Shared Memory bank conflicts

https://youtu.be/2jwAMSuMxVE

1. 공유 메모리

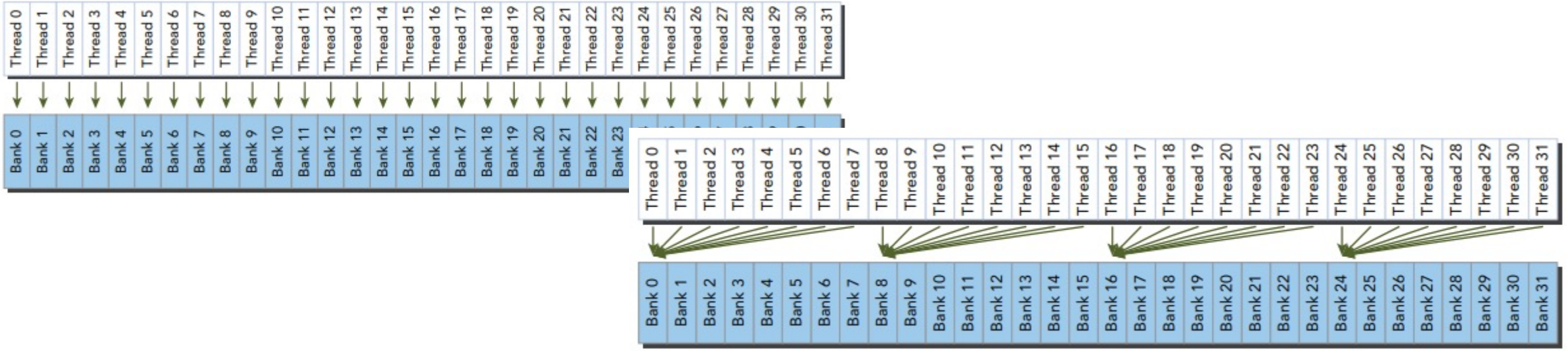
- 공유메모리는 onchip에 위치해 글로벌 메모리보다 빠른 처리속도를 보여주는 메모리.

- 동적과 정적으로 사용할 수 있고, 정적으로 사용할 경우 48KB 까지 사용할 수 있고 동적으로는 100KB 정도까지 사용 가능하다.
  - 공유메모리가 L1 캐시랑 자원을 공유한다고 하는데, 레지스터가 많이 필요할 경우에는 공유메모리를 동적으로 많이 할당해서 사용하는 것이 오히려 안 좋을 수도 있다고 함.

2. Bank Conflicts

- 공유메모리는 높은 bandwidth를 위해서 뱅크라는 32개의 동일한 사이즈의 메모리 모듈로 나뉨
  - 워프 단위로 처리가 되는데, 워프는 32개의 thread 그룹

- 서로 다른 스레드가 동일한 bank에 접근하면서 병렬로 처리되어야 할 것이 순차적으로 처리되는 문제가 발생.



3. 분석 계기

- Camellia GPU 구현 후 프로파일링을 하였는데, Bank conflict 문제가 있다고 나옴.
  - store 과정에서 뱅크 충돌이 발생하고 있음.

```
__shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
u32 ptInit[4];

if (threadIdx.x < TABLE_SIZE)
{
    for (u8 bankIndex = 0; bankIndex < SHARED_MEM_BANK_SIZE; bankIndex++)
    {
        tS[threadIdx.x][bankIndex] = SBOX[threadIdx.x];
    }
}
__syncthreads();
```

```
OPT   Est. Speedup: 93.96%
      The memory access pattern for shared stores might not be optimal and causes on average a 32.0 - way bank
      conflict across all 262144 shared store requests.This results in 8126465 bank conflicts,  which represent
      96.87% of the overall 8388636 wavefronts for shared stores. Check the Source Counters section for
      uncoalesced shared stores.
```

3. 분석 계기

- Camellia GPU 구현 후 프로파일링을 하였는데, Bank conflict 문제가 있다고 나옴.
  - store 과정에서 뱅크 충돌이 발생하고 있음.

```
__shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
u32 ptInit[4];

if (threadIdx.x < TABLE_SIZE)
{
    for (u8 bankIndex = 0; bankIndex < SHARED_MEM_BANK_SIZE; bankIndex++)
    {
        tS[threadIdx.x][bankIndex] = SBOX[threadIdx.x];
    }
}
__syncthreads();
```

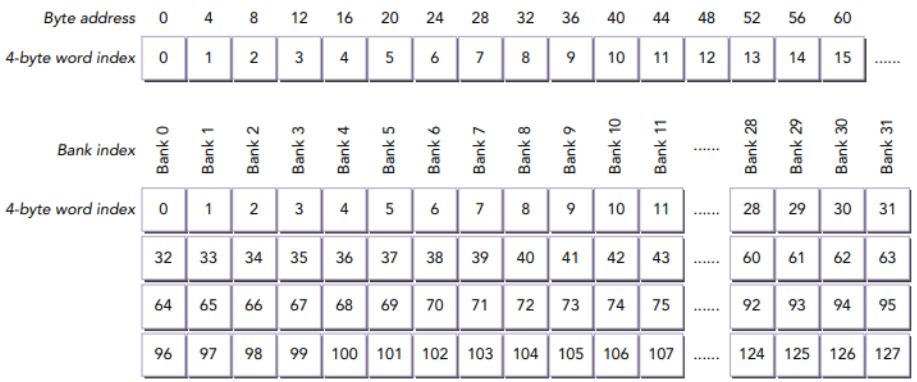
Bank 0	Bank 1	Bank 2	...	...	Bank 29	Bank 30	Bank 31
tS[0][0]	tS[0][1]	tS[0][2]	...	...	tS[0][29]	tS[0][30]	tS[0][31]
Sbox[0]	Sbox[0]	Sbox[0]	...	...	Sbox[0]	Sbox[0]	Sbox[0]

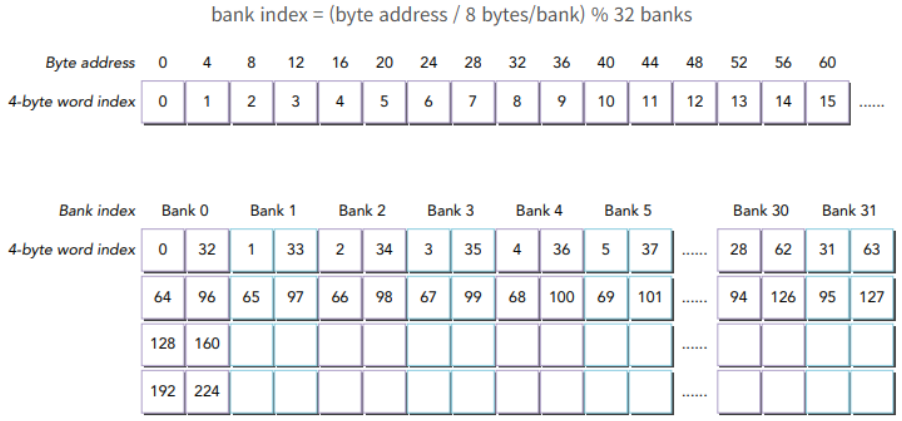
tS[255][0]	tS[255][1]	tS[255][2]	...	...	tS[255][29]	tS[255][30]	tS[255][31]
Sbox[255]	Sbox[255]	Sbox[255]	...	...	Sbox[255]	Sbox[255]	Sbox[255]

4. 원인 분석

- Kepler부터는 shared memory가 2가지 address mode의 32 bank를 가진다고 함.

2012	Kepler	GeForce 600, 700
2014	Maxwell	GeForce 750, 900
2016	Pascal	GeForce 10
2018	Turing	GeForce 16, RTX 20
2020	Ampere	RTX 30
2022	Ada Lovelace	RTX 40





확인되지 않음

https://junstar92.tistory.com/288

4. 원인 분석

```
__shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
u32 ptInit[4];

if (threadIdx.x < TABLE_SIZE)
{
    for (u8 bankIndex = 0; bankIndex < SHARED_MEM_BANK_SIZE; bankIndex++)
    {
        tS[threadIdx.x][bankIndex] = SBOX[threadIdx.x];
    }
}
__syncthreads();
```

```
.This results in 8126465 bank conflicts,
```

32*256*1024 = 8,388,608

Bank 32개

Block size 1024

Bank 0	Bank 1	Bank 2	...	...	Bank 29	Bank 30	Bank 31
tS[0][0]	tS[0][1]	tS[0][2]	...	...	tS[0][29]	tS[0][30]	tS[0][31]
Sbox[0]	Sbox[0]	Sbox[0]	...	...	Sbox[0]	Sbox[0]	Sbox[0]

스레드 256개

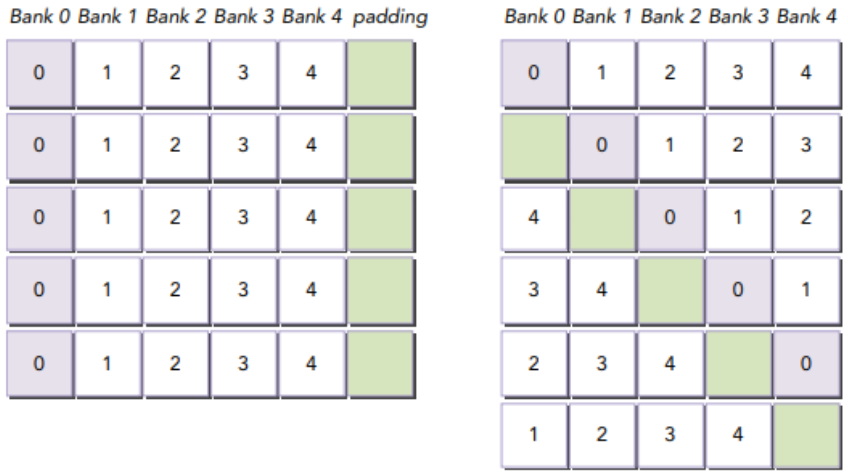
tS[255][0]	tS[255][1]	tS[255][2]	...	...	tS[255][29]	tS[255][30]	tS[255][31]
Sbox[255]	Sbox[255]	Sbox[255]	...	...	Sbox[255]	Sbox[255]	Sbox[255]

5. 문제 해결

- 패딩 방법: 배열 크기를 임의로 +1 해서 하나씩 미루는 방법.

```
# warpThreadIndex = threadIdx.x & 31;
# __shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- -------------
Metric Name                                                Metric Unit  Metric Value
---------------------------------------------------------- ----------- -------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                13,862,869.43
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                   13,879,511
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                   13,848,110
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                  415,886,083
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                   270,882.27
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                      277,760
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                      269,824
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                    8,126,468
---------------------------------------------------------- ----------- -------------
Time elapsed: 7.207182 sec
```

```
# warpThreadIndex = threadIdx.x & 31;
# __shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE+1];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- -----------------
Metric Name                                                Metric Unit      Metric Value
---------------------------------------------------------- ----------- -----------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                11,681,513,181.97
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                   11,977,893,725
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                   11,635,343,647
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                  350,445,395,459
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                             0.40
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                                3
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                                0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                               12
---------------------------------------------------------- ----------- -----------------
Time elapsed: 10.126232 sec
```



6. 성능 비교 분석

- 뱅크 충돌 최소화 방법
  - 7.2초

```
# warpThreadIndex = threadIdx.x & 31;
# __shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- ------------
Metric Name                                                Metric Unit  Metric Value
---------------------------------------------------------- ----------- ------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                13,862,869.43
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                   13,879,511
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                   13,848,110
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                  415,886,083
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                   270,882.27
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                      277,760
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                      269,824
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                    8,126,468
---------------------------------------------------------- ----------- ------------
Time elapsed: 7.207182 sec
```

- 글로벌 메모리 사용
  - 9.2초

```
# __device__ u32 SBOX[TABLE_SIZE];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- ------------
Metric Name                                                Metric Unit Metric Value
---------------------------------------------------------- ----------- ------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                           0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                           0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                           0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                           0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                           0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                           0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                           0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                           0
---------------------------------------------------------- ----------- ------------
Time elapsed: 9.720719 sec
```

- 공유메모리 일반적인 방법
  - 157.6초

```
# warpThreadIndex = 0;
# __shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- ------------------
Metric Name                                                Metric Unit       Metric Value
---------------------------------------------------------- ----------- ------------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg             132,914,617,868.97
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                136,288,613,554
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                132,393,135,953
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum               3,987,438,536,069
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                      270,882.23
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                         277,760
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                         269,824
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                       8,126,467
---------------------------------------------------------- ----------- ------------------
Time elapsed: 157.662399 sec
```

6. 성능 비교 분석

- 라운드키 공유메모리 사용 분석
- Sbox 처럼 32개 복사
- 그냥 하나만
  - broadcast access
  - 모든 스레드가 동일한 뱅크 사용할 때

```
# warpThreadIndex = threadIdx.x & 31;
# __shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- ------------
Metric Name                                                Metric Unit  Metric Value
---------------------------------------------------------- ----------- ------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg             13,862,869.43
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                13,879,511
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                13,848,110
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                415,886,083
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                270,882.27
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                   277,760
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                   269,824
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                 8,126,468
---------------------------------------------------------- ----------- ------------
Time elapsed: 7.207182 sec
```

```
# warpThreadIndex = threadIdx.x & 31;
# __shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
# __shared__ u32 rkC[68][SHARED_MEM_BANK_SIZE];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- ------------
Metric Name                                                Metric Unit Metric Value
---------------------------------------------------------- ----------- ------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                5,532,367
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                5,541,337
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                5,523,139
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum              165,971,010
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                341,879.53
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                   350,560
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                   340,544
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                10,256,386
---------------------------------------------------------- ----------- ------------
Time elapsed: 9.161233 sec
```

```
# warpThreadIndex = threadIdx.x & 31;
# __shared__ u32 tS[TABLE_SIZE][SHARED_MEM_BANK_SIZE];
# __shared__ u32 rkC[68];
Camellia_128_CTR(unsigned int *, unsigned int *, unsigned int *, unsigned long long *) (1024, 1, 1)x(512, 1, 1), Context 1, Stream 7, Device 0, CC 8.6
Section: Command line profiler metrics
---------------------------------------------------------- ----------- ------------
Metric Name                                                Metric Unit Metric Value
---------------------------------------------------------- ----------- ------------
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                4,444,360.17
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                   4,497,085
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                   4,431,534
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                 133,330,805
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                270,882.27
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                   277,760
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                   269,824
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                 8,126,468
---------------------------------------------------------- ----------- ------------
Time elapsed: 9.138584 sec
```

7. 이후 GPU 공부 방향

- Warp 투표/매치/리듀스/셔플 기능 활용 방안

1. Warp Vote Functions (워프 투표 기능)

- 워프 내의 모든 스레드들이 각자 조건을 평가하고, 그 결과(참 또는 거짓)를 투표하듯 모아서 확인하는 기능입니다.
- 주로 모든 스레드가 특정 조건을 만족하는지, 아니면 하나라도 만족하는지 빠르게 체크할 때 사용됩니다.
- 예: 모든 스레드가 특정 값 이상인지 빠르게 검사할 때 유용합니다.

2. Warp Match Functions (워프 매치 기능)

- 워프 내의 스레드들 중에서 특정 값이 일치(match)하는 스레드를 빠르게 찾는 기능입니다.
- 특정 데이터 값을 가진 스레드들을 그룹화하여 데이터 교환이나 추가 연산을 할 때 사용됩니다.
- 예: 워프 내에서 동일한 값을 가진 스레드끼리 묶어서 처리할 수 있습니다.

3. Warp Reduce Functions (워프 리듀스 기능)

- 워프 내의 모든 스레드가 가진 값을 하나의 값으로 빠르게 줄이는(reduce) 연산입니다.
- 대표적으로 합(sum), 최대값(max), 최소값(min) 등의 값을 빠르게 계산할 때 사용됩니다.
- 예: 워프 내 모든 스레드가 가진 숫자를 빠르게 더해 하나의 합계 값을 얻을 때 유용합니다.

4. Warp Shuffle Functions (워프 셔플 기능)

- 워프 내의 스레드들이 서로 데이터를 직접 교환(shuffle)하는 기능입니다.
- 공유 메모리(shared memory)를 사용하지 않고도 빠르게 데이터를 교환할 수 있어 성능 최적화에 매우 유리합니다.
- 예: 특정 스레드가 가진 값을 워프 내 다른 스레드들과 효율적으로 공유하고 싶을 때 사용합니다.

https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html?highlight=shared%2520memory#warp-vote-functions

감 사 합 니 다