Course Goal

- Define a pattern language to capture common solutions to hardware design problems
- Inspirations: Christopher Alexander, Software Patterns, UPCRC OPL
Caveats

- This is not educational TV
- This is an exploratory graduate seminar
- i.e., you will be helping find/build this material
- Class participation critical to success
Course Overview

Application(s)

(Berkeley) Hardware Pattern Language

Hardware (RTL)
BHPL Version 0.1

Application Patterns (from OPL)
- Structural Patterns:
  - Pipes
  - Model-View-Controller
  - Event Based
  - Process Control
  - Iteration
  - Map-Reduce
  - Layered Systems
  - Task Graphs

- Computational Patterns:
  - Circuits
  - Dense Linear Algebra
  - N-Body Methods
  - Sparse Linear Algebra
  - Spectral Methods
  - Unstructured Grids
  - Graph Traversal
  - Structured Grids
  - Graph Algorithms
  - FSMs

Machine Organizations
- Systolic
- SIMD Shared Memory
- SIMD Distributed Memory
- MIMD Shared Memory
- Homogeneous MIMD Distributed Memory
- Heterogeneous MIMD Distributed Memory

Processing
- FSM
- Microcoded Pipeline
- In-Order Pipeline
- Out-of-Order Pipeline

Memory
- Banked Memory
- Bypassed Memory
- Cached Memory

Switch
- Bus
- Crossbar
- Multi-Stage Networks

Hardware Building Blocks
- FIFO
- Multiport Memory
- CAM
- Arbiter
- Communication Channel
Machine Organizations

- Describes overall structure of machine
- Can be composed hierarchically
- Some Important dimensions:
  - SIMD versus MIMD
  - Distributed versus shared memory
- Systolic ?
SIMD Distributed Memory

Examples: MPP, ICL DAP, CM-1, CM-2, MasPar, Sony Playstation-2 Graphics Engine, Vision processing chips
SIMD Shared Memory

Examples: STARAN, BSP, TI ASC, CDC
Star-100, Multi-Lane Vector Machines
MIMD Shared Memory

Examples: Burroughs B5x00 series, Multicore
Homogeneous MIMD Distributed Memory

- Examples: Caltech Cosmic Cube, Transputer, nCube, Clusters
Heterogeneous MIMD Distributed Memory

Examples: Signal Processing Pipelines,
Systolic

- Examples: Warp, Raw, Motion Estimation Engines,
Processor Types

- State Machine
  - limited/no datapath
- Microcoded Engine
  - single-cycle datapath
- In-Order Pipeline Engine
  - Single stream of pipelined operations
- Out-of-Order Pipeline Engine
  - Stream of operations can be reordered
- Threaded Pipeline
  - Several streams of operations are time-multiplexed
Memory Structures

- True Multiport Memory
- Banked Memory
  - Interleave lesser-ported banks to provide higher bandwidth
- Cached Memory
  - Memory hierarchy to provide higher-bandwidth, lower latency for predictable accesses
- Bypassed Memory
  - Reduce latency of pipelined dependent memory accesses
Switches

- Connects multiple agents (processors and/or memories)

- Bus
  - Low-cost, ordered

- Crossbar
  - High-performance

- Multi-stage network
  - Trade cost/performance
Course Organization

- Instructors: Krste Asanovic, John Wawrzynek
- Meet once a week, Thursdays at 2pm
- First few weeks is general discussion, reading, and example gathering
- Middle of semester is group refinement of space and students selecting patterns
- Towards end of semester, we have pattern workshops to refine patterns students are authoring
Course Credit

- For 1 unit of credit, attend class, participate in discussion, and gather and present examples
- For 3 units of credit, each student will in addition author a pattern, and shepherd another pattern
First Assignments

- Read DeHon et. al., “Design Patterns for Reconfigurable Computing”, FCCM 2004, for next week (9/3)
- Find one hardware engine (not a conventional programmable processor), and bring meeting in two weeks time (9/10) with preliminary pattern breakdown