**EECS 392: FPGA Systems Design Projects – Midterm Report**

**Outline**

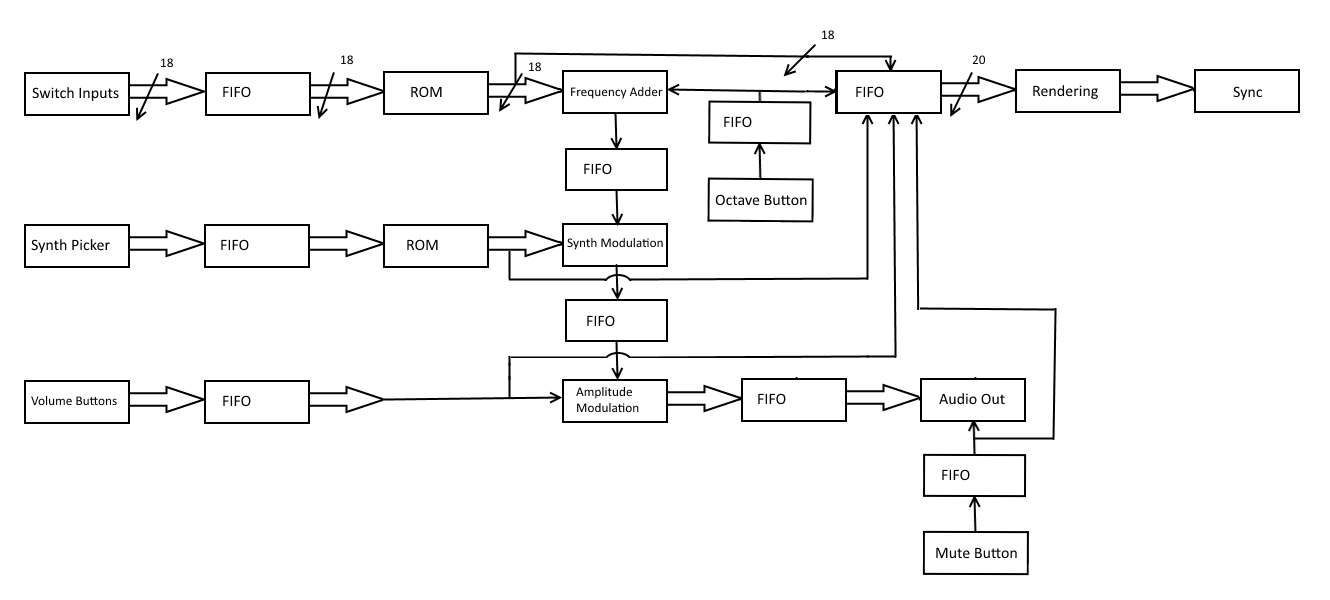
**Introduction:** In the introduction, state the goal of your design project and give a brief overview of your approach.

Our goal for this design project is to build JS3Inth, a 17-key synthesizer with multiple synth modulations and multi-state volume control implemented in real-time. We will implement JS3Inth on the Cyclone IV FPGA, and each switch, from left to right, will be mapped to a particular frequency corresponding to notes on a traditional keyboard, expanding through an octave and a half. We plan to implement JS3Inth by creating VHDL entities that will take the inputs of which keys are turned on, get the corresponding notes from a ROM, add up the notes, amplify the result based on the current volume level, output this resultant sound to the speakers, and display information about the current state on the monitor.

**Design constraints and requirements:** Comment on the key requirements, components, peripherals, etc. Show block diagrams of your implementation.

The user should be able to cycle through several ranges of frequencies which represent octaves in a traditional keyboard. We will also allow the user to switch between multiple forms of sound modulation through a button input that will cycle through various programmed sound textures. We will also include a mute button that silences/un-silences the system. We will be also implementing a graphical user interface (GUI) which will be outputted through a video-graphics array (VGA), showing what keys are turned on, what volume level is being used, whether mute is on or off, and which synth modulation is being used. The only components/peripherals will be the Cyclone IV FPGA, the monitor, and the speakers.

Block Diagram:



**Design description:** This section is where you describe your design in detail. Begin with an overview of your design and follow this with sections describing particular sub-systems and features. Describe the data that is passed between functional units? If it is a streaming architecture, analyze the required sampling rate of data between functional units. What optimizations are you using or planning to use to meet design constraints (area, performance, timing, etc).

All inputs and outputs are sent through FIFO buffer to preserve the integrity of the signals. Initially, we have 18 switch inputs which are connected to a read-only memory (ROM). Each switch input will correspond to a point in memory which will then be outputted to a frequency adder. Additionally, we will be multiplying the output of the frequency adder by a certain amount determined by the state of the octave state machine.

A finite state machine is implemented to determine the current synthesizer being used. Once a certain synth has been decided, the correct function inputs are pulled from the ROM and put into the synth modulation block, which combines with the frequency adder outputs to create a modified audio wave.

The volume is controlled through a finite state machine that cycles through 5 levels, these levels determine the maximum amplitude of the output signal. The output signal of the synth modulator is multiplied is multiplied to increase the amplitude until it hits the correct volume. This signal is sent to the Line Out module to the user’s speakers. The mute button toggles the Line Out module on and off.

All inputs are also passed through our VGA module, which reads the inputs and displays a simple user interface to help the user keep track of any of the keys, volumes, and octaves running in the system.

**Performance/testing:** In this section discuss how you have tested your design in simulation and assessed its performance. Provide some simulation wave diagrams if appropriate to show proper functionality. Don't overdo it though. Point out any simulation errors or problems you are working on.

**Development Plan:** Provide an updated gantt chart and description of where you are at in terms of completion of tasks. What are the next steps? If you have synthesized your design, have you placed it on the board? What is required to complete the design on schedule?

Original Gantt (OG) Chart:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task Name | Duration |  | Start | Finish | Predecessors | Resource Names |
| **Simulation** | **14.75 days** |  | **Fri 4/10/15** | **Thu 4/30/15** |  |  |
| **Theory** | **10 days** |  | **Sun 4/12/15** | **Sun 4/26/15** |  |  |
| Determine how the waveform should look | 8 hrs |  | Sun 4/12/15 | Thu 4/16/15 |  | Jason A, Spencer W |
| Identify design bottlenecks | 2 hrs |  | Sun 4/26/15 | Sun 4/26/15 |  | Jason A, Spencer W |
| **VGA** | **13.63 days** |  | **Sun 4/12/15** | **Thu 4/30/15** |  |  |
| Determine how the screen output should look | 5 hrs |  | Sun 4/12/15 | Sun 4/12/15 |  | Sebastian R |
| **VHDL** | **10.13 days** |  | **Thu 4/16/15** | **Thu 4/30/15** | **6** |  |
| Write Rendering Block | 8 hrs |  | Thu 4/16/15 | Sun 4/19/15 |  | Sebastian R |
| Write Sync Block | 2 hrs |  | Thu 4/23/15 | Thu 4/23/15 |  | Sebastian R |
| Write test bench | 2 hrs |  | Sun 4/26/15 | Sun 4/26/15 | 8,9 | Sebastian R |
| Testing + Debugging | 6 hrs |  | Sun 4/26/15 | Thu 4/30/15 | 10 | Sebastian R |
| VGA Done | 0 days |  | Thu 4/30/15 | Thu 4/30/15 | 11 | Sebastian R |
| **Synthesizer** | **13.75 days** |  | **Sun 4/12/15** | **Thu 4/30/15** |  |  |
| **Theory** | **1.38 days** |  | **Thu 4/16/15** | **Sun 4/19/15** |  |  |
| Determine Synth functions | 8 hrs |  | Thu 4/16/15 | Sun 4/19/15 | 3 | Jason A, Spencer W |
| **VHDL** | **13.75 days** |  | **Sun 4/12/15** | **Thu 4/30/15** |  |  |
| Write FIFO | 4 hrs |  | Sun 4/12/15 | Sun 4/12/15 |  | Ian F |
| Write Rom | 4 hrs |  | Sun 4/12/15 | Thu 4/16/15 |  | Ian F |
| Write Synthesizer | 6 hrs |  | Thu 4/23/15 | Sun 4/26/15 | 15 | Ian F |
| Write test bench | 3 hrs |  | Sun 4/26/15 | Sun 4/26/15 | 17,18,19 | Ian F |
| Testing + Debugging | 4 hrs |  | Sun 4/26/15 | Thu 4/30/15 | 20 | Ian F |
| Synthesizer Done | 0 days |  | Thu 4/30/15 | Thu 4/30/15 | 21 | Ian F |
| **Amplitude + Frequency** | **11 days** |  | **Fri 4/10/15** | **Sun 4/26/15** |  |  |
| **Amplitude Modulation** | **11 days** |  | **Fri 4/10/15** | **Sun 4/26/15** |  |  |
| **Theory** | **6 days** |  | **Fri 4/10/15** | **Sun 4/19/15** |  |  |
| Determine how to amplify signal based on different volume | 6 hrs |  | Fri 4/10/15 | Sun 4/19/15 |  | Jason A, Spencer W |
| **VHDL** | **10 days** |  | **Sun 4/12/15** | **Sun 4/26/15** |  |  |
| Write Input | 2 hrs |  | Sun 4/12/15 | Thu 4/16/15 |  | Sebastian R |
| Write FIFO | 3 hrs |  | Sun 4/19/15 | Thu 4/23/15 |  | Sebastian R |
| Write Amplitude Modulation Block | 4 hrs |  | Sun 4/19/15 | Thu 4/23/15 | 26 | Sebastian W |
| Design testbench | 2 hrs |  | Sun 4/26/15 | Sun 4/26/15 | 28,29,30 | Jason A, Spencer W |
| Testing + Debugging | 3 hrs |  | Sun 4/26/15 | Sun 4/26/15 | 31 | Sebastian W |
| Amplitude Done | 0 days |  | Sun 4/26/15 | Sun 4/26/15 | 32 | Sebastian W |
| **Frequency Adder** | **6.38 days** |  | **Thu 4/16/15** | **Sun 4/26/15** |  |  |
| **Theory** | **0.38 days** |  | **Thu 4/23/15** | **Thu 4/23/15** |  |  |
| Determine how to add signals | 4 hrs |  | Thu 4/23/15 | Thu 4/23/15 |  | Jason A, Spencer W |
| **VHDL** | **6.38 days** |  | **Thu 4/16/15** | **Sun 4/26/15** |  |  |
| Write Inputs | 2 hrs |  | Thu 4/16/15 | Thu 4/16/15 |  | Ian F |
| Write FIFO | 3 hrs |  | Sun 4/19/15 | Sun 4/19/15 |  | Ian F |
| Write ROM | 3 hrs |  | Sun 4/19/15 | Sun 4/19/15 |  | Ian F |
| Write Frequency Adder block | 3 hrs |  | Sun 4/26/15 | Sun 4/26/15 | 36 | Sebastian W |
| Design Test bench | 2 hrs |  | Sun 4/26/15 | Sun 4/26/15 | 38,39,40,41 | Jason A, Spencer W |
| Testing + Debugging | 3 hrs |  | Sun 4/19/15 | Sun 4/19/15 |  | Sebastian W |
| Frequency Done | 0 days |  | Sun 4/19/15 | Sun 4/19/15 | 43 | Sebastian W |
| **Output** | **5 days** |  | **Sun 4/12/15** | **Sun 4/19/15** |  |  |
| Research how to output audio | 2 hrs |  | Sun 4/12/15 | Sun 4/12/15 |  | Sebastian W |
| Write Audio out block | 3 hrs |  | Sun 4/12/15 | Sun 4/12/15 | 46 | Sebastian W |
| Write Test bench | 2 hrs |  | Sun 4/12/15 | Thu 4/16/15 | 47 | Sebastian W |
| Testing + Debugging | 4 hrs |  | Thu 4/16/15 | Sun 4/19/15 | 48 | Sebastian W |
| Audio Output Done | 0 days |  | Sun 4/19/15 | Sun 4/19/15 | 49 | Sebastian W |
| **Integration** | **4.75 days** |  | **Thu 4/30/15** | **Thu 5/7/15** | **1** |  |
| Write Test bench | 2 hrs |  | Thu 4/30/15 | Sun 5/3/15 |  | Jason A, Spencer W |
| Testing + Debugging | 6 hrs |  | Sun 5/3/15 | Thu 5/7/15 | 52 | Ian F, Jason A, Sebastian W, Sebastian R, Spencer W |
| Integration Done | 0 days |  | Thu 5/7/15 | Thu 5/7/15 | 53 |  |
| **Implementation** | **3.63 days** |  | **Sun 5/10/15** | **Thu 5/14/15** | **51** |  |
| Load code onto FPGA | 2 hrs |  | Sun 5/10/15 | Sun 5/10/15 |  | Ian F, Jason A, Sebastian W, Sebastian R, Spencer W |
| Debug | 6 hrs |  | Sun 5/10/15 | Thu 5/14/15 | 56 | Ian F, Jason A, Sebastian W, Sebastian R, Spencer W |
| Project Complete | 0 days |  | Thu 5/14/15 | Thu 5/14/15 | 57 |  |
| **Other** | **20 days** |  | **Thu 5/7/15** | **Thu 6/4/15** |  |  |
| **Documentation** | **6.5 days** |  | **Thu 5/7/15** | **Sun 5/17/15** |  |  |
| Write Project Report | 3 hrs |  | Thu 5/7/15 | Thu 5/7/15 | 51 | Ian F, Jason A, Sebastian W, Sebastian R, Spencer W |
| Write Final Report | 5 hrs |  | Thu 5/14/15 | Sun 5/17/15 | 55 | Ian F, Jason A, Sebastian W, Sebastian R, Spencer W |
| **Major Milestones** | **20 days** |  | **Thu 5/7/15** | **Thu 6/4/15** |  |  |
| Project Report | 0 days |  | Thu 5/7/15 | Thu 5/7/15 |  |  |
| Final Report + Demo | 0 days |  | Thu 6/4/15 | Thu 6/4/15 |  |  |

**Problems Encountered Along the Way and How They Were Solved:**

The biggest hurdle we encountered during development of the prototype of the JS3INTH was figuring out how to get a digital audio signal out of the Altera DE2-115 successfully. This proved to be much more tedious than what we were originally anticipating since we were not sure on where to send our standard 24 bit audio signal in order to have it recognized and decoded by the on board Wolfson WM8731 audio codec properly. After scouring the internet and DE2 user manual on how to set up the signal in the necessary serialized format, we eventually had to spend several days studying the datasheet for the Wolfson codec to learn how to communicate digitally with the chip as to set it up to receive and correctly understand the signal being sent to it through the I2C protocol. Another more minor problem we had at the beginning of the quarter was the fact that 2 of our members had no prior FPGA experience, however they very quickly picked up programming in VHDL and are now very valuable members of the team.

**Recent Changes in the Design**

The most recent changes in the design have been how audio has been processed, changing from a range of 18 base tones to a range of 16 tones as the more easily instantiate and implement ROM modules (easier usage of hexadecimal).