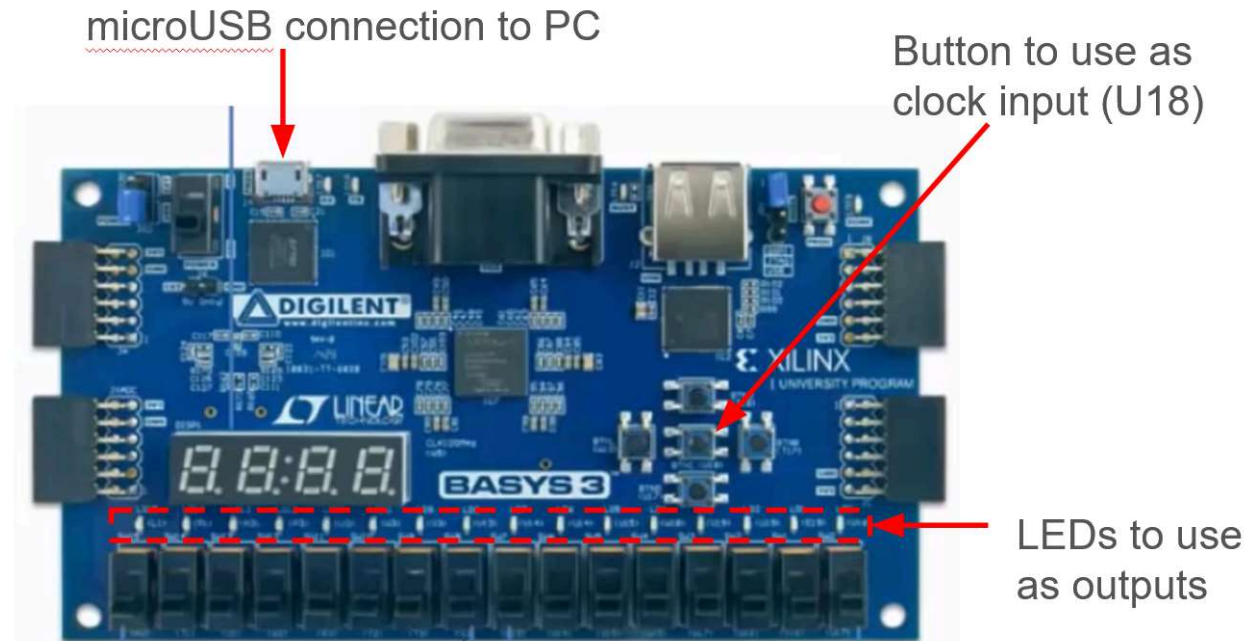


CSE 490/590 Spring 2024
Project 1 Hardware Requirement

Basys Board Layout:



For this project, you will only be responsible for connecting the above pins to your module. No switches will be used, no clocks will be used, and no other General Purpose Input/Output will be used besides the center switch on your Basys 3 board. If you look closely at the board, the pin value is placed next to the component. This value will be required for your pin assignment.

Instructions for Demo:

Your demo will involve both an in-person demonstration that the code simulates the outputs correctly, proof of placing the program on the Basys board correctly, and an individual questionnaire. Further instructions for testing your code output for both the simulation and the hardware will be given on the day of your demo. Please make sure all members are present for your demo with at least one group member having the built code to quickly show results. If anyone can not make it to the day of the demo they signed up for, they are responsible for rescheduling and having access to the entire code to demo again for us. Completely missing the demo will result in a 0 for the project. For any questions about demo availability (times, days, and location will be posted later), contact either a TA who will relay the question to Sridhar, or directly contact Sridhar during office hours or by email.

Hardware Requirements:

For the second part of the demonstration, placing the code onto the Basys board is a requirement. Your input, clk/clock, should be connected to a push button, specifically the center push button labeled BTNC <U18>. The value inside the arrows < > refers to the pin number. Use this pin number for pin assignment. This button will be used to simulate the clock. If there are any issues that arise with this push button, we may require reassignment of this pin on short notice.

For outputs, all outputs will be placed on the 16 LEDs placed above the switches. Eight of these LEDs should correlate to the value inside one register and the other eight for a different register. The first eight LEDs (LEDs 0 - 7) should correlate to the value inside rs of the instruction loaded. Therefore, while holding down the push button, we should see the value loaded from rs (could be any of the four registers) displayed on these eight LEDs. The LSB (least significant bit) will be assigned to LD0 and the MSB will be assigned to LD7. The second set of eight LEDs (LEDs 8 - 15) should correlate to the value inside rt. MSB should be on LD15 and LSB should be on LD8.

After the completion of one instruction when the push button is released, the new value of rt (if updated) should be displayed on LEDs 0-7. If rs is equal to rt, then rs should also update its value on the LEDs.

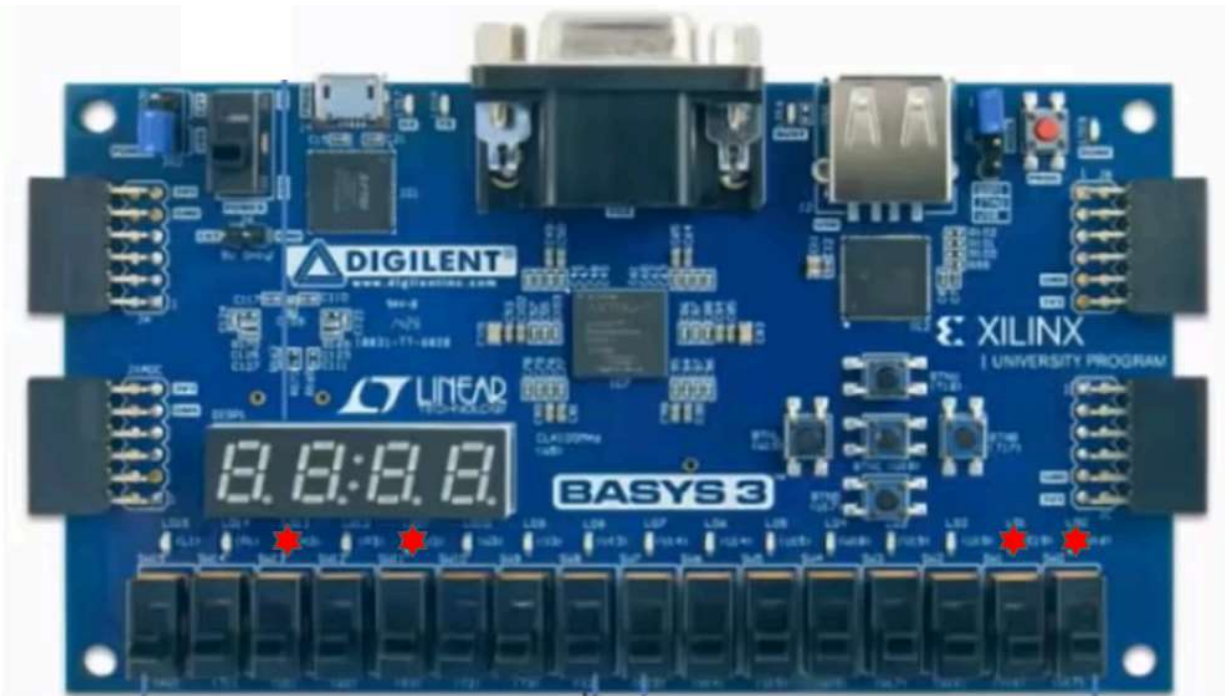
Examples:

Assuming initial values r0: 3, r1: 249, r2: 5, r3: 15

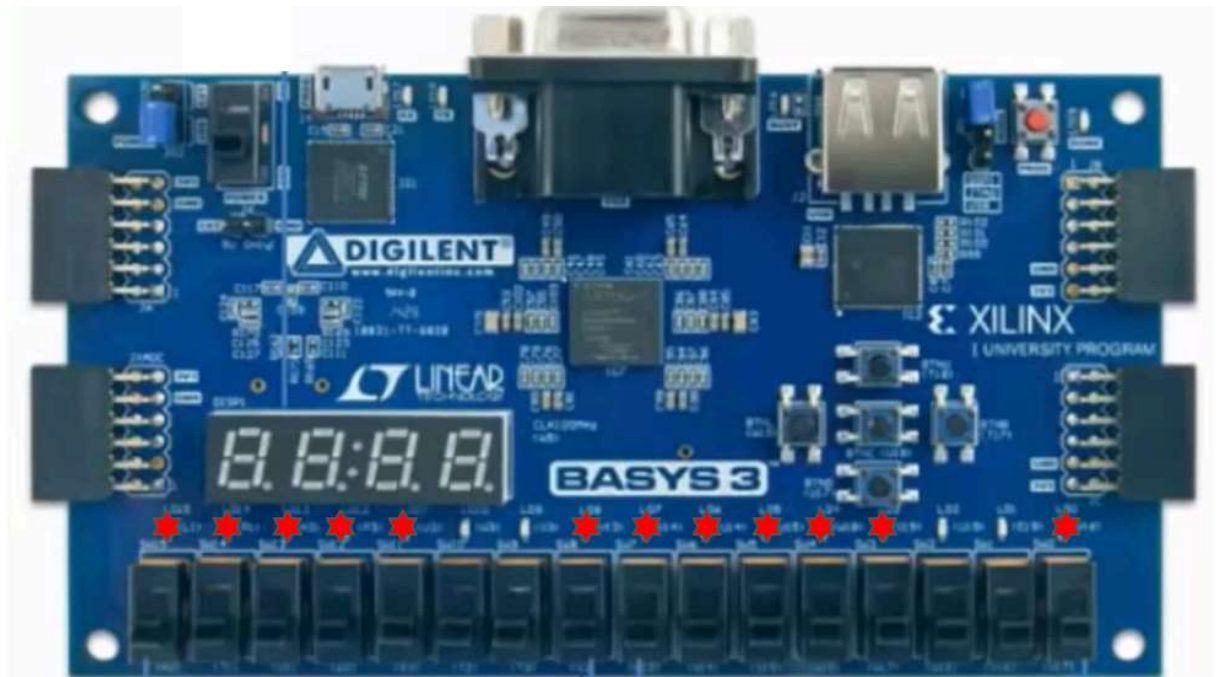
Using the instruction sll r2, r0 which correlates to sll r2, r2, r0, the output should look like this while holding down the button:



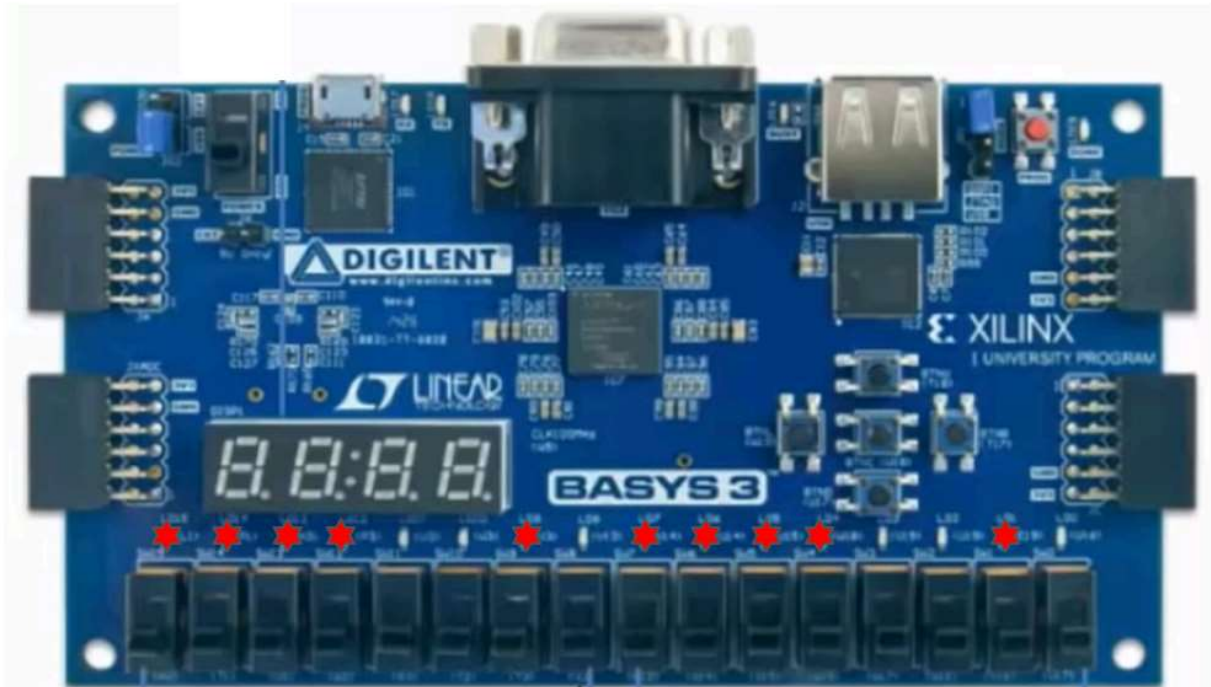
After releasing the button, the value of r_t (in this case r_2) should update to 40:



Say the next instruction is add r_1 , r_1 , then the output after pushing down on the button should be as follows:



Then, after releasing the button, the output should be:



Note that both sides changed since r_s is equal to r_t here.

Edge Cases:

- It does not matter what is outputted on your LEDs before the first instructions runs
- The first instruction should start running on the first button press
- If you have questions about pin assignment, please refer to the Hardware Manager under Program and Debug. Try stuff out, make mistakes, and try to figure it out yourself
 - If you've spent a considerable amount of time on it and still can't figure it out, then bring the question to a TA

If you have any questions or see any mistakes, please don't hesitate to make a post on piazza. Unless your question is already answered in this document. Then please hesitate before posting.