ECE 3829: Advanced Digital System Design with FPGAs - C Term 2018

Design of a MicroBlaze MCS based System
Report due at beginning of class Friday March 2nd

Implement a MicroBlaze Micro Controller System. This lab (and report) can be completed individually or with a lab partner – it will be worth 30% of your lab grade. This lab demonstrates the use of an embedded micro-controller implemented with in the FPGA and integrated with custom logic.

Lab Sign-off: During a lab section or posted help session or office hour, demonstrate what you have working to a member of the course staff for sign-off. You should get sign-offs as your go along instead of waiting to get them all at once. Please BRING your VERILOG and C code listings for the TAs to check and sign (they will return these to you to hand in with the report). The lab report is due in class on Friday March 2nd.

Note: Due to the end of term it is recommended that you complete this lab and have it signed off as soon as possible. *No lab sign-offs or lab reports will be accepted after Friday March 2nd.*

Description

This lab will introduce you to implementing a FPGA design with an embedded microprocessor – the Xilinx MicroBlaze MCS. We will use the MicroBlaze to interface to the switches and the leds on the Basys3 board as well as interfacing to your custom IP (the seven segment display).

***Background*:**

- Read the DS685 data sheet.
- Complete the MicroBlaze MCS Tutorial including adding the UART, switches and LEDs
    - Set the MicroBlaze memory size to 32KB (this will be implemented using Artix-7 block RAM, BRAM)
    - Make the UART BAUD 9600 so it will work with the lab PCs
- Use a MMCM to generate 100MHz for the Microblaze MCS and other sequential logic
- Use your seven-segment display from previous projects (you should not need to modify the module)

Part 1*: *Hardware additions to MicroBlaze MCS

- Connect 4 push buttons to a 4-bit GPI port on the Microblaze so you can read their value in software.
- Connect the 16 LEDs to a 16-bit GPO port so you can control them in software.
- Connect the lower 12-bits of your seven segment display to a new 12-bit GPO port on the Microblaze so that it can be controlled by software.

- Connect the upper 4-bits of the display module to a hardware counter (not software) that counts from ‘0’ to ‘F’ at a rate of 1Hz.

Part 2*: *Software additions to MicroBlaze MCS - Modify the C program from the tutorial to:

- In response to a push-button press, display a message on the PC with you name(s).
- If a second button is pressed display a message on the PC with current count value from the 4-bit hardware counter.
- If another push-button is pressed, prompt the user to enter a three digit hex number on the PC keyboard (e.g. ‘6B0’) and display this number on the bottom three seven segment displays.
- If another push-button is pressed turn on the left-most discrete LED. Each time the same push-button is pressed turn on the next led one place to the right until all the LEDs are on.
- Your software should allow for the previous described actions to be carried out in any order.

Notes:

Prepare a sign-off sheet and demonstrate your system during one of your lab sessions or during posted course office hours before the deadline. ***BOTH TEAM MEMBERS MUST BE PRESENT FOR ALL SIGN-OFFS!*** Have your printed Verilog and C source files ready so they can be checked during the demo (don’t forget to header blocks for your Verilog modules including names, description, and comments).

Write a *short* report including:

- an overview of your design with a good block diagram,
- a summary of the FPGA resources used (flip-flops, LUTs, BRAM) - both as a number and also as a percentage of total FPGA resources,
- a listing and explanation of synthesis warnings (ignore the Microblaze related warnings),
- screen captures of PC terminal output,
- a conclusion describing any problems or issues you had, and lessons learned.
- Include your signoff sheet and your source files in an appendix.

Grading Guidelines

- [55 pts] Implementation
    - Design works on board and meets requirements
- [20 pts] Source Code – Verilog and C code in Appendix
    - Code style and comments (well-commented and tab-indented code!)
    - Good modular design
    - No latches or other synthesis problems
- [25 pts] Short Lab Report
    - [15 pts] Brief introduction, and general overview of design along with good block diagram and screen captures of PC terminal showing output along with explanation and descriptions.
    - [5 pts] FPGA Resource usage and listing and explanation of warning messages (don't copy all the Xilinx reports – just the relevant sections)
    - [5 pts] Conclusions
        - Problems faced in implementation
        - Solutions used to solve problems
        - Lessons learned from the project
        - Suggestions for further improvements and extensions

ECE 3829: Lab 4 sign-off sheet

Name: _____ ECE Box #: _____

Name: _____ ECE Box #: _____

Both partners MUST be present at all sign-offs!

1Hz counter (0 to F) on seven segment display _____

Hexidecimal number entered on keyboard is displayed on seven segment display _____

Name(s) displayed on PC _____

Hardware count displayed on PC _____

LEDs turned on _____

All actions can be continuously carried out _____