EEE4120F High Performance Digital Embedded Systems

FPGA Project

SSDC: Sea Sounds Data Caster

Hiding some digital transmissions in calming noises

15 March 2019

Projects to be done as a team of 2 or 3 students

Table of Contents

This project concerns the design, and experimental implementation and testing of a special purpose high performance embedded computing system. We've tried to come up with a project that will be both interesting, to get you to think and learn about some physical phenomena, but at the same time to calm your nerves a bit… as we know how stressful final year of undergraduate study can be. So, without further ado, welcome to the "Sea Sounds Data Caster (SSDC)" system that will produce relaxing sea sounds (which are a kind of modulated white noise) and can include some 'subliminal' messages (at least detectable by machines) of positive quotes or entertaining says – or just interesting bit sequences. Part of the 'fun' is to try and come up with messages to encode into your relaxing sounds that are suitably pleasing… or curious.

The following submissions are required for this project:

1) M1 (Milestone 1) Essay (10%), max 5 pages: essay on sea sounds, diagrams should be included. This milestone is mainly to encourage you to read up on these natural phenomena and properties of natural waves, which links to the natural science element of this ELO to which this project relates. It is essential that your essay be organized and written in view of the clear objective of developing the SSDC as a high performance embedded computing system, though largely simplified.
2) M2 design review and Competency Test (ELO assessment) (10%): figures, screenshots and a written or recorded oral narration through which you need to present a refined design of the system before completing the implementation. The system design should demonstrate your knowledge and appropriate application to system analysis and design, of suitable mathematical and statistical concepts and techniques such as the Gaussian distribution (noise model) and Nyquist theorem. We are not having formal in-person design reviews for the course this year as the student numbers are too high, which will make the scheduling logistics difficult – the online submission approach is a compromise whereby you can still get thorough feedback but have the advantage of preparing the submission whenever and wherever you want to. Competency test to be help after this due date.
3) M3 Demonstration (20%): this will be a +-10 minute presentation to the lecturer, tutor ~~or approved volunteer assessor (we planning to ask postgraduates working in this field to assist with demo assessments).~~ The demos provide a final intervention before the final report is due so that you can refine your report before submission. You are welcome to bring a copy of your partially completed report to show to the assessor for some quick feedback.
4) M4 Final Report (60%): this will be a full report of the project including introduction, background, methodology, SSDC-R concept design, SSDC-R design, SSDC-T prototype design, implementation, results and conclusions. The completed report to be in PDF format and should be uploaded using the Vula assignment for this project which will also indicate the due date.

1 OPERATION OF SSDC

Putting it briefly, the SSDC needs to generate modulated white noise that sounds like sea sounds. The volume and pitch of these natural wave sounds can vary – but they should not have excessive variation – as sounds with excessive variation are not predictable and relaxing for human listeners. Part of your task is to read up about sea waves (you can also investigate wind, as these are similar) to gain a deeper insight into their characteristics and the variations of these sounds. In order to demonstrate your understanding, you are tasked with writing a short essay (M1) on your findings.

1.1 System Overview

You need to develop both a transmitter and receiver for the SSDC system, a high-level illustration of the system is provided in Figure 1. The transmitter, I strongly advise that you develop it first as a PC application using OCTAVE, see Section 1.2 for SSDC-T description. Once a experimental OCTAVE SSDC-T has been constructed (golden measure) then the transmitter should be developed as an FPGA prototype, which can receive and transmit messages to be modulated from a host PC.

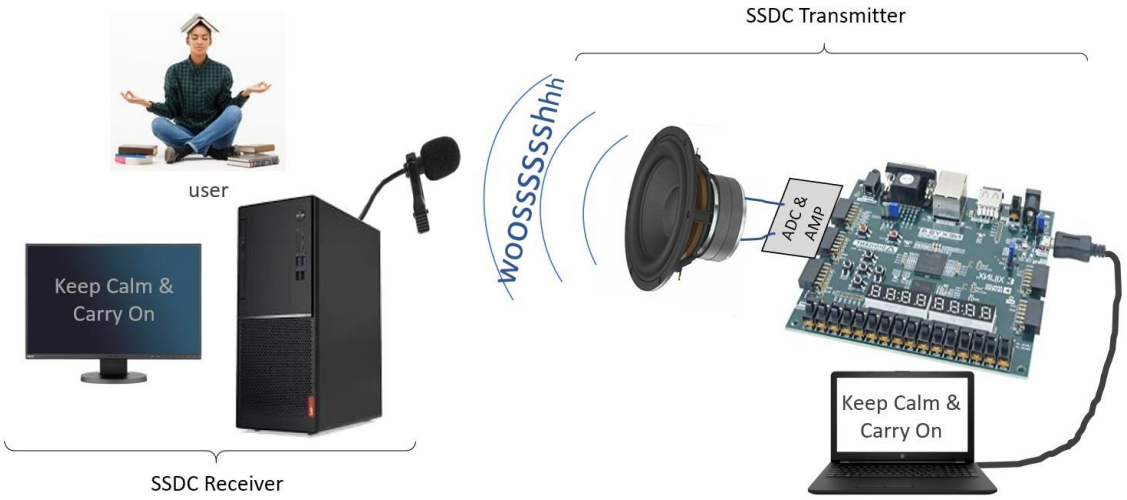


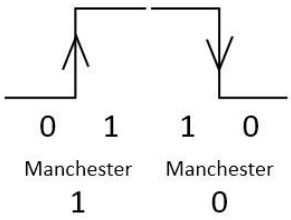
Figure 1: SSDC System Overview

The receiver, SSDC-R, is to be *designed* more in accordance to a HPEC system, for which you need to provide a digital system architectural design (on paper)… but the implementation of the receiver can be entirely PC-based, it can be written in OCTAVE / C / C++ / Python; you could use an alternative language/tool, but if you choose to do so then please seek permission to do so from the lecture as we will need to have a suitable prepared examiner to assess your work. See Section 1.3 for more detail on the SSDC-R.

1.2 SSDC Transmitter

Sample files will be given to you together with an example encoding technique, with a sample solution for this provided as additional sample files. The complete code for doing this encoding with not be provided, as you need to develop your own design and implementation for that aspect.

For your SSDC transmitter you can decide on your own modulation technique, for example you could use FM, AM or a combination of these techniques or something else. For the design review (M2) you need to propose the technique you will use. A suggestion is to have two modulation plans: a highly robust Plan A that might be a bit slow and inefficient, and a quicker but perhaps less reliable Plan B. Your Plan A could, for instance, use AM and Manchester encoding. As you hopefully remember, Manchester encoding involves sending two bits for each message bit so that the message bit is encoded as either an up-going or down-going edge, see illustration on right.



You should (ideally) include samples of the modulated files you would have experimented with in OCTAVE (or MATLAB). When submitting your demo code, you can include these samples as audio files (e.g. WAV or MP3 files) that the assessor of your demo can listen to.

For this part of the project you need to develop an FPGA-based implementation of your modulation scheme. You will need to construct HDL code to operate on the FPGA, for example implementing a timing core to control the system's timing, a message FIFO, a modulation core, a function generator code (that may need to have a sin function encoded in BRAM used to generate smooth frequencies).[1] A near final version of this implementation is to presented in the demo, together with an operational SSDC-R receiver, working in OCTAVE or alternate language, to demonstrate messages can be send.

[1] *Due to time constraints, you do not need to implement an FPGA-based transmitter. However, you need to provide system level design and explanation of what you would need to do if you were to implement an FPGA-based transmitter.*

The design, implementation and testing details need to be written up in the final report, together with clear explanations of your chosen modulation scheme.

You can use an off-the-shelf speaker to transmit your sounds, but to avoid disturbing other lab users – or rather avoid causing them to fall asleep – you could just connect a wire between your transmitter prototype and PC while testing. Though at some stage you *do* need to use actual sound to generate the wave sounds to confirm transmission and reception of the modulated messages over sound waves.

1.3 SSDC Receiver

The receiver side is likely going to be more complicated in its functioning, particularly if considered as being done as an FPGA implementation. For this reason, the SSDC Receiver only needs to be implemented as a PC solution, implemented using OCTAVE or approved alternate language. In your design review and for your final report you are still required to explain your receiver design and to recommend a (high level) design approach for how it could be implemented as a FPGA solution, or a heterogeneous computing solution that integrates an FPGA digital accelerator and a PC user interface. But note: you are not required to develop an FPGA-based solution for the receiver (you only need do an FPGA-based solution for the transmitter). However, if you would like to take on the challenge of implementing the SSDC-R as an FPGA or partly FPGA solution then you are most welcome to attempt this – but note that while you may receive high marks in consideration for such an effort there are no bonus marks allocated for making the attempt to create a FPGA-based receiver.

The first take on your receiver can be considered a conceptual design project – assume that the design space is more flexible than it really is: consider that you are going to develop the computer board which will have an FPGA (indeed you could even hypothetically consider the FPGA as being replace eventually by a ASIC chip of your own design). Let's refer to this as your "SSDC Receiver Concept" or just your design concept – which is explained in 1.3.1. In Section 1.3.2 it is explained what you need to implement for the receiver to show that your transmitter is working, which you will use when experimenting with and demonstrating your transmitter prototype.

1.3.1 SSDC Receiver Concept

For this part you need to propose design solutions for your receiver system. For instance answering design questions such as: what mic you would use, what amp (built your own, or use a SOC, or suggest a COTS solution?), the ADC, the FPGA, and the power source and other components you might need. You would probably want to have some memory (say 128MB) into which you can store data (although you might not really need this since the modulations are likely going to be so slow). Draw up this component-based design, it can be as a block diagram and a parts list, you are not required to provide a detailed schematic design (you are not going to implement this concept). Then explore the digital logic design (also you can do this as fairly high level) that will run on the FPGA and which will analyze the recorded sea sounds and output the demodulated messages to e.g. a USB connection (or just a simple UART). The design drawings and explanation for these need to be provided in the Receiver Design Concept section of your final report.

1.3.2 SSDC Receiver Prototype

As mentioned earlier, the receiver prototype only needs to be implemented as a PC application. But you need to provide the software design for this, including a high-level design of the application, flow-charts and explanations of important aspects of your code with the inclusion code snippets where relevant to assist the explanations. These details need to be provided in the "SSDC-R design" section of the report. Make sure to include the implementation of your SSDC receiver program in your final submission so that your solution can be tested by the assessor if needed.

2 TODO: The Report

The main submission for this project is a **report**. This counts for 60% of the marks for this assignment. Complete the report according to the report structure described below. Estimations for number of pages for parts are given in italics in parentheses – consider these as lower bound estimations; it is fine to use more space if needed. The page limit of 40 pages including diagrams (but I don't think anyone will produce that much, more likely 10-20 pages is expected).

2.1 Report Structure

The first two pages (page i and page ii) must have no page number. The third page and remaining pages (i.e. the main body of the report) must be numbered from 1. With the exception of the Abstract and Table of Contents, all headings are to be numbered up to level 3 (this document illustrates this numbering style).

- **Cover page** (page i – this page should not show a page number)
  - Your name and student number, title of the project, course code, date.
  - A UCT badge (not the old version!) must be on the cover. You may add an appropriate image if you want to make it look more exciting.
- **Preamble** (Page ii – this page should not show a page number)
  - Abstract (max. 200 words, no images)
  - Table of contents (from Introduction onwards, up to maximum of three heading levels).
- **Introduction** *(1-2 pages)*
  - Provide an introduction to the topic and the report. You can include some highlights of your milestone 1 essay. Describe what is going to be covered and provide a brief summary of major design decisions that you made.
  - Include some reflection on the development process (this will help substantiate ELOs), you could discuss how you might be consider the spiral model e.g. mention risks anticipated.
- **Receiver Concept Design** (*4 pages*)
  - Refined block diagram *(1 page; if needed you can use A3)*.
  - Draft Schematic for the components of the system (if you choose to do a core, this can be limited to just a partial schematic). You can discuss with the lecturer / TA what you propose to design to confirm that it is enough work.
  - Description of major design decisions (e.g. ways in which components are connected, if a particular bus protocol is needed, etc.)
  - Remember the intention of the concept design is *not* to take things to the implementation level; rather the discussion should focus on the design level and reasons for important choices made.
- **Receiver Design** (2 pages)
  - Software design documentation of your actual (executable) SSDR-R receiver program, i.e. the program you use for confirming that your ~~FPGA-based~~ transmitter is working.
- **Transmitter Design** (4 page)
  - This can be done in two pieces: the first piece can mainly be a summary of the design review, explaining your modulation scheme and how OCTAVE, or whatever PC-based program you use, is used to generate test signals. The second piece, a more substantial one, should detail the actual design for your FPGA-based SSDC transmitter solution that you would have implemented using the Nexys4 or other FPGA board.
  - Discuss the main processing blocks that you may built/incorporated, how data will be sent from the PC to indicate the message to send (if needed your could use the LEDs or 7-segment display to show what is being sent, or if you can't get comms from the PC working your could

use the pushbuttons or slide switches as a backup for specifying the message that needs to be modulated and transmitted).
    - You should design both a Golden Measure (that runs just on the PC, e.g. in OCTAVE) and the main implementation (system under test) that could either run entirely on the Nexys board or it may be implemented as a heterogeneous system that comprises both an FPGA solution and a PC-based component (e.g. to send over messages and to capture performance results).
- **Implementation** *(1-2 pages)*
    - Present some code snippets, particularly signal processing solution and FPGA code snippets that you used in the project.
- **Results** *(2-3 pages)*
    - Reflections on the design. Discussion of benefits & drawbacks. Show performance results, e.g. BER (Bit Error Rate) at different distances between transmitter and receiver, performance (speed of deciding messages), etc.
    - Make use of appropriate mathematical and statistical concepts and techniques to analyse and evaluate your system.
- **Discussion and Conclusions** *(1-2 pages)*
    - Reflections on the design. Discussion of benefits & drawbacks.
- **References**
    - If you build on ideas from the literature or more generally work and ideas obtained from others, please cite these in your report and add an entry to the reference. You can use numbered references (e.g. IEEE or Vancouver Style[2]) or author-date (e.g. Harvard[3]). Only use footnotes for adding additional information, not for references.

[2] Details of the Vancouver style can be found at: http://www.lib.monash.edu.au/tutorials/citing/vancouver.html
[3] A guide to Harvard is at: http://libweb.anglia.ac.uk/referencing/files/Harvard_referencing_2011_quick.pdf

2.2 Marking Schema for Report

The report will be marked out of 200. Higher marks will be awarded to logical and elegant solutions presented in a professional style.

Criteria	Marks
Cover Page	5
Preamble Abstract. Table of contents.	5
Introduction (Sect. 1) Outlines the system and should also include a brief summary of major design decisions. Reflect on design process / aspects of spiral model. Outline how the transmissions are produced and (de)modulated.	25
Receiver Concept (Sect. 3) Explain the receiver concept, largely a pen on paper exercise, this potentially ASIC concept is not going to be implemented. Include use of images.	25
Receiver Design (Sect. 2) Outline the software design of your receiver, make sure it is clear how the signals will be received and demodulated.	40
Transmitter Prototype Design (Sect. 4) This relates to your design documentation for the transmitter, the implementation marks are largely part of the demo. Marks here awarded based on clarity, working, structure of descriptions. Higher marks will be given for managing to make any headway on the HDL implementation for this.	30
Implementation (Sect. 5) The demo assesses the implementation mainly, but these few marks are to award any refinements or polishing that was done after the demo. As per Sect 4 higher marks will be awarded for any attempt at an HDL implementation.	10
Results (Sect. 6) The demo also assesses results	10
Discussion & Conclusion (Sect. 7)	10
Creativity / innovative solutions. This criterion is used to award good design ideas. Creativity marks will be awarded for elements of creativity in either your design or implementation.	20
Style and professionalism. Including page numbers, headings, language, references.	20
Total:	200

2.3 Schedule

A summary of the schedules for the different project milestones is given below.

Milestone	Due
M0: Project briefing	2 April
M1: Essay	23 April
M2: Design Review *ELO competency Test date to be announced*	3 May
M3: Demonstration	10 May
M4: Final Report	15 May

Due Date: 15 May. (Late penalty: 10% per day late, excluding Sundays)

Submit your assignment online using the appropriate VULA assignment.