

Progress on two projects:

Efficient space radiation computation with parallel FPGA High-performance Martian Space Radiation Mapping

Liwen Shih

ABSTRACT—In support of NASA’s goals for safe, efficient, and cost-effective deep space missions, we applied high-performance computer techniques to enhance space radiation analysis for the Moon, Mars, and beyond. We designed a fault-tolerant field-programmable gate array (FPGA) that is double precision IEEE-compliant, floating-point parallel core, and designed to handle bottleneck functions impeding NASA’s space radiation computation times. We also are looking into OpenMP computation cluster/grid enhancement for space radiation analysis at UH TLC² as well as TeraGrid.

INTRODUCTION

In 2007, we continued to explore state-of-the-art reconfigurable FPGA (field programmable gate array) computer hardware and OpenMP multi-thread execution on large computer clusters/grids to help modernize NASA space radiation HZETRN (high-charge energy transport) computation. HZETRN is the space radiation dosage/flux software provided by NASA to estimate high-energy nuclear transport across materials being tested. The HZETRN model has been developed as an accurate scientific model, but the implementation of the model in FORTRAN-77 code based on a VAX 4000 mainframe is slow and inefficient. Currently, radiation exposure is underestimated by 15-30 percent which dangerously affects the safety of the mission and the cost-effectiveness of spacecraft shielding.

We examined multithread code optimization and parallel FPGA options for the major performance bottleneck functions in the HZETRN source code, including the Phi/interpolation function (which alone takes 34 percent of HZETRN run time). Our preliminary 8-bit floating point FPGA prototypes for bottleneck functions showed up to 325-650 times speedup over VAX. In 2007, we have extended the co-processor accelerator design for IEEE-compliant 64-bit double-precision floating-point (FP) calculations, and with multiple 64-bit FP cores on one FPGA chip with built-in fault-tolerance. Even though researchers at NASA LaRC intend to lend UHCL the Starbridge Hypercomputer-38 with 10-chip FPGA array (also called the “supercomputer-in-a-box”), we have chosen to look for easily replaceable COTS FPGA alternatives, due to the concern regarding maintenance and liability issues with loaned equipment. We thus developed a weighted-score system selection approach and are now turning our attention on to a reconfigurable DRC dual-core board.



Liwen Shih

WEIGHTED-SCORE FPGA ACCELERATOR TARGET PLATFORM SELECTION

With the proliferation of faster and denser semi-conductor technologies, current state-of-the-art logic devices provide viable application-specific alternatives to general-purpose computing. Applying dedicated logic to perform computational tasks with FPGAs while in tandem with standard CPUs achieves higher performance gains than could otherwise be attained with clusters of standard CPUs. In the realm of reconfigurable computing (RC), the selection process must be carefully conducted to leverage cost, performance, and ease of use. We examined different criteria used to select applicable

system architecture as well as hardware accelerator boards for NASA space radiation HZETRN software. The selection criteria were scored and weighted, with the final selection resulting in the DRC Corporation Reconfigurable Processor Unit. This is a small-form device that fits into a CPU socket on an AMD Opteron system with a HyperTransport bus. (See Figure 1.) In addition, the DRC RPU 110-L200 has the largest local SRAM of all that we compared. The larger local memory is a clear advantage when performing calculations that will require transfers between larger data structures.

PARALLEL FAULT-TOLERANT, FLOATING-POINT, MULTI-CORE FPGA ACCELERATOR DESIGN

To improve the floating-point performance for scientific computation such as HZETRN, we further developed the implementation of generic parallel IEEE-compatible floating point units implemented in an FPGA along with a simplified memory interface to a low-cost FPGA development board.

The implementation of the floating-point core was completed in a VHDL (VHSIC Hardware Description Language)

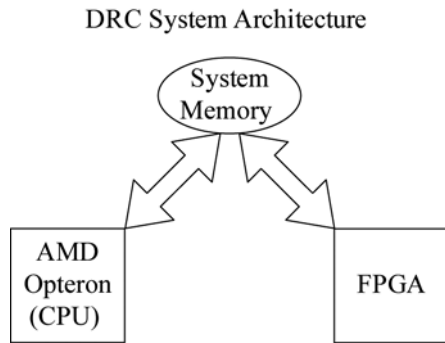


Figure 1. DRC system architecture with duo slots for AMD core + a custom FPGA Accelerator

specification of about 40 pages. Each entity implementation in VHDL was set with a global constant representing the data path bit-width for the module. Scalable design allows for easy data path width tuning from 32-bit to 64-bit or even down to 8-bit floating-point. (See Figure 2) Data path bit-width scalability is possible as each of the entities shares these global constants to set its variables, loops, and such.

In order to ensure that the floating-point core operated as intended when placed on an FPGA, a low-cost development board (Xilinx Spartan-3 FPGA along with 2 MB of RAM) was used for testing purposes. (See Figure 3.) The placing and routing were achieved with success. To further verify functionality, a “hard-wired” test was performed on the FP core by setting a static input to the two inputs of the FP core along with pin output to the output of the FP core. It was then started and tested using a breadboard with pin outputs and a multi-meter.

A dual-port interface to a SDRAM memory was also experimented with and made functional. Data flowed back and forth between a host PC and the target FPGA. However, the interconnect was far from ideal, as expected with our limited, low-cost, student-lab-grade testbed. Commercial solutions do exist that are made exclusively for high-bandwidth connections from a host PE to shared memory on an FPGA for future integration with HZETRN code.

Also, a sequencer was architected to load sequential data into shared memory and have an FPGA execute a parallel operation on the data. (See Figure 4.) For a parallel load application, an FPGA with a large number of multiplexers would be necessary to link up the four individual arithmetic units (add, subtract, multiply, and divide), as well as the FP core itself. Some other observations were:

- Memory load was slow from PC to FPGA target board. Loading 100K elements takes ~78 seconds through a parallel port, suggesting the need for much higher-grade bandwidth interconnect
- The sequencer clocked at 100MHz; the FP cores clocked with divide-by-2 at 50MHz
- Limited space meant not enough room for subjective testing of our limited hardware testbed
- The Xilinx Spartan 3S1000 supports a total of 15,360 logic slices
- Dual FP64 cores with glue (4370 x 2) = 8740 slices

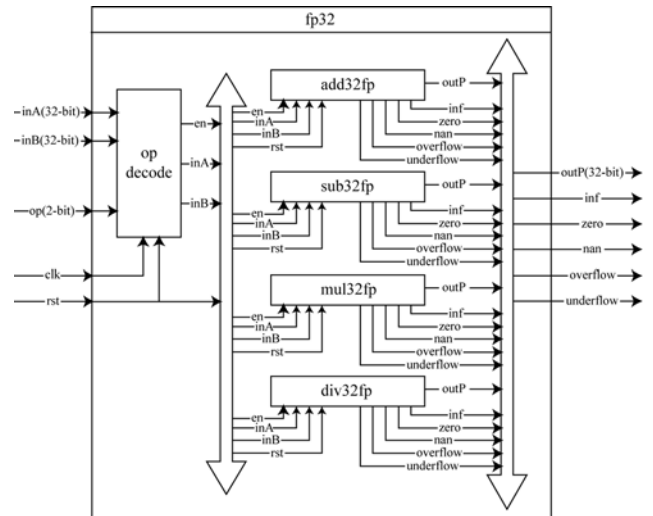


Figure 2. Extending 8-bit float FPGA prototype core to IEEE 64-bit double-precision

- Dual-port SDRAM->parallel interface = 350 slices
- Parallel sequencer and associated logic = 3970 slices
- Device utilization:
 - 13060 out of 15360 logic slices ~85 percent (only fits 2 FP cores in this hardware test)
 - Block RAM (registers) ~10 percent

With the addition of better tools and more convenient densities, it is highly possible to expand this architecture to different silicon. It would be beneficial to see the performance and features of more advanced FPGA with larger interconnect blocks. We will look into the reconfigurable FPGA fabric with a built-in PowerPC core that can communicate within nanoseconds.

HZETRN95 WITH OPENMP ON MASSIVE TLC² COMPUTATION GRIDS/CLUSTERS

Yet another effort in 2007 was to explore the large computation cluster/grid approach of the HZETRN program using parallel processors. The hardware and platform used for this parallel processing on a Linux operating system clusters were provided by the Advanced Computing Laboratory of the Texas Learning and Computation Center (TLC²). The goal of this work was to determine whether OpenMP parallelization structure has any further potential for accelerating the execution of the HZETRN program. We found so far that the original, very little parallelism-minded HZETRN software design fares slightly better with OpenMP on larger tightly coupled share-memory systems.

In conclusion, we continue to work with NASA LaRC and JSC on HZETRN code modernization and the planned future expansions that can benefit design and engineering of lighter and more cost-effective shielding material for use in NASA spacecraft, e.g., CEV Orion. With the newly emerging technology in parallel network clusters/grids (e.g., TeraGrid), reconfigurable FPGA fabric, fast high-bandwidth interconnects and FPGA arrays (e.g., NASA’s Hypercomputer-38), it is highly promising that a high-performance improvement in

