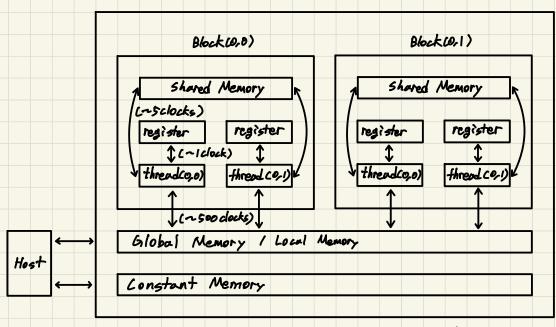
CUDA memory hierarchy

Grid



- 1. Max 1024 threads per 1 block 2. 1 block per SM
- 3. (32 thread 5 = 1 warp) per 5m
- -global void ternel Func (float x dst, const float x src) {

 float p = src[threadIdx.x); // register per thread

 float heap[(0]; // (ocal per thread
 - __shared__ float partial_sum[1024]; //shared Memory per
 Block

TILED MATRIX MULTIPLICATION Basic

		lhs						h	LS				
ſ	tiread (e, e)	thread (o, 1)	Hiread (0,0)	Haread C 0, 1)	firead (0,0)	thread (o, 1)		firead (e, o)	Hiread (0,1)				
	Hiread (1,0)	thread (1,1)	thread (1,0)	thread (1,1)	Hread (1,0)	thread (1,1)		Hread (1,0)	thread (1,1)				
								Haread (0,0)	faread (0,1)				
Ī								Hread (1,0)	thread (1,1)				
								timend (e, e)	thread (o, 1)				
Ī								firend (1,0)	thread (1,1)				
Ī													
Ī													
Ī			, ,										
×(0,	.0	1	olozk]c	1x CO, 1)								

blackIdx(0,0 blackIdx (0,1)

blackPim.y

timal timal (0,0) (0,1)

```
pseudo code

int x = blockIdx.x x blockDim x f threadIdx.x;

int y = blockIdx.y x blockDim.y f threadIdx.y;

double sum = 0.0;

for Cint i=0; i (ths_cols; +i)

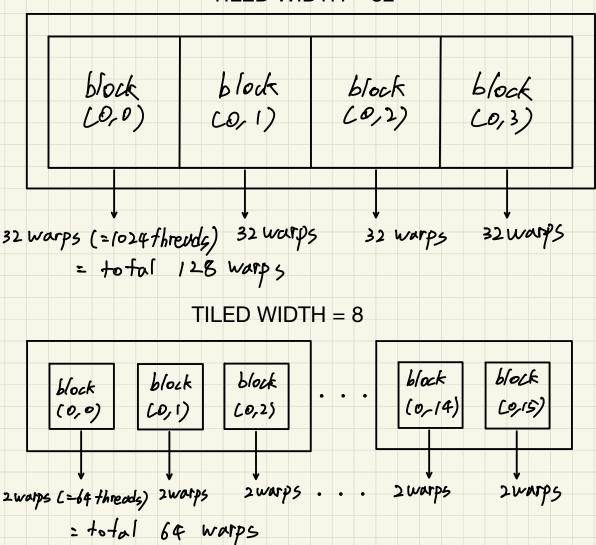
{
    sum f ths[y x ths_cols + i] x ths [i x ths_cols + x];

}

result[y x rows +x] = sum;
```

Result matrix dimension 256 x 1 TILED MATRIX MULTIPLICATION

TILED WIDTH = 32



										1				
										<u> </u>				