

# Challenges of Remote FPGA Configuration for Space Applications<sup>12</sup>

Ms. Mindy Surratt<sup>3</sup>, Dr. Herschel H. Loomis<sup>3</sup>, Dr. Alan A. Ross<sup>3</sup>, Dr. Russ Duren<sup>4</sup>  
<sup>3</sup>Naval Postgraduate School  
777 Dyer Rd., Bldg 233  
Monterey, CA 93950  
(831) 656-3213  
mlsurrat@nps.edu  
<sup>4</sup>Baylor University  
Waco, Texas

*Abstract*—There are many unique challenges associated with providing remote access to space experimental payloads. The limited bandwidth to the space craft, the inability to physically monitor and probe the payload, and the management of access time for various researchers working on the project all compound to create a challenging work environment.

The Configurable Fault Tolerant Processor (CFTP) project aims to alleviate many of the difficulties associated with remote payload operation. We have made use of modular FPGA design, which allows us to transfer only small application modules rather than full configuration files. This dramatically reduces the bandwidth required to upload new applications as we discover new experiments for the CFTP after launch.

Another unique aspect of the CFTP project is the collaborative effort in its development. We must manage access time for universities and research institutions across the country for running experiments on the CFTP, downloading CFTP documents, and analyzing telemetry after launch.

## TABLE OF CONTENTS

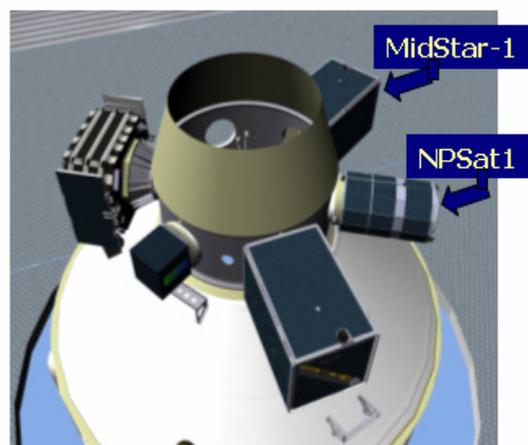
1. INTRODUCTION.....	1
2. DEVELOPMENT CONCERNS.....	3
3. IN-FLIGHT CONCERNS .....	4
4. CONCLUSIONS .....	7
REFERENCES .....	7
BIOGRAPHY .....	8
APPENDIX A – CFTP TERMS.....	9

## 1. INTRODUCTION

The Configurable Fault Tolerant Processor (CFTP) is a test bed for space based reconfigurable hardware. It is a shared platform used by researchers located across the country to

perform fault tolerant computing and reconfiguration experiments. The CFTP employs SRAM-based reconfigurable FPGAs as opposed to anti-fuse, one-time programmable FPGAs to obtain the necessary flexibility [1].

The CFTP is currently manifested as a Space Test Program (STP) experimental payload on the Naval Postgraduate School’s NPSat1 and the United States Naval Academy’s MidStar-1 LEO satellites, both of which are scheduled for launch in September 2006 (Figure 1).



**SPACE TEST PROGRAM (STP-1),  
ATLAS V LAUNCH VEHICLE,  
Est. Launch September 2006**

**Figure 1 - Image depicting the locations of NPSat1 and MidStar-1 on the STP launch vehicle.**

### *CFTP Payload Description*

The CFTP payload consists of a Printed Circuit Board (PCB) of 5.3 inches x 7.3 inches, and a power supply board.

On NPSat1, the CFTP board will be part of the spacecraft’s in-satellite controller stack, communicating directly with the

<sup>1</sup> U.S. Government work not protected by U.S. copyright.

<sup>2</sup> IEEEAC paper 1437, version 1, updated November 3, 2004

Command and Data Handler (C&DH). On MidStar-1, the CFTP will consist of a fully enclosed box containing the CFTP board, a power supply, and an interface and control ARM processor running embedded Linux. The CFTP requires 28V of power from MidStar (drawing no more than 5W) and a serial link to MidStar's C&DH for communications.

The CFTP board itself contains two Xilinx XQVR600 FPGAs—a control FPGA and an experiment FPGA. The control FPGA connects to the processor board through a modified PC/104 bus interface.

The PCB is designed with requisite supporting circuitry, including a EEPROM bootloader, six SDRAM program memory chips, and an Intel flash memory chip (Figure 3).

The six Elpida SDRAM chips provide a total of 128MB available storage capacity that will be used as program and data memory during operation. An additional 32MB of SDRAM is used to contain the error detection and correction (EDAC) code for the memory. A 4Mbit Xilinx EEPROM holds one initial configuration (3,601,968 bits per full configuration file) to serve as a failsafe bootloader. A 32 Mbit Intel Flash memory is available to store additional configurations. (Figure 2). The flash memory has a minimum of 100,000 block erase cycles. As the flash will be used to store configuration files, which will rarely be removed, the write cycle life of the memory should not be an issue on orbit.

The initial experiment FPGA configuration will be an instantiation of a triple modular redundant (TMR) processor, with included Error Detection and Correction (EDAC) and memory controller circuitry [2]. One of the main goals of the CFTP project is to evaluate the effectiveness of reprogrammable logic in creating a system that is both resistant to single event upsets caused by radiation in the space environment ([3] provides a concise outline of radiation effects to SRAM-based FPGAs in the space environment), as well as maintaining a flexible and modifiable experiment platform.

Several Masters students in the CFTP program at the Naval Postgraduate School have completed theses going into more detail on the radiation effects in the space environment and methods of mitigating errors. Dean Ebert, “Design and Development of a Configurable Fault Tolerant Processor (CFTP) for Space Applications,” gives an overview of the harshness of the space environment and methods of mitigating its effects [4]. Rong Yuan, “Triple Modular Redundancy (TMR) in a Configurable Fault Tolerant Processor (CFTP) for Space Applications,” goes into detail on the use of TMR to create an SEU resistant instantiation of a processor on the FPGAs [5]. Charles Hulme, “Testing and Evaluation of the Configurable Fault Tolerant Processor (CFTP) for Space-Based Applications,” outlines various test methods to verify the integrity of data and configurations on the CFTP [6].

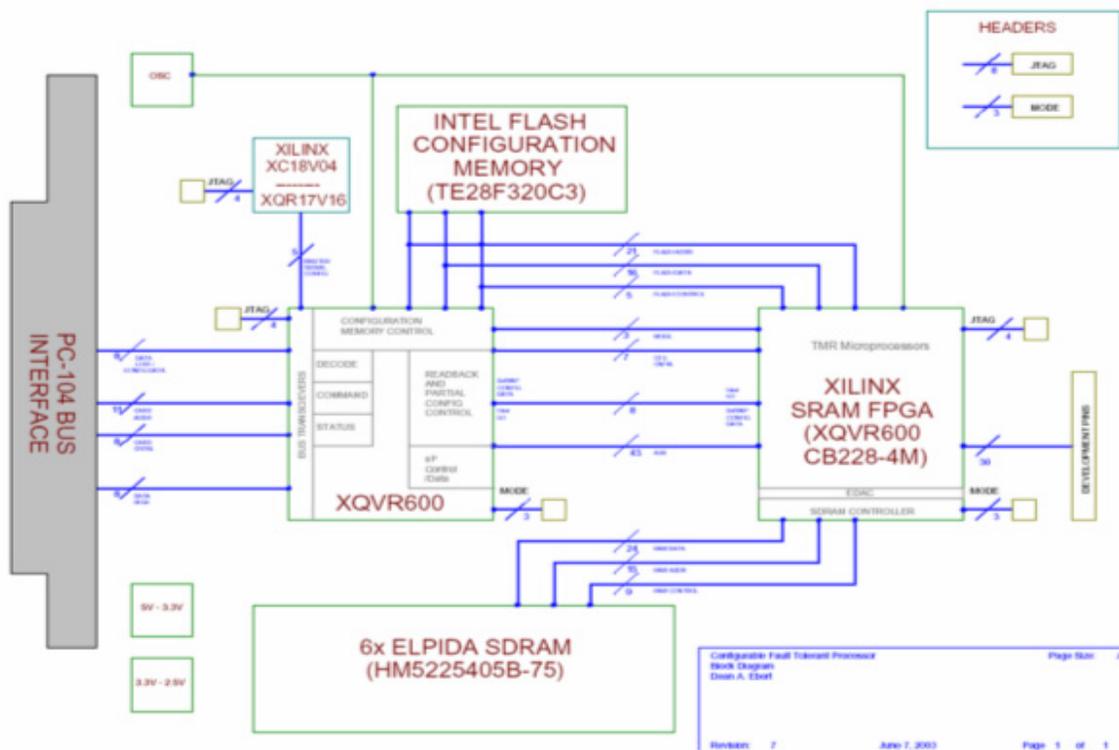
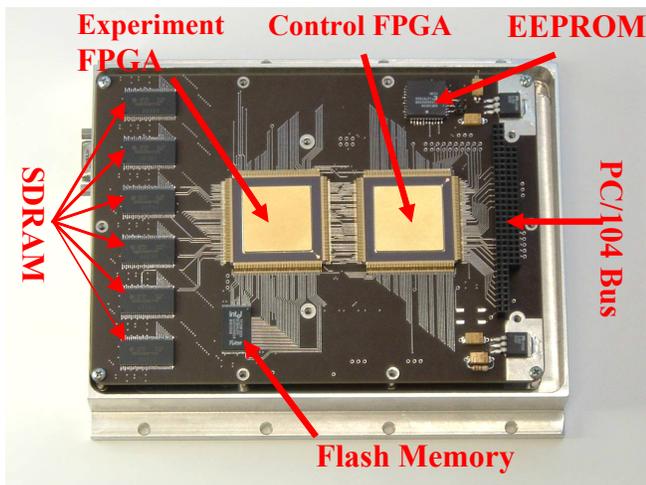


Figure 2 – Diagram of the CFTP board



**Figure 3 - CFTP Board**

*Remote Development Requirements*

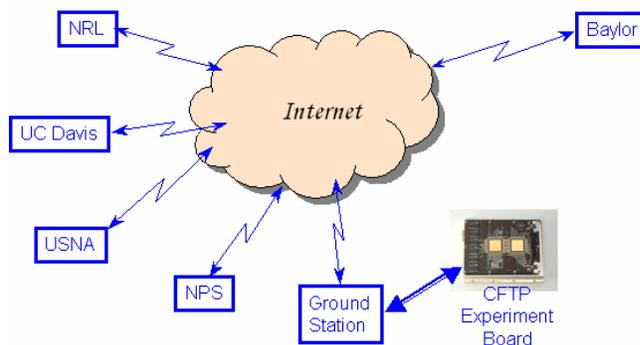
The breadboard CFTP stack used during development is identical to the flight version, with the exception of a surrogate processor board running embedded Linux being used in place of the ARM processor in order to interface with the outside world. This provides us with a remotely accessible development unit that simulates that which will be on board the spacecraft.

The CFTP project is a perfect platform for collaboration of many different organizations. The reconfigurable aspect of the project allows for researchers with a wide range of experimental interests to program the board to fit their needs. We have members of our team from Baylor University in Texas, the Naval Research Laboratory in Washington, DC, the U. S. Naval Academy in Maryland, UC Davis, as well as the Naval Postgraduate School here in Monterey (Figure 4). Since we can't afford multiple breadboards, all of these researchers must have access to the board and any related CFTP documents and data, both during development and after launch.

**2. DEVELOPMENT CONCERNS**

*The CFTP Server*

One of the initial concerns for the collaborative CFTP project was providing remote access to CFTP tools, documents, and data for the various researchers involved. We have set up an Internet-based system to host all CFTP related files and information. This server is connected to the CFTP stack through a PPP link over a serial port on the surrogate processor board (Figure 4).



**Figure 4 - Remote configuration development**

*Web page*—A simple web interface has been provided for researchers to download and post documents pertaining to the CFTP, <http://cftp.nps.navy.mil/>. This includes an archive of all design documents, ICDs, software interface guides, experimenter design guidelines, design and development tools, and any other pertinent CFTP information. Researchers who desire access to CFTP documents and tools can contact the webmaster through the website for access information.

*Web-based GUI applications*—We are designing web-based applications for basic monitoring of status and health. The same interface that is used for the development of the CFTP will be used during flight operations. Geographically separate experimenters will have access to the same display available to an operator at a ground site.

*Secure shell access*—Secure shell access is also provided to any CFTP collaborator who requires it. This gives the researcher full access to all CFTP documents, as well as the ability to run command line tools that have been developed by other CFTP researchers.

*Scheduling CFTP Access*

Nearly all of the CFTP development can be done concurrently. Different users can all develop their own experiments to run on the CFTP using the tools located on the CFTP server.

The only exception involves the actual programming of the board. There must be some guarantee that once a certain experiment has been loaded into the CFTP board, it will not be disturbed until testing and experimentation is complete.

For this reason, we have developed a method of “checking out” the CFTP board, similar to scheduling meeting times in a conference room. Once a user has successfully signed up for access to the CFTP board for a specified period, no one else can access the board during that time. The active user has sole access to the JTAG port on the board and the serial link to the surrogate processor.

### *Remote Physical Control*

One drawback of remotely operating a system that is under development is the inability to physically manipulate the board. You cannot push the power button or apply a probe to an output pin when you are trying to debug the system from the other side of the country.

*Power control*—In order to remove the need for someone local to the Naval Postgraduate School to continually power on, power off, and power cycle the CFTP stack, we are in the process of developing a method to remotely control power to the system. Safeguards must be put in place such that no damage can be done to the system.

*Status telemetry*—It is also highly desirable to have some physical feedback from the system during the development phase. We are developing a system for monitoring the current, voltage, and temperature of the system. We will also set up a webcam in the lab area so the researcher can have some visual evidence concerning the status of the CFTP.

In addition to monitoring the physical state of the system, we will also devise a method of remotely accessing physical output from the system, such as oscilloscope readings. Since placing probes on the output pins will continue to have to be done manually, the probes will remain on specified output pins (currently I/O pins 10 and 15 on the control FPGA).

## **3. IN-FLIGHT CONCERNS**

Many of the in-flight concerns for operating the CFTP will coincide with the development concerns. We will continue to need an accessible remote link to the CFTP with some sort of scheduling system in place, as well as some form of remote physical control of the system. One of the major differences after launch is the limited uplink to the spacecraft for transferring configurations and other large files.

### *The CFTP Server*

After the CFTP has launched, all collaborators will continue to have access to the CFTP documentation, development tools, and a ground-based development CFTP stack. Researchers can continue to produce experiments to run on the CFTP using our pre-launch scheduling system after it has launched.

### *Flight Queue*

Once an experiment is ready for flight, we will determine the amount of time and resources required for its operation, and on which spacecraft it will be run (Midstar or NPSAT-1). All queued experiments will be given a priority and

handed off to the Midstar or NPSAT-1 ground station to determine when it will get operation time on the spacecraft.

### *Scheduled Uplink*

The day before a scheduled experiment, all of the required files and resources will be transferred to the spacecraft and prepared for operation. Necessary files could include full configuration files, modular configuration files, scripts to manage input and output between the spacecraft and the CFTP board, or data to be transferred to the CFTP.

We will also have the capability to send commands to the CFTP from ground. These could include a request to power cycle the CFTP, load a certain configuration, send a certain configuration to ground for inspection, etc.

### *Scheduled Downlink*

The data from the CFTP experiments will be processed on the CFTP's processor board, and any relevant data will be transferred to the C&DH of the spacecraft and queued for eventual transfer to ground.

The CFTP will also produce a status file at regular intervals (once every 60 seconds) containing status telemetry (see section 2, *Remote Physical Control*), and a time stamp. These status files will be queued on the spacecraft for eventual transfer to ground.

The downlink on NPSat-1 has identical bandwidth to the uplink.

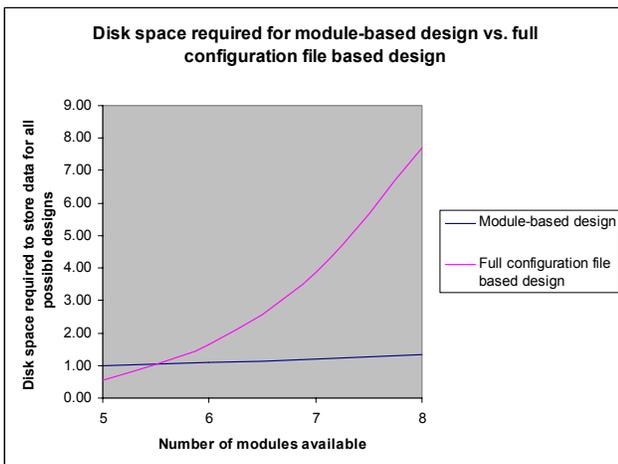
### *Partial Reconfiguration and bandwidth limitation concerns*

The limited bandwidth available for uploading files to the spacecraft necessitates a method of minimizing the size of uploads. The uplink on NPSat-1, for example, is 100kbs, with approximately 30 minutes of access time per day from our NPS ground station. This provides approximately 22 MB of data transfer per day. NPSat-1 has allotted approximately 15 MB of data per day for all experiments onboard. This equates to an average of approximately 5 MB uplink per day for CFTP, with a maximum of the full 15 MB (if no other experiments require uplink) and a minimum of as little as 100 KB if the other experiments have critical data to transfer to the spacecraft.

One of the largest items that the CFTP would need to upload is a configuration file. For the Xilinx FPGAs we are using on the Midstar and NPSAT-1 boards, one configuration file contains 3,601,968 bits, or about 440KB. Larger FPGAs can run upwards of 3-4MB per configuration file. The Virtex-II XC2V8000 chip, for example, has a bitstream configuration file of 3.5 MB. Future iterations of the CFTP project intend to implement newer, higher gate count Xilinx chips, such as the Virtex-II or Virtex-II Pro, so limiting file size will become even more critical.

With a total uplink of less than 5 MB to the spacecraft on average, it is very important to try to limit the number and/or size of configuration files. One method we have investigated is using partial reconfiguration to allow us to transfer only partial bitstreams to the spacecraft rather than full configuration files, greatly reducing the bandwidth required to upload the necessary files for a CFTP experiment. The Xilinx Virtex parts we are using provide partial reconfiguration features, allowing the user to reconfigure the FPGA, changing as little as one frame at a time [7]. Each frame in the XCV600 (which is structurally identical to the radiation hardened XQVR600 part) contains 960 of the 3,601,968 configuration bits [8].

The main idea behind partial reconfiguration is a modular approach to system development. We will be exploiting the fact that a large number of the CFTP experiments will have aspects of their configurations in common. This will allow us to piece together an experiment using several different configuration modules that are located on the spacecraft. These pieces can be used in different configurations to create several different experiments. This also allows us to upload only partial bitstreams if we need the CFTP to perform an additional task. Each module can then perform a specific task that is not already located on the spacecraft, as opposed to uploading a full 440KB configuration file.



**Figure 5 - We are assuming an average of 4 configuration modules are needed to create a full design. The disk space required for designs using full configuration files is determined based on the number of viable configurations possible using module-based design.**

Suppose, for example, that each configuration requires an average of 4 modules, plus the full base configuration. If there were a total of 8 modules and one base configuration local to the spacecraft, that provides the pieces to generate 70 (10 choose 4) unique configurations. If only 25 percent of these configurations are valid and useful designs, we are still left with about 16 unique configurations at a fraction of

the disk space required to store full bit files. Supposing each module is approximately 1/4 of the size of a full configuration file (since we are assuming an average configuration uses 4 distinct modules), we would get the functionality of 16 unique designs, but only use the disk space required to store 3 full configurations.

Figure 5 compares the amount of disk space needed for module based design to the amount of disk space needed to store full configuration files. We quickly make large gains in disk space savings by utilizing multiple modules in our designs. These savings can be translated to bandwidth savings as we begin to transfer new designs to the spacecraft.

The Xilinx tool bitgen simplifies the task of creating partial bitstreams. There are primarily two separate methods of partial reconfiguration, module-based and difference-based. Difference-based simply takes the design from which you want to create a bitstream, compares it to a bitstream that is already loaded on the FPGA, and generates an additional, much smaller bit file of only the differences between the two configurations. This new bitstream, which can be as much as 95% smaller than a full bitstream, can then easily be transferred up to the spacecraft and loaded into the FPGA while the current instantiation continues to run. [9]

The difference-based partial reconfiguration method would be useful for making small changes to a currently running experiment, or for repairing errors that have been found during operation [7]. It has two major benefits over doing a complete reload. 1) The transfer time to the spacecraft is greatly reduced and 2) the operation of the experiment does not have to be interrupted.

Difference-based partial reconfiguration is somewhat limited in its applications, however. In order to develop and run an entirely new, complicated experiment, module-based partial reconfiguration is better suited. Module-based partial reconfiguration involves having many different application modules local to the spacecraft, as well as a base FPGA load. The base load includes all of the inputs and outputs of the experiment, but no internal circuitry. The individual application modules contain all of the functional applications (Figure 6).

Module-based partial reconfiguration allows us to instantiate the base FPGA load, and then add in any additional application modules the experiment requires. These application modules could contain processor soft cores, Triple Modular Redundancy (TMR) circuitry, Error Detection and Correction (EDAC) circuitry, or other specialized tasks.

The process of determining which modules are needed to complete a certain configuration and obtaining those modules is illustrated in Figure 7.

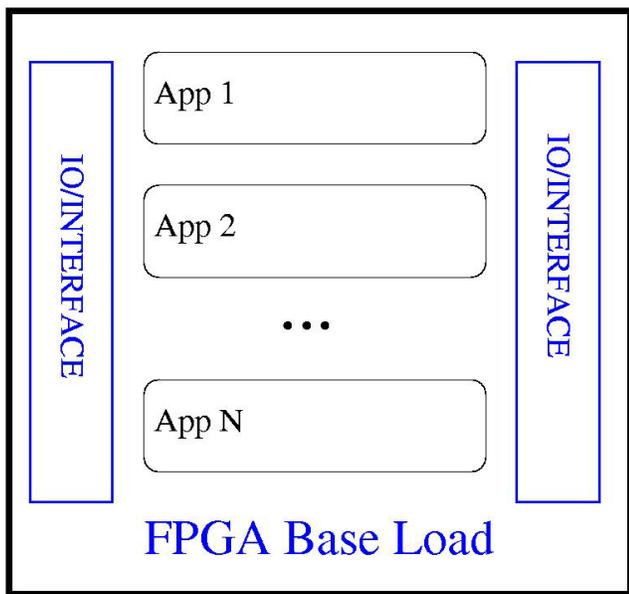


Figure 6 - The framework for partial FPGA loads

The base FPGA loads in this illustration are blank slate configurations that determine only inputs, outputs, and interface logic of the instantiation. Ideally, we would need only one base FPGA load between all experiments, but we always keep the option open of adding additional base loads if a certain experiment has unique I/O or interface needs.

The configuration modules contain the actual functionality of the experiment. We will make an effort to develop small independent modules that can effectively be pieced together to create many different experiments. This makes development more efficient, eliminating redundant code, as well as reducing the necessity to transfer large configuration files over the limited uplink to the spacecraft.

The Flash memory on the CFTP board contains a number of configuration modules and base FPGA loads. Before launch, the flash memory will be populated with as many configuration modules and base loads as we have prepared for experimentation. This provides great flexibility and minimizes the need to transfer large amounts of data over the limited uplink. The intent is to have as many pieces available on the spacecraft from the beginning to create new experiments as the need arises.

### Configuration Modules and Base FPGA Loads

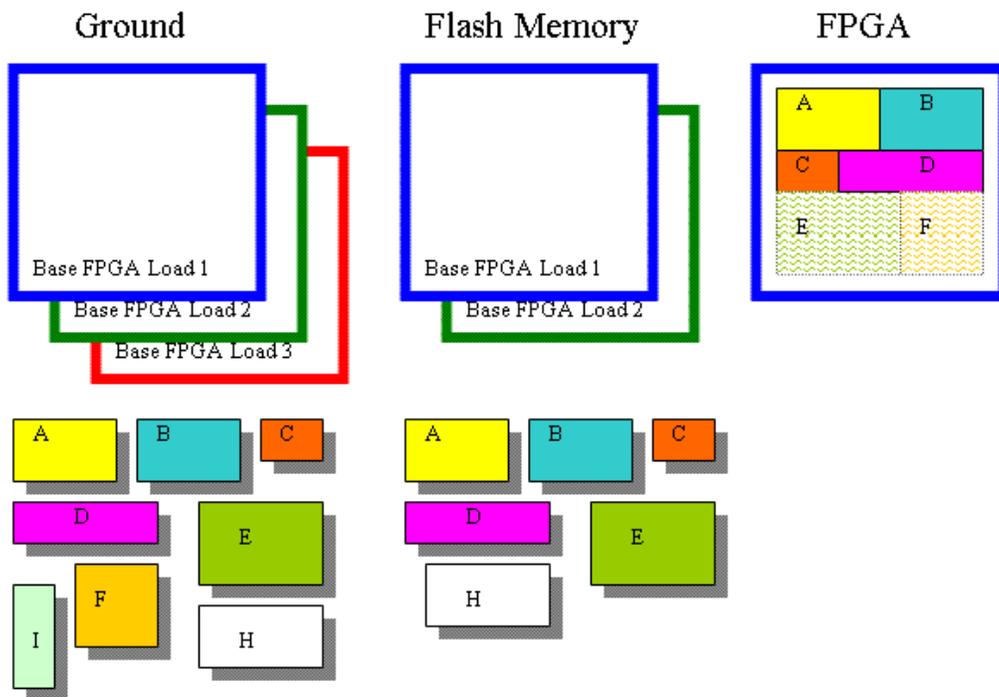


Figure 7 - Illustration of module-based partial reconfiguration

For example, suppose we have a current configuration instantiated on the FPGA consisting of the base FPGA load 1, and modules A, B, C, and D. We need 2 additional modules, E and F, in order to complete our desired configuration.

On the ground, we determine which modules are necessary to complete our desired configuration. We note that modules A, B, C, and D are already loaded on the FPGA base load 1, and that module E is located in the Flash memory. That leaves module F to be transferred from ground to the spacecraft.

#### 4. CONCLUSIONS

The CFTP project provides a unique opportunity to develop effective methods of remote configuration and operation. The widely spaced researchers who are contributing to the project necessitate the development of tools to remotely monitor and operate the system. These tools can easily be transferred to the on-orbit CFTP, facilitating the design of on-orbit configuration.

The extensive collaboration on this project also facilitates the distribution of data to interested parties after the CFTP has launched. The web-based system of providing access to CFTP tools and documentation supplies a perfect platform for data distribution. Anyone who is interested in accessing data that has resulted from a CFTP experiment can directly acquire it from the CFTP server.

The CFTP project also encourages limiting file size and uplink usage. The ever-changing repertoire of experiments that will be run on the CFTP require frequent uploads of new configurations and scripts to the spacecraft. This encourages taking measures to decrease the number and size of the configuration files necessary for reconfiguring the system. Both module-based and difference-based partial reconfiguration are excellent tools for realizing this goal.

#### REFERENCES

- [1] J. Lyke, "Reconfigurable Systems: A Generalization of Reconfigurable Computational Strategies for Space Systems," 2002 IEEE Aerospace Applications Conference Proceedings, 9-16 March 2002.
- [2] C. A. Hulme, H. H. Loomis, A. A. Ross, and R. Yuan, "Configurable Fault-Tolerant Processor (CFTP) for Spacecraft Onboard Processing," 2004 IEEE Aerospace Applications Conference Proceedings, 7-12 March 2004.
- [3] M. Wirthlin, E. Johnson, N. Rollins, M. Caffrey, and P. Graham, "The Reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets," 2003 IEEE Symposium on Field-Programmable Custom Computing Machines Proceedings, 9-11 April 2003.
- [4] D. A. Ebert, "Design and Development of a Configurable Fault Tolerant Processor (CFTP) for Space Applications," Master's Thesis, Naval Postgraduate School, Monterey, California. June 2003.
- [5] R. Yuan, "Triple Modular Redundancy (TMR) in a Configurable Fault Tolerant Processor (CFTP) for Space Applications," Master's Thesis, Naval Postgraduate School, Monterey, California. December 2003.
- [6] C. A. Hulme, "Testing and Evaluation of the Configurable Fault Tolerant Processor (CFTP) for Space-Based Applications." Master's Thesis, Naval Postgraduate School, Monterey, California. December 2003.
- [7] M. Gokhale, P. Graham, E. Johnson, N. Rollins, and M. Wirthlin, "Dynamic Reconfiguration for Management of Radiation-Induced Faults in FPGAs," 2004 International Parallel and Distributed Processing Symposium, 26-30 April 2004
- [8] Xilinx, Application Note 138, "Virtex FPGA Series Configuration and Readback," 5 November 2001.
- [9] Xilinx, Application Note 290, "Two Flows for Partial Reconfiguration: Module-based or Difference-based," 9 September 2004.

## **BIOGRAPHY**

**Mindy Surratt** is a Research Assistant in the Electrical and Computer Engineering Department at the Naval Postgraduate School in Monterey, CA. She received her Bachelors in Mathematics from the University of Maryland in College Park in 2002. She provides interface and communications support for the CFTP project.

**Herschel H. Loomis, Jr.** is a Professor of Electrical and Computer Engineering at the Naval Postgraduate School in Monterey, CA. He has a Bachelors degree in Electrical Engineering from Cornell University, a Masters in Electrical Engineering from the University of Maryland, College Park, and a Ph.D. degree in Electrical Engineering from the Massachusetts Institute of Technology.

**Alan Ross** is the TENCAP visiting chair professor in Space Systems at the Naval Postgraduate School. He has a BS in Engineering from UC Davis, a Masters in Electrical Engineering from the Air Force Institute of Technology, and a PhD in Computer Engineering from UC Davis.

**Russ Duren** is a Professor at Baylor University in Waco, Texas.

## APPENDIX A

### *CFTP Terms*

**CFTP Board** – The reconfigurable experiment board itself

**Surrogate processor board** – The link between the CFTP board and the CFTP server on the breadboard system.

**Processor board** – The link between the CFTP board and the spacecraft on the flight system.

**CFTP Stack** – The CFTP board, processor board, and power supply together

**CFTP server** – The web-accessible development PC which is connected to the CFTP stack through the serial port

**Control FPGA** – FPGA X1 on the CFTP board that interfaces with the processor board.

**Experiment FPGA** – FPGA X2 on the CFTP board which carries out the various CFTP experiments.

### *Other terms*

**Configuration memory** – EEPROM that stores initial configuration for the CFTP

**Flash memory** – Non-volatile memory that stores alternate configurations for the CFTP

**Partial reconfiguration** – ability to reconfigure a portion of the FPGA while the remainder of the design is still operational

- **Module-based partial reconfiguration**
- **Difference-based partial reconfiguration**