www.hcs.ufl.edu
High-performance Computing & Simulation Research Lab

UF UNIVERSITY of FLORIDA

Experimental Analysis of Multi-FPGA Architectures over RapidIO for Space-Based Radar Processing

Chris Conger, David Bueno, and Alan D. George

HCS Research Laboratory
College of Engineering
University of Florida

Project Overview

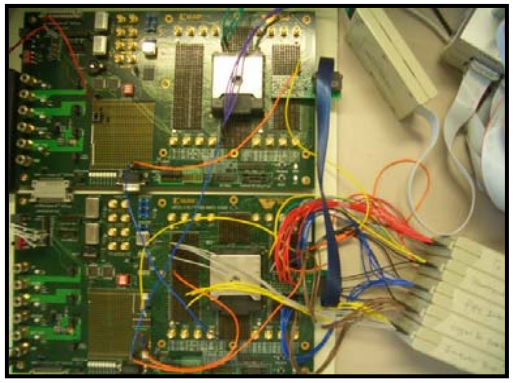
- Considering advanced architectures for on-board satellite processing
  - Reconfigurable components
  - High-performance, packet-switched interconnects
- RapidIO as candidate interconnect technology
- Ground-Moving Target Indicator (GMTI) case study application
- Sponsored by Honeywell Electronic Systems Engineering & Applications
- Experimental research, with focus on node-level design and memory-processor-interconnect interface architectures
  - FPGAs for main processing nodes, parallel processing of radar data
  - Computation vs. communication: application requirements, component capabilities
  - Hardware-software co-design
  - Numerical format and precision

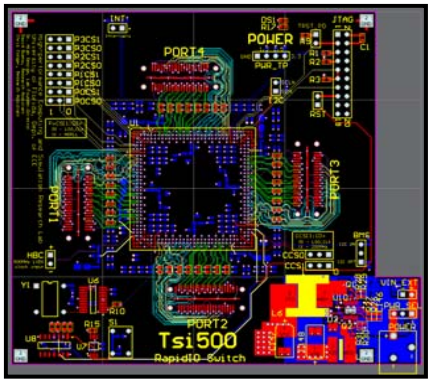
Image courtesy [1]

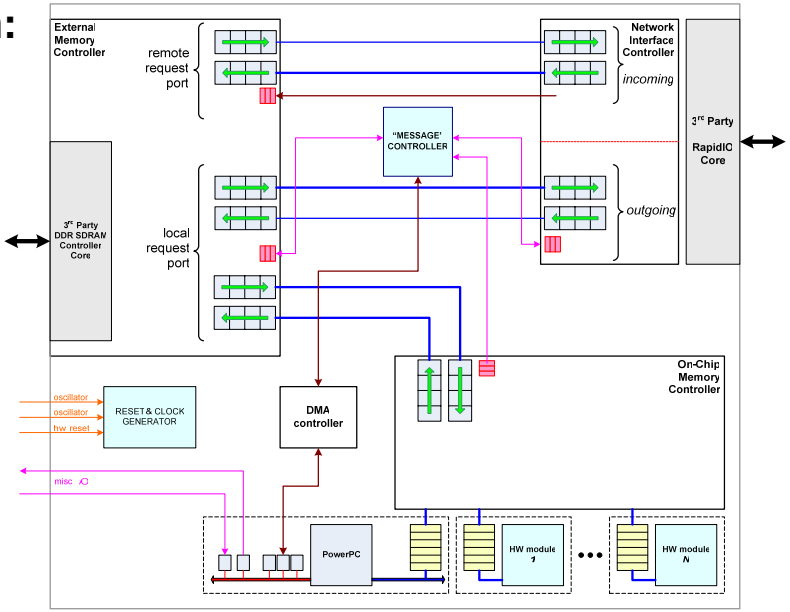
[1] http://www.noaanews.noaa.gov/stories2005/s2432.htm

Hardware Testbed

- **Custom-built hardware testbed, composed of:**
  - Xilinx Virtex-II Pro FPGAs (XC2VP20-FF1152-6), RapidIO IP cores
  - 128 MB SDRAM (64-bit @ 125 MHz for 8 Gbps memory bandwidth per-node)
  - Custom-designed PCBs for enhanced node capabilities
  - Novel processing node architecture (HDL) connecting processing elements, external memory, and network interconnect
- **Performance measurement and debugging with:**
  - 500 MHz, 80-channel logic analyzer
  - Xilinx's ChipScope Pro JTAG debugger

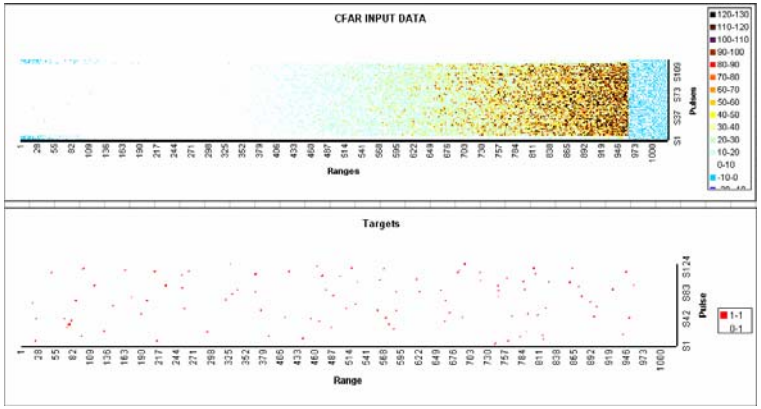


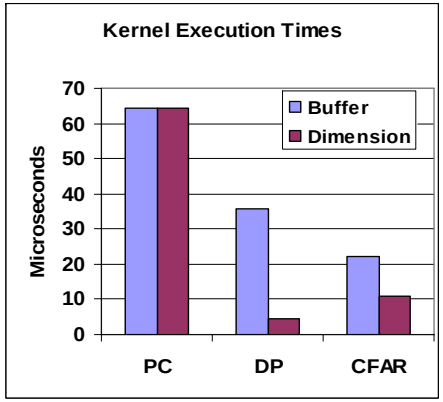


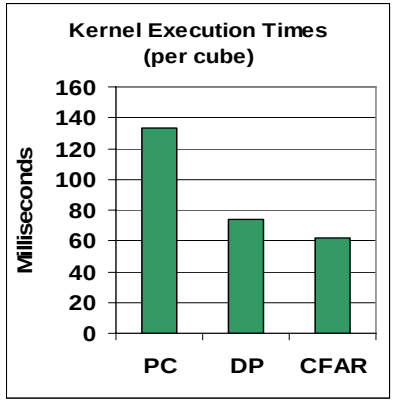


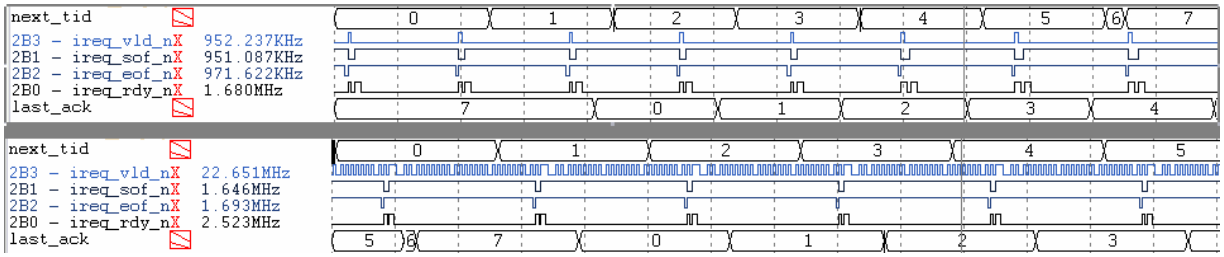
Experimental Results

- Low-level, fine-grained visibility into system operation
- Variable architectures and hardware parameters due to reconfigurable components
- Gaining insight into optimal memory hierarchy design approaches, feasibility of reconfigurable technology for space-based radar processing









$T_{kernel} = 2 \cdot T_{trans} + N \cdot T_{steady}$,

where

$T_{trans} = DMA_{blocking} + MAX[DMA_{blocking}, PROC_{buffer}]$

$T_{steady} = 2 \cdot MAX[(2 \cdot DMA_{blocking}), PROC_{buffer}]$

N = general term for number of iterations

$DMA_{blocking}$ = time to complete a blocking DMA transfer of M elements

$PROC_{buffer}$ = time to process M elements of buffered data (depends on task)