**Module 1 - Discussion - Section 82**

Discussion Forums are not like class homework. You are expected to help each other, and ask questions. If something is not working out, ask right away. Remember, your grade in the discussion forum is based on how much you participate. I will jump in and help, and add my comments to the class, however the forum is really a conversation. Your observations, insight, hints, and solutions are how we measure if the material is being understood, and if the class needs additional help.

How do you start the discussion? Don't feel that you have to jump in with a solution as your first post. Talk about what is being asked, get ideas from your classmates. Post issues that you run into trying to solve the problem. Post your code if you are having problems with it. Don't be afraid to ask for help. Each post helps the class find it's way through the problem.

Your grade is not strictly bookkeeping, While I am looking for a minimum of 6 postings spread out over the week, I am looking for something significant at least once during the week. A posting that shows how to do something with the code, with the tools, or to solve a problem counts. In order to keep the grading from being arbitrary, I have adopted a grading criteria for the discussion forum:

* 5 - 6 you will probably get a 10 if you turned in a good post, and helped out the discussion
* 4 you will probably get a 9 if you turned in a good post, but If you didn't you would probably get a 7-8 depending on what you posted
* 3 - If you turned in a good post, you will probably get a 7 (I have been very generous up to now). You can also get a 5-6 if you did not do a detailed post.
* 2 - You will get a 6 if you turned in a good post, and a 5 if you didn't
* 1 - you will get a 5 if you turned in a good post, and 4 if you didn't
* 0 - You will get a 0

This semester we will be simulating and experimenting with various Verilog designs. The class will be using Xilinx Vivado 2020.2 or newer:

* Xilinx Vivado 2020.2 (You could install the latest version. The Li verilog implementations will work on the latest version.)
* Download CodeScape (MIPS SDK): https://www.mips.com/develop/tools/codescape-mips-sdk/download-codescape-mips-sdk-essentials/ This will allow you to assemble MIPS code into hex files that will be loaded into the Vivado code. A script on our GitHub (assemble.sh) will take the MIPS assembly and generate hex files. Note that incase this link does not work, the files will also be located in the readings section of this module.

Vivado is a large package and should be downloaded to your system, licenses (free) should be obtained.

We will also be using the Nexys 4 DDR evaluation board (or Nexys A7 board) along with the pmodUSBUart module. To facilitate its use, we are first providing two basic designs via zip file (the Readings). You should also apply to the bitbucket.org site and request a free account. Once you have the account, provide your username to Nick Beser. We  will add you to the repository and you will have access to the rest of the class designs. The login to the bitbucket.org site is: https://id.atlassian.com/login

Additional reading is provided from the system:  Once your account is setup, you can follow this link to copy the repository onto your computer: https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud

There is the source code (Verilog) to the single cycle MIPS processor which is discussed in both Li's Chapter 5 and Patterson and Hennessey's text. This is provided as a way of verifying you can run the following:

1. Codescape assembly tools
2. Vivado tools which include both simulator and bitstream for Nexys 4 DDR board
3. Able to download a bitstream to the Nexys 4 DDR board
4. Able to manipulate the verilog code to change parameters and watch the program execute on the simulator
5. Able to modify the assembly code play with memory mapped I/O (switches and LED's).
6. Become comfortable using the school github so that as we modify these designs to add I/O, instructions, etc, we will try not to break the basic design, and build on our new designs.

A word about order of operations: Our class follows the Patterson and Hennessey Text  which means that we learn MIPS, HDL followed by computer arithmetic, and then various processor designs, followed by memory, cache/VM and I/O. In order to do that in a meaningful way with hardware labs, we will have to jump around a bit, and learn things in different order. I am providing you with a complete (although toy) design for a MIPS processor so we can get comfortable simulating and working designs. We will be able to learn MIPS assembly by using the various tools and then trying them out on a "Monolithic" design that we will eventually break into to add features. As we progress we will add new designs, and modify existing designs.

So, for this week's discussion forum do the following:

1. Get your tools in place (Download Vivado (latest version) and CodeScape). We will also be using the pmodUSBUart to download program code into the processor emulations on the FPGA board.
2. Get your account setup on bitbucket - send your username to Nick so we can add you to the class repository.
3. Unzip the design files to get started. (located in the readings section)
4. Assemble the MIPS code
5. Simulate (modify the assembly to reduce the loop time if necessary) Play with the test bench to vary the input and see the processor execute from the inside.
6. Implement the design, and generate the bit stream
7. Download it and check out the I/O
8. During this entire exercise report back on the discussion forum your observations, issues, questions, and comments.
9. Try out the pmodUSBUart interface by downloading a new program into the existing design. In the mips-cpu.zip file under software is a readme file which is also documented in the getting started guide. There will be updates made early in the week to provide a means of generating intelHex files.
10. Try out different MIPS programs, using the Vivado simulator, or the MARS simulator. Don't be afraid to break the code, you learn more that way.

The two MIPS assembly examples that are on the repository do the following:

1. write 8 digits to the LED's on the board
2. read the Switches and set the LED's.

As an exercise, modify the program to program the LED's to count in ascending order. You could also modify the LED sequence to perform a fibonacci count instead of increasing sequence.

Note that the single cycle processor has a limited instruction set. You can examine the verilog code to see the instructions, and also examine Chapter 5 from Li's text (The pdf is included in this weeks readings.).

I am also trying something new this year. I have been corresponding with Yamin Li about extending his verilog designs to RISC-V. The focus of this class is largely on MIPS, however, the work that Yamin Li has done offers the opportunity to try out some original design in a similar architecture. In the next few modules when we start to discussion arithmetic operations, I will introduce you to the multiply and divide approach used by RISC-V. When we examine the single cycle MIPS and the multi-icycle MIPS, we can also look at the RISC-V design. Many of the design concepts that Yamin Li has provided in his text will work on the new design. As background, here is a briefing on RISC-V instruction set:

[525.612Module01RISCVInstructionSetArchitectureV2.pdf](https://jhu.instructure.com/courses/60023/files/9678219?wrap=1)

[Download 525.612Module01RISCVInstructionSetArchitectureV2.pdf](https://jhu.instructure.com/courses/60023/files/9678219/download?download_frd=1)

The RISC-V simulators are in the readings section. Try it out.

**Homework Format**

I am supplying this comment as guidance for turning in homework.

 If you have multiple documents, please zip them into a single zip file. I prefer pdf format. The way I grade is I create a directory on the school OneDrive, and copy them in. My ipad will access it directly, and I can write on the pdf's. It makes grading easy and if anybody wants to see the graded paper, i can send it back to them.

It would be helpful if the filename had your name on it, and the homework number. Makes it easier to put in the student's folder.

If you have hand written homework, please scan it in and make sure it is readable. I have often received a mix of hand written and typed. As long as I can follow it, and can read it, it's fine.

**A gentle reminder**

I enjoy the discussion forum, it let's me see how the class is doing in terms of the topic of the week. I want to remind the class that each discussion forum ends at Sunday night at midnight EST. I grade the discussion forum based on the number of posts. This week's forum has the grading rubic in it. I always tell the class, think of this as a conversation. Do not think of the discssion forum as a polished presentation. Everyone has something to contribute. If you wait until Sunday night, your classmates who have been participating all week long, might not respond. I like to think of it as a conversation while everyone is running out of the door.  A great percentage of the class is participating, but there are some who have not posted.

The rubric is quite generous in terms of grading, but there is one criteria that I always apply. If you don't post, you get zero points for the week.