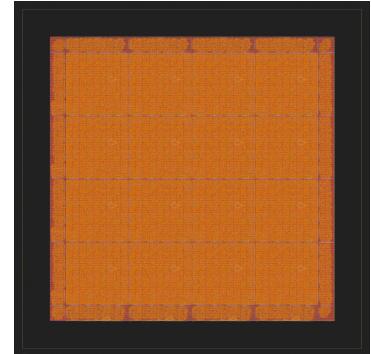


zer|0|

Lessons Learned Implementing an FPGA in 12nm using OpenROAD

Peter Gadfort
Zero ASIC
June 24, 2025



Overview

- SiliconCompiler introduction
- FPGA design
- Physical flow in OpenROAD
- Timing extraction flow in OpenSTA
- Lessons learned

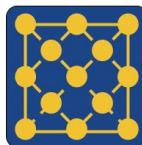
Overview

- SiliconCompiler introduction
- FPGA design
- Physical flow in OpenROAD
- Timing extraction flow in OpenSTA
- Lessons learned

SiliconCompiler Overview

SiliconCompiler is an open source build system that automates translation from source code to silicon.

"MAKE for silicon"



SiliconCompiler

- Python user API
- Flowgraph based execution model
- Drivers for all flow tools (executables)
- Run-time tracking of all actions/metrics

Type	Supported Technologies
Design Languages	C, Verilog , SV, VHDL, Chisel, Migen/Amaranth, Bluespec, MLIR
Simulation Tools	Verilator, Icarus, GHDL, Xyce
Synthesis	Yosys , Vivado, Synopsys, Cadence
ASIC APR	OpenROAD , Synopsys, Cadence
FPGA APR	VPR , nextpnr, Vivado
Layout Viewer	kLayout , OpenROAD , Cadence, Synopsys
DRC/LVS	kLayout, Magic, Synopsys, Siemens
PDKs	sky130, ihp130, gf180, asap7, freepdk45, gf12lp , gf22fdx, intel16

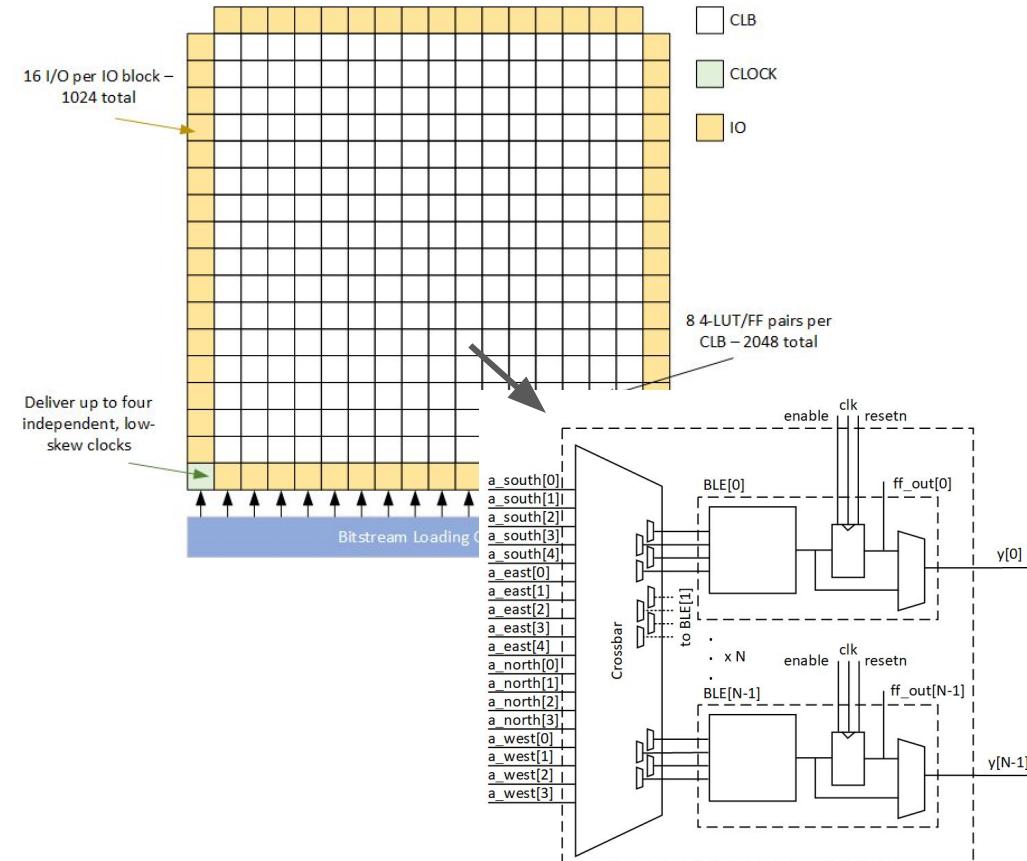
Overview

- SiliconCompiler introduction
- FPGA design
- Physical flow in OpenROAD
- Timing extraction flow in OpenSTA
- Lessons learned

Embedded FPGA (Z1000)



- LUTs: 2048
- Regs: 2048
- IO: 1024
- Clocks: 4
- Size: 1036.8um x 1037.2um
- VPR based rtl-2-bitstream

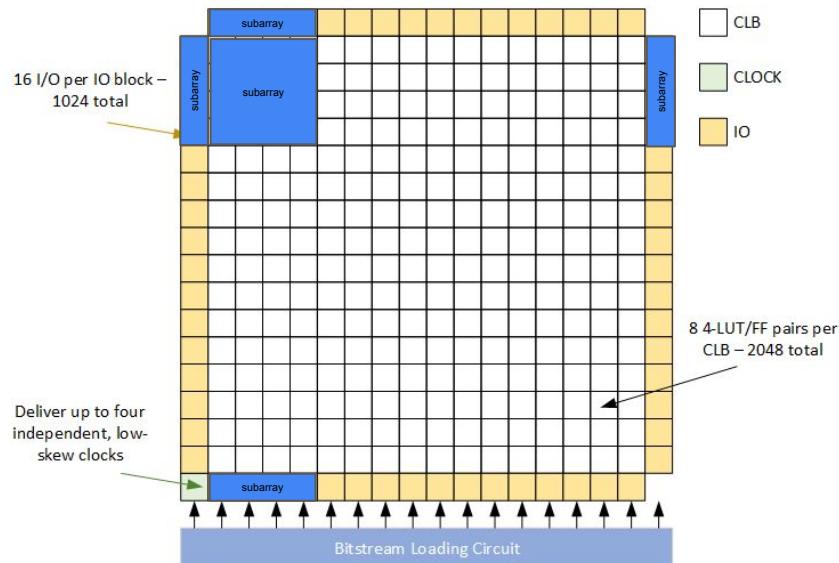


<https://github.com/siliconcompiler/logiklib/tree/main/logiklib/zeroasic/z1000>

Physical Flow Preplanning



- Careful selection of tile size
 - Integer multiple of standard cell site
- Careful placement of pins to ensure abutment of arrays
- Careful construction of the powergrid to ensure continuity with abutment
- Total subarrays: 1 Core, 4 IOs, 4 corners

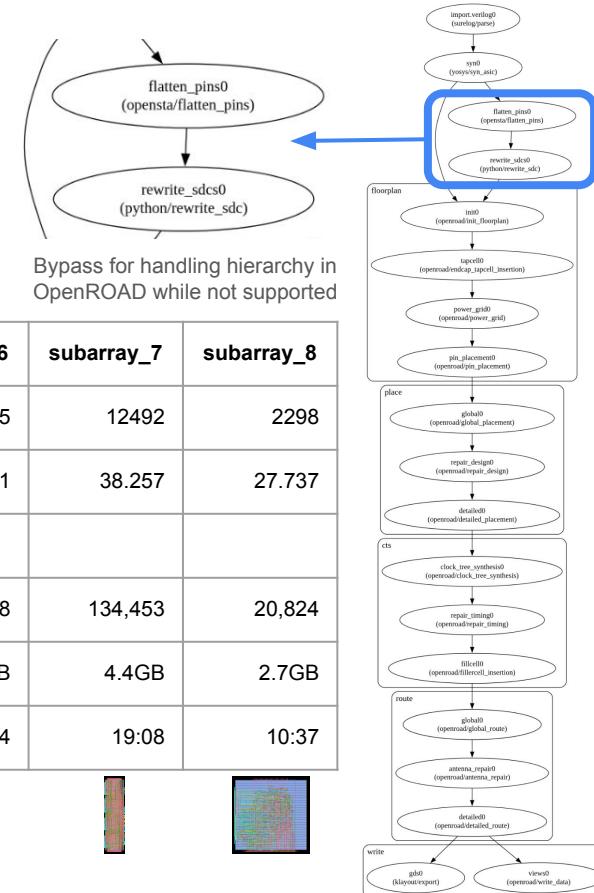


Overview

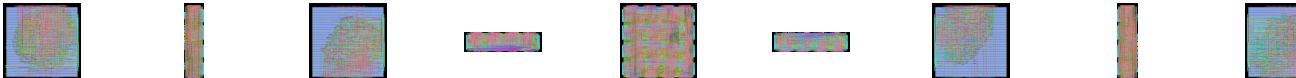
- SiliconCompiler introduction
- FPGA design
- Physical flow in OpenROAD
- Timing extraction flow in OpenSTA
- Lessons learned

Subarray Tile Flow

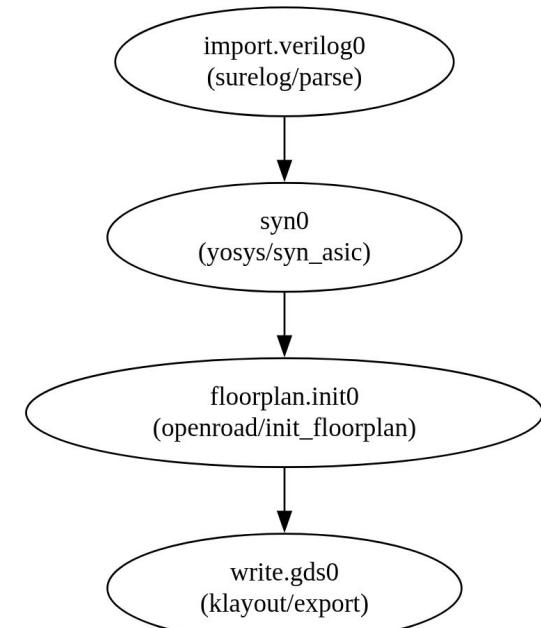
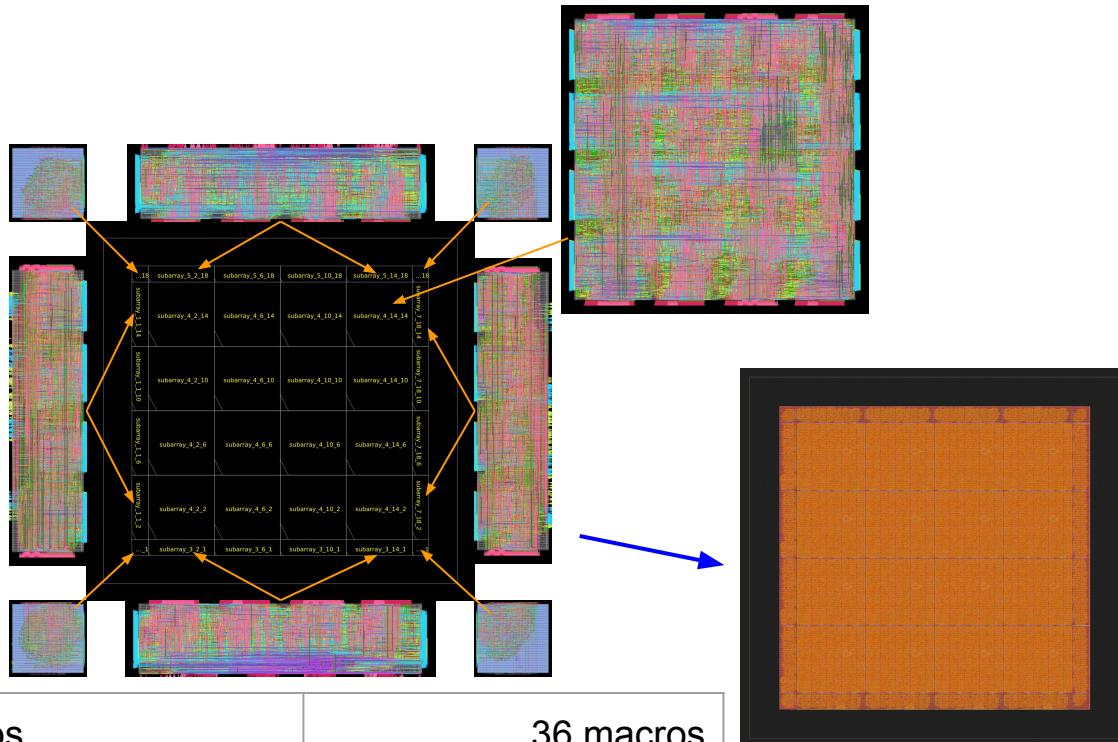
- Needed to mock hierarchy to preserve timing arcs
 - Use dont_touch to avoid modifying hierarchy and modify SDCs to correct for lack of hierarchical ports



	subarray_0	subarray_1	subarray_2	subarray_3	subarray_4	subarray_5	subarray_6	subarray_7	subarray_8
Gates	2916	12508	2295	13152	71686	12448	2295	12492	2298
Utilization	33.822	38.657	27.747	41.694	55.271	38.866	27.741	38.257	27.737
Fmax					1.058GHz				
Wirelength	27,343	146,344	20,509	160,110	656,488	149,909	19,998	134,453	20,824
Memory	2.3GB	4.4GB	1.8GB	4.6GB	17.0GB	4.4GB	2.0GB	4.4GB	2.7GB
Runtime	15:27	25:14	10:22	30:12	1:23:18	19:47	10:24	19:08	10:37



Array Assembly Flow



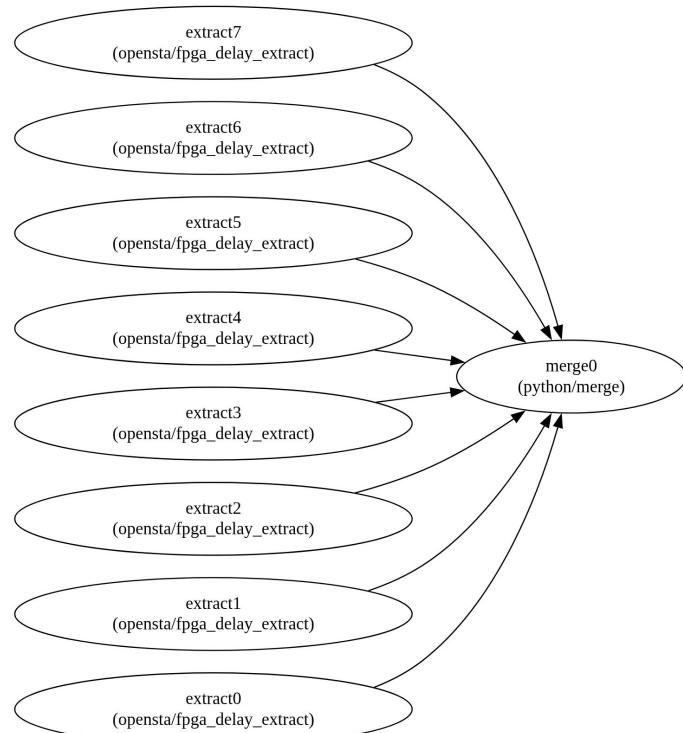
Overview

- SiliconCompiler introduction
- FPGA design
- Physical flow in OpenROAD
- Timing extraction flow in OpenSTA
- Lessons learned

FPGA Timing Extraction for VPR

- Extract timing arcs required for VPR timing enablement

	subarray_0	subarray_1	subarray_2	subarray_3	subarray_4
Arcs	1398	5652	1048	5652	29112
Memory	520.0MB	535.3MB	518.2MB	536.3MB	2.2GB
Runtime	0:30	4:15	0:24	4:36	14:16:00
	subarray_5	subarray_6	subarray_7	subarray_8	Total
Arcs	5502	1048	5502	1198	56112
Memory	534.0MB	518.3MB	534.2MB	518.5MB	N/A
Runtime	3:52	0:24	3:54	0:25	14:34:23



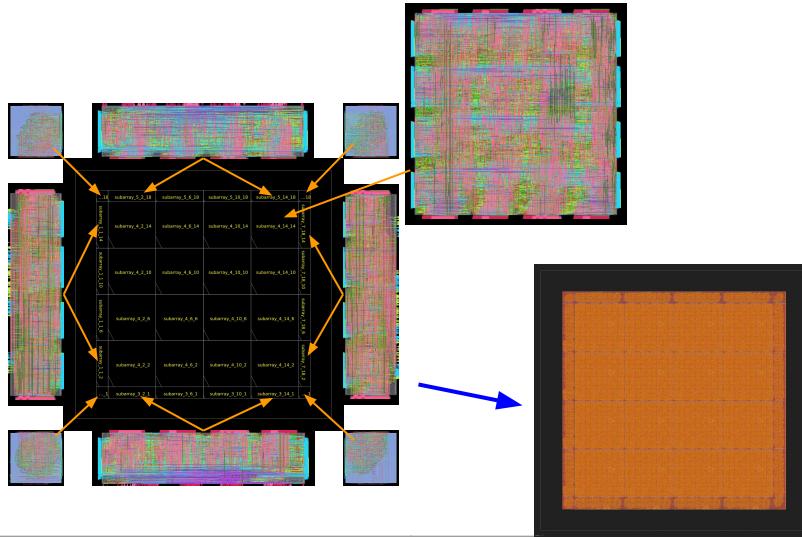
Overview

- SiliconCompiler introduction
- FPGA design
- Physical flow in OpenROAD
- Timing extraction flow in OpenSTA
- Lessons learned

Lessons Learned

- Hierarchy is your friend
 - Need to improve support, openroad loses all hierarchical pins complicating timing signoff
- Embrace parallel
 - Timing extraction would have taken ~100 hours if limited to one instance
- Open source gaps for GF12:
 - GF12 DRC, LVS
 - GF12 PEX (requires reference pex to build OpenROAD deck)

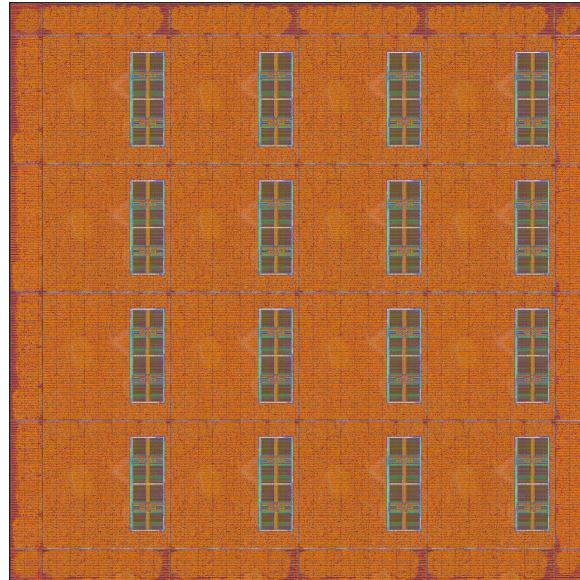
Questions?



Gates	1.360M
Fmax	1.058GHz
Wirelength	12.95m
Size	1036.8um x 1037.2um
Runtime	3:47:15 + 14:34:23 = 18:21:38

What is next?

- Integrate BRAMs and DSPs

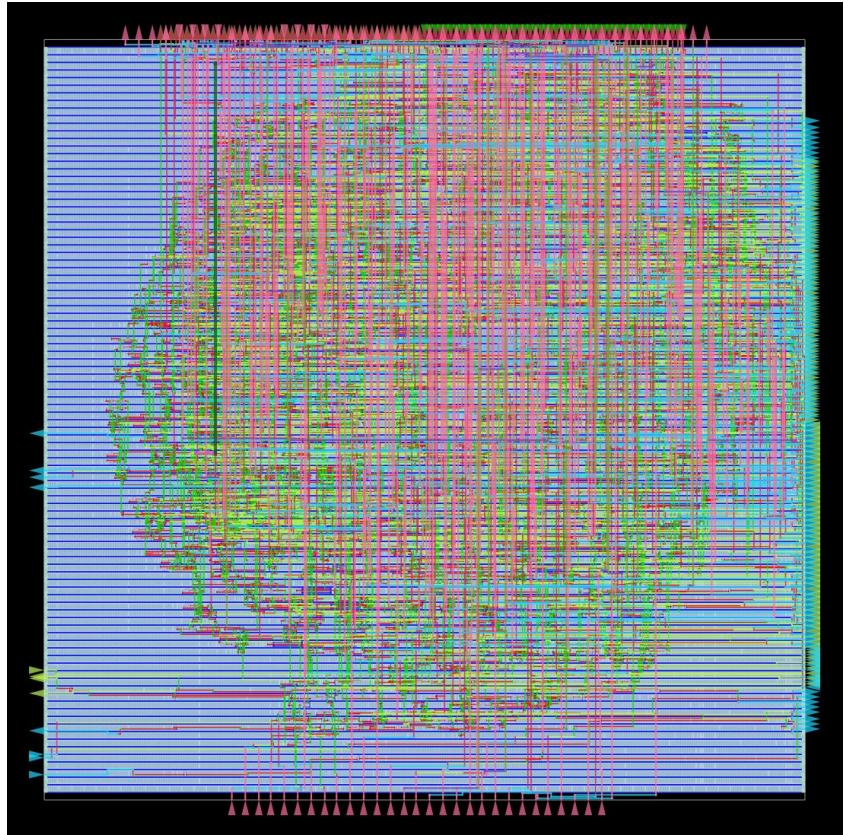
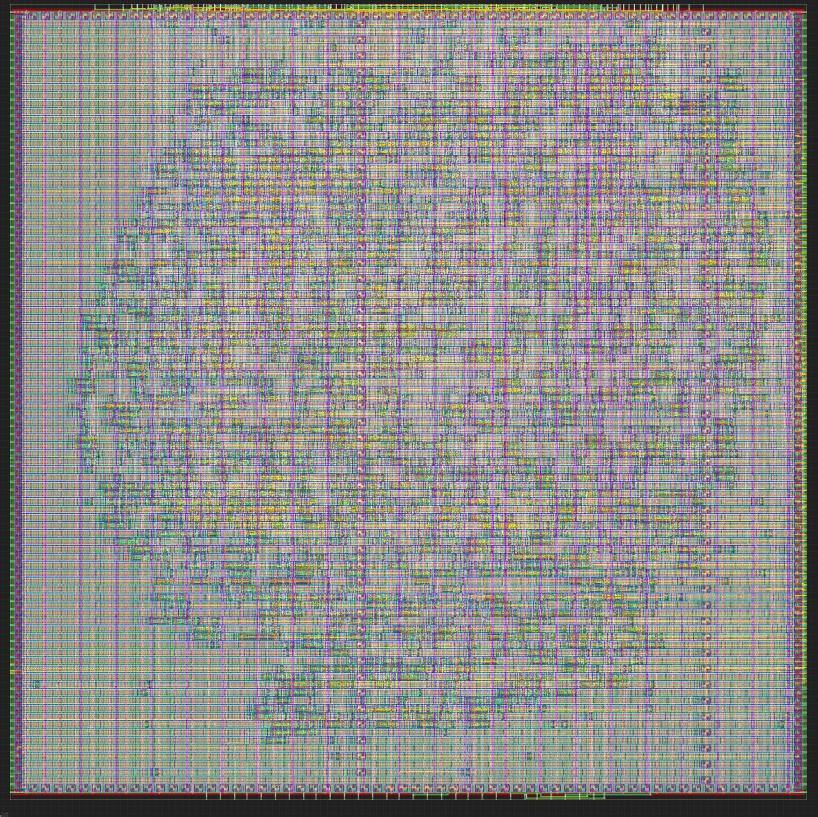


VPR Counter Example Timing

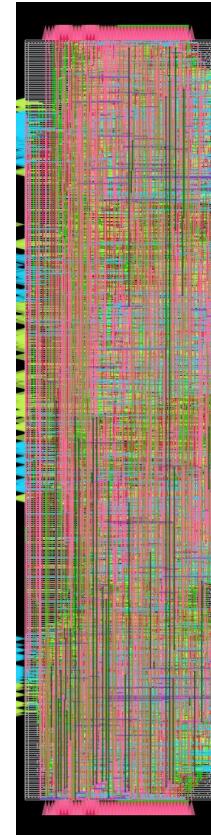
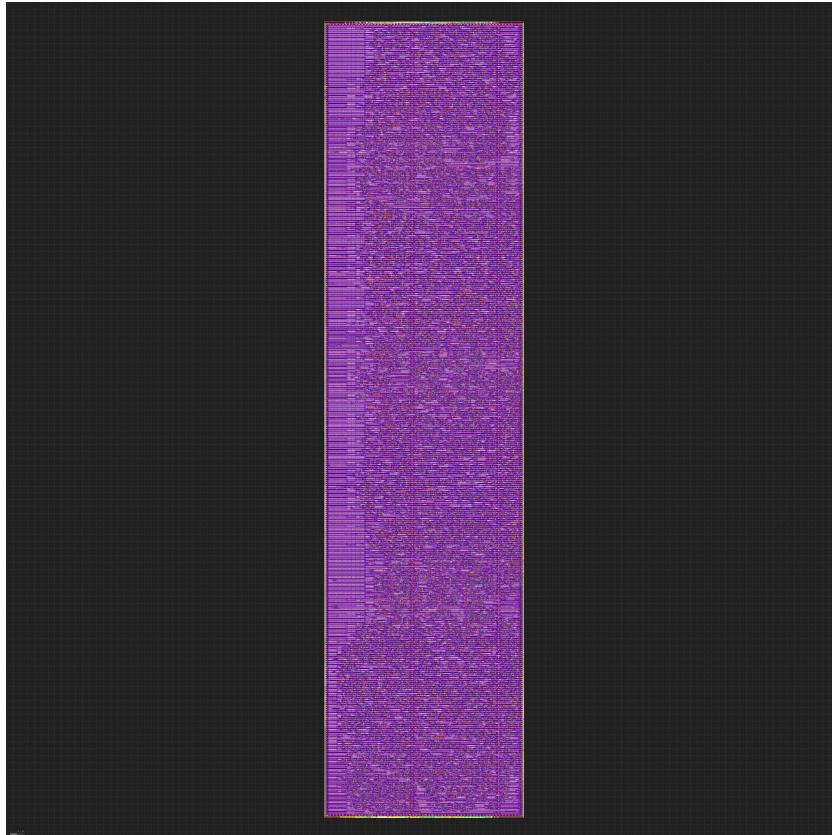
```
#Path 1
Startpoint: y[0].Q[0] (dff clocked by clk)
Endpoint   : y[6].D[0] (dff clocked by clk)
Path Type  : setup
```

Point	Incr	Path
clock clk (rise edge)	0.000	0.000
clock source latency	0.000	0.000
clk.inpad[0] (.input)	0.000	0.000
y[0].clk[0] (dff)	0.112	0.112
y[0].Q[0] (dff) [clock-to-output]	0.118	0.230
\$abc\$494\$new_n22.in[1] (.names)	0.115	0.346
\$abc\$494\$new_n22.out[0] (.names)	0.128	0.473
\$abc\$494\$new_n21.in[3] (.names)	1.448	1.921
\$abc\$494\$new_n21.out[0] (.names)	0.128	2.048
\$abc\$494\$auto\$rtlil.cc:2985:MuxGate\$371.in[2] (.names)	1.101	3.150
\$abc\$494\$auto\$rtlil.cc:2985:MuxGate\$371.out[0] (.names)	0.038	3.188
y[6].D[0] (dff)	0.000	3.188
data arrival time		3.188
clock clk (rise edge)	1.000	1.000
clock source latency	0.000	1.000
clk.inpad[0] (.input)	0.000	1.000
y[6].clk[0] (dff)	0.112	1.112
clock uncertainty	0.000	1.112
cell setup time	-0.059	1.054
data required time		1.054
data required time	1.054	
data arrival time		-3.188
slack (VOLATED)		-2.134

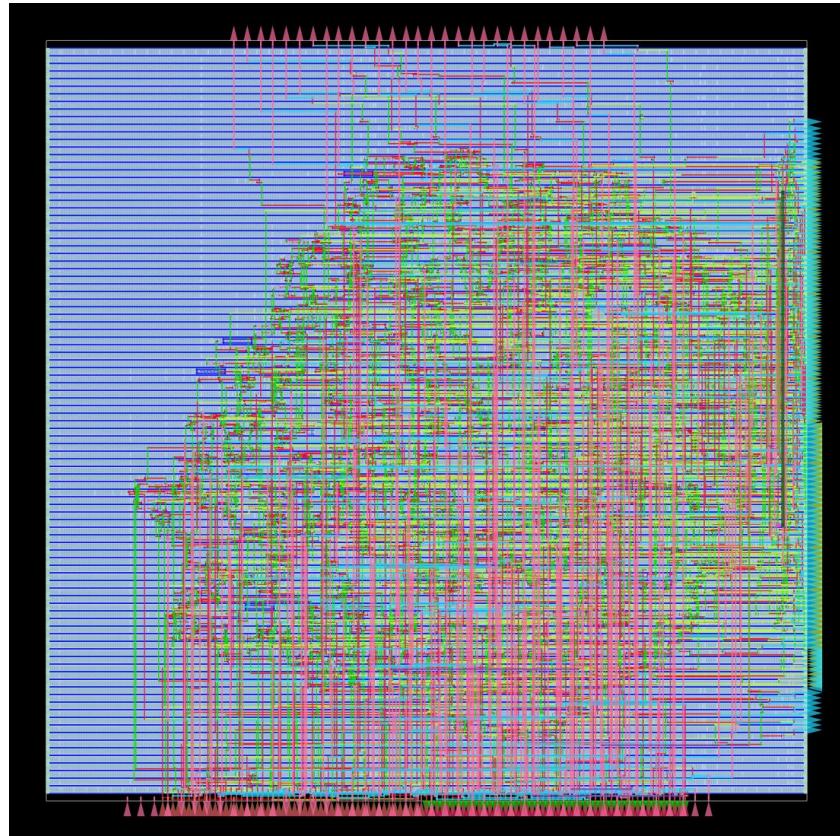
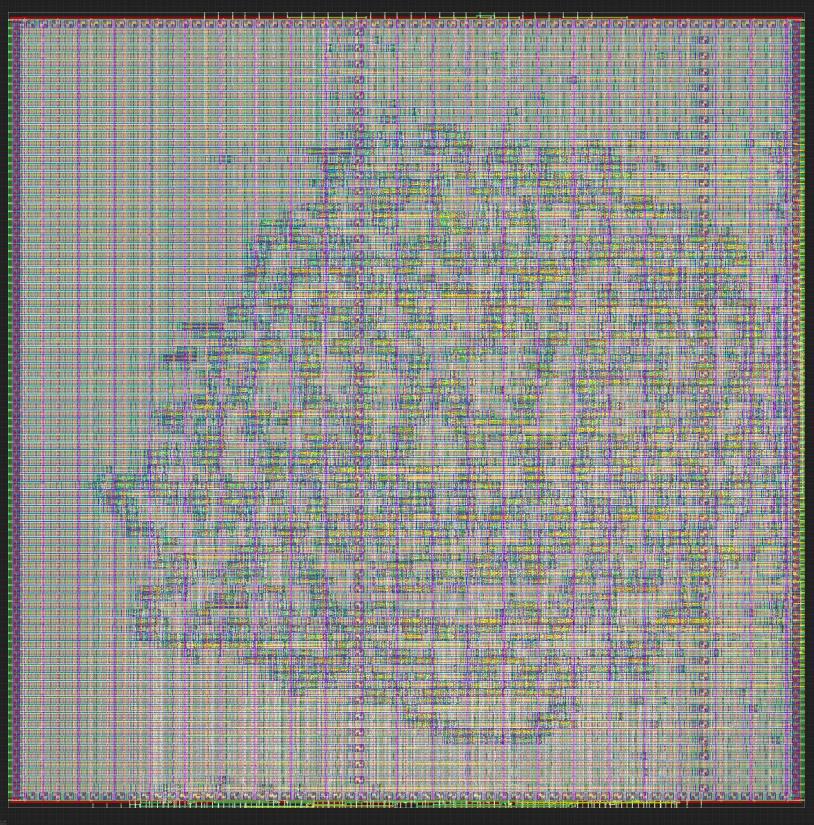
Images - subarray_0 - lower left



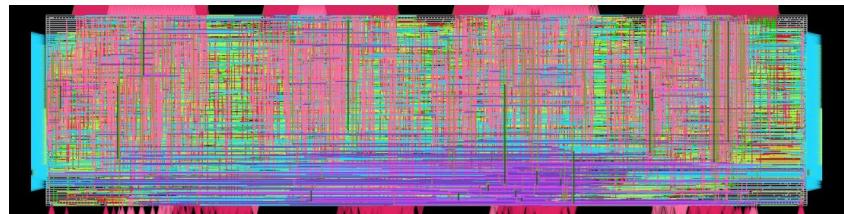
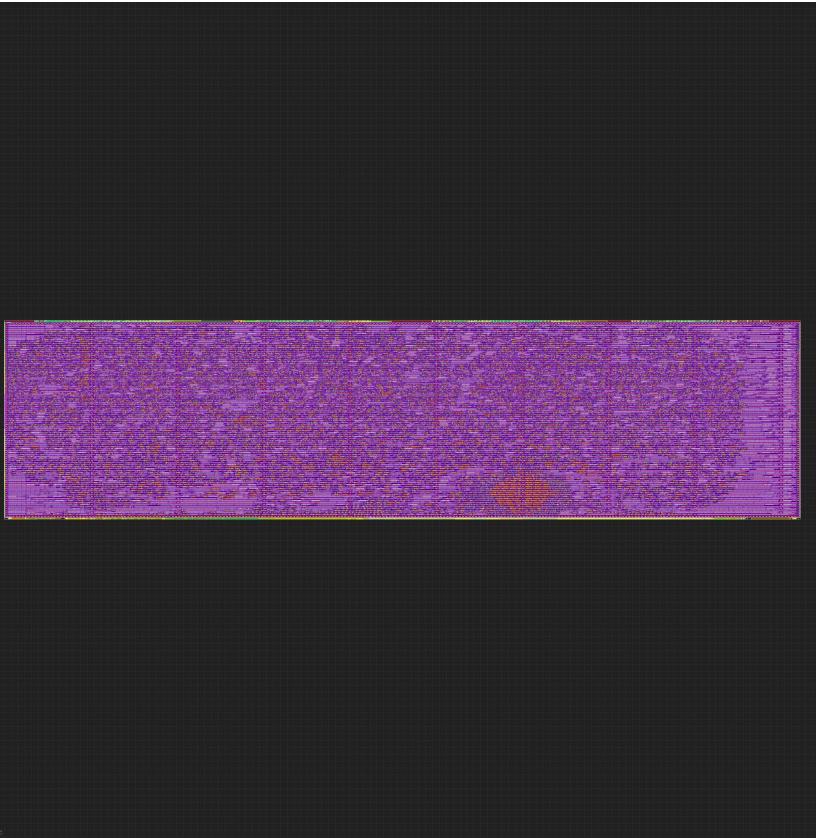
Images - subarray_1 - left



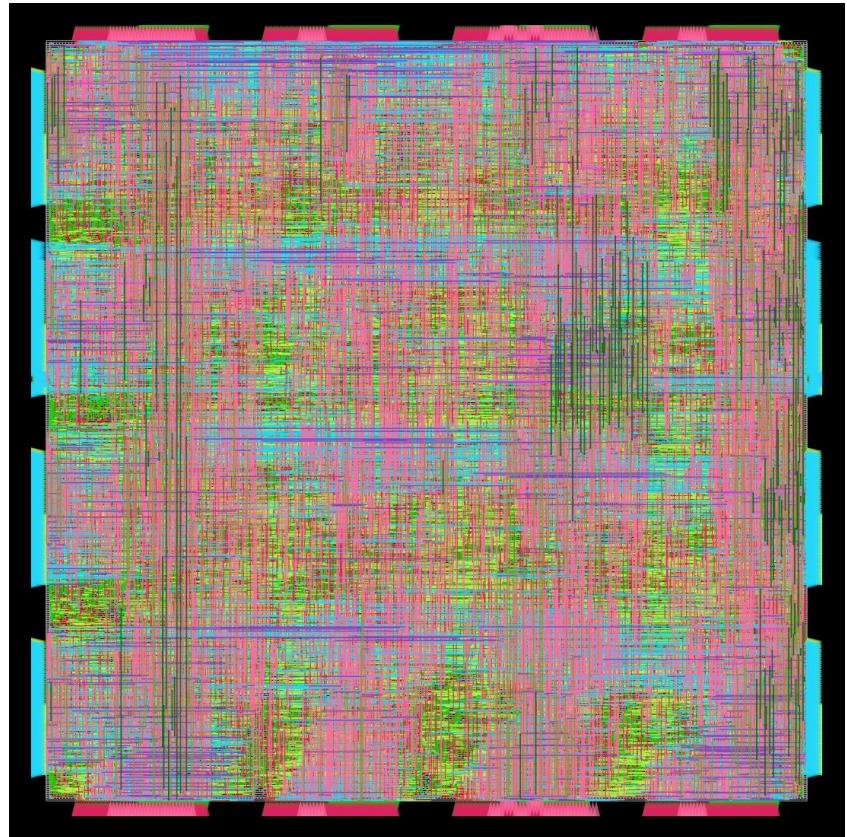
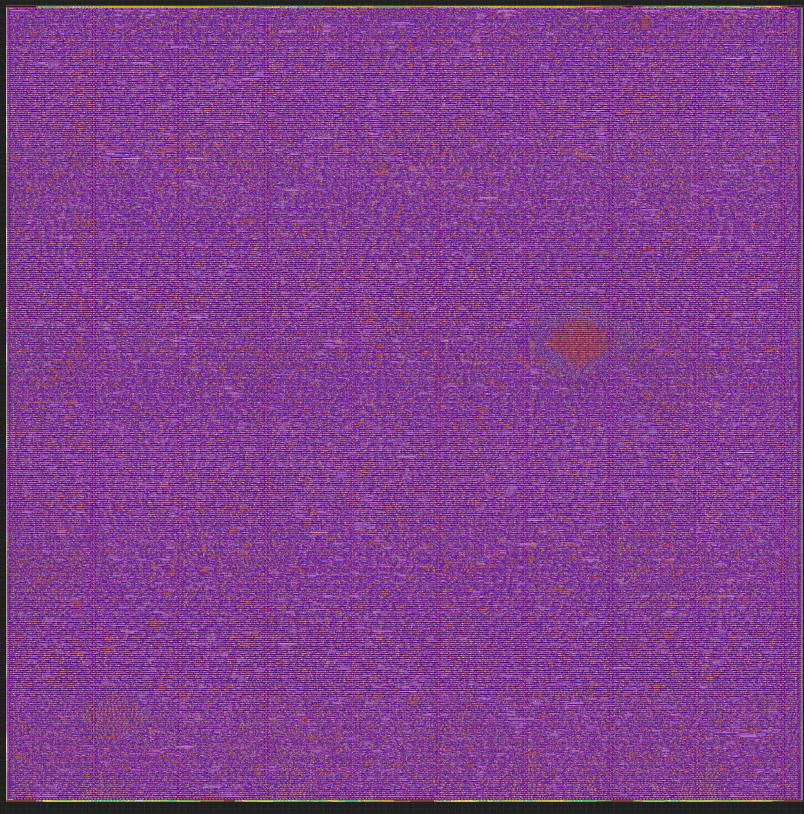
Images - subarray_2 - upper left



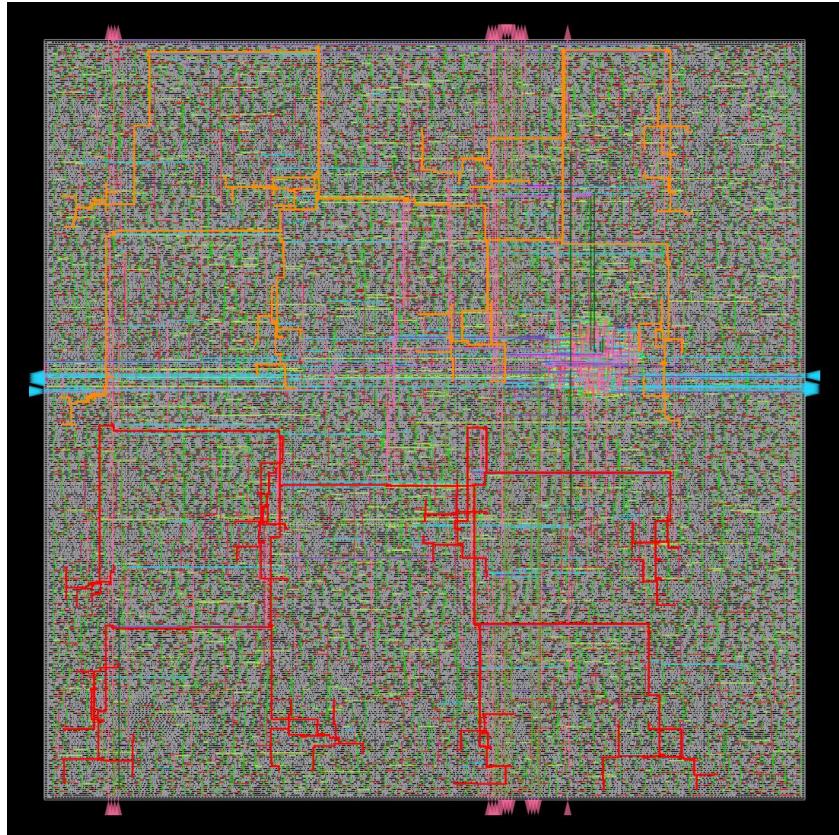
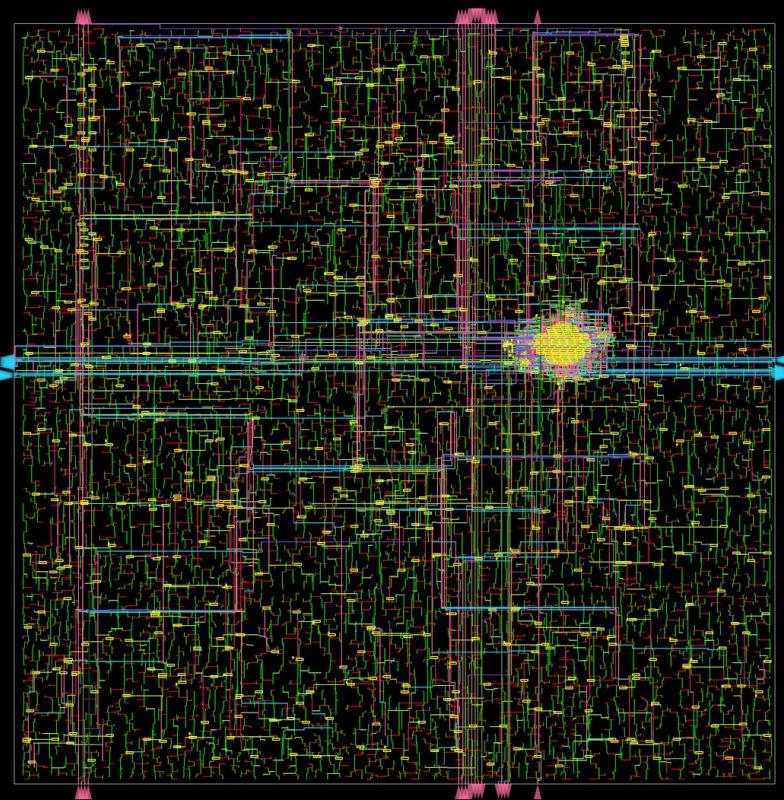
Images - subarray_3 - bottom



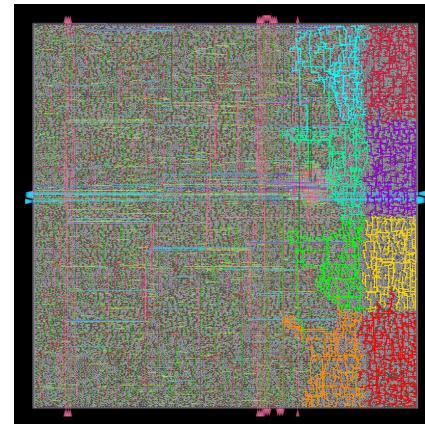
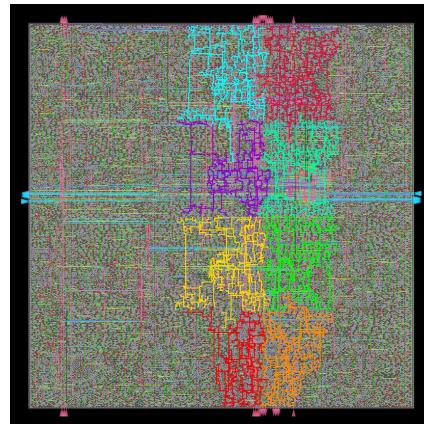
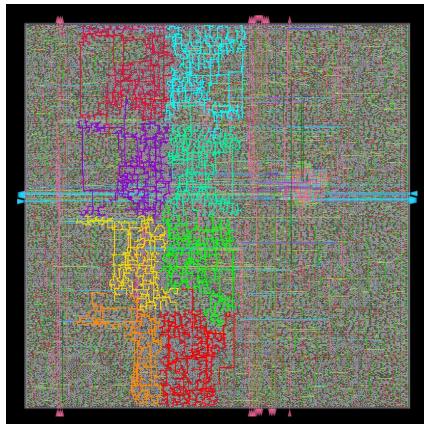
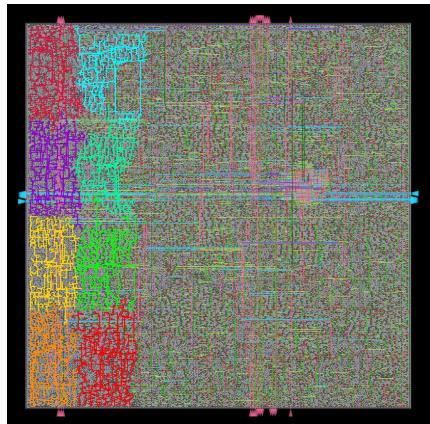
Images - subarray_4 - core



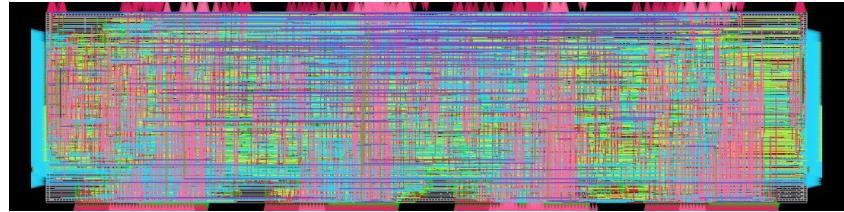
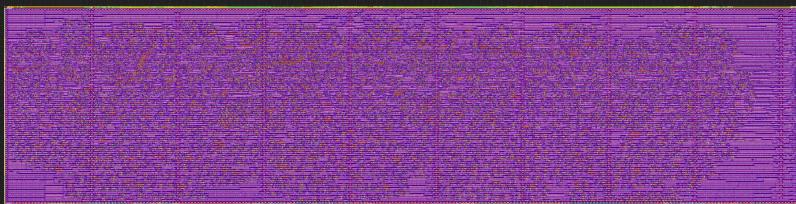
Images - subarray_4 - core - clocks (1/2)



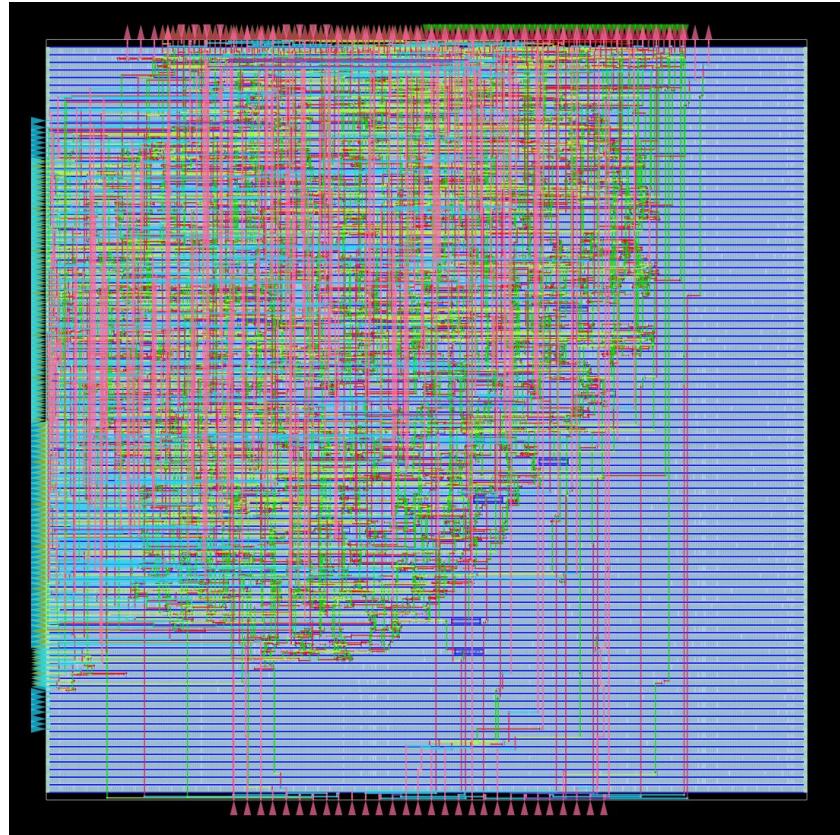
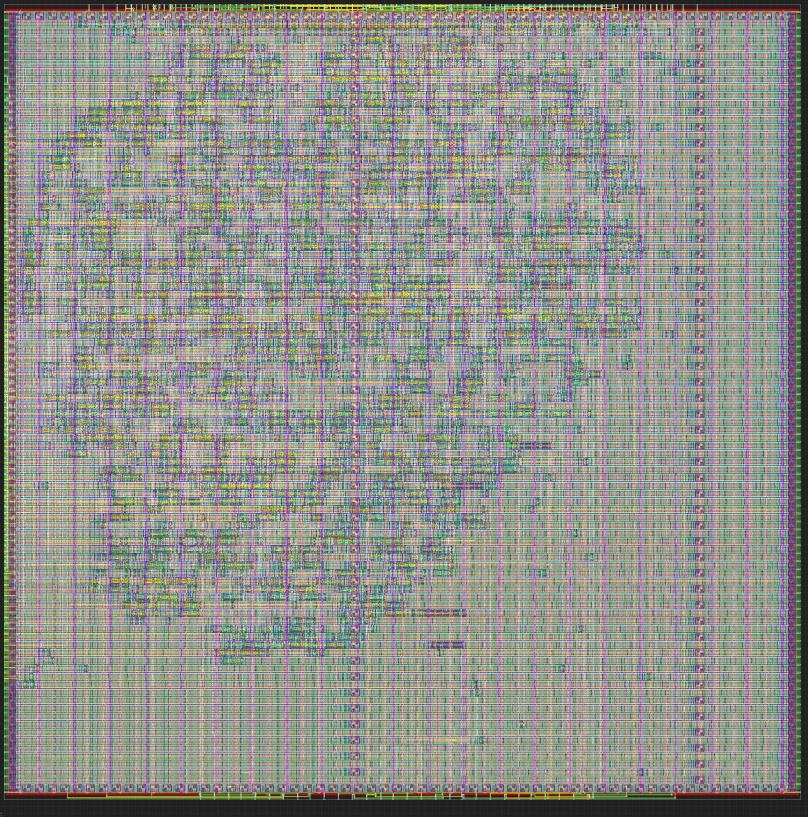
Images - subarray_4 - core - clocks (2/2)



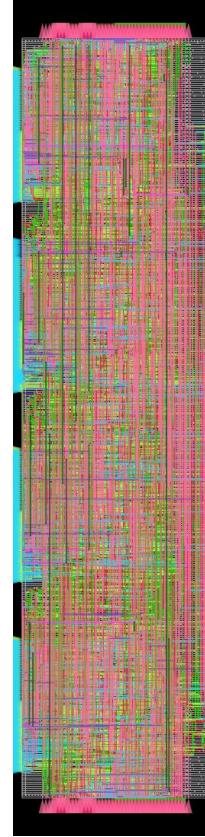
Images - subarray_5 - top



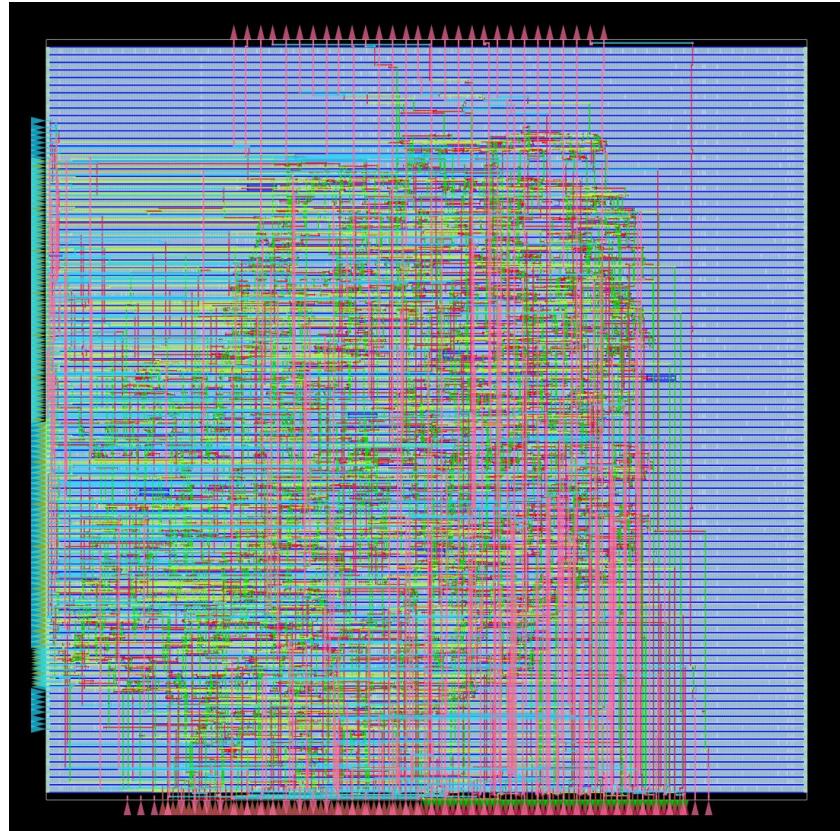
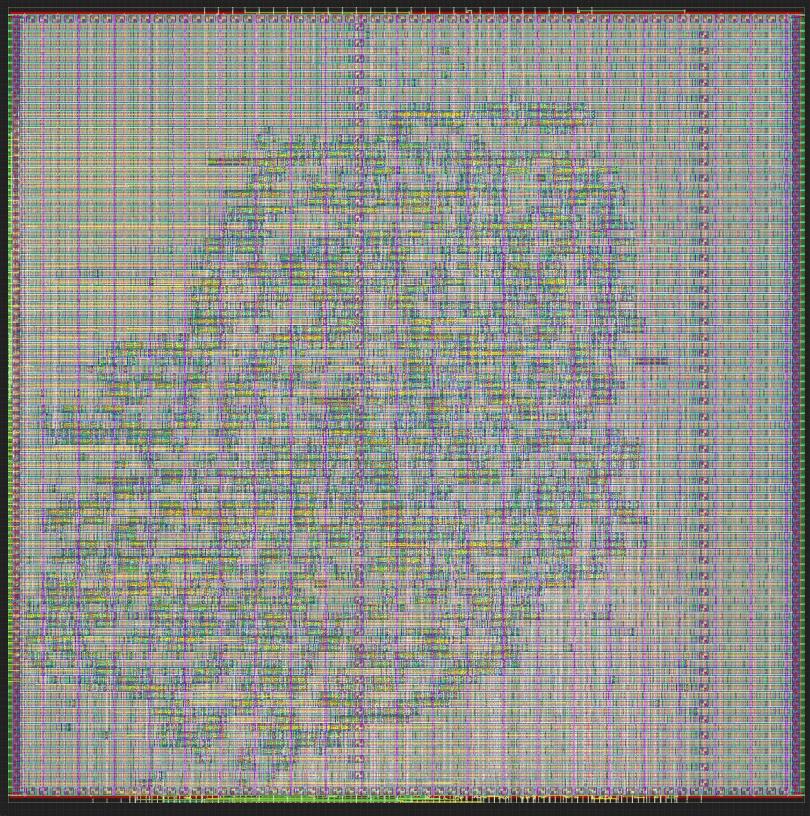
Images - subarray_6 - bottom right



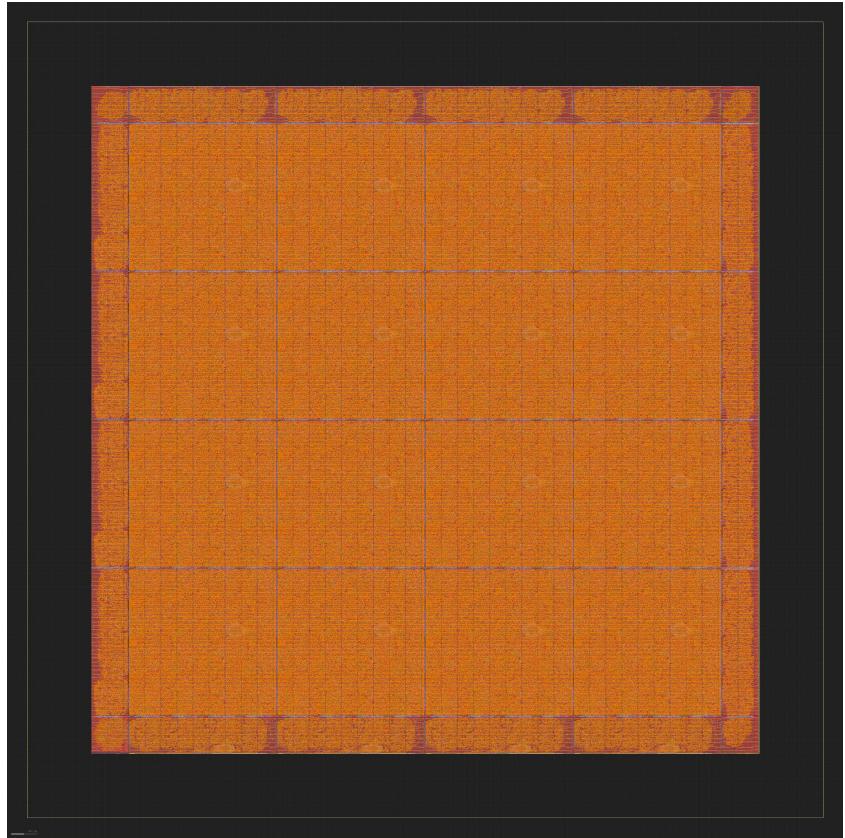
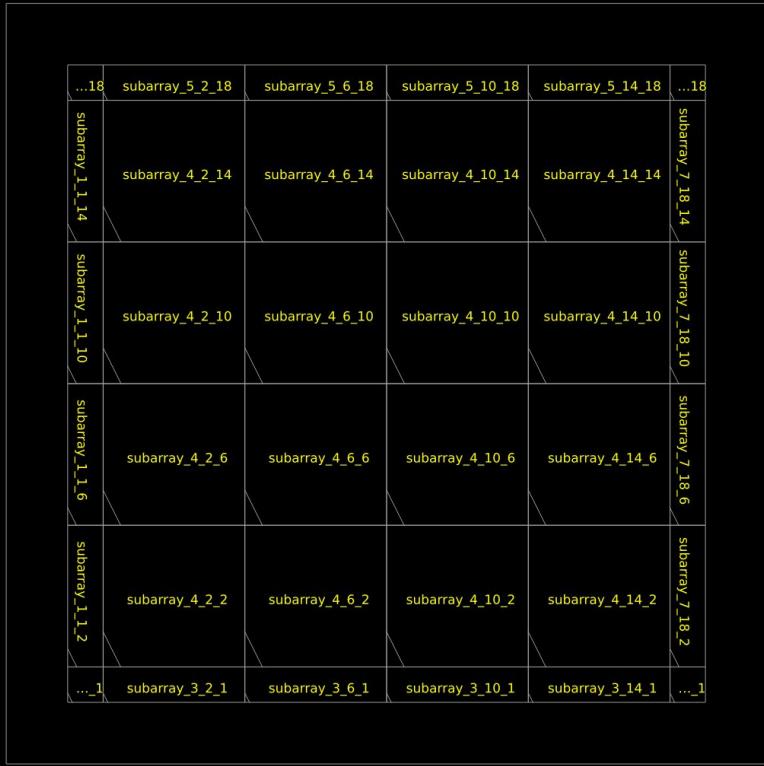
Images - subarray_7 - right



Images - subarray_8 - top right

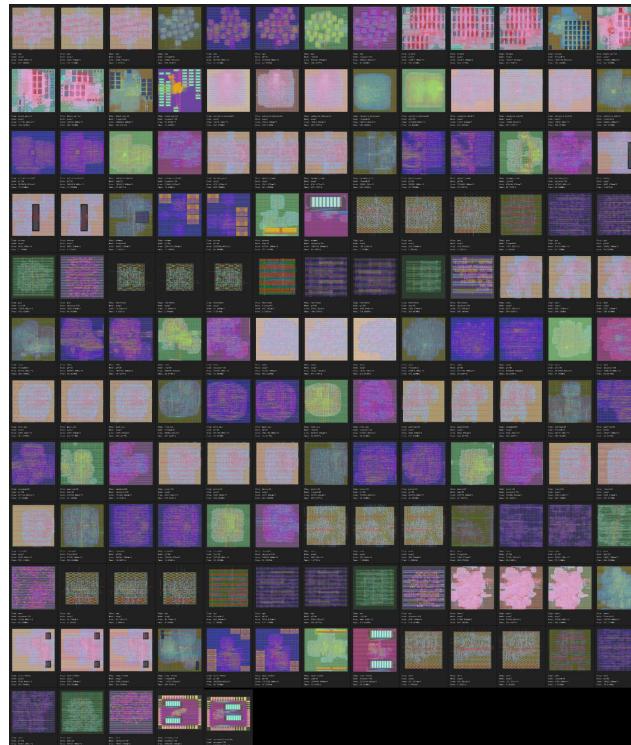


Images - full assembly





Dashboard



<https://www.zeroasic.com/dashboard>



Physical Stats

	subarray_0	subarray_1	subarray_2	subarray_3	subarray_4	subarray_5	subarray_6	subarray_7	subarray_8	assembly
Gates	2916	12508	2295	13152	71686	12448	2295	12492	2298	N/A
Utilization	33.822	38.657	27.747	41.694	55.271	38.866	27.741	38.257	27.737	100
Fmax	2.582G	2.214G	2.598G	2.101G	1.058G	2.230G	2.603G	2.228G	2.577G	1.058G
Wirelength	27343	146344	20509	160110	656488	149909	19998	134453	20824	N/A
Memory	2.315G	4.431G	1.838G	4.559G	17.027G	4.421G	1.998G	4.427G	2.663G	2.716G
Runtime	15:27	25:14	10:22	30:12	1:23:18	19:47	10:24	19:08	10:37	2:41

	Total
Gates	1.360M
Fmax	1.058G
Wirelength	12.95m
Runtime	3:47:15

Timing Extraction Stats

	subarray_0	subarray_1	subarray_2	subarray_3	subarray_4	subarray_5	subarray_6	subarray_7	subarray_8	Total
Arcs	1398	5652	1048	5652	29112	5502	1048	5502	1198	56112
Memory	520.094M	535.250M	518.227M	536.285M	2.243G	534.000M	518.309M	534.215M	518.457M	N/A
Runtime	30	04:15	24	04:36	14:16:00	03:52	24	03:54	25	14:34:23

Comparison with ORFS

Feature	SiliconCompiler	ORFS
Interface Language	Python	Make
Flow	Graph Based	Fixed
Parallel scheduling	yes	no
Package handling	yes	no
Tool build scripts provided	yes	yes
Cloud/distributed runnable	yes	no
Frontend languages	Verilog, SystemVerilog, VHDL, Bluespec*, Chisel*, C*, MLIR*	Verilog, SystemVerilog
Metrics	yes, recorded in python schema	yes, recorded in files/logs