An FPGA-Based Experiment Platform for Hardware Software Co-Design

Yajuvendra Nagaonkar and Mark Manwaring PhD

Electrical and Computer Engineering, Brigham Young University, Provo UT 84602, USA

Introduction

To create a rapid-prototyping environment for computer architecture experiments and for simulation of Hardware-Software Co-Design problems.

The system is targeted to contribute to research efforts and to be used in computer architecture education.

1. Architecture Emulation

The FPGA and SRAM combination allow for emulation of a processor soft-core with different architectural features.

The microprocessor and ROM allow for the system to function as a stand alone machine. The ROM stores the FPGA configuration bit-stream, which is loaded on start-up.

The system provides a real-time prototyping environment for custom soft-cores with all the available resources and standard communication interfaces with other peripheral devices.

2. Academic Environment

This system can prove very beneficial in an academic setting such as a course on Computer Architecture.

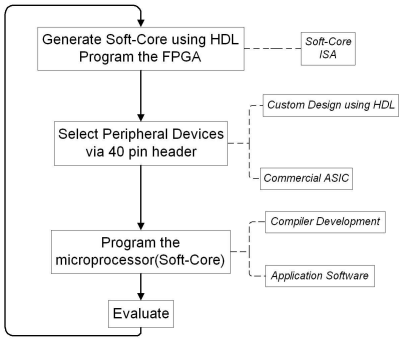
It allows for experimentation with different computer architecture concepts in different stages.

A student equipped with the knowledge of a HDL is able to implement hardware relating to computer architecture concepts.

The rapid prototyping feature allows for implementing different concepts within the time frame of a semester.

Advanced architecture features can be experimented with, and the abundant I/O on the modules enable interfacing with various peripheral devices.

Co-Design Flow



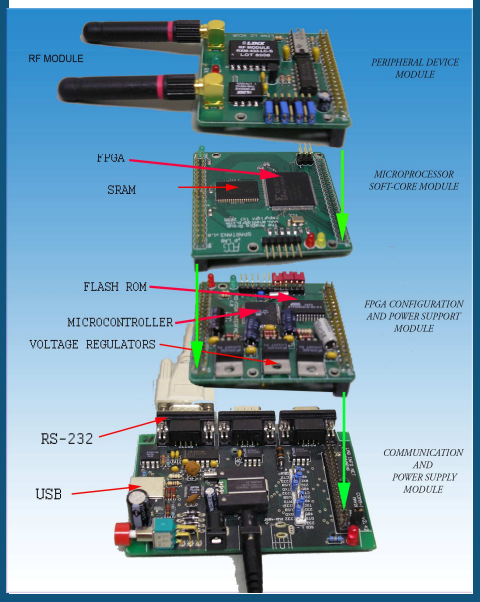
Example of a Co-Design Problem

What is the most efficient soft-CPU (Microprocessor soft-core) to support application on an RF module?

The RF module has a weak communication link and requires error detection and correction.

The co-design problem can be approached by either moving the error detection and correction to hardware or software.

An appropriate soft-CPU has to be selected or designed, and various choices and tradeoffs can be evaluated.



Hardware Software Co-Design

The problem of Co-Design consists of:

Processor Selection : In this system, the processor can be selected and implemented as a soft-core on the FPGA module.

The soft-core selected might be a standard available soft-core processor such as Microblaze® or it could be a custom or modified soft-core.

Programming the Processor: Software needs to be developed, based on the processor selected. Optimizations like target specific compilers and microprocessor specific application software is developed.

Peripheral Devices: The selection of peripheral devices is based on factors of complexity, availability, interface to the world and on-board standard I/O.

Based on the above factors the peripheral device can be implemented as a soft-core on the FPGA or an actual device that is interfaced to the microprocessor soft-core.

Components

Communication and Power Supply Module

•Commercial Voltage Supply Compliant (9V-15V).

•Standard Communication platforms- RS-232, USB.

Microcontroller module

•8-bit RISC Atmel Atmega128 Microcontroller.

•8-Mbit Atmel Dataflash ROM.

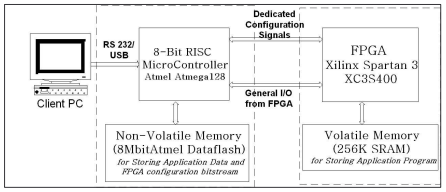
•Voltage Regulators (5V(source) → 3.3V, 2.5V, 1.2V).

FPGA Module

•400K Gate Xilinx® Spartan-3® XC3S400 FPGA.

•(256K X 16) 4Mb Cypress Semiconductor® SRAM.

System Overview



The interface is kept simple and intuitive to the user by providing a standard communication interface between the PC and the module.

The configuration bit-stream is downloaded via RS-232/USB to the Flash ROM via the microcontroller, and the FPGA is configured with a slave serial arrangement, avoiding the complex vendor specific software and expensive configuration hardware.

Conclusion

The system offers an efficient platform for computer architecture emulation and hardware software co-design.

We anticipate several improvements in the module design such as combining the microcontroller and FPGA modules into a single compact PCB.

Other possible improvements would include adapting different vendor technology specific FPGAs from Alterra®, Atmel®, Actel®, etc. The selection would depend on any architectural features of the FPGA which would support maximum implementation of microprocessor soft-cores.

Contact

Yajuvendra Nagaonkar yaju@ee.byu.edu

801-422-1240

Mark Manwaring, PhD manwaring@ee.byu.edu

801-422-7582