The Waffle House

Updated Cohort Progress Report

November 16, 2018

Description:

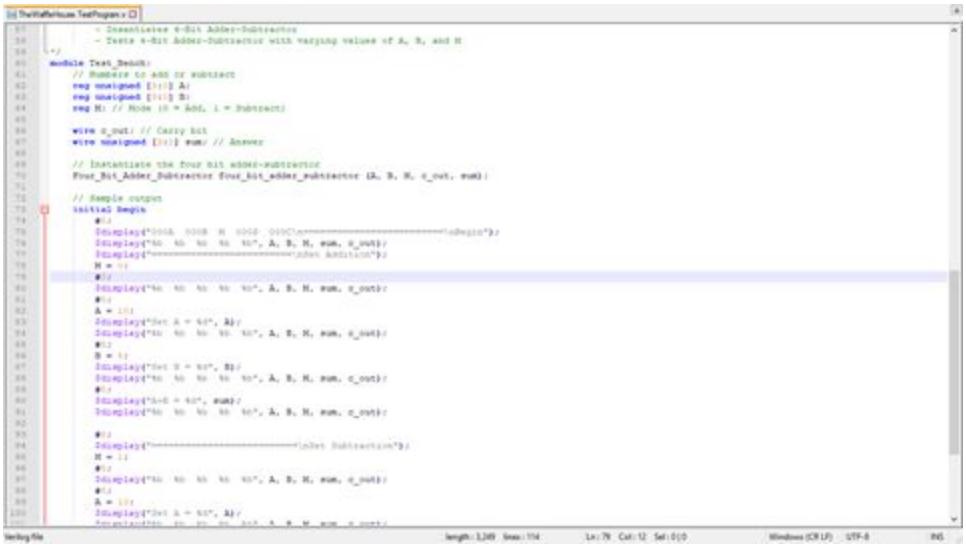
For our project we plan to create an ALU. An ALU is an arithmetic-logic unit that is a part of all computer processors. From simply adding, subtracting, multiplying, and dividing two numbers to shifting numbers left and right, the ALU allows all computation and processing to take place in any device that requires arithmetic. A simple version of an ALU is used in things that do not require complex computations to take place, such as in a microwave. These ALU's only have the addition and subtraction functionality, as if it needs to multiply or divide it can just make several passes through either addition or subtraction to simulate multiplication and division respectively. The complete ALU's that are used in more complex equipment like phones or tablets have their own dedicated systems for multiplication and division, as well as a logic unit that allows us to access logic gates like AND, or NOT whenever the computer needs to. What is so impressive about an ALU is that it is really all that is necessary to start building the immensely complicated things such as the control module for the ISS or video game engines and programming languages. In addition to the ALU, we plan to also have a project created using redstone in Minecraft. Redstone can be used to simulate logic gates and make logical operations. Currently the project idea is to create a clocktower, which will showcase circuit logic and memory.

Member Tasks:

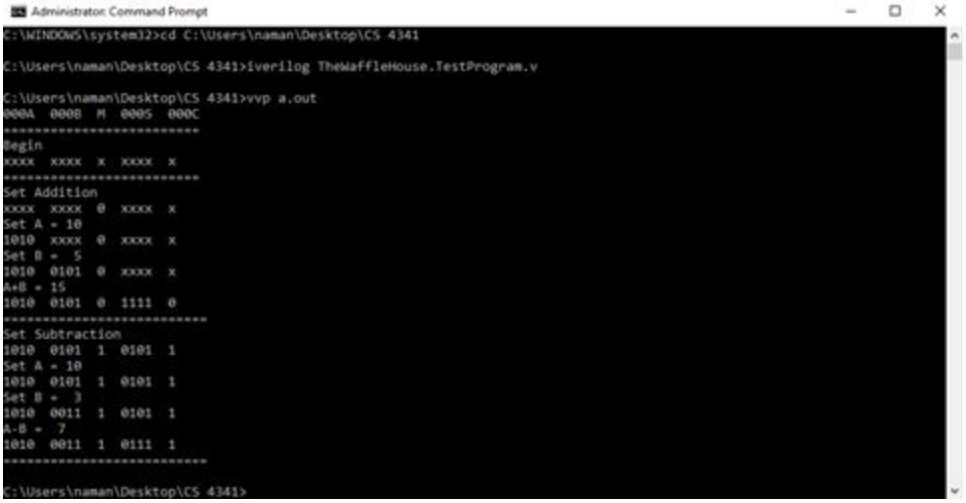
Tasks\Members	Tennessee	Naman	Eric	Ben	Philip
Verilog code		X			
Circuit Diagrams	X				
Extra Credit					X
Parts List	X				
Diagrams			X		
Writeup			X		
Input List				X	
Output list				X	
Interface List				X	
Module List	X				
State Machines		X			

Software Discovery:

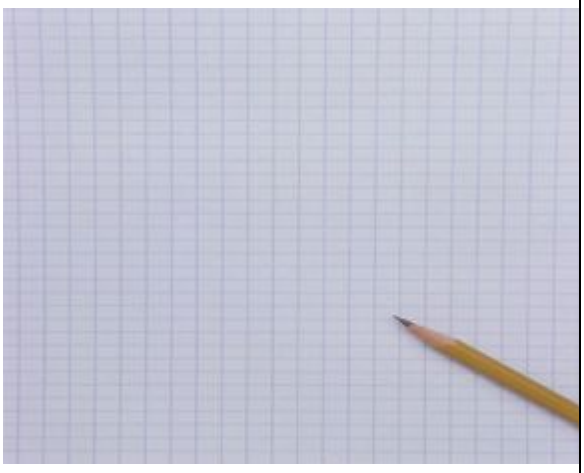
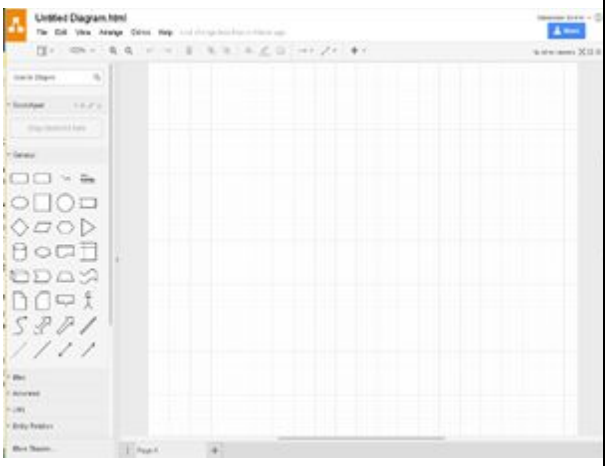
Editing source code of Verilog programs (Notepad++):



Compiling and running Verilog programs in iVerilog 10.1.1 (Windows 10 Administrator Command Prompt):



Diagrams and Circuit Drawings:

Initial Software	~~Backup~~
	

Notepad++ was chosen for editing Verilog source code, because it has an intuitive sense of syntax highlighting and tabbing that coincides with Verilog which can make editing much easier and allow us to focus directly on the content of the code required in the project. It is also widely available for the operating systems that everyone in the group uses, so sharing files should be seamless.

On the cohort computer's machine, Windows 10 is the operating system, so a compatible version of iVerilog 10.1.1 was installed for it. The Windows Administrator Command Prompt is used for easily compiling and running the programs and can be widely used across other group members Windows machines.

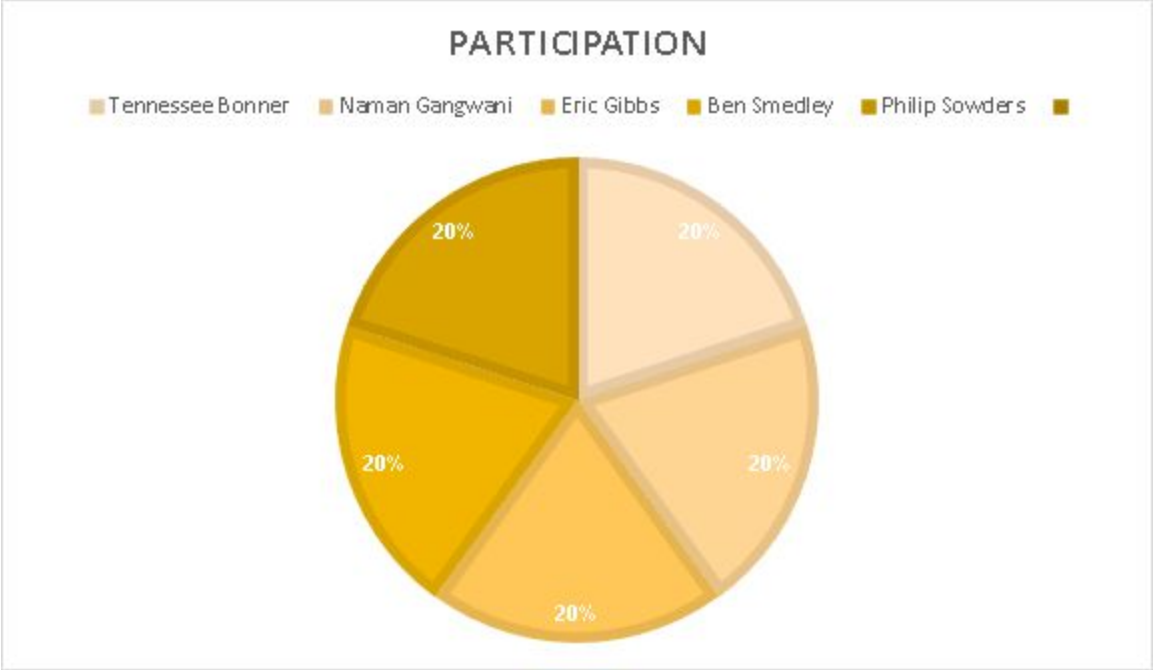
The preferred circuit drawing software this group uses is good old fashion pen/pencil and paper. However, when things get too complicated and unwieldy to do on a sheet of paper, we use Draw.io to lay out any circuit design we need to do. Pen and paper was chosen because it is really easy to wrap your head around in the moment and tends to be faster for us. There isn't any

having to wrestle with the computer and menu systems to jump in and write exactly what we want, only finding a piece of paper and something to write on it. There's something much more personal about physically writing or drawing something out that helps us really understand what we are trying to accomplish. While mistakes are liable to happen, and erasing big mistakes ends with the paper looking all messy and maybe even difficult to read, there's a sense of accomplishment when something that you drew personally ends up coming together to form the correct picture. We've found that through the simplicity and general ease of being able to pick up anything that we can mark up and drawing out the problem tends to be our go to for any circuit drawing needs.

While pen and paper are our preferred method of drawing circuits~~, there's something to be said about using software to draw circuits as well. Since it was mentioned in class and is free, we decided to investigate using Draw.io and found that it works really well. It took a while to find out just how to draw logic gates and circuit designs in the interface, seeing as though there is no tutorial and the menu system doesn't have a specific label for logic gates. However, once we found out where everything is and how to get it to work all together, we found it surprisingly easy to put together circuits. No longer did we have to go through the pain of having to write out the same logic gates over and over againall it took to put down as many as we needed is a few clicks of the mouse. When dealing with complex and large circuits, we could see the true use of Draw.io to keep everything neat and orderly, while reducing a bit overhead and hand strain.~~

We did all of our circuit diagrams and state machines with pen and paper.

Participation Census:



Cohort Meeting	1	2	3	4	5	6	7	8	9	10
Tennessee Bonner	X	X	X	X	X	X	X	X	X	X
Naman Gangwani	X	X	X	X	X	X	X	X	X	X
Eric Gibbs	X	X	X	X	X	X	X	X	X	X
Ben Smedley	X	X	X	X	X	X	X	X	X	X
Philip Sowders	X	X	X	X	X	X	X	X	X	X

	Cohort Task Completed?
Tennessee Bonner	X
Naman Gangwani	X
Eric Gibbs	X
Ben Smedley	X
Philip Sowders	

Everyone has attended all the cohort meetings and ~~as of 10/5/2018 none of us have completed our cohort tasks.~~ as of 11/16/2018, all of us have completed our Cohort Tasks except for Phillip, who has begun working on the bonus and is on track for finishing it by bonus presentation day.