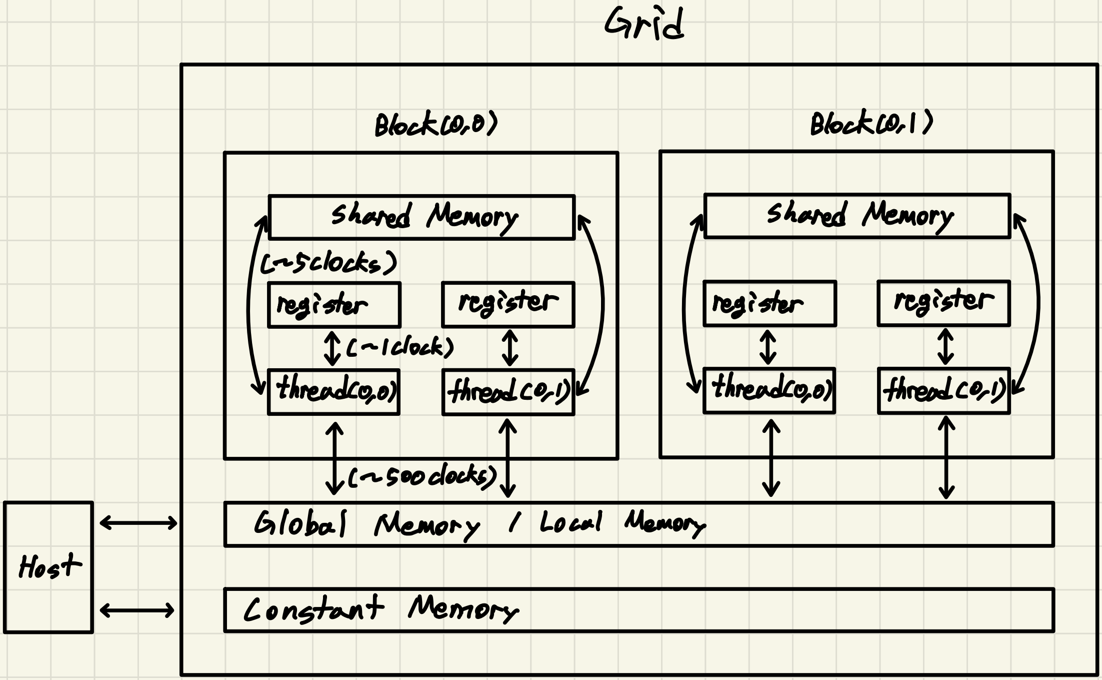
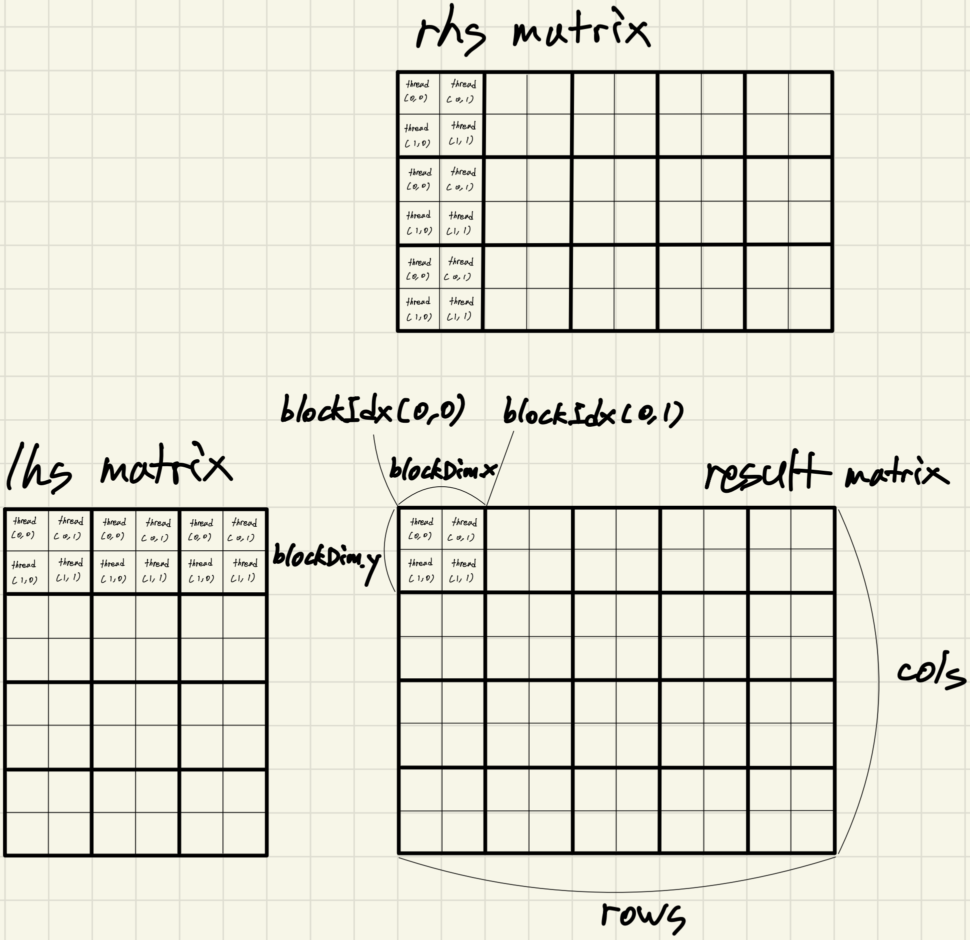
CUDA memory hierarchy



1. max 1024 threads per 1 block
2. 1 block per SM
3. (32 threads = 1 warp) per SM

```
__global__ void kernelFunc (float* dst, const float* src) {
    float p = 0; // register per thread
    float arr[10]; // local per thread
    __shared__ float partial_sum[1024]; //shared Memory per Block
}
```

TILED MATRIX MULTIPLICATION Basic

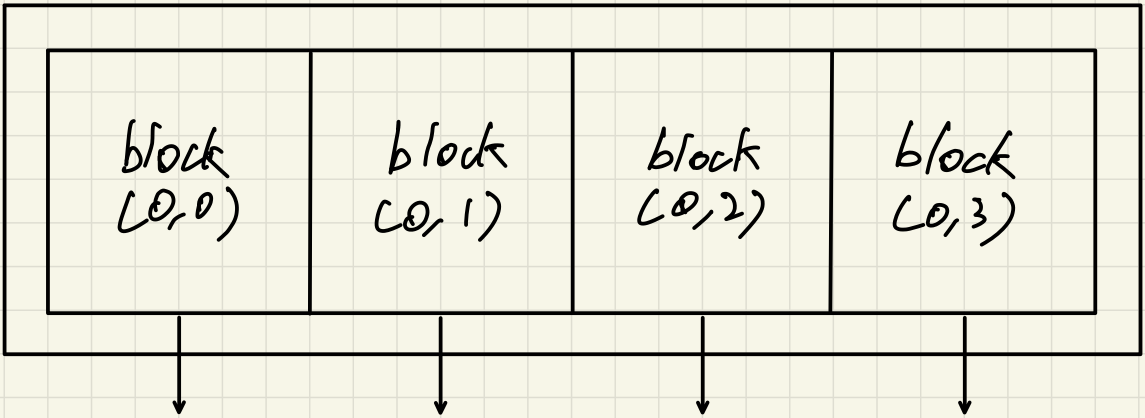


pseudo code

```
int x = blockIdx.x x blockDim.x + threadIdx.x
int y = blockIdx.y x blockDim.y + threadIdx.y
result[y, x] = sum ( lhs[y, :] x rhs[:, x])
```

Calculate the number of warp
(result matrix dimension 256 x 1)

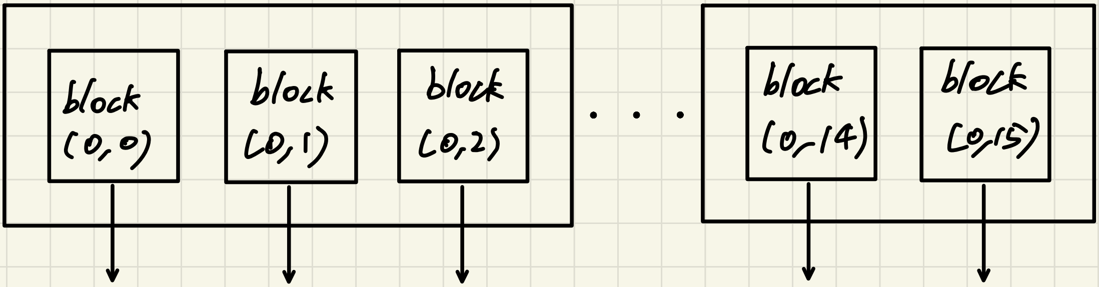
TILE WIDTH = 32



32 warps (=1024 threads) 32 warps 32 warps 32 warps

= total 128 warps

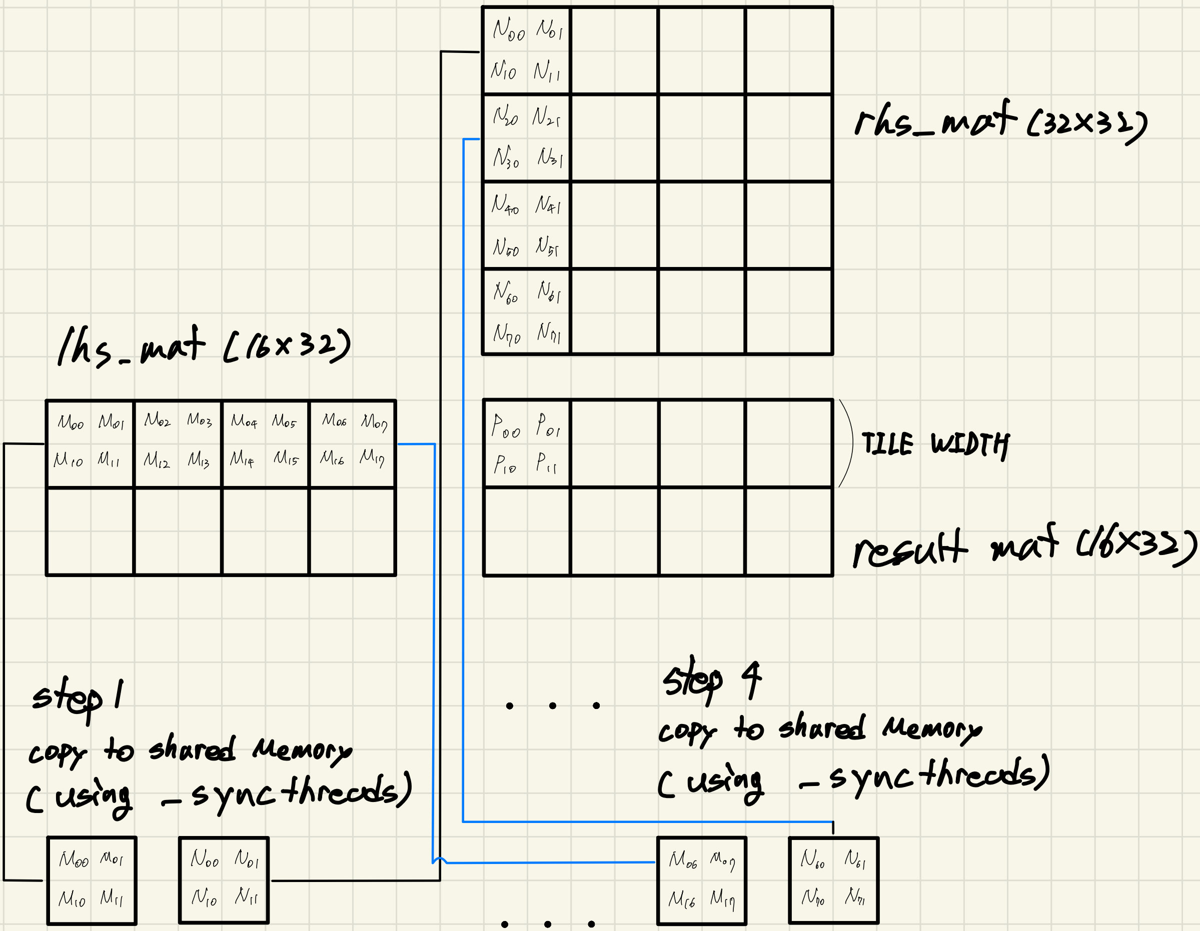
TILE WIDTH = 8



2 warps (=64 threads) 2 warps 2 warps . . . 2 warps 2 warps

= total 64 warps

TILED MATRIX MULTIPLICATION
using shared memory (TILE WIDTH = 2)



calculation

$P_{00} \pm M_{00} \times N_{00} + M_{01} \times N_{10}$

$P_{01} \pm M_{00} \times N_{01} + M_{01} \times N_{11}$

$P_{10} \pm M_{10} \times N_{00} + M_{11} \times N_{10}$

$P_{11} \pm M_{10} \times N_{01} + M_{11} \times N_{11}$

calculation

$P_{00} \pm M_{06} \times N_{60} + M_{07} \times N_{70}$

$P_{01} \pm M_{06} \times N_{61} + M_{07} \times N_{71}$

$P_{10} \pm M_{16} \times N_{60} + M_{17} \times N_{70}$

$P_{11} \pm M_{16} \times N_{61} + M_{17} \times N_{71}$