

CHREC Tools (existing and upcoming)

NSF CHREC Center

Latest update: March 10, 2009

Tool name: Assorted core libraries, benchmarks, and applications

Project(s): Various projects, all four CHREC sites

Summary: TBA

Tool name: Autonomy

Project(s): V3-09

Summary: This tool suite and API provides a different use-model for FPGAs that could be used to create new capabilities to members' products, and provide a potentially new method for fault recovery. This tool is intended to ultimately make a system that is to a degree "self-aware" and can configure itself autonomously at a fine granularity in response to external stimuli. A system with these capabilities could do system-level link maintenance (cognitive and software-defined radios), change the topology of a computation (HPC), universal modularity for in-system adaptation, or rework a system in response to a highly localized failure.

Tool name: Bitstream Relocator

Project(s): F4-08

Summary: Partial Reconfiguration (PR) of FPGAs presents many opportunities for application design flexibility, enabling tasks to dynamically swap in and out of the FPGA without entire system interruption. However, mapping a task to any available PR region (PRR) requires a unique partial bitstream for each PRR. This replication can introduce significant overheads in terms of bitstream storage and communication requirements. Bitstream relocation is a process which allows transforming a single partial bitstream to map to any available PRR. Bitstream relocation has high potential for usage on adaptive RFT (Reconfigurable Fault Tolerance) and runtime management of FPGA resources. The bitstream relocation tool is C-based and supports Xilinx Virtex-4 devices.

Tool name: DAPR: Design Automation for Partially Reconfigurable FPGA Systems

Project(s): F4-09

Summary: Partial reconfiguration (PR) is a powerful feature of Xilinx FPGAs which permits individual FPGA region reconfiguration at the runtime while the remaining FPGA regions continue operation. PR systems have high potential for applications requiring hardware adaptation or time-multiplexing of a subset of the FPGA resources. However, design and implementation of PR systems currently is a very highly specialized process involving many manual and time-consuming steps. DAPR is a C- and Perl-based software tool set that reads a PR system description implemented by the system designer using high-level PR-augmented VHDL. DAPR automatically performs module profiling, module mapping, and system floor-plan creation. DAPR's output files are used to automatically run the standard Xilinx PR Design Flow to implement the final PR system.

Tool name: EDIF TMR Tools

Project(s): B3-08

Summary: This software provides several APIs for parsing and manipulating EDIF netlists. A key component of this API is the ability to automatically triplicate EDIF circuit designs and insert voters to support Triple Modular Redundancy (TMR). A variety of other reliability tools are included with this package as well as a user guide. The original development of this tool at BYU predates CHREC, but was updated in CHREC and posted as open source via project B3 in 2008.

Tool name: Intermediate Fabrics Exploration Tool

Project(s): F2-09

Summary: This tool suite assists the exploration of intermediate fabrics (IFs), virtual PAR-specialized reconfigurable fabrics that act as an intermediate translation layer between a netlist and an underlying physical device. Preliminary results show that IFs achieve > 600x speedup in PAR time for commercial-off-the-shelf FPGAs, enabling rapid in-circuit debugging. IFs also enable portability across devices, acting as an abstraction of underlying device resources. The suite includes tools to generate HDL for a variety of IF architectures from a flexible specification, as well as PAR tools for those architectures. Current work is focused on reducing area overhead and integrating the suite with existing FPGA vendor tool chains.

Tool name: PFIF: Portable Framework Interface

Project(s): G5-08

Summary: This tool is an Eclipse-based IDE for developing applications on various reconfigurable computers. The user chooses the targeted reconfigurable platform (Cray XD1, SGI RC100) and also specifies memory interfaces to be used from a drop-down list. The tool guides the user through a simple wizard and generates the required template files in a high-level language (Impulse-C), to assist the user in developing the application. The tool also generates the necessary wrapper files as well as PFIF files specific to the selected platform, and carries out the required synthesis, PAR, and bitstream generation.

Tool name: Profiling and Scheduling IDE

Project(s): G1-07, G6-08

Summary: This tool provides a graphical front-end for profiling and analyzing applications written in C, for visualizing and identifying computation- and data-intensive functions. The application developer can also view the call graph of the application. The call graph is treated as a DFG and partitioned/scheduled across multiple FPGA configurations, and the corresponding task clusters are indicated in the call graph. Instead of using C source files, the user also has the option of providing the task graph of the application as an input to the tool directly, by using a text file that describes the graph.

Tool name: RAT: RC Amenability Test

Project(s): F3-07, F3-08, F1-09

Summary: This tool is an Excel worksheet that, using the RAT performance model, enables analytical performance modeling of a specific RC application targeting a specific RC platform. Users provide key application and platform parameters as defined by the tool, which are used to calculate the performance predictions. RAT encompasses both a device-level model for analysis of single-FPGA systems (F3-07, F3-08) and modeling aspects being leveraged and added (F1-09) from LogGP to provide system-level prediction for scalable, multi-FPGA platforms. The goal is to maintain efficient and reasonably accurate prediction prior to expensive implementation iterations.

Tool name: RCML: RC Modeling Language Editor and Analysis Toolset

Project(s): F1-08, F1-09

Summary: This tool is an Eclipse plug-in that allows users to create, save, and edit models based upon the RC Modeling Language (RCML) created in CHREC in 2008. The graphical editor provides users the ability to drag-and-drop model components and edit models (application and machine) through graphical interfaces and dialogs. Analysis tools are integrated into the RCML

editor allowing users to analyze the performance and behavior of their RCML model using system-level methodologies and tools developed in CHREC, namely RAT and RCSE. A core-level modeling and analysis methodology was also developed (F1-08) and will be integrated into this tool. New methods for automated mapping and design-space exploration of RCML system models are in development and will be integrated into this tool as well.

Tool name: RCSE: RC Simulation Environment

Project(s): F1-07, F1-08

Summary: This tool is a library of discrete-event models built within the commercial simulation environment called Mission-Level Designer (MLD) to support the rapid simulative analysis of RC systems, comprised of applications and machines and their mappings. The application models take as input a script representing a trace of the application under study. Scripts represent the application as a sequence of major events to drive simulative analysis. The architecture models in the library are highly generic components that use parameter files to tune the generic models to behave as a specific device technology.

Tool name: ReCAP: Reconfigurable Computing Application Performance

Project(s): F2-07, F2-08, F2-09

Summary: ReCAP provides the ability to instrument, measure, and present performance data from a reconfigurable application (e.g. multi-CPU/FPGA application) via a major extension to the Parallel Performance Wizard (PPW) tool that provides performance analysis for conventional parallel systems. The hardware portion of the application may be written using a hardware description language (VHDL or Verilog) or a high-level language (Impulse-C or Carte-C). ReCAP includes an HDL instrumenter to gain access to application data as well as a hardware measurement module (HMM) to record, store, and permit runtime access of application performance data on the FPGA. ReCAP also provides knowledge-based bottleneck detection capabilities for quickly locating common performance bottlenecks and is currently being extended to provide runtime debug capabilities particularly useful to large-scale (e.g. multi-FPGA) applications.

Tool name: SCF: System Coordination Framework Toolset

Project(s): F1-09

Summary: SCF features an Eclipse plug-in that allows users to describe applications as task graphs and define a mapping of tasks onto resources. A backend parser reads the task graph and mapping information to generate communication infrastructure. It auto-generates communication constructs for all the tasks in the application which are written in one of the supported IDEs and for a SCF-compliant target system. Also to be included with SCF is an early prototype of extensions to the SHMEM parallel programming model and/or implementation for coordination of accelerators in next-generation, heterogeneous HPC machines.

Tool name: SPFFI: Simple Portable FPGA Fault Injector

Project(s): F4-08, F6-09

Summary: SPFFI is a simple fault injector that emulates radiation-induced SEUs onto Virtex-4 FPGAs. This process is performed by injecting faults into the configuration memory of the FPGA. The architecture of the tool is divided into four major components: Campaign Generator, Injection Engine, Test Generator, and Logging Engine. The Campaign Generator uses a debug bitstream file generated by the Xilinx ISE tools to determine the random locations of the errors to be injected. The Injection Engine creates crafted partial bitstream files which are then programmed over JTAG to inject and remove faults. The Test Generator is a user-provided program that verifies the

operation of the design of the FPGA and provides that information to the Logging Engine, which combines and saves all the injection details into a log file that can be used for post-mortem analysis.

Tool name: VAPRES: Virtual Architecture for Partially Reconfigurable Embedded Systems

Project(s): F4-08, F4-09

Summary: VAPRES is a multipurpose ready-to-use base platform for rapid-deployment of partially reconfigurable embedded systems. VAPRES is created around the Xilinx MicroBlaze soft-core processor and currently targets Virtex-4 devices. Multiple H/W coprocessors are connected to the MicroBlaze through FSL (Fast Simplex Link) streaming interfaces. Special features of VAPRES include: 1) independently clocked HW coprocessors; 2) H/W coprocessors runtime reconfiguration; and 3) inter-coprocessor communication support via SCORES (Scalable Communication Architecture for Modular Reconfigurable Systems). VAPRES includes architectural parameters to enable application customization. A set of S/W tools for rapid-development of VAPRES S/W applications and reconfigurable H/W coprocessors is also included for the VAPRES system.