Loop Acceleration For Tightly-Coupled CPU+FPGA System

Cheng Liu
Supervisor: Dr. Hayden Kwok-Hay So
Co-supervisor: Dr. Ngai Wong

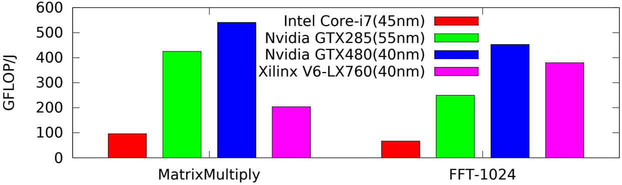
Department of Electrical and Electronic Engineering
The University of Hong Kong

June 10, 2013

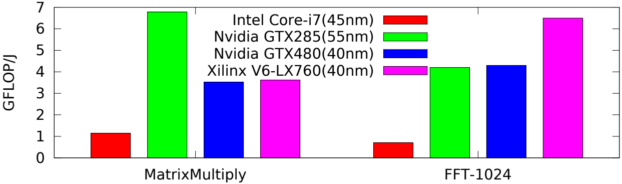
FPGA vs. CPU vs. GPU

FPGA has competitive computation capability and energy efficiency.

Performance comparison [1]



Energy efficiency comparison [1]



[1] Eric S. Chung, etc., Single-Chip Heterogeneous Computing: Does the future include customized logic, FPGA and GPGPUs?, IEEE International Symposium of Microarchitecture, 2010

Why isn’t FPGA the mainstream computing device?

Main obstacles

- High barrier-to-entry
  - Require extensive hardware knowledge,
  - while software engineers usually don’t have.
  - ...
- Low design productivity
  - Low level abstraction and long development time
  - Long compilation and implementation time
  - Poor portability and design reuse
  - Difficult to support complex software like OS
  - ...

What has the community done to overcome the obstacles?

Design methodologies

- High level synthesis languages and tools

  Vivado(Xpilot, AutoESL), LegUP, Impulse-C, ...
- Virtual overlay on top of commerical FPGA

  VirtualRC, Soft coarse-grained reconfigurale array (SCGRA), ...

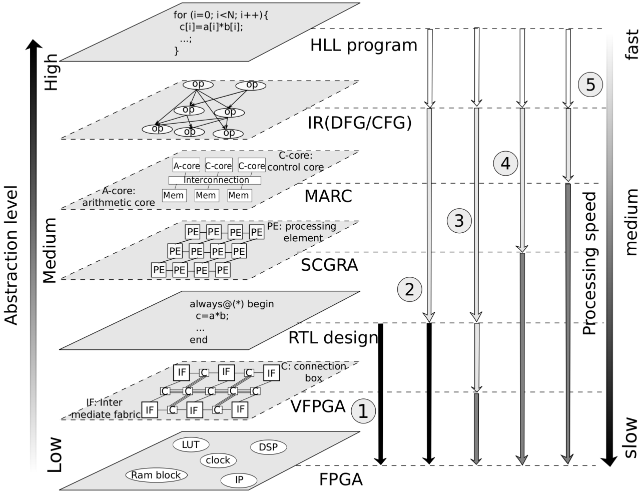
CPU+FPGA based hybrid computation

- Hybrid computation architectures

  Embedded softcore+FPGA, Embedded hardcore+FPGA, General CPU+FPGA, ...
- Communication libraries, unified memory interfaces, and integrated environments

  CoRAM, LEAP, ...

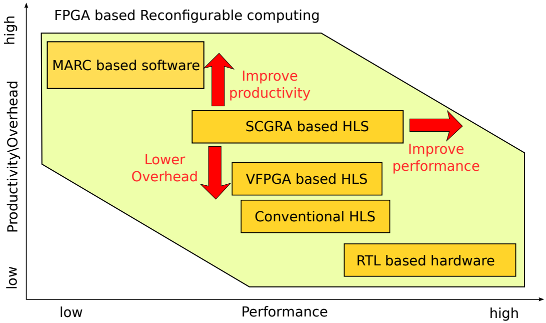
Differences and relations of the design methodologies



1 RTL design flow 2 Conventional HLS 3 Virtual FPGA(VFPGA) based HLS

4 SCGRA based HLS 5 Many-core approach to reconfigurable computing(MARC)

Performance vs. productivity vs. overhead

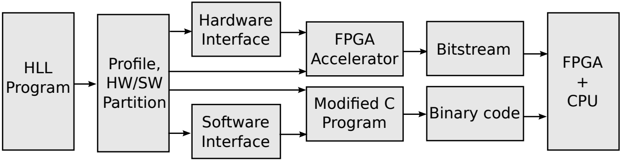


Why SCGRA based HLS has potential to provide better performance?

performance = operations per cycle × **implementation frequency**

CPU+FPGA based hybrid computation

Hardware/software co-design



Loop and computation kernel

- Most algorithms are implemented following a sequential programming model.
- Loops are typical computation kernels with large parallel operations.

CPU+FPGA based hyrbid computation architecture

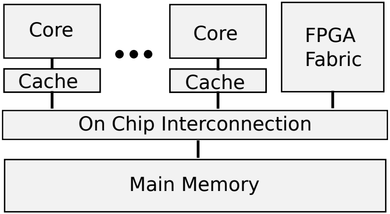


Figure 1 : Single chip multicore with FPGA accelerator

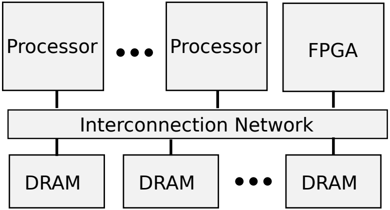


Figure 2 : Shared memory multiprocessor with FPGA accelerator

Previous SCGRA Work

What have been done?

- Introduced the SCGRA layer for HLS,
- showed potential design productivity improvement,
- and proved its energy efficiency using an application specific SCGRA topology

What are still missing?

- The relationship between a holistic loop and its kernel data flow graph,
- influence of the communication between CPU and FPGA on the SCGRA based HLS.

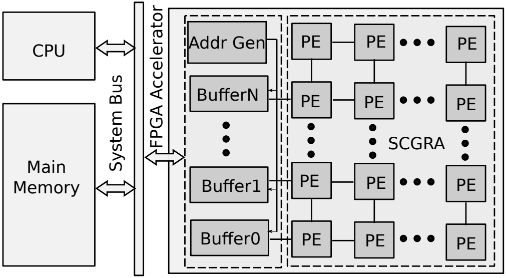
Main goal of this work

Accelerate loop on a CPU+FPGA system

- Optimal loop unrolling for the SCGRA based accelerator
- Application specific on-chip buffering including data prefetching and buffer structure customization

Hardware infrastructure

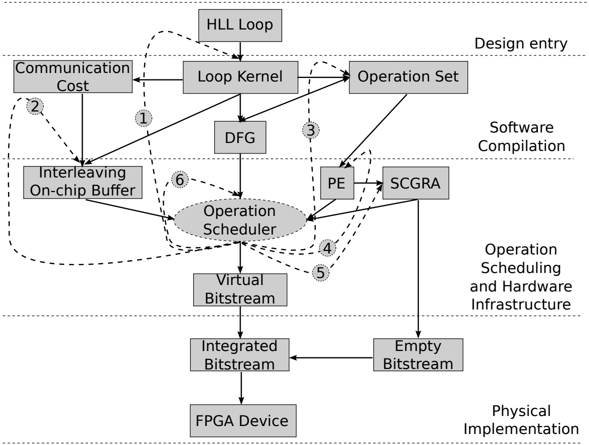
SCGRA based CPU+FPGA accelerator



Softness of the accelerator

- SCGRA structure could be reconfigurable
- On chip buffer could be reconfigurable

SCGRA based accelerator design flow



1 loop unrolling factors

2 On-chip buffer size, interleaving scheme, data fetching scheme

3 Primitive operations supported by the hardware infrastructure

4 PE pipeline depth, local memory port number and allocation

5 Topology of the computation array, array size

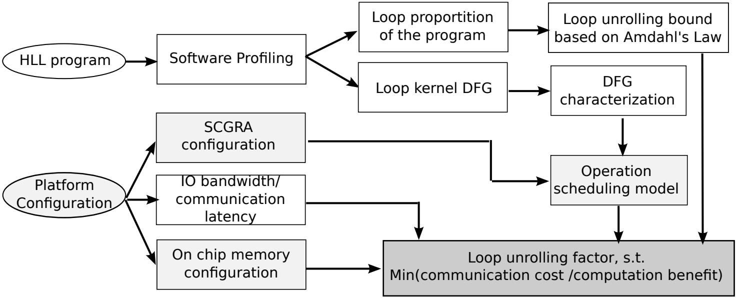
6 Scheduling algorithm, scheduling strategies

Optimal loop unrolling

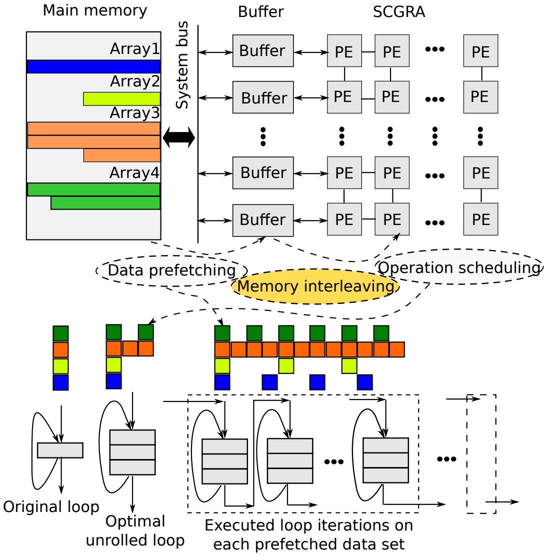
Why loop unrolling and why not fully unroll the loop?

- Increases parallel operations and improves performance
- Induces larger hardware overhead performance
- Benefit may be limited by system constrains.

Simplified loop unrolling

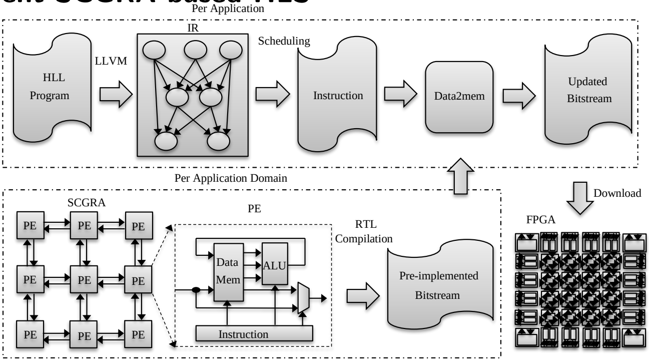

HLL program
Software Profiling
Loop proportition of the program
Loop unrolling bound based on Amdahl's Law
Loop kernel DFG
DFG characterization
SCGRA configuration
Operation scheduling model
Platform Configuration
IO bandwidth/ communication latency
On chip memory configuration
Loop unrolling factor, s.t. Min(communication cost /computation benefit)

On-chip buffering



Quantify the productivity of the SCGRA based HLS

Current SCGRA based HLS

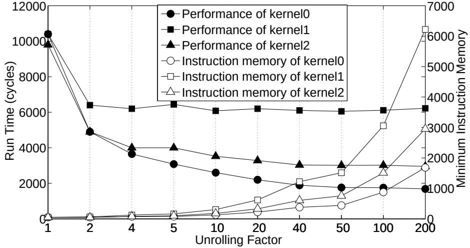


Colin's work on the SCGRA based HLS

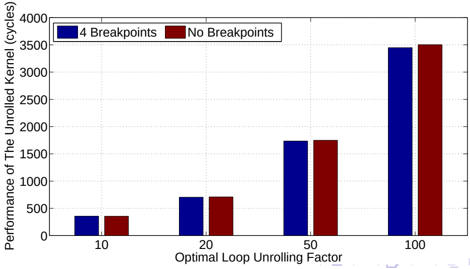
- Operation scheduling
- SCGRA design and implementation
- Application specific SCGRA topology synthesis

Preliminary loop unrolling analysis

loop unrolling influence on performance and overhead

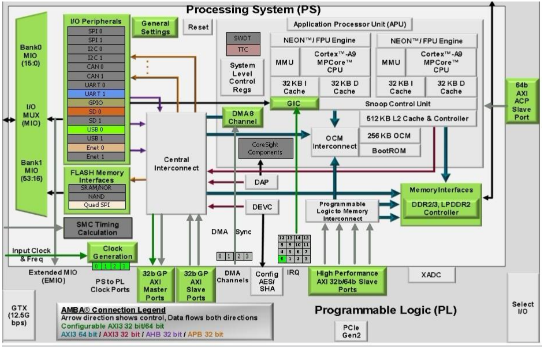


Irregular loop bound



HW/SW communication on Zedboard

Zedboard platform



Different communication methods

- Accelerator coherence port
- Central DMA, Video DMA, XDMA
- GPIO

Self-evaluation

About the progress and publciation

- Didn't work hard enough
- Didn't not balance well between the engineering work and research focus
- Have taken 10 RPG courses up to now