

# Loop Acceleration For Tightly-Coupled CPU+FPGA System

Cheng Liu

Supervisor: Dr. Hayden Kwok-Hay So

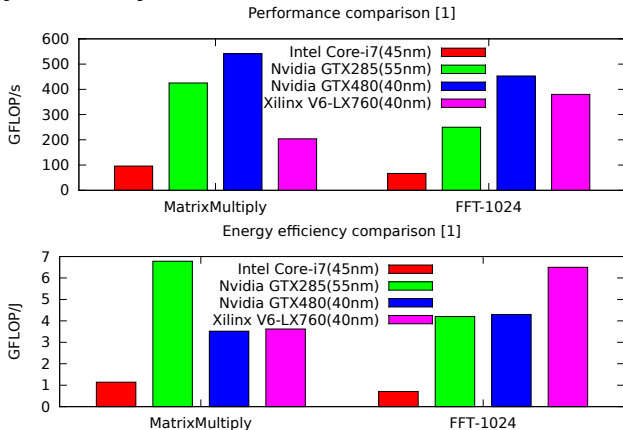
Co-supervisor: Dr. Ngai Wong

Department of Electrical and Electronic Engineering  
The University of Hong Kong

June 14, 2013

# FPGA vs. CPU vs. GPU

**FPGA has competitive computation capability and energy efficiency.**



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

Background

Related work

Research  
scheme

Current  
progress

Conclusion

# Challenges and progress on FPGA computing

## Challenges

- High barrier-to-entry (Hardware knowledge, ...)
- Low design productivity (Low abstraction level, long compilation time, ...)

## Progress

- High level synthesis (HLS), e.g., LegUp, AotoESL, Impulse-C, ROCCC, ...
- Virtual overlays
  - ✓ Reconfigurable many-core, e.g., MARC, WPPA(Weakly programmable processor array), ...
  - ✓ Coarse-grained reconfigurable array, e.g., QUKU, SCGRA, Heterogeneous CGRA, ...
  - ✓ Virtual FPGA, e.g., Intermediate Fabric, ZUMA, CARBON, MALIBU, ...
- Other techniques, e.g., Partial reconfigurable technique, Hard Macros, Communication library, ...

Background

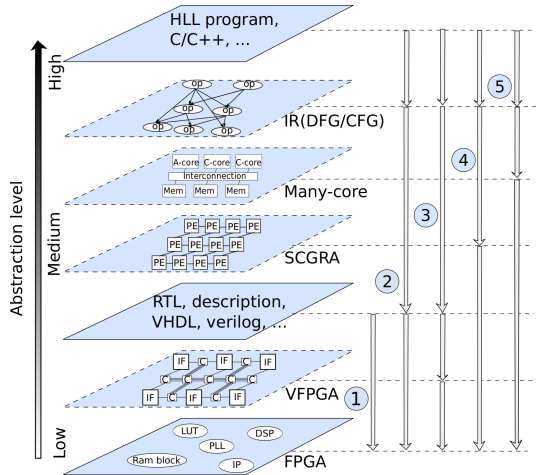
Related work

Research  
scheme

Current  
progress

Conclusion

# Differences and relations of the overlays



- ① RTL design flow
- ② Conventional HLS
- ③ VFPGA based HLS
- ④ SCGRA based HLS
- ⑤ Reconfigurable many-core architecture

# SCGRA work in our group

Background

Related work

Research  
scheme

Current  
progress

Conclusion

## 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.

## Focus of my work

- Automatic loop acceleration on a tightly-coupled CPU+FPGA using the SCGRA overlay

# Why loop acceleration?

## Loop and computation kernel

- Loops usually form the most computationally intensive kernel of a program
- Regularity of loops provide ample of data parallelism
- Loops are important optimization targets for the parallel computing architectures including Multi-core processor, GPU, CGRA and FPGA.

## Difference from previous work

- Hardware infrastructure (SCGRA and communication) is changing with the loop optimization
  - ✓ Not possible with hard CGRA
  - ✓ Take advantage of the softness of the FPGA
  - ✓ Application-specific buffering, loop unrolling, and scheduling

Background

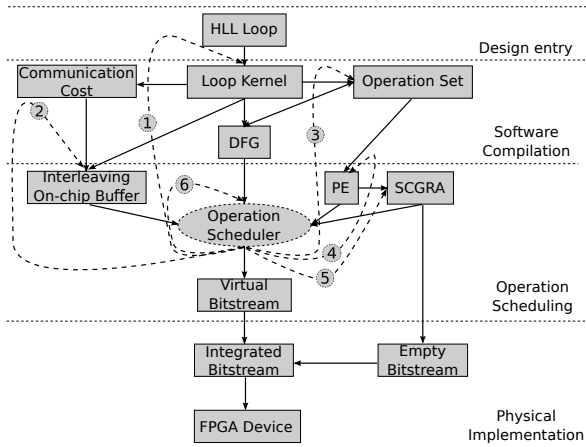
Related work

Research  
scheme

Current  
progress

Conclusion

# 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 constraints.

## Loop unrolling problem

- Assumptions: Bounded loop, and data dependency known at compiling
- Input: Sequential program proportion, kernel DFG, loop iteration bound, ...
- Optimization target: Min(loop execution time/communication cost)
- Constraint: hardware overhead, IO bandwidth
- Model: Operation Scheduling model+Data prefetching model

Background

Related work

Research  
scheme

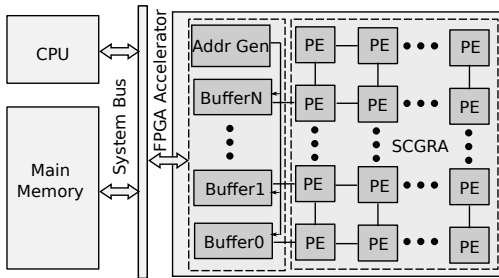
Current  
progress

Conclusion



# Hardware infrastructure

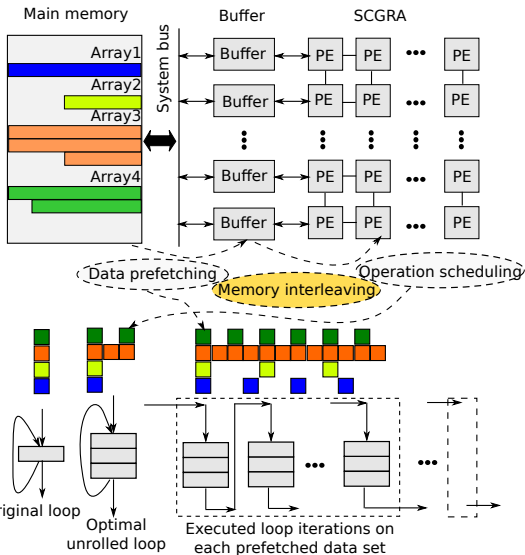
## SCGRA based CPU+FPGA accelerator



## Softness of the accelerator

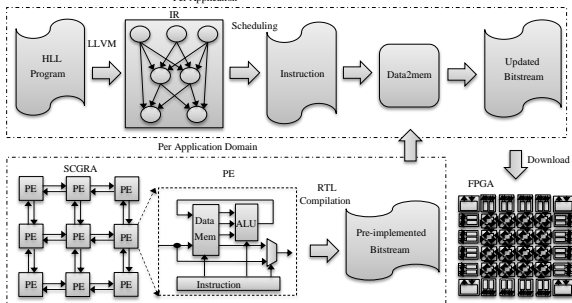
- SCGRA structure could be reconfigurable
- On chip buffer could be reconfigurable

# On-chip buffering

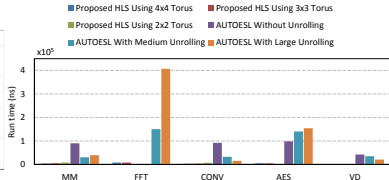
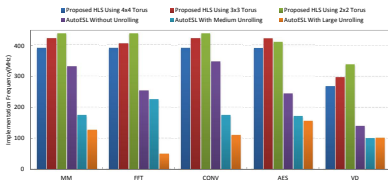


# SCGRA based HLS optimization for both design productivity and frequency

## Optimized SCGRA based HLS

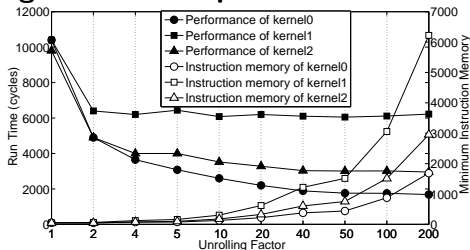


## Experiment results

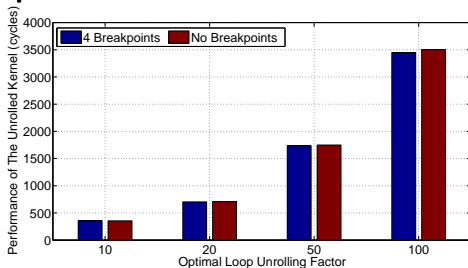


# Preliminary loop unrolling analysis

## loop unrolling influence on performance and overhead



## Irregular loop bound



Background

Related work

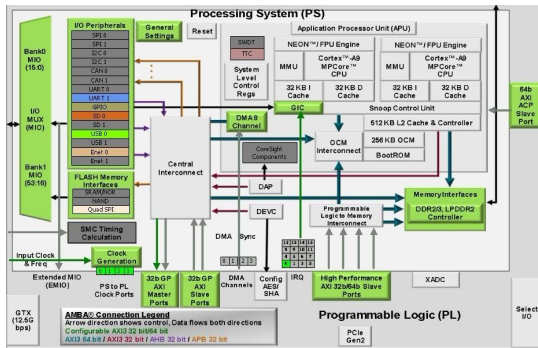
Research  
scheme

Current  
progress

Conclusion

# HW/SW communication on Zedboard

## Zedboard platform



## Different communication methods

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

# Conclusion

## Potential contribution

- Analyze the relationship between loop and its kernel data flow graph. Hopefully, an optimal partial loop unrolling may help resolve the BRAM-consuming problem in previous work.
- Provide a systemic solution to loop acceleration on a CPU+FPGA system and therefore a more friendly high level interface to the end users.

Background

Related work

Research  
scheme

Current  
progress

Conclusion