ISAAC Newton Package Tutorial

Jason Zheng

July 11, 2008

Contents

1 Introduction

This document explains the usage of ISAAC Newton package, including checking out from the ISAAC repository, compiling, simulating, and logic synthesis. Here are the basic stages and their associated tools:

1. Package check-out. Subversion (svn) is used to manage the ISAAC repository and also used for check-out. Subversion is a free tool released under GNU Public License (GPL).
2. Compile and simulation. The default compiler and simulator is Icarus Verilog (iverilog). VCD dump files can be viewed with GTKWave. Both iverilog and GTKWave are freely available under GPL.
3. Logic Synthesis and Place-n-Route. This step requires the Xilinx ISE Foundation or Alliance installation. The ISE can be purchased from Xilinx.
4. Post Place-n-Route Simulation and Verification. This stage again uses iverilog and GTKWave by default.

All the stages above have been programmed into the package using standard make files, which requires GNU Make utility. All the above stages should be carried out in a POSIX-compliant environment, or the tool flow may need additional modifications.

Through this document, a example package called qm_fir is used to demonstrate the tool. This is the main iCore package for ISAAC Newton.

2 Package Checkout

Detailed discussion of svn is outside the scope of this tutorial, so only the basic checkout command is provided. On any POSIX-compliant environment (GNU/Linux, Solaris, etc.), issue the following svn command to check out the qm_fir package to the current directory:

```
$svn co svn+ssh://isaacdev/proj/isaac/repo/newton/qm_fir/trunk
```

The checkout process copies the entire design package to the current directory. Here's a sample directory tree:

```
$ tree
.
|-- Makefile
|-- qm_fir.ucf
|-- tb
|   |-- Makefile
|   |-- QM_FIR_tb.v
|   `-- xov_binary.txt
`-- v
    |-- LUT1.v
    |-- LUT2.v
    |-- LUT3.v
    |-- MAC1.v
    |-- QM.v
    |-- QM_FIR.v
    |-- firdecim.v
    |-- firdecim_m2_n50.v
    |-- firdecim_m5_n15.v
    `-- firdecim_m5_n20.v
```

3 Compile and Simulation

The first step after package checkout is to verify that the RTL design is correct through RTL simulation. As mentioned before, this step requires, by default, Icarus Verilog and GTKWave. To compile the design:

```
$ cd tb
$ iverilog QM_FIR_tb ../v/*.v -o qm_fir.o
```

This command will compile the design and generate an object file called qm_fir.o. Now simulation can be run with the following command:

```
$ vvp -l rtl.log -v qm_fir.o
```

A log file rtl.log is created after the simulation, as well as a VCD dump file called dump.vcd. The VCD dump file can be viewed by GTKWave by issuing the following command:

```
$ gtkwave dump.vcd
```

The compile and simulation commands have been captured in the Makefile that accompanies the tb directory. So the first two steps can be accomplished by issuing the following command:

```
$ cd tb
$ make rtl
```

4 Synthesis, Place-n-Route

This step compiles the RTL design and translates it to a format that can be used to implement a Xilinx FPGA. To make the process consistent, all the steps in logic synthesis and place-n-route have been captured in the Makefile in the root directory. The Makefile calls the ISE backend tool xflow, which in turn calls other backend tools such as xst, ncdbuild, par, and trce. More detailed understanding of these tools can be obtained from reading the official Xilinx documentation on the backend development tools. For purpose of this tutorial, just issue the following command:

```
$ make all
```

The process will probably continue for 10-30 minutes before the design is fully placed and routed. The default target device embedded in the Makefile is xc2v3000-4bf957 (Virtex2-3000, speedgrade -4). To change the target device, one can edit the Makefile in any text editor, and replace xc2v3000-4bf957 with a suitable target device name.

A basic static timing analysis is done by the make command. To view the basic static timing report, open the qm_fir.twr file under the (newly created) work directory:

```
$ view work/qm_fir.twr
```

Additional timing analysis can be done with the Xilinx trce tool. Here's an exmample of generating a static timing report with a different speedgrade (6) and 10 expanded paths:

```
$ trce -s 6 -v 10 qm_fir.ncd
$ view qm_fir.twr
```

5 Post Place-n-Route Simulation

This step is very similar to Section 3. All the necessary commands are again captured in the Makefile under the tb directory. To run the post place-n-route simulation with back-annotated timing, simply issue the following command:

```
$ cd tb
$ make gate
```