Lecture 7 - Part 2
Getting up to speed with DE1-SoC board: HPS+FPGA Projects

Cristinel Ababei
Dept. of Electrical and Computer Engineering, Marquette University

1. Objective

The objective of this hands-on lecture-tutorial is to continue to learn about how to use the DE1-SoC board to create projects that are HPS+FPGA systems. This will be done through a new example, a bit more complex than those from the first part.

***NOTES:** This is a time-consuming assignment; so, please make sure to allocate plenty of time to it. Start early!*

2. Prerequisites

I assume you have successfully done the examples from Part 1 of this series of hands-on tutorials. The examples are numbered in continuation of the numbering from the previous part. Because in Part 1, we looked at three examples, the example in this Part 2 is numbered as Example #4.

There is one more thing to do (very important):
Follow the steps inside: C:\intelFPGA_lite\22.1\nios2eds\bin\README
To complete the installation of QUARTUS ECLIPSE.

3. Examples

EXAMPLE #4: FPGA+HPS+NIOS II system that uses LEDs, switches, and 7-segment display

Plan A: The main way this example is intended to be done is to follow steps from the tutorial Guide:
"SoC-FPGA Design Guide [DE1-SoC Edition].pdf":
https://github.com/sahandKashani/SoC-FPGA-Design-Guide/tree/master/DE1_SoC/SoC-FPGA%20Design%20Guide

The above Guide is an excellent tutorial written by Sahand Kashani-Akhavan and René Beuchat, and is perhaps one of the **best publicly available online HPS+FPGA examples.** Sahand and René did a terrific job at compressing the huge Cyclone V Hard Processor System Technical Reference Manual:
https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-v/cv_5v4.pdf
to something readable in a shorter time.

To do this example, you must read and do the steps from the following sections in the Guide (though, you may want to read the whole guide; it's good stuff):

- Section 8
- Section 9
- Section 10
- Section 13.8 dealing with creating the application, which you can compile as we did in Example #3 from previous Part 1 tutorial. No need to do the steps involving the debugger here. Just take the binary "de1_soc" and copy it to the microSD card, then run it from within a Putty terminal.

Summary of the example: *A system is created in which both the HPS and FPGA can do some computation simultaneously. More specifically, we want the following capabilities:*

1) *A Nios II processor on the FPGA must be able to use the 10 LEDs and 10 switches connected to the FPGA portion of the device. The Nios II processor will create a strobing light effect on the 10 LEDs, with the 10 switches acting as enable signals for the corresponding LEDs.*
2) *The Nios II processor will use its SDRAM instead of any form of on-chip memory.*
3) *The HPS must be able to use the LED and button that are directly connected to the HPS PORTION of the device. Pressing the button should toggle the LED.*
4) *The HPS must be able to use 2 buttons and the six 7-segment displays connected to the FPGA PORTION of the device. The HPS will increment and decrement a counter that will be shown on the 7-segment displays. Pressing the first button should invert the counting direction, and pushing the second button should reset the counter to 0.*

Plan B: In the .zip archive provided with this lecture tutorial, I have included the complete example after I did all the above steps from those sections. You can find the entire worked-out project inside **hps_fpga_moodle_epfl_ex1/**, provided on the class website. You can use my files if you have issues that you cannot resolve when you try to doo it on your own following the Guide.

NOTES:

- The demo project from Sahand Kashani-Akhavan and René Beuchat can be used as a **template project** to start from and then change to implement additional desired functionality for new projects. It is really nice as it combines HPS, NIOS II (softcore), and FPGA. Also, the top-level design is described in VHDL, **hw/hdl/DE1_SoC_top_level.vhd**, which is a plus compared to the examples from Terasic.
- A nice presentation of the general design flow for HPS+FPGA systems is presented in the following lab; you may want to briefly read it to re-enforce some of the steps that are involved: https://www.ee.ryerson.ca/%7Ecourses/coe838/labs/lab3.pdf

4. Summary

After going through the above steps, you should hopefully have a bit better idea about HPS+FPGA systems, which involve also NIOS II softcores, in addition to HPS and FPGA fabric.

5. References

[1] Sahand Kashani-Akhavan, SoC FPGA Design Guide, DE1-SoC, https://github.com/sahandKashani/SoC-FPGA-Design-Guide

[2] Systems-on-Chip Design COE838 / EE8221, Ryerson University, https://www.ee.ryerson.ca/%7Ecourses/coe838/announcements.html

[3] ECE 5760, Advanced Microcontroller Design and system-on-chip, Cornell, https://people.ece.cornell.edu/land/courses/ece5760