Make FPro

Keith Schubert

September 28, 2023

1 Vivado

Your first step will be to make a Vivado RTL project. **DO NOT** check the Vitis extension. This makes a dynamic project, and you want a static project.

Here are the basic steps. You will need the fpga_mcs_sv_src folder (or the Nexsys4 vanilla folder) to add the following.

1. Add the Pong Chu's Nexsys4DDR config file (in config folder)
2. Add from bridge in sys:
   (a) chu_iomap.sv
   (b) chu_mcs_bridge.sv
3. Add from subsys in sys:
   (a) mmio_sys_vanilla.sv
4. Add from top in sys:
   (a) mcs_top_vanilla.sv
5. Add from mmio_support in mmio:
   (a) chu_mmio_controller.sv
   (b) in fifo:
      i. fifo.sv
      ii. fifo_ctrl.sv
      iii. reg_file.sv
6. Add from mmio_basic in mmio:
   (a) chu_gpi.sv
   (b) chu_gpo.sv
   (c) chu_timer.sv
7. Add from uart in mmio:
   (a) baud_gen.sv
   (b) chu_uart.sv
   (c) uart.sv
   (d) uart_rx.sv
   (e) uart_tx.sv

At this point you need to add the cpu. Don't use his, it is from 2017, and it is so easy to get a fresh one.

1. From your flow navigator window, select IP Catalog.
2. Search for "embedded" and then in processors, select " MicroBlaze MCS".
3. A dialog will appear. Change the "Component Name" field to "cpu" then click the "MCS" tab.
4. In MCS set the memory size to 128KB and select the "Enable IO Bus" checkbox. Select Ok.

Now you can generate the bitstream. When it generates you need to export the hardware.

1. From the file menu's export submenu, select "Export Hardware".
2. A dialog appears and you must select next.
3. Select include bitstream, then next.
4. Name the xsa whatever you want, and export somewhere you will find it.
5. Press finish.

From the tools menu select "Launch Vitis IDE".

2 Vitis

Note where your workspace is (files you need will be saved here).

First you must make a platform project (File, New, Platform Project...). This is the hardware description and system that your app will be built on. Give it a name you will remember and select next. From the create a new platform from hardware (XSA), select browse and navigate to the xsa you exported. Finish up. When it is initially built it will be out of date, just build it (the hammer on the toolbar), should be fine now.

Now you want to make an application project (File, New, Application Project...). Your platform project should be available int the repository. Select it and click next. Name your project, make sure your system is selected, and press finish. Now add the following files from fpga_mcs_sv_src

1. Add from drv in cpp:
   (a) chu_init.cpp
   (b) chu_init.h
   (c) chu_io_map.h
   (d) chu_io_rw.h
   (e) gpio_cores.cpp
   (f) gpio_cores.h
   (g) timer_core.cpp
   (h) timer_core.h
   (i) uart_core.cpp
   (j) uart_core.h
2. Add from app in cpp:
   (a) main_vanilla_test.cpp

Build it! (hammer button). It should produce an "elf" file.

If so we can program. Attach your board.

In Vitis, select program device from the Xilinx menu. Verify the Project, bitstream, and BMM/MMI are correct. Change the "bootloop" on microblaze_I to "browse..." and double click ELF/Mem file to select your elf file (in the workspace's project's debug folder). Press program. Enjoy!