CORALU Group

Excalibur Nios CPU board

Application note

New Wannasatit (nw@ualberta.ca)

Ruud Benjaminsen (benjamins@ee.ualberta.ca)

Rev 1.1 **December, 2001**

Table of Contents:

Check out http://www.altera.com for all your latest Nios developments

Referenced Documents:

Nios Embedded Processor Development Board Manual (mnl_nios.pdf) – 1.1 March 2001
Nios Embedded Processor Hardware Tutorial (nios_tutorial.pdf) – 1.0 November 2000

1. Introduction to the Excalibur NIOS board

The Altera Nios board is similar to the old UP1 board in ways, but adds a new dimension to the design by including embedded processor core (Nios). Nios is a RISC based processor. The added processor core allows for flexible system-on-a-chip (SOPC) designs, providing ample opportunity for reprogramming and reconfiguration. Combined with user defined logic it can be configured for a wide range of applications.

This application note gives a brief overview of some of Nios' more interesting features, and gives the basic ways to instantiate it in the Quartus II software suite.

2. Features

Some of the main features of the Excalibur NIOS board will be outlined below:

The Nios development board supports an APEX20K200E, with a maximum of 526,000 system gates (typically 211,000). The Nios CPU normally occupies around 30% of the logic available, offering a lot of free unused logic for other PLD purposes. It further offers two kinds of on-board memory systems, a 1MB (512K x 16-bit) flash memory and 2 SRAM chips (64K x 16-bit each). It is notable that a pre-configured Nios CPU setup can be stored in flash memory, to be downloaded to the APEX device on power-up. For memory hungry developers there is also another 144-pin SODIMM connection socket (for use with 64-bit wide SDRAM modules) available.

Some more notable features:

- 1MB (512K x 16-bit) flash memory
- 256KB (two 64K x 16-bit) SRAM memory
- 144-pin SODIMM connection socket
- 3.3V and 5V expansion / prototype headers (40 user I/O pins each)
- A standard DB-9 RS-232 serial connector
- 2 IEEE-1386 compliant PCI mezzanine connectors
- JTAG interface connector for Byteblaster and Masterblaster cables
- A two-digit 7-segment display
- 8-Dip switch board, 4 general purpose push-buttons and 2 LEDs

3. Nios CPU instantiation in Quartus II

This is a short summary of the longer hardware tutorial found in the altera/excalibur sub directory.

For any custom project that involves a custom instantiation of the Nios CPU core there are basically four steps to be taken:

* Create a top-level design entity for your Nios CPU.
* Instantiate a custom Nios CPU system using the MegaWizard Plug-In Manager function of Quartus II.
* Insert your own AHDL/VHDL/Verilog project.
* After compilation, download the programming files to the board using the JTAG connection and customize any configuration options in the flash memory portion of the board..

When creating a new project in Quartus II, you can specify the top-level entity. Normally this is going to be your custom Nios CPU system, let's call it *my_nios_cpu*. After creating a new project, this will be the only entity given in the "Compilation Hierarchies" window of Quartus II. Most designs (like those in the nios_sample_designs directory) start of with inserting the Nios CPU system in a Block Design File. This gives a nice abstraction if you are only concerned with interaction of the Nios CPU system with your own project. After adding this BDF file to your project, you are ready the get your own custom Nios CPU system.
From the MegaWizard Plug-In Manager, select Altera Excalibur Nios as your custom megafunction variation. The wizard will lead you through (reasonable simple) process of instantiating the Nios CPU system. It is here where you can add those nifty features like an UART device or an external SRAM peripheral. Choose those options that are necessary for your project.
After the instantiating the NIOS CPU system, you are ready to insert your own project. Again, the easiest way to do this is to first add Block Design Files for each component of your project, and then include your favourite description language files into a Block Symbol File, which then can be added to the Block Design File.
Lastly, you want of course to test out your design on the board it self. After compiling your project, load the the particular sof file for your project into the programmer. Set the right JTAG switch configuration on the Nios board and download your file to the board. To customize the data in flash memory of the board, you would use the GERMS monitor.