ENGN8537: Embedded Systems Major Project

Description

The major project for Embedded Systems will require the students to design and implement an Embedded System centred on an Altera DE2-115 Development Board. The students will critically analyse the structure of their design and justify implementation decisions regarding HDL or procedural implementation, real time constraints, physical interfacing assumptions and others as relevant to the particular project chosen.

The project is designed to allow the student to explore areas of knowledge that have interested them over the course of their degree. There is no fixed task to be accomplished as students are encouraged to formulate projects based on problems they've been exposed to in their study of other areas such as Computer Vision, Control Systems, Telecommunications, Robotics etc. Inspiration for projects may be taken from the list of Project Ideas listed below.

The project is to be undertaken by each student individually. The design must be targeted for the Altera DE2-115 Development Board with any extra hardware support to be discussed with the lecturers. The project and lab work will form 40% of the assessment for the course with the breakdown described below. Weekly lab sessions will be available to all students for development and testing of their designs, however it will be assumed that the students continue to develop the project in simulation or otherwise on their own time.

Students are free to make use of code given as part of the example package as well as any other code bases they have access to (including online) so long as the following conditions are met:

- The student must have the right to use and redistribute the code (licence compliance)
- The student must clearly attribute the source of the code, both in the code base itself and in the accompanying documentation
- The student need not completely understand how the used code works, however they must be able to describe what the code does and how it interfaces to the rest of the project and/or the environment.

Code that has not been written by the student will not be assessed, but should be used to facilitate other assessable outcomes.

Students are required to submit a formal project report of approximately 10 A4 pages including figures with architecture details, interface specifications, quantitative performance reviews and benchmarks of different implementation options for critical portions of their design. This will be reviewed by the assessors prior to the project demonstration.

The project will be assessed at a scheduled demonstration. All students will present on the same day, schedule allowing. Present at the demonstration will be the lecturers, lab demonstrators and possibly external moderators as required.

Available Hardware

In addition to the DE2-115 board, the following hardware devices are available. If you wish to do a project requiring hardware support but don't see the required peripherals on the list, please discuss with the lecturer as soon as possible.

- Ultrasonic range finders XL-MaxSonar-EZ/AE MB1200
- Microphones, Speakers
- Infrared remote controls
- PAL Camera device (limited numbers)
- WifiBot mobile robotic platforms (limited numbers)
- TurtleBot mobile robotic platforms (limited numbers)

Note that the audio peripherals on the board may be used as general purpose analogue I/O, within the voltage and bandwidth limitations specified by the codec device. In addition, the VGA DACs may be used as general purpose high speed, high resolution Digital to Analogue converters.

Deliverables

Item	Date
Report	11th October
Final Demonstration	18th October

Report

Students are required to submit a formal report. This should cover at least the following:

1. Intended function and operational instructions
2. Architecture and design
3. Functionality and functional limitations
4. Analysis of implementation options with respect to stated metrics (performance, FPGA utilisation, accuracy of results etc.).
5. Conclusion of design success including suggestions for future work or improvement

It must include a one-page Executive Summary and full references. Full marking criteria are found in the rubric below.

Project Ideas

Students are encouraged to design their own projects; the ideas below are mostly for inspiration.

RSA/DES/AES encrypter/decrypter

Embedded Systems often require encrypted communication channels. This project would look at software vs. hardware implementations of common encryptions standards, investigating the advantages of hardware parallelism. Variations and extensions on this project could include brute-force, dictionary or rainbow-table based attacks on encrypted data with unknown key.

Protocol Analyser

A good way to understand a communications interface is to 'snoop' on existing traffic, note and record patterns. Rather than requiring external busses, the analyser may be set up to monitor traffic on other busses on the board, for example I2C (to the audio or video codec) or USB device/host interaction. A basic implementation may be able to produce reports via the serial interface where a more advanced one would display the data trace on the VGA monitor, highlighting framing or error conditions.

Complex impedance (Bode) plotter

In many embedded systems, external devices are measured and controlled by their interaction with known signals. The audio interface may be used to generate and analyse signals for phase shift and attenuation in order to determine the complex impedance of the device connected. A Bode plot could be drawn to the VGA monitor and/or the values of simple RLC circuits identified. Extensions to this might include a water level sensor based on change of capacitance between two plates, metal detectors based on a change in inductance etc. An advanced implementation may be able to detect proximity of a hand or other water body based on change in capacitance (the way that many touch screens now work).

Doppler velocity meter

A key item in robotics is object detection and avoidance. Using the microphone and speakers, a system may be created to determine the distance to, and velocity of, a remote object by round-trip delay and Doppler shift. This information may be combined with range data from Sonar or IR range finders by means of a Kalman Filter, Monte-Carlo algorithm or otherwise.

Audio effects board

Using the audio interface and implementing a selection of filter algorithms, the DE2-115 may be used as an audio effects board. The parameters of such filters and effects engines should be able to be selected by the user at run time and the current parameter set could be shown on the VGA display. Visualisation of the audio streams and effects via time domain, frequency domain and/or Bode plots could form part of an advanced implementation.

Analysis of different implementation options (e.g. combinatorial FFT vs. sequential/pipelined) could be investigated for performance and complexity.

Computer Vision

The field of CV has many algorithms that may be accelerated using FPGA technology. A project may investigate the implementation of edge detection, feature detection, object recognition or many other such algorithms in the highly parallel FPGA environment.

Audio Control

An accelerometer may be attached to a speaker cone and used to determine the movement of that cone relative to the input signal. A control algorithm could be developed to improve the frequency response of the speaker based on this closed-loop information.

Linux on the Nios II processor

This is an advanced project and should only be undertaken by students with experience in Linux programming.

Many Embedded Systems now use Linux on their core processors. The Nios II processor for the Altera DE2-115 board is capable of running Linux [1] and can be used to demonstrate the advantages and disadvantages of this approach to embedded design. Simply getting Linux to run on the board is not sufficient for the project; the student must critically evaluate response time and throughput metrics for Linux implementations of some task as compared to pure hardware and/or no-OS implementations.

[1] http://www.alterawiki.com/wiki/Linux

Project Notes

Quantitative Data

Students should, where possible, use proper statistical methods to justify claims such as "Implementation B is faster than implementation A". At the most basic level, the experiment should be repeated several times with averages and variances (standard deviations) noted. If a difference between two sets of numbers is asserted, the means should be more than one standard deviation away from each other. In an HD-level justification of the difference of two implementations, a "Student's Two-Sample t-Test" could be used. If the student is not familiar with this method from ENGN2226 or similar, a summary can be found at [1] while the MATLAB function `ttest2` can perform the calculation. Your choice of alpha value (confidence interval) should be justified.

[1] http://www.socialresearchmethods.net/kb/stat_t.php

Marking

Marking will be split in to three parts. The project itself and the report form 30% and 5% respectively of the student's overall grade for the semester. They will be marked according to the rubric found below. The student's attendance at the Guided Project Sessions, run during normal lab slots in weeks 7-10, will be recorded and each instance of attendance worth 1.25% of the student's overall grade for the semester.

Lab openings outside the Guided Project Sessions are not counted towards the attendance mark.

Learning Outcomes

The student is able to:

1. Analyse the strengths and limitations of Microcontrollers and FPGAs for Embedded Systems
2. Construct real time constraints from problem specifications
3. Effectively integrate sensor, actuator and interface devices with Microcontrollers and FPGAs
4. Optimise the interaction of Embedded hardware designs with their software

Weighting

The lab sessions and project and will form 40% of the student's overall grade for the course. This will be split and assessed as follows:

Project Performance & Presentation	**Report**	**Lab Attendance**
30%	5%	5%

Rubric

Category	Pass	Credit	Distinction	High Distinction
Design and Architecture	• Architecture of the project is described	• Architecture of the project is described • Some areas are noted as being critical for performance • Most decisions made regarding the specific implementation of architectural elements and/or the design as a whole are justified	• Architecture of the project is described • Some areas are noted as being critical for performance and some attempt is demonstrated at optimising these sites • All decisions made regarding the specific implementation of architectural elements and/or the design as a whole are justified	• Architecture of the project is described • All reasonable areas are noted as being critical for performance and many of them have demonstrably and effectively been optimised • Different implementation techniques are demonstrated depending on the design requirements • All decisions made regarding the specific implementation of architectural elements and/or the design as a whole are justified
Functionality	• basic functionality works reliably • most functional limitations, limits on the operating environment and conditions for correct operation are noted in the accompanying documentation	• basic functionality works reliably • some advanced functionality exists and works subject to particular operating conditions • most functional limitations, limits on the operating environment and conditions for correct operation are noted in the accompanying documentation	• Advanced functionality works reliably • All functional limitations, limits on operating environment and conditions for correct operation are noted in the accompanying documentation • The stated operating environment covers most conditions that might reasonably be expected to be encountered during operation • Some consideration has been given to system reliability and safety where appropriate.	• Advanced functionality works reliably • All functional limitations, limits on operating environment and conditions for correct operation are noted in the accompanying documentation • The stated operating environment covers all conditions that might reasonably be expected to be encountered during operation • Conditions outside the operating environment lead to graceful (safe) degradation of functionality, within the limitations of the hardware • Thorough consideration has been given to system reliability and safety where appropriate.
Interface Specification	• Interfaces between modules within the project are identified • Interfaces with the environment are identified	• Interfaces between modules within the project are identified • Interfaces with the environment are identified • Most interfaces between modules within the project are specified and documented with respect to data standards, required timings and other interface semantics	• Interfaces between modules within the project are identified, specified and documented with respect to data standards, required timings and other interface semantics • Interfaces with the environment are identified • Most interfaces with the environment, including time, are specified and documented	• Interfaces between modules within the project are identified, specified and documented with respect to data standards, required timings and other interface semantics • Interfaces with the environment, including time, are identified, specified and documented

Category	Pass	Credit	Distinction	High Distinction
Code Quality	• Most code is written in a consistent and neat style • Some in-line commenting exists • All code not written by the student is correctly attributed	• All code is written in a consistent and neat style • Most pieces of code whose functionality is obfuscated, unique or special have in-line comments • All code not written by the student is correctly attributed	• All code is written in a consistent and neat style • All code whose functionality is obfuscated, unique or special have in-line comments • Some modules have interface or usage documentation embedded within them, or in an accompanying document (which may not be the deliverable document) • All code not written by the student is correctly attributed	• All code is written in a consistent and neat style • All code whose functionality is obfuscated, unique or special have in-line comments • Most modules have interface or usage documentation embedded within them, or in an accompanying document (which may not be the deliverable document) • All code not written by the student is correctly attributed
Presentation	• Student clearly presents the key points of their project	• Student clearly presents the key points of their project • Student clearly presents and explains limitations of their design and implementation • Student gives satisfactory answers to some questions from the examiners	• Student clearly presents the key points of their project • Student clearly presents and explains limitations of their design and implementation • Student gives satisfactory answers to most questions from the examiners	• Student clearly presents the key points of their project • Student clearly presents and explains limitations of their design and implementation • Student presents qualitative justification of design decisions • Student gives satisfactory answers to most questions from the examiners
Report	• Report shows good spelling, grammar and thoughtful layout • Report states key design decisions and hurdles • Report presents a reflective conclusion	• Report shows good spelling, grammar and thoughtful layout • Report states key design decisions and hurdles • Report gives qualitative justification of some key design decisions • Report presents a reflective conclusion including concepts for future work	• Report shows good spelling, grammar and thoughtful layout • Report states key design decisions and hurdles • Report gives qualitative and quantitative justification of some key design decisions • Report presents a reflective conclusion including critical statement of design strengths, limitations and concepts for future work	• Report shows good spelling, grammar and thoughtful layout • Report states key design decisions and hurdles • Report gives qualitative and/or quantitative justification of all key design decisions • Report presents a critical analysis of how the implementation has encountered and overcome the core challenges of Embedded Systems. • Report presents a reflective conclusion including critical statement of design strengths, limitations and concepts for future work