Seamless Hardware Module Swapping for Partially Reconfigurable Stream Processing Systems

Abelardo Jara-Berrocal, Joseph Antoon and Ann Gordon-Ross

NSF Center of High Performance Reconfigurable Computing (CHREC)
Department of Electrical and Computer Engineering
University of Florida. Gainesville, FL 32611
berrocal@chrec.org, joe.antoon@ufl.edu, ann@chrec.org
http://www.chrec.org

Abstract
Adaptation is a key requirement for space-based systems due to harsh and rapidly changing conditions outside of the Earth's ecosystem. These systems must be able to adapt data rates to counter changes in available bandwidth (e.g., inter-satellite and satellite-earth communications) and bit-error rates [2] (e.g., radiation, single event upsets (SEUs)) and adapt functionality to account for human error [1]. Additionally, in some situations a designer may need to repurpose a space-based system for an entirely different mission. To provide these adaptabilities while limiting system overhead (e.g., system down-time, area), the partial reconfiguration (PR) capability of modern FPGA devices is advantageous for such missions. PR devices provide the flexibility to dynamically load and unload internal hardware components/modules on demand without forcing a system reset or shutdown, which could result in catastrophic system failure due to the unavailability of vital system functionality.

PR-enabled system fabrics typically consist of a static region and a set of partitions connected to the static region. The static region is configured during startup, and contains all hardware functionality that remains fixed during system execution. One or more partitions can be independently reconfigured while the rest of the system, including other partitions, remains operational. This architectural structure enables efficient dynamic runtime placement of hardware modules inside available partitions, a technique known as dynamic hardware module switching. Hardware module switching is an enabling technology in novel operating system frameworks [3] and artificial intelligence systems, and is comparable to the dynamic swapping of components in an FPGA-based system-on-a-chip (SoC).

Hardware module switching capabilities are particularly advantageous for reconfigurable stream processing systems (RSPSs), such as those which process sensor data in space-based missions. RSPSs are composed of a set of hardware modules and software modules connected together to transform a data input stream into a processed data output stream. On FPGA SoC architectures, software modules execute on an embedded microprocessor (hard or soft core) implemented inside the FPGA. Software modules communicate with the hardware modules through either industry standard interfaces, like Fast Simplex Links (FSLs), CoreConnect, the Advanced Microcontroller Bus Architecture (AMBA), or custom-designed interfaces. An RSPS may also use an inter-module communication network to pass a data stream between peripheral components without interrupting software execution. An example of an overall system architecture is an audio/video processing RSPS. The RSPS may include hardware functionality such as MPEG decoding, rasterizing of the video fields, and audio and video digital-to-analog conversion while the software module functionality may include digital audio filtering, subtitle insertion, and audio data parsing to skip silent periods. Since input data particulars (e.g., illumination, image resolution, contrast, etc.) dictate the type of decoding hardware modules or desired filtering effects, the RSPS can dynamically exchange/replace specialized hardware modules. In addition, since application bandwidth requirements dictate processing throughput, different hardware modules specialized for different compression formats can be included to alter processing throughput.

For space-based missions, hardware availability is a critical requirement, and any hardware module switching should be fast and incur minimum stream processing interruption, a goal we define as seamless

module switching. Unfortunately, a serious drawback of PR is lengthy reconfiguration time, as reconfiguration of a reasonably sized partition (around 800 slices on Xilinx Virtex-4 FPGAs) can take on the order of tens to hundreds of milliseconds [4]. A possible solution to lengthy reconfiguration times is to overlap system processing with PR. A new module aimed to replace an older module can be placed in a separate partition while the RSPS continues operation uninterrupted. After reconfiguration, streaming data flows entering the older hardware module must be redirected to the new hardware module, and the new hardware module's output must be redirected to the appropriate downstream hardware module. The inter-module communication architecture can facilitate hardware module swapping by enabling dynamic redirection of the streaming data flows between the hardware modules [6]. In general, the inter-module communication architecture enables RSPS runtime assembly, which is the process of dynamically constructing an RSPS at runtime from a subset of hardware modules ready to execute. A system's ability to perform RSPS runtime assembly significantly enhances reconfigurability, and thus presents a mechanism to effectively hide reconfiguration time.

In this work, we design and demonstrate a scheme for seamless module switching using the VAPRES (Virtual Architecture for Partially Reconfigurable Embedded Systems) architecture introduced in [5]. Our system performs real-time target tracking of a sample object (a bouncing ball for this demo, but the target could be any object such as a moon, planet, asteroid, etc.) followed by a camera. Target tracking is a path/trajectory prediction method used to follow environmental targets such as vehicles, missiles, animals, hostiles, or even unidentified objects. Since Kalman filters are a popular target tracking kernel (FPGA implementations of Kalman filters demonstrate excellent performance in comparison to software implementations) [7], we chose Kalman filters as processing modules to implement the target tracking functionality in our system. Previous reported FPGA implementations of target tracking systems based on Kalman filters [7] do not consider targets with different characteristics, such as widely varying speeds and tracking criticality, resulting in increased processing time and energy consumption. Our implemented system leverages contributions over previous work by performing a seamless module switching scheme over two different versions of Kalman filters: a basic Kalman filter (suitable for targets with non-constant velocity) and a constant-gain Kalman filter (suitable for targets with constant velocity).

Acknowledgments
This work was supported in part by the I/UCRC Program of the National Science Foundation (NSF) under Grant No. EEC-0642422 and by the NSF under Grant No. ECCS-0901706. We gratefully acknowledge tools provided by Xilinx and Vasudha Chaudhary and Srikanth S. for the design and implementation of the camera interface.

References
[1] Anderson Y. and Scarffe V. and Heventhall W. and Doody, D. A. Solving Cassini's Data Glitch Problem during Coherency Mode Transition for Titan Radar Observation, Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2006
[2] Arnon S and Kopeika N. Adaptive bandwidth for satellite optical communication. IEE Proc. on Optoelectron,. April 1998 Volume 145, Issue 2, p.109-115
[3] Handa M. and Vemuri R. An efficient algorithm for finding empty space for online fpga placement. Proceedings of the 41st annual Design Automation Conference (DAC'07). ACM, New York, USA, 960–965.
[4] Hymel R. and George A. and Lam H. Evaluating partial reconfiguration for embedded FPGA applications. In Proceedings of International Symposyum on High Performance Computing (HPC'07)
[5] Jara-Berrocal A. and Gordon-Ross, A. VAPRES: Virtual Architecture for Partially Reconfigurable Embedded Systems. In Proceedings of the Design, Automation Test in Europe Conference Exhibition. (DATE '10).
[6] Jara-Berrocal A. and Gordon-Ross, A Runtime temporal partitioning assembly to reduce FPGA reconfiguration time. In Proceedings of International Conference on Reconfigurable Computing and FPGAs (Reconfig '09)
[7] Meltzer A. and S. Soatto. Fast visual feature selection and tracking in a hybrid reconfigurable architecture. Proceedings of the Workshop on Applications of Computer Vision (ACV'06)