Getting to Work with OpenPiton

Princeton University

http://openpiton.org
ASIC SYNTHESIS AND BACKEND
Whats 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

• Coming soon: Gate-level simulation
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-RMLJ-2013.12-SP2
  - Settings:
    - RTL Source Format: VERILOG

- QoR Strategy: DEFAULT
- Physical Guidance: TRUE
- Hierarchical Flow: TRUE
- MCM Flow: FALSE
- Multi-Voltage UPF: FALSE
- Clock Gating: TRUE
- Leakage Power: TRUE
- DFT Synthesis: FALSE
- Lynx Compatible: FALSE
- Static Timing Analysis
  - PT-RM.j-2013.12
Patching Synopsys-RM

```
[openpiton]mmckeown@hanoi$ ls
README  README.md  build  docs  piton
[openpiton]mmckeown@hanoi$ synrm_patch -h

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
./solvnet_scripts/ICC-RM_I-2013.12-SP4/
```
Patching Synopsys-RM

```
[openpiton]mmckeown@hanoi$ ls
README  README.md  build  docs  piton
[openpiton]mmckeown@hanoi$ synrm_patch -h

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
../solvnet_scripts/ICC-RM_I-2013.12-SP4/
```
Patching Synopsys-RM

```
[openpiton]mmckeown@hanoi$ ls
README  README.md  build  docs  piton
[openpiton]mmckeown@hanoi$ synrm_patch -h

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

```
Patching Synopsys-RM

[openpiton]mmckeown@hanoi$ ls
README  README.md  build  docs  piton

[openpiton]mmckeown@hanoi$ synrm_patch -h

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

Patching Synopsys-RM

synrm_patch: # OpenPiton Patch Synopsys RM Flow #
synrm_patch: Sympnys RM... synrm_patch: Sympnys RM integrity check passed.
synrm_patch: Copying Synopsys RM files from '../solvent_script/DC-RM_I-2013.12-SP2/' to '/tank/mmckeown/research/projects/piton/op
enpiton/piton/tools/synopsys/script/'
synrm_patch: Successfully copied Synopsys RM files from '../solvent_script/DC-RM_I-2013.12-SP2/'.
synrm_patch: Patching Synopsys RM '../solvent_script/DC-RM_I-2013.12-SP2/' to OpenPiton...
synrm_patch: Successfully patched Synopsys RM '../solvent_script/DC-RM_I-2013.12-SP2/' to OpenPiton.
synrm_patch: Copying Synopsys RM files from '../solvent_script/PT-RM_I-2013.12/' to '/tank/mmckeown/research/projects/piton/openpi
ton/piton/tools/synopsys/script/'
synrm_patch: Successfully copied Synopsys RM files from '../solvent_script/PT-RM_I-2013.12/'.
synrm_patch: Patching Synopsys RM '../solvent_script/PT-RM_I-2013.12/' to OpenPiton...
synrm_patch: Successfully patched Synopsys RM '../solvent_script/PT-RM_I-2013.12/' to OpenPiton.
synrm_patch: Copying Synopsys RM files from '../solvent_script/ICC-RM_I-2013.12-SP4/' to '/tank/mmckeown/research/projects/piton/op
enpiton/piton/tools/synopsys/script/'
synrm_patch: Successfully copied Synopsys RM files from '../solvent_script/ICC-RM_I-2013.12-SP4/'.
synrm_patch: Patching Synopsys RM '../solvent_script/ICC-RM_I-2013.12-SP4/' to OpenPiton...
synrm_patch: Successfully patched Synopsys RM '../solvent_script/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]$
Patching Synopsys-RM

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/synopsys/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/synopsys/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/synopsys/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$
Patching Synopsys-RM

```
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/synopsys/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/synopsys/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/synopsys/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$```
Patching Synopsys-RM

```
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/mmcckeown/research/projects/piton/openpiton/piton/tools/synopsys/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/mmcckeown/research/projects/piton/openpiton/piton/tools/synopsys/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/mmcckeown/research/projects/piton/openpiton/piton/tools/synopsys/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]mmcckeown@hanoi$ 
```
Directory Structure and Scripts

• All scripts written in Tcl
• Two primary locations
  – Module generic scripts
  – Module specific scripts

```
piton/
  ├── design/
  │    └── chip/
  └── tools/
      └── calibre/
          └── synopsys/
```
Directory Structure and Scripts

• All scripts written in Tcl
• Two primary locations
  – Module generic scripts
  – Module specific scripts

```
$ pwd
/tank/mmckeown/research/projects/piton/openpiton/piton/design/chip/tile/sparc
$ ls
exu  ffu  ifu  lsu  mul  rtl  spu  srams  synopsys  tlu
$ ls rtl/
Flist.sparc_common     bw_clk_cl_sparc_cmp.v     cpx_spc_buf.v     sparc.v     spc_pcx_buf.v
Flist.sparc_top         cfgasi.v               cpx_spc_rpt.v     sparc_core.v
$ 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
```
Directory Structure and Scripts

- All scripts written in Tcl
- Two primary locations
  - Module generic scripts
  - Module specific scripts
Directory Structure and Scripts

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

```
$ pwd
/tank/mmckeown/research/projects/piton/openpiton/piton/design/chip/tile/sparc
$ ls
exu ffu ifu lsu mul rtl spu srams synopsis tlu
$ 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
$ ls synopsis/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
```
Directory Structure and Scripts

• All scripts written in Tcl
• Two primary locations
  – Module generic scripts
  – Module specific scripts

```bash
[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  tlun
[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  config.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
```
Directory Structure and Scripts

- All scripts written in Tcl
- Two primary locations
  - Module generic scripts
  - Module specific scripts
Porting to a Process Technology

- `${PITON_ROOT}/piton/tools/synopsys/script/common/env_setup.tcl`
- `${PITON_ROOT}/piton/tools/synopsys/script/common/process_setup.tcl`
- `${PITON_ROOT}/piton/tools/calibre/script/common/calibre_env`
Porting to a Process Technology

- `${PITON_ROOT}/piton/tools/synopsys/script/common/env_setup.tcl`
- `${PITON_ROOT}/piton/tools/synopsys/script/common/process_setup.tcl`
- `${PITON_ROOT}/piton/tools/calibre/script/common/calibre_env`

```tcl
# Setup environment variables to be used throughout scripts
set HOSTNAME ::env(HOSTNAME)
# Add environment variables to point to your
# std cell libraries and PDK. Use these in
# scripts to reference files as opposed to
# absolute paths to maintain system portability
#set IBM_PDK ::env(IBM_PDK)
#set IBM_PDK_SYNOPSYS ::env(IBM_PDK_SYNOPSYS)
#set IBM_SRAMS ::env(IBM_SRAMS)
#set IBM_IP ::env(IBM_IP)
#set ARM_32NM_IP ::env(ARM_32NM_IP)
```
Porting to a Process Technology

- `${PITON_ROOT}/piton/tools/synopsys/script/common/env_setup.tcl`
-`${PITON_ROOT}/piton/tools/synopsys/script/common/process_setup.tcl`
-`${PITON_ROOT}/piton/tools/calibre/script/common/calibre_env`
Porting to a Process Technology

- `$PITON_ROOT/piton/tools/synopsys/script/common/env_setup.tcl`
- `$PITON_ROOT/piton/tools/synopsys/script/common/process_setup.tcl`
- `$PITON_ROOT/piton/tools/calibre/script/common/calibre_env.tcl`

```tcl
# Do not include module specific SRAMs in the below lists, these variables
# should only include standard cell libraries. Module specific SRAMs are
# handled in module specific TCL scripts

# Target standard cell technology libraries
set TARGET_LIBRARY_FILES  "TARGET_STDCELL_LIBRARIES_HERE (.db, put multiple Vt as space separated list)"

# Target standard cell technology verilog
set TARGET_LIBRARY_VERILOG  "TARGET_STDCELL_LIBRARY_TEST_SIMULATION_VERILOG_HERE (.v)"

# List of max min library pairs for setup and hold analysis
# List should follow this format: "max1 min1 max2 min2 max3 min3..."
# Should follow same order as target libraries, but two here for each one in target library
set MIN_LIBRARY_FILES  "MAX_SETUP_STDCELL_LIBRARY_HERE (.db) MIN_HOLD_STDCELL_LIBRARY_HERE (.db)"

# Target standard cell Milkyway reference libraries
set MW_REFERENCE_LIB_DIRS "TARGET_STDCELL_MW_REFERENCE_LIBRARIES_HERE (MW directory)"

# Target standard cell GDS merge files if necessary
# (depends on if CEL views were extracted for above MW libs.
# This may depend on how they were provided to you, but to be
# safe you can place the GDS for stdcells here, it may just increase
# runtime in merge gds
set TARGET_LIBRARY_GDSII "" ; # Optional

# Target standard cell libarry SP merge files for LVS
set TARGET_LIBRARY_SP "TARGET_STDCELL_LIBRARY_LVS_SPICE_HERE (.cdl/.sp)"

# MW reference control file (not really used, do not need to set)
set MW_REFERENCE_CONTROL_FILE ""

# Tcl script with library modifications to set don't use cells (not necessary)
set LIBRARY_DONT_USE_FILE ""

# Milkyway technology file for process
set TECH_FILE  "TARGET_STDCELL_TECH_FILE_HERE (.tf)"
# Mapping file for TLUplus
set MAP_FILE "PDK_TLUPLUS_MAP_FILE_HERE (.map)"
# TLUplus max and min files
set TLUPLUS_MAX_FILE  "PDK_TLUPLUS_MAX_FILE_HERE (.tluplus)"
```

13
Porting to a Process Technology

- **${PITON_ROOT}/piton/tools/synopsys/script/common/env_setup.tcl**

- **${PITON_ROOT}/piton/tools/synopsys/script/common/process_setup.tcl**

- **${PITON_ROOT}/piton/tools/calibre/script/common/calibre_env.tcl**

```tcl
# Do not include module specific SRAMs in the below lists, these variables
# should only include standard cell libraries. Module specific SRAMs are
# handled in module specific TCL scripts

# Target standard cell technology libraries
set TARGET_LIBRARY_FILES "TARGET_STDCELL_LIBRARIES_HERE (.db, put multiple Vt as space separated list)"

# Target standard cell technology verilog
set TARGET_LIBRARY_VERILOG "TARGET_STDCELL_LIBRARY_TEST_SIMULATION_VERILOG_HERE (.v)"

# List of max min library pairs for setup and hold analysis
# List should follow this format: "max1 min1 max2 min2 max3 min3..."
# Should follow same order as target libraries, but two here for each one in target library
set MIN_LIBRARY_FILES "MAX_SETUP_STDCELL_LIBRARY_HERE (.db) MIN_HOLD_STDCELL_LIBRARY_HERE (.db)"

# Target standard cell Milkyway reference libraries
set MW_REFERENCE_LIB_DIRS "TARGET_STDCELL_MW_REFERENCE_LIBRARIES_HERE (MW directory)"

# Target standard cell GDS merge files if necessary
# (depends on if CEL views were extracted for above MW libs.
# This may depend on how they were provided to you, but to be
# safe you can place the GDS for stdcells here, it may just increase
# runtime in merge gds
set TARGET_LIBRARY_GDSII "" ; # Optional

# Target standard cell libarry SP merge files for LVS
set TARGET_LIBRARY_SP "TARGET_STDCELL_LIBRARY_LVS_SPICE_HERE (.cdl/.sp)"

# MW reference control file (not really used, do not need to set)
set MW_REFERENCE_CONTROL_FILE ""

# Tcl script with library modifications to set don't use cells (not necessary)
set LIBRARY_DONT_USE_FILE ""

# Milkyway technology file for process
set TECH_FILE "TARGET_STDCELL_TECH_FILE_HERE (.tf)"

# Mapping file for TLUplus
set MAP_FILE "PDK_TLUPLUS_MAP_FILE_HERE (.map)"

# TLUplus max and min files
set TLUPLUS_MAX_FILE "PDK_TLUPLUS_MAX_FILE_HERE (.tluperplus)"
```
Porting to a Process Technology

- \(\text{${PITON_ROOT}/piton/tools/synopsys/script/common/env_setup.tcl}\)
- \(\text{${PITON_ROOT}/piton/tools/synopsys/script/common/process_setup.tcl}\)
- \(\text{${PITON_ROOT}/piton/tools/calibre/script/common/calibre_env}\)
Porting to a Process Technology

- `${PITON_ROOT}/piton/tools/synopsys/script/common/env_setup.tcl`
  - DRC Runset and optional golden GDS file
    - OPENPITON_CALIBRE_DRC_DECK="CALIBRE_DRC_DECK_HERE (.drc.cal)"
    - OPENPITON_CALIBRE_GOLDEN_GDS_FILE="" ; # Optional, (.gds)
  - 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.
    - These should be space separated list of standard cells
    - You want black boxed
      - OPENPITON_GDS2SP_BBOX_CELLS=""
      - OPENPITON_LVS_BBOX_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

- `${PITON_ROOT}/piton/tools/synopsys/script/common/env_setup.tcl`
- `${PITON_ROOT}/piton/tools/synopsys/script/common/process_setup.tcl`
- `${PITON_ROOT}/piton/tools/calibre/script/common/calibre_env.tcl`

# DRC Runset and optional golden GDS file
OPENPITON_CALIBRE_DRC_DECK="CALIBRE_DRC_DECK_HERE (.drc.cal)"
OPENPITON_CALIBRE_GOLDEN_GDS_FILE="" ; # Optional, (.gds)

# 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.
# These should be space separated list of standard cells
# you want black boxed
OPENPITON_GDS2SP_BBOX CELLS=""
OPENPITON_LVS_BBOX 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/`

<table>
<thead>
<tr>
<th>Process Specific File Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>common/env_setup.tcl</td>
<td>Environment variable setup. Gets environment variables that are used throughout the scripts.</td>
</tr>
<tr>
<td>common/process_setup.tcl</td>
<td>Main process specific setup file. Includes standard cell libraries, technology files, layer mapping files, Milkyway libraries, etc.</td>
</tr>
<tr>
<td>common/calibre_env</td>
<td>Calibre process specific setup file. Sets the DRC and IVS decks and any environment variables used by these decks.</td>
</tr>
<tr>
<td>common/design_setup.tcl</td>
<td>Setup variables specific to a design. Includes things like clock frequency, allowed routing metal layers, etc.</td>
</tr>
<tr>
<td>common/floorplan/common_pgn.tcl</td>
<td>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.</td>
</tr>
<tr>
<td>common/floorplan/core_pgn_mesh.tpl</td>
<td>Templates for PGN meshes. Called from common/floorplan/common_pgn.tcl and module specific PGN scripts.</td>
</tr>
<tr>
<td>common/floorplan/common_post_floorplan.tcl</td>
<td>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.</td>
</tr>
<tr>
<td>common/dbl_via_setup.tcl</td>
<td>Double via definition script. This is usually process specific but is also sometimes optional. Can define which vias are used for double via insertion.</td>
</tr>
<tr>
<td>common/pt_eco_drc_buf.tcl</td>
<td>This is also optional, but specifies a list of standard cell buffers that Synopsys PT can use for DRC fixing.</td>
</tr>
<tr>
<td>common/vt_group_setup.tcl</td>
<td>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.</td>
</tr>
</tbody>
</table>
Porting to a Process Technology

Table 3: Process specific synthesis and backend scripts. Referenced from `~/.piton/tools/synopsys/script/`

<table>
<thead>
<tr>
<th>Process Specific File Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>common/env_setup.tcl</td>
<td>Environment variable setup. Gets environment variables that are used throughout the scripts.</td>
</tr>
<tr>
<td>common/process_setup.tcl</td>
<td>Main process specific setup file. Includes standard cell libraries, technology files, layer mapping files, Milkyway libraries, etc.</td>
</tr>
<tr>
<td>common/calibre_env</td>
<td>Calibre process specific setup file. Sets the DRC and LVS decks and any environment variables used by these decks.</td>
</tr>
<tr>
<td>common/design_setup.tcl</td>
<td>Setup variables specific to a design. Includes things like clock frequency, allowed routing metal layers, etc.</td>
</tr>
<tr>
<td>common/floorplan/common_pgn.tcl</td>
<td>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.</td>
</tr>
<tr>
<td>common/floorplan/core_pgn_mesh.tpl</td>
<td>Templates for PGN meshes. Called from common/floorplan/common_pgn.tcl and module specific PGN scripts.</td>
</tr>
<tr>
<td>common/floorplan/common_post_floorplan.tcl</td>
<td>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.</td>
</tr>
<tr>
<td>common/dbl_via_setup.tcl</td>
<td>Double via definition script. This is usually process specific but is also sometimes optional. Can define which vias are used for double via insertion.</td>
</tr>
<tr>
<td>common/pt_eco_drc_buf.tcl</td>
<td>This is also optional, but specifies a list of standard cell buffers that Synopsys PT can use for DRC fixing.</td>
</tr>
<tr>
<td>common/vt_group_setup.tcl</td>
<td>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.</td>
</tr>
</tbody>
</table>
# Porting to a Process Technology

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

<table>
<thead>
<tr>
<th>Process Specific File Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>common/env_setup.tcl</td>
<td>Environment variable setup. Gets environment variables that are used throughout the scripts.</td>
</tr>
<tr>
<td>common/process_setup.tcl</td>
<td>Main process specific setup file. Includes standard cell libraries, technology files, layer mapping files, Milkyway libraries, etc.</td>
</tr>
<tr>
<td>common/calibre_env</td>
<td>Calibre process specific setup file. Sets the DRC and LVS decks and any environment variables used by these decks.</td>
</tr>
<tr>
<td>common/design_setup.tcl</td>
<td>Setup variables specific to a design. Includes things like clock frequency, allowed routing metal layers, etc.</td>
</tr>
<tr>
<td>common/floorplan/common_pgn.tcl</td>
<td>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.</td>
</tr>
<tr>
<td>common/floorplan/core_pgn_mesh.tpl</td>
<td>Templates for PGN meshes. Called from common/floorplan/common_pgn.tcl and module specific PGN scripts.</td>
</tr>
<tr>
<td>common/floorplan/common_post_floorplan.tcl</td>
<td>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.</td>
</tr>
<tr>
<td>common/dbl_via_setup.tcl</td>
<td>Double via definition script. This is usually process specific but is also sometimes optional. Can define which vias are used for double via insertion.</td>
</tr>
<tr>
<td>common/pt_eco_drc_buf.tcl</td>
<td>This is also optional, but specifies a list of standard cell buffers that Synopsys PT can use for DRC fixing.</td>
</tr>
<tr>
<td>common/vt_group_setup.tcl</td>
<td>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.</td>
</tr>
</tbody>
</table>
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

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

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

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffu</td>
<td>OpenSPARC T1 core floating-point front-end unit</td>
<td>Small module with one SRAM macro</td>
</tr>
<tr>
<td>sparc</td>
<td>OpenSPARC T1 core</td>
<td>Large module with many SRAM macros</td>
</tr>
<tr>
<td>dynamic_node</td>
<td>OpenPiton on-chip network router</td>
<td>Small module with no IP macros</td>
</tr>
<tr>
<td>tile</td>
<td>OpenPiton tile</td>
<td>Large, hierarchical module with many SRAM macros</td>
</tr>
</tbody>
</table>

- `${PITON_ROOT}/piton/tools/synopsys/block.list`
Running the Flow

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

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

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

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffu</td>
<td>OpenSPARC T1 core floating-point front-end unit</td>
<td>Small module with one SRAM macro</td>
</tr>
<tr>
<td>sparc</td>
<td>OpenSPARC T1 core</td>
<td>Large module with many SRAM macros</td>
</tr>
<tr>
<td>dynamic_node</td>
<td>OpenPiton on-chip network router</td>
<td>Small module with no IP macros</td>
</tr>
<tr>
<td>tile</td>
<td>OpenPiton tile</td>
<td>Large, hierarchical module with many SRAM macros</td>
</tr>
</tbody>
</table>
Launch Flow

bash
$ rttf sparc --slurm
rttf: Running full tool flow for 1 modules
rttf: Running full tool flow for sparc
Submitted batch job 147182
Submitted batch job 147183
Submitted batch job 147184
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

bash
$ squeue

<table>
<thead>
<tr>
<th>JOBD</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>CPUS</th>
<th>QOS</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>147188</td>
<td>eco1_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147192</td>
<td>drc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147193</td>
<td>lvs_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147183</td>
<td>stal_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147184</td>
<td>rvs1_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147186</td>
<td>stal_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147187</td>
<td>rvs1_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147189</td>
<td>stal_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147190</td>
<td>rvs1_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147191</td>
<td>mgds_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147185</td>
<td>par1_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>long (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147182</td>
<td>syn1_sparc</td>
<td>mmckeown</td>
<td>R</td>
<td>0:04</td>
<td>1</td>
<td>6</td>
<td>short hanoi</td>
<td></td>
</tr>
</tbody>
</table>
Launch Flow

```
[openpiton]mmckeown@hanoi$ rftf sparc --slurm
rftf: Running full tool flow for 1 modules
rftf: Running full tool flow for sparc
Submitted batch job 147182
Submitted batch job 147183
Submitted batch job 147184
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

<table>
<thead>
<tr>
<th>JOBD</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>CPUS</th>
<th>QOS</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>147188</td>
<td>eco1_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147192</td>
<td>drc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147193</td>
<td>lvs_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147183</td>
<td>stal_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147184</td>
<td>rvs1_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147186</td>
<td>stal_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147187</td>
<td>rvs1_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147189</td>
<td>stal_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147190</td>
<td>rvs1_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147191</td>
<td>mgds_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147185</td>
<td>parl_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>long  (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147182</td>
<td>syn1_sparc</td>
<td>mmckeown</td>
<td>R</td>
<td>0:04</td>
<td>1</td>
<td>6</td>
<td>short hanoi</td>
<td></td>
</tr>
</tbody>
</table>
```
Launch Flow

```bash
[openpiton]mmckeown@hanoi$ rftf sparc --slurm
rftf: Running full tool flow for 1 modules
rftf: Running full tool flow for sparc
Submitted batch job 147182
Submitted batch job 147183
Submitted batch job 147184
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

<table>
<thead>
<tr>
<th>JOBD</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>CPUS</th>
<th>QOS</th>
<th>NODELIST(_REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>147188</td>
<td>eco1_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147192</td>
<td>drc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147193</td>
<td>lvs_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147183</td>
<td>stal_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147184</td>
<td>rvs1_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147186</td>
<td>stal_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147187</td>
<td>rvs1_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147189</td>
<td>stal_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147190</td>
<td>rvs1_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147191</td>
<td>mgds_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147185</td>
<td>par1_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>long (Dependency)</td>
<td></td>
</tr>
<tr>
<td>147182</td>
<td>syn1_sparc</td>
<td>mmckeown</td>
<td>R</td>
<td>0:04</td>
<td>1</td>
<td>6</td>
<td>short hanoi</td>
<td></td>
</tr>
</tbody>
</table>
```
Launch Flow

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

<table>
<thead>
<tr>
<th># BlockID</th>
<th>BlockPath</th>
<th>BlockSynMemReq</th>
<th>BlockPARMReq</th>
<th>FTFPasses</th>
<th>ECO</th>
</tr>
</thead>
<tbody>
<tr>
<td>tile</td>
<td>chip/tile</td>
<td>32000</td>
<td>128000</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>dynamic_node</td>
<td>chip/tile/dynamic_node</td>
<td>8000</td>
<td>16000</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>sparc</td>
<td>chip/tile/sparc</td>
<td>48000</td>
<td>128000</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>ffu</td>
<td>chip/tile/sparc/ffu</td>
<td>8000</td>
<td>12000</td>
<td>1</td>
<td>Y</td>
</tr>
</tbody>
</table>

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 eco1_sparc mmckeown PD  0:00 1  10 short (Dependency)
147192 drc_sparc mmckeown PD  0:00 1  10 short (Dependency)
147193 lvs_sparc mmckeown PD  0:00 1  10 short (Dependency)
147183 stal_dc_sparc mmckeown PD  0:00 1  1 short (Dependency)
147184 rvs1_dc_sparc mmckeown PD  0:00 1  1 short (Dependency)
147186 stal_icc_sparc mmckeown PD  0:00 1  1 short (Dependency)
147187 rvs_1_icc_sparc mmckeown PD  0:00 1  1 short (Dependency)
147189 stal_eco_sparc mmckeown PD  0:00 1  1 short (Dependency)
147190 rvs_1_eco_sparc mmckeown PD  0:00 1  1 short (Dependency)
147191 mgds_sparc mmckeown PD  0:00 1  1 short (Dependency)
147185 parl_sparc mmckeown PD  0:00 1  10 long (Dependency)
147182 syn1_sparc mmckeown R  0:04 1  6 short hanoi
```
Launch Flow

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

rtf: Running full tool flow for 1 modules
rtf: Running full tool flow for sparc
Submitted batch job 147182
Submitted batch job 147183
Submitted batch job 147184
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

<table>
<thead>
<tr>
<th>JOID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>CPUS</th>
<th>QOS</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>147188</td>
<td>eco1_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147192</td>
<td>drc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147193</td>
<td>lvs_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147183</td>
<td>stal_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147184</td>
<td>rvs1_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147186</td>
<td>stal_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147187</td>
<td>rvs1_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147189</td>
<td>stal_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147190</td>
<td>rvs1_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147191</td>
<td>mgds_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147185</td>
<td>parl_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>long</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147182</td>
<td>syn1_sparc</td>
<td>mmckeown</td>
<td>R</td>
<td>0:04</td>
<td>1</td>
<td>6</td>
<td>short</td>
<td>hanoi</td>
</tr>
</tbody>
</table>
```
Launch Flow

```
[openpiton]mmckeown@hanoi$ rftf sparc --slurm
rftf: Running full tool flow for 1 modules
rftf: Running full tool flow for sparc
Submitted batch job 147182
Submitted batch job 147183
Submitted batch job 147184
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

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>CPUS</th>
<th>QOS</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>147188</td>
<td>eco1_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147192</td>
<td>drc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147193</td>
<td>lvs_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147183</td>
<td>stal_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147184</td>
<td>rvs1_dc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147186</td>
<td>stal_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147187</td>
<td>rvs1_icc_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147189</td>
<td>stal_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147190</td>
<td>rvs1_eco_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147191</td>
<td>mgds_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>1</td>
<td>short</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147185</td>
<td>parl_sparc</td>
<td>mmckeown</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>10</td>
<td>long</td>
<td>(Dependency)</td>
</tr>
<tr>
<td>147182</td>
<td>syn1_sparc</td>
<td>mmckeown</td>
<td>R</td>
<td>0:04</td>
<td>1</td>
<td>6</td>
<td>short</td>
<td>hanoi</td>
</tr>
</tbody>
</table>
```
Flow Runtimes

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

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.

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

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.

<table>
<thead>
<tr>
<th>Command</th>
<th>Results Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsyn</td>
<td>reports/dc_shell, reports/dc_shell_pass*</td>
</tr>
<tr>
<td>rsta</td>
<td>reports/pt_shell, reports/pt_shell_dc_pass*, reports/pt_shell_icc_pass*</td>
</tr>
<tr>
<td>rrvs</td>
<td>reports/fm_shell, reports/fm_shell_dc_pass*, reports/fm_shell_icc_pass*</td>
</tr>
<tr>
<td>rpar</td>
<td>reports/icc_shell, reports/icc_shell_pass*</td>
</tr>
<tr>
<td>reco</td>
<td>reports/eco_shell, reports/echo_shell_pass*</td>
</tr>
<tr>
<td>merge_gds</td>
<td>results/</td>
</tr>
<tr>
<td>rsrc</td>
<td>reports/&lt;design_name&gt;.drc.summary (bottom of file)</td>
</tr>
<tr>
<td>rlvs</td>
<td>reports/&lt;design_name&gt;.lvs.report (second result)</td>
</tr>
</tbody>
</table>
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
eco1_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
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.acts
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
rsv1_dc_sparc
rsv1_eco_sparc
rsv1_icc_sparc
sta_status_dc_pass1.log
sta1_dc_sparc
sta1_eco_sparc
sta1_icc_sparc
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
sta1 dc_sparc
sta1 eco_sparc
sta1 icc_sparc
sta_status_dc_pass1.log
syn1_sparc
syn1_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
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
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.acts
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
rvs_1_eco_sparc
rvs_1_icc_sparc
rvs_status_dc_pass1.log
rvs_status_eco_pass1.log
rvs_status_icc_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
sta1_dc_sparc
sta1_eco_sparc
sta1_icc_sparc
sta_status_dc_pass1.log
sta_status_eco_pass1.log
sta_status_icc_pass1.log
syn1_sparc
syn_status_dc_pass1.log
syn_status_eco_pass1.log
syn_status_icc_pass1.log
Flow Outputs
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.acts
par1_sparc
parasitics_command.log
pna_output
pt_shell_command.log
pt_shell_dc_pass1.log
pt_shell_icc_pass1.log
reports
results
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
sparc_core.def
sparc_core_LIB
sparc_core_LIB.tf_checker
sparc_core_port_map.0
sta1_dc_sparc
sta1_eco_sparc
sta1_icc_sparc
sta_status_dc_pass1.log
syn1_sparc
syn_status_pass1.log
set_pad_attributes_on_cell_sparc_core.tcl
```
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.mapped.sdc  sparc_core.output.sbpf.max
sparc_core  sparc_core.icc.lib  sparc_core.mapped.sdf  sparc_core.output.sbpf.min
sparc_core.dc.lib  sparc_core.icc.scope  sparc_core.mapped.spdf  sparc_core.output.sdc
sparc_core.dc.scope  sparc_core.icc Constr.pt  sparc_core.mapped.svf  sparc_core.output.sdf
sparc_core.dc Constr.pt  sparc_core.icc.lib.db  sparc_core.mapped.v  sparc_core.output.v
sparc_core.dc_lib.db  sparc_core.icc_test.db  sparc_core.output.dc.v svdb
sparc_core.dc_test.db  sparc_core.initial.fp  sparc_core.output.def
sparc_core.drc.results  sparc_core.mapped.dcc  sparc_core.output.pg.lvs.v
sparc_core.elab.dcc  sparc_core.mapped.fp  sparc_core.output.pt.sdf
sparc_core.erc.results  sparc_core.mapped.pt.sdf  sparc_core.output.pt.v

[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
fм_shell_dc_pass1  icc_shell_pass1  pt_shell_icc_pass1  sparc_core.gds2sp.report.ext
```
Flow Outputs

```
[synopsy]$ mmckeown@hanoi$ pwd
/tank/mmckeown/research/projects/piton/piton_master
[piton/design/chip/tile/sparc/synopsys]
[synopsy]$ mmckeown@hanoi$ ls results/
eco_changes_dc.tcl sparc_core.gds sparc_core.mapped.sdc sparc_core.output.sbpf.max
sparc_core sparc_core.icc.lib sparc_core.mapped.sdf sparc_core.output.sbpf.min
sparc_core.dc.lib sparc_core.icc.scope sparc_core.mapped.spef sparc_core.output.sdc
sparc_core.dc.scope sparc_core.icc_constr.pt sparc_core.mapped.svf sparc_core.output.sdf
sparc_core.dc_constr.pt sparc_core.icc_lib.db sparc_core.mapped.v sparc_core.output.v
sparc_core.dc_lib.db sparc_core.icc_test.db sparc_core.unmerged.gds svdb
sparc_core.dc_test.db sparc_core.initial.fp sparc_core.output dc.v
sparc_core.drc.results sparc_core.output dc.dv
sparc_core.elab.ddc sparc_core.mapped.ddc sparc_core.output pg.lvs.v
sparc_core.erc.results sparc_core.mapped.ddc sparc_core.output pt.sdf
sparc_core.erc_results sparc_core.mapped.pt.sdf sparc_core.output pt.v
[synopsy]$ 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
```
Flow Outputs

```bash
[ synopsis ] mmckeown@hanoi$ pwd
/tank/mmckeown/research/projects/piton/piton_master/piton/design/chip/tile/sparc/synopsys
[ synopsis ] mmckeown@hanoi$ ls results/
eco_changes_dc.tcl sparc_core.gds sparc_core.mapped.sdc sparc_core.output.sbpf.max
sparc_core sparc_core.icc.lib sparc_core.mapped.sdf sparc_core.output.sbpf.min
sparc_core.dc.lib sparc_core.icc.scope sparc_core.mapped.spdef sparc_core.output.sdc
sparc_core.dc.scope sparc_core.icc_constr.pt sparc_core.mapped.svf sparc_core.output.sdf
sparc_core.dc_constr.pt sparc_core.icc_lib.db sparc_core.mapped.v sparc_core.output.v
sparc_core.dc_lib.db sparc_core.icc_test.db sparc_core.output.dc.v
sparc_core.dc_test.db sparc_core.initial.fp sparc_core.output.def
 sparc_core.drc.results sparc_core.mapped.ddc sparc_core.output.pg.lvs.v
sparc_core.elab.ddc sparc_core.mapped.fp sparc_core.output.pt.sdf
sparc_core.erc.results sparc_core.mapped.pt.sdf sparc_core.output.pt.v
[ synopsis ] 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
```
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.mapped.sdc  sparc_core.output.sbpf.max
sparc_core         sparc_core.icc.lib  sparc_core.mapped.sdf  sparc_core.output.sbpf.min
sparc_core.dc.lib  sparc_core.icc.scope sparc_core.mapped.spef  sparc_core.output.sdc
sparc_core.dc.scope sparc_core.icc_constr.pt sparc_core.mapped.svf  sparc_core.output.sdf
sparc_core.dc_constr.pt sparc_core.icc.lib.db sparc_core.mapped.v  sparc_core.output.v
sparc_core.dc_lib.db sparc_core.icc_test.db sparc_core.output.dc.v
sparc_core.dc_test.db sparc_core.initial.fp  sparc_core.output.def
sparc_core.drc.results sparc_core.mapped.ddc sparc_core.output.pg.lvs.v
sparc_core.elab.ddc  sparc_core.mapped.fp  sparc_core.output.pt.sdf
sparc_core.erc.results sparc_core.mapped.pt.sdf sparc_core.output.pt.v
[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
```
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.mapped.sdc  sparc_core.output.sbpf.max
sparc_core         sparc_core.icc.lib  sparc_core.mapped.sdf  sparc_core.output.sbpf.min
sparc_core.dc.lib  sparc_core.icc.scope sparc_core.mapped.spef  sparc_core.output.sdc
sparc_core.dc.scope sparc_core.icc_constr.pt sparc_core.mapped.svf   sparc_core.output.sdf
sparc_core.dc_constr.pt sparc_core.icc_lib.db sparc_core.mapped.v      sparc_core.output.v
sparc_core.dc_lib.db sparc_core.icc_test.db sparc_core.output.dc.v  svdb
sparc_core.dc_test.db sparc_core.initial.fp  sparc_core.output.def
sparc_core.drc.results sparc_core.mapped.ddc sparc_core.output.pg.lvs.v
sparc_core.elab.ddc  sparc_core.mapped.fp  sparc_core.output.pt.sdf
sparc_core.erc.results sparc_core.mapped.pt.sdf sparc_core.output.pt.v
[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
```
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.mapped.sdc   sparc_core.output.sbpf.max
sparc_core           sparc_core.icc.lib  sparc_core.mapped.sdf   sparc_core.output.sbpf.min
sparc_core.dc.lib    sparc_core.icc.scope  sparc_core.mapped.spef
sparc_core.dc.scope  sparc_core.icc_constr.pt  sparc_core.mapped.svf
sparc_core.dc_constr.pt sparc_core.icc_lib.db  sparc_core.mapped.v
sparc_core.dc_lib.db  sparc_core.icc_test.db  sparc_core.output.dc.v
sparc_core.drc.results  sparc_core.initial.fp  sparc_core.output.def
sparc_core.elab.ddc  sparc_core.mapped.ddc  sparc_core.output.pg.lvs.v
sparc_core.erc.results  sparc_core.mapped.ddc  sparc_core.output.pt.sdf
sparc_core.erc.results  sparc_core.mapped.pt.sdf  sparc_core.output.pt.v

[synopsys]mmckeown@hanoi$ ls reports/
dc_shell_pass1  fm_shell_eco_pass1  pt_shell_dc_pass1  sparcl_core.drc.summary  sparcl_core.lvs.report
eco_shell_pass1  fm_shell_icc_pass1  pt_shell_eco_pass1  sparcl_core.erc.summary
fm_shell_dc_pass1  icc_shell_pass1  pt_shell_icc_pass1  sparcl_core.gds2sp.report.ext
```
Flow Outputs

```
$ pwd
/tank/mmckeown/research/projects/piton/piton_master/piton/design/chip/tile/sparc/synopsys
$ ls results/
eco_changes_dc.tcl       sparc_core.gds       sparc_core.mapped.sdc       sparc_core.output.sbpf.max
sparc_core               sparc_core.icc.lib    sparc_core.mapped.sdf       sparc_core.output.sbpf.min
sparc_core.dc.lib        sparc_core.icc.scope  sparc_core.mapped.spef      sparc_core.output.sdc
sparc_core.dc.scope      sparc_core.icc_constr.pt sparc_core.mapped.svf       sparc_core.output.sdf
sparc_core.dc_constr.pt  sparc_core.icc_lib.db  sparc_core.mapped.v         sparc_core.output.v
sparc_core.dc_lib.db     sparc_core.icc_test.db  sparc_core.output.dc.v      svdb
sparc_core.dc_test.db    sparc_core.initial.fp  sparc_core.output.def
sparc_core.drc.results   sparc_core.mapped.ddc  sparc_core.output.pg.lvs.v
sparc_core.elab.ddc      sparc_core.mapped.fp   sparc_core.output.pt.sdf
sparc_core.erc.results   sparc_core.mapped.pt.sdf sparc_core.output.pt.v
$ 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
```
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.mapped.sdc           sparc_core.output.sbpf.max
sparc_core                   sparc_core.icc.lib        sparc_core.mapped.sdf          sparc_core.output.sbpf.min
sparc_core.dc.lib            sparc_core.icc.scope       sparc_core.mapped.spef         sparc_core.output.sdc
sparc_core.dc.scope          sparc_core.icc Constr.pt   sparc_core.mapped.svf          sparc_core.output.sdf
sparc_core.dc Constr.pt      sparc_core.icc_lib.db      sparc_core.mapped.v            sparc_core.output.v
sparc_core.dc_lib.db         sparc_core.icc_test.db     sparc_core.output dc.v          svdb
sparc_core dc_test.db        sparc_core.initial.fp      sparc_core.output.def         svdb
sparc_core.drc.results       sparc_core.mapped.dcc      sparc_core.output.pg.lvs.v     svdb
sparc_core.elab.dcc          sparc_core.mapped.fpp      sparc_core.output pt.sdf       svdb
sparc_core.erc results       sparc_core.mapped.pt.sdf   sparc_core.output pt.v         svdb

[synopsys]mmckeown@hanoi$ ls reports/
dc_shell_pass1               fm_shell_eco_pass1         pt_shell_dc_pass1            sparc_core.drc.summary
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
sparc_core.lvs.report         sparc_core.erc.summary    ```
Opening the Design

[synopsys]mmckeown@hanoi$ icc_shell -gui
Opening the Design
Opening the Design

[synopsys]mmckeown@hanoi$ icc_shell -gui
Opening the Design

```
[synopsys]mmcckeown@hanoi$ icc_shell -gui
```
Opening the Design
Opening the Design
Opening the Design
Opening the Design
Opening the Design
Opening the Design