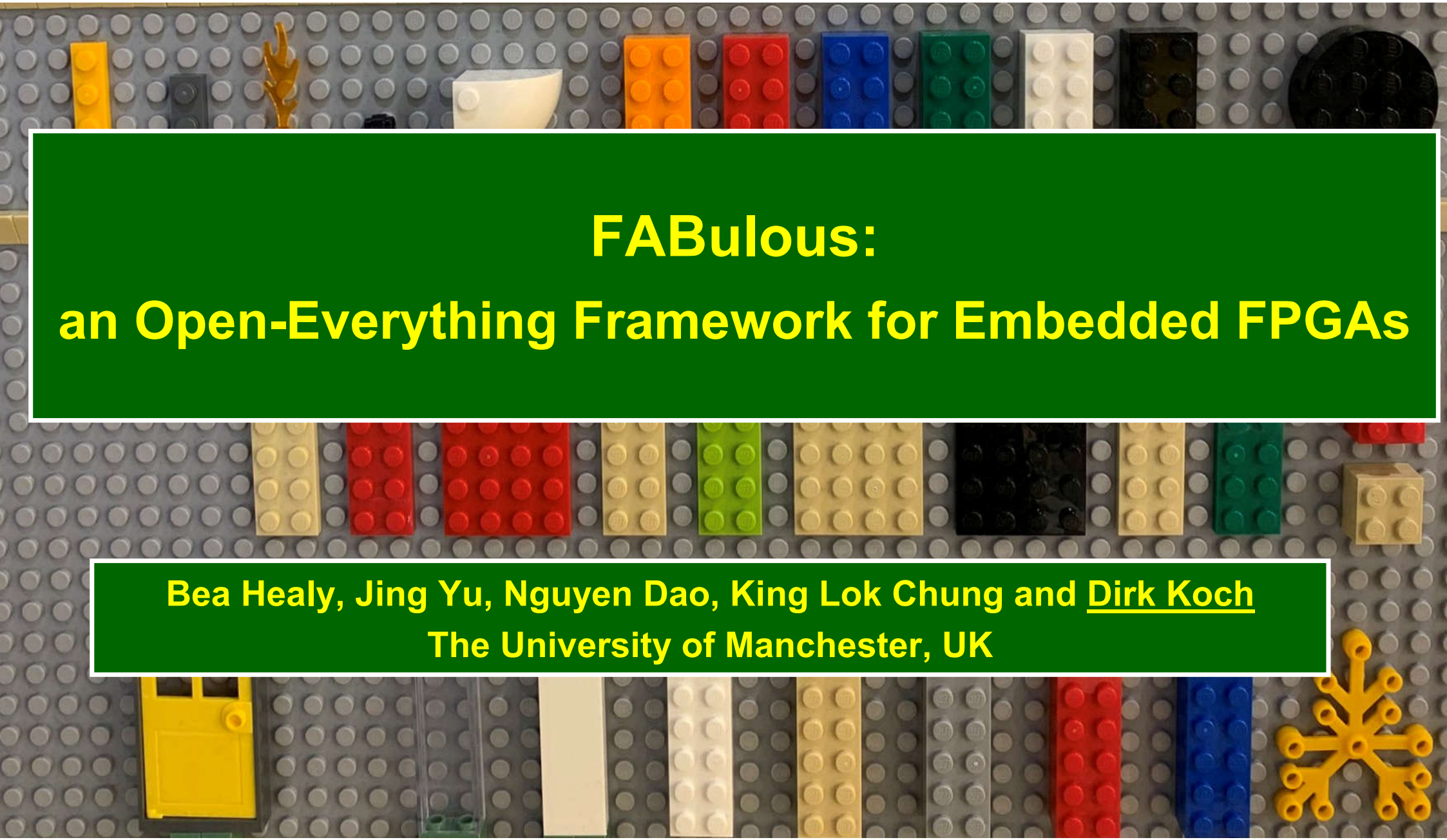


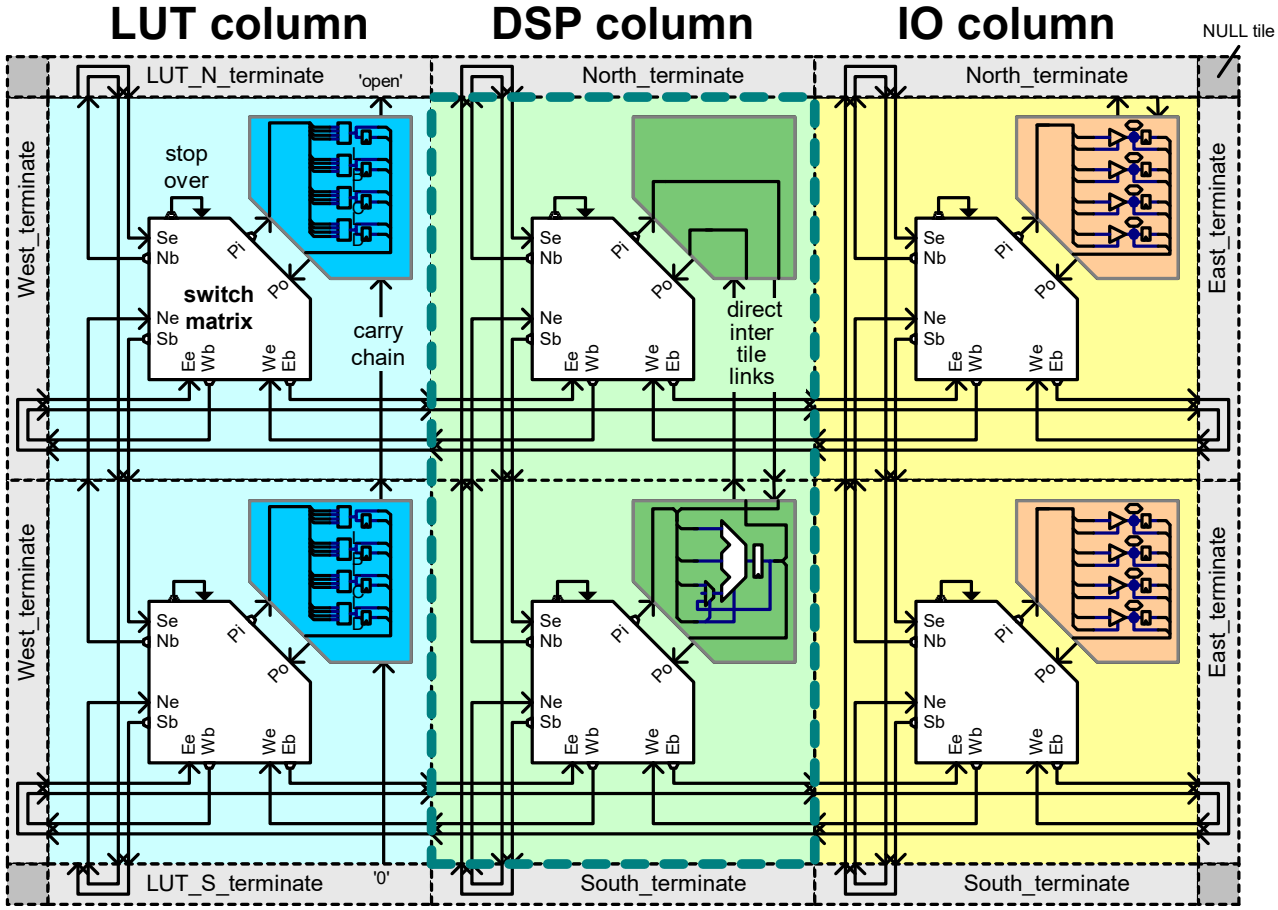
FABulous:
an Open-Everything Framework for Embedded FPGAs

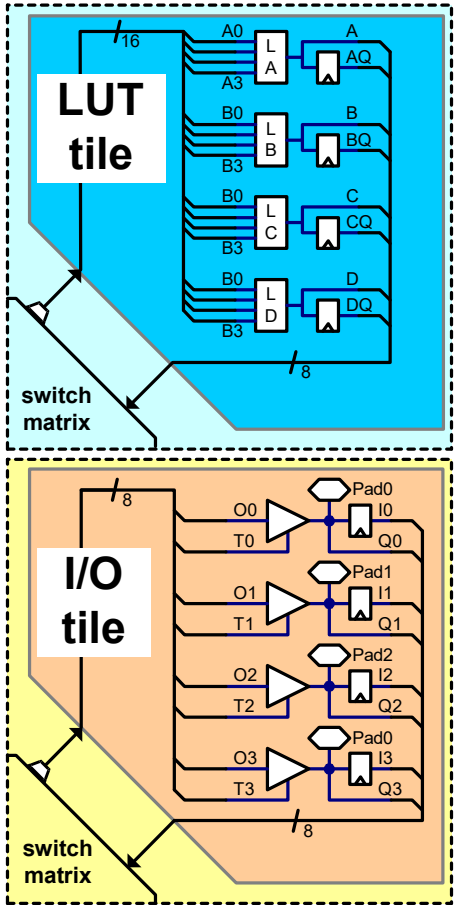
Bea Healy, Jing Yu, Nguyen Dao, King Lok Chung and Dirk Koch
The University of Manchester, UK

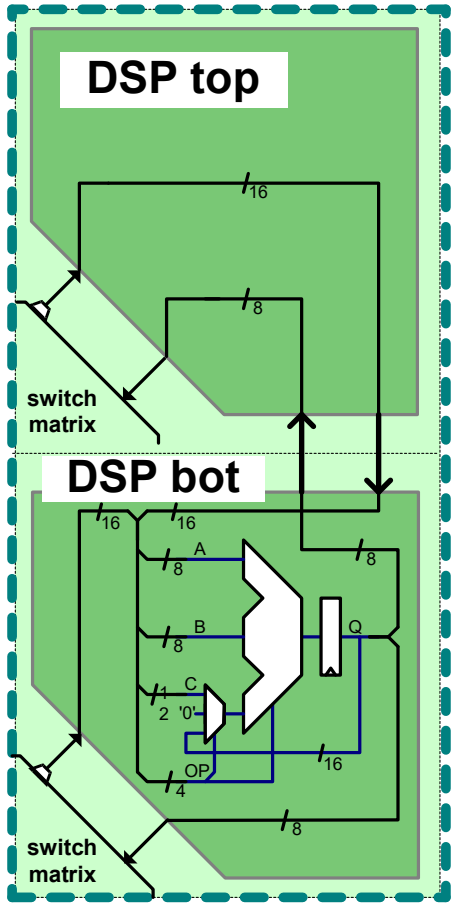
What is FABulous offering?

- Fully integrated open-source FPGA framework with good quality of results (area & performance)
- Entirely open and free, including commercial use (we integrated many other projects)
- Supports custom cells (if provided)
- Supports partial reconfiguration
- Designed for ease of use while providing full control as needed
- Versatile
  - Different flows (OpenLane ←→Cadance) (Yosys/nextpnr ←→ VPR)
  - Easy to customize, including the integration of own IP

Basic concepts

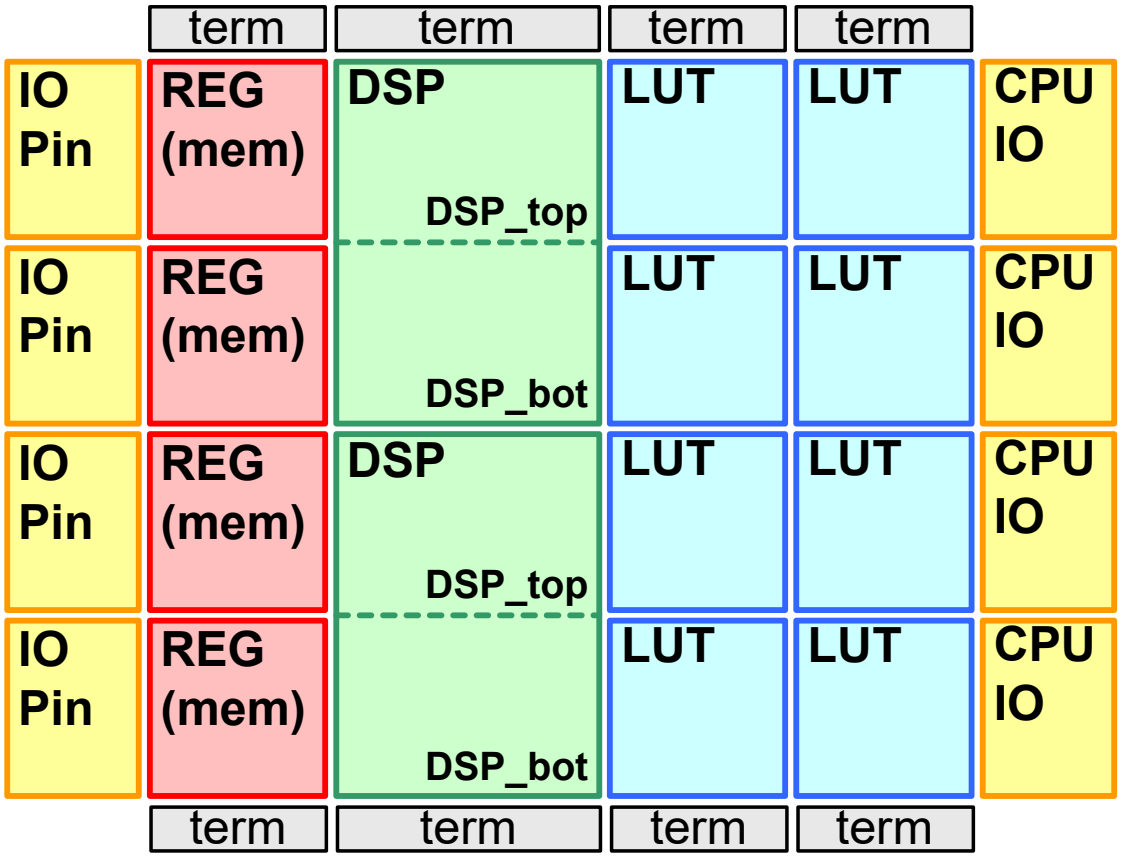






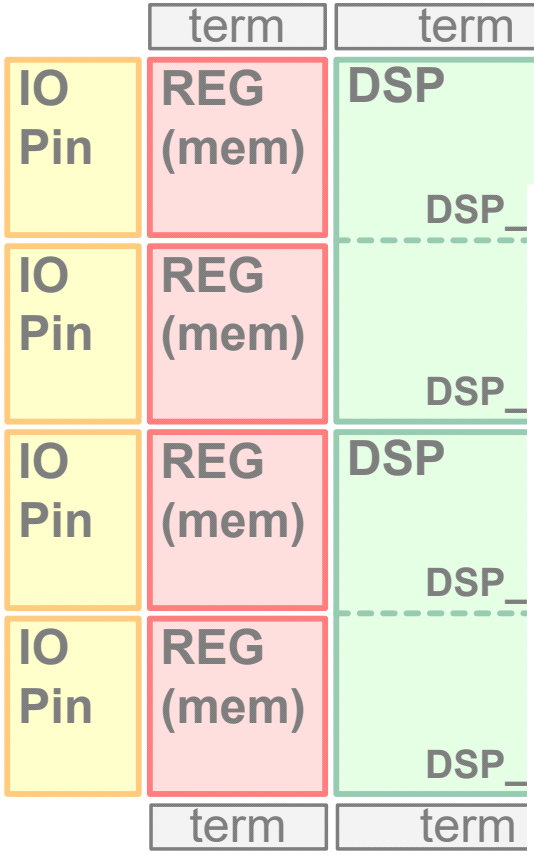
- Basic tiles have same height, but type-specific width (for logic tiles, DSPs, etc.)
- Adjacent tiles can be fused for more complex blocks (see the DSP example)

Let‘s build a small eFPGA: Fabric Definition



- 4 x register file, 2 x DSPs, 4 x LUTs (CLB), I/Os left and right,

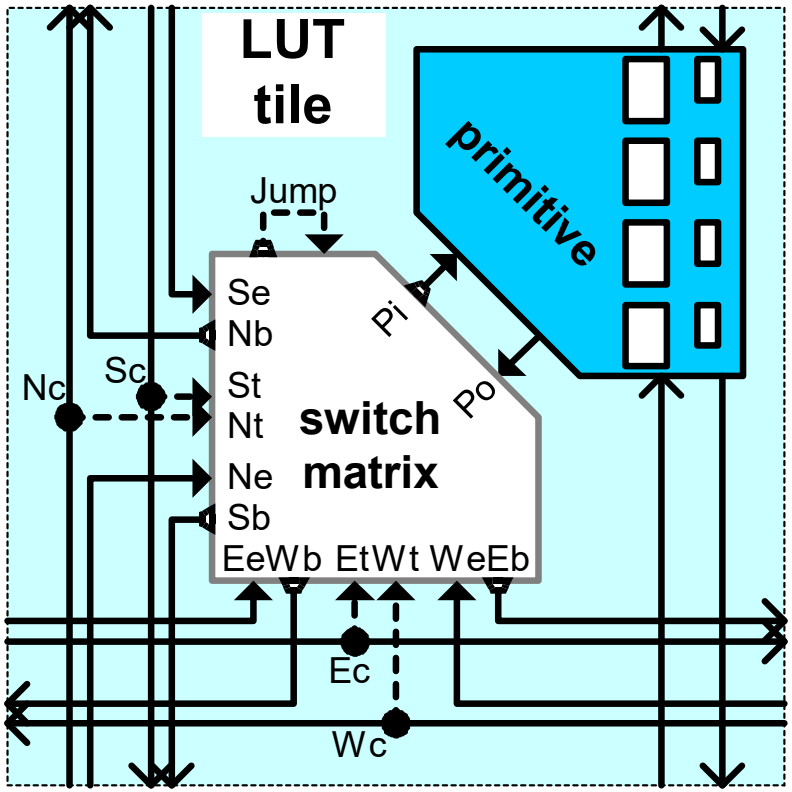
Let‘s build a small eFPGA: Fabric Definition



	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*					

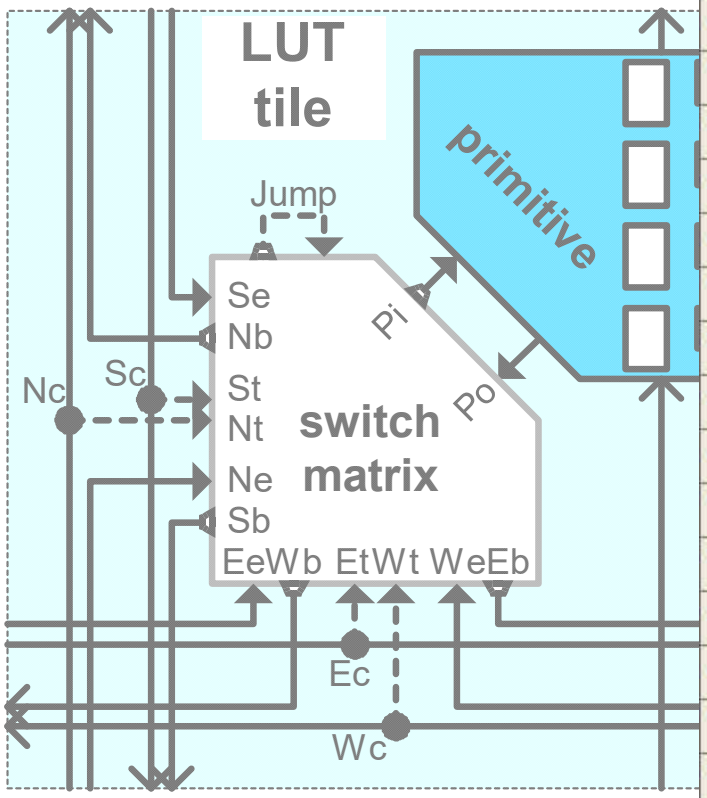
- 4 x register file, 2 x DSPs, 8 x LUT-tiles (CLB), I/Os left and right,
- A fabric is modelled as a spreadsheet (tiles are references to tile descriptors)

Let‘s build a small eFPGA: Tile Definition



- Wires
- Primitives (basic elements)
- Switch matrix

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

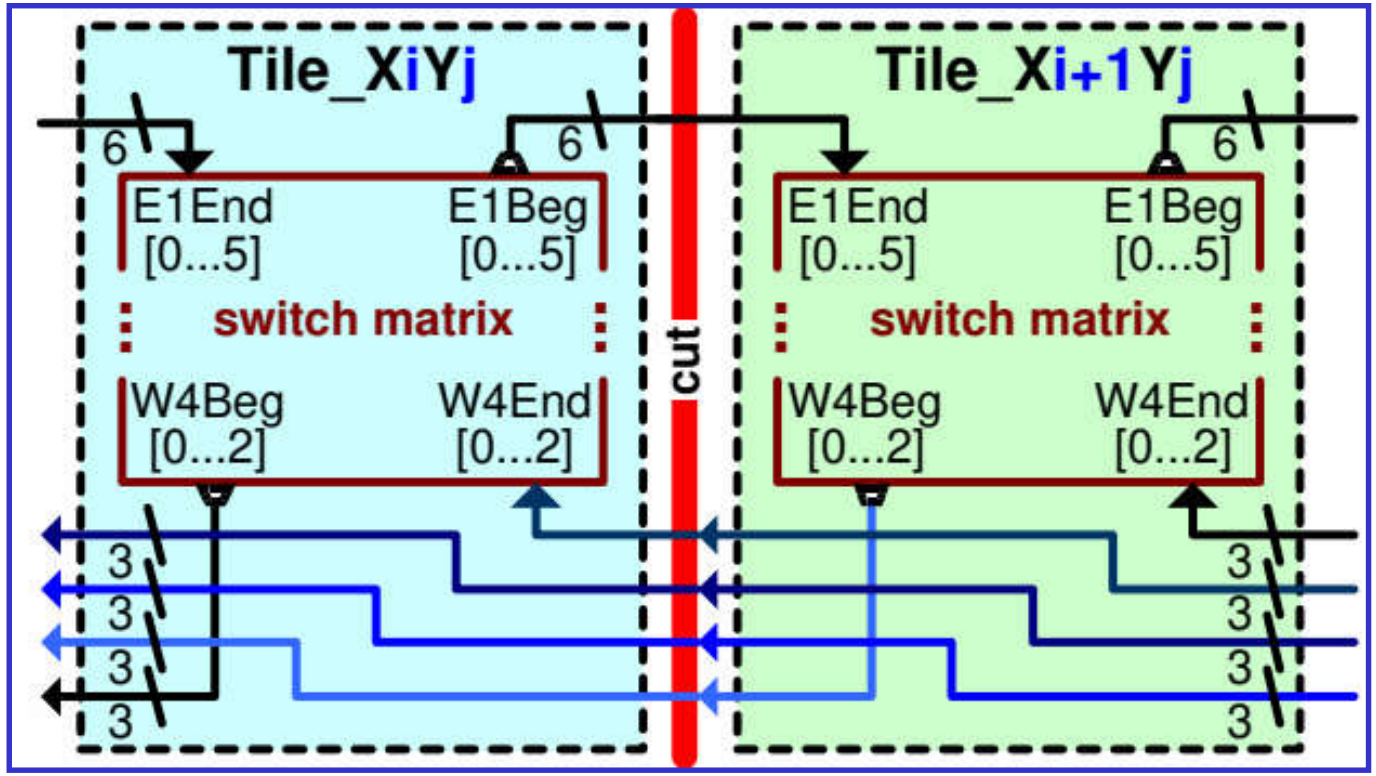
Let‘s build a small eFPGA: Tile Definition

121	*TILE*	LUT4AB				
122	#direction	source	X-offset	Y-offset	destination	wires
123	EAST	E1BEG	1	0	E1END	6
124	WEST	W4BEG	-4	0	W4END	3

Wires

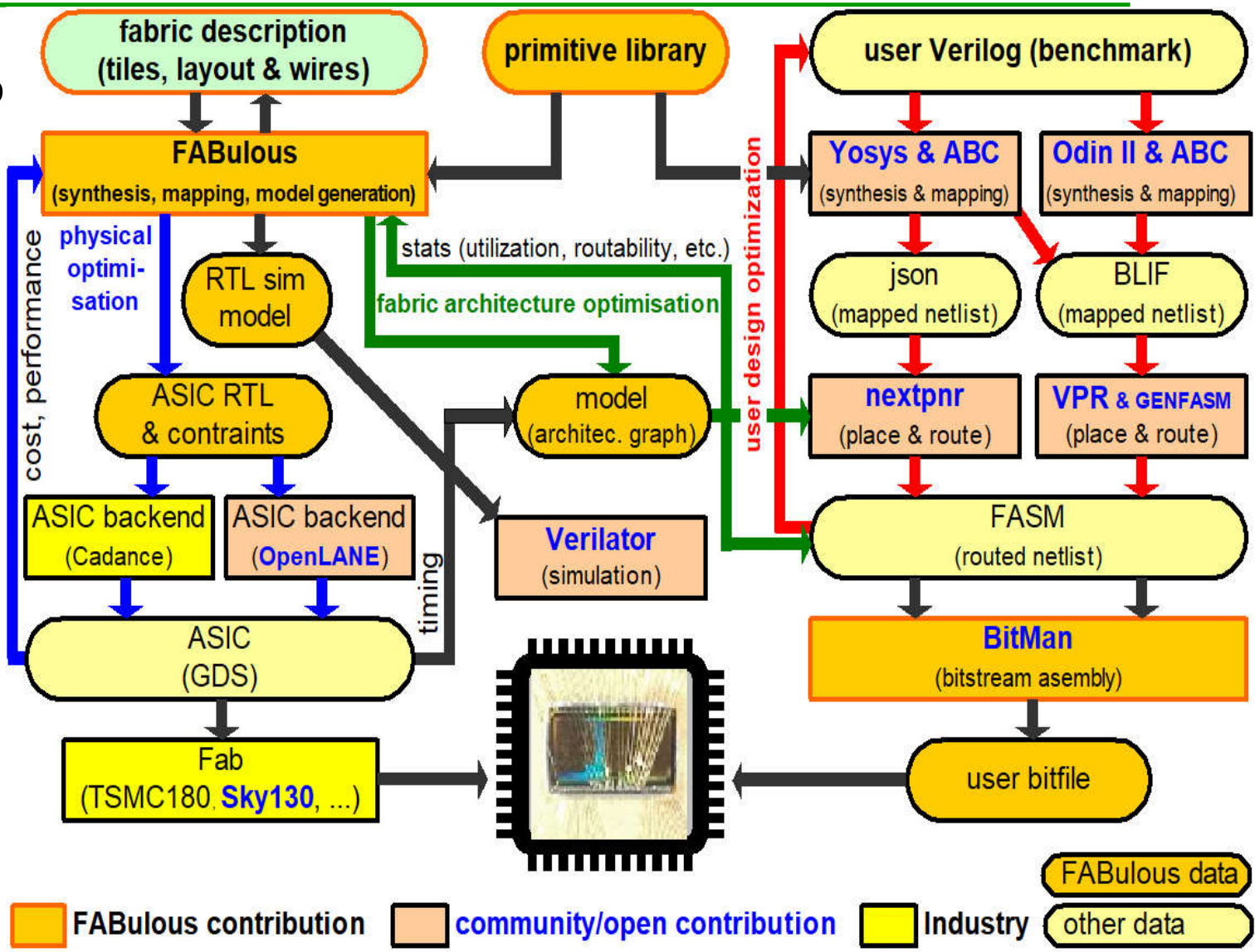
Switch matrix adjacency

	A	B	C
1	**LUT4AB**	N1END0	N1END1
2	N1BEG0	1	0
3	N1BEG1	0	0
4	N1BEG2	0	1
5	N1BEG3	1	0
6	N2BEG0	0	0
7	N2BEG1	0	0
8	N2BEG2	0	1
9	N2BEG3	0	0
10	N2BEG4	0	1
11	N2BEG5	0	0



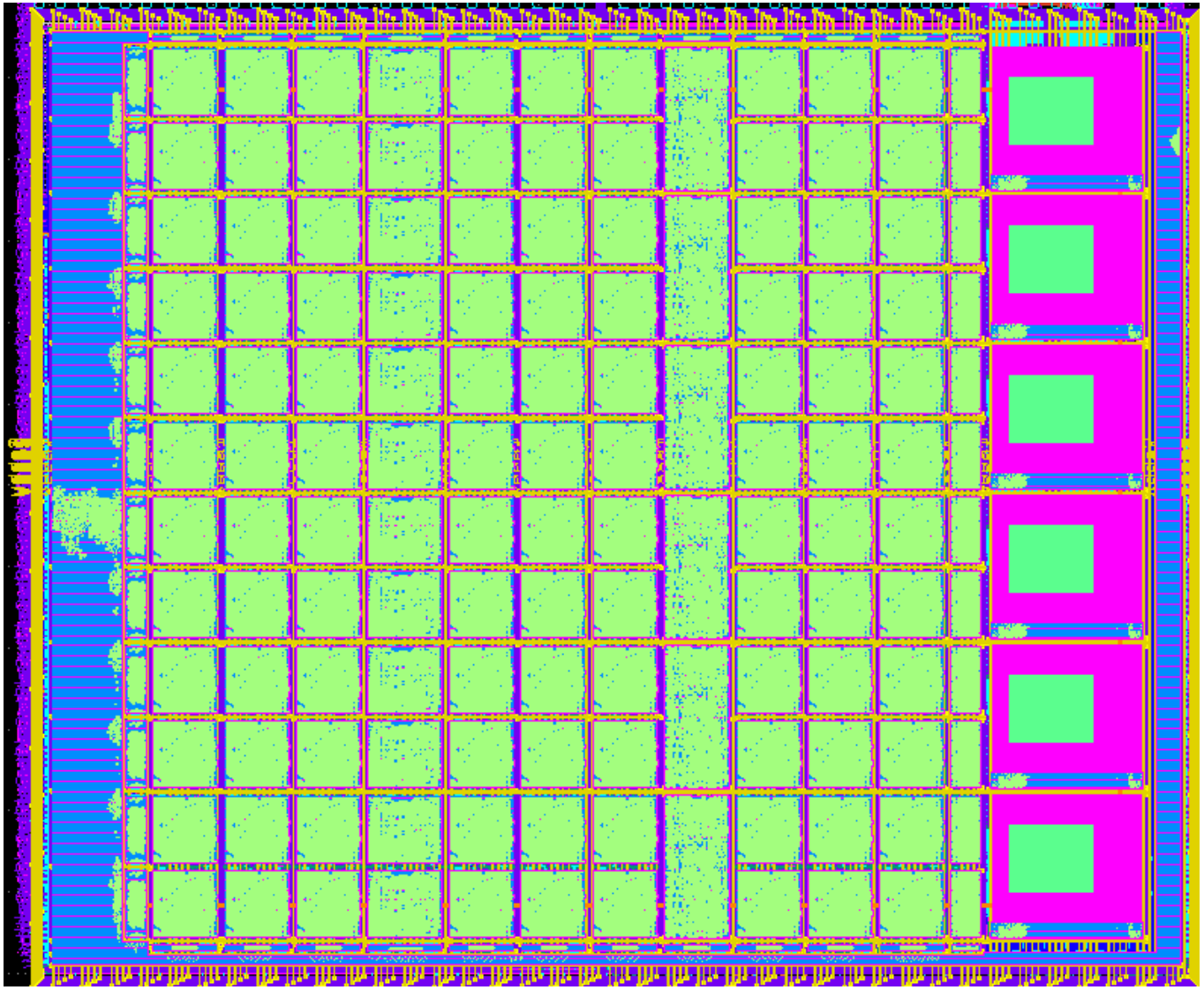
The FABulous eFPGA Ecosystem

- FABulous eFPGA generato
  - 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”



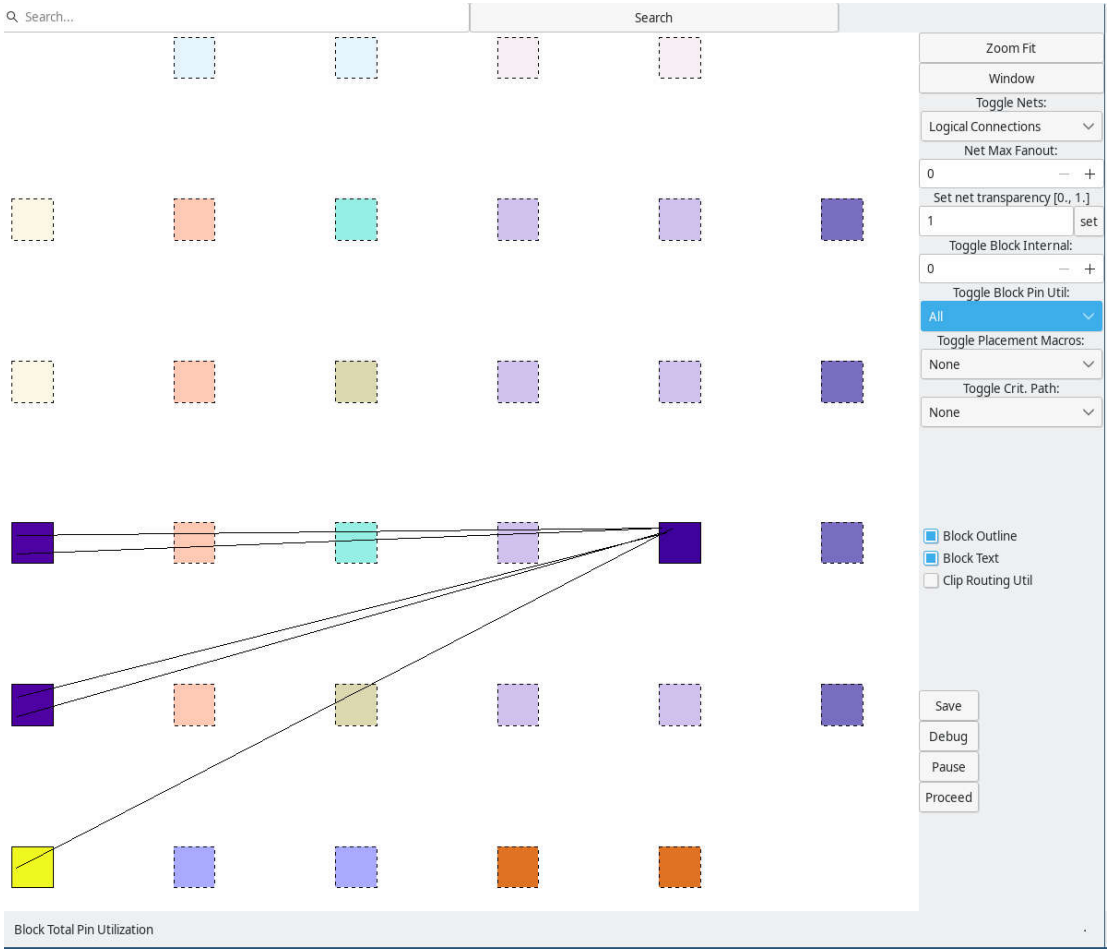
eFPGA – Customization and Integration

- Case study in 130 nm (~10mm^2) Skywater process (open PDK)
- 864 x LUT-4,
- 6 x DSPs (8x8 and 20 bit ACC)
- 12 register-file slices
  - 4bit wide 32 entries deep
  - 1 write, 2 read ports
- 6 x 16kb RAM blocks (using OpenRAM)
  - 1 read, 1 write port
  - Programmable aspect ratios for read and write ports
- Partial reconfiguration
- Internal self-configuration port



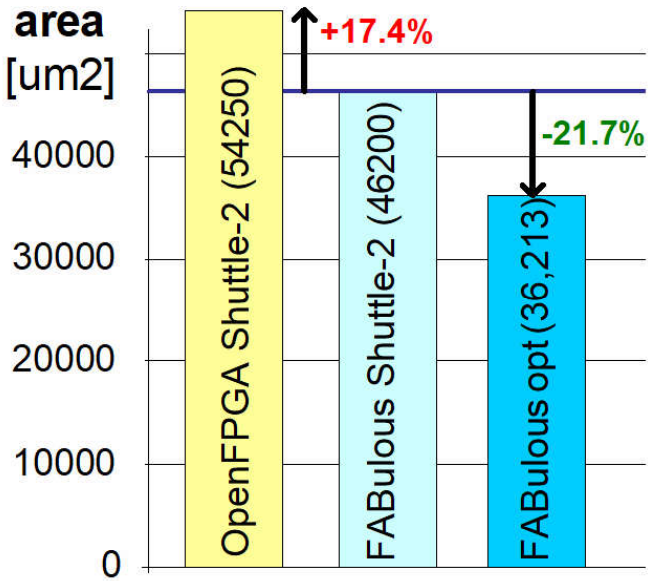
Implementing User Circuits (Verilog-to-Bitstream)

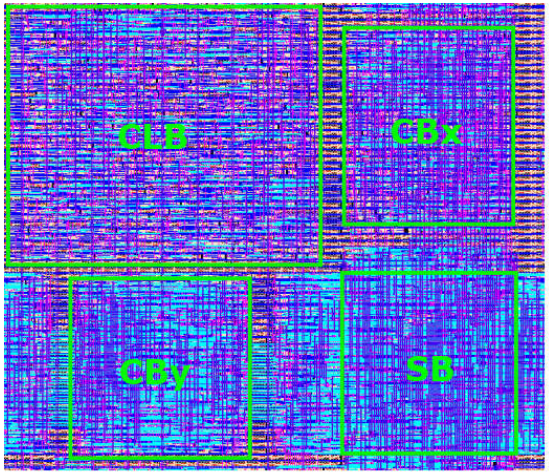
```
module demo(
    clock,
    a,
    b,
    c,
    out
);
    input   clock;
    input   a;
    input   b;
    input   c;
    reg     d;
    output  out;
    // ASSIGN STATEMENTS
    always @(posedge clock)
    begin
        d <= a & b;
    end
    assign out = c | d;
endmodule
```



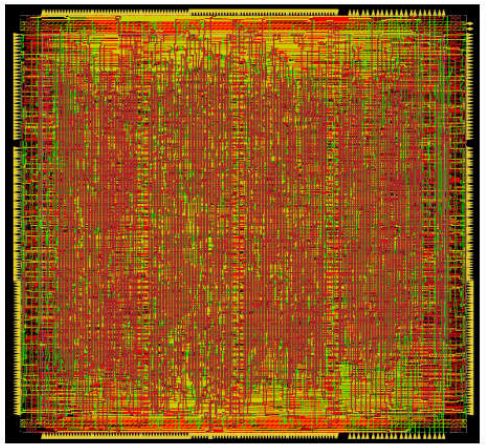
- Verilog-to-bitstream using VPR or Yosys/nextpnr
- Result is a FASM that we translate into the bitstream

FABulous versus OpenFPGA (on Sky130)

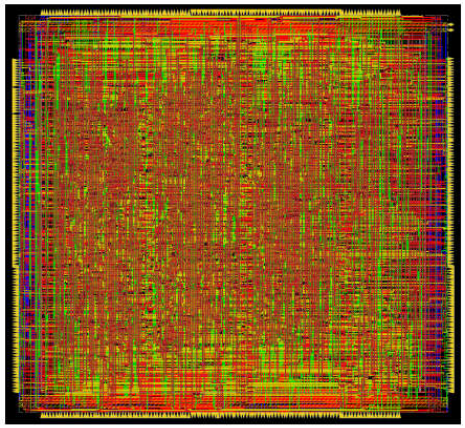




OpenFPGA Shuttle 2
250x217 um2
(+17.4%)



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!
- Shuttle3: we plan an open-everything FPGA with better density than Shuttle2 (Cadance)

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

~1200 MUX2

FABulous Contributors

People:

Nguyen Dao	nguyen.dao@manchester.ac.uk
Jing Li	jing.li@manchester.ac.uk
Khoa Pham	khoa.pham@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@manchester.ac.uk

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