ESE532: System-on-a-Chip Architecture

Day 18: October 30, 2019
Design Space Exploration

Today
- Design-Space Exploration
  - Generic
  - Concrete example:
    - Fast Fourier Transform (FFT)

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

Design-Space Exploration
Generic

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

• Have many choices for implementation
  - Alternatives to try
  - Parameters to tune
  - Mapping options
• This is our freedom to impact implementation costs
  - Area, delay, energy
Preclass 1

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

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)

Design-Space Choices

• Type of parallelism
• How decompose / organize parallelism
• Area-time points (level exploited)
• What resources we provision for what parts of computation
• Where to map tasks
• How schedule/order computations
• How synchronize tasks
• How represent data
• Where place data; how manage and move
• What precision use in computations

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?

Finding Optima

• Kapre, FPL 2009
• Kadric, TRETS 2016

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
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

Tools

- Sometimes tools will directly help you explore design space
  - What SDSoC/Vivado HLS support?
- Often they will not
  - What might you want that does not support?

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
- Often they will not
  - Need to reshape functions and loops
  - Line buffers
  - Data representations and sizes

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

Design-Space Exploration

Example FFT

Sound Waves

Hz = 1/s
1kHz = 1000 cycles/s

Source: http://www.mediacollege.com/audio/01/sound-waves.html
Discrete Sampling

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

Time-Domain & Frequency-domain

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

Frequency-domain

- Can represent sound wave as linear sum of frequencies

Time vs. Frequency

- Can represent sound wave as linear sum of frequencies

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

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

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

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

Fast-Fourier Transform (FFT)

- Efficient way to compute FT
- $O(N^\log_2(N))$ computation
- Contrast $N^2$ for direct computation
  - Each dot product has $N$ points (multiply-adds)

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

FFT

- Large space of FFTs
- Radix-2 FFT Butterfly

Basic FFT Butterfly

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

Preclass 2

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

FFT Parallelism

- Spatial
- Pipeline
- Streaming
- By column
  - Choose how many Butterflies to serialize on a PE
- By subgraph
- Pipeline subgraphs
Streaming FFT

Preclass 3

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

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

Accelerator Building Blocks

• What common subgraphs exist in the FFT?

Common Subgraphs

Processor Mapping

• How map butterfly operations to processors?
  – Implications for communications?
Preclass 4a

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

Preclass 4b

- How change evaluation order to reduce local storage memory?

Preclass 4b

- Evaluation order

Streaming FFT

Communication

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

Data Precision

- Input data from A2D likely 12b
- Output data, may only want 16b
- What should internal precision and representation be?
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(mpy) \sim W^2, A(add) \sim W

Heterogeneous Precision

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

W 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....

FFT (partial) Design Space

- Parallelism
- Decompose
- Size/granularity of accelerator
  - Area-time
- Sequence/share
- Communicate
- Representation/precisions
- Twiddle

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
Admin

- P1 milestone
  - Due Friday
- P2 out
  - Asks you to identify design space