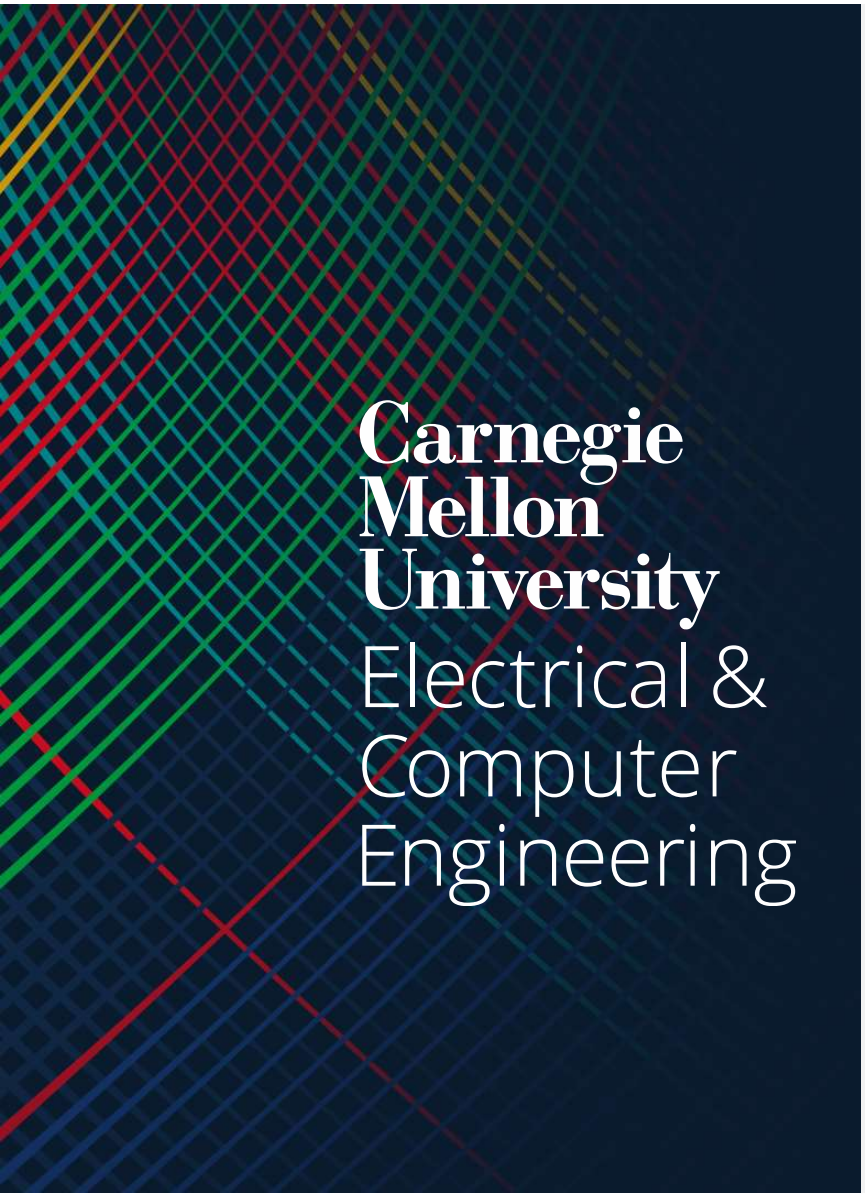


Halo Classes in VLSI Design

October 14, 20201

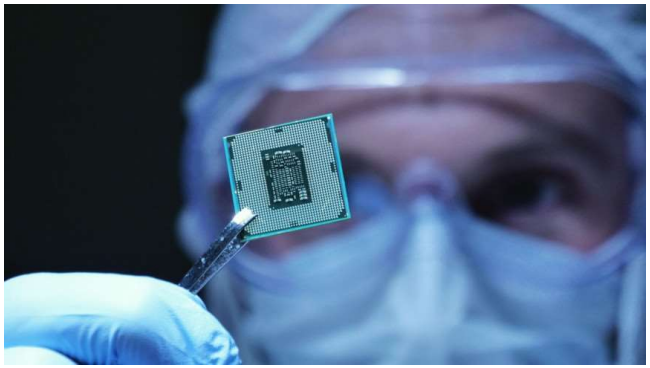
Ken Mai

Student Goals

What are the student goals?

- Graduate
- Go to graduate school
- Find a good job

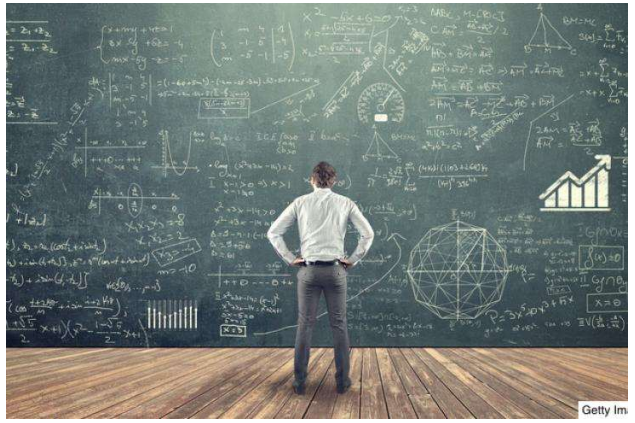
- Build something fun and exciting



Academia Goals

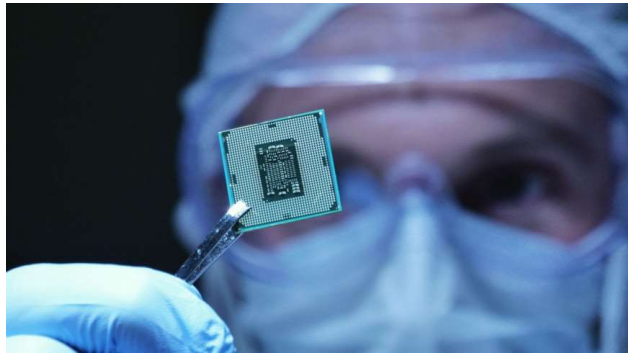
Educate students

- Teach them how to think & design
- Teach them how to drive SOTA tools
- Teach them how to work in SOTA process nodes
- Show them the entire design stack



Research

- High risk & unclear profit path
- 5-10+ years from deployment
- Hopefully relevant to the field at large
- Build something exciting …



Halo Products

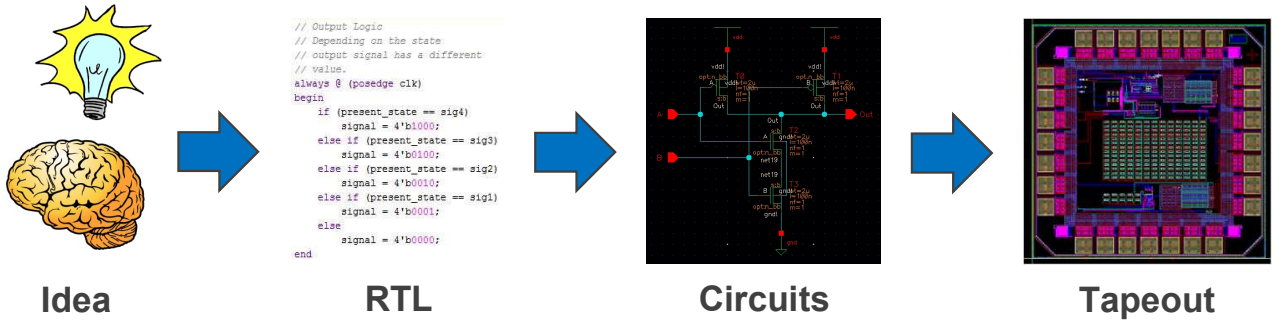
- Have to attract them *to* the field before you can do that
- How to make them excited about the field?

Audi R8

Apple iPod

Build 18: Infinite Fish Tank

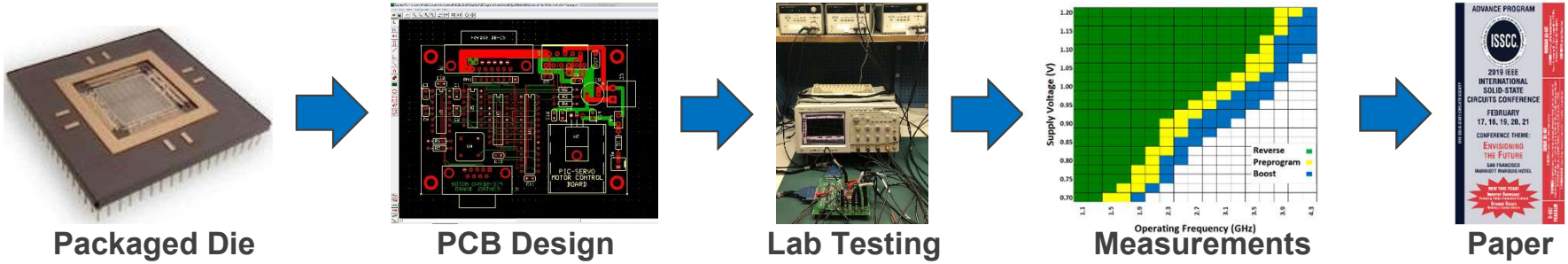
Halo Class Sequence in VLSI Design



Two semester VLSI design and test class sequence

- Design (Spring) → Fab (Summer) → Test (Fall)
- ~30 PhD and UG/MS students working in teams of 2-4
- Digital, mixed-signal, and analog/RF designs
  - Students decide what to design ← ownership is key
  - Some advisor driven projects

Halo Class Sequence in VLSI Design (cont.)



Two semester VLSI design and test class sequence

- Commercial tool flow
  - Mainly Cadence at CMU
- 28nm planar CMOS process (NDA w/ lock-out period, export control)
  - Industrial funding support
- Custom test setups
  - Want students to see all of the design/test/verification stack

Process Technologies

Two paths in the class

- Industrial process & industrial tools
  - 28nm planar
  - Cadence tools
- Generic process & industrial tools ← university required equivalent path
  - Cadence 45nm GPDK
  - Cadence tools

Industrial process requires NDA and export control

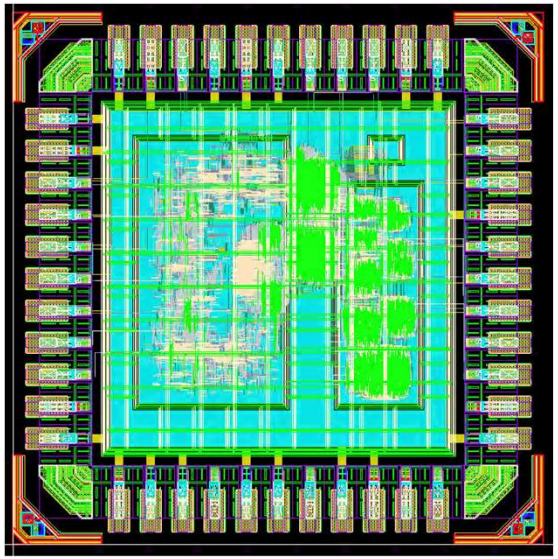
- Lock-out period for sub-28nm technologies
- Export control of PDK
  - Unexpected interaction with CoViD and students located in foreign countries

Rough Chip Specs

1mm^2 die slot per team

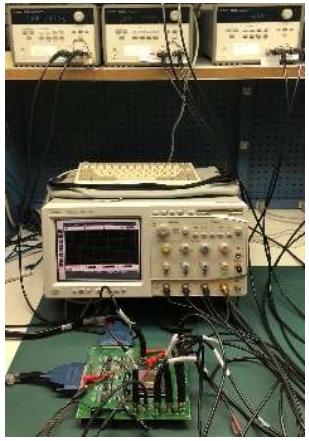
- 700mm x 700mm core area
- ~200k logic gates
- ~2Mb SRAM
- 40 I/O pads

MOSIS TinyChip-ish

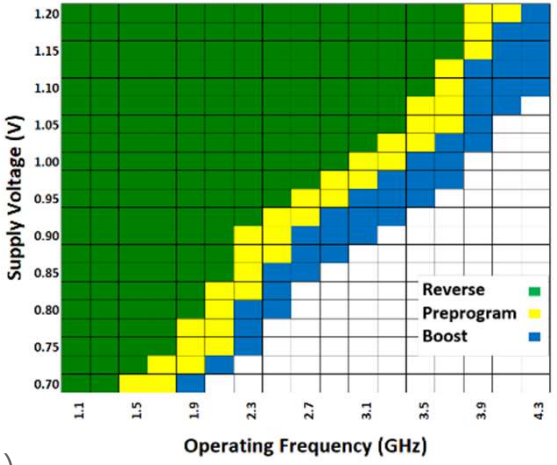


Victory conditions

- Test the chip

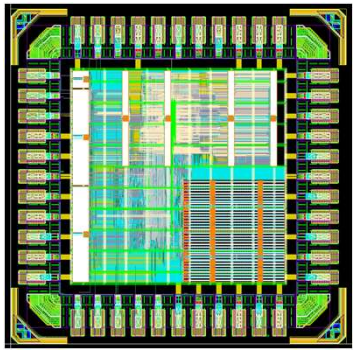


- If the chip is working
  - Gather data
  - Presentation/paper/demo

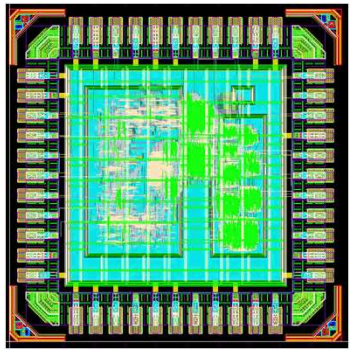


- If the chip is not working
  - Figure out why the chip is not working
  - Implement a fix
  - Verify the fix (fixes the issue & no new ones)
  - Presentation/paper/demo
  - Fab again?

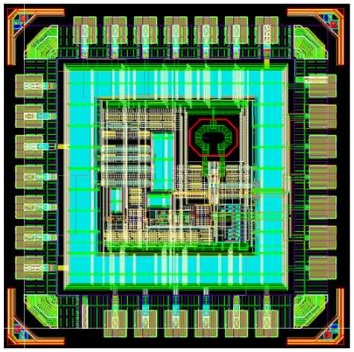
18-725 Spring 2020 Chips



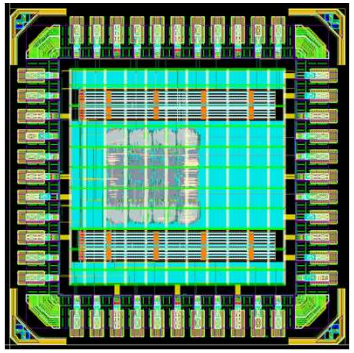
Graphics Processor



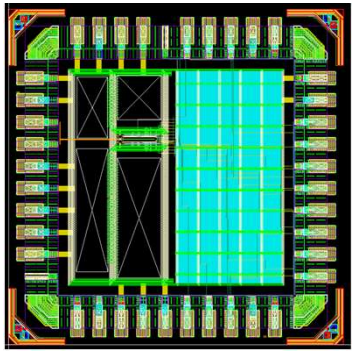
CryptoLogic



Phase Locked Loop

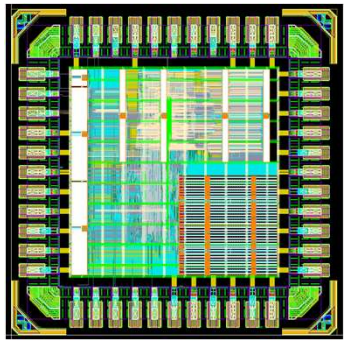


FPGA



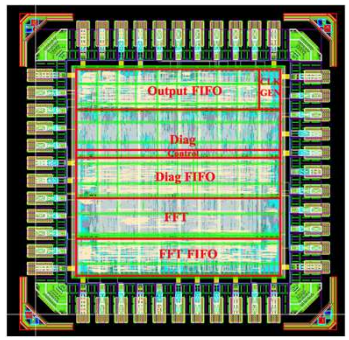
SAR ADC

18-725: Projects in IC Design-Taped Out - **Spring 2021**



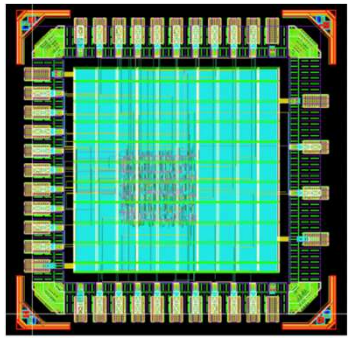
Lab on a chip

- Capacitance-to-freq. sensor array
- Transmits data by RF near field antenna. Sensitivity 2.8 MHz/fF
- Autonomous wireless 29.8uW., Pad powered 88.2uW



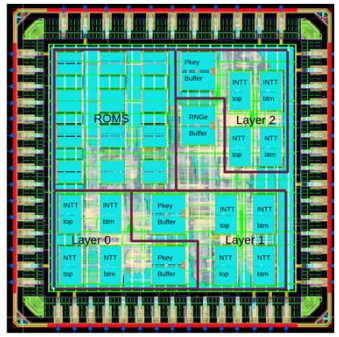
CNN-FFT

- High throughput 8-point DFT kernel, 45 I/O,
- Peak thoughput 56GFLOPs, FFT8 latency 17ns
- 87.47 mW at 500 MHz



64-tile General Purpose FPGA

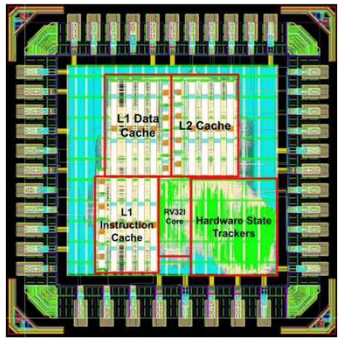
- 8x8 CLB array,
- Test case: 6 bit adder, 5 &4bit multiplier clock divider, 64 bit shift register
- 37 IOPads, 50 MHz (FPGA core)



Homomorphic Encryption Accelerator

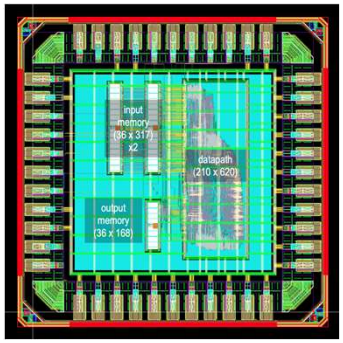
- BFV zero-encryption 126x faster than baseline software and uses 6.6 less energy
- 67.64 mW, Time 0.616 ms for zero encryption operation

18-725: Projects in IC Design-Taped Out - **Spring 2021**



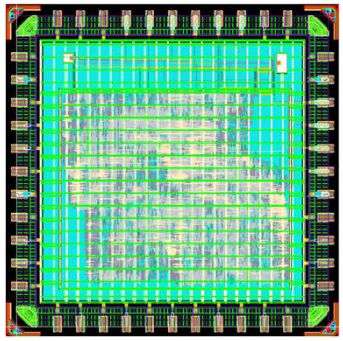
Root of Trust

- Monitoring firmware/SW control flow integrity. RISC-V core. 50 IO Pads
- Memory Capacity 8KB
- 5mW Power, Clk 100MHz



Fingerprint NN

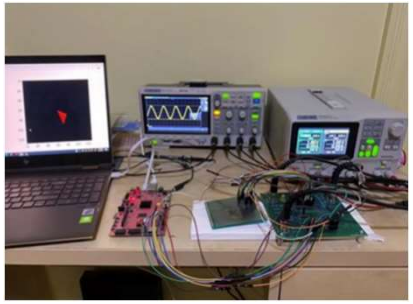
- DNN accelerator for local operation on edge devices
- 6.91 mW at 100MHz



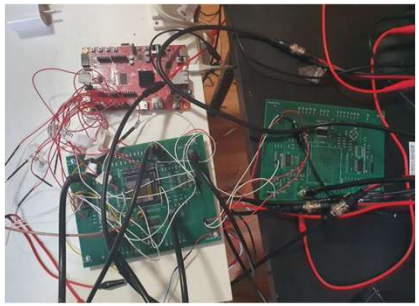
Hard disk Read-Channel NN

- CNN data detector. 11 hidden layers with 15 filters
- 283 mW at 200MHz

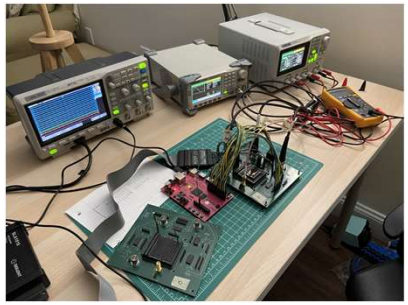
18-726: First Silicon Board Bring up & Measurement



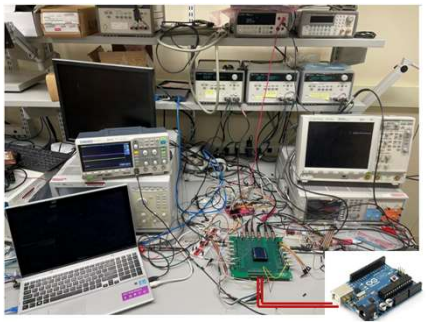
OpenGL
Graphics Pipeline



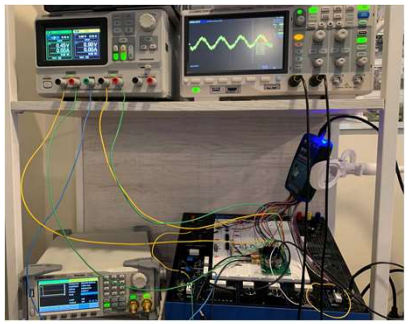
Obfuscated Fixed-Weight
Vector MAC Unit



FPGA

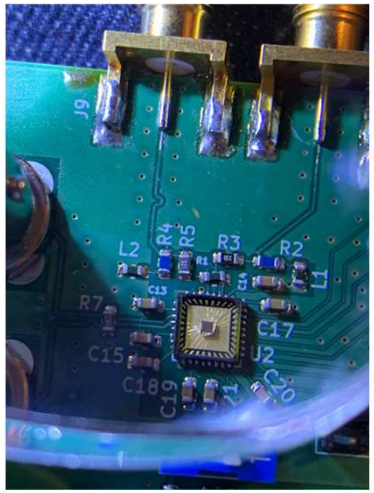


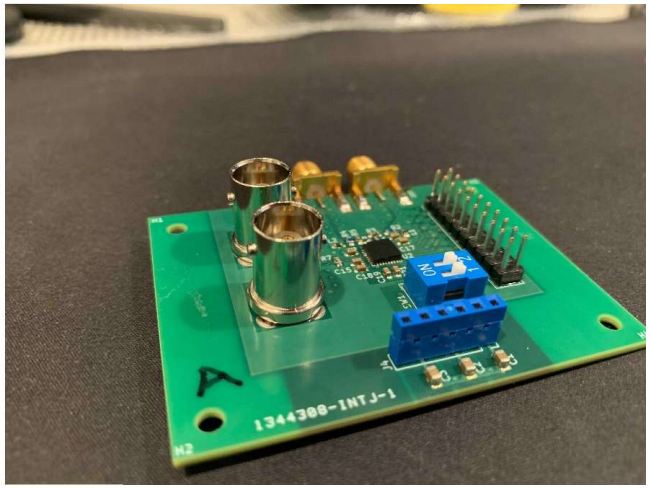
Manic- Vector data flow for
embedded system

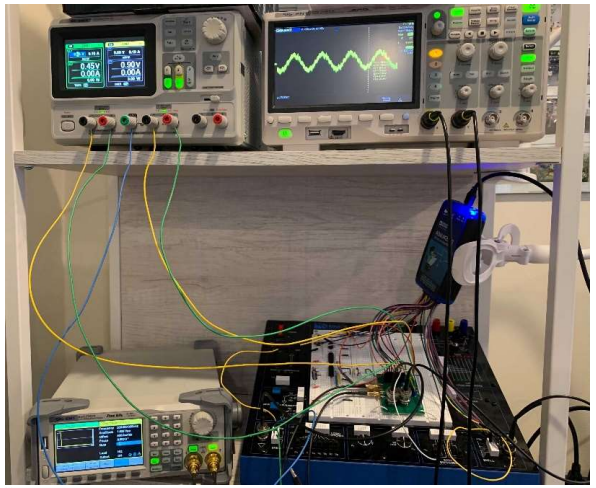


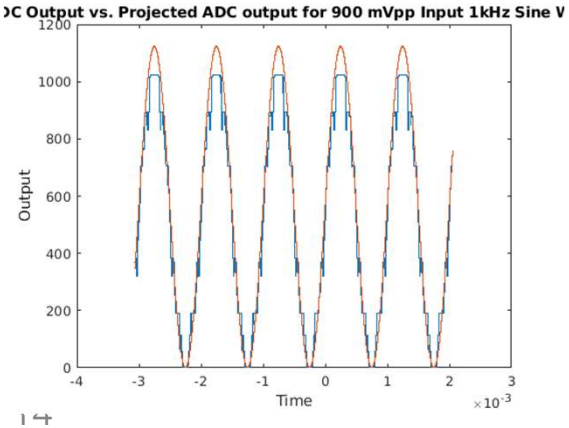
SAR ADC

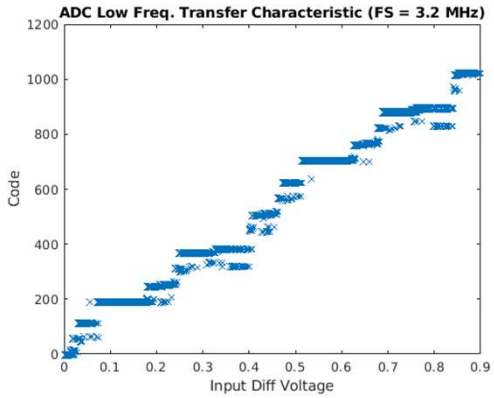
Putting It All Together

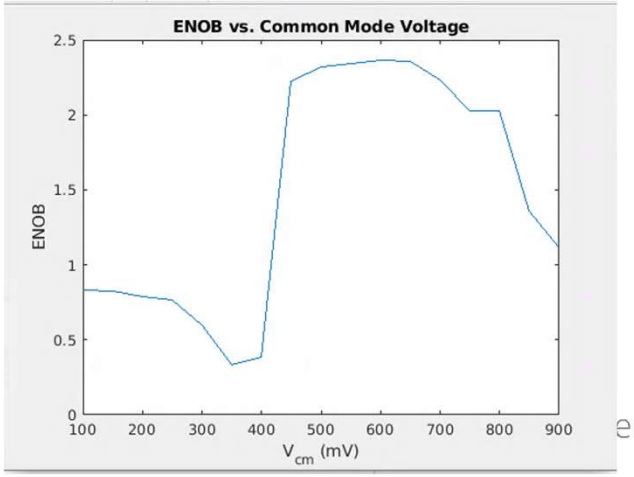












ering

Lessons Learned: Halo Effect

- Currently mid second year of effort
- Students who finished sequence
  - Most pursued jobs or graduate school in VLSI
  - Many reported their projects were major topic of interviews
- Halo effect seems to be working
  - Significant increase in UG/MS students taking circuits/VLSI sequence
  - Significant increase in faculty sending PhD students to sequence
  - Intro to VLSI design class grew from ~10 to ~50

Lessons Learned: Process Technology

Do we need to use advanced processes (e.g., 28nm)?

- Expensive-ish
- NDA
- Export control
- Still not that close to SOTA industry

What's good about it

- Industrial PDK
- Industrial tools
- Complete standard cell, IO, memory compiler, etc. support
- Modern enough to publish

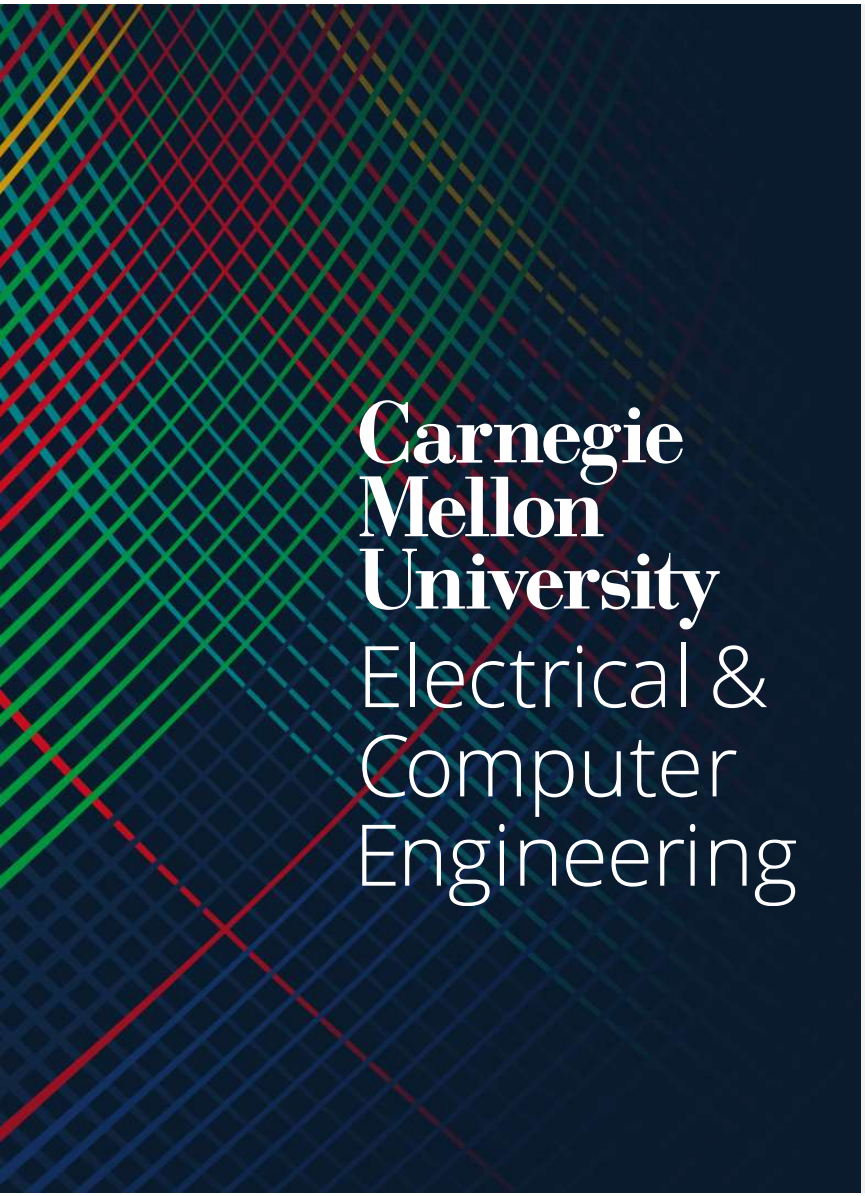
Likely get majority of benefit from older node (e.g., 65nm)

- If you're not looking to publish

Lessons Learned: Suggestions

- Industrial tools?
  - Output quality
  - Compatibility and lower number of bugs
  - Community and user base
  - Full integrated set of tools
  - What students will see in industry
  - Low cost for academia

- Process technology?
  - Supported PDK on toolset
  - Basic IP collateral – standard cells, IO, memory compiler, etc.
  - Doesn't have to be state-of-the-art (for education)
  - Do we need/want research buy in?
  - Efabless ChipIgnite?



Soft eFPGA Generator

October 14, 20201

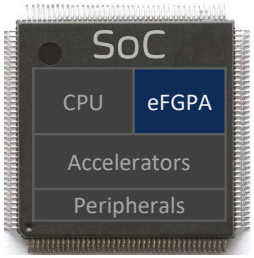
Ken Mai

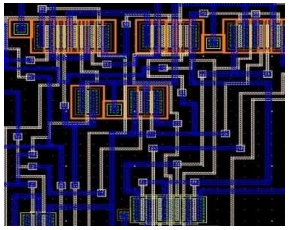
Embedded FPGA Fabrics – Hard vs Soft Fabrics

- eFPGA are increasingly used in SoCs
  - Accelerators → faster & lower power compared to CPU
  - Custom logic → for interfacing with external components/sensors
  - Feature addition, bug fixes, security, etc.

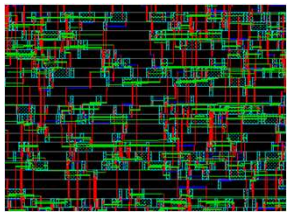
- Hard eFPGA fabrics use custom layout
  - Maximizes density, performance, and energy efficiency
  - Custom layout → extensive time/effort & poor process portability
  - Tight SoC schedules, budgets, and fab diversity needs

- Soft eFPGA fabrics use std-cells & automatic PnR tools
  - Quick arch. spec to layout turn around time (~1-2 days)
  - Specify eFPGA fabric as RTL → flexible architecture specs





Custom layout → manual & slow

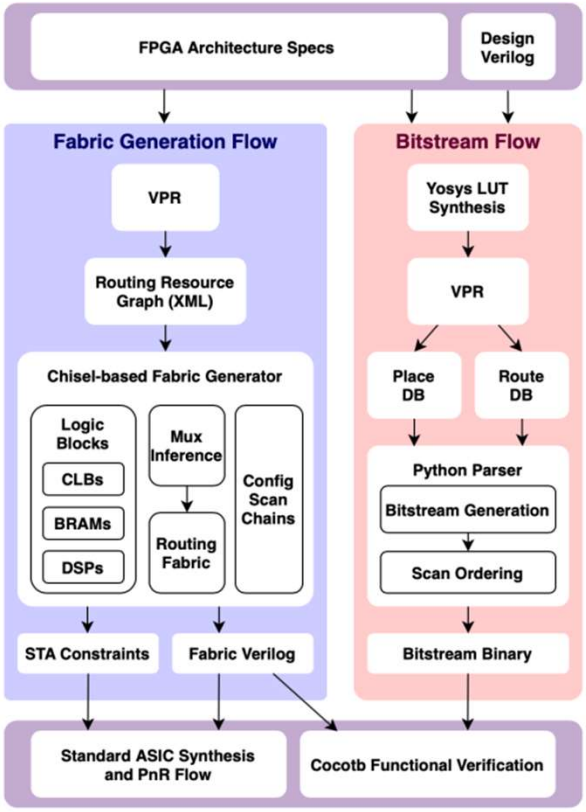


Std-cell layout → automated & fast

Fabric Generation Flow for Silicon Implementation

- Fabric generation flow
  - Uses Chisel hardware construction language & VPR
  - Input → VPR routing graph & logic blocks
  - Output → eFPGA Verilog + timing constraints (global timing view)
  - Standard ASIC synthesis, PnR, and physical verification flow

- Bitstream & functional verification flow
  - Uses Yosys + VPR to map designs on the fabric
  - Support functional Verilog simulation using Cocotb
  - Formal verification using Conformal Equivalence Checker



Proof of Concept Soft eFPGA Testchip

- 16nm FinFET CMOS process
- Soft eFPGA fabric
  - 16x16 tiles with 80 routing channels
  - 2K LUTs → 8 6-input LUTs per tile
  - eFPGA core: 1.32mm X 1.32mm = 1.74 mm^2
- Implementation details
  - PnR using top-down design methodology
  - JTAG like configuration scan-chains
  - FIFOs around eFPGA core for at speed testing
- Taped out designs in 65nm, 28nm, 22nm, and 16nm
  - Soft design allows for easy process portability
  - Including RHBD 6k LUT design on 22nm