

# **Verification, Testing, and System Integration of Reconfigurable FPGA Signal Processors<sup>1</sup>**

**Thomas J. Palermo, Louis F. Lentz, and Tim H. Noh**  
MTL Systems, Incorporated  
3481 Dayton-Xenia Road  
Dayton, OH 45432-2796

**Jeffrey Summers**  
U.S. Air Force, Wright Laboratories  
Wright-Patterson AFB, OH 45433

## **Abstract**

In this paper, we describe a totally integrated FPGA development platform, which very effectively provides the means for design, verification, testing, and system integration of reconfigurable logic designs based on the latest in FPGA technology. This unique integrated system concept makes complete the development cycle for a reconfigurable FPGA signal processor design, by merging in a seamless fashion, the capabilities of custom-developed design automation methodologies with conventional EDA Tools and a unique real-time reconfigurable FPGA hardware platform. This flexible board architecture represents a precise duplication of an optimized FPGA processing element architecture. This is contrasted to typical commercial FPGA development platforms, in which the hardware does not represent the actual architecture or allow operation at the clock speed of the target design.

VHDL is used extensively throughout the system. In fact, the common and essential link in the integrated system is use of VHDL-based EDA tools throughout, which allows implementation of an end to end testing, verification, and system integration concept.

## **1. Introduction**

Field Programmable Gate Array (FPGA) based reconfigurable processing and computing are finding many applications throughout the commercial and military marketplace. These systems offer the flexibility of a software implementation, yet approach the speed performance of dedicated hardware implementations, such as ASICs. Unfortunately, the development cycle for applications with these systems is currently a lengthy hardware design task. Development of high-performance designs requires that the design engineers have both significant FPGA experience, as well as detailed knowledge of the application and the targeted reconfigurable hardware devices.

In this paper, we describe a unified development system that effectively integrates the design entry, logic synthesis, simulation, and technology mapping steps with real-time verification, testing, and system integration. This unique integrated system concept makes complete the development cycle for a reconfigurable FPGA signal processor design, by merging the capabilities of custom-developed design automation methodologies with conventional EDA Tools and real-time reconfigurable FPGA hardware. VHDL is used extensively throughout the system, not only for design input, but also for functional and behavioral simulations for verification throughout the design process flow.

The flexible board architecture developed on the DRASTIC Program represents a precise duplication of an optimized FPGA processing element (PE) architecture. This is in contrast to most commercial FPGA development platforms, in which the hardware does not represent the actual architecture or operate at the full clock speed of the target design. The DRASTIC FPGA Board provides real-time I/O ports for direct interfacing to external data paths at the system level, but in addition provides internal data paths to the host PC platform. The internal data paths, based on the high-speed PCI Bus, allow design verification at real-time or near real-time, by allowing full speed data interchange with the host. For preliminary test and integration, the design can be operated at full clock speed, with stimulus data supplied from host memory. The resulting output data can be captured and analyzed by the host, again using the PCI interface. An additional feature of the PCI Bus interfaced FPGA board is that it may be used as an end platform for the processing of data.

The key features of the integrated FPGA development environment are:

- An integrated EDA Tool environment and FPGA hardware platform, linked together by the VHDL hardware description language
- VHDL design entry, with VHDL simulation at all intermediate steps of the design cycle
- MCM and FPGA level logic partitioning
- Real-time FPGA logic design verification in the actual target hardware architecture

## **2. System Overview**

The DRASTIC System, as shown in Figure 1, provides an effective, integrated environment for developing FPGA based hardware and systems. Through a tight integration of EDA and custom developed tools, it supports an efficient and effective design flow. This provides the user with an optimal development path that merges design automation tools with the reprogrammable hardware. At the top-level, the host system contains an integrated Tool Suite and the DRASTIC Reprogrammable Board.

Within the Integrated Tool Suite, the design flow is organized into the following major processes:

- Design Capture
- Logic Synthesis and Technology Mapping
- Partitioning
- Floorplaning
- Routing
- Design Verification

Each process in the suite utilizes VHDL simulation as a means to verify correct operation. This Integrated Tool Suite is described in more detail in Section 3.

Once a design has been implemented with the Integrated Tool Suite, an FPGA based reprogrammable board, contained within the system, is used to verify the design. The reprogrammable board is based on multi-chip module (MCM) architecture, in which each MCM consists of three Xilinx 4025E FPGAs and one megabyte of high-speed static random access memory (SRAM). This board is scalable and it contains, in its smallest configuration, four MCMs which can map a 300,000 gate sized design. In general, for design verification, stored stimulus data, used throughout the design flow, is used along with the capture and storage of output data processed by the reprogrammable board. For system testing and integration, real-time I/O is provided, which can be used in conjunction with internal stimulus and data storage capabilities via the host computer's local PCI bus. The architecture of the reprogrammable board is illustrated in a detail in Sections 4 and 5.

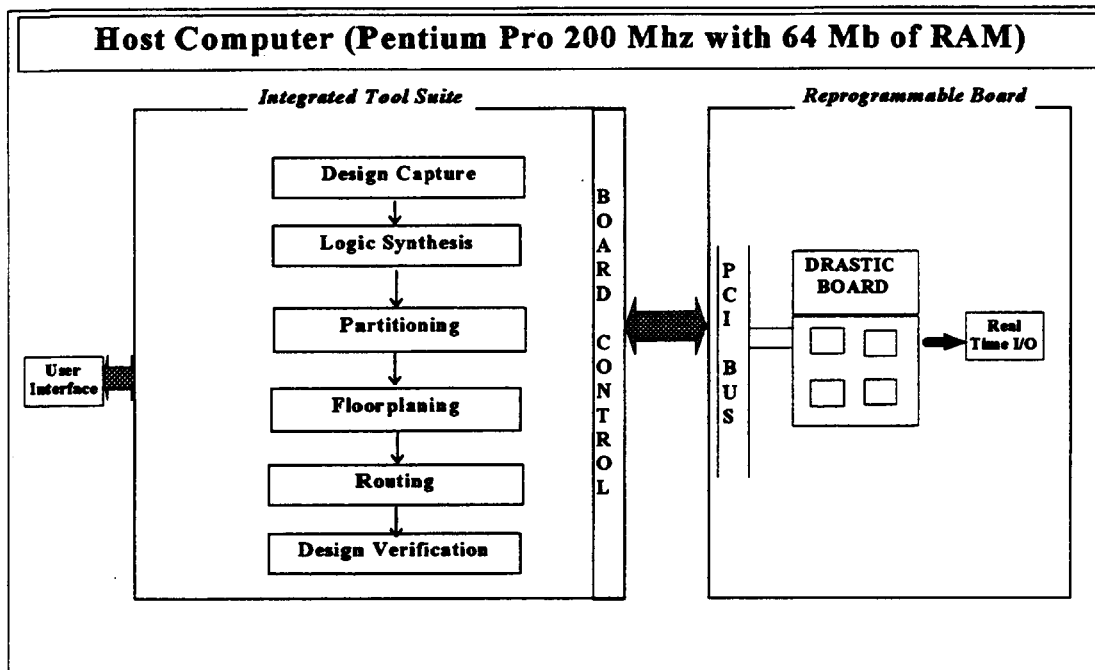


Figure 1. The DRASTIC System Overview

### 3. Integrated Development Environment (IDE), Testing, and Verification of Design

Overview of the Integrated Development Environment (IDE), which encompasses the integrated tool suite, for Reconfigurable FPGA Signal Processing is shown in Figure 2. It consists of six major processes and three supporting processes. The major processes are: Design Capture, Logic Synthesis & Technology mapping, Macro Function-based MCM-Level Partitioning, Macro Function-based Floorplanning, Routing, and Design Verification and System Integration. Three supporting processes are Macro Function library development, XNF File Merge, and Design Rule Checking (DRC) and Logic trimming. Now we begin to describe each major process. The supporting processes will be discussed when appropriate. Note that, in our Integrated Development Environment, VHDL and VITAL are utilized heavily for design and verification of each process.

#### 3.1 Design Capture

In this process, the concepts of a design or a system are captured in a simulated form, for design verification. The designer can utilize two primary design entry methods: Schematic Capture and VHDL description. Each design capture method utilizes a highly-optimized and high performance Macro Function library to assist the designer. The designs are captured in technology independent fashion and functionality is verified through VHDL simulation.

#### 3.2 Macro Library Development

Macro Functions are synchronous logic functions for Xilinx 4000 family. They have been optimized for the maximum possible clock frequency, while keeping resource demands to a minimum. Each Configurable Logic Block (CLB) of a macro has been specifically programmed, although the Macro Function's internal routing has not been defined. However, the CLBs have been placed so as to minimize the internal routing. Therefore, a design using these Macro Functions achieve optimum clock performance. As shown in Figure 2, each Macro Function is represented in three different forms: VHDL, Xilinx Netlist Format (XNF), and Logic Cell Array

(LCA). The VHDL model is necessary to support functional simulation. The XNF file is used to facilitate information exchange among design processes. The LCA file is needed during the routing process.

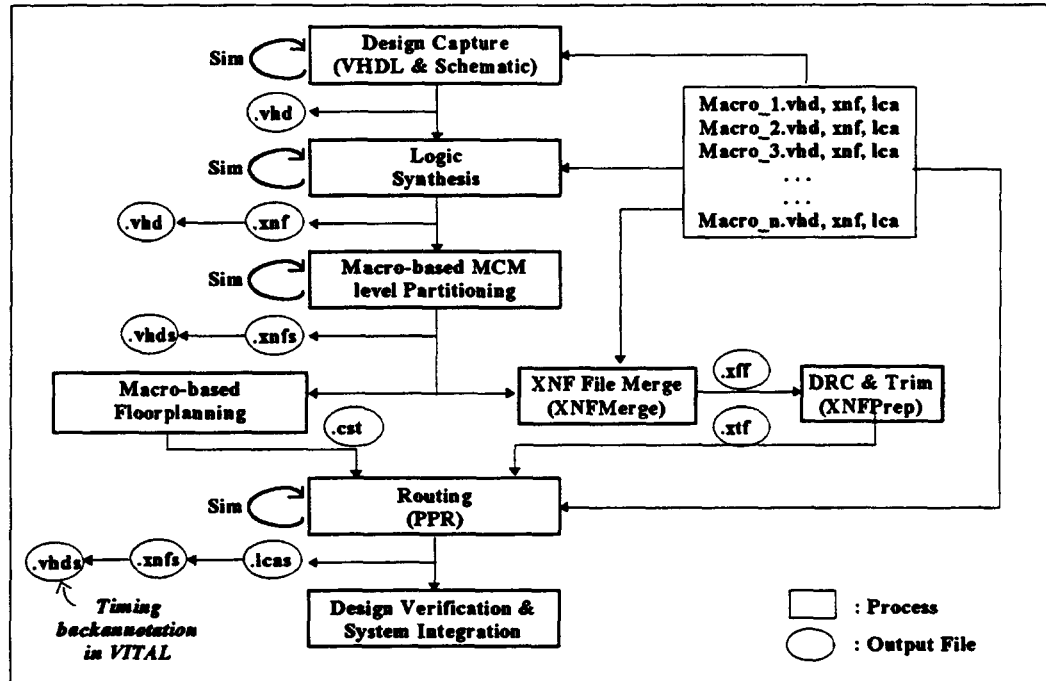


Figure 2. Overview of the Integrated Development Environment

### 3.3 Logic Synthesis and Technology Mapping

In this process, the top level VHDL description is synthesized into a technology specific macro-level XNF file. This XNF file will be input to the Macro Function-based MCM-Level Partitioning process. For verification, the synthesized XNF file is translated into a structural VHDL model for functional simulation. This translation requires a two step process. First, the top level XNF file is merged or flattened using the XNF files available in Macro Function library. Then, the merged XNF is translated into structural VHDL for simulation.

### 3.4 Macro-Based MCM-Level Partitioning

Implementation of a large digital system requires partitioning the design into a collection of interconnected circuits. These individual circuits must then be implemented in a collection of heterogeneous devices without disturbing functionality or performance. Traditionally this has been a manually intensive task. A custom tool was developed to partition these macro-based design between FPGAs [1]. Our Macro Function-based partitioner utilizes spectral methods which results in partitions which form a natural cluster. The partitioner generates multiple XNF files for our reconfigurable board. Using a board architecture configuration file, the partitioner can handle virtually any FPGA based board architecture.

For verification, the XNF files are converted into corresponding VHDL files as discussed in section 3.3. A board level VHDL model is constructed for functional simulation. Due to the reprogrammable nature of the FPGA, however, creating a static board level model required special attention. This board level VHDL model is discussed in the following section.

### 3.4.1 Board Level VHDL Model

Since all FPGAs on the board are reprogrammable, the task of creating the board level model presented some challenges. For example, one of I/O pins may be programmed as an input pin for one specific design. However, the same I/O pin may be programmed as an output pin for different design. The board level model should be able to handle this programmable nature of the FPGA. One solution to this problem is to construct a customized board model specific to each design. However, this approach is a very manually intensive task and error prone. Another approach would be to use the VHDL linkage port. Using the linkage port, we can create a generic and static board level model that can work with various designs. The definition of the linkage port is described in detail in the IEEE Standard VHDL Language Reference Manual (IEEE Std 1076-1993) [2]. The following illustrates the usage of the linkage port for the board level model.

```

entity DRASTIC is -- Entity declaration
port(
  Input_port: in std_logic_vector(15 downto 0);
  output_port: out std_logic_vector(15 downto 0);
  .....
  TMS: in std_logic;
  TCLK: in std_logic;
  TDI: in std_logic; -- into mcm1fpga1
  TDO: out std_logic; -- out of mcm4fpga3
);
end DRASTIC;

Architecture structure of DRASTIC is
component mcm1fpga1
port(
  m1f1p1 :linkage std_logic;
  m1f1p2 :linkage std_logic;
  m1f1p3 :linkage std_logic;
  .....
  m1f1p304 :linkage std_logic);
end component;
component mcm1fpga2
port(
  m1f1p1 :linkage std_logic;
  .....
  m1f1p304 :linkage std_logic);
end component;
.....
.....
component mcm4fpga3
port(
  m1f1p1 :linkage std_logic;
  m1f1p2 :linkage std_logic;
  .....
  m1f1p304 :linkage std_logic);
end component;

-- Configuration Statements
for f1: mcm1fpga1 use entity
work.mcm1fpga1(topdown);
for f2: mcm1fpga2 use entity
work.mcm1fpga2(topdown);
.....
for f12: mcm4fpga3 use entity
work.mcm4fpga3(topdown);
signal sig_con1, sig_con2, VCC, GND: std_logic;
begin
f1: mcm1fpga1 port map (
  m1f1p1 => sig_con1 ,
  m1f1p2 => sig_con2,
  m1f1p3 => M1_A_M3_H(31),
  .....
  m1f1304 => GND);
f2: mcm1fpga2 port map (.....);
.....
f12: mcm4fpga3 port map (.....);
end structure;

```

In this example, all components are declared to have formal ports of mode *linkage*. According to VHDL Language Reference Manual, the formal port of mode *linkage* can be associated to an actual port of any mode. The mode *linkage* is a key construct that enable us to create a static board level model. Now a static board level model can handle the dynamic reprogrammable nature of the FPGA.

### **3.5 Macro-Based Floorplanning**

Floorplanning is the process of arranging the circuit components on a layout surface. The input to the floorplanning process is a set of XNF files generated from the partitioner. In this process, the partitioned XNF files are individually floorplanned for the Xilinx 4025E. Macro-based, timing driven placement achieves high CLB utilization while meeting the performance requirements [3,4]. It generates a constraint file (.cst) for each XNF file to drive the routing process. The floorplanner also considers the routability during the placement process.

### **3.6 XNF File Merge and DRC & Logic Trim**

These are supporting processes that: prepare a flattened file (.XFF), perform design rule checks, and generate a trimmed file (.XTF) for each partitioned XNF file. The XTF file along with its corresponding constraint file are input to the routing process.

### **3.7 Routing**

The placed logic elements and macro blocks are interconnected in this process. For each of the FPGAs in a design, the routing process is guided by: a constraint file, a trimmed file, and a set of Macro Function files (in LCA format). After the routing, the timing of each design is back-annotated into a new LCA file. Then, the LCA file is translated into a XNF file which in turn is translated into a VHDL file which accommodates timing information in VITAL compliant format. VHDL simulation is carried out for functional and timing verification. The board model described in Section 3.4.1 is used for the board level VHDL simulation.

### **3.8 Design Verification & System Integration**

In this process, the routed design is down-loaded to the reconfigurable board. After configuration of the board, the design is operated in real-time, or near real-time, in an array of Xilinx XC4025E FPGAs for final test and verification.

## **4. FPGA Reconfigurable Array Architecture**

The architecture of the DRASTIC Reconfigurable FPGA Array Board was developed with the following guidelines and requirements in mind:

- Selection of a universal FPGA array architecture which efficiently implements the distributed processing element (PE) and PE array architectures typically needed by digital signal processing applications.
- Design of a real-time direct emulation hardware platform, which duplicates a target universal architecture, and allows implementation of a complete design with operation at full clock speed.
- Design of a hardware board architecture which provides design verification, testing and system integration of implemented applications, totally within the Integrated Reconfigurable Environment.
- Design of a reconfigurable FPGA board which can be used as an end platform for the processing of data on implemented algorithms.

Recently proposed and built FPGA board architectures have typically consisted of an array of ring connected or crossbar connected processing elements (PEs), composed of globally reconfigurable FPGAs. Two of the more notable programs are SPLASH<sup>2</sup> and CHAMP<sup>3</sup>. Most of these architectures are intended to support standard parallel and pipelined processing applications, which include single-instruction-multiple-data (SIMD), single-dimension pipelined

systolic, and in some cases, higher dimensional systolic distributed processor models. In particular, the results from the CHAMP (Configurable Hardware Algorithm Mappable Preprocessor) Program, conducted by Wright Laboratories, AASH, provided many practical examples of the application of a reconfigurable processor architecture, based on the Xilinx FPGA device technology. CHAMP was designed to meet the needs of reprogrammable and evolving avionics applications.

This initial work formed the basis for the development, of the DRASTIC optimized FPGA-based PE array and board architecture, which utilized the latest high density Xilinx devices. The design tradeoffs and the performance issues that led to the choice of the PE array and board architecture are discussed in more detail in the following paragraphs.

#### 4.1 The CHAMP II MCM Configuration

In consideration of all the requirements for FPGA-implemented PEs it was determined that the CHAMP 2 MCM [5] represented an optimum design for use in the DRASTIC Reconfigurable FPGA Array Board. The effective use of the pin-outs from the 3 FPGAs, to implement a maximum number of bi-directional data ports, coupled with the logic capacity of three XC4025E devices (3072 CLBs), provides a suitable architecture for the PE element of the DRASTIC FPGA board. The core PE elements are easily extended into the board architecture needed for the DRASTIC platform.

The CHAMP 2 MCM block diagram is shown in Figure 3. Two of the XC4025Es are interconnected through a set of 128 bi-directional data lines, which implements a pipeline configuration through a pair of FPGAs (#2 and #3). Each of the two processor PEs has two 32-bit busses, providing external interconnection to other PEs, and a third 32-bit bus for interconnection to FPGA #1, which is the memory controller. FPGA #1 provides independent address control and data paths to the two segments of local memory. FPGA #1 can also serve as additional processing resources, with data interchange between FPGAs #2 and #3 via the above mentioned 32-bit internal busses. As shown in Figure 3, the external busses of the MCM are labeled A through H, where the last two, G and H, provide external access to the data paths of the two segments of local memory.

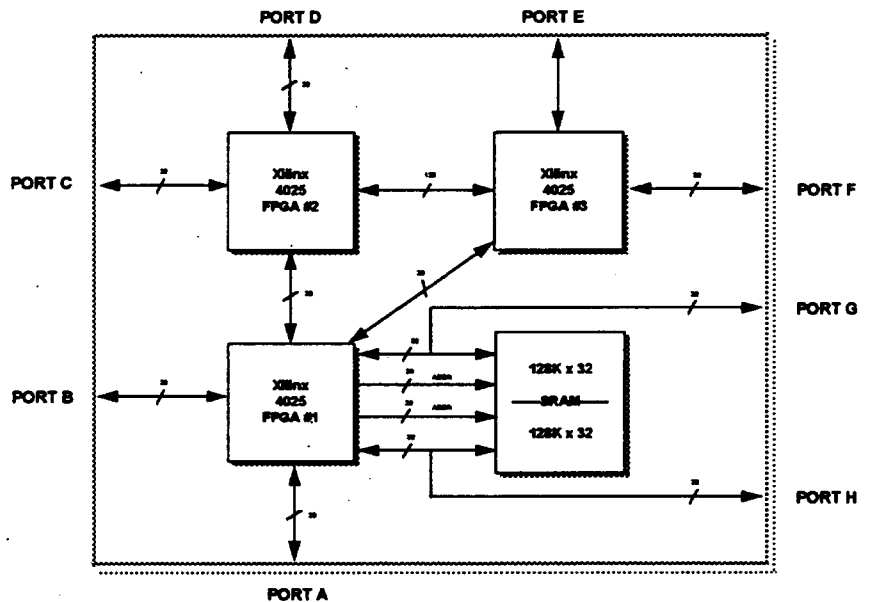


Figure 3. CHAMP 2 MCM, Architecture and Data Paths

#### 4.2 The Reconfigurable FPGA Board Architecture

The DRASTIC board architecture is designed to provide both the PE configurability and the necessary system data paths to support the requirements of an integrated reconfigurable platform. The interconnection of the core MCM PE to form a complete reconfigurable FPGA platform is shown in Figure 4. To support pipelined and ring interconnected architectures, two independent sets of external I/O ports are provided. These ports, consisting of two 32-bit input

busses and two 32-bit output busses, divide the board into two independent channels consisting of MCMs 1 and 2 and MCMs 3 and 4. This provides the board with the capability to accept external real-time data and to output processed data to an external device. These data paths allow integration of the processor array into existing systems or into systems under development. Internal data paths to the host computer PCI Bus also exist (not shown in Figure 4), to allow design verification on the host platform. Through the high-speed PCI interface, the capability exists to send stimulus data to the FPGA array and to capture the array's output data. Further details of the DRASTIC Reconfigurable FPGA Array Board implementation are described in the next section.

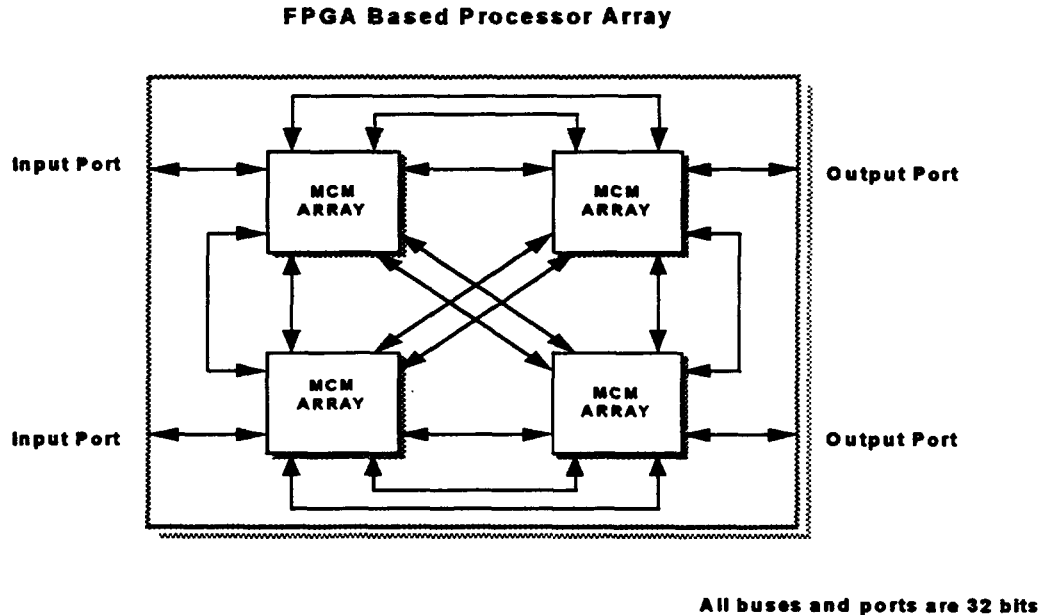


Figure 4. DRASTIC FPGA Based Processor Array

## 5. DRASTIC Reconfigurable FPGA Array Board Implementation

The FPGA-based reprogrammable board was designed specifically to reside in the DRASTIC Integrated Reprogrammable System. The board performs three basic functions: (1) design verification of implemented applications, (2) developmental verification of the DRASTIC tool suite functionality, and (3) implementation of a versatile FPGA-based hardware platform for the processing of data with implemented algorithms.

The DRASTIC FPGA processor array, which was modeled after the optimal PE and memory configuration concept of the CHAMP II MCM, provides additional functions to support the design verification process. These additional features include a programmable clock generation and distribution section, and a buffered PCI Bus interface, which allows host memory access to/from the I/O ports of the FPGA array. The architecture of the complete DRASTIC reprogrammable board is shown in Figure 5.

### 5.1 FPGA-Based Processor Array

The FPGA based processor array is the heart of the reconfigurable board. The FPGA array emulates an array of MCMs which contain Xilinx XC4025E FPGAs and SRAM local memory. Data enters the array via two 32-bit data busses that have two possible sources: (1) data input from an external source or (2) data transferred from the host computer memory, via the PCI interface. Data leaving the processor array can be simultaneously output to an external device

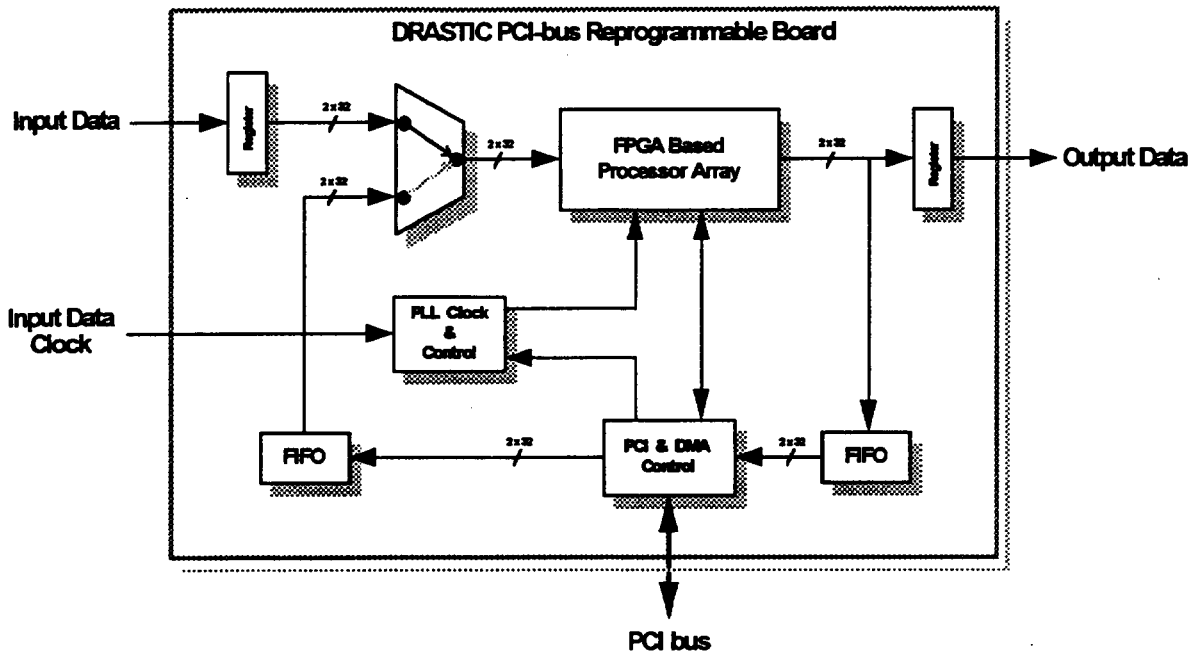


Figure 5. PCI Local Bus Reconfigurable FPGA Board

interface and read into host memory via the PCI Bus. Stimulus data from the computer will consist of user supplied data files residing on hard disk or in the host computers main memory. Likewise, data captured from the FPGA array will be stored in files on hard disk or in the host computers main memory. Both the stimulus data and output data are buffered through FIFO memories. The PCI controller section sets up DMA transfers and coordinates data flow.

External data I/O interface requirements will vary greatly in test environments, or in design verification environments that connect to other system components. In practice, the actual line receiver and output driver circuits required for some of these applications can reside on a dedicated expansion board in the computer. A standard ISA-Bus board can be used for this I/O circuitry, when required. The board can be general purpose board, such as a HSVB (High Speed Video Bus) I/O, or a more specialized module such as an A/D (Analog to Digital) or DAC (Digital to Analog) conversion board.

## 6. Conclusions

The DRASTIC Integrated Reconfigurable System design was evolved from a set of requirements for an integrated, universal FPGA development system, which included merging real-time hardware resources with CAD tools, to provide a much more capable development platform than those commercially available. The advanced FPGA board architecture, developed both for the DRASTIC Integrated System as well as for an end use target platform, represented the results of an extensive FPGA processing element study. The resulting board architecture incorporates the latest in FPGA technology, coupled with flexible ancillary system functions to facilitate a high-speed interface to the host platform, real-time system interfacing, and execution of all implemented FPGA designs at full clock speed in the actual target architecture.

The common and essential link in the DRASTIC Integrated System is use of VHDL-based EDA tools throughout, which allows implementation of an end to end testing, verification, and system integration concept that makes possible the correlation of results between all process steps, and problem solving at all intermediate levels of the design process.

### **Acknowledgment**

The authors wish to acknowledge Ms. Kerry Hill for her continuing support and technical assistance. This work was supported in part by Wright Laboratory AASH under U.S. Air Force Contract No. F33615-95-C-1715.

### **References**

- [1] Vijayanand Sankarasubramanian and Dinesh Bhatia. Multiway Partitioner for High Performance FPGA based Board Architectures. IEEE Conference on Computer Design: VLSI in Computers, (ICCD-96), October 1996.
- [2] IEEE Standard VHDL Language Reference Manual (IEEE Std 1076-1993), 1993.
- [3] Jianzhong Shi and Dinesh Bhatia. Macro Block Based FPGA Floorplanning. Technical Report, University of Cincinnati, TR 189/02/96/ECECS, 1996.
- [4] Vijayanand Sankar, Jianzhong Shi and Dinesh Bhatia. File Format of Architecture File for Architectural Constraint Based Partitioning.. Technical Report, Department of ECECS, University of Cincinnati, Design Automation Laboratory Internal Report, 1996.
- [5] Brian Box and John Nieznanski. Common Processor Element Packaging for CHAMP. In IEEE Symposium on FPGAs for Custom Computing Machines, pages 39-44, IEEE, 1995.

### **Endnotes**

1. This research was supported by Wright Laboratory AASH, U.S. Air Force Contract No. F33615-95-C-1715, the DRASTIC program, Dynamically Reprogrammable Avionics System Technology Implementation Concept.
2. SPLASH 2 program, performed by the Supercomputer Research Center, Bowie, Maryland.
3. The CHAMP program, performed by Sanders, A Lockheed Martin Company, under contract from the U. S. Air Force Wright Laboratories, AASH.