CHAPTER ONE

Introduction

Virtual Computer Corporation has combined the latest reconfigurable component technology with a number of advanced software tools in one easy to use 'system approach' to digital design. **The H.O.T. Works Development System** contains all the components necessary for configurable computing implementation in one integrated package. The H.O.T. Works Package enhances the development cycle for those involved in hardware/software co-design.

The H.O.T. Works Board utilizes the new XC6200 Reconfigurable Processing Unit (RPU) and the XC4000 FPGA from Xilinx. Inc. The H.O.T. Works PCI Board offers a standard platform with both fine and course grain reconfigurable components. The board includes a standard PCI mezzanine card area as well as various data path options for maximum flexibility on the hardware side. On the software side, the H.O.T. Development System bundles a suite of development tools for design entry, implementation and debugging.
The H.O.T. Works Development System represents a major step in breaking the barriers between hardware and software; offering Hardware-On-Demand™.

CONTENTS

BLANK PAGE

INTRODUCTION

Overview

The H.O.T. Works Development System is one of an entire family of configurable computing products created by Virtual Computer Corporation. We designed the H.O.T. Works Development System with the hardware and software engineer in mind and offers a unique logic design, implementation and rapid prototyping environment.

The H.O.T. Works Development System features both hardware and software components. The integrated board and software tools make the H.O.T. Works Development System an easy to use platform for hardware / software co-design. Co-verification on the H.O.T. Works Development System is possible with the real-time connection between your hardware design and the system environment. The choice of options on both the hardware side and software side of H.O.T. Works makes it ideal for rapid product development. You have dynamic control of the communications between digital, structural logic, timing and data. With the use of a custom or third party mezzanine card, you can add analog components to the system.

With the H.O.T. Works Development System you can:

- Easily implement "what ifs",
- Test your designs with data in a computing environment
- Integrate your design with a PCI Bus
- Develop PCI Mezzanine Cards
- Use Rapid Run-Time Reconfiguration Techniques
- Evaluate the fine grained XC6200 RPU

With the use of a High Level Hardware Description Language (HDL), Hardware Object Technology (H.O.T.) and the PCI plug-in co-processing board, the engineer can begin to use configurable computing techniques for algorithm acceleration, design emulation and rapid prototyping.

- Implement and test designs in real-time using real data
- Simple method for run-time reconfiguration of hardware from executable programs
- Build upon the prototyping features of the H.O.T. Works Development System
- Choose from a number of different development tools included with the system --
  1) Use the new HDL language Lola, by the author of Pascal
  2) convert your VHDL designs with supplied tools or
  3) import your schematic designs into the Xilinx XActStep map, place and route tools,

4) convert your designs into 'C' program language functions with VCC's H.O.T. tools, download, run and debug through simple custom executables

Simply Speaking

Parts of the following first appeared on Virtual Computer Corp.'s Website in February 1995 (http://www.vcc.com). For those who understand hardware, FPGAs and electronic engineering, we ask your patience, as this is written for those unfamiliar with this technology.

Hardware Object Technology

Why is it H.O.T. ?

Current computers are fixed hardware systems based upon microprocessors. As powerful as the microprocessor is, it must handle far more functions then just the application at hand. With each new generation of microprocessors, the application's performance increases only incrementally (≤2X). In many cases the application must be rewritten to achieve this incremental performance enhancement.

Configurable Computing Systems are those computing platforms whose architecture can be modified by the software to suit the application at hand. To obtain maximum through-put, an algorithm must be placed in hardware (i.e., ASIC, DSP, etc.). Dramatic performance gains (>25X) can be obtained through the 'hardwiring' of the algorithm in the reconfigurable portion of the H.O.T. Works Board. The software code being downloaded into the configurable computer is a formatted digital design created for that specific algorithm.

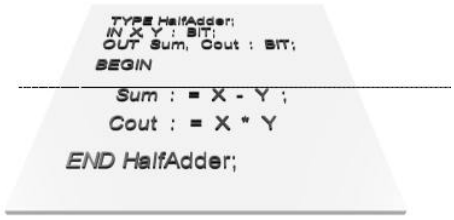


Figure 1 Software Code

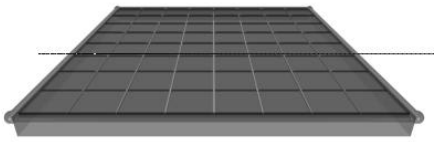


Figure 2 A Configurable Computer

The barriers between hardware and software begin to blur when the software can configure the hardware at run-time. This is the **H.O.T. Crossover**.

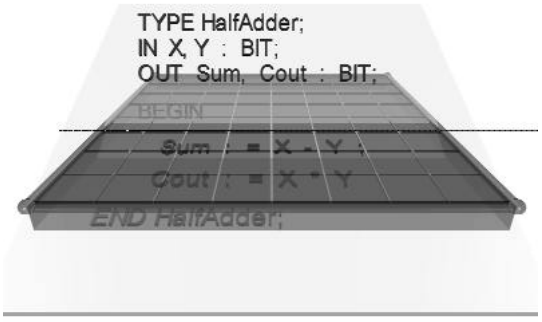


Figure *3* *The H.O.T. Crossover*

This process of altering configurable logic on-the-fly from within an executable program is called **Run-Time Reconfiguration.** The Run-Time Programming of Configurable Computer Systems gives executable programs the power to alter the 'logic' level of hardware to suit its own needs. Hardware-On-Demand™ is possible with Hardware Object Technology and the Virtual Computer H.O.T. Works PCI Board. VCC's Hardware Object Technology enables the designer to use digital designs with standard 'C' language programs. Your digital design is downloaded from within an application program (as a 'C' language function). VCC's unique implementation of reconfigurablity and ease-of-use allows real-time debugging of digital designs.

The Configurable Hardware (**H.O.T. Works PCI Board**) is an extension of your existing host computer. To the host computer it is another processor. Like the microprocessor (MPU), this Reconfigurable Processing Unit (RPU) can perform complex calculations. Unlike the MPU, the RPU can be customized to fit a problem, not handled efficiently by the von Neumann based MPU.

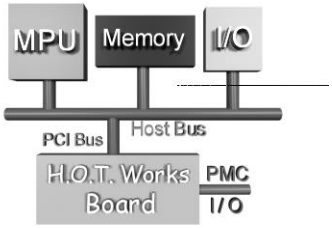


Figure *4* *The Host System*

The Standard Computer is a Fixed Hardware System

The computer has three major parts:

1. The microprocessor Unit (MPU); which is the engine of the computer, handling the management of all computer functions as well as performing the calculations (running the application program).
2. Memory; the place in the computer where the program and data to be used by the MPU is stored.
3. I/O Devices (Hard Disk and other devices such as keyboard, monitor, printer, modem, etc.); These input/output devices are attached to the main computer and handle everything from displaying data to storing data and programs.

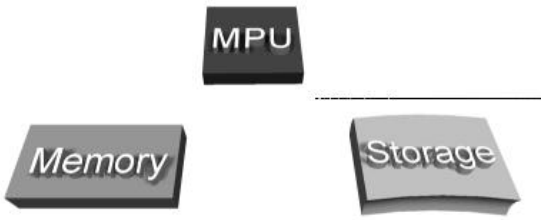
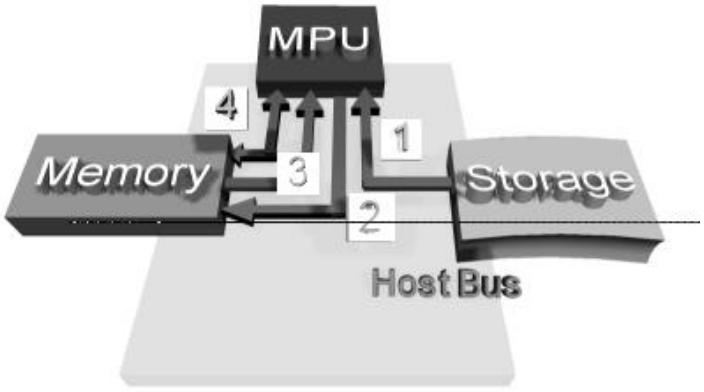


Figure 5 The Standard Computer

The interaction between these components takes place on the computer 'BUS'. The BUS is the electrical wires or highway that allows data and programs to pass from one device to another. The BUS width is the size of the highway; the larger the bus width, the greater amounts of data can pass through at one time (i.e., 8bit, 32bit, 64bit).

The heart beat of a computer is the 'CLOCK RATE'. The Clock Rate is the pulse that pushes the entire computer. This pulse is measured in Megahertz (one MHz equals one million heart beats per second); thehigh the number of MHz, the faster the computer performs. A 100 MHz clock rate is twice as fast as a 50 MHz clock rate. For the actual running of a program, the clock rate on the system's MPU determines how fast system functions and calculations are handled. Figure 6 shows the steps a computer system takes to run an application program.



.

Figure 6 --- The Running of an application

1. The MPU downloads the program and data from storage
2. The MPU transfers to the on board memory (RAM).
3. The MPU reads the program's instructions into the MPU for program execution.
4. The MPU transfers the data, back and forth to the system memory. Depending upon the complexity of the application and the amount of data being processed, this step can be repeated many**thousands**of times.

Within the MPU are a number of standardized functions. Some dedicated functions handle the additions, subtractions, and multiplications called for by the application program. Other functions handle the management of the I/O, while other functions are used only within the MPU to translate the application's program code into segments and instructions needed for the particular MPU to perform calculations.

The inside of the MPU has five internal units that function together to perform all the tasks necessary to run the computer.

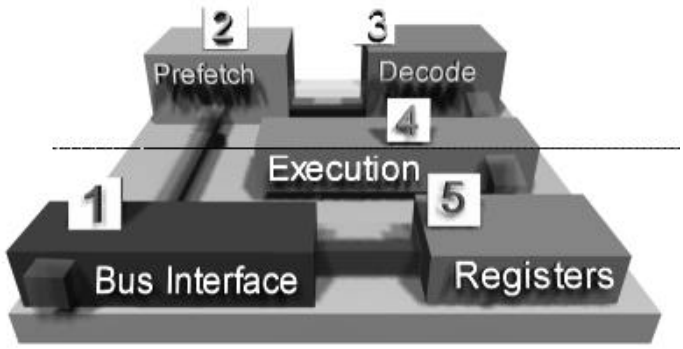


Figure 7 Microprocessor MPU

The major units of an MPU:

1. The Bus Interface Unit (BIU) retrieves the program code or data from the memory (RAM). The BIU converts the code segments for use inside the MPU. The results from the MPU pass through the BIU to the rest of the computer system (e.g., hard disk, monitor, etc.)
2. The Prefetch Unit takes the code and data from the BIU (when it is not busy); it then separates the data from the code and places them in a unique queue for the instruction unit.
3. The Decode Unit reads the program code segments and translates them into the internal MPU language. These simpler (low level) instructions include both what is to be calculated and how those calculation will take place. The translated program code is then placed in it's unique queue awaiting the Execution Unit's request.
4. The Execution Unit retrieves the instructions waiting in the Decode Unit; places the data to be processed into the Registers and begins executing the first instruction in the queue.
5. The data is moved in and out of the Registers as each instruction is executed.
6. After the final instruction of a series has been completed, the data is moved out of the Registers and on to the BIU. The BIU then converts the data back into the format necessary

for the computer system to handle. If you had loaded a program that added two plus two, the answer (four) passes here on the way to the monitor screen or the printer.

The Virtual Computer™ is a Reconfigurable Hardware System

Configurable Computing systems are those computing platforms whose architecture is modified by the software to suit the application at hand. This means that within the application program a software routine has been written to download a digital design (chip design) directly into the Reconfigurable Processing Unit (RPU) of the Reconfigurable Computer. It is like having custom processor chip manufactured for your application program. Most Reconfigurable Computing Systems are plug-in boards made for standard computers such as PCs and workstations. The Reconfigurable board acts as a Co-processor to the main MPU. Calculations normally done within the MPU are now carried out in the RPU instead. The main reasons for using this approach are:

1. Very High Speed Performance Gains; the fastest version of any program calculation is one in which a computer chip has been designed just to perform that specific calculation only,

2. Flexible Hardware; as new ways of solving problems arise, the RPU optimizes the compute power by implementing the calculations directly into a custom computing chip.

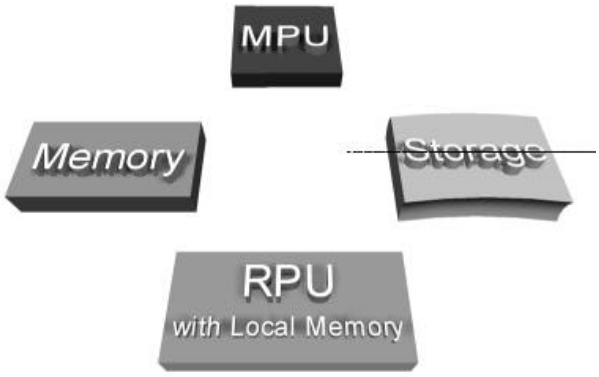


Figure 8 Configurable Computer

The Reconfigurable Computer has four major parts:

1. The Microprocessor Unit (MPU); which is the engine of the computer, handling the management of all computer functions as well as performing the calculations (running the application program).
2. Memory; the place in the computer where the program and data to be used by the MPU is stored.
3. Devices (Hard Disk and other devices such as keyboard, monitor, printer, modem, etc.); These Input/output devices are attached to the main computer and handle everything from displaying data to storing data and programs.
4. The Reconfigurable Processor Unit (RPU) acts as a Co-Processor to the MPU. The RPU allows the software to configure the RPU's internal architecture to perform a given unique calculation. The RPU runs that particular calculation (without the MPU), returns the answer and is ready to be reconfigured by the software with another calculation.

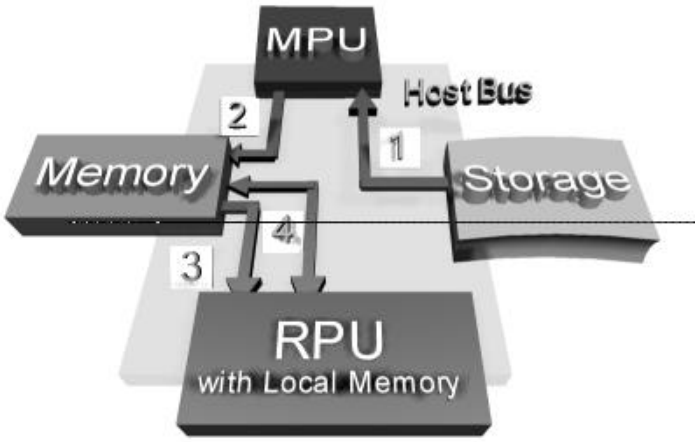


Figure 9 Configurable Computer Running an Application

1. The MPU downloads the program and data
2. The MPU transfers to the on board memory (RAM).
3. The MPU downloads the program's instructions into the RPU for program execution.
4. The RPU loads the data to be processed by the application program back and forth to the system memory. Depending upon the complexity of the application and the amount of data

being processed, this step may be repeated many thousands of times. The calculation takes place inside the RPU.

Within the MPU are a number of standardized functions. The RPU has the ability of receiving numerous unique calculation download from the application program. The data flows into and through the calculation that has configured the RPU. Due to the fact that the RPU need only carry out the specific calculation, no time (heart beats or clock cycles) is spent on general function routines as the MPU does.

In general terms, the inside of the RPU has four internal parts which allow the chip to be reconfigurable. The enabling technology of the RPU is the Field Programmable Gate Array or FPGA. The FPGA is designed to help cut the time and cost of making custom computer chips. The particular FPGA used in Reconfigurable Computing is a SRAM-based FPGA. This class of FPGAs allows one to reconfigure the inside over and over again (millions of times).

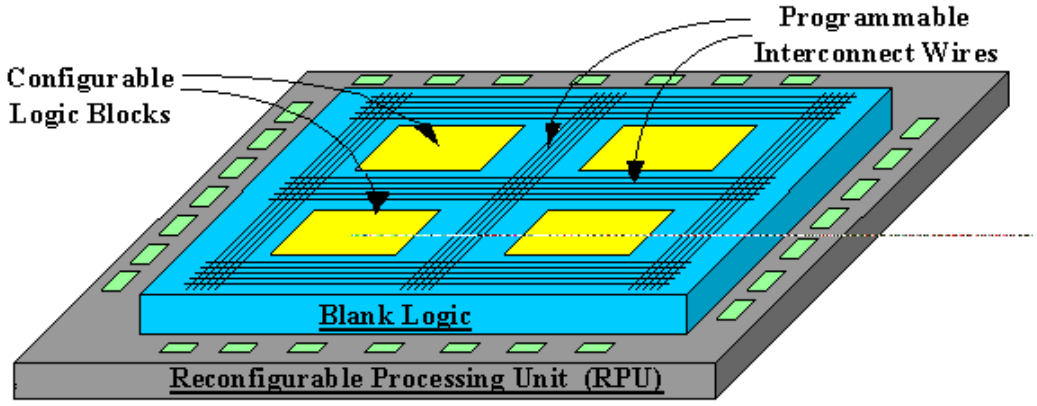


Figure 10 RPU

There are three major parts to RPU:

1. The Configurable Logic Blocks (CLB); Configurable Logic Blocks implement most of the logic in an RPU. The Logic Design within the CLBs connect to the programmable interconnect resources outside the block.

2. The Programmable Interconnect Resources; All internal connections are composed of metal segments or wires with programmable switching points to implement the desired routing

(connections between CLBs & IOs). There are four main types of interconnect, three of them are distinguished by the relative length of their segments: single-length lines, double-length lines and Long-lines.

3. The Input & Output Blocks (IOB); User-configurable input/output blocks provide the interface between outside world and the internal logic. (Green blocks in Figure 11).

Standard Microprocessor Operation / Reconfigurable Processor Operation

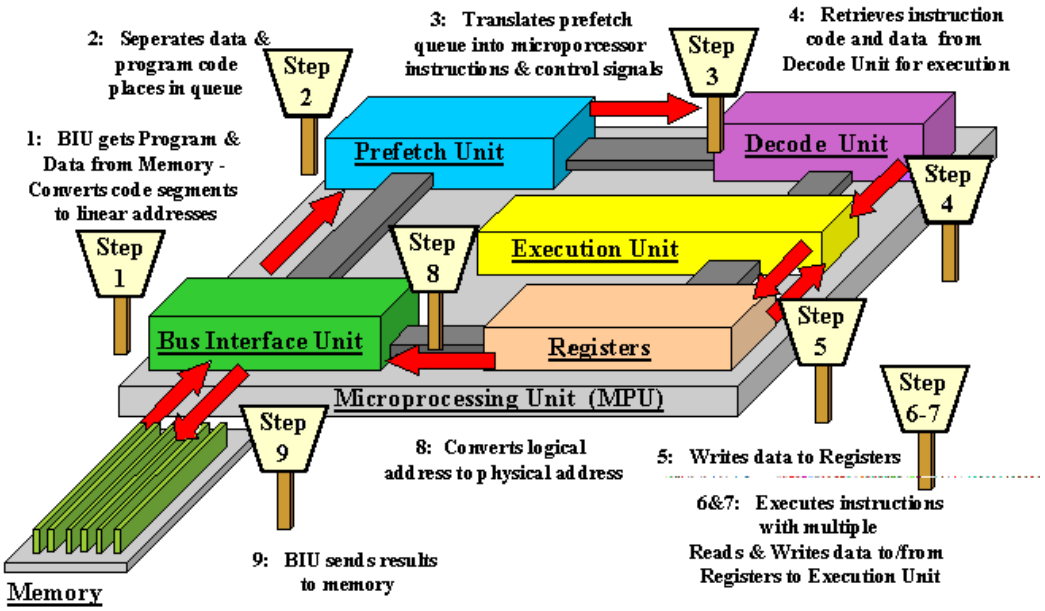


Figure 11 The MPU Function Flow

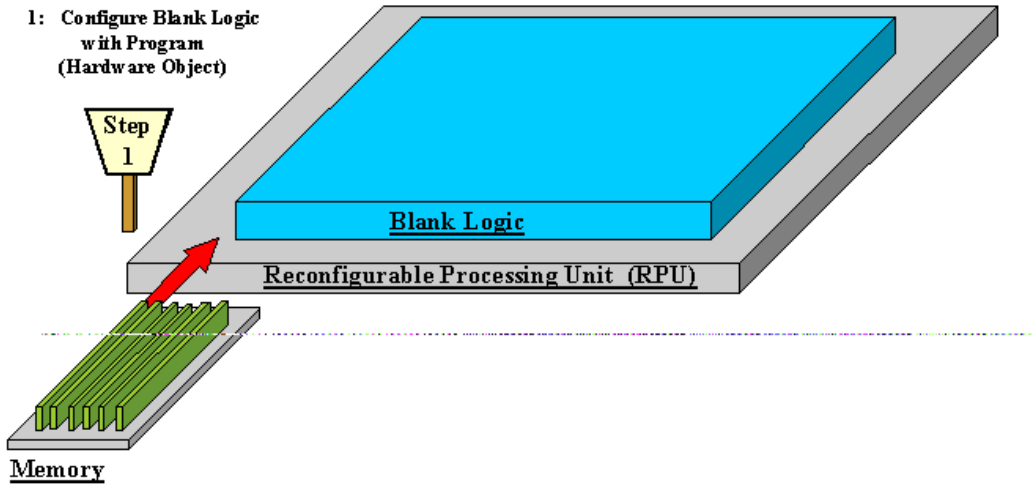


Fig. 12 RPU Function Flow (1)

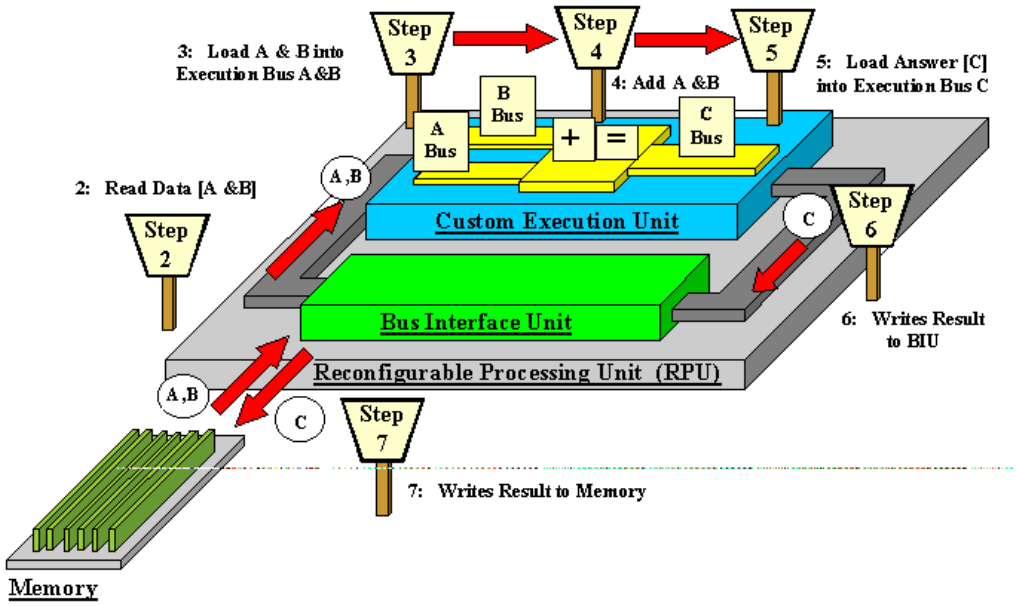
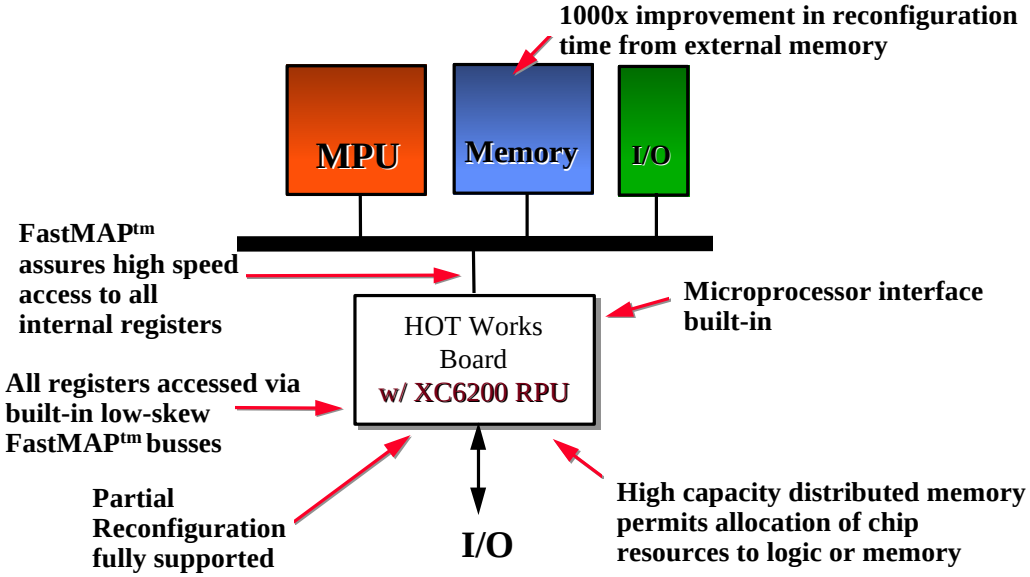


Figure 13 RPU Function Flow (2)

HOT Works Development System features meet Real Time Computing requirements



Definition of Configurable Computer

Configurable computers are those machines that use the reconfigurable aspects of Reconfigurable Processing Units (RPUs) and Field Programmable Gate Arrays (FPGAs) to implement an algorithm. The algorithms are partitioned into a sequence of hardware implementable objects (Hardware Objects). These hardware objects represent the serial behavior of the algorithm and can be executed sequentially. The use of Hardware Objects gives the developer a logic-on-demand capability that is the basis of configurable computing.

Hardware/Software Co-Design System

With the rapidly changing demands for quicker turn-around times and higher performance at a lower cost, a development platform that allows for control and debugging of designs in real-time with real-data can enhance the engineer's success. The problems of integrating the hardware and software components are greater as the logic density and application sizes increase. A Hardware/Software Co-design Development System is becoming essential to success. Unlike co-design systems of the past, these new platforms must address the needs and skills of both the hardware and the software engineers. As the boundaries between hardware and software blur so do the boundaries between the hardware engineer and software engineer.

We have assembled a Hardware/Software Co-design System that answers the above needs. We have combined the latest reconfigurable component technology with a number of advanced software tools in one easy to use 'system approach' to digital design. Through seamless integration of the hardware and software the developer is able to instantly test their designs in real-time with real-data.

For further study on digital design for hardware and programming languages, We highly recommend the following book on Digital Design:

Digital Circuit Design, An Introductory Textbook
by Niklaus Wirth
ISBN 3-540-58577-X Springer-Verlag

H.O.T.Works Development System Components

Overview -- Hardware

The H.O.T. Works PCI Virtual Computer board is a standard single slot, PCI card for any PCI compatible platform. It features the first commercially available Reconfigurable Processing Unit (RPU), the XC6200 by Xilinx Inc. The H.O.T. Works Board includes one XC6200 RPU, one Field Programmable Gate Array device (XC4000 FPGA Family), 2MB Fast SRAM, On-Board Programmable Oscillator (360 KHz to 100MHz), PCI interface, and PCI Mezzanine Card (PMC) Standard interface connectors for the addition of optional daughter board cards. The H.O.T. Works Board is an enhanced version of the Xilinx Reference Design (1997) and complies to PCI industry standards.

Overview -- Software

The H.O.T Works software bundle includes the complete XACT6000 design software by Xilinx Inc. on one CD and a suite of other design entry, editing, implementation, debugging tools, and program examples with source code on the H.O.T. Works CD.

I. **The Lola Programming System** an Object Oriented Hardware Description Language and compiler developed by Niklaus Wirth, author of Pascal and Modula2. This complete design entry, and implementation tool includes: a) Lola, the Logic Language (HDL) design entry tool, b) XC Editor, a graphical layout editor with a circuit checker, a mapper, placer and router, and c) XC Loader, a bit stream generator and loader for the XC6200 RPU.

II. **Xilinx's XACTS*tep*** is a Map, Place and Router for the XC6200 family of RPUs. XACTStep accepts EDIF formatted files from any design capture system generating the configuration data file for the XC6200. XACTStep allows varying degrees of user interaction from fully automatic place and route to detailed user control. It also provides a single point of access to tools such as timing Analysis and Constraint Editing.

III. **Velab**, a VHDL programming tool for conversion of structural VHDL coded designs into the XC6200 RPU EDIF file format.

IV. **WebScope**, a Java-based tool for real-time design emulation (hardware debugger). WebScope enables one to load and test designs over the network or internet. Source code is provided for the quick understanding of programming techniques.

V. **Device Drivers** for Windows 95.

VI. **Design & Program Examples**

H.O.T. Works Development System -- CD Contents

There are many resources on the two CDs provided with the H.O.T. Works Development System. The following outlines the various programs and files included on the two CDs.

XACT6000 CD (ver1.1 for IBM PC or compatibles)

The default installation directory is `C:\Xact6000` .

XACTstep6000 Development Software: `C:\Xact6000\bin\Xact6000.exe`

The Map, Place and Routing Software from Xilinx Inc. for the XC6200.

Related program files in: `C:\Xact6000\data , C:\Xact6000\Vl_libs`

XACTstep6000 Examples: `C:\Xact6000\examples`

Design Examples used in XACT6000 Software

XACTstep6000 Tutorial: `C:\Xact6000\Ralib\tutorial`

Tutorial example (in three steps) for using the XACTstep tools. PLEASE NOTE: An updated version (in four steps) of this tutorial for use with the H.O.T. Works Development System can be found in `C:\Hotworks\Tutorial` (see below).

RAL Class & Library Files: `C:\Xact6000\Ralib`

RAL Class & Library Files for reconfiguration control software.

Document Files (in PDF Format): `C:\Xact6000\Docs`

`An68020.pdf` : Interfacing XC6200 to Microporcessors (MC68020 Example -- Application Note from Xilinx Inc.

`Antms320.pdf` : : Interfacing XC6200 to Microprocessors (TMS320C50 Example -- Application Note from Xilinx Inc.

`cic.pdf` : : High Performance, Low Area Interpolator Design for the XC6200 -- Application Note from Xilinx Inc.

`Dsheet.pdf` : : Data Sheet on the XC6200 Family from Xilinx Inc.

Interactive Demo: This interactive demo can be found either on the CD under demo or if you installed it on your harddisk, at: `C:\Xact6000\demo Xactdemo.exe.` This is an excellent presentation and over view of both the XC6200 Architecture and the XACTstep tools.

H.O.T. Works Development Software Package CD (ver1.1)

The default installation directory is `C:\Hotworks`

Please Note: Many of the programs and examples included on this CD have **`'.ini', '.vpj'`** files and other files with default directories. If you install the H.O.T. Works CD material in any other directory location than **C:\Hotworks**, you will have to alter those files.

Root Default Directory

Directory: C:\Hotworks\

1. Applications

Directory: C:\Hotworks\correlator

A. Correlator Program (see: Tutorials Chapter for an overview of the Correlator Program)

Image correlation is performed by passing a template over an image and determining at each pixel position, if a match has been found. The design described uses a 32x16-pixel match image and a 32x16-pixel mask image to construct the template. The match image is a small image to be located within a larger image. The mask image allows masking out of background regions in the match image. Only unmasked pixels will be correlated. If the number of matching pixels at any point in the image exceed a threshold, a Hit is detected.

The Correlator subdirectory contains the application file and supporting files for the Correlator.

C:\Hotworks\ correlator\code\ correlator.exe - the executable Correlator Program
C:\Hotworks\ correlator\ code\ correlator.cal - the digital design file
C:\Hotworks\ correlator\ code\ correlator.ral - the Run-Time library file
C:\Hotworks\correlator\ code\ correlator.sym- the symbol file
C:\Hotworks\ correlator\ code\ mask0.pbm & mask6.pbm
C:\Hotworks\ correlator\ code\ match0.pbm & match6.pbm
C:\Hotworks\ correlator\ code\ images\ - 3 ppm images

Correlator source code files directories:

C:\Hotworks\ correlator\code
C:\Hotworks\ correlator\viewlogic
C:\Hotworks\ correlator\ xact6000

B. WebScope Program (see: Tutorials Chapter for an overview of the WebScope Program)

The WebScope directory contains a Java application for test design debugging. WebScope provides a graphical user interface that interacts with designs using the Xilinx XC6200 device, in particular, the H.O.T. Works Board. WebScope is implemented in the Java programming language. Its implementation permits WebScope to run on a variety of hosts, from PCs to workstations, using the same small set of files. Additionally, WebScope may be run using local hardware, or remotely using H.O.T. Works Board on another host. Finally, WebScope may be run as either a standalone application using a Java interpreter, or as an applet running from a Web Browser such as Netscape or the Microsoft Internet Explorer.

As a standalone application, all that is needed is a Java interpreter. These are available from a variety of sources. Sun's Java Development Kit (JDK) is included on the H.O.T. Works CD (see below). Once the Java interpreter is installed, WebScope may be run.

A complete description of how to use the WebScope application is contained within the HTML files:

C:\Hotworks\ webscope\webscope.html - HTML WebScope Documentation
C:\Hotworks\ webscope*.class - Webscope.class - the top level class C:\Hotworks\ webscope*.cal - various CAL example designs C:\Hotworks\ webscope*..sym - sym files for the CAL examples C:\Hotworks\webscope\java_source\ *.java - java source code

2. C++ Classes & Documentation

Directory: C:\Hotworks\Api

This directory contains the C++ application development interface files to the device driver. The following classes provide a C++ interface to the XC6200 PCI board (XC6200DS). The classes allow the user to interface to the PCI card from their own C++ code with a few simple function calls. The code uses the XC6200.vxd device driver to interface to the PCI board. The Hardware Object Technology CAL file convertion program is in this directory.

C:\Hotworks\api\pciboard.cpp & pciboard.h
C:\Hotworks\api\ pcicore.cpp & pcicore.h
C:\Hotworks\api\xc6200ds.cpp & xc6200ds.h
C:\Hotworks\api\pciboard.cpp & pciboard.h
C:\Hotworks\api\Ntdioctl.h
C:\Hotworks\Api\cal2h.exe --- Hardware Object Conversion Program
C:\Hotworks\Api\pcilass.html

The complete class description can be found in HTML format **'pciclass.html'** in the C:\Hotworks/docs directory.

3. Device Drivers & Documentation

Directory: C:\Hotworks\drivers

This directory contains the WIN95 drivers for the H.O.T. Works board and contains the following driver files:

C:\Hotworks\drivers \hotworks.inf - the information setup file
C:\Hotworks\drivers \hotwork.vxd- the device driver

An earlier version of the driver is xc6200.inf & xc6200.vxd

4. Java Software, Documentation & Examples

Directory: C:\Hotworks\java

This 1.0.2 release of the Java Developers Kit (JDK) lets you write applets that conform to the 1.0 Java applet API. The related subdirectories and files are used by WebScope.

5. Lola Programming System Software, Documentation & Examples

Directory: C:\Hotworks\Lola\Lola.exe

The Lola Programming System for the XC6200. There are many related program files in Lola\ subdirectories. These files are automatically accessed by the Lola Program.

Lola Documentation & Examples

All documentation, tutorials and examples are on-line.

6. PCI Test Software & Documentation

Directory: C:\Hotworks\pci_test\

Commands to test the PCI board can be entered via the command line of the PCITest window. These core commands interface directly to the XC6200DS functions. The commands can be combined together in Command files allowing the user to create their own command sequence. Currently within PCITest there are a suite of command files allowing core testing of the board functionality of the board.

PCITest Documentation on-line

7. RAL Libraries, Documentation & Examples

RAL Files: C:\Hotworks\Api\ral

Libraries used for Run-Time programming . RAL Class & Library Files for reconfiguration control software.

8. Velab Software, Documentation & Examples

Velab Software: C:\Hotworks\Velab\velab.exe

The VHDL Elaborator from Xilinx Inc. for the XC6200.

Related program files in: C:\Xact6000\multiadd , C:\Xact6000\popcnt, etc.

Velab Documentation: C:\Hotworks\docs\Paramlib.pdf & ..\Velab\macros

This PDF Format file contains the preliminary documentation of libraries for Xilinx 6200 series FPGAs. There are 30 macros, all of them captured in structural VHDL and parametrised by word width. Some of them are also parametrised by pin separation. Placement constraints are specified using the attribute facility in VHDL, and all designs involve only nearest neighbor routing. A short summary is given in the "Library Summary" table.

For each library, we include a short description of its function, its parameters and their default values, and the pin definitions, in addition to the VHDL code. Examples indicating the key attributes for each macro are provided; the attributes include its aspect ratio, the number of 6200 cells used, and the location of the input and output pins. Various circuit diagrams generated from Synopsys and XACTstep6000 tools are also included. A separate document containing only the VHDL entity definitions and the essential information for using the library designs, has also been compiled.

9. XC6200 Design Example Macros

Schematic Files: C:\Hotworks\schematics

There are seven design examples with schematics (Viewlogic) and related files. These files are called up with XACTstep6000. CAL files are included. Each design is located in it's own subdirectory. The various example designs for the XC6200 are contained in the following sub-directories. These are the test designs used within PCITest.

C:\Hotworks\ schematics\32con_rd\ - 6200 IOB test
C:\Hotworks\ schematics\6k_ram\ - 6200 to SRAM interface test
C:\Hotworks\ schematics\clk_step\ - demonstrates clock stepping
C:\Hotworks\ schematics\counter\ - a software clocked counter design

C:\Hotworks\ schematics\filter\ - reconfiguarable image filter
C:\Hotworks\ schematics\ugeq\ - unsigned greater than or equal to example
C:\Hotworks\ schematics\toggle\ - toggles flip-flops for on board current analysis

Design Flow

Designing with the H.O.T. Works Development System requires knowledge and understanding of digital design and 'C' or JAVA programming languages. The starting point is the design entry step. The final step creates an executable program containing the digital design (as a Hardware Object) on the host system using the H.O.T. Works Board as the configurable co-processor.

The development process is defined in five steps. (see diagram below)

I. Design Entry or Import

II. Map, Place, Route Design Editing

III. Make Bitstream

IV. Convert to Run-Time Format

V. Run-Time use on the H.O.T. Works Board

I . Design Entry or Import

The first step in reconfigurable H.O.T. Works is entering a design capture of the design creating run-time Hardware with the Development System digital design with a program. The output mapped, placed capture tool is than RPU (Step II.). and routed for the

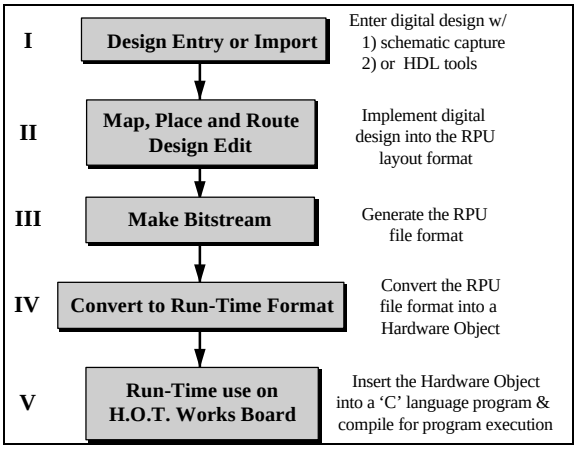


Figure 1 Design Flow

Design capture systems can be classified into two categories:

- Schematic Entry Tools
- Hardware Description Languages (HDL)

The H.O.T. Works Development Software Package contains two different design capture programs. The first is an HDL text editor and compiler called *'Lola'* (see description above) and the other is a VHDL conversion program called *'Velab'*.

You can use any third party design capture system outputing EDIF 2. 0 .0 netlist files and supports XC6200 libraries (e.g. Viewlogic) to enter your design. The EDIF netlist file can be imported directly by *XACT6000* for design implementation (Step II.).

II . Design Implementation -- Map, Place, Route and Edit

Once your design has been entered, the next step is to map, place and route the design onto the XC6200 RPU. There are two software programs included in the H.O.T. Works Development System that implement designs.

The *Lola Programming System's XC Editor* imports the compiled *Lola HDL* design into a graphical layout tool for placement and routing. The output of this process includes the making of the RPU bitstream (Step III.).

The *XACT6000* imports EDIF files. The placement and routing may than be viewed and edited in the *Layout Editor* and *Design Browser*. *XACT6000* includes a sophisticated tool that analyzes timing for completed designs. When your design is finished you can than make the RPU bitstream file (Step III.).

III. Make Bitstream

The bitstream file configures the RPU. The bitstream file has the extension *CAL* and is generated by user's commands in both the *Lola Programming System* and *XCATStep6000*.

IV. Convert Bitstream File into Run Time Program Mode.

Two methods for Run-Time Reconfiguration of the H.O.T. Works Board are supported by the H.O.T. Works Development Package.

With one method, the CAL file (your digital design) is loaded from the harddisk to the H.O.T. Works Board via a program command. This method requires use of C++ support software for the H.O.T. Works Board. These files contain routines to support Low-level board interface, plug and play support, XC6200 Device configuration support, XC6200 Runtime support and XC6200 Debug support.

With the other method your digital design is compiled into the executable program. The downloading of the design to the H.O.T. Works Board occurs at program execution time. This method requires the conversion of the *CAL* file to a Hardware Object . It also requires C++ routines for control of the H.O.T. Works Board. This is VCC's Hardware Object Technology Method of Run Time Reconfiguration.

V. Run-Time use on H.O.T. Works Board

Design Flow for the Three Programming Methods

The following diagrams explain the three programming methods design flow. The differences between the three are in the combination of software tools.

The H.O.T. Works Development System supports three Design Paths for development and execution of designs using the H.O.T. Works Board. These are:

1. Schematic Design Entry to Program Execution.
2. VHDL Design Entry to Program Execution.
3. Lola Design Entry to Program Execution.

Design Flow for Schematic Entry Method

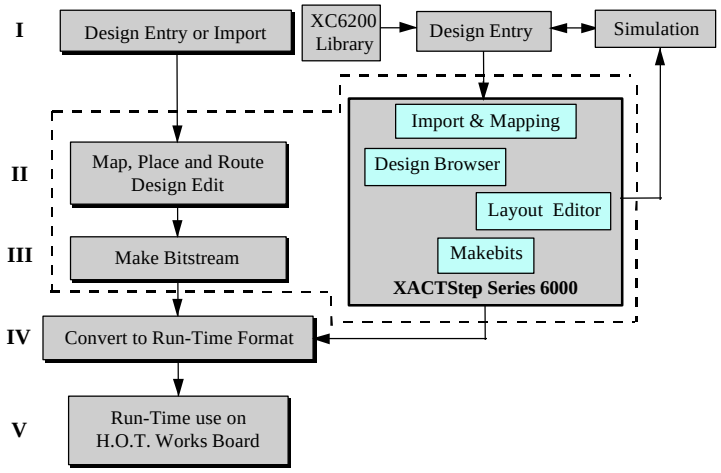


Figure 2

Schematic design entry and simulation must be done with third party software tools (not included in H.O.T. Works Package). The output file of the design entry program must be in EDIF format for the XACT6000 software to import. (See the XACT6000 Manual for more information).

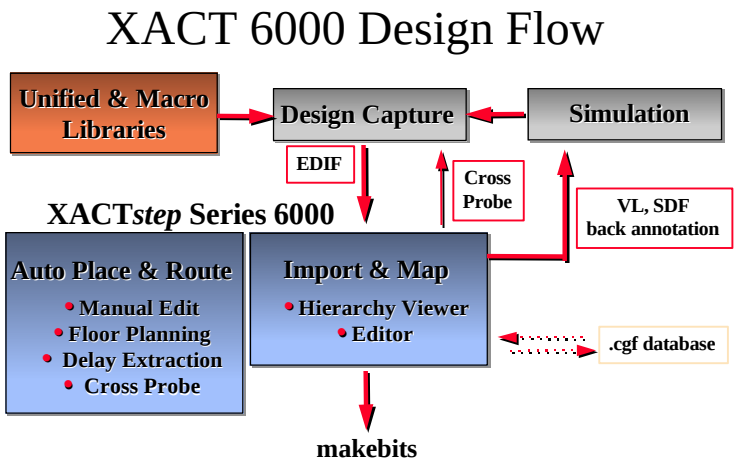


Figure 3

Design Flow for VHDL Entry Method

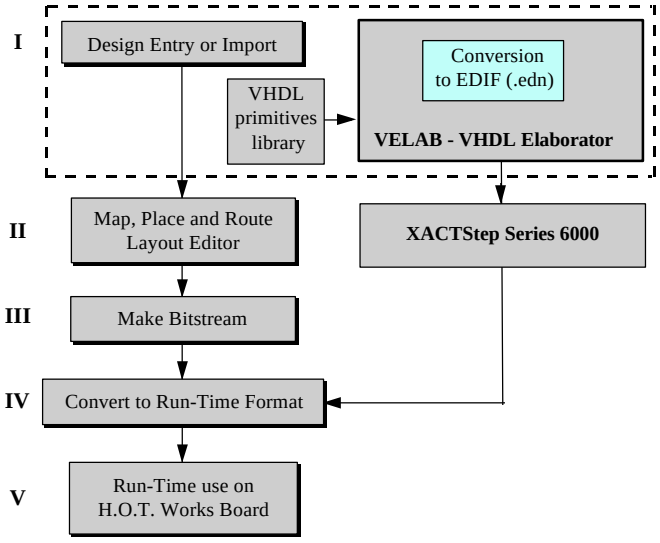


Figure 4

This tutorial assumes that you know how to create a design using structural VHDL. Velab converts the VHDL code into EDIF Format file for use with the XACT6000 software.

VeLab Design Flow

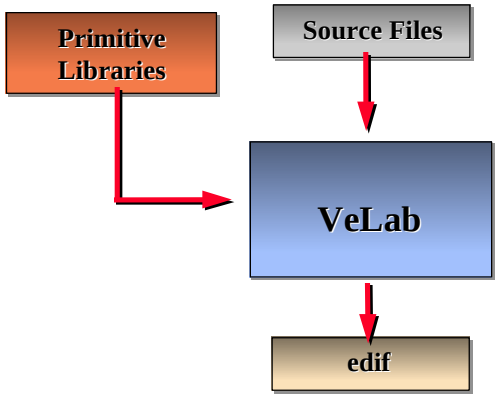


Figure 5

Design Flow for Lola Entry Method

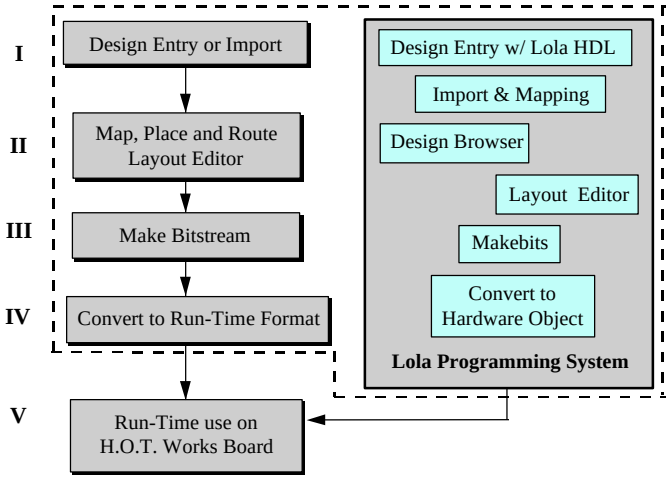


Figure 6

Features of the Lola Programming System include a Suite of Tightly Integrated Tools for efficient design & implementation of algorithms on the H.O.T. Works Platform.

The Lola Programming System contains Lola HDL & Lola Compiler, Layout Editor, Circuit Checker, a technology mapper, a placer & router, and a bit-stream generator.

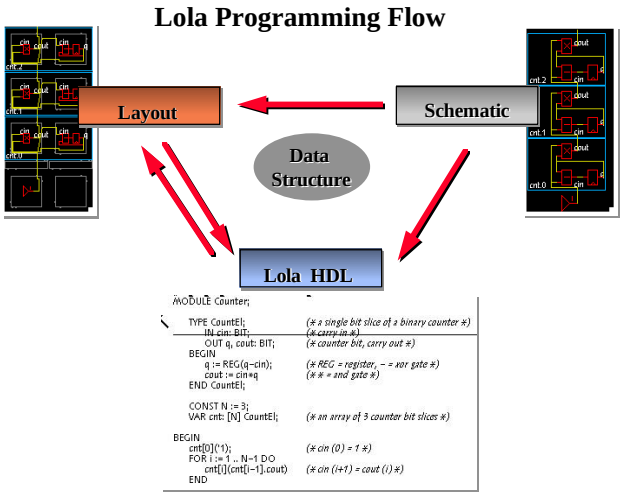


Figure 7

Run-Time Programming

Design Flow for Run-Time Reconfiguration

The two methods for Run-Time Reconfiguration of the H.O.T. Works Board differ in the method by which the digital design is downloaded into the RPU.

With one method, the CAL file (your digital design) is loaded from the harddisk to the H.O.T. Works Board via a program command. This method requires use of C++ support software for the PCI/6200 hardware. These files contain routines to support Low-level board interface, plug and play support, XC6200 Device configuration support, XC6200 Runtime support and XC6200 Debug support. With the other method your digital design is compiled into an executable program. The downloading of the design to the H.O.T. Works Board occurs at program execution time. This method requires the conversion of the *.CAL* file to a Hardware Object . It also requires C++ routines for control of the H.O.T. Works Board. This is VCC's Hardware Object Technology Method of Run Time Reconfiguration.

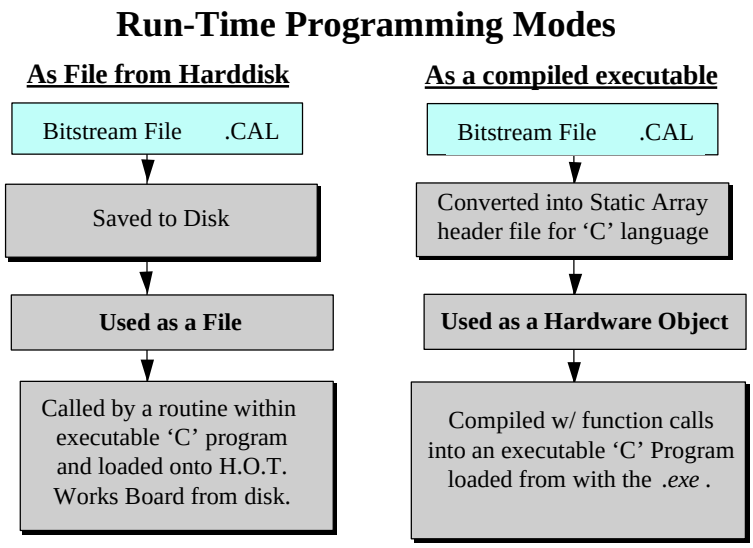


Figure 8

File from the Harddisk --- Mode

There are three classes providing a C++ interface to the H.O.T. Works Board. These classes allow the user to interface to the H.O.T. Works Board from their own C++ code with a few simple function calls. The code uses the **`hotworks.vxd`** device driver to interface to the PCI board. See Chapter Five for details on the H.O.T. Works Class Files.

These are three classes that make up the H.O.T. Works application interface:

1. The **XC6200DS** is the top level class and provides a complete interface to the XC6200 Development System. This class contains all the 6200 chip access functions, including the reading/writing of the control registers and loading and accessing the user design.

2. The **PCI Board** is a subclass of the XC6200DS class. This class contains the board IOSpace register access functions.

3. The supporting low level class **PCICore** handles the actual reading and writing to the H.O.T. Works Board as a memory mapped device. This class contains all the necessary code for interfacing to the device driver.

Building the Executable Program

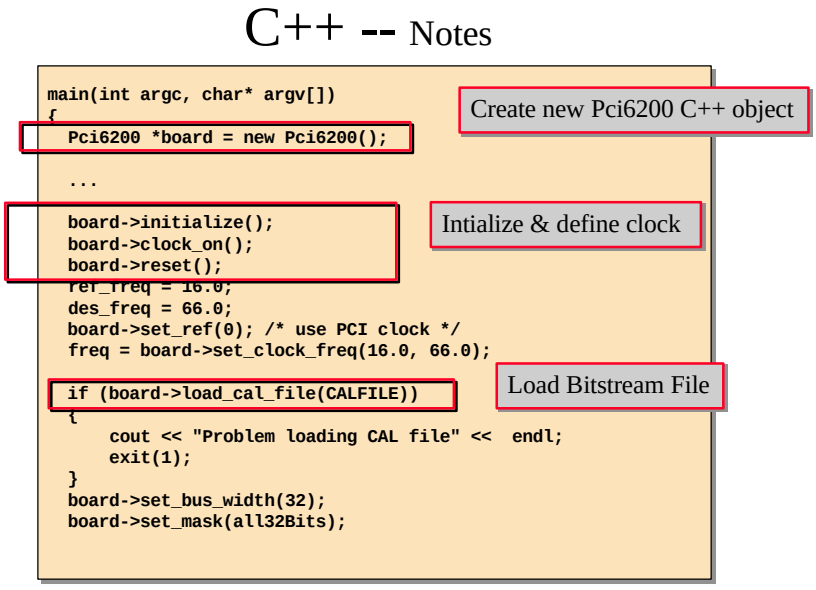


Figure 9

C++ -- Notes

```
for(int i = 0; i < 1000; i++)
{
  // Generate 4 random nos (11-bit)
  a = rand()%2048;
  b = rand()%2048;
  x = rand()%2048;
  y = rand()%2048;

  // Write these data input registers
  board->set_map(all32Bits, noBits);
  board->set_column(multin, a + (x << 16));
  board->set_map(noBits, all32Bits);
  board->set_column(multin, b + (y << 16));
  // Read back the answer
  board->set_map(all32Bits, noBits);
  c = (board->get_column(addout)) & 0x007fffff;

  fout << a << "*" << x << " + " << b << "*"
       << y << " = " << c << endl;
```

Write

Read

Figure 10

Figure 9 shows the Initialization of the H.O.T. Works Board and the Design File (CAL) downloading C++ functions.

Figure 10 shows The Read and Write C++ functions. See the C++ Interface Section below for more information on the Interface Classes and their functions.

Hardware Object Technolgy --- Mode

Hardware Objects are converted digital designs called from within a software program. The following describes a technique for creating Hardware Objects and embedding them into a compiled 'C++' program.

You can reuse the Hardware Object over and over or in combination with other Hardware Objects. An integrated software driver and bus interface gives flexibilty and freedom from bus protocols. You need only convert the digital design into a Hardware Object to enable its use in an application programming environment.

After the final placement and routing of your design the resulting CAL file is converted into a **.h** file to be included within a 'C++' program application. Once the design has been converted

by a program called **cal2h** and inserted into an application, the Hardware Object is ready for use. The H.O.T Conversion Program is located `C:\Hotworks\Api`

`cal2h.exe` takes the `<filename>.CAL` and outputs `<filename>.h`

The **`.h`** file must be included in your program for compilation with:

#include "<header file name>"
The file name is the array name of the design - created by the **cal2h** program used earlier in the design development cycle.

This Hardware Object downloading routine is used to load the Hardware Object.

loadHOT();
loadHOT(<filename>)
For example: **loadHOT(my_design)** downloads the Hardware Object called **my_design** into the H.O.T. Works Board.

With the above two commands and the other C++ routines provided, you can use your design from within your application program.

Before you install H.O.T. Works Development System

We urge those not familiar with hardware/software co-design or configurable computing technology to read the manual first. The H.O.T. Works Board is bleeding edge technology, and faulty design implementations can damage the RPU, FPGA or other components on the board. Virtual Computer Corporation is not responsible for damage due to faulty designs. Each development system has been thoroughly checked and is in working order prior to shipment. Please see the enclosed H.O.T. Works User's Guide for more information on warranty policy.

Check to make sure you have received a complete package and that your system has the minimum requirements before proceeding with installation.

Checking package contents

Your H.O.T. Works Development System package should contain the following:

- H.O.T. Works PCI Virtual Computer Board
- A program CD with software and on-line documentation titled -- *H.O.T. Works Development Software Package*
- A program CD with software and on-line documentation titled -*XACT6000*

  Two printed manuals titled --*H.O.T. Works User's Guide & XACT6000*

  Release documents, Registration, warranty and license agreement inserts

If your package is not complete, call customer service at Virtual Computer Corporation. (818)342-8294 (U.S.) or email support@vcc.com.

Making sure your system meets the minimum requirements

The following table lists the minimum requirements for H.O.T. Works.

System Requirements

Components	Requirements
Processor	Pentium® /66 megahertz or higher
RAM	16MB (minimum) 32MB preferably
Available disk space	65 MB
Operating System	Window95®
CD-ROM drive	needed for software installation

Installing the H.O.T. Works Development System

We recommend that you quit all other applications during this installation process.

- **Please follow this procedure for installation of the hardware and software components on your system.**

⇒ Put the *XACT 6000* CD into your CD-ROM drive. This CD contains the XACT 6000 software, interactive demo and Adobe® Acrobat® Reader.

⇒ The XACT6000 installation program assumes you have already installed **Adobe Acrobat Reader**. To install the Acrobat Reader version 3.0; Run *ACROBAT/AR16E30.EXE* on the CD. Adobe and Acrobat are trademarks of Adobe Systems Inc.

⇒ Run *SETUP.EXE* on the CD to install the XACT6000 software. Follow the instruction on the monitor screen. {Note: Default Directory for this installation is `C:\Xact6000`}

⇒ Read the enclosed Release Document titled -- *Series 6000 Reconfigurable Processing Unit* for further installation information and options for XACT6000.

⇒ Put the *H.O.T. Works Development Software* CD into your CD-ROM drive. This CD contains necessary drivers for your system as well as software programs, documentation, source code and design examples.

⇒ Run *SETUP.EXE* on the *H.O.T. Works Development Software* CD to continue installing the software programs and files. Follow the instruction on the monitor screen. Reboot after this installation. {Note: Default Directory for this installation is `C:\Hotworks`}

This completes the software installation process.

- **We recommend that you follow the instructions provided by the manufacturer of your computer for installing plug-in cards. Please remember that the H.O.T. Works Board is a PCI compatible board and must be installed into a PCI slot of the computer.**

⇒ Turn off the computer and remove the cover for installation of the H.O.T. Works Virtual Computer Board.

⇒ Install the H.O.T. Works Board in one of the PCI slots. Be sure to use caution when installing hardware into your computer.

⇒ Turn on the computer.

⇒ When the Window95 **New Hardware Found** dialog appears; select the **' .inf '** file located in the Hotworks directory. C:\Hotworks\drivers\hotworks.inf . See figures 11, 12, 13.

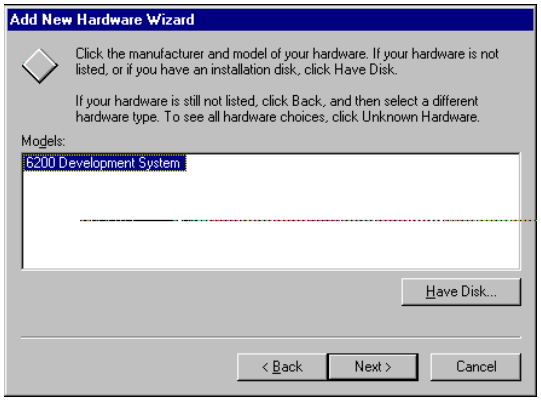


Figure 11

⇒ `Hot Works Development System` should be highlighted.

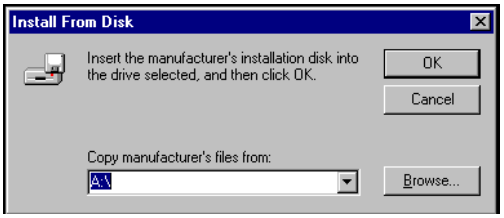


Figure 12

⇒ Goto c:\Hotworks directory.

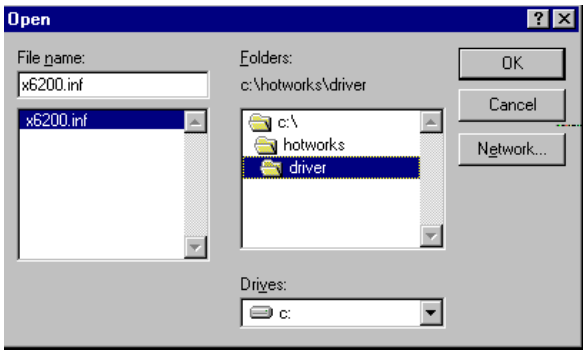


Figure 13

⇒ Select `Hotworks.inf`

- You should have successfully installed the hardware and software components of the H.O.T. Works Development System. Let's test the H.O.T. Works Board now.

Testing the H.O.T. Works PCI Virtual Computer Board

The following procedure will test the PCI Board to determine whether the installation process was successful.

⇒ Reboot computer

⇒ Upon rebooting of the computer, the system should have recognized the H.O.T. Works PCI Board. To confirm the presence of the board and it's correct identification in the Window95 system goto the**Control Panel** and select **System.** See figure 14.

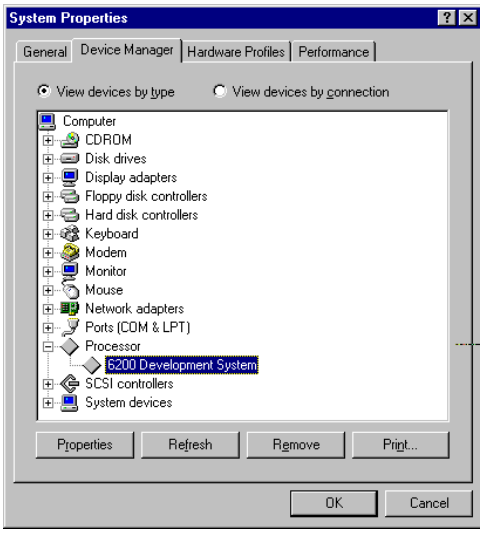


Figure 14

Testing the Board

The following procedure will test the H.O.T. Works Board by loading test designs and accessing the various features on the board using PCITest program located in:

```
C:\Hotworks\pci_test
```

PCITest Program is your initial doorway to the H.O.T. Works Board. It enables you to probe the XC6200RPU, read and write instructions, data and commands to the board and SRAM. The

following is an introduction to it's features and use. More information on PCITest program can be found in Chapter 2.

⇒ Under Windows, the Hotworks Setup installs a folder on the Start Menu Called **Hotworks**. In this folder one of the items is the **PCITest Program** OR/
⇒ Run `PCITest.exe` Located in: `C:Hotworks\pci_test` directory.
⇒ Choose Help. Select Test Command Files . See figure 15

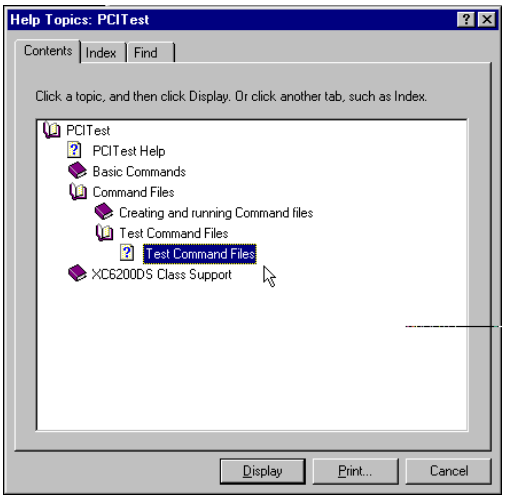


Figure 15

The Test Command Help window displays the various test commands utilizied by PCITest program. Select the first command test **`cur1_tst.cmd`** , an explanation of this test is displayed. See figure 16.

⇒ Select the title command name and copy it (ctrl.c).

⇒ Return to the PCITest main window. Type **`obey <space>`** and paste the command name. It should read: **`obey cur1_tst.cmd`**

⇒ Press Run. See Figure 17.

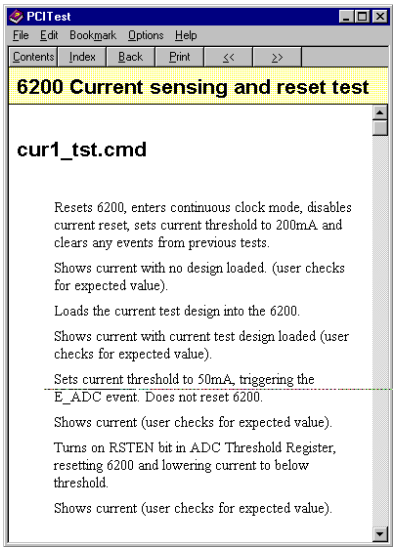


Figure 16

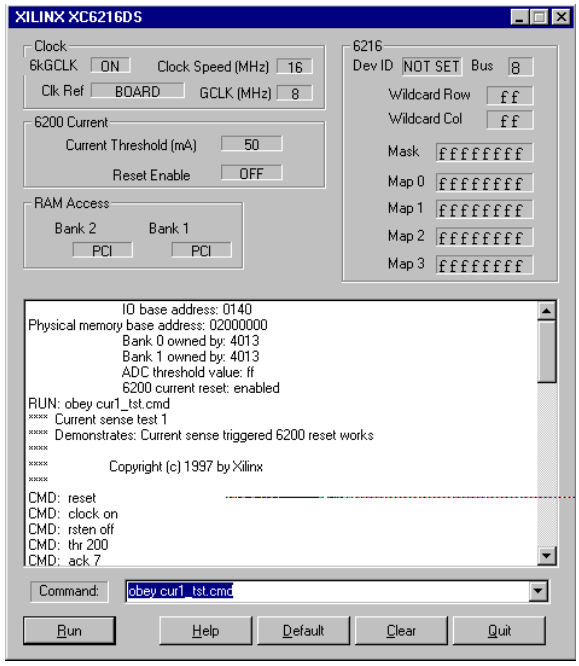


Figure 17

Upon running the Current sensing and reset test, the command operation is display in the PCITest information window. The figures below is the result of running this test. Figures 18 thru 21.

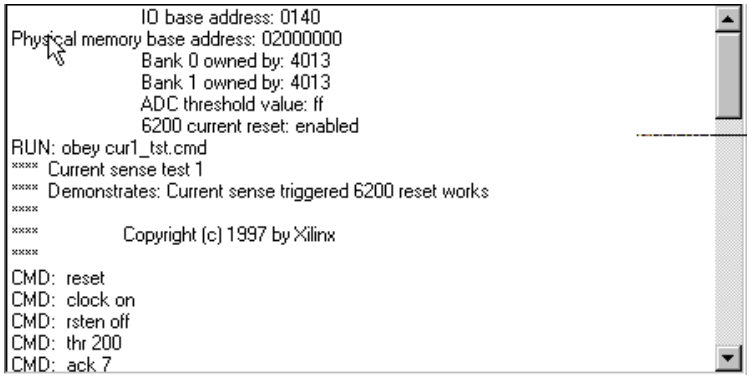


Figure 18

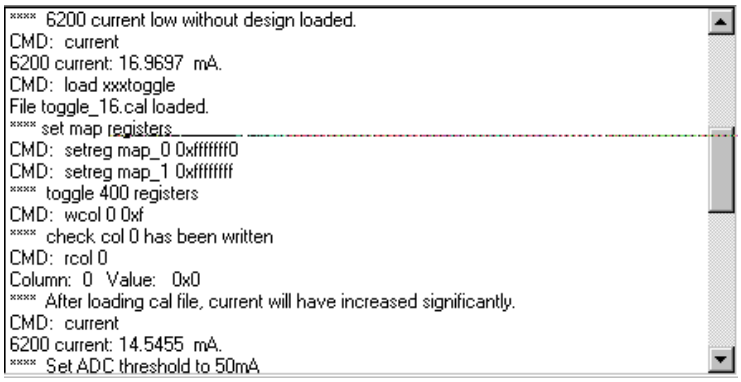


Figure 19

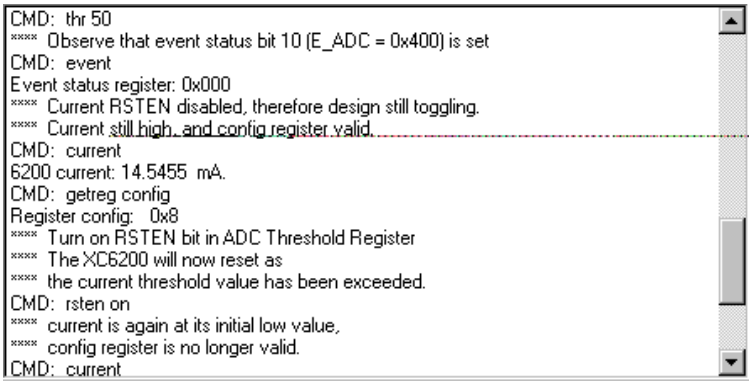


Figure 20

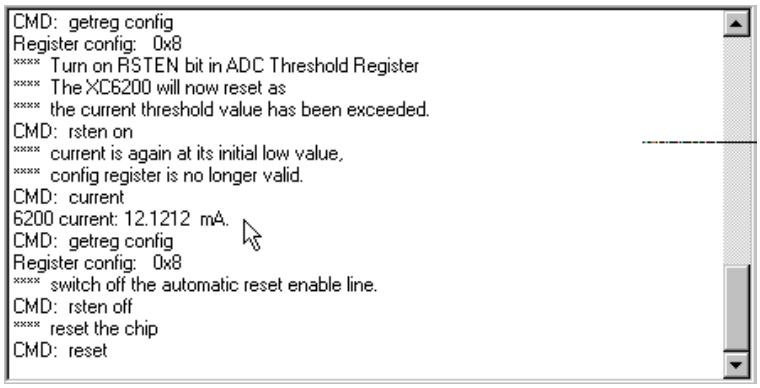


Figure 21

Errors are indicated in the information window.

You may run other test by selecting the test command file name and running them with the **obey** command.

This complete the installation process. We at VCC wish you all the best in you explorations with the H.O.T. Works Development System.

Be sure to send the your registration card in. You will receive a subscription notice and password to participate in the Hotuser's Electronic Forum. The purpose of the Hotuser's Forum, an email reflector, is to provide a forum for ideas, comments, questions and hopefully answers to issues regarding the H.O.T. Works Development System. Virtual Computer will post update and version information to the H.O.T. Works Users on the Forum.