

Tutorial 101: Benchmarking and Architecture Exploration



OpenFPGA Tutorial

Tutorial 101: Benchmarking and Architecture Exploration

ORGANIZERS








Tutorial 101: Benchmarking and Architecture Exploration

OpenFPGA Github Tutorial Repositor

https://github.com/Inis-uofu/OpenFPGA_tutorials

Tutorial 101: Benchmarking and Architecture Exploration

Tutorial Repo

	AllenDBoston Update readme.md	46cd4f4 2 hours ago	 67 commits
	DATE23	Update readme.md	2 hours ago
	LICENSE	Initial commit	yesterday
	README.md	Update README.md	19 hours ago

README.md



Welcome to OpenFPGA Tutorials

Here we provide a repository with hands on tutorials on how to use OpenFPGA. Navigate to appropriate folder for your in-person session.

[Design Test and Automation in Europe 2023](#): M03 Embedded FPGAs (eFPGA) and Applications to IP Protection via eFPGA Redaction




Tutorial 101: Benchmarking and Architecture Exploration

Tutorial Repo



AllenDBoston Update readme.md

46cd4f4 2 hours ago 67 commits

 DATE23	Update readme.md	2 hours ago
 LICENSE	Initial commit	yesterday
 README.md	Update README.md	19 hours ago

README.md 

Welcome to OpenFPGA Tutorials

Here we provide a repository with hands on tutorials on how to use OpenFPGA. Navigate to appropriate folder for your in-person session.

[Design Test and Automation in Europe 2023: M03 Embedded FPGAs \(eFPGA\) and Applications to IP Protection via eFPGA Redaction](#)

Tutorial 101: Benchmarking and Architecture Exploration

Open the Interactive Binder

Contributor

13 lines (7 sloc) | 690 Bytes

Welcome to M03 Embedded FPGAs (eFPGA) and Applications to IP Prot Redaction

Part 1: OpenFPGA

Get started by opening a interactive notebook by clicking the binder icon located [here](#)

Slides for the tutorials are provided here:

[Tutorial 101](#) Benchmarking and Architecture Exploration

[Tutorial 102](#) FPGA Fabric Netlist Generation

[OpenFPGA Git Repository](#)

[Give feedback](#)

Tutorial 101: Benchmarking and Architecture Exploration

Tutorial Slides

🔗 Welcome to M03 Embedded FPGAs (eFPGA) and Applications to IP Protection

Part 1: OpenFPGA

Get started by opening a interactive notebook by clicking the binder icon located [here](#)

Slides for the tutorials are provided here:

[Tutorial 101](#) Benchmarking and Architecture Exploration

[Tutorial 102](#) FPGA Fabric Netlist Generation

[OpenFPGA Git Repository](#)

[Give feedback](#)

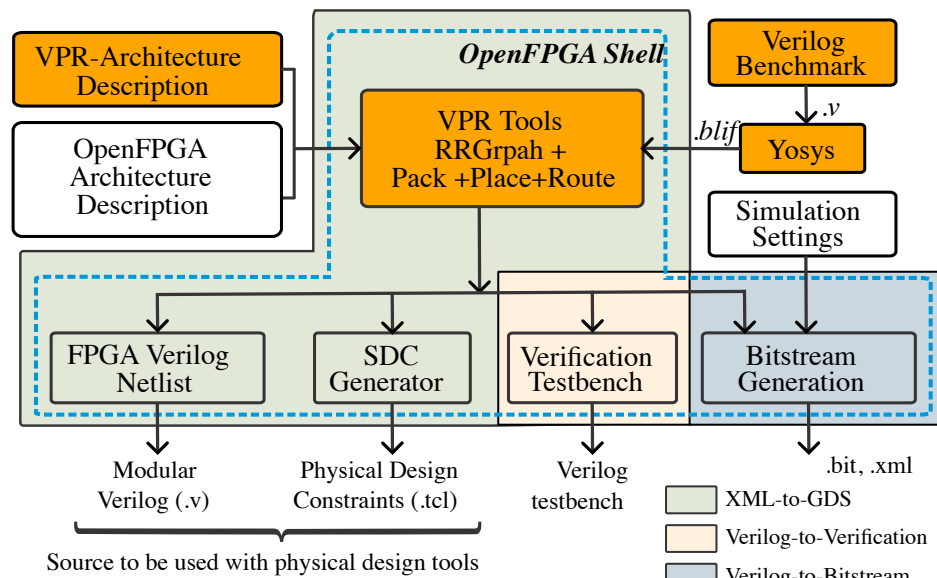
Tutorial 101: Benchmarking and Architecture Exploration

Objective

Evaluate the impact of **different FPGA architecture choice** on the given set of benchmarks **using OpenFPGA**.

Tutorial 101: Benchmarking and Architecture Exploration

Architecture Exploration Flow



Tutorial 101: Benchmarking and Architecture Exploration

Given set of Benchmarks

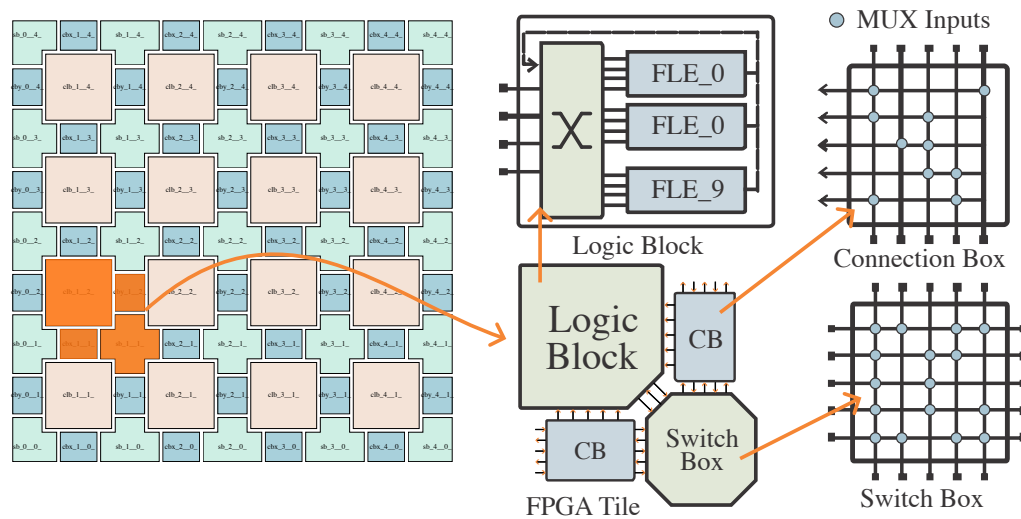
1. **ch_intrinsics** : Memory Init
2. **diffeq1** : Arithmetic Unit
3. **diffeq2** : Arithmetic Unit
4. **sha** : Cryptography Unit

More benchmarks are available at [openfpga_flow](#)/benchmarks/`

Tutorial 101: Benchmarking and Architecture Exploration

Candidate Architectures

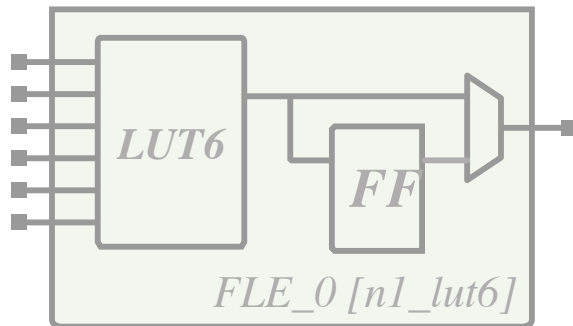
1. Homogeneous FPGA architecture
2. With 300 tracks/channels
3. All wires are length-4
4. All architectures have a full crossbar in the CLB



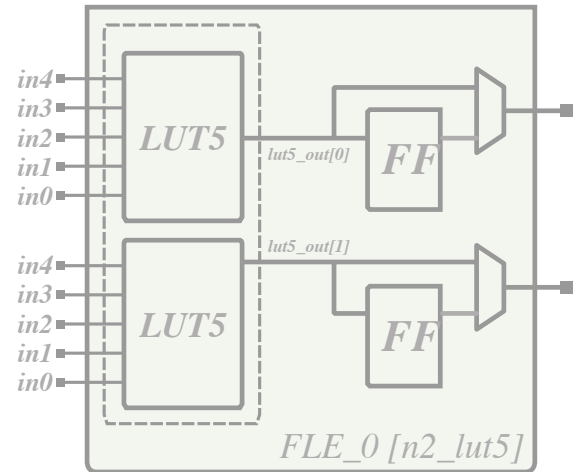
Tutorial 101: Benchmarking and Architecture Exploration

Fracturable Logic Block (FLE)

Arch 1: 6-input LUT (1× 6-input)



Arch 2: Fracturable LUT (1× 6-input or 2× 5-input LUT)

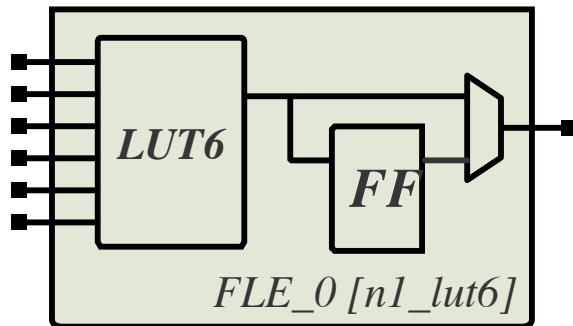


More example architecture are available at openfpga_flow/vpr_arch/

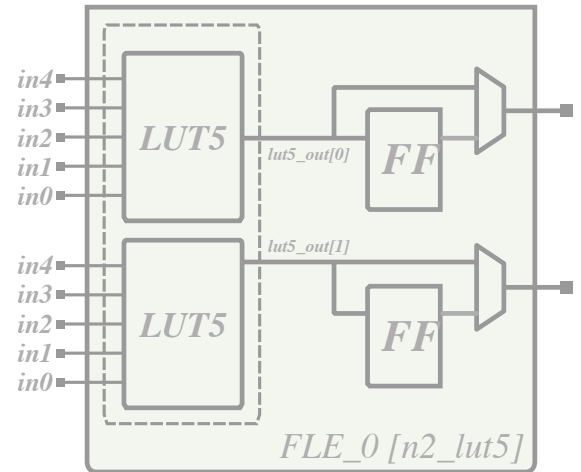
Tutorial 101: Benchmarking and Architecture Exploration

Fracturable Logic Block (FLE)

Arch 1: 6-input LUT (1× 6-input)



Arch 2: Fracturable LUT (1× 6-input or 2× 5-input LUT)

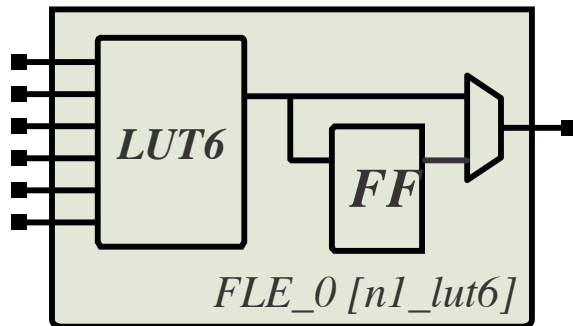


More example architecture are available at openfpga_flow/vpr_arch/

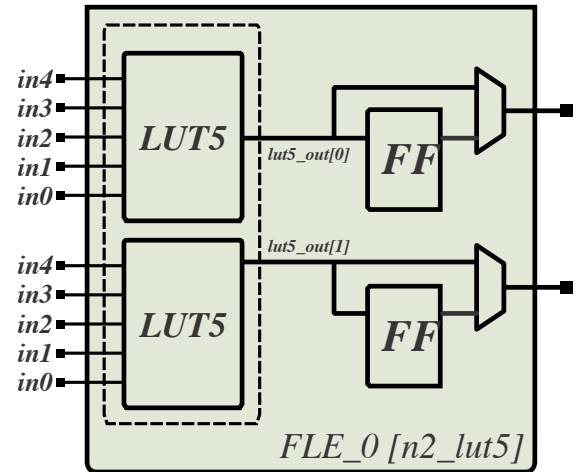
Tutorial 101: Benchmarking and Architecture Exploration

Fracturable Logic Block (FLE)

Arch 1: 6-input LUT (1× 6-input)



Arch 2: Fracturable LUT (1× 6-input or 2× 5-input LUT)

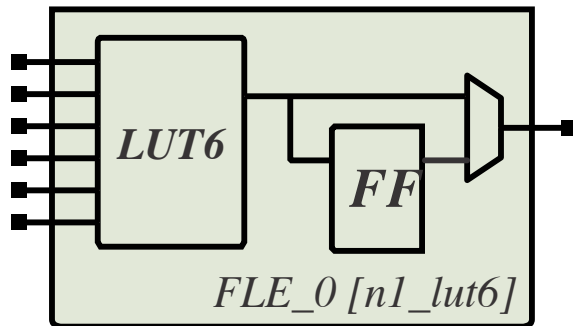


More example architecture are available at openfpga_flow/vpr_arch/

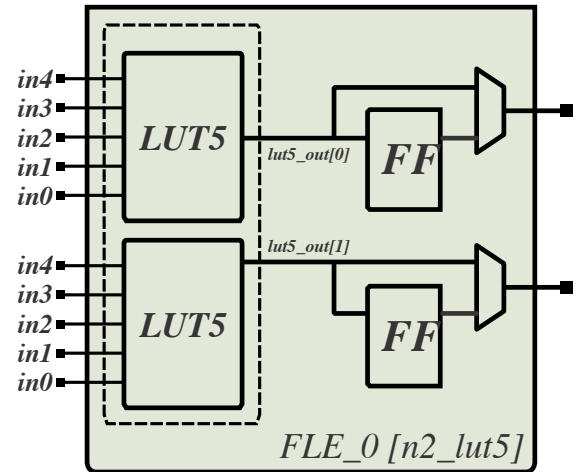
Tutorial 101: Benchmarking and Architecture Exploration

Fracturable Logic Block (FLE)

Arch 1: 6-input LUT (1× 6-input)



Arch 2: Fracturable LUT (1× 6-input or 2× 5-input LUT)



More example architecture are available at openfpga_flow/vpr_arch/

Tutorial 101: Benchmarking and Architecture Exploration

Binder

The screenshot displays the Binder web interface at hub.gke2.mybinder.org. The interface is divided into three main sections:

- EXPLORER:** A sidebar on the left showing the file system structure. It includes folders like `.github`, `.vscode`, `build`, `cmake`, `dev`, `docker`, `docs`, `libs`, `openfpga`, `openfpga_flow`, `vtr-verilog-to-routing`, `yosys`, `yosys-plugins`, `.clang-format`, `.dockerignore`, `.gitignore`, `.gitmodules`, `.readthedocs.yml`, `CMakeLists.txt`, `Dockerfile`, `LICENSE`, `Makefile`, `openfpga.sh`, `README.md`, `requirements.txt`, `VERSION.md`, and `wavetrace.wavetrace.vsix`. It also shows an `OUTLINE` and `TIMELINE` view.
- code-server:** The main workspace area with the heading "Editing evolved". It includes a "Start" section with options: "New File...", "Open File...", and "Clone Git Repository...". Below this is a "Recent" section stating "You have no recent folders, open a folder to start." and a "Walkthroughs" section with links: "Get Started with VS Code for the Web", "Learn the Fundamentals", and "Boost your Productivity".
- Next Up:** A section titled "Deploy code-server for your team" with a description of Code Server and a "Get started" link. Below this is a "Templates" section showing a table of available templates.

Name	Used By
Kubernetes	128 developers
Windows	21 developers

The bottom status bar shows the current branch as `master*` and the layout as `U.S.`.

Press **ctrl+shift+`** to open terminal

Tutorial 101: Benchmarking and Architecture Exploration

How to use OpenFPGA

Load OpenFPGA Environment

```
> source openfpga.sh
OPENFPGA_PATH=/opt/openfpga
shopt
```

Create OpenFPGA Task

```
# create-task <new_task_dir_name> <template_name>
> create-task lab1 template_tasks/frac-lut-arch-explore_template
Creating task      lab1
Template project   template_tasks/frac-lut-arch-explore_template
```

Run OpenFPGA Task

```
# run-task <task_dir_name>
> run-task lab1
```

Hint: use **Tab** to auto complete

Tutorial 101: Benchmarking and Architecture Exploration

How to use OpenFPGA

Load OpenFPGA Environment

```
> source openfpga.sh
OPENFPGA_PATH=/opt/openfpga
shopt
```

Create OpenFPGA Task

```
# create-task <new_task_dir_name> <template_name>
> create-task lab1 template_tasks/frac-lut-arch-explore_template
Creating task      lab1
Template project   template_tasks/frac-lut-arch-explore_template
```

Run OpenFPGA Task

```
# run-task <task_dir_name>
> run-task lab1
```

Hint: use **Tab** to auto complete

Tutorial 101: Benchmarking and Architecture Exploration

How to use OpenFPGA

Load OpenFPGA Environment

```
> source openfpga.sh
OPENFPGA_PATH=/opt/openfpga
shopt
```

Create OpenFPGA Task

```
# create-task <new_task_dir_name> <template_name>
> create-task lab1 template_tasks/frac-lut-arch-explore_template
Creating task      lab1
Template project   template_tasks/frac-lut-arch-explore_template
```

Run OpenFPGA Task

```
# run-task <task_dir_name>
> run-task lab1
```

Hint: use **Tab** to auto complete

Tutorial 101: Benchmarking and Architecture Exploration

Content of the task directory

Any directory with `config/task.conf` file is an OpenFPGA task directory

```
> tree lab1 -L 2
lab1/
├── config
│   └── task.conf
├── k6_frac_N10_tileable.xml
├── k6_N10_tileable.xml
└── vtr_benchmark_template_script.openfpga
```

OpenFPGA-Shell Commands

(Similar to the TCL script file of any EDA tool)

```
# Execute VPR for architecture exploration
vpr ${VPR_ARCH_FILE} ${VPR_TESTBENCH_BLIF} \
    --route_chan_width ${VPR_ROUTE_CHAN_WIDTH} \
    --constant_net_method route
exit
```

Tutorial 101: Benchmarking and Architecture Exploration

Content of the task directory

Any directory with `config/task.conf` file is an OpenFPGA task directory

```
> tree lab1 -L 2
lab1/
├── config
│   └── task.conf
├── k6_frac_N10_tileable.xml
├── k6_N10_tileable.xml
└── vtr_benchmark_template_script.openfpga
```

OpenFPGA-Shell Commands

(Similar to the TCL script file of any EDA tool)

```
# Execute VPR for architecture exploration
vpr ${VPR_ARCH_FILE} ${VPR_TESTBENCH_BLIF} \
    --route_chan_width ${VPR_ROUTE_CHAN_WIDTH} \
    --constant_net_method route
exit
```

Tutorial 101: Benchmarking and Architecture Exploration

Configuration File Content

General Section

```
[GENERAL]
run_engine=openfpga_shell # default
power_tech_file = ${PATH:OPENFPGA_PATH}/openfpga_flow/tech/PTM_45nm/45nm.xml
power_analysis = false
spice_output=false
verilog_output=true
timeout_each_job = 20*60
fpga_flow=yosys_vpr # yosys_vpr or vpr_blif
```

Openfpga_shell Section

```
[OpenFPGA_SHELL]
openfpga_shell_template=${PATH:TASK_DIR}/vtr_benchmark_template_script.openfpga
openfpga_arch_file=${PATH:OPENFPGA_PATH}/openfpga_flow/openfpga_arch/k6_frac_N10
openfpga_sim_setting_file=${PATH:OPENFPGA_PATH}/openfpga_flow/openfpga_simulation
vpr_route_chan_width=300
```

- **\${PATH:TASK_DIR}** : Points to the root directory of the task
- **\${PATH:OPENFPGA_PATH}** : Points to the root directory of the OpenFPGA repository

Tutorial 101: Benchmarking and Architecture Exploration

Configuration File Content

General Section

```
[GENERAL]
run_engine=openfpga_shell # default
power_tech_file = ${PATH:OPENFPGA_PATH}/openfpga_flow/tech/PTM_45nm/45nm.xml
power_analysis = false
spice_output=false
verilog_output=true
timeout_each_job = 20*60
fpga_flow=yosys_vpr # yosys_vpr or vpr_blif
```

Openfpga_shell Section

```
[OpenFPGA_SHELL]
openfpga_shell_template=${PATH:TASK_DIR}/vtr_benchmark_template_script.openfpga
openfpga_arch_file=${PATH:OPENFPGA_PATH}/openfpga_flow/openfpga_arch/k6_frac_N10
openfpga_sim_setting_file=${PATH:OPENFPGA_PATH}/openfpga_flow/openfpga_simulation
vpr_route_chan_width=300
```

- **\${PATH:TASK_DIR}** : Points to the root directory of the task
- **\${PATH:OPENFPGA_PATH}** : Points to the root directory of the OpenFPGA repository

Tutorial 101: Benchmarking and Architecture Exploration

Configuration File Content

General Section

```
[GENERAL]
run_engine=openfpga_shell # default
power_tech_file = ${PATH:OPENFPGA_PATH}/openfpga_flow/tech/PTM_45nm/45nm.xml
power_analysis = false
spice_output=false
verilog_output=true
timeout_each_job = 20*60
fpga_flow=yosys_vpr # yosys_vpr or vpr_blif
```

Openfpga_shell Section

```
[OpenFPGA_SHELL]
openfpga_shell_template=${PATH:TASK_DIR}/vtr_benchmark_template_script.openfpga
openfpga_arch_file=${PATH:OPENFPGA_PATH}/openfpga_flow/openfpga_arch/k6_frac_N10
openfpga_sim_setting_file=${PATH:OPENFPGA_PATH}/openfpga_flow/openfpga_simulation
vpr_route_chan_width=300
```

- **\${PATH:TASK_DIR}** : Points to the root directory of the task
- **\${PATH:OPENFPGA_PATH}** : Points to the root directory of the OpenFPGA repository

Tutorial 101: Benchmarking and Architecture Exploration

VPR Architecture Section (2 architectures)

[ARCHITECTURES]

```
arch0=${PATH:TASK_DIR}/k6_N10_tileable.xml
arch1=${PATH:TASK_DIR}/k6_frac_N10_tileable.xml
```

- `${PATH:VPR_ARCH_PATH}` : Points to VPR arch file in the openfpga repository

Benchmark Section (4 benchmarks)

[BENCHMARKS]

```
bench1=${PATH:BENCH_PATH}/vtr_benchmark/ch_intrinsics.v
bench2=${PATH:BENCH_PATH}/vtr_benchmark/diffeq1.v
bench3=${PATH:BENCH_PATH}/vtr_benchmark/diffeq2.v
bench4=${PATH:BENCH_PATH}/vtr_benchmark/sha.v
```

- `${PATH:BENCH_PATH}` : Points to benchmarks in the openfpga repository

Synthesis Parameters

[SYNTHESIS_PARAM]

```
# Yosys script parameters
bench_read_verilog_options_common = -nolatches
bench_yosys_common=${PATH:OPENFPGA_PATH}/openfpga_flow/misc/ys_tmpl_yosys_vpr_flow
# Benchmark top-module name
bench1_top = memset
bench2_top = diffeq_paj_convert
bench3_top = diffeq_f_systemC
bench4_top = sha1
```

run job name format <arch_num>_<top_module>_

Tutorial 101: Benchmarking and Architecture Exploration

VPR Architecture Section (2 architectures)

[ARCHITECTURES]

```
arch0=${PATH:TASK_DIR}/k6_N10_tileable.xml
arch1=${PATH:TASK_DIR}/k6_frac_N10_tileable.xml
```

- `${PATH:VPR_ARCH_PATH}` : Points to VPR arch file in the openfpga repository

Benchmark Section (4 benchamarks)

[BENCHMARKS]

```
bench1=${PATH:BENCH_PATH}/vtr_benchmark/ch_intrinsics.v
bench2=${PATH:BENCH_PATH}/vtr_benchmark/diffeq1.v
bench3=${PATH:BENCH_PATH}/vtr_benchmark/diffeq2.v
bench4=${PATH:BENCH_PATH}/vtr_benchmark/sha.v
```

- `${PATH:BENCH_PATH}` : Points to bencharks in the openfpga repository

Synthesis Parameters

[SYNTHESIS_PARAM]

```
# Yosys script parameters
bench_read_verilog_options_common = -nolatches
bench_yosys_common=${PATH:OPENFPGA_PATH}/openfpga_flow/misc/ys_tmpl_yosys_vpr_flow
# Benchmark top-module name
bench1_top = memset
bench2_top = diffeq_paj_convert
bench3_top = diffeq_f_systemC
bench4_top = sha1
```

run job name format <arch_num>_<top_module>_

Tutorial 101: Benchmarking and Architecture Exploration

VPR Architecture Section (2 architectures)

[ARCHITECTURES]

```
arch0=${PATH:TASK_DIR}/k6_N10_tileable.xml
arch1=${PATH:TASK_DIR}/k6_frac_N10_tileable.xml
```

- `${PATH:VPR_ARCH_PATH}` : Points to VPR arch file in the openfpga repository

Benchmark Section (4 benchamarks)

[BENCHMARKS]

```
bench1=${PATH:BENCH_PATH}/vtr_benchmark/ch_intrinsics.v
bench2=${PATH:BENCH_PATH}/vtr_benchmark/diffeq1.v
bench3=${PATH:BENCH_PATH}/vtr_benchmark/diffeq2.v
bench4=${PATH:BENCH_PATH}/vtr_benchmark/sha.v
```

- `${PATH:BENCH_PATH}` : Points to bencharks in the openfpga repository

Synthesis Parameters

[SYNTHESIS_PARAM]

```
# Yosys script parameters
bench_read_verilog_options_common = -nolatches
bench_yosys_common=${PATH:OPENFPGA_PATH}/openfpga_flow/misc/ys_tmpl_yosys_vpr_flow
# Benchmark top-module name
bench1_top = memset
bench2_top = diffeq_paj_convert
bench3_top = diffeq_f_systemC
bench4_top = sha1
```

run job name format <arch_num>_<top_module>_

Tutorial 101: Benchmarking and Architecture Exploration

Script Parameters Section

```
[SCRIPT_PARAM_]
# empty
```

Post execution result extraction

```
[DEFAULT_PARSE_RESULT_VPR]
01_lut6_use = "lut6      : ([0-9]+)", int
02_lut5_use = "lut5      : ([0-9]+)", int
```

Tutorial 101: Benchmarking and Architecture Exploration

Architecture XML File Difference

```
# ===== k6_N10_tileable.xml =====  
architecture/  
  |— models  
  |— tiles  
  |— layout  
  |— device  
  |— switchlist  
  |— segmentlist  
  |— complexblocklist  
    |— <pb_type name="io" ...  
      |— .....  
    |— <pb_type name="clb" ...  
      |— <pb_type name="fle" ...  
        |— mode [n1_lut6]
```

```
# ===== k6_frac_N10_tileable.xml =====  
architecture/  
  |— models  
  |— tiles  
  |— layout  
  |— device  
  |— switchlist  
  |— segmentlist  
  |— complexblocklist  
    |— <pb_type name="io" ...  
      |— .....  
    |— <pb_type name="clb" ...  
      |— <pb_type name="fle" ...  
        |— mode [n1_lut6]  
        |— mode [n2_lut5] <<<<<
```

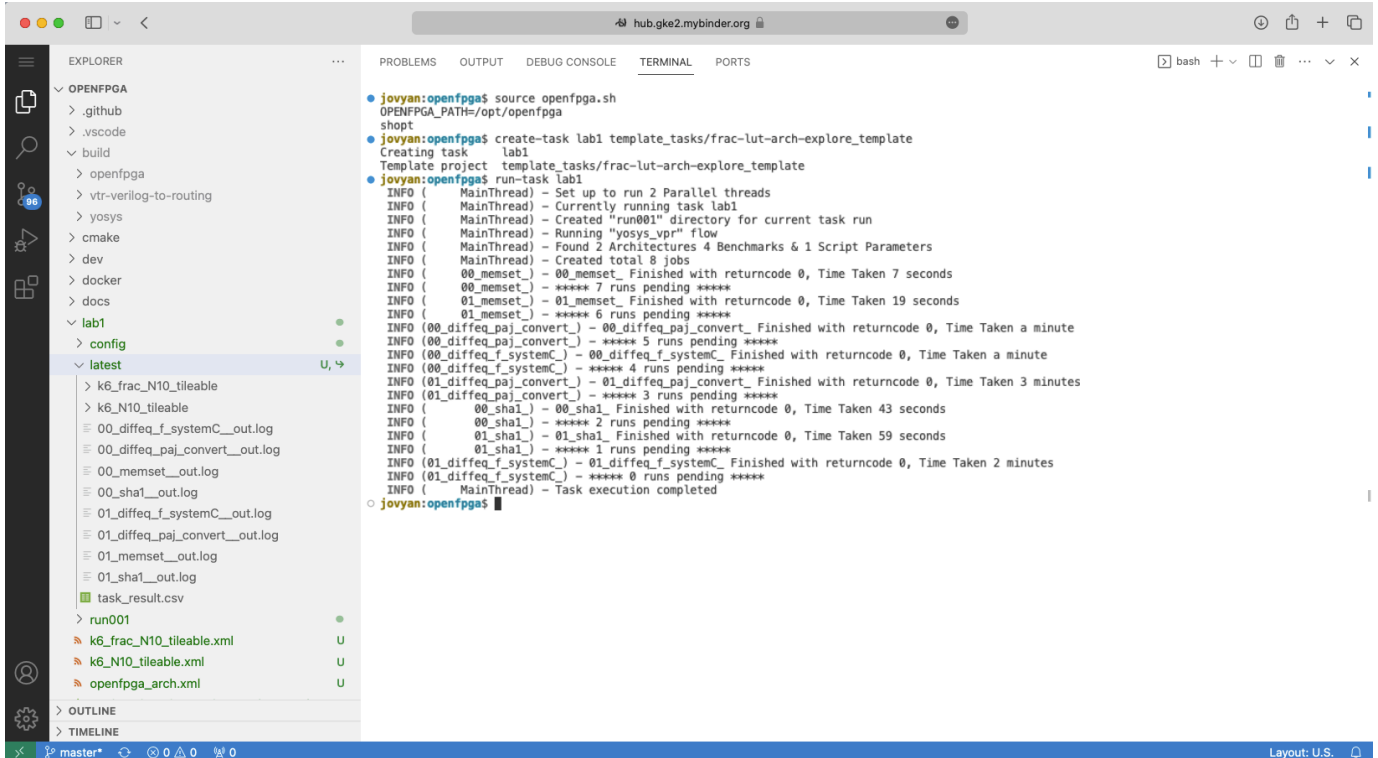
Added **<mode name="n2_lut5">...</mode>**

Run Directory (after task execution)

```
lab1/
├── config
│   └── task.conf
├── k6_frac_N10_tileable.xml
├── k6_N10_tileable.xml
├── vtr_benchmark_template_script.openfpga
├── latest
└── run001
    ├── k6_frac_N10_tileable      <<<< arch0
    │   ├── k6_N10_tileable      <<<< bench0
    │   ├── diffeq_paj_convert   <<<< bench1
    │   ├── diffeq_f_systemC     <<<< bench2
    │   └── sha1                 <<<< bench3
    ├── k6_N10_tileable          <<<< arch1
    │   ├── memset               <<<< bench0
    │   ├── diffeq_paj_convert   <<<< bench1
    │   ├── diffeq_f_systemC     <<<< bench2
    │   └── sha1                 <<<< bench3
    └── task_result.csv          <<<<<<<<<<<<<<<<
```

Tutorial 101: Benchmarking and Architecture Exploration

Task Completed

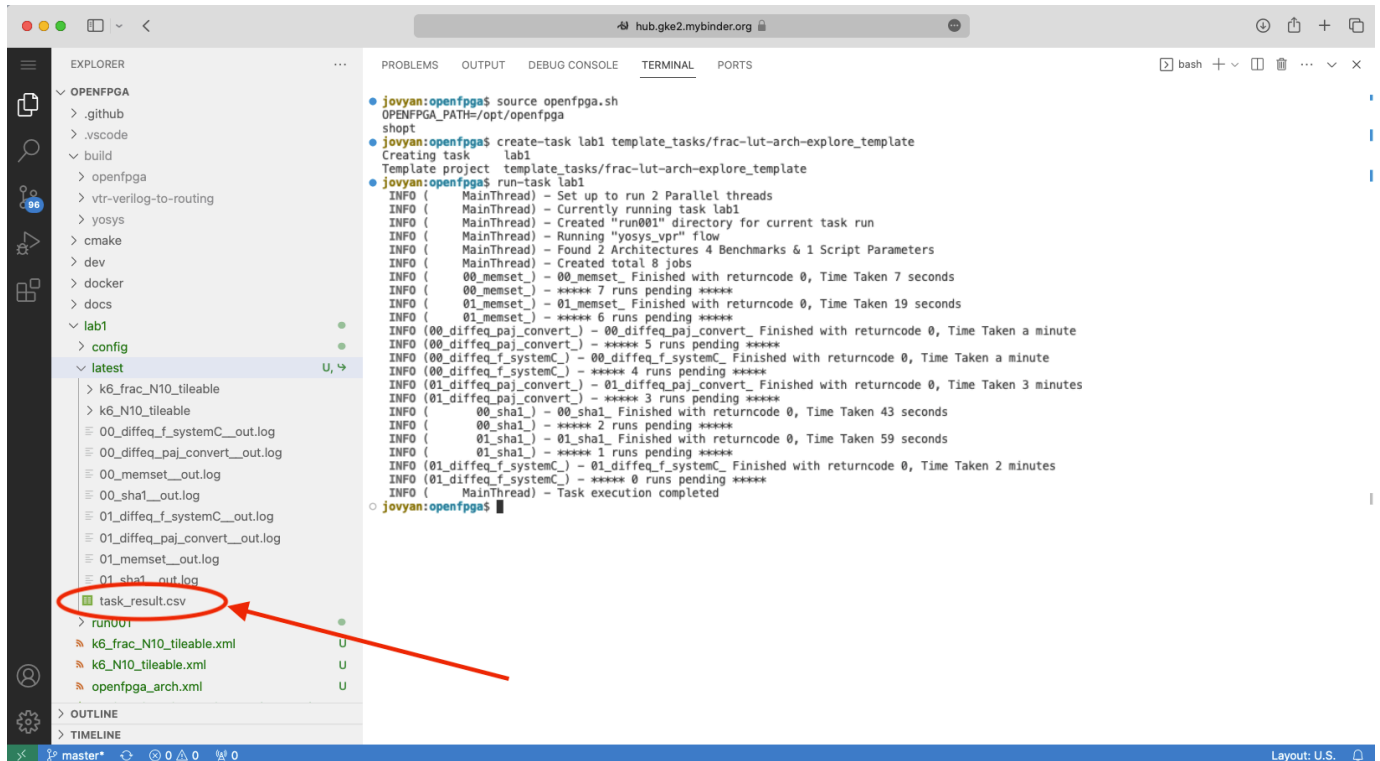


```
jovyan:openfpga$ source openfpga.sh
OPENFPGA_PATH=/opt/openfpga
shopt
jovyan:openfpga$ create-task lab1 template_tasks/frac-lut-arch-explore_template
Creating task lab1
Template project template_tasks/frac-lut-arch-explore_template
jovyan:openfpga$ run-task lab1
INFO ( MainThread) - Set up to run 2 Parallel threads
INFO ( MainThread) - Currently running task lab1
INFO ( MainThread) - Created "run001" directory for current task run
INFO ( MainThread) - Running "yosys_vpr" flow
INFO ( MainThread) - Found 2 Architectures 4 Benchmarks & 1 Script Parameters
INFO ( MainThread) - Created total 8 jobs
INFO ( 00_memset ) - 00_memset_Finished with returncode 0, Time Taken 7 seconds
INFO ( 00_memset ) - ***** 7 runs pending *****
INFO ( 01_memset ) - 01_memset_Finished with returncode 0, Time Taken 19 seconds
INFO ( 01_memset ) - ***** 6 runs pending *****
INFO ( 00_diff_eq_paj_convert ) - 00_diff_eq_paj_convert_Finished with returncode 0, Time Taken a minute
INFO ( 00_diff_eq_paj_convert ) - ***** 5 runs pending *****
INFO ( 00_diff_eq_f_systemC ) - 00_diff_eq_f_systemC_Finished with returncode 0, Time Taken a minute
INFO ( 00_diff_eq_f_systemC ) - ***** 4 runs pending *****
INFO ( 01_diff_eq_paj_convert ) - 01_diff_eq_paj_convert_Finished with returncode 0, Time Taken 3 minutes
INFO ( 01_diff_eq_paj_convert ) - ***** 3 runs pending *****
INFO ( 00_sha1 ) - 00_sha1_Finished with returncode 0, Time Taken 43 seconds
INFO ( 00_sha1 ) - ***** 2 runs pending *****
INFO ( 01_sha1 ) - 01_sha1_Finished with returncode 0, Time Taken 59 seconds
INFO ( 01_sha1 ) - ***** 1 runs pending *****
INFO ( 01_diff_eq_f_systemC ) - 01_diff_eq_f_systemC_Finished with returncode 0, Time Taken 2 minutes
INFO ( 01_diff_eq_f_systemC ) - ***** 0 runs pending *****
INFO ( MainThread ) - Task execution completed
jovyan:openfpga$
```

Result located: **lab1/latest/task_results.csv**

Tutorial 101: Benchmarking and Architecture Exploration

Finding the Results



```
jovyan:openfpga$ source openfpga.sh
OPENFPGA_PATH=/opt/openfpga
shopt
jovyan:openfpga$ create-task lab1 template_tasks/frac-lut-arch-explore_template
Creating task lab1
Template project template_tasks/frac-lut-arch-explore_template
jovyan:openfpga$ run-task lab1
INFO ( MainThread) - Set up to run 2 Parallel threads
INFO ( MainThread) - Currently running task lab1
INFO ( MainThread) - Created "run001" directory for current task run
INFO ( MainThread) - Running "yosys_vpr" flow
INFO ( MainThread) - Found 2 Architectures 4 Benchmarks & 1 Script Parameters
INFO ( MainThread) - Created total 8 jobs
INFO ( 00_memset ) - 00_memset_Finished with returncode 0, Time Taken 7 seconds
INFO ( 00_memset ) - ***** 7 runs pending *****
INFO ( 01_memset ) - 01_memset_Finished with returncode 0, Time Taken 19 seconds
INFO ( 01_memset ) - ***** 6 runs pending *****
INFO ( 00_diff_eq_paj_convert ) - 00_diff_eq_paj_convert_Finished with returncode 0, Time Taken a minute
INFO ( 00_diff_eq_paj_convert ) - ***** 5 runs pending *****
INFO ( 00_diff_eq_f_systemC ) - 00_diff_eq_f_systemC_Finished with returncode 0, Time Taken a minute
INFO ( 00_diff_eq_f_systemC ) - ***** 4 runs pending *****
INFO ( 01_diff_eq_paj_convert ) - 01_diff_eq_paj_convert_Finished with returncode 0, Time Taken 3 minutes
INFO ( 01_diff_eq_paj_convert ) - ***** 3 runs pending *****
INFO ( 00_sha1 ) - 00_sha1_Finished with returncode 0, Time Taken 43 seconds
INFO ( 00_sha1 ) - ***** 2 runs pending *****
INFO ( 01_sha1 ) - 01_sha1_Finished with returncode 0, Time Taken 59 seconds
INFO ( 01_sha1 ) - ***** 1 runs pending *****
INFO ( 01_diff_eq_f_systemC ) - 01_diff_eq_f_systemC_Finished with returncode 0, Time Taken 2 minutes
INFO ( 01_diff_eq_f_systemC ) - ***** 0 runs pending *****
INFO ( MainThread ) - Task execution completed
jovyan:openfpga$
```

Result located: **lab1/latest/task_results.csv**

Tutorial 101: Benchmarking and Architecture Exploration

Analyze Results

name	01_lut6_use	02_lut5_use	clb_blocks	total_wire_length
00_memset_	270		31	2210
01_memset_	46	224	26	2203
00_diffeq_paj_convert_	3540		368	43628
01_diffeq_paj_convert_	1316	2224	275	38642
00_diffeq_f_systemC_	3392		354	37096
01_diffeq_f_systemC_	1215	2177	260	33857
00_sha1_	1616		168	16027
01_sha1_	885	731	154	15182

Tutorial 101: Benchmarking and Architecture Exploration

Exercise

1. Consider more VTR benchmarks for performance comparison: **stereovision3**, **blob_merge**, and **bgm**
2. Extend the evaluation metrics and identify number final gr size of the FPGA for each benchmark (*Hint*: look for **"FPGA sized to"** sentence in *openfpgashell.log* file)

Tutorial 101: Benchmarking and Architecture Exploration

Exercise

1. Consider more VTR benchmarks for performance comparison: **stereovision3**, **blob_merge**, and **bgm**
2. Extend the evaluation metrics and identify number final grid size of the FPGA for each benchmark (*Hint*: look for **"FPGA sized to"** sentence in *openfpgashell.log* file)

Answer: `03_grid_size = "FPGA sized to(.*) x", str`