Dataflow Blocks: Modular Time-multiplexing for CGRAs

Xuesi Chen, Nishanth Subramanian, Karthik Ramanathan, Nathan Beckmann, Brandon Lucia

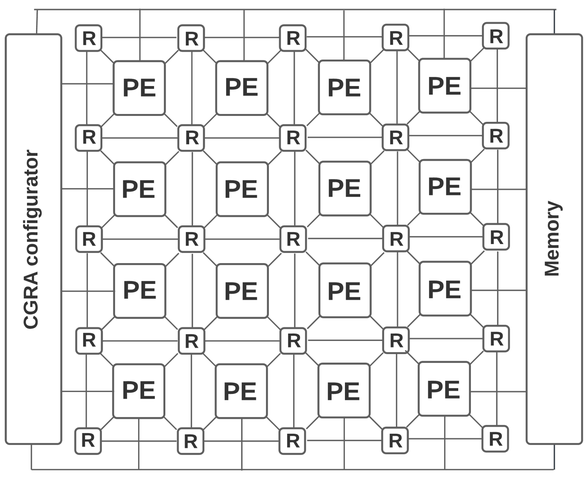
Carnegie Mellon University

Introduction

We provide a compiler and architecture co-design to improve performance per area with a small cost in energy on CGRA-based energy-minimal processors.

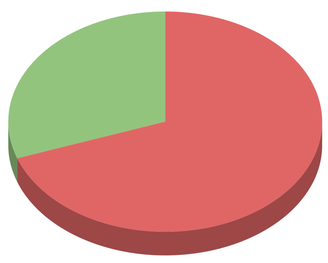
chip-scale satellite | wildlife tracking | medical devices

Background



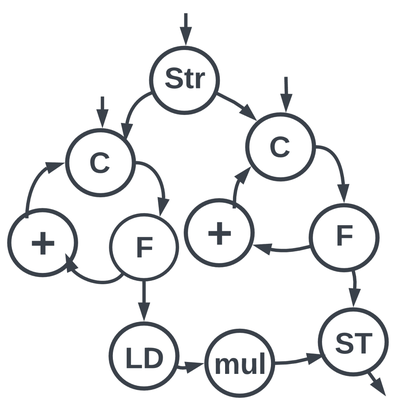
A Potential CGRA Design

hardware resource utilization on RipTide[1]

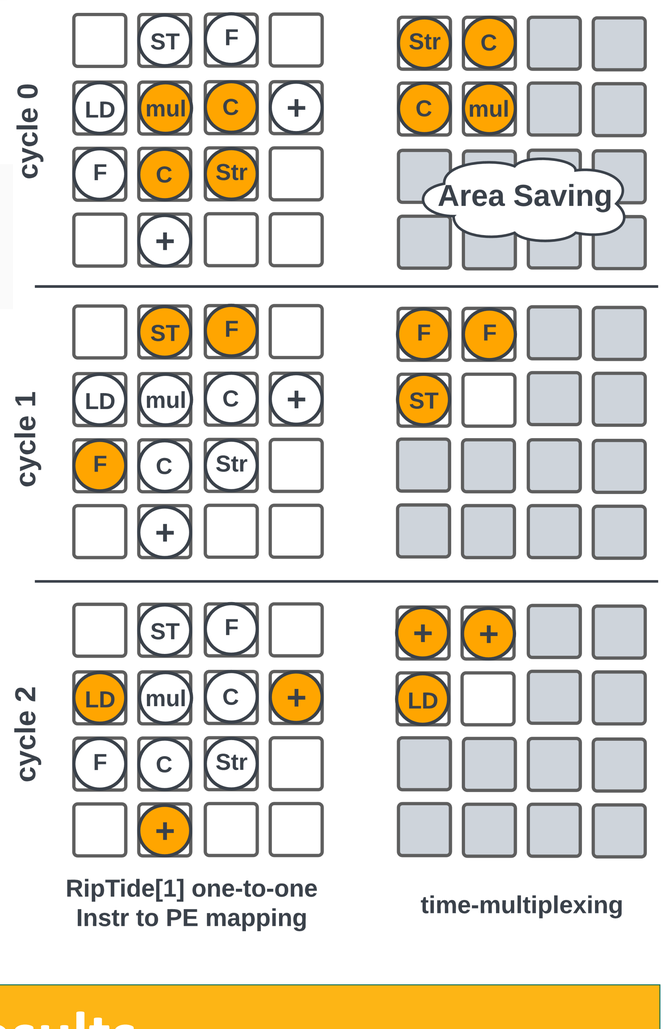



```
for (int i = 0; i < size; i++) {
  *dest++ = *src++ * 5;
}
```

C Program



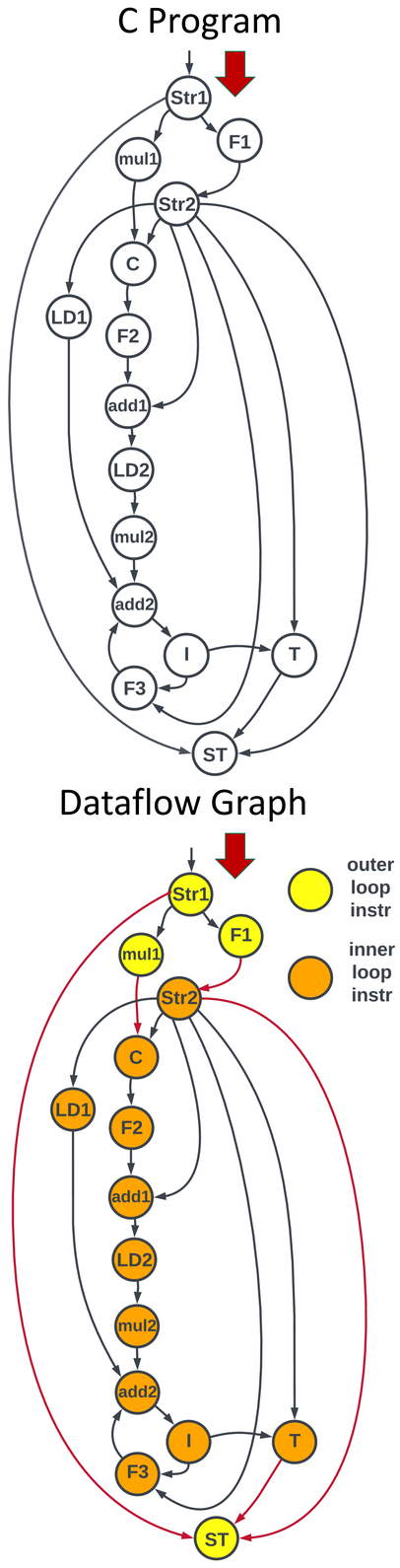
Dataflow Graph



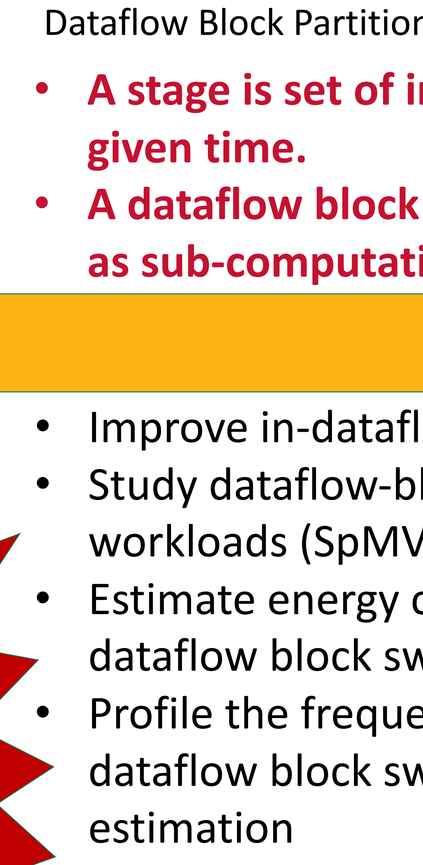
Systems Overview

```
for (int i = 0; i < size; i++) {
  int w = 0;
  for (int j = 0; j < n; j++)
    w += A[i*n+j]*b[j];
  z[i] = w;
}
```

C Program

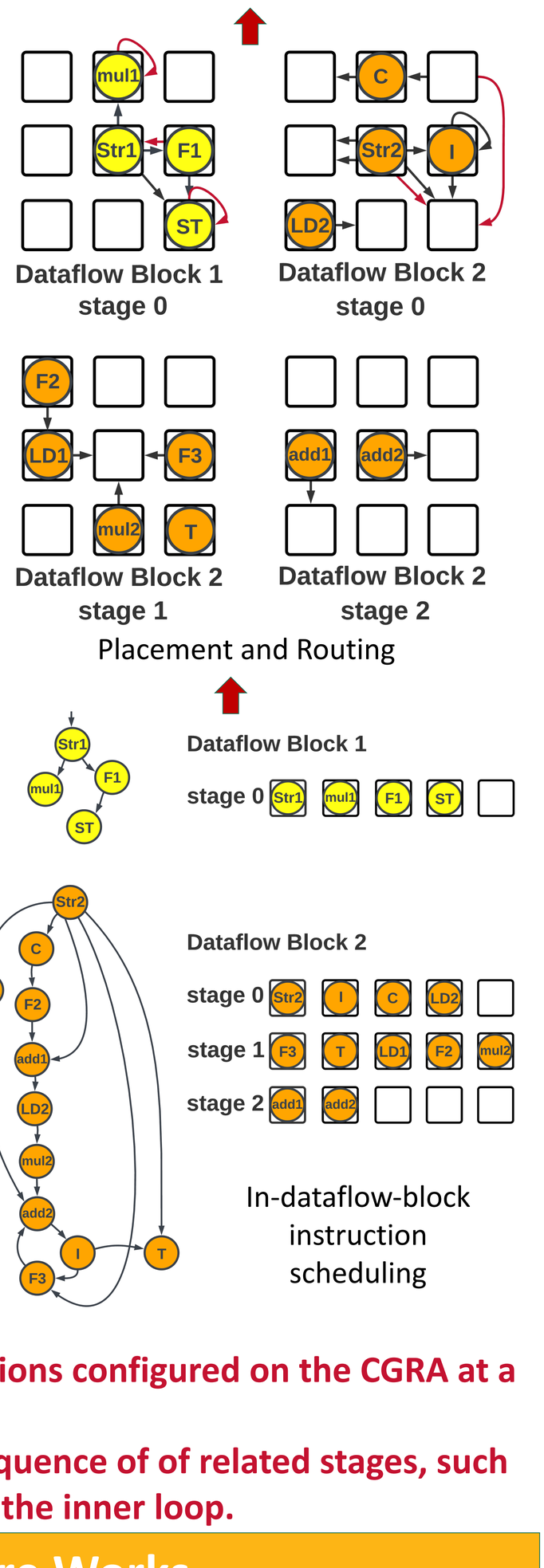


Dataflow Graph



Dataflow Block Partition

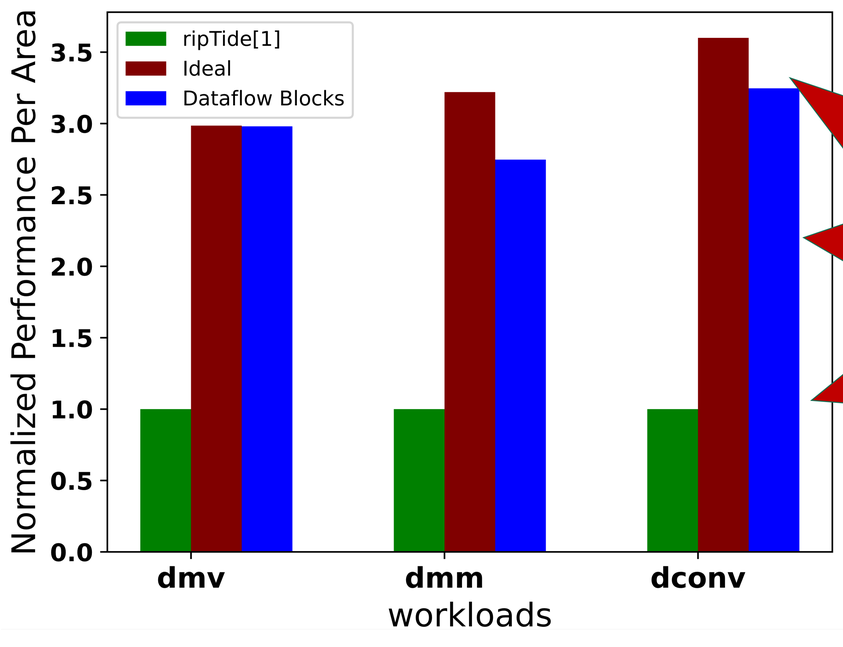
Execution:
static cycle-by-cycle stage rotation
dynamic dataflow block scheduling



In-dataflow-block instruction scheduling

- **A stage is set of instructions configured on the CGRA at a given time.**
- **A dataflow block is a sequence of of related stages, such as sub-computations of the inner loop.**

Results



2.7x-3.3x performance per area improvement

Future Works

- Improve in-dataflow-block static scheduling
- Study dataflow-block partition for other irregular workloads (SpMV, fft, bfs, etc.) beyond nested for loops
- Estimate energy cost for stage reconfiguration and dataflow block switching in RTL
- Profile the frequency of stage reconfiguration and dataflow block switching in the simulator for energy estimation
- Expand compiler algorithm to accommodate spatial loop unroll and multi-tenancy

[1] G. Gobieski, S. Ghosh, M. Heule, T. Mowry, T. Nowatzki, N. Beckmann, and B. Lucia, "A programmable, energy-minimal dataflow compiler and architecture," pp. 546–564, 2022.