Project Progress and Future Work

A discussion of the progress on our project, problems encountered, and avenues of future work

---

Contents

Introduction

This document is a brief summary of our work on the Hardware-Software Co-Design project with Xilkernel (formerly, QNX). It highlights the important parts of our work, while discussing the various problems we encountered along the way. It concludes with avenues of future work that future teams can focus on.

Progress

1. Finished the process of integrating Impulse C into the Xilkernel real-time operating system.
2. Created several different example Xilkernel projects that utilize Impulse C applications.
3. Developed partial automation scripts to aid user during the integration process.
4. Implemented and tested instrumentation library (C header file) that can be used to aid developer in gather cycle-based measurements of execution time.
5. Generated extensive documentation covering the integration process, our results with using the profiling tools and running some experiments, and design notes to provide insight into POSIX-compatible Impulse C implementations.
6. Attempted to add partial reconfiguration support to the Xilkernel base system.

Current Problems

1. Could not obtain license for PlanAhead in the SE lab, despite using a floating license from the CE department. Future teams will need to work closely with Emilio Del Plato (the current CE systems admin) early on to resolve this issue.
2. A PlanAhead project was created in the CE lab, but coming up to speed with the extensive number of features and the complexity of the Impulse C project and the base

system was very time consuming. Special attention needs to be paid to the connectivity between the Impulse C module and the rest of the system, or else reserving an area for the reconfigurable partition will fail. Hopefully future teams will benefit from the documentation provided by this effort.

3. The PlanAhead software available in the CE labs (version 12.3) will not implement a partial reconfiguration design. The implementation process will begin and then queue indefinitely with no error or other explanation. The cause of this problem is unknown. Attempts to resolve this issue with Emilio and Sam Skalicky (PhD candidate in the CS department) were fruitless.

Future Work

1. Explore and finish setting up a partial reconfiguration compliant system that treats Impulse C modules as blackboxes with the same entity-level interface. This implies that each Impulse C module must have the same parameter list for every hardware process that is partitioned into the FPGA. The contents of this interface should be varied, as they are dependent on the specific application of Impulse C.
2. Explore and document the Xilinx and Impulse C interface in more detail.
3. Explore dynamic partial reconfiguration of the FPGA by a task within the Xilkernel RTOS.
4. Follow the translation script logic and use it to add Impulse C support to the Xilinx SDK eclipse environment. This will help make the integration process more inclusive within the Xilinx toolchain, with the only external development done using the Impulse C tools.
5. Add macro support to Impulse C translation script to dynamically generate parameter lists, structures, and initialize the fields.