Chapter 1. Introduction

GPU is specialized for compute-intensive, highly parallel computation, therefore it is designed such that more transistors are devoted to data processing rather than data caching and flow control.

CUDA: a general-purpose parallel computing platform and programming model.

A GPU is built around an array of Streaming Multiprocessors.

Chapter 2. Programming Model

Kernels allow the programmer to define C functions using the _global_ declaration specifier.

Thread hierarchy

Memory hierarchy

Heterogeneous Programming, the CUDA programming model assumes that the CUDA threads execute on a physically separate device that operates as a coprocessor to the host running the C program. The CUDA model assumes that both the host and the device maintain their own separate memory spaces in DRAM, referred to as host memory and device memory, respectively.

Exercise

1. One-dimensional block, two-dimensional block, three-dimensional block.

2. Per-thread local memory, per-block shared memory, global memory.

Questions

1. what does ‘a multithreaded program is partitioned into blocks of threads’ mean?

2. Is per-block shared memory independent from per-thread local memory?