UNIVERSITEIT • STELLENBOSCH • UNIVERSITY
jou kennisvennoot • your knowledge partner

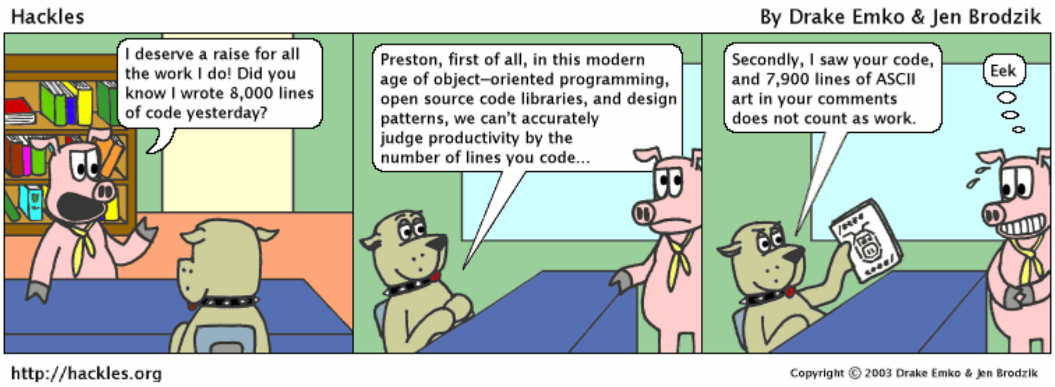
Computer Systems 414

Practical 7

2021

Aim of Practical 7:

1. To get practical experience on the Terasic DE0 FPGA board


Hackles
By Drake Emko & Jen Brodzik
I deserve a raise for all the work I do! Did you know I wrote 8,000 lines of code yesterday?
Preston, first of all, in this modern age of object–oriented programming, open source code libraries, and design patterns, we can't accurately judge productivity by the number of lines you code...
Secondly, I saw your code, and 7,900 lines of ASCII art in your comments does not count as work.
Eek
http://hackles.org
Copyright © 2003 Drake Emko & Jen Brodzik

Rules of Engagement

1. Attendance is required for this practical – if you cannot attend the practical in person make an arrangement with the lab manager to sign out a development board at a time of his choosing.

2. Students are allowed to work in groups of two, but each student must write and submit their own code.

3. Supporting documentation is available on learn.sun.ac.za, but students also have to do their own research.

4.The report is to be submitted on learn.sun.ac.za. The deadline for the reports is (14:00) Wednesday, 02 June 2021. **No late submissions will be accepted.**

5. Do not forget to cite and give credit for any information reported which is not your property.

8. Google is your friend. Any information not given is left out on purpose. Search for your solution and on the internet or relevant manuals and documentation..

Task (Use the uploaded files from SunLearn as required)

Assignment 7A

For this practical we will use Quartus 13.1 and the Terasic DE0 FPGA board.

Please refer to the additional files on the Sunlearn course webpage.

Task (Use the uploaded files from SunLearn as required)

1. Sign out a DE0 FPGA board from the lab manager

2. Read through the user manual for the board.

3. Plug in the board and install the USB Blaster drivers

4. Install the DE0 Control Panel and play with the LED, buttons and the 7 segment display.

5. Launch Quartus and follow the instructions to compile and program the base project.

6. Observe how the compilation process differs from the GHDL projects of our previous practicals. What is the output of the compiled project? Is this a functional simulation only? Is the timing and gate-level placement and routing details available?

7. Modify the base project to include a state machine that uses all three buttons and two more LED's. The state machine has to wait for a Button 0 press before turning on LED0, then for a Button 1 press before turning on LED1 (while LED0 stays on) and then for a Button 2 press before turning LED2 on. When the button sequence is repeated the LED's must turn off in reverse sequence (i.e. LED2 turns off when Button 0 is pressed).

   Please note that the buttons actually need to be debounced. Since this involves timers and can be complex, a workaround would be to design your state machine in such a way that multiple presses of the same button will be discarded.

   This practical also do not include instructions for the use of Modelsim to simulate your design. The state machine is deemed simple enough that it can be debugged by button presses and Modelsim should not be needed.

   If you do feel the need to simulate your design you are welcome to use either Modelsim or GHDL to accomplish this. With Modelsim it is possible to use a waveform editor to generate simulation waveforms – Google for a Modelsim tutorial.

   Please be aware that use Modesim you will have to go to Tools → Options → EDA Tools and ensure that the Modelsim path is correct.

8. Write a report answering the questions of (6) above, a state diagram of your state machine, a 'Current state – Next state table', your VHDL code and an image of the LED's in operation. Submit the report to Sunlearn before the deadline.