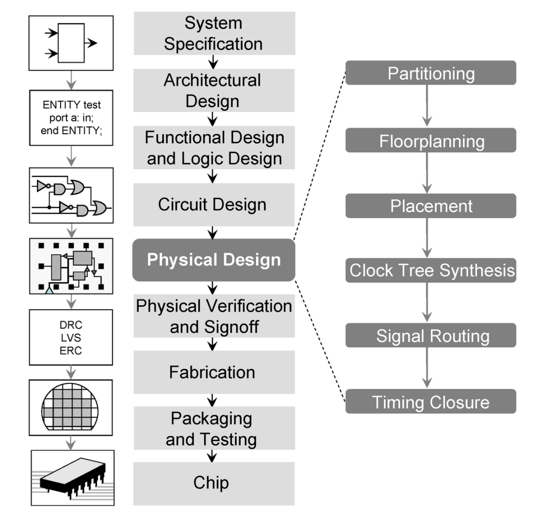
Informing Physical Design in SCD

Predicting parameter cost

Robert D'Angelo

March 26, 2019

What is physical design?



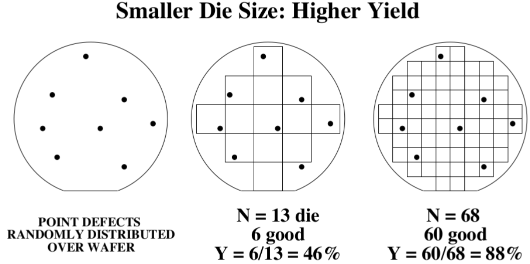
How can SCD help?

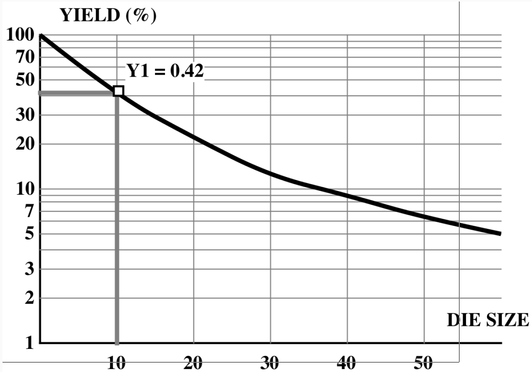
Real time feedback on cost of configuration choices.

- "Impact scores" per knob
  - Area
  - Power
  - Timing
  - Compute cost (PNR/timing optimization), e.g. clock gating
  - I/O requirements
  - Benchmarks

Why show this information? Motivating example

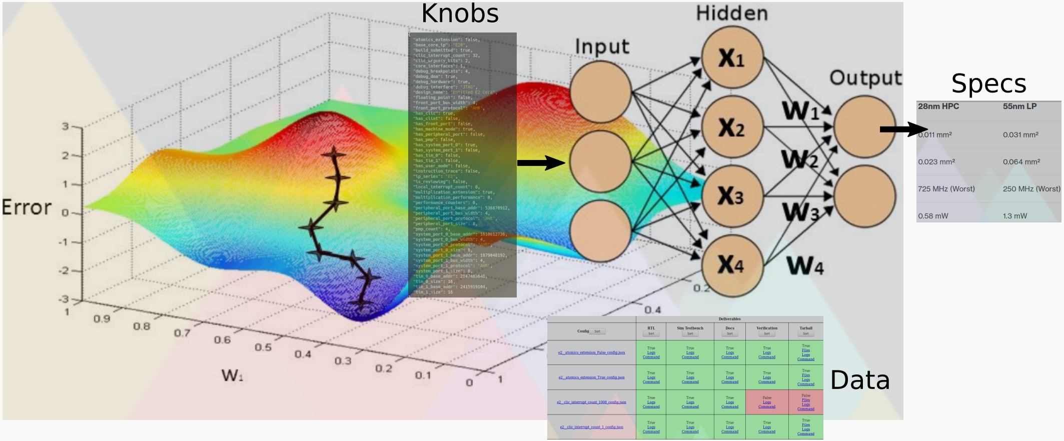
Yield/Cost vs. Area





4x area reduction → 10x cost reduction

Real time knob cost predictions in your browser?!?




```
{
  "atomics_extension": false,
  "base_core_ip": "E20",
  "build_submitted": true,
  "clic_interrupt_count": 32,
  "clic_urgency_bits": 2,
  "core_interfaces": 1,
  "debug_breakpoints": 4,
  "debug_dma": true,
  "debug_hardware": true,
  "debug_interface": "JTAG",
  "design_name": "Untitled E2 Core",
  "floating_point": false,
  "front_port_bus_width": 4,
  "front_port_protocol": "AHB",
  "has_clic": true,
  "has_clint": false,
  "has_front_port": false,
  "has_machine_mode": true,
  "has_peripheral_port": false,
  "has_pmp": false,
  "has_system_port_0": true,
  "has_system_port_1": false,
  "has_tim_0": false,
  "has_tim_1": false,
  "has_user_mode": false,
  "instruction_trace": false,
  "ip_series": "E2",
  "is_reviewing": false,
  "local_interrupt_count": 8,
  "multiplication_extension": true,
  "multiplication_performance": 8,
  "performance_counters": 0,
  "peripheral_port_base_addr": 536870912,
  "peripheral_port_bus_width": 4,
  "peripheral_port_protocol": "AHB",
  "peripheral_port_size": 8,
  "pmp_count": 4,
  "system_port_0_base_addr": 1610612736,
  "system_port_0_bus_width": 4,
  "system_port_0_protocol": "AHB",
  "system_port_0_size": 8,
  "system_port_1_base_addr": 1879048192,
  "system_port_1_bus_width": 4,
  "system_port_1_protocol": "AHB",
  "system_port_1_size": 8,
  "tim_0_base_addr": 2147483648,
  "tim_0_size": 16,
  "tim_1_base_addr": 2415919104,
  "tim_1_size": 16
```


28nm HPC	55nm LP
0.011 mm²	0.031 mm²
0.023 mm²	0.064 mm²
725 MHz (Worst)	250 MHz (Worst)
0.58 mW	1.3 mW

	Deliverables				
Config Sort	RTL Sort	Sim Testbench Sort	Docs Sort	Verification Sort	Tarball Sort
e2__atomics_extension_False_config.json	True Logs Command	True Logs Command	True Logs Command	True Logs Command	True Files Logs Command
e2__atomics_extension_True_config.json	True Logs Command	True Logs Command	True Logs Command	True Logs Command	True Files Logs Command
e2__clic_interrupt_count_1008_config.json	True Logs Command	True Logs Command	True Logs Command	False Logs Command	False Files Logs Command
e2__clic_interrupt_count_1_config.json	True Logs Command	True Logs Command	True Logs Command	True Logs Command	True Files Logs Command

Need a model

Challenges to making front end prediction

- Compute time (Too many parameters)
- Modeling (Complex designs)
- Licensing (Expensive software, prohibitive use)

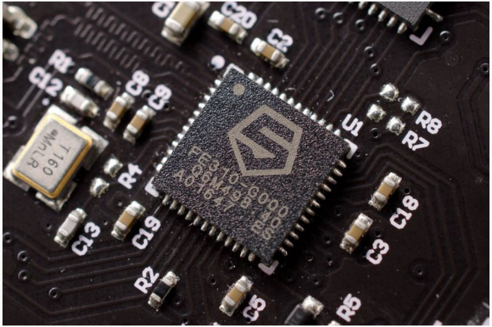
Need a model, Part 2

Single input, single output neural network may not be enough. Ideally, create a model that can map data from different correlated data sources to qualitative performance metrics.

- Compute time $\rightarrow$ Sparse sampling + Pattern recognition/ML
- Modeling $\rightarrow$ Statistical dependence over physical models
- Licensing $\rightarrow$ Remote execution and FOSS

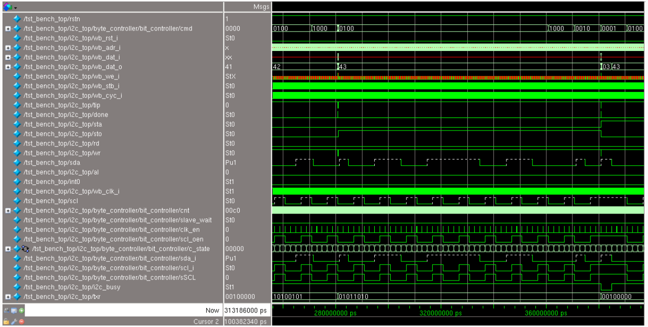
Need data

- Hardware measurements
  - Best, ground truth, fast (speed of RL)
  - Expensive (\$10k-\$1M), Time consuming (months to years)
  - Error prone (humans make mistakes), outlier effects (small sample size)
  - Cutting edge prohibitively expensive ($> 14nm$)



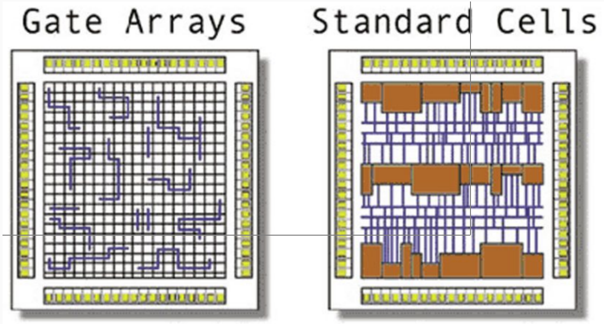
Need data

- ASIC Simulations (SPICE, PNR annotated gate sim, .lib delay models)
  - Accurate, time consuming (hours to days to weeks)
  - Can do some statistics for free with stat params from PDK
  - Expensive ($40k per license per year), time (days to weeks, 10 hrs with prometheus)
  - Cutting edge ($< 14nm$ simulations)



Need data

- FPGA
  - Apples vs. oranges, relative information possible, still real hardware
  - Time consuming (minutes to hours)
  - Cheap (\$500-\$3k/board, \$300/license)
  - Limited to process on board (14-32nm)



Need data, Part 2

- Scoreboard great for integration testing, but not suitable for data collection
- Mini-scoreboard + Deputy + Wake = Data
- xarrays `https://github.com/pydata/xarray`, N-D extension of pandas
- database vs netCDF `https://www.unidata.ucar.edu/blogs/developer/entry/netcdf_schema_language`
- Tensorflow(.js) to deploy trained models for front end design feedback

Typical gamma workload (lots of idle compute to be exploited):