# The University of _____ Reduced Instruction Set Computer (MARC)


## Abstract

We present our design of a VHDL-based, RISC processor instantiated on an FPGA for use in undergraduate electrical engineering courses and research. Our architecture incorporates a simple instruction set serially executed in a 5-stage cycle (fetch, decode, execute, memory, write back). The design utilizes SRAM memory to store instructions and pushbuttons, switches, LEDs, and 7-segment displays provide feedback and user input. Specifically, we use the Altera Cyclone II to instantiate our system running on the Altera DE2 development and education board. We also leverage the associated CAD tools to build this system. Our implementation allows students the ability to easily modify and adapt their designs for a specific application. We target computer architecture courses where students learn fundamental CPU concepts such as the data path, control unit, ALU, register file, and memory.


## 1. Introduction

At the University of _____, we study and utilize FPGAs in many of our electrical engineering courses. In the Department of Electrical Engineering and Computer Science, EE majors take a number of core courses to include EE360 (Digital Logic) and EE375 (Computer Architecture using VHDL). The program also offers a number of depth threads to provide focus on a particular area of electrical engineering (i.e. computer architecture). Students learn digital logic in EE360 through the use of MSI (medium-scale integration) logic devices, CPLDs (complex programmable logic devices) and FPGAs. In EE375, students continue to learn computer architecture and VHDL by studying the MARC processor. The final course in the digital thread is EE484 (Advanced Computer Architecture using VHDL) covering topics to include branch prediction, static and dynamic scheduling, multiprocessors, memory and cache coherency, interconnects, and various I/O interfaces. The purpose of this paper is to fully describe the MARC architecture implementation on an Altera DE2 board and how it can be utilized in undergraduate electrical and computer engineering coursework.

## 2. Related Work

Many universities have FPGAs and in-house VHDL processor cores integrated into their curriculum. We have leveraged the educational resources of Altera's University Program. This support has been instrumental in adapting our digital thread curriculum to utilize FPGAs in the classroom. In EE360, Digital Logic, students learn the basics of combinational and sequential digital logic to include the design and implementation of finite state machines (FSMs) and algorithmic state machines (ASMs). In order to develop their understanding of FSM and ASM design, they are given a practical exercise using the iRobot® Create robot. As part of the process, students develop their FSM or ASM diagrams based upon their interpretation of the given signals (i.e. robot position and movement) and problem statement and implement their design in VHDL for synthesis onto an Altera Cyclone II FPGA on the DE2 board (Figure 1).

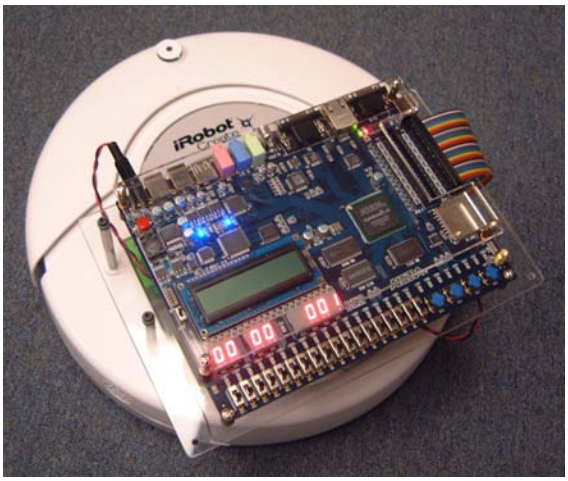

Figure 1: DE2 board controlling Create robot.[1]

EE484, Advanced Computer Architecture using VHDL, builds upon the computer architecture course. As noted in the title, the course utilizes VHDL throughout the semester implementing designs on the DE2 board. This senior-level elective uses multiple soft processor cores on an FPGA to study multi-core designs in a laboratory exercise. Our students instantiate multiple cores using the Altera Nios® II soft processor and associated Computer-Aided Design (CAD) tools. We demonstrate that multiprocessor systems can be developed, implemented, and studied by undergraduate students due to the availability and accessibility of design tools and FPGA development boards.[2] Furthermore, these systems enhance the learning of multiprocessors and aptly compliment advanced computer architecture courses covering topics such as shared memory, synchronization, sequential consistency, and memory coherency.

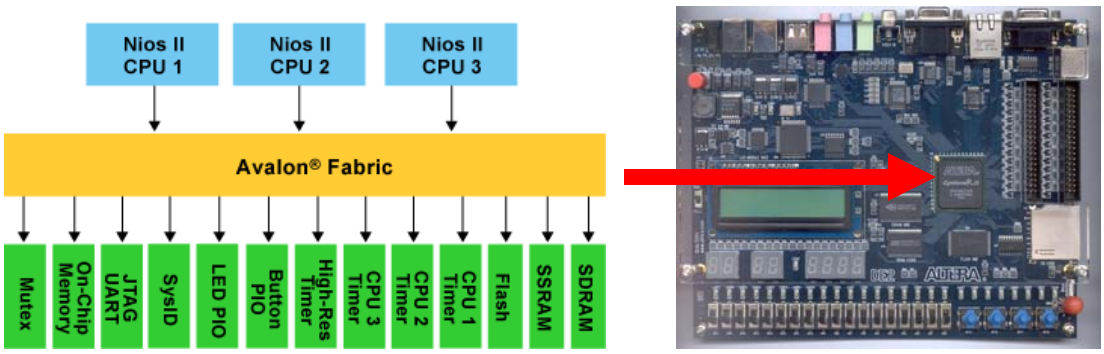


Figure 2: EE484 Multiprocessor Laboratory Exercise Block Diagram.[3]

We have also looked at FPGA work at other universities, namely Cornell University and Georgia Tech. At the Cornell University School of Electrical and Computer Engineering, students learn embedded design by using FPGAs to develop SOC (system-on-chip) devices. Their program has done extensive work with using FPGAs for embedded control and in electronic design.[4] At Georgia Tech, FPGAs are also integrated into the classroom where they are used to develop the skills that are necessary of an electrical engineer. FPGA-based SoPC development boards have been used over the past few years in their undergraduate classes to allow for a wider variety of

student projects as an alternative to more traditional off-the-shelf microcontrollers or other basic FPGA boards.[5] [6]

## 3. Methodology

Our Computer Architecture course (EE375) at University of _____ was the last course in the digital thread to integrate the DE2 board. By making this transition, we are able to use a common platform throughout the thread in addition to designing, simulating, building, and testing the MARC processor in the classroom. The design of the MARC has evolved over the last ten years. In 1999, our department began considering simulation as a substitute for hardware. The proposed solution was to use a hardware description language for simulation only due to the increasing complexity of programmable device technology.[7] At this time, the first version of the MARC was coded in VHDL. In 2003, faculty members redesigned the MARC (MARC2) which is still the current version studied in EE375.[8] The new version of the MARC had two variations: one for classroom use and one to program an FPGA. The classroom version removed most of the I/O from the top-level to emphasize the internal components. This version was used with test benches to simulate the execution of the processor. The FPGA version of the MARC was instantiated on a Spartan 2e FPGA utilizing only 25% of the programmable core and 85% of the available SRAM on a Xilinx board. It was possible to instantiate the MARC on an FPGA in 2003 but due to cumbersome CAD tools, it was too difficult and time consuming to do in the classroom. In the past five years, the technology and tools have improved to allow undergraduate students the ability to easily use these tools. A student can implement a simple AND gate on an FPGA or build industrial quality RISC-based processor cores for high-end designs using the same CAD tool suite.

## 4. Hardware and CAD Tools

The Altera DE2 board, with its large number of I/O devices and peripherals, was specifically designed as an educational tool. The board includes switches, buttons, LEDs, 7 segment displays, and an LCD display to aid students with visualizing the functionality of the system programmed on to the FPGA. This board is equipped with a Cyclone II FPGA and comes with devices such as a built-in programmer, Ethernet, RS232, video, and USB. There are four types of memory: SDRAM, SRAM, Flash and a SD memory card slot.[9] The Quartus II CAD tool allows the student to develop, analyze, compile, synthesize, and download their designs onto the DE2 board.

## 5. MARC2 Architecture

The MARC2 (MARC version 2, Figure 3) is a simple 16-bit Reduced Instruction Set Computer. It has eight registers with an Arithmetic Logic Unit (ALU) capable of executing 19 different instructions. The MARC2 is a multi-cycle machine with five stages: fetch, decode, execute, memory access and write back. The top-level contains an address bus, data bus, memory control signals, pushbuttons (one button for reset), switches, 7-segment displays, and a 50 MHz clock. The address and data bus allow the CPU to fetch instructions and to load and store data. The MARC2 is broken into two subcomponents: the datapath and the control unit. The control unit contains four registers: program counter (PC), stack pointer (SP), instruction register (IR), and

control word (CTLWD). The datapath contains a register file, ALU, and logic to change the datapath depending on the control word signal (Ctl_wd) from the control unit. The SRAM memory (512 KB) is built using a CAD tool called SOPC (System on a Programmable Chip) Builder. The memory interface is the only portion of the design created using SOPC Builder. When the design is downloaded onto the FPGA, a memory instantiation file configures the SRAM with the program code for the processor.

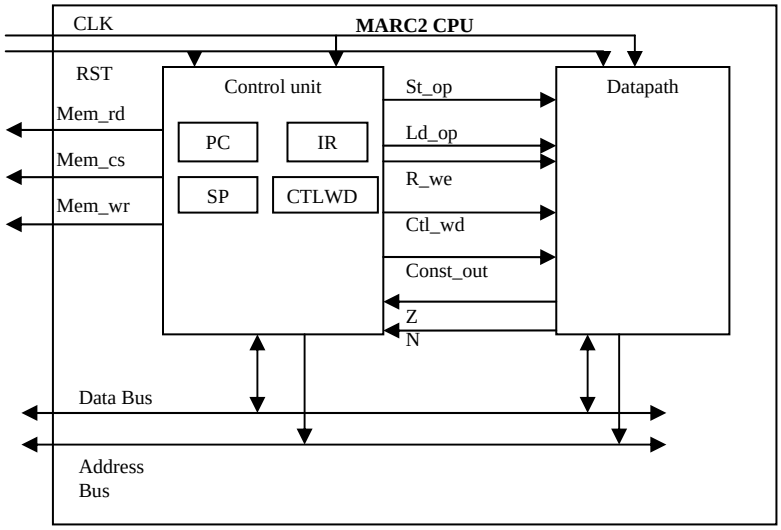


Figure 3: MARC2 Architecture Summary

The I/O devices on the DE2 board provide many forms of feedback to the student. The 16 toggle switches can be used to determine which of the following internal registers to display: eight data registers, four control unit registers, and four internal datapath registers. A student can trace and debug the expected values from their simulation to their actual implementation on the DE2 board. The output display is broken down into three parts: stage of the processor, register, and hex value. The stage of the processor is denoted on the first 2 7-segment displays: "1F" Instruction Fetch, "1D" Instruction Decode, "E" Execute, "A" Memory Access, and "B" Writeback. The second set of 7-segment displays indicates the register. The last four 7-segment displays show the hex value for the selected 16-bit register.

## 6. MARC2 Implementation

Taking the classroom version of the MARC2 and instantiating it on an FGPA required a fair amount of engineering effort from the students and faculty involved in the design. The memory interface proved to be the most difficult portion to integrate. Fortunately, only the top-level module needed the most change with the lower-level modules primarily intact. With the MARC2 implemented on an FPGA, we are able to enhance the design cycle for the students. In EE375, students design the datapath in Lab 3, and given a control unit, simulate the entire processor after Lab 4. In Lab 4, students use an assembler/emulator designed specifically to work with the MARC2.[10] These laboratory exercises ended at a functional simulation (see left side of Figure 4). By using the DE2 board, students are able to synthesize their design, target actual hardware, and run their processor on an FPGA (right side of Figure 4). Further, they are

able to debug their system during execution and can introduce improvements in the design such as pipelining or data forwarding.

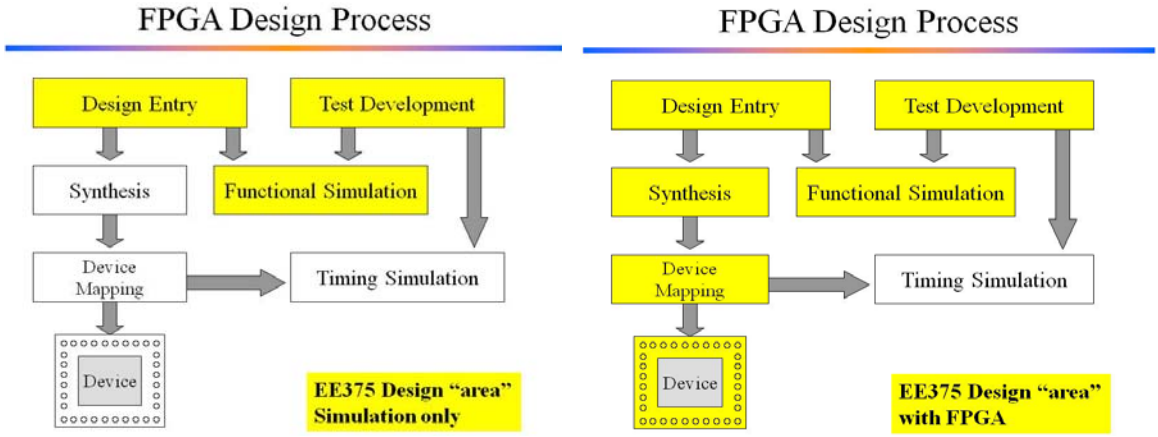


Figure 4: Design Process (Simulation only and with an FPGA)

## 7. Future Work

We are currently incorporating the FPGA-based MARC2 into a graded laboratory exercise. The existing version of the assembler must be upgraded to generate memory files compatible with the DE2 board SRAM memory interface. We also plan to utilize the LCD screen to provide additional I/O feedback to the students. The internal processor data shown on the 7-segment displays could instead be displayed on the LCD which would create a better interface for the students. There are also many architectural improvements that students could make to their own processors in follow-on courses.

## 8. Conclusions

Implementing the MARC2 on the Altera DE2 board in our Computer Architecture course provides a common hardware platform across the digital thread in our program. The initial results are encouraging where students find making a processor work is exciting and improves their motivation to do proper simulation. The MARC2 allows students to synthesize their designs and run assembly programs in hardware, improving student confidence and learning.

The views expressed are those of the authors and do not reflect the official policy or position of the University of _____, the U.S. Department of the Army, the U.S. Department of Defense or the United States Government.

**Bibliography**


1. iRobot Create and Altera DE2 Board Laboratory Exercise developed by LTC Bob McTasney, United States Military Academy, 2008.
2. Christopher Korpela and Robert McTasney, “An FPGA Multiprocessor System for Undergraduate Study”, American Society for Engineering Education Annual Conference in Austin, Texas, 14 June 2009, pending.
3. Creating Multiprocessor Nios II Systems, PDF File, Altera Corporation, http://www.altera.com/literature/tt/tt_nios2_multiprocessor_tutorial.pdf
4. ECE 5760 Advanced Microcontroller Design and System-on-chip, Professor Bruce Land, http://instruct1.cit.cornell.edu/courses/ece576
5. J.O. Hamblen, T.S. Hall, Using an FPGA Processor Core and Embedded Linux for Senior Design Projects, IEEE International Conference on Microelectronic Systems Education; pp33-34.
6. T. S. Hall and J. O. Hamblen, "System-on-a-Programmable-Chip Development Platforms in the Classroom," IEEE Transactions on Education, vol. 47, no. 4, pp. 502-507, Nov. 2004.
7. W. Kleinfelder, D. Gray, and G. Dudevoir, “A Hierarchical Approach to Digital Design Using Computer-Aided Design and Hardware Description Languages”, *29th ASEE/IEEE Frontiers in Education Conference*, San Juan, Puerto Rico, November 10-13, 1999, pp. 13c6-18-22.
8. MARC2 Design Summary developed by MAJ Paul Maxwell, United States Military Academy, 2004.
9. Altera’s DE2 Development and Education Board, Altera Corporation http://university.altera.com/materials/boards/unv-de2-board.html
10. Military Academy RISC Assembler (MARASM) created by COL Eugene Ressler, United States Military Academy, 2003