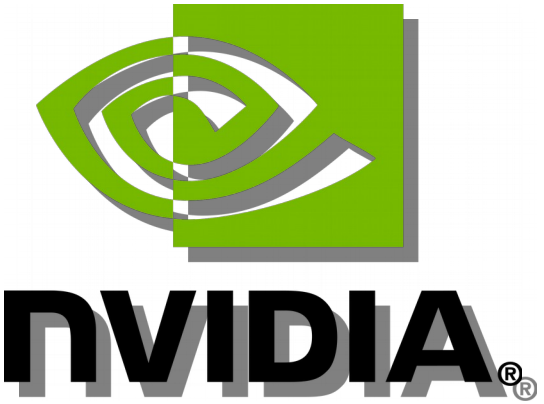
CUDA Programming – Execution Model

by
Dr. Nileshchandra Pikle
Assistant Professor
&
"A certified CUDA instructor by NVIDIA"



A typical C program

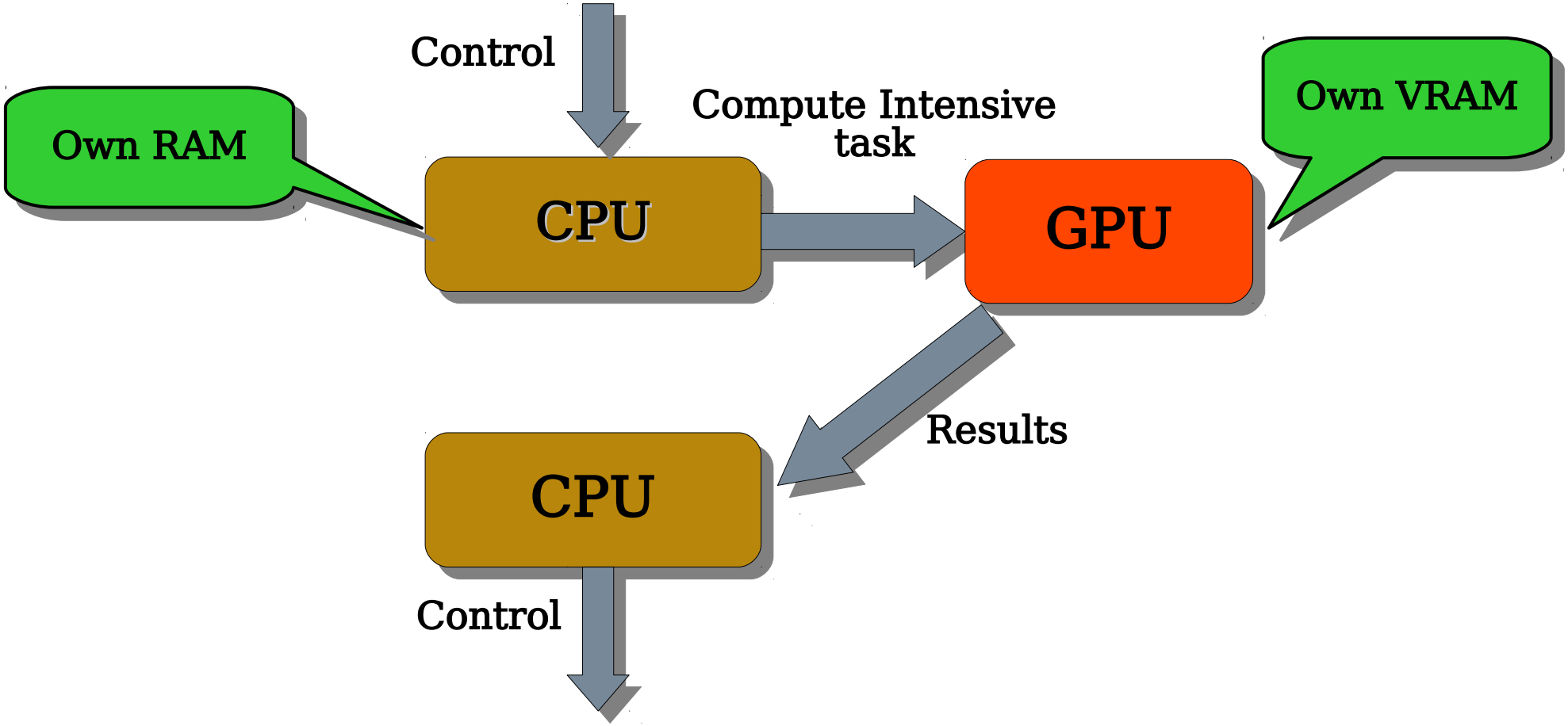
1. Include libraries
2. Declare variables
3. Allocate memory to the variables
4. Initialize variables
5. Perform computations
6. Store results
7. Free varables

GPU is a coprocessor/ accelerator

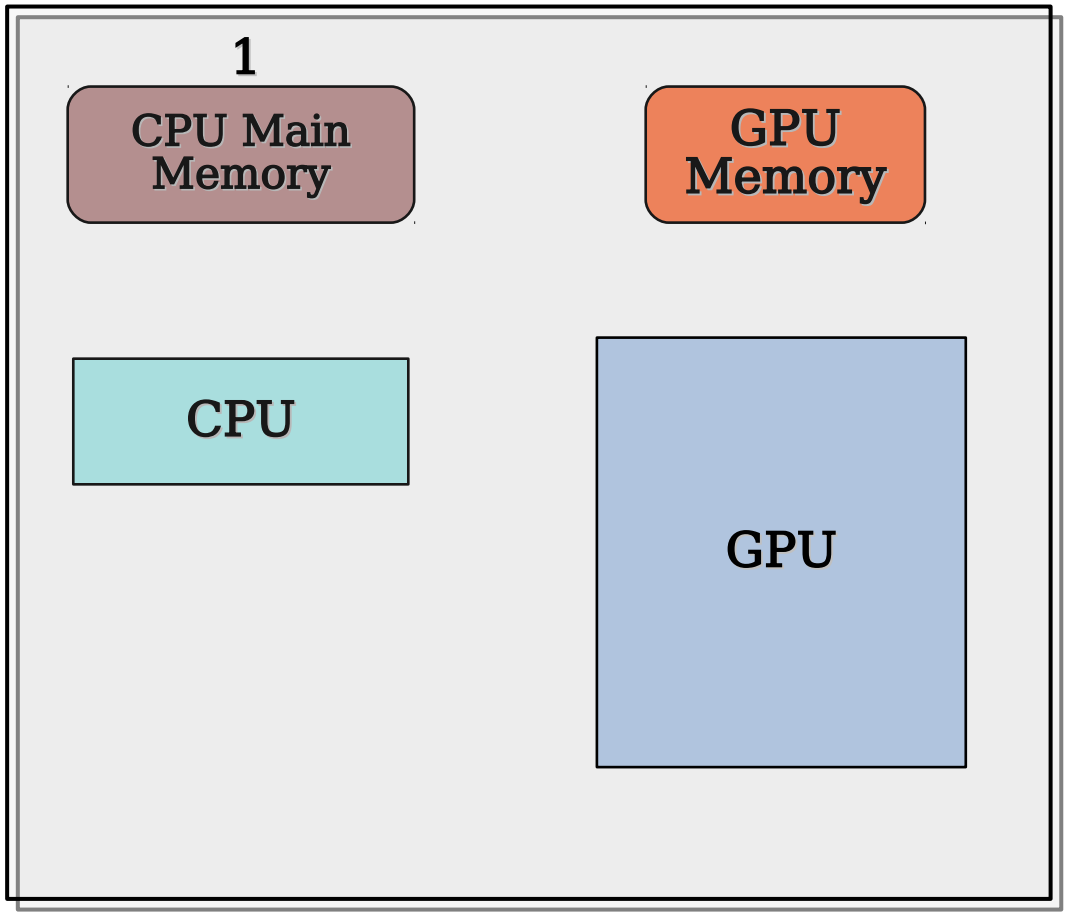
Only computationally intensive jobs are diverted towards GPU

GPU is a coprocessor/ accelerator

Only computationally intensive jobs are diverted towards GPU

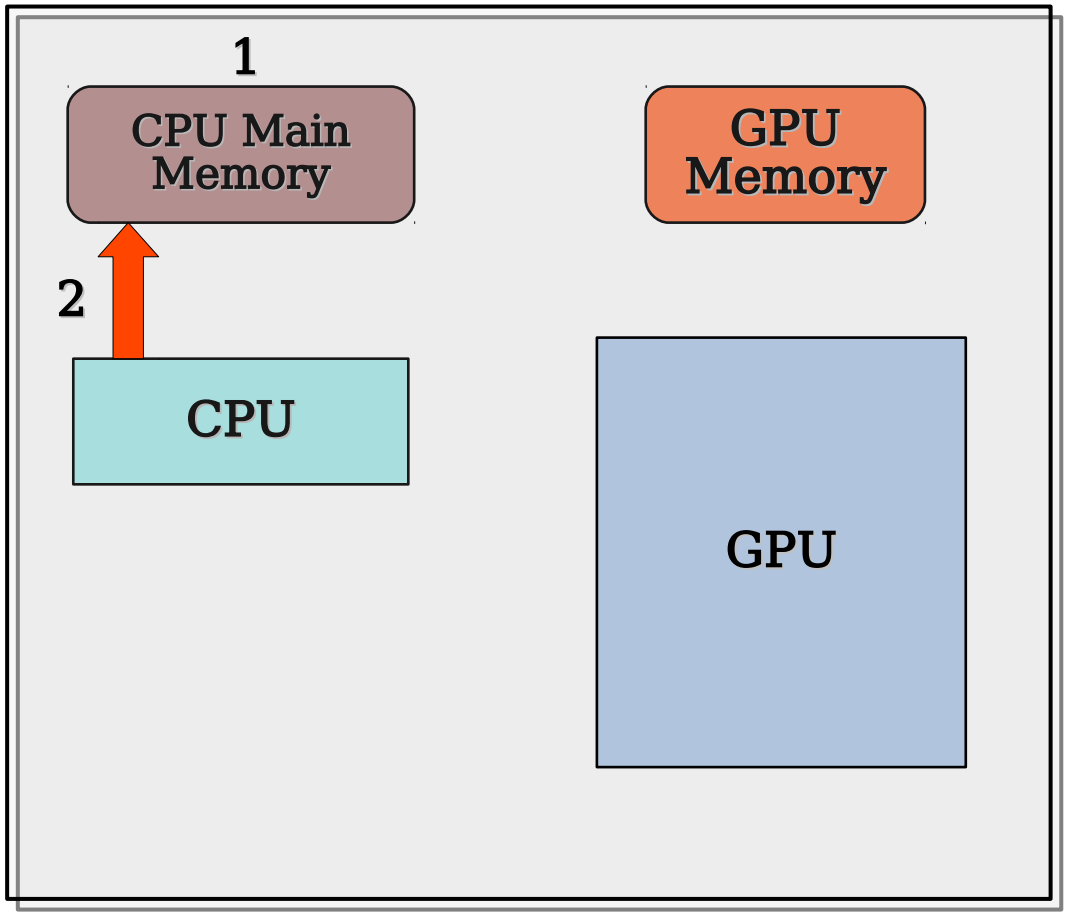


GPU Program Execution Model



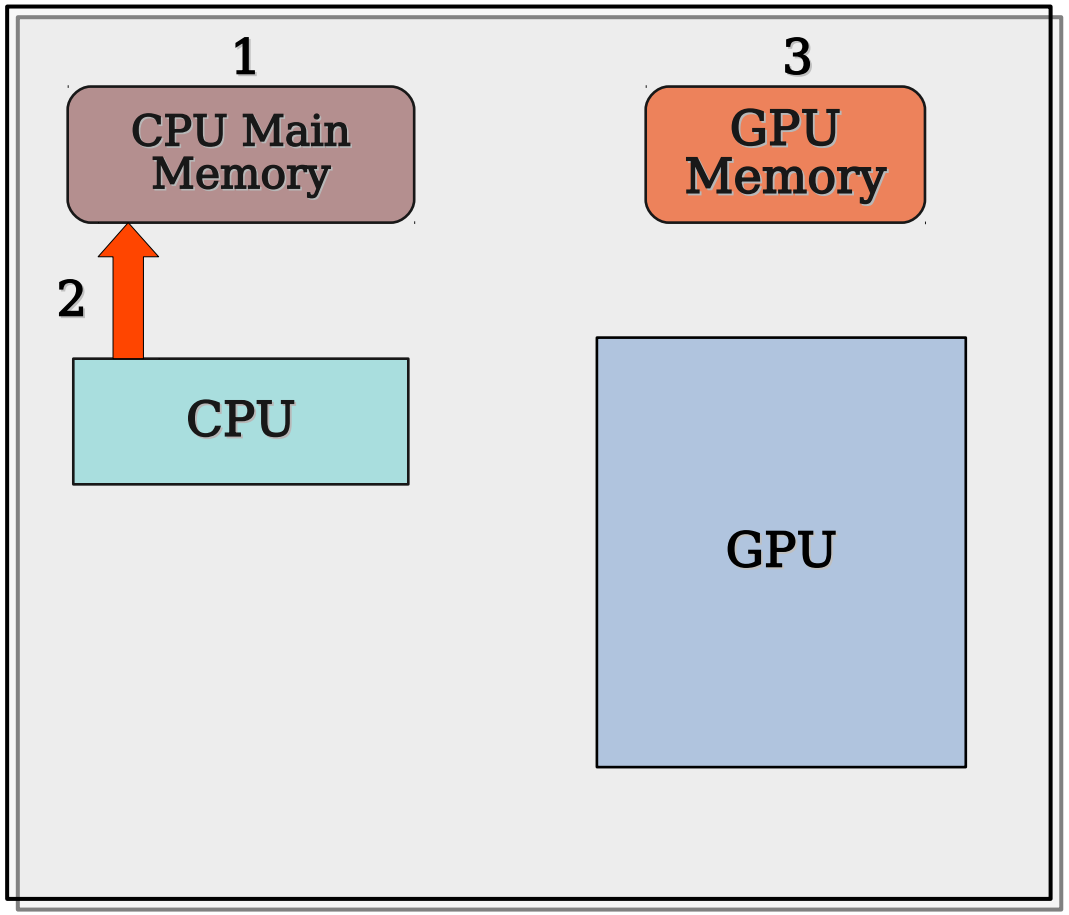
1. Declare CPU variables

GPU Program Execution Model



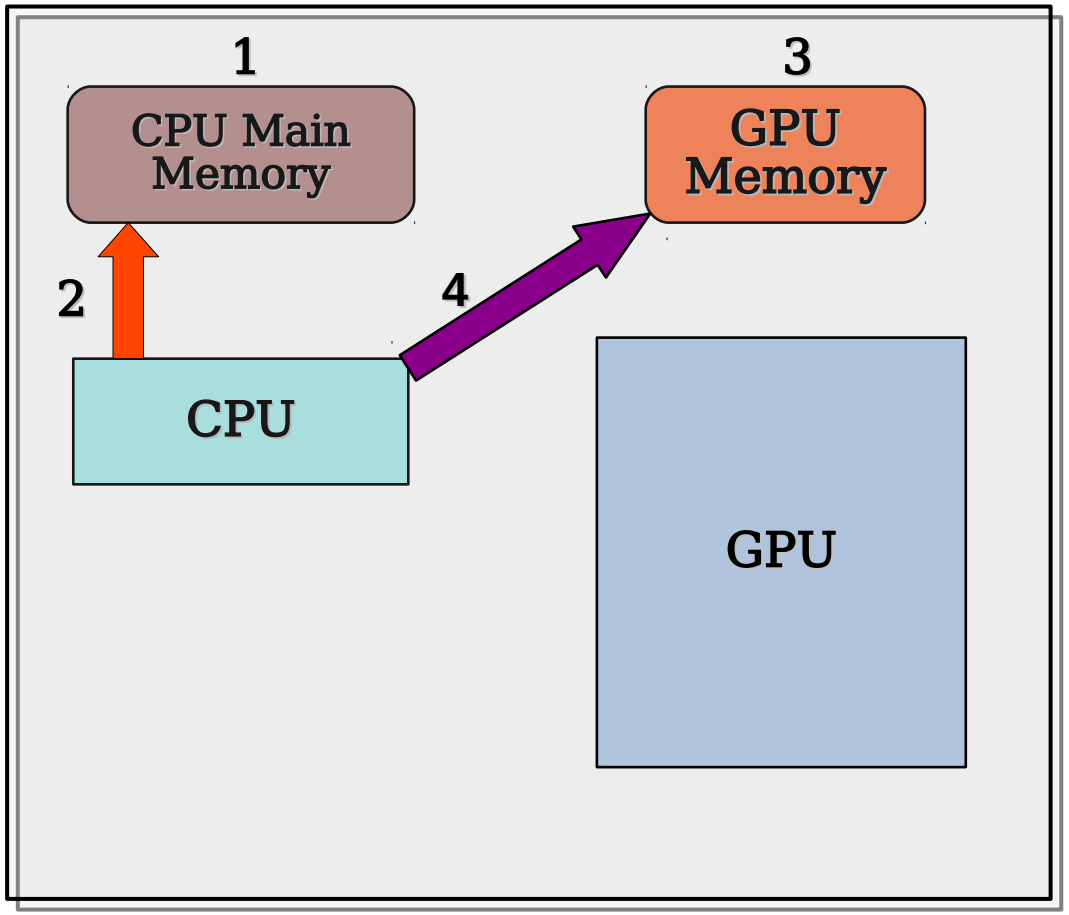
1. Declare CPU variables
2. Allocate memory to CPU variables

GPU Program Execution Model



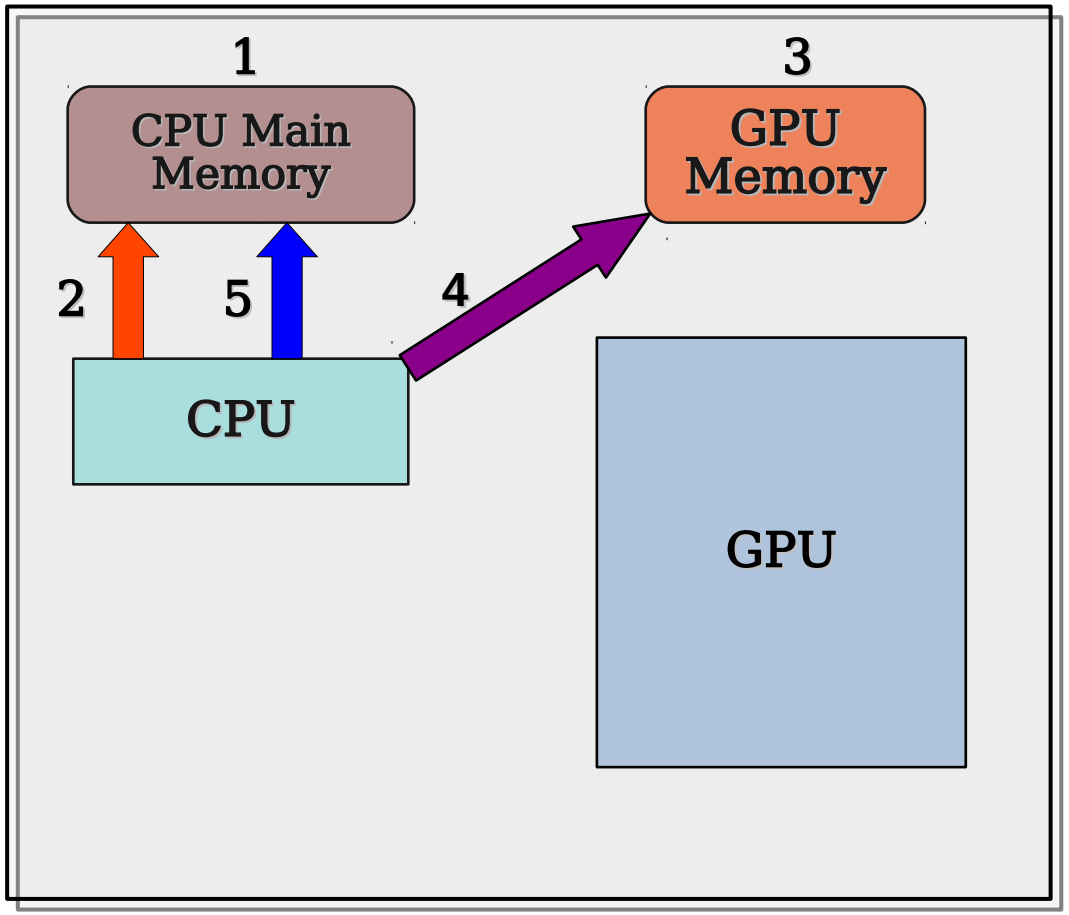
1. Declare CPU variables
2. Allocate memory to CPU variables
3. Declare GPU variables

GPU Program Execution Model



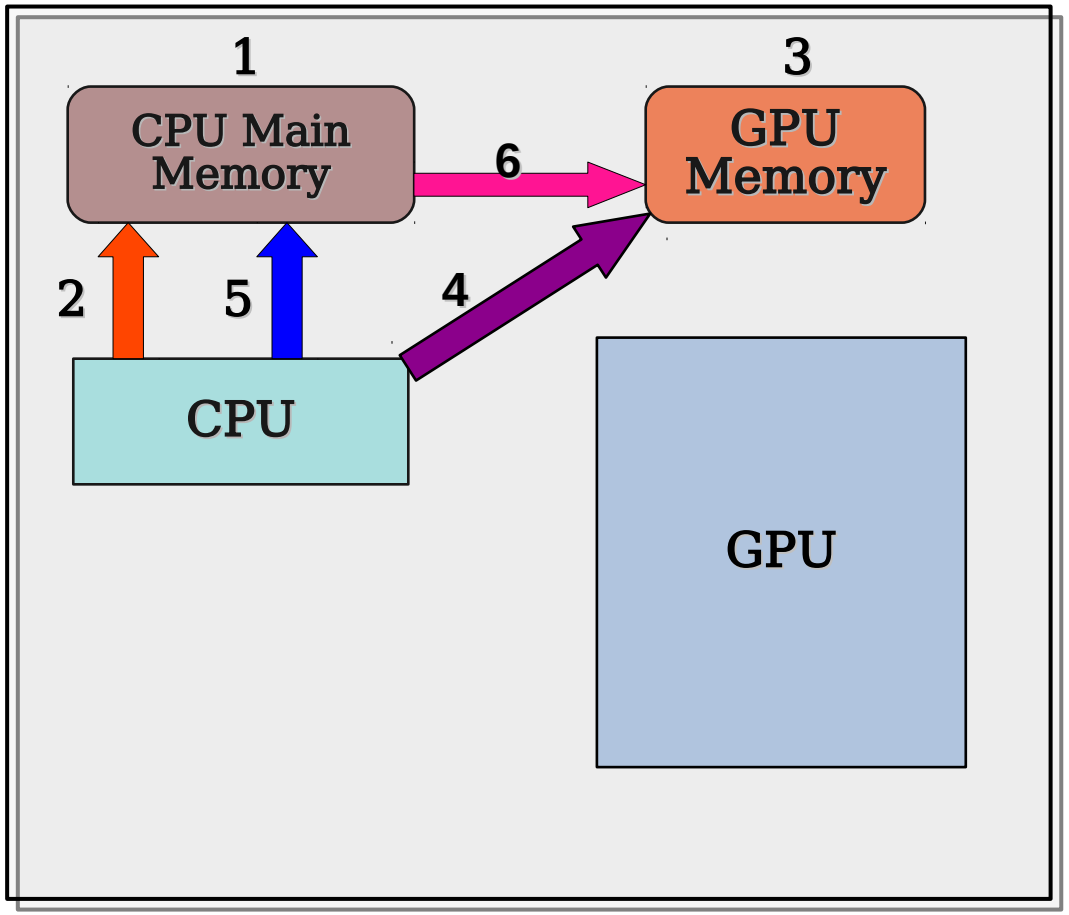
1. Declare CPU variables
2. Allocate memory to CPU variables
3. Declare GPU variables
4. Allocate memory to GPU variables

GPU Program Execution Model



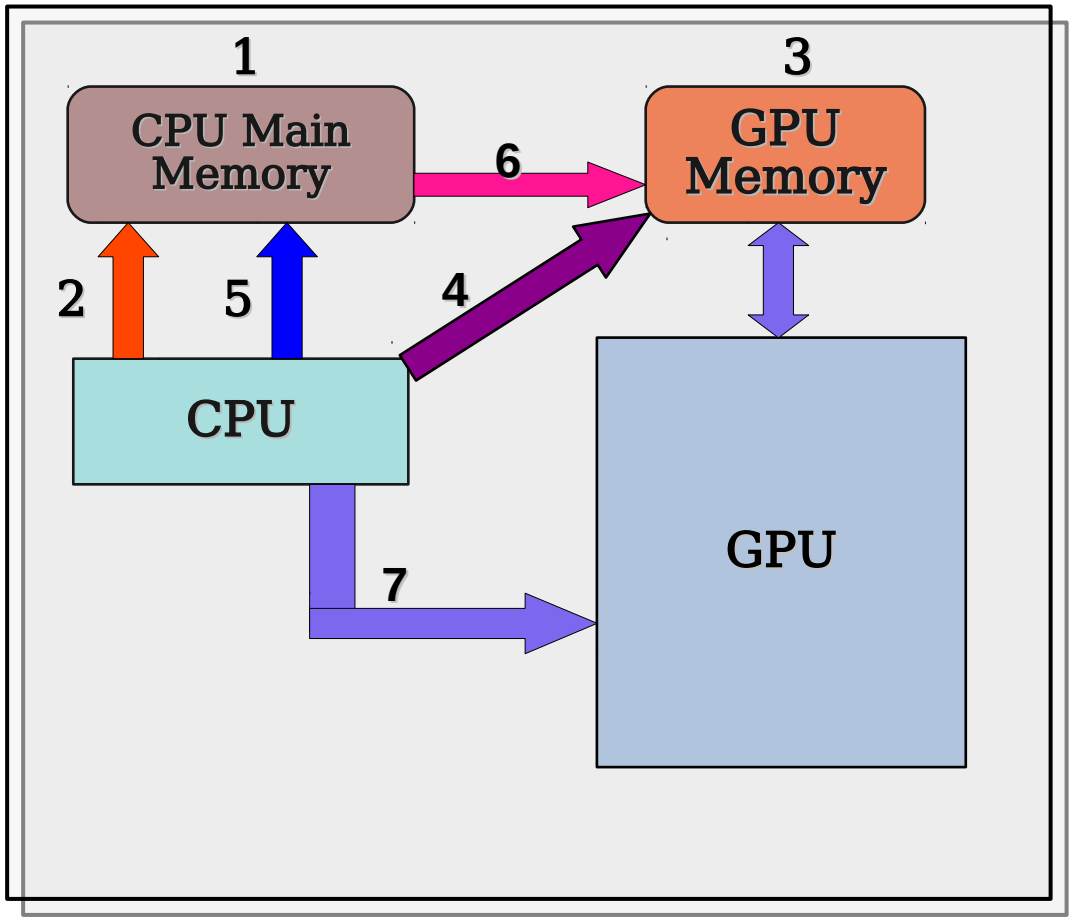
1. Declare CPU variables
2. Allocate memory to CPU variables
3. Declare GPU variables
4. Allocate memory to GPU variables
5. Initialize data in CPU memory

GPU Program Execution Model



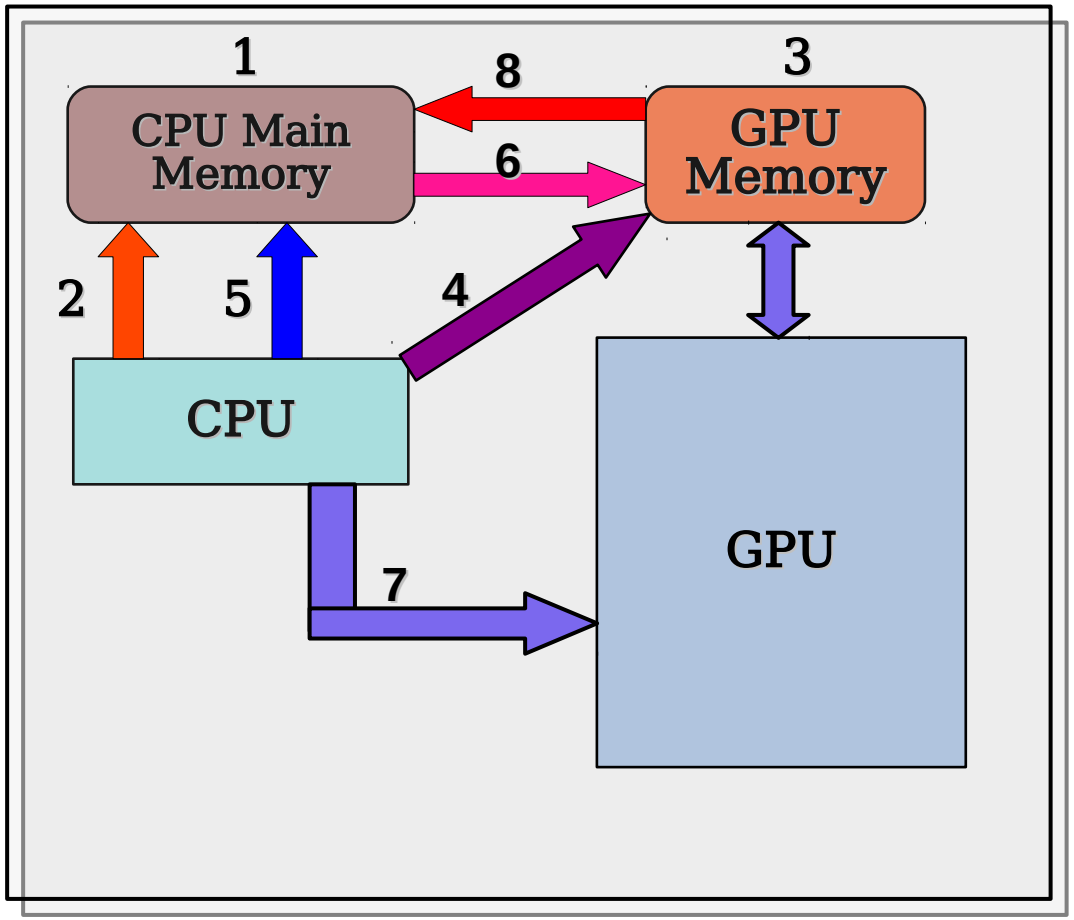
1. Declare CPU variables
2. Allocate memory to CPU variables
3. Declare GPU variables
4. Allocate memory to GPU variables
5. Initialize data in CPU memory
6. Copy data from CPU memory to GPU memory

GPU Program Execution Model



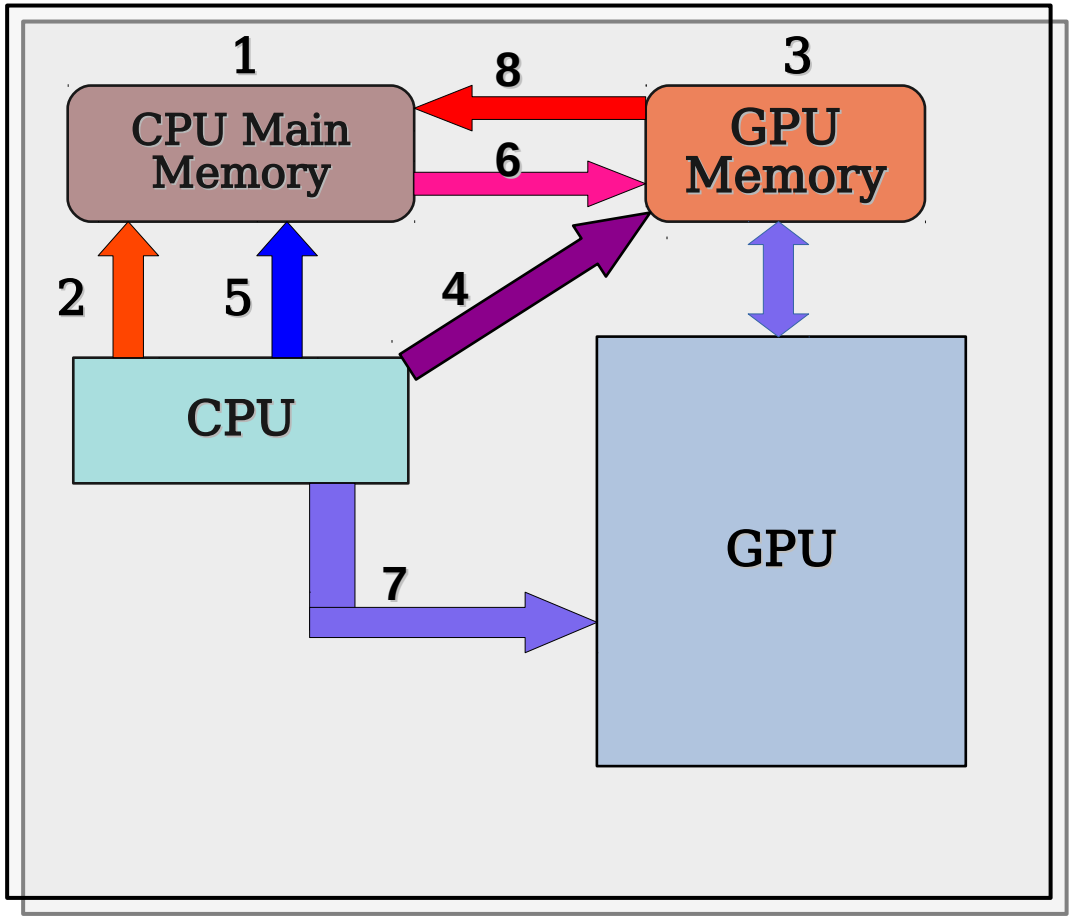
1. Declare CPU variables
2. Allocate memory to CPU variables
3. Declare GPU variables
4. Allocate memory to GPU variables
5. Initialize data in CPU memory
6. Copy data from CPU memory to GPU memory
7. CPU instruct to GPU for parallel Execution

GPU Program Execution Model



1. Declare CPU variables
2. Allocate memory to CPU variables
3. Declare GPU variables
4. Allocate memory to GPU variables
5. Initialize data in CPU memory
6. Copy data from CPU memory to GPU memory
7. CPU instruct to GPU for parallel Execution
8. Copy results back from GPU Memory to CPU memory

GPU Program Execution Model



1. Declare CPU variables
2. Allocate memory to CPU variables
3. Declare GPU variables
4. Allocate memory to GPU variables
5. Initialize data in CPU memory
6. Copy data from CPU memory to GPU memory
7. CPU instruct to GPU for parallel Execution
8. Copy results back from GPU Memory to CPU memory

9. Free both CPU & GPU memories

Summary

- Bypass the computationally intensive tasks to the GPU
- GPU acts as a co-processor / accelerator
- Hybrid programming model – CPU + GPU