GPU 기초

송민호

유튜브 주소: https://youtu.be/C9n_pNvg0Zo

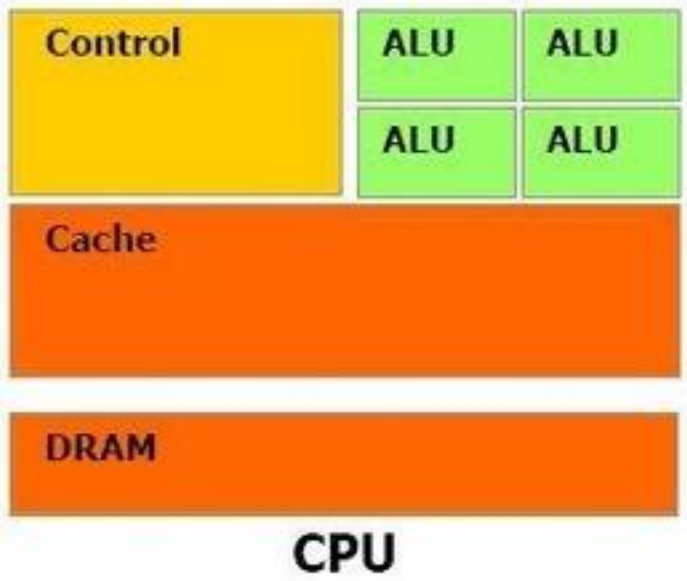
한성대학교 HANSUNG UNIVERSITY

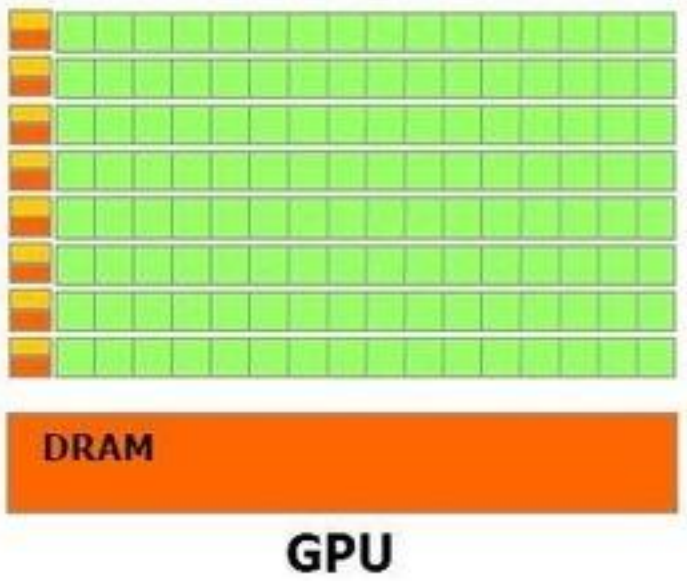
CryptoCraft LAB

GPU

- 그래픽 처리장치(Graphic Processing Unit)
  - 그래픽, 3D를 위한 프로세서로 개발

  - CPU는 **직렬** 명령 처리
  - GPU는 **병렬** 명령 처리



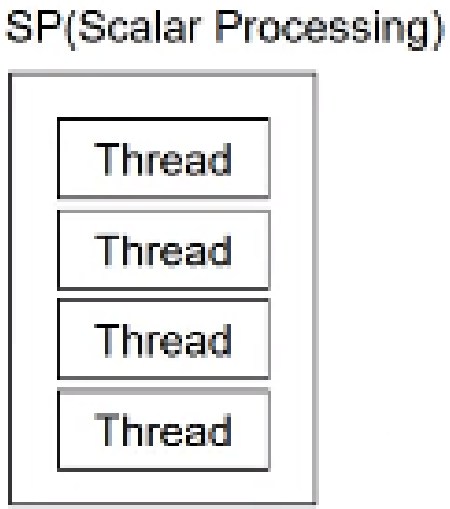


GPU 구성요소

- Thread
  - GPU 내에는 여러 개의 멀티 프로세서가 존재
  - 멀티 프로세서에서 작동되는 하나의 코어

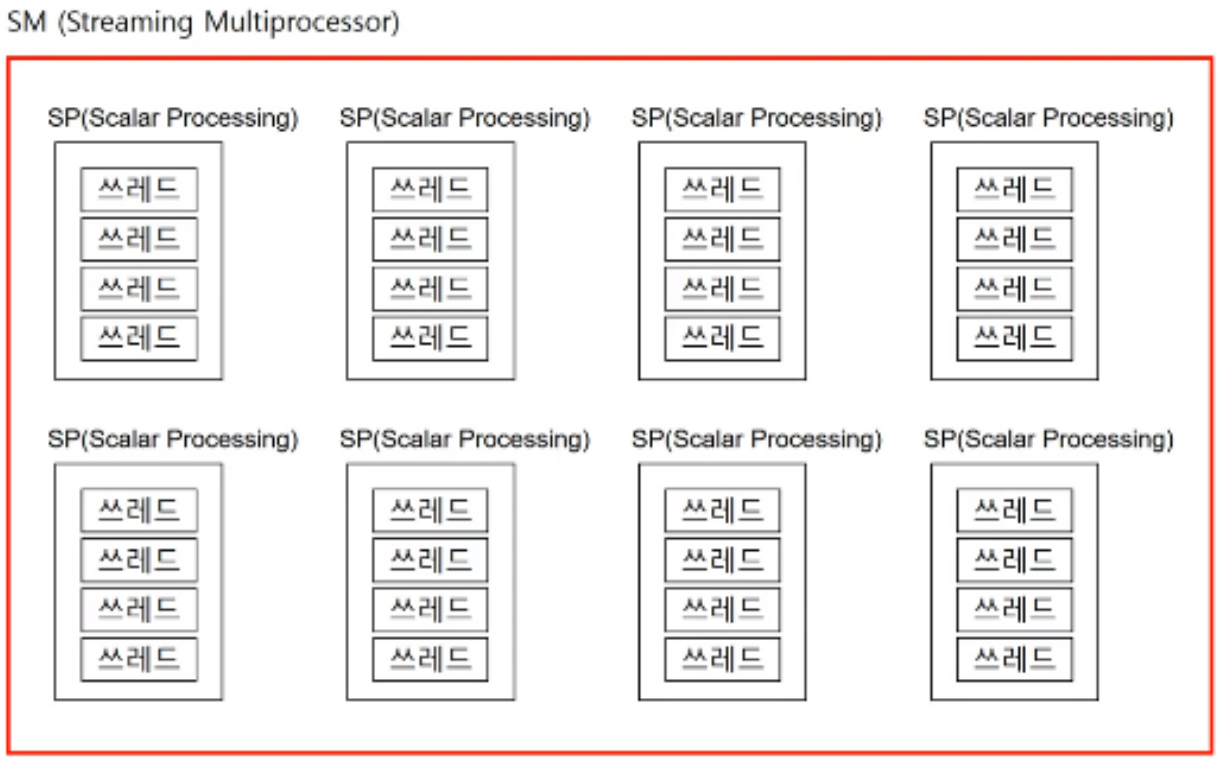
- Scalar Processing(SP)
  - GPU 칩의 가장 작은 단위
  - 4개의 Thread로 구성

- Streaming Multiprocessor(SM)
  - 8개의 SP로 구성



GPU 구성요소

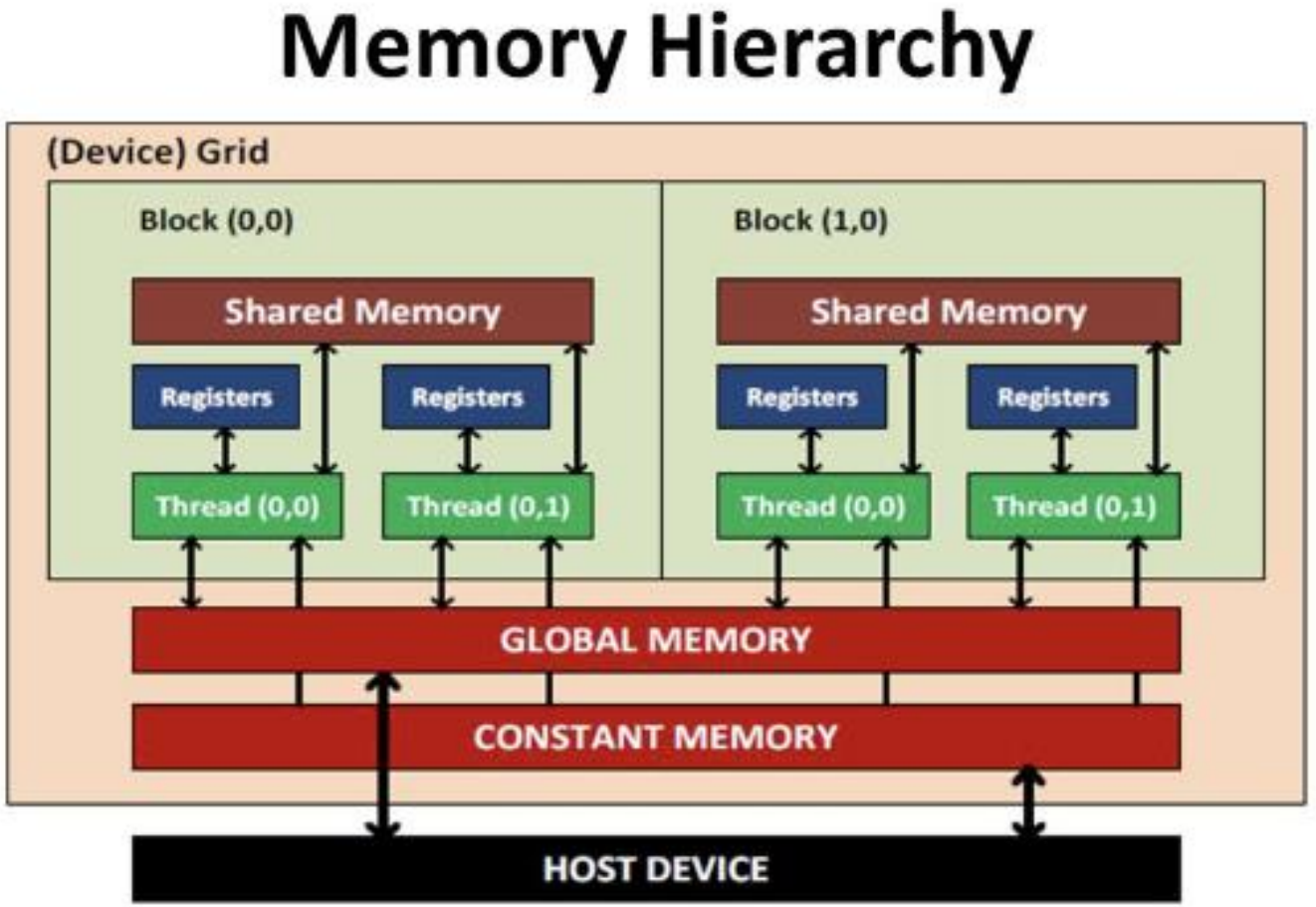
- Warp
  - Thread를 32개의 단위로 나눔
  - 한 명령을 실행하기 위한 최소 단위
- Block
  - Thread의 모음
- Grid
  - Block의 모음
  - 65,536개의 Block으로 구성



GPU 메모리

- Register
  - 접근 속도가 **가장 빠름**
- Shared memory
  - 접근 속도가 **2번째로 빠름**
  - Block 단위

- Local memory
- Constant memory
- Texture memory
- Global memory

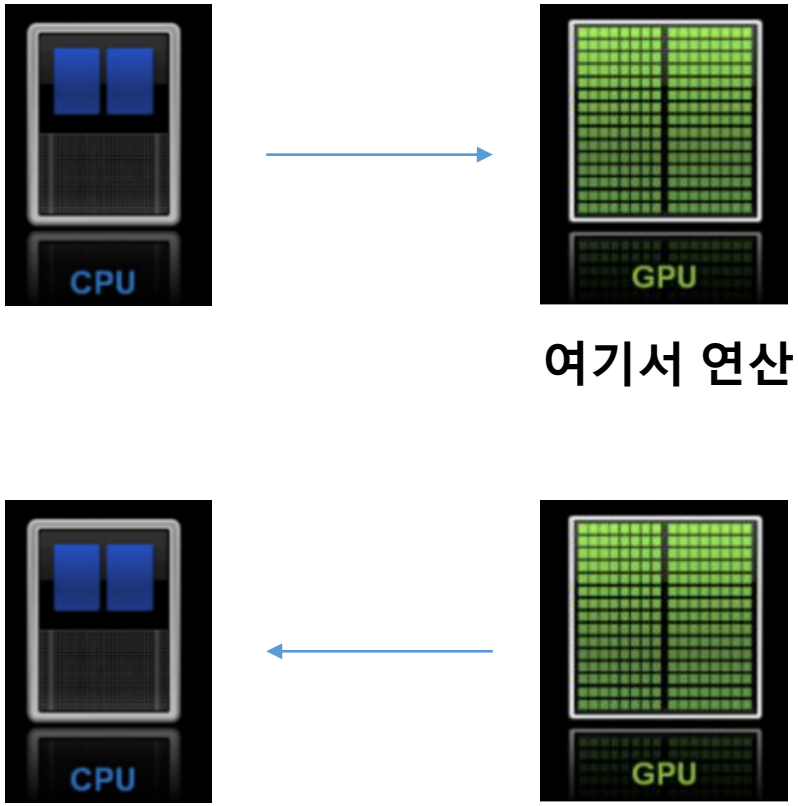


CUDA Memory Hierarchy

CUDA

- GPU 병렬 명령 처리를 프로그래밍 언어를 통해 수행할 수 있게 하는 기술

- CPU에서 GPU로 메모리 복사
- GPU에서 연산 수행
- GPU에서 CPU로 메모리 복사



CUDA

- __device__
  - **GPU**에서 수행되는 함수
- __host__
  - **CPU**에서 수행되는 함수
- __device__ __host__
  - CPU, GPU 두 곳에서 사용
- __global__
  - CPU에서 선언, GPU에서 수행

```
// __global__ 키워드를 붙이면 Device에서 작동된다.
__global__ void kernel( void ) {

}

int main( void ) {
        kernel<<<1,1>>>();
        printf( "Hello, World!\n" );
}
```

GPU 메모리

- <<< BlockNum, ThreadNum >>>
  - 커널 함수에서 사용
  - Block, Thread 개수 조절
  - 1번째 파라미터는 Block
  - 2번째 파라미터는 Thread

```
// add 함수를 1번 실행
add<<< 1, 1 >>>( dev_a, dev_b, dev_c );

// N개의 블록으로 1개의 스레드를 통해 실행(Parallel Block)
add<<< N, 1 >>>( dev_a, dev_b, dev_c );

// 1개의 블록으로 N개의 스레드를 통해 실행(Parallel Thread)
add<<< 1, N >>>( dev_a, dev_b, dev_c );
```

Q & A