# Design of Digital Circuits and Systems

## Course Wrap-Up

**Instructor:** Justin Hsia

**Teaching Assistants:**

Colton Harris
Deepti Anoop
Gayathri Vadhyan
Jared Yoder
Lancelot Wathieu
Matthew Hung

Thanks to Thierry Moreau and Katie Lim for their contributions to the following slides!

# Relevant Course Information

- Quiz 5 (STA, Pipelining, CDC) today @ **11:30** am
  - Scientific calculator allowed!
- Homework 6 (Advanced Testing) due Monday (5/27)
  - Monday office hours as scheduled, but remote only
- Lab 6 due the next Monday (6/3)
  - Demo, lab report, and video
  - You can demo before submitting your report
  - Next week's lecture slots will be extra office hours ***in the lab***
- Course eval will be released over the weekend
  - Will also post on Ed Discussion with an anonymous Google Form to give feedback to the TAs

# Review: Constraints

- Reminders:
  - A standard constraint expression includes 1+ randomizable variables and at most one comparison operator
  - Range: `[A:B]` means all integers from A to B, inclusive
  - Sets: `<var> inside {<set>};`
  - Distribution: `<var> dist {<distribution>};`

- Given `low`, `mid`, `high` write constraints such that:
  - `low` is 1 half the time, 2 one-third of the time, and 3 one-sixth of the time `constraint c_low { low dist {1:=3, 2:=2, 3:=1}; }`
  - `mid` is evenly distributed between `low` and `high` `constraint c_mid { mid inside {[low:high]}; }`
  - `high` is one of 9, 11, or 13 `constraint c_high { high inside {9, 11, 13}; }`

# Review: Constraints

- Given `low`, `mid`, `high` write constraints such that:
  - `low` is 1 half the time, 2 one-third of the time, and 3 one-sixth of the time
  - `mid` is evenly distributed between `low` and `high`
  - `high` is one of 9, 11, or 13

- What are the possible values of `mid`?

$mid_{min} = low_{min} = 1$

$mid_{max} = high_{max} = 13$

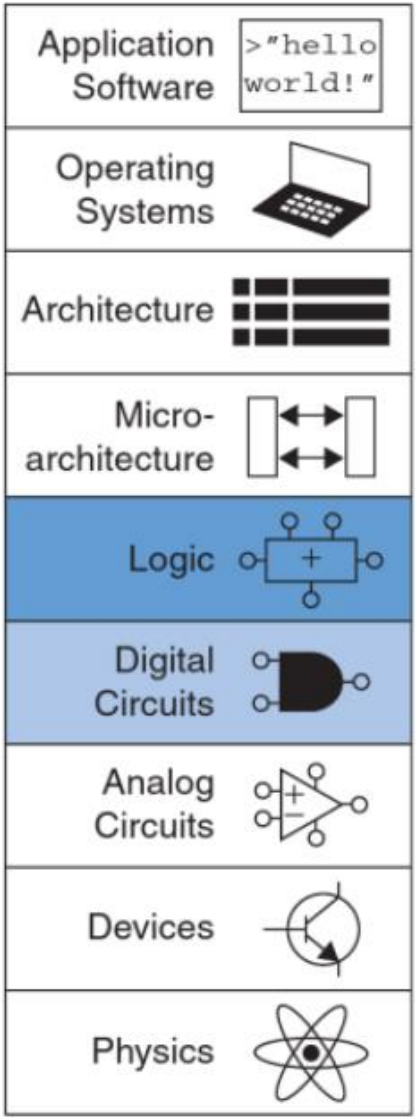
mid can be all integers from 1 to 13

# Review: Course Motivation

- More advanced digital logic design
  - Higher-level circuit design and analysis techniques
  - Interfacing with various devices/peripherals
  - How to implement algorithms in hardware
  - Practical timing analysis
  - "Verilog finishing school"

- This course is the end of the road at UW for gate-level design
  - Can start doing FPGA development

Harris and Harris. "Digital Design and Computer Architecture" 2nd ed.

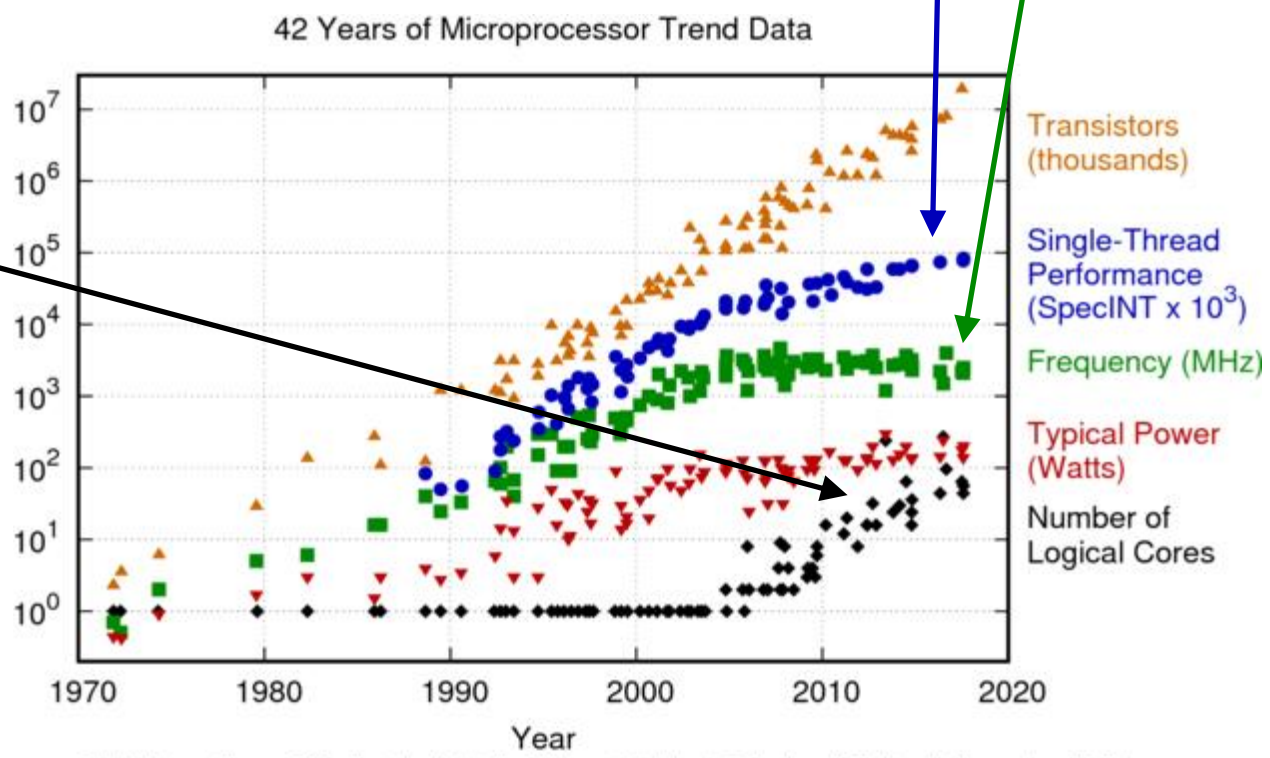


# Engines of Innovation in Computing

- Moore's Law
  - Transistor density doubling roughly very 2 years
  - *Expected trend:* more transistors with which to make more complex processors
- Dennard Scaling
  - Smaller transistors should use less power and have shorter delays
  - *Expected trend:* clock frequency should increase as transistors get smaller
- Parallelism
  - Multiple CPUs/cores can work on tasks simultaneously
  - *Expected trend:* more and more cores

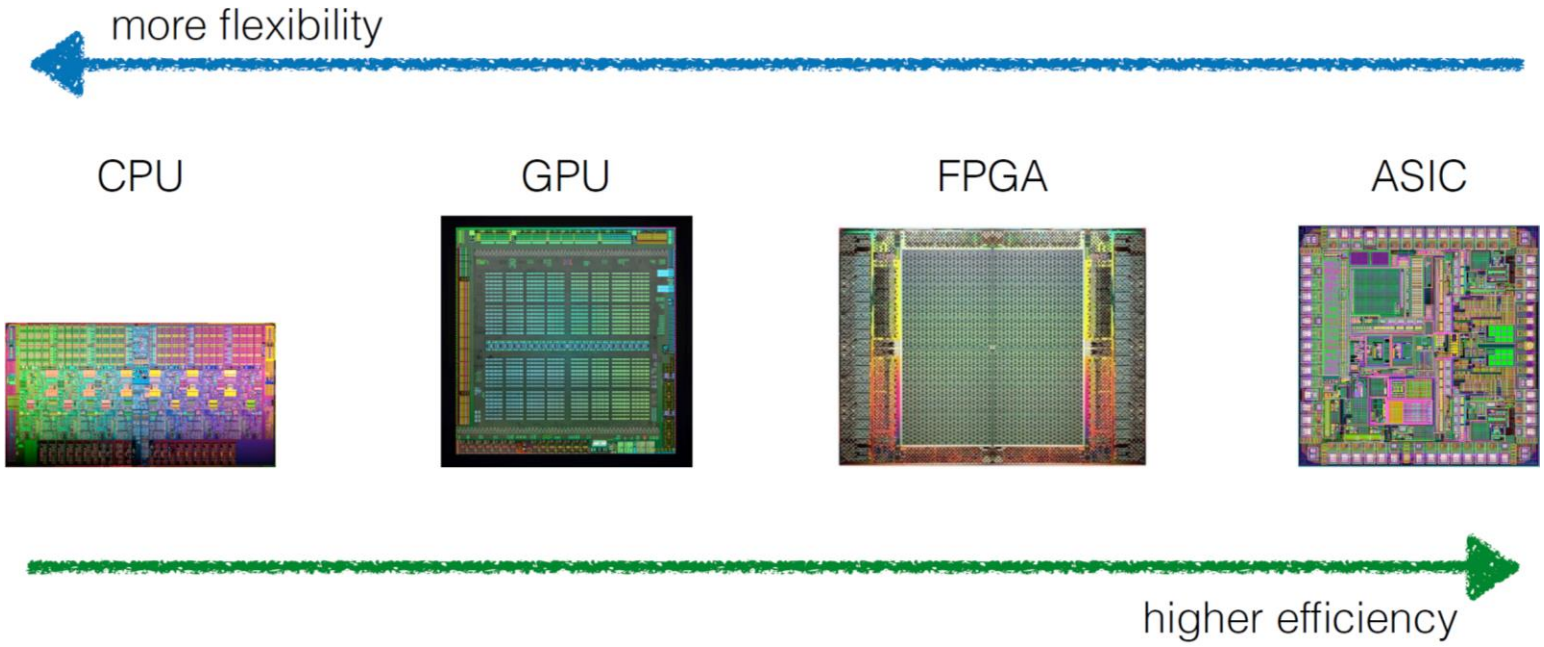
# Engines of Innovation in Computing

- These trends no longer hold!
  - Translating transistors into single-threaded performance is hard
  - Transistors are hitting physical and economic viability limits, so Dennard scaling no longer holds
  - Parallel performance is difficult to do right



# Engines of Innovation

- Specialization: squeeze more work out of those transistors
  - Tailor your chip architecture to the characteristics of a **stable** workload

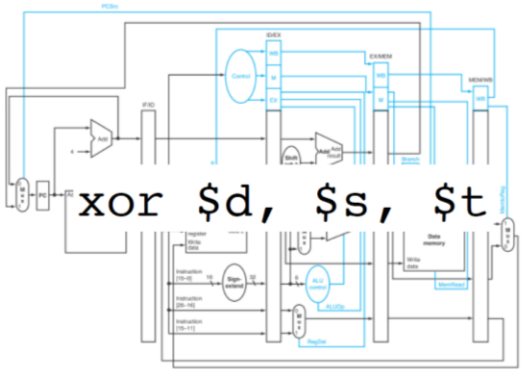


# Engines of Innovation

- Specialization: squeeze more work out of those transistors
  - Tailor your chip architecture to the characteristics of a **stable** workload

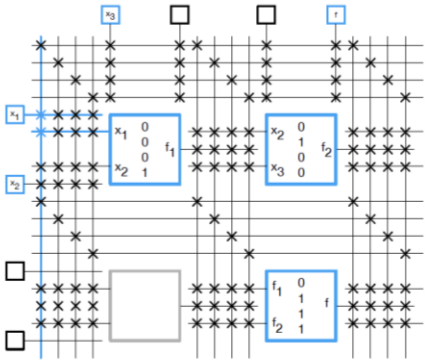
**XOR Example**


**CPU**



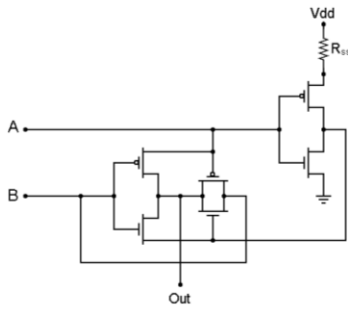

1 instruction, but billions of transistors

**FPGA**



1 lookup table, but lots of static power

**ASIC**

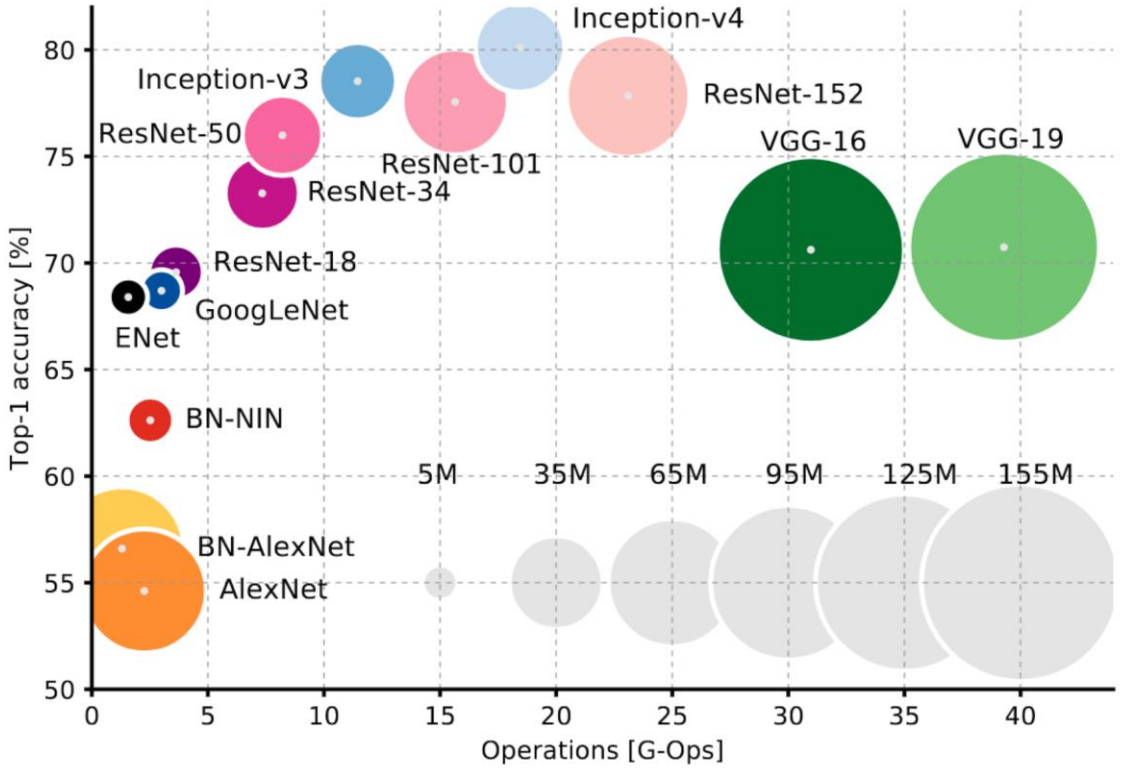



just 6 transistors

# Reliance on Specialization Will Increase

- Compute demand is growing
  - Fueled by heavy computations in web search, data science, machine learning, and blockchain technologies

Source: Eugenio Culurciello, An Analysis of Deep Neural Network Models for Practical Applications, arXiv:1605.07678



# Reliance on Specialization Will Increase

- We're in a golden age of hardware specialization

wired.com

**Building an AI Chip Saved Google From Building a Dozen New Data Centers**

*Author: Cade MetzCade Metz*

5-6 minutes

wired.com

**Tesla's New Chip Holds the Key to 'Full Self-Driving'**

*Author: Tom SimoniteTom Simo*

6-8 minutes

nytimes.com

**Amazon's Homegrown Chips Threaten Silicon Valley Giant Intel**

7-9 minutes

Andy Jassy, chief executive officer of Amazon's cloud computing service. Amazon is now building its own chip for its servers, giving the company new leverage over its longtime supplier, Intel.CreditDavid Paul Morris/Bloomberg

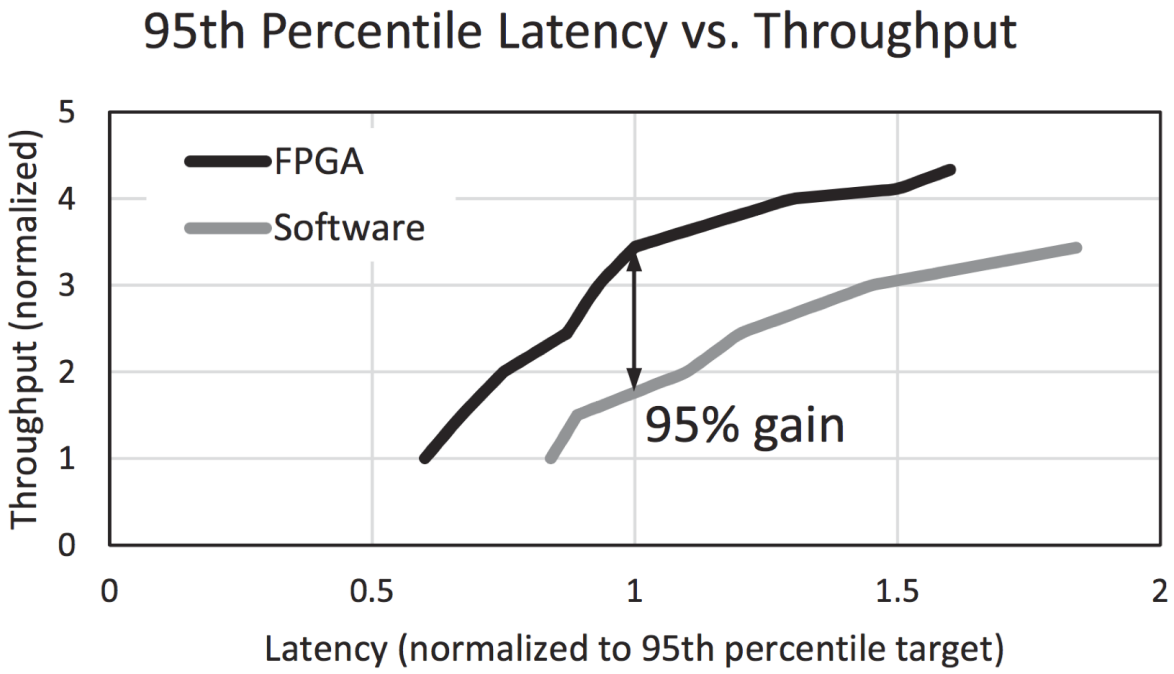
# Reliance on Specialization Will Increase

- We're in a golden age of hardware specialization
  - Tons of companies are building specialized chips


- https://www.google.com/search?q=fpga+jobs
- https://www.google.com/search?q=verification+engineer+jobs

# FPGAs in the Datacenter

- FPGAs have been used in deployment to accelerate Bing search among other things at Microsoft *




* Putnam et al., A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services? [ISCA14]

# FPGAs in the Datacenter

- FPGAs have been used in deployment to accelerate Bing search among other things at Microsoft

techcrunch.com

**Microsoft launches Project Brainwave, its deep learning acceleration platform – TechCrunch**

*Frederic Lardinois@fredericl / 1 year agocomment Comment*

2 minutes

Microsoft today announced at its Build conference the preview launch of Project Brainwave, its platform for running deep learning models in its Azure cloud and on the edge in real time.

While some of Microsoft's competitors, including Google, are betting on custom chips, Microsoft continues to bet on FPGAs to accelerate its models, and Brainwave is no exception. Microsoft argues that FPGAs give it more flexibility than designing custom chips and that the performance it achieves on standard Intel Stratix FPGAs is at least comparable to that of custom chips.

By highly tailoring the accelerator architecture to the workload (something we cannot do with ASICs without sacrificing generality), they were able to achieve 5x less latency than Google's TPU!

# Research Around FPGAs (at UW)

- Prof. Scott Hauck (ECE)
  - "The application of FPGA technology to situations that can make use of their novel features: HPC, reconfigurable subsystems for SoCs, computer architecture education, hyperspectral image compression, and other areas."
- Prof. Rania Hussein (ECE)
  - FPGAs for education (LabsLand)
- Prof. David Kohlbrenner (CSE)
  - Developing attacks against cloud FPGA security systems
- Prof. Michael Taylor (ECE/CSE)
  - BlackParrot: An Agile Open-Source RISC-V Multicore for Accelerator SoCs [FPGAs as testing environment]

# Research Around FPGAs (outside of UW)

- Emerging "MLSys" research area centered around software and hardware systems for deploying modern ML methods
  - https://arxiv.org/pdf/1904.03257.pdf

- Look for general Systems keywords like *hardware specialization* and *heterogeneous architectures*
  - "In heterogeneous architectures, an integrated circuit (ASIC) or a field-programmable gate array (FPGA) is used instead of just one CPU or GPU to perform highly specialized tasks."

# Where To Go From Here?

- This course is the *end of the road at UW* for gate-level design, but still other related areas
  - Good building block for embedded systems, digital VLSI, and computer architecture

- Future classes
  - Logic Gates: EE 331
  - Embedded Systems: EE/CSE 474, EE/CSE 475 (capstone)
  - Computer Architecture: EE/CSE 469, EE/CSE 470
  - Digital VLSI: EE 476, EE 477, EE 478 (capstone)

# Thanks for a great quarter!

- Huge thanks to your awesome TAs!


- Thanks to the original course creator: (redesigned in 18au)


- Hope you had fun this quarter and best of luck in the future!

# Ask Me Anything (AMA)


That's all Folks!