ECEN 5593 Advanced Computer Architecture

Implementation Report

October 2021

Team Members: Pranav Bharadwaj, Malcolm McKellips

**Project Implementation:**

The design flow of implementing the SweRVolf EH1 core on the Digilent Nexys A7 board is as follows:

1. The work environment in a virtual machine running Ubuntu 18.04 on VirtualBox. The minimum memory requirements for the same is 4096 Mb, and a hard drive space of 100 Gb.
2. The softcore processor is synthesized using Xilinx Vivado onto the board (uses .bit files generated post compilation). Since the board communicates via UART over microUSB, only one host software can communicate with the board at any given time. For software control over the processor’s operation, we require the use of Visual Studio (along with Whisper, which is an instruction set simulator for the SweRVolf RISC-V core).
3. Verilator and GTKWave are chosen as the hardware simulator and waveform viewer respectively. These are open source and can be easily modified for the SweRVolf core.

Building the SweRVolf core, generating the bitstream for it and uploading it to the board is the first major milestone to be achieved. The SweRVolf EH1 core is composed of the following modules that need to be compiled:

1. CPU (EH1 core complex)
2. Interconnect (AXI, AXI2WB interconnect)
3. Peripherals (System controller, GPIO)

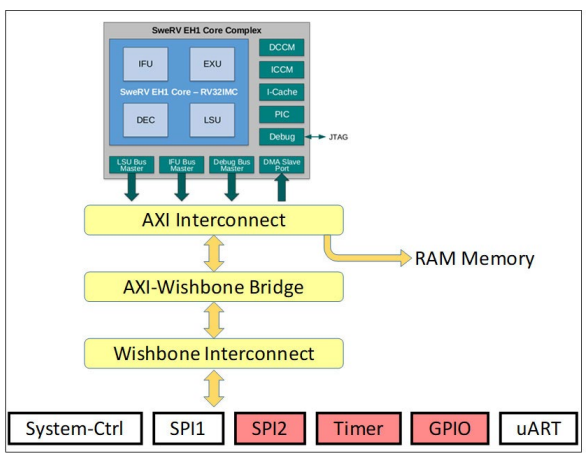


Fig 1: SweRVolf EH1 core complex diagram

Xilinx’s Vivado is used to create an RTL project that generates the bitstream required for the softcore to be uploaded to the board. Since the EH1 core complex does not have on-board memory, we use the 128 MB RAM present on the Nexys A7 board for the same. This requires the memory initialization files for the same.

Further, we add the Verilog modules to initialize the on-board registers, write and read buses and memory controllers for the litedram module on the board. We build a high-level design using schematic entry as the hard IP for all modules are readily available. The wrapper for the SweRV core is used as the master component, with the memory controller and GPIO components following it. The BootROM for the softcore processor is included as a block entry as well. This handles the memory map for the processor upon start up. The clock input for the processor is provided via software, through Visual Studio. Hence, the clock pin from the block diagram is made external for use. The final block design consists of the following modules:

1. SweRV core complex
2. Interconnect wrapper
3. BootROM module
4. GPIO module
5. System controller
6. Bidirectional GPIO module

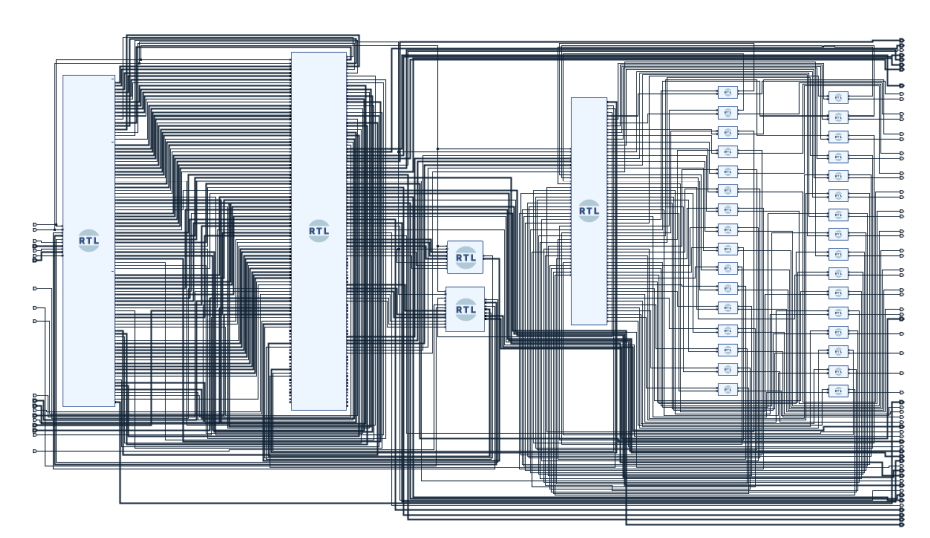


Fig 2: Finalized Schematic entry for the block design