GateMate FPGA Synthesis with Third-Party Tools

FPGA-Conference Europe

Patrick Urban
Cologne Chip AG

July 7th, 2021

Meet us during the conference

Feel free to arrange a personal meeting with our engineers in our virtual room during the conference from July 6th - 8th 2021!

- Use the link to our virtual room or contact us via e-mail: info@colognechip.com
- Presentation slides and additional material in content library
- If you see this note after the conference, feel free to contact us at any time

Agenda

1. Introduction to GateMate FPGA

   Features, Architecture Overview

2. FPGA Design Flow

   File formats, Toolchain

3. Synthesis Tools

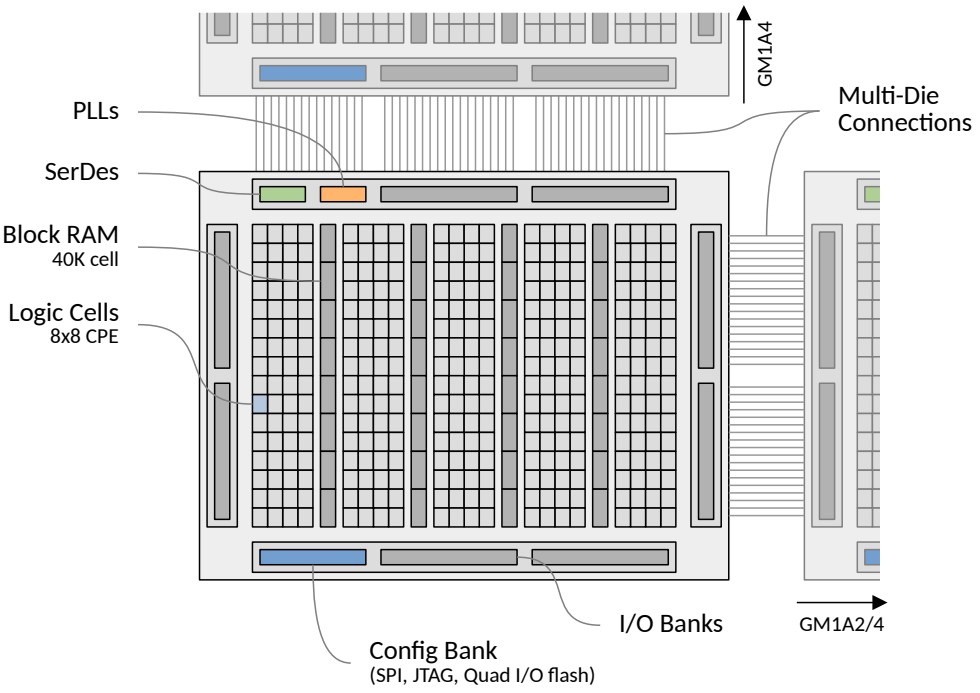
   Primitives Library

4. Outlook

   Roadmap, LUT-tree mapping

GateMate FPGA
Features at a glance

- Globalfoundries™ 28 nm SLP process
- 20,480 programmable elements
  - 1x8 inputs or 2x4 independent inputs
  - 1- or 2-bit full adder or 2x2-bit multiplier
  - 2 Flip-flops
- All 162 GPIO configurable as single-ended or LVDS differential pairs with DDR support
- 32 40Kbit RAM cells (Total 1,280 Kbit)
  - 1x40K or 2x20K cells, up to 80 (SDP) / 40 (TDP) inputs
  - FIFO mode
- 4 Clock Generators (PLL)
- 2.5 Gb/s SerDes Controller
- Core voltage from 0.9V to 1.1V
- A1, A2+A4: 324-ball FPGA package (15x15 mm)



GateMate FPGA Series

Feature Summary by Device

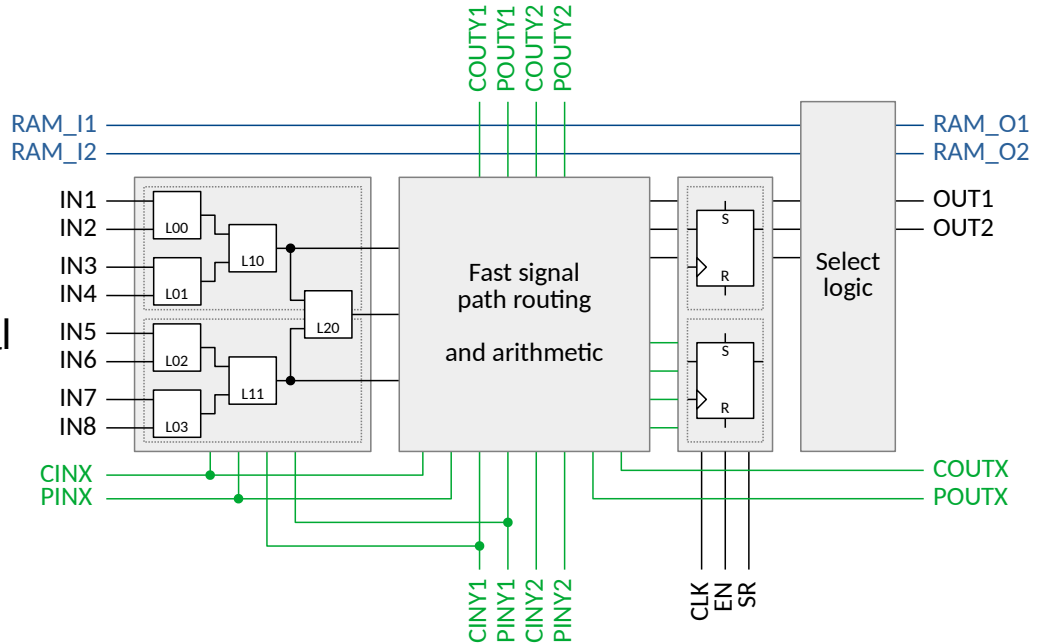
Device	Size	CPEs	FFs	Block RAM		PLLs	SerDes	GPIO		Package
				20K	40K			Single-ended	Diff. Pairs (LVDS)	
CCGM1A1	1	20,480	40,960	64	32	4	1	162	81	324 FBGA 15x15 mm
CCGM1A2	2	40,960	81,920	128	64	8	2	162	81	324 FBGA 15x15 mm
CCGM1A4	4	81,920	163,840	256	128	16	4	162	81	324 FBGA 15x15 mm
...										
CCGM1A25	25	512,000	1,240,000	1600	800	100	25	tba	tba	tba

GateMate FPGA
Central Programming Element

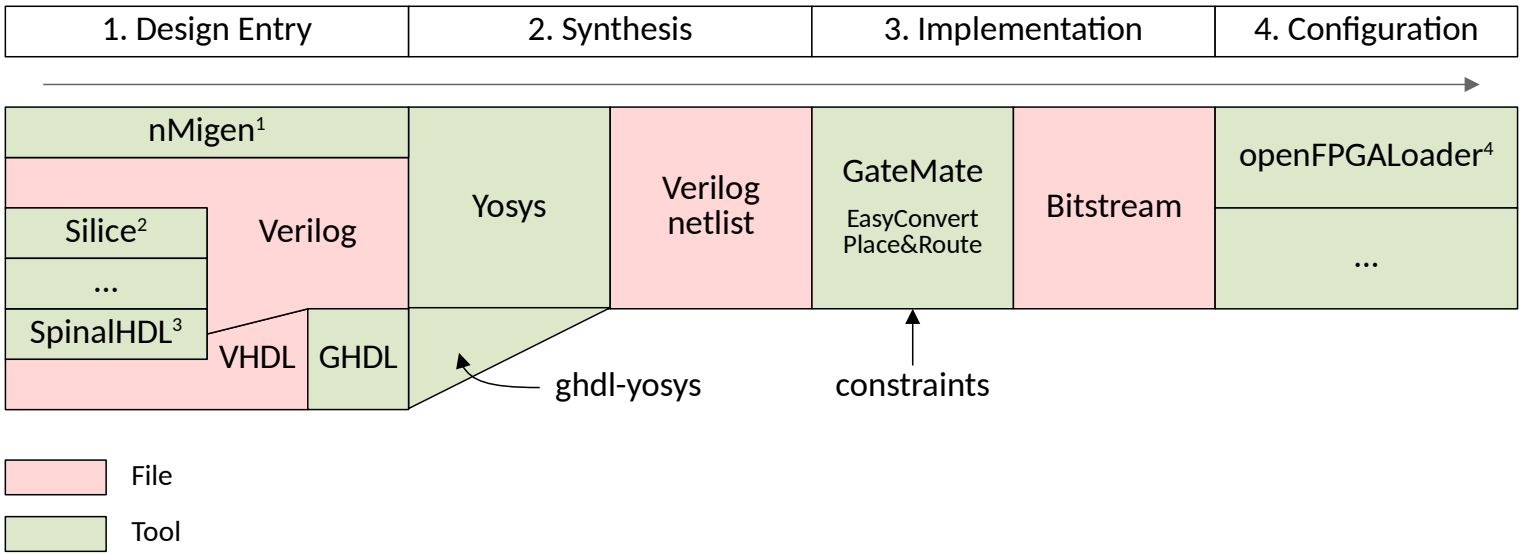
Combinatorical
- 8-input function with LUT2-tree
- 2 independent 4-input functions
- 6 inputs for MUX-4 function

Arithmetic
- 1-bit or 2-bit full adder, horizontal or vertical
- 2x2-bit multiplier, expandable to any size

Sequential
- 2 Flip-flops or latches
- 8+3 inputs for MUX-8 function



FPGA Design Flow

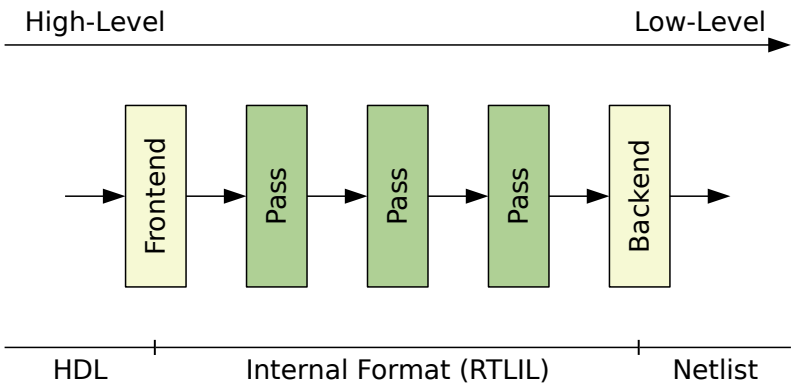


[1] https://github.com/nmigen/nmigen
[2] https://github.com/sylefeb/Silice
[3] https://github.com/SpinalHDL/SpinalHDL
[4] https://github.com/trabucayre/openFPGALoader

Synthesis Tools
Overview

Tool research

- Leonardo Spectrum (Saros Technology)
- Genus (Cadence)
- Synthesis based on »Liberty Files«
  - IEEE Standard for defining standard cells
  - Limitation: No IO buffer insertion
  - Limitation: No multiplier or block RAM inference
- Yosys offers more flexibility in form of »passes«
- First step: Primitive Specification



[1]

[1] http://www.clifford.at/yosys/files/yosys_presentation.pdf, slide 9

Primitives Library
Overview

IO Components

- CC_IBUF / OBUF / TOBUF
- CC_IOBUF
- CC_LVDS_IBUF / OBUF / TOBUF
- CC_LVDS_IOBUF
- CC_IDDR / ODDR

CPE Components

- CC_LUT1/2/3/4
- CC_MX2/4/8
- CC_DFF / CC_DLT

Arithmetic Components

- CC_ADDF
- CC_MULT

Block RAM Components

- CC_BRAM_20K / 40K
- CC_FIFO_40K

Other Components

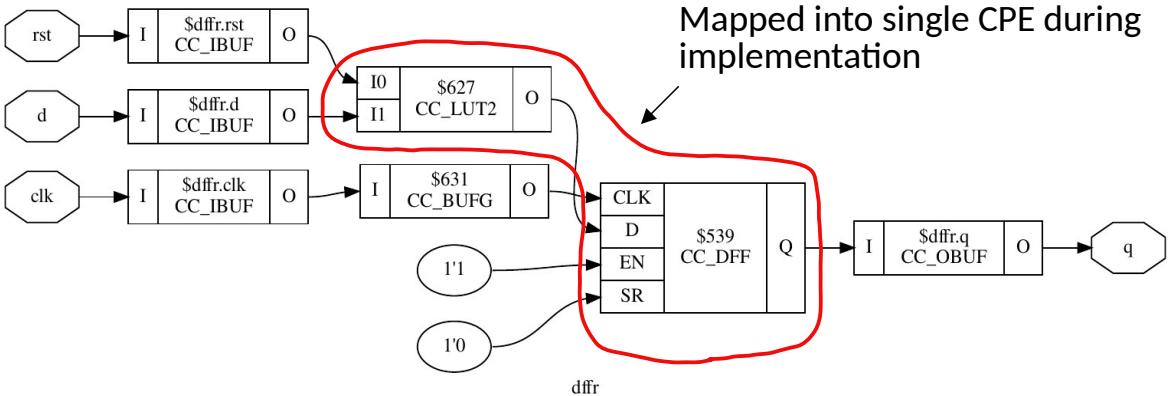
- CC_PLL
- CC_SERDES
- CC_CFG_CTRL

Netlist with Primitves
Technology-Mapping

(I) Design entry before Technology-Mapping

```
module dffr(
      input clk, rst, d,
      output reg q);

      always @(posedge clk) begin
            if (!rst)
                  q <= 1'b0;
            else
                  q <= d;
      end
endmodule
```

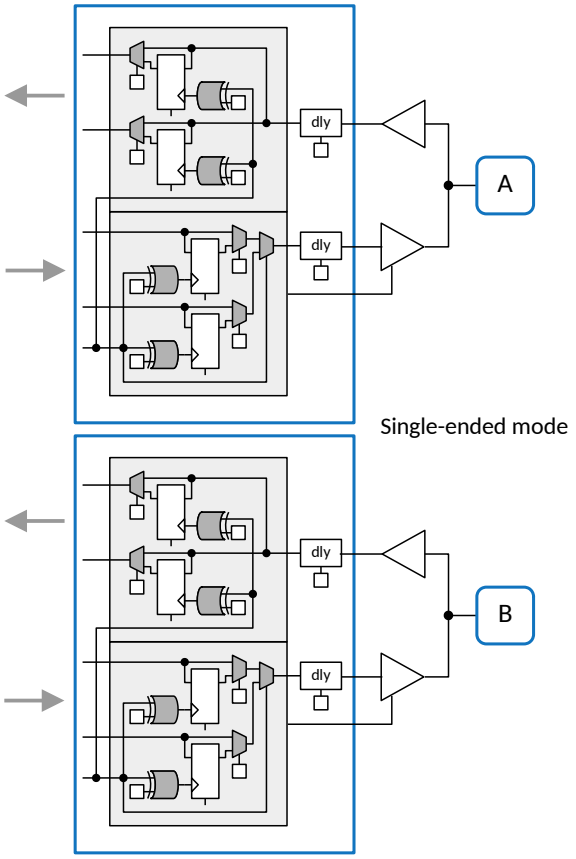
(II) Netlist after Technology-Mapping



Generated with yosys -p 'read -sv dffr.v; synth_gatemate; show'

Primitives Library
IO Cells

- Single-ended IO cells are automatically mapped during synthesis in yosys
- Manual instantiation allows parameterization of
  - IO delays
  - Pin placement
  - Drive strengths
  - Slew rate control
  - Pullup/pulldown configuration
  - Mapping of registers into IO cells
- LVDS and DDR primitves may be instatiated manually



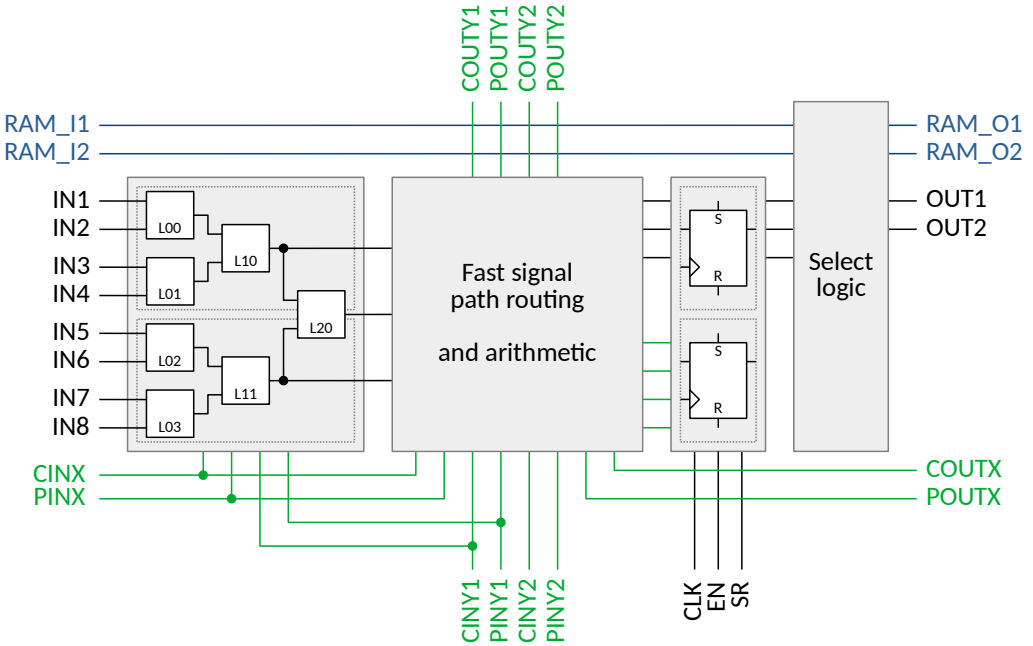
Primitives Library
CPE Cells

Combinatorical

- LUT2-tree is a subset of a LUT4
- We use »ABC« to map LUT4 during synthesis and re-map to LUT2-tree in implementation
- Extract MUX2/4/8 before LUT-mapping

Sequential

- D-type Flip-flops have asynchronous reset
- Add additional LUT for synchronous reset



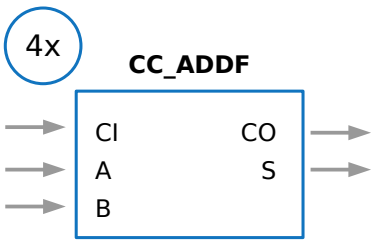
Primitives Library
Arithmetic Cells

- Arithmetic functions such as multipliers or adders are automatically inferred during synthesis

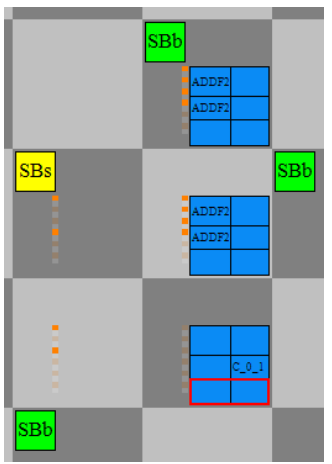
(I) Design Entry

```
module adder(
       input wire [3:0] in1,
       input wire [3:0] in2,
       output [3:0] out
);

       assign out = in1 + in2;

endmodule
```

(II) Synthesis Netlist



(III) Implementation

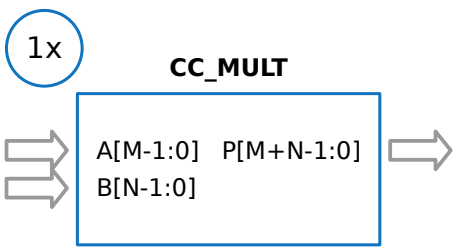


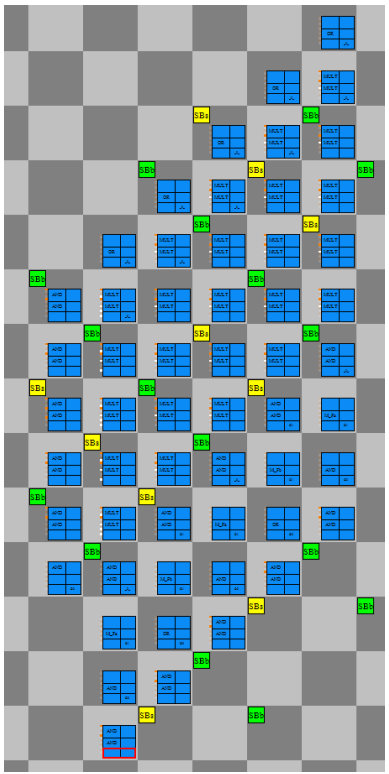
Primitives Library
Multiplier Cell

- Common DSP-slices are rare, have fixed positions and limited size
- CPEs may be used to build arbitrary-sized multipliers
- In- and output pipelining registers can be integrated

- Parameterized inputs and outputs in the CC_PLL primitive
- Automatically infered and mapped during synthesis with yosys
- Place&Route automatically builds CPE-array

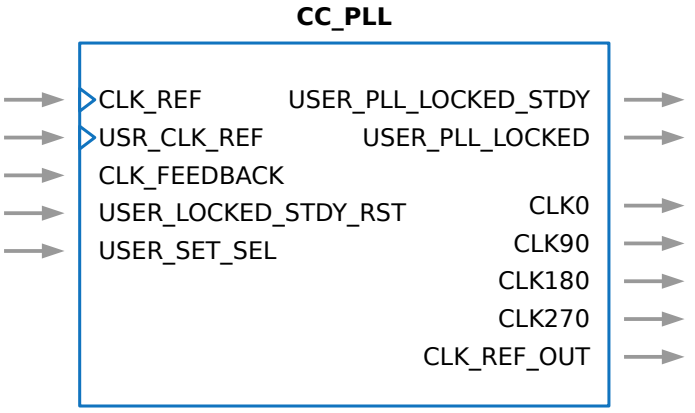
```
wire [M-1:0] a;
wire [N-1:0] b;
reg [M+N-1:0] p;

// Plain M*N multiplier
always @(*)
begin
      p <= a * b;
end
```





Primitives Library
PLL Cell

- Each die has 4 built-in clock generators
- No external tool for primitive generation required
- Parameter setting configures PLL
  - Matching configuration is generated during P&R

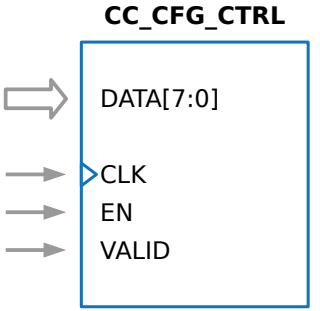



```
CC_PLL #(
       .REF_CLK("10.0"),
       .OUT_CLK("50.0"),
       .PERF_MD("ECONOMY"), // LOWPOWER, ECONOMY, SPEED
       .LOW_JITTER(1),      // 0: disable, 1: enable low jitter mode
       .CI_FILTER_CONST(2), // optional CI filter constant
       .CP_FILTER_CONST(4)  // optional CP filter constant
) pll_inst (
       .CLK_REF(clk), .CLK_FEEDBACK(1'b0), .USER_CLK_REF(1'b0),
       .USER_LOCKED_STDY_RST(1'b0), .USER_SET_SEL(1'b0),
       .USER_PLL_LOCKED_STDY(usr_pll_lock_stdy),
       .USER_PLL_LOCKED(usr_pll_lock),
       .CLK0(clk0)
);
```

Primitives Library
Other Cells (Blackbox)

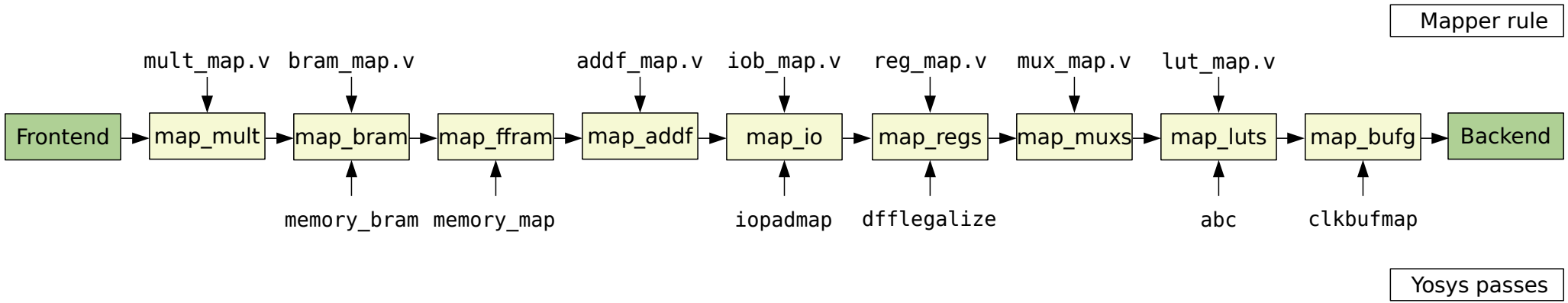
→ Blackbox cells are looped directly into the final netlist

- SerDes Controller Primitive
- (Re-) Configuration Interface
  - Acces to configuration controller from CPE array
  - e.g. to build wrappers for alternative configuration interfaces



Yosys Synthesis Flow

»synth_gatemate« subpasses



Outlook

Yosys

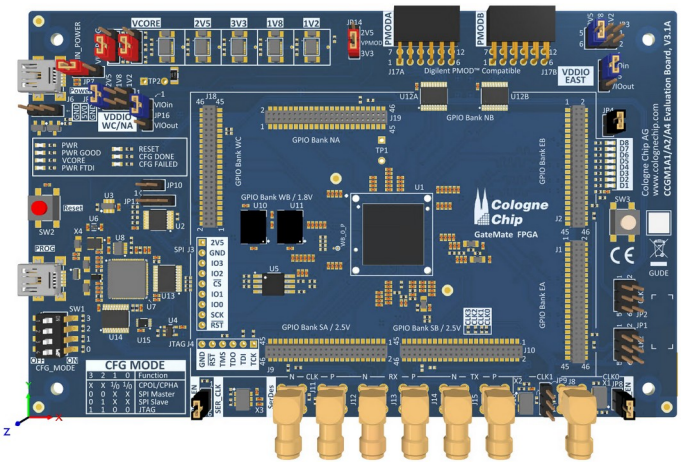
- Pull request for GateMate »techlib« in Yosys[1]
- Investigations for LUT-tree mapping required

Hardware

- Evaluation Boards and FPGAs will be available soon
- Pull request pending for symbol and footprint in KiCad[2]

Further plans

- Device support in nextpnr[3]



[1] https://github.com/colognechip/yosys
[2] https://github.com/pointhi/kicad-footprint-generator/pull/611
[3] https://github.com/YosysHQ/nextpnr

Contact

Patrick Urban
Cologne Chip AG

Eintrachtstr. 113
50668 Köln

https://colognechip.com
info@colognechip.com

References

- Cologne Chip GitHub

  https://github.com/colognechip

- Primitives Library Documentation

  https://colognechip.com/docs/ds1008-gatemate1-primitives-library-latest.pdf

- Yosys Headquarters

  https://github.com/YosysHQ