MAVEN SILICON

Centre of Excellence in VLSI

Design Compiler and ICC2

www.maven-silicon.com

Maven Silicon Confidential

All the presentations, books, documents [hard copies and soft copies] labs and projects [source code] that you are using and developing as part of the training course are the proprietary work of Maven Silicon and it is fully protected under copyright and trade secret laws. You may not view, use, disclose, copy, or distribute the materials or any information except pursuant to a valid written license from Maven Silicon

Design Compiler Flow

Basically Design Compiler optimizes a design to provide miniaturized and fastest representation of logical function. It also provides topographical technology, which allows us to predict accurate post-layout, timing, area and power during RTL synthesis. We know that Design compiler is a Graphical tool which helps us in finding a multicore-multimode designs and allows us to create and modify floorplan .I came across that it also reduces routing congestion and it improves area similarities. Usually we came across logic synthesis which is the process of converting Verilog to gate-level netlist which is mapped to a specific logic library and we check whether the synthesized design meets required functionality timing, power, and area requirements.

Design Flow

Basically design flow consists of three steps i.e.

- ✓ Synthesis
- ✓ Optimization
- ✓ **Compile**

Synthesis

As we know in this stage it generates a gate-level netlist from a RTL and design is created. Synthesis is a process where different inputs are required to carry out the design.

- ✓ Technology Files (.tf)
- ✓ Reference libraries(ref.lib)(.DB'S)
- ✓ RTL(Verilog code)
- ✓ Design constraints(.sdc)
- ✓ Floorplan constraint
- ✓ TLU-plus files(parasitic files)
- ✓ DFT inputs

Optimization

Followed by synthesis we have optimization in this step we implement a combination of library cells and we have to check whether we obtained required function, area and power.

Compile

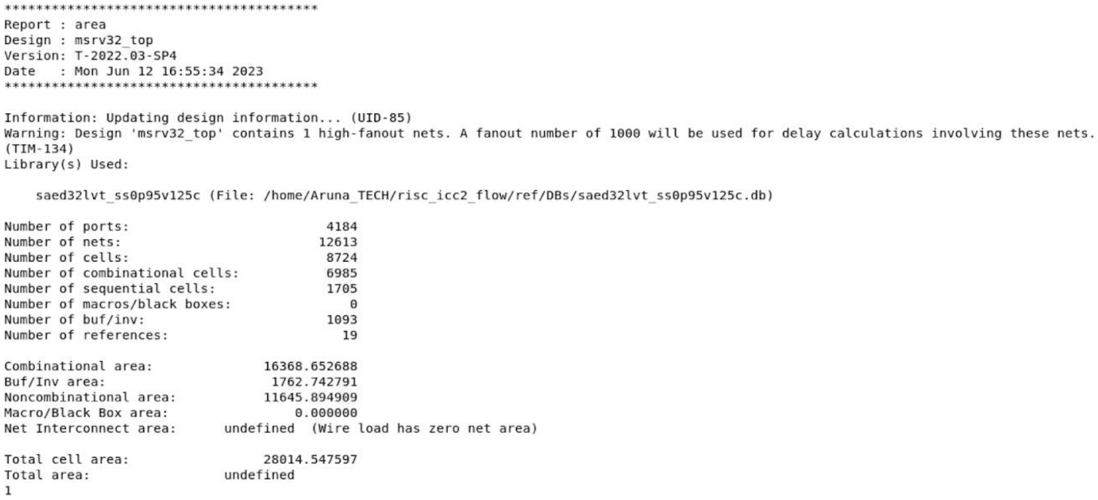
Compile is the Design Compiler process that executes the synthesis and optimization steps. After you read in the design and perform other necessary tasks, you run the "compile_ultra or compile" command to generate a gate-level netlist for the design.

The following are the output which are generated after synthesis

- ✓ Area
- ✓ Power
- ✓ Timing
- ✓ Design details
- ✓ Netlist
- ✓ Design constraints file
- ✓ Parasitic extraction
- ✓ SDF file (Standard delay format)

1. Area

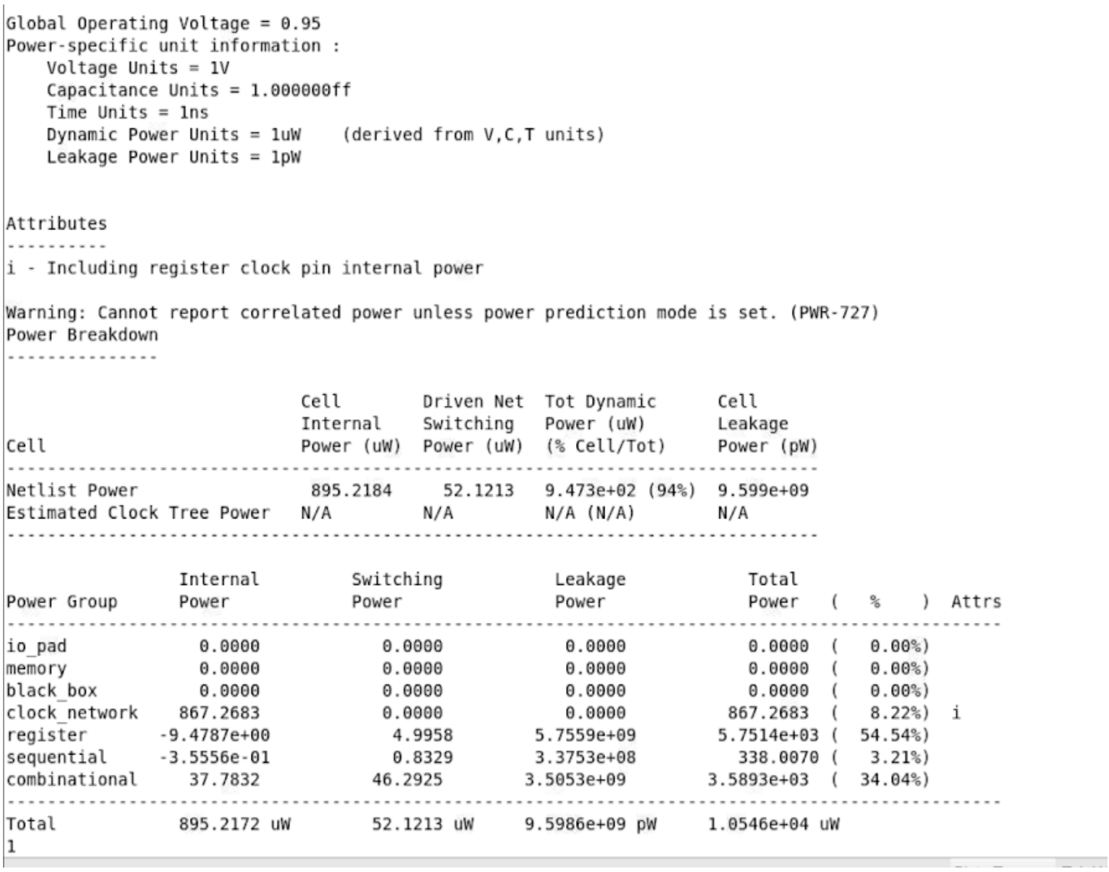
- ✓ The area report gives a summary of the area of each component in the current design .The report gives the number of gates and the area size based on the specified technology library.
- ✓ The command used to get area report is **report_area.**



```
****************************************
Report : area
Design : msrv32_top
Version: T-2022.03-SP4
Date   : Mon Jun 12 16:55:34 2023
****************************************

Information: Updating design information... (UID-85)
Warning: Design 'msrv32_top' contains 1 high-fanout nets. A fanout number of 1000 will be used for delay calculations involving these nets.
(TIM-134)
Library(s) Used:

    saed32lvt_ss0p95v125c (File: /home/Aruna_TECH/risc_icc2_flow/ref/DBs/saed32lvt_ss0p95v125c.db)

Number of ports:                         4184
Number of nets:                         12613
Number of cells:                         8724
Number of combinational cells:           6985
Number of sequential cells:              1705
Number of macros/black boxes:               0
Number of buf/inv:                       1093
Number of references:                      19

Combinational area:              16368.652688
Buf/Inv area:                     1762.742791
Noncombinational area:           11645.894909
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 28014.547597
Total area:                 undefined
1
```

2. Power

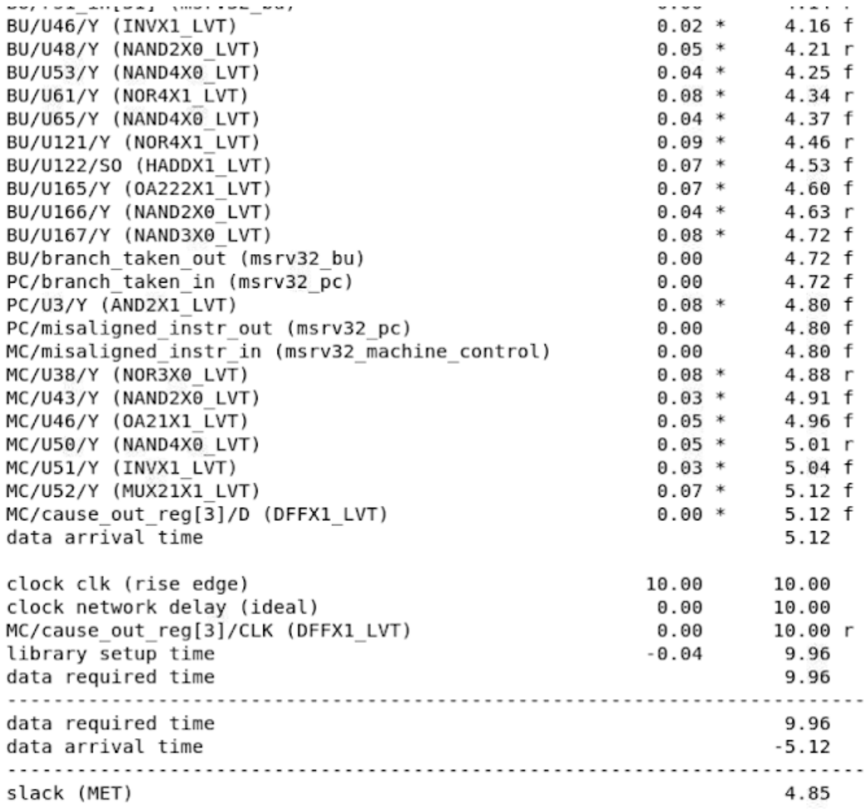
- ✓ Using Design compiler we can automatically minimizes power consumption at the RTL and gate level, and enables concurrent timing,area,power and test optimizations within the Design Compiler
- ✓ The command used to get power report is report_power.



```
Global Operating Voltage = 0.95
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1pW


Attributes
----------
i - Including register clock pin internal power

Warning: Cannot report correlated power unless power prediction mode is set. (PWR-727)
Power Breakdown
---------------

                             Cell        Driven Net  Tot Dynamic      Cell
                             Internal    Switching   Power (uW)       Leakage
Cell                         Power (uW)  Power (uW)  (% Cell/Tot)     Power (pW)
--------------------------------------------------------------------------------
Netlist Power                 895.2184     52.1213   9.473e+02 (94%)  9.599e+09
Estimated Clock Tree Power   N/A         N/A         N/A (N/A)        N/A
--------------------------------------------------------------------------------

                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000            0.0000            0.0000  (   0.00%)
memory             0.0000           0.0000            0.0000            0.0000  (   0.00%)
black_box          0.0000           0.0000            0.0000            0.0000  (   0.00%)
clock_network    867.2683           0.0000            0.0000          867.2683  (   8.22%)  i
register        -9.4787e+00          4.9958        5.7559e+09       5.7514e+03 (  54.54%)
sequential      -3.5556e-01          0.8329        3.3753e+08         338.0070 (   3.21%)
combinational     37.7832          46.2925        3.5053e+09       3.5893e+03  (  34.04%)
--------------------------------------------------------------------------------------------------
Total            895.2172 uW       52.1213 uW     9.5986e+09 pW     1.0546e+04 uW
1
```

3. Timing

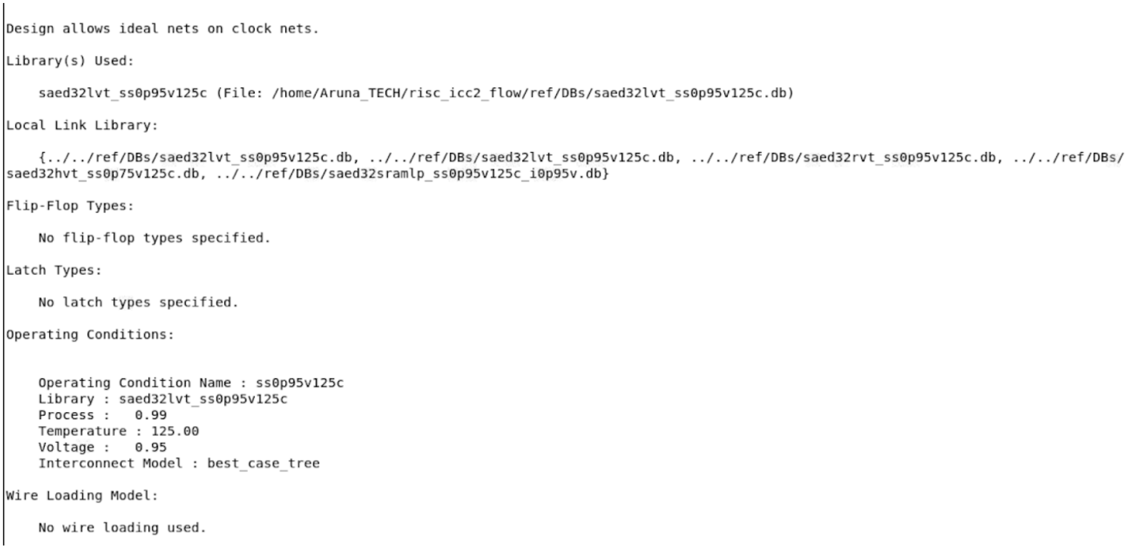
- ✓ In Timing we can consider the following types of paths for timing analysis: Clock path, a path from a clock input port or cell pin, through one or more buffers or inverters, to the clock pin of a sequential element; for data setup and hold checks. Clock-gating path.
- ✓ The command used to get timing report is **report_timing**.



```
BU/U46/Y (INVX1_LVT)                                   0.02 *     4.16 f
BU/U48/Y (NAND2X0_LVT)                                 0.05 *     4.21 r
BU/U53/Y (NAND4X0_LVT)                                 0.04 *     4.25 f
BU/U61/Y (NOR4X1_LVT)                                  0.08 *     4.34 r
BU/U65/Y (NAND4X0_LVT)                                 0.04 *     4.37 f
BU/U121/Y (NOR4X1_LVT)                                 0.09 *     4.46 r
BU/U122/SO (HADDX1_LVT)                                0.07 *     4.53 f
BU/U165/Y (OA222X1_LVT)                                0.07 *     4.60 f
BU/U166/Y (NAND2X0_LVT)                                0.04 *     4.63 r
BU/U167/Y (NAND3X0_LVT)                                0.08 *     4.72 f
BU/branch_taken_out (msrv32_bu)                        0.00       4.72 f
PC/branch_taken_in (msrv32_pc)                         0.00       4.72 f
PC/U3/Y (AND2X1_LVT)                                   0.08 *     4.80 f
PC/misaligned_instr_out (msrv32_pc)                    0.00       4.80 f
MC/misaligned_instr_in (msrv32_machine_control)        0.00       4.80 f
MC/U38/Y (NOR3X0_LVT)                                  0.08 *     4.88 r
MC/U43/Y (NAND2X0_LVT)                                 0.03 *     4.91 f
MC/U46/Y (OA21X1_LVT)                                  0.05 *     4.96 f
MC/U50/Y (NAND4X0_LVT)                                 0.05 *     5.01 r
MC/U51/Y (INVX1_LVT)                                   0.03 *     5.04 f
MC/U52/Y (MUX21X1_LVT)                                 0.07 *     5.12 f
MC/cause_out_reg[3]/D (DFFX1_LVT)                      0.00 *     5.12 f
data arrival time                                                 5.12

clock clk (rise edge)                                 10.00      10.00
clock network delay (ideal)                            0.00      10.00
MC/cause_out_reg[3]/CLK (DFFX1_LVT)                    0.00      10.00 r
library setup time                                    -0.04       9.96
data required time                                                9.96
------------------------------------------------------------------------
data required time                                                9.96
data arrival time                                                -5.12
------------------------------------------------------------------------
slack (MET)                                                       4.85
```

4. Design Details

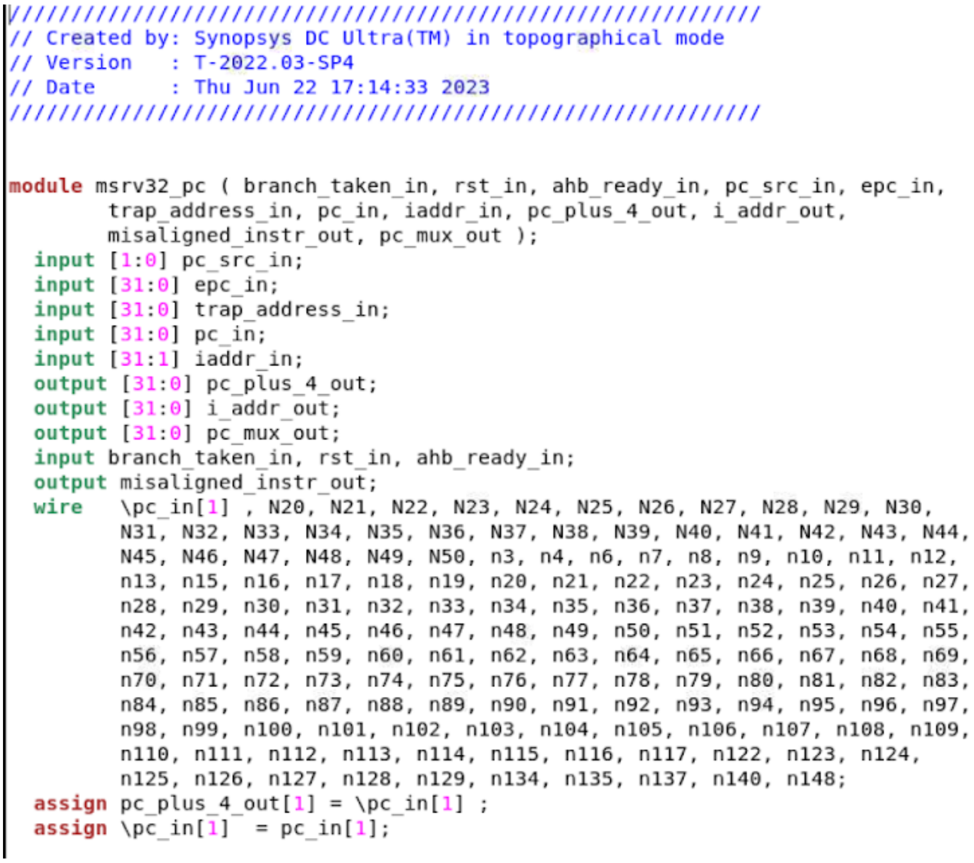
- ✓ The command used to get a Design details is report_design.



```
Design allows ideal nets on clock nets.

Library(s) Used:

    saed32lvt_ss0p95v125c (File: /home/Aruna_TECH/risc_icc2_flow/ref/DBs/saed32lvt_ss0p95v125c.db)

Local Link Library:

    {../../ref/DBs/saed32lvt_ss0p95v125c.db, ../../ref/DBs/saed32lvt_ss0p95v125c.db, ../../ref/DBs/saed32rvt_ss0p95v125c.db, ../../ref/DBs/
saed32hvt_ss0p75v125c.db, ../../ref/DBs/saed32sramlp_ss0p95v125c_i0p95v.db}

Flip-Flop Types:

    No flip-flop types specified.

Latch Types:

    No latch types specified.

Operating Conditions:


    Operating Condition Name : ss0p95v125c
    Library : saed32lvt_ss0p95v125c
    Process :   0.99
    Temperature : 125.00
    Voltage :   0.95
    Interconnect Model : best_case_tree

Wire Loading Model:

    No wire loading used.
```

5. Netlist

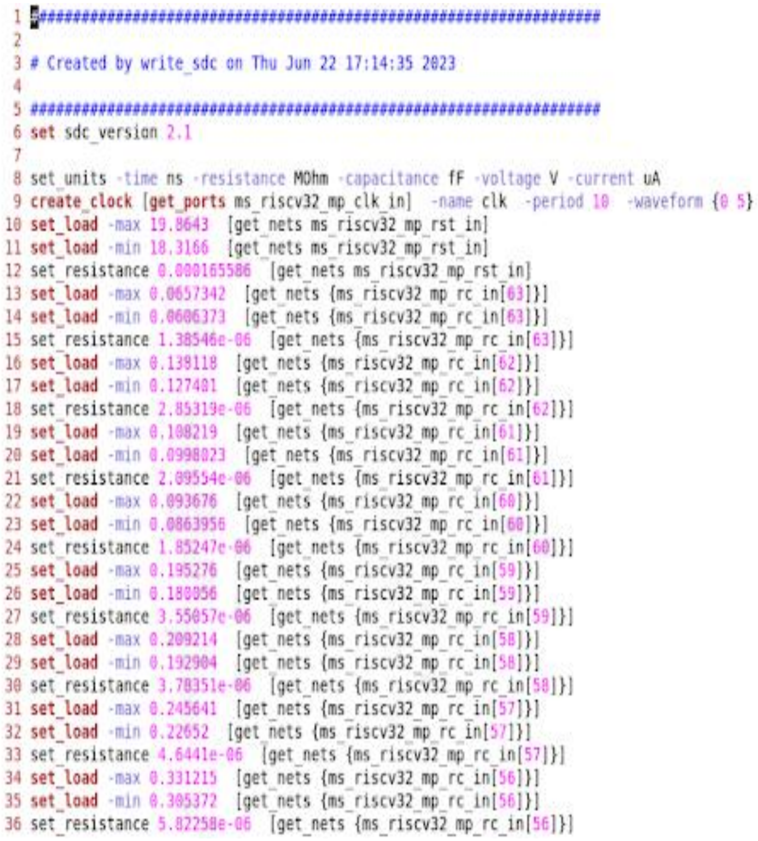
   ✓ The command used to get information about Netlist file is **write_verilog**



```
/////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Ultra(TM) in topographical mode
// Version   : T-2022.03-SP4
// Date      : Thu Jun 22 17:14:33 2023
/////////////////////////////////////////////////////////////////


module msrv32_pc ( branch_taken_in, rst_in, ahb_ready_in, pc_src_in, epc_in,
        trap_address_in, pc_in, iaddr_in, pc_plus_4_out, i_addr_out,
        misaligned_instr_out, pc_mux_out );
  input [1:0] pc_src_in;
  input [31:0] epc_in;
  input [31:0] trap_address_in;
  input [31:0] pc_in;
  input [31:1] iaddr_in;
  output [31:0] pc_plus_4_out;
  output [31:0] i_addr_out;
  output [31:0] pc_mux_out;
  input branch_taken_in, rst_in, ahb_ready_in;
  output misaligned_instr_out;
  wire   \pc_in[1] , N20, N21, N22, N23, N24, N25, N26, N27, N28, N29, N30,
         N31, N32, N33, N34, N35, N36, N37, N38, N39, N40, N41, N42, N43, N44,
         N45, N46, N47, N48, N49, N50, n3, n4, n6, n7, n8, n9, n10, n11, n12,
         n13, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27,
         n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41,
         n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55,
         n56, n57, n58, n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69,
         n70, n71, n72, n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83,
         n84, n85, n86, n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97,
         n98, n99, n100, n101, n102, n103, n104, n105, n106, n107, n108, n109,
         n110, n111, n112, n113, n114, n115, n116, n117, n122, n123, n124,
         n125, n126, n127, n128, n129, n134, n135, n137, n140, n148;
  assign pc_plus_4_out[1] = \pc_in[1] ;
  assign \pc_in[1]  = pc_in[1];
```

6. Design Constraints File

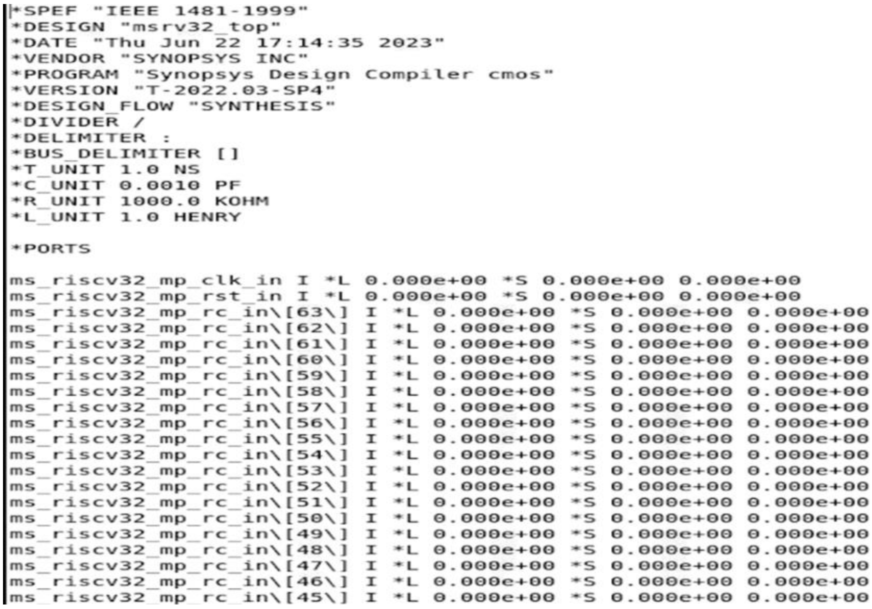
   ✓ The command used to get a design constraints are **write_sdc**



```
###################################################################

# Created by write_sdc on Thu Jun 22 17:14:35 2023

###################################################################
set sdc_version 2.1

set_units -time ns -resistance MOhm -capacitance fF -voltage V -current uA
create_clock [get_ports ms_riscv32_mp_clk_in]  -name clk  -period 10  -waveform {0 5}
set_load -max 19.8643  [get_nets ms_riscv32_mp_rst_in]
set_load -min 18.3166  [get_nets ms_riscv32_mp_rst_in]
set_resistance 0.000165586  [get_nets ms_riscv32_mp_rst_in]
set_load -max 0.0657342  [get_nets {ms_riscv32_mp_rc_in[63]}]
set_load -min 0.0606373  [get_nets {ms_riscv32_mp_rc_in[63]}]
set_resistance 1.38546e-06  [get_nets {ms_riscv32_mp_rc_in[63]}]
set_load -max 0.139118  [get_nets {ms_riscv32_mp_rc_in[62]}]
set_load -min 0.127481  [get_nets {ms_riscv32_mp_rc_in[62]}]
set_resistance 2.85319e-06  [get_nets {ms_riscv32_mp_rc_in[62]}]
set_load -max 0.108219  [get_nets {ms_riscv32_mp_rc_in[61]}]
set_load -min 0.0998023  [get_nets {ms_riscv32_mp_rc_in[61]}]
set_resistance 2.09554e-06  [get_nets {ms_riscv32_mp_rc_in[61]}]
set_load -max 0.093676  [get_nets {ms_riscv32_mp_rc_in[60]}]
set_load -min 0.0863956  [get_nets {ms_riscv32_mp_rc_in[60]}]
set_resistance 1.85247e-06  [get_nets {ms_riscv32_mp_rc_in[60]}]
set_load -max 0.195276  [get_nets {ms_riscv32_mp_rc_in[59]}]
set_load -min 0.180056  [get_nets {ms_riscv32_mp_rc_in[59]}]
set_resistance 3.55057e-06  [get_nets {ms_riscv32_mp_rc_in[59]}]
set_load -max 0.209214  [get_nets {ms_riscv32_mp_rc_in[58]}]
set_load -min 0.192904  [get_nets {ms_riscv32_mp_rc_in[58]}]
set_resistance 3.70351e-06  [get_nets {ms_riscv32_mp_rc_in[58]}]
set_load -max 0.245641  [get_nets {ms_riscv32_mp_rc_in[57]}]
set_load -min 0.22652  [get_nets {ms_riscv32_mp_rc_in[57]}]
set_resistance 4.6441e-06  [get_nets {ms_riscv32_mp_rc_in[57]}]
set_load -max 0.331215  [get_nets {ms_riscv32_mp_rc_in[56]}]
set_load -min 0.305372  [get_nets {ms_riscv32_mp_rc_in[56]}]
set_resistance 5.82258e-06  [get_nets {ms_riscv32_mp_rc_in[56]}]
```

7. Parasitic Extraction

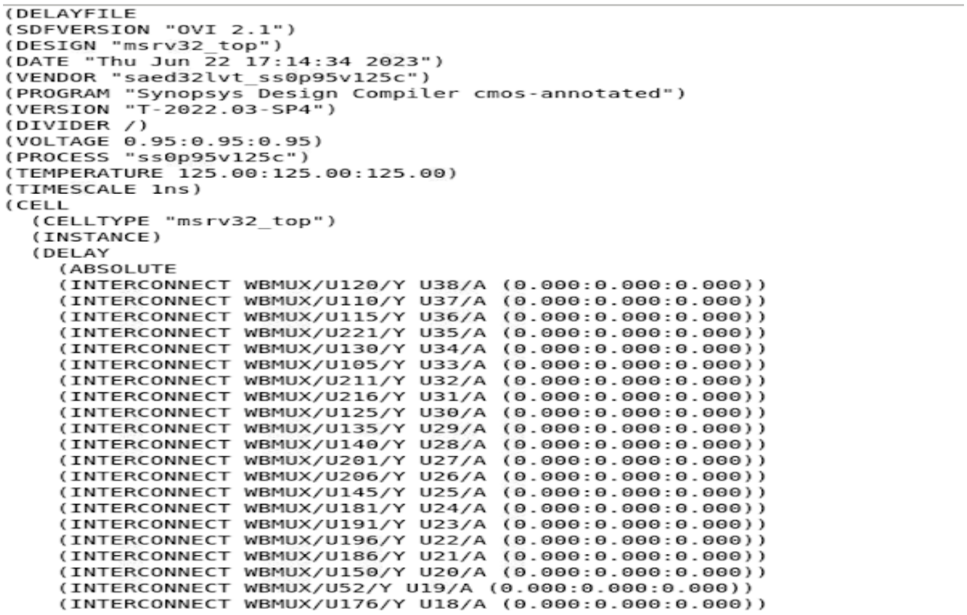
- ✓ The command used to get information about parasitic is write_parasitic



```
*SPEF "IEEE 1481-1999"
*DESIGN "msrv32_top"
*DATE "Thu Jun 22 17:14:35 2023"
*VENDOR "SYNOPSYS INC"
*PROGRAM "Synopsys Design Compiler cmos"
*VERSION "T-2022.03-SP4"
*DESIGN_FLOW "SYNTHESIS"
*DIVIDER /
*DELIMITER :
*BUS_DELIMITER []
*T_UNIT 1.0 NS
*C_UNIT 0.0010 PF
*R_UNIT 1000.0 KOHM
*L_UNIT 1.0 HENRY

*PORTS

ms_riscv32_mp_clk_in I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rst_in I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[63\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[62\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[61\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[60\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[59\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[58\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[57\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[56\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[55\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[54\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[53\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[52\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[51\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[50\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[49\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[48\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[47\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[46\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
ms_riscv32_mp_rc_in\[45\] I *L 0.000e+00 *S 0.000e+00 0.000e+00
```

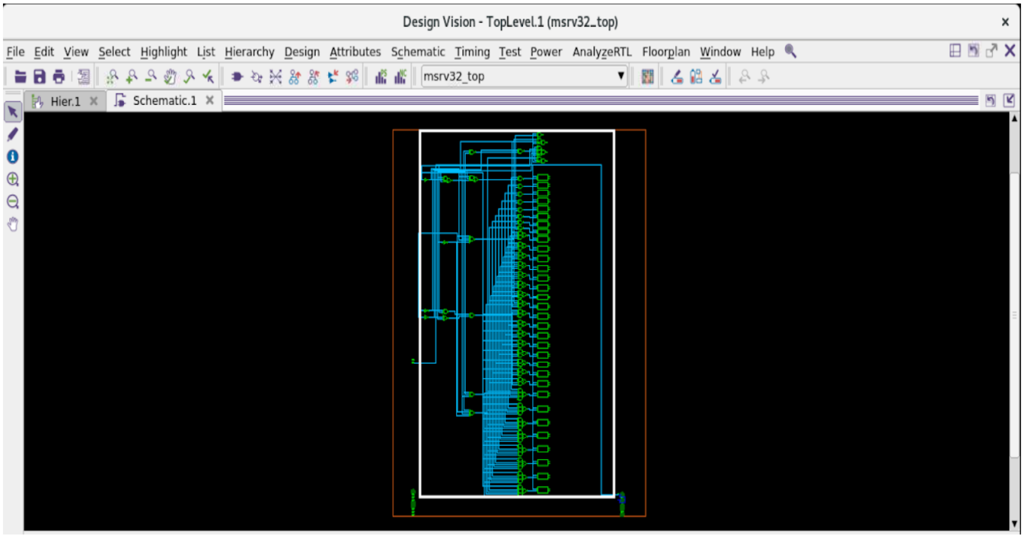
8. SDF File (Standard Delay File)

- ✓ The command used to get information about delay files is **write_sdf.**



```
(DELAYFILE
(SDFVERSION "OVI 2.1")
(DESIGN "msrv32_top")
(DATE "Thu Jun 22 17:14:34 2023")
(VENDOR "saed32lvt_ss0p95v125c")
(PROGRAM "Synopsys Design Compiler cmos-annotated")
(VERSION "T-2022.03-SP4")
(DIVIDER /)
(VOLTAGE 0.95:0.95:0.95)
(PROCESS "ss0p95v125c")
(TEMPERATURE 125.00:125.00:125.00)
(TIMESCALE 1ns)
(CELL
  (CELLTYPE "msrv32_top")
  (INSTANCE)
  (DELAY
    (ABSOLUTE
    (INTERCONNECT WBMUX/U120/Y U38/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U110/Y U37/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U115/Y U36/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U221/Y U35/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U130/Y U34/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U105/Y U33/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U211/Y U32/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U216/Y U31/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U125/Y U30/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U135/Y U29/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U140/Y U28/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U201/Y U27/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U206/Y U26/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U145/Y U25/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U181/Y U24/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U191/Y U23/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U196/Y U22/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U186/Y U21/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U150/Y U20/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U52/Y U19/A (0.000:0.000:0.000))
    (INTERCONNECT WBMUX/U176/Y U18/A (0.000:0.000:0.000))
```

- ✓ Schematic View Generated by Design Compiler



Design Vision - TopLevel.1 (msrv32_top)
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power AnalyzeRTL Floorplan Window Help
msrv32_top
Hier.1
Schematic.1

- ✓ We also have IC22 Compatible files which are generated by Design compiler
  - ✓ msrv32_top.floorplan
  - ✓ msrv32_top.icc2_script.tcl
  - ✓ msrv32_top.MCMM
  - ✓ msrv32_top.settings.tcl
  - ✓ msrv32_top.v
  - ✓ write_icc2_files.log

ICC II Compile Flow

ICC II Compiler is a complete netlist-GDS implementation, which, includes innovative for flat and hierarchical design planning, early design exploration, Congestion aware placement and optimization, clock tree synthesis, advanced node routing convergence, manufacturing compliance and signoff closure.

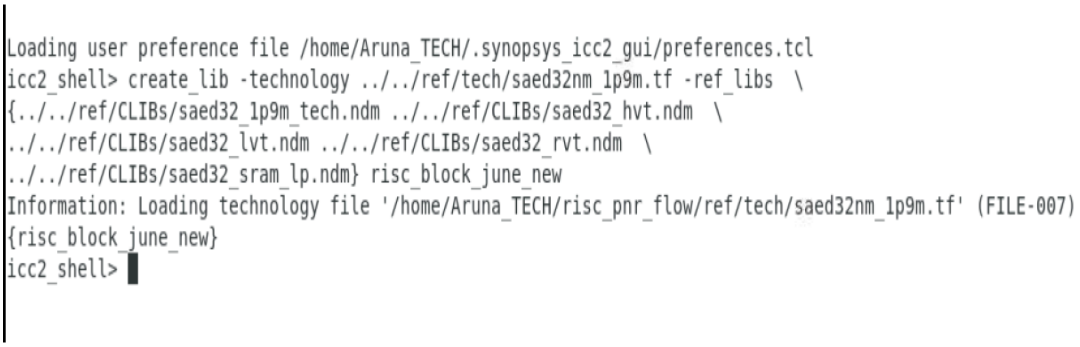
We have some set of inputs which we have to provide for ICC II tool such as

- ✓ Technology files(.tf) & Netlist
- ✓ NDM library's
- ✓ TLV-plus files
- ✓ Design constraints file from DC

ICC II tool is basically a PNR (Power and Routing) tool which can process some particular steps as follows.

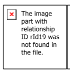
- ✓ Floor Planning
- ✓ Placement
- ✓ Clock Tree Synthesis
- ✓ Routing

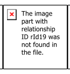
Step 1: First step in ICC II compiler is creating a library, we can create a library by using the same library file and Technological files .To create a library Create_lib command is used where in which it is supported with all the technology files and library name.

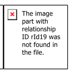


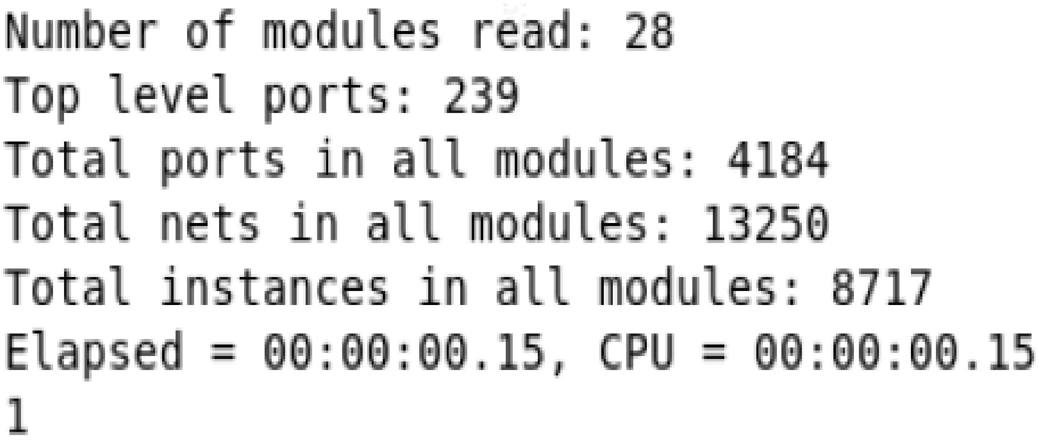
```
Loading user preference file /home/Aruna_TECH/.synopsys_icc2_gui/preferences.tcl
icc2_shell> create_lib -technology ../../ref/tech/saed32nm_1p9m.tf -ref_libs  \
{../../ref/CLIBs/saed32_1p9m_tech.ndm ../../ref/CLIBs/saed32_hvt.ndm  \
../../ref/CLIBs/saed32_lvt.ndm ../../ref/CLIBs/saed32_rvt.ndm  \
../../ref/CLIBs/saed32_sram_lp.ndm} risc_block_june_new
Information: Loading technology file '/home/Aruna_TECH/risc_pnr_flow/ref/tech/saed32nm_1p9m.tf' (FILE-007)
{risc_block_june_new}
icc2_shell> 
```

Step 2: In this step ICC II reads a libraries in the form of NDM's i.e., Reading RTL and the reading SDC which is generated from Design compiler.


The image part with relationship ID rId19 was not found in the file.


The image part with relationship ID rId19 was not found in the file.


The image part with relationship ID rId19 was not found in the file.

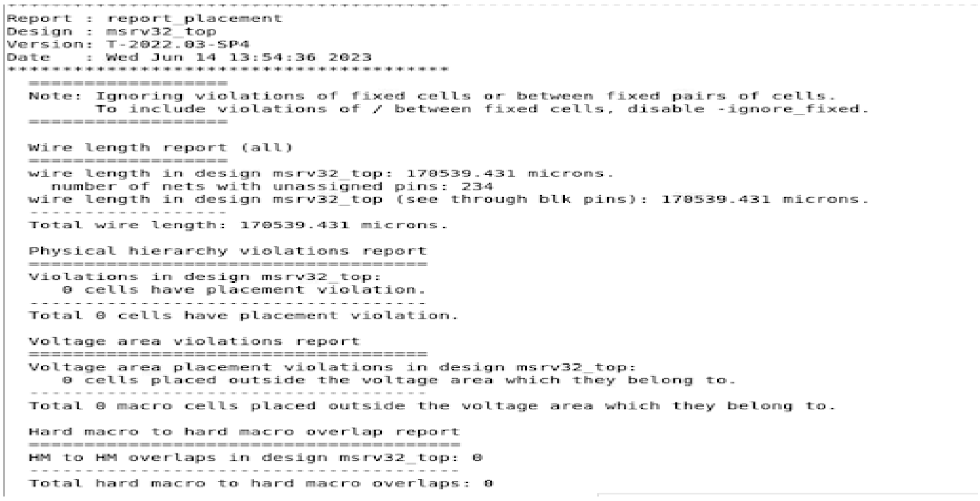


```
Number of modules read: 28
Top level ports: 239
Total ports in all modules: 4184
Total nets in all modules: 13250
Total instances in all modules: 8717
Elapsed = 00:00:00.15, CPU = 00:00:00.15
1
```

Step 3: This step involves floor planning which in turn involves determining the locations, shape, size of modules in a chip and we can get a rough estimation of the chip area, delay and the wiring congestion thereby providing a ground work for layout.

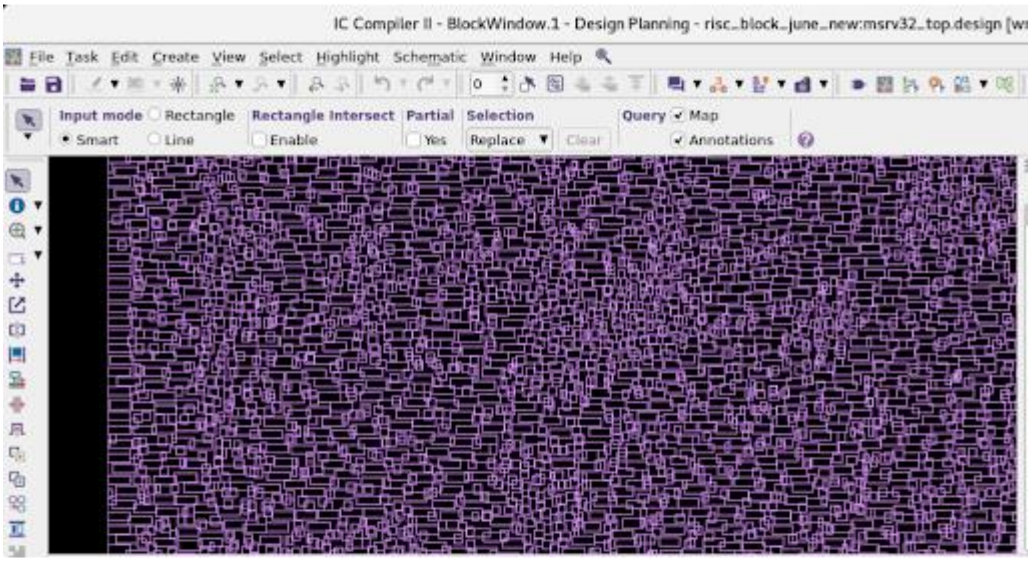
Step 4: In this step a default parasitic model is created with targeting of TLU +files and its related map files & all the related scenarios targeting to different PVTs should be created.

Step 5: Placement Creation



```
****************************************
Report : report_placement
Design : msrv32_top
Version: T-2022.03-SP4
Date   : Wed Jun 14 13:54:36 2023
****************************************
  ==================
  Note: Ignoring violations of fixed cells or between fixed pairs of cells.
        To include violations of / between fixed cells, disable -ignore_fixed.
  ==================

  Wire length report (all)
  ==================
  wire length in design msrv32_top: 170539.431 microns.
    number of nets with unassigned pins: 234
  wire length in design msrv32_top (see through blk pins): 170539.431 microns.
  ------------------
  Total wire length: 170539.431 microns.

  Physical hierarchy violations report
  ====================================
  Violations in design msrv32_top:
     0 cells have placement violation.
  ------------------------------------
  Total 0 cells have placement violation.

  Voltage area violations report
  ====================================
  Voltage area placement violations in design msrv32_top:
     0 cells placed outside the voltage area which they belong to.
  ------------------------------------
  Total 0 macro cells placed outside the voltage area which they belong to.

  Hard macro to hard macro overlap report
  =======================================
  HM to HM overlaps in design msrv32_top: 0
  ---------------------------------------
  Total hard macro to hard macro overlaps: 0
```

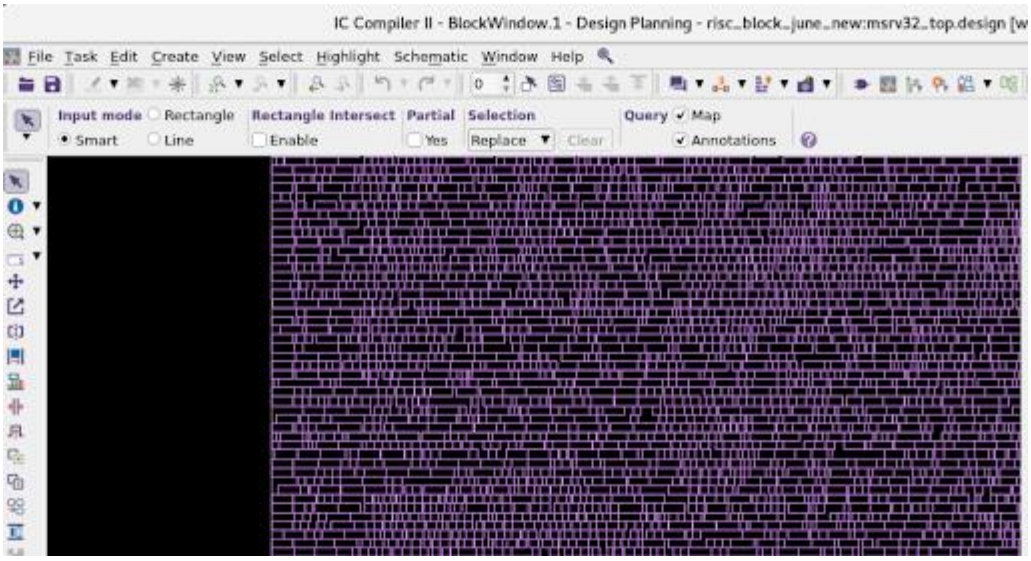
In create placement stage all the Standard cells which are placed inside the floor plan area in the global method, here all the cells are placed in overlapped manner.



IC Compiler II - BlockWindow.1 - Design Planning - risc_block_june_new:msrv32_top.design [w
File Task Edit Create View Select Highlight Schematic Window Help
0
Rectangle Intersect
Clear

Input mode ● Smart ○ Rectangle ○ Line
Rectangle Intersect □ Enable
Partial □ Yes
Selection Replace
Query ☑ Map ☑ Annotations

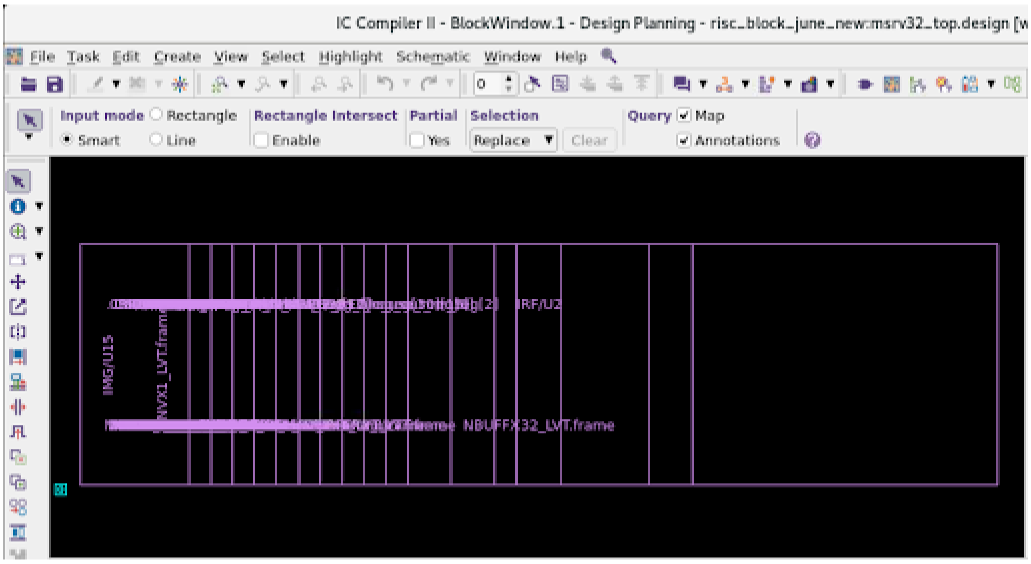
To avoid overlapping we have to go for legalized placement, so that Standard cells are placed without overlapping.



IC Compiler II - BlockWindow.1 - Design Planning - risc_block_june_new:msrv32_top.design [w
File Task Edit Create View Select Highlight Schematic Window Help
0
Rectangle Intersect
Clear

Input mode ● Smart ○ Rectangle ○ Line
Rectangle Intersect □ Enable
Partial □ Yes
Selection Replace
Query ☑ Map ☑ Annotations

After placing a standard cells we are going to place the input output pins.


IC Compiler II - BlockWindow.1 - Design Planning - risc_block_june_new:msrv32_top.design [w
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
IRF/U2
IMG/U15
NVX1_LVT.frame
NBUFFX32_LVT.frame

Step 6: Clock Tree Synthesis

Clock Tree Synthesis is a technique for distributing the clock equally among all sequential parts. The purpose of Clock tree Synthesis is provided the placement data as well as the clock tree limitations as input .
A signal with constant rise and fall with ideally equal width (50 % rise and 50 % fall of the signal width) helps to control data propagation through the clock elements like Flip-Flop, Latches etc. The clock source mostly present in top-level design and from there propagation. PLL Oscillator like constant sources are being used normally in designs to get the clock.

Inputs of CTS

- ✓ Placement DB
- ✓ CTS Spec File

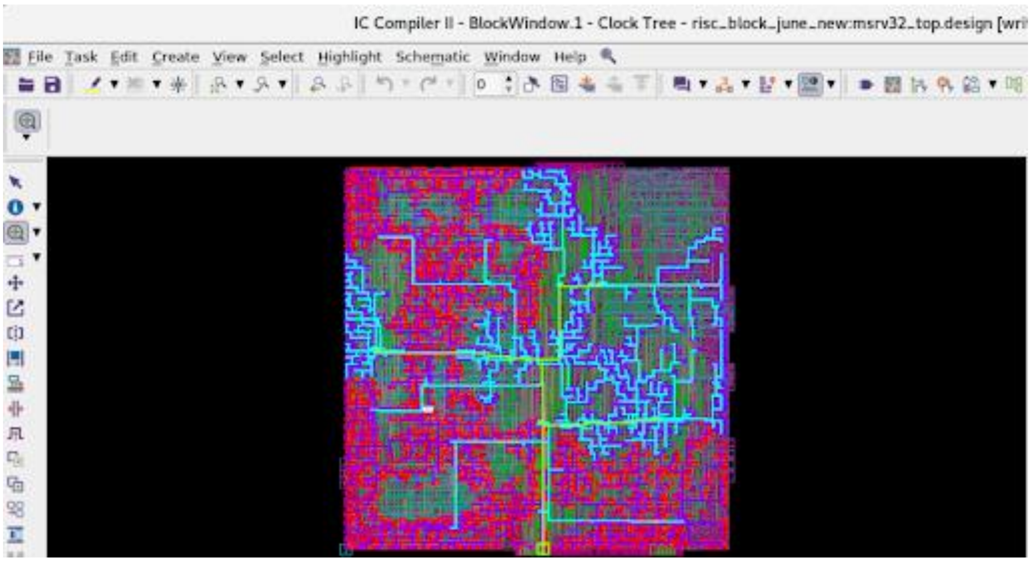
Placement DB

Placement DB contains Placement completed Netlist, DEF, LIB, LEF, SDC, UPF and other information which contain all the files from the placement database. This can be a zipped file. This DB is also known as PLACE EXIT db. Which means we are not going to do any standard cell placement and related things here onward.

CTS Spec File

CTS spec file contains the below information:

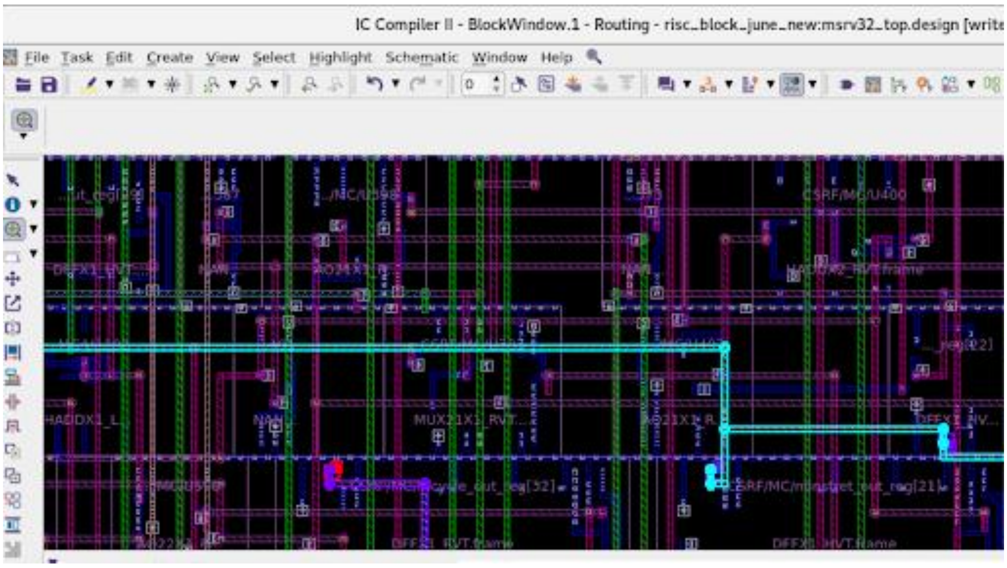
- ✓ Inverters or buffers to be defined which will be used to balance the clock tree.
- ✓ CTS Exceptions (End points of clock tree).
- ✓ Skew group information.
- ✓ Contains target Skew, max target transition and other timing constraints as per clock tree.
- ✓ Top layer and bottom layer route info. VIA's information which will be used during clock route.
- ✓ 6 Clock related info (Generated clocks {Eg. Clock divider, Clock multiplier etc}).
- ✓ 7 NDR Rule definition.


IC Compiler II - BlockWindow.1 - Clock Tree - risc_block_june_new:msrv32_top.design [wri
File
Task
Edit
Create
View
Select
Highlight
Schematic
Window
Help
0

Step 7: Routing

Routing is the process of path selection in any network. A computer network is made of many machines, called nodes, and paths or links that connect those nodes. Communication between two nodes in an interconnected network can take place through many different paths.
There are three types of Routing:

- ✓ Static Routing.
- ✓ Default Routing.
- ✓ Dynamic Routing.


IC Compiler II - BlockWindow.1 - Routing - risc_block_june_new:msrv32_top.design [write
File Task Edit Create View Select Highlight Schematic Window Help
DFFX1_HVT...
HADDX1_L...
MUX21X1_RVT...
AO21X1_R...
CSRF/MC/U400
HADDX2_RVT.frame
DFFX1_HV...
DFFX1_RVT.frame
DFFX1_HVT.frame