

# ESE532: System-on-a-Chip Architecture

Day 23: April 12, 2017  
Parallelism and Energy



# Today

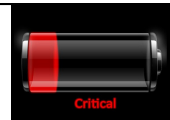
- How does parallelism impact energy?
- Refine
  - Locality?
  - Problem size?

# Message

- Can tune parallelism to minimize energy
- Typically, the more parallel implementation costs less energy

Day 22

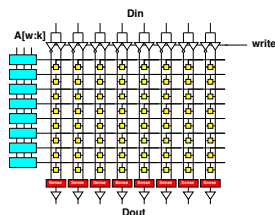
# Energy



- 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

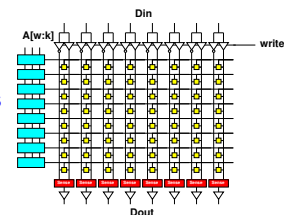
# Memory Energy

- Memory operations cost energy
- Must move data for memory bit to outside of array
- Wires of length  $\sqrt{N}$
- Energy  $\sqrt{N}$



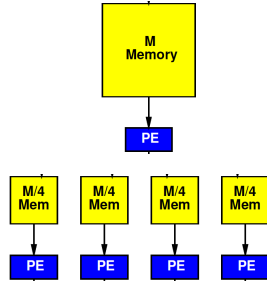
# Memory Energy

- Energy  $\sqrt{N}$
- Large memories cost more energy than small memories



## Preclass 1

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

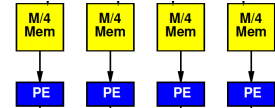


Penn ESE532 Spring 2017 -- DeHon

7

## Local Consumption

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

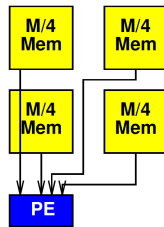


Penn ESE532 Spring 2017 -- DeHon

8

## Cheat?

- What if we broke the memory into 4 blocks, but still routed to a single processor?

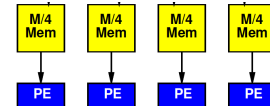


Penn ESE532 Spring 2017 -- DeHon

9

## Exploit Locality

- Must consume data near computation



Penn ESE532 Spring 2017 -- DeHon

10

## Inter PE Communication

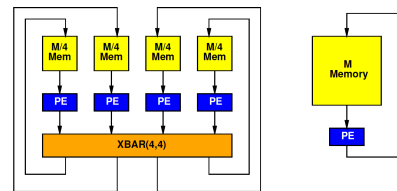
- May need to communicate between parallel processing units (and memories)
- Must pay for energy to move data between PEs

Penn ESE532 Spring 2017 -- DeHon

11

## Preclass 2

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

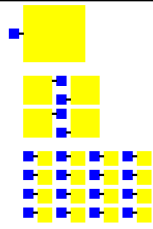


Penn ESE532 Spring 2017 -- DeHon

12

## 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

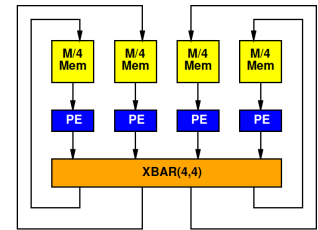
- Where do we minimize total energy?
  - Both memory and communication
- Design axis P – number of PEs
  - What P minimizes energy?

## 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

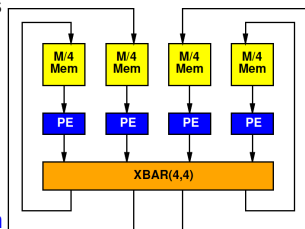
## Simple Model: Memory

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



## Simple Model: Communication

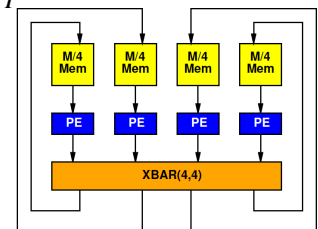
- Crossbar with P inputs and P outputs
- $E_{\text{xbar}} = 4 * P * P$
- Crossbar used N/P times
- Crossbar energy:  $4 * N * P$
- **Communication energy increase with P**



## Simple Model

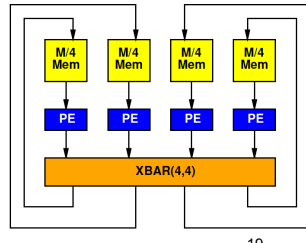
$$N \times \sqrt{\left(\frac{N}{P}\right)} + N \times 4 \times P$$

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



### Preclass 3

- For  $N=10^6$
- $$N \times \left( \sqrt{\frac{N}{P}} + 4 \times P \right)$$
- Per operation becomes:
- $$\left( \frac{10^3}{\sqrt{P}} \right) + 4 \times P$$



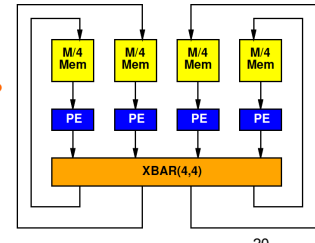
Penn ESE532 Spring 2017 -- DeHon

19

### Preclass 3

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

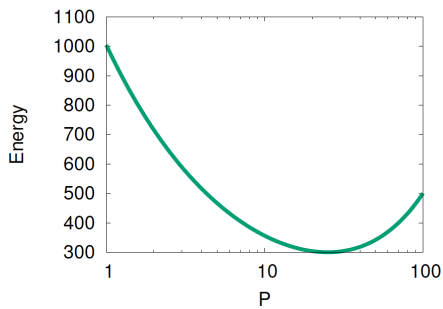
$$\left( \frac{10^3}{\sqrt{P}} \right) + 4 \times P$$



Penn ESE532 Spring 2017 -- DeHon

20

### Graph



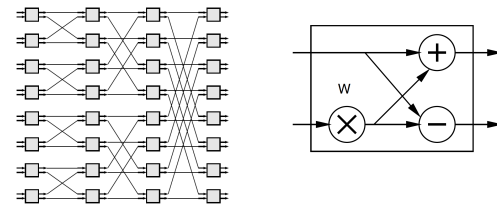
Penn ESE532 Spring 2017 -- DeHon

21

### Day 14

### FFT

- Large space of FFTs
- Radix-2 FFT Butterfly



Penn ESE532 Spring 2017 -- DeHon

22

### FFT Example

- Tune number of PEs,  $P$

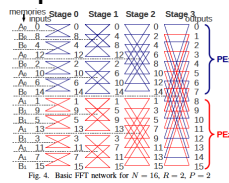


Fig. 4. Basic FFT network for  $N = 16, R = 2, P = 2$

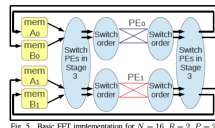


Fig. 5. Basic FFT implementation for  $N = 16, R = 2, P = 2$

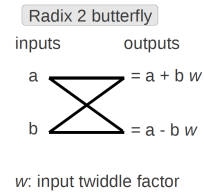
Penn ESE532 Spring 2017 -- DeHon

[Kadric, FCCM 2014]

23

### FFT on Stratix-IV

- PE is Radix 2 Butterfly



$w$ : input twiddle factor

PEs	Hier	Mem	Components (per point)			Total	
			Type	Mem	route		comp
1	1	M144K	2300	140	900	1300	4700
1	16	M144K	1300	210	920	1400	3800
4	4	M9K	880	75	1000	1500	3500
16	1	M9K	420	85	900	1200	2600
32	1	M9K	210	85	910	1200	2400

DeHon--Altera, Feb. 2015

[Kadric, FCCM 2014]

24

### FFT

- PE is Radix 2 Butterfly

Radix 2 butterfly

inputs outputs

$$\begin{matrix} a \\ b \end{matrix} \begin{matrix} \diagup \\ \diagdown \end{matrix} \begin{matrix} = a + b w \\ = a - b w \end{matrix}$$

w: input twiddle factor

PEs	Hier	Mem	Components (per point)				Total
			Mem	route	comp	route	
1	1	M144K	2300	140	900	1300	4700
1	16	M144K	1300	210	920	1400	3800
4	4	M9K	880	75	1000	1500	3500
16	1	M9K	420	85	900	1200	2600
32	1	M9K	210	85	910	1200	2400

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014] 25

### FFT

- PE is Radix 2 Butterfly

Radix 2 butterfly

inputs outputs

$$\begin{matrix} a \\ b \end{matrix} \begin{matrix} \diagup \\ \diagdown \end{matrix} \begin{matrix} = a + b w \\ = a - b w \end{matrix}$$

w: input twiddle factor

PEs	Hier	Mem	Components (per point)				Total
			Mem	route	comp	route	
1	1	M144K	2300	140	900	1300	4700
1	16	M144K	1300	210	920	1400	3800
4	4	M9K	880	75	1000	1500	3500
16	1	M9K	420	85	900	1200	2600
32	1	M9K	210	85	910	1200	2400

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014] 26

### FFT

- PE is Radix 2 Butterfly

Radix 2 butterfly

inputs outputs

$$\begin{matrix} a \\ b \end{matrix} \begin{matrix} \diagup \\ \diagdown \end{matrix} \begin{matrix} = a + b w \\ = a - b w \end{matrix}$$

w: input twiddle factor

PEs	Hier	Mem	Components (per point)				Total
			Mem	route	comp	route	
1	1	M144K	2300	140	900	1300	4700
1	16	M144K	1300	210	920	1400	3800
4	4	M9K	880	75	1000	1500	3500
16	1	M9K	420	85	900	1200	2600
32	1	M9K	210	85	910	1200	2400

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014] 27

### FFT

- PE is Radix 2 Butterfly

Radix 2 butterfly

inputs outputs

$$\begin{matrix} a \\ b \end{matrix} \begin{matrix} \diagup \\ \diagdown \end{matrix} \begin{matrix} = a + b w \\ = a - b w \end{matrix}$$

w: input twiddle factor

PEs	Hier	Mem	Components (per point)				Total
			Mem	route	comp	route	
1	1	M144K	2300	140	900	1300	4700
1	16	M144K	1300	210	920	1400	3800
4	4	M9K	880	75	1000	1500	3500
16	1	M9K	420	85	900	1200	2600
32	1	M9K	210	85	910	1200	2400

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014] 28

### Tune Parallelism: Stratix-IV

WFilter128, Sort8K, MMul128

(nJ/data/cycle)

P

■ logic ■ route ■ mem ■ limit

DeHon--Altera, Feb. 2015 [Kadric, FPGA 2015] 29

### GMM

- Compute Gaussian model of pixel background for every pixel

PEs	Mem	Components (per PE)			Total
		Type	Mem	route	
1	M144K	29	4.4	9.8	28
2	M144K	27	6.8	9.5	16
4	M9K	12	1.3	9.9	19
8	M9K	4.7	1.0	10	17
16	M9K	4.6	1.0	9.8	13
32	M9K	5.7	3.7	9.3	8

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014] 30

### GMM

- Compute Gaussian model of pixel background for **every** pixel

PEs	Mem		Components (per PE)				Total
	Type	Mem	route	Comp	route	mW/PE	
1	M144K	29	4.4	9.8	28		71
2	M144K	27	6.8	9.5	16		59
4	M9K	12	1.3	9.9	19		42
8	M9K	4.7	1.0	10	17		33
16	M9K	4.6	1.0	9.8	13		29
32	M9K	5.7	3.7	9.3	8		27

31

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014]

### GMM

- Compute Gaussian model of pixel background for **every** pixel

PEs	Mem		Components (per PE)				Total
	Type	Mem	route	Comp	route	mW/PE	
1	M144K	29	4.4	9.8	28		71
2	M144K	27	6.8	9.5	16		59
4	M9K	12	1.3	9.9	19		42
8	M9K	4.7	1.0	10	17		33
16	M9K	4.6	1.0	9.8	13		29
32	M9K	5.7	3.7	9.3	8		27

32

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014]

### GMM

- Compute Gaussian model of pixel background for **every** pixel

PEs	Mem		Components (per PE)				Total
	Type	Mem	route	Comp	route	mW/PE	
1	M144K	29	4.4	9.8	28		71
2	M144K	27	6.8	9.5	16		59
4	M9K	12	1.3	9.9	19		42
8	M9K	4.7	1.0	10	17		33
16	M9K	4.6	1.0	9.8	13		29
32	M9K	5.7	3.7	9.3	8		27

33

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014]

### GMM

- Compute Gaussian model of pixel background for **every** pixel

PEs	Mem		Components (per PE)				Total
	Type	Mem	route	Comp	route	mW/PE	
1	M144K	29	4.4	9.8	28		71
2	M144K	27	6.8	9.5	16		59
4	M9K	12	1.3	9.9	19		42
8	M9K	4.7	1.0	10	17		33
16	M9K	4.6	1.0	9.8	13		29
32	M9K	5.7	3.7	9.3	8		27

34

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014]

### Window Filter

a) 5x5 Gaussian Filter (Coefficients shown)

b) Single memory (C cycles per pixel)

c) Add 4 Line buffers (1 pixel per cycle)

d) Add 3 PEs (total of 4) (4 pixels per cycle)

PEs	Line	Bits	Mem	Components (per pixel)				Total
				Mem	route	comp	route	
1	N	1	M144K	400	50	130	280	850
1	Y	1	M9K	110	11	81	120	320
2	Y	1	M9K	45	6	67	84	200
4	Y	1	M9K	24	3	61	75	160
4	Y	2	M9K	15	3	59	63	140
4	Y	4	M9K	15	3	57	60	135

35

DeHon--Altera, Feb. 2015 [Kadric, FCCM 2014]

### High Locality

- If communication is local, don't need crossbar
- Communication energy scales less than P<sup>2</sup>
- Can scale as low as P
- As see GMM, WinF

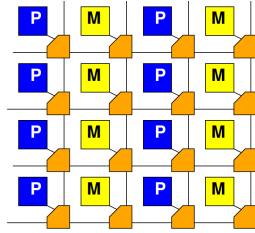
36

Penn ESE532 Spring 2017 -- DeHon

## Model for High Locality

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

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



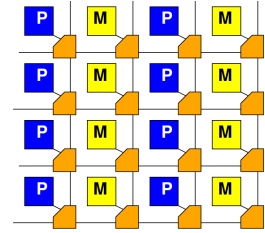
37

Penn ESE532 Spring 2017 -- DeHon

## Preclass 4

- What is energy minimizing P?

$$\left( \frac{10^3}{\sqrt{P}} \right) + 10$$



38

Penn ESE532 Spring 2017 -- DeHon

## Task Locality Matters

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

39

Penn ESE532 Spring 2017 -- DeHon

## Impact of Problem Size

- Optimal P changes with problem size
- Changes N
  - Changes cost of memories

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

40

Penn ESE532 Spring 2017 -- DeHon

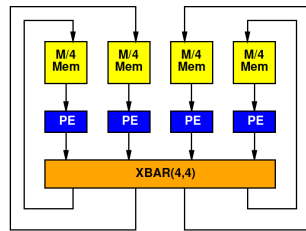
## Preclass 5

- For  $N = 6.4 \times 10^7$

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

- Per operation becomes:

$$\left( \frac{8 \times 10^3}{\sqrt{P}} \right) + 4 \times P$$



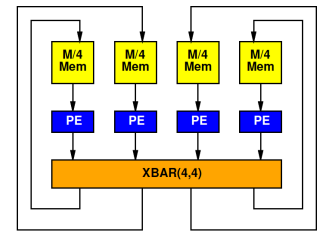
41

Penn ESE532 Spring 2017 -- DeHon

## Preclass 5

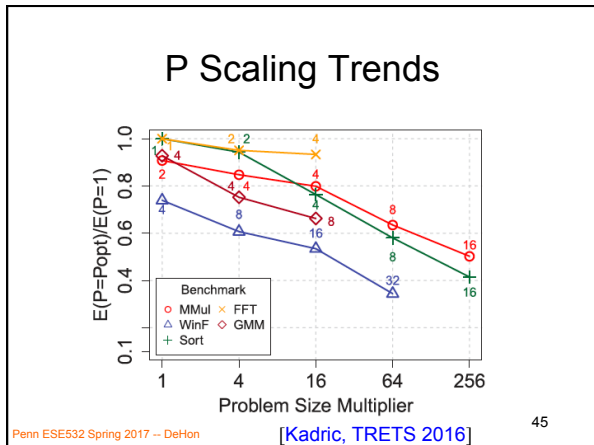
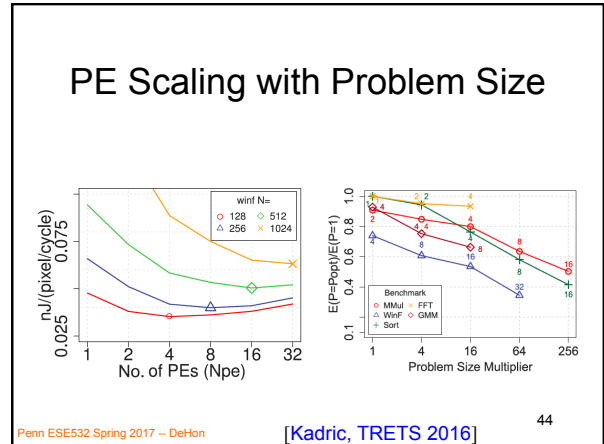
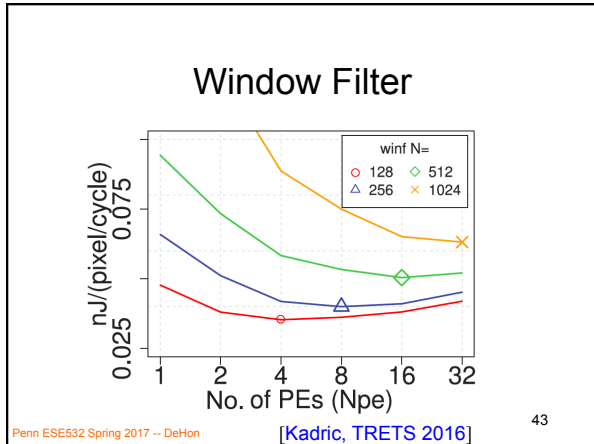
- Energy minimizing P?

$$\left( \frac{8 \times 10^3}{\sqrt{P}} \right) + 4 \times P$$



42

Penn ESE532 Spring 2017 -- DeHon



### Balanced Memory and Compute

- In general
  - Want memory to be about the size of the PE
  - Then replicate that to handle larger problems
    - Rather than making memory larger

The diagram shows a 3x8 grid of squares. The top row has 4 yellow squares followed by 4 blue squares. The middle row has 2 yellow squares followed by 6 blue squares. The bottom row has 1 yellow square followed by 7 blue squares. This illustrates a balanced memory and compute architecture where memory size is comparable to compute size.

Penn ESE532 Spring 2017 -- DeHon 46

### Balanced Memory and Compute

- Memory  $\approx$  Compute size
  - $E_{mem}$  comparable to  $E_{route}$ 
    - Doesn't make local computation much larger
  - If does need distant memory
    - Not much worse than one large memory

The diagram shows a 3x8 grid of squares. The top row has 4 yellow squares followed by 4 blue squares. The middle row has 2 yellow squares followed by 6 blue squares. The bottom row has 1 yellow square followed by 7 blue squares. This illustrates a balanced memory and compute architecture where memory size is comparable to compute size.

Penn ESE532 Spring 47

### Balanced Memory and Compute

- Memory  $\gg$  PE size
  - Not exploiting locality

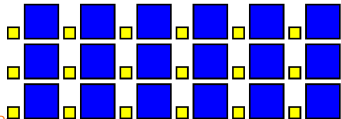
The diagram shows a large yellow square and a small blue square. This illustrates an architecture where memory size is much larger than the PE size, leading to poor locality.

Penn ESE532 Spring 2017 -- DeHon 48



## Balanced Memory and Compute

- Memory  $\gg$  PE size
  - Not exploiting locality
- Memory  $\ll$  PE size
  - Energy routing to distant memory can be much larger than if each memory bank was larger

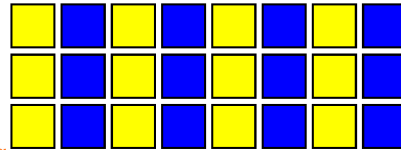


Penn ESE532 Spring 2017 -- DeHon

49

## Balanced Memory and Compute

- In general
  - Want memory to be about the size of the PE
  - Then replicate that to handle larger problems
    - Rather than making memory larger
    - $P \propto$  Problem Size



Penn ESE532 Spring 2017 -- DeHon

50

## Big Ideas

- Large memories are energy expensive
  - Force large data movement
- Small memories with local compute
  - Reduce data movement, save energy
- Parallel design exploiting locality can reduce energy
- Optimal parallelism for problem
  - Driven by communication structure
  - Depends on problem size

Penn ESE532 Spring 2017 -- DeHon

51

## Admin

- Project energy Milestone
  - Due Friday

Penn ESE532 Spring 2017 -- DeHon

52