

OpenPiton in Action

Princeton University

<http://openpiton.org>



OpenPiton

ASIC SYNTHESIS AND BACKEND

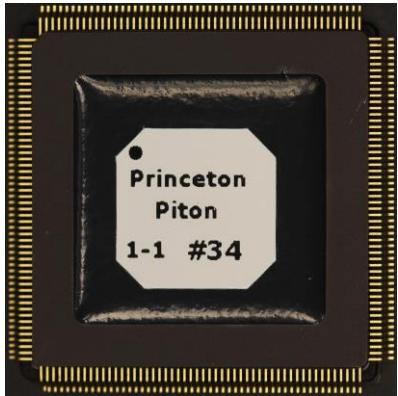
What's in the Box?

- Synthesis
 - Synopsys Design Compiler
- Static timing analysis (STA)
 - Synopsys Primetime
- Formal equivalence checking (RVS)
 - Synopsys Formality
- Place and route (PAR)
 - Synopsys IC Compiler
- Layout versus schematic (LVS)
 - Mentor Graphics Calibre
- Design rule checking (DRC)
 - Mentor Graphics Calibre

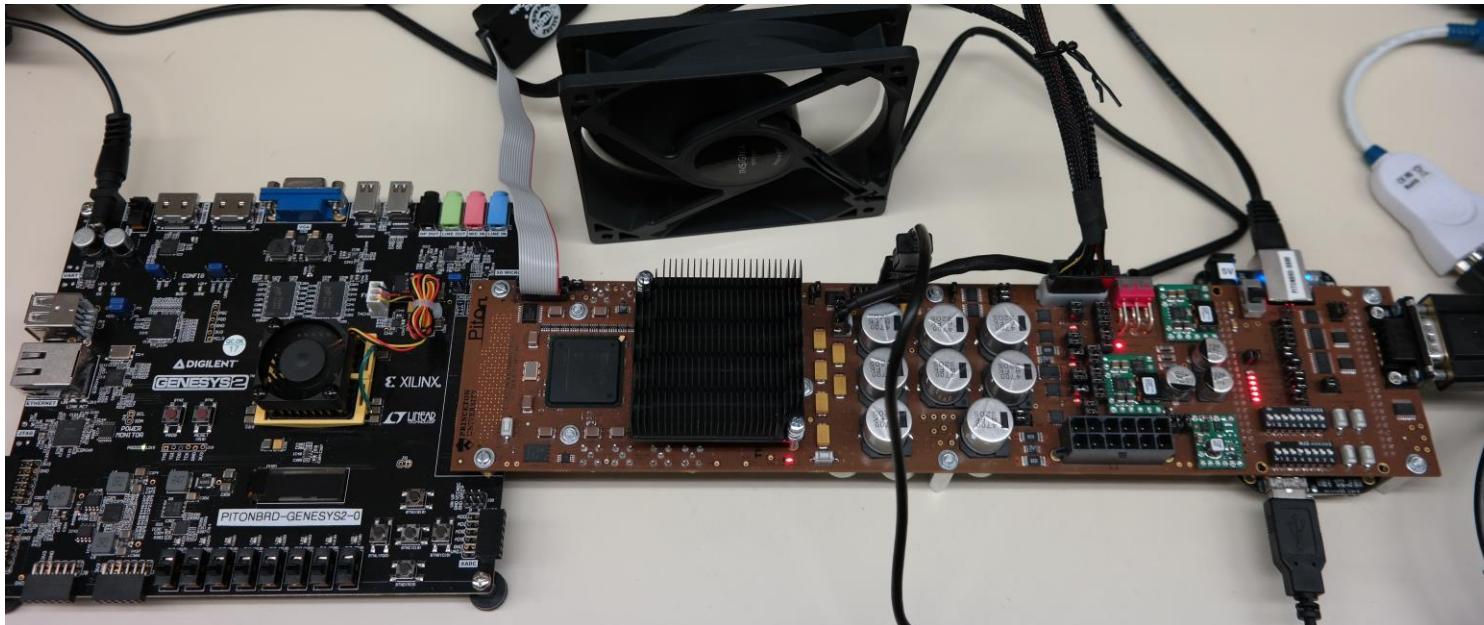
Why is it Useful?

- Research studies
 - Architecture, EDA, and other HW research
- ASIC tapeout
- Education

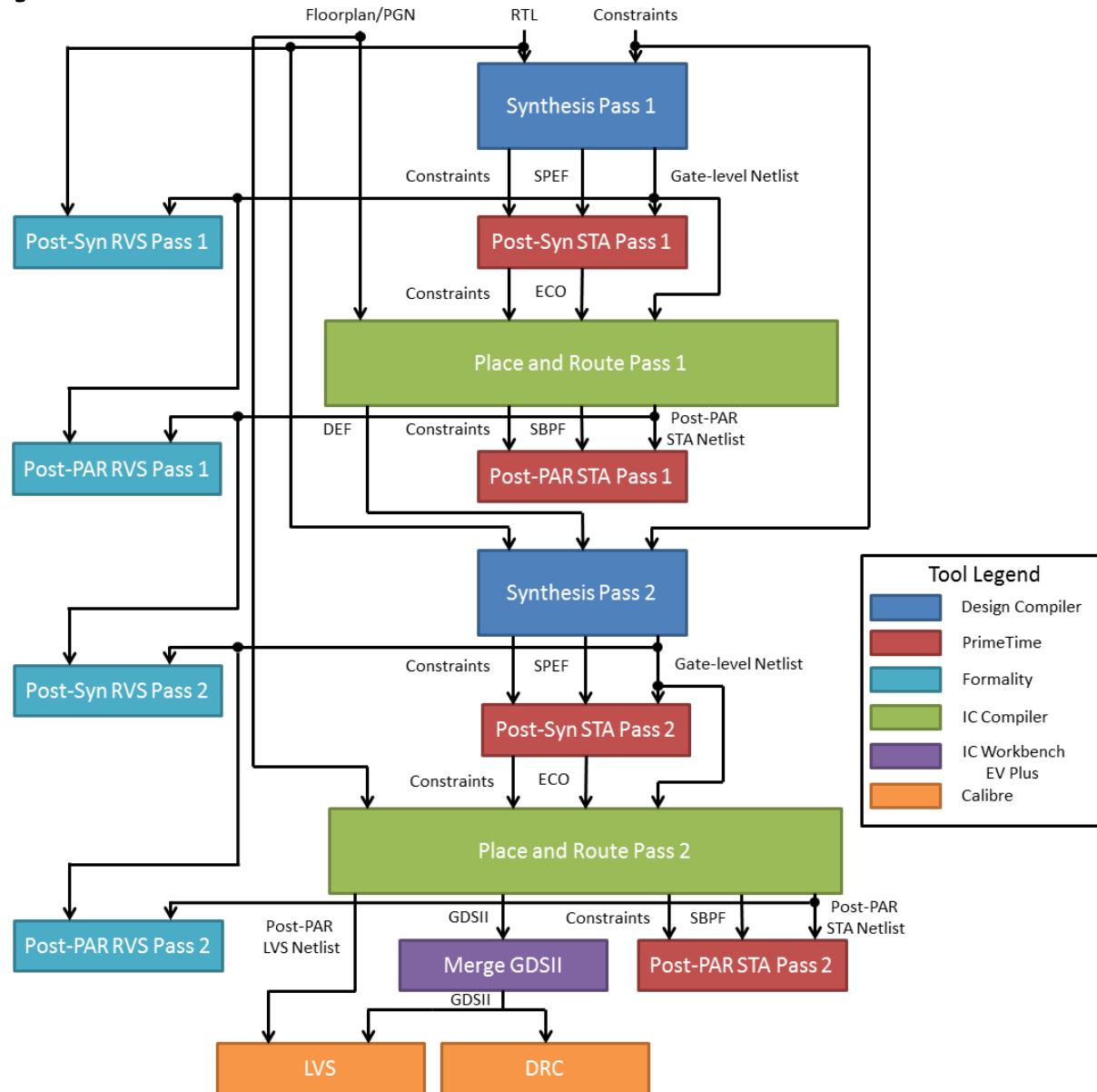
Piton ASIC



- 25 tiles
- IBM 32nm SOI
- 36 mm² (6mm x 6mm)
- 1 GHz Target Frequency
- Tested working in silicon!



Synthesis and Backend Flow



What do you need?

- OpenPiton
- Synopsys License
 - Tools and Reference Methodology (RM)
- Mentor Graphics License
 - Calibre (for LVS and DRC only)
- Standard cell library and process development kit

Getting Started

- Download Synopsys-RM
- Patch Synopsys-RM
- Familiarize with directory structure and scripts
- Port to process technology
- Running the flow

Download Synopsys-RM

- Synopsys Solvnet
- See OpenPiton
Synthesis and Backend
Manual
 - Specify version
 - Specify settings
- Broader support

2.2.8 Reference Methodology

The OpenPiton synthesis and back-end flow is based on the Synopsys Reference Methodology (RM). Because of IP issues, the OpenPiton synthesis and back-end scripts have been released as a patch to the Synopsys RM. Thus, users will need access to Synopsys RM in order to make use of the OpenPiton synthesis and back-end flow. The OpenPiton synthesis and back-end flow supports patching from the following versions and settings of the Synopsys RM:

- Synthesis
 - DC-RM.I-2013.12-SP2
 - * Settings:
 - RTL Source Format: VERILOG
 - QoR Strategy: DEFAULT
 - Physical Guidance: TRUE
 - Hierarchical Flow: TRUE
 - MCMM Flow: FALSE
 - Multi-Voltage UPF: FALSE
 - Clock Gating: TRUE
 - Leakage Power: TRUE
 - DFT Synthesis: FALSE
 - Lynx Compatible: FALSE
 - Static Timing Analysis
 - PT-RM.I-2013.12

Patching Synopsys-RM

```
[openpiton]mmckeown@hanoi$ ls  
README README.md build docs piton  
[openpiton]mmckeown@hanoi$ synrm_patch -h  
usage: synrm_patch [-h] [-g] [-d DC_RM_PATH] [-p PT_RM_PATH] [-i ICC_RM_PATH]
```

OpenPiton generate back-end flow from Synopsys RM scripts and OpenPiton patch. This script can also be used to generate patches from the current OpenPiton back-end flow state.

optional arguments:

- h, --help show this help message and exit
- g, --gen Generate a patch from the current state of OpenPiton back-end scripts (default is to generate back-end flow from Synopsys RM scripts and OpenPiton patch)
- d DC_RM_PATH, --dc_rm_path DC_RM_PATH Path to directory containing extracted Synopsys Design Compiler RM scripts
- p PT_RM_PATH, --pt_rm_path PT_RM_PATH Path to directory containing extracted Synopsys Primetime RM scripts
- i ICC_RM_PATH, --icc_rm_path ICC_RM_PATH Path to directory containing extracted Synopsys IC Compiler RM scripts

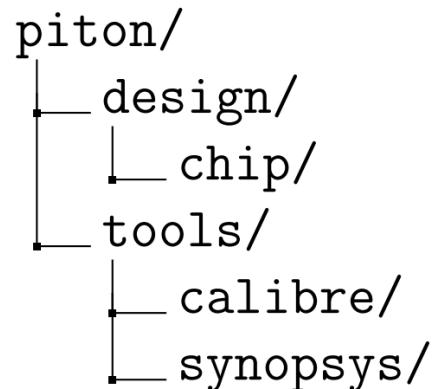
```
[openpiton]mmckeown@hanoi$ synrm_patch --dc_rm_path=../solvnet_scripts/DC-RM_I-2013.12-SP2/ --pt_rm_path=../solvnet_scripts/PT-RM_I-2013.12/ --icc_rm_path=../solvnet_scripts/ICC-RM_I-2013.12-SP4/
```

Patching Synopsys-RM

```
[openpiton]mmckeown@hanoi$ synrm_patch --dc_rm_path=../solvnet_scripts/DC-RM_I-2013.12-SP2/ --pt_rm_path=../solvnet_scripts/PT-RM_I-2013.12/ --icc_rm_path=../solvnet_scripts/ICC-RM_I-2013.12-SP4/
synrm_patch: #####
synrm_patch: # OpenPiton Patch Synopsys RM Flow #
synrm_patch: #####
synrm_patch: Checking integrity of Synopsys RM...
synrm_patch: Synopsys RM integrity check passed.
synrm_patch: Copying Synopsys RM files from '../solvnet_scripts/DC-RM_I-2013.12-SP2/' to '/tank/mmckeown/research/projects/piton/openpiton/piton/tools/synopsis/script/'...
synrm_patch: Successfully copied Synopsys RM files from '../solvnet_scripts/DC-RM_I-2013.12-SP2/'.
synrm_patch: Patching Synopsys RM '../solvnet_scripts/DC-RM_I-2013.12-SP2/' to OpenPiton...
synrm_patch: Successfully patched Synopsys RM '../solvnet_scripts/DC-RM_I-2013.12-SP2/' to OpenPiton.
synrm_patch: Copying Synopsys RM files from '../solvnet_scripts/PT-RM_I-2013.12/' to '/tank/mmckeown/research/projects/piton/openpiton/piton/tools/synopsis/script/'...
synrm_patch: Successfully copied Synopsys RM files from '../solvnet_scripts/PT-RM_I-2013.12/'.
synrm_patch: Patching Synopsys RM '../solvnet_scripts/PT-RM_I-2013.12/' to OpenPiton...
synrm_patch: Successfully patched Synopsys RM '../solvnet_scripts/PT-RM_I-2013.12/' to OpenPiton.
synrm_patch: Copying Synopsys RM files from '../solvnet_scripts/ICC-RM_I-2013.12-SP4/' to '/tank/mmckeown/research/projects/piton/openpiton/piton/tools/synopsis/script/'...
synrm_patch: Successfully copied Synopsys RM files from '../solvnet_scripts/ICC-RM_I-2013.12-SP4/'.
synrm_patch: Patching Synopsys RM '../solvnet_scripts/ICC-RM_I-2013.12-SP4/' to OpenPiton...
synrm_patch: Successfully patched Synopsys RM '../solvnet_scripts/ICC-RM_I-2013.12-SP4/' to OpenPiton.
synrm_patch: Checking integrity of OpenPiton flow...
synrm_patch: OpenPiton integrity checks passed.
synrm_patch: Successfully patched Synopsys RM to OpenPiton flow.
[openpiton]mmckeown@hanoi$
```

Directory Structure and Scripts

- All scripts written in Tcl
- Two primary locations
 - Module generic scripts
 - Module specific scripts



```
[sparc]mmckeown@hanoi$ pwd  
/tank/mmckeown/research/projects/piton/openpiton/piton/design/chip/tile/sparc  
[sparc]mmckeown@hanoi$ ls  
exu ffu ifu lsu mul rtl spu srams synopsys tlu  
[sparc]mmckeown@hanoi$ ls rtl/  
Flist.sparc_common bw_clk_cl_sparc_cmp.v cpx_spc_buf.v sparc.v spc_pcx_buf.v  
Flist.sparc_top cfg_asi.v cpx_spc_rpt.v sparc_core.v  
[sparc]mmckeown@hanoi$ ls synopsys/script/  
calibre.lvs.excpt floorplan.tcl module_setup.tcl preplace_srams.tcl pt.excpt  
connect_pg.tcl icc.excpt pgn.tcl preroute_stdcells.tcl sparc_core.constraints.tcl
```

Porting to a Process Technology

- \${OPENPITON_DRC_RUN+x} tcl
 - \${OPENPITON_LVS_RUN+x} tcl
 - \${OPENPITON_GDS2SP_BBOX_CELLS+x} tcl
- ```
Do not include module specific SRAMs in the below lists - these variables
DRC Runset and optional golden GDS file
OPENPITON_CALIBRE_DRC_DECK="CALIBRE_DRC_DECK_HERE (.drc.cal)"
OPENPITON_CALIBRE_GOLDEN_GDS_FILE="" ; # Optional, (.gds)
OPENPITON_GDS2SP_BBOX_CELLS="" ; # space separated list of standard cells
OPENPITON_LVS_BB0X_CELLS="" ; # space separated list of standard cells

LVS Runset
OPENPITON_CALIBRE_LVS_DECK="CALIBRE_LVS_DECK_HERE (.lvs.cal)"

GDS2SP (for doing LVS) and LVS black box cells
These can be due to internal LVS problems on IP macros,
std cells that don't get outputted from ICC, etc.
OPENPITON_GDS2SP_BB0X_CELLS="" ; # space separated list of standard cells
OPENPITON_LVS_BB0X_CELLS="" ; # space separated list of standard cells

Put any environment variables needed by DRC and LVS here
Examples (should be specified by DRC/LVS deck):
export TECHDIR=
export LAYOUT_SYSTEM=GDSII

if [-z ${OPENPITON_DRC_RUN+x}]; then
 # Put any DRC specific environment variables here
 :
fi

if [-z ${OPENPITON_LVS_RUN+x}]; then
 # Put any LVS specific environment variables here
 :
fi
```

# Porting to a Process Technology

Table 3: Process specific synthesis and backend scripts. Referenced from  `${PITON_ROOT}/piton/tools/synopsys/script/`

| Process Specific File Path                              | Description                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>common/env_setup.tcl</code>                       | Environment variable setup. Gets environment variables that are used throughout the scripts.                                                                                                                                                                                           |
| <code>common/process_setup.tcl</code>                   | Main process specific setup file. Includes standard cell libraries, technology files, layer mapping files, Milkyway libraries, etc.                                                                                                                                                    |
| <code>common/calibre_env</code>                         | Calibre process specific setup file. Sets the DRC and LVS decks and any environment variables used by these decks.                                                                                                                                                                     |
| <code>common/design_setup.tcl</code>                    | Setup variables specific to a design. Includes things like clock frequency, allowed routing metal layers, etc.                                                                                                                                                                         |
| <code>common/floorplan/common_pgn.tcl</code>            | Preroutes the power and ground network (PGN) for modules without SRAMs or other IP macros. For modules with SRAMs or other IP macros, a module specific PGN script will be required.                                                                                                   |
| <code>common/floorplan/core_pgn_mesh.tpl</code>         | Templates for PGN meshes. Called from <code>common/floorplan/common_pgn.tcl</code> and module specific PGN scripts.                                                                                                                                                                    |
| <code>common/floorplan/common_post_floorplan.tcl</code> | Post floorplan steps that are common to all modules. This is usually process specific but is also sometimes optional. We used this to place certain cells at a specific density throughout a module after floorplanning is complete, per a requirement from the foundry.               |
| <code>common/dbl_via_setup.tcl</code>                   | Double via definition script. This is usually process specific but is also sometimes optional. Can define which vias are used for double via insertion.                                                                                                                                |
| <code>common/pt_eco_drc_buf.tcl</code>                  | This is also optional, but specifies a list of standard cell buffers that Synopsys PT can use for DRC fixing.                                                                                                                                                                          |
| <code>common/vt_group_setup.tcl</code>                  | Optional, can group target libraries into threshold voltage groups for reporting. Can generate a report of what percentage of different groups were used, useful for knowing things like what portion of your critical path is made up of the lowest threshold voltage standard cells. |

# Porting to a Process Technology

- Module specific scripts suggested for review:
  - module\_setup.tcl
  - floorplan.tcl
  - <design\_name>.constraints.tcl

# Running the Flow

Table 4: OpenPiton Synthesis and Back-end Flow Run Commands

| Command                | Flow Step                              | Tool                                 | Checking Script         |
|------------------------|----------------------------------------|--------------------------------------|-------------------------|
| <code>rsyn</code>      | Synthesis                              | Synopsys Design Compiler             | <code>csyn</code>       |
| <code>rsta</code>      | Static Timing Analysis                 | Synopsys Primetime                   | <code>csta</code>       |
| <code>rrvs</code>      | RTL vs. Schematic Equivalence Checking | Synopsys Formality                   | <code>crvs</code>       |
| <code>rpar</code>      | Place and Route                        | Synopsys IC Compiler                 | <code>cpar</code>       |
| <code>reco</code>      | Run ECO                                | Synopsys IC Compiler                 | <code>cpar</code>       |
| <code>merge_gds</code> | Merge GDSII Designs                    | Synopsys IC Workbench Edit/View Plus | <code>cmerge_gds</code> |
| <code>rdrc</code>      | Design Rule Checking                   | Mentor Graphics Calibre              | <code>cdrc</code>       |
| <code>rlvs</code>      | Layout vs. Schematic Checking          | Mentor Graphics Calibre              | <code>clvs</code>       |
| <code>rftf</code>      | Full tool flow                         | All of the above                     | N/A                     |

Table 5: OpenPiton Synthesis and Back-end Flow Supported Modules

| Module Name               | Description                                     | Purpose                                          |
|---------------------------|-------------------------------------------------|--------------------------------------------------|
| <code>ffu</code>          | OpenSPARC T1 core floating-point front-end unit | Small module with one SRAM macro                 |
| <code>sparc</code>        | OpenSPARC T1 core                               | Large module with many SRAM macros               |
| <code>dynamic_node</code> | OpenPiton on-chip network router                | Small module with no IP macros                   |
| <code>tile</code>         | OpenPiton tile                                  | Large, hierarchical module with many SRAM macros |
| <code>chip</code>         | OpenPiton top-level chip                        | Large, top-level hierarchical module             |

| # Format:    | BlockPath              | BlockSynMemReq | BlockPARMemReq | FTFPasses | ECO |
|--------------|------------------------|----------------|----------------|-----------|-----|
| # BlockID    |                        |                |                |           |     |
| chip         | chip                   | 48000          | 128000         | 1         | Y   |
| tile         | chip/tile              | 32000          | 128000         | 1         | Y   |
| dynamic_node | chip/tile/dynamic_node | 8000           | 16000          | 1         | N   |
| sparc        | chip/tile/sparc        | 64000          | 128000         | 1         | Y   |
| ffu          | chip/tile/sparc/ffu    | 8000           | 12000          | 1         | Y   |

# Launch Flow

```
[openpiton]mmckeown@hanoi$ rftf sparc --slurm
```

| # Format: | BlockID      | BlockPath              | BlockSynMemReq | BlockPARMemReq | FTFPasses | ECC |
|-----------|--------------|------------------------|----------------|----------------|-----------|-----|
|           | chip         | chip                   | 48000          | 128000         | 1         | Y   |
|           | tile         | chip/tile              | 32000          | 128000         | 1         | Y   |
|           | dynamic_node | chip/tile/dynamic_node | 8000           | 16000          | 1         | N   |
|           | sparc        | chip/tile/sparc        | 64000          | 128000         | 1         | Y   |
|           | ffu          | chip/tile/sparc/ffu    | 8000           | 12000          | 1         | Y   |

```
Submitted batch job 147185
Submitted batch job 147186
Submitted batch job 147187
Submitted batch job 147188
Submitted batch job 147189
Submitted batch job 147190
Submitted batch job 147191
Submitted batch job 147192
Submitted batch job 147193
```

```
[openpiton]mmckeown@hanoi$ squeue
```

| JOBID  | NAME            | USER     | ST | TIME | NODES | CPUS | QOS   | NODELIST(REASON) |
|--------|-----------------|----------|----|------|-------|------|-------|------------------|
| 147188 | ecol_spark      | mmckeown | PD | 0:00 | 1     | 10   | short | (Dependency)     |
| 147192 | drc_spark       | mmckeown | PD | 0:00 | 1     | 10   | short | (Dependency)     |
| 147193 | lvs_spark       | mmckeown | PD | 0:00 | 1     | 10   | short | (Dependency)     |
| 147183 | stal_dc_spark   | mmckeown | PD | 0:00 | 1     | 1    | short | (Dependency)     |
| 147184 | rvs1_dc_spark   | mmckeown | PD | 0:00 | 1     | 1    | short | (Dependency)     |
| 147186 | stal_icc_spark  | mmckeown | PD | 0:00 | 1     | 1    | short | (Dependency)     |
| 147187 | rvs_1_icc_spark | mmckeown | PD | 0:00 | 1     | 1    | short | (Dependency)     |
| 147189 | stal_eco_spark  | mmckeown | PD | 0:00 | 1     | 1    | short | (Dependency)     |
| 147190 | rvs_1_eco_spark | mmckeown | PD | 0:00 | 1     | 1    | short | (Dependency)     |
| 147191 | mgds_spark      | mmckeown | PD | 0:00 | 1     | 1    | short | (Dependency)     |
| 147185 | par1_spark      | mmckeown | PD | 0:00 | 1     | 10   | long  | (Dependency)     |
| 147182 | syn1_spark      | mmckeown | R  | 0:04 | 1     | 6    | short | hanoi            |

# Flow Runtimes

| <b>Submodule</b>  | <b>SYN</b> | <b>PAR</b> | <b>STA</b> | <b>ECO</b> | <b>DRC</b> | <b>LVS</b> | <b>Total</b> |
|-------------------|------------|------------|------------|------------|------------|------------|--------------|
| <b>NoC Router</b> | 0.32       | 2.35       | 0.02       | N/A        | 0.04       | 0.04       | 2.80         |
| <b>L1.5</b>       | 0.40       | 15.45      | 0.05       | 3.55       | 0.24       | 0.12       | 34.18        |
| <b>Core</b>       | 1.16       | 36.82      | 0.19       | 3.51       | 1.09       | 0.32       | 78.04        |
| <b>Tile</b>       | 1.11       | 22.95      | 0.12       | 2.69       | 2.10       | 0.49       | 55.89        |
| <b>Chip</b>       | 4.20       | 79.19      | 0.67       | 11.36      | 9.04       | 0.93       | 104.91       |

Table 5: Time durations (in hours) of selected synthesis and back-end stages for selected submodules. If any stage executes more than once (STA or for multi-pass flow), the maximum duration is shown.

| <b>Submodule</b>  | <b>SYN</b> | <b>PAR</b> | <b>STA</b> | <b>ECO</b> | <b>DRC</b> | <b>LVS</b> | <b>Peak</b> |
|-------------------|------------|------------|------------|------------|------------|------------|-------------|
| <b>NoC Router</b> | 5.38       | 3.55       | 0.57       | N/A        | 1.40       | 1.33       | 5.38        |
| <b>L1.5</b>       | 19.91      | 5.97       | 1.43       | 5.00       | 1.73       | 1.46       | 19.91       |
| <b>Core</b>       | 27.19      | 7.99       | 1.96       | 15.05      | 2.54       | 4.03       | 27.19       |
| <b>Tile</b>       | 28.04      | 7.71       | 1.90       | 13.62      | 2.48       | 8.84       | 28.04       |
| <b>Chip</b>       | 8.37       | 64.54      | 1.34       | 64.65      | 8.54       | >41        | 64.65       |

Table 6: Peak memory usage (in GByte) of selected synthesis and back-end stages for selected submodules. If any stage executes more than once (STA or for multi-pass flow), the maximum peak usage is shown.

# Flow Reports

Table 6: OpenPiton Synthesis and Back-end Flow Results Locations. Referenced from module specific `synopsys` directory.

| Command                | Results Location                                                                                                 |
|------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>rsyn</code>      | <code>reports/dc_shell</code> , <code>reports/dc_shell_pass*</code>                                              |
| <code>rsta</code>      | <code>reports/pt_shell</code> , <code>reports/pt_shell_dc_pass*</code> , <code>reports/pt_shell_icc_pass*</code> |
| <code>rrvs</code>      | <code>reports/fm_shell</code> , <code>reports/fm_shell_dc_pass*</code> , <code>reports/fm_shell_icc_pass*</code> |
| <code>rpar</code>      | <code>reports/icc_shell</code> , <code>reports/icc_shell_pass*</code>                                            |
| <code>reco</code>      | <code>reports/eco_shell</code> , <code>reports/echo_shell_pass*</code>                                           |
| <code>merge_gds</code> | <code>results/</code>                                                                                            |
| <code>rdrc</code>      | <code>reports/&lt;design_name&gt;.drc.summary</code> (bottom of file)                                            |
| <code>rlvs</code>      | <code>reports/&lt;design_name&gt;.lvs.report</code> (second result)                                              |

# Flow Outputs

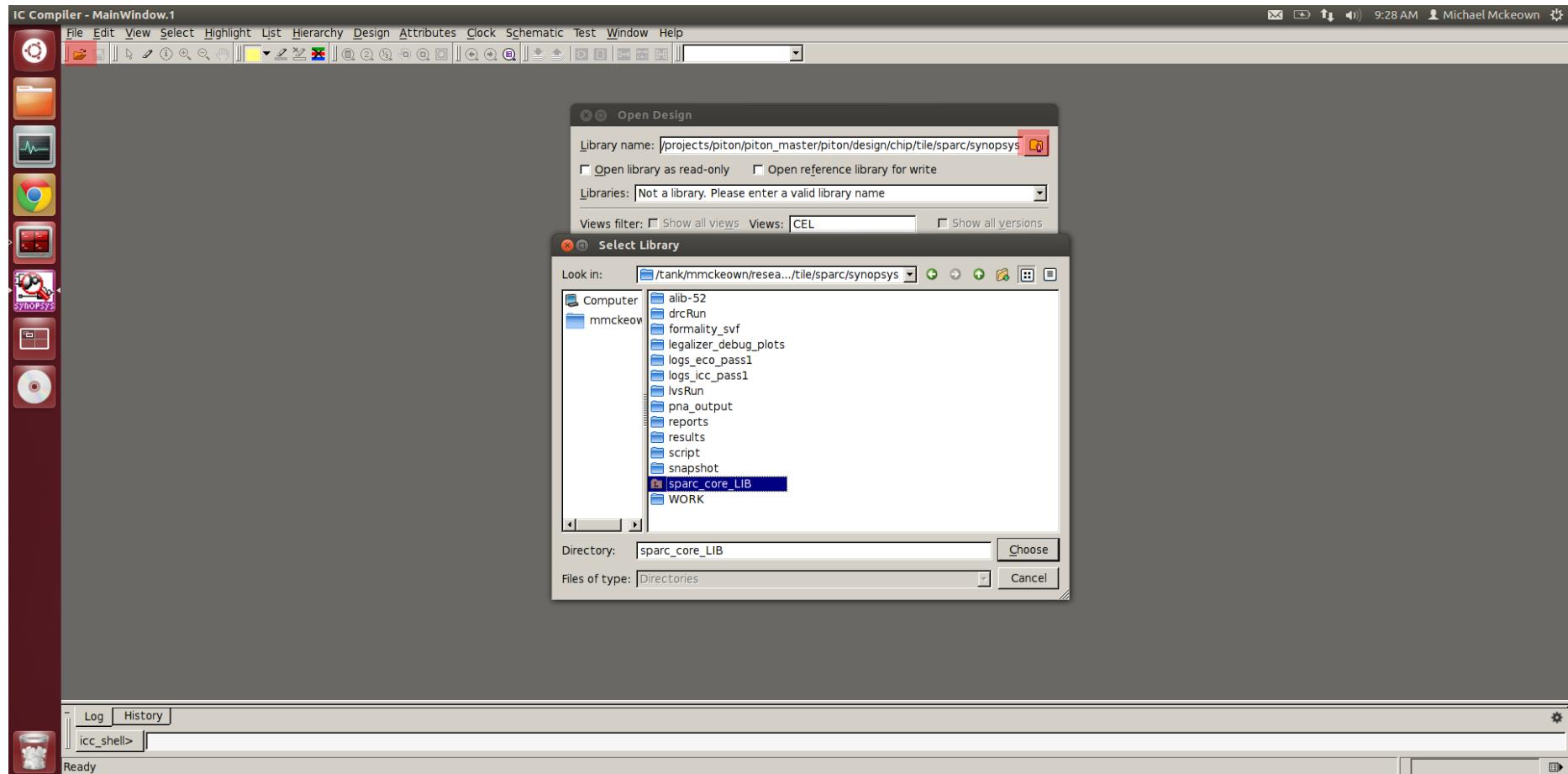
```
[synopsys]mmckeown@hanoi$ pwd
/tank/mmckeown/research/projects/piton/piton_master/piton/design/chip/tile/sparc/synopsys
[synopsys]mmckeown@hanoi$ ls
WORK
alib-52
command.log
dc_shell_pass1.log
drcRun
drc_sparc
ecol_sparc
eco_shell_pass1.log
filenames_16101_D20160412.log
filenames_17779_D20160412.log
filenames_25593_D20160412.log
filenames_27689_D20160413.log
filenames_30063_D20160413.log
filenames_31848_D20160413.log
filenames_38009_D20160413.log
filenames_45017_D20160413.log
filenames_58085_D20160414.log
filenames_58451_D20160414.log
filenames_58476_D20160414.log
filenames_59231_D20160414.log
filenames_63773_D20160414.log
filenames_64960_D20160413.log
fm_shell_command.log
fm_shell_dc_pass1.log
fm_shell_eco_pass1.log
fm_shell_icc_pass1.log
formality.log
formality.svf
icc_shell_pass1.log
legalizer_debug_plots
logs_eco_pass1
logs_icc_pass1
lvsRun
lvs_sparc
merge_gds.log
mgds_sparc
net.act
par1_sparc
parasitics_command.log
pna_output
pt_shell_command.log
pt_shell_dc_pass1.log
pt_shell_eco_pass1.log
pt_shell_icc_pass1.log
reports
results
rvs1_dc_sparc
rvs1_eco_sparc
rvs1_icc_sparc
rvs_status_dc_pass1.log
script
set_pad_attributes_on_cell_sparc_core.tcl
slurm-106342.out
slurm-106343.out
slurm-106344.out
slurm-106345.out
slurm-106346.out
slurm-106347.out
slurm-106348.out
slurm-106349.out
slurm-106350.out
slurm-106351.out
slurm-106352.out
slurm-106353.out
snapshot
sparc_core.def
sparc_core_LIB
sparc_core_LIB.tf_checker
sparc_core_port_map.0
stal_dc_sparc
stal_eco_sparc
stal_icc_sparc
sta_status_dc_pass1.log
syn1_sparc
syn_status_pass1.log
```

# Flow Outputs

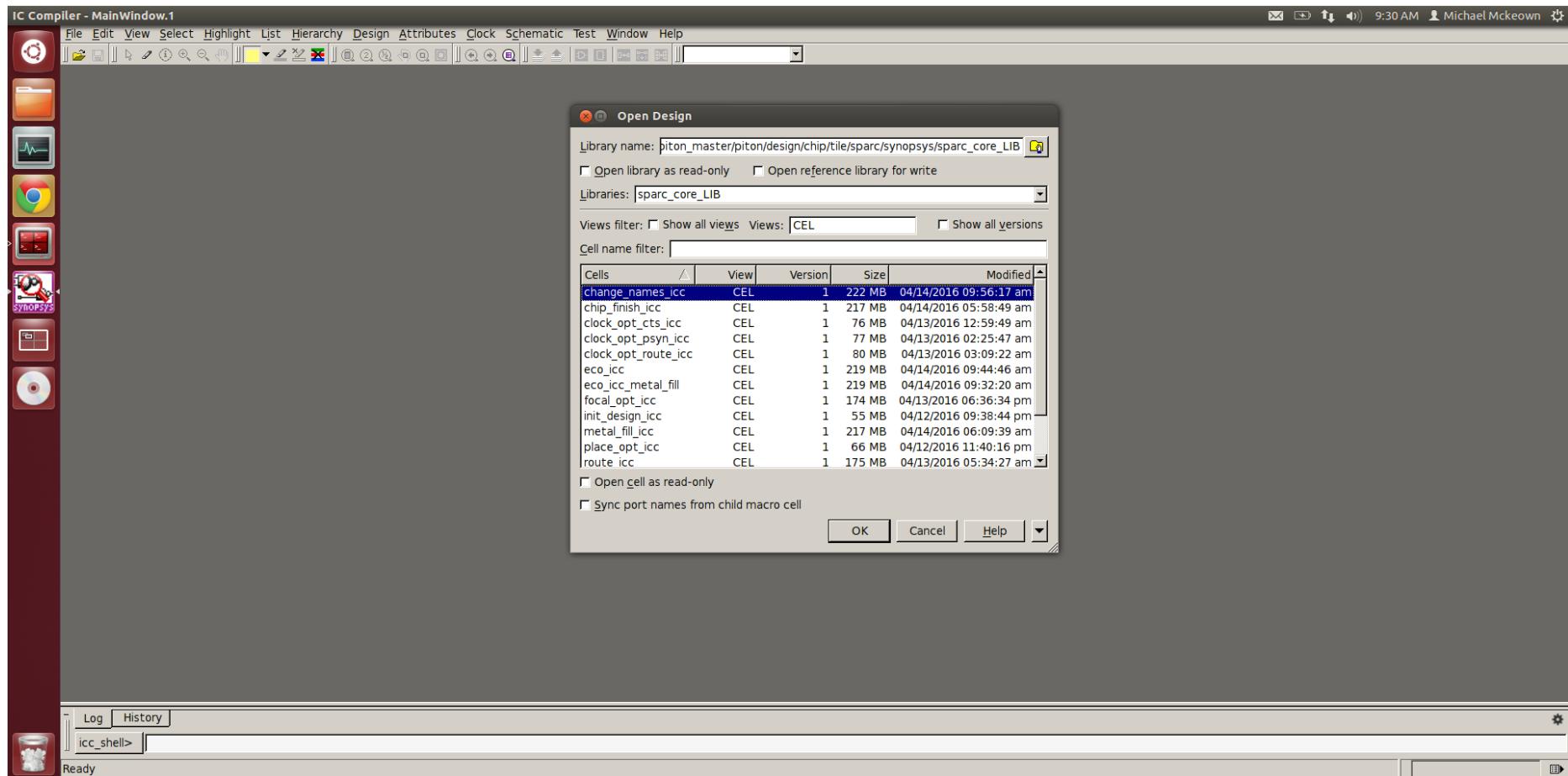
```
[synopsys]mmckeown@hanoi$ pwd
/tank/mmckeown/research/projects/piton/piton_master/piton/design/chip/tile/sparc/synopsys
[synopsys]mmckeown@hanoi$ ls results/
eco_changes_dc.tcl sparc_core.gds
sparc_core sparc_core.icc.lib
sparc_core.dc.lib sparc_core.icc.scope
sparc_core.dc.scope sparc_core.icc_constr.pt
sparc_core.dc_constr.pt sparc_core.icc_lib.db
sparc_core.dc_lib.db sparc_core.icc_test.db
sparc_core.dc_test.db sparc_core.initial.fp
sparc_core.drc.results sparc_core.mapped.ddc
sparc_core.elab.ddc sparc_core.mapped.fp
sparc_core.erc.results sparc_core.mapped.pt.sdf
[synopsys]mmckeown@hanoi$ ls reports/
dc_shell_pass1 fm_shell_eco_pass1 pt_shell_dc_pass1 sparc_core.drc.summary sparc_core.lvs.report
eco_shell_pass1 fm_shell_icc_pass1 pt_shell_eco_pass1 sparc_core.erc.summary
fm_shell_dc_pass1 icc_shell_pass1 pt_shell_icc_pass1 sparc_core.gds2sp.report.ext
```

# Opening the Design

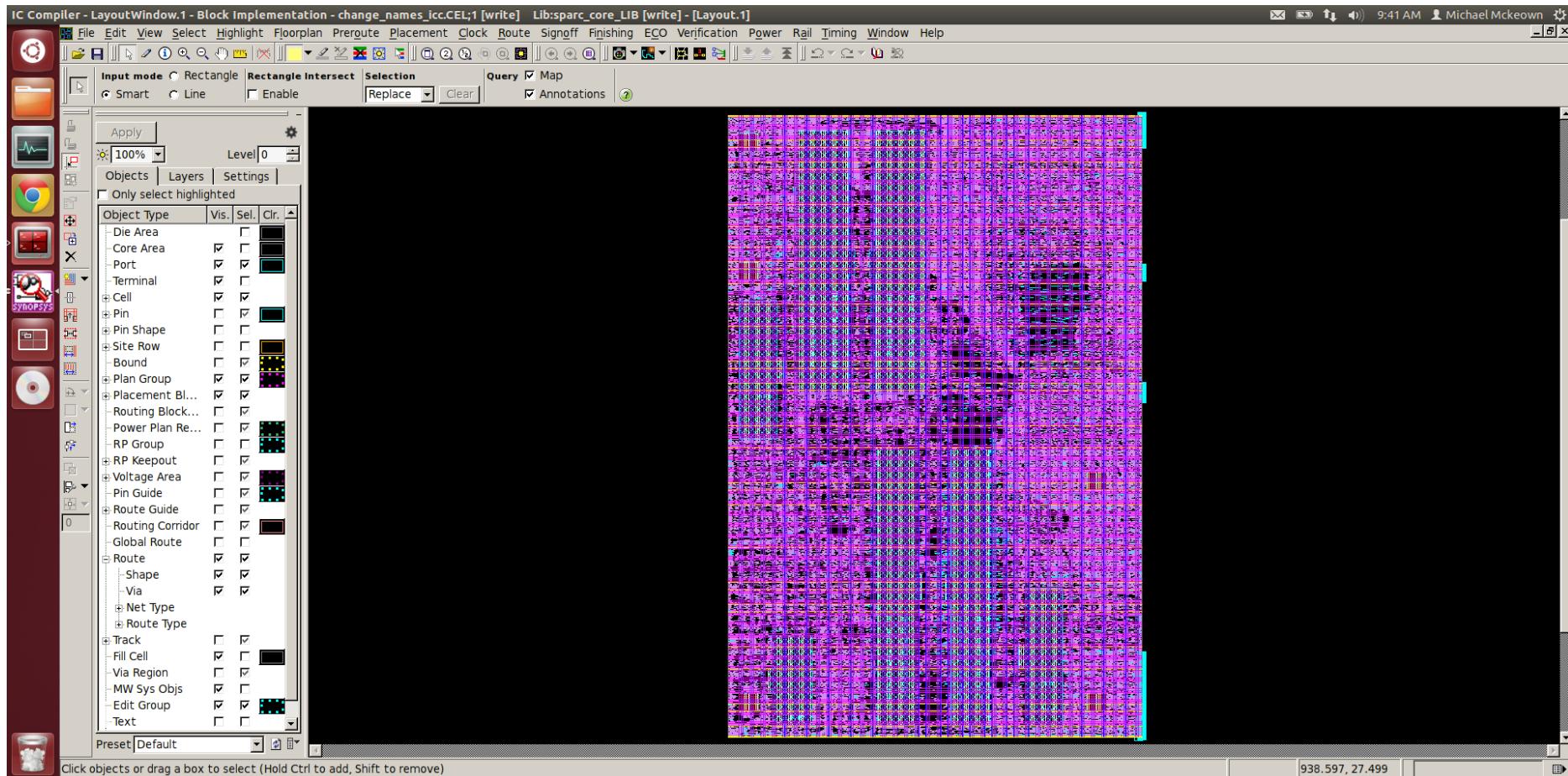
```
[synopsys]mmckeown@hanoi$ icc_shell -gui
```



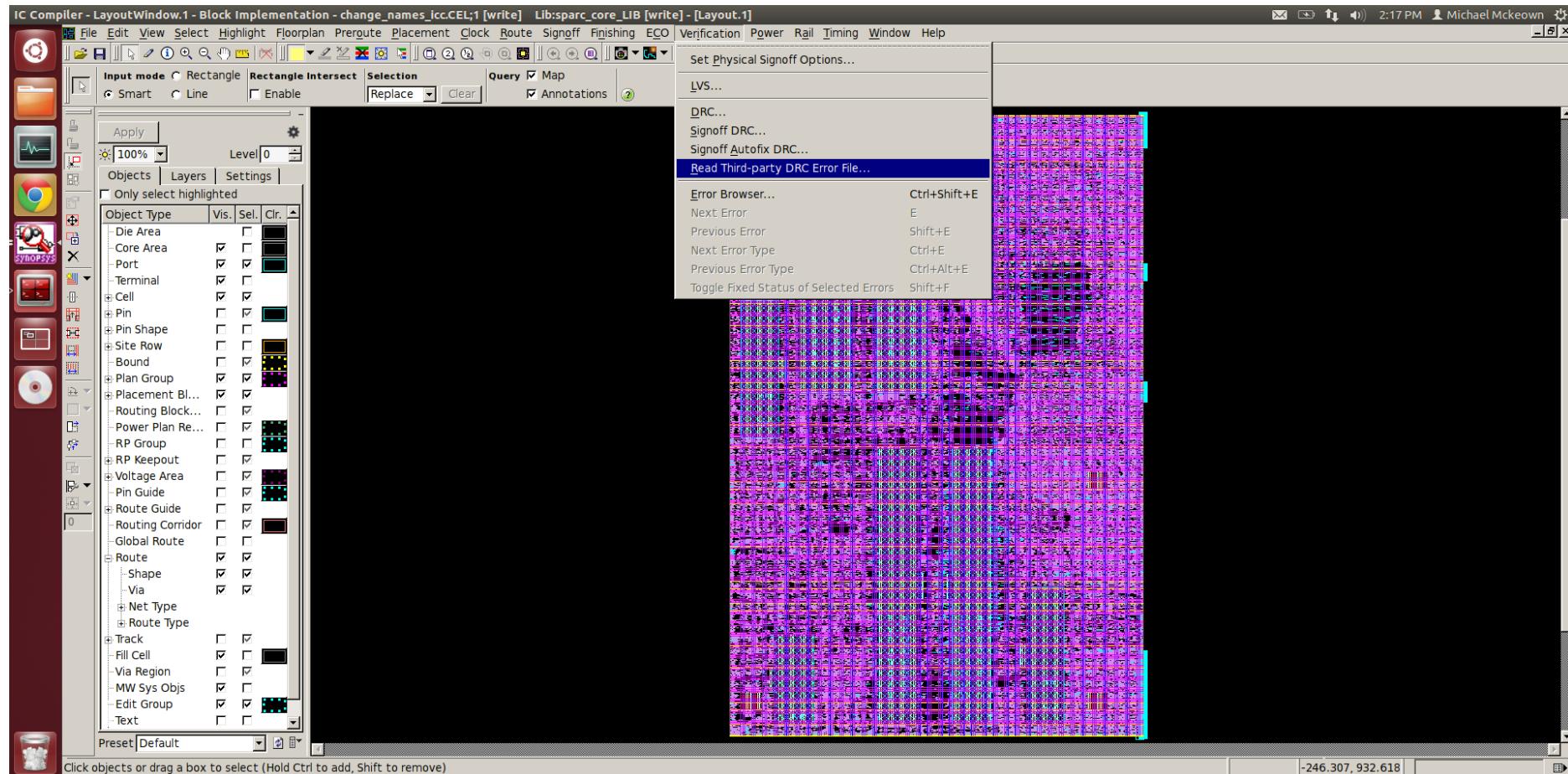
# Opening the Design



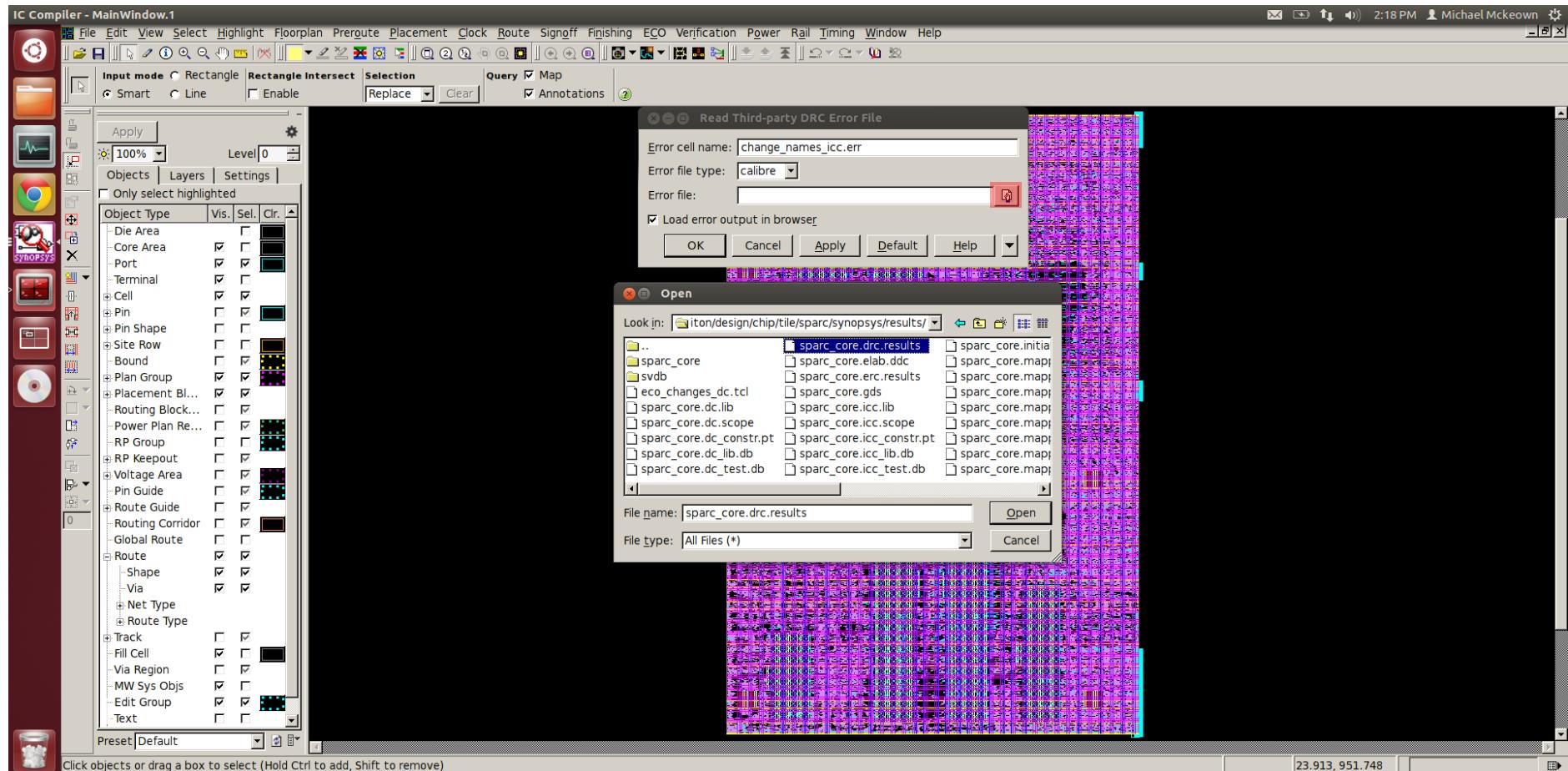
# Opening the Design



# Opening the Design



# Opening the Design



# Opening the Design

