A PGAS infrastructure for Heterogeneous Reconfigurable Computing

Ruediger Willenberg and Paul Chow

High-Performance Reconfigurable Computing Group
University of Toronto

October 4, 2013
Overview

• A brief FPGA primer
• Programming models and our group’s approach
• FPGA components for Global Address Space
• Implementation systems
• Upcoming work
FIELD-PROGRAMMABLE GATE ARRAYS
FIELD-PROGRAMMABLE GATE ARRAYS

© H.Roesner

© flickr / MadPhysicist
Hardwired FPGA functions

- ARM A9
- ARM A9
- SRAM
- Ser Des
- DSP
- 1GE MAC
Computing with FPGAs

• Fully customized dataflow and buffering

• Tightly coupled pipelining of computations

• Very low energy / computation ratio

• Computation cores can be switched out in msecs with partial reconfiguration
Example: Smith-Waterman DNA sequencing (Dynamic Programming)

49x – 980x speedup (I/O dependent) on Xilinx V4-LX160 FPGA vs. 2.2GHz AMD Opteron

(Storaasli/Cray 2009)
FPGA programmability drawbacks

• Implementation (compile-equivalent) takes **hours**
• Established Hardware Description Languages (Verilog HDL, VHDL) are very low-level
• Downside of design flexibility: No established programming models
• Interfaces are often specific to vendors, platforms, FPGA boards
Programmability improvements

• Higher-level HDLs
  – SystemC, SystemVerilog, Bluespec, *Chisel (UC Berkeley)*

• High-Level-Synthesis ("C-to-gates")
  – Very active area in research and industry, but:
  – Only useful if programmer understands hardware
Programmability improvements (2)

OpenCL as an interface to FPGAs

– Abstracts hardware interface as it does for GPUs
– Sets a programming model (memory, core hierarchy)
– Implies High-Level-Synthesis for actual code kernels
– But GPU model not a perfect fit for FPGA
  • Overconstrained dataflow
Classic accelerator model: Master-Slave
Our programming model philosophy

- Use a common API for Software and Hardware
Common SW/HW API

• CPU and FPGA components can initiate data transfers

• SW and HW components use similar call formats

• For distributed memory and message-passing, this was implemented by TMD-MPI (TMD: Toronto Molecular Dynamics)

• Our contribution: Building hardware infrastructure for a common API for PGAS
Why again a common API?

- Easier development: SW Prototyping → Migration

- Model makes no distinction between CPUs and FPGAs (in terms of data communication, synchronization)

- FPGA-initiated communication relieves CPU (even more so for 1-sided comm.)

- FPGA-only systems (or 1 CPU + many FPGAs) can work efficiently
From MPI to GASNet

• Chose GASNet Core API as the “common API” to use for PGAS software and hardware

• GASNet is widely used, well-defined PGAS API

• Core API’s “Active Message” types (S/M/L/LA) and handler codes cover essential hardware needs
GASNet on FPGA: GAScore
GAScore: RDMA for FPGA components

GAScore

CPU

SRAM

Memory Interface

Transmit

Token Buffer

Receive

To Network

From Network

AM request

Handler call

GAScore
GAScore

• Remote memory communication engine (RDMA)
• Controlled through FIFOs (FSLs)
• Configuration parameters are the same as for GASNet Active Message function calls
  – Simple software layer for embedded CPUs
GASNet on FPGA: GAScore
Programmable Active Message Sequencer

• Controls custom hardware operations
• Handles reception/transmission of Active Messages
• Custom hardware ops and Active Messages are initiated based on:
  – Custom hardware state
  – Number of received messages with a specific code
  – Amount of received data
• (Re-)programmable through Active Messages
On-chip GASNet system

- FPGA
  - SRAM
  - CPU
  - CH
  - Gc
- DRAM
- Memory Controller
- On-Chip Network
- Off-Chip
- Host via PCIe, QPI, HT, Ethernet
- Other FPGAs via PCIe, Ethernet, Infiniband
BEE3 multi-FPGA platform
BEE4 heterogeneous system

PCIe 2 x8
3.2 GB/s

2x 16GB
DDR3-800

1 Gb Eth

October 4, 2013
High-Performance Reconfigurable Computing Group ∙ University of Toronto
Undergraduate Project: GASNet on Zynq
Current and upcoming work

• Hardware components:
  – External memory support
  – Strided/vectored memory accesses
  – Host interface (PCIe; later QPI, HT)

• Software components:
  – GASNet integration on CPU side
  – PGAS C++ library for heterogeneous systems
Heterogeneous C++ PGAS library

C++ PGAS Application

Heterogeneous C++ PGAS Library

GASNet Library

Compile

CPU-based Host

GASNet

Dynamic generation

PAMS

Custom FPGA Hardware

Static generation

Manual or HLS

October 4, 2013

High-Performance Reconfigurable Computing Group    ∙     University of Toronto
Conclusion

GASNet API + GAScore (+PAMS) enable

- Universal abstraction for FPGA interfaces
- Simple heterogeneous communication and synchronization
- Simplified kernel migration to accelerators
Thank you for your attention!

More details in my ACM FPGA 2013 paper:

“A Remote Memory Access infrastructure for Global Address Space programming models in FPGAs”

http://www.eecg.toronto.edu/~willenbe/publications

Questions?