The Transmogrifier: The University of Toronto Field-Programmable System

David Galloway, David Karchmer, Paul Chow,
David Lewis, Jonathan Rose

Department of Electrical and Computer Engineering
University of Toronto

Abstract

A fully programmable circuit board consisting of FPGAs, RAMs and programmable interconnect chips has been constructed. The hardware is described, along with the software used to program it. Three sample applications, a Viterbi decoder, a re-organizable memory, and a logarithmic arithmetic unit have been implemented on the board.

1. Introduction

Field-programmable logic devices in the form of FPGAs have many advantages compared to custom logic devices. They offer a high degree of flexibility, reducing the number of distinct part types that need to be kept in inventory. They allow designers to rapidly and inexpensively implement circuit designs in a few minutes at a cost of a few hundred dollars, compared to months and tens of thousands of dollars for custom designs.

FPGAs, however, suffer from lower logic density and lower speed than custom logic. The low logic density problem can be addressed by constructing larger field-programmable systems consisting of multiple FPGAs, RAMs, and a programmable interconnection network. Together with the appropriate software tools, a field-programmable system allows a user to create designs that are too large to fit into any single FPGA chip.

Previous systems containing multiple FPGAs have been constructed for logic emulation [Butts92], the acceleration of software algorithms [Gokhale91], and rapid prototyping [VandenBout92]. Most previous systems used a fixed interconnect structure. Recently introduced programmable interconnection chips allow a more flexible programmable interconnection network to be used.

We have constructed a field-programmable system called Transmogrifier 1, or TM-1. We have used it to support the design of several sample circuits and are developing CAD tools to support the system.

Section 2 of this paper describes the hardware architecture of our system. Section 3 discusses the CAD synthesis software we are using and developing. Section 4 discusses the interconnection network architecture of TM-1. Section 5 describes how the designs are tested. Section 6 discusses three example designs that have been implemented on TM-1, and Section 7 describes ongoing work.

2. Hardware Architecture

A field-programmable interconnect chip (FPIC) is a new kind of field-programmable device. Like an SRAM-based FPGA, it contains programmable switching circuitry that can pass signals around the chip. Unlike an FPGA, it contains no programmable logic blocks. Instead, an FPIC contains a large number of I/O pins, and the routing and switching resources needed to passively connect almost any combination of those pins.

Our current field-programmable system, TM-1, is based on an Aptix AXB-GP2 field-programmable circuit board [Aptix93a]. The board is approximately 300 mm by 175 mm in size. It contains two Aptix AX1024 FPIC chips that are connected to over 1700 holes on the board. Standard DIP packages can be directly inserted into these holes, and the FPIC devices can be programmed to wire these packages in any way. Provision is made for connecting pins to power and ground, and these pins are indicated as such to the routing software.

TM-1 uses the Aptix board to interconnect four Xilinx XC4010 FPGA chips [Xilinx92a] and four 32Kx9 SRAM chips [Motorola92], mounted on special PGA and PLCC adapters. Ribbon cables carry 72 bi-directional signals to a separate interface board, as illustrated in Figure 1.

The interface board contains a single XC4010 FPGA, that is programmed to act as a communications controller. The interface board has 72 signals leading to TM-1, and a 16 bit parallel port cable leading to a SUN workstation. The communications controller implements a simple protocol that allows a program on the SUN to communicate with a design running in TM-1.

Separate cables are used by the SUN to program the Aptix FPIC chips and the Xilinx FPGAs.

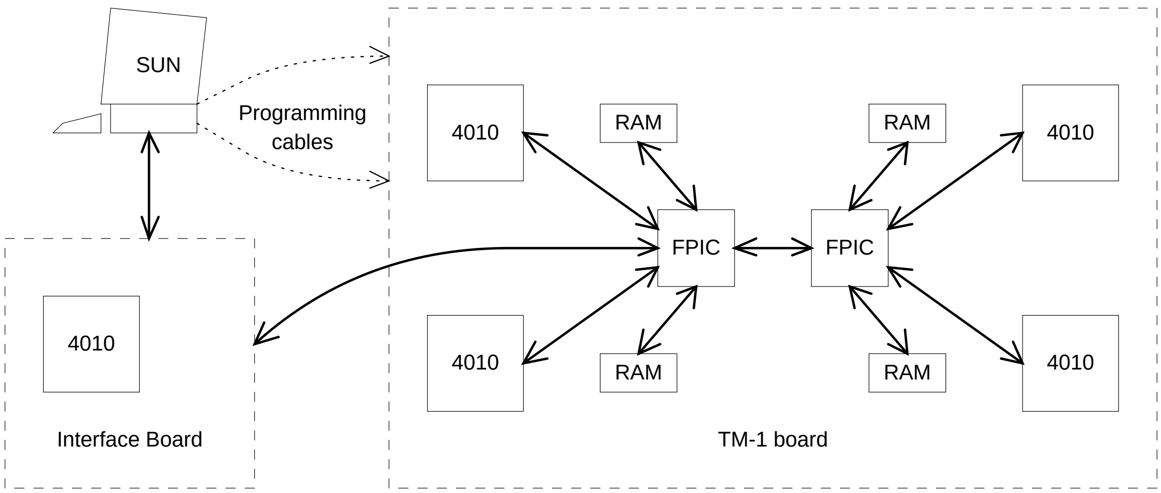

Figure 1: Transmogrifier 1

3. CAD Synthesis Software

A circuit to be implemented on TM-1 is first created in the Xilinx XNF format using commercial or university developed CAD tools. Once individual XNF files have been created for each of the FPGAs on the board, they are processed by the Xilinx technology mapping, placement and routing program, **ppr** [Xilinx92b], which produces the FPGA programming files.

To make the individual XNF files that **ppr** needs, the circuit must be broken into pieces that are small enough to fit into the chips available on TM-1. This partitioning step can be done manually, but manual partitioning can be time consuming and error prone. We are developing an automatic partitioning program that can take a large design in XNF format and partition it into pieces that will fit into the individual FPGAs. The program, called **part**, is based on the Fiduccia and Mattheyses partitioning algorithm [Fiduccia82], with extensions for multi-way partitioning and timing sensitive preclustering [Shih92].

While placing and routing each FPGA, **ppr** chooses the pins to be used for each I/O signal. The pin assignments are extracted from the **ppr** output, and are included in a netlist that describes the inter-chip wiring and the external input and output positions. This netlist is taken by the Aptix **axess** software [Aptix93b], and used to program the FPIC chips. After the FPIC chips are programmed, the remainder of the chips on the board are powered up, and the Xilinx **xchecker** software programs the FPGAs.

4. TM-1 Interconnection Network

In typical multi-FPGA boards, e.g. [Gokhale91], the wires that connect the FPGAs together are usually hard-wired from pin to pin. This places two restrictions on the circuit implementation: first, the number of signals that can be passed between any two FPGAs is limited by the existing wires, which places severe restrictions on the multi-chip partitioning and leads directly to lower utilization of the FPGAs. Secondly, the CAD programs that are used to place and route the FPGAs must agree on which pins are to be used to transfer the signals.

In TM-1, all inter-FPGA signals pass through an FPIC chip, which can be programmed to connect them in any way. This allows the partitioner to use the pins on each FPGA to best advantage. It can, if the circuit requires it, connect 90% of the pins on one FPGA to a second FPGA, which would not be possible on a general purpose board with hard-wired connections. As well, the Xilinx **ppr** place and route software is allowed to make its own decisions about which pins are used for external signals. This freedom eases the task of placing and routing the internal logic of each chip, which in turn should give better density and speed. The programmable interconnect of TM-1 can then connect the inter-chip signals, regardless of which pins **ppr** happens to use on each chip.

The disadvantage of using an FPIC for inter-chip communication, however, is that all signals incur an extra delay, (from 5 to 15 ns) whereas hard-wired connections add little delay. We hope to mitigate this cost by using better partitioning algorithms. After more experience, we will have a better idea of the ideal mixture of hard-wired and programmable connections to use in future field-programmable system designs.

5. Design Testing

Designs running in TM-1 can be observed using a standard logic analyzer. An analyzer can be connected to a set of connectors on the TM-1 board, and any signal that is external to a chip on the board can be easily routed to the connectors by the Aptix software.

Programs running on the SUN workstation can also observe or modify any of the 72 signal lines that are connected to the interface board. The parallel port on the SUN is memory mapped into a program's memory space, and a simple protocol is used to pass data between the SUN program and the design through the interface board.

6. Example Applications

This section describes three working applications that have been tested on TM-1.

6.1 A Viterbi Decoder

We are constructing a long constraint length Viterbi decoder. A Viterbi decoder interprets a form of convolutional code that has been used to improve communication over noisy channels such as deep-space links. Our architecture consists of 13 processors with local memory connected in a ring.

The full version of our Viterbi decoder will probably consist of at least 13 Xilinx 4010s for the processors, 12 Xilinx 4010s for RAM controllers, and 13 Xilinx 4005s for data switches. There will also be 39 256x9 RAMs and a few other small components. Most of the FPGA designs have been completed, with some minor modifications yet to be implemented.

A system consisting of one processor, one RAM controller, and three RAMs was being verified through simulation, but this became tedious, so it was decided to use TM-1. The system was partitioned by hand, with one 4010 programmed as a test vector generator, one used as the processor, and one used as the RAM controller.

The test generator FPGA and processor worked at 10MHz; however, the RAM controller did not function correctly, and is currently being re-designed. It was desired that the system could be tested at higher than 10MHz, but the clock signal became degraded at higher speeds because of high fanout through the FPIC. No special treatment was being given to the clock signals. However, the use of TM-1 has allowed verification and testing of the Viterbi processor node at much higher speeds than could be done with software simulation.

6.2 A Memory Organizer

A locally written CAD tool called MemPacker produces a circuit that will let a small number of physical RAM chips emulate a larger number of logical memory chips of different shapes, sizes and speeds.

The input to this tool is a list of required logical memories with their parameters (ie. width, depth and access time). The tool produces a netlist used to build the circuit (ie. the multiplexers and control) in a programmable device (in our case, Xilinx's XC4000 series FPGAs).

The program can generate the circuitry needed to use the memory in TM-1 as any number or types of memories, if they fit in the available space, and their required access time is met.

For example, MemPacker can configure logical RAM circuits that satisfy these sample constraints, that come from industrial ASIC designs:

- a RAM with 736 entries, each 16 bits wide; and four 368 entry RAMs each 16 bits wide
- three 688 entry by 4 bit; three 22 by 32; one 1024x4; two 168x12
- two 1620x3; two 366x11; two 168x12

TM-1 was used to test several circuits produced by MemPacker. The tool works, using about 5% of the space in the FPGAs for a typical example. The other 95% of the space can be used to implement the application's logic.

6.3 A Logarithmic Arithmetic Chip

Another application that has been implemented is a processor for logarithmic number system (LNS) arithmetic. LNS arithmetic represents a number by its base 2 logarithm using a fixed-point representation. Multiplication and division are easy in LNS, and are performed using fixed point addition and subtraction. Addition and subtraction are difficult and require the evaluation of some complicated transcendental functions [Lewis93].

The target application was to redesign an LNS processor that had been previously been designed for full-custom 1.2 micron CMOS. Using this example, we hope to compare the area and speed of the TM-1 implementation to a full-custom implementation. The processor can perform two multiplies or divides per clock cycle, and one addition or subtraction.

The TM-1 implementation offered many interesting differences compared to the full-custom design. New algorithms using a quadratic function interpolator and interleaved memory had been designed, and offer high efficiency in full-custom CMOS. The novel memory structure proposed in the full-custom implementation was not possible using FPGAs, but memory costs were much less in the TM-1 design, due to the available SRAM. The algorithm was also redesigned to reduce algorithm complexity at the cost of increased memory size. One hot controllers were used for the many barrel shifters that would be used in a full-custom design, but tightly encoded control reduced the amount of shift logic needed in the TM-1 design. The TM-1 design needed four 4010s to implement the circuit functions with a 5Mhz clock rate. The critical path in the TM-1 design was a 19*8 multiplier which is used twice per clock cycle to implement a 19*16 multiply.

7. Future Work

In a joint research project, Bell-Northern Research is producing a multi-chip module that is similar to TM-1. It will contain four XC4010 FPGAs, four 32x9 SRAMs, and a single Aptix AX1024 FPIC chip. Its dimensions will be 2.4x2.4 inches. The module will have 216 general purpose external signal pins. All the internal and external pins on the module will be wired to the FPIC chip.

We intend to implement other example circuits on TM-1 and on the BNR module when it becomes available. The experience gained will be used to design an improved second generation module, with higher speed interconnections and larger FPGAs.

8. Summary

This paper describes an ongoing project to examine the costs and benefits of field programmable systems and their associated software.

A fully programmable circuit board consisting of FPGAs, RAMs and programmable interconnect chips has been constructed. A CAD synthesis system has been assembled to allow designs to be tested on the board. Three sample applications have been tested, and more are in progress.

Acknowledgements

We are grateful to Jess Lee and David Yeh, who were the first real users of the system. The project is funded by grants from NSERC and BNR.

References

[Aptix93a] FPCB AXB-GP2 Data Sheet, Aptix Corporation, San Jose, California, February 1993

[Aptix93b] Aptix FPCB Development System Utilities, Aptix Corporation, San Jose, California, December 1993

[Babb93] J. Babb, R. Tessier and A. Agarwal, "Virtual Wires: Overcoming Pin Limitations in FPGA-based Logic Emulators", IEEE Workshop on FPGAs for Custom Computing Machines, April 1993.

[Butts92] M. Butts, J. Batcheller, J. Varghese, "An Efficient Logic Emulation System", Proceedings of ICCD 1992, pp 138-141.

[Fiduccia82] C.M. Fiduccia and R.M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions", Proc. of the Design Automation Conference, DAC-82, pp 175-181.

[Gokhale91] M. Gokhale, W. Holmes, A. Kopser, S. Lucas, R. Minnich, D. Sweeny, and D. Lopresti. "Building and using a highly parallel programmable logic array", Computer, Vol 24, No. 1, January 1991.

[Lewis93] David M. Lewis, "An Accurate LNS Arithmetic Unit Using Interleaved Memory Function Interpolator", *Proc. of Arith-11*, June, 1993, pp 2-9.

[Motorola92] MCM62110 32Kx9 Bit Synchronous Dual I/O Fast Static RAM with Parity Checker Data Sheet, Motorola Semiconductor, March 1992

[Shih92] Minshine Shih, Ernest S. Kuh, "Performance-Driven System Partitioning on Multi-Chip Modules", Proc. of the Design Automation Conference, DAC-92, pp 53-56.

[VandenBout92] D. Van den Bout, J. Morris, D. Thomae, S. Labrozzi, S. Wingo and D. Hallman, "Anyboard: An FPGA-base, reconfigurable system.", IEEE Design and Test of Computers, September 1992, pp 21-30.

[Xilinx92a] XC4000 Logic Cell Array Family, Xilinx, Inc, San Jose, California, August 1992.

[Xilinx92b] XACT Reference Guide - Volume 1, Xilinx, Inc, San Jose, California, October 1992.