
# Design and Implementation of a Reconfigurable Computing Course for efficient Hardware/Software Co-Design in Reconfigurable Systems

DANIEL LLAMOCCA

**Electrical and Computer Engineering Department,**

**Oakland University**

*April 30th, 2016*

OAKLAND UNIVERSITY

# Outline

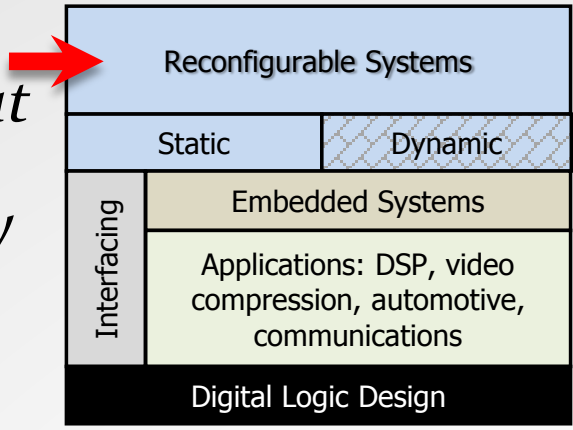


- Motivation
- Hardware/Software Equipment
- Course Structure
- Analysis
- Outcomes
- Conclusions

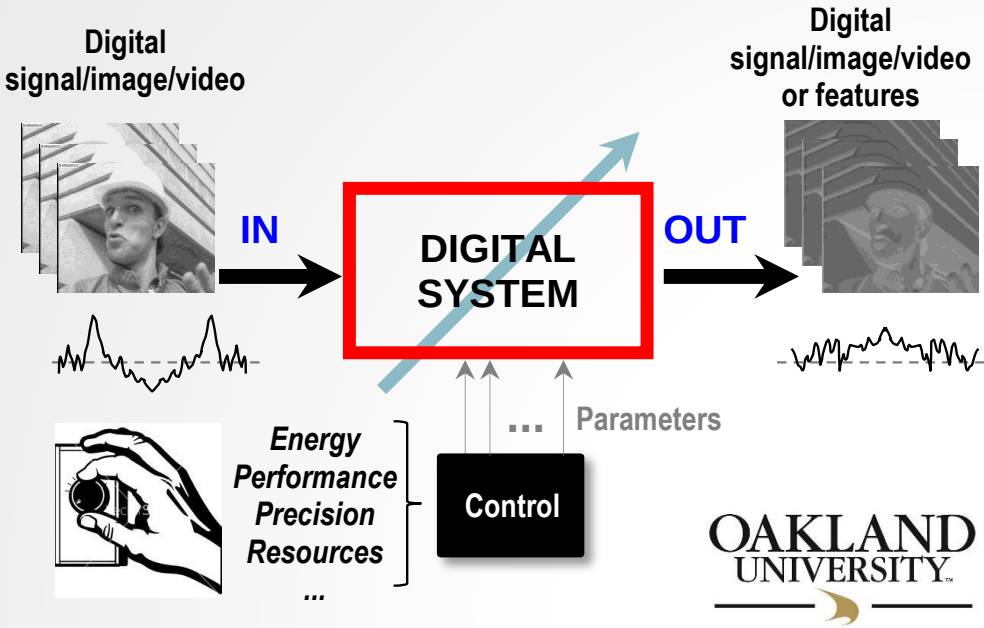
# Motivation



- *ECE students take standard classes in:*
  - *Digital logic: digital circuit fundamentals*
  - *Embedded system design: software and peripheral control.*
- *Reconfigurable Computing: It deals with the implementation of reconfigurable systems that alter (often on the fly) the functionalities and interconnection of their components to satisfy time varying requirements.*




- *Reconfigurable computing builds on embedded design and digital logic: hardware units are often connected to an embedded microprocessor via an interface.*
- *This emerging area is rapidly finding its way into many applications: smart antennas, image analysis, video compression, automotive.*



# Hardware/Software Equipment

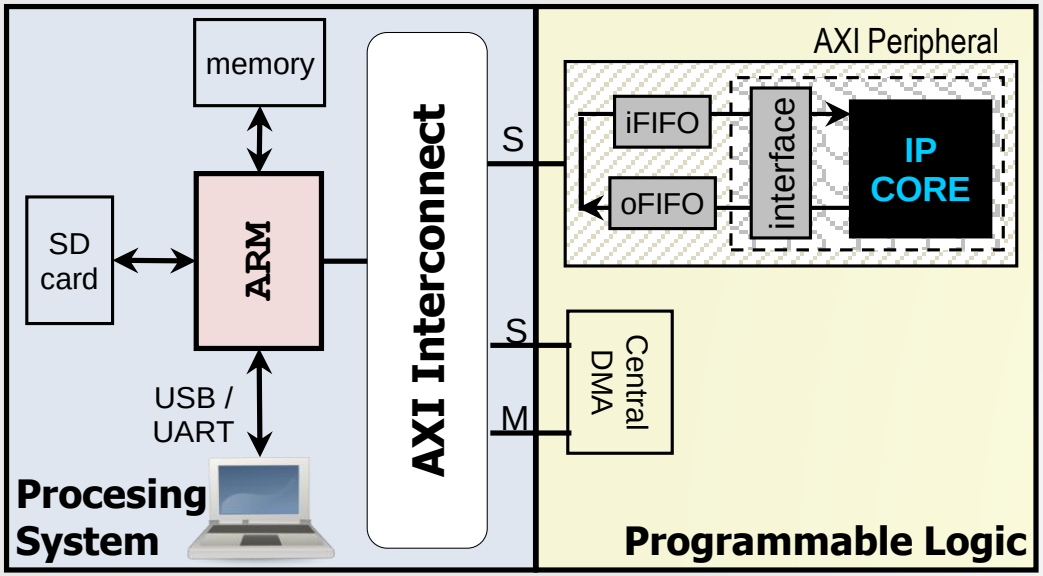


- ***Reconfigurable Computing*** *encompasses a large diversity of topics:*
  - *Hardware design: computer arithmetic, pipelining, parallelization*
  - *Embedded Systems: embedded interface design, embedded software development, hardware/software codesign.*
  - *High-level modeling: for verification (MATLAB®, Octave).*
- *Programmable System-on-Chip (SoC) devices: A great platform for teaching Reconfigurable Computing concepts. These devices include a reconfigurable fabric and a processor system. The user can design and test embedded systems containing both software and hardware units.*
  - *Many common peripherals (e.g.: USB, Ethernet) are available as libraries from the software tool.*
  - *The user can include custom-built peripherals, which involves digital hardware and embedded interfacing.*

# Hardware/Software Equipment



- *Xilinx® Zynq-7000 All-Programmable SoC: They integrate a feature-rich dual-core ARM® CortexTM-A9 processor as well as a reconfigurable fabric that allows for run-time reconfiguration.*
- *The peripherals and the processor are interconnected via the AXI (Advanced eXtensible Interface) bus. There are three types: AXI4-Lite (low-speed, simple version), memory-mapped AXI4-Full (high speed), and AXI4-Stream.*

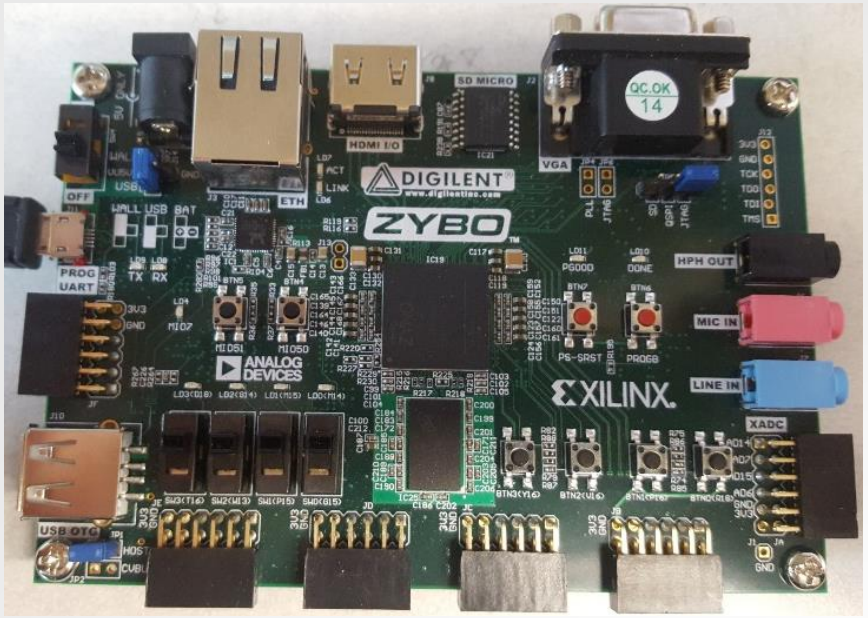



- *The figure depicts a block diagram of an embedded system with common peripherals and a custom hardware peripheral.*

# Hardware/Software Equipment



- *ZYBO Board: Low cost, entry-level embedded software and digital circuit development platform. It houses a Zynq-7000 device along with on-board memories, video and audio I/O, dual-role USB, Ethernet, and SD slot.*




- *Course: It uses both the AXI4-Lite and the AXI4-Full 32-bit Slave Interfaces:*
  - *AXI4-Lite peripheral: A simple register-based interface.*
  - *AXI4-Full peripheral: It allows for burst transfers, but needs a custom-built interface (with FIFOs) around the hardware core . This interface is provided to the students.*

# Course Structure



- ***Topics**: These were carefully selected and grouped into six units. Lecture notes were developed for each unit:*
  - *Advanced topics in Computer Arithmetic: Selection of numerical representation is vital for algorithm implementation on hardware. We analyze numerical bounds and accuracy of: fixed point, floating point, and the non-standard dual fixed-point arithmetic.*
  - *Specialized arithmetic circuits and techniques: Hardware implementation of simple and complex common arithmetic units as well as techniques for resource-efficient implementation.*
  - *Advanced coding in Hardware Description Language (HDL): This unit provides details on custom-defined types, parametric coding, and dealing with I/O files for Synthesis and Simulation.*
  - *Pipelining and unfolding: These techniques leverage the power of FPGAs by allowing a system to output data at every clock cycle.*
  - *Embedded system on a System-on-Chip (SoC): Students learn to partition into hardware and software components*
  - *Dynamic Partial Reconfiguration: This powerful technology allows hardware portions to be altered (or turned off) on-the-fly, while the rest of the system is still operating.*

# Course Structure



- ***Assignments**: The evaluations were organized as follows:*
  - *Four (4) homeworks.*
  - *Five (5) laboratories: These included tasks related to the design of embedded systems on a Programmable SoC: hardware design, embedded interface design, and embedded software development:*
  - *In-class midterm exam.*
  - *Final Project (groups of 2). Students were evaluated in their ability to successfully partition a system into hardware and software components to implement a reconfigurable system. They also submitted a final report that includes their methodology and results.*
- *The table lists the topics along with the associated assignments: homeworks (HW), laboratories (LAB)*

| Topic (# of lectures) | HW | LAB |
|---|---|---|
| Advanced topics in Computer Arithmetic (4) | 1, 2 | |
| Specialized arithmetic circuits and techniques (4) | 2, 3 | 3 |
| Advanced coding in HDL (2) | 4 | 1, 3 |
| Pipelining and Unfolding (4) | 4 | |
| Embedded System on a SoC (6) | | 2, 4, 5 |
| Dynamic Partial Reconfiguration (3) | | |

# Course Structure



- *Laboratory assignments are organized as follows:*

| Lab | Description |
|---|---|
| 1 | Xilinx® Vivado Webpack Design Flow: Synthesis, Simulation, Bitstream Generation, and Programming |
| 2 | Introduction to Embedded System Design: block-based design and embedded software development |
| 3 | Hardware implementation of iterative algorithm for computation of trigonometric functions |
| 4 | Embedded System: Design of a custom peripheral using the AXI4-Lite Interface. |
| 5 | Embedded System: Design of a custom peripheral using the high-speed AXI4-Full Interface. |

- ***Online Material**: This entire material is freely available at the Reconfigurable Computing website at Oakland University:*
  - *Lecture notes for the six units.*
  - *Four homeworks with solutions*
  - *Final project guidelines with students' final reports and presentations.*
  - *Five laboratories*
  - *Five tutorials on embedded system design with Vivado and SDK. This includes and software/hardware examples.*

# Analysis



- ***Practices**: The work in Wieman et al, "The Teaching Practices Inventory: A New Tool for Characterizing College and University Teaching in Mathematics and Science", provides a comprehensive list of practices that support student learning and teacher effectiveness. This course implemented the following:*
  - *Knowledge organization: The class includes a detailed syllabus listing student learning outcomes, grading scheme, schedule of assignments, laboratory materials, final project guidelines, and online material.*
  - *Motivation: The course featured the design of real-world applications combining both software and hardware components, discussed state-of-the-art topics, and provided research opportunities.*
  - *Practice: The class includes homework assignments, student demonstration of working systems in laboratory assignments and final project, oral presentation of the final project, and final report preparation.*
  - *Group Learning: The class relies heavily on team-based laboratory assignments and final project.*

# Analysis



- ***Impact of Learning***: *Learning outcomes include competence in topics, embedded design proficiency, and oral/ written presentation. We list them with the associated class activities:*

| Student Learning Outcomes | Activities |
|---|---|
| Design custom architectures using fixed-point, dual fixed-point and floating-point arithmetic | HW1, L1, HW2, L3 |
| Learn advanced coding and testbench techniques in Hardware Description Language | HW3 |
| Describe how to unfold a sequential algorithm to turn it into a fully pipelined architecture | HW4 |
| Design an embedded system using FPGA fabric and an embedded microprocessor | L2, L4, L5 |
| Work in a team environment to design a reconfigurable system and communicate the results in a written report and an oral presentation | Final Project |

- *Two main reasons have been identified that improve student engagement, learning outcomes, and student success:*
  - *Hardware/software co-design: Student engagement improved by integrating custom hardware and embedded software routines. Students have been learning these topics throughout college.*
  - *Opportunities for research in state-of-the-art topics: Cutting edge topics were included: non-standard computer arithmetic, dynamic partial reconfiguration, AXI interfacing.*

# Outcomes

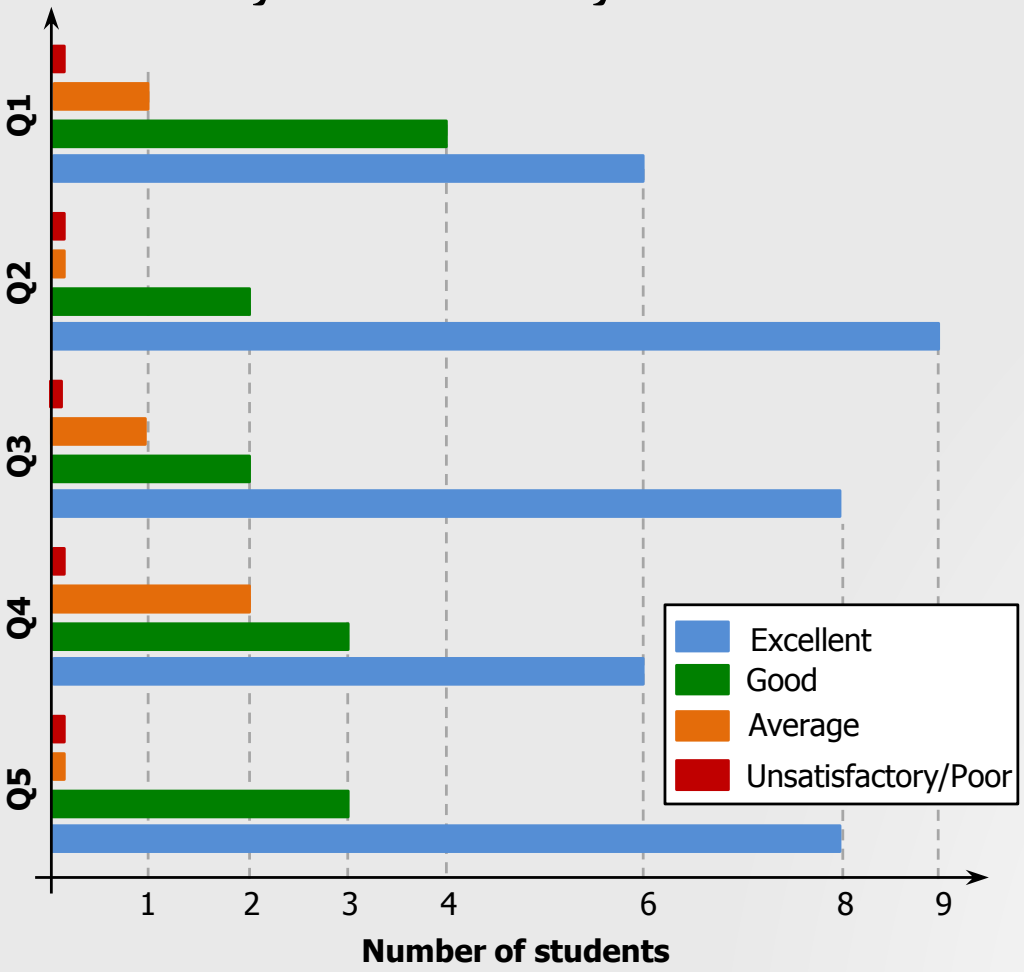


- ***Student Projects***:
  - *Hardware for powering function ($x^y$): Developed for both single and double precision. For hardware validation, a software routine controls an AXI4-Full interface and a SD card peripheral.*
  - *Development of Dual Fixed Point Arithmetic Units: Addition, subtraction, multiplication, and division were designed. Hardware validation: embedded software routine with a AXI4-Lite interface.*
  - *Hyperbolic CORDIC in Dual Fixed Point Arithmetic: Architecture for hyperbolic functions using non-standard arithmetic that provide trade-off between floating and fixed point arithmetic.*
  - *Architectures for floating point units: Addition, subtraction, and multiplication units were designed. An AXI4-Lite interface was included for hardware validation using a software routine.*
  - *Image filtering and RGB-to-grayscale conversion: These two hardware components were independently developed and tested using an AXI4-Lite interface and embedded software.*
  - *Graphics Processing units on an FPGA: Design of a small GPU architecture for matrix multiplication, division, and a line drawing algorithm. An AXI4-Lite interface configures the hardware, while a Master AXI4-Full interface sends video frames to a DDR memory for display.*

# Outcomes



- ***Student Survey:*** *11/13 students completed an end-of-semester anonymous survey. This is a selection of the questions:*




- *Q1: The instructor did a good job of making the objectives of the course clear to me.*
- *Q2: The instructor stimulated and deepened my interest in the subject.*
- *Q3: The instructor motivated me to do my best work.*
- *Q4: Value of the laboratory component of the course.*
- *Q5: Overall rating of this course as a learning experience.*

- *We show how students rated each question (Q1-Q5) and the number of students that gave a specific rating.*

# Outcomes



- ***Course Implementation Challenges:*** *To design dedicated architectures and then incorporate them into an embedded system, students need to learn: computer arithmetic, specialized arithmetic circuits, and resource-efficient and high performance techniques.*

- *Several students commented that they would have preferred to integrate the microprocessor and hardware earlier in the semester instead of focusing on theoretical topics. This can be addressed by a complete overhaul of the course structure so students can start working with embedded systems after the 3rd week of class.*

- *Most students successfully displayed their results using a serial interface (laboratory and final project). To improve student engagement, a series of tutorials must be included on how to work with other audio and visual interfaces.*

- *Dynamic Partial Reconfiguration was introduced, but not included in the assignments. The lack of a Graphical User Interface in the Vivado 2015 software hindered our efforts (the intricate procedure could only be taught in command-line mode). Extra effort must be put on the step-by-step tutorials and examples.*

# Conclusions



- *We presented a course structure, results and analysis from the implementation of a course in reconfigurable computing using hardware/software co-design for embedded systems.*

- *Results are encouraging: improved student engagement, availability of open-source material, and opportunities for research in state-of-the-art topics.*

- *Students successfully completed the assignments and final projects, and they highly rated their overall course learning experience. Based on student feedback and instructor experience, several improvements are listed that will be addressed in future implementations of this course: the freely available class material and embedded design tutorials will be updated on a regular basis.*