

HANDS-ON — CUDA SDK - BASIC CONCEPTS

Siegfried Höfinger

VSC Research Center, TU Wien

October 22, 2023

→ <https://tinyurl.com/cudafordummies/ii/ho3/notes-ho3.pdf>

CUDA 4 DUMMIES — OCT 24-25, 2023

HANDS-ON — CUDA SDK - BASIC CONCEPTS

Exercise

- Q1)** *A simple example to probe tensor core operation is given in the SDK — 3_CUDA_Features/cudaTensorCoreGemm. Create your own space (maybe in a separate parallel dir my_cudaTensorCoreGemm), go there, copy over the sources, examine the *.cu, compile it and run it. Can we check whether the GPU is really doing something ? Could we make the compute process on the GPU a bit “heavier” so that we can monitor it better ?*

15 min

- A1)** *Go into the SDK and copy over the mentioned directory,
cd wherever_the_SDK_may_be/Samples/3_CUDA_Features
cp -r ./cudaTensorCoreGemm ./my_cudaTensorCoreGemm
cd ./my_cudaTensorCoreGemm
make
./cudaTensorCoreGemm
In a second xterm run
watch -n 0.1 nvidia-smi
and observe GPU 0/1 activity. If we increase M/N/K_TILES to 1024,
re-compile and re-launch, GPU-Util in nvidia-smi will be monitoring >0%.*

→ https://tinyurl.com/cudaforummies/ii/ho3/cudaTensorCoreGemm_v2.cu

Exercise

Q2) *Consider the SDK sample 0_Introduction/simplePrintf. Again, copy/compile/run it in some private directory. Think about 2 modifications inserting assert() calls in the kernel code, one causing no termination, the other triggering exit/abortion, ideally dependent on some value of threadIdx/blockIdx.*

15 min

A2) *Repeat the copying/compiling of 0_Introduction/simplePrintf then examine the kernel code. A simple non-harmful `assert()` could be `assert(val < 100);` while another one causing exit could be `assert(k < 3);` with*

$$k = (\text{blockIdx.y} * \text{gridDim.x}) + \text{blockIdx.x};$$

→ https://tinyurl.com/cudaforndummies/ii/ho3/simplePrintf_v2.cu

→ https://tinyurl.com/cudaforndummies/ii/ho3/simplePrintf_v3.cu