

# ESE5320: System-on-a-Chip Architecture

Day 19: November 4, 2024  
Design-Space Exploration



Penn ESE5320 Fall 2024 -- DeHon

1

## Today

- Design-Space Exploration
  - Generic (Part 1)
  - Concrete example: (Part 2)
    - Fast Fourier Transform (FFT)

Penn ESE5320 Fall 2024 -- DeHon

2

2

## Message

- The universe of possible implementations (design space) is large
  - Many dimensions to explore
- Formulate carefully
- Approach systematically
- Use modeling along the way for guidance

Penn ESE5320 Fall 2024 -- DeHon

3

3

## Design-Space Exploration

Generic

Penn ESE5320 Fall 2024 -- DeHon

4

4

## Design Space

- Have many choices for implementation
  - Alternatives to try
  - Parameters to tune
  - Mapping options
- This is our freedom to impact implementation costs
  - Area, delay, energy

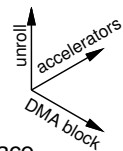
Penn ESE5320 Fall 2024 -- DeHon

5

5

## Design Space

- Ideally
  - Each choice orthogonal axis in high-dimensional space
  - Want to understand points in space
  - Find one that best meets constraints and goals
- Practice
  - Seldom completely orthogonal
  - Requires cleverness to identify dimensions
  - Messy, cannot fully explore
  - But... can understand, prioritize, guide



Penn ESE5320 Fall 2024 -- DeHon

6

6

## Preclass 1

- What choices (design-space axes) can we explore in mapping a task to an SoC?
- What showed up in homework so far?

Penn ESE5320 Fall 2024 -- DeHon

7

7

## From Homework?

- Types of parallelism
- Mapping to different fabrics / hardware
- How manage memory, move data
  - DMA, streaming
  - Data access patterns
- Levels of parallelism
- Pipelining, unrolling, II, array partitioning
- Data size (precision)

Penn ESE5320 Fall 2024 -- DeHon

8

8

## Generalize Continuum

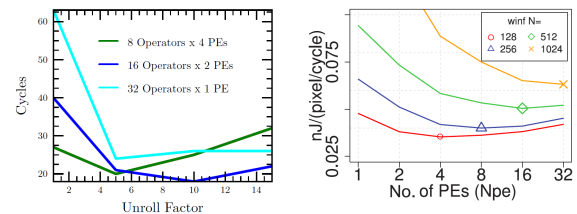
- Encourage to think about parameters (axes) that capture continuum to explore
- Start from an idea
  - Maybe can compute with 8b values
  - Maybe can put matrix-mpy computation on FPGA fabric
  - Move data in 1KB chunks
- Identify general knob
  - Tune intermediate bits for computation
  - How much of computation go on FPGA fabric
  - What is optimal data transfer size?

Penn ESE5320 Fall 2024 -- DeHon

10

10

## Finding Optima



- Kapre, FPL 2009
  - Kadric, TRETs 2016
- PE = Processing Element

Penn ESE5320 Fall 2024 -- DeHon

11

11

## Design Space Explore

- Think systematically about how might map the application
- Avoid overlooking options
- Understand tradeoffs
- The larger the design space
  - more opportunities to find good solutions
  - Reduce bottlenecks

Penn ESE5320 Fall 2024 -- DeHon

12

12

## Elaborate Design Space

- Refine design space as you go
- Ideally identify up front
- Practice bottlenecks and challenges
  - will suggest new options / dimensions
    - If not initially expect memory bandwidth to be a bottleneck...
- Some options only make sense in particular sub-spaces
  - Bitwidth optimization not a big issue on the 64b processor
    - More interesting on vector, FPGA

Penn ESE5320 Fall 2024 -- DeHon

13

13

## Tools

- Sometimes tools will directly help you explore design space
  - Unrolling, pipelining, II
  - Array packing and partitioning
  - Some choices for data movement
  - DMA pipelining and transfer sizes
  - Some loop transforms
  - Granularity to place on FPGA

Penn ESE5320 Fall 2024 -- DeHon

14

14

## Tools

- Often tools will not help you with design space options
  - Need to reshape functions and loops
  - Line buffers
  - Data representations and sizes
  - C-slow sharing
  - Communications overlap
  - Picking hash function parameters

Penn ESE5320 Fall 2024 -- DeHon

15

15

## Code for Exploration

- Can you write your code with parameters (#define) that can easily change to explore continuum?
  - Unroll factor?
  - Number of parallel tasks?
  - Size of data to move?
- Want to make it easy to explore different points in space

Penn ESE5320 Fall 2024 -- DeHon

16

16

## Part 2: Design-Space Exploration

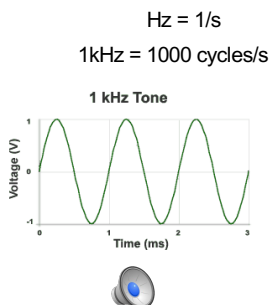
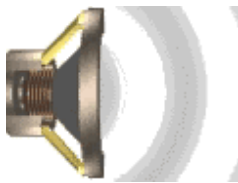
Example FFT

Penn ESE5320 Fall 2024 -- DeHon

17

17

## Sound Waves



Source: <http://www.mediacollege.com/audio/01/sound-waves.html>

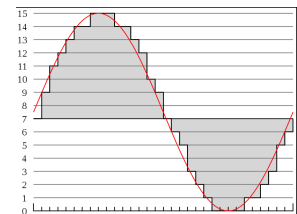
Penn ESE5320 Fall 2024 -- DeHon

18

18

## Discrete Sampling

- Represent as time sequence
- Discretely sample in time
- What we can do directly with an Analog-to-Digital (A2D) converter



<http://en.wikipedia.org/wiki/File:Pcm.svg>

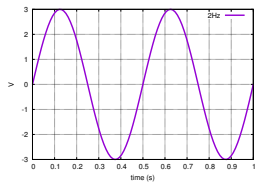
Penn ESE5320 Fall 2024 -- DeHon

19

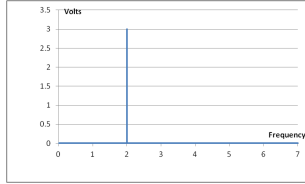
19

## Time-Domain & Frequency-domain

- **example...have a pure tone**
  - If period:  $T = 1/2$  and **Amplitude = 3 Volts**
  - $s(t) = A \sin(2\pi ft) = A \sin(2\pi t)$



Time domain representation



Frequency domain representation

Penn ESE5320 Fall 2024 -- DeHon

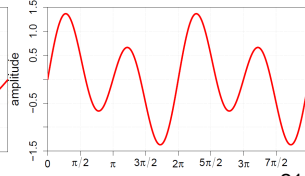
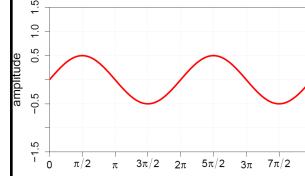
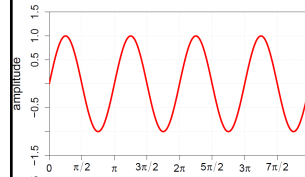
20

20



## Frequency-domain

- Can represent sound wave as linear sum of frequencies



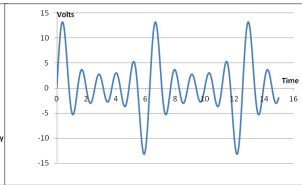
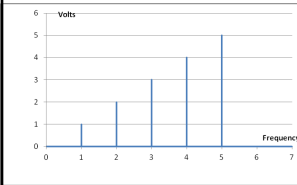
Penn ESE5320 Fall 2024 -- DeHon

21

21



## Time vs. Frequency

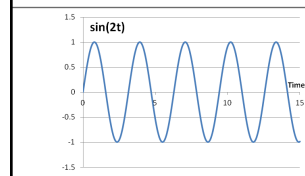
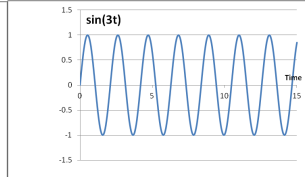
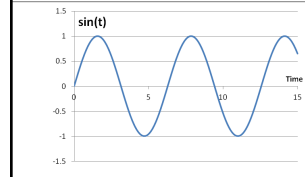


Penn ESE5320 Fall 2024 -- DeHon

22

22

## Fourier Series



- The  $\cos(nx)$  and  $\sin(nx)$  functions form an orthogonal basis: they allow us to represent any periodic signal by taking a linear combination of the basis components without interfering with one another

Penn ESE5320 Fall 2024 -- DeHon

23

23

## Fourier Transform

- Identify spectral components (frequencies)
- Convert between Time-domain to Frequency-domain
  - E.g. tones from data samples
  - Central to audio coding – e.g. MP3 audio

$$Y[k] = \sum_{j=0}^{n-1} \left( X[j] e^{-2i\pi \frac{k}{n}} \right)$$

Constant

Penn ESE5320 Fall 2024 -- D

24

24

## FT as Matching

- Fourier Transform is essentially performing a dot product with a frequency
  - How much like a sine wave of freq.  $f$  is this?

$$Y[k] = \sum_{j=0}^{n-1} \left( X[j] e^{-2i\pi \frac{k}{n}} \right)$$

Constant

Penn ESE5320 Fall 2024 -- D

25

25

## Fast-Fourier Transform (FFT)

- Efficient way to compute FT
- $O(N \cdot \log(N))$  computation
- Contrast  $N^2$  for direct computation
  - $N$  dot products
    - Each dot product has  $N$  points (multiply-adds)

$$Y[k] = \sum_{j=0}^{n-1} \left( X[j] e^{-2i\pi \frac{jk}{n}} \right)$$

Constant

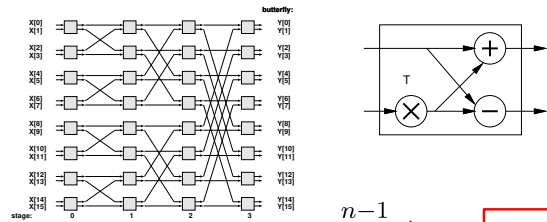
Penn ESE5320 Fall 2024 -- D

26

26

## FFT

- Large space of FFTs
- Radix-2 FFT Butterfly



$$Y[k] = \sum_{j=0}^{n-1} \left( X[j] e^{-2i\pi \frac{jk}{n}} \right)$$

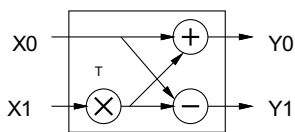
T

Penn ESE5320 Fall 2024 -- DeHon

27

## Basic FFT Butterfly

- $Y_0 = X_0 + T(\text{stage, butterfly}) * X_1$
- $Y_1 = X_0 - T(\text{stage, butterfly}) * X_1$
- Common sub expression, compute once:  $T(\text{stage, butterfly}) * X_1$



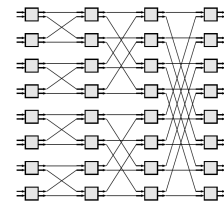
Penn ESE5320 Fall 2024 -- DeHon

28

28

## Preclass 2

- What parallelism options exist?
  - Single FFT
  - Sequence of FFTs



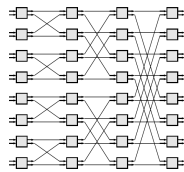
Penn ESE5320 Fall 2024 -- DeHon

29

29

## FFT Parallelism

- Spatial
- Pipeline
- Streaming
- By column
  - Choose how many Butterflies to serialize on a PE (processing element)
- By subgraph
- Pipeline subgraphs



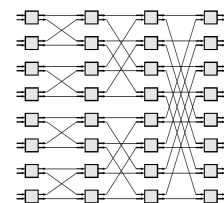
Penn ESE5320 Fall 2024 -- DeHon

30

30

## Preclass 3

- How large of a spatial FFT can implement with 360 multipliers?



Penn ESE5320 Fall 2024 -- DeHon

31

31

## Bit Serial

- Could compute the add/multiply bit serially
  - One full adder per adder
  - $W$  full adders per multiply
  - $W=16$ , maybe 20—30 LUTs
  - 70,000 LUTs
    - $\approx 70,000/30 \approx 2330$  butterflies
    - 512-point FFT has 2304 butterflies
- Another dimension to design space:
  - How much serialize word-wide operators
  - Use LUTs vs. DSPs

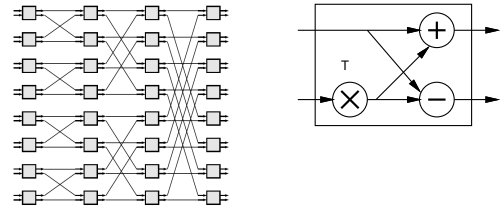
Penn ESE5320 Fall 2024 -- DeHon

32

32

## Accelerator Building Blocks

- What common subgraphs exist in the FFT?

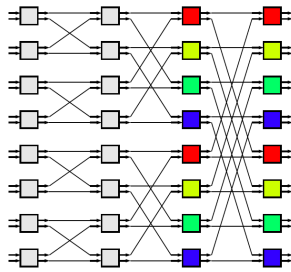


Penn ESE5320 Fall 2024 -- DeHon

33

33

## Common Subgraphs



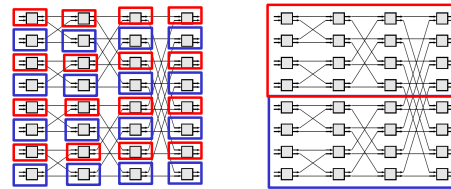
Penn ESE5320 Fall 2024 -- DeHon

34

34

## Processor Mapping

- How map butterfly operations to processors?
  - Implications for communications?



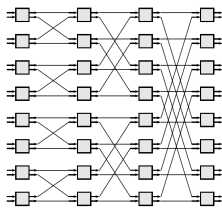
Penn ESE5320 Fall 2024 -- DeHon

35

35

## Preclass 4a

- How large local memory to communicate from stage to stage?



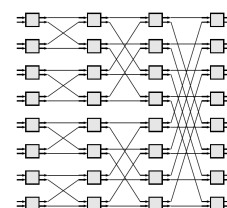
Penn ESE5320 Fall 2024 -- DeHon

36

36

## Preclass 4b

- How change evaluation order to reduce local storage memory?



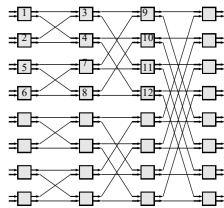
Penn ESE5320 Fall 2024 -- DeHon

37

37

## Preclass 4b

- Evaluation order

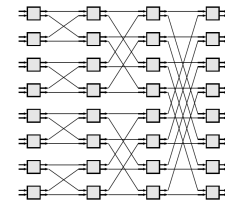
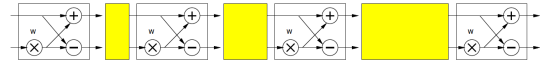


Penn ESE5320 Fall 2024 -- DeHon

38

38

## Streaming FFT



Penn ESE5320 Fall 2024 -- DeHon

39

39

## Communication

- How implement the data shuffle between processors or accelerators?
  - Memories / interconnect ?
  - How serial / parallel ?
  - Network?

Penn ESE5320 Fall 2024 -- DeHon

40

40

## Data Precision

- Input data from A2D likely 12b
- Output data, may only want 16b
- What should internal precision and representation be?

Penn ESE5320 Fall 2024 -- DeHon

41

41

## Number Representation

- Floating-Point
  - IEEE standard single (32b), double (64b)
    - With mantissa and exponent
    - ...half, quad ...
- Fixed-Point
  - Select total bits and fraction
    - E.g. 16.8 (16 total bits, 8 of which are fraction)
      - Represent 1/256 to 256-1/256
  - $A(\text{mpy}) \sim W^2$ ,  $A(\text{add}) \sim W$ 
    - $W$  here width – not FFT weight value

Penn ESE5320 Fall 2024 -- DeHon

42

42

## Operator Sizes

Operator	LUTs	LUTs + DSPs
Double FP Add	712	681+3 DSPs
Single FP Add	370	219+2 DSPs
Fixed-Point Add (32)	16	
Fixed-Point Add (n)	n/2	
Double FP Multiply	2229	223+10 DSPs
Single FP Multiply	511	461+3 DSPs
Fixed Multiply (32x32)	1099	
Fixed Multiply (16x16)	283	1 DSP
Fixed Multiply (18x25)		1 DSP
Fixed Multiply (n)	$\sim n^2$	

FP (Floating Point) sizes from:  
[https://www.xilinx.com/support/documentation/ip\\_documentation/ru/floating-point.html](https://www.xilinx.com/support/documentation/ip_documentation/ru/floating-point.html)

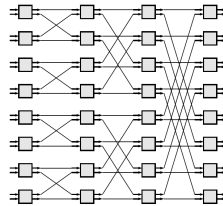
Penn ESE5320 Fall 2024 -- DeHon

43

43

## Heterogeneous Precision

- May not be same in every stage
  - W factors less than 1
  - Non-fraction grows at most 1b per stage



Penn ESE5320 Fall 2024 -- T

44

44

## T Coefficients

- Precompute and store in arrays
- Compute as needed
  - How?
    - sin/cos hardware?
    - CORDIC?
    - Polynomial approximation?
- Specialize into computation
  - Many evaluate to 0,  $\pm 1$ ,  $\pm \frac{1}{2}$ , ....
  - Multiplication by 0, 1 not need multiplier

$$Y[k] = \sum_{j=0}^{n-1} \left( X[j] e^{-2i\pi \frac{jk}{n}} \right)$$

Penn ESE5320 Fall 2024 -- DeHon

45

## FFT (partial) Design Space

- Parallelism
- Decompose
- Size/granularity of accelerator
  - Area-time
- Sequence/share
- Communicate
- Representation/precisions
- W coefficients (precompute, store, specialize...)

Penn ESE5320 Fall 2024 -- DeHon

46

46

## Big Ideas:

- Large design space for implementations
- Worth elaborating and formulating systematically
  - Make sure don't miss opportunities
- Think about continuum for design axes
- Model effects for guidance and understanding

Penn ESE5320 Fall 2024 -- DeHon

47

47

## Admin

- Feedback, including P1
- P2 due Friday
  - Asks you to identify design space

Penn ESE5320 Fall 2024 -- DeHon

48

48