CS294-48: Hardware Design Patterns
Class Wrap Up

Krste Asanovic
UC Berkeley
Fall 2009
Overall Problem Statement

MP3 bit string

Audio

Application(s)

(Berkeley) Hardware Pattern Language

MP3 bit string

Audio

Hardware (RTL)
BHPL Goals

- BHPL captures problem-solution pairs for creating hardware designs (machines) to execute applications

BHPL Non-Goals

- Doesn’t describe applications themselves, only machines that execute applications and strategies for mapping applications onto machines
Progress over This Semester
BHPL 0.5 Overview

Applications (including OPL patterns)

BHPL

- FFT to SIMD array

App-to-UTL Mappings Layer

- Problem: Application Computation
  - Solution: UTL Machine

UTL-to-UTL Transformation Layer

- Time-Multiplexing
- Unrolling

- Problem: UTL violates constraint (too big, too slow)
  - Solution: Transformed UTL

UTL-to-RTL Transformation Layer

- Microcoded Engine
  - In-Order Pipeline Engine

- Problem: UTL design
  - Solution: RTL behavior

RTL-to-Technology

- Interleaved Memory
- FIFO

- Problem: RTL behavior
  - Solution: Structural design
Machine Vocabulary

- Machines described using a hierarchical structural decomposition

- Units (processing engines)
- Memories
- Networks (connect multiple entities)
- Channels (point-to-point connections)

(Memories, Networks, and Channels are just specialized Units)
Pattern Write-Up Guidance

- Should begin a collection of related patterns with a preamble to define context and maybe some terminology.
- Each pattern writeup has following structure.
Pattern Name

**Problem:** Describe the particular problem the pattern is meant to solve. Should include some context (small, high throughput), and also the layer of the pattern hierarchy where it fits.

**Solution:** Describe the solution, which should be some hardware structure with a figure. Solution is usually the pattern name. Should not provide a family of widely varying solutions - these should be separate patterns, possibly grouped under a single more abstract parent pattern.

**Applicability:** Longer discussion of where this particular solution would normally be used, or where it would not be used.

**Consequences:** Issues that arise when using this pattern, but only for cases where it is appropriate to use (use **Applicability** to delineate cases where it is not appropriate to use). These might point at sub-problems for which there are sub-patterns. There might also be limitations on resulting functionality, or implications in design complexity, or CAD tool use etc.
Where class patterns fit?

Applications (including OPL patterns)

BHPL

App-to-UTL Mappings Layer

- Systolic Array

UTL-to-UTL Transformation Layer

- Time-Multiplexing
- Pipelining
- Unrolling
- Multithreading

UTL-to-RTL Transformation Layer

- Microcoded Engine
- Switch with Memory
- Clock Domain Crossings

RTL-to-Technology

- Interleaved Memory
- Regfile
- FIFO