The FABulous Open Source eFPGA Framework

Dirk Koch

Heidelberg University, The University of Manchester

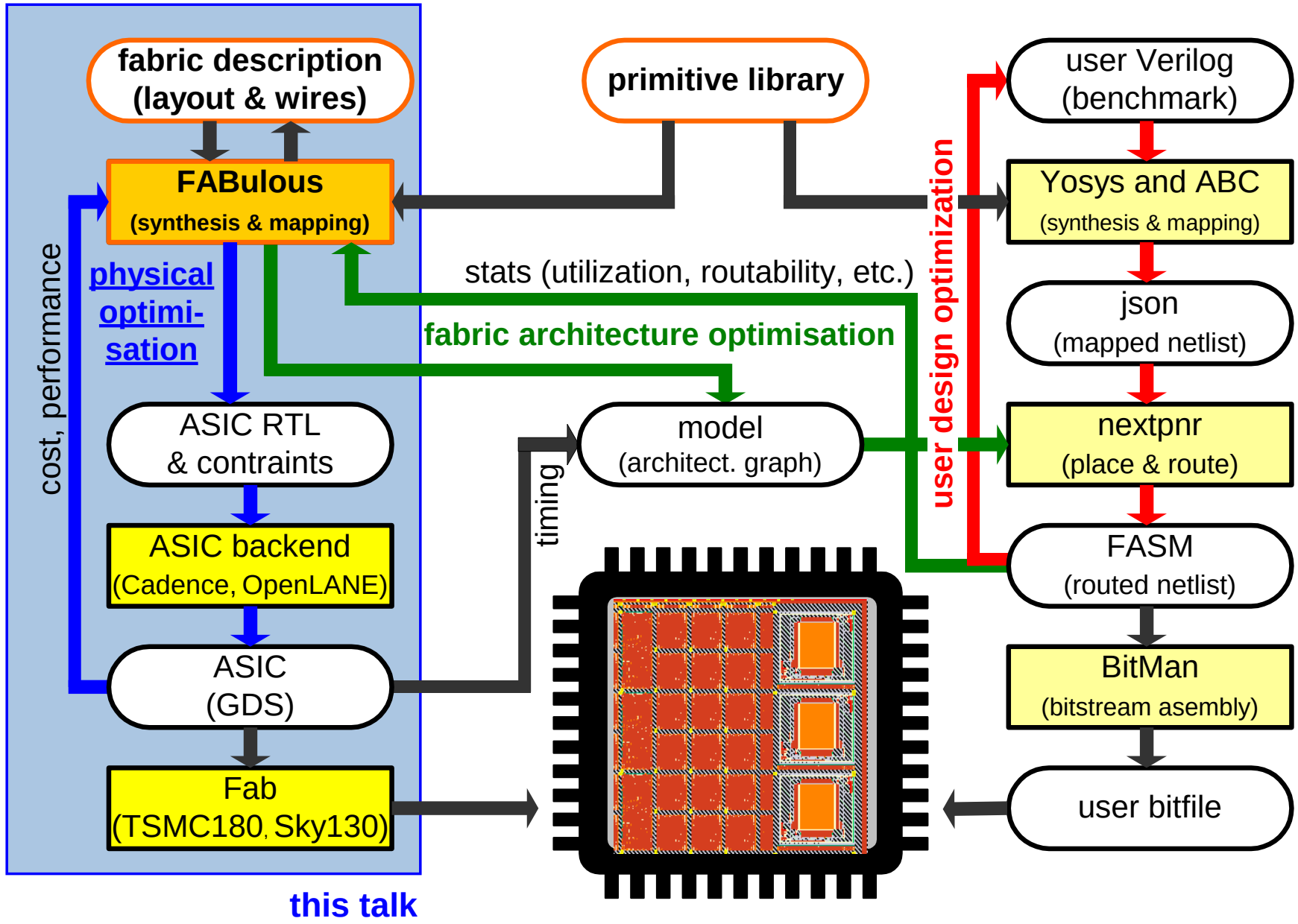
dirk.koch@ziti.uni-heidelberg.de

The FABulous Framework

- Fully integrated framework for eFPGAs

Uses many projects:

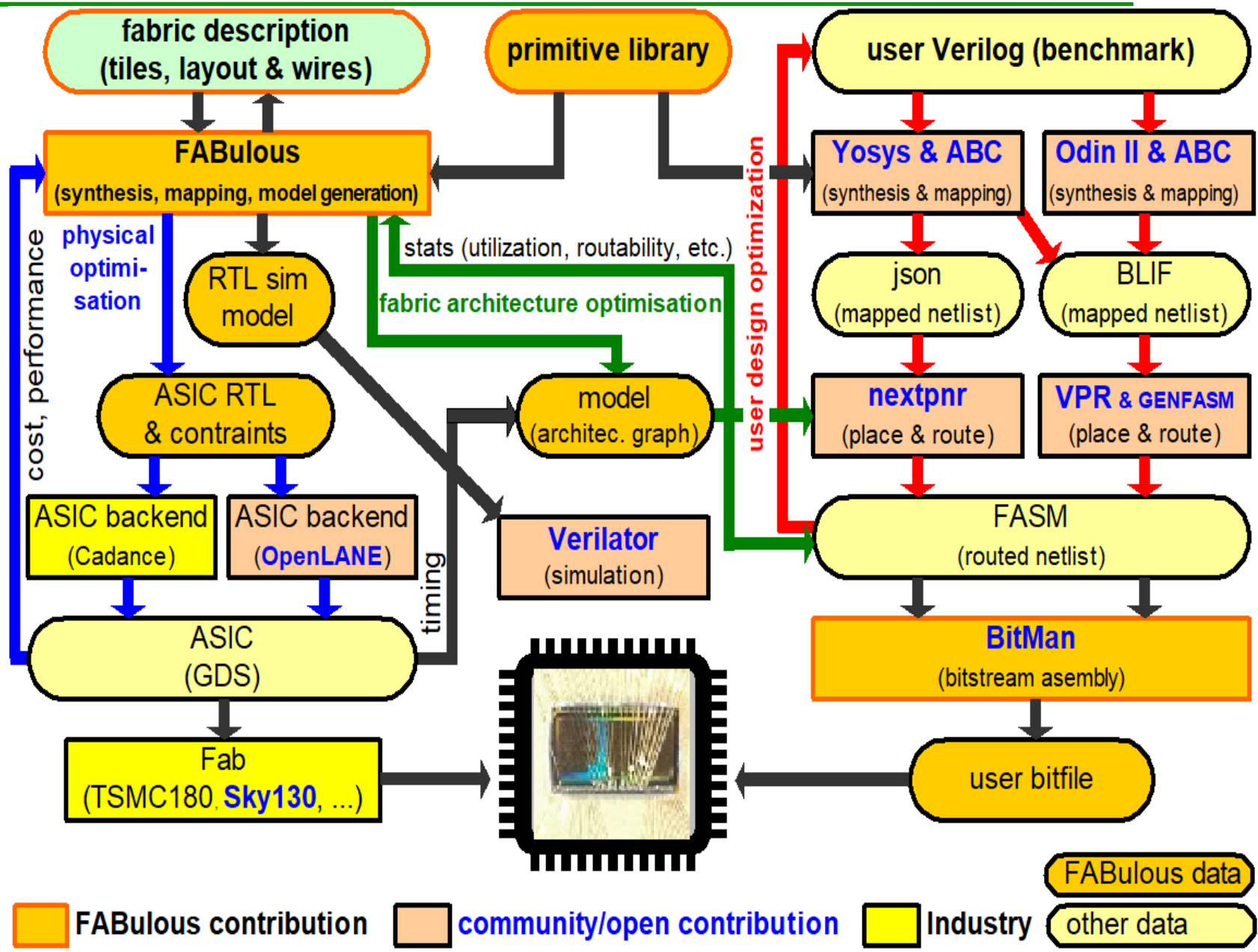
- Yosys & ABC
- nextpnr
- OpenLANE
- VPR
- OpenRAM
- Verilator



The FABulous Framework

- FABulous eFPGA generator
  - ASIC RTL and constraints generation
  - Generating models for nextpnpr/VPR flows
  - FPGA emulation
- Virtex-II, Lattice clones (patent-free!)

- See our FPGA 2021 paper „FABulous: An Embedded FPGA Framework”



The first open-everything FPGA

- Built using open tools

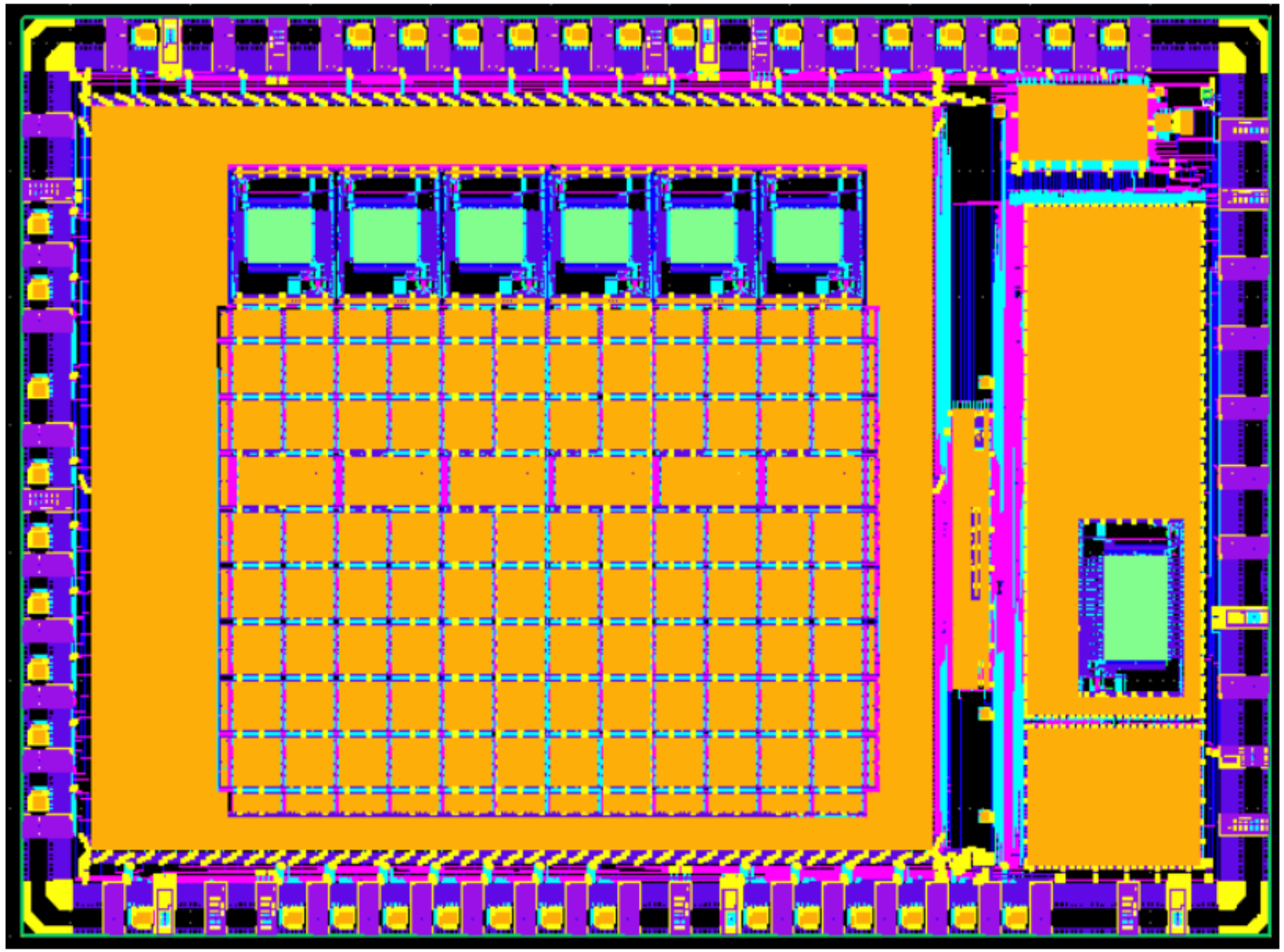
  (Yosys, OpenLane, Verilator…)

- Open PDK

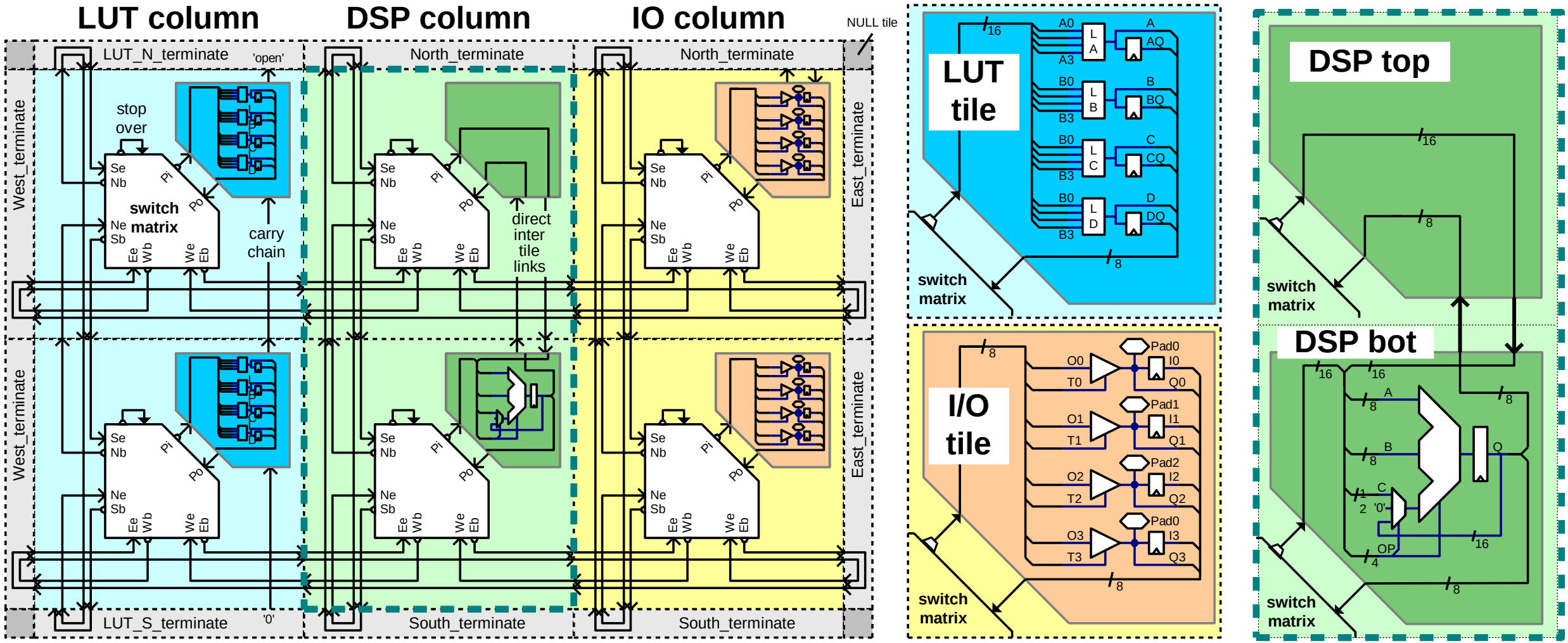
  (Skywater 130 process)

- Google Shuttle (MPW5):

  https://github.com/nguyendao-uom/open_eFPGA

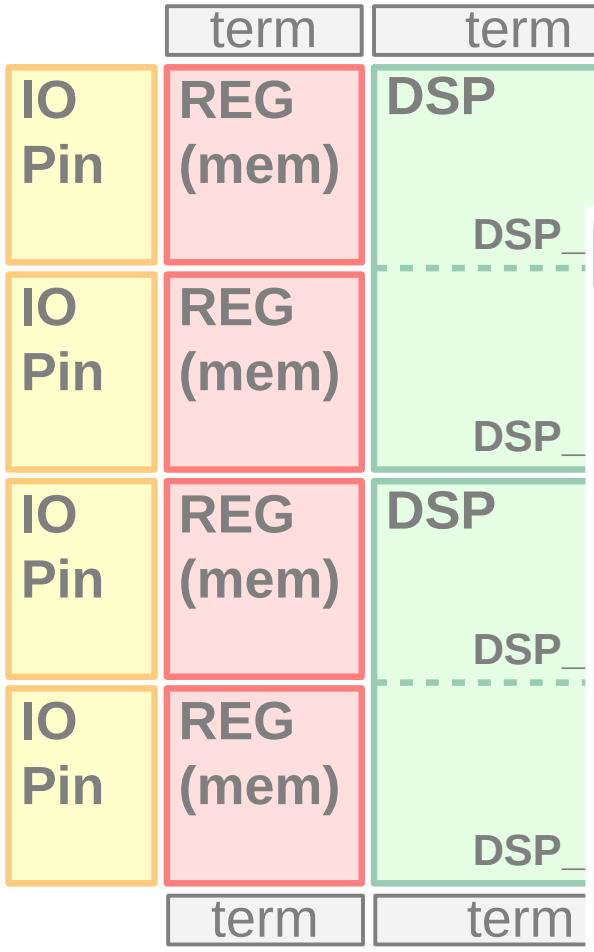


Basic concepts



- Basic tiles have same height, but type-specific width (for logic tiles, DSPs, etc.)
- Adjacent tiles can be fused for more complex blocks (see the DSP example)

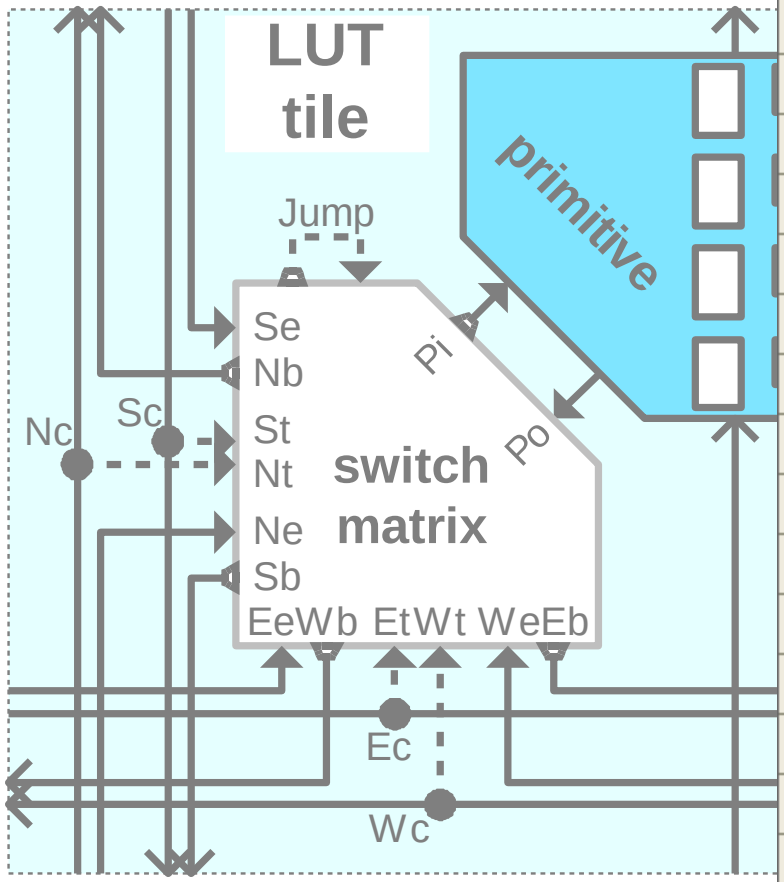
Let‘s build a small eFPGA: Fabric Definition



	A	B	C	D	E	F
1	*FabricBegin*					
2	NULL	N_term	N_term	N_term	N_term	NULL
3	W_IO	RegFile	DSP_top	LUT4AB	LUT4AB	CPU_IO
4	W_IO	RegFile	DSP_bot	LUT4AB	LUT4AB	CPU_IO
5	W_IO	RegFile	DSP_top	LUT4AB	LUT4AB	CPU_IO
6	W_IO	RegFile	DSP_bot	LUT4AB	LUT4AB	CPU_IO
7	NULL	S_term	S_term	S_term	S_term	NULL
8	*FabricEnd*					

- 4 x register file, 2 x DSPs, 8 x LUT-tiles (CLB), I/Os left and right,
- A fabric is modelled as a spreadsheet (tiles are references to tile descriptors)

Let‘s build a small eFPGA: Tile Definition



121	*TILE*	LUT4AB					
122	#direction	source	X-offset	Y-offset	destination	wires	
123	NORTH	N1BEG	0	1	N1END	8	
124	NORTH	N2BEG	0	2	N2END	4	
125	NORTH	Co	0	1	Ci	1	# carry
126	EAST	E1BEG	1	0	E1END	8	
127	EAST	E4BEG	4	0	E4END	2	
128	SOUTH	S1BEG	0	-1	S1END	8	
129	SOUTH	S2BEG	0	-2	S2END	4	
130	WEST	W1BEG	-1	0	W1END	8	
131	WEST	W4BEG	-4	0	W4END	2	
132	JUMP	J_BEG	0	0	J_END	42	
133	BEL	LUT4.vhdl	LA_				
134	BEL	LUT4.vhdl	LB_				
135	BEL	LUT4.vhdl	LC_				
136	BEL	LUT4.vhdl	LD_				
137	BEL	MUX8LUT.vhdl					
138	MATRIX	LUT4AB_switch_matrix.vhdl					
139	*EndTILE*						

- Wires
- Primitives (basic elements)
- Switch matrix

eFPGA Ecosystem – Switch Matrix Definition

```
# LUT4AB
# double with MID cascade : [N,E,S,W]2BEG --- [N,E,S,W]2MID -> [N,E,S,W]2BEGb --- [N,E,S,W]2END (
[N|E|S|W]2BEGb[0|1|2|3|4|5|6|7],[N|E|S|W]2MID[0|1|2|3|4|5|6|7]

############## LUT Inputs ##############
############## LUT Inputs ##############
############## LUT Inputs ##############

# shared double MID jump wires
J2MID_ABa_BEG[0|0|0|0],[JN2END3|N2MID6|S2MID6|W2MID6]
J2MID_ABa_BEG[1|1|1|1],[E2MID2|JE2END3|S2MID2|W2MID2]
J2MID_ABa_BEG[2|2|2|2],[E2MID4|N2MID4|JS2END3|W2MID4]
J2MID_ABa_BEG[3|3|3|3],[E2MID0|N2MID0|S2MID0|JW2END3]


# Carry chain  Ci -> LA_Ci-LA_Co -> LB_Ci-LB_Co -> ... ->
LA_Ci,Ci0
L[B|C|D|E|F|G|H]_Ci,L[A|B|C|D|E|F|G]_Co
Co0,LH_Co
```

	A	B	C	D	E	F
1	CLB	N1END0	N1END1	N1END2	N2END0	N2EN
2	N1BEG0	0	1	1	1	1
3	N1BEG1	1	0	1	0	0
4	N1BEG2	1	0	1	0	1
5	N2BEG0	0	1	0	1	0
6	N2BEG1	1	0	0	0	0
7	N2BEG2	1	1	1	0	0
8	N4BEG0	0	1	0	1	1
9	N4BEG1	1	1	1	1	1
10	E1BEG0	1	0	1	0	1
11	E1BEG1	1	1	0	1	1

- Describes the adjacency in a symbolic way
  <mux_output>,<mux_input>
- Alternatively adjacency matrix

FPGA-Research-Manchester / FABulous

Public

Pin | Watch 10 | Fork 9 | Star 53

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

master | 10 branches | 0 tags | Go to file | Add file | Code

y8j31 Update Usage.rst — 2e4d557 19 days ago — 388 commits

.github/workflows	Create fab flow script for pair generation and add to CI t...	4 months ago
docs	Update Usage.rst	19 days ago
fabric_files	clean up	2 months ago
fabric_generator	Update create_basic_files.sh	19 days ago
nextpnr @ 8a0fed7	submodule update	2 months ago
.gitattributes	Add .gitattributes and renormalize line endings	11 months ago
.gitignore	I/O mapping complete	17 months ago
.gitmodules	Nextpnr is a sub-repository of Fabulous	3 months ago
.readthedocs.yaml	documentation update	2 months ago

About

Fabric generator and CAD tools

fabulous.readthedocs.io/en/latest/

Readme
Apache-2.0 License
53 stars
10 watching
9 forks

Releases

No releases published
Create a new release

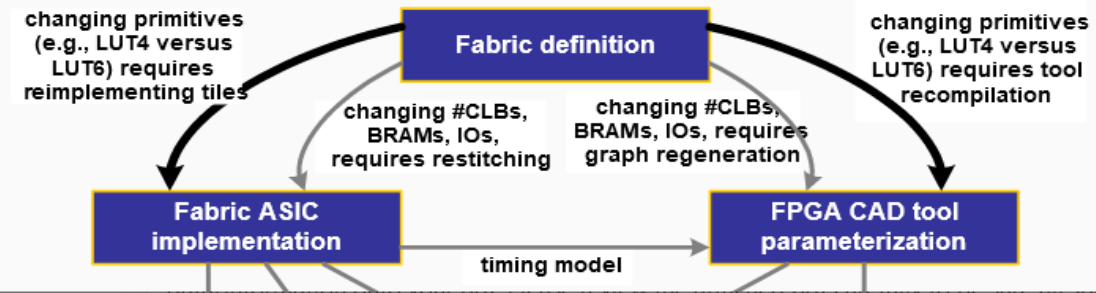
FABULOUS
eFPGAs made easy

FABulous: an Embedded FPGA Framework

FABulous is designed to fulfill the objectives of ease of use, maximum portability to different process nodes, good control for customization, and delivering good area, power, and performance characteristics of the generated FPGA fabrics. The framework provides templates for logic, arithmetic, memory, and I/O blocks that can be easily stitched together, whilst enabling users to add their own fully customized blocks and primitives.

The FABulous ecosystem generates the embedded FPGA fabric for chip fabrication, integrates SymbiFlow toolchain release packages, deals with the bitstream generation and after fabrication tests. Additionally, we will provide an emulation path for system development.

This guide describes everything you need to set up your system to develop for FABulous ecosystem.



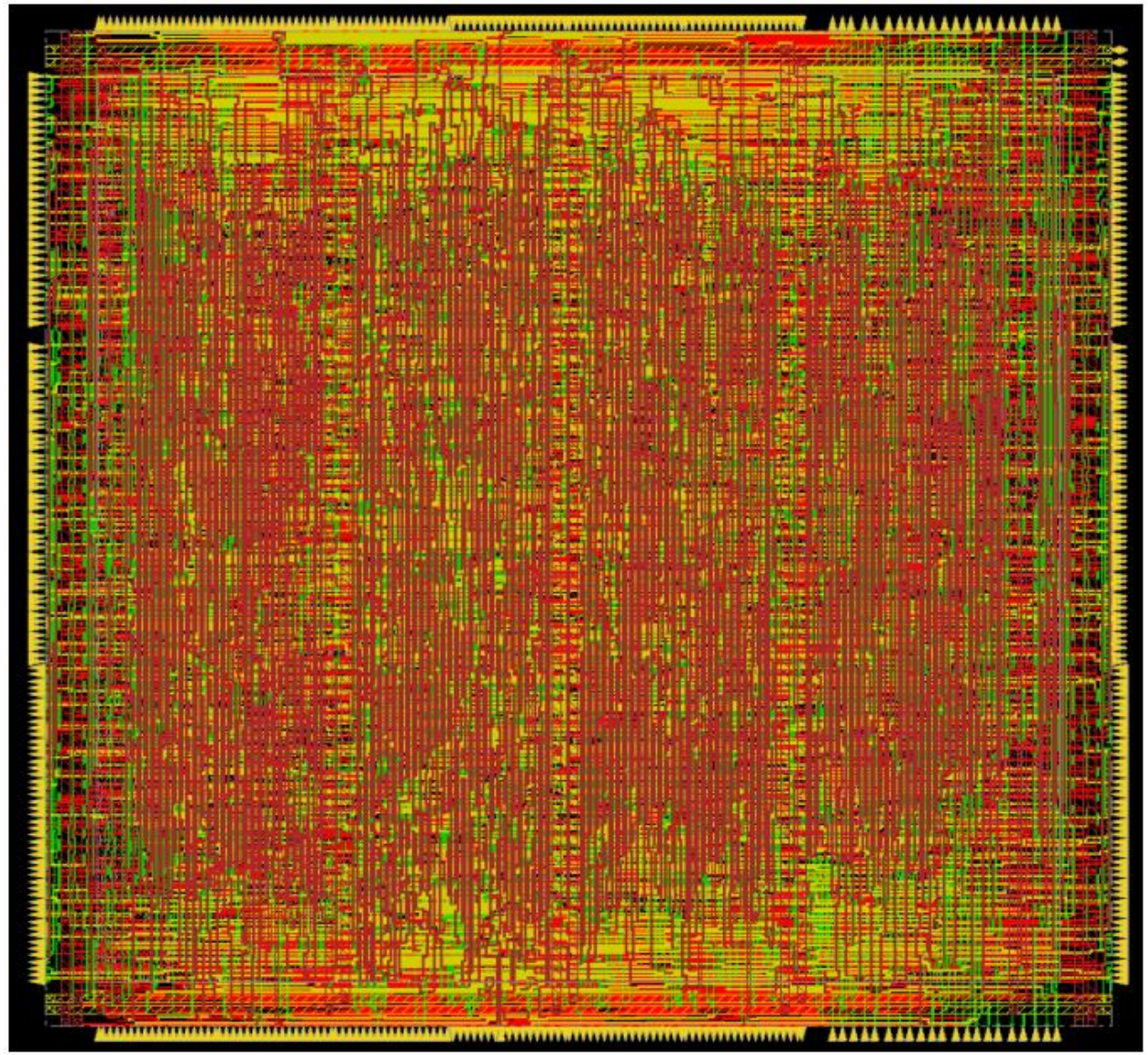
Tile-based Design in FABulous

- Tile-based design
  - Consistent timing
  - Reliable implementation
  - Reuse optimization effort

    → Share tiles as GDS macros

    (Thanks Google for the Shuttle Program)
- Routing fabric is dominating area

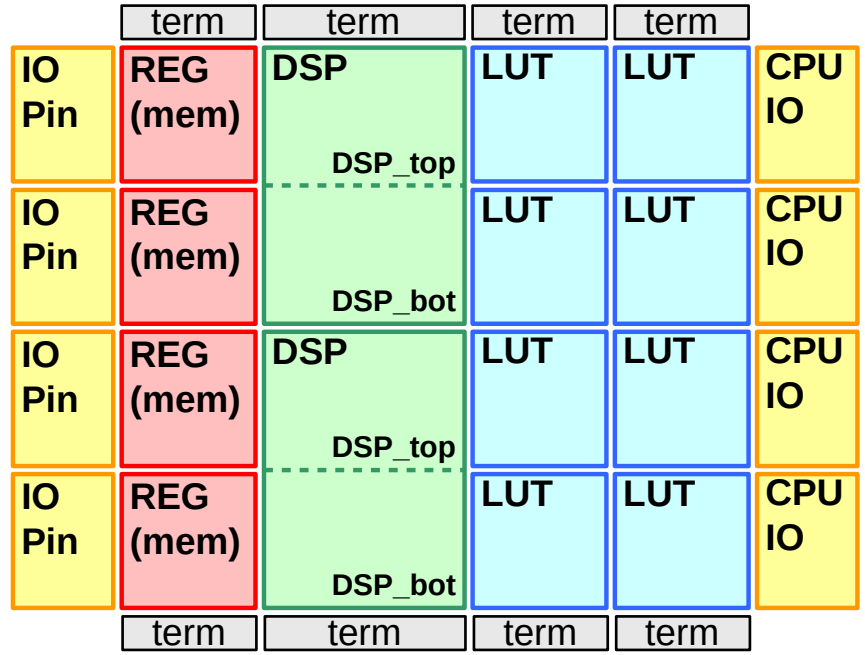
  → simple optimizations gain most optimization potential
  - Use cheap configuration memory (custom cells or latches)
  - Use custom multiplexers



Tile-based Design in FABulous

- Define CLB (pragmatically) as smallest square tile
- Defines the grid height
- Widths depend on tile-type (CLB, REG, DSP, etc.)
- Replace standard cell multiplexers with custom mux-4

$A_{std\text{-}cell} - A_{c\text{-}mux4} \times N = (33.8\,\mu m^2 - 17.5\,\mu m^2) \times 376 = 6{,}116\,\mu m^2$



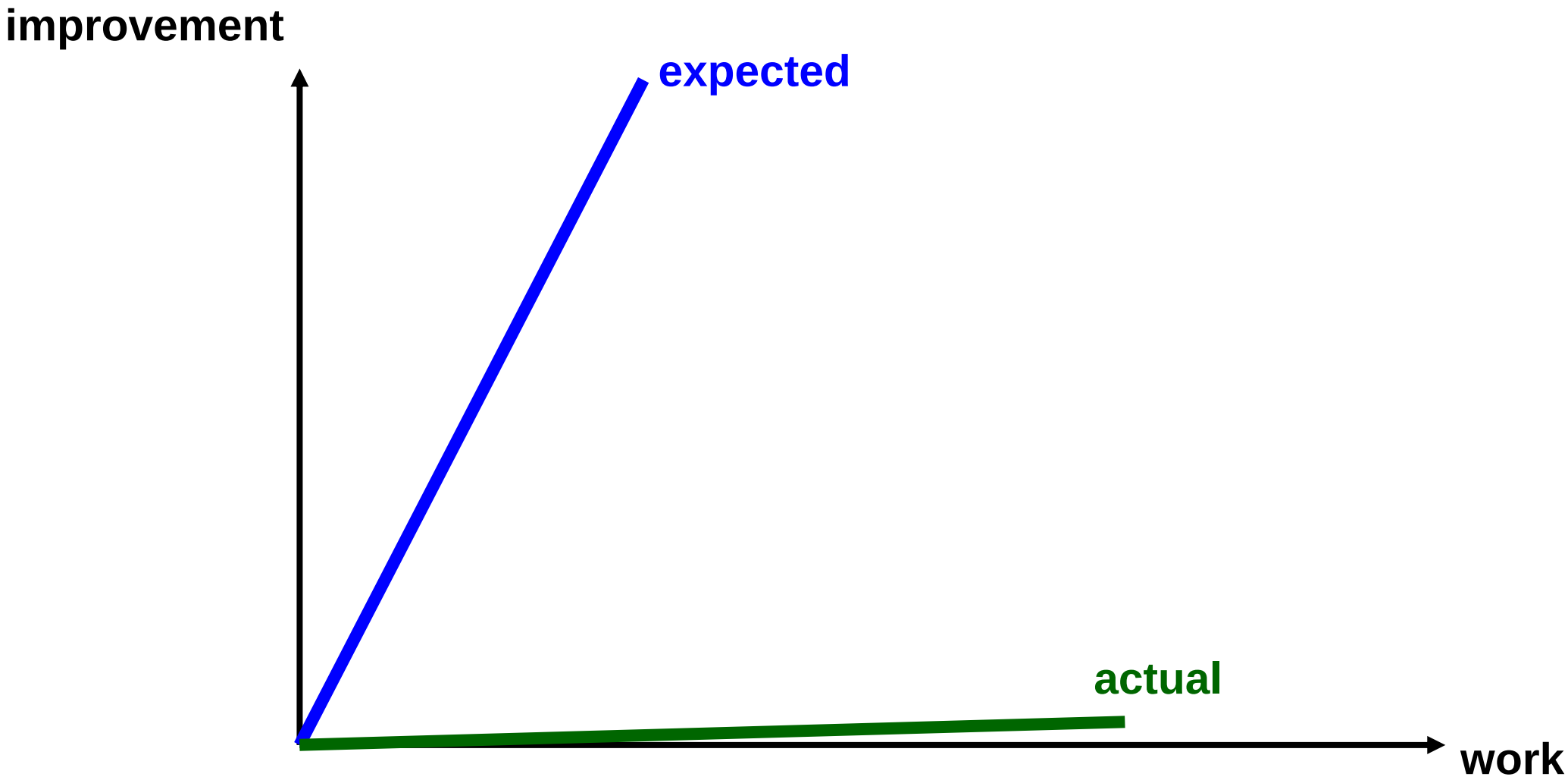
	Standard cell				Custom mux-4	
	height	width	**area**	util.	**area**	util.
CLB	219 µm	219 µm	**47,961**	81.8%	**46,225**	60.7%
REG	219 µm	214 µm	**46,866**	84.1%	**46,655**	64.3%
DSP	443 µm	185 µm	**81,955**	80.9%	**81,780**	56.7%

Observation:

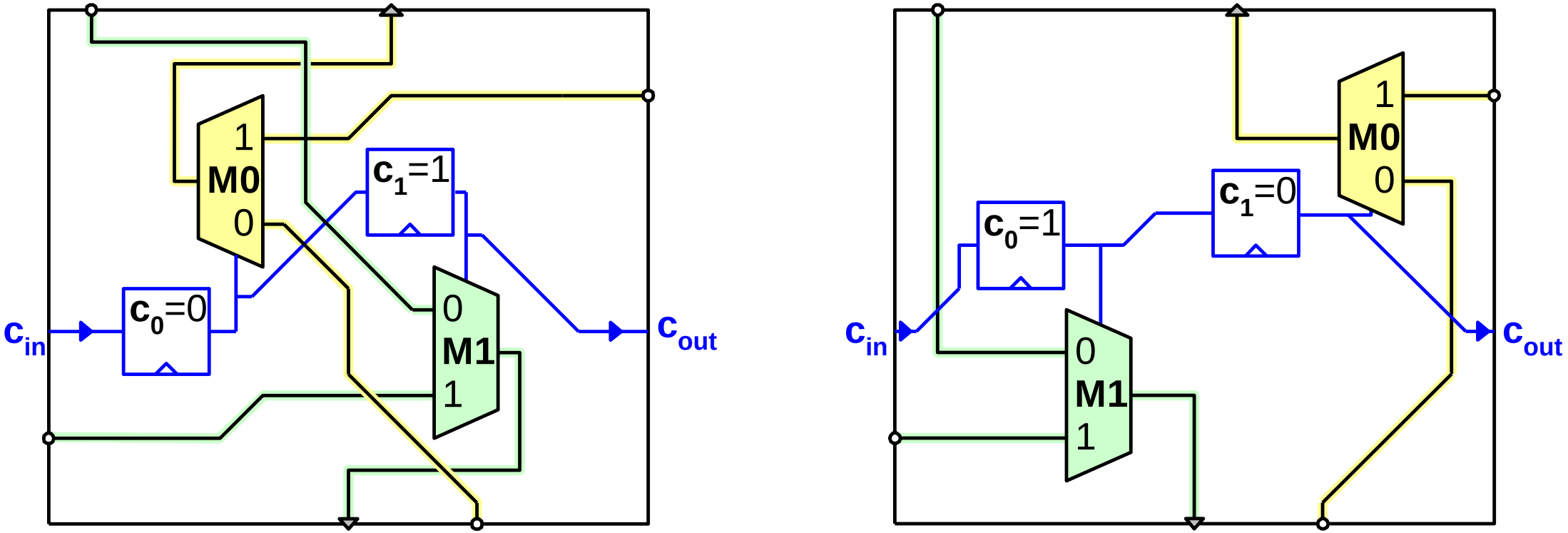
- No area improvement
- Instead: core utilization went down

→ Congested tile routing

In short



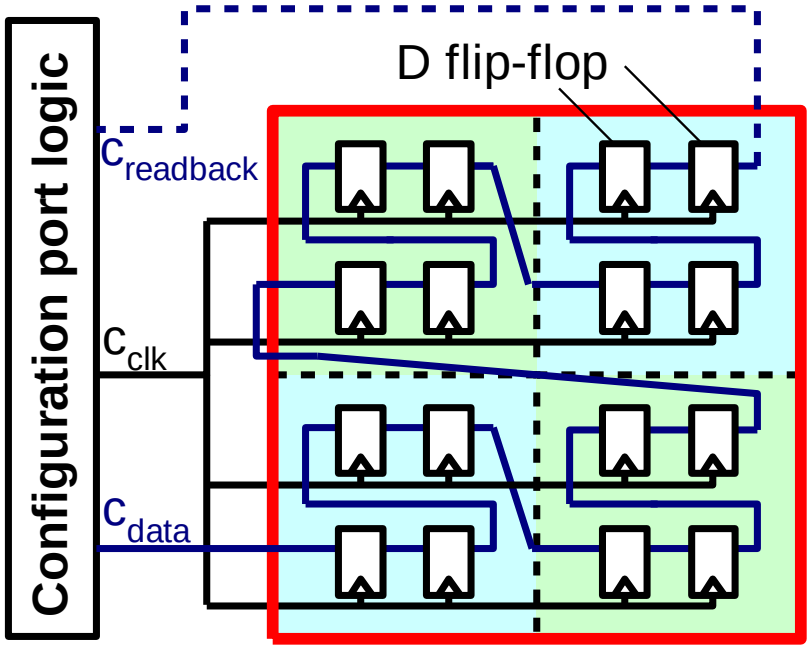
Optimization: Bitstream Remapping



- The configuration bit cells may induce inferior placement of multiplexers
- We can remap configuration bits → requires remapping of the bitstream (trivial)

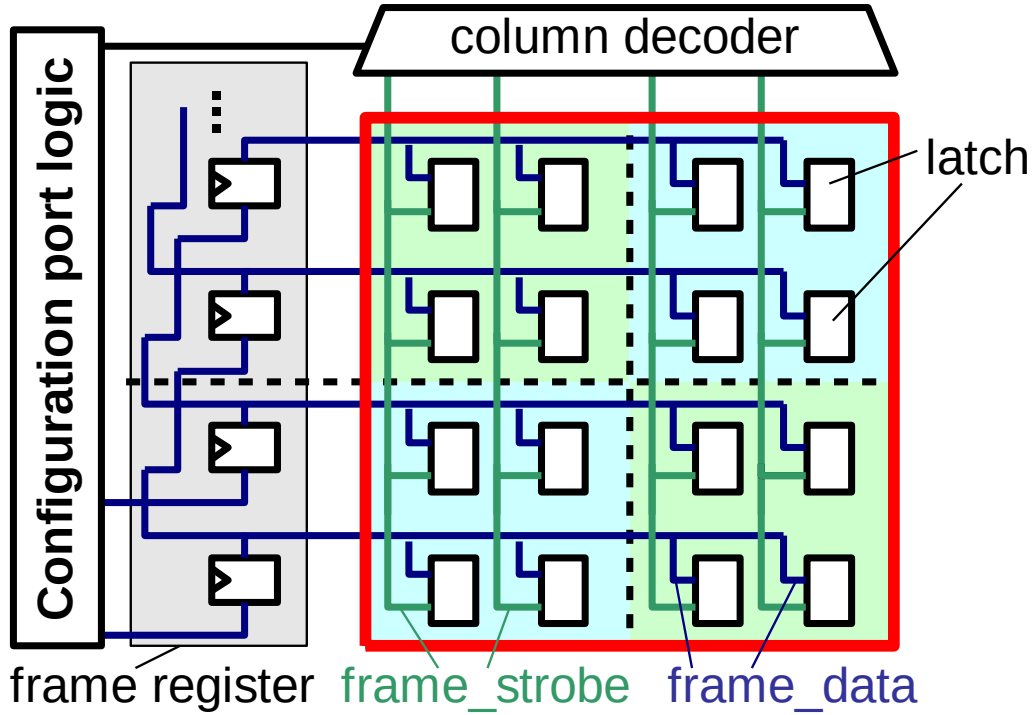
Optimization: Bitstream Remapping

Shift-register Configuration



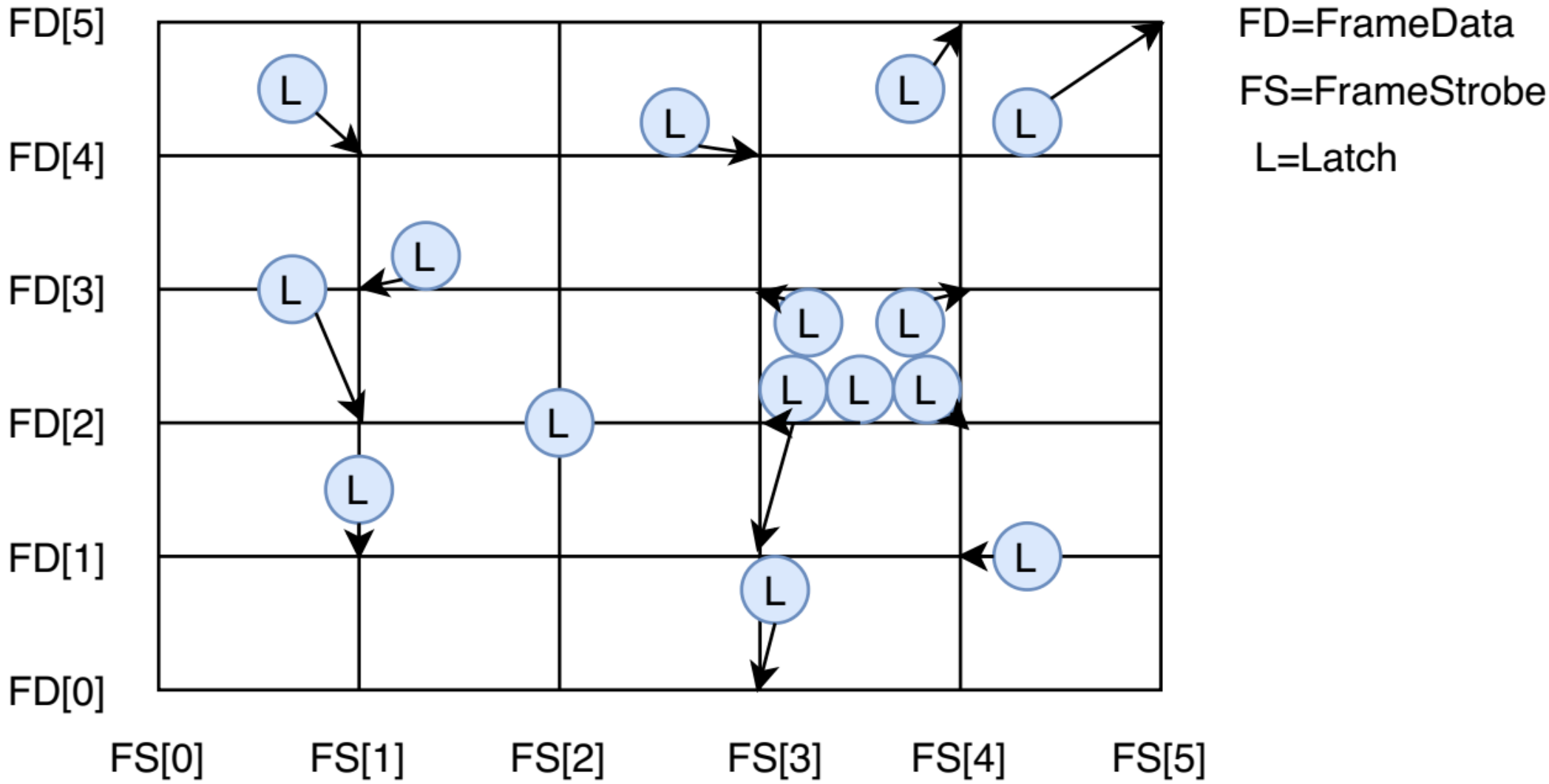
- Change order of flip-flops in the chain

Frame-based Configuration



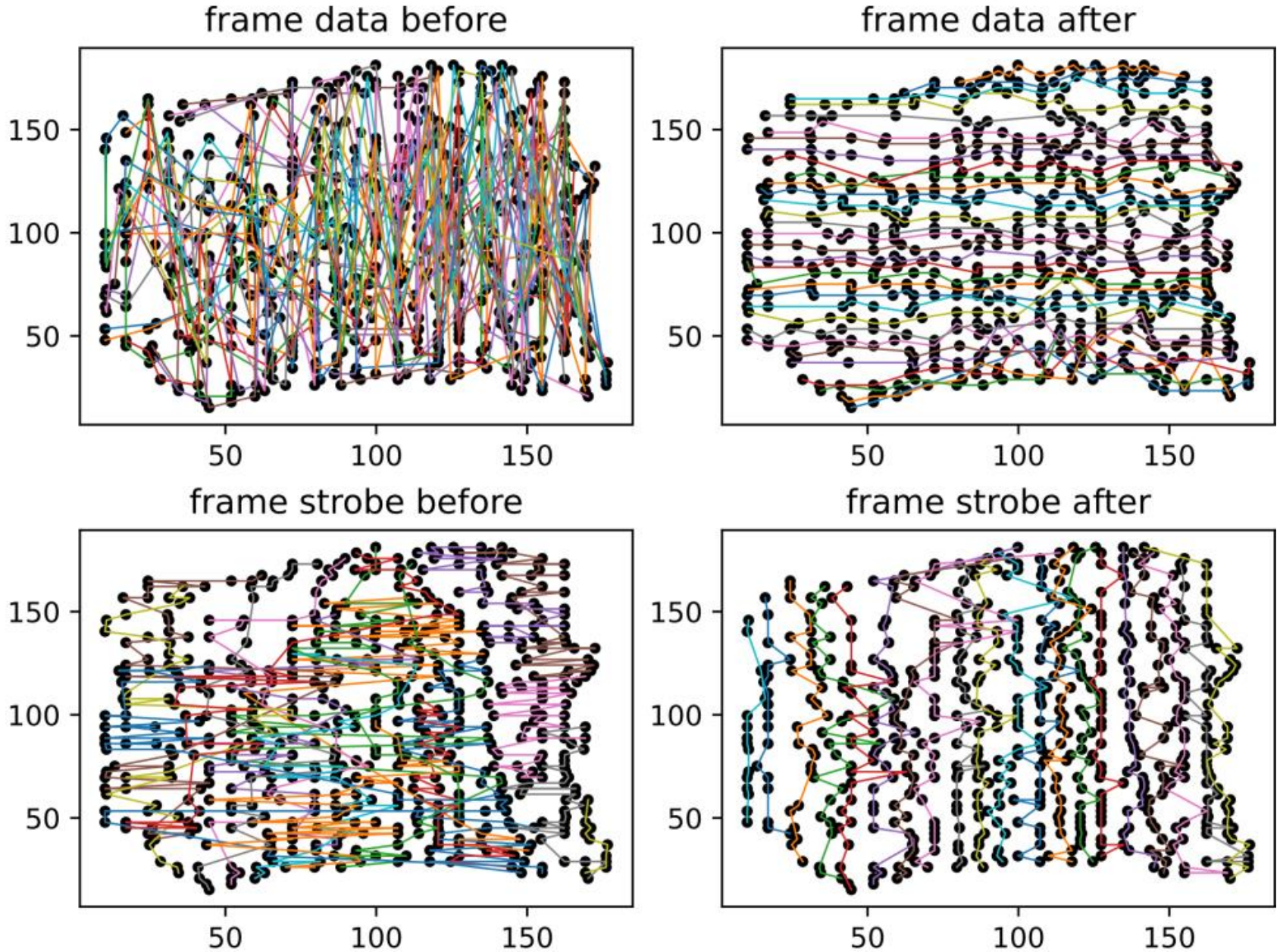
- Change latch-to-gridpoint mapping (with frame-data and frame-strobe gridpoints)

Optimization: Bitstream Remapping

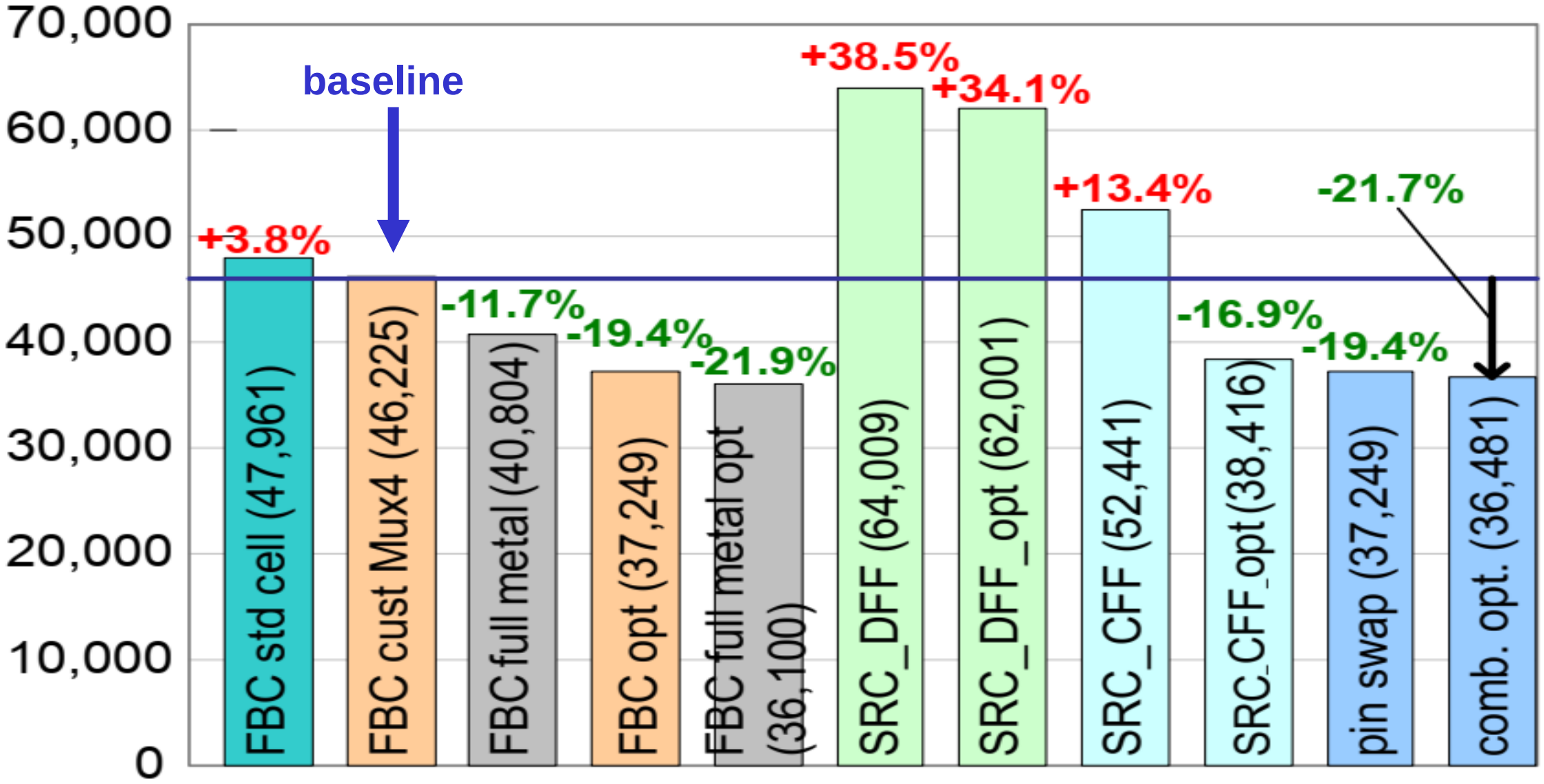


- We use Google's Operations Research tools to compute the grid points (https://github.com/google/or-tools)

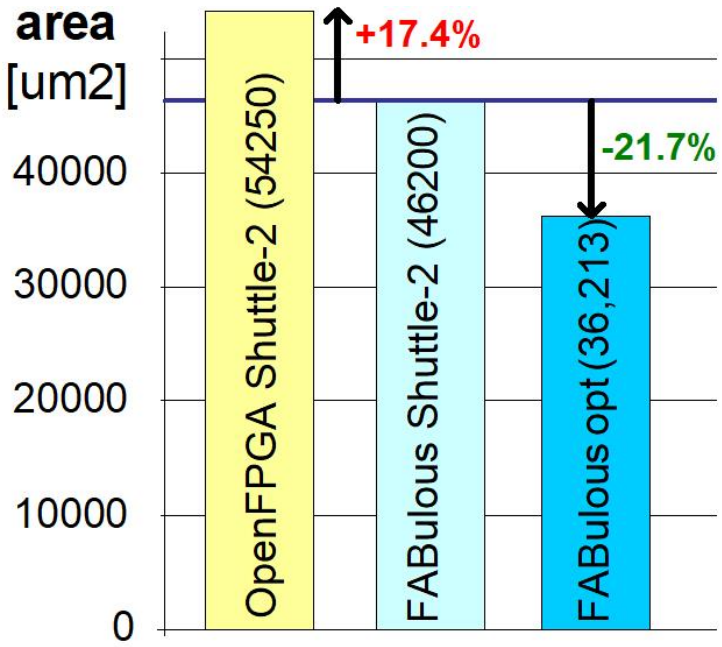
Optimization: Bitstream Remapping

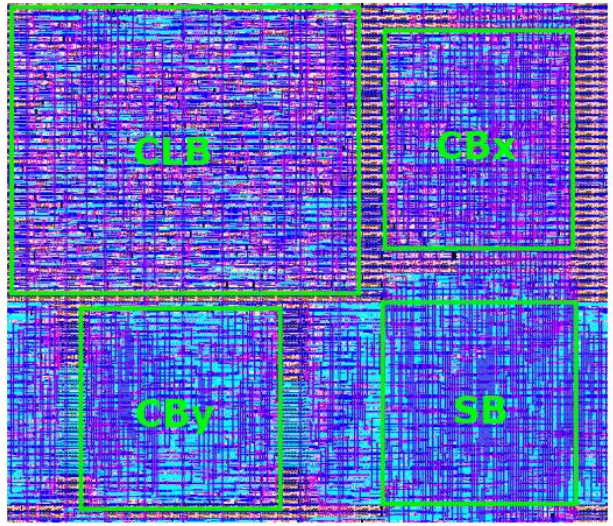


Bitstream Remapping – Results

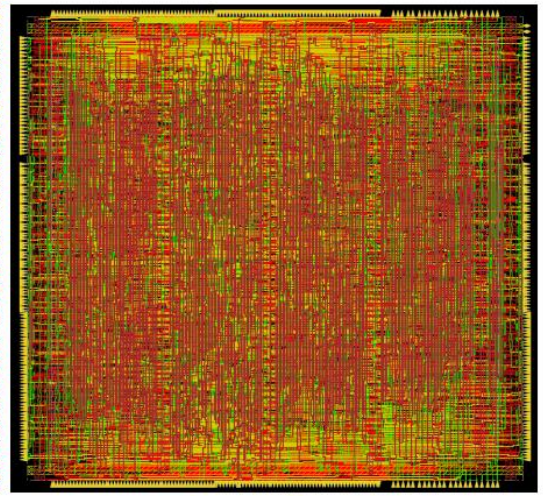


FABulous versus OpenFPGA (on Sky130)

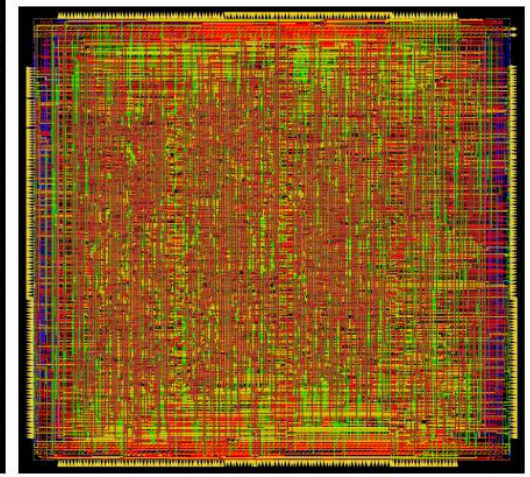




OpenFPGA Shuttle 2
250x217 um2
(+17.4%)



Fabulous Shuttle 2
220x210 um2
(+0%)



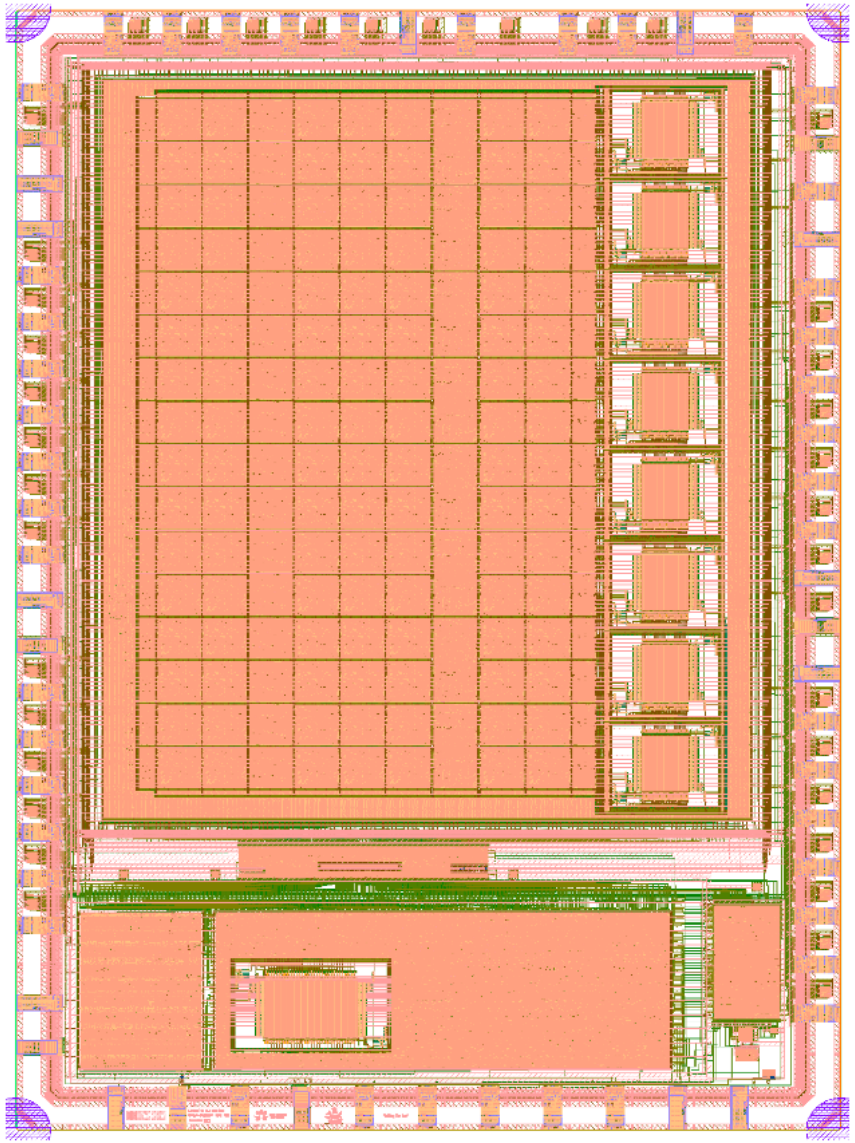
Fabulous Opt.
193x193 um2
(-21.7%)

- FABulous and OpenFPGA have a Google Shuttle2 submission
- ~ same physical impl. problem
- OpenFPGA CLBs are 17% bigger
- New optimizations gave us further 21.7% in density on the same netlist!

	resources
OpenFPGA	1300xMUX2 / 530xDFF
FABulous	376xMUX4 / 46xMUX2 / 8xFF / 586xlatch

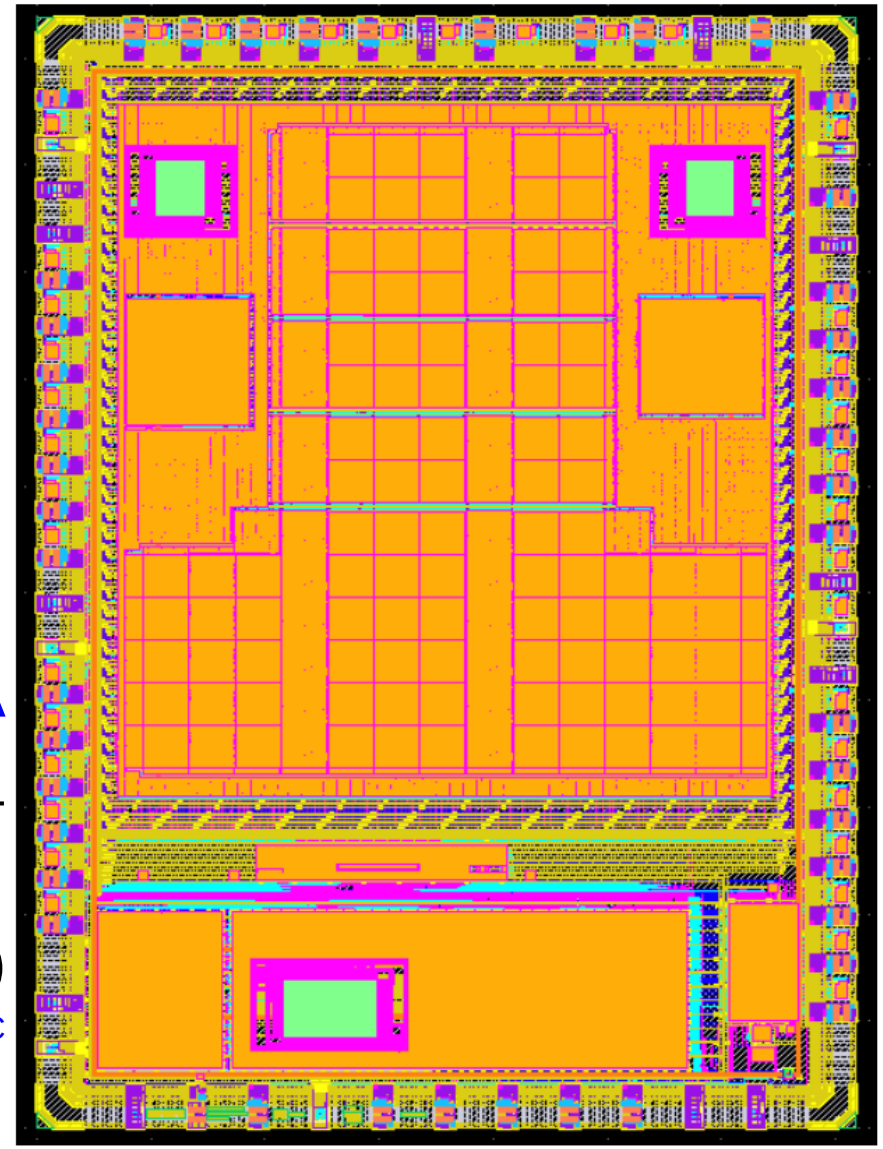
~1200 MUX2

FABulous Chip Gallery



Sky130 with CLBs, DSPs, RegFiles, BRAMs
Google Shuttle - MPW-2
(can implement RISC-V)
https://github.com/nguyendao-uom/eFPGA_v3_caravel

Dual-Ibex-Crypto-eFPGA
Google Shuttle - MPW-4
(custom instructions, T-shaped fabric)
https://github.com/nguyendao-uom/ICESOC



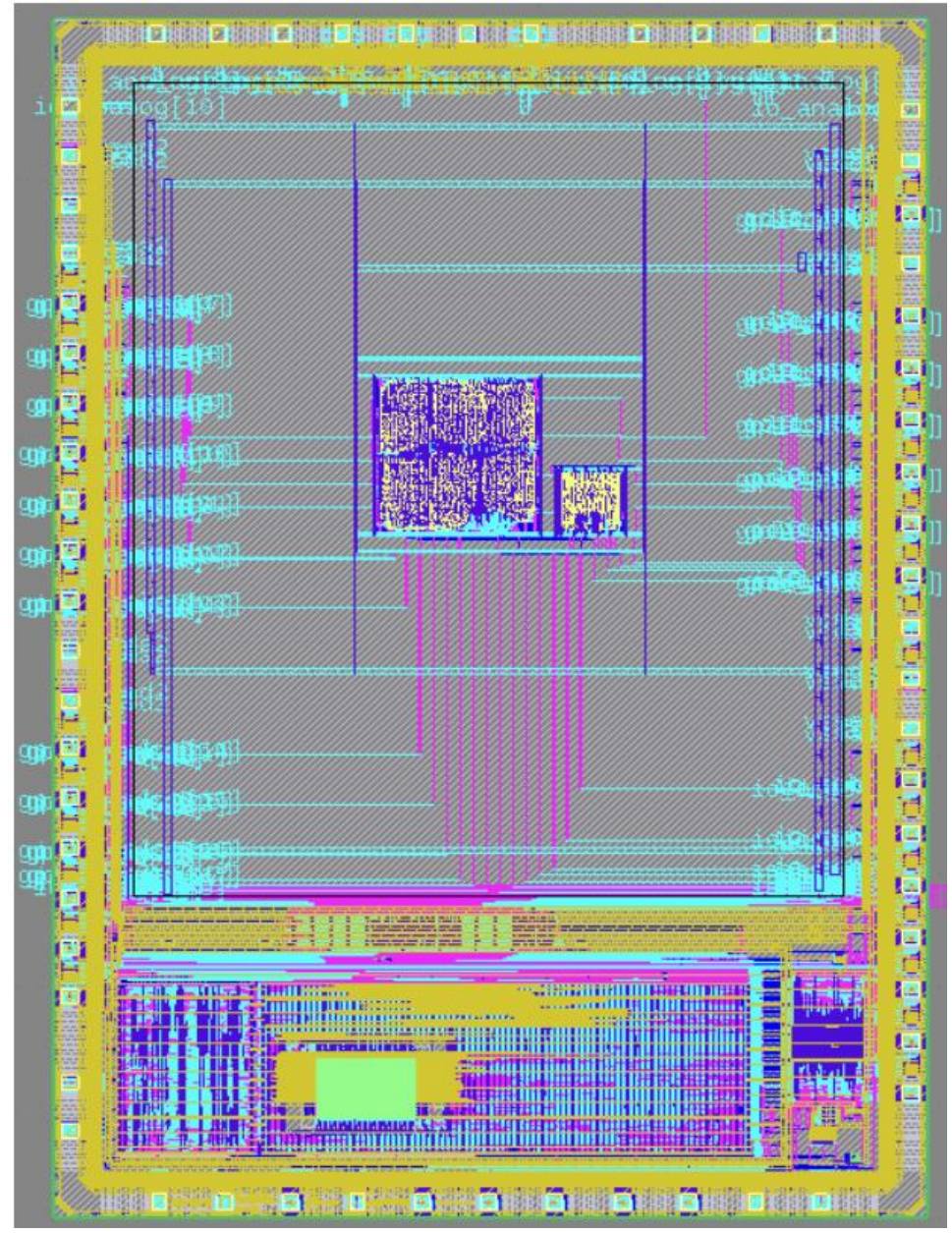
FABulous Chip Gallery

Open ReRAM FPGA test chip

- Sky130, Google Shuttle
  MPW4https://github.com/nguyendao-uom/rram_testchip
- Just enough logic to send
  „Hello World“ to a UART
- Different configuration modes

Posible advantages of ReRAM FPGAs

- Security (user circuit is encoded in resitsive states)
- Reliability (ReRAM is radiation hard)
- Probably density
- Instantanous on
- CMOS friendly



The FABulous eFPGA Framework – Wrap-up

- Heterogeneous (FPGA) fabric (DSBs, BRAMs, CPUs, custom blocks)
  - Multiple tiles can be combined for integrating more complex blocks
  - Custom blocks can be instantiated directly in Verilog and are integrated in Yosys, VPR/nextpnr CAD tools (Synthesis, Place&Route) (as primitive blocks)
- Support for dynamic partial reconfiguration (some elements of XC6200, like wildcard configuration)
- Configuration through shift registers or latches (or custom cells)
- Support for custom cell primitives (passgate multiplexers)
- Good performance / area / power figures (about 1.3x worse than Xilinx) (could be narrowed down through customization)
- Usable by FPGA users (you don‘t have to be an FPGA architect)
  → there are FPGA classics that we have/will clone
- ToDo: multiple clock domains, mixed-grained granularity, more ReRAM, processes, ...

FABulous Team

People:

Nguyen Dao	**nguyen.dao@manchester.ac.uk**
Jing Yu	**jing.yu@manchester.ac.uk**
Bardia Babaei	**bardia.babaei@postgrad.manchester.ac.uk**
King Chung	**king.chung@student.manchester.ac.uk**
Tuan La	**tuan.la@manchester.ac.uk**
Andrew Attwood	**a.j.attwood@ljmu.ac.uk**
Bea Healy	**tabitha.healy@student.manchester.ac.uk**
Dirk Koch	**dirk.koch@ziti.uni-heidelberg.de**

MANCHESTER 1824
The University of Manchester

FORTE | Forming the future of nanoelectronics

EPSRC
Engineering and Physical Sciences Research Council

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

See our projects under:

https://github.com/FPGA-Research-Manchester

This work is kindly supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant EP/R024642/1

We also thank Google for sponsoring the Efabless Open MPW Shuttle Program.