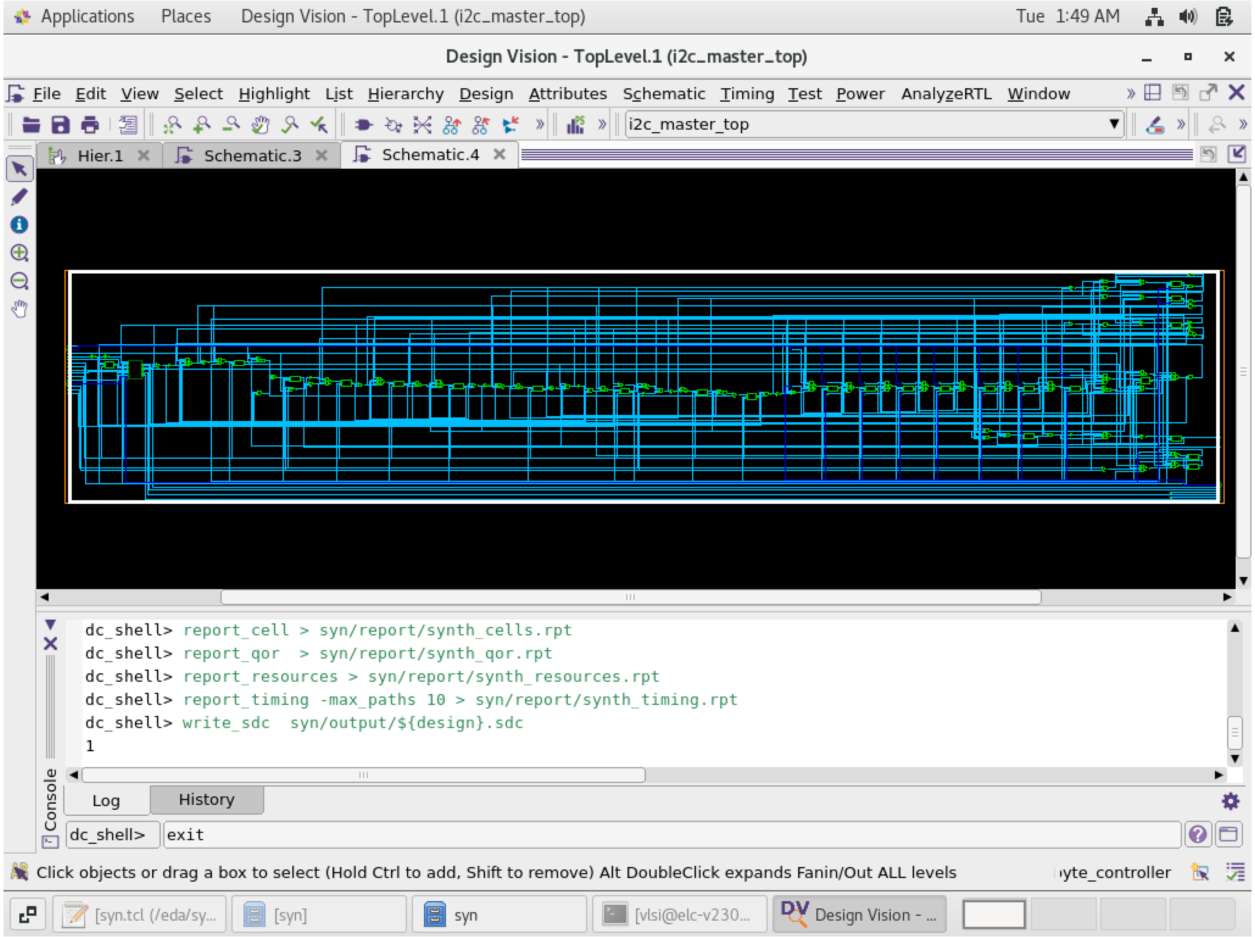
Specify zoom out box (Click opposite corners or drag)

Design Vision - TopLevel.1 (i2c_master_top)
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power AnalyzeRTL Window
i2c_master_top
Hier.1
Schematic.3
Log
History
Console

Figure 1:synthesis of top module



```
dc_shell> report_cell > syn/report/synth_cells.rpt
dc_shell> report_qor  > syn/report/synth_qor.rpt
dc_shell> report_resources > syn/report/synth_resources.rpt
dc_shell> report_timing -max_paths 10 > syn/report/synth_timing.rpt
dc_shell> write_sdc  syn/output/${design}.sdc
1
```


Applications
Places
Design Vision - TopLevel.1 (i2c_master_top)
Tue 1:49 AM
Design Vision - TopLevel.1 (i2c_master_top)
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power AnalyzeRTL Window
i2c_master_top
Hier.1
Schematic.3
Schematic.4
Console
Log
History
Click objects or drag a box to select (Hold Ctrl to add, Shift to remove) Alt DoubleClick expands Fanin/Out ALL levels
ıyte_controller
[syn.tcl (/eda/sy...
[syn]
syn
[vlsi@elc-v230...
Design Vision - ...

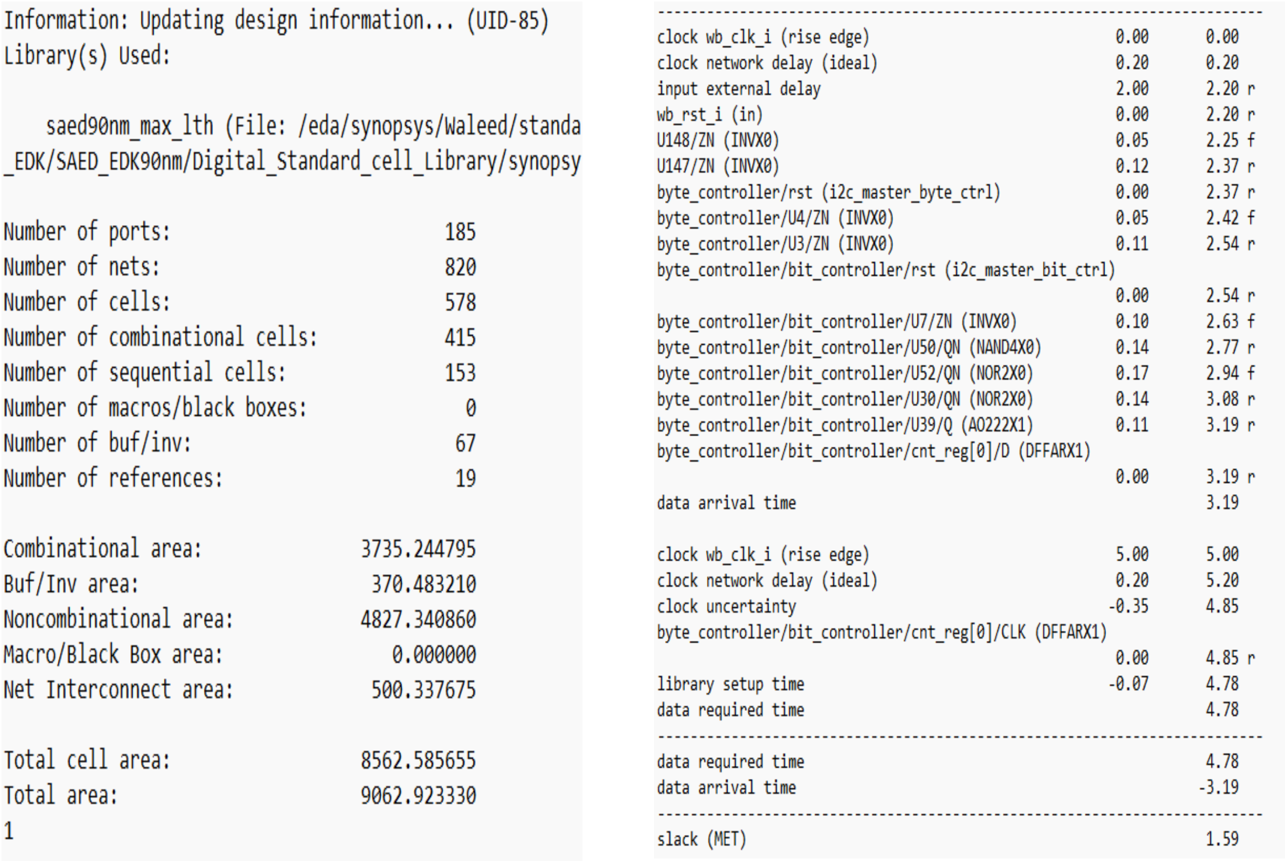
```
dc_shell> report_cell > syn/report/synth_cells.rpt
dc_shell> report_qor  > syn/report/synth_qor.rpt
dc_shell> report_resources > syn/report/synth_resources.rpt
dc_shell> report_timing -max_paths 10 > syn/report/synth_timing.rpt
dc_shell> write_sdc  syn/output/${design}.sdc
1
```

```
dc_shell> exit
```

Figure 2:synthesis of byte controller

It’s clear that the synthesis has been done smoothly and in the following part we are going to show both area & worst slack timing path also you will find all of these reports attached to the project on github

Timing & Area reports:



```
Information: Updating design information... (UID-85)
Library(s) Used:

    saed90nm_max_lth (File: /eda/synopsys/Waleed/standa
_EDK/SAED_EDK90nm/Digital_Standard_cell_Library/synopsy

Number of ports:                          185
Number of nets:                           820
Number of cells:                          578
Number of combinational cells:            415
Number of sequential cells:               153
Number of macros/black boxes:               0
Number of buf/inv:                         67
Number of references:                      19

Combinational area:               3735.244795
Buf/Inv area:                      370.483210
Noncombinational area:            4827.340860
Macro/Black Box area:                0.000000
Net Interconnect area:             500.337675

Total cell area:                  8562.585655
Total area:                       9062.923330
1
```

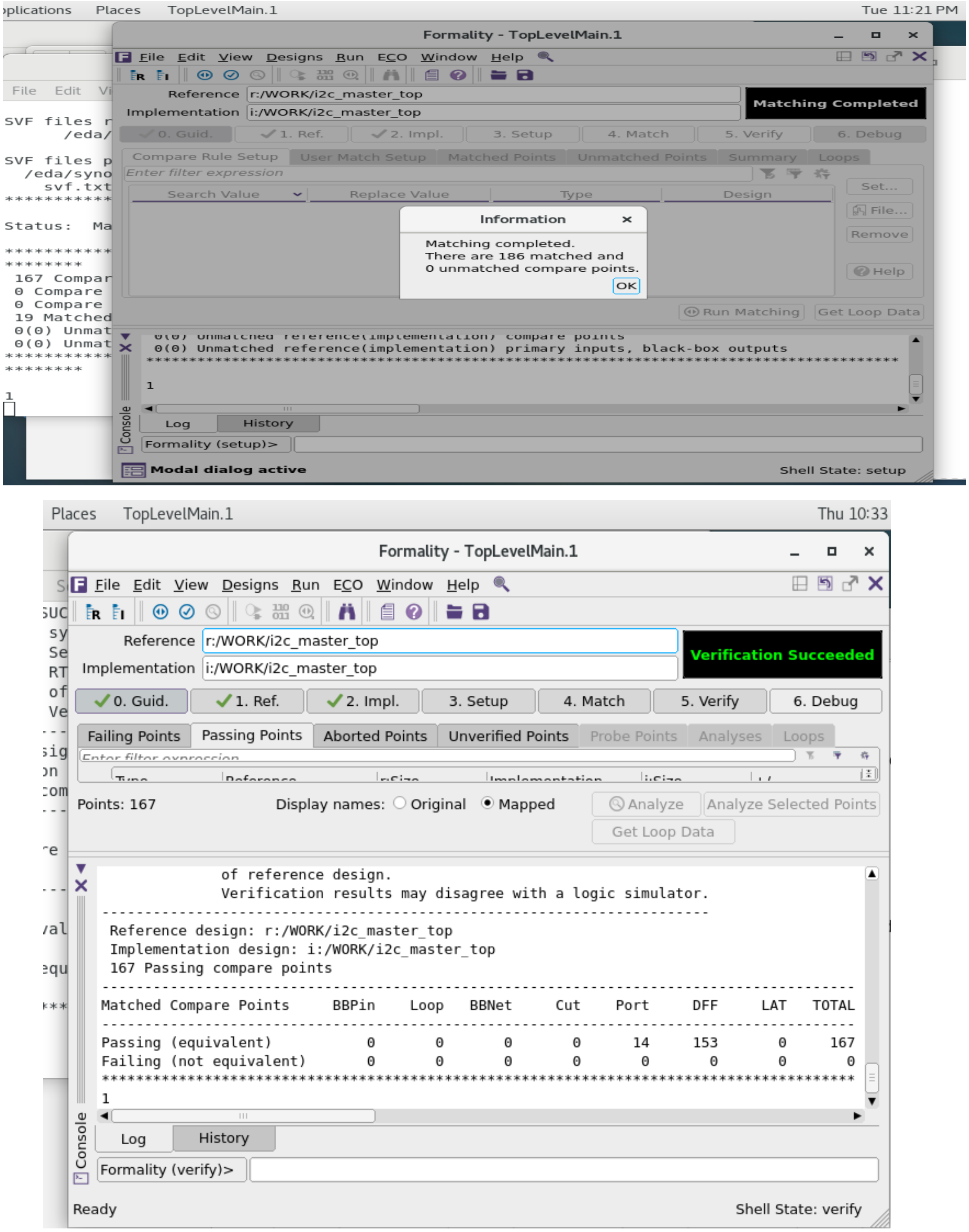
```
---------------------------------------------------------------------------
clock wb_clk_i (rise edge)                                0.00       0.00
clock network delay (ideal)                               0.20       0.20
input external delay                                      2.00       2.20 r
wb_rst_i (in)                                             0.00       2.20 r
U148/ZN (INVX0)                                           0.05       2.25 f
U147/ZN (INVX0)                                           0.12       2.37 r
byte_controller/rst (i2c_master_byte_ctrl)                0.00       2.37 r
byte_controller/U4/ZN (INVX0)                             0.05       2.42 f
byte_controller/U3/ZN (INVX0)                             0.11       2.54 r
byte_controller/bit_controller/rst (i2c_master_bit_ctrl)
                                                          0.00       2.54 r
byte_controller/bit_controller/U7/ZN (INVX0)              0.10       2.63 f
byte_controller/bit_controller/U50/QN (NAND4X0)           0.14       2.77 r
byte_controller/bit_controller/U52/QN (NOR2X0)            0.17       2.94 f
byte_controller/bit_controller/U30/QN (NOR2X0)            0.14       3.08 r
byte_controller/bit_controller/U39/Q (AO222X1)            0.11       3.19 r
byte_controller/bit_controller/cnt_reg[0]/D (DFFARX1)
                                                          0.00       3.19 r
data arrival time                                                    3.19

clock wb_clk_i (rise edge)                                5.00       5.00
clock network delay (ideal)                               0.20       5.20
clock uncertainty                                        -0.35       4.85
byte_controller/bit_controller/cnt_reg[0]/CLK (DFFARX1)
                                                          0.00       4.85 r
library setup time                                       -0.07       4.78
data required time                                                   4.78
---------------------------------------------------------------------------
data required time                                                   4.78
data arrival time                                                   -3.19
---------------------------------------------------------------------------
slack (MET)                                                          1.59
```

Figure 3:timing & area reports

The worst path has a slack with 1.59 which is a good slack depending on our requirements which just need no violations where all the setup & hold paths are MET depending on applied constraints

<u>Formal verification:</u>



Reference: r:/WORK/i2c_master_top

Implementation: i:/WORK/i2c_master_top

Matching Completed

Information

Matching completed.
There are 186 matched and
0 unmatched compare points.

OK

```
SVF files r
     /eda/

SVF files p
  /eda/syno
     svf.txt
***********

Status:  Ma

***********
********
 167 Compar
 0 Compare
 0 Compare
 19 Matched
 0(0) Unmat
 0(0) Unmat
***********
********

1
```

```
 0(0) Unmatched reference(implementation) compare points
 0(0) Unmatched reference(implementation) primary inputs, black-box outputs
******************************************************************************

1
```

Formality (setup)>

Modal dialog active

Shell State: setup

Reference: r:/WORK/i2c_master_top

Implementation: i:/WORK/i2c_master_top

Verification Succeeded

Points: 167

Display names: ○ Original ● Mapped

```
              of reference design.
              Verification results may disagree with a logic simulator.
 ----------------------------------------------------------------
 Reference design: r:/WORK/i2c_master_top
 Implementation design: i:/WORK/i2c_master_top
 167 Passing compare points
--------------------------------------------------------------------------------
Matched Compare Points     BBPin    Loop   BBNet     Cut    Port     DFF     LAT   TOTAL
--------------------------------------------------------------------------------
Passing (equivalent)           0       0       0       0      14     153       0     167
Failing (not equivalent)       0       0       0       0       0       0       0       0
********************************************************************************
1
```

Formality (verify)>

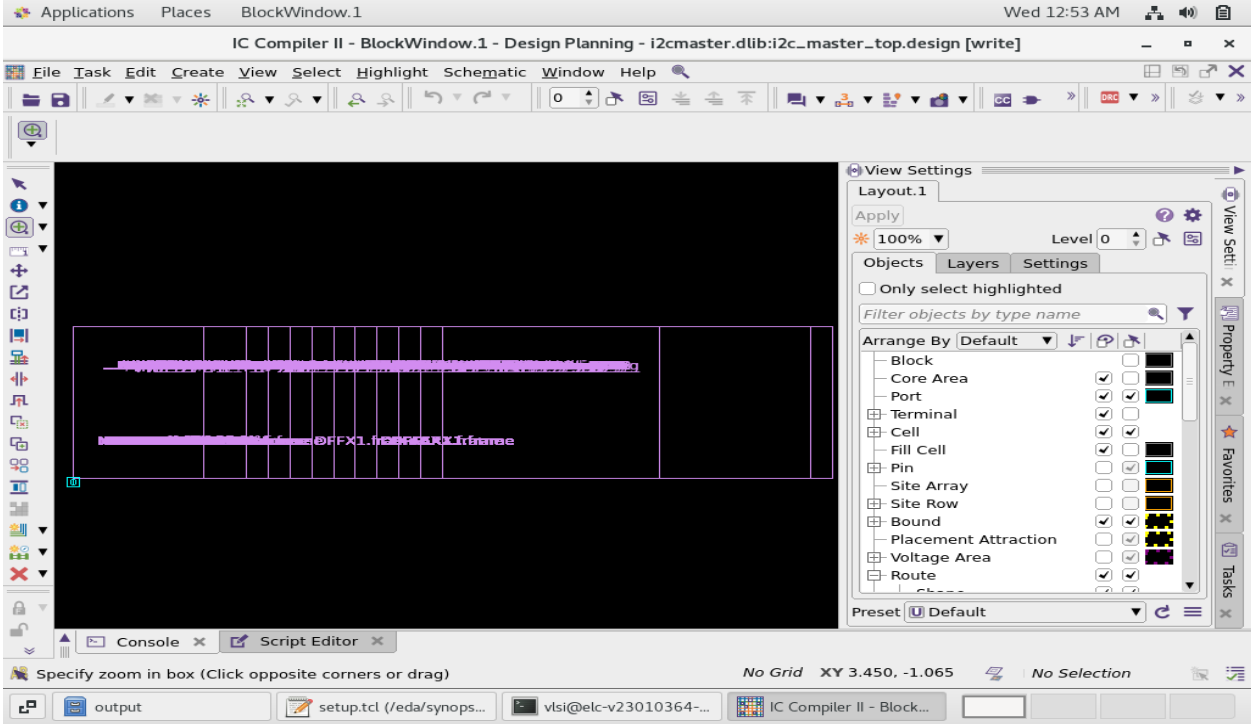
Ready

Shell State: verify

Figure 4:formality output

It’s obvious that the formal verification has passed successfully without any falling points

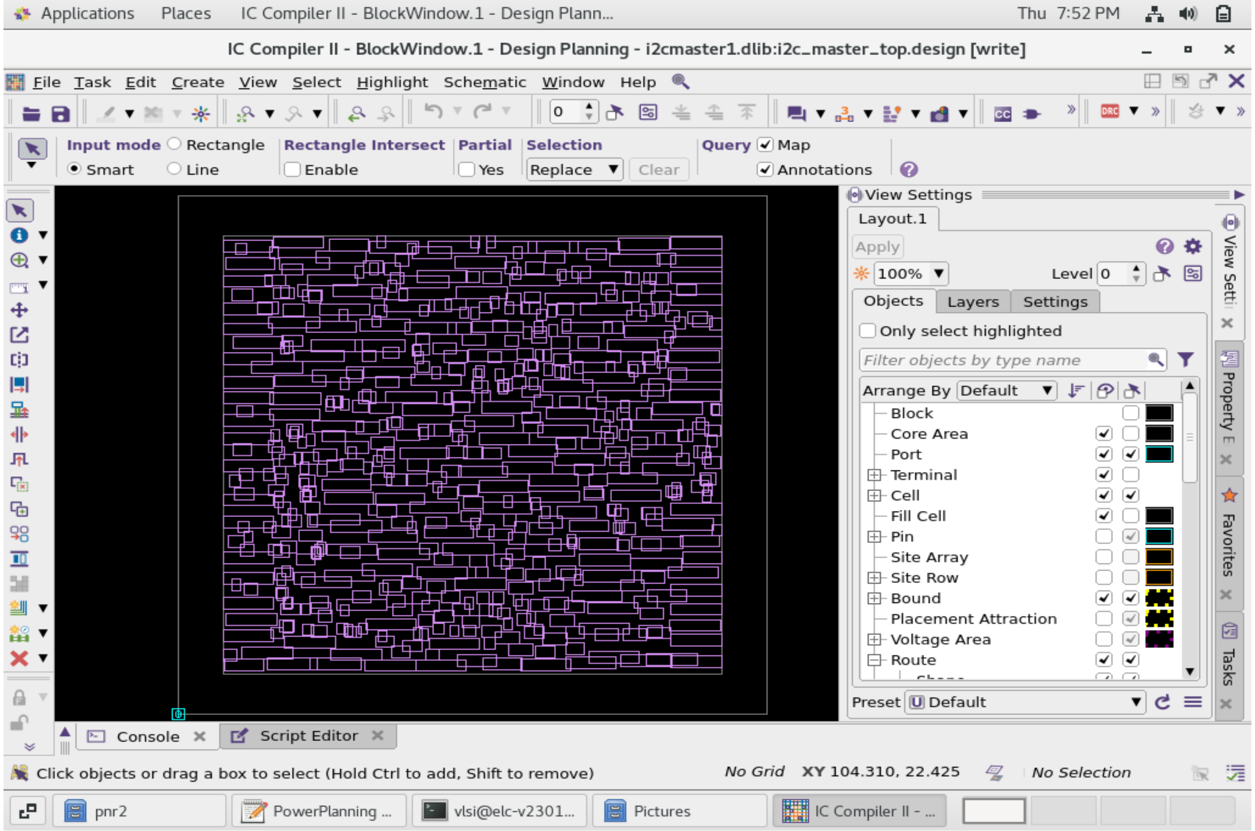
PnR Flow:



IC Compiler II - BlockWindow.1 - Design Planning - i2cmaster.dlib:i2c_master_top.design [write]

Specify zoom in box (Click opposite corners or drag)   No Grid   XY 3.450, -1.065   No Selection

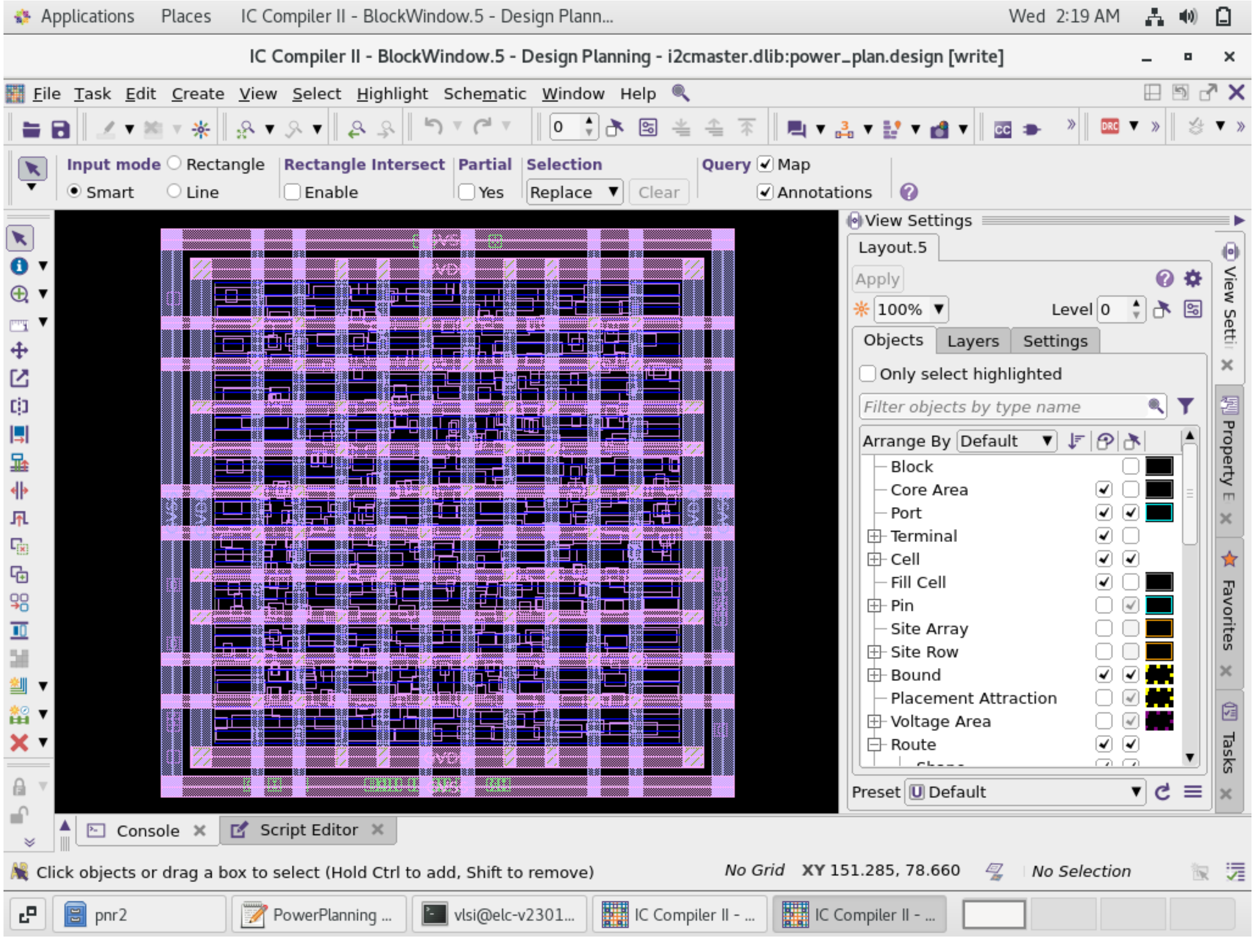
Figure 5:floorplan output



IC Compiler II - BlockWindow.1 - Design Planning - i2cmaster1.dlib:i2c_master_top.design [write]

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)   No Grid   XY 104.310, 22.425   No Selection

Figure 6:initial placment output


IC Compiler II - BlockWindow.5 - Design Planning - i2cmaster.dlib:power_plan.design [write]

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)   No Grid   XY 151.285, 78.660   No Selection

Figure 7:powerplan output

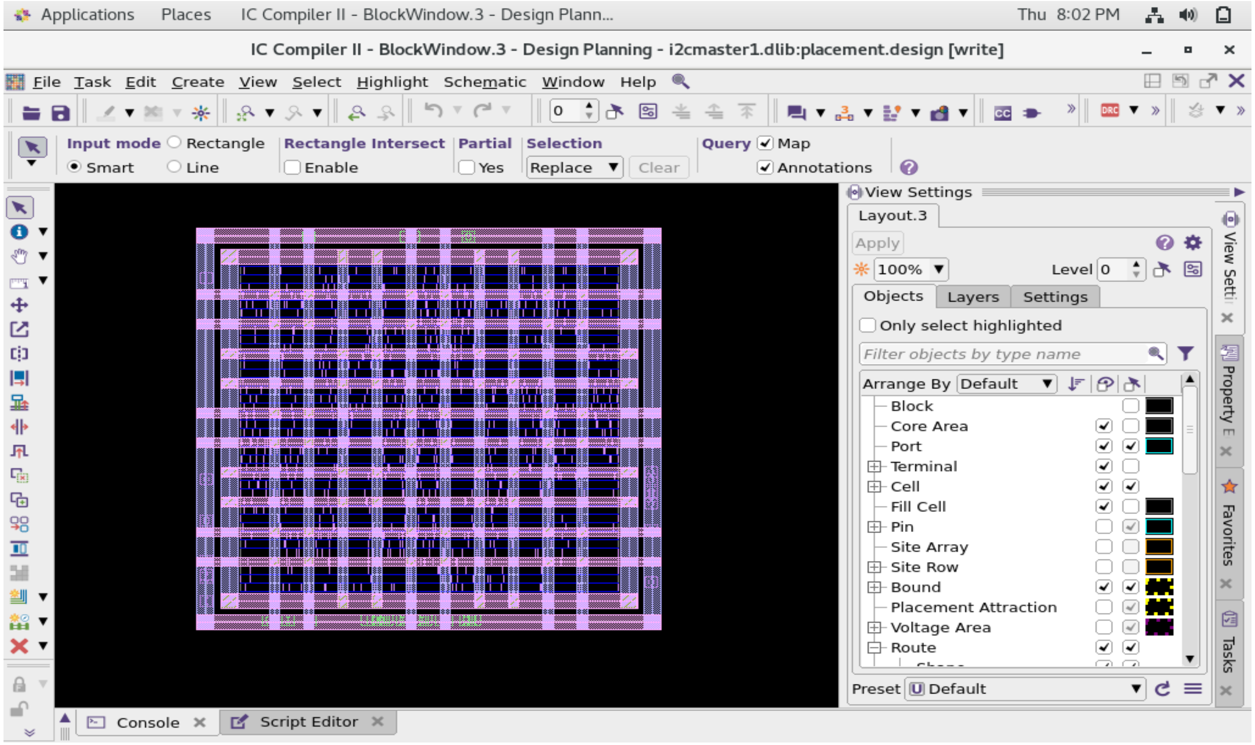

IC Compiler II - BlockWindow.3 - Design Planning - i2cmaster1.dlib:placement.design [write]

Figure 8:final placement output

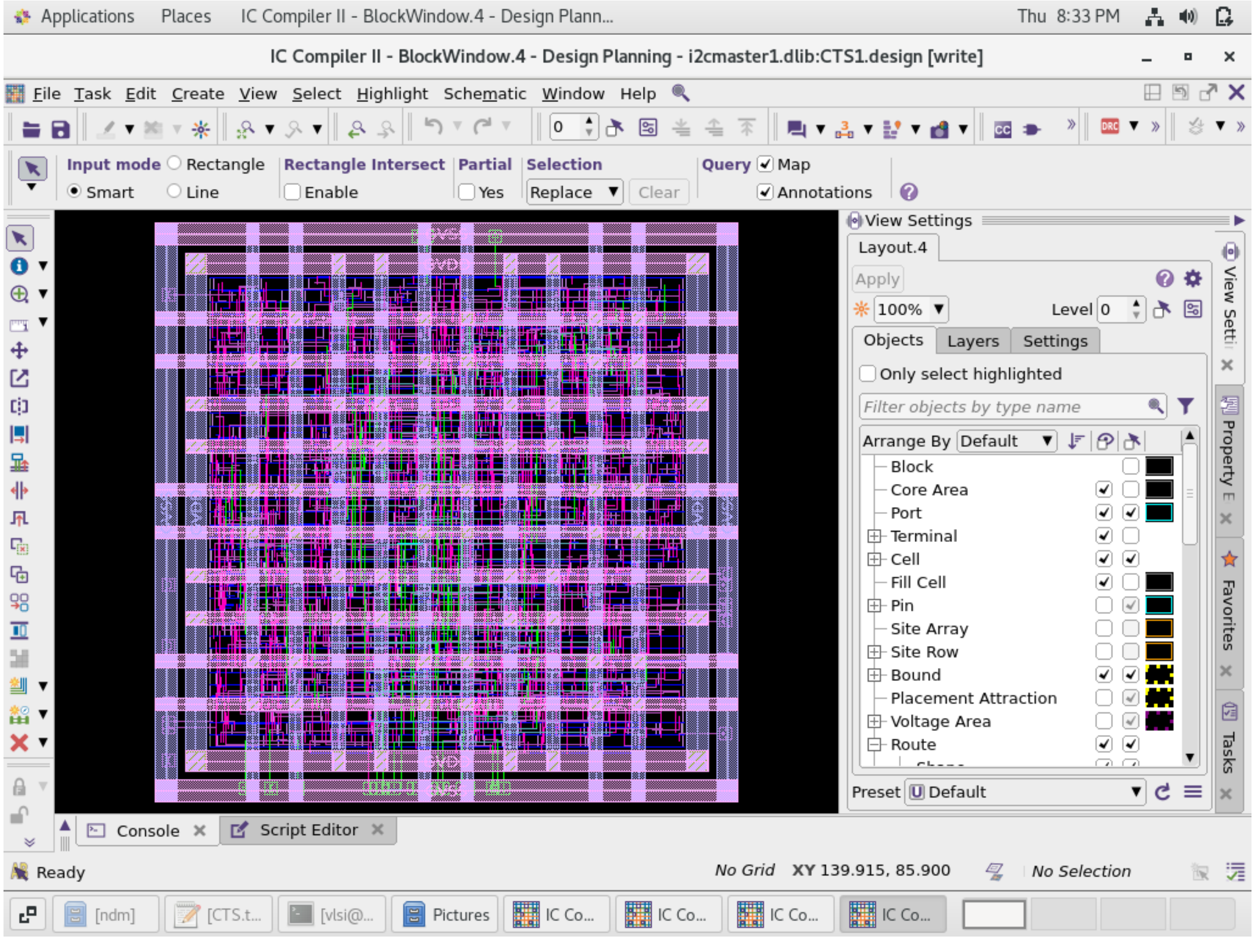

Applications Places IC Compiler II - BlockWindow.4 - Design Plann...
Thu 8:33 PM
IC Compiler II - BlockWindow.4 - Design Planning - i2cmaster1.dlib:CTS1.design [write]
File Task Edit Create View Select Highlight Schematic Window Help
Input mode ○ Rectangle ● Smart ○ Line
Rectangle Intersect □ Enable
Partial □ Yes
Selection Replace Clear
Query ☑ Map ☑ Annotations
View Settings
Layout.4
Apply
100% Level 0
Objects Layers Settings
□ Only select highlighted
Filter objects by type name
Arrange By Default
Block
Core Area
Port
Terminal
Cell
Fill Cell
Pin
Site Array
Site Row
Bound
Placement Attraction
Voltage Area
Route
Preset Default
Property E
Favorites
Tasks
Console Script Editor
Ready
No Grid XY 139.915, 85.900
No Selection
[ndm] [CTS.t... [vlsi@... Pictures IC Co... IC Co... IC Co... IC Co...

Figure 9:Output after CTS

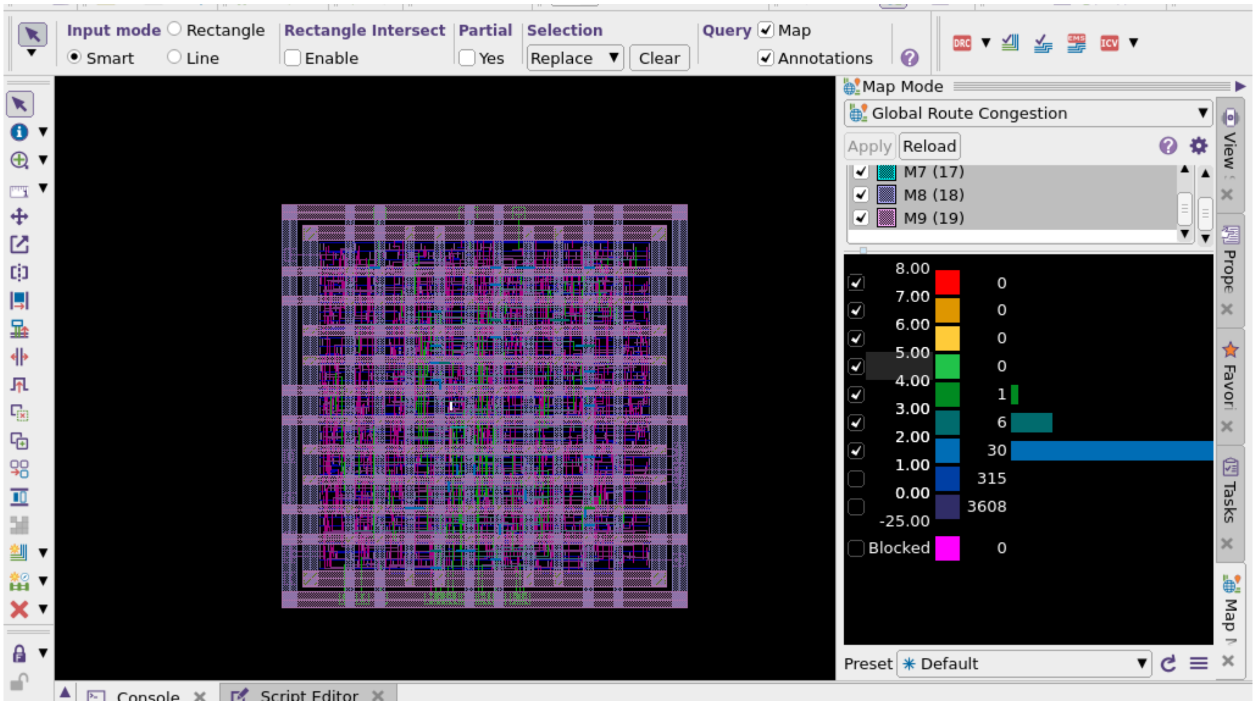


Figure 10:Global route congestion view

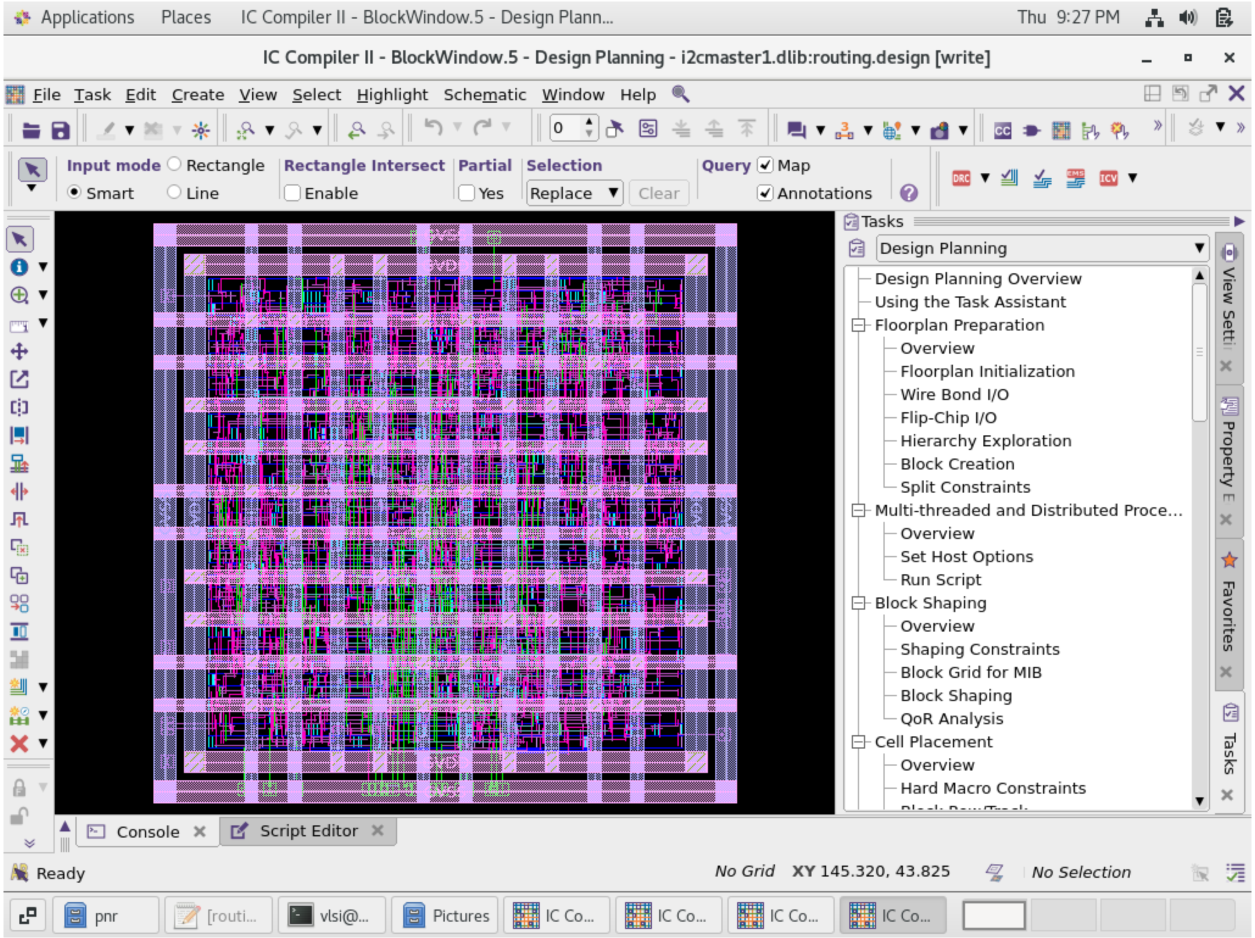


Figure 11:output After Routing

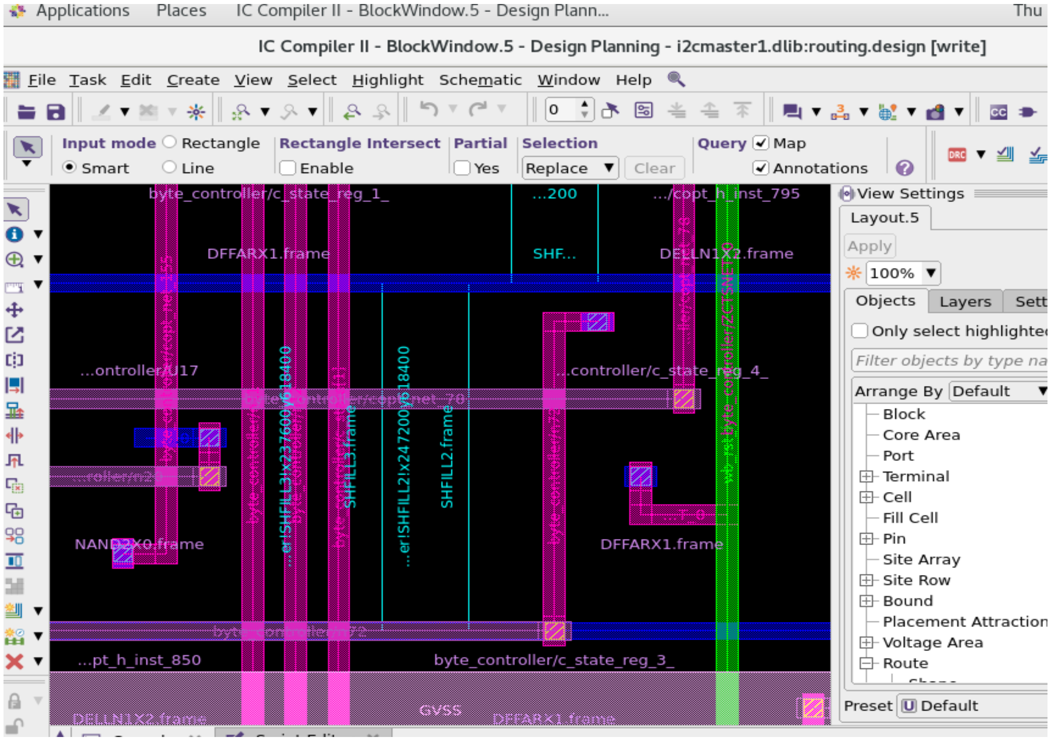
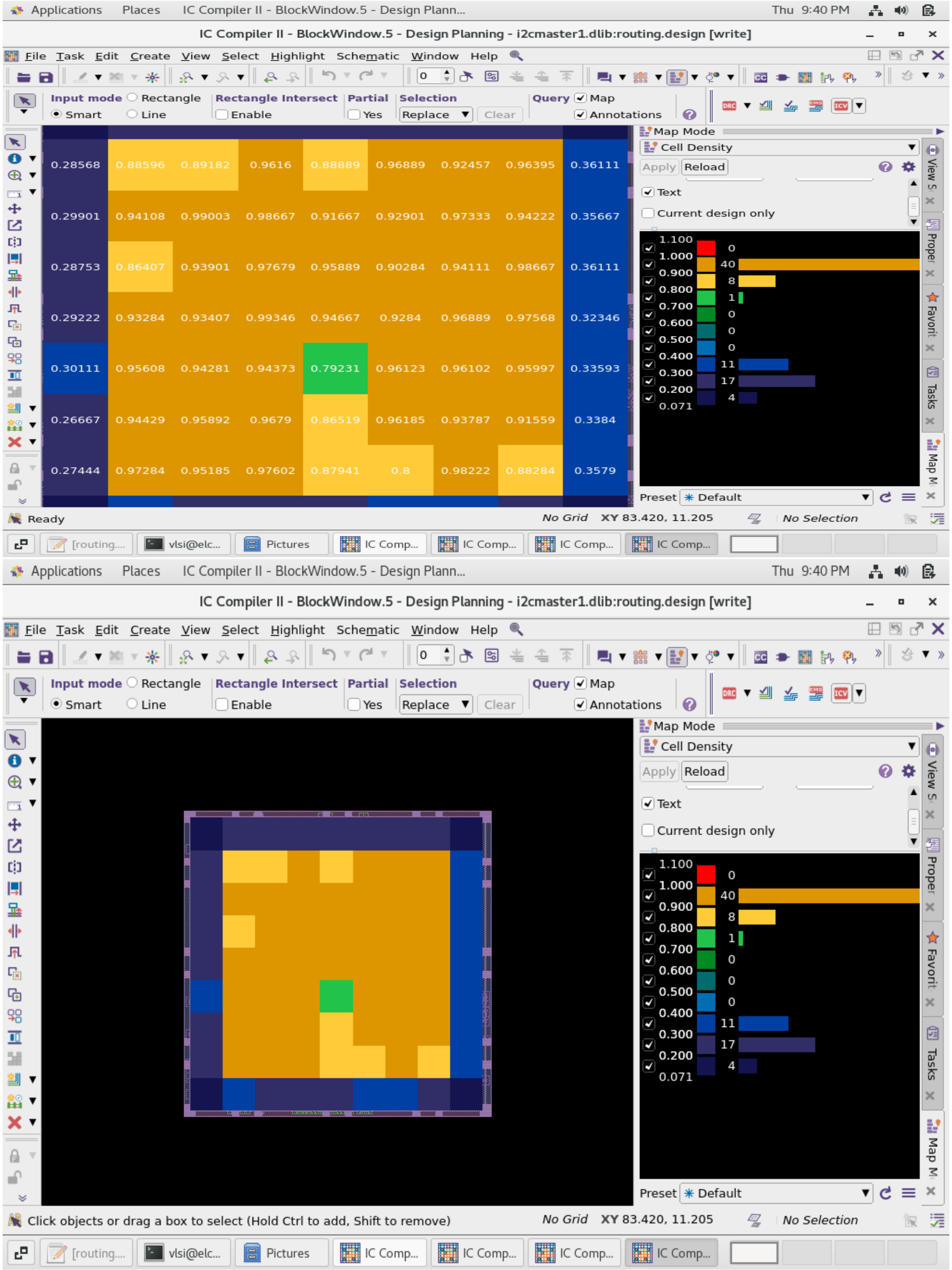


Figure12:Adding filler Cells



0.28568	0.88596	0.89182	0.9616	0.88889	0.96889	0.92457	0.96395	0.36111
0.29901	0.94108	0.99003	0.98667	0.91667	0.92901	0.97333	0.94222	0.35667
0.28753	0.86407	0.93901	0.97679	0.95889	0.90284	0.94111	0.98667	0.36111
0.29222	0.93284	0.93407	0.99346	0.94667	0.9284	0.96889	0.97568	0.32346
0.30111	0.95608	0.94281	0.94373	0.79231	0.96123	0.96102	0.95997	0.33593
0.26667	0.94429	0.95892	0.9679	0.86519	0.96185	0.93787	0.91559	0.3384
0.27444	0.97284	0.95185	0.97602	0.87941	0.8	0.98222	0.88284	0.3579

☑ 1.100	0
☑ 1.000	40
☑ 0.900	8
☑ 0.800	1
☑ 0.700	0
☑ 0.600	0
☑ 0.500	0
☑ 0.400	11
☑ 0.300	17
☑ 0.200	4
0.071	

Ready

No Grid XY 83.420, 11.205 No Selection

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)

No Grid XY 83.420, 11.205 No Selection

Figure 13:Cell Congestion view

Finally after all of these steps we have got the GDSII File with no violations and achieving the requirements, so we have done Synthesis of the code, Performing static timing analysis, Performing Formal Verification, Creating the floor plan, Creating the power plan, Performing clock tree synthesis, Placing and routing, Design and layout of the pad ring with all the required inputs and outputs, Signoff and final complete layout

Implementation of I2C protocol new approach:

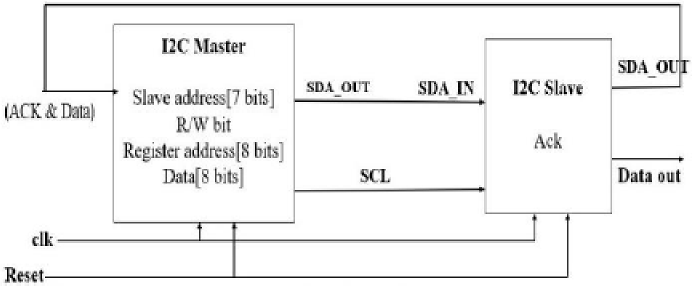


Figure 14: I2C block Diagram

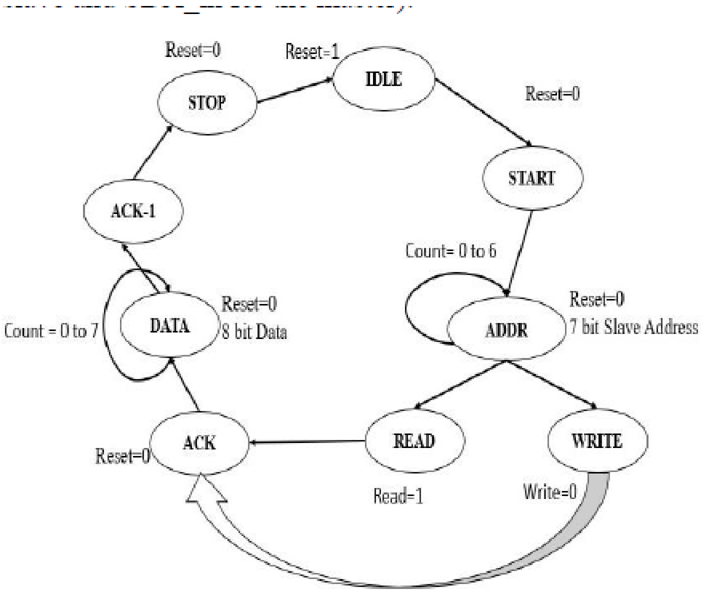


Figure 15:FSM of the Design

There are the following states

1. State 0 = IDLE (reset =1, initiate =1)

Else when reset =0 and initiate =0 then the following states appear.

2. State 1 = START CONDITION - SDA is pulled from high to low when SCL is still high

3. State 2= SLAVE ADDRESS – Normal mode so, 7-bit slave address is sent.

4. State 3= READ/WRITE BIT- Write is 0 and Read is 1.

5. State 4 = ACK/NACK BIT- SDA line is low if acknowledged and SDA line is high if NACK (not acknowledged)

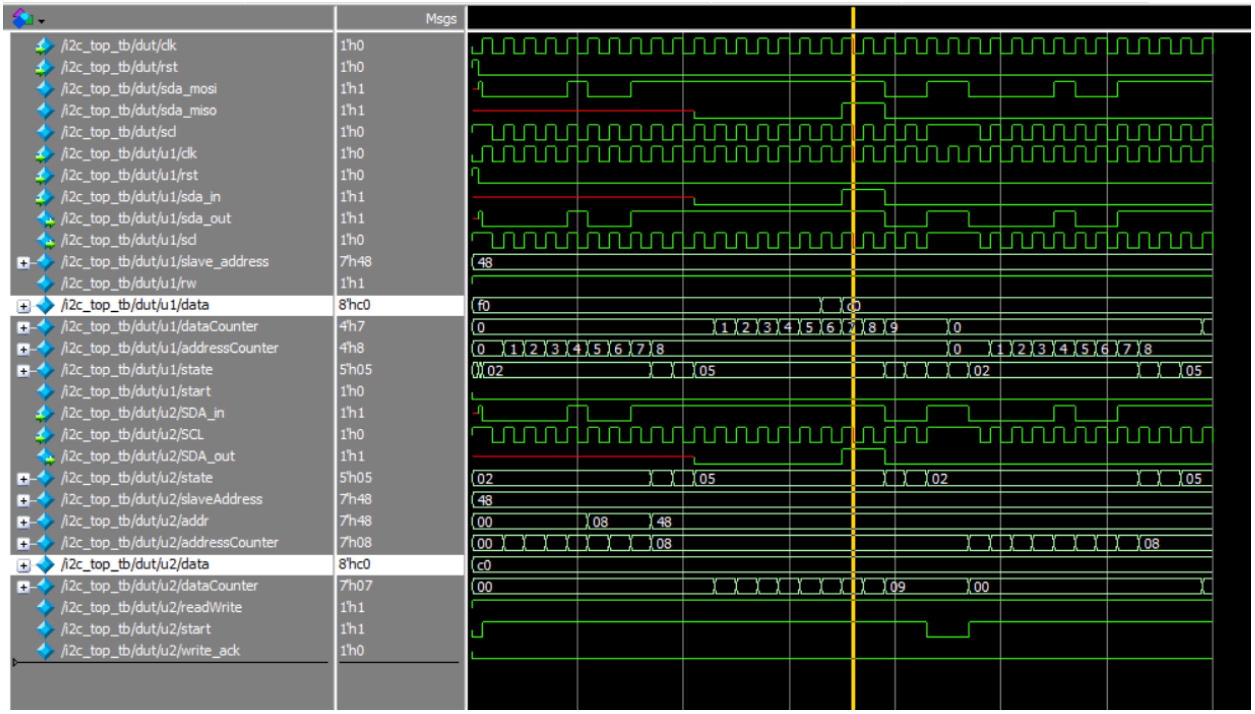
6. State 5 = DATA- 8-bit data is sent or received at once

7. State 6 = ACK_1 BIT – After receiving the data ACK bit sent i.e. SDA line is pulled low. NACK bit means SDA line is high

8. State 7 = STOP CONDITION- Master will terminate the communication by pulling the SDA line from low to high when SCL line is still high.

FSM (finite state machine) gives the exact flow of the Protocol. Using this FSM, I2C protocol can be implemented in Verilog where SDA line which is a bidirectional one can be taken as SDA_out and SDA_in for master as well as slave later they can be combined as SDA_om and SDA_os respectively.

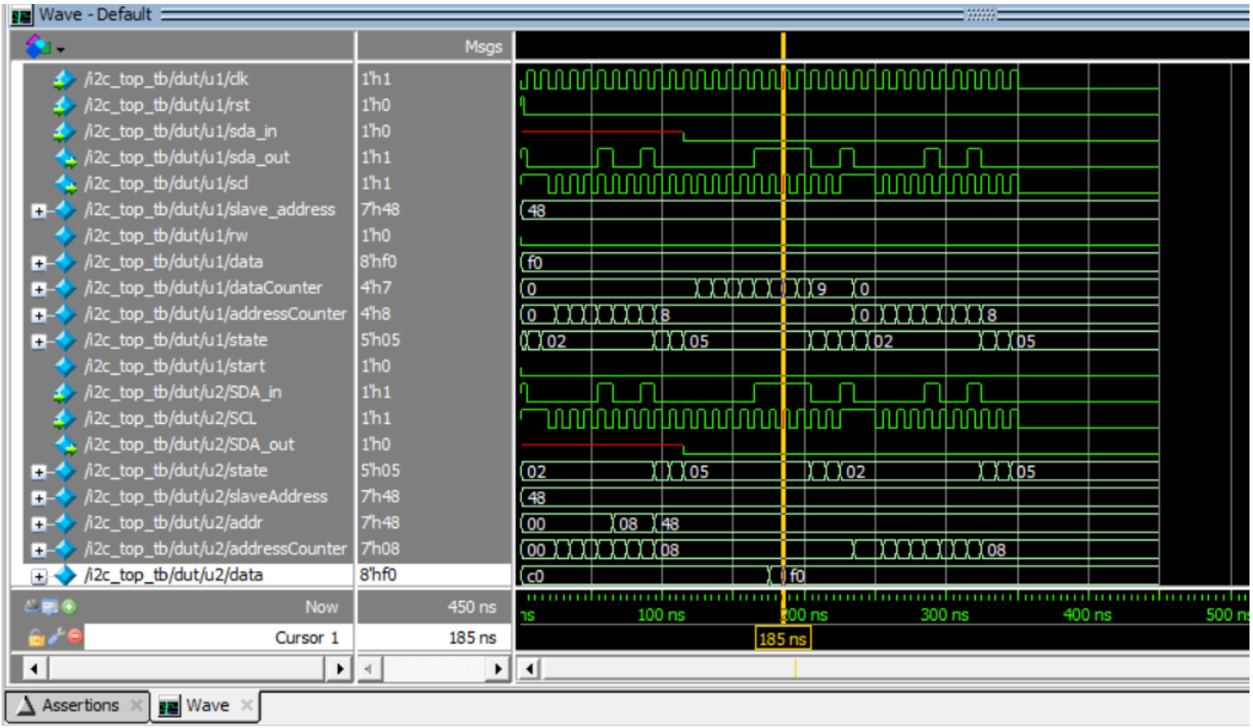
Simulation results:



	Msgs
/i2c_top_tb/dut/clk	1'h0
/i2c_top_tb/dut/rst	1'h0
/i2c_top_tb/dut/sda_mosi	1'h1
/i2c_top_tb/dut/sda_miso	1'h1
/i2c_top_tb/dut/scl	1'h0
/i2c_top_tb/dut/u1/clk	1'h0
/i2c_top_tb/dut/u1/rst	1'h0
/i2c_top_tb/dut/u1/sda_in	1'h1
/i2c_top_tb/dut/u1/sda_out	1'h1
/i2c_top_tb/dut/u1/scl	1'h0
/i2c_top_tb/dut/u1/slave_address	7'h48
/i2c_top_tb/dut/u1/rw	1'h1
/i2c_top_tb/dut/u1/data	8'hc0
/i2c_top_tb/dut/u1/dataCounter	4'h7
/i2c_top_tb/dut/u1/addressCounter	4'h8
/i2c_top_tb/dut/u1/state	5'h05
/i2c_top_tb/dut/u1/start	1'h0
/i2c_top_tb/dut/u2/SDA_in	1'h1
/i2c_top_tb/dut/u2/SCL	1'h0
/i2c_top_tb/dut/u2/SDA_out	1'h1
/i2c_top_tb/dut/u2/state	5'h05
/i2c_top_tb/dut/u2/slaveAddress	7'h48
/i2c_top_tb/dut/u2/addr	7'h48
/i2c_top_tb/dut/u2/addressCounter	7'h08
/i2c_top_tb/dut/u2/data	8'hc0
/i2c_top_tb/dut/u2/dataCounter	7'h07
/i2c_top_tb/dut/u2/readWrite	1'h1
/i2c_top_tb/dut/u2/start	1'h1
/i2c_top_tb/dut/u2/write_ack	1'h0

Figure15:read operation

It’s obvious that the data stored in the slave has been read by the master at u1/data where u1 is module of master while u2 is the module of slave

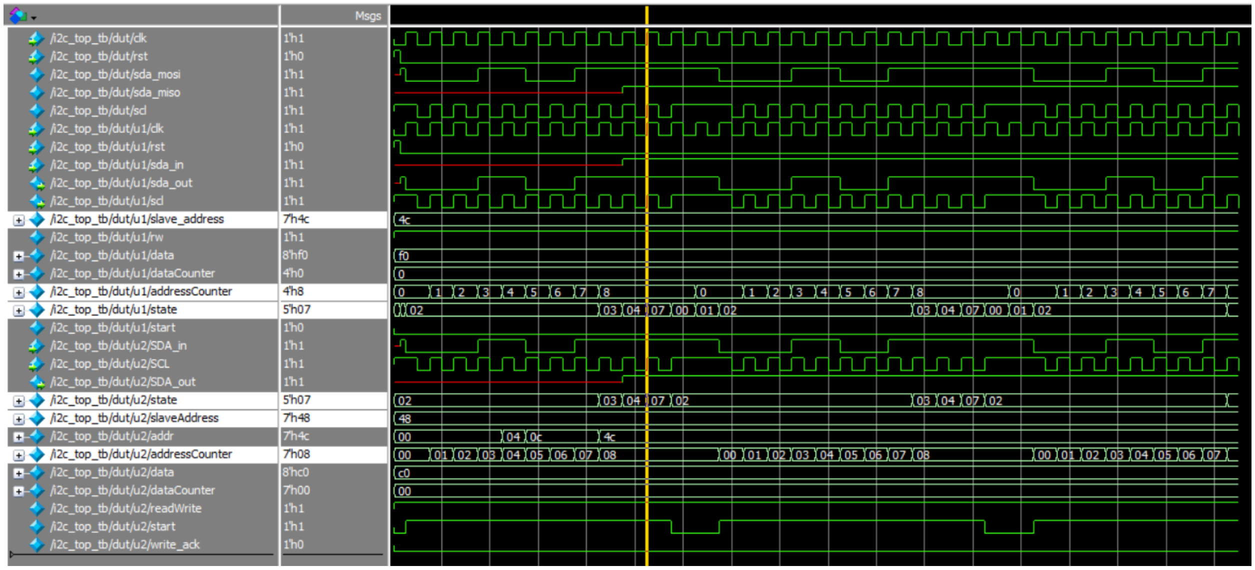


Wave - Default

	Msgs
/i2c_top_tb/dut/u1/clk	1'h1
/i2c_top_tb/dut/u1/rst	1'h0
/i2c_top_tb/dut/u1/sda_in	1'h0
/i2c_top_tb/dut/u1/sda_out	1'h1
/i2c_top_tb/dut/u1/scl	1'h1
/i2c_top_tb/dut/u1/slave_address	7'h48
/i2c_top_tb/dut/u1/rw	1'h0
/i2c_top_tb/dut/u1/data	8'hf0
/i2c_top_tb/dut/u1/dataCounter	4'h7
/i2c_top_tb/dut/u1/addressCounter	4'h8
/i2c_top_tb/dut/u1/state	5'h05
/i2c_top_tb/dut/u1/start	1'h0
/i2c_top_tb/dut/u2/SDA_in	1'h1
/i2c_top_tb/dut/u2/SCL	1'h1
/i2c_top_tb/dut/u2/SDA_out	1'h0
/i2c_top_tb/dut/u2/state	5'h05
/i2c_top_tb/dut/u2/slaveAddress	7'h48
/i2c_top_tb/dut/u2/addr	7'h48
/i2c_top_tb/dut/u2/addressCounter	7'h08
/i2c_top_tb/dut/u2/data	8'hf0
Now	450 ns
Cursor 1	185 ns

Figure16:Write operation

It’s obvious that the data stored in the master has been written by the master to slave at u2/data=8’f0 where u1 is module of master while u2 is the module of slave



	Msgs
/i2c_top_tb/dut/clk	1'h1
/i2c_top_tb/dut/rst	1'h0
/i2c_top_tb/dut/sda_mosi	1'h1
/i2c_top_tb/dut/sda_miso	1'h1
/i2c_top_tb/dut/scl	1'h1
/i2c_top_tb/dut/u1/clk	1'h1
/i2c_top_tb/dut/u1/rst	1'h0
/i2c_top_tb/dut/u1/sda_in	1'h1
/i2c_top_tb/dut/u1/sda_out	1'h1
/i2c_top_tb/dut/u1/scl	1'h1
/i2c_top_tb/dut/u1/slave_address	7'h4c
/i2c_top_tb/dut/u1/rw	1'h1
/i2c_top_tb/dut/u1/data	8'hf0
/i2c_top_tb/dut/u1/dataCounter	4'h0
/i2c_top_tb/dut/u1/addressCounter	4'h8
/i2c_top_tb/dut/u1/state	5'h07
/i2c_top_tb/dut/u1/start	1'h0
/i2c_top_tb/dut/u2/SDA_in	1'h1
/i2c_top_tb/dut/u2/SCL	1'h1
/i2c_top_tb/dut/u2/SDA_out	1'h1
/i2c_top_tb/dut/u2/state	5'h07
/i2c_top_tb/dut/u2/slaveAddress	7'h48
/i2c_top_tb/dut/u2/addr	7'h4c
/i2c_top_tb/dut/u2/addressCounter	7'h08
/i2c_top_tb/dut/u2/data	8'hc0
/i2c_top_tb/dut/u2/dataCounter	7'h00
/i2c_top_tb/dut/u2/readWrite	1'h1
/i2c_top_tb/dut/u2/start	1'h1
/i2c_top_tb/dut/u2/write_ack	1'h0

Figure 17:Wrong slave address sent by master

It’s obvious that the slave addres is 7’h48 while the address sent by the master is 7’h4c which is wrong so both master & slave go to the stop state as the checking is wrong