The XC6200DS Development System

Stuart Nisbet[1] and Steven A. Guccione[2]

[1] Xilinx Development Corporation
52 Mortonhall Gate
Edinburgh EH16 6TJ (Scotland)
Stuart.Nisbet@xilinx.com
[2] Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124 (USA)
Steven.Guccione@xilinx.com

Abstract. The *XC6200DS* is a development system for reconfigurable logic design. The hardware features an *XC6216* reconfigurable logic device, a 33 MHz PCI bus interface and up to 2 MB of on-board SRAM. Support software including interface libraries in both *C++* and *Java*, as well as the *WebScope* graphical debug interface and sample applications are also discussed.

1 Introduction

The *XC6200DS* [12] development system is a complete development platform, including hardware and software for producing applications based around the *XC6200* [11] reconfigurable logic device. This system features a PCI bus board containing an *XC6216* reconfigurable logic device and up to 2 MB of SRAM. In addition, software support in the form of the *XACT/6000* placement and routing tool, as well as run-time support libraries in both *C++* and *Java* are available.

2 The System Hardware

Figure 1 shows a diagram of the *XC6200DS* hardware. The board features a 33 MHz *PCI* host interface bus [9] and up to 2 MB of SRAM. In addition, a PCI mezzanine connector is supplied. This permits custom interfaces to the *XC6200DS* hardware to be developed within a high performance, standard framework.

Other support hardware is provided on the board. This includes a programmable clock generator and a current sensing meter. Figure 1 diagrams the *XC6200DS* hardware.

2.1 The XC6200 Reconfigurable Logic Device

The *XC6216* reconfigurable logic device [11] is the first member of Xilinx's XC6200 FPGA family. This device features a microprocessor compatible interface bus for configuration and data access to the device. In addition, all logic,

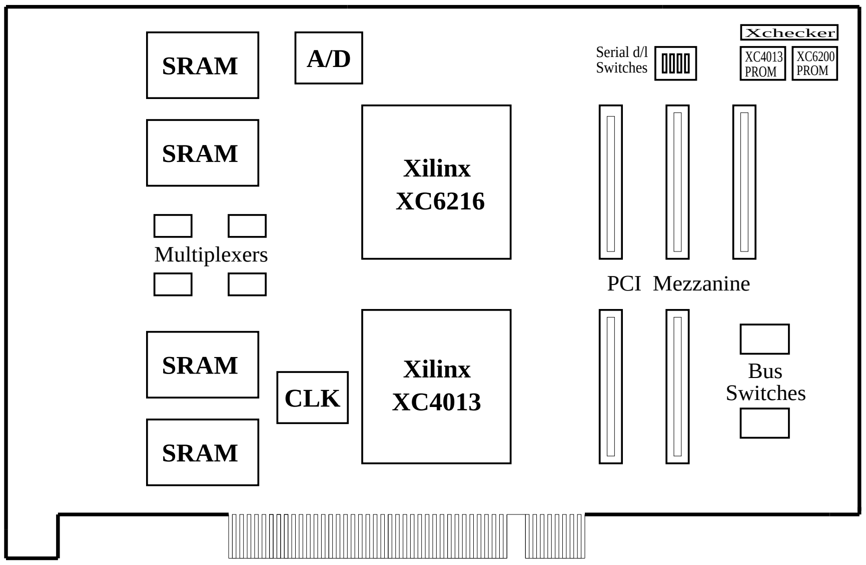

Fig. 1. The XC6200DS hardware.

routing and I/O on the *XC6200* family are dynamically reprogrammable. As an added feature, all routing in the *XC6200* family is uni-directional. This provides fast interconnect and avoids the possibility of internal logic conflicts, which are capable of consuming large amounts of power, as well as potentially damaging the device itself. This makes the *XC6200* family ideal for coprocessing applications, particularly those making use of dynamic reconfiguration.

Currently, the *XC6200DS* system uses a *XC6216* device in a 240 pin package. This device has an array of 64 x 64 logic cells. Since this device has one register per cell, this provides as many as 4096 bits of register. In addition, the gate density is approximately 16,000 to 24,000 gates [11].

The *XC6200DS* is designed to support newer, denser devices currently under development. As denser devices become available, they may be used in place of the *XC6216*.

Several recent papers have been published describing the *XC6200* device and its applications [5] [4] [2] architecture [3] [10] and tools [1] [6] [7] [8]. The reader is referred to these publications, as well as the *Xilinx XC6200 FPGA Family* datasheet [11] for more information on the *XC6200* family.

2.2 The PCI Interface

The PCI interface is implemented using a Xilinx *XC4013E* FPGA and Xilinx's *LogiCore (tm)* PCI bus interface macrocell. Rather than use an ASIC to provide the PCI bus interface, the *LogiCore* approach provided added flexibility.

In the current implementation, only approximately 40% of the device is used to support the PCI interface. This leaves the remainder available for other glue logic and interface tasks.

2.3 The Memory Interface

The *XC6200DS* contains 2 banks of RAM, each containing as much as 1 MB. Each bank is organized as 1M 16-bit words of data. Multiplexers and bus switches on the *XC6200DS* permit the data and address paths to the RAM to be dynamically reconfigured. This provides the different configurations suitable for different processing requirements. Figure 2 shows the 4 major modes of memory operation.

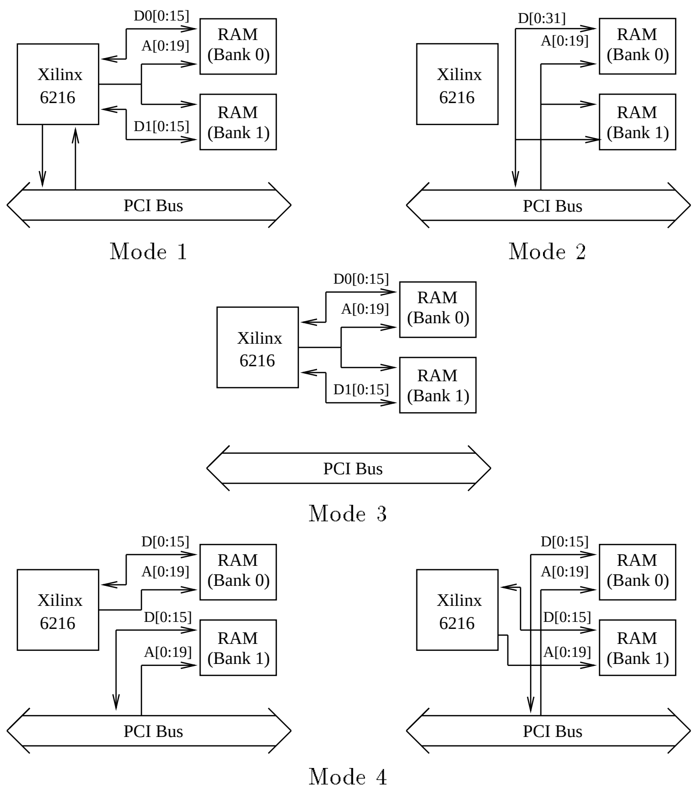

Fig. 2. The memory access paths.

Mode 1 sends the same 20 bit address to both RAM banks, and provides a data path from the RAM to the XC6200. This permits RAM to be accessed as a single large bank. RAM may be accessed as 8-, 16- or 32-bit words by the

XC6200. In addition, the PCI bus is free to communicate in parallel with the *XC6200* while it accesses the RAM. This mode is useful for circuits which wish to process data to/from the PCI bus from/to the RAM, with the *XC6200* in this data path, performing processing.

Mode 2 again groups the 2 banks of 16-bit RAM to appear as a single bank or RAM. Unlike *Mode 1*, the RAM is accessed via the PCI bus. This allows fast loading and unloading of the on board RAM directly from the host processor. In this mode, the RAM may be accessed as 16- or 32-bit words.

Mode 3 is essentially the same as *Mode 1*, but without the PCI bus communicating with the *XC6200*. This permits the *XC6200DS* to run independent of any external bus activity. This mode is useful for using the *XC6200DS* to develop embedded applications which will not use the host processor.

Finally, *Mode 4* permits one bank of RAM to be accessed by the *XC6200* and the other by the PCI bus. The bank attached to the *XC6200* may be switched with the bank attached to the PCI bus. This mode permits the *XC6200* device to perform operations in parallel on one bank of RAM while the PCI bus loads or unloads the other.

2.4 Other Features

In addition to the PCI interface, the RAM and the *XC6200* device, the *XC6200DS* board has additional support hardware. An analogue to digital converter is connected to a current sensing resistor on the *XC6200* power supply. This permits real-time monitoring of device current. Because of the unidirectional routing on the *XC6200* device, excessive current due to internal bus conflicts is not a problem. The current meter is provided simply for those concerned with power consumption, not as a safeguard feature.

A programmable clock generator is also supplied. The clock generator permits circuits in the *XC6200* to be clocked over a wide range of frequencies. This permits circuits in the *XC6200* to be run at their maximum speed without requiring the user to produce clock divider circuits. The *XC6200* may also be clocked in single or multiple clock mode. This mode is useful when debugging circuit designs.

A standard PCI mezzanine interface permits the user to attach suplementary designs to the *XC6200DS*. All *XC6200* I/O pins are attached to this mezzanine, giving full flexible access to the reconfigurable hardware. This mezzanine connector permits daughtercards to perform functions such as video or real-time control interfacing.

Finally, the *XC6200DS* provides support for serial downloading of data. Support for serial PROMs for both the XC6200 and XC4013 are on the *XC6200DS* board. In addition, the *XC6200DS* contains a serial *Xchecker* cable connection. This supports the development of embedded designs intended to run independent of the host processor.

3 The Support Software

3.1 The XACTstep Series 6000 Software

XACTstep Series 6000 is a graphical tool for 6200 Family designs. This system is a back-end tool with *EDIF* as its primary input. Because of the use of standard *EDIF* for design input, design data can be generated with any existing CAD tool which produces EDIF. Existing schematic capture and hardware design language tools, when combined with libraries for the *XC6200* can be used with the *The XACTstep Series 6000 software.*

The *XACTstep Series 6000* editor preserves the hierarchy of the input design. This hierarchy information is used to support both top down design through floorplanning and bottom up design through either manual or automatic techniques. In addition, fully automatic place and route is supported. The graphical editor gives full access to all resources on the 6200 Family architecture.

While the The XACTstep Series 6000 software is the primary means of producing design data for the *XC6200DS*, detailed description of the software is beyond the scope of this paper.

3.2 Run-time Support

The *XC6200DS* system comes with run-time support software for both *C/C++* and *Java* programming languages. The basic support is in the form of a class name *Pci6200.* This class contains the interface to the *XC6200DS* hardware. Figure 3 shows a diagram of the *C/C++* interface library.

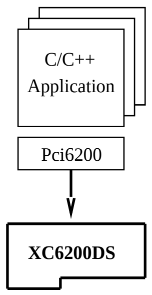

Fig. 3. The C/C++ support software.

The *Pci6200* class contains all functions necessary to read and write the *XC6200* device, the on-board RAM and all support devices, such as the clock generator and current meter.

The *Java* interface class contains identical functionality to the *C/C++* class. In fact, the *java* class was derived directly from the *C/C++* version of *Pci6200.* Figure 4 shows the *Java* interface.

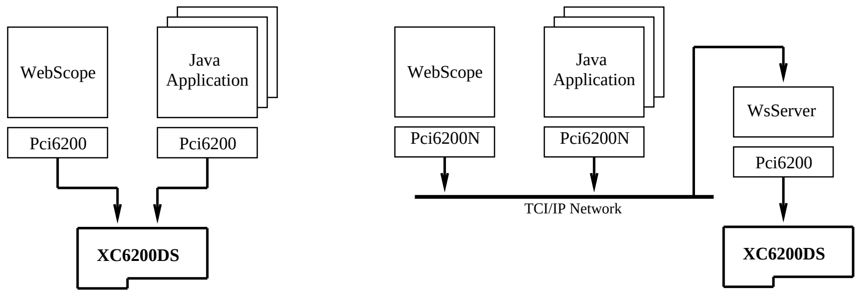

Fig. 4. The Java support software.

While the *Java* support software is based on the original *C/C++* code, there are some differences. First, the *Java* code defines two implementations of the interface, one for direct connection to the hardware, the other for remote access.

The remote access class takes advantage of the sophisticated network support in *Java* and allows remote execution of any function in the *Pci6200* class. While this is usually substantially slower than direct access over the PCI bus, it permits hardware to be easily shared in a networked environment.

Finally, the *Java* support software includes *WebScope* a *Java*-based debug tool for the *XC6200*.

3.3 The WebScope Debug Tool

WebScope is a portable graphical debug interface for the *XC6200* device implemented in *Java*. This tool interacts with the *XC6200* device to aid in the design and debug of circuits.

WebScope provides a point and click interface which permits the logic value of each cell in the *XC6200* to be probed. In addition, all control registers in the *XC6200* are graphically displayed.

In addition to displaying the state of the *XC6200* device, *WebScope* also provides symbolic access to data in the *XC6200*. Selected cells in columns of the *XC6200* may be read back from the microprocessor bus. These groups of cells may function as "variables" such as those in a software program. *WebScope* permits such variables to be defined for a given design. These variables are then displayed either textually or graphically by *WebScope*.

Finally, *WebScope* provide all basic control functions for the *XC6200* device, such as *reset* and *clock*ing. As with the *XACTstep Series 6000* software, details of *WebScope* are beyond the scope of this paper.

4 Conclusions

The *XC6200DS* provides a complete development platform for reconfigurable logic development. Support for activities such as coprocessing applications, embedded applications and research into reconfigurable computing are all possible uses for this system.

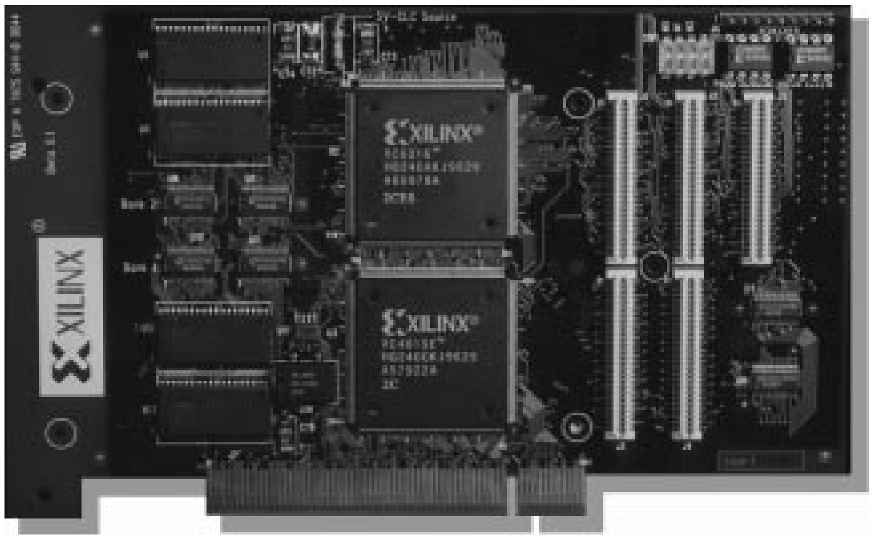

Fig. 5. A photograph of the XC6200DS hardware.

Application development both inside and outside of Xilinx is continuing. As of the writing of this paper, demonstration applications including 2D image matching, cellular automata and image processing have been developed on the *XC6200DS*.

In addition, Xilinx has made the design information for the *XC6200DS* an open standard. Several third party vendors are already producing hardware which is compatible with the *XC6200DS* specification.

Acknowledgements

Thanks to Tom Kean and Nabeel Shirazi for early work on the *XC6200DS* system. Thanks also to the engineering staff of the Xilinx Development Corporation for their efforts in tool and application development for the XC6200DS.

References

1. Gordon Brebner. A virtual hardware operating system for the Xilinx XC6200. In Reiner W. Hartenstein and Manfred Glesner, editors, *Field-Programmable Logic:*

Smart Applications, New Paradigms and Compilers, pages 327–336, 1996. Proceedings of the 6th International Workshop on Field-Programmable Logic and Applications, FPL 96. Lecture Notes in Computer Science 1142.

2. Gordon Brebner and John Gray. Use of reconfigurability in variable-length code detection at video rates. In Will Moore and Wayne Luk, editors, *Field-Programmable Logic and Applications*, pages 429–438, 1996. Proceedings of the 5th International Workshop on Field-Programmable Logic and Applications, FPL 95. Lecture Notes in Computer Science 972.
3. Stephen Churcher, Tom Kean, and Bill Wilkie. The XC6200 FastMap $^{\mathrm{TM}}$ processor interface. In Will Moore and Wayne Luk, editors, *Field-Programmable Logic and Applications*, pages 36–43, 1996. Proceedings of the 5th International Workshop on Field-Programmable Logic and Applications, FPL 95. Lecture Notes in Computer Science 972.
4. J. P. Heron and R. F. Woods. Architectural strategies for implementing an image processing algorithm on XC6000 FPGA. In Reiner W. Hartenstein and Manfred Glesner, editors, *Field-Programmable Logic: Smart Applications, New Paradigms and Compilers*, pages 317–326, 1996. Proceedings of the 6th International Workshop on Field-Programmable Logic and Applications, FPL 96. Lecture Notes in Computer Science 1142.
5. Tom Kean, Bernie New, and Bob Slous. A constant coefficient multiplier for the XC6200. In Reiner W. Hartenstein and Manfred Glesner, editors, *Field-Programmable Logic: Smart Applications, New Paradigms and Compilers*, pages 230–236, 1996. Proceedings of the 6th International Workshop on Field-Programmable Logic and Applications, FPL 96. Lecture Notes in Computer Science 1142.
6. Stefan H.-M. Ludwig. Design of a coprocessor board using Xilinx's XC6200 FPGA – an experience report. In Reiner W. Hartenstein and Manfred Glesner, editors, *Field-Programmable Logic: Smart Applications, New Paradigms and Compilers*, pages 77–86, 1996. Proceedings of the 6th International Workshop on Field-Programmable Logic and Applications, FPL 96. Lecture Notes in Computer Science 1142.
7. W. Luk, S. Guo, N. Shirazi, and N. Zhuang. A framework for developing parameterized FPGA libraries. In Reiner W. Hartenstein and Manfred Glesner, editors, *Field-Programmable Logic: Smart Applications, New Paradigms and Compilers*, pages 24–33, 1996. Proceedings of the 6th International Workshop on Field-Programmable Logic and Applications, FPL 96. Lecture Notes in Computer Science 1142.
8. Wayne Luk, Nabeel Shirazi, and Peter Y. K. Cheung. Modelling and optimising run-time reconfigurable systems. In Kenneth L. Pocek and Jeffrey Arnold, editors, *IEEE Symposium on FPGAs for Custom Computing Machines*, pages 167–176, Los Alamitos, CA, April 1996. IEEE Computer Society Press.
9. Edward Solari and George Willse. *PCI Hardware and Software*. Annabooks, 11848 Bernardo Plaza Court, Suite 110, San Diego, CA 92128 (USA), 1994.
10. R. Woods, A. Cassidy, and J. Gray. Architectures for field programmable gate arrays: A case study. In Kenneth L. Pocek and Jeffrey Arnold, editors, *IEEE Symposium on FPGAs for Custom Computing Machines*, pages 2–9, Los Alamitos, CA, April 1996. IEEE Computer Society Press.
11. Xilinx, Inc. *The Programmable Logic Data Book*, 1996.
12. Xilinx, Inc. *XC6200 Development System*, 1997.