Generating TV and Launch Pattern Files

Table of contents

Since the cuBB 22-2.2 release, the test vectors are not included in the release package. You must generate the TV files before running cuPHY examples or cuBB end-to-end test.

Note

TV generation is **NOT** supported on ARM because Matlab Compiler SDK doesn't support it yet.

Using Aerial Python mcore Module

No Matlab license is required to generate TV files using Aerial Python mcore module. The cuBB container already has aerial_mcore installed.

To generate the test vectors required for end-to-end testing:

1. Run the following inside the Aerial container. It completes in less than a minute.

```
cd ${cuBB_SDK}/5GModel/aerial_mcore source scripts/setup.sh
./scripts/gen_e2e_ota_tvs.sh ls -lh GPU_test_input/ cp GPU_test_input/*
${cuBB_SDK}/testVectors/
```

Example output of the above:

Note

The cuPhyChEstCoeffs.h5 file can be found in the /opt/nvidia/cuBB/testVectors directory of both the x86 and ARM containers.

aerial@c_aerial_aerial:/opt/nvidia/cuBB/5GModel/aerial_mcore$ source ./scripts/setup.sh [Aerial

```
Python]aerial@c_aerial_aerial:/opt/nvidia/cuBB/5GModel/aerial_mcore$
./scripts/gen_e2e_ota_tvs.sh Finished genCuPhyChEstCoeffs Elapsed
time: 1.166473150253296 seconds [Aerial
Python]aerial@c_aerial_aerial:/opt/nvidia/cuBB/5GModel/aerial_mcore$
ls -lh ../GPU_test_input/ -rw-rw-r– 1 aerial aerial 90K Oct 17 2023
../cuPhyChEstCoeffs.h5
```

2. Copy the output to the testVectors folder.

To generate the all of TV files:

1. Run the following commands inside the Aerial container.

> **Note**
>
> The TV generation may take a few hours on the devkit with the current isocpus parameter setting in the kernel command line. The host must have 64GB memory and 300GB available disk space. Hyperthreading must be enabled.

```
cd ${cuBB_SDK}/5GModel/aerial_mcore/examples source ../scripts/setup.sh
export REGRESSION_MODE=1 time python3 ./example_5GModel_regression.py
allChannels echo $? ls -alF GPU_test_input/ du -h GPU_test_input/
```

Example output is shown below. The “real” time takes less than one hour on a 24 cores x86 host. The “echo $?” shows the exit code of the process, which should be 0. A non-zero exit code indicates a failure.

```
Channel Compliance_Test Error Test_Vector Error Performance_Test Fail ---------
------------------------------------------------------------------ SSB 37 0 42 0 0 0 PDCCH 71 0
80 0 0 0 PDSCH 274 0 286 0 0 0 CSIRS 86 0 87 0 0 0 DLMIX 0 0 1049 0 0 0
PRACH 60 0 60 0 48 0 PUCCH 469 0 469 0 96 0 PUSCH 388 0 398 0 41 0 SRS
125 0 125 0 0 0 ULMIX 0 0 576 0 0 0 BFW 58 0 58 0 0 0 ----------------------------------
```

```
---------------------------------------- Total 1568 0 3230 0 185 0 Total time for runRegression is 2147 seconds Parallel pool using the 'local' profile is shutting down. real 36m51.931s user 585m1.704s sys 10m28.322s
```

To generate the launch pattern for each test case using cubb_scripts:

```
cd $cuBB_SDK cd cubb_scripts python3 auto_lp.py -i ../5GModel/aerial_mcore/examples/GPU_test_input -t launch_pattern_nrSim.yaml
```

Copy the launch pattern and TV files to testVectors repo:

```
cd $cuBB_SDK cp ./5GModel/aerial_mcore/examples/GPU_test_input/*h5 ./testVectors/. cp ./5GModel/aerial_mcore/examples/GPU_test_input/launch_pattern* ./testVectors/multi-cell/.
```

Using Matlab

To generate TV files using Matlab:

1. Run the following command in Matlab:

   ```
   cd('nr_matlab'); startup; [nTC, errCnt] = runRegression({'TestVector'}, {'allChannels'}, 'compact', [0, 1] );
   ```

   All the cuPHY TVs are generated and stored under `nr_matlab/GPU_test_input`.

2. Generate the launch pattern for each test case using cubb_scripts:

   ```
   cd $cuBB_SDK cd cubb_scripts python3 auto_lp.py -i ../5GModel/nr_matlab/GPU_test_input -t launch_pattern_nrSim.yaml
   ```

3. Copy the launch pattern and TV files to testVectors repo.

```
cd $cuBB_SDK cp ./5GModel/nr_matlab/GPU_test_input/TVnr_* ./testVectors/. cp
./5GModel/nr_matlab/GPU_test_input/launch_pattern* ./testVectors/multi-cell/.
```