

Custom Multi-FPGA System for Rapid Prototyping of Innovative PIM Architectures

Maciej Brodowicz
maciek@cct.lsu.edu

Thomas Sterling
tron@cct.lsu.edu

Center for Computation and Technology
302 Johnston Hall
Louisiana State University
Baton Rouge, LA 70803

phone: (225) 578-4012
fax: (225) 578-5362

This paper presents an FPGA-based testbed designed and built for quick development, verification and refinement of architectural concepts related to the implementation of advanced processor architectures. The original purpose of this platform is to evaluate the design of individual components of the class of architectures called **Processor-In-Memory**, as well as a reduced-scale multi-PIM system running at clock frequencies within an order of magnitude of its silicon implementation. While the system has already proven its usefulness in several PIM component studies, its possible applications range much farther thanks to its reconfigurable logic capacity, connectivity options and flexibility of control and interaction.

Keywords: *FPGA, PIM, rapid prototyping*

1 Introduction

The Processor-In-Memory architecture [1,2] exploits available memory bandwidth by placing the computing logic at close proximity to the memory array (at the secondary sense amplifiers) and taking advantage of wide buses to perform data transactions. The latency of memory accesses is also mitigated, firstly by avoiding transferring the data through the package pins, and secondly by providing a direct access to the contents of row buffer. Such an arrangement, a PIM node, is replicated multiple times on a silicon die, yielding very high cumulative processing speed and aggregate memory bandwidth. The nodes in PIM architecture are tightly connected both on the chip using a low-latency communication bus, as well as across multiple chips through an external interconnection network.

To support the development of PIM architecture efficiently, the test platform must satisfy the following set of requirements:

- The aggregate logic resources on the board must accommodate the entire PIM node structure. At the time of development this implied a multi-FPGA platform, as either no single FPGA device on the market was offering the required capacity, was prohibitively expensive, or its availability was questionable.
- The prototype system has to simulate interactions between several PIM nodes. To accomplish this, multiple FPGA boards are necessary together with a means enabling flexible (in terms of latency, throughput and network topology) inter-board communications.
- The test platform has to mimic the wide memory buses used by computing nodes by deploying FPGAs with a sufficient number of I/O pins and combining a number of discrete memory devices to act as a single, wide data word module.
- The design has to be modular, allowing the addition of extra boards when the expansion of the test setup is necessary. The built-in interconnect cannot inherently enforce a limited node count topology.
- The prototype board must embed a supervisor logic, which initializes the communication with an external controlling entity (such as a PC), configures the FPGAs using bitstreams provided by the user, monitors the board hardware, and reports status of these operations. A secondary out-of-band channel for data exchange between the logic implemented in the FPGA and the controlling PC is a desirable addition. All these functions can be conveniently carried out by a dedicated microcontroller with a custom firmware.
- The test platform should also feature a minimal, highly customizable monitoring and display circuitry independently, if possible, from the main FPGAs and the supervisor. That way, critical events can be intercepted and analyzed even in a case of system lockup. An additional benefit is that event counters can be flexibly defined and interfaced to without consuming much of the main FPGAs' resources. This functionality can be implemented in a cheap, small-scale FPGA.
- Since the final PIM specifications are not known, the testbed has to be flexible enough to adapt to the changing requirements. This postulates that the design has to exhibit some degree of symmetry to enable comparable layout and functionality if, for example, two simplified PIM nodes must be instantiated on a single board.

2 Architecture

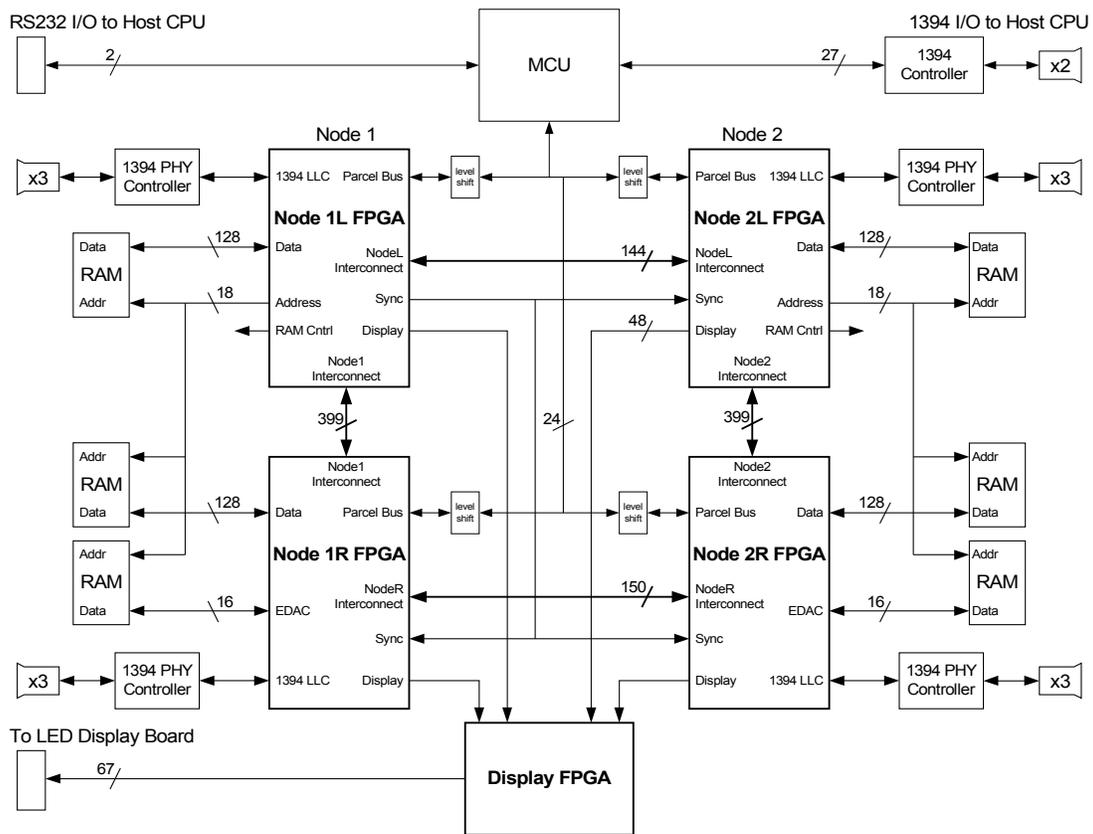


Figure 1. Structure of the FPGA testbed system

The simplified testbed board diagram is shown in Figure 1. Its major components include:

1. Four Xilinx Virtex-E FPGAs (part number XCV2000E-6FG1156) with 2.5 million system gates¹ per device. The FPGAs are arranged in two pairs, each of which is equipped with a local memory bank. The interconnect between the “left” and “right” FPGA of each pair is over 300 signal lines wide. Each of the FPGAs attaches to an IEEE1394 driver to support an independent, out-of-board communication link.

The corresponding FPGAs in each pair also connect via buses over 100 lines wide. The wide interconnects make possible low-latency, high-bandwidth data transfers between PIM components distributed across the FPGAs, thus enabling their good integration and minimizing the potentially negative effects of multi-FPGA approach. Due to the short span of the interconnects, the signaling protocol can be single-ended, resulting in a more efficient pin allocation.

2. Two 8MB SRAM memory banks, each composed of 17 chips with 16 data lines and 10ns access time. This arrangement provides a 256-bit wide memory bus with an additional 16 lines for error correction/redundancy. The main data lines are split uniformly between the FPGAs in each pair (128 signals). To balance the pin assignment, the memory address bus connects to the “left” FPGA and EDAC lines to the “right” FPGA in each pair. Note that the memory banks are disjoint, hence configuring a two-node PIM testbed is trivial.
3. The board supervisor is a 32-bit Hitachi/Renesas SuperH-2E RISC MCU with a flash program memory running at 40MHz. Besides being able to steer the FPGA configuration lines, this microcontroller also drives a 24-bit bus attached directly to I/O pins in each FPGA, thus enabling

1 According to the Xilinx datasheet.

data transfers between the synthesized logic cores and an external computer. The PC connectivity may be achieved either through the IEEE1394 link layer controller, or, for the initial stages of firmware development, a standard RS-232 serial link.

4. The display and auxiliary I/O circuit is based on a Xilinx XC2S200-5FG456 FPGA. This component can communicate with each “large” FPGA independently through dedicated 48-bit buses. It also interfaces with two miniature connectors to allow an outboard I/O and the attachment of an external display board. Since the logic structure of this part can be freely defined, a number of possible additional applications is possible. For example, since the data transfer over IEEE1394 channels is not always desirable or convenient, it may be used as a fast, configurable parallel I/O interface. Another use could be as a collection of hardware counters with dedicated event signal lines sourced by other FPGAs, or logic debugger with signal analysis implemented in hardware. Such an arrangement is convenient, as direct pin access of any FPGA packaged in a BGA case frequently proves quite complicated.
5. Communication interfaces can be divided in two groups: one supporting system scaling, and the second interacting with the external control unit. The first group includes IEEE1394 physical layer drivers (four per board), supporting 400Mbps wire rates and a 3-connector interface each. The choice of PHY-only drivers was dictated by the necessity of latency minimization and the fact that the full link layer functionality would introduce an unwanted overhead increasing the logic requirements on the FPGA side. Since IEEE1394 has point-to-point connectivity, a large variety of interconnect topologies can be built, ranging from a simple ring to trees of hypercubes. The control unit (PC) link relies on the standard IEEE1394 LLC from Fujitsu (also 400Mbps) driving two connectors, which allow easy daisy-chaining of multiple boards without the need for multiple control units to supervise the stack of PIM emulators.
6. Auxiliary circuits, not shown in the diagram, perform less prominent, yet essential functions. They include programmable clock generators (configured by the board supervisor), reset sequence generator, FPGA die temperature monitor, regulated power supplies delivering 5V, 3.3V, 2.5V and 1.8V used by board components, logic level shifters, and JTAG interface.

3 Status

The current prototype board, shown in Figure 2, is a second revision (*beta*) of the original design. This iteration addressed a significant number of the shortcomings of the *alpha* board, which effectively made it hard to use and inflexible. Available funding and PCB fabrication errors limited the number of assembled boards to four from the intended sixteen; all of them passed verification and testing, and already run valid experiments.

To facilitate the testbed use, an integrated configuration and deployment software package was developed. This application, running on MS Windows-compatible machines, provides a menu-driven GUI invoking a number of tasks, including board bootstrap and connection to the supervisor, download of MCU firmware extension, change of settings of the selected board components (FPGA clock frequency, the value of critical FPGA core temperature, *etc.*), selection of the FPGA configuration bitstreams, FPGA configuration and configuration reset, node logic reset, start and interpretation of various system tests (memory, on-board interconnect and IEEE1394 drivers), and finally the initialization of an interactive session with the supervisor.

The configuration data implementing the logic structures on the FPGAs are generated using a standard Foundation software from Xilinx, which supports both VHDL and Verilog (including mixed-language designs). The component descriptions are validated prior their testing on the FPGAs using ModelSim package.

4 Initial Experiments

The FPGA testbed has been successfully used to verify the implementation of two components of the PIM node: parcel handler and IEEE754-compliant floating-point unit. Both experiment sets were

performed on a single board. The first test focused on the elementary functionality of the parcel² handler. The sequence of inspected operations included parcel reception from the controlling PC, accumulation in an internal buffer, extraction of arguments and operation type, invocation of the operation in a primitive integer ALU, creation of the return parcel containing the result and its emission to the PC. The objective was to verify that parcel receive and transmit pipes are working correctly, and the parcels can be properly decomposed, interpreted, used to modify the node's state, and assembled for transmission.

The second experiment involved a 4-stage FPU performing additions and multiplies on the full range of double precision floating-point numbers (including infinities, NaNs and denormalized numbers). A script in Python was generating random pairs of arguments, which were then passed through the control GUI to the FPGA logic. The output stream containing the results was piped back to the script, which compared the outcomes with the results computed locally. The test run continuously over several days with no errors detected.

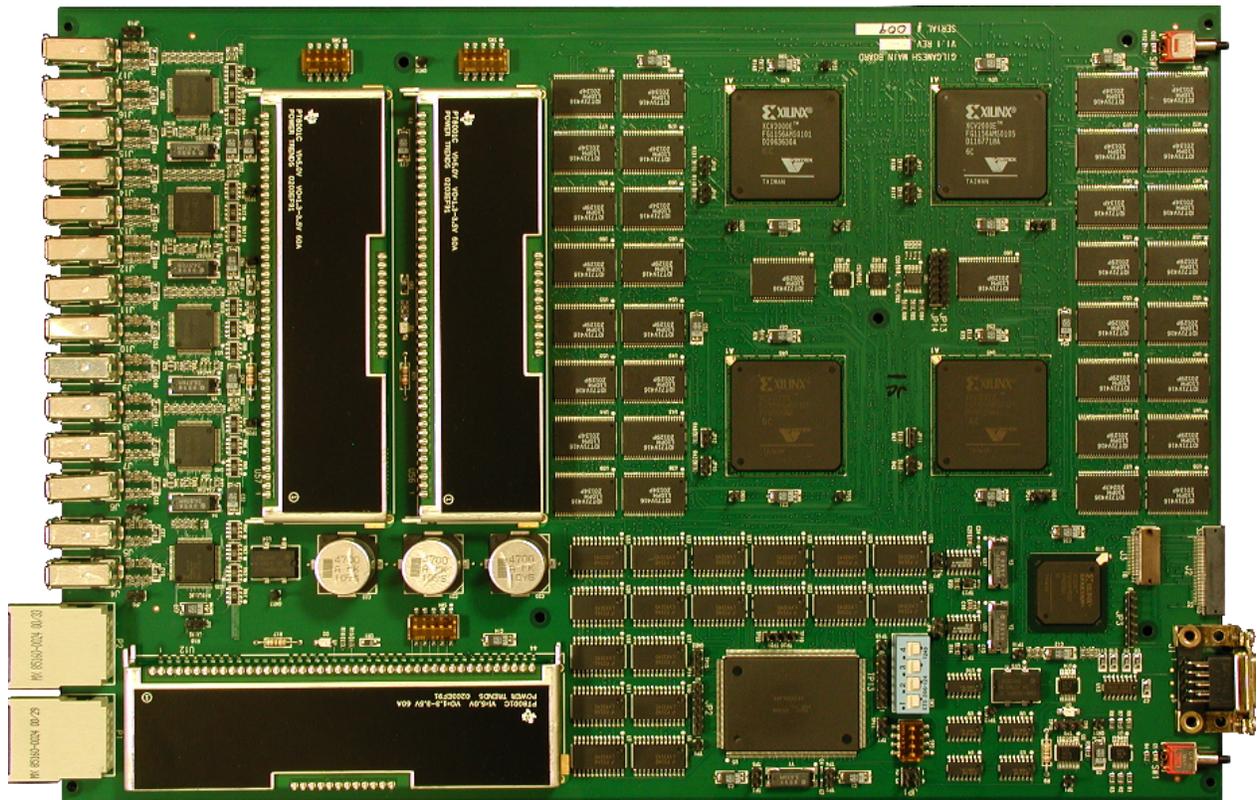


Figure 2. Assembled FPGA board

5 Proposed Work

The developed testbed will continue to be used in a variety of further experiments. The primary goal is the verification in hardware of the Verilog implementation of the remaining components of the PIM node. Its most prominent elements include: message-driven dispatch unit, multi-threaded scheduler, distributed memory management and wide ALU.

Another planned application of the FPGA testbed is a small-scale emulation of the Continuum Computing Architecture [3], which essentially models interactions in advanced cellular automata network as a foundation for highly scalable parallel computer. The FPGAs lend themselves naturally to this project, as they offer interconnect-rich topology and support easy replication of small structures.

2 Parcel is a **parallel control element**, an atomic communication unit capable of carrying data between the nodes and triggering actions at remote sites.

Finally, the boards will provide the basis for the new architecture laboratory being established at the Center for Computation and Technology at the LSU. The authors hope that this new facility will help attract the students to the exciting area of computer architecture and provide hands-on experience, perfectly complementing the theoretical coursework.

Acknowledgments

The authors would like to express their deep gratitude to William Whitaker of NASA/JPL for his passionate involvement in the development of the FPGA board and many important contributions to the PIM node layout. We also wish to thank Keith Underwood of Sandia National Laboratories for providing the working set of VHDL sources containing the floating-point unit design and spending his valuable time helping us to port it to the prototype board, and debug the problems.

References

- [1] T. Sterling and H. Zima, "Gilgamesh: A Multithreaded Processor-In-Memory Architecture for Petaflops Computing," in Proc. Supercomputing '02.
- [2] E. Upchurch, T. Sterling and J. Brockman, "Analysis and Modeling of Advanced PIM Architecture Design Tradeoffs," in Proc. Supercomputing '04.
- [3] M. Brodowicz and T. Sterling, "Continuum Computer Architecture for Nano-scale and Ultra-high Clock Rate Technologies," in Proc. IWIA '05.