FAQ of
Embedded SoPC Design with Nios II Processor and Verilog Examples

Last updated 8/25/2012

If your question is not answered, please e-mail me at p.chu@csuohio.edu and I'll try to incorporate them into FAQ in future update

General

Q. What is this book for?
A. The book focuses on the development of custom hardware and integration of the hardware with a soft-core processor in an FPGA device. It uses a "learning-by-doing" approach and illustrates the design and development process by a series of hands-on experiments and projects.

Q. What is this book **not** for?
A. The book is not intended to provide comprehensive coverage of the following:

- Verilog. Verilog is complex language. The book only uses and discusses a small, synthesizable subset of Verilog.
- Altera software/device. The book includes a general overview of FPGA and several tutorials of Altera software tools, just providing enough information for the experiments and projects. Detailed information can be found in Altera data sheets and manuals.
- Testbench. The book includes two simple testbench templates, which can be used for simple combinational and sequential circuits, and dose not provide detailed discussion on testbench. Many testing circuits are included to physically verify the module's operation.

Q. What is the difference between the book *Embedded SoPC Design with Nios II Processor and Verilog Examples* and the book *FPGA Prototyping by Verilog Examples*?
A. The latter book focuses more on hardware and covers a stand-alone 8-bit microcontroller (PicoBlaze). The former emphasizes the hardware and software co-design and integrates the 32-bit processor in the development flow. Also, the latter uses Xilinx boards but the former uses Altera boards. The first part (about 25%) of the two books is the same.

Q. What is the difference between the book *Embedded SoPC Design with Nios II Processor and VHDL Examples* and the book *Embedded SoPC Design with Nios II Processor and Verilog Examples*?
A. The two books cover the same experiments and examples. One is using the VHDL language and the other is using the Verilog language.

Q. Is there a quick way to see the demonstration?
A. The FPGA configuration file and several software image files for the DE1 and DE2 board are generated and can be downloaded. They are stored in the de1_build and de2_build directories, respectively. They can be used to set up the demonstration with a simple command-line window. The instruction is in the embedded pdf file.

Q. Can I use the board without involving HDL and synthesis?
A. A comprehensive Nios II system that includes book's main IP cores is developed in Section 17.10. The FPGA configuration .sof file (to program the FPGA device) and the .sopcinfo file (to generate BSP) for this system are included in the de1_build and de2_build directories. They can be used for software development without invoking Quartus II at all. However, since the main purpose of SoPC is to develop custom hardware and software, it is not a good idea to use an FPGA board just for software development.

Prototyping Board

Q. Which prototyping board can be used?
A. The book is intended to be used with Altera *DE series* boards. If the DE1 board is used, the codes and pin-assignment files can be downloaded from the book's companion website and used directly. Minor modifications of HDL may be needed for the DE2 board.

Q. What kind of modifications is needed for the DE2 board?
A. A revised VGA controller core is required for the DE2's video DAC chip and several miscellaneous files must be updated. All the revised files are included in the chu_ip_vlog_de2 directory and can be downloaded.

Q. Why are the DE series boards chosen?
A. The DE series boards contain a useful collection of memory modules and I/O peripherals, including SRAM, SDRAM, VGA port, audio codec, PS2 port, and SD card slot. They are quite diversified and practical but not exceedingly complex (such as DDR3 memory and HDMI). For novice learners, these boards are sophisticated enough but not overwhelming.

Q. Why is Altera chosen?
A. Because of the availability of the DE series boards.

Q. How about other Altera boards?
A. An IP core and code can be used as long as a board has a similar I/O peripheral configuration.

Q. How about Xilinx boards?
A. Xilinx has its own MicroBlaze soft-core processor and its own tool-chain. The design tools discussed in the book thus cannot be applied. However, the HDL and C codes of the IP cores are mostly device independent. The needed modifications are discussed in the IP cores section.

Altera software

Q. Which version of Altera software is used in the book?
A. Altera Quartus II Web Edition v10 sp1 and Altera Nios II EDS v10 sp1.

Q. Why is the obsolete version of Altera software used in the book?
A. Publishing a book like this is a tedious and time-consuming process. Even after the draft is completed, it takes more than half year to do the copy-editing, correcting, publishing, distribution etc. On the other hand, since the soft-core processor and hardware-software co-design are still a new and evolving area, Altera software is updated in a regular basis (new version or service pack). It is impossible to use the most updated version in the book.

Q. Can I use the book to learn Altera software?
A. No. The book intends to introduce development flow but not a specific suite of software tools. The relevant Altera software tutorials in the book just provide enough information to "jump-start" the process. After learning the basics, you should consult Altera documentation for the detailed information.

Q. Are the older versions of Altera software available?
A. Yes. Altera has an archive that contains older versions. The link is
https://www.altera.com/download/archives/arc-index.jsp
The v10 version can be downloaded. Of course, one disadvantage of using an older version is that it does not include the most up-to-date bug fixes and software driver.

IP Cores

Q. How "general" are the book's IP cores?
A. The HDL codes and C driver codes are basically provide the "bare bone" functionality. They should work for "normal" condition but are not robust or configurable. However, despite of simplicity, the design and coding follow disciplined practice and the HDL and C codes can be easily expanded to meet additional requirements or used as the basis for a full-ledged IP core.

Q. Why are only the "bare bone" IP cores developed in the book?
A. The development of an SoPC core covers a wide range of subjects, from gate-level timing to the application file format. A full-fledge IP core needs to provide good performance and be flexible, configurable, and robust. Consider the SD card core. It involves the SPI bus standard, the SD card standard, and the FAT file format. There are many features and variations of these standards and their specifications involve hundred pages of documentation. It is not a simple task to develop an IP core that exercises all features of the SD card, supports the maximal data transfer rate, and checks all possible error conditions. It requires a separate book to cover a single full-fledged IP core.

Q. What if I just want to "plug in" an IP core and am not interested in its internal design?
A. It is better to use the existing IP cores provided by the Altera University Program. Their software drivers are more robust and integrated with HAL framework. The hardware portion is almost transparent to the user.

Q. Are the HDL codes of the IP cores portable?

A. The HDL codes are generally device neutral and thus portable. However, each IP core includes a top-level "wrapping circuit" to interface Altera Avalon interconnect. Modification of this circuit is needed to accommodate other bus protocols.

Q. Are the C codes of the IP cores portable?

A. The codes use standard ANSI C and libraries. However, the driver routines include three Altera-specific libraries:

- alt_types.h: define a collection of data types with explicitly width
- io.h: define macros for direct memory access
- system.h: define the symbolic name for an IP core's base address

The definitions and macros in these libraries must be replaced for non-Altera platform.