DESIGN, AUTOMATION & TEST IN EUROPE
17 – 19 April 2023 · Antwerp, Belgium
The European Event for Electronic System Design & Test

Embedded FPGAs (eFPGA) and Applications to IP Protection via eFPGA Redaction

Hands on Tutorial on ALICE (eFPGA Redaction)

Luca Collini

lc4976@nyu.edu

NYU | TANDON SCHOOL OF ENGINEERING

ALICE repository

Github Repo: https://github.com/Lucaz97/ALICE

ALICE: An automatic design flow for eFPGA redaction

launch binder

ALICE is a framework for eFPGA redaction. It helps identifying the best module combination to fit on the eFPGA with the fabric of your choice. you can either specify the relevant modules or select what outputs you want to protect and let ALICE do the work.

If you use our tool for your research please cite us!

```
Tomajoli, C. M., Collini, L., Bhandari, J., Moosa, A. K. T., Tan, B., Tang, X., Gaillardon, P. E.
ALICE: An Automatic Design Flow for eFPGA Redaction. In Proceedings of the 59th ACM/IEEE Design A
DAC 2022 (pp. 781–786). (Proceedings – Design Automation Conference). Institute of Electrical and
https://doi.org/10.1145/3489517.3530543
```

OPEN ACCESS PAPER PDF

ALICE repository

ALICE: An automatic design flow for eFPGA redaction

launch binder

Binder

ALICE is a framework for eFPGA redaction. It hel … on the eFPGA with the fabric of your choice. you can either spe … you want to protect and let ALICE do the work.

If you use our tool for your research please cite u …

```
Tomajoli, C. M., Collini, L., Bhandari, … aillardon, P. E.
ALICE: An Automatic Design Flow for eFPG … CM/IEEE Design A
DAC 2022 (pp. 781–786). (Proceedings – D … f Electrical and
https://doi.org/10.1145/3489517.3530543
```

OPEN ACCESS PAPER PDF

Demo 1: Two smaller eFPGA

- **2 smaller (4x4) eFPGAs**
- **Perform I/O filtering and only explore the Datapath module**

- **To run:**

```
make 2small
```

Demo 1: Two smaller eFPGA

2_small_efpgas.yaml

```yaml
define: []
exclude_modules: []
include: []
file_list:
- ../tests/openroad/gcd/src/gcd.v
max_fpga_num: '2' # 2 efpgas
max_io_num: '64' # 64 is the max IO for a 4x4
method: io+out+rank # filter by io count, output signal
module_names: []
not_allowed_size: [5,6,7] # we want a 4x4
out_dir: work
rank: '1'
signal_names: outputs
topmodule: gcd
openfpga_arch: ${OPENFPGA_FLOW_PATH}/openfpga_arch/k4_frac_N4_40nm_cc_openfpga.xml
vpr_arch: ${OPENFPGA_FLOW_PATH}/vpr_arch/k4_frac_N4_40nm.xml
```

Demo 1: Two smaller eFPGA

Terminal output

```
Top_epfga_12 --> score = 78.57142857142857
Top_epfga_14 --> score = 114.28571428571429
Top_epfga_15 --> score = 98.57142857142857
Top_epfga_16 --> score = 200.0
Top_epfga_17 --> score = 171.42857142857144
Top_epfga_18 --> score = 171.42857142857144
Top_epfga_20 --> score = 108.57142857142857
Top_epfga_22 --> score = 112.85714285714286
Time elapsed (process time) for third phase (eFPGA characterization + selection): 109.82210850715637s
Total number of modules = 11
-----
Number of allowed modules = 9
-----
Number of eFPGA candidates = 28
Number of eFPGA valid alternatives = 8
-----
Number of solutions = 16
```

Demo 1: Two smaller eFPGA

```
Number of allowed modules = 9
-----
Number of eFPGA candidates = 28
Number of eFPGA valid alternatives = 8
-----
Number of solutions = 16
-----
Total number of used eFPGA = 2
    - 4x4
    - 4x4
Total number of redacted modules = 2
```

Terminal output

```
eFPGA to be generated
Top_epfga_16
[gcd.dpath.b_mux]
MPM: gcd.dpath
Instance name: ctrl
to_fix: dpath
Instance name: dpath
to_fix: dpath
eFPGA to be generated
Top_epfga_17
[gcd.dpath.a_reg]
MPM: gcd.dpath
Instance name: ctrl
to_fix: dpath
Instance name: dpath
to_fix: dpath
```

Demo 1: Two smaller eFPGA

```
module gcd
(
  input  wire clk,
  input  wire [  31:0] req_msg,
  output wire req_rdy,
  input  wire req_val,
  input  wire reset,
  output wire [  15:0] resp_msg,
  input  wire resp_rdy,
  output wire resp_val
);
```

Original vs new top module interface

Added ports in top module of redacted design

```
  input wire Top_epfga_16_pReset,
  input wire Top_epfga_16_prog_clk,
  input wire Top_epfga_16_set,
  input wire Top_epfga_16_clk,
  input wire Top_epfga_16_ccff_head,
  output wire Top_epfga_16_ccff_tail,
  input wire Top_epfga_17_pReset,
  input wire Top_epfga_17_prog_clk,
  input wire Top_epfga_17_set,
  input wire Top_epfga_17_clk,
  input wire Top_epfga_17_ccff_head,
  output wire Top_epfga_17_ccff_tail
);
```

Demo 2: One bigger eFPGA

- **Single eFPGA, let us aim for a 5x5**
- **Perform I/O filtering and identify candidate modules from relevant outputs**

- **To run:**

```
make 1big
```

Demo 2: One bigger eFPGA

1_big_efpga.yaml

```yaml
define: []
exclude_modules: []
include: []
file_list:
- ../tests/openroad/gcd/src/gcd.v
max_fpga_num: '1' # 1 efpgas
max_io_num: '96' # 96 is the max IO for a 4x4
method: io+module # filter by io count, output signal
module_names: [GcdUnitDpathRTL_0x4d0fc71ead8d3d9e] # we want to look into the data path module only
not_allowed_size: [3,4,6,7,8] # we are aiming at a 5x5
out_dir: work
rank: '1'
signal_names: outputs
topmodule: gcd
openfpga_arch: ${OPENFPGA_FLOW_PATH}/openfpga_arch/k4_frac_N4_40nm_cc_openfpga.xml
vpr_arch: ${OPENFPGA_FLOW_PATH}/vpr_arch/k4_frac_N4_40nm.xml
```

Demo 2: One bigger eFPGA

Terminal output

```
Top_epfga_1 --> score = 180.0
Top_epfga_6 --> score = 135.0
Top_epfga_8 --> score = 180.0
Top_epfga_14 --> score = 107.5
Top_epfga_16 --> score = 155.0
Top_epfga_18 --> score = 142.5
Top_epfga_21 --> score = 122.5
Top_epfga_23 --> score = 162.5
Top_epfga_25 --> score = 122.5
Time elapsed (process time) for third phase (eFPGA characterization + selection): 89.760328769683845
```

```
Total number of modules = 11
-----
Number of allowed modules = 8
-----
Number of eFPGA candidates = 26
Number of eFPGA valid alternatives = 9
-----
Number of solutions = 9
-----
```

Demo 2: One bigger eFPGA

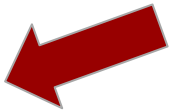
```
_____
Total number of used eFPGA = 1
    - 5x5
Total number of redacted modules = 2
eFPGA to be generated
Top_epfga_1
[gcd.dpath.a_reg, gcd.dpath.b_mux]
MPM: gcd.dpath
Instance name: ctrl
to_fix: dpath
Instance name: dpath
to_fix: dpath
```

Terminal output

Demo 2: One bigger eFPGA

```
module gcd
(
  input wire clk,
  input wire [31:0] req_msg,
  output wire req_rdy,
  input wire req_val,
  input wire reset,
  output wire [15:0] resp_msg,
  input wire resp_rdy,
  output wire resp_val,
  input wire Top_epfga_1_pReset,
  input wire Top_epfga_1_prog_clk,
  input wire Top_epfga_1_set,
  input wire Top_epfga_1_clk,
  input wire Top_epfga_1_ccff_head,
  output wire Top_epfga_1_ccff_tail
);
```

New top module interface



Added ports in top module of redacted design

Demo 3: A different fabric

- **1 eFPGA 4x4 with a bigger fabric (6 inputs LUTs, 8 LUTs in CLBs)**
- **Perform I/O filtering and identify candidate modules from relevant outputs**

- **To run:**

```
make fabric
```

Demo 3: A different fabric

different_fabric.yaml

```yaml
define: []
exclude_modules: []
include: []
file_list:
- ../tests/openroad/gcd/src/gcd.v
max_fpga_num: '1' # 2 efpgas
max_io_num: '96' # 64 is the max IO for a 4x4
method: io+module # filter by io count, output signal
module_names: [GcdUnitDpathRTL_0x4d0fc71ead8d3d9e] # we want to look into the data path module only
not_allowed_size: [3,6,7,8] # we want to see if using the bigger fabric we can get a 4x4
out_dir: work
rank: '1'
signal_names: outputs
topmodule: gcd
openfpga_arch: ${OPENFPGA_FLOW_PATH}/openfpga_arch/k6_frac_N8_40nm_openfpga.xml
vpr_arch: ${OPENFPGA_FLOW_PATH}/vpr_arch/k6_frac_N8_tileable_40nm.xml
```

Demo 3: A different fabric

```
Time elapsed (process time) for third phase (eFPGA characterization + selection): 116.40164542198181s
Total number of modules = 11
-----
Number of allowed modules = 8
-----
Number of eFPGA candidates = 26
Number of eFPGA valid alternatives = 22
-----
Number of solutions = 20
-----
Total number of used eFPGA = 1
    - 4x4
Total number of redacted modules = 1
eFPGA to be generated
Top_epfga_18
[gcd.dpath.sub]
MPM: gcd.dpath
Instance name: ctrl
to_fix: dpath
Instance name: dpath
to_fix: dpath
```

Terminal output

The end

Thank you for your participation!

lc4976@nyu.edu

https://github.com/Lucaz97/ALICE