Zero ASIC

Removing the barrier to custom silicon

Lessons learned from using SiliconCompiler in advanced node tapeouts

Open Source Birds-Of-Feather, DAC 2023

Presented by Andreas Olofsson
andreas@zeroasic.com

SiliconCompiler is an open source cloud native build system that automates reproducible compilation from source code to silicon.

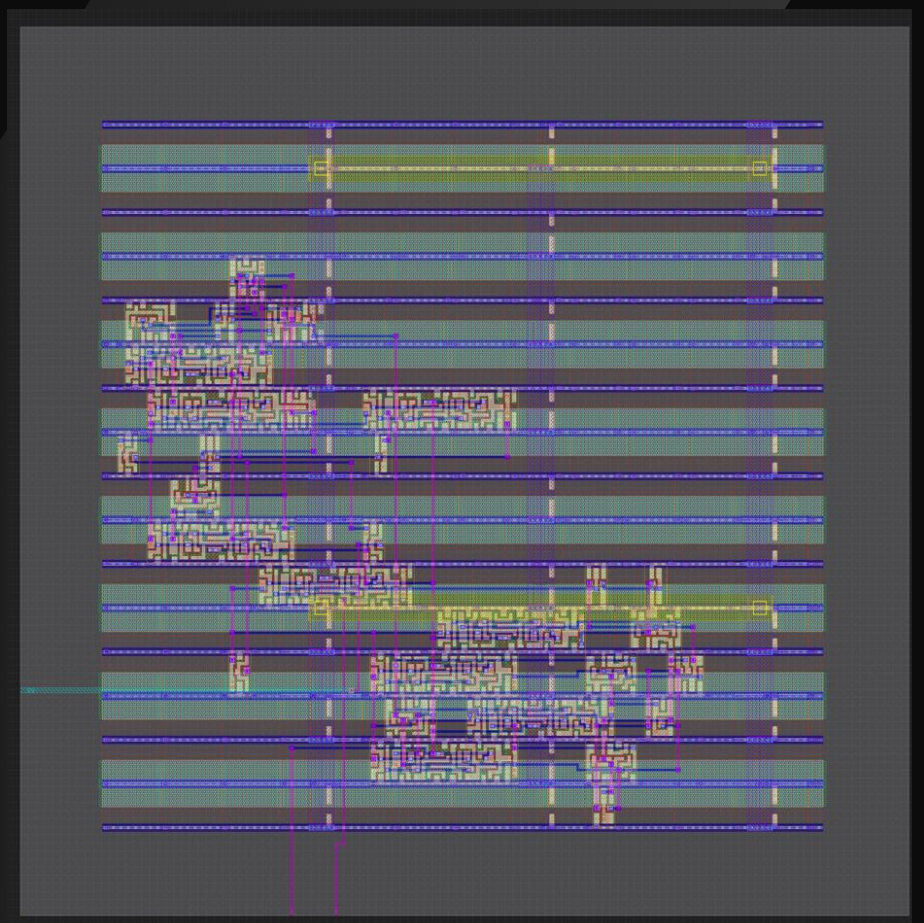
“MAKE for silicon”

DEMO TIME!

```
$ pip install --upgrade siliconcompiler
```

```
$ sc -target asic_demo -remote
```

You won't get it until you try it…



```
Chip: heartbeat
Node: skywater130
Area: 1941.860um^2
Fmax: 219.197MHz
```

https://github.com/siliconcompiler

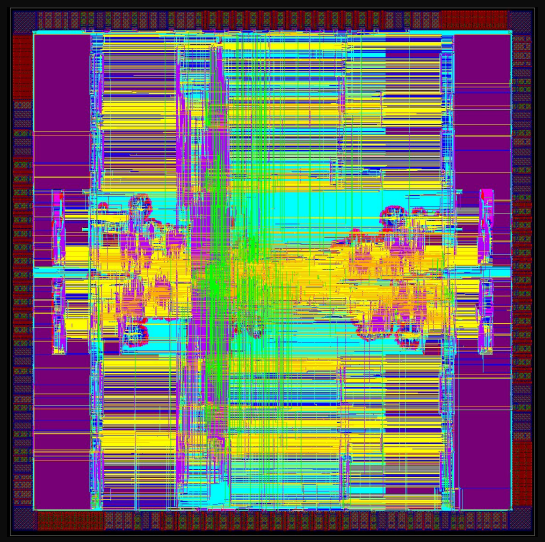
Under The Hood: Surelog,Yosy, Openroad, Klayout, sky130, Slurm (+lots of Python

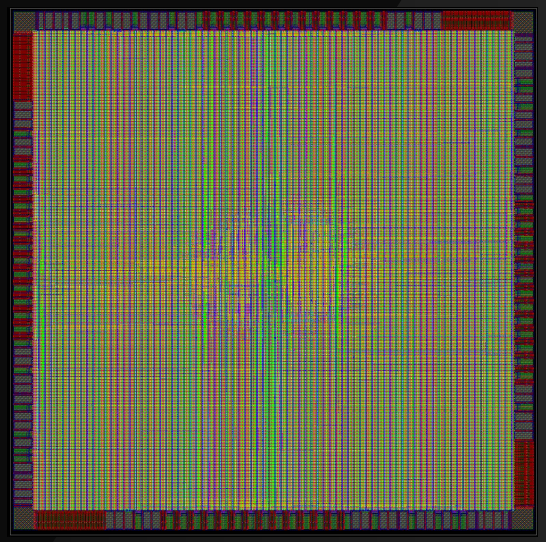
Designer view of SiliconCompiler

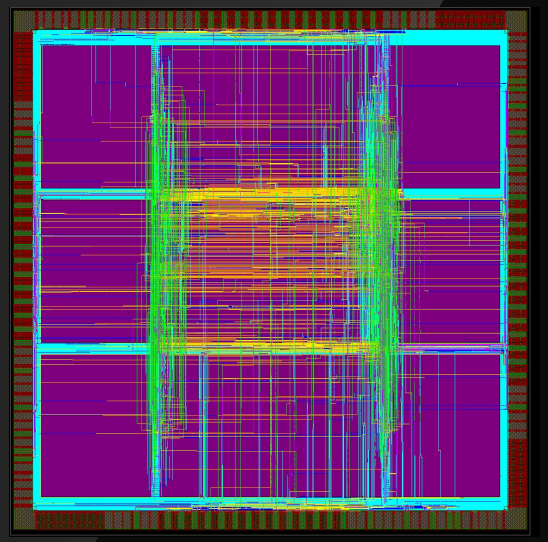
```
2
3   import siliconcompiler                          # import python package
4
5   def main():
6       chip = siliconcompiler.Chip('heartbeat')    # create chip object
7       chip.input('heartbeat.v')                   # define list of source files
8       chip.input('heartbeat.sdc')                 # set constraints file
9       chip.load_target("freepdk45_demo")          # load predefined target
10      chip.run()                                  # run compilation
11      chip.summary()                              # print results summary
12      chip.show()                                 # show layout file
```

- Relies on a standard schema that can describe any chip design flow
- Permissive open source: (https://github.com/siliconcompiler)
- Extensively documented (https://docs.siliconcompiler.com/en/latest/)

3 Tapeouts in 8 Weeks (Sept-Oct 2022)







	GOTLAND	MAUI	KODIAK
TYPE	CPU	FPGA	MEMORY
SIZE	2 x 2 mm	2 x 2 mm	2 x 2 mm
ORIGIN	UCB - ROCKET	ZA	ZA
METRIC	Quad Core RV64GC CPU	–	3MB
DESIGNERS	2	2	2
WALL TIME	< 8 weeks	< 8 weeks	< 8 weeks
RUN TIME	< 24hrs	< 24hrs	< 24hrs

Our Agile Chip Design Team (Sept 2022)

	Andreas Olofsson	Dr. Steven Herbst	Dr. Peter Grossman	Benazeer Noorani	Wenting Zhang	Noah Moroze	William Ransohoff
Experience	27	13	13	15	3	3	9
Role	“The Architect”	“The Analyst”	“Tall thin designer”	“The Expert”	“Tall thin designer”	“Full stack”	“Full stack”
Scope	Architecture, design infrastructure	Emulation, verification, software	FPGA	Implementation, floorplanning, signoff	CPU	Reference flows, SC	Cloud, SC

Agile chip design is fundamentally different from traditional waterfall chip design and requires different people and tools.

Playing Silicon Compiler Tapeout Bingo

	EDA-A	EDA-B	EDA-C	FOSS
INTEL16	2	0	2	1
GF12LP	0	4	3	0
SKY130	0	0	0	1

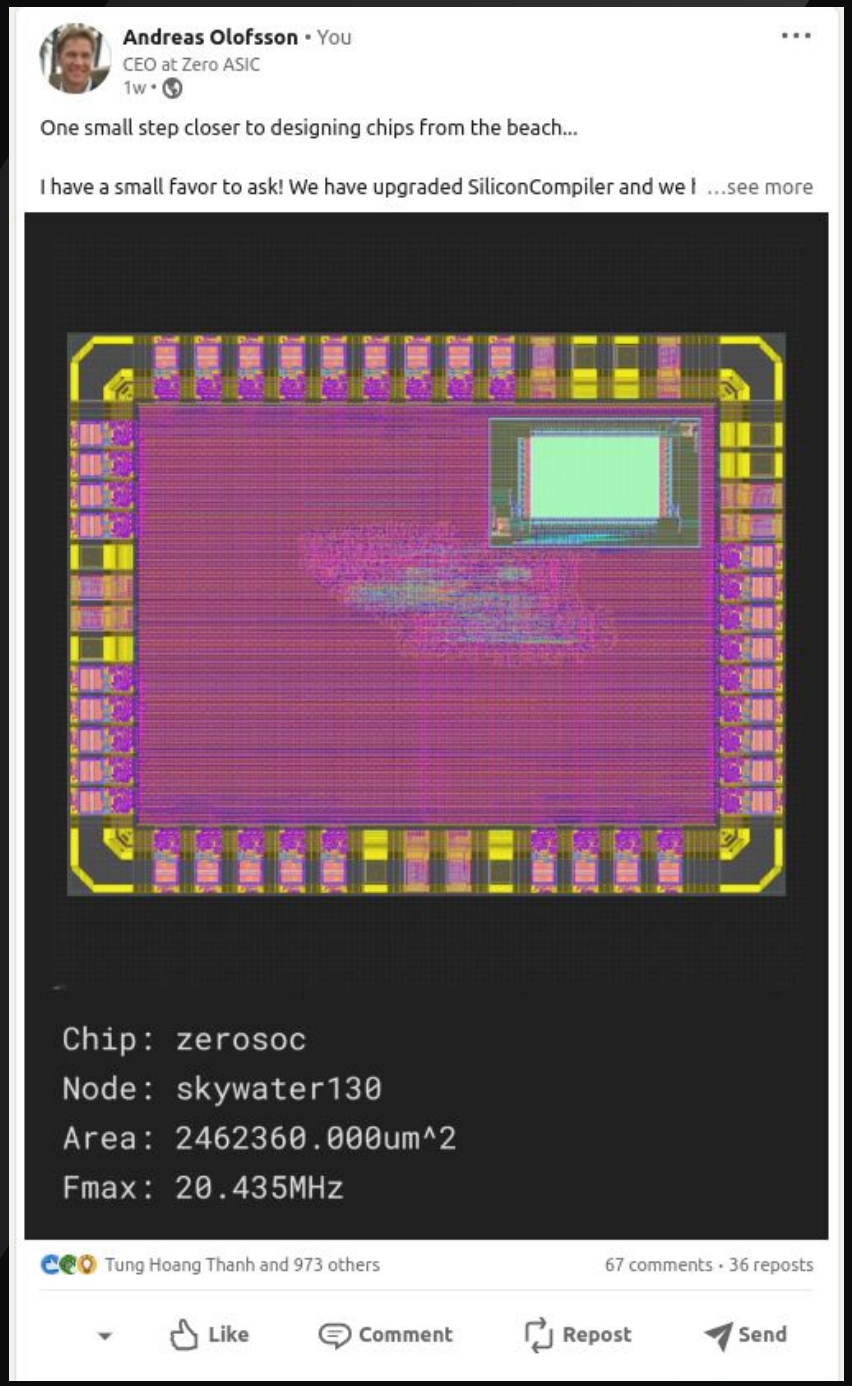
Free (as in Free Beer) Compilation

- No money barrier (free)
- No registration barrier
- No time barrier

....459 compilations in first week

```
$ pip install --upgrade siliconcompiler
```

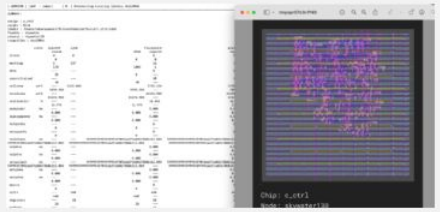
```
$ sc -target asic_demo -remote
```



Testimonials

Omkar Padaki • 1st
MS-CE @ Texas A&M University | SiFive
1w

Worked flawlessly on my design !



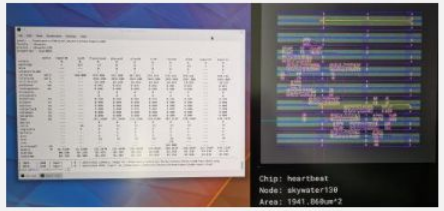
Like · 4 | Reply

SAYED MAHAMUD • 1st
Custom Layout Design | IC Mask Design | Ser

Wow, that's impressive!

Peer Schmitt • 1st
ADICSYS is the Avant-Garde of technology node indepent FPGAs.

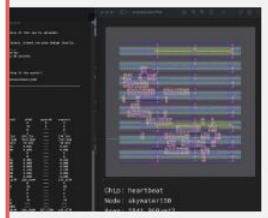
Success from ubuntu 18_04 inside virtualbox



Like | Reply

Anıl Çelebi, PhD • 1st
Associate Professor at Kocaeli University, Managing Partner at KuanTek
1w

irst ASIC flow i have ever tried si
S. 👋



Rahul B. • 2nd
Senior Engineer at Tenstorrent | Teaching VLSI at quicksilicon.in
1w (edited)

Tried the verilog solution for the following problem and it worked like a charm!
https://quicksilicon.in/course/rtl-design/module/fifo-flush

PS: Tested it on MacOS

Like · 16 | Reply · 5 Replies

1w

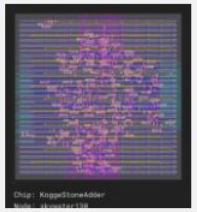
Reply

Prashant Pandey (He/Him) • 2nd
Student at Indian Institute of Technology, Guwahati
7h

Implemented an N-bit Kogge-Stone adder in Verilog and performed its ASIC flow for N = 64. It worked like a charm!

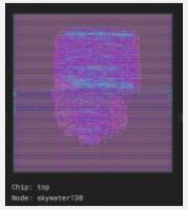
Thanks Zero ASIC!

…see more



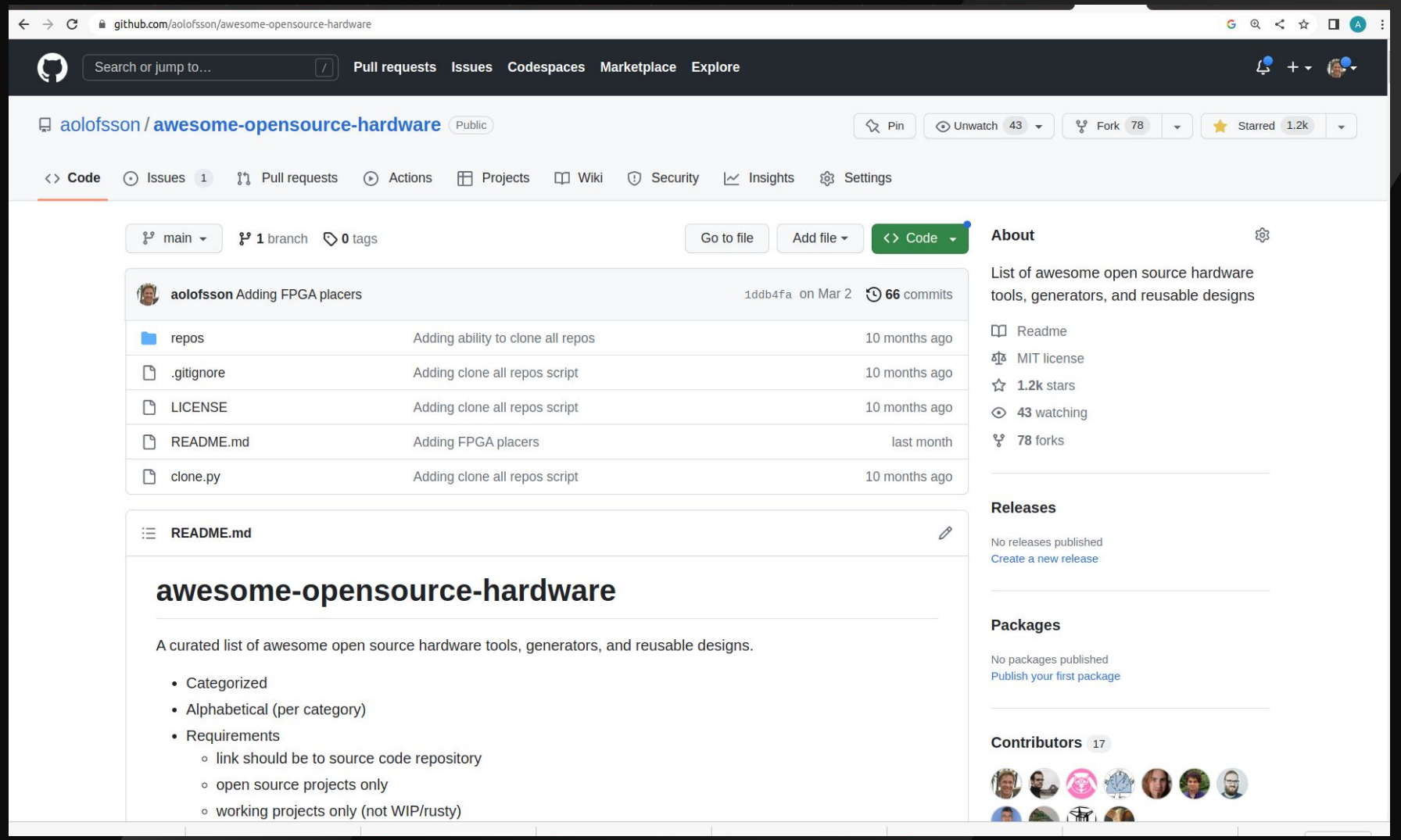
Steve Hoover (He/Him) • 1st
Founder of Redwood EDA
1w

Good timing on this announcement. We're just wrapping up a workshop on CPU design for FPGAs now (https://makerchip.com/sandbox?tabs=Courses), and students were able to also implement their designs for ASIC! Here's a RISC-V CPU:



Like · 11 | Reply · 1 Reply

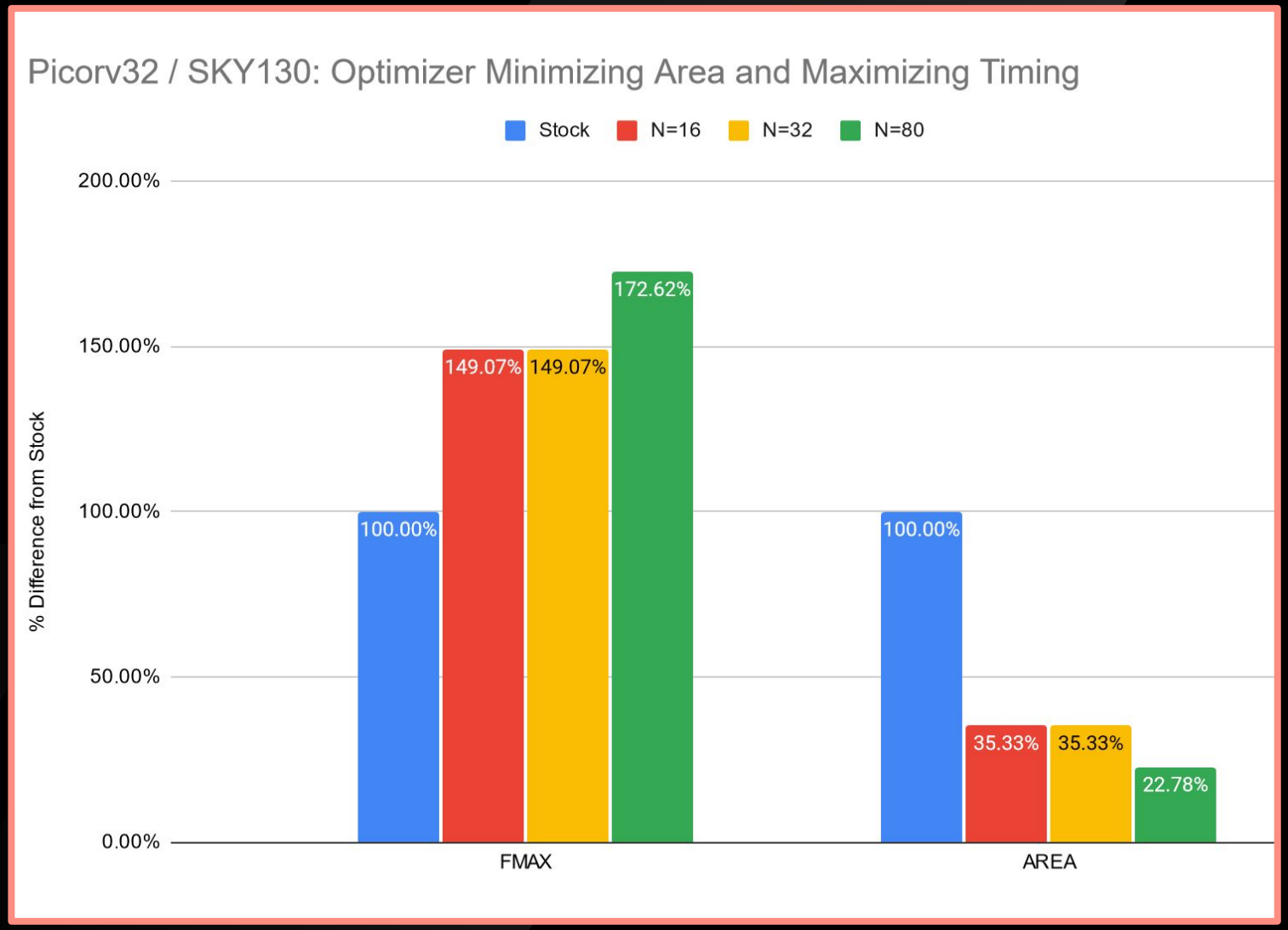
Tracking Open Source Goodness (300+ and counting)



https://github.com/aolofsson/awesome-opensource-hardware

Autotuning Experiment With Azure

- SC captures metrics
- Google Vizier used for hyperparameter tuning
- SLURM used for scheduling
- 16 experiments per iteration
- 1 experiment per server
- ~8 Hr run time
- Very promising results!



N = total #experiments, Parameters : Density, syn strategy, place density

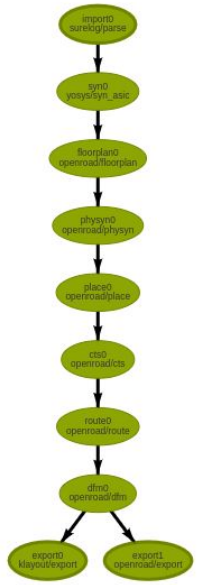
Metrics, Metrics, Metrics, ...

ibex dashboard

default

Metrics | Manifest | File Preview | Design Preview

Flowgraph



Data Metrics

Transpose

	import0	syn0	floorplan0	physyn0	place0	cts0	route0	dfm0	export0	export1
errors	0	0	0	0	0	0	0	0	0	0
warnings	37	9	43	31	31	698	1039	1032	1	31
drvs	None	None	9679	9679	46	110	115	0	None	6
unconstrained	None	None	6	6	6	6	6	6	None	6
cellarea (um^2)	None	2166.967	2125.550	2125.550	2340.370	2442.350	2442.350	2442.350	None	2442.350
totalarea (um^2)	None	None	21661.100	21661.100	21661.100	21661.100	21661.100	21661.100	None	21661.100
utilization (%)	None	None	9.813	9.813	10.805	11.275	11.275	11.275	None	11.275
peakpower (mw)	None	None	17.143	17.143	20.701	21.870	21.526	18.680	None	21.984
leakagepower (mw)	None	None	0.002	0.002	0.002	0.003	0.003	0.003	None	0.003
holdpaths	None	None	1650	1650	1650	1650	1653	1650	None	1681

Select Parameters

Node Information

	import0
errors	0
warnings	37
memory (B)	429.895M
exetime	08.449

	import0
endtime	2023-07-06 09:10:02
scversion	0.12.3
starttime	2023-07-06 09:09:53
task	parse

files

- outputs
- slpp_all
- inputs
- reports
  - import.log
  - import.errors

sc-dashboard -cfg build/ibex/job0_asap7_demo/ibex.pkg.json

zero

SiliconCompiler State of the Union

- Used in production environment every day at ZA
- New commits almost every day
- Funding secured to continue development for next three years
- Countless improvements in flight:
  - Package management
  - Dashboard
  - Support for new nodes
  - Zero install installation
  - FPGA compilation

- What else should we work on??

Thank you!