Designing a Coarse-grained Reconfigurable Architecture for Performance and Power Efficiency

Allan Carroll, Stephen Friedman, Robin Panda, Brian Van Essen, Aaron Wood, Benjamin Ylvisaker

Carl Ebeling and Scott Hauck
Accelerating DSP & Scientific Computing

- Digital signal processing & scientific computing
  - Video
  - Networking
  - Encryption
  - DNA sequencing
  - Protein Folding
  - ...

- Large, repetitive, computation
- Want flexibility
Achieving Parallelism: Multi-core

+ general purpose
- doesn’t exploit regularity
Achieving Parallelism: ASIC

+ high performance

- high cost
Achieving Parallelism: FPGA

+ flexibility and fast

- inefficient
Achieving Parallelism: CGRA
What is a CGRA

• Lots of ALUs
• Embedded memory
• Scheduled interconnect
• Shared resources
  – time-multiplexed
• Streaming I/O
Communication via wires

• Communicating via wires not load and store through memory
• Many concurrent hardware resources
Exploring the CGRA design space

- Previous coarse-grained architectures
  - Monolithic systems
  - Lacked analysis of individual architectural features

- Identify key architectural features
  - Evaluate for performance and power efficiency

- Complete system
  - Language
  - Architecture
  - Programming environment
Architectural Opportunities

• Interconnect (communication) is bulk of power budget
• Programmability is just overhead

Distribution of Power in FPGA

- 60% Routing
- 35% Logic
- 5% Clock
Example: Interconnect topology

nearest neighbor

nearest neighbor + switched network
Architectural Exploration

• Dedicated vs. shared interconnect channels
  – shared channels can be statically scheduled or dynamically controlled
• Heterogeneous vs. homogeneous logic mix
• Memory hierarchy
• Specialization vs. generalization
• Time-multiplexing / virtualization of resources
• Computational & control style
• Clocking strategies
Designing an architecture

- Logic Synthesis
  - Tool: Macah Compiler
  - Unified Mapper (Schedule, Place & Route)
    - Tool: SPR

- Power Analysis
  - Tool: Verilog + PLI

- Architecture Generation
  - Tool: Electric VLSI

- Dataflow graph
- Datapath graph configuration
- Datapath graph
- Benchmarks
- Simulation
- Area & Delay
- Power
Representing prototype architectures

• Datapath graph
  – computing resources
  – storage
  – interconnect
Measuring power

Benchmarks

Logic Synthesis
Tool: Macah Compiler

dataflow graph

Unified Mapper
(Schedule, Place & Route)
Tool: SPR

datapath graph configuration

Power Analysis
Tool: Verilog + PLI

datapath graph

dataflow graph

Architecture Generation
Tool: Electric VLSI

Area & Delay

Power
Application driven power modeling

- Modeling power consumption requires accounting for correlation in application data
  - application driven power simulation
Creating benchmarks

- **Logic Synthesis**: Tool: Macah Compiler
- **Unified Mapper**: (Schedule, Place & Route) Tool: SPR
- **Power Analysis**: Tool: Verilog + PLI
- **Architecture Generation**: Tool: Electric VLSI

Diagram:

- Dataflow graph configuration
- Datapath graph
- Datapath graph configuration
- Datapath graph

Output:

- Simulation
- Area & Delay
- Power
Macah

- Good abstractions for explicitly managing spatial resources
  - compute resources
  - kernel bandwidth
  - memory
Programming an architecture

- Logic Synthesis
  - Tool: Macah
  - Compiler
- Unified Mapper (Schedule, Place & Route)
  - Tool: SPR
- Power Analysis
  - Tool: Verilog + PLI

Architecture Generation
- Tool: Electric VLSI

Benchmarks

Simulation

Area & Delay

Power
Mapping applications in space

• Placement and routing of an application to an architecture is a traditional CAD problem

• Manage architectures reconfiguration by adapting CAD algorithms to have a time-sharing capability
Managing kernels in space and time

- Some common operations have a long latency
  - Can force your application to run at a slower rate
  - Lead to under-utilization of dedicated resources

- Resource requirements may exceed availability

- Resource sharing (i.e. virtualization) allows for more efficient implementation.
  - Time-multiplex multiple operations onto single physical resource
Mapping applications in space and time

• Unified Schedule, Place, & Route
  – simplifies task of mapping
  – improves results
Summary

• Power efficiency becoming first-order concern for digital signal processing and scientific computing

• CGRAs well suited for application domain

• Flexibility in implementing applications is crucial for evolving applications
  – Overhead of flexibility is too high in current architectures

• Exploring the design space to identify which architectural features offer an advantage in power efficiency
Questions?

• Visit us at the poster session in:

Hardware and Embedded Systems Lab
CSE 505

Allan Carroll, Stephen Friedman, Robin Panda, Brian Van Essen, Aaron Wood, Benjamin Ylvisaker

Faculty:
Carl Ebeling and Scott Hauck