**EECS 392: FPGA Systems Design Projects – Project Plan**

*Team Name*

FPOGs

*Team Members*

* Jason Arkin
* Ian Feeney
* Sebastian Rodriguez
* Spencer Williams
* Sebastian Witkowski

*Project Title*

JS3Inth

*Project Description / Overview*

JS3Inth is an 18-key synthesizer with multiple synth modulations and state volume control implemented in real-time. This project will be implemented in the Cyclone IV EP4CE115F29C7N field programmable gate array (FPGA). Each switch, from left to right, will be mapped to a particular frequency corresponding to notes on a traditional keyboard format, expanding through an octave and a half. The user will 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. Also included is a mute button that provides the option to silence the system. In addition to this functionality, we will be also implementing a graphics user interface (GUI) which will be outputted through a video-graphics array (VGA).

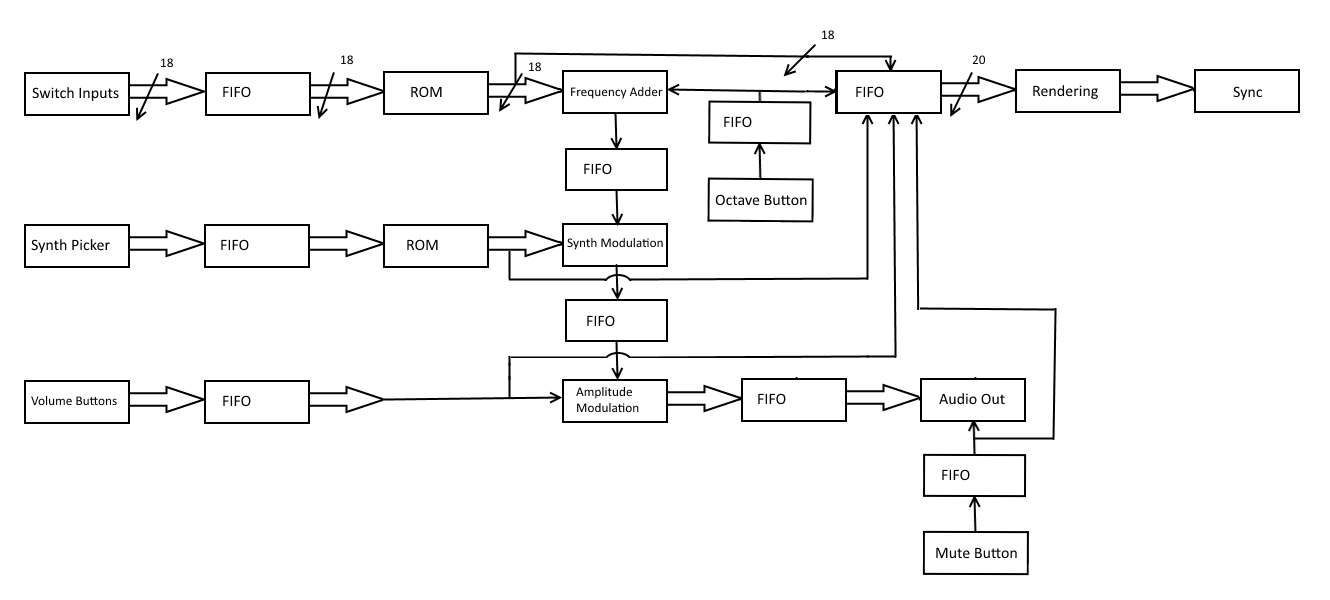
*System Architecture / IO / Peripherals / Memory*

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.



*Tasks / Gantt Chart*

Note that the Gantt Chart was built in Microsoft Project. For a better representation of the subtask structure, please email Spencer Williams (spencerwilliams2016@u.northwestern.edu).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |  |  |

*Members Responsibilities*

* VGA Module – Sebastian R
* Synthesizer Module – Jason A, Ian F, Spencer W
* I/O Module – Sebastian W

In general, people will help out other components once they’ve completed theirs.

*Development Plan*

Our project will be done in 3 major steps:

* Simulation
* Integration
* Implementation

Simulation is composed of the 3 major modules which will be led by different members on the team. These are:

* VGA
* Synthesizer
* Amplitude + Frequency

Specifically, the Amplitude + Frequency module will be broken down into the sub-modules listed below:

* Amplitude Modulation
* Frequency Modulation
* Audio Output

These divides were made so that we could create better workflow with our members, divide the difficulty of the project among our members with VHDL experience, and hit all project deadlines. In general our development plan is to build multiple smaller modules and continually integrate and test them until we have a testable full project. Full test benches will be created and used on all major modules and sub-modules before they can be integrated into the Final Project. After all of this is completed, we will move on to implementation on the FPGA and ensure that our simulations are accurate.

*Deliverables*

* Working JS3Inth implemented on the Cyclone IV E FPGA; demonstration on speakers and VGA monitor
* Final Report

*Project Costs*

*by Project Components*

|  |  |
| --- | --- |
| Component | Cost |
| Simulation | $19,950 |
| Integration | $5,100 |
| Implementation | $6,000 |
| Other | $3,000 |

*by Person*

|  |  |  |
| --- | --- | --- |
| Name | Hours | Cost |
| Jason Arkin | 45 | $6,750 |
| Ian Feeney | 46 | $6,900 |
| Sebastian Rodriguez | 46 | $6,900 |
| Spencer Williams | 45 | $6,750 |
| Sebastian Witkowski | 46 | $6,900 |