QET Labs

An Electronic-Photonic Co-Design Approach with *piel*

Dario A Quintero Dominguez
PhD Student

Github: daquintero/piel

BIG PHOTON LAB

University of BRISTOL

University of BRISTOL

Outline

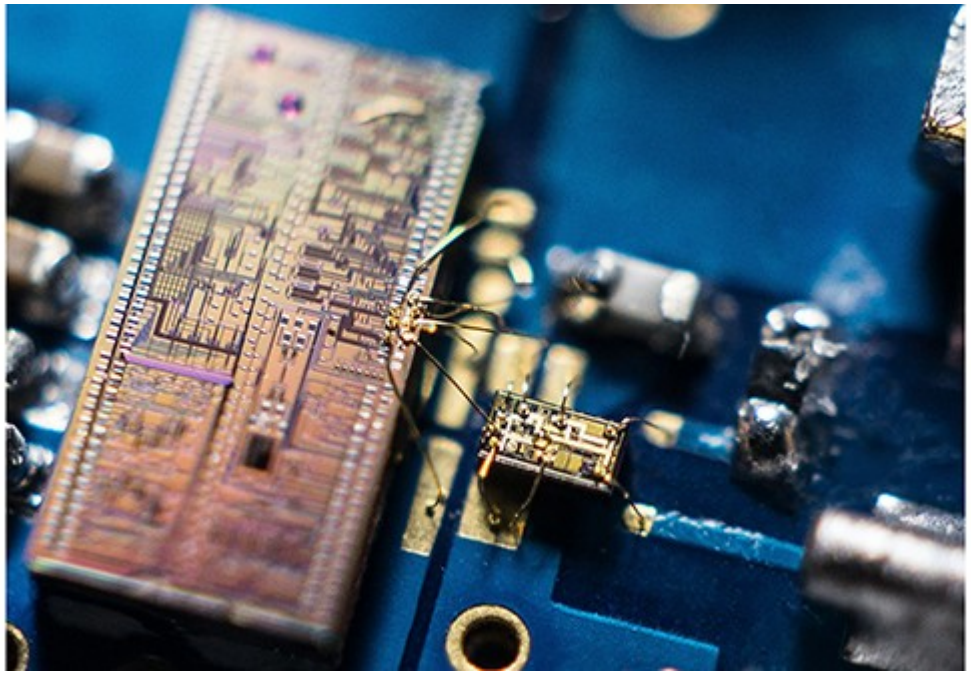
I. Motivations for Electronic-Photonic Co-Design
   I. Why Open-Source?

II. The piel Toolchain

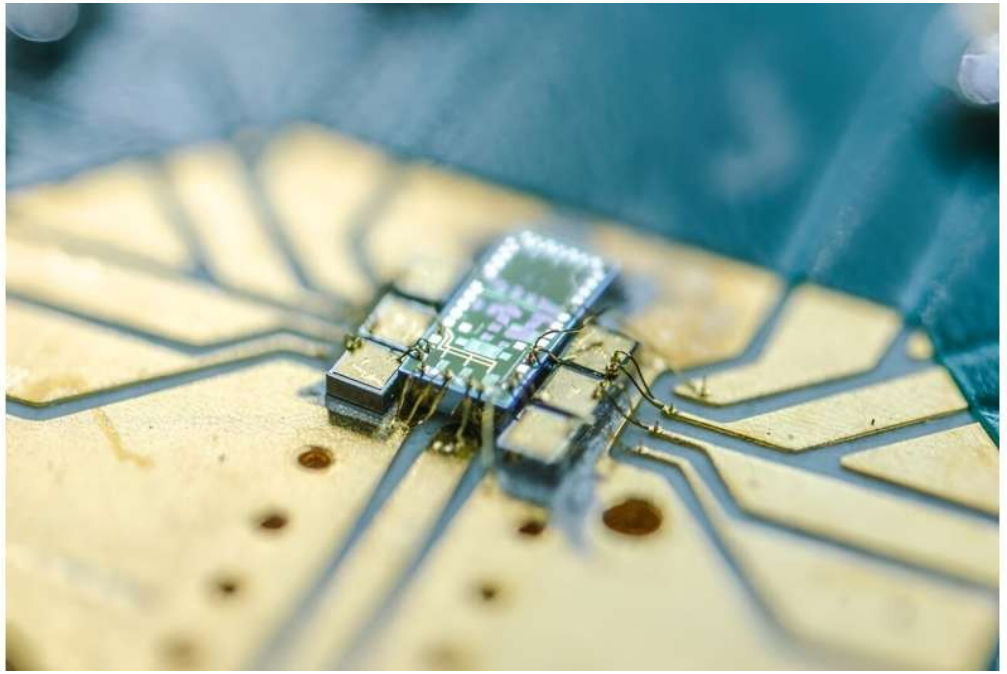
III. Examples towards a Full-Flow Demo

Github: daquintero/piel

Motivations for Co-Design



Tasker, Joel F., et al. "Silicon photonics interfaced with integrated electronics for 9 GHz measurement of squeezed light." Nature Photonics 15.1 (2021): 11-15.



Tasker, J. F., Frazer, J., Ferranti, G., & Matthews, J. C. (2024). A Bi-CMOS electronic photonic integrated circuit quantum light detector. Science Advances, 10(20), eadk6890.

Applications:

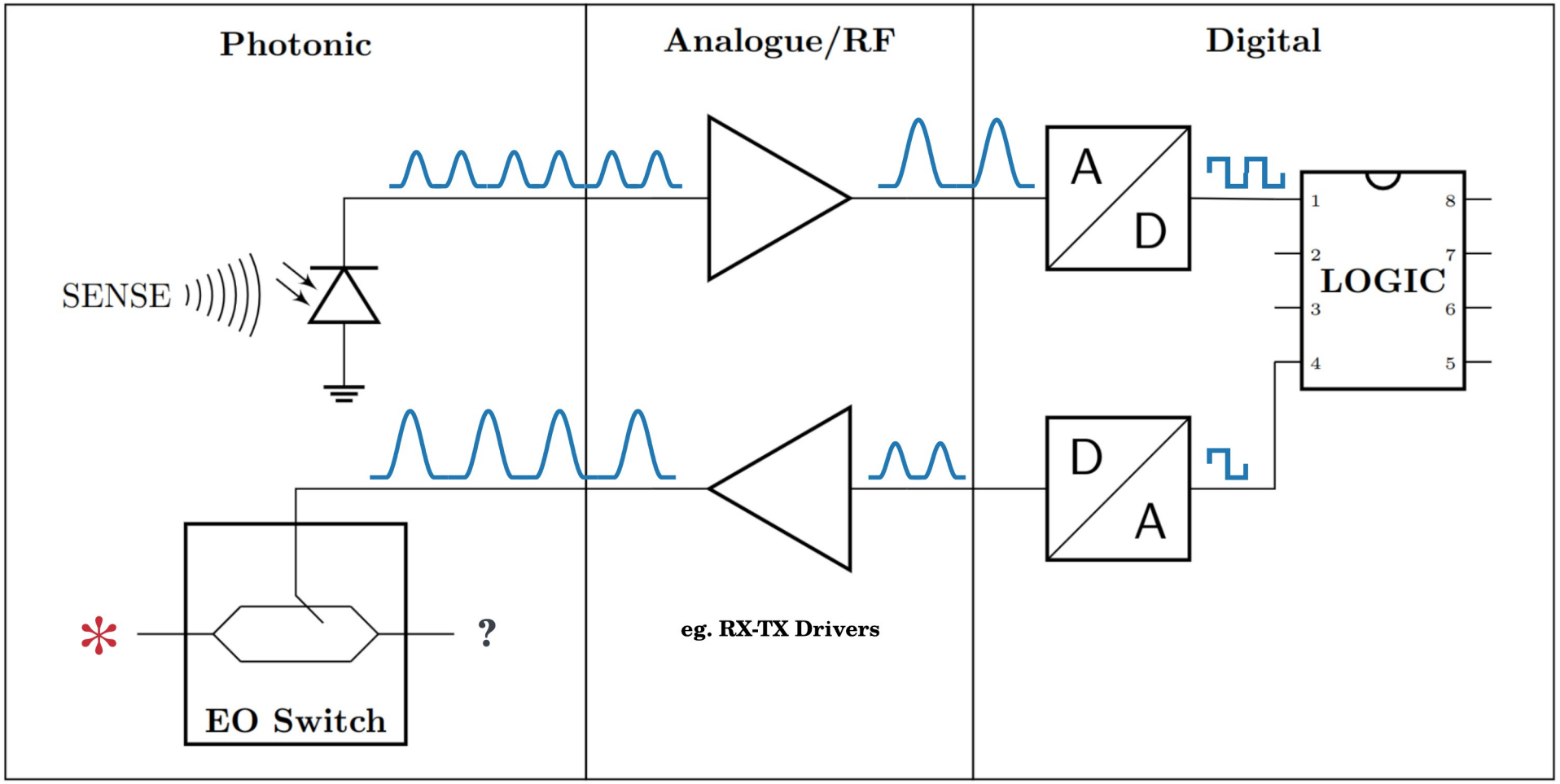
- Quantum Photonics/Computation (my research)
- Telecommunications / HPC
- Sensing / LIDAR / Automotive
- Photonic / Neuromorphic Computation

Challenges:

- Co-simulation feedback complexity
- Co-design tools are not well integrated
- One domain design decision could break the whole system.

University of BRISTOL

A Fully Concurrent Electronic-Photonic System



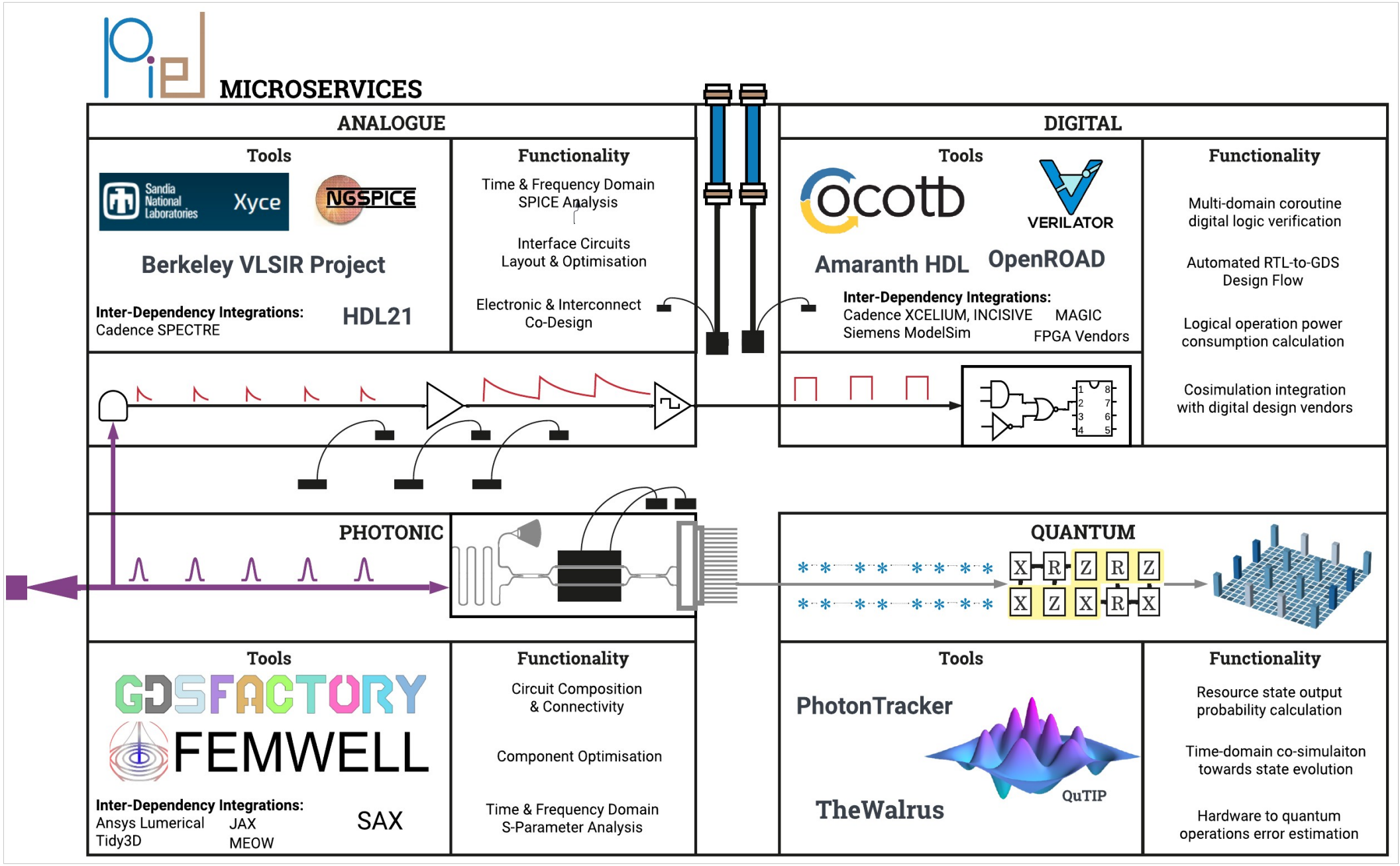
QET Labs

University of BRISTOL

The *piel* Toolchain

Project Status:

- Very Alpha
- Very Active



QET Labs

Dependency Management

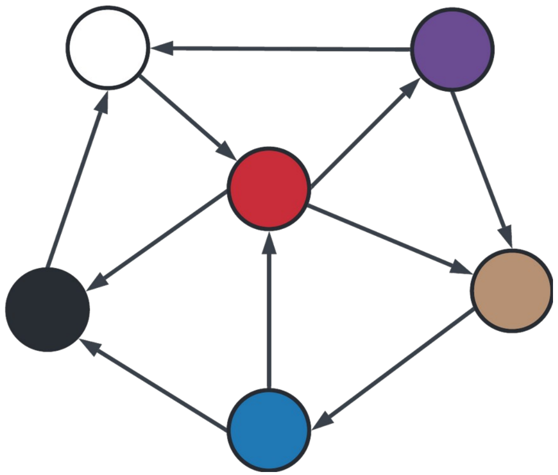
Extending:

efabless/nix-eda
efabless/openlane2

Poetry2nix
(piel poetry.lock)

Thanks Donn!

```
2024-06-20 10:44:14 ⌚ dxps in ~/phd/piel
± |further_full_demo S:1 U:3 ?:1 ✗| → ls
CONTRIBUTING.rst  flake.lock  MANIFEST.in   poetry.lock     ruff.toml
docs              flake.nix   openlane_run  pyproject.toml  tests
environment       LICENSE     piel          README.md
```



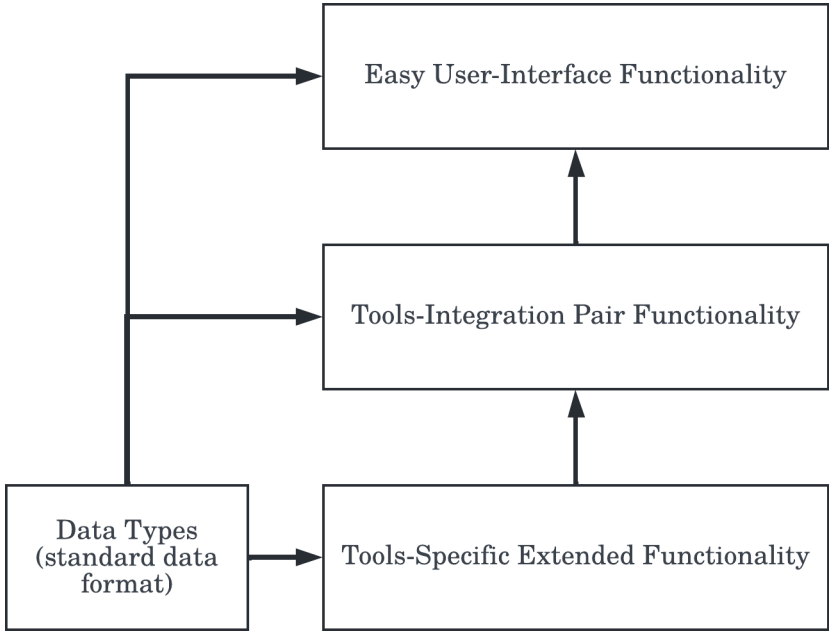
```
2024-06-20 10:40:36 ◷ dxps in ~/phd/piel
± |further_full_demo S:1 U:3 ×| → nix shell . github:efabless/nix-eda#{ngspice,xschem,verilator,yosys} github:efabless/openlane2 nixpkgs#verilog nixpkgs#gtkwa
warning: Git tree '/home/daquintero/phd/piel' is dirty
warning: ignoring the client-specified setting 'trusted-public-keys', because it is a restricted setting and you are not a trusted user
warning: ignoring the client-specified setting 'trusted-public-keys', because it is a restricted setting and you are not a trusted user
warning: ignoring the client-specified setting 'trusted-public-keys', because it is a restricted setting and you are not a trusted user
warning: ignoring the client-specified setting 'trusted-public-keys', because it is a restricted setting and you are not a trusted user
warning: ignoring the client-specified setting 'trusted-public-keys', because it is a restricted setting and you are not a trusted user
warning: ignoring the client-specified setting 'trusted-public-keys', because it is a restricted setting and you are not a trusted user
warning: ignoring the client-specified setting 'trusted-public-keys', because it is a restricted setting and you are not a trusted user

2024-06-20 10:41:36 ◷ dxps in ~/phd/piel
± |further_full_demo S:1 U:3 ×| → pyenv which piel
/nix/store/d7iy49gi0zncvnkl71wx0sr7q1b56hls-python3.11-piel-0.0.56/bin/piel

2024-06-20 10:41:42 ◷ dxps in ~/phd/piel
± |further_full_demo S:1 U:3 ×| → pyenv which openlane
/nix/store/9jb8wsk32ny2yy5ghcaq3y7mbmmavi2c-python3.11-openlane/bin/openlane
```

University of BRISTOL

Co-Design Project Structure

Compatibility Between Toolsets

- Hierarchy Management
- Encoded Directory Operation Mapping



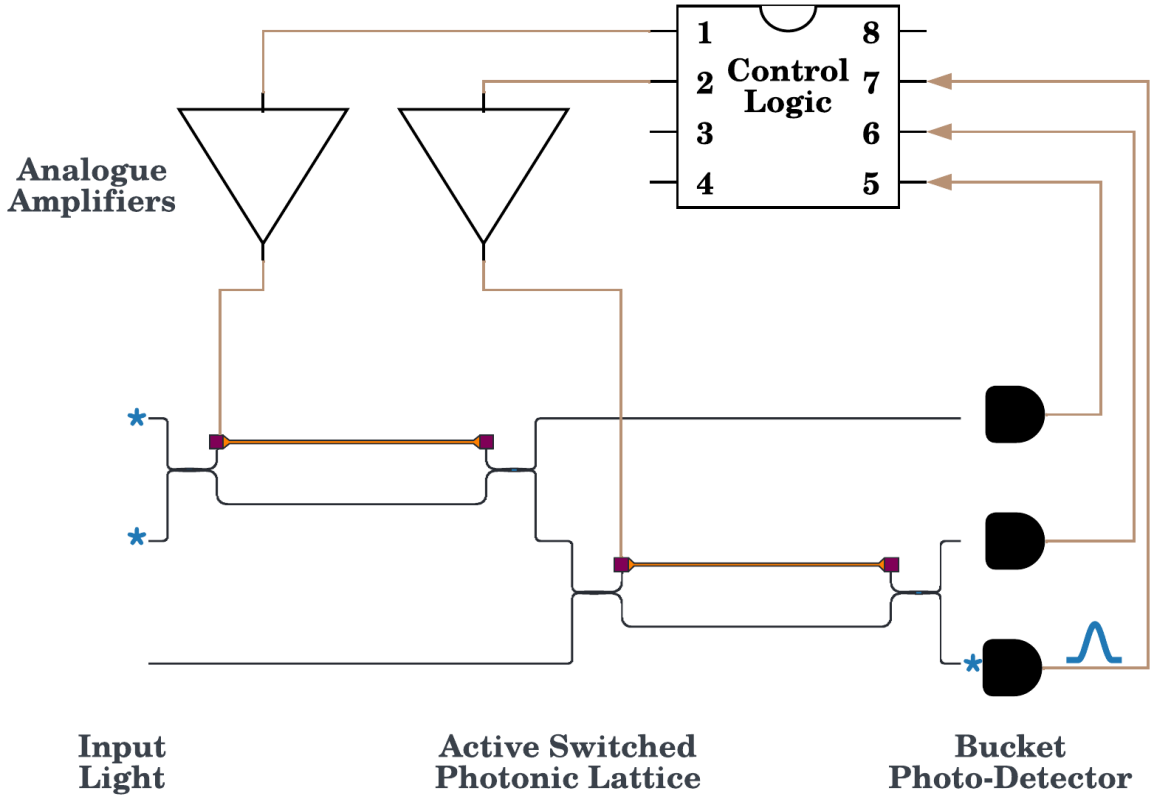

```
design_folder_name
    docs/
        README.md # Optional: Document your design
    design_folder_name/
        __init__.py
        analogue/
            __init__.py # Analog GDSFactory top-level layout files in here.
        components/
            __init__.py # Optional: Custom layout `gdsfactory` components in here.
            photonics/
                __init__.py # `gdsfactory` components in here.
            analogue/
                __init__.py # `gdsfactory` custom analogue layout in here.
            digital/
                __init__.py # Any digital macros part of the OpenLane, `amaranth`, `cocotb`
        io/
            pin_order.cfg # CUSTOM REQUIRED: OpenLane
        models/
            __init__.py # Optional: `gplugins` models in here.
            analogue/
                __init__.py # `hdl21 SPICE` models too.
            frequency/
                __init__.py # `sax` and related photonics models here.
            logic/
                __init__.py # `digital` behavioural models here.
            physical/
                __init__.py # `femwell`-generated models here on particular components.
            transient/
                __init__.py # Photonic and specific behavioural analogue time-domain models
        photonic/
            __init__.py # Photonic GDSFactory top-level layout files in here.
        scripts/
            openlane.sh # eg. generated by piel for automation with script based tools.
        sdc/
            design.sdc # Required: OpenLane SDC timing configuration files.
        src/
            source_files.v # Digital flow source files in here.
        runs/
            openlane_run_folder # AUTO GENERATED: OpenLane v1 digital layout outputs.
        tb/
            __init__.py
            Makefile # # AUTO GENERATED: cocotb in here
            test_design.py # # AUTO GENERATED: Required cocotb
            out/
                output.csv # # AUTO GENERATED: Recommended: cocotb simulation output files g
        config.json # Optional, Required only for OpenLane V1
    .gitignore
    setup.py
    README.md
```

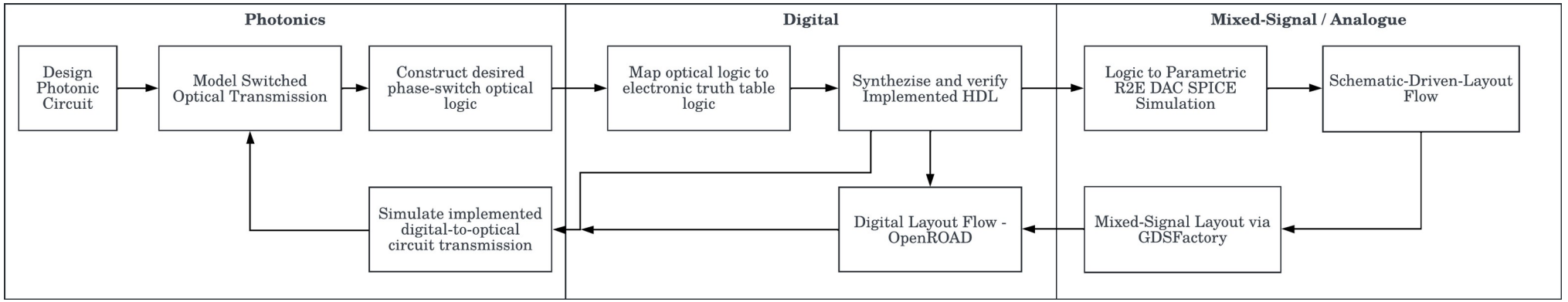
QET Labs

University of BRISTOL

Designing and Simulating

Towards an (incomplete) Full Flow Example



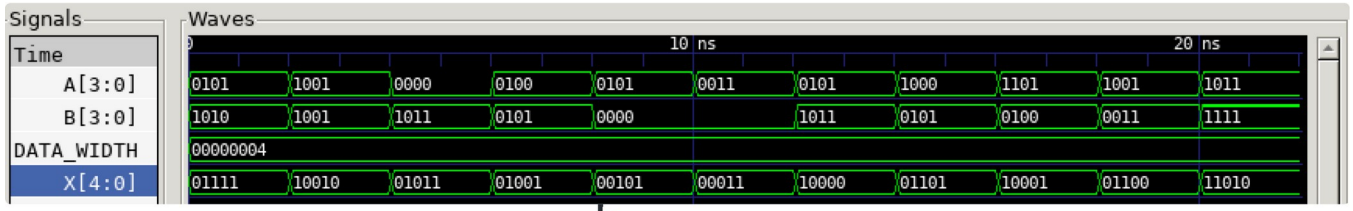


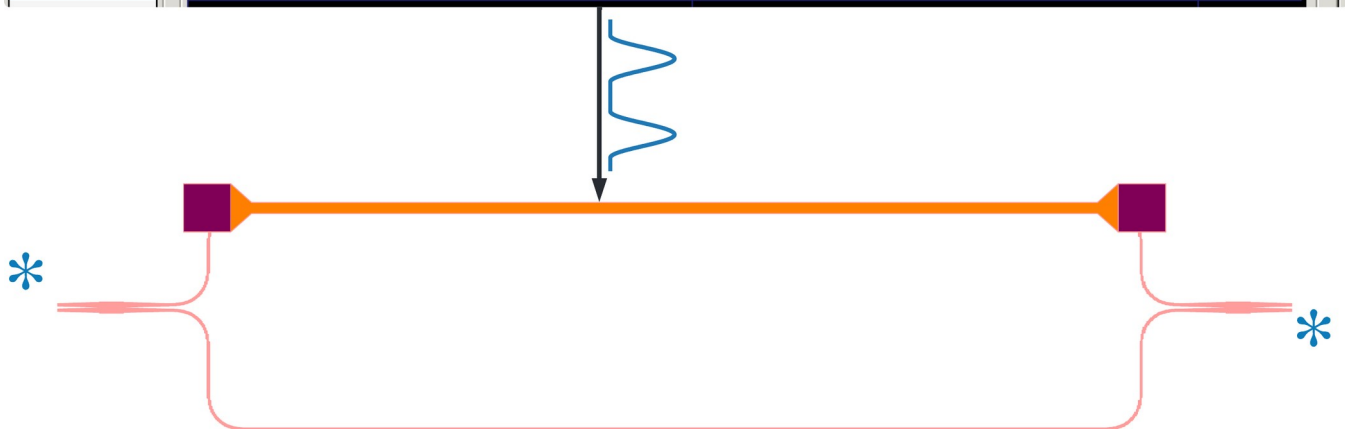
QET Labs

University of BRISTOL

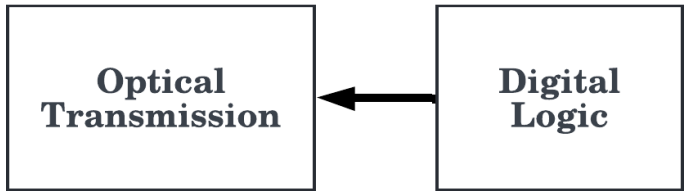
Simulation Capabilities

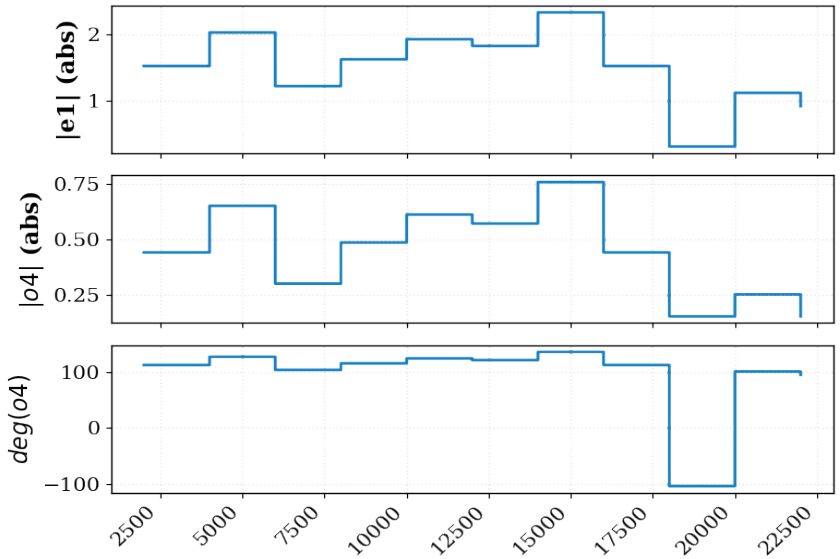
Example 3a

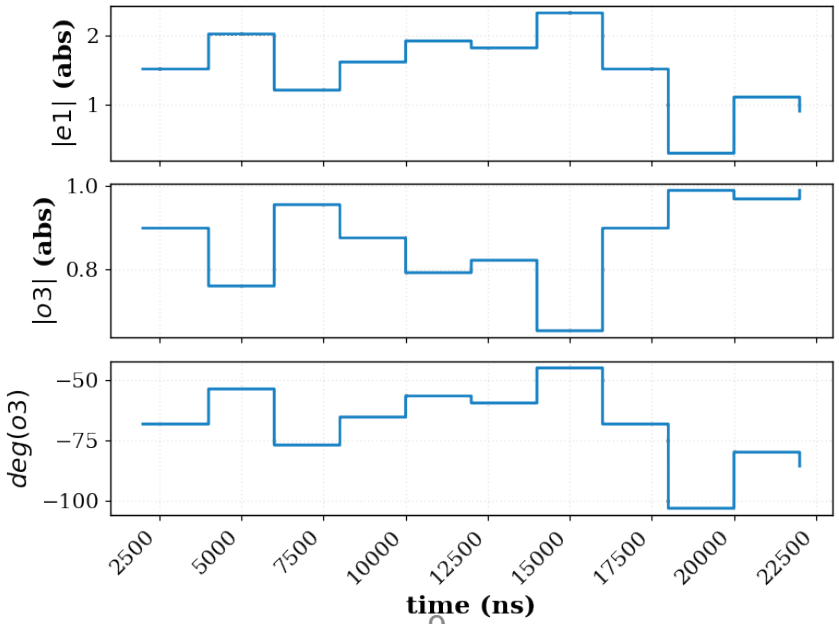




Example 03a



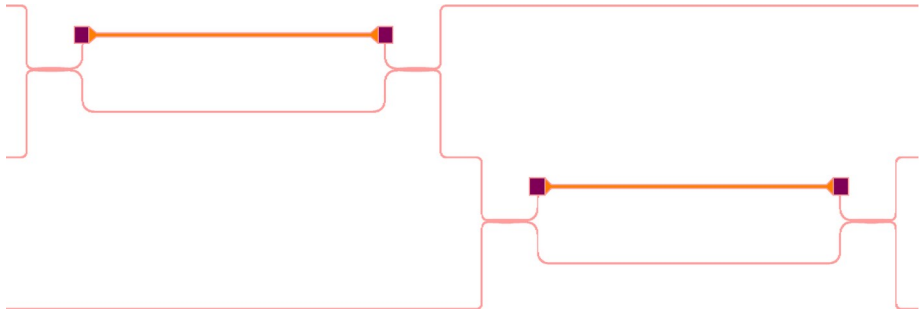




University of BRISTOL

Simulation Capabilities

Example 7



```
chain_fock_state_transitions = piel.flows.get_state_phase_transitions(
    circuit_component=chain_3_mode_lattice_circuit,
    models=optical_logic_verification_models,
    mode_amount=3,
    target_mode_index=2,
)
```

```
chain_fock_state_transitions.transition_dataframe
```

	phase	input_fock_state	output_fock_state	target_mode_output
0	(0, 0)	(1, 0, 0)	(0, 0, 1)	1
1	(0, 0)	(0, 1, 0)	(1, 0, 0)	0
2	(0, 0)	(0, 0, 1)	(0, 1, 0)	0
3	(0, 3.141592653589793)	(1, 0, 0)	(0, 1, 0)	0
4	(0, 3.141592653589793)	(0, 1, 0)	(1, 0, 0)	0
5	(0, 3.141592653589793)	(0, 0, 1)	(0, 0, 1)	1
6	(3.141592653589793, 0)	(1, 0, 0)	(1, 0, 0)	0
7	(3.141592653589793, 0)	(0, 1, 0)	(0, 0, 1)	1
8	(3.141592653589793, 0)	(0, 0, 1)	(0, 1, 0)	0
9	(3.141592653589793, 3.141592653589793)	(1, 0, 0)	(1, 0, 0)	0
10	(3.141592653589793, 3.141592653589793)	(0, 1, 0)	(0, 1, 0)	0
11	(3.141592653589793, 3.141592653589793)	(0, 0, 1)	(0, 0, 1)	1

Example 07 (18/06/24)


```
truth_table = (
    piel.flows.digital_electro_optic.convert_optical_transitions_to_truth_table(
        optical_state_transitions=chain_fock_state_transitions,
        bit_phase_map=basic_ideal_phase_map,
        logic="implementation",
    )
)
truth_table.dataframe
```

1

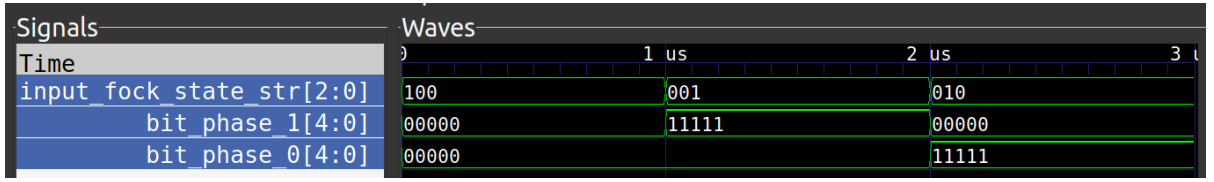
	input_fock_state_str	bit_phase_0	bit_phase_1
0	100	00000	00000
1	001	00000	11111
2	010	11111	00000

University of BRISTOL

Simulation Capabilities

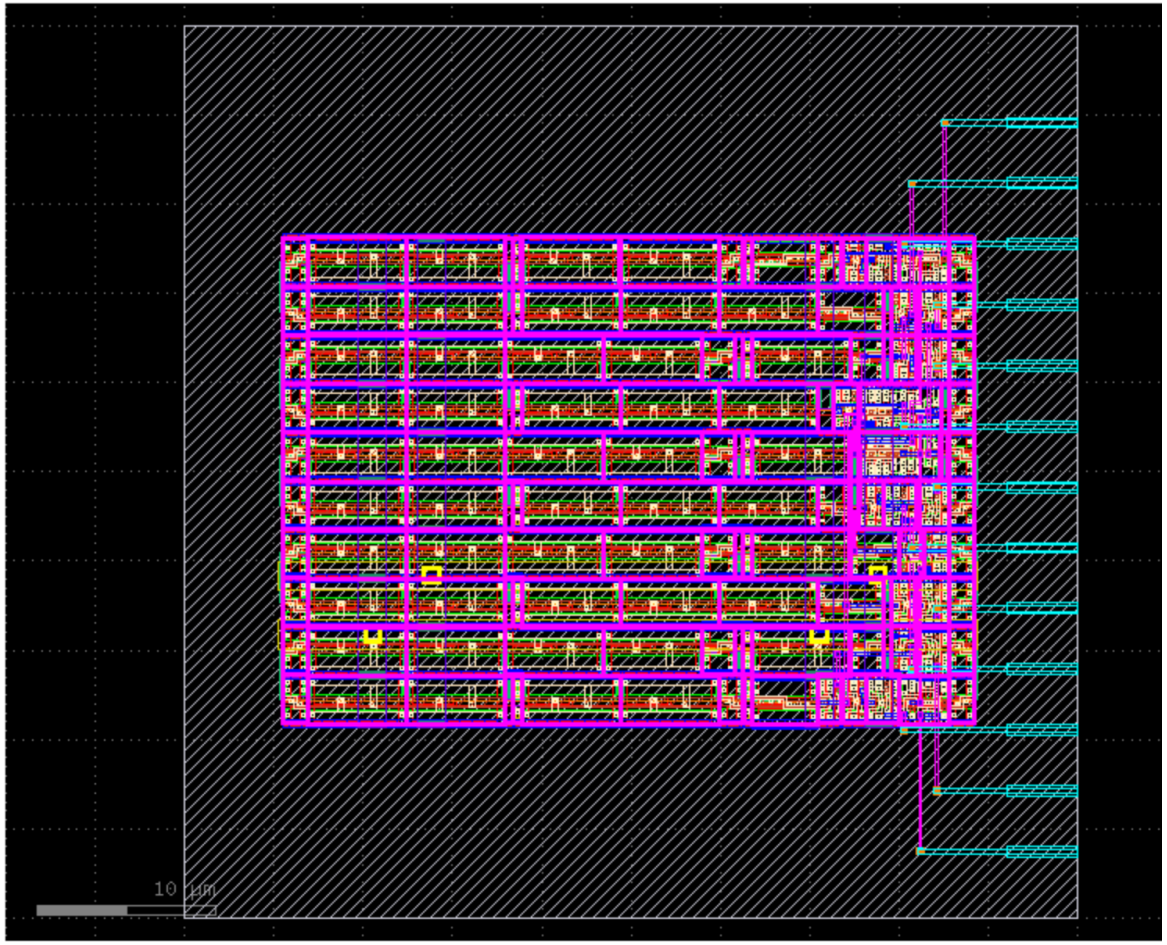
Example 7

```
/* Generated by Amaranth Yosys 0.40 (PyPI ver 0.40.0.0.post95, git sha1 a1bb0255d) */

(* top =  1  *)
(* generator = "Amaranth" *)
module top(bit_phase_0, bit_phase_1, input_fock_state_str);
  reg \$auto$verilog_backend.cc:2352:dump_module$1  = 0;
  (* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:88" *)
  output [4:0] bit_phase_0;
  reg [4:0] bit_phase_0;
  (* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:88" *)
  output [4:0] bit_phase_1;
  reg [4:0] bit_phase_1;
  (* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:84" *)
  input [2:0] input_fock_state_str;
  wire [2:0] input_fock_state_str;
  always @* begin
    if (\$auto$verilog_backend.cc:2352:dump_module$1 ) begin end
    (* full_case = 32'd1 *)
    (* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:114" *)
    casez (input_fock_state_str)
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:117" */
      3'h4:
          bit_phase_0 = 5'h00;
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:117" */
      3'h1:
          bit_phase_0 = 5'h00;
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:117" */
      3'h2:
          bit_phase_0 = 5'h1f;
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:126" */
      default:
          bit_phase_0 = 5'h00;
    endcase
  end
  always @* begin
    if (\$auto$verilog_backend.cc:2352:dump_module$1 ) begin end
    (* full_case = 32'd1 *)
    (* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:114" *)
    casez (input_fock_state_str)
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:117" */
      3'h4:
          bit_phase_1 = 5'h00;
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:117" */
      3'h1:
          bit_phase_1 = 5'h1f;
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:117" */
      3'h2:
          bit_phase_1 = 5'h00;
      /* src = "/home/daquintero/phd/piel/piel/tools/amaranth/construct.py:126" */
      default:
          bit_phase_1 = 5'h00;
    endcase
  end
endmodule
```




```
component = piel.flows.get_latest_digital_run_component(
    module=full_flow_demo,
)
component.plot()
```

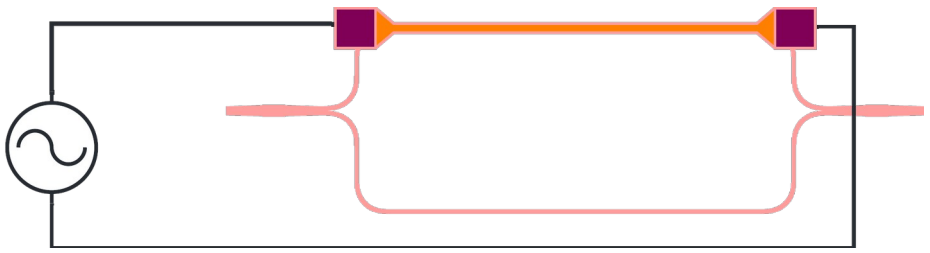
```
Importing this gds: /home/daquintero/phd/piel/docs/examples/07_full_flow_demo_electronic_photonic/full_flow_demo
```

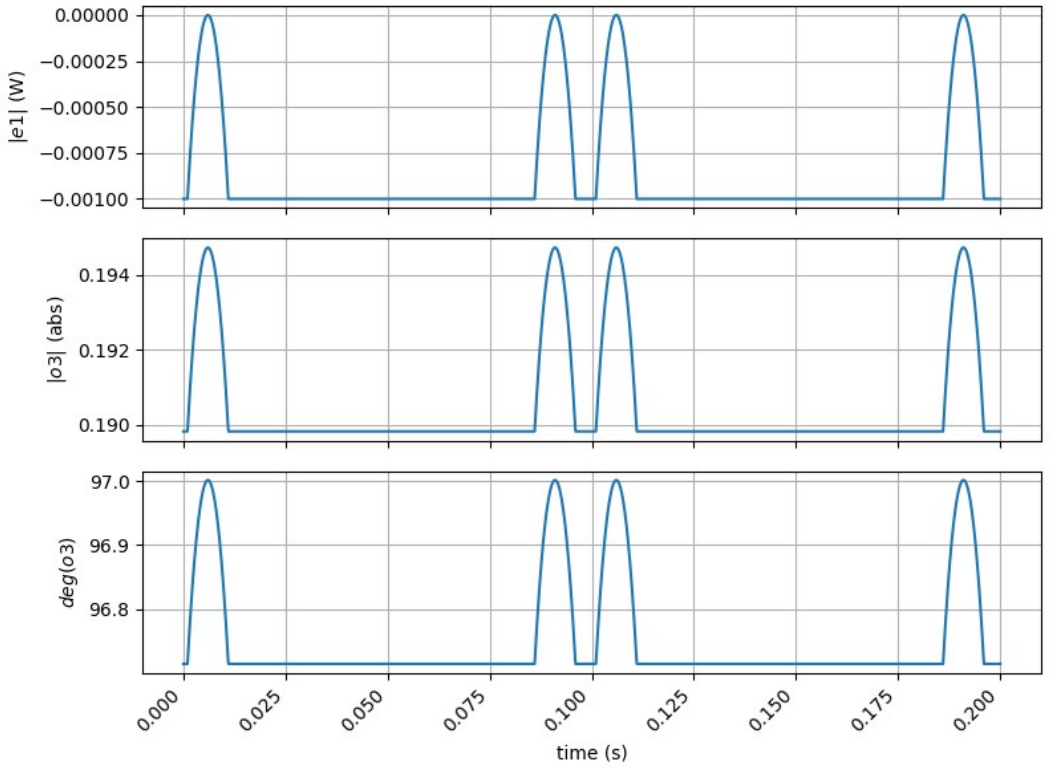


University of BRISTOL

Simulation Capabilities

Example 4





Example 04


```
piel.flows.extract_component_spice_from_netlist(
    component=straight_heater_metal_simple(),
)
```

```
* Anonymous `circuit.Package`
* Generated by `vlsirtools.SpiceNetlister`
*

.SUBCKT Straight
+ e1 e2
* No parameters

rr1
+ e1 e2
+ 1000
* No parameters


.ENDS

.SUBCKT Taper
+ e1 e2
* No parameters

rr1
+ e1 e2
+ 1000
```

Future Plans:

- Steps in the direction of fully-modeling a electronic-photonic system, finish full demo.

- Integrate sky130nm analog models working with the gdsfactory team.

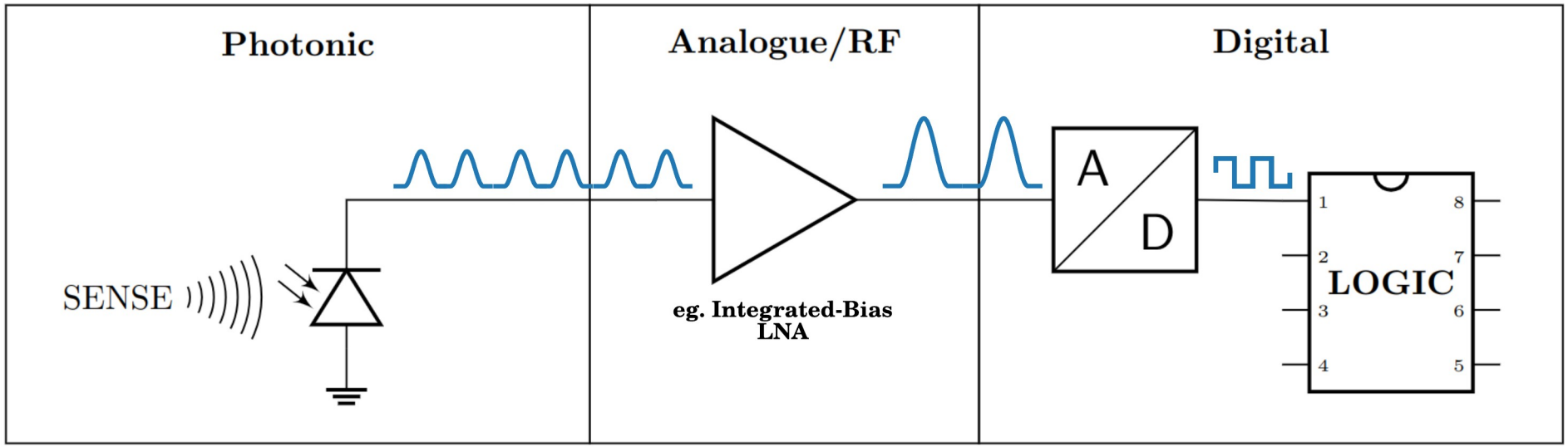
- Fully concurrent simulation: Use the ngspice-42 external-trigger input (ch12.5) to write a subroutine to integrate a sax time-domain simulator with mixed-signal simulation.

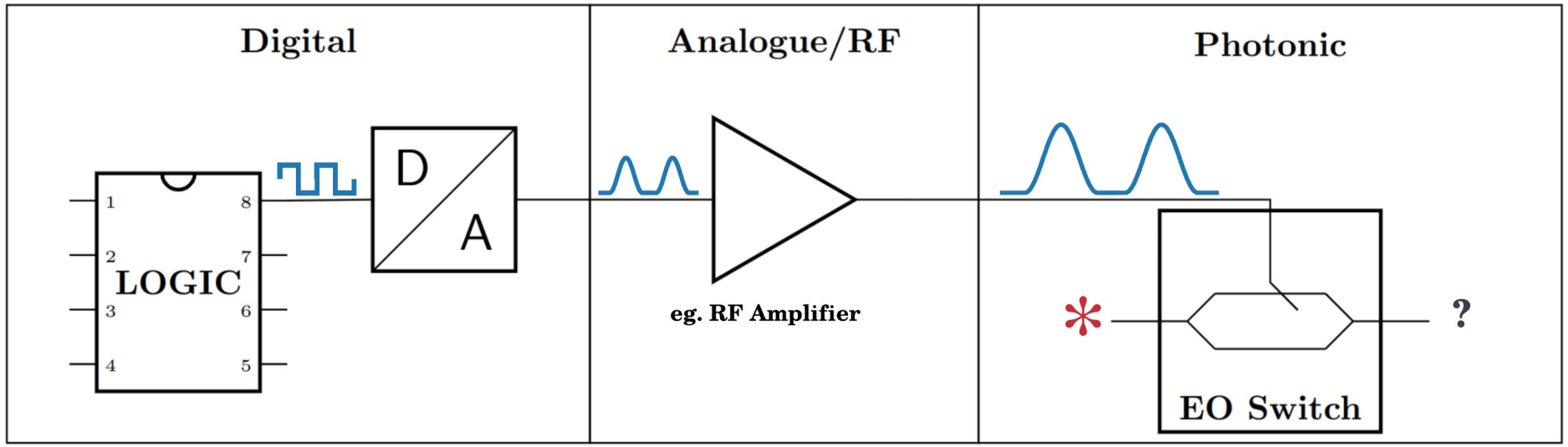
- Finish the demo paper showing a quantum-photonic-electronic system co-design flow.

Especially thanks to :
- Joaquin
- Floris
- Donn
- The gdsfactory/gplugins team and Efabless.

You are the best!

Appendix


Photonic
Analogue/RF
Digital
SENSE
eg. Integrated-Bias LNA
A
D
LOGIC
1
2
3
4
5
6
7
8


Digital
Analogue/RF
Photonic
LOGIC
D
A
eg. RF Amplifier
EO Switch
?
1
2
3
4
5
6
7
8