Project Nock-off Entertainment System

Cody Anderson
Ben Nollan
Morgan Skrabut
Ryan Price

October 8, 2017

Jeremy Thomas

Dept. of Electrical Computer Engineering

ECE 310L, ECE410L, Conjoined 3rd and 4th Year CE Project

Abstract

Project Nock-off Entertainment System (NoES) aims to create a Nintendo Entertainment System (NES) clone in SystemVerilog, which will be instantiated on an FPGA. The design of the NES clone will be nearly identical to the design of the actual NES console. In order to create a very similar clone, the discrete IC's on the NES will be modeled by discrete modules in SystemVerilog. Extensive validation and testing will go into checking the correctness of the timing and results of CPU and Picture Processing Unit (PPU) instructions. If we succeed in creating this clone, the NoES will be capable of running original NES games.

1 Introduction

Remembered and loved by many individuals today, the Nintendo Entertainment System released in North America in 1985 is a gaming console icon of nostalgia and joyful memories. This console has been brought back to life by few individuals on the FPGA and we propose to achieve the same endeavor this year. We are motivated to recreate the NES ourselves because it is a very rare and highly sought after console with its roots deep in the hearts of many gamers including ours and those of our friends, families, and academic colleagues. We will be referencing two individuals who completed this project themselves (Dan Strother [1] and Jonathan Ganyer [2]). Our success in completing this project will be defined by our ability to run the game Super Mario Bros. on our FPGA embedded console by the end of the Fall semester.

2 Methods, Techniques, and Design

2.1 Block Diagram

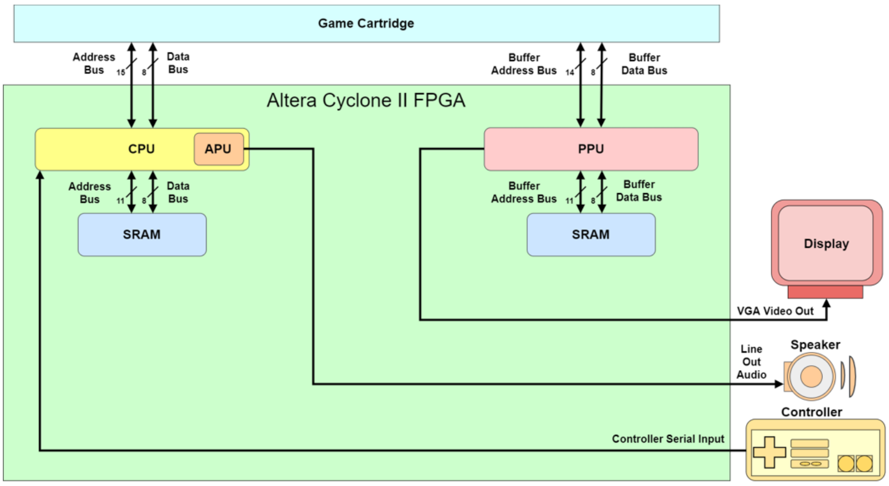


Figure 1: Project Block Diagram showing simplified flow of data and physical structure of components and modules

2.2 Central Processing Unit (CPU)

The CPU in the NES is based on the 6502 CPU, but with a few changes. The CPU operates at a frequency of 1.79 MHz and the Audio Processing Unit (APU) is built into the same package. In terms of I/O, the CPU has a 16-bit address bus and a bi-directional 8-bit data bus.

2.3 Picture Processing Unit (PPU)

The NES PPU is used for generating two-dimensional scenes by using composite video output. It operates at 21.48 Mhz and outputs one pixel per clock cycle. It also has a dedicated 2K SRAM Chip for storing data. This chip communicates with the CPU by using the CPU data bus. It also is connected to three of the CPU address lines, these are used to select the PPU register.

2.4 Chip Interconnects

In the NES, all of the devices share the data lines and take turns outputting to them. This method won't work when using SystemVerilog to describe the hardware as it doesn't have the capability to resolve multiple driving nets. The current solution that we intend to use is the method of separating the inputting and outputting of each device. Then all the data buses will be passed through a multiplexer. This system should mitigate some of the issues with multi driver nets and shouldn't pose any problems with data propagation.

2.5 Hardware Interface

We are going to connect a cartridge interface and two controllers to the FPGA using the two 40-pin expansion ports on the FPGA. The cartridge is a 72-pin socket, most of which are used. Each controller connector has 7 pins, four of which connect to the CPU data bus.

2.6 Audio Processing Unit (APU)

The APU is made up of five output channels. These channels include:

- Two (2) square wave generators
- One (1) triangle wave generator
- One (1) sample generator
- One (1) noise generator

Each of these channels are fed into their own DAC. Each channel is then combined in the APU Mixer. The APU Mixer is an analog circuit, but can be approximated easily with digital logic.[3]

3 Schedule and Task Breakdown

Task Distribution Among Team Members:

- Cody Anderson is in charge of overlooking the task of CPU creation
- Morgan Skrabut is in charge of overlooking the task of PPU creation
- Ryan Price is in charge of the cartridge and controller connectivity as well as the APU
- Ben Nollan is in charge of designing module framework and all testing and verification

First Half Semester Milestones (Week 7):

- Implement twenty-five opcodes from Super Mario Bros. on the CPU
- PPU Video Output of an image from Super Mario Bros.
- Cartridge to FPGA connectivity

Second Half Semester Milestones (Week 12):

- Implement all CPU opcodes required to run Super Mario Bros.
- Finished basic implementation of the PPU
- Complete controller and cartridge connectivity

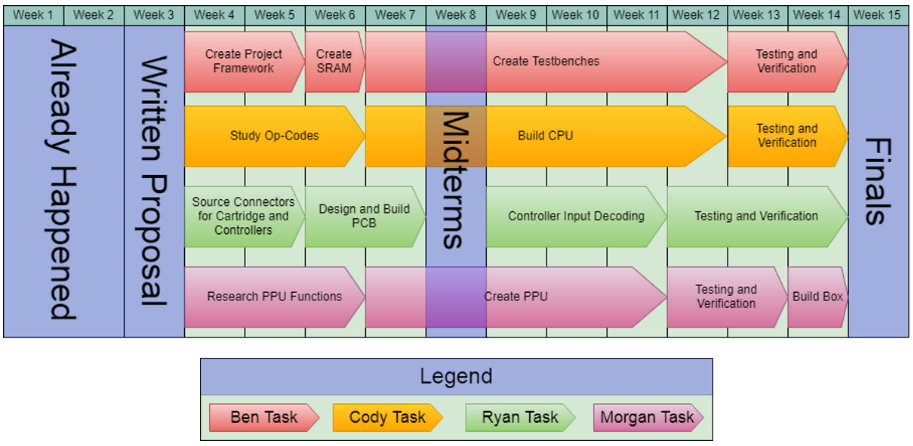
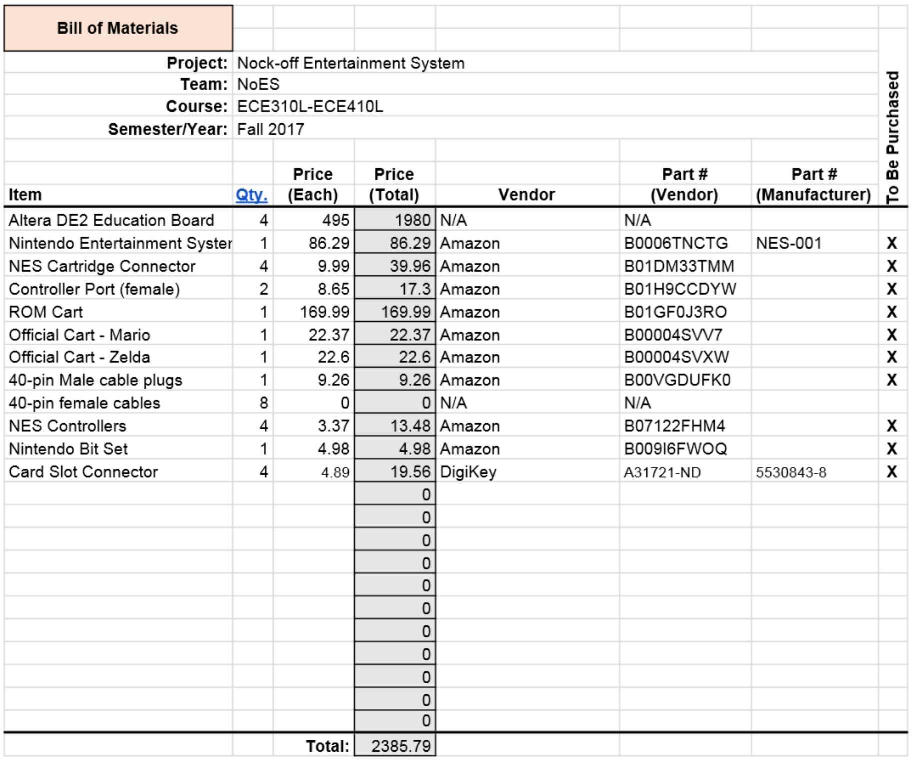


Figure 2: Project task breakdown for this semester.

4 Bill of Materials



Bill of Materials

Project: Nock-off Entertainment System
Team: NoES
Course: ECE310L-ECE410L
Semester/Year: Fall 2017

		Price	Price		Part #	Part #	
Item	**Qty.**	**(Each)**	**(Total)**	**Vendor**	**(Vendor)**	**(Manufacturer)**	**To Be Purchased**
Altera DE2 Education Board	4	495	1980	N/A	N/A		
Nintendo Entertainment Syster	1	86.29	86.29	Amazon	B0006TNCTG	NES-001	X
NES Cartridge Connector	4	9.99	39.96	Amazon	B01DM33TMM		X
Controller Port (female)	2	8.65	17.3	Amazon	B01H9CCDYW		X
ROM Cart	1	169.99	169.99	Amazon	B01GF0J3RO		X
Official Cart - Mario	1	22.37	22.37	Amazon	B00004SVV7		X
Official Cart - Zelda	1	22.6	22.6	Amazon	B00004SVXW		X
40-pin Male cable plugs	1	9.26	9.26	Amazon	B00VGDUFK0		X
40-pin female cables	8	0	0	N/A	N/A		
NES Controllers	4	3.37	13.48	Amazon	B07122FHM4		X
Nintendo Bit Set	1	4.98	4.98	Amazon	B009I6FWOQ		X
Card Slot Connector	4	4.89	19.56	DigiKey	A31721-ND	5530843-8	X
			0				
			0				
			0				
			0				
			0				
			0				
			0				
			0				
			0				
			0				
			0				
		Total:	2385.79				

Figure 3: Project Bill of Materials

References

[1] D. Strother, "Fpga nes," 2010. [Online]. Available: https://danstrother.com/fpga-nes/

[2] J. Ganyer, "Nes fpga." [Online]. Available: https://jonathanganyer.wordpress.com/nes-fpga/

[3] 2017. [Online]. Available: http://wiki.nesdev.com/w/index.php/Nesdev_Wiki