HUNT ENGINEERING
Chestnut Court, Burton Row,
Brent Knoll, Somerset, TA9 4BP, UK
Tel: (+44) (0)1278 760188,
Fax: (+44) (0)1278 760199,
Email: sales@hunteng.demon.co.uk
URL: http://www.hunteng.co.uk

Developing FPGA Applications – The Development Flow

Rev 1.0 R.Weir 8-02-2001

Introduction

The HERON-FPGA family is ideal for many of the building blocks of digital communications. Providing large, easily-programmed gate arrays, often combined with interface elements like ADC or DACs, they can be used to implement many system components.

There are several toolsets which can be used to generate the code for a HERON-FPGA module. This paper introduces the main options.

The Development Process

The typical development cycle is as follows:

1. Develop the FPGA code using one of the toolsets described later. Interface to the hardware using the libraries provided for your FPGA module – we do not recommend you attempt to modify these libraries!
2. Download this FPGA pattern to the hardware using the Hunt Engineering tools
3. Load any DSP code and test the system.

As can be seen, if your system includes both DSP processor and FPGA we recommend treating development as two separate stages. Develop the FPGA and make sure it's right; then switch to developing the DSP processor code. This can be done iteratively – develop a basic FPGA, then a basic DSP program, then return to enhance the FPGA code…

The Toolsets

Three main toolsets are available. We recommend the first and have little or no experience of the others; however, this should not preclude you from using them if they suit your requirements.

1. Xilinx Foundation series tools. This is the option we use. It is available in different levels; the basic level supports only the small FPGAs, but at reduced cost; while the "top of the range" pack is expensive, and covers all current Xilinx devices. You can purchase a system that will support your current project, or go for a system that will cover any FPGA you may use in the future.
2. Xilinx WebPack. This is a freely downloadable toolset. It supports a small subset of the FPGA modules we offer – mainly those equipped with the Spartan-II array. It has reduced functionality, but it's free.
3. The Xilinx Simulink System Generator. This allows the developer to build his application in The MathWorks' popular Simulink architectural design package. This allows some link back to the Matlab DSP design package.

Each has strengths and weaknesses, as we'll outline in the following sections.

Foundation

Foundation is the toolset we use and recommend. It is not the cheapest option (WebPack is free), but it does provide all the facilities that you are likely to need. It's available in a range of price/functionality options, ranging from basic support to a full system with simulators and support for all Xilinx FPGAs.

Designs are entered using VHDL or schematic capture, and the package includes the "core generator". This is an extremely useful utility that can build you a FIR or FFT processing engine (many more blocks are available) – leaving you to drop in coefficients and data to get a working system. It also allows you to use other IP cores in your design.

A major consideration is that the full version of Foundation supports the FPGAs used on ALL the HERON-FPGA modules. The basic package supports many modules, but will not support the latest arrays, or the largest size options. For these, the full version of Foundation is the only choice.

As with the other tools, Foundation is continually being extended. Check the Xilinx website for details on the current software release.

WebPack

WebPack is a free toolset available from the web. It is far more basic than Foundation, but can be used for basic designs in a subset of the HERON-FPGA Range.

WebPack only supports a limited range of devices – currently the Virtex-V300E and the SPARTAN-II family. If you intend to use an FPGA module with any other device (e.g. Virtex, all Virtex-II) you must buy the Foundation package.

Designs can be entered as HDL or, with the right combination of downloads, in schematic form. However, the tools are not as good as in Foundation. The core generator is also not available, so many of the easy examples we show will not work.

We do not recommend the use of WebPack for signal processing applications. Only use it if you are designing simple logic – like an I/O interface. Also, make sure that the HERON module you are targeting uses an array that is supported by WebPack.

Simulink System Generator

This is really a link between Simulink and the Foundation package. It cannot generate FPGA programs on its own. To use it you must own Simulink (from The MathWorks) and Foundation.

This makes it an expensive option. In its full form however it has many benefits, especially for the signal processing developer.

You can use Simulink to create FPGA designs and then test them, in a signal-based environment. For example, you can create a filter in Matlab, import it in Simulink, and simulate it exactly as it will operate in the FPGA – include quantization effects, wordlength effects and so forth. When you dump this design into Foundation, you can be sure that the signal processing part of the FPGA works – a major benefit!

One drawback of Simulink is that to get the performance from the FPGA, you must use a restricted set of blocks. These blocks map directly to cores generated by the Xilinx Foundation core generator. This is not an option for existing designs, and not ideal for migrating Matlab designs to FPGA.

Because it sits on top of Foundation, Simulink inherits all the benefits of that package – like support for all arrays used in HERON modules. As Simulink generates blocks that you will integrate into a Foundation design, you also have the option of adding extra functionality using schematic or HDL-style design entry.

Summary

Simple - we recommend Foundation. You can extend this with Simulink if required, but on its own it gives a powerful set of tools for supporting the HERON-FPGA range.

If you are working on a tight budget and building a very simple system, or have a very in-depth knowledge of the systems you want to create, then WebPack may be adequate. It is available as a free download, so you can download it and try it – but do not expect all our examples to work, as we use the core generator!

	Foundation	WebPack	Simulink
VHDL	Yes	Yes	Via Foundation
Core Generator	Yes	No	Via Foundation
Schematic Entry	Yes	Sort of	Via Foundation
Cost	Moderate	Free	High (Foundation)
Arrays Supported	All (affects cost)	Restricted to Spartan-II	All
Signal-Level Simulation	No	No	Yes
System-Level Capability	No	No	Yes
Filter Design Tools	No	No	Yes