



Tutorial 3: Processor Based Design in an FPGA

Tutor Notes

1 Introduction

The aim of this tutorial is to give an introduction to the implementation of a processor-based design in an FPGA. This includes the creation of a design consisting of a processor and memory and also the associated software. Future tutorial sessions will use processor-based designs to practice material covered in lectures. The laboratory in the second half of the semester will also use processor cores.

Information on the processor core used in this tutorial and also another one which will be used in other tutorials can be found in the Lecture Notes Part II Appendix B.2 *Selected Components* section. The documents are also available as part of the help facility or on-line:

- CR0115 TSK51x MCU
(http://www.altium.com/files/AltiumDesigner6/LearningGuides/CR0115_TSK51x_MCU.pdf)
- CR0117 TSK80x MCU
(http://www.altium.com/files/AltiumDesigner6/LearningGuides/CR0117_TSK80x_MCU.pdf)

There is only one task to complete for this tutorial. Spend some time reading through the information on the processor cores to understand the function of all it's inputs and outputs.

Rather than just completing the design as described, make sure that the students understand what they are doing. It is important that they spend some time reading the documentation. Unfortunately, because of the Easter vacation and public holidays, not much material has been covered in lectures before completing this tutorial.

2 Main Task

Complete the attached *Implementing a simple processor-based design in an FPGA* tutorial provided by Altium. This tutorial is also available as part of the help facility or on-line:

- TU0118 Implementing a simple processor-based design in an FPGA
(http://www.altium.com/files/AltiumDesigner6/LearningGuides/TU0118_Implementing_a_Simple_Processor-Based_Design_in_an_FPGA.pdf)

The tutorial from Altium is not attached to the tutor notes. Refer to the work sheet instead.

Use the Xilinx FPGA rather than the Altera FPGA. Try not to rush through the task. Consider both the design and also the methodology used in the design tools. You may also want to modify the sample code to perform other tasks than just counting.

If students seem to be completing the task very fast, you might want to just sit down with each group and discuss what they have done. Interesting points of discussion are:

- *Why has RAM been connected to the processor rather than ROM.*
- *Understand the way that hard and soft JTAG links are used.*
- *Consider how software is downloaded to the RAM.*
- *How to use the debugger to step through program execution and to view registers and the address space.*

You may also think of other thing to talk with the students about. Anything that makes them think about what they are doing. I will also let you know if there are other tasks that the students can do. If they have not finished all of the tasks from the previous tutorial (such as simulating the bi-direction octal register) they can work on that.