

ESE5320: System-on-a-Chip Architecture

Day 1: August 31, 2022
Introduction and Overview
(lecture start target 10:20am)

Masks required in Lecture

Note: slides linked to web

www.seas.upenn.edu/~ese5320/fall2022/fall2022.html

- Preclass (work now)
- Feedback form (turn in end of lecture)



Penn ESE5320 Fall 2022 -- DeHon

1

Today

- Part 1: Case for Programmable SoC
- Part 2
 - Course Goals
 - Outcomes
 - Risks, Tools
- Part 3: Sample Optimization
- Part 4: This course
 - (including policies, logistics)

Penn ESE5320 Fall 2022 -- DeHon

2

Apple A15 Bionic

- 108mm², 5nm
- 15 Billion Tr.
- iPhone 13
- 6 ARM cores
 - 2 fast (3.2GHz)
 - 4 low energy (2GHz)
- 5 custom GPUs
- 16 Neural Engines
 - 11 Trillion ops/s?

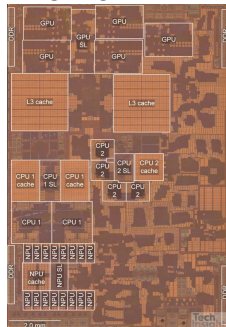


Image from <https://semianalysis.com/apple-a15-die-shot-and-annotation-ip-block-area-analysis/>
<https://www.anandtech.com/show/16983/the-apple-a15-soc-performance-review-faster-more-efficient>

Penn ESE5320 Fall 2022 -- DeHon

3

3

Questions

- Why do today's SoC look like they do?
- How approach programming modern SoCs?
- How design a custom SoC?
- When building a System-on-a-Chip (SoC)
 - How much area should go into:
 - Processor cores, GPUs, FPGA logic, memory, interconnect, custom functions (which) ?



Penn ESE5320 Fall 2022 -- DeHon

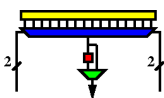
6

FPGA

Field-Programmable Gate Array

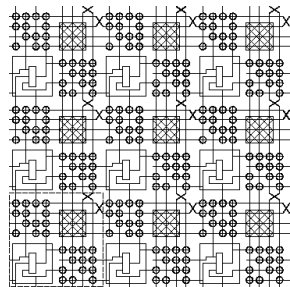
K-LUT (typical k=4 or 6)

Compute block
w/ optional
output Flip-Flop



ESE1500, CIS5710

Penn ESE5320 Fall 2022 -- DeHon



7

7

Case for Programmable SoC

Penn ESE5320 Fall 2022 -- DeHon

8

8

End of microprocessor Scaling

Old

- Moore's Law scaling delivered faster transistors
- Processors rode Moore's Law
 - Turning transistors into performance
- Could wait and ride technology curve

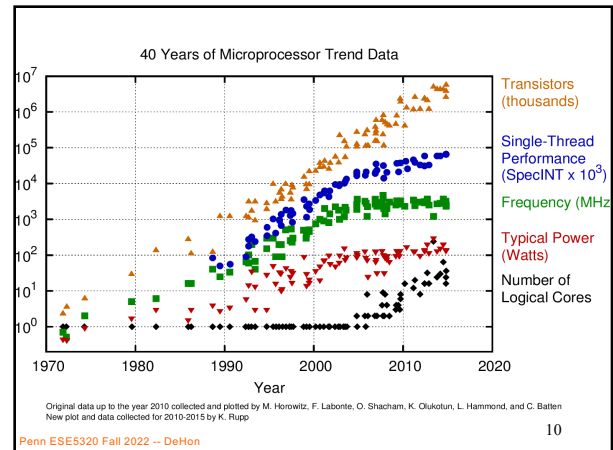
Now

- Dennard's Law kicked in
 - How need to scale voltage with size
- microprocessors were burning more power
- Lost ability to scale down voltage
- Processor performance stalled

Penn ESE5320 Fall 2022 -- DeHon

9

9



10

10

The Way things Were

30 years ago

- Wanted programmability
 - used a processor
- Wanted it a little faster
 - Next year's processor would run faster...
- Wanted high-throughput
 - used a custom IC
- Wanted product differentiation
 - Got it at the board level
 - Select which ICs and how wired
- Build a custom IC
 - It was about gates and logic

Penn ESE5320 Fall 2022 -- DeHon

11

11

Today

- Microprocessor may not be fast enough
 - (but often it is)
 - Or low enough energy
- Single core processor scaling has ended
- Time and Cost of a custom IC is too high
 - \$100M's of dollars for development, Years
- FPGAs promising
 - But build everything from prog. gates?
- Premium for small part count
 - And avoid chip crossing

Penn ESE5320 Fall 2022 -- DeHon

12

12

Non-Recurring Engineering (NRE) Costs

- Costs spent up front on development
 - Engineering Design Time
 - Prototypes
 - Mask costs
- Recurring Engineering
 - Costs to produce each chip

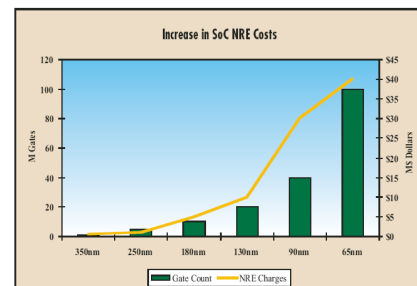
$$Cost(N_{chips}) = Cost_{NRE} + N_{chips} \times Cost_{perchip}$$

Penn ESE5320 Fall 2022 -- DeHon

13

13

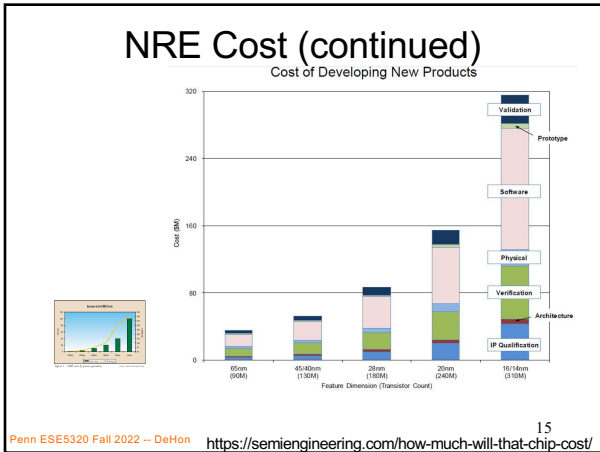
NRE Costs



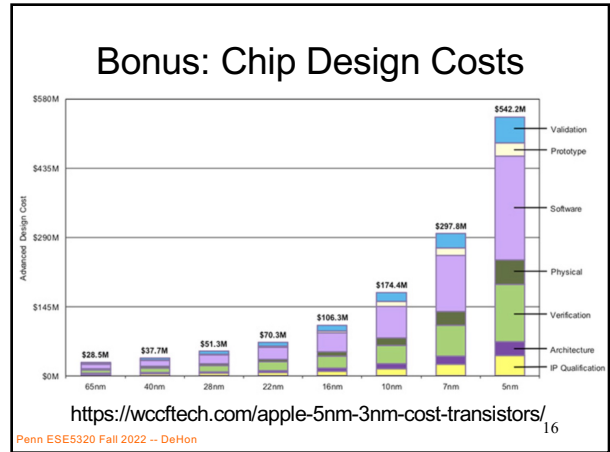
Penn ESE5320 Fall 2022 -- DeHon

14

14



15



16

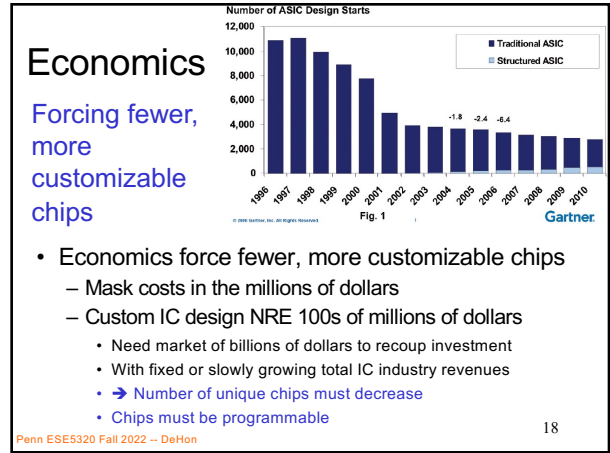
Amortize NRE with Volume

$$Cost(N_{chips}) = Cost_{NRE} + N_{chips} \times Cost_{perchip}$$

$$Cost = \frac{Cost_{NRE}}{N_{chips}} + Cost_{perchip}$$

Penn ESE5320 Fall 2022 -- DeHon

17



18

Large ICs

- Now contain significant software
 - Almost all have embedded processors
- Must co-design SW and HW
- Must solve complete computing task
 - Tasks has components with variety of needs
 - Some don't need custom circuit
 - 90/10 Rule

Penn ESE5320 Fall 2022 -- DeHon

19

Given Demand for Programmable

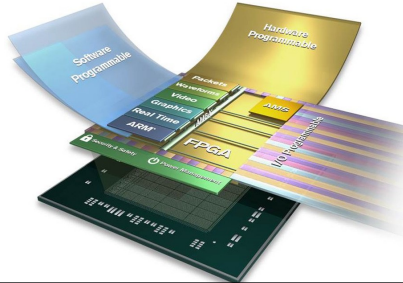
- How do we get higher performance than a processor, while retaining programmability?

Penn ESE5320 Fall 2022 -- DeHon

20

Programmable SoC

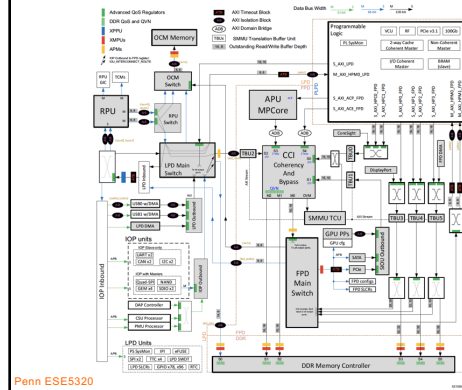
- Implementation Platform for innovation
 - This is what you target (avoid NRE)
 - Implementation vehicle



Penn ESE5320 Fall 2022 -- DeHon

21

Programmable SoC



Penn ESE5320

UG1085
Xilinx
UltraScale
Zynq
TRM
(p27)

22

Then and Now

30 years ago

- Programmability?
 - use a processor
- Faster
 - Processors scaled
- High-throughput
 - used a custom IC
- Wanted product differentiation
 - board level
 - Select & wired IC
- Build a custom IC
 - It was about gates and logic

Today

- Programmability?
 - uP, FPGA, GPU, PSoC
- Faster
 - Can't get with single core
- High-throughput
 - FPGA, GPU, PSoC, custom
- Wanted product differentiation
 - Program FPGAs, PSoC
- Build a custom IC
 - System and software

Penn ESE5320 Fall 2022 -- DeHon

23

23

Part 2: Course Goals, Outcomes

Penn ESE5320 Fall 2022 -- DeHon

24

24

Goals

- Create Computer Engineers
 - SW/HW divide is wrong, outdated
 - Computer engineers understand computation
 - HW and SW are just tools and design options
 - Parallelism, data movement, resource management, abstractions
 - Cannot build a chip without software
- SoC user – know how to exploit
- SoC designer – architecture space, hw/sw codesign
- Project experience – design and optimization

Penn ESE5320 Fall 2022 -- DeHon

25

Roles

- PhD Qualifier
 - One broad Computer Engineering
- CMPE Concurrency Lab
- Hands-on Project course

Penn ESE5320 Fall 2022 -- DeHon

26

26

Outcomes

- Design, optimize, and program a modern System-on-a-Chip.
- Analyze, identify bottlenecks, design-space
 - Modeling → write equations to estimate
- Decompose into parallel components
- Characterize and develop real-time solutions
- Implement both hardware and software solutions
- Formulate hardware/software tradeoffs, and perform hardware/software codesign

Penn ESE5320 Fall 2022 -- DeHon

27

27

Outcomes

- Understand the system on a chip from gates to application software, including:
 - on-chip memories and communication networks, I/O interfacing, design of accelerators, processors, firmware and OS/infrastructure software.
- Understand and *estimate* key design metrics and requirements including:
 - area, latency, throughput, energy, power, predictability, and reliability.

Penn ESE5320 Fall 2022 -- DeHon

28

28

Tools

- Are complex
- Will be challenging, but good for you to build confidence can understand and master
- Tool runtimes can be long
- Learning and sharing experience will be part of assignments

Penn ESE5320 Fall 2022 -- DeHon

29

29

Distinction

CIS2400, 4710, 5710

- Best Effort Computing
 - Run as fast as you can
- Binary compatible
- ISA separation
- Shared memory parallelism

ESE5320

- Hardware-Software codesign
 - Willing to recompile, maybe rewrite code
 - Define/refine hardware
- Real-Time
 - Guarantee meet deadline
- Non shared-memory parallelism models

Penn ESE5320 Fall 2022 -- DeHon

30

30

Distinction

ESE5390:

- Hardware/Software Co-Design for Machine Learning
- Deep on Application (ML)
 - More accessible to CS
 - Less previous experience with circuits and architecture
 - Won't be as deep on understanding HW and optimization
 - Program in Pytorch, OpenCL

ESE5320:

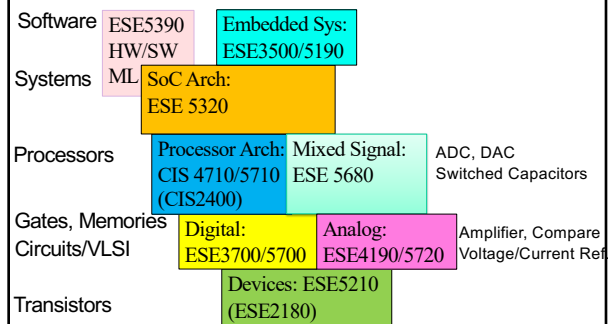
- Deep computer engineering
- Broad application
- Program in C
- Suitable followup if want to dig deeper

Penn ESE5320 Fall 2022 -- DeHon

31

31

Abstraction Stack



Penn ESE5320 Fall 2022 -- DeHon

32

32

Part 3: Approach -- Example

33

Penn ESE5320 Fall 2022 -- DeHon

33

Abstract Approach

- Identify requirements, bottlenecks
- Decompose Parallel Opportunities
 - At extreme, how parallel could make it?
 - What forms of parallelism exist?
 - Thread-level, data parallel, instruction-level
- Design space of mapping
 - Choices of where to map, area-time tradeoffs
- Map, analyze, refine
 - Write equations to understand, predict

34

Penn ESE5320 Fall 2022 -- DeHon

34

Example SPICE Circuit Simulator

Pass Transistor Cascade (after inverter reduction)

W=L=1 for each

35

Penn ESE5320 Fall 2022 -- DeHon

35

Example: SPICE Circuit Simulator

Matrix Solve
 $Ax=B$
 A matrix
 B vector
 x unknown vector
 Solve for x
 = KCL, KVL *

Linear Algebra
 solving n eqns
 in n unknowns.

Example: Kapre+DeHon, TRCAD 2012

* Kirchoff {Current,Voltage} Laws ³⁶

36

Penn ESE5320 Fall 2022 -- DeHon

36

Analyze

37

Penn ESE5320 Fall 2022 -- DeHon

37

Analyze

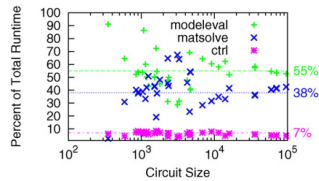
- $T = T_{modeval} + T_{matsolve} + T_{ctrl}$

38

Penn ESE5320 Fall 2022 -- DeHon

38

Speedup



- $T = T_{\text{modelevel}} + T_{\text{matsolve}} + T_{\text{ctrl}}$
- What should we speedup first?
- What happens if only speedup modelevel?
 - $T = T_{\text{matsolve}} + (T_{\text{modelevel}})/S + T_{\text{ctrl}}$

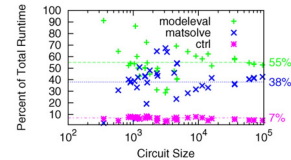
Penn ESE5320 Fall 2022 -- DeHon

39

39

Analyze

- If only accelerated model evaluation only about 2x speedup
- If want better than 14x speed, must also attack control



Penn ESE5320 Fall 2022 -- DeHon

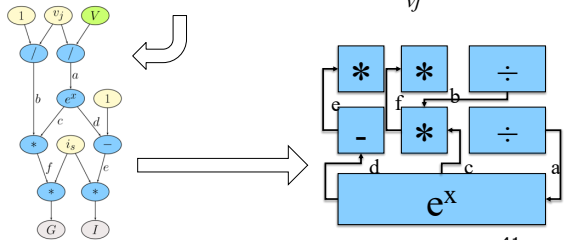
40

40

Model Evaluation: Trivial Hardware Implementation

$$I_{D1} = I_s \times (e^{V_{D1}/V_j} - 1)$$

$$G_{D1} = \frac{d}{dV_{D1}}(I_{D1}) = I_s \times e^{V_{D1}/V_j} \times \frac{1}{V_j}$$



Penn ESE5320 Fall 2022 -- DeHon

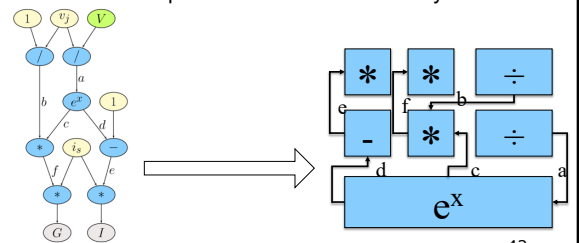
Verilog-AMS as Domain-Specific Language

41

41

Spatial, Pipelined Parallelism

- Every operation (*, +) gets dedicated hardware.
- Implement task in space → use additional area for each operator.
- Parallel – all operations occur simultaneously.



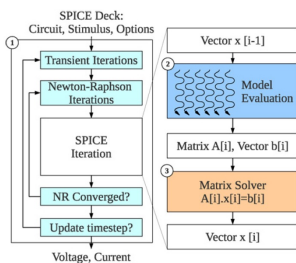
Penn ESE5320 Fall 2022 -- DeHon

42

42

Parallelism: Model Evaluation Data Parallel

- Every device independent
- Many of each type of device
- Can evaluate in parallel
 - $T = T_{\text{seq}}/N_{\text{proc}}$
- Build pipelined circuit for model
 - $T_{\text{seq}} = N_{\text{comp}} * T_{\text{cycle}}$
 - vs. $T_{\text{pipe}} = T_{\text{cycle}}$

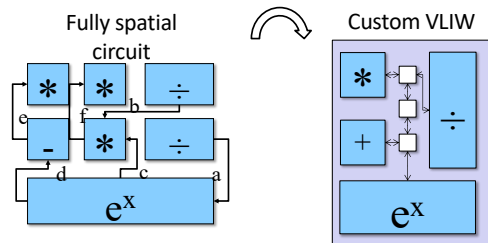


Penn ESE5320 Fall 2022 -- DeHon

43

43

Spatial Too Big?



~100x Speedup
Multiple FPGAs

~10x Speedup
1 FPGA (2010)

VLIW=Very Long Instruction Word
exploits Instruction-Level Parallelism

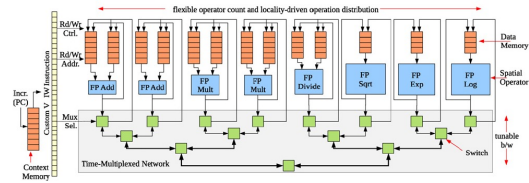
Penn ESE5320 Fall 2022 -- DeHon

44

44

Parallelism: Model Evaluation

- Spatial end up bottlenecked by other components
- Use custom evaluation engines
- ...or GPUs



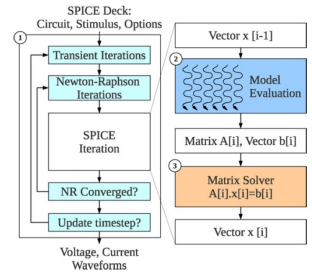
Penn ESE5320 Fall 2022 -- DeHon

45

45

Parallelism: Matrix Solve

- Needed direct solver?
- E.g. Gaussian elimination
- Data dependence on previous reduce
 - Limited data parallelism
- Parallelism in subtracts
- Some row independence

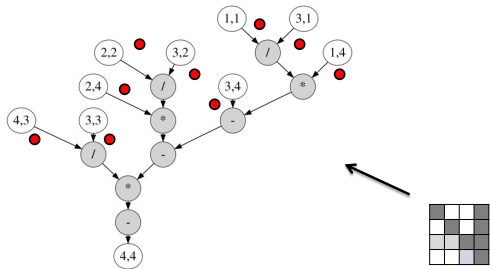


Penn ESE5320 Fall 2022 -- DeHon

46

46

Example Matrix

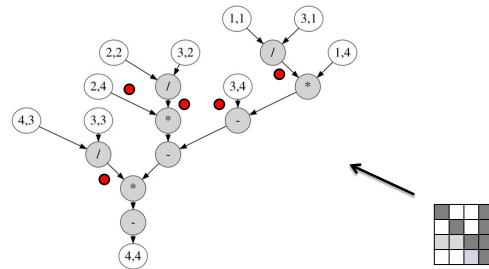


Penn ESE5320 Fall 2022 -- DeHon

47

47

Example Matrix

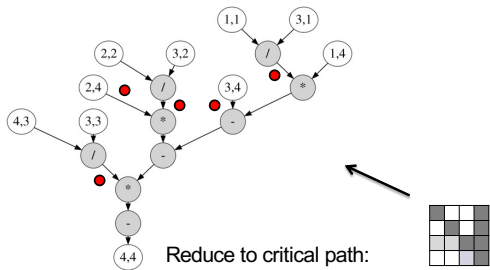


Penn ESE5320 Fall 2022 -- DeHon

48

48

Example Matrix



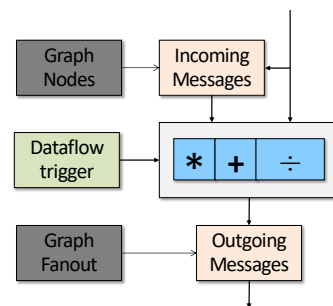
Reduce to critical path:
from 9 sequential operations
to path of 5 operations.

Penn ESE5320 Fall 2022 -- DeHon

49

49

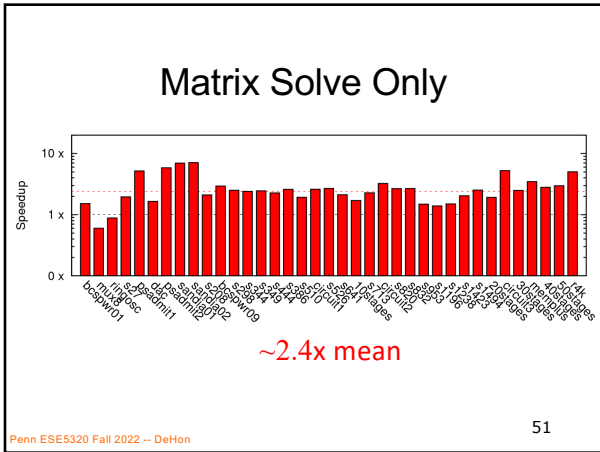
Dataflow Processing Element (PE)



Penn ESE5320 Fall 2022 -- DeHon

50

50



51

Parallelism: Matrix Solve

- Settled on constructing dataflow graph
- Graph can be iteration independent
 - Statically scheduled
 - (cheaper)
- This is bottleneck to further acceleration

Penn ESE5320 Fall 2022 -- DeHon

52

Parallelism Controller?

- Could leave sequential
- For some designs, becomes the bottleneck once others accelerated
- Has internal parallelism in condition evaluation

$$T = T_{\text{modeleval}}/S_1 + (T_{\text{matsolve}})/S_2 + T_{\text{ctrl}}$$

Penn ESE5320 Fall 2022 -- DeHon

53

Parallelism Controller

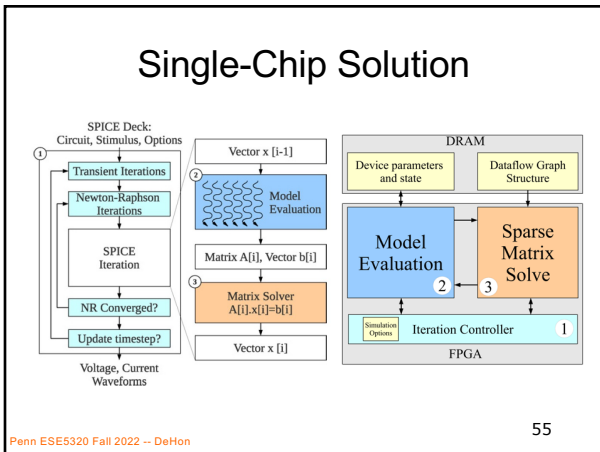
- Customized datapath controller

$$T_{\text{scqctrl}} = N_{\text{add}} + N_{\text{mul}} + 10 * N_{\text{divide}}$$

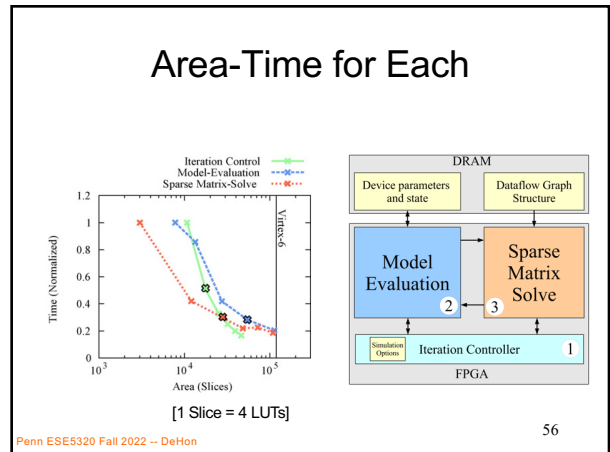
$$T_{\text{vliwctrl}} = \text{Max}(N_{\text{add}}/2, N_{\text{mul}}, 10 * N_{\text{divide}})$$

Penn ESE5320 Fall 2022 -- DeHon

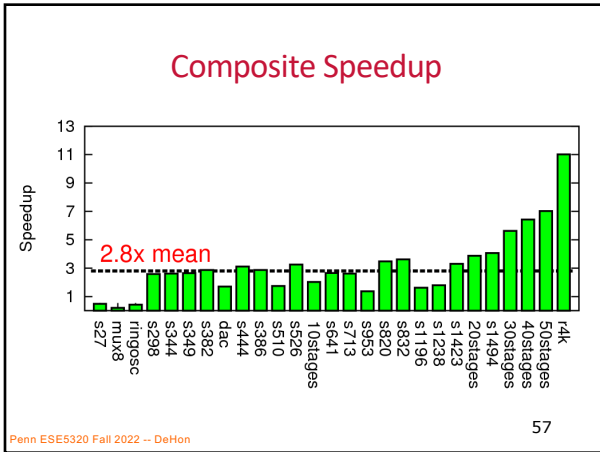
54



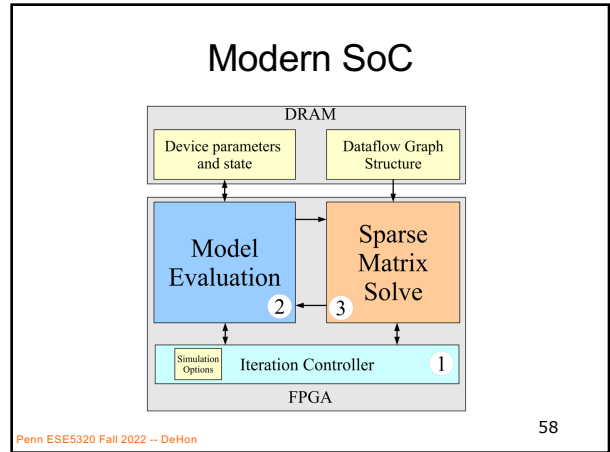
55



56



57

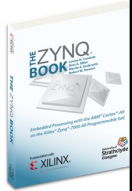


58

Part 4: Class Components

Penn ESE5320 Fall 2022 -- DeHon

59

- ### Class Components
- Lecture (incl. preclass exercise)
 - In-person (not hybrid, don't expect recordings)
 - Slides on web before class (print if you want)
 - N.B. I encourage class participation
 - In class; Questions ("warm" calls)
 - Daily Quiz
 - Reading [~1 required paper/lecture]
 - online: Canvas, IEEE, ACM, also ZynqBook, Parallel Programming for FPGAs
 - Homework
 - (1 per week due F5pm Eastern)
 - Project – open-ended (~6 weeks)
- 
- Penn ESE5320 Fall 2022 -- DeHon
- Note syllabus, course admin online

60

- ### First Half
- Quickly cover breadth
 - Metrics, bottlenecks
 - Memory
 - Parallel models
 - SIMD/Data Parallel
 - Thread-level parallelism
 - Spatial, C-to-gates
 - Line up with homeworks
- Penn ESE5320 Fall 2022 -- DeHon

61

- ### Second Half
- Use everything on project
 - Going deeper
 - Real-time
 - Reactive
 - Memory
 - Networking
 - Energy
 - Scaling
 - Chip Cost
 - Verification
 - VLIW
 - Reduce
- Penn ESE5320 Fall 2022 -- DeHon

62

Teaming

- HomeWorks (HW) in Groups of 2 (after 0, 1)
- HW: we assign
- Individual assignment writeup
- Project in Groups of 3
- Project: you propose team of 3, we review
 - Most portions group writeup
 - Few components individual writeup

Penn ESE5320 Fall 2022 -- DeHon

63

63

Office & Lab Hours

- Andre: M 4:00pm—5:00pm
 - Levine 270, Zoom
 - See canvas
- TAs – Ketterer (**plan, confirming...**)
 - Tuesday 7 pm
 - Wednesday 7 pm (not today)
 - Thursday 4-5pm, 8:15pm-9:15pm (first office hours tomorrow)

Penn ESE5320 Fall 2022 -- DeHon

64

64

Diagnostic Assessment

- Course will rely heavily on C
 - Program both hardware and software in C
- If you cannot read/write code in C, this course will be a challenge
- Diagnostic Assessment intended as a quick indication if you aren't ready
 - Should be able to complete quickly
 - Better to find out now than after you're stuck in the course
 - Due next Wednesday (9/7)

Penn ESE5320 Fall 2022 -- DeHon

65

65

C Review

- Course will rely heavily on C
 - Program both hardware and software in C
- HW1 has some C warmup problems
- TAs will hold C review
 - on Sept. 6th, 5:00pm
 - (before our next class meeting since Monday 9/5 is Labor day)

Penn ESE5320 Fall 2022 -- DeHon

66

66

Preclass Exercise

- Motivate the topic of the day
 - Introduce a problem
 - Introduce a design space, tradeoff, transform
- Available before lecture (11:10am)
 - Should work before lecture starts
 - Won't be available later
- Do bring/use calculator
 - Will be numerical examples

Penn ESE5320 Fall 2022 -- DeHon

67

67

Daily Quiz

- Count for Engagement Points
- Only available until next lecture
- Incentive to keep up with material

Penn ESE5320 Fall 2022 -- DeHon

68

68

Lecture Timeline

- Preclass available before class
 - In class hardcopy circa 10:10am
- Start lecture at 10:20am
- Lecture until 11:40am
- (most days) stay for remaining questions
 - Pending course after us
- **Masks required in lecture**

Penn ESE5320 Fall 2022 -- DeHon

69

Feedback

- Will have anonymous feedback for each lecture
 - Clarity?
 - Speed?
 - Vocabulary?
 - General comments
 - Specificity most helpful
 - X was unclear because of Y
 - Subtopic Z went too fast
 - Need an example for Q

Penn ESE5320 Fall 2022 -- DeHon

70

Policies

- Canvas turn-in of assignments
- No handwritten work
- Due on time
 - Individual assignments only
 - 3 free late days total
- Collaboration
 - Tools – allowed
 - Designs – limited to project teams as specified on assignments
- See web page

Penn ESE5320 Fall 2022 -- DeHon

71

Your action: Admin

- Feedback sheet for today
- Find course web page
 - Read it, including the policies
 - Find Syllabus
 - Find diagnostic assessment, homework 1
 - Find lecture slides
 - » Will try to post before lecture
 - Find reading assignments
- Find reading for lecture 2 on canvas and web
 - ...for this lecture if you haven't already
- Find/join Ed Discussion group for course
- Signup for detkin/ketterer access
- Complete/submit diagnostic assessment

Penn ESE5320 Fall 2022 -- DeHon

72

Big Ideas

- Programmable Platforms
 - Key delivery vehicle for innovative computing applications
 - Reduce TTM (Time-to-Market), risk
 - More than a microprocessor
 - Heterogeneous, parallel
- Demand hardware-software codesign
 - Soft view of hardware
 - Resource-aware view of parallelism

Penn ESE5320 Fall 2022 -- DeHon

73

Questions?

Penn ESE5320 Fall 2022 -- DeHon

74