Communicating High-level Programs with FPGA-based Acceleration Hardware

A. Cruz, M. Jiménez, D. Rodríguez, and N. Santiago
Electrical and Computer Engineering Department
University of Puerto Rico at Mayagüez, Mayagüez, Puerto Rico

1 Abstract
Reconfigurable computing devices have the potential of accelerating computationally intensive software applications by moving some of its operations into hardware. To make this possible, a communication mechanism is required between the software and the target hardware. It is generally assumed that there are readily available solutions for establishing such a communication, but this is hardly the case. This project presents the details of implementing a communication scheme between high-level software and FPGA-based accelerators residing in a Nallatech FPGA board. The implemented scheme was tested with a Corner Turning block, which is part of an image formation algorithm in a Synthetic Aperture Radar data processing system. Data was successfully transferred between the Corner Turning block and the host system with low overhead in timing and area consumption.

2 Development Platform
- **Motherboard:** Nallatech BenNUXEY-PCI-4EP50-6A 64-bit PCI board.
- **PCI Interface FPGA:** Spartan II; permits communication between the host system and the board. Also contains some board functions.
- **Integrated User FPGA:** Xilinx Virtex 2 Pro 50; contains the Communication Core to transfer data to and from the PCI interface.
- **BenDATA Module:** Nallatech BenDATA/4-5X55-11.
- **Target FPGA:** Xilinx Virtex 4 4V5X55; receives and sends data from the Integrated User FPGA and contains the user’s design.
- **Software:** Consists of Nallatech’s DIMEScript interpreter and the code for board and FPGA setup, data transfer, and process control.

3 System Implementation
Communication Core
- Data exchanges between the FPGAs and the PCI interface are performed using a proper communication protocol.
- Users can implement this protocol either directly or using Nallatech’s Spartan-to-Virtex IP core, which was the method used in this work.
- This is a data bus and control signals for DMA transfers to FIFO elements on the target FPGA.
- Also supplies data bus, address bus, and read and write strobes which connect to registers on the Target FPGA.
- User’s design application must be interfaced to this core.

4 Results
- **Virtex 4 Interface and User’s Design**
  - Resource: Number Available Percent
  - Total Number of Slice Registers: 438 438 100%
  - Used for McIP: 437 437 100%
  - User legalization: 1 1 1%
  - Total 4 Input LUTs: 704 484 100%
  - Bonded IOs: 79 940 125%
- **FIFO Generator Resource Report**
  - 32 bit, 32 depth - block memory
  - Slice Flip Flops: 162%
  - Gated Flip Flops: 118%
  - Total 4 Input LUTs: 100%
- **time Latencies during validation**
  - Operation: Time (μs)
  - Data Write: 2644.391
  - Data Read: 92.006

5 Conclusions and Future Work
- Describes a communication scheme that is platform dependent but easy to implement and that gives the user control of the communication components for adapting different designs.
- Scheme demonstrated with the use of a Corner Turning operation.
- Area consumption and timing statistics were obtained.
- Future directions include acquiring throughput measurements and quantifying how it is affected by the transmitted data size.
- Also additional communication components shall be added to access the BenDATA module’s external memory banks.
- Other tasks include interfacing different hardware designs, and exploring the use of other languages besides DIMEScript.

6 References

*This work was supported in part by Lockheed Martin Corporation*