RADIATION TOLERANT MANY-CORE COMPUTING SYSTEM FOR AEROSPACE APPLICATIONS

by

Clinton Francis Gauer

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2010

APPROVAL

of a thesis submitted by

Clinton Francis Gauer

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citation, bibliographic style, and consistency and is ready for submission to the Division of Graduate Education.

Dr. Brock J. LaMeres

Approved for the Department of Electrical and Computer Engineering

Dr. Robert C. Maher

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Clinton Francis Gauer

April 2010

ACKNOWLEDGEMENTS

I would like to thank Dr. Brock LaMeres for bringing me into this project and serving as my advisor. I would also like to thank Dr. Robert Ray and Dr. Andrew Keys from NASA's Advanced Avionics and Processor Systems for mentoring this project, Dr. Ross Snider and Randy Larimer for serving as committee members, NASA sponsored Senior Design teams at Montana State University for their contributions, David Racek for helping me during the summer of 2009, and my family and friends for their support throughout my college career. Funding for this project came from Montana Space Grant Consortium (NASA EPSCoR) and the NASA Exploration Systems Mission Directorate Higher Education Program.

TABLE OF CONTENTS

TABLE OF CONTENTS - CONTINUED

LIST OF TABLES

Table Page

LIST OF FIGURES

LIST OF FIGURES - CONTINUED

LIST OF FIGURES - CONTINUED

Figure Page

ABSTRACT

When integrated circuits are exposed to ionizing radiation, a variety of fault conditions can occur. This draws concern to the aerospace community as they look toward integrating more complex computing systems into flight applications. The detrimental effects that radiation can have on integrated circuits can be broken up into two categories: single event effects and total ionizing dose. Single event effects refer to non-destructive electron hole pairs that are created by the radiation which can lead to logical failures. Total ionizing dose refers to the permanent damage to a device caused by the electron hole pairs getting trapped prior to recombination and results in oxide breakdown and leakage current. In order to provide a robust computing platform for aerospace applications, both of these effects must be addressed. This thesis presents a set of novel fault mitigation strategies to increase the reliability of aerospace flight computers by exploiting the reconfigurability of Field Programmable Gate Arrays. First, redundant circuitry and a voting system are used to recover from non-destructive faults. Secondly, spare circuitry is used to spatially avoid faults and replace permanently damaged circuitry. Finally, partial reconfiguration of the Field Programmable Gate Array is used to repair faults in the configuration memory of the device. These fault mitigation techniques are all combined into a complete system to provide a robust computing platform for aerospace applications.

1

INTRODUCTION

When integrated circuits (ICs) are used in space environments, they are bombarded with a wide range of radiation particles, typically in the form of electrons, protons, and heavy ions from solar flares [1]. These particles penetrate into the device causing electron/hole pairs which can lead to possible circuit failures. ICs used in aerospace applications need to be able to tolerate these radiation effects in order to provide a reliable computing platform. Historically, radiation tolerance has been accomplished with costly fabrication process techniques or with redundant circuitry. Both of these techniques lead to increased cost and reduced performance of the device.

Field Programmable Gate Arrays (FPGAs) are ICs containing configurable logic blocks (CLBs) with configurable interconnects that can be programmed into a variety of digital logic circuits [2]. CLBs are made up of two slices that each contain look up tables (LUTs), flip-flops, and logic gates. In addition, FPGAs provide block RAM (BRAM), clock buffers, digital clock managers (DCMs), phase lock loops (PLLs) and other necessary components required for digital circuit implementation. The following figure shows a Xilinx Virtex-5 FPGA.

Figure 1.1: A Xilinx Virtex-5 FPGA.

Over the past decade FPGA performance has significantly increased. Commercial FPGAs are now comparable in performance to microprocessor implementations in addition to providing greater power savings [3-7]. SRAM-based FPGA configuration memory allows these devices to be reconfigured during operation, making them even more attractive than microprocessors.

The performance, flexibility, customization, and reconfigurability of FPGAs make them desirable for use in aerospace applications. The real-time reconfiguration of FPGAs enables the practical deployment of reconfigurable computing systems. Furthermore, modern FPGA fabrics provide the necessary resources for redundant circuitry and other novel fault mitigation techniques. The low power consumption and increasing performance of FPGAs make them attractive choices for computationally demanding flight systems.

This thesis describes the design and implementation of radiation tolerant computing systems that can be used for aerospace applications. The systems consist of a variety of novel fault mitigation techniques in order to provide fault tolerance based on exploiting the reconfigurability of commercial SRAM-based FPGAs. In these systems, soft-core processors are used in conjunction with dedicated recovery circuitry to mitigate the effects of ionizing radiation on ICs. An FPGA provides the hardware fabric for the soft processors and the fault recovery circuitry and is an attractive choice due to its inherent flexibility.

An FPGA can be reconfigured by writing a completely new programming file to the entire chip. For real-time reconfiguration any variable information used by the

system needs to be stored off-chip prior to reconfiguration and then reloaded once the programming is complete. The programming information is stored off-chip and loaded using custom programming circuitry. Full FPGA reconfiguration is a relatively timely process and is not desirable in the event of sudden radiation exposure. However, the same FPGA can be used for many different systems thus lowering the number of parts onboard a flight vehicle.

Another method of reconfiguring the FPGA is with partial reconfiguration (PR). Partial reconfiguration is a dynamic process that changes the circuit configuration in only certain regions of the FPGA while the other regions remain functioning and uninterrupted. The partial programming files are stored off-chip but the programming circuitry is contained internally on the FPGA. This reconfiguration technique enables much lower programming time compared to that of the full FPGA reconfiguration. It can also be used to repair corruptions to the FPGA configuration memory caused by radiation effects.

The first computing system presented in this work dynamically adapts its architecture by reconfiguring the entire FPGA for three different modes of operation depending on current system needs. The three modes are: low power, parallel processing, and radiation tolerant. When the system is operating simple tasks or a sequential algorithm, the FPGA can be configured with a single processing unit to conserve power (i.e., low power mode). When performance demands become greater, the FPGA can be reconfigured with a parallel processing scheme to provide increased computation. Both of these modes assume that the FPGA is not being exposed to

ionizing radiation. In the event that radiation suddenly becomes present, the system can switch to a radiation fault mitigation mode in which redundant processors are configured in TMR to recover from radiation strikes through majority voting. As proof of concept, a prototype system has been created in which the processors control simple I/O peripherals (lcd, keyboard, and mouse) and faults are induced manually.

The second system presented in this work exploits partial reconfiguration. The Xilinx PicoBlaze soft processor is used to form a many-core architecture. The PicoBlaze processor is a simple core that is small enough to fit dozens of processors on a single FPGA. It is an 8-bit processor that provides enough processing power to handle basic computing tasks and lower computation needs. The processors are arranged in a 3+13 design, meaning that at any given time 3 processors are active in a redundant fault mitigating scheme and 13 are reserved for spares. The active processors can switch among the spares in an effort to spatially avoid radiation strikes. Each processor can be partially reconfigured to their original condition in the event of radiation damage to the configuration SRAM corresponding to that processor. Also, if one of the active processors becomes permanently damaged from the radiation effects, a spare can be deployed as a replacement.

For applications that require a high level of processing power, a final computing system is presented. This system uses the more sophisticated Xilinx MicroBlaze 32-bit soft processor that can handle much more complex algorithms and applications. It features a 3+1 design. Again, the 3 processors are arranged in a redundant fault mitigating scheme while 1 processor is reserved for spare replacement. Partial

reconfiguration capabilities are also included as a mitigation strategy in this system for configuration SRAM faults.

These systems have been prototyped using Xilinx Virtex-5 FPGAs. The first system was developed using the Xilinx ML505 project board. It was demonstrated that a mode configuration change required 527.2ms and a soft fault recovery was accomplished in 23.2ms. The second and third systems were developed using the Xilinx XUPV5-LX110T project board. In the 3+13 PicoBlaze system, it was demonstrated that soft fault recovery took 23.2ms and PR of a damaged processor took 66ms. In the 3+1 MicroBlaze system, it was demonstrated that soft fault recovery took 11.9ms and PR of a damaged processor took 525ms.

6

MOTIVATION

Radiation Effects on Integrated Circuits

When integrated circuits (IC) are exposed to space environments, they are susceptible to ionizing radiation, typically in the form of electrons, protons, and heavy ions from solar flares and other cosmic particles [1]. These particles ionize the semiconductor material used in the circuit resulting in the occurrence of fault conditions. High energy protons and heavy ions create fault conditions called Single Event Effects (SEE) [8]. Figure 2.1 shows a MOS transistor being struck with ionizing radiation. SEEs do not cause permanent damage to the device; however, the ionizing effects cause free charge to be created in the diffusion region of the device. If the charge becomes large enough, a voltage spike on the output may be observed by the receiving gate leading to possible logic failure in the circuit [1,9]. This voltage strike is known as a Single Event Transient (SET) and is shown in Figure 2.2. When an inadvertent logic transition due to an SET is latched into a flip-flop or other digital storage element, this is referred to as a Single Event Upset (SEU) [10]. When an SEU occurs in the configuration SRAM of an FPGA, it is referred to as a Single Event Functional Interrupt (SEFI) and will alter the physical circuitry of the fabric [11].

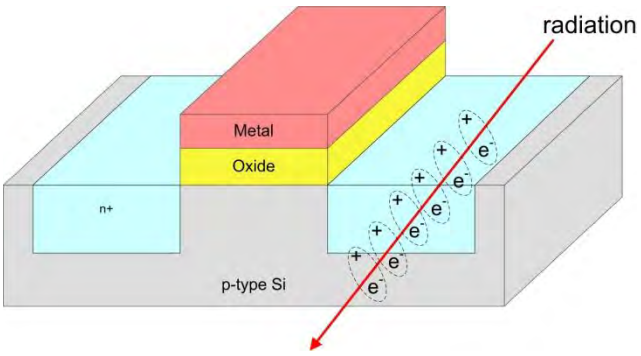

Figure 2.1: Ionizing radiation striking a MOS transistor.

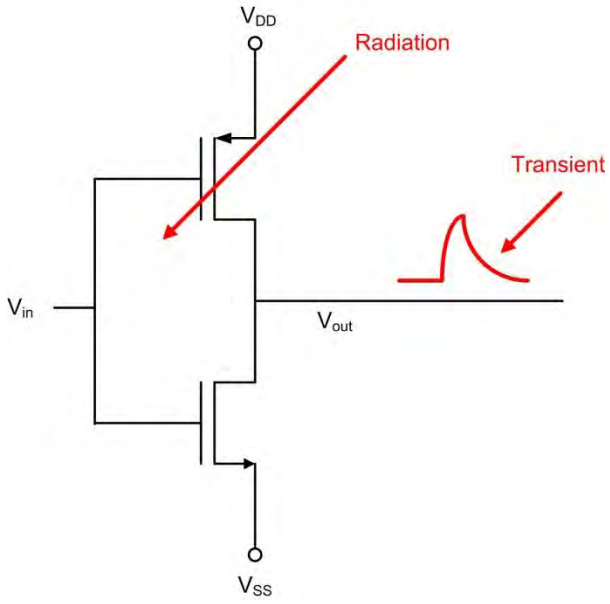

Figure 2.2: Example of an SET on an inverter circuit.

Ionizing radiation can also cause permanent damage to the circuit fabric of FPGAs. Total Ionizing Dose (TID) refers to long term damage of the device mainly due to low energy electrons and protons [12]. This occurs when charge carriers get trapped in the insulating or more lightly doped regions of the device. Electron mobility of semiconductor materials tends to be higher than hole mobility ($\mu_n > \mu_p$). When radiation

strikes cause an electron/hole pair to form, these carriers attempt to move back together to find an electrostatic equilibrium. Due to the time the carriers take to recombine, the hole charge carriers have a higher chance of getting trapped in the insulating or more lightly doped regions of the device. This degrades the transistor resulting in threshold shifts, increased device leakage, timing changes, and functional failure of the device [8]. Generally, radiation hardened systems are specified to withstand a lifetime TID amount which is sufficient to last throughout the mission plan. However, for long term missions specifying a lifetime TID tolerance becomes complicated and it is desirable to monitor the degradation so that replacements can be made prior to a critical system failure. The following figure shows a MOS transistor that has suffered oxide breakdown and substrate doping as a result of TID.

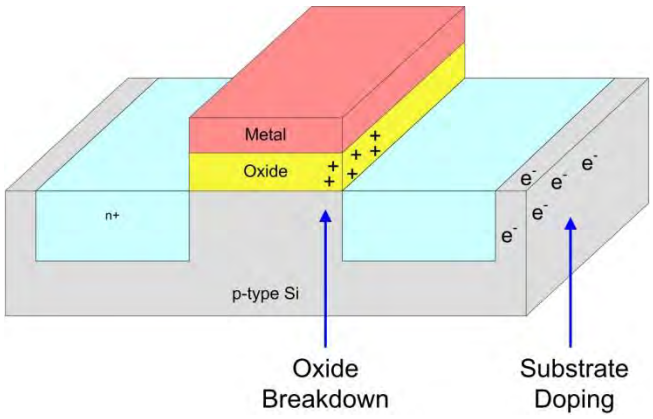

Figure 2.3: Oxide breakdown and substrate doping in a MOS transistor as a result of TID.

Existing Mitigation Strategies

Many techniques have been proven to help mitigate SEU failures. One of the most commonly used techniques is Triple Modulo Redundancy (TMR) [13-14]. TMR can be implemented at the hardware level using three redundant circuits to detect and correct logical errors [13]. A voter circuit consisting of a combinational logic comparator is used to produce an output dependant on the majority of outputs from the three redundant circuits. A recovery sequence can be added to the TMR system to reset and reinitialize the circuitry when a fault is detected in microprocessor-based systems [15]. The following figures show a simple logic circuit with and without TMR.

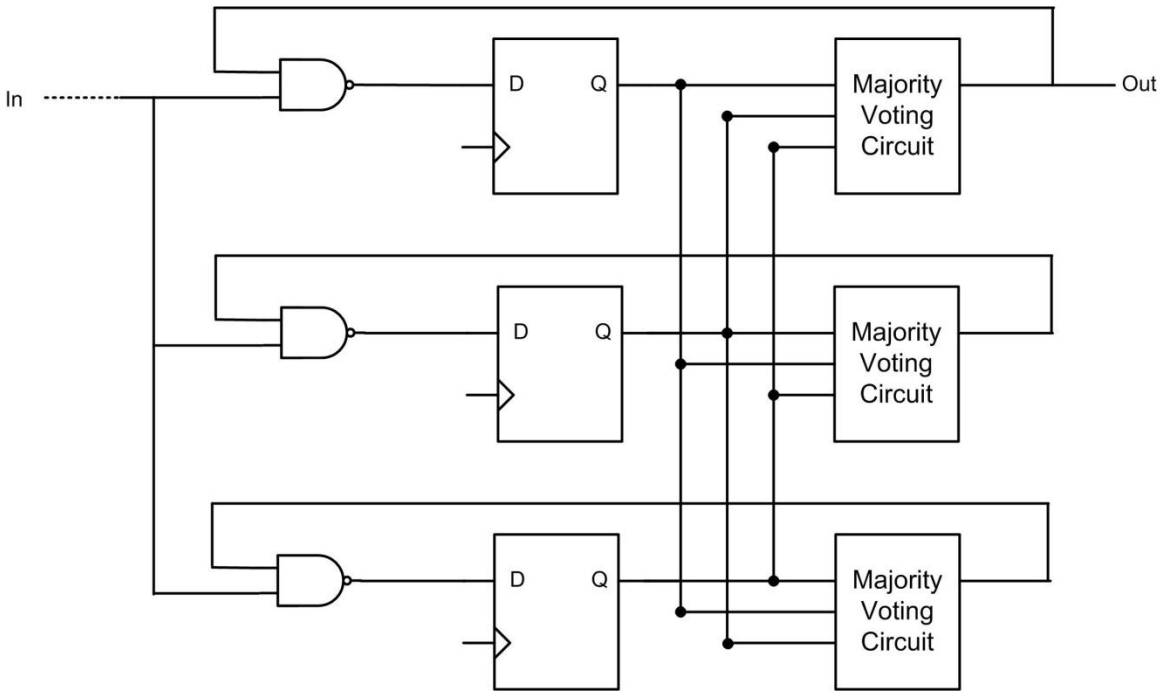

Figure 2.4: Simple logic circuit with TMR.

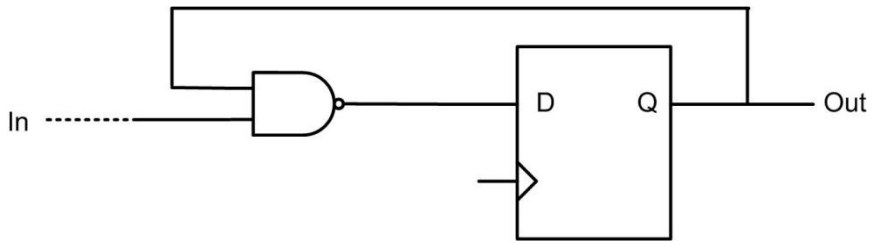

Figure 2.5: Simple logic circuit without TMR.

TMR can also be implemented at the application layer, known as *radiation hardened by software*, in which redundant processes are running with the voting performed in software [16]. Watchdog timers can also be deployed as a fault mitigation technique. Watchdog timers monitor the system and reset the circuitry if the system has been idle for too long [15]. All of these techniques can easily be implemented in an FPGA. However, the downsides to these approaches are increased area, reduced performance, additional power consumption, and increased development time. Also, finding the optimal fault observation nodes is challenging due to the impact the observation circuitry has on the operation of the circuit being monitored [11,14]. Care must be taken as to where TMR is placed so that fault coverage will actually be achieved. Poorly placed TMR systems can increase susceptible circuit area resulting in worse system performance [10]. The voter and recovery circuitry can also suffer faults and it becomes difficult to detect and correct them.

Total Ionizing Dose has historically been addressed by altering the physical circuit in an attempt to make it less susceptible to permanent radiation damage. Techniques including isolation trenches, substrate doping, and non-standard layouts have been developed to make integrated circuits radiation hardened during the fabrication

process. Isolation trenches are created using field oxide which is trenched into the silicon to reduce the amount of charge buildup at the silicon/oxide junction and prevent current leakage between devices [17]. Substrate doping attempts to lower the resistivity of the semiconductor substrate to increase the probability that the electron/hole pairs will recombine. Gold doping is a common technique that has been demonstrated which produces a less resistive transistor body [18] and decreases the amount of charge buildup in the device. Non-standard layout techniques, such as enclosed-layout-transistors (ELT), use a modified transistor layout where the area of the susceptible diffusion region is reduced to lower the probability of a radiation strike occurring [19]. Radiation hardened parts are specified to withstand particular radiation dosages (i.e., lifetime TID). The major drawback of these techniques is that they require a dedicated radiation hardened fabrication process resulting in higher production costs. Also, logic mitigation techniques are still required because TID hardened parts do not prevent SEUs caused by high energy protons and heavy ions.

Radiation hardened ICs were of great interest to the Department of Defense (DoD) in the 1970's and 1980's [20]. The DoD sponsored initiatives to produce radiation hardened semiconductor devices that could withstand the radiation aftermath of a nuclear weapon discharge. During the 1990's the DoD shifted away from wide scale nuclear warfare and the radiation hardened foundry infrastructure began to deteriorate as demand and government subsidies declined.

Commercial processors and other ICs that do not require radiation hardening continue to increase in performance. Radiation hardening by architecture or fabrication

decreases the performance of the devices. The mismatch in performance between commercial and radiation hardened parts can lead to hardware and software compatibility issues with emerging technology [21] in addition to computing platforms that don't meet the computational need of future aerospace missions. The following figure shows the performance lag between commercial processors and radiation hardened processors is 10 years.

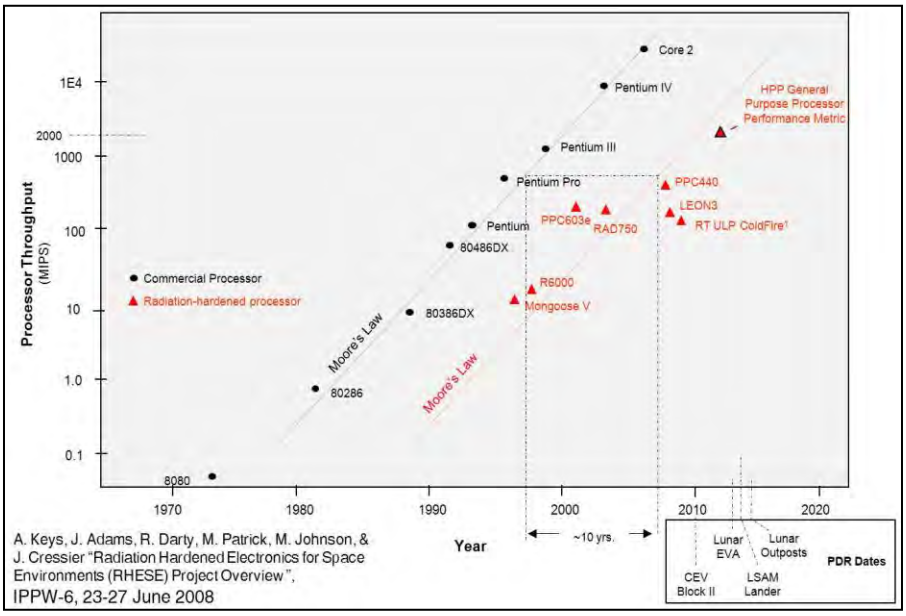

Figure 2.6: Performance lag between commercial processors and radiation hardened processors [12].

FPGAs as a Platform for Radiation Hardened Systems

With the increasing performance and inherent flexibility of SRAM-based FPGAs, the aerospace community is showing an increased interest in using them for flight systems [22-23]. This means that fault detection and recovery techniques must be

developed for FPGAs that can mitigate radiation induced errors. When an SEU occurs in the logic fabric of an FPGA, it is referred to as a *soft fault* because no permanent damage is done to the circuit. These faults can typically be recovered using a reset [14]. A major problem with FPGAs exposed to ionizing radiation is the susceptibility of the configuration SRAM to SEUs [24-25]. SEFIs cause no permanent damage to the FPGA, but in order to restore the initial state of the circuit, the configuration SRAM needs to be reconfigured. Recent advances in development tools for FPGAs have enabled direct access to the configuration SRAM and allow partial reconfiguration techniques to be used as a fault mitigation technique [26]. FPGAs are also susceptible to TID just as any other IC.

Typically, FPGAs deployed in high radiation environments use fused based configuration ROM to avoid SEFIs. However, the flexibility of the design is limited because the FPGA can only be programmed once, eliminating the option for dynamic reconfiguration in the field. In FPGAs with SRAM-based configuration memory, a scrubber circuit is typically used to detect and correct SEFIs. A scrubber circuit continually compares the data in the configuration SRAM to the original configuration data stored in an off-chip non-volatile device [27]. In the event of error detection, the scrubber simply writes the original configuration data back to the SRAM. Scrubbing can be done with a circuit independent to the TMR system that simply traverses through the memory addresses of the configuration SRAM. However, the scrubber has no insight to where an error has occurred leading to significant latency between the detection and repair of the SEFI.

FPGAs can be divided up into tiles allowing certain circuit elements to be implemented into that area. These tiles can then be partially reconfigured to a new or original circuit configuration [28]. By constraining soft processors into these tiles, SEFIs can be dynamically repaired for a particular processor while the other processors continue operation.

Proposed Technique

This thesis presents two tile-based soft processor reconfigurable computing systems as well as a full FPGA reconfigurable computing system. These systems combine many fault mitigation techniques in an effort to build a stable and robust computing system that can handle extreme radiation environments.

The full FPGA reconfigurable computing system features three separate FPGA configurations that provide low power or high performance computing solutions for radiation free environments and a third mode in the event that the system enters a radiation harsh environment. This radiation tolerant mode implements three PicoBlaze [29] soft processors arranged in a TMR scheme. A voter and recovery system is added to handle fault detection and resynchronization in the event of radiation effects.

In the tile-based approach, the FPGA is divided into equally sized tiles representing a quantum of resources that can implement a PicoBlaze or MicroBlaze [30] soft processor and can be individually reprogrammed using partial reconfiguration. At any given time, three of the processors are configured in TMR and the remaining processors are reserved as spares. In the event of an SEU, the voter and recovery circuit

will attempt to reset and resynchronize the running processors. If a processor fails to recover, it is marked as damaged and a spare processor is brought online for replacement. The damaged processor undergoes a repair process in which the tile is partially reconfigured in an effort to mitigate any SEFIs. The newly repaired processor is added to the list of spares. If the system attempts to bring it back online and it fails again, the processor is marked as *permanently TID damaged* and is no longer available for use.

The tile-based, 8-bit PicoBlaze system offers 13 spares for spatial avoidance and TID replacement. Only one spare is available to the MicroBlaze system, but the 32-bit processor is more capable for heavy computing tasks and more complex algorithms. To demonstrate usability, the processors are programmed to control a set of basic I/O peripherals: keyboard, mouse, and liquid crystal display (LCD). In addition, the MicroBlaze system executes a Turing machine to show computational effectiveness [31].

Like all TMR systems, the full FPGA reconfigurable computing platform offers the ability to mitigate SEUs in the event of harsh radiation bombardment. However, when traditional radiation tolerant systems move out of a radiation harsh environment, they continue to operate in TMR mode resulting in unnecessary power consumption. This reconfigurable system has the ability to switch into a low power mode to conserve power or a parallel processing mode to provide increased computation.

The 3+13 PicoBlaze and 3+1 MicroBlaze systems offer novel radiation mitigation capabilities over traditional radiation tolerant systems. Traditional TMR approaches do not provide a solution to TID failure and TID hardened parts do not offer SEU fault mitigation. Also, using FPGAs as a hardware fabric for a TMR system adds the

susceptibility to SEUs in the FPGA configuration SRAM. The second and third systems presented in this thesis offer the ability to mitigate all three of the major sources of radiation faults in FPGA-based flight systems (SEUs, SEFIs, and TID).

SYSTEM OVERVIEW

Full FPGA Reconfigurable Computing System

The full FPGA reconfigurable computing system is implemented on the Xilinx ML505 project board and uses the PicoBlaze soft processor for its computational platform. It features three modes of operation: low power, radiation tolerant, and parallel processing. The following figure shows an ML505 project board with a Virtex-5 LX50T FPGA.

Figure 3.1: A Xilinx ML505 project board.

Low Power Mode

During low power mode operation, one PicoBlaze processor is implemented to control a keyboard, mouse, and liquid crystal display (LCD). The keyboard and mouse

clock lines are connected to an interrupt on the PicoBlaze processor. Information is passed to the PicoBlaze processor using the PS2 protocol. The processor keeps track of mouse movements and writes the X,Y location to the bottom line of a 32 character LCD while keyboard characters are written to the top line of the LCD. Dip switches on the project board are used to specify which configuration mode is desired. When a mode change is requested, the processor offloads its 64 bytes of RAM and 16 general-purpose data registers to an off-chip EEPROM via I^2C protocol. These bytes contain the mouse X,Y location, keyboard cursor position, and the LCD top line characters. A dedicated state machine then configures the System ACE controller to write the new mode configuration file to the FPGA.

Parallel Processing Mode

The parallel processing mode uses three PicoBlaze processors that work in parallel to handle the IO peripherals. One processor communicates with the mouse while a second processor handles the keyboard interaction. Both of these processors pass their information to the third processor which drives the LCD interface.

Radiation Fault Tolerant Mode

The radiation tolerant mode uses three PicoBlaze processors placed into a TMR scheme. These processors are configured to handle the IO peripherals in the same way as the low power mode processor. A voter is added to the memory address and instruction lines of the processors to check for faults. In the event of an SEU fault detection, a non-corrupted processor offloads its variable data (64 bytes of RAM and 16 data registers) to

the off-chip EEPROM. The voter resets all three processors so they can simultaneously read in the valid variable data from the EEPROM before continuing normal operation. Pushbuttons on the project board are connected to the memory lines to emulate SEUs in the processors.

3+13 PicoBlaze System

The 3+13 PicoBlaze system is implemented on the Xilinx XUPV5-LX110T project board. The XUPV5-LX110T project board is identical to the ML505 except that it uses a Virtex-5 LX110T FPGA. This system uses the PicoBlaze processor to control a keyboard, mouse, and LCD. During any given time, three PicoBlaze processors are configured in a TMR scheme with a voter to detect and recover from SEUs. The voter and recovery circuit also has the ability to switch any of the active processors to one of 13 spare PicoBlaze processors in the event of TID failure. As in the fault tolerant mode of the first system, the off-chip EEPROM is used to reinitialize faulted processors as well as spare replacements. SEUs are simulated with project board pushbuttons tied to the memory lines.

On the Virtex-5 LX110T floorplan, the processors are arranged in a 4x4 grid using the Xilinx PlanAhead 10.1 software. PlanAhead creates individual programming files for each of the sixteen processors allowing them to be partially reconfigured in the event of SEFI corruption. Partial reconfiguration is handled with a MicroBlaze soft processor added to the system. A graphical user interface (GUI) connects to the system using the RS232 protocol and is used to simulate a SEFI / TID failure. The GUI

"corrupts" an active processor so the voter circuit will switch to a spare and partial reconfiguration can be performed on the corrupted processor.

3+1 MicroBlaze System

The 3+1 MicroBlaze system is also implemented on the XUPV5-LX110T project board and uses the MicroBlaze soft processor as a computational platform. In addition to controlling the keyboard, mouse, and LCD peripherals, this system runs a Turing machine to show computational effectiveness. At any given time, three MicroBlaze processors are arranged in a TMR scheme with one reserved as a spare. Fault mitigation is handled the same for this system as it is for the 3+13 PicoBlaze system. The pushbuttons and a GUI are used to simulate SEUs, SEFIs and TID failure. The processors are floorplanned into one column and a separate MicroBlaze processor handles the partial reconfiguration for the system.

FULL FPGA RECONFIGURABLE COMPUTING SYSTEM

The full FPGA reconfigurable computing system is implemented on the Xilinx ML505 board and is designed to control basic peripherals (keyboard, mouse, and LCD). This system offers three modes of operation: low power, parallel processing, and radiation tolerant. When the system enters a radiation harsh environment, it automatically reconfigures into the highest priority mode, radiation tolerant. When the system is not in radiation, it can reconfigure itself into either parallel processing mode or low power mode depending on desired performance. This system also contains a power-up mode that is used to initialize the EEPROM, mouse, and LCD when the system is first turned on. The system was developed in VHDL using the Xilinx Integrated Software Environment (ISE) development software. The processing is performed with a PicoBlaze soft-core processor.

PicoBlaze Soft-Core Processor

For this system, the PicoBlaze soft processing core was chosen as the computational platform. The PicoBlaze is an 8-bit soft processor provided by Xilinx to be used with their ISE development software. In the ISE project, the PicoBlaze VHDL files are imported and glue logic is created to connect all system components.

The PicoBlaze is programmed in assembly language using the Mediatronix pBlazIDE development environment. This program compiles the assembly language into a VHDL file containing the PicoBlaze hardware language (opcodes and operands). This file is imported into ISE along with the other PicoBlaze files.

System Power-On

When the system is first powered on, the EEPROM, mouse, and LCD need to be initialized. The ML505 contains an STMicroelectronics M24C08 I^2C EEPROM. The EEPROM is used during system operation for offloading valid processor data before a configuration mode change or to recover a faulted processor. During the power-on stage, EEPROM bytes that are used to hold typed keyboard characters, keyboard cursor position, and mouse X,Y location are cleared. The system is then reconfigured into the desired operating mode which loads the EEPROM data into the processors and starts the system from a clean slate.

Also during the power-on stage, the PS2 mouse is initialized into *stream mode*. This mode produces a stream of packets that indicate mouse movement and button presses. The mouse used for this project and all projects presented in this thesis is the Microsoft Mouse Port Compatible Mouse 2.1A.

Finally, during the power-on stage the Tianma TM162VBA6 16x2 line LCD is initialized. During the initialization, the LCD is placed into 4-bit mode in order to interface with the ML505 project board. Also, the LCD is set for 2-line mode and the cursor blink is turned off.

Low Power Mode

The low power mode implements a single PicoBlaze processor that is programmed to control all three of the peripherals. This mode reduces power by using a

single processor but sacrifices the speed advantages of having multiple dedicated processors.

When the system first enters this mode, the processor reads the data from the EEPROM allowing system operation to continue in the same state as the previously configured mode. The keyboard and mouse clocks are connected to the interrupt port of the PicoBlaze processor. Whenever mouse movement is detected or a button is pressed or released, the peripherals begin sending data. When a key is pressed on the keyboard, the make-code for that key is transmitted. The PicoBlaze accumulates the data bits from the keyboard until it has received a complete make-code. When the key is released, the keyboard sends a break-code. Break-codes are ignored for all systems presented in this thesis. The make-codes are passed through a make-code-to-ascii converter and then written to the top line of the LCD. Packets from the mouse contain three bytes of data that are translated by the PicoBlaze. Mouse button clicks are displayed on the bottom left of the LCD and mouse X,Y movement is accumulated and displayed on the bottom center of the LCD. The following figure shows a block diagram of the low power mode configuration.

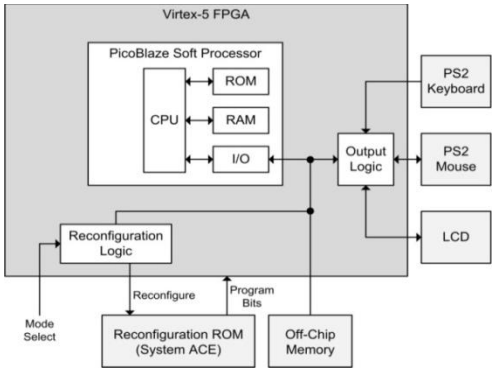

Figure 4.1: Block diagram for low power mode.

Xilinx *ChipScope* was used to monitor the internal signals of the FPGA to verify the operation of the system and the other systems presented in this thesis. ChipScope is an integrated logic analyzer core that is implemented on the FPGA. The measured signals are communicated to a PC GUI to view logic waveforms using a JTAG boundary scan interface. The following figure shows ChipScope waveforms of the address and instruction busses within the processor while in low power mode.

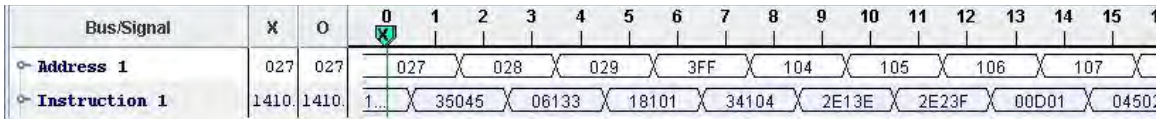

Figure 4.2: ChipScope waveforms showing the low power mode of operation.

Parallel Processing Mode

The parallel processing mode configures the FPGA to use three PicoBlaze processors that are each dedicated to controlling a single peripheral. The mouse and keyboard processors use interrupts to notify the LCD processor that new information has been received and needs to be displayed. Upon entering this mode, the LCD processor reads in the EEPROM data and maintains responsibility for keeping track of typed keyboard characters, keyboard cursor position, and mouse X,Y location. The following figure shows a block diagram of the parallel processing mode configuration.

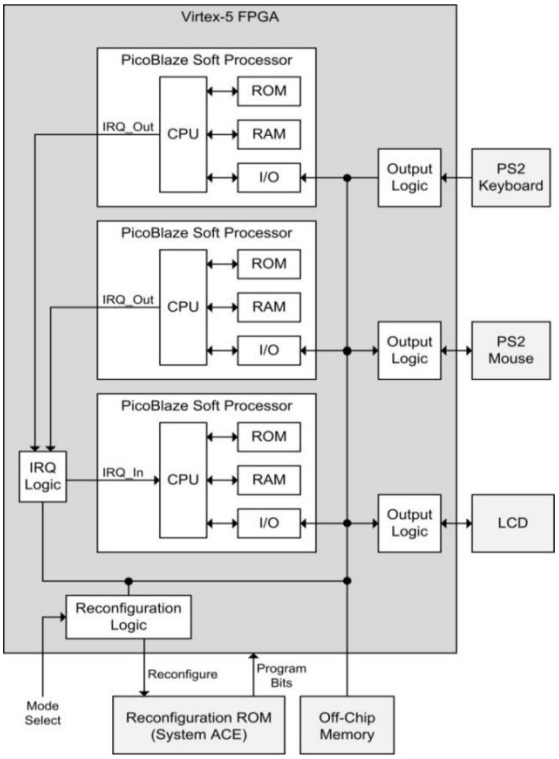

Figure 4.3: Block diagram for parallel processing mode.

Figure 4.4 shows a ChipScope screenshot of the address and instruction busses within the processors while in parallel processing mode. This screenshot shows the bus states when the mouse processor interrupts the LCD processor to indicate that new data has been received and needs to be written.

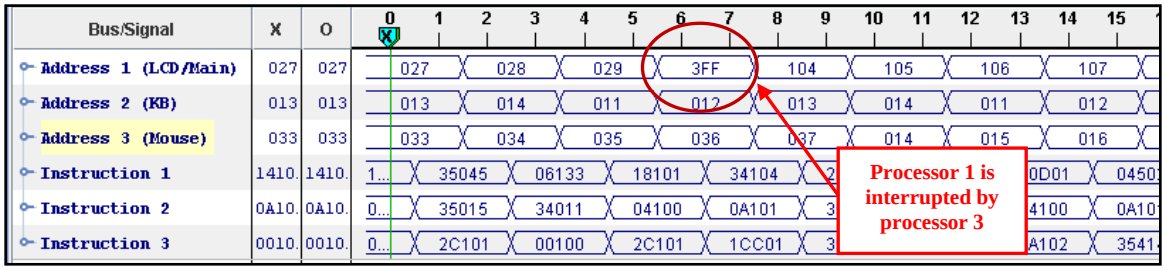

Figure 4.4: ChipScope waveforms showing the parallel processing mode of operation. Address 0x3FF indicates that the LCD processor has been interrupted by the mouse processor.

Radiation Tolerant Mode

The third mode of operation for this system is the radiation tolerant mode. In this mode the FPGA is configured to use three redundant PicoBlaze processors to control the peripherals. A TMR voting and recovery circuit is added to detect and repair faults. The TMR voting is performed on the address and instruction lines of the processor memory. The following figure shows a block diagram of the radiation tolerant mode configuration.

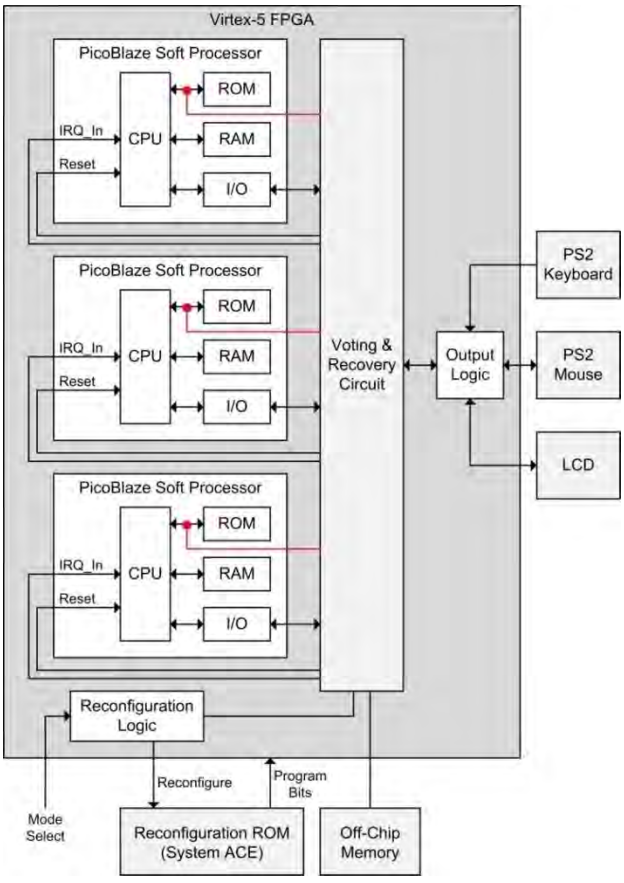

Figure 4.5: Block diagram for radiation tolerant mode.

When the system enters the radiation tolerant mode, all three processors read the variable data from the EEPROM and begin servicing the peripherals. If at any time the

voter detects a fault, the processors are interrupted to set an *error flag*. Upon completion of the main program loop, the processors poll the *error flag*. If the flag is set, the processors offload their variable data to the EEPROM and signal the recovery circuit when complete. The voter ensures that only valid information is written to the EEPROM. The recovery circuit then resets all three processors at the same time in order to synchronize them. The processors load the variable data from the EEPROM and begin servicing the peripherals once again. The following figure shows the soft fault recovery flow for the soft processors and the recovery circuitry.

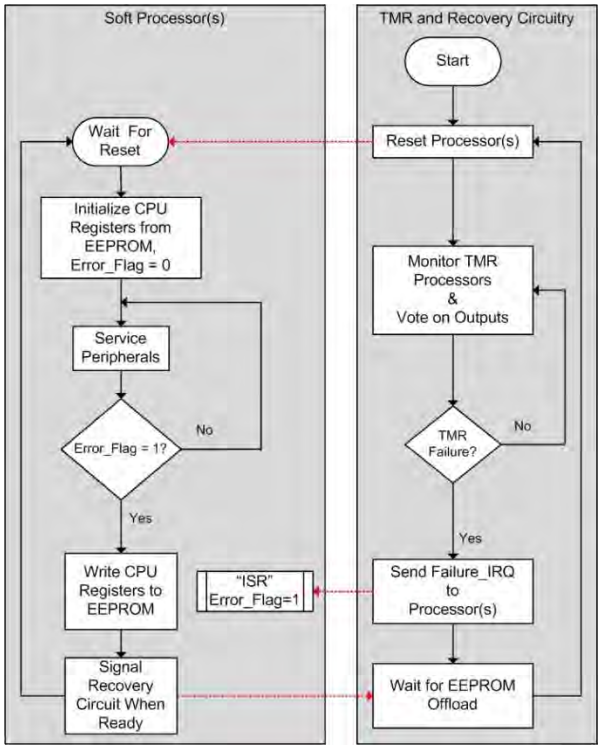

Figure 4.6: Soft fault recovery process during radiation tolerant mode.

Figure 4.7 shows ChipScope waveforms of the processor busses during a reset event. The recovery sequence was triggered by a fault on processor 1. If the fault is continuous, the processors are reset each time through the main loop allowing the peripherals to be serviced. This is shown in Figure 4.8.

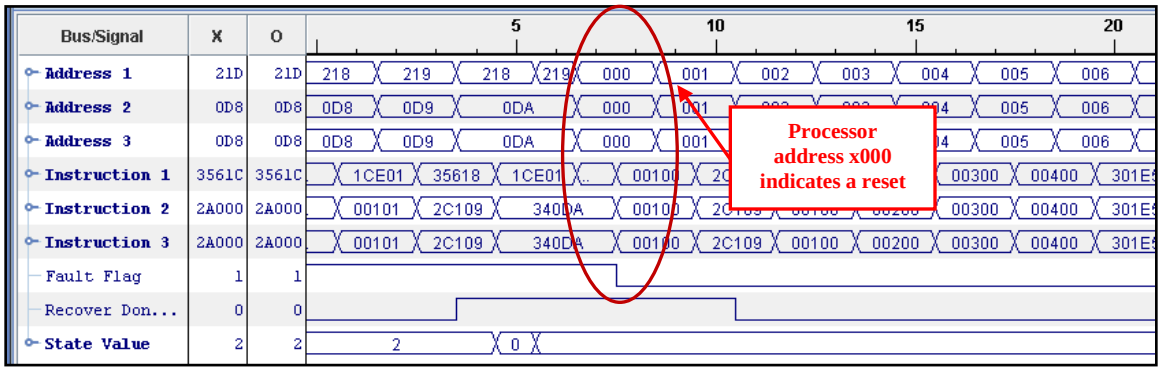

Figure 4.7: ChipScope waveforms showing the processor busses during a reset event. A soft fault detection on processor 1 triggered this recovery sequence.

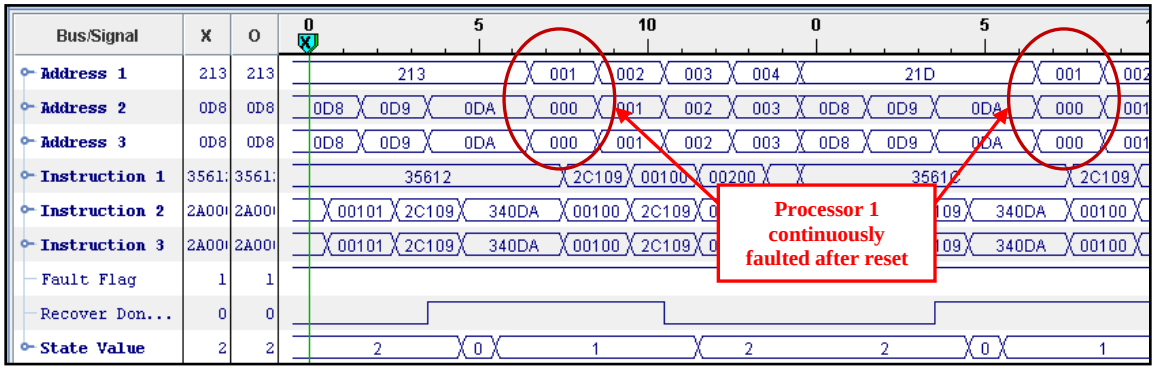

Figure 4.8: ChipScope waveforms showing a continuous fault on processor 1. Processor recovery is attempted each time through the main loop allowing the peripherals to be serviced and ensuring continuous system operation.

Full FPGA Reconfiguration

Xilinx FPGAs are programmed with configuration files called BIT files. BIT files can be downloaded to the device through the JTAG Boundary-Scan port using a USB JTAG programmer and the Xilinx iMPACT software. Virtex-5 FPGAs can also be programmed from on-board PROM or from the System ACE control device. The System ACE is a separate IC on the ML505 and XUPV5 project boards that connects to the FPGA JTAG interface and a Compact Flash (CF) memory card. BIT files are stored on the CF card and transferred to the JTAG port using the System ACE controller. Eight different BIT files can be stored on the CF card and dedicated DIP switches on the project board tell the System ACE which configuration to load during power-up.

The System ACE contains a Microprocessor Interface (MPU) that can be used for monitoring the status of and controlling the System ACE controller [32]. The MPU interface is composed of a set of registers that provide a means for communicating with resources in the System ACE controller. By connecting the MPU to the FPGA, this system tells the System ACE to load the desired BIT file and replace the current FPGA configuration. A simple state machine was created to interface with the MPU and switch between configuration modes. For demonstrative purposes, this design uses general purpose DIP switches on the ML505 to select the desired mode of operation. The following figure shows the full FPGA reconfiguration flow for all modes.

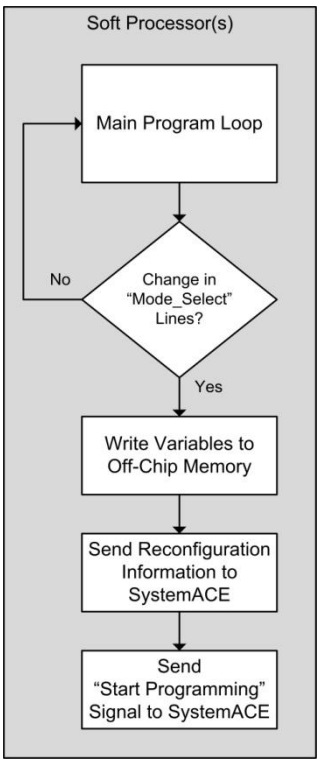

Figure 4.9: Reconfiguration sequence for all modes.

System Performance

One of the main timing parameters for this system is the time it takes to recover from faults. Using a Virtex-5 FPGA with a system clock of 100MHz, it took 23.2ms to recover a faulted processor. This time is dependent on the amount of variable information that is stored off-chip during each recovery. Equations 4.1 and 4.2 give the amount of time it takes to write to and read from the off-chip memory via I^2C protocol.

$$t_{OCMW} = \frac{(n+2)\times 9}{f_{SCL}} + t_{WR} \times \left(\left\lceil \frac{n}{N} \right\rceil - 1 \right) \tag{4.1}$$

$$t_{OCMR} = \frac{(n+3)\times 9}{f_{SCL}} \tag{4.2}$$

In equations 4.1 and 4.2, n is the number of bytes to read or write, f_{SCL} is the I^2C clock rate, t_{WR} is the page write time, and N is the page size. For this system, f_{SCL} is 400 kHz, t_{WR} is 5ms and N is 16. During a fault recovery, t_{SM} is the amount of time it takes the recovery state machine to execute.

Another important timing parameter is the amount of time it takes to reconfigure the FPGA. This value is given is in equation 4.3.

$$t_{sysace} = \frac{k}{f_{SACLK}} + t_{RC} \tag{4.3}$$

In equation 4.3, k is the number of MCU register writes to the System ACE, f_{SACLK} is the clock frequency of the System ACE, and t_{RC} is the time it takes the System ACE to reconfigure the FPGA. For this design k is 4, f_{SACLK} is 33MHz, and t_{RC} is 504ms. This time is the same regardless of which configuration is being loaded due to the FPGA needing all logic to be configured each time it is programmed. Table 4.1 shows the timing impact of the design and table 4.2 shows the amount of FPGA resources used for each of the three configurations used in this system.

Table 4.1: Timing impact of important system events for the full FPGA reconfigurable computing system.

System Event	Timing Dependencies	Total Time
Initial Power On	$t_{mouse} + t_{LCD} + t_{OCMW} + t_{sysace} + t_{OCMR}$	0.7ms + 1.67ms + 21.6ms + 504.0ms + 1.6ms = **529.6ms**
Mode Configuration Change	$t_{OCMW} + t_{sysace} + t_{OCMR}$	21.6ms + 504.0ms + 1.6ms = **527.2ms**
Fault Recovery	$t_{OCMW} + t_{SM} + t_{OCMR}$	21.6ms + 30.0ns+ 1.6ms = **23.2ms**

Table 4.2: FPGA resource usage for the three configuration modes.

FPGA Resources	Resources Used		
	Parallel Processing	Low Power	Radiation Tolerant
Number of Slice Registers	261	102	355
Number of Slice LUTs	470	194	562
Number of occupied Slices	187	96	280
Number of LUT Flip Flop pairs	498	210	635

System Summary

This system provides a platform that can fully reconfigure itself for three different modes of operation. If radiation is present, the system configures into a TMR architecture with circuitry to recover faulted processors. Soft fault recovery was accomplished in 23.2ms. When not in a radiation environment, the system configures into a high performance, parallel processing mode or a single-processor low power mode. A full FPGA mode reconfiguration was accomplished in 527.2ms.

3+13 PICOBLAZE SYSTEM

The 3+13 PicoBlaze system is a many-core system that contains 16 homogenous tiles each containing a PicoBlaze soft processor. The system is implemented on the Xilinx XUPV5-LX110T project board and is designed to control basic peripherals (keyboard, mouse, and LCD). Active processors are configured in a TMR scheme to mitigate soft faults caused by ionizing radiation. In the event that one of the active processors becomes TID damaged, a spare replacement is brought online. The system also has the ability to partially reconfigure each of the 16 tiles to repair SEFIs. A GUI is used to emulate TID and SEFI fault conditions. Figure 5.1 shows a block diagram for the 3+13 PicoBlaze many-core system.

Soft Faults

Soft fault detection and recovery is implemented exactly the same for this system as it is for the full FPGA reconfigurable computing system. Active processors are placed in a TMR scheme and a voter monitors the address and data lines between the microprocessor and its instruction memory. When a fault is detected, a recovery state machine interrupts the processors and sets an *error flag*. After the processors finish servicing the peripherals, the *error flag* is checked. If it is set, they store their variable data to the off-chip EEPROM. The recovery state machine resets the processors and the variable data is then loaded back in. Soft faults are emulated by pressing push buttons on the project board which induce SEUs on various address and data lines between the processor and memory.

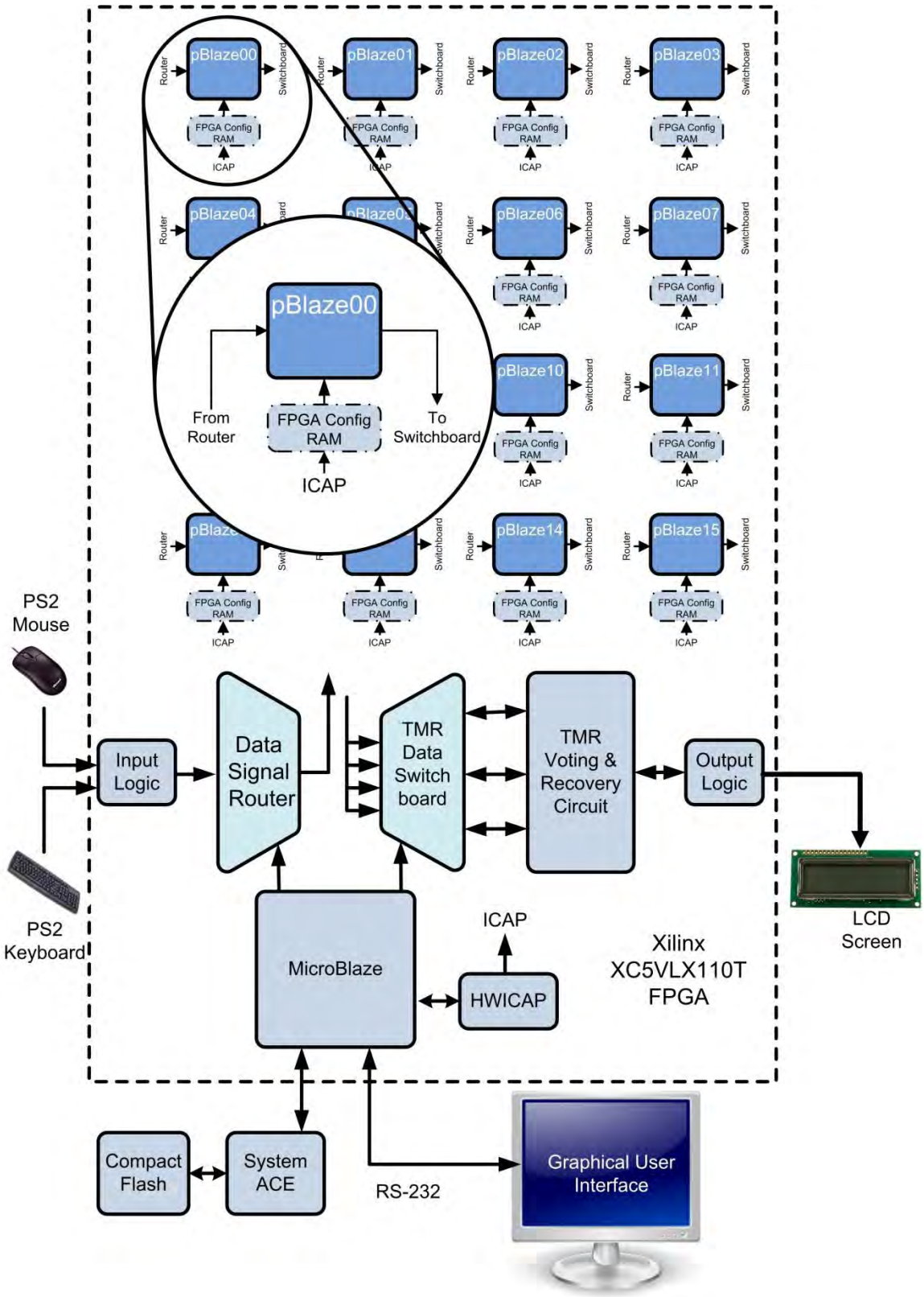

Figure 5.1: A block diagram of the 3+13 PicoBlaze (pBlaze) many-core system.

Graphical User Interface

A GUI was developed to monitor which of the 16 soft processors are active at any given time. The GUI was developed in C# (C sharp) using the Microsoft Visual Studio development environment and communicates with the system via RS232 protocol. The GUI allows the user to inject SEFI faults into the configuration SRAM. Faulted tiles can be manually reconfigured or the GUI can be set for automatic reconfiguration. Tiles can also be "permanently damaged" to emulate a TID failed tile. The following figure shows a screenshot of the GUI having processors 0, 1, and 2 active with the other processors available as spares.

Figure 5.2: Screenshot of the 3+13 PicoBlaze GUI (Blue = Active, Red = Corrupted, Gray = Spare).

Spatial Avoidance of TID Failures

When the system detects an error that is not caused by a soft fault, a new tile is brought online in an attempt to spatially avoid the fault. For this prototype, spatial avoidance is triggered by injecting faults using the GUI or by monitoring back-to-back faults on a processor that has undergone the soft fault recovery sequence.

Spare tiles are enabled using the same process flow as the soft fault recovery procedure; however, upon reset the recovery circuit selects a spare tile to replace the faulted processor. The new processor is initialized with the same variable data that the two remaining good processors are loaded with. In this way, a TMR triplet can be formed with any of the 16 processors available to the system. The system contains a log of which processors are available as spares and which ones are marked as *damaged*.

The following figure shows ChipScope measurements of the system after it has undergone a fault on processor 2. The system brings processor 3 online to form a TMR triplet.

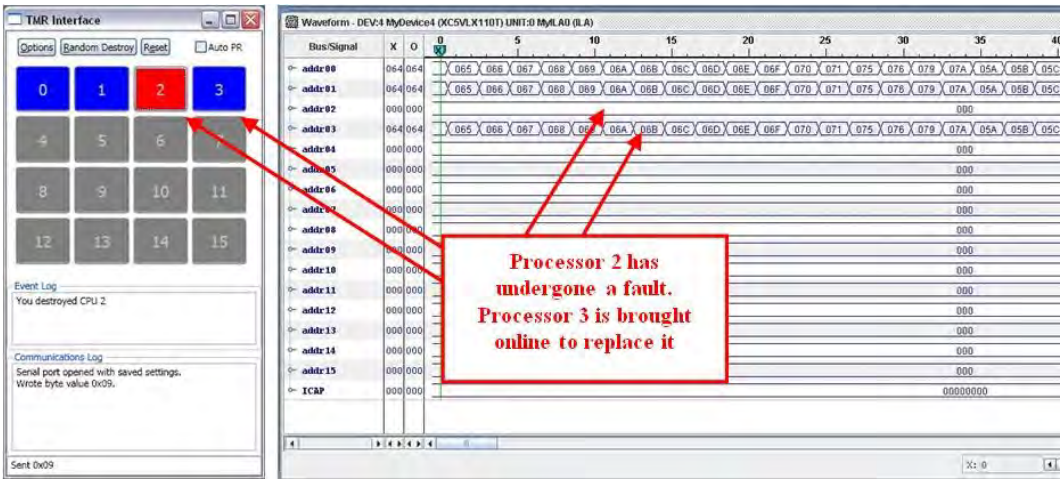

Figure 5.3: Processor 2 has undergone a fault and the system has brought on processor 3 to form the TMR triplet.

The following figure shows ChipScope measurements after multiple faults have occurred in the system. Processors 2, 4, 6, and 7 have been faulted and the system activated processors 3, 5, and 8 to form the TMR triplet.

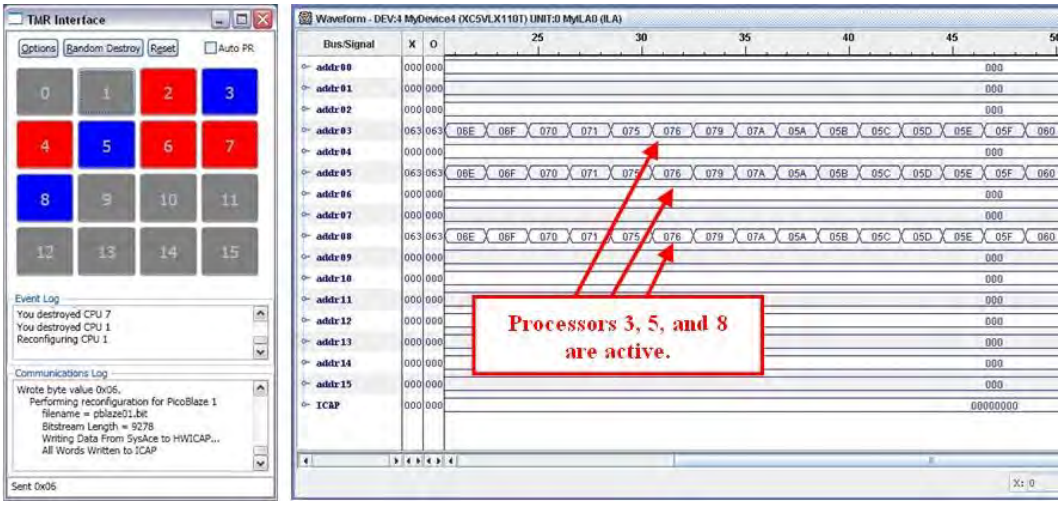

Figure 5.4: The system is spatially avoiding faulted processors 2, 4, 6, and 7 by bringing online processors 3, 5, and 8.

Partial FPGA Reconfiguration

The Virtex-5 contains an Internal Configuration Access Port (ICAP) that provides direct read/write access to the configuration SRAM of the FPGA. This allows designated tiles of the FPGA to be dynamically reconfigured without effecting operation of the remaining FPGA areas. Partial bitstream data that is written to the ICAP comes from partial BIT files created using the Xilinx PlanAhead software. Partial BIT files are stored on a Compact Flash card and read through the System ACE using a dedicated MicroBlaze soft-processor. This processor, referred to as the ICAP processor, then writes the partial bitstream to the ICAP through the *HWICAP* core. The HWICAP core

contains a series of FIFO blocks which handle retiming the bitstream so it can be driven into the ICAP.

Using Xilinx ISE 9.2 with the PR_14 patch, the static system design is created with the ICAP processor included. PR modules are synthesized as black-box components and are connected to the static design using bus-macros. Bus-macros ensure fixed routing channels between the static design and PR modules [28]. With the exception of clocks, every signal connected to the static design must have a bus-macro. PR modules are synthesized separately from the static design without added IO buffers.

The synthesized netlists are then brought into Xilinx PlanAhead 10.1 and the project is set to PR mode. PlanAhead provides a graphical representation of FPGA resources as they are laid out on the IC and allows the developer to manually place circuit design components throughout the device. PR modules are grouped into physical blocks (Pblocks) and then laid out on the FPGA floorplan into a partial reconfiguration region (PRR). For the Virtex-5 LX110T FPGA, PR tiles are 20 CLB high and 1 CLB wide. PRRs can contain as many full PR tiles as needed to implement resource requirements of the PR module. The PlanAhead PR flow requires that all bus-macros, clock buffers, and digital clock management (DCM) units be manually placed on the FPGA floorplan. For the Virtex-5, bus-macros are placed anywhere inside their corresponding PRR. Once the floorplan passes the PR design rules check (DRC), the place and route process is ran on the static design and PR modules separately. After completion, the PR assemble process is called to create the partial BIT files for the PR modules and merge the static and PR design to create one full BIT file.

SEFI Recovery Using PR

When a spare processor is brought online, a repair attempt is made on the faulted tile using partial reconfiguration. This recovery process will mitigate SEUs that have occurred in the configuration SRAM of the FPGA. The recovery sequence is performed independently of the normal system operation using the ICAP processor. After reconfiguration, the tile is entered back into the system log as an available spare.

The processor tiles are laid out in a 4 by 4 grid on the Virtex-5 LX110T floorplan. The following figure shows the floorplan as laid out in PlanAhead.

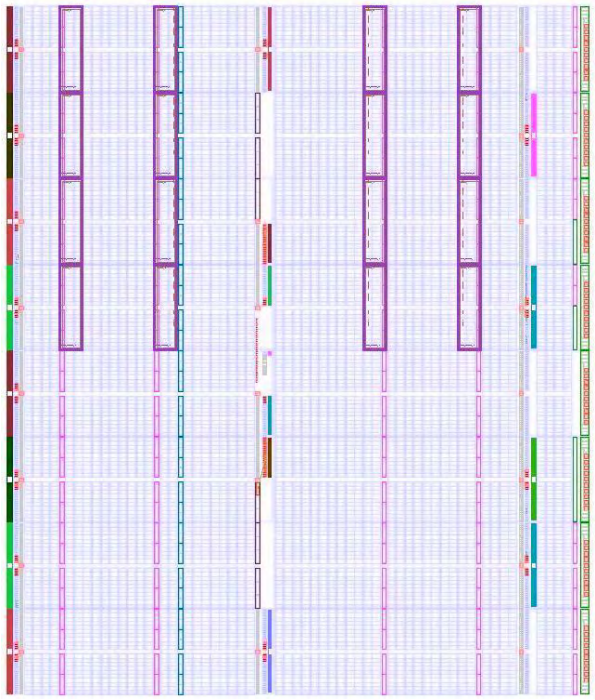

Figure 5.5: 3+13 Picoblaze system floorplan on the Virtex-5 LX110T.

Tile sizes for the PicoBlaze processor were found to be 24 CLBs and 1 BRAM. The PR tools require that 20 CLBs or 4 BRAMs be reconfigured at a time. Thus, the

final tile size for the PicoBlaze processor is 40 CLBs and 4 BRAMs. The following figure shows a zoom of the floorplan for one of the PR tiles.

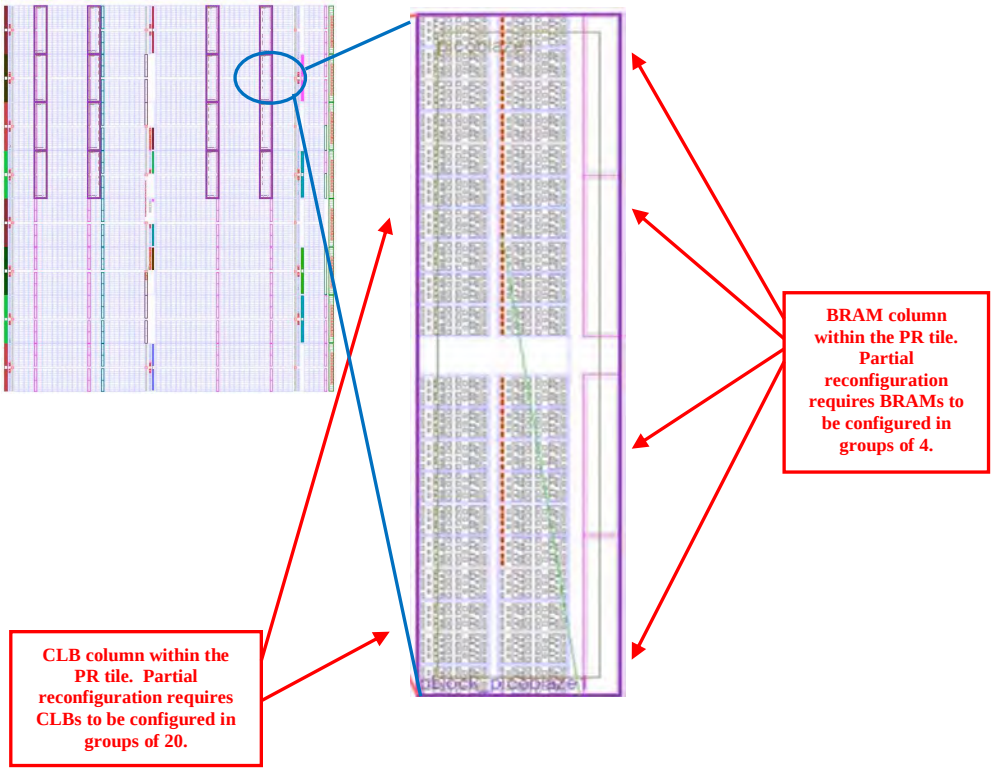

Figure 5.6: Floorplan zoom of a PicoBlaze PR tile.

The PR tools generate a unique configuration bitstream file for each of the 16 processor tiles. For this system, the files are 31.6k bytes in size. Each of the partial BIT files contains a header with unique information about the file, the bitstream length, a constant 54 bytes of data, and then the unique bitstream. The following figure shows the partial BIT file for processor 0 opened in a hex editor to display the bitstream.

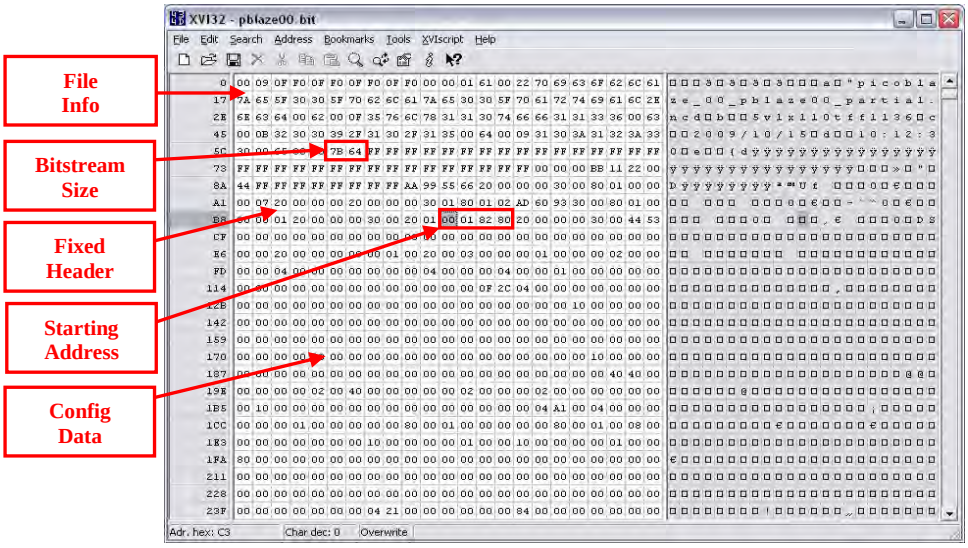

Figure 5.7: Hex translation of the partial BIT file for PicoBlaze tile 0.

Byte number xC3 in the partial BIT file contains the unique starting address of the bitstream that corresponds to a location in the configuration SRAM. The following table shows the configuration SRAM starting addresses for all 16 partial BIT files.

Table 5.1: Configuration SRAM starting addresses for the PicoBlaze tiles.

PicoBlaze Tile	Configuration SRAM Starting Address (HEX)
0	x00018280
1	x00018780
2	x00019400
3	x00019980
4	x00010280
5	x00010800
6	x00011400
7	x00011980
8	x00008280
9	x00008800
10	x00009400
11	x00009980
12	x00000280
13	x00000800
14	x00001400
15	x00001980

The ICAP was observed with ChipScope to monitor the partial reconfiguration occurring in the background while the active processors are running. Figure 5.8 shows partial reconfiguration for processor tile 0 while processors 3, 5, and 8 are running in a TMR triplet. In this case, processors 0, 1, 2, 4, 6, and 7 are faulted with a SEFI. The GUI indicates that processor 0 has been repaired and is now available as a spare. The ICAP bitstream measured in ChipScope shows the configuration SRAM starting address in sample 24 (x00018280).

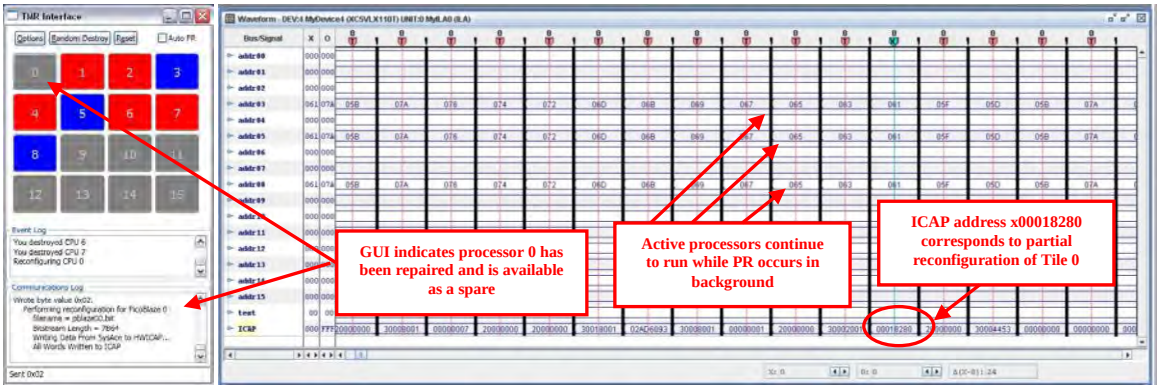

Figure 5.8: Recovery from a SEFI on PicoBlaze processor 0 using partial reconfiguration.

Figure 5.9 shows partial reconfiguration for processor tile 1 while processors 3, 5, and 8 are active in a TMR triplet. Appendix A contains figures showing partial reconfiguration for all PicoBlaze processor tiles. These figures also show the partial BIT file hex translation for the corresponding processor tile.

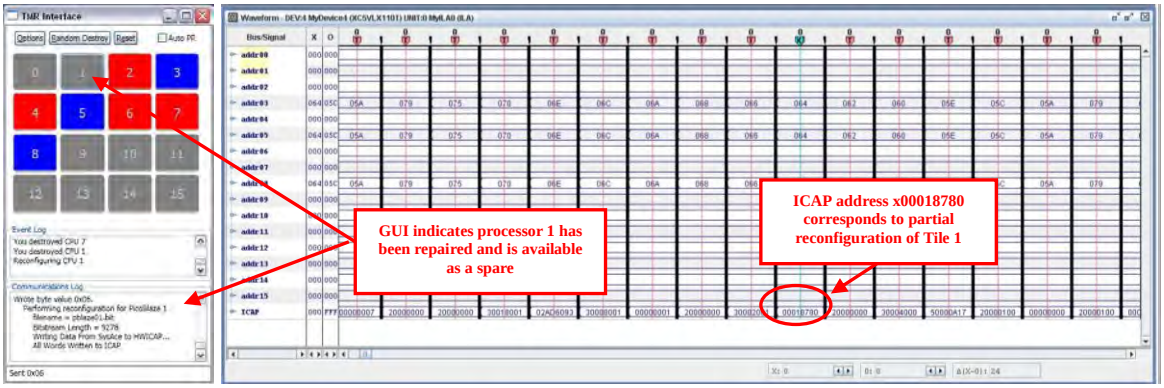

Figure 5.9: Recovery from a SEFI on PicoBlaze processor 1 using partial reconfiguration.

System Performance

The amount of time it takes to recover from a soft fault is exactly the same is it is for that of the radiation tolerant mode in the full FPGA reconfigurable computing system. Using equations 4.1 and 4.2 and a 100MHz system clock, the soft fault recovery time is 23.2ms. Bringing a spare processor online to spatially avoid TID requires the same variable data offload and reload from the EEPROM. Thus, the time it takes to bring a spare processor online is also 23.2ms.

The amount of time it takes to perform partial reconfiguration depends on the size of the tile bitstream. For this system, the tile bitstreams are 31.6k bytes in size. Partial reconfiguration for each tile takes approximately 200 clocks for each byte of configuration data. This corresponds to a PicoBlaze processor tile PR time of 66ms using a 100MHz system clock.

System Summary

This system provides a platform that uses three fault mitigation techniques to recover from radiation induced failures on an FPGA. The FPGA is partitioned into equally sized tiles each containing a PicoBlaze soft processor. SEUs in the FPGA fabric are mitigated using a reset sequence while SEUs in the FPGA configuration SRAM are mitigated using partial reconfiguration of the tile. Finally, TID damage is mitigated using spatial avoidance of the effected region. Soft fault recovery and TID avoidance were achieved in 23.2ms. Partial reconfiguration of a single tile was achieved in 66ms.

3+1 MICROBLAZE SYSTEM

The 3+1 MicroBlaze system is a many-core system that contains four homogenous tiles each containing a MicroBlaze soft processor. The MicroBlaze is a more sophisticated soft core processor than the PicoBlaze and provides a higher level of processing power. The system is implemented on the Xilinx XUPV5-LX110T project board and is designed to control basic peripherals (keyboard, mouse, and LCD) while executing a Turing machine to show computational effectiveness. Active processors are configured in a TMR scheme to mitigate soft faults caused by ionizing radiation. In the event that one of the active processors becomes TID damaged, the spare replacement is brought online. The system also has the ability to partially reconfigure each of the four tiles to repair SEFIs. A GUI is used to inject TID and SEFI fault conditions. Figure 6.1 shows a block diagram for the 3+1 MicroBlaze many-core system.

MicroBlaze Soft-Core Processor

This system uses the MicroBlaze soft-core processor to provide a higher level of processing power over the 3+13 PicoBlaze system. The MicroBlaze is a 32-bit RISC Harvard architecture soft processor provided by Xilinx to be used under license with the Xilinx Platform Studio (XPS) development software. For this system, the MicroBlaze processor uses 64kB of BRAM and contains a hardware multiplier for increased signal processing capability.

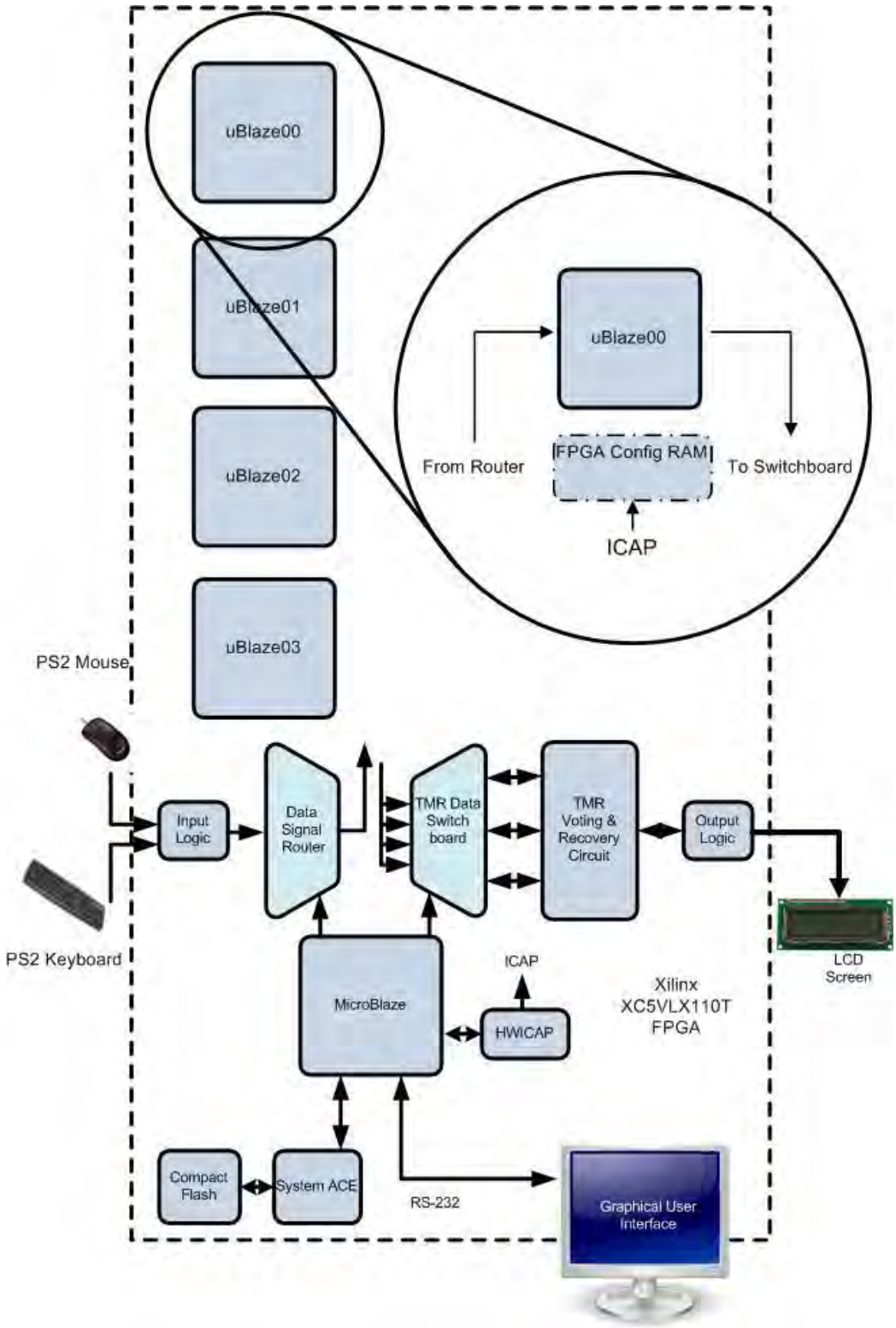

Figure 6.1: A block diagram of the 3+1 MicroBlaze (uBlaze) many-core system.

The MicroBlaze is programmed in the C language using the XPS Software Development Kit (SDK). This program compiles the C code into an Executable and Linkable Format (ELF) file that contains the MicroBlaze hardware language (opcodes and operands). This file is merged into the configuration BIT file using the Xilinx Data2Mem tool.

Soft Faults

Soft fault detection and recovery is implemented exactly the same for this system as it is for the full FPGA reconfigurable computing system and the 3+13 PicoBlaze many-core computing system. The amount of variable data stored to the off-chip EEPROM for this system is less than the other systems. The MicroBlaze has the ability to change the amount of allocated RAM to accommodate the application requirements. Therefore, only necessary variables are stored, unlike the PicoBlaze systems that store their entire 64 bytes of variable RAM.

Graphical User Interface

The GUI from the 3+13 PicoBlaze many-core computing system was modified to monitor which of the four soft processors are active at any given time. The GUI offers the same functionality as before, allowing the user to inject SEFI faults into the configuration SRAM. Faulted tiles can be manually reconfigured or the GUI can be set for automatic reconfiguration. Tiles can also be permanently damaged to emulate a TID

failed tile. The following figure shows a screenshot of the modified GUI having processors 0, 1, and 2 active with processor 3 available as a spare.

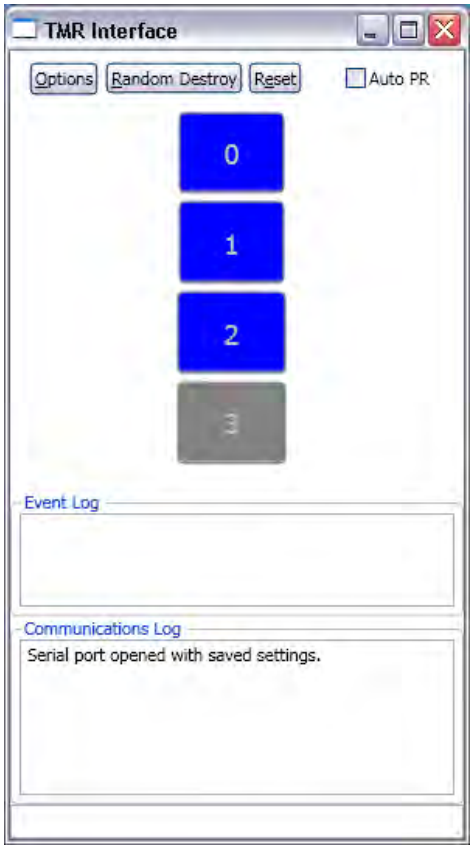

Figure 6.2: Screenshot of the 3+1 MicroBlaze GUI (Blue = Active, Red = Corrupted, Gray = Spare).

Spatial Avoidance of TID Failures

Spatial avoidance of TID failures is achieved exactly the same for this system as it is for the 3+13 PicoBlaze system. The following figure shows ChipScope measurements of the system after it has undergone a fault on processor 2. The system brings processor 3 online to form the TMR triplet.

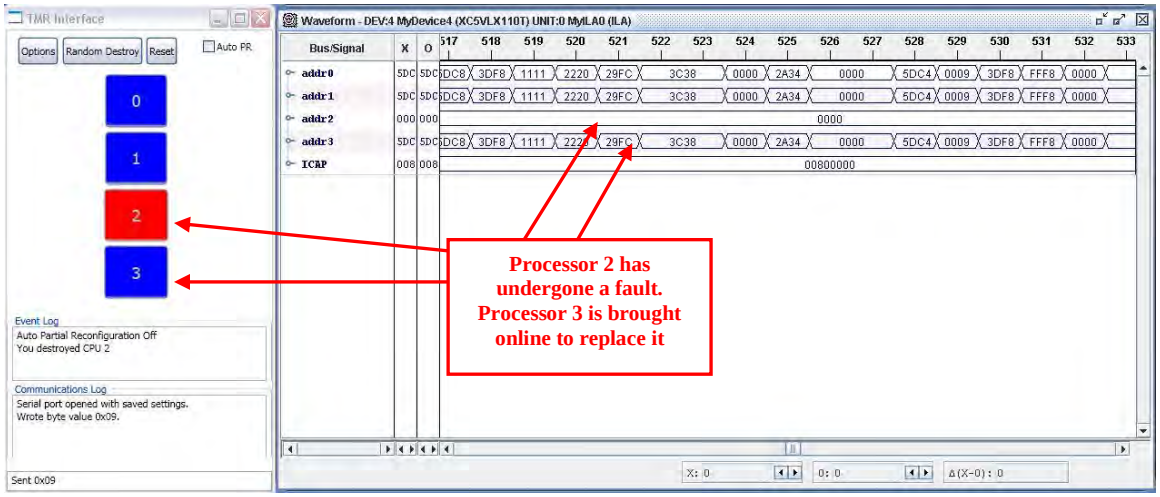

Figure 6.3: Processor 2 has undergone a fault and the system has brought on processor 3 to form the TMR triplet.

SEFI Recovery Using PR

As in the 3+13 PicoBlaze system, when a spare processor is brought online, a repair attempt is made on the faulted tile using partial reconfiguration. This recovery process will mitigate SEUs that have occurred in the configuration SRAM of the FPGA. The recovery sequence is performed independently of the normal system operation using the ICAP processor. After reconfiguration, the tile is entered back into the system log as an available spare.

The processor tiles are laid out in a 4 by 1 column on the Virtex-5 LX110T floorplan. The following figure shows the floorplan as laid out in Xilinx PlanAhead.

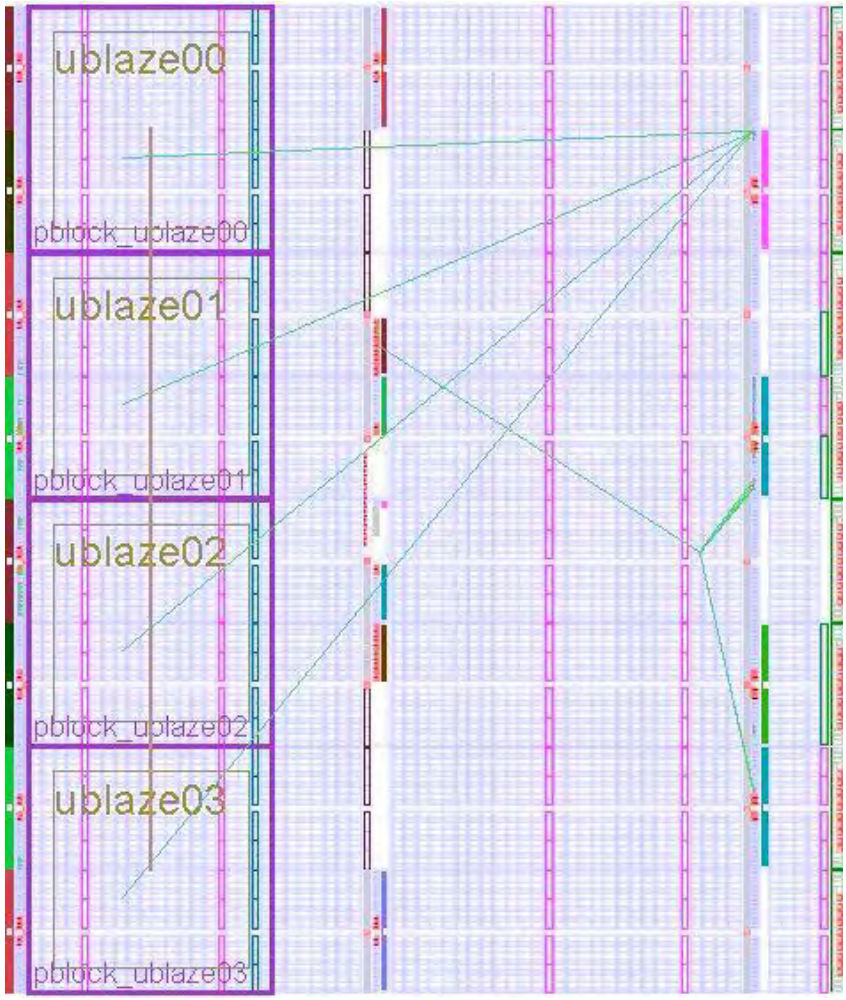

Figure 6.4: 3+1 MicroBlaze system floorplan on the Virtex-5 LX110T.

The MicroBlaze processors used in this system require 16 BRAMs, three DSP slices, and roughly 280 CLBs; however, due to PR tool constraints and routing limitations, the final tile size for the MicroBlaze processor is 16 BRAMs, 16 DSP slices, and 680 CLBs. The following figure shows a zoom of the floorplan for one of the PR tiles.

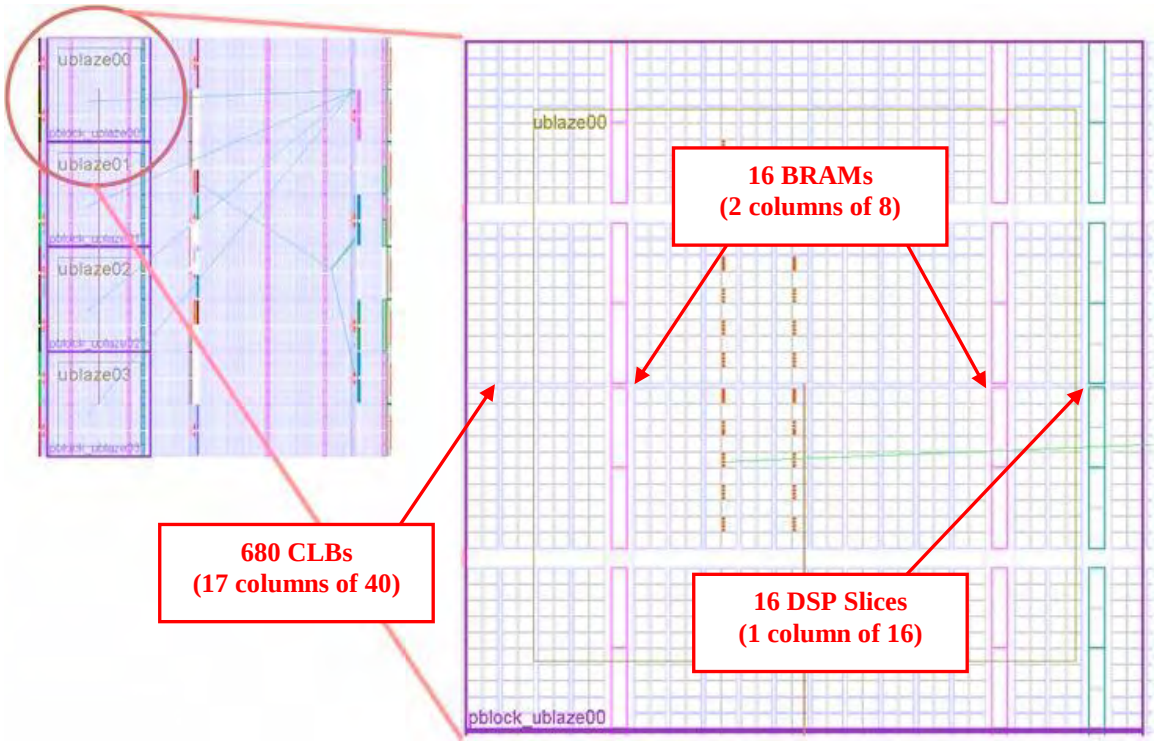

Figure 6.5: Floorplan zoom of a MicroBlaze PR tile.

The PR tools generate a unique configuration bitstream file for each of the four processor tiles. For this system, the files are 262.4k bytes in size. Each of the partial BIT files contains a header with unique information about the file, the bitstream length, a constant 54 bytes of data, and then the unique bitstream. The following figure shows the partial BIT file for processor 0 opened in a hex editor to display the bitstream.

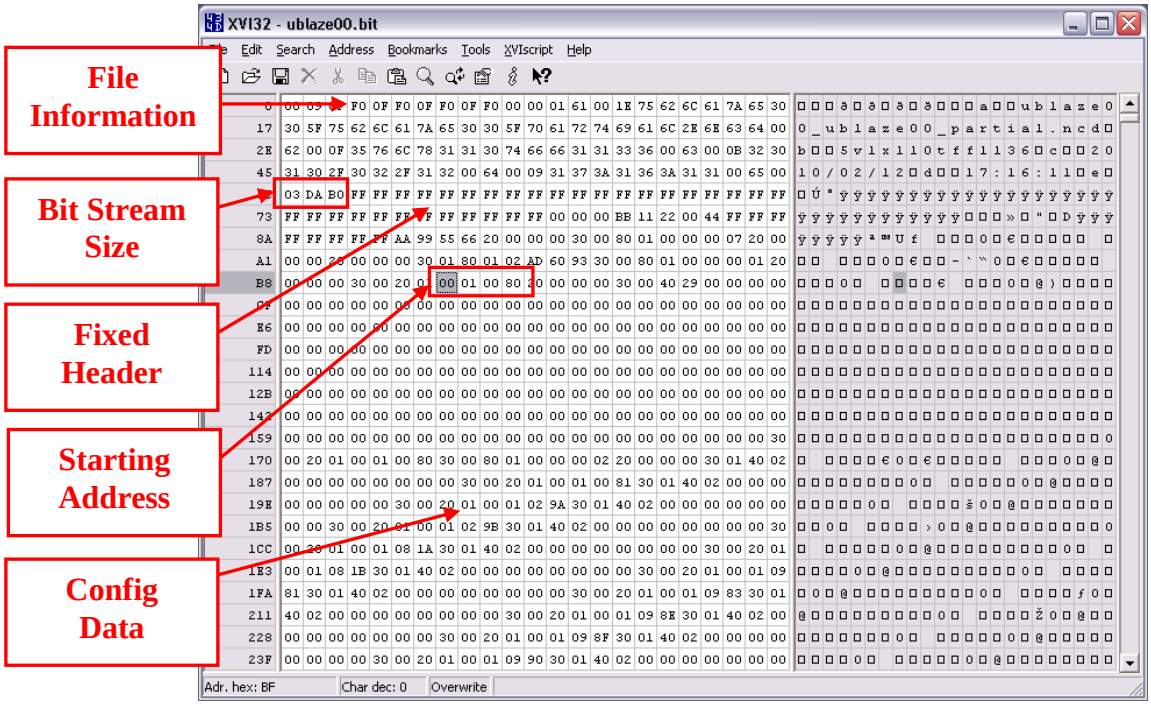

Figure 6.6: Hex translation of the partial BIT file for MicroBlaze tile 0.

Byte number xBF in the partial BIT file contains the unique starting address of the bitstream that corresponds to a location in the configuration SRAM. The following table shows the configuration SRAM starting addresses for all four partial BIT files.

Table 6.1: Configuration SRAM starting addresses for the MicroBlaze tiles.

MicroBlaze Tile	Configuration SRAM Starting Address (HEX)
0	x00010080
1	x00000080
2	x00100080
3	x00110080

The ICAP was observed with ChipScope to monitor the partial reconfiguration occurring in the background while the active processors are running. Figure 6.7 shows partial reconfiguration for processor tile 0 while processors 1, 2, and 3 are running in a TMR triplet. In this case, processor 0 has been faulted with a SEFI. The GUI indicates that processor 0 has been repaired and is now available as a spare. The ICAP bitstream measured in ChipScope shows the configuration SRAM starting address in sample 24 (x00010080). Appendix B contains figures showing partial reconfiguration for all MicroBlaze processor tiles. These figures also show the partial BIT file hex translation for the corresponding processor tile.

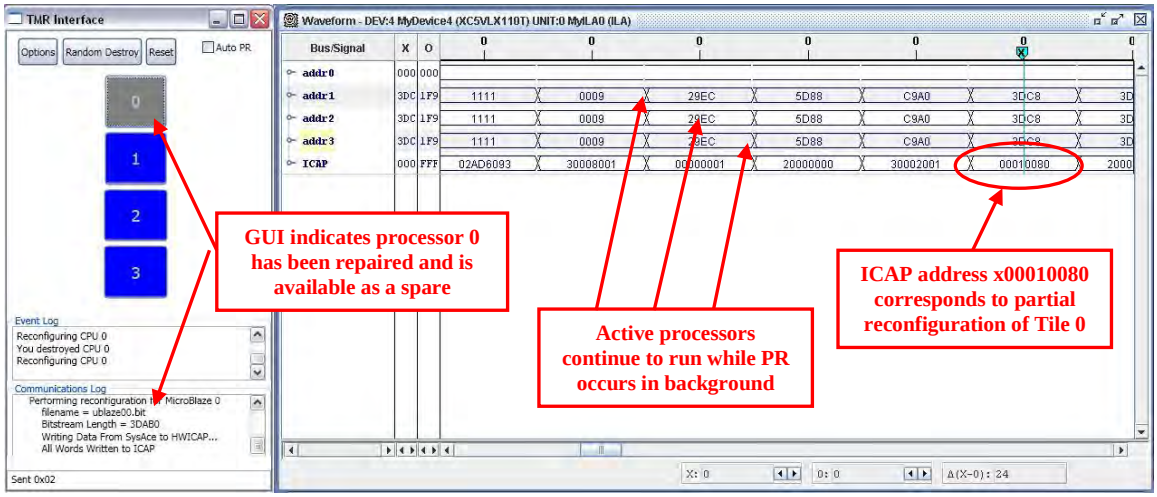

Figure 6.7: Recovery from a SEFI on MicroBlaze processor 0 using partial reconfiguration.

System Performance

The amount of time it takes to recover from a soft fault can be found using equations 4.1 and 4.2. For an f_{SCL} of 400 kHz, the soft fault recovery time is 11.9ms.

Bringing a spare processor online to spatially avoid TID requires the same variable data offload and reload from the EEPROM. Thus, the time it takes to bring a spare processor online is also 11.9ms.

The amount of time it takes to perform partial reconfiguration depends on the size of the tile bitstream. For this system, the tile bitstreams are 262.4k bytes in size. Partial reconfiguration for each tile takes approximately 200 clocks for each byte of configuration data. This corresponds to a MicroBlaze processor tile PR time of 525ms using a 100MHz system clock.

System Summary

This system provides a platform that uses three fault mitigation techniques to recover from radiation induced failures on an FPGA. The FPGA is partitioned into equally sized tiles each containing a MicroBlaze soft processor. The MicroBlaze provides greater processing power than the PicoBlaze soft-core processor used in the 3+13 PicoBlaze computing system. SEUs in the FPGA fabric are mitigated using a reset sequence while SEUs in the FPGA configuration SRAM are mitigated using partial reconfiguration of the tile. Finally, TID damage is mitigated using spatial avoidance of the effected region. Soft fault recovery and TID avoidance were achieved in 11.9ms. Partial reconfiguration of a single tile was achieved in 525ms.

FUTURE WORK

System Improvements

A major downfall of the 3+13 PicoBlaze system and the 3+1 MicroBlaze system is that the ICAP processor uses many FPGA resources and has no radiation fault tolerance. This processor will be replaced with a simple state machine that reads the PR bitstream data from the System ACE and writes it to the ICAP. Radiation fault mitigation techniques, such as TMR, can then be applied to the state machine. Also, the TMR voter and recovery state machine offer no radiation fault tolerance. These circuits can be placed in a TMR scheme or moved to an off-chip radiation hardened device. Fault injection needs to be added that actually flips a bit in the configuration SRAM instead of emulating the corruption with the GUI. Dynamic bitstream relocation would allow only the corrupted portion of a destroyed processor tile to be moved instead of moving the entire tile. This would save on FPGA resource usage. For demonstration purposes, the 3+1 MicroBlaze system will be adapted to handle streaming video from a webcam to a VGA monitor. Digital Signal Processing (DSP) algorithms, such as Fast Fourier Transforms (FFTs), will then be applied to the streaming video to demonstrate real-time computational effectiveness.

Spatial Radiation Sensor

Integrating these systems with a spatial radiation sensor will provide information of radiation strikes before system faults occur. The radiation fault tolerant mode of the

full FPGA reconfigurable computing system will be deployed based on sensor activity. The 3+13 PicoBlaze system and the 3+1 MicroBlaze system will monitor the sensor information and bring online spare processors to spatially avoid radiation strikes before a fault is detected.

Scrubber

Adding a scrubber to the system will add enhanced SEFI fault mitigation. The scrubber will scan the FPGA configuration SRAM in a sequential fashion repairing configuration errors. It will be made spatially aware by integrating it with the radiation sensor. If a radiation strike is detected, the scrubber will immediately begin scanning in the affected area to reduce latency between detection and correction of SEFIs. Radiation sensor integration will also reduce power by only running the scrubber when radiation events are detected.

Ion Chamber Testing

The 3+13 PicoBlaze system and the 3+1 MicroBlaze system will be tested in an ion chamber to yield the fault tolerance of the systems. Radiation tests on various ion sizes, energy levels, and incident angles will be tested on both systems. The testing will be performed at the Texas A&M Radiation Effects Facility.

CONCLUSION

By exploiting the reconfigurability of commercial SRAM-based FPGAs, novel fault mitigation techniques were developed to provide fault tolerant computing systems that can be used for aerospace applications. Soft-core processors were used in conjunction with dedicated recovery circuitry to mitigate the effects of ionizing radiation on FPGAs. These systems offer the ability to mitigate all three of the major sources of radiation faults in FPGA-based systems (SEUs, SEFIs, and TID).

The full FPGA reconfigurable computing platform offers the ability to mitigate SEUs in the event of harsh radiation bombardment. It also has the ability to switch into a low power mode to conserve power or a parallel processing mode to provide increased computation. SEU recovery was achieved in 23.2ms and a full FPGA mode reconfiguration was accomplished in 527.2ms.

The 3+13 PicoBlaze and 3+1 MicroBlaze systems offer novel radiation mitigation capabilities. They provide a solution to TID failure and, unlike TID hardened parts, they provide SEU fault mitigation as well. Also, a partial reconfiguration strategy is added to mitigate SEU faults in the FPGA configuration SRAM. For the 3+13 PicoBlaze system, soft fault recovery and TID avoidance were achieved in 23.2ms. Partial Reconfiguration of a single PicoBlaze processor tile was achieved in 66ms. For the 3+1 MicroBlaze system, soft fault recovery and TID avoidance were achieved in 11.9ms. Partial reconfiguration of a single MicroBlaze processor tile was achieved in 525ms.

As the aerospace community looks toward integrating more complex computing systems into flight applications, fault conditions caused by ionizing radiation need to be

addressed. By providing a variety of radiation fault mitigation techniques, the systems presented in this thesis provide an attractive option for implementing FPGA-based flight systems.

REFERENCES CITED

[1] C. Claeys, E. Simoen, “Radiation Effects in Advanced Semiconductor Materials and Devices”, Berlin Heidelberg, ISBN 3-540-43393-7, Springer-Verlag, 2002.

[2] Clive Maxfield, “The Design Warrior’s guide to FPGAs”, Burlington, Newnes, 2004.

[3] T.J. Todman, G.A. Constantinides, S.J.E. Wilton, O. Mencer, W. Luk and P.Y.K. Cheung, “Reconfigurable computing: architectures and design methods”, IEE Proc.-Comput.Digit. Tech., Vol 152, No.2, March 2005.

[4] Chen Chang, John Wawrzynek, and Robert W. Brodersen, “BEE2: A High-End Reconfigurable Computing System”, IEEE Design & Test of Computers, March-April 2005

[5] Keith Underwood, “FPGAs vs. CPUs: Trends in Peak Floating-Point Performance”, FPGA'04, February 2004, Monterey, CA, USA.

[6] Scott Hauck, “The Roles of FPGAs in Reprogrammable Systems”, Proceedings of the IEEE, Vol. 86, No. 4, pp. 615-638, April, 1998.

[7] Gadi Taubenfeld, “Synchronization Algorithms and Concurrent Programming”, Pearson/Prentice Hall, 2006.

[8] A. Holmes-Siedle, L. Adams, “Handbook of radiation Effects”, 2nd edition, New York, Oxford University Press, 2002.

[9] John Cochran – “A SET Resistant Majority Voting Circuit”, Military / Aerospace Programmable Logic Devices (MAPLD) Conference, 2009.

[10] Melanie Berg, “Design for Radiation Effects”, Military / Aerospace Programmable Logic Devices (MAPLD), Conference, Annapolis, MD, 2008.

[11] Melanie Berg, “Embedding Asynchronous FIFO Memory Blocks in Xilinx Virtex Series FPGAs Targeted for Critical Space System Applications”, Military/Aerospace PLD (MAPLD) Conference, 2009.

[12] A. Keys, J. Adams, R. Darty, M. Patrick, M. Johnson, & J. Cressier "Radiation Hardened Electronics for Space Environments (RHESE) Project Overview ", International Planetary Probes Workshop (IPPW), Atlanta, GA, June 2008.

[13] C. Carmichael, B. Bridgford, G. Swift, M. Napier, "A Triple Module Redundancy Scheme for SEU Mitigation of Static Latch-Based FPGAs", Military/Aerospace PLD (MAPLD) Conference, 2004.

[14] M. Stetller, "Radiation Effects and Mitigation Strategies for modern FPGAs", 10th annual workshop for LHC and Future experiments, Los Alamos National Laboratory, USA, 2004.

[15] Ricky W. Butler, "A Primer on Architectural Level Fault Tolerance", NASA Scientific and Technical Information (STI) Program Office, Report No. NASA/TM-2008-215108, Feb. 2008.

[16] Mehlitz, P.C., Penix, J.J., and Markosian, L. Z., "Radiation-Hardened Software for Space Flight Science Applications", American Geophysical Union, Fall Meeting 2005, abstract #IN41A-0316, 2005.

[17] Shaneyfelt, M.R.; Dodd, P.E.; Draper, B.L.; Flores, R.S.; , "Challenges in Hardening Technologies Using Shallow-Trench Isolation," *Nuclear Science, IEEE Transactions on* , vol.45, no.6, pp.2584-2592, Dec 1998.

[18] Dawes, W.R.; Derbenwick, G.F.; Gregory, B.L.; , "Process technology for radiation-hardened CMOS integrated circuits," *Solid-State Circuits, IEEE Journal of* , vol.11, no.4, pp. 459- 465, Aug 1976.

[19] Binzaid, S.; Attia, J.O.; Schrimf, R.D.; , "Enclosed Layout Transistor with Active Region Cutout," *Region 5 Conference, 2008 IEEE* , vol., no., pp.1-5, 17-20 April 2008.

[20] J.W. Gambles, G.K. Maki, "Rad-Tolerant Flight VLSI From Commercial Foundries", *39th IEEE Midwest Symposium on Circuits and Systems*, vol 3, pp. 1227-1230, August 18-21, 1996.

[21] M. Johnson, “Radiation Hardened, High Performance, Power Efficient Processing – An Objective of the NASA Exploration Systems Technology Development Program”, *13th NASA Symposium on VLSI Design*, Post Falls, ID, June 5-6, 2007.

[22] Bernard Bancelin – “ATMEL ATF280E Rad Hard SRAM Based FPGA: SEE test results and fault injection”, Military / Aerospace Programmable Logic Devices (MAPLD) Conference, 2009.

[23] Andrew S. Keys, Howell, “Technology Developments in Radiation-Hardened Electronics for Space Environments”, NASA Technical Reports Server (NTRS), Document No. 20080032798, [Online], Available: ntrs.nasa.gov, March 2010.

[24] S. Franklin, K. Jentung, B. Spence, M. McEachen, S. White, J. Samson, R. Some, J. Zsoldos, “The Space Technology 8 Mission”, 2006 IEEE *Aerospace Conference*, pp 16, March 4-11, 2006.

[25] J. Greco, G. Cieslewski, A. Jacobs, I.A. Troxel, A.D. George, “HW/SWinterface or high-performance space computing with FPGA coprocessors”, 2006 IEEE *Aerospace Conference*, pp 10, March 4-11, 2006.

[26] Marshall C. Patrick, “RHESE Reconfigurable Computing (RC) Task”, Military / Aerospace Programmable Logic Devices (MAPLD), Conference, Annapolis, MD, 2008.

[27] G. Alonzo Vera, “A Programmable Configuration Scrubber for FPGAs”, Military / Aerospace Programmable Logic Devices (MAPLD) Conference, 2009.

[28] “Early Access Partial Reconfiguration User Guide”, Xilinx Document No. UG208 (v1.2), [Online], Available: www.xilinx.com, March 2010.

[29] “PicoBlaze™ 8-bit Embedded Microcontroller User Guide”, Xilinx Document No. UG129 (v2.0), [Online], Available: www.xilinx.com, March 2010.

[30] “MicroBlaze™ Processor Reference Guide”, Xilinx Document No. UG081 (v9.0), [Online], Available: www.xilinx.com, March 2010.

[31] David Barker-Plummer, “Turing Machines”, *Stanford Encyclopedia of Philosophy*. [Online], Available: http://plato.stanford.edu/entries/turing-machine/, March 2010.

[32] “System ACE CompactFlash Solution”, Xilinx Document No. DS080 (v2.0), [Online], Available: www.xilinx.com, March 2010.

APPENDICES

APPENDIX A

PICOBLAZE PR FIGURES

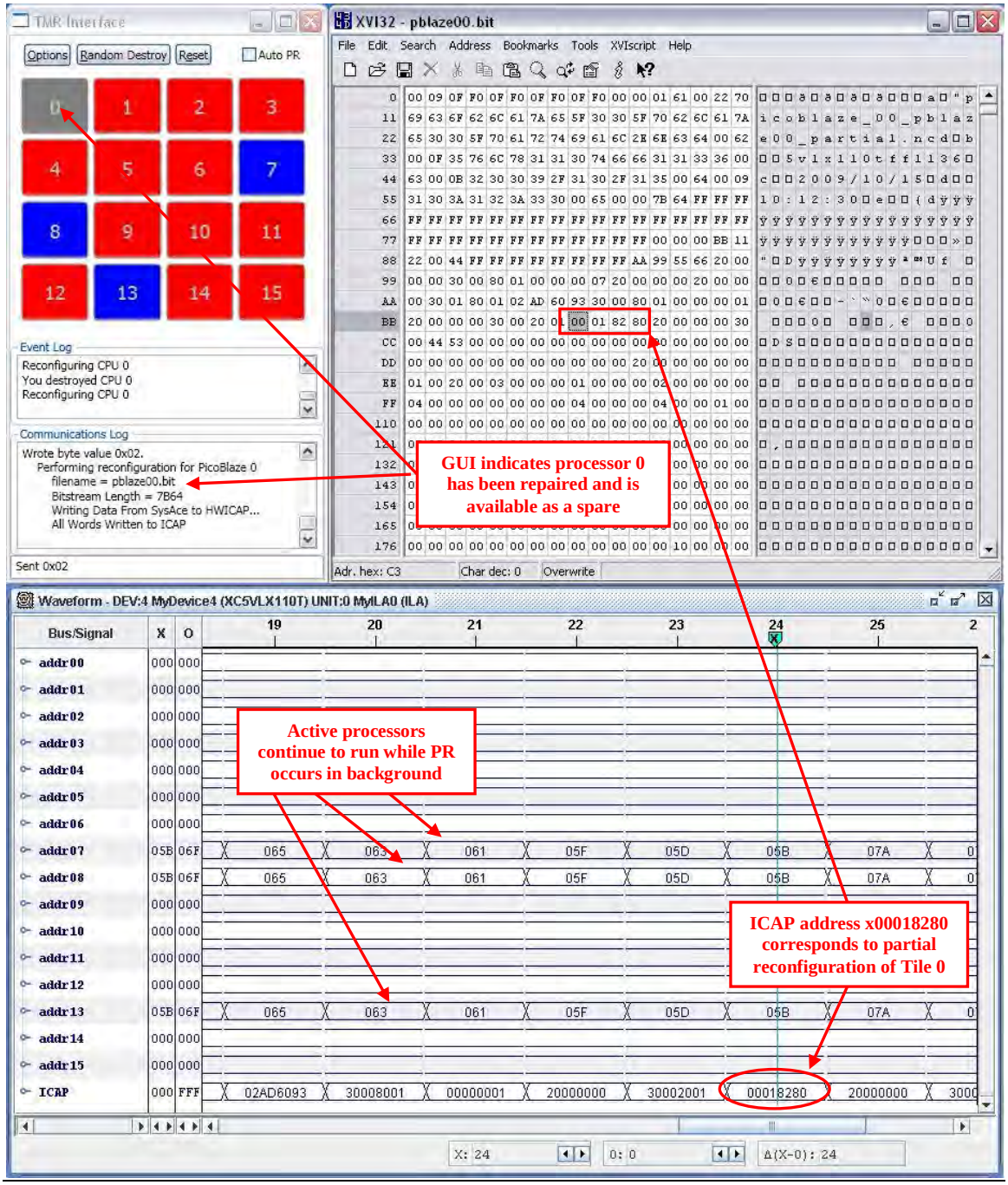

Figure A.1: Partial reconfiguration of PicoBlaze processor tile 0.

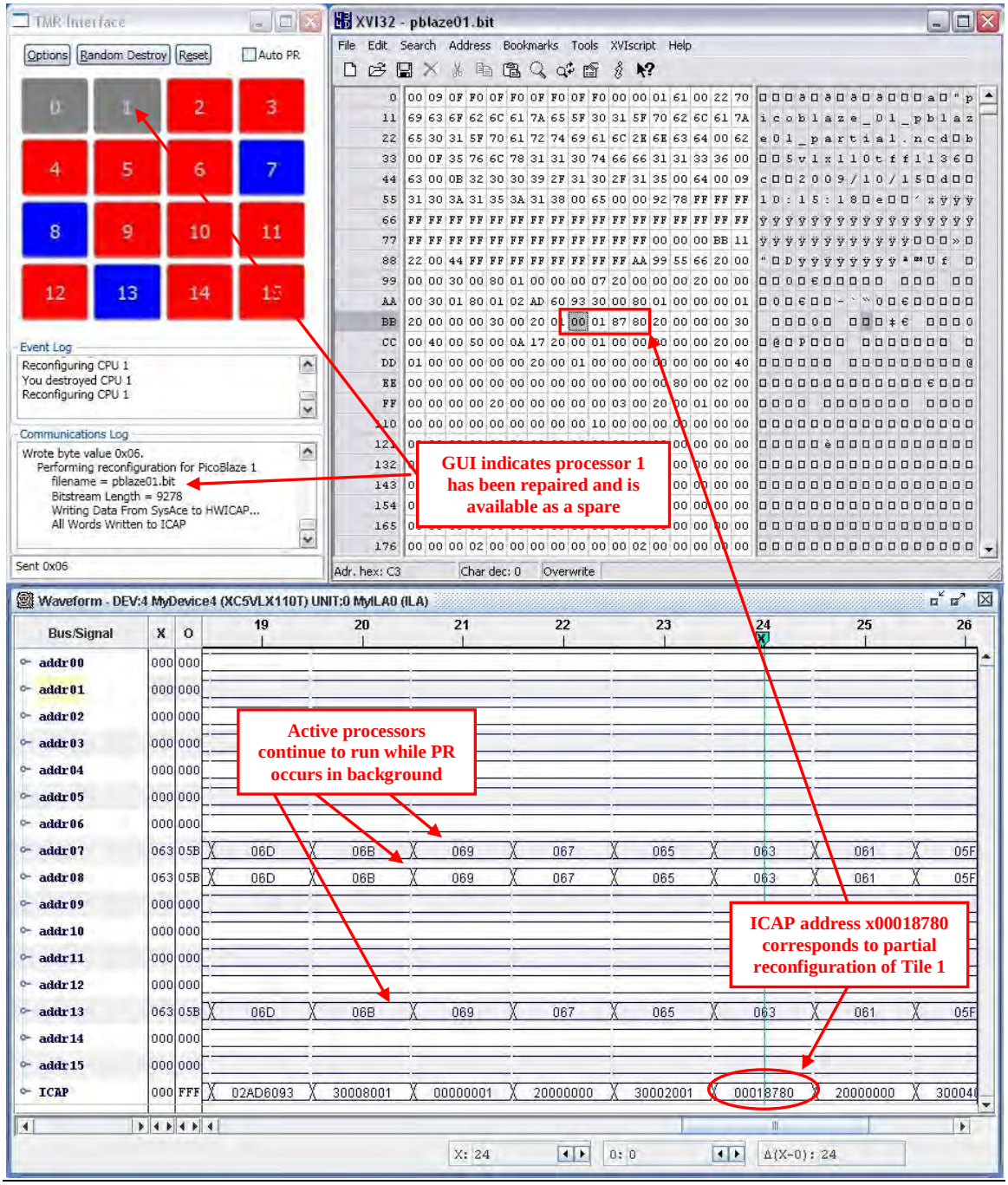

Figure A.2: Partial reconfiguration of PicoBlaze processor tile 1.

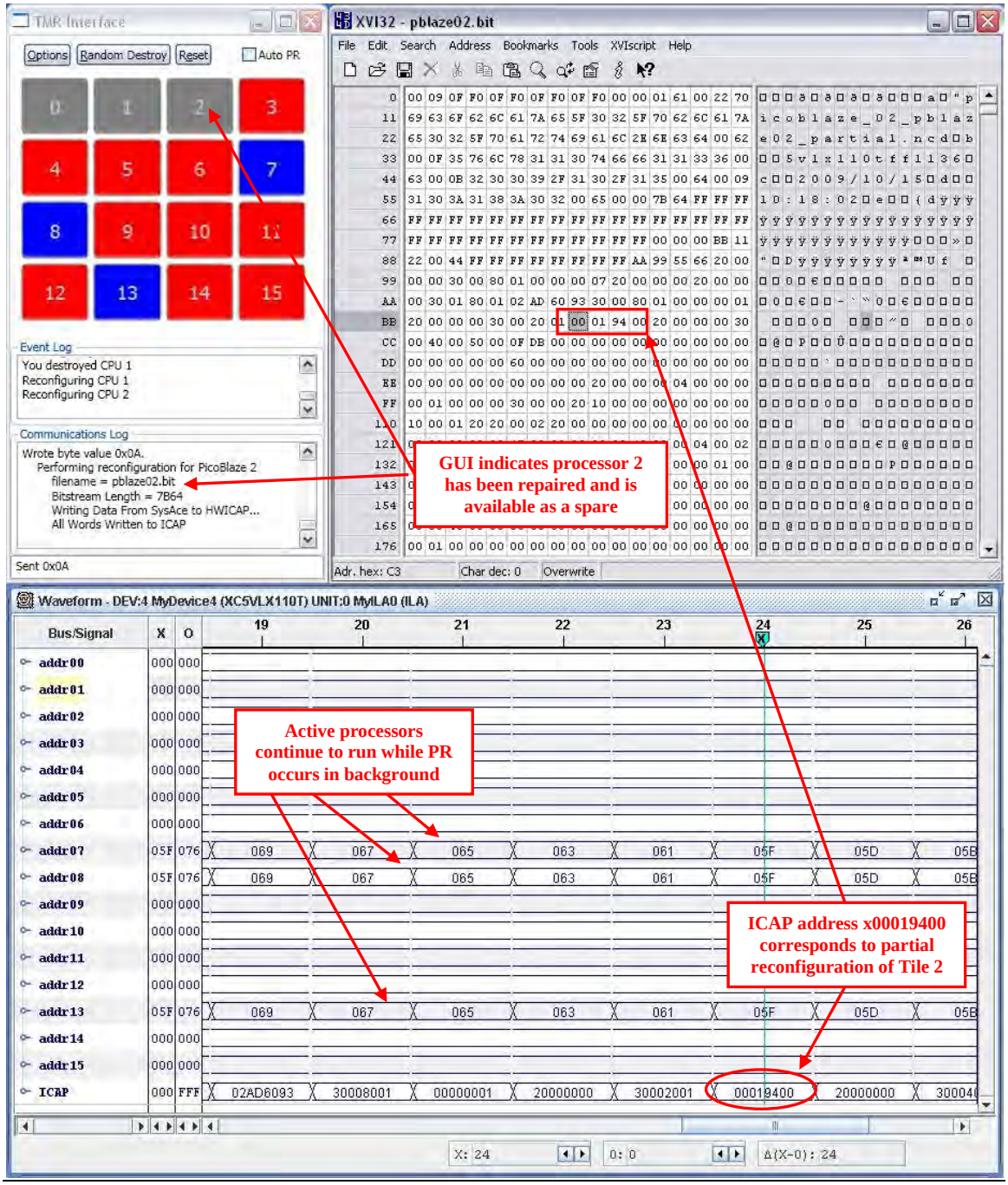

Figure A.3: Partial reconfiguration of PicoBlaze processor tile 2.

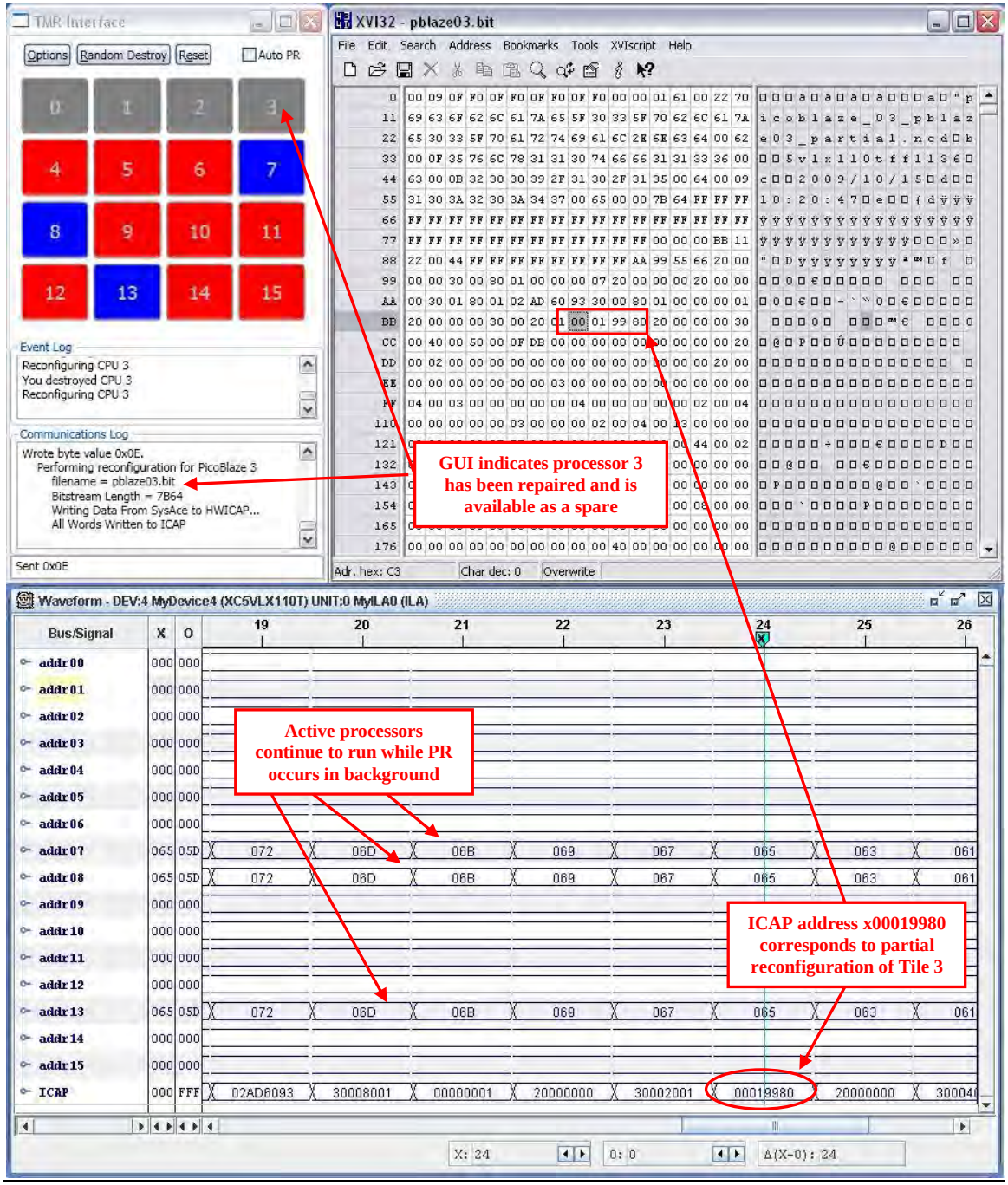

Figure A.4: Partial reconfiguration of PicoBlaze processor tile 3.

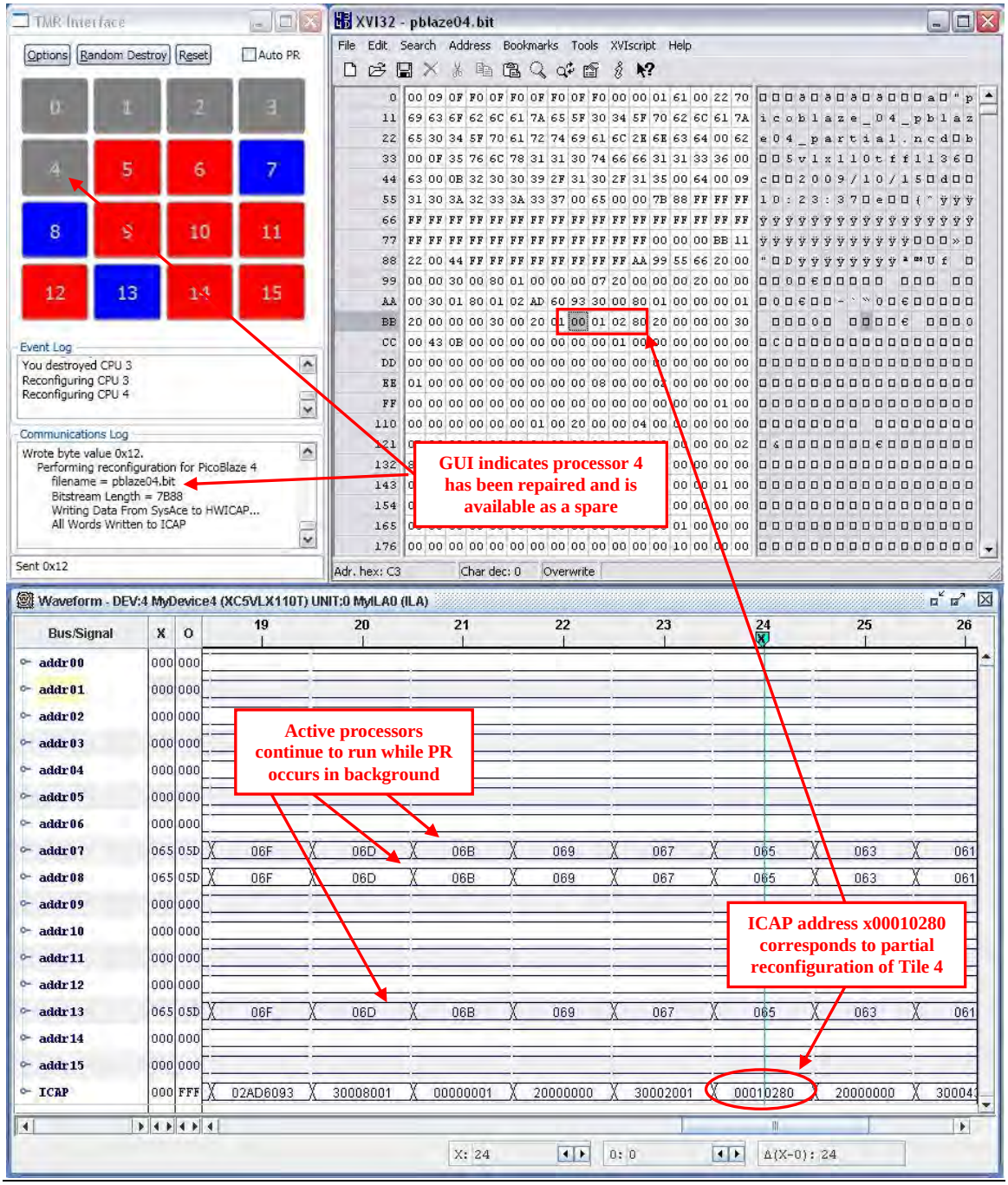

Figure A.5: Partial reconfiguration of PicoBlaze processor tile 4.

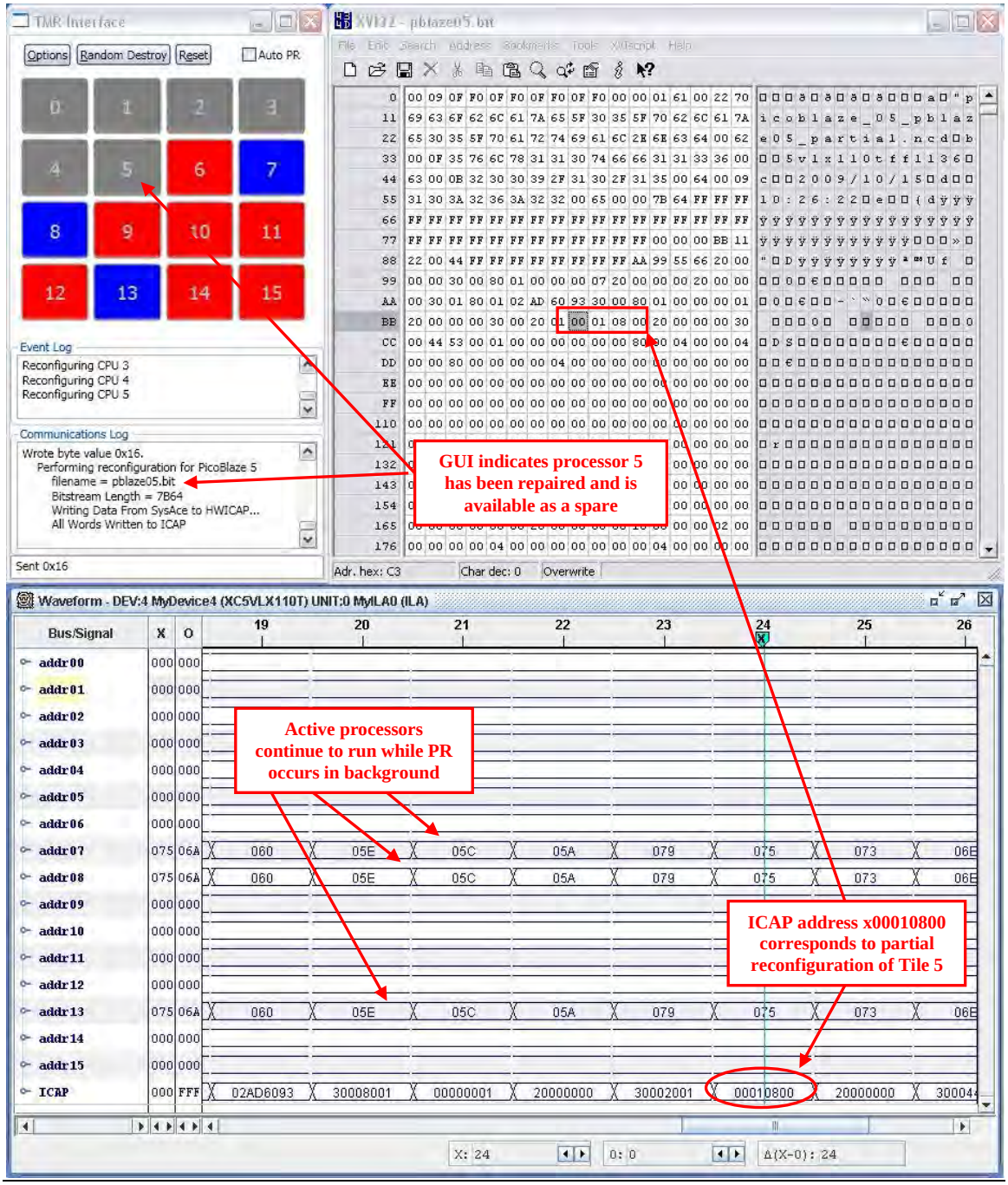

Figure A.6: Partial reconfiguration of PicoBlaze processor tile 5.

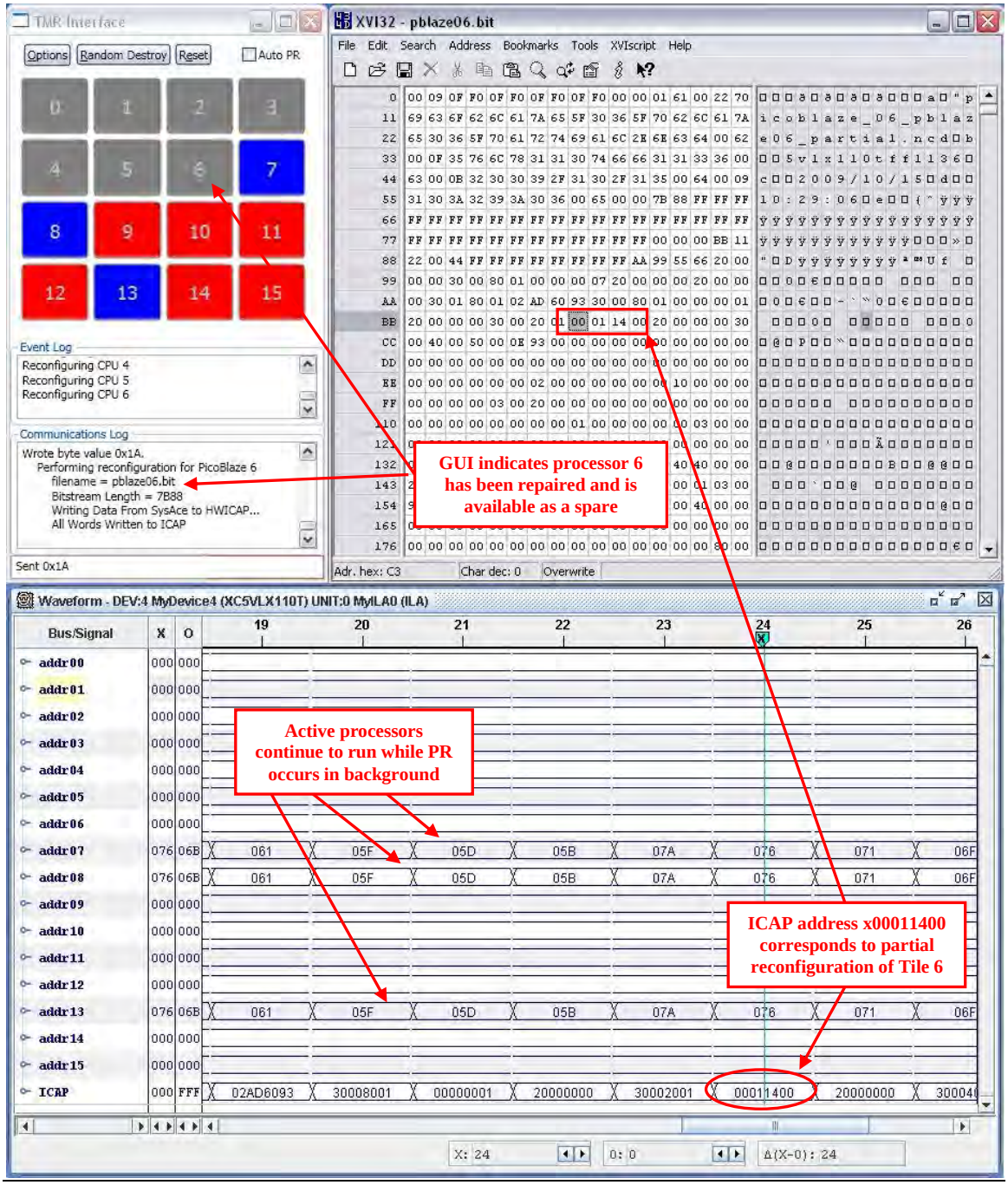

Figure A.7: Partial reconfiguration of PicoBlaze processor tile 6.

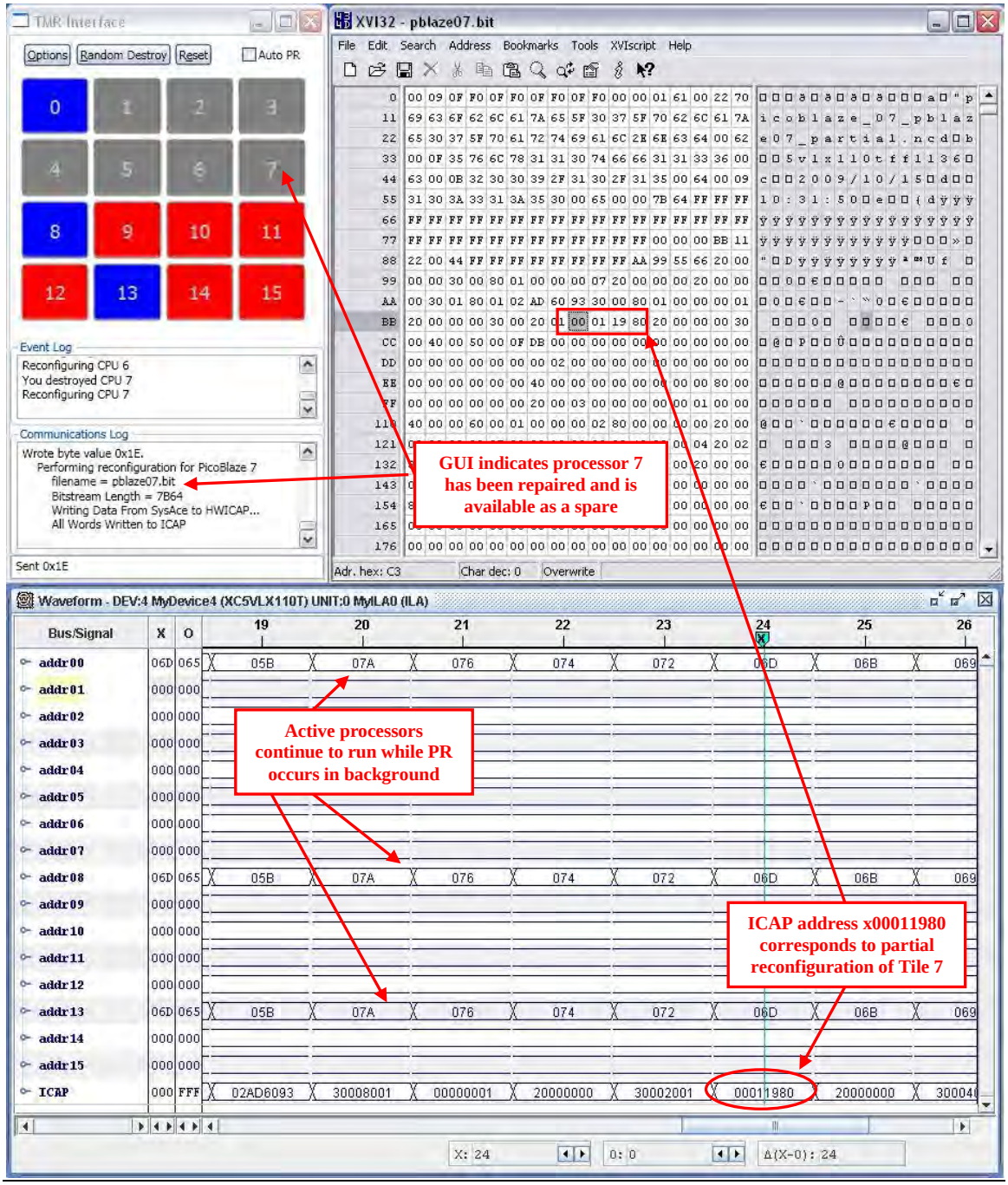

Figure A.8: Partial reconfiguration of PicoBlaze processor tile 7.

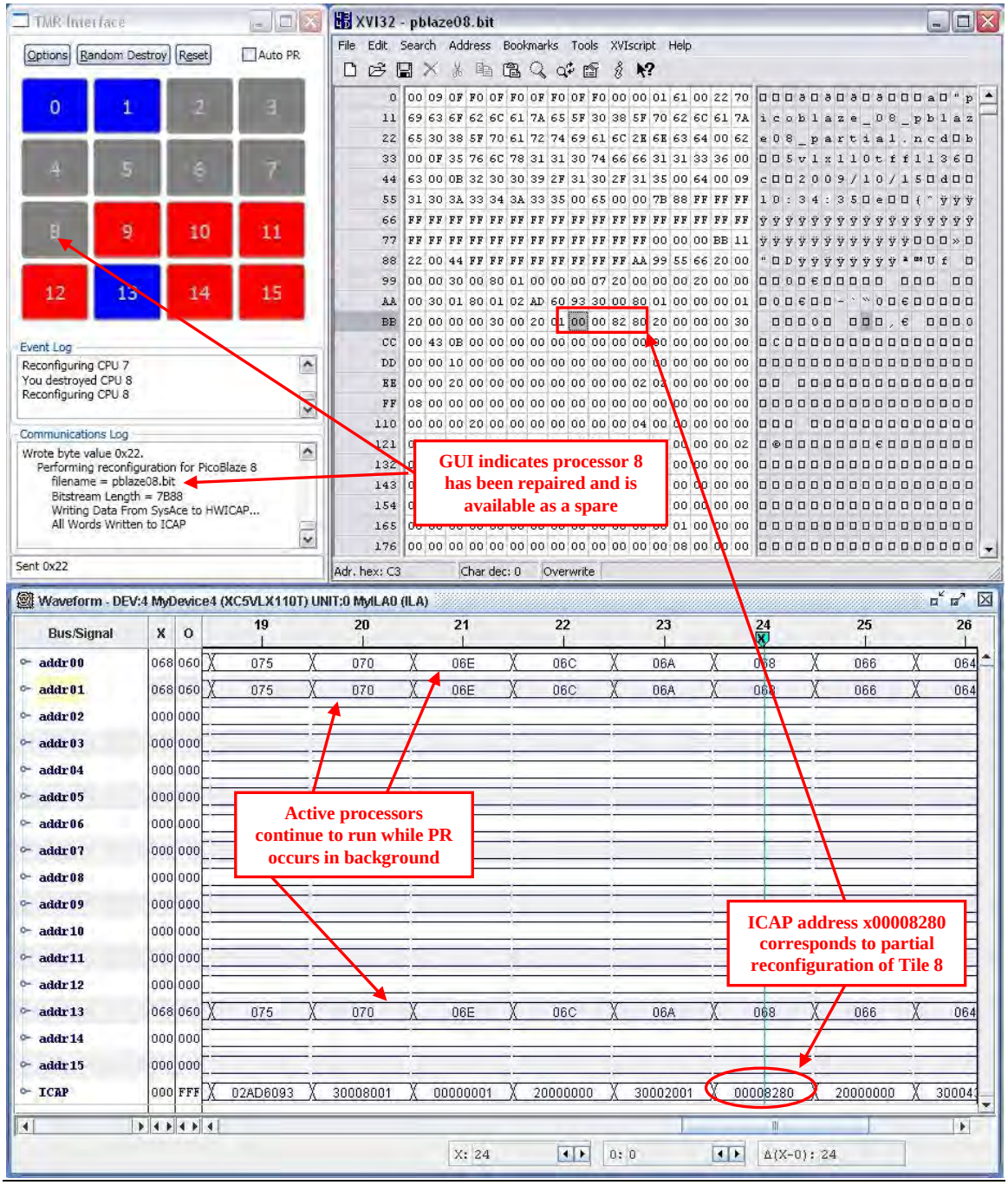

Figure A.9: Partial reconfiguration of PicoBlaze processor tile 8.

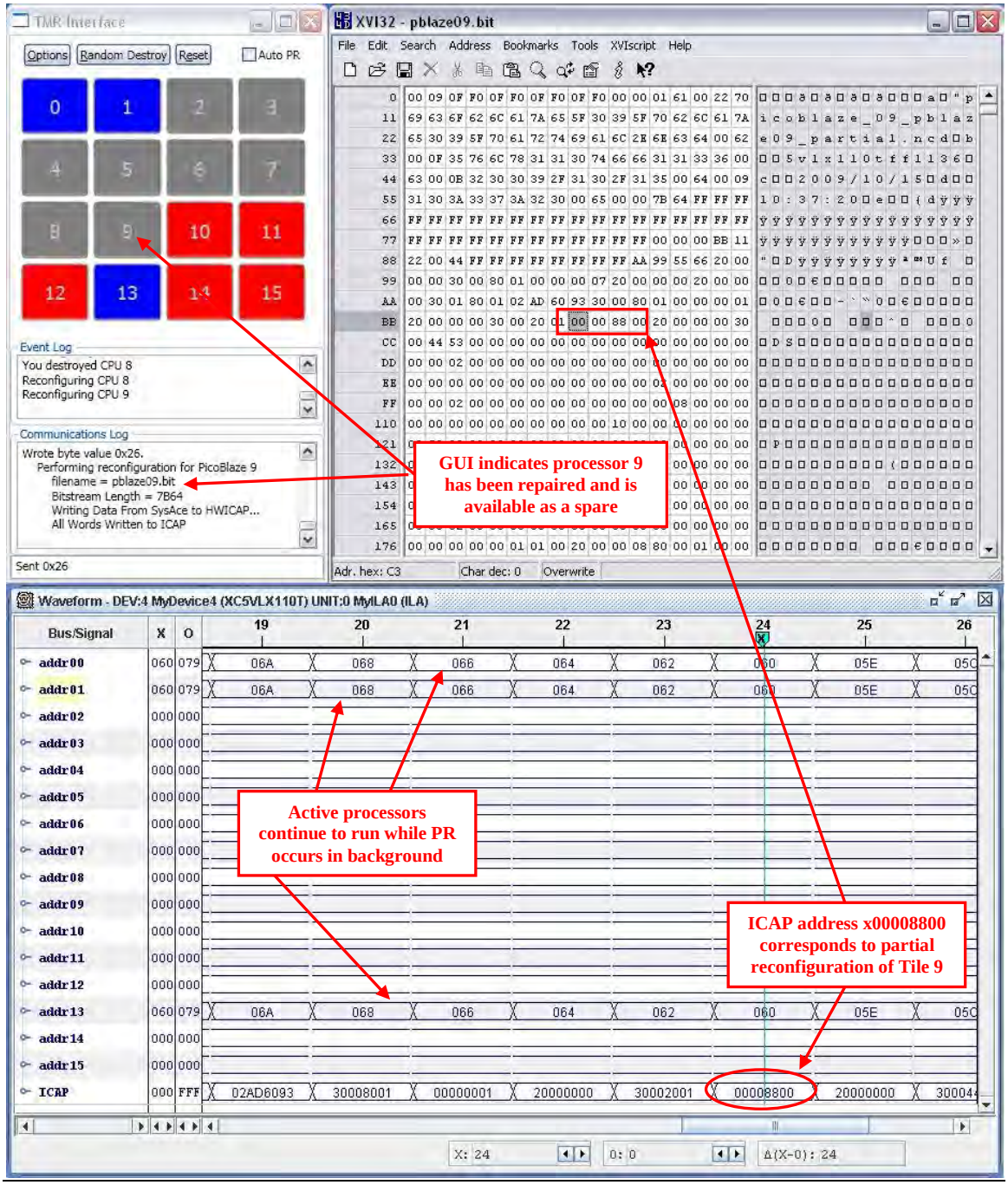

Figure A.10: Partial reconfiguration of PicoBlaze processor tile 9.

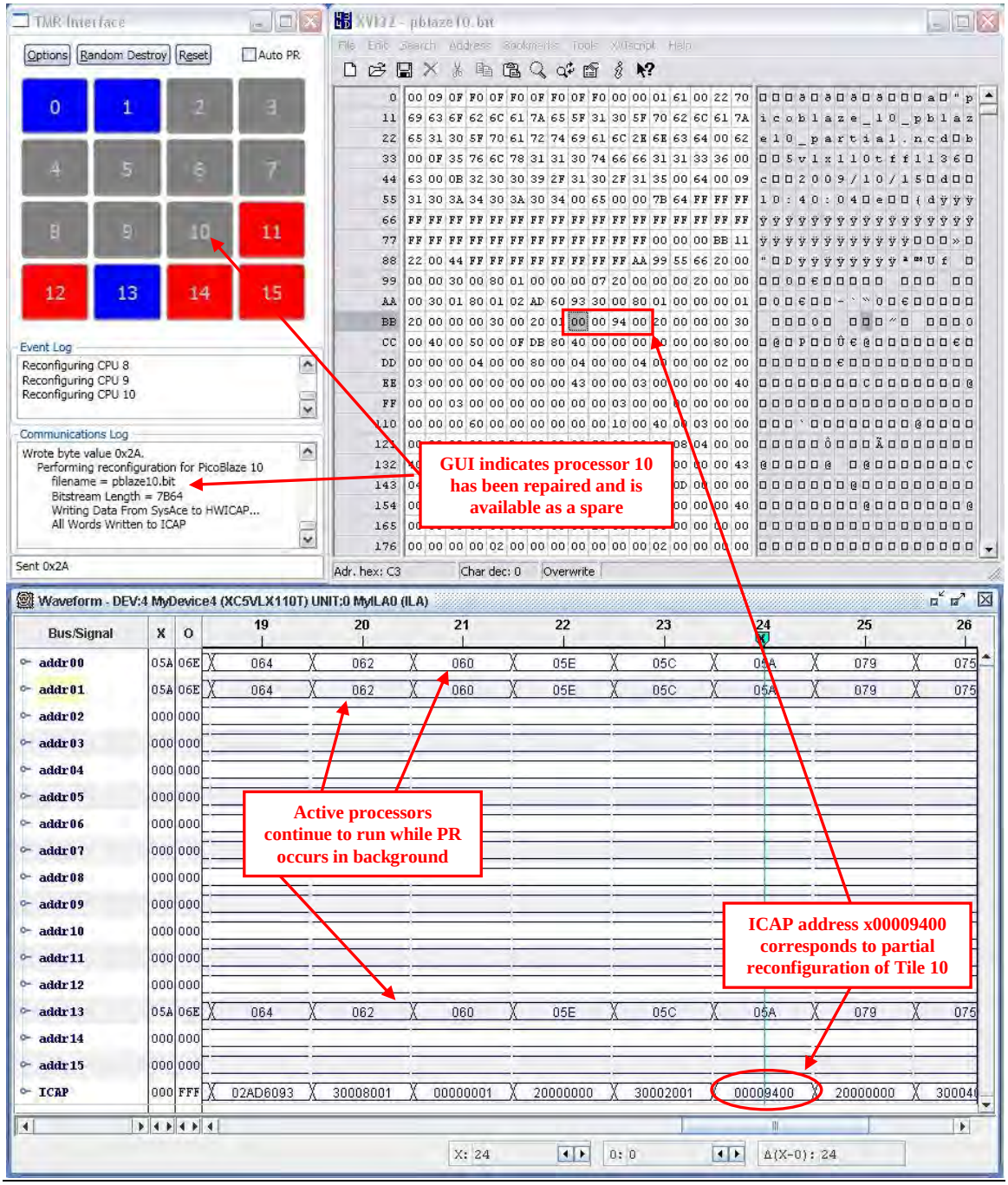

Figure A.11: Partial reconfiguration of PicoBlaze processor tile 10.

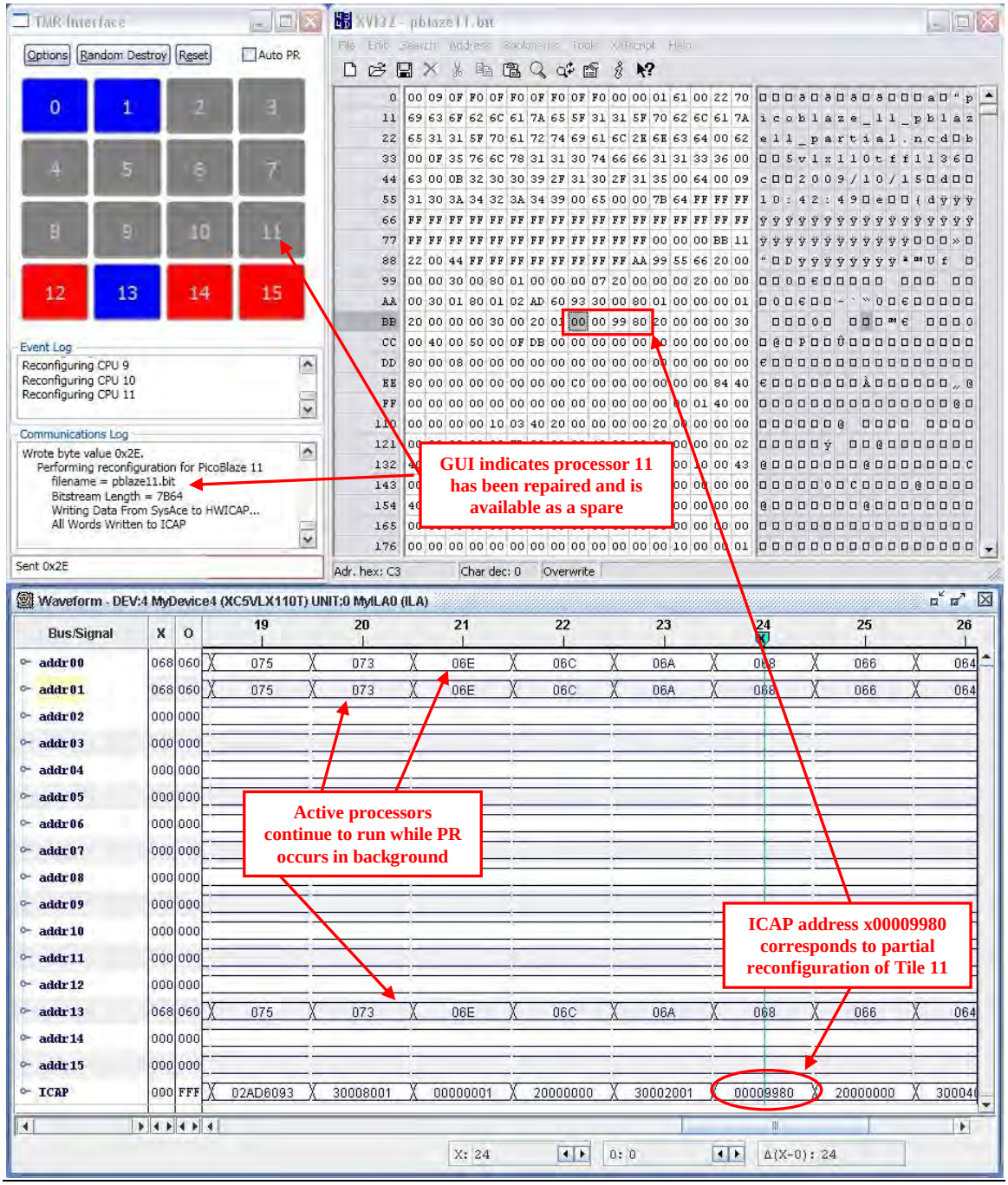

Figure A.12: Partial reconfiguration of PicoBlaze processor tile 11.

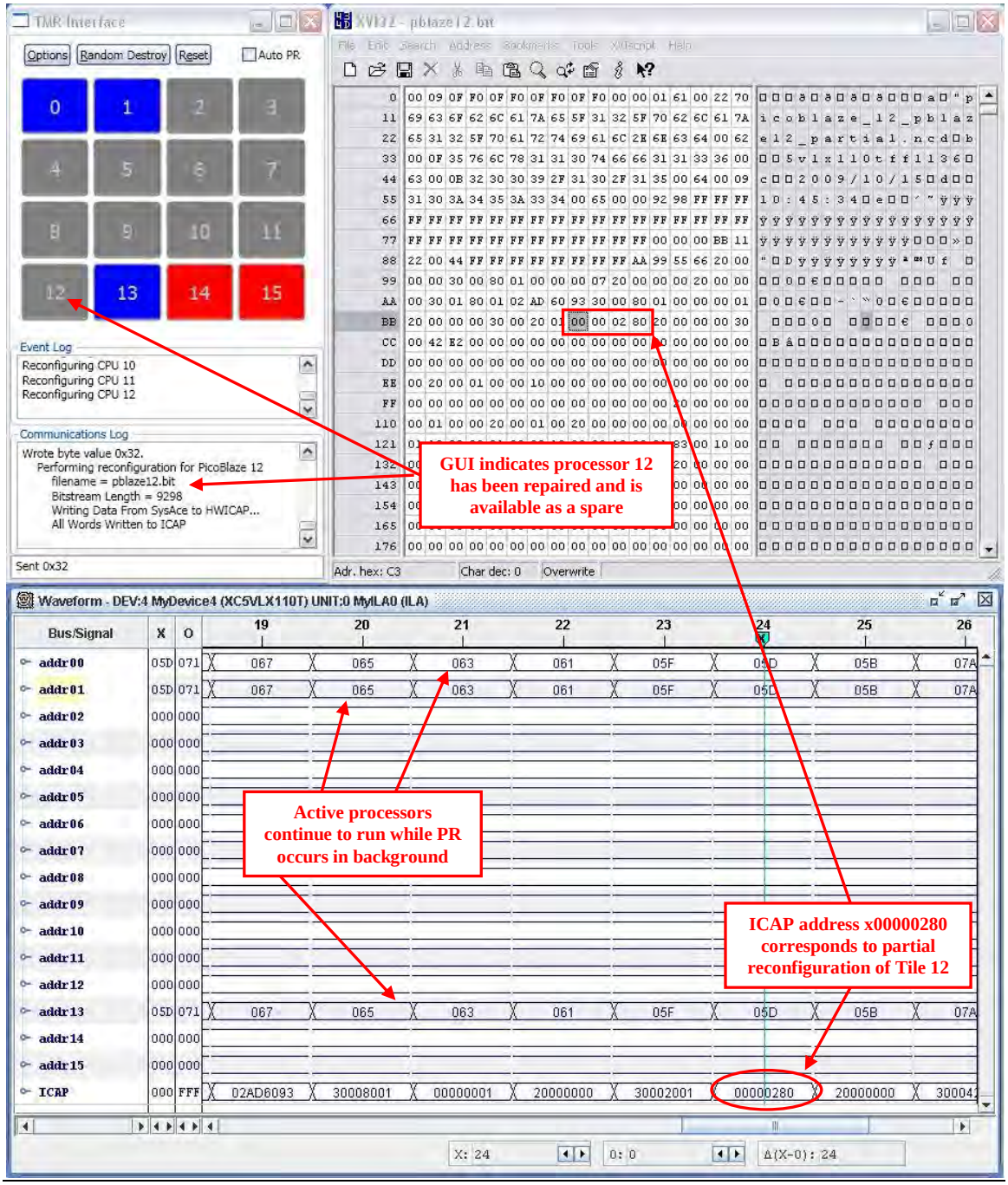

Figure A.13: Partial reconfiguration of PicoBlaze processor tile 12.

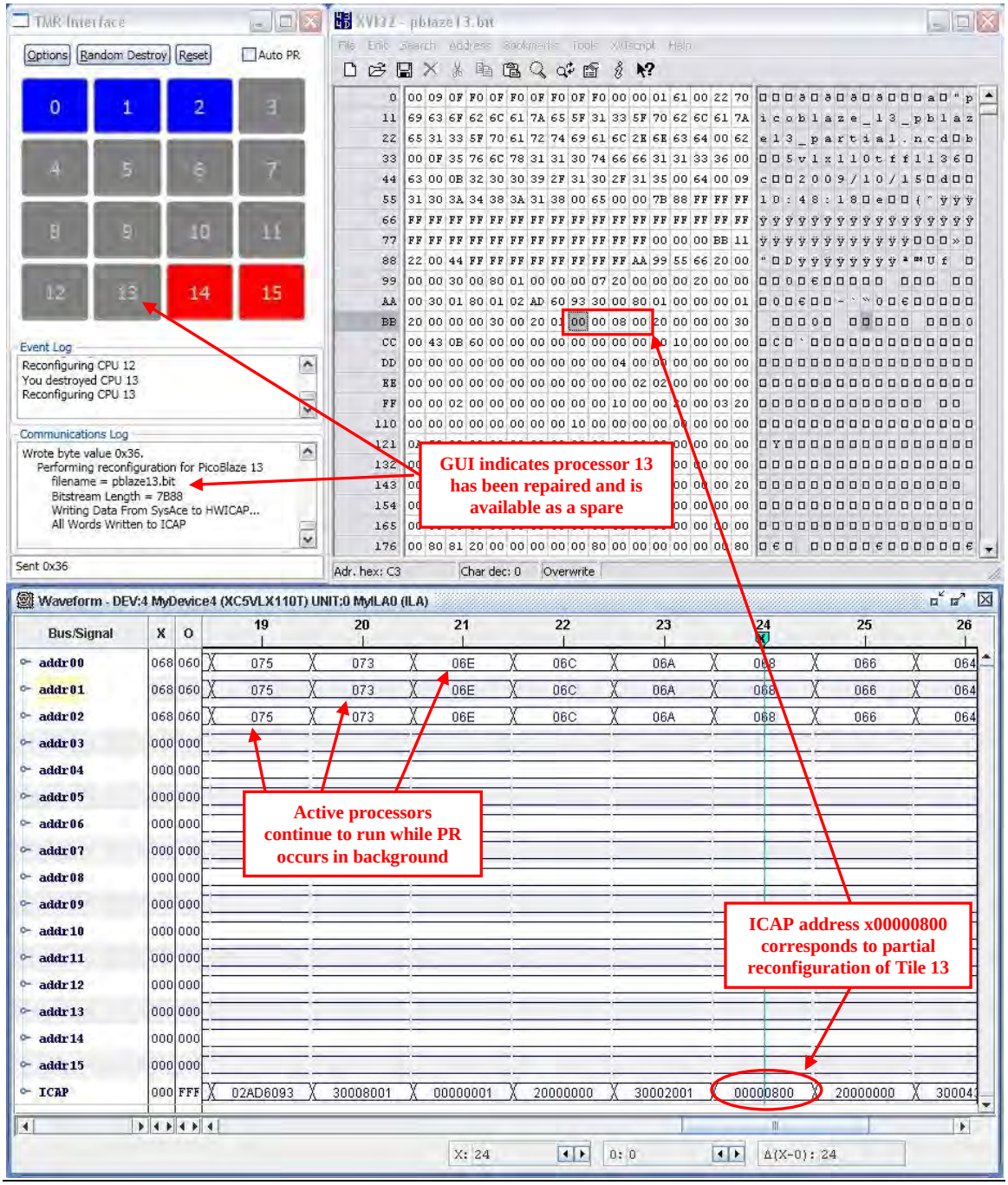

Figure A.14: Partial reconfiguration of PicoBlaze processor tile 13.

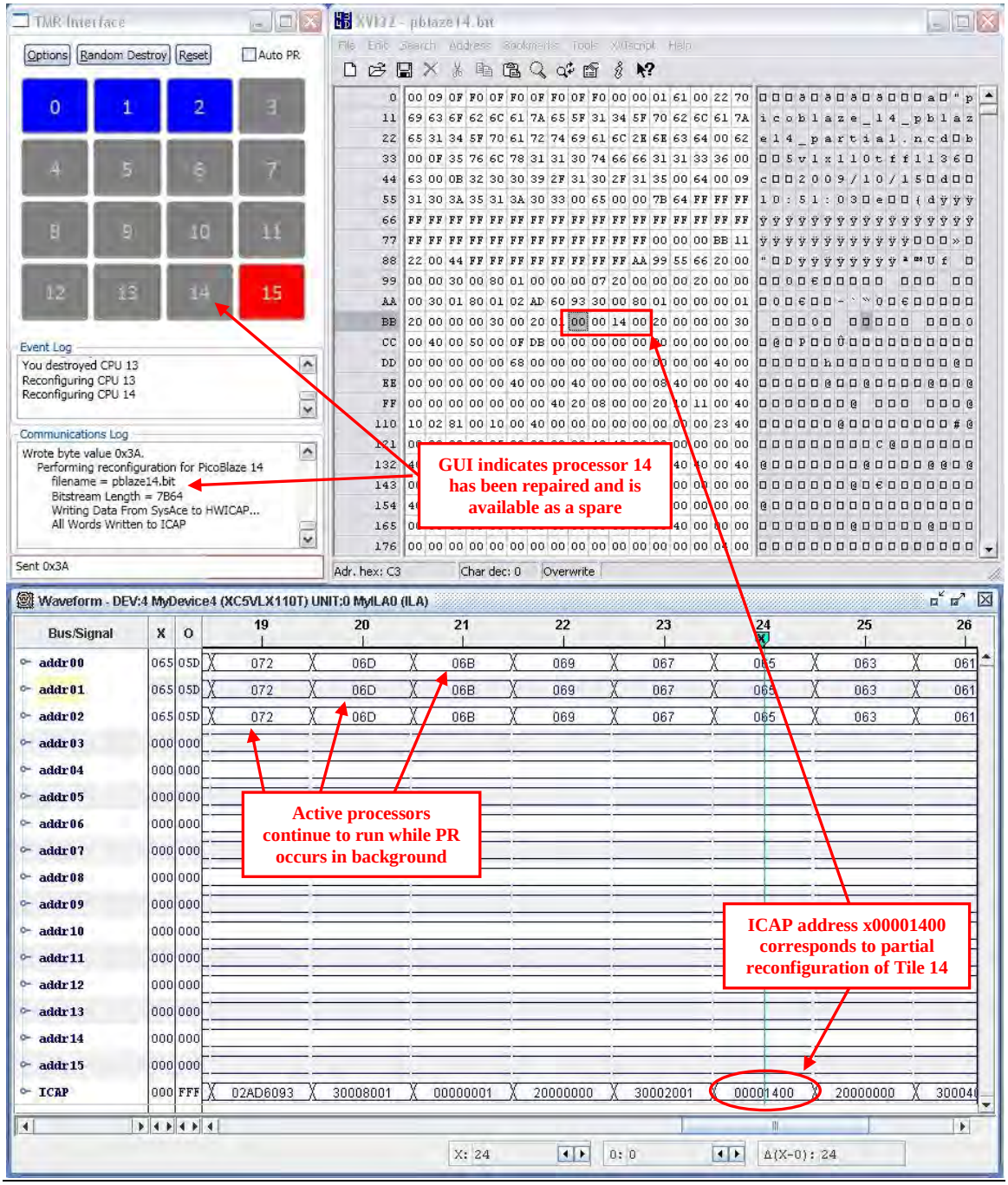

Figure A.15: Partial reconfiguration of PicoBlaze processor tile 14.

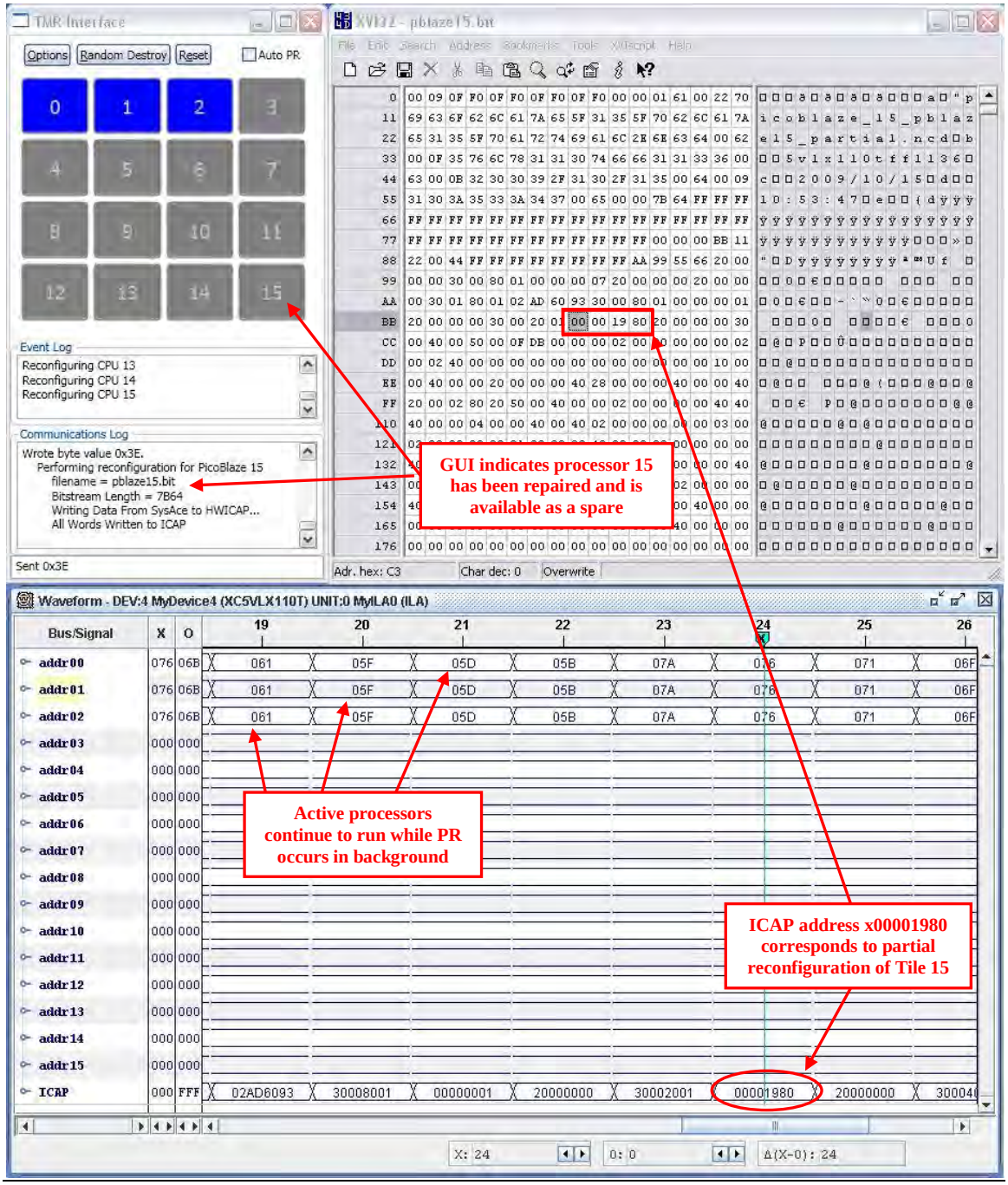

Figure A.16: Partial reconfiguration of PicoBlaze processor tile 15.

APPENDIX B

MICROBLAZE PR FIGURES

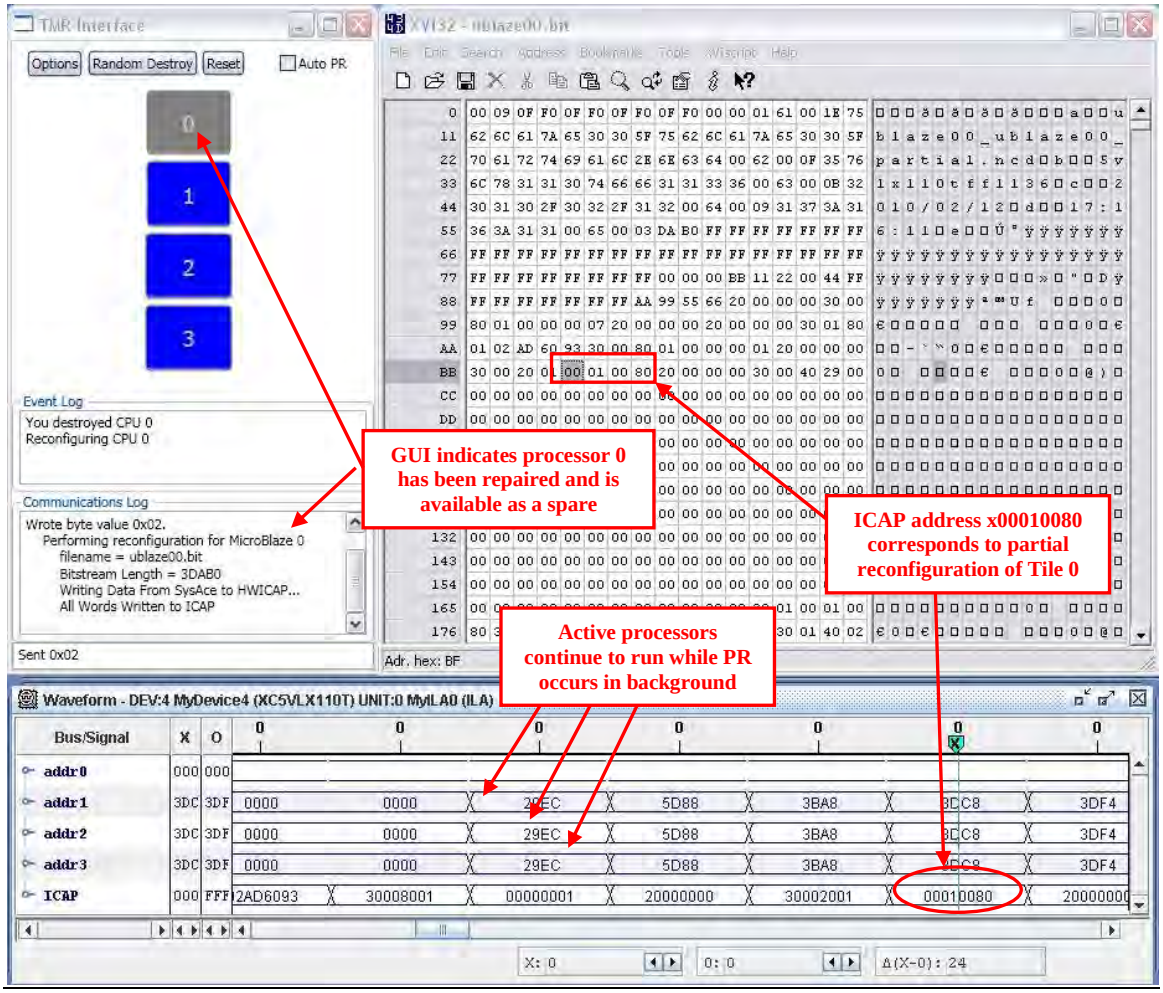

Figure B.1: Partial reconfiguration of MicroBlaze processor tile 0.

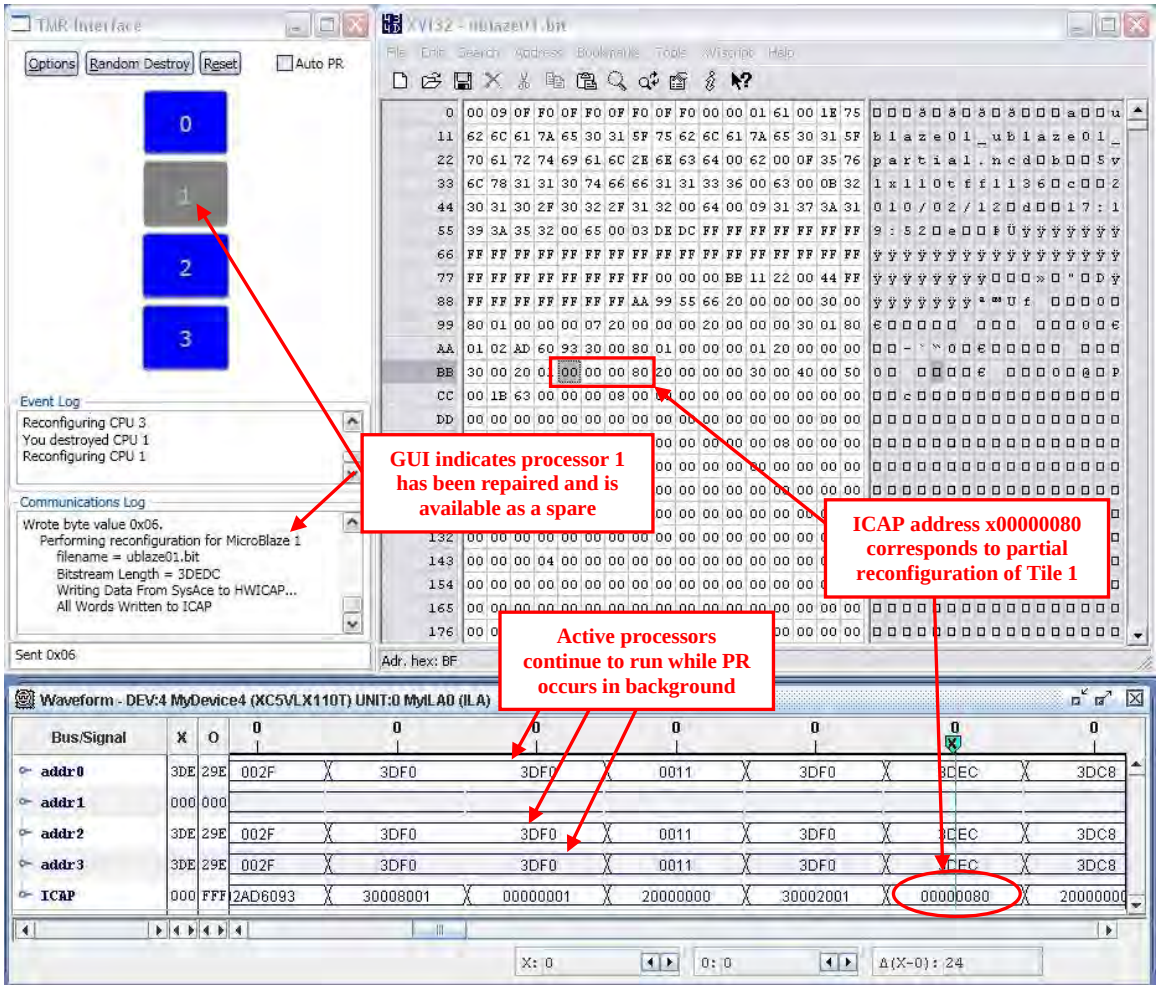

Figure B.2: Partial reconfiguration of MicroBlaze processor tile 1.

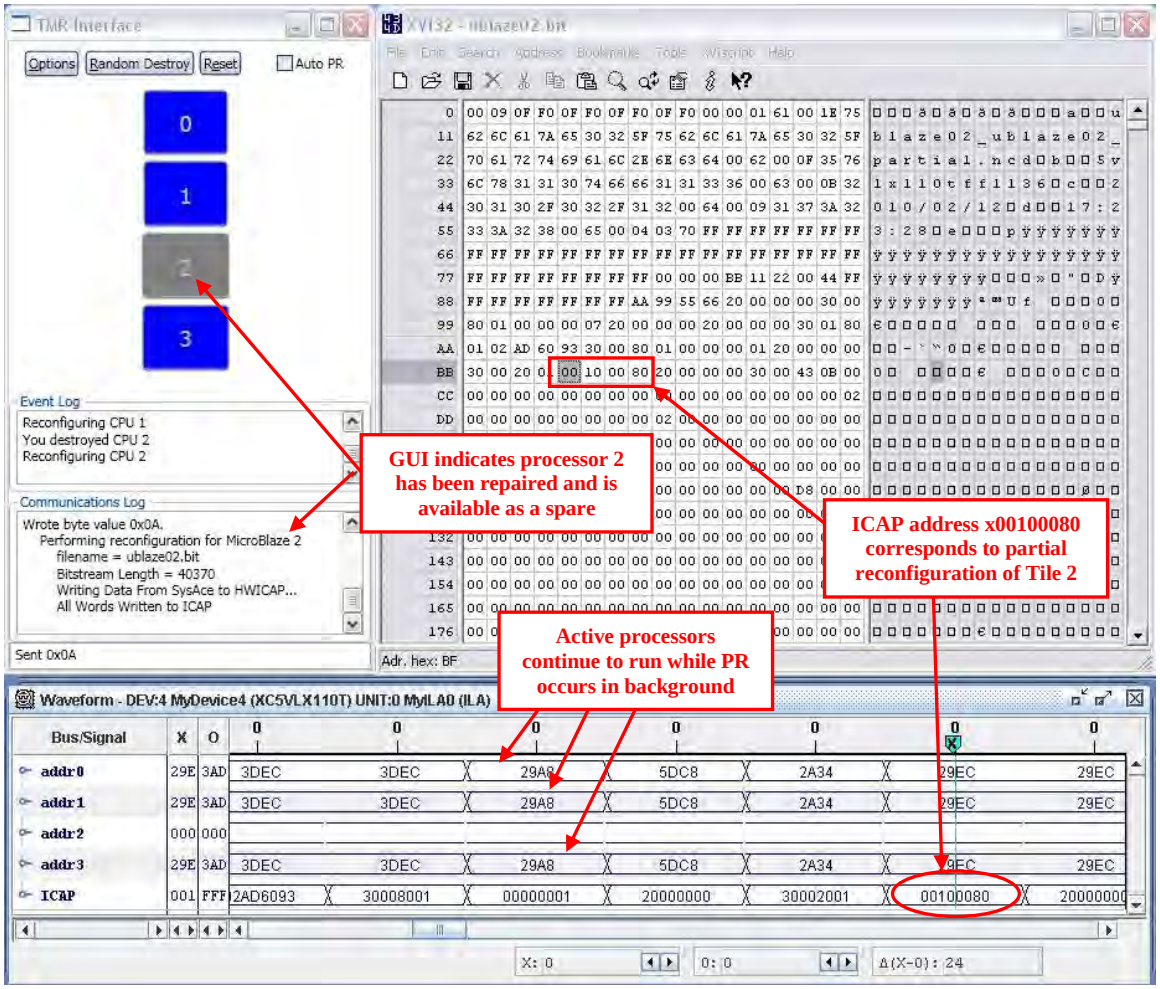

Figure B.3: Partial reconfiguration of MicroBlaze processor tile 2.

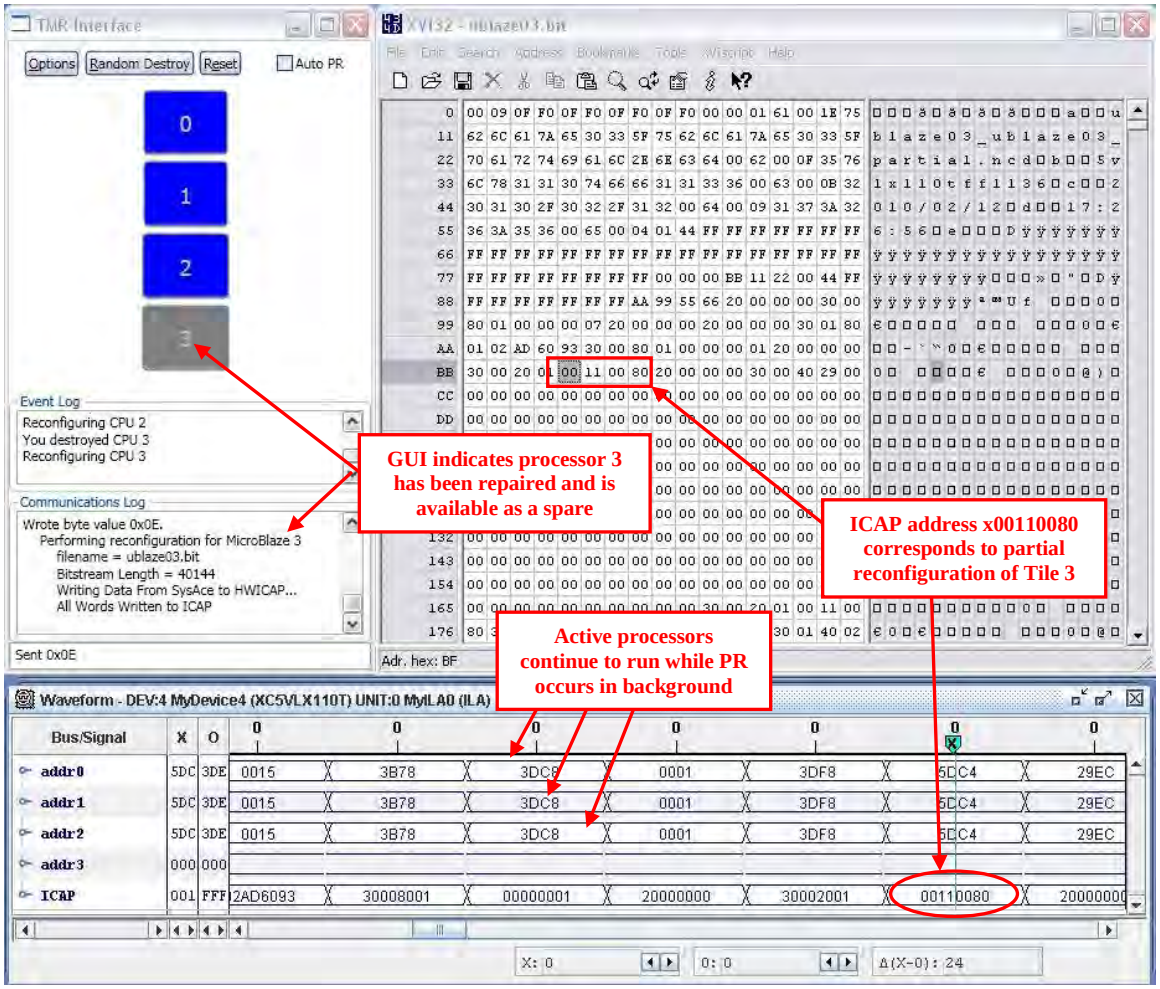

Figure B.4: Partial reconfiguration of MicroBlaze processor tile 3.