

Organization and Usage of Source Files of “FPGA Prototyping Using VHDL Examples: MCS edition”

Last updated on November 10, 2017

1 File Organization

The folders and files in the zip file are organized as follows:

- `chapter_listing`: The folder contains the listings of VHDL codes and C++ codes. The subfolders are arranged chapter by chapter in chronological order.
- `hdl`: The folder contains the VHDL files to construct an FPro system. The subfolders are organized as follows:
 - `sys`: system level components
 - `bridge`: `chu_io_map.vhd` and `bridge`
 - `mcs_v2017`: `cpu.xci`, Xilinx IP description file for MicroBlaze MCS generated in Vivado 2017.2
 - `subsys`: various mmio and video subsystems
 - `top`: various top-level FPro system
 - `mmio`: I/O cores
 - `mmio_support`: supporting files, including mmio controller and FIFO buffer, for the mmio subsystem
 - `mmio_basic`: `gpi`, `gpo`, and timer cores
 - `others`: individual i/o cores
 - `video`: video cores
 - `video_support`: supporting files, such as video controller and clock management, for the video subsystem
 - `others`: individual video cores
- `cpp`: The folder contains the C++ and header files for software development. The subfolders are organized as follows:
 - `app`: various application (main) program
 - `drv`:
 - 4 basic definition/initialization files:
`chu_io_map.h`, `chu_io_rw.h`, `chu_init.h`, `chu_init.cpp`
 - `others`: drivers for individual cores
- `constraint`: The folder contains the Xilinx constraint (`.xdc`) files.

2 Usage

The source files are best used as follows:

- Testing of prototyping circuits in Part I
 - copy the needed file from the `chapter_listing` folder.
- Construction of an FPro system
 - Select a top-level FPro file from the `sys/top` folder.
 - Select a mmio subsystem file and a video subsystem file (if needed) from the `sys/subsys` folder.
 - Add the `sys/mcs_v2017` and `sys/bridge` folders.

- Add the needed mmio and video core folders. For the sampler system and the daisy system, just add the mmio and video subfolders, respectively.
- Add a constraint file from the constraint folder.
- Software development
 - Import the 4 basic definition/initialization files from the cpp/drv folder.
 - Select and import the needed driver files (both .h and .cpp files) from the cpp/drv folder.
 - Select and import a main program from the cpp/app folder.

3 Pre-built FPro system

An alternative to constructing an FPro system from source files is to import a pre-built Vivado project. Three compressed project files incorporate the three FPro systems used in Part II, III, and IV of the book: can be found on the companion website:

- vanilla.xpr.zip: vanilla FPro system of Part II
- sampler.xpr.zip: sampler FPro system of Part III
- daisy.xpr.zip: vanilla-daisy FPro system of Part IV

They can be found on the companion website.

To use a project file:

- Unzip the file. The hdl files and the constraint file can be found in the corresponding subfolders.
- In Vivado, select File →Open Project...” and then navigate to the uncompressed folder.
- The HDL and constraint files can be found in proj_name.srscs folder.

The project files are constructed under Vivado 2017.2. They may need to be updated for new versions.

4 Compatibility

Most HDL and C++ codes in the book are portable. However, four Xilinx proprietary IP cores are used:

- MicroBlaze MCS
- MMCM clock management
- XADC ADC converter
- BRAM based dual-clock fifo buffer

These cores are obtained/generated in Vivado v2017.2 and may need to be revised or updated in other versions. They can be re-generated following the tutorial in Appendix A.4.

The codes are developed around the Digilent Nexys 4 DDR boards. However, other Xilinx boards can be used as well. The revision is mainly based on the available I/O peripherals on the board. The basic procedure is

- Determine the needed MMIO and video cores and assigned a slot number for each core.
- Revise the relevant files as follows:
 - Update the system clock rate and slot assignment in chu_io_map.vhd and chu_io_map.h.
 - Develop a new mmio subsystem and a video subsystem.
 - Develop a new top-level FPro subsystem.
 - Update the constraint (.xdc) file for clock specification and pin assignment.