

ECE 383 – Embedded Systems II

Final Project Worksheet

1 Proposal

1.1 Objective Statement

The objective statement defines the purpose of the system you are designing and the need the project supposes to fulfill.

1.2 Requirements

Define the requirements that you will need to meet in order to achieve minimum functionality. Likewise define B and A-level functionality. Minimum functionality should be a barebones, proof-of-concept level implementation. B and A-level functionality should add features to build on the minimum functionality.

In addition, you must incorporate an external device to achieve A-level functionality. This could be the NES controller/ARCADE cabinet or another device of your choosing but the device must be more than a simple switch or button. Other options include other game peripherals (e.g. joysticks, other controllers, racing wheels), IR remotes, MIDI keyboards, sensors (e.g. temperature, light, sound, distance, cameras).

1.3 Level-0 Description & Top Level Design

A level-0 function table defines the overall input, output, and behavior of the system that you are building. You need to carefully consider what interactions a user will have with your system. This will give you clear goals for your project.

- **Graphical** - Draw your system with a box for each major subsystem (For example, FPGA, Input Sensors, Output Displays) and with the top level signal inputs and outputs between the subsystems. Have a specific line for each input/output signal, defining what the signal is (i.e., temperature, velocity, etc), what the specific interface link is (i.e., USB, Bluetooth, etc), and what port it connects to on each device (i.e., UART port, GPIO pin, etc). For the FGPA, draw internal blocks to delineate if the system is just Custom Hardware (FSM and Datapath), just Microblaze, or a combination of Microblaze and custom hardware.

As you think about what you are proposing for your requirements and this design, you might want to consider some of these “hint” files on the class website (under Resources)

Final Project Help

- [IR controller hints](#)
- [NES controller hints](#)
- [PS2 Mouse interface hints](#)

- [Example Grid Memory\(zip\)](#)
- [Example Graphics Memory, 2bits per pixel\(zip\)](#)
- [Reading UART keyboard without being locked out](#)
- [Analog to Digital Converter hints](#)
- [Random Number Hints](#)
- [How to 3D print a case for your FPGA board](#)

You might also want to look over the list of past projects (some with videos)

[Final Project Ideas](#)

2 Plan

The project plan defines how you are going to go about implementing the design set forth in your proposal. The plan should then go on to include the following sections.

2.1 Proposal

Include the body of the updated proposal document - include any edits made to the original proposal by the instructor.

2.2 Detailed Architecture and Sub-System Design

You need to provide the detailed design of your system. A detailed design should be split into level-1 subsystems, such as datapath and control.

2.2.1 Level-1 Design

A level-1 design breaks the level-0 design into further subsystems and modules; each module generally corresponds to some physical chip, sensor, or actuator. A system boundary should be drawn showing where the system interacts with the user (signals which cross this boundary are the I/O from the level-0 function table). Example subsystems within your FPGA include Datapath, Control (FSM), Microblaze, and Key I/O ports (like UART, Bluetooth, etc). Describe the internal organization of the chip using a datapath and control similar to those presented in Lab 1, 2, and 4. Your datapath should be composed of several basic building block (BBBs), like in Lab 1, 2, and 4. Make sure that the top level-0 signals match those in the Level-1 diagram.

The Level-1 Design should include a block diagram, showing all I/O signals and interfaces between the subsystems. Subsystems requiring software should have either a flowchart or FSM design.

2.3 Calculations/Analysis/Drawings

You may need to perform some calculations, equations, or analysis in order for your project to work or to derived the key specifications for your system. Any unusual data structures which have bit fields which meet some requirements should be explained in this section, e.g. fixed point. Any math functions implemented in look-up-tables should be defined.

If your system is creating images on the HDMI monitor (like scopeface), you should include a drawing of the example scopeface display with the objects and game area drawn to scale (row/col pixel scale)

2.4 Bill of Materials

~~Include a list of the parts that you will need to complete the project; include parts that you may already own. For each component, please indicate if we have it and if not, will you be purchasing it, or will USAFA need to purchase it. If you need an item purchased, please provide the exact part number, cost, and URL where it can be ordered.~~

2.5 Milestone I

A milestone represents an intermediate level of technical accomplishments required in the overall system. Your first milestone should focus on getting every unit defined in your level-1 design operational and communicating with your reprogrammable chip. The milestones will take on the form of tests which will be run to verify that each milestone was met. For example, if you are interfacing a thermometer to your chip then a milestone would be, "move thermometer from hot water to cold water and observe temperature change." Make sure to document these tests with pictures or measurements, so that they can be included in the final project write-up.

2.6 Milestone II

The second milestone will represent a more advanced level of system functionality. At this point you should be examining the interactions between combinations of level 1 components. Again, the milestones should take the form of tests which verify that modules are working correctly. These tests should be built around the overall stimulus response of your final project. A second milestone based on the previous example might be, "move thermometer into hot water and wait for warning LED to come on."

2.7 Updated Functionality and Requirements

Define the detailed requirements that you will need to meet in order to achieve minimum functionality. Include any edits to the requirement made to the original proposal by the instructor.

3 Milestone I

You need to document that you have or have not met your deliverable obligations for the first milestone.

4 Milestone II

You need to document that you have or have not met your deliverable obligations for the second milestone.

5 Final Demonstration and Test Results

This section should document the test and demonstration results the overall system was able to achieve.

6 Poster Presentation

This year, in lieu of a presentation, we will have a poster/demo session. You will need to have a single slide overview of your project and have a working demo ready to show to attendees.

7 Write-up

The final write-up should have sections 2 to 6 above. If you have modified your milestone goals, functionality goals, Level-0 or Level-1 Designs since your proposal/plan, then these should be updated in your final report. Don't forget section 5.

The write-up should be posted in GitHub as a README.

Also include this appendix:

Appendix A: Running the Project

Describe the procedure for a faculty member or student to duplicate your demo. I would like to have your project as a resource for students in later semesters so its important describe for them all the details necessary to make your project work. Include any special assembler or compilers that you used.

8 Git Repo (GitHub)

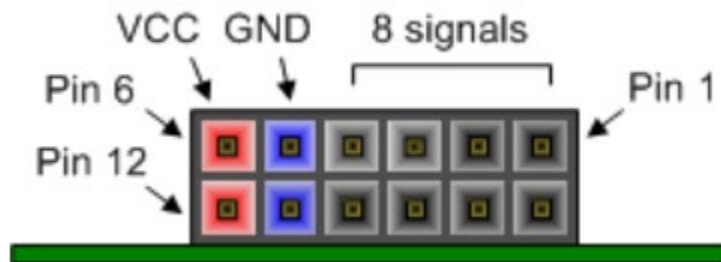
Ensure your final project code along with your poster slide, a clean screen shot or photo (ask about HDMI capture), and a final bitstream (packaged with an ELF if required) are in your repo.

9 Notes on the ARCADE Cabinet

If you choose to make your design work on the Applications of Reconfigurable Computing And Digital Electronics (A.R.C.A.D.E.) machine you will need to adhere to the following standards.

- The ARCADE cabinet emulates an NES controller. There are two controllers available and they are connected to JA1-3 and JA7-9.

JA1: P1 Latch	JA7: P2 Latch
JA2: P1 Pulse	JA8: P2 Pulse
JA3: P1 Data	JA9: P2 Data



- The cabinet will also have RGB LEDs around the monitor or marquee. These can be controlled via a one-wire protocol on JA4.
- A raspberry pi inside the cabinet is responsible for loading bitstreams. You should make your design run without requiring any user input aside from the NES controller buttons. For example, you should not rely on a user typing commands via a UART Terminal.
- You can run a MicroBlaze design on the cabinet if you package your executable file (ELF) within your hardware bitstream. This means you will create your hardware design, export the hardware to Vitis, complete your software design, compile the software to ELF, then go back to Vivado and insert the ELF before regenerating the bitstream.
- We will use an 800x600 image as a preview of your game/application within the menu, so please create a nice image for this and include it in your GitHub repo.