COMPUTER ORGANIZATION AND ARCHITECTURE
COA LAB 2023
GROUP NUMBER: 13

GPGPU-Sim provides a detailed simulation model of contemporary NVIDIA GPUs running CUDA and/or OpenCL workloads.

This documentation is meant for installing GPGPU Simulator and executing the applications using GPGPU-Simulator.

The version of GPGPU-Sim has been tested with a subset of CUDA version 4.2, 5.0, 5.5, 6.0, 7.5, 8.0, 9.0, 9.1, 10, and 11.

The commands and references used in the following stages are based on the usage of ***Ubuntu 20.04.06 LTS and CUDA 11.1 ToolKit only.***

- Install linux in a dual-boot format (prefer not to use virtual-box, as it may crash sometimes) using the above ubuntu version.

- **STEP 1: INSTALLING DEPENDENCIES**

  - First install all the dependencies using the following commands:

  - Install gcc, g++, git using the following commands:
    - sudo apt-get install gcc-7 g++-7 git
    - cd /bin
    - sudo mv gcc gcc-9.4
    - sudo mv g++ g++-9.4
    - sudo mv gcc-7 gcc
    - sudo mv g++-7 g++

  - GPGPU-Sim Dependencies
    - sudo apt-get install build-essential xutils-dev bison zlib1g-dev flex libglu1-mesa-dev

  - GPGPU-Sim documentation dependencies:
    - *sudo apt-get install doxygen graphviz*

  - AerialVision dependencies:
    - *sudo apt-get install python-pmw python-ply python-numpy libpng-dev python3-matplotlib*

  - CUDA SDK dependencies:

- *sudo apt-get install libxi-dev libxmu-dev freeglut3-dev*

- **STEP 2: INSTALLATION OF CUDA TOOLKIT**
  - Now download the CUDA ToolKit using the following commands in order:
    - *wget https://developer.download.nvidia.com/compute/cuda/11.1.0/local_installers/cuda_11.1.0_455.23.05_linux.run*
    - *sudo apt-get install nvidia-cuda-toolkit*
    - *sudo sh cuda_11.1.0_455.23.05_linux.run*
    - During the execution of this code
      - Wait for a while unless a EULA acceptance screen appears.
      - After that ensure that you deselect the NVIDIA driver if your system has a NVIDA Graphic Card of whatever size may be.
      - Now proceed with installation.

- **STEP 3: RUN THE APPLICATION**
  - Now build the GPGPU-Sim
  - Enter the following command in order:
  - Path Variable Changes
    - sudo nano ~/.bashrc
    - ***Add the following two lines at the end the file:***
    - export CUDA_INSTALL_PATH="/usr/local/cuda-11.1"
    - export PATH="/usr/local/cuda-11.1/bin:$PATH"
    - ***Exit the file and run the following command:***
    - source ~/.bashrc
    - Re-start the terminal
  - **Enter the following commands in the terminal present in the root directory:**
    - git clone https://github.com/gpgpu-sim/gpgpu-sim_distribution.git
    - cd gpgpu-sim_distribution/
    - source setup_environment

- make
- mkdir run_code
- cd run_code

- **Now add your test file which you want to run using GPGPU-Sim. (extension of file name should be .cu)**

  Now copy any one your required GPU Hardware into the current working directory using the following command:

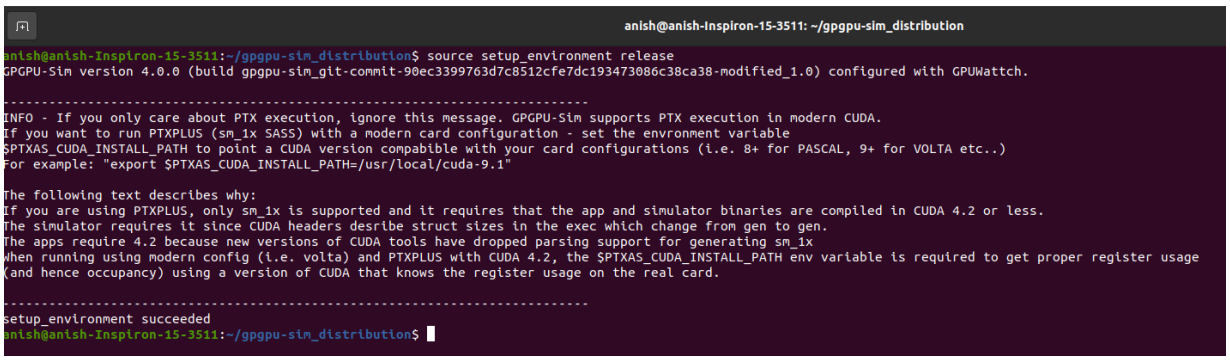
  In the below line go to the directory gpgpusim_distribution/configs/tested-cfgs **and choose the hardware and then write that name in the command.**

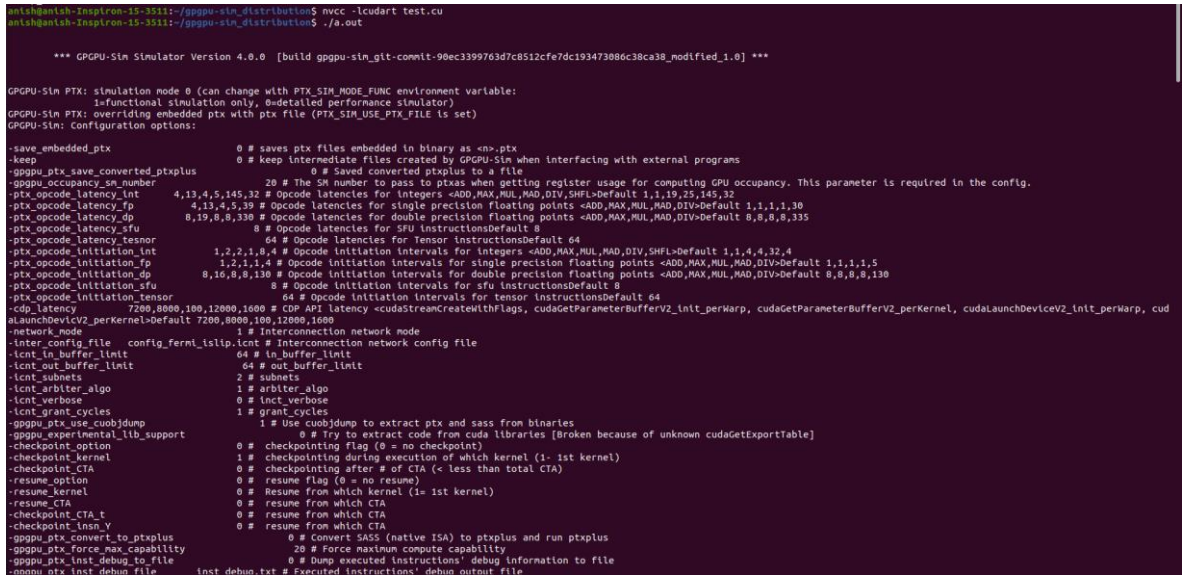
  - cp -r ~/gpgpusim_distribution/configs/tested-cfgs/{**choose your required harware and replace it over here**}/* ./
  - source ~/gpgpu-sim_distribution/setup_environment
  - nvcc -lcudart {**Enter your file_name**}.cu
  - ./a.out

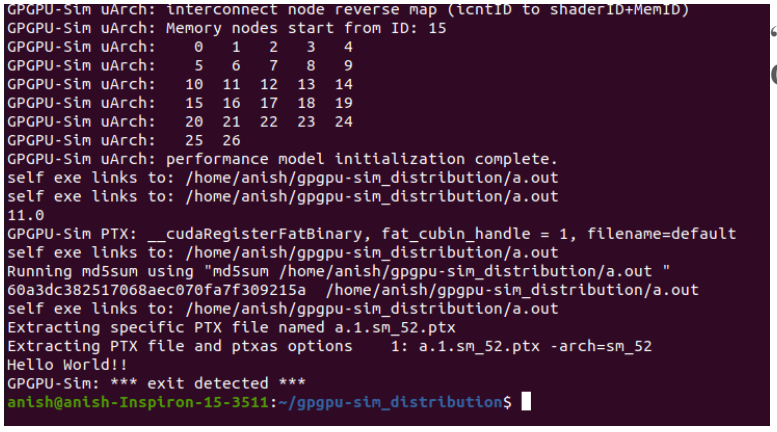
INPUT:

OUTPUT:

- ***The following is the output when the following commands are run in the terminal from the directory where the test file is present:***
  - source ~/gpgpu-sim_distribution/setup_environment
  - nvcc -lcudart *test*.cu
  - ./a.out



- 



- **We can see the “Hello World!” Printed through GPGPU-Sim.**