

ESE532: System-on-a-Chip Architecture

Day 21: November 12, 2018
Energy



Today

Energy

- Today's bottleneck
- What drives
- Efficiency of
 - Processors, FPGAs, accelerators
- How does parallelism impact energy?

Message

- Energy dominates
 - Including limiting performance
- Make memories small and wires short
 - Small memories cost less energy per read
- Accelerators reduce energy
 - Compared to processors
- Can tune parallelism to minimize energy
 - Typically, the more parallel implementation costs less energy

Energy



- Growing domain of portables
 - Less energy/op → longer battery life
- Global Energy Crisis
- Power-envelope at key limit
 - E reduce → increase compute in P-envelope
 - Scaling
 - Power density **not** transistors limit sustained ops/s
 - Server rooms
 - Cost-of-ownership **not** dominated by Silicon
 - Cooling, Power bill

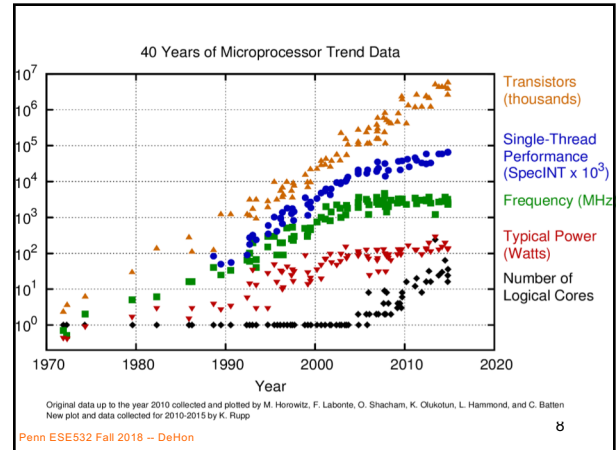
Preclass 1--4

- 200K gates/mm²
- 5*10⁻¹⁵ J/gate switch
- Gates on 1cm²
- Energy to switch all?
- Power at 1GHz?
- Fraction can switch with 10W/cm² power budget?

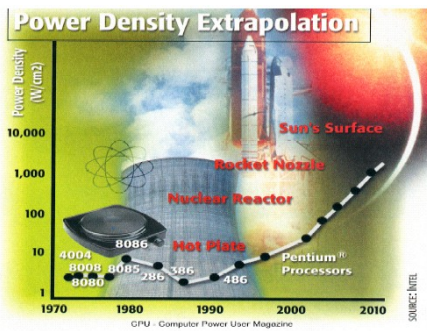
Challenge: Power

Origin of Power Challenge

- Limited capacity to remove heat
 - ~100W/cm² force air
 - 1-10W/cm² ambient
- Transistors per chip grow at Moore's Law rate = $(1/F)^2$
- Energy/transistor must decrease at this rate to keep constant power density
- $P/tr \propto CV^2f$
- $E/tr \propto CV^2$
 - ...but V scaling more slowly than F

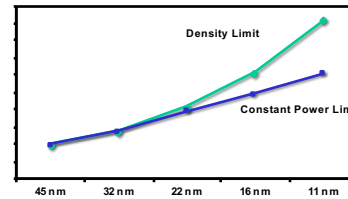


Intel Power Density



Impact

Power Limits Integration



Source: Carter/Intel

Impact

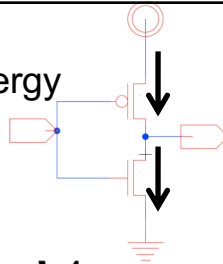
- Power density is limiting scaling
 - Can already place more transistors on a chip than we can afford to turn on!
- Power is potential challenge/limiter for all future chips.
 - Only turn on small percentage of transistors?
 - Operate those transistors at much slower frequency?

Energy

$$E_{total} = E_{switch} + E_{leak}$$

Leakage Energy

- I_{leak}
 - Subthreshold leakage
 - (possibly) Gate-Drain leakage



$$P_{leak} = I_{leak} \times V$$

$$E_{leak} = P_{leak} \times T$$

Penn ESE532 Fall 2018 -- DeHon

13

Switching Energy

$$E_{switch} \propto \alpha CV^2$$

- C – driven by architecture
- V – today, driven by variation, aging
- α – driven by architecture, coding/information

Penn ESE532 Fall 2018 -- DeHon

14

Energy

$$E_{total} = E_{switch} + E_{leak}$$

$$E_{switch} \propto \alpha CV^2$$

$$E_{leak} = I_{leak} \times V \times T$$

Penn ESE532 Fall 2018 -- DeHon

15

Voltage

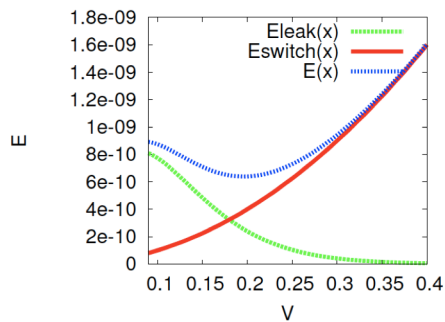
$$E_{switch} \propto \alpha CV^2 \quad E_{leak} = I_{leak} \times V \times T$$

- We can set voltage
- Reducing voltage
 - Reduces E_{switch}
 - Increases delay \rightarrow cycle time, T
 - Increases leakage

Penn ESE532 Fall 2018 -- DeHon

16

Total Energy vs. Voltage



Penn ESE532 Fall 2018 -- DeHon

17

Switching Energy

$$E_{switch} \propto \alpha CV^2$$

- C – driven by architecture
 - Also impacted by variation, aging
- V – today, driven by variation, aging
- α – driven by architecture, information

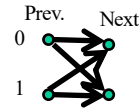
Penn ESE532 Fall 2018 -- DeHon

18

Data Dependent Activity

- Consider an 8b counter
 - How often do each of the following switch?
 - Low bit?
 - High bit?
 - Average switching across all 8 output bits?
- Assuming random inputs
 - Activity at output of nand4?
 - Activity at output of xor4?

Gate Output Switching (random inputs)



$$P_{switch} = P(0@i) * P(1@i+1) + P(1@i) * P(0@i+1)$$

Switching Energy

$$E_{switch} = \left(\sum_i \alpha_i C_i \right) V^2$$

C_i == capacitance driven by each gate (including wire)

Switching Rate (α_i) Varies

- Different logic (low/high bits, gate type)
- Different usage
 - Gate off unused functional units
- Data coded
- Entropy in data
- Average α 5--15% plausible

$$E_{switch} = \left(\sum_i \alpha_i C_i \right) V^2$$

Switching Energy

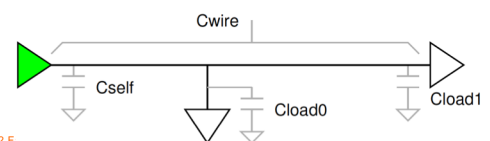
$$E_{switch} \propto \alpha C V^2$$

- C – driven by architecture
- V – today, driven by variation, aging
- α – driven by architecture, information

Wire Driven

$$E_{switch} = \left(\sum_i \alpha_i C_i \right) V^2$$

- Gates drive
 - Self
 - Inputs to other gates
 - Wire routing between self and other gates
- Typically: $C_{wire} > C_{self} + C_{load}$



Wire Capacitance

- How does wire capacitance relate to wire length?

Wire Capacitance

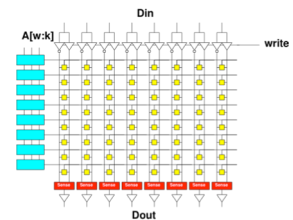
- $C = \epsilon A/d = \epsilon W * L_{\text{wire}}/d = C_{\text{unit}} * L_{\text{wire}}$
- Wire capacitance is linear in wire length
- E.g. 1.7pF/cm (preclass)
- Remains true if buffer wire
 - Add buffered segment at fixed lengths

Wire Driven Implications

- Care about locality
 - Long wires are higher energy
 - Producers near consumers
 - Memories near compute
 - Esp. for large α_i 's
- Care about size/area
 - Reduce (worst-case) distance must cross
- Care about minimizing data movement
 - Less data, less often, smaller distances
- Care about size of memories

Preclass 5

- Primary switching capacitance in wires
- C: How does energy of a read grow with capacity (N) of a memory bank?
- D: Energy per bit?



Memory Implications

- Memory energy can be expensive
- Small memories cost less energy than large memories
 - Use data from small memories as much as possible
- Cheaper to re-use data item from register than re-reading from memory

Architectural Implications

Component Numbers

Energy Per Operation (0.13µm, 1.2V)

TABLE 1

Operation	Energy
32-bit arithmetic operation	5 pJ
32-bit register read	10 pJ
32-bit 8KB RAM read	50 pJ
32-bit traverse 10mm wire	100 pJ
Execute instruction	500 pJ

Penn ESE532 Fall 2018 -- DeHon

[Dally, March 2004 ACM Queue] ³¹

Component Numbers

- Processor instruction 100x more than arithmetic
- Register read 2x
- RAM read 10x
- Why processor instruction > arith operation?

Energy Per Operation (0.13µm, 1.2V)

TABLE 1

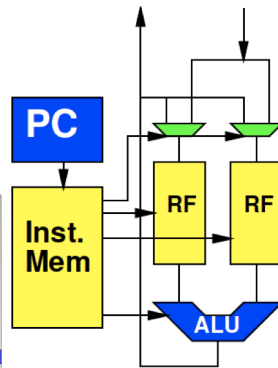
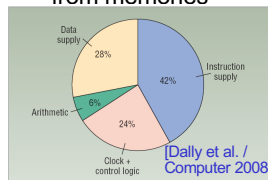
Operation	Energy
32-bit arithmetic operation	5 pJ
32-bit register read	10 pJ
32-bit 8KB RAM read	50 pJ
32-bit traverse 10mm wire	100 pJ
Execute instruction	500 pJ

Penn ESE532 Fall 2018 -- DeHon

[Dally, March 2004 ACM Queue] ³²

Processors and Energy

- Very little into actual computation
- Determine and Fetch Instruction
- Read and Write data from memories



P

ARM Cortex A9

Estimate find: 0.5W at 800MHz in 40nm

- $0.5/0.8 \times 10^{-9}$ J/instr
- ~600pJ/instr
- Scale to 28nm
 - maybe $0.7 \cdot 600$ — $0.5 \cdot 600$
 - 300—400pJ/instr ?
- Is superscalar w/ neon, so not as simple a processor as previous example

Penn ESE532 Fall 2018 -- DeHon

34

Zynq

Operation	PL Resource	ARM A9 Resource	ARM A9 energy/OP (pico Joules or mW/GOP/sec)	PL energy/OP (pico Joules or mW/GOP/sec)
Logical Op of 2 var	LUT/FF	ALU		1.3
32-bit ADD	LUT/FF	ALU		1.3
16x16 Mult	DSP	ALU		8.0
32-bit Read/Write register	LUTRAM	L1		1.4
32-bit Read/Write AXI register	LUT/FF	AXI		30
32-bit Read/Write local RAM	BRAM	L2		23.7/17.2
32-bit Read/Write OCM	AXI/OCM	CPU/OCM		44
32-bit Read/Write DDR3	AXI/DDR	CPU/DDR		541/211

- ARM A9 instruction 300—400pJ
- ARM A9 L1 cache read 23pJ

Penn ESE532 Fall 2018 -- DeHon

Xilinx UG585 – Zynq TRM ³⁵

Compare

- Assume ARM Cortex A9 executes 4x32b Neon vector add instruction for 300pJ
- Compare to 32b adds on FPGA?

Operation	PL Resource	ARM A9 Resource	ARM A9 energy/OP (pico Joules or mW/GOP/sec)	PL energy/OP (pico Joules or mW/GOP/sec)
Logical Op of 2 var	LUT/FF	ALU		1.3
32-bit ADD	LUT/FF	ALU		1.3
16x16 Mult	DSP	ALU		8.0
32-bit Read/Write register	LUTRAM	L1		1.4
32-bit Read/Write AXI register	LUT/FF	AXI		30
32-bit Read/Write local RAM	BRAM	L2		23.7/17.2
32-bit Read/Write OCM	AXI/OCM	CPU/OCM		44
32-bit Read/Write DDR3	AXI/DDR	CPU/DDR		541/211

P

Compare

- Assume ARM Cortex A9 executes 8x16b Neon vector multiply instruction for 300pJ
- Compare to 16x16 multiplies on FPGA?

Operation	PL Resource	ARM A9 Resource	ARM A9 energy/OP [pico Joules or mW/GOP/sec]	PL energy/OP [pico Joules op mW/GOP/sec]
Logical Op of 2 var	LUT/FF	ALU		1.3
32-bit ADD	LUT/FF	ALU		1.3
16x16 Mult	DSP	ALU		8.0
32-bit Read/Write register	LUTRAM	L1		1.4
32-bit Read/Write AXI register	LUT/FF	AXI		30
32-bit Read/Write local RAM	BRAM	L2		23.7/17.2
32-bit Read/Write OCM	AXI/OCM	CPU/OCM		44
32-bit Read/Write DDR3	AXI/DDR	CPU/DDR		541/211

Programmable Datapath

- Performing an operation in a pipelined datapath can be orders of magnitude less energy than on a processor
 - ARM 300pJ vs. 1.3pJ 32b add
 - Even neon 300pJ vs. 4x1.3pJ for 4x32b add
 - 300pJ vs. 8x8pJ for 8 16x16b multiplies

Penn ESE532 Fall 2018 -- DeHon

38

Zynq

Operation	PL Resource	ARM A9 Resource	ARM A9 energy/OP [pico Joules or mW/GOP/sec]	PL energy/OP [pico Joules op mW/GOP/sec]
Logical Op of 2 var	LUT/FF	ALU		1.3
32-bit ADD	LUT/FF	ALU		1.3
16x16 Mult	DSP	ALU		8.0
32-bit Read/Write register	LUTRAM	L1		1.4
32-bit Read/Write AXI register	LUT/FF	AXI		30
32-bit Read/Write local RAM	BRAM	L2		23.7/17.2
32-bit Read/Write OCM	AXI/OCM	CPU/OCM		44
32-bit Read/Write DDR3	AXI/DDR	CPU/DDR		541/211

- Reading from OCM order of magnitude less than from DRAM
- ...and BRAM half that

Penn ESE532 Fall 2018 -- DeHon

Xilinx UG585 – Zynq TRM 39

FPGA vs. Std Cell Energy

- 90nm
- FPGA: Stratix II
- STMicro CMOS090

TABLE VI
DYNAMIC POWER CONSUMPTION RATIO (FPGA/ASIC)

Name	Method	Logic Only	Logic & DSP	Logic & Memory	Logic, Memory & DSP
booth	Sim	26			
rs.encoder	Sim	52			
cordic18	Const	6.3			
cordic8	Const	5.7			
des.area	Const	27			
des.perf	Const	9.3			
fir.restruct	Const	9.6			
mac1	Const	19			
aes192	Sim	12			
fir3	Const	12	7.5		
diffeq	Const	15	12		
diffeq2	Const	16	12		
molecular	Const	15	16		
rs.decoder1	Const	13	16		
rs.decoder2	Const	11	11		
atm	Const			15	
aes	Sim			13	
aes_inv	Sim			12	
ethernet	Const			16	
serialproc	Const			16	
fir24	Const				5.3
pipeproc	Const				8.2
raytracer	Const				8.3
Geomean		14	12	14	7.1

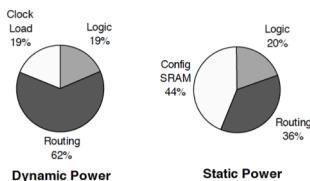
[Kuon/Rose TRCADv26n2p203–215 2007]

Penn ESE532 Fall 2018 -- DeHon

40

FPGA Disadvantage to Custom

- Interconnect Energy
 - Long wires → more capacitance → more E
 - Switch Energy is an overhead



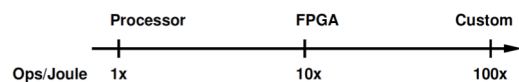
[Tuan et al./FPGA 2006]

41

Penn ESE532 Fall 2018 -- DeHon

Simplified Comparison

- Processor two orders of magnitude higher energy than custom accelerator
- FPGA accelerator in between
 - Order of magnitude lower than processor
 - Order of magnitude higher than custom



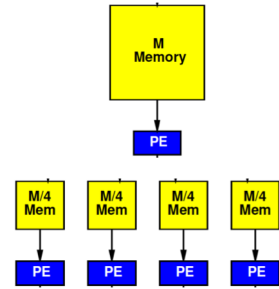
Penn ESE532 Fall 2018 -- DeHon

42

Parallelism and Energy

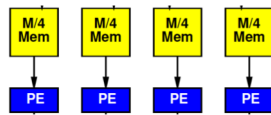
Preclass 6

- Energy
 - Per read from $M=10^6$ memory?
 - Per read from $10^6/4$ memory?



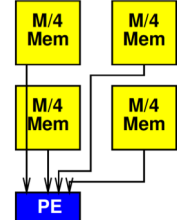
Local Consumption

- To exploit, we must consume the data local to the memory.



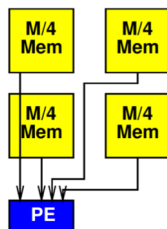
Cheat?

- What if we broke the memory into 4 blocks, but still routed to a single processor?
 - Hint: Interconnect energy reading from top right?



Cheat?

- $E = E_{mem}(M/4) + 2 * \sqrt{M/4} * C_{wire}$
- (simplify assume $2C_{wire} \approx 1$ unit)
- $E = \sqrt{M/4} + \sqrt{M/4} \approx \sqrt{M}$



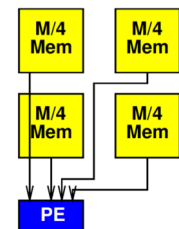
Cheat?

- Near mem = $E(M/4) = (1/2) \sqrt{M}$
- Far mem = \sqrt{M} [previous slide]
- Other two = $E(M/4) + \sqrt{M/4} * C_{wire} = (1/2 + 1/4) \sqrt{M}$

- Average:

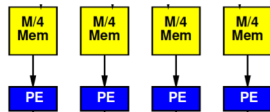
$$\sqrt{M} * (1/4) (1/2 + 2 * 3/4 + 1)$$

$$= 3/4 \sqrt{M}$$



Exploit Locality

- Must consume data near computation

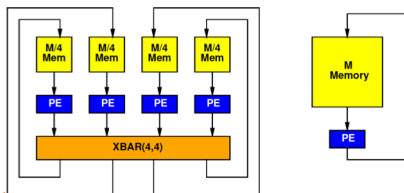


Inter PE Communication

- May need to communicate between parallel processing elements [PEs] (and memories)
- Must pay for energy to move data between PEs

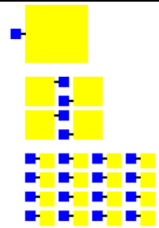
Preclass 7

- Energy: Read 4 memories $10^6/4$, route 4×4 crossbar, write $4 \times 10^6/4$ memories?
- Energy: 4 reads from 10^6 memory, 4 writes from 10^6 memory?



Parallel Larger

- More parallel design
 - Has more PEs
 - Adds interconnect
- Total area > less parallel design
 - More area → longer wires → more energy in communication between PEs
 - Could increase energy!



Continuum Question

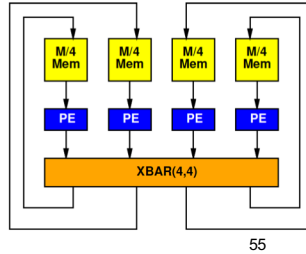
- Where do we minimize total energy?
 - Both memory and communication
- Design axis P – number of PEs
 - What P minimizes energy?

Simple Model

- $E_{\text{mem}} = \sqrt{M}$
- Communication = $E_{\text{xbar}}(I,O) = 4 * I * O$
- P Processors
- N total data
- Possibly communicate each result to other PEs

Simple Model: Memory

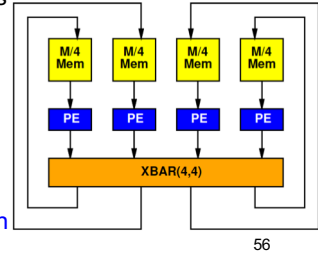
- Divide N data over P memories
- $E_{\text{mem}} = \sqrt{N/P}$
- N total memory operations
- Memory energy: $N \cdot \sqrt{N/P}$
- Memory energy decrease with P



Penn ESE532 Fall 2018 -- DeHon

Simple Model: Communication

- Crossbar with P inputs and P outputs
- $E_{\text{xbar}} = 4 \cdot P \cdot P$
- Crossbar used N/P times
- Crossbar energy: $4 \cdot N \cdot P$
- Communication energy increase with P

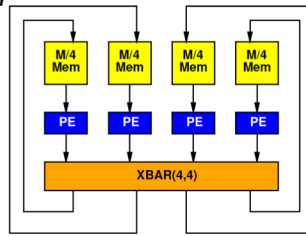


Penn ESE532 Fall 2018 -- DeHon

Simple Model

$$N \times \sqrt{\left(\frac{N}{P}\right)} + N \times 4 \times P$$

$$N \times \left(\sqrt{\left(\frac{N}{P}\right)} + 4 \times P \right)$$



Penn ESE532 Fall 2018 -- DeHon

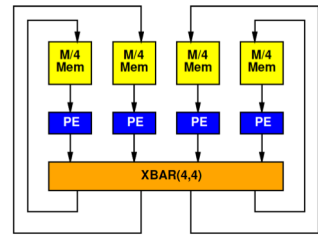
Preclass 8

- For $N=10^6$

$$N \times \left(\sqrt{\left(\frac{N}{P}\right)} + 4 \times P \right)$$

- Per operation becomes:

$$\frac{10^3}{\sqrt{P}} + 4P$$

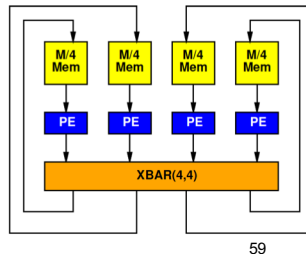


Penn ESE532 Fall 2018 -- DeHon

Preclass 8

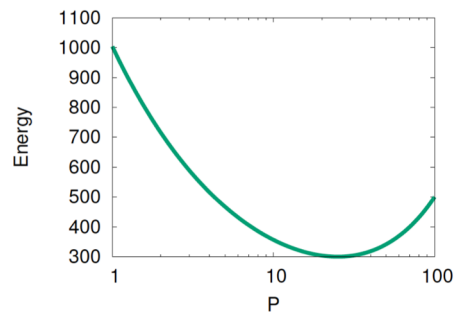
- Energy for:
 - P=1
 - P=4
 - P=100
- Energy minimizing P?
 - Energy?

$$\frac{10^3}{\sqrt{P}} + 4P$$



Penn ESE532 Fall 2018 -- DeHon

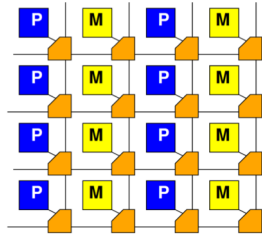
Graph



Penn ESE532 Fall 2018 -- DeHon

High Locality

- If communication is local, don't need crossbar
- Communication energy scales less than P^2
- Can scale as low as P



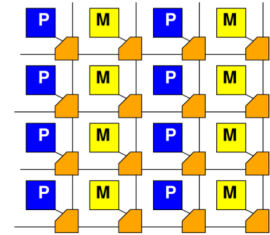
61

Penn ESE532 Fall 2018 -- DeHon

Model for High Locality

- $E_{\text{comm}} = \text{constant}$
- $E_{\text{comm}} = 10$
- Total comm: $N * 10$

$$N \times \left(\sqrt{\frac{N}{P}} + 10 \right)$$



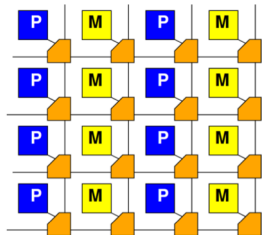
62

Penn ESE532 Fall 2018 -- DeHon

Preclass 9

- What is energy minimizing P ?

$$\frac{10^3}{\sqrt{P}} + 10$$



63

Penn ESE532 Fall 2018 -- DeHon

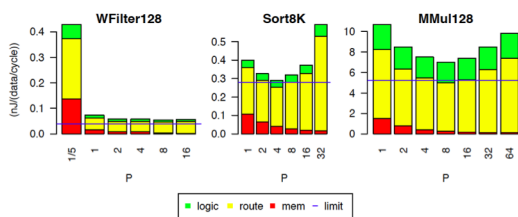
Task Locality Matters

- Optimal P depends on communication locality
 - Very local problems always benefit from parallelism
 - Highly interconnected problems must balance energies \rightarrow intermediate parallelism

64

Penn ESE532 Fall 2018 -- DeHon

Tune Parallelism: Stratix-IV

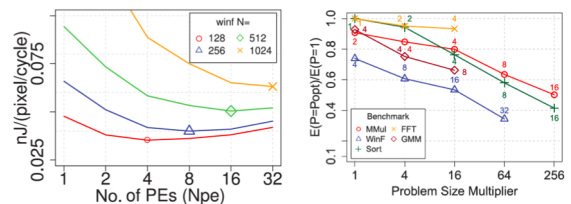


65

Penn ESE532 Fall 2018 -- DeHon

[Kadric, FPGA 2015]

PE Scaling with Problem Size



66

Penn ESE532 Fall 2018 -- DeHon

[Kadric, TRETS 2016]

Big Ideas

- Energy dominance
- With power-density budget
 - The most energy efficient architecture delivers the most performance
- Make memories small and wires short
- SoC, accelerators reduce energy by reducing processor instruction execution overhead
- Parallel design exploit locality → reduce energy
- Optimal parallelism for problem
 - Driven by communication structure, size

Penn ESE532 Fall 2018 -- DeHon

67

Admin

- P2 design space
 - Should have feedback on canvas
- Back to verification on Wednesday
- P3 I/O and Energy Milestone
 - Due Friday

Penn ESE532 Fall 2018 -- DeHon

68