Efficient Compilation of Complex Data-Dependent Code to Coarse-Grained Reconfigurable Architectures

Stephen Friedman, Benjamin Ylvisaker, Robin Panda, Carl Ebeling and Scott Hauck

CGRAs
This work is part of the MOSAIC Project -- exploring a tool-chain and Coarse-Grained Reconfigurable Architecture designs to provide energy efficient acceleration of HPC kernels.

Architecture Features:
- Lots of ALUs
- Internal distributed memory
- Modulo Scheduled time multiplexing
- Fully compile-time configured

Goal: Sharing Resources on Mutually Exclusive Paths
Enhanced loop flattening is used to transform control flow into a spatially distributed computation -- mutual exclusivity of code paths provides potential for runtime sharing of resources. This will lessen the burden of "cold paths", improving energy efficiency and performance in complex kernels.

Compiler Abstractions for Data-dependent Execution in a Modulo Scheduled Architecture

Datapath Control of Multiplexing:
Modulo Counter + (Few) Datapath Signals

Tracking Distribution of Control:
Regions
Colors represent areas controlled by the same predicate bits.

Routing
Mutually exclusive routes can join where the predicates are available to switch the route.
If routes match direction once joined, they can share paths even in regions without the predicate available.

A: Generalize from work on sharing signals on non-multiplexed interconnect!

Q: How do we get routes to match?

Scheduling
Adopt techniques from VLIW community, provides for optimistic scheduling of CGRA:

Placement
Based on Simulated Annealing
- Allow multi-placement
- Track compatibility of operations on devices
- Track availability of predicates within regions