Local FPGA Guide

ACT Lab

1 Prerequisites

For our system we have used the following:

1. OS - Ubuntu 18.04.4 LTS
2. RTL Simulation Tool - Xilinx Vivado 2020.2 and Xilinx Vitis 2020.2

2 Running GeneSys Compiler

3 Building Genesys on a local FPGA

We use Xilinx FPGA, Xilinx XRT and Vitis to run tests on the FPGA. The host/driver code uses OpenCL. Please make sure that you have OpenCL installed on your machine.

You can use the fpga_framework/host/openCL_version.cpp script to check the OpenCL version on your machine. Before running the below steps, please make sure that the FPGA is active and detectable.
If using Xilinx FPGA compatible with Xilinx XRT, use xbutil query to check the FPGA status.

1. Clone the genesys directory onto your local computer or server.

   $ cd <desired_directory>

   $ git clone https://github.com/actlab-genesys/GeneSys.git
2. Please make note of the FPGA platform you are running on as well as the part name. You will need to make changes to the following locations to take into account the platform and part names. Below, we have included the locations you should check.
   a. genesys-fpga/fpga_framework/Makefile

i. Line 36, 39

b. genesys-fpga/fpga_framework/pack_kernel.tcl
   i. Line 6, 7, 8
c. genesys-fpga/fpga_framework/systolic_fpga.cfg
   i. Line 1

3. Execute the following commands in the fpga_framework directory to build the design:

```
make clean
make pack_kernel
tmux new -s syn
make build_hw
```

The output of this process should be a .xclbin file.

3.2 Building Genesys Hardware Emulation Binary

1. For generating FPGA Emulation binary, you will need to complete steps 1 and 2 in Section 2.1.
2. Edit line 42 genesys-fpga/fpga_framework/Makefile to be hw_emu.
3. Run the following commands:

```
$ make pack_kernel
$ export XCL_EMULATION_MODE=hw_emu
$ make build_hw
```

4 Generating Software Executable

1. Place the test/instruction directory you wish to execute outside of the genesys directory. If you have used the compiler to generate your instructions, this is what you will need to move.
2. Proceed to the following folder: genesys-fpga/fpga_framework/host. Inside of this folder, find the genesys_driver_b2b.h file, and search for the variable REPO_PATH and set it to where the test directory is.
3. Open the genesys_driver_b2b.cpp. Here, you will find the variable binaryFile. Replace the string “*.xclbin”, with the awsxclbin filename.
4. From the fpga_framework directory, run the following command:

```
make build_sw TARGET_DIR=<bin_folder_name> TEST_NAME=<name_of_test>
```

5 Simulating in Xilinx Environment

1. Clone the genesys repo

2. Choose a configuration for your test. For example, if you want to run a test on a 16x16 systolic array, edit genesys_systolic/source/config.vh to ensure that ARRAY_N and ARRAY_M are set to 16. Also make sure that the compiler is configured for a 16x16 systolic array. Once you have compiled the instructions.
3. Open genesys_systolic/testbench/generic_tb_files/systolic_fpga_benchmark_config.vh and add an entry with the respective instruction, input, and output files generated from the compiler. You could use one of the example entries too for sanity check or as an example. Ensure you use absolute paths to avoid errors. Ensure valid paths are given for all the file variables as shown in the template even if it is not applicable to your test. For example, ADD_ONLY test does not need a bias input, nevertheless a valid path is given for the variable.
4. Start Vivado and add all the files in the subdirectories of genesys_systolic/source/ and genesys_systolic/testbench/generic_tb_files/ as sources.
5. You will need to generate the AXI Verification IPs for running simulation. In Vivado, go to IP Catalog and look for AXI Verification IP. Six AXI VIPs need to be created and use the same names as below. This might require a Xilinx Vivado License.

6 Running Genesys on a test

6.1 Running on a local FPGA

1. Connect to your local FPGA and copy the xclbin and executable files to a directory accessible by the FPGA. Run any necessary environment setup scripts.
2. Next, run the following:

   ./<executable> <awsxclbinfile>

   Example: ./FPGA_resnet50_gemm57 aws_systolic_fpga.awsxclbin

   If using one of the example tests, if the test passes, then “TEST PASSED” will be displayed.

6.2 Running on local FPGA using Python Driver

If you use the python driver available under fpga_framework/host_py/host.py , then you do not need to worry about Section 5. This driver’s inputs are your test/instructions and awsxclbin file.

1. In fpga_framework/host_py/host.py, set the following paths - test_name, data_info_file, base_path, genesys_binary
2. Run the following command:

   $ python3 host.py

7 DC Synthesis

1. Move *read_rtl_genesys.tcl* and *run_dc_SIMD_4x4.tcl* to a directory containing the needed rtl files which are listed in *read_rtl_genesys.tcl*
2. Run $ dc_compiler to launch the compilation tool
3. Finally, run $ source run_dc_genesys.tcl