ReconfAST: Early-Stage Identification Tool to Detect Similar Hardware Implementations
An HLS add-on to find Shared Accelerators

Parnian Mokri & Maziar Amiraski & Yuelin Liu & Mark Hempstead
Tufts University

Abstract
We propose an early detection tool that complements existing High-level Synthesis tools by identifying computationally similar synthesizable kernels that are used to build Shared Accelerators (SAs). SAs are specialized hardware accelerators that execute very different software kernels but share the common hardware function between them. SAs can provide increased coverage if both dataflow and control flow similarities between seemingly very different workloads are detected. Existing methods are either dynamic traces or analyze register transfer level (RTL) implementations to find these similarities which requires deep knowledge of RTL and time-consuming design process.

Introduction
Current design methodologies fail to efficiently assist computer architecture designers to maximize overall system performance and workload coverage, especially under a specific system-wide area constraint.

List of contributions:
1. Introduction of shared accelerators (SAs) and their architecture implications.
2. ReconfAST, a methodology for extracting SA candidates by detaching common kernels using AST representations of source-code.
3. Introducing Clustered-AST (CAST): A transformed AST representation that removes unnecessary syntax, summarizes common patterns, and aids in the efficient comparison of hardware similar workloads.
4. Study of ReconfAST on MachSuite demonstrating the potential of SAs to increase coverage.
5. ASC and FPGA implementations of example SAs and analysis of hardware costs. Classification of workload patterns that result in good and poor performing SA implementations.

Problem: No Methodical Approach to Design Dedicated Accelerators

Solution: Increase The Coverage of Each Accelerator: Shared Accelerator

Shared Accelerators (SAs) can execute multiple kernels by including all of the hardware for both kernels. Common hardware kernels are automatically discovered and shared, reducing area costs.

Choosing Effective Maps

Our methodology finds statically similar software kernels in pairs of workloads by evaluating if two subtrees are isomorphic. To judge the efficacy of Shared Accelerator candidates we need to estimate the shared subtree’s fraction of total execution-time.

Hardware Implementation

SAs accelerate workloads by 5x on average, and reduce Flipflop usage by 37%, DSPs by 16%, LUTs by 10% over an average of dedicated accelerators.

References