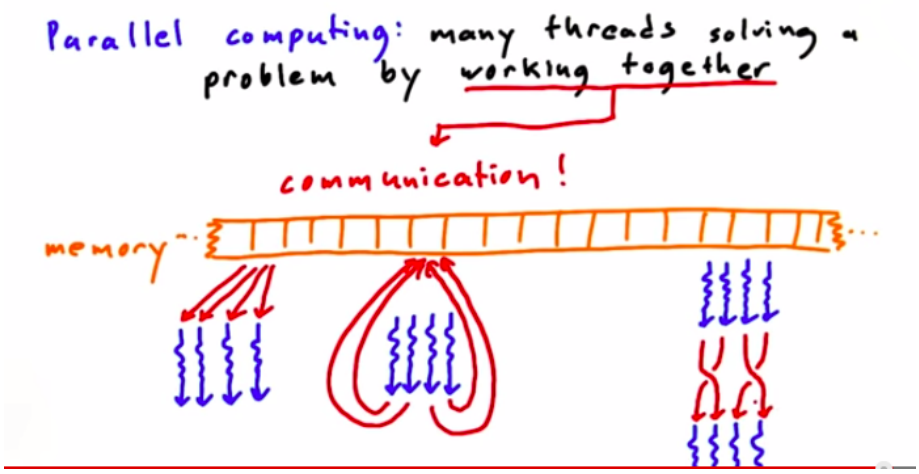
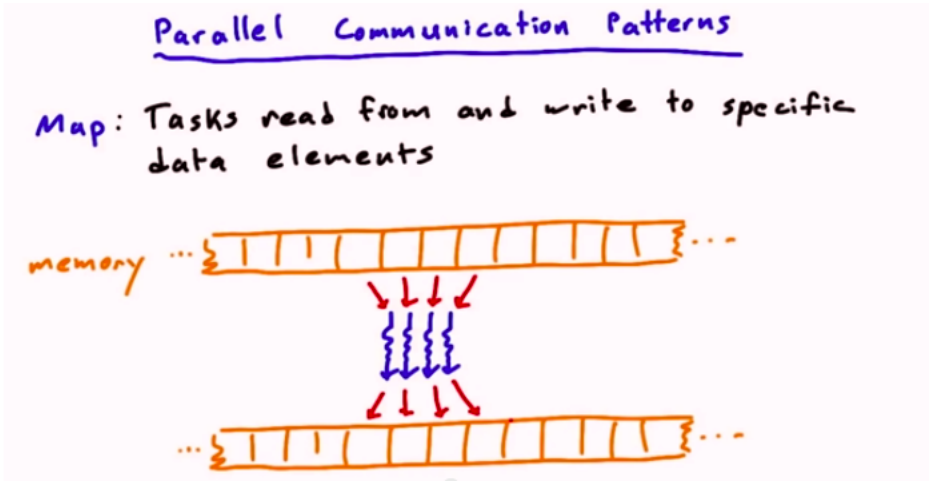
CS344 Introduction to Parallel Programming

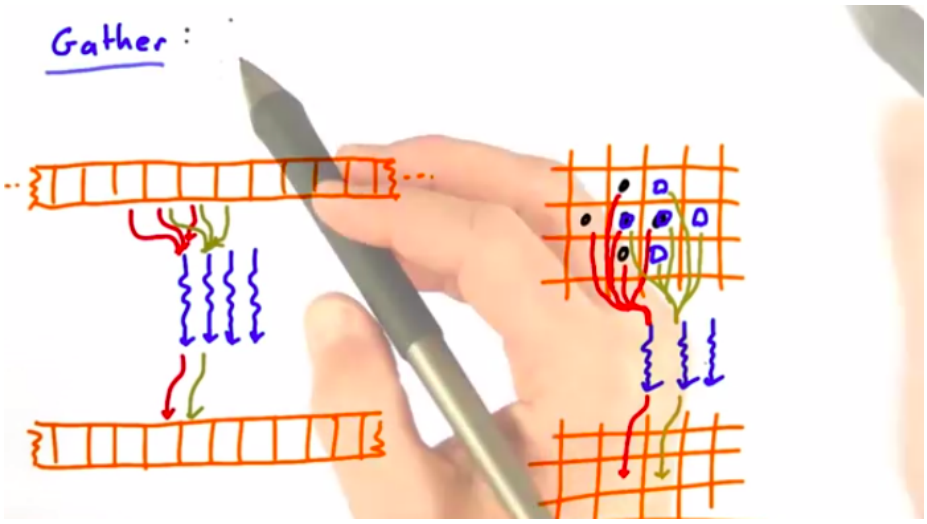
Lesson 2: GPU Hardware and Parallel Communication Patterns

L2-2.2-Communication Patterns



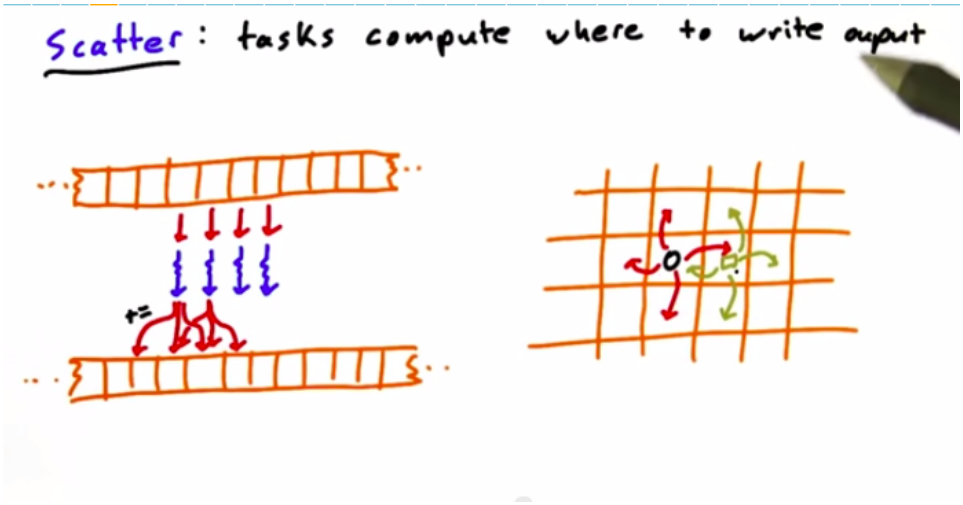
L2-2.3-Map and Gather



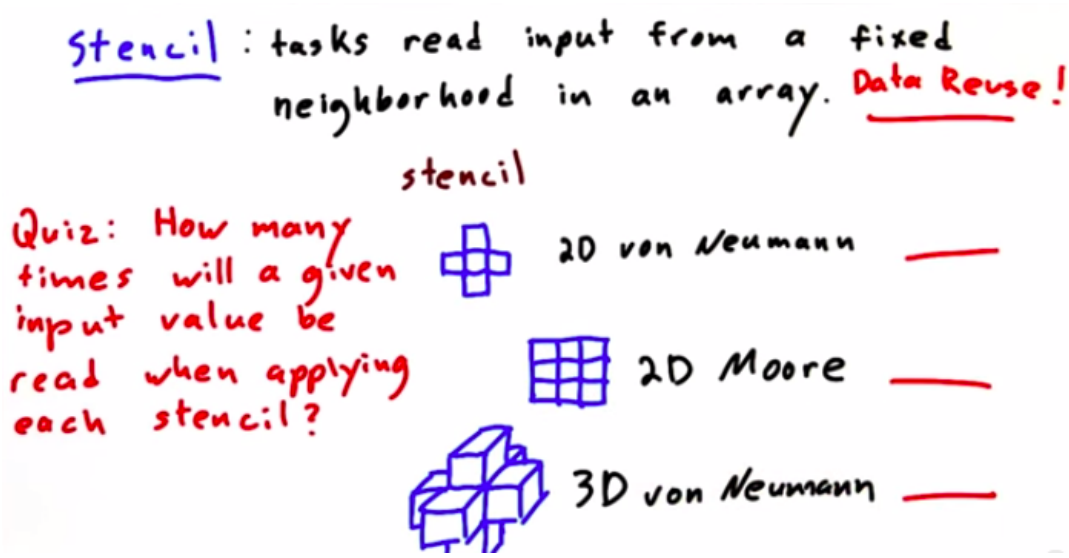


[CS344 Intro to Parallel Programming, Lesson 2]

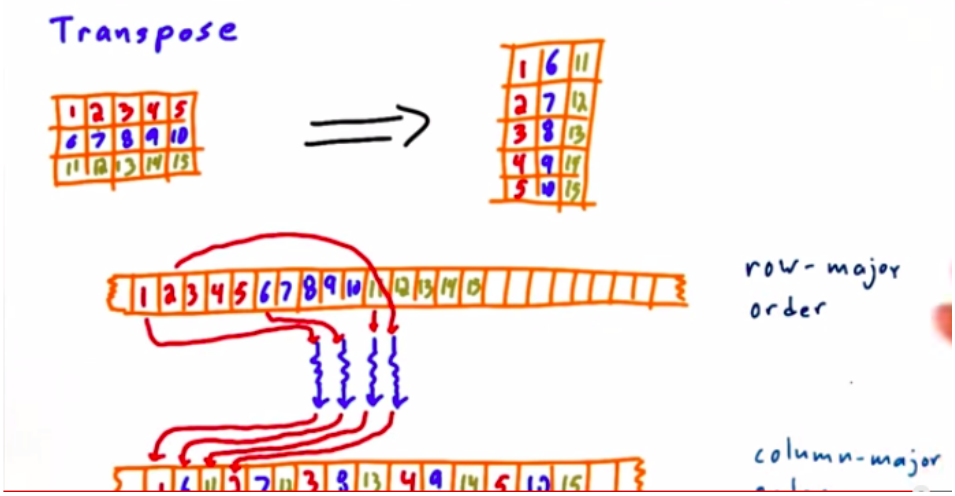
L2-2.4- Scatter Quiz



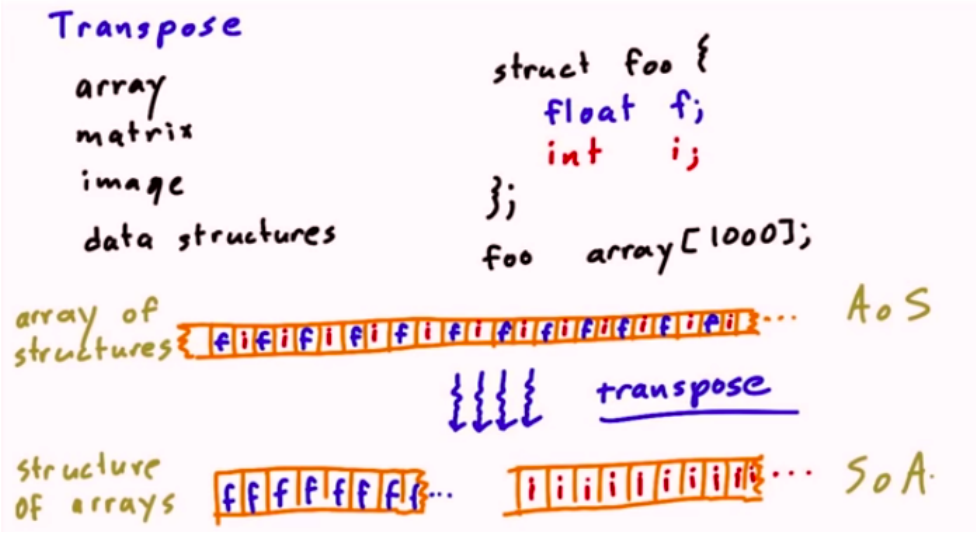
L2-2.5-Stencil-Quiz



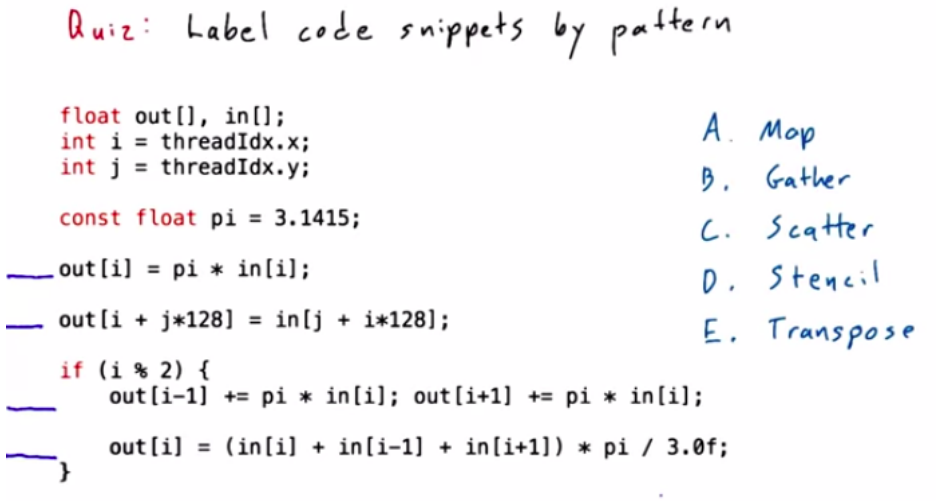
L2-2.6-Transpose Part 1



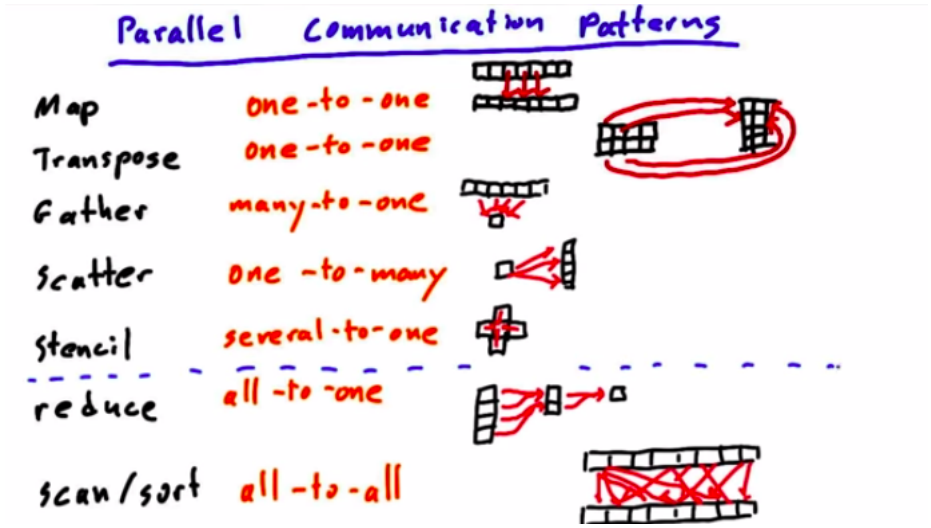
L2-2.7-Transpose Part 2



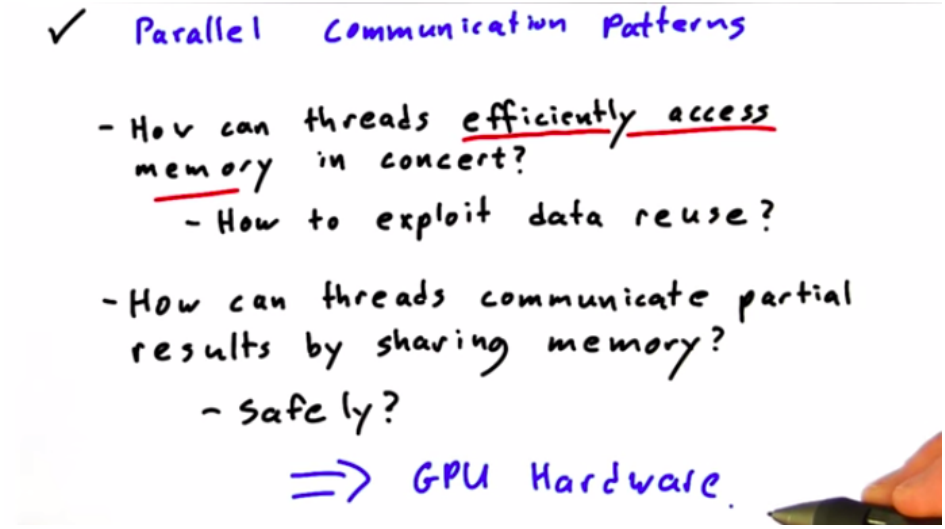
L2-2.8-What Kind of Communication Pattern - Quiz



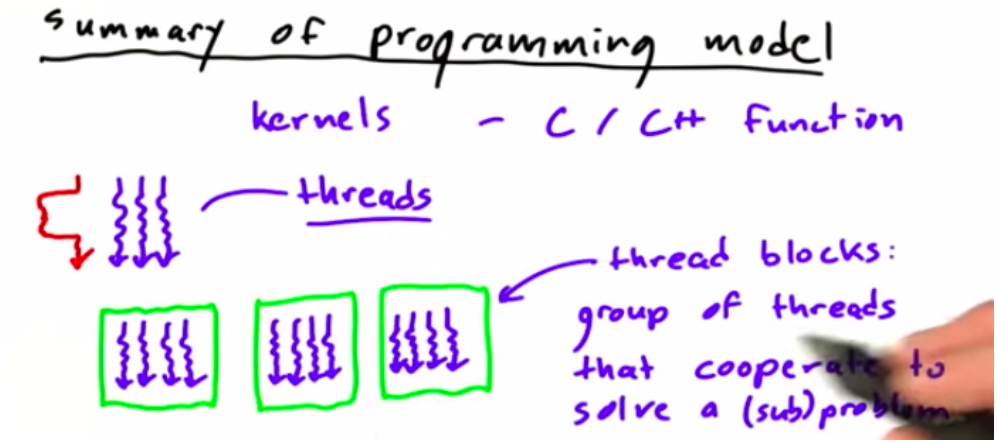
L2-2.9 Parallel Communication Patterns Recap

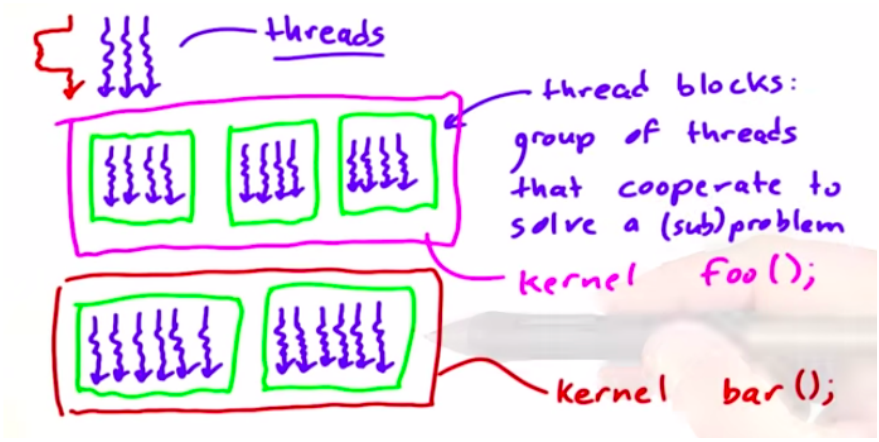


L2-2.10-Let Us Talk About GPU Hardware

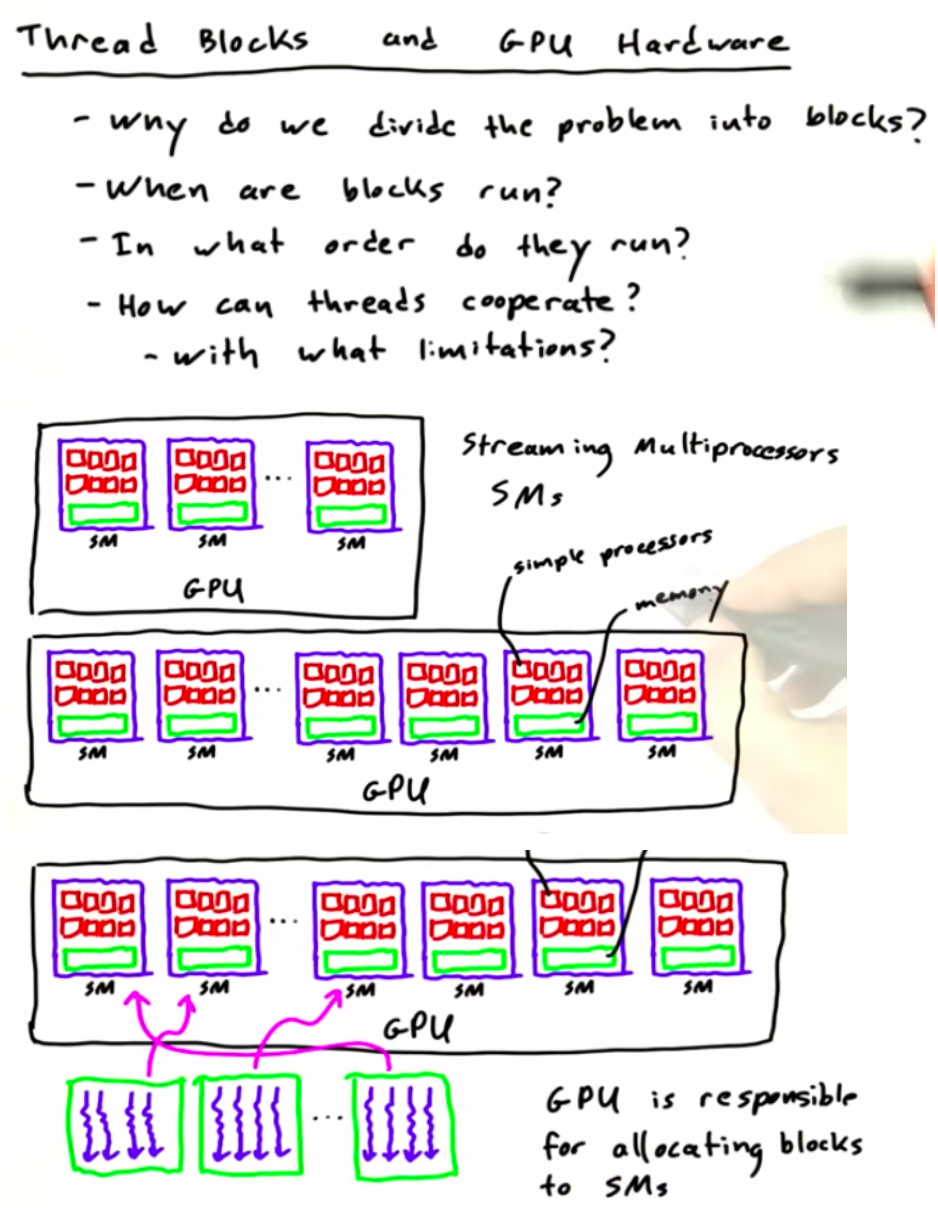


L2-2.11-Programmer View of the GPU



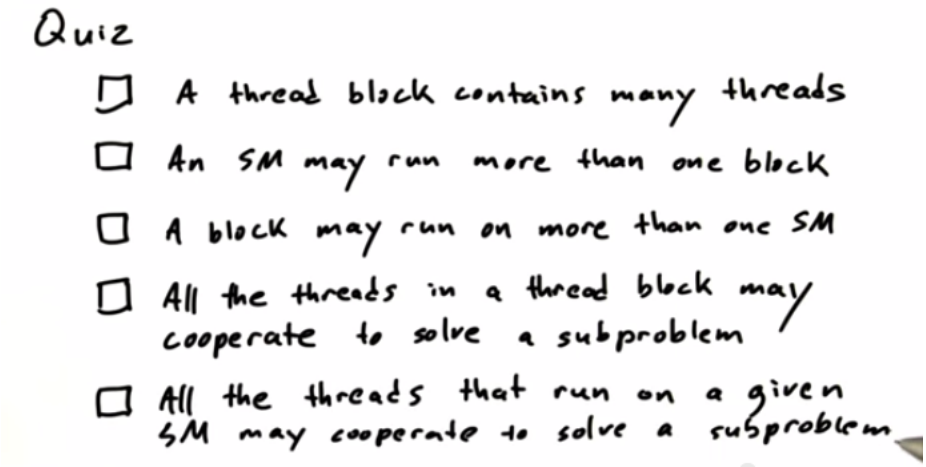


L2-2.12-Thread Blocks And GPU Hardware

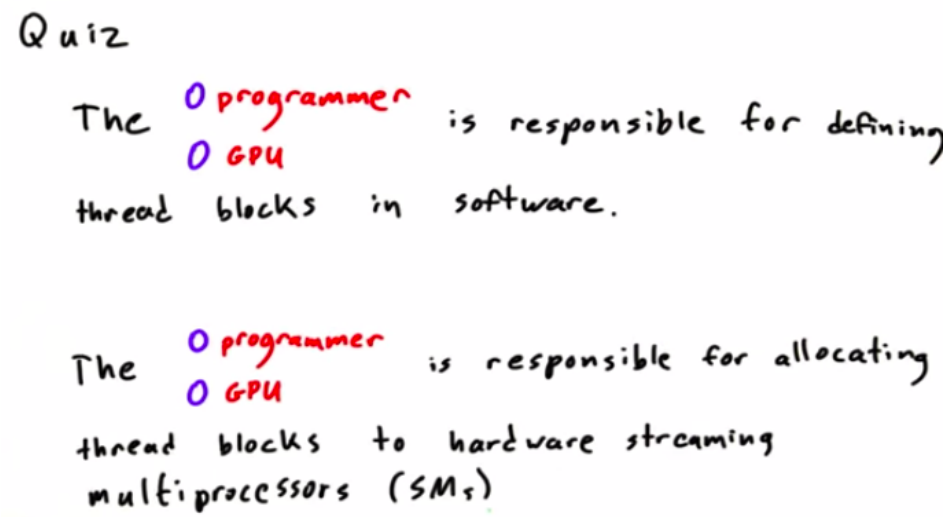


[CS344 Intro to Parallel Programming, Lesson 2]

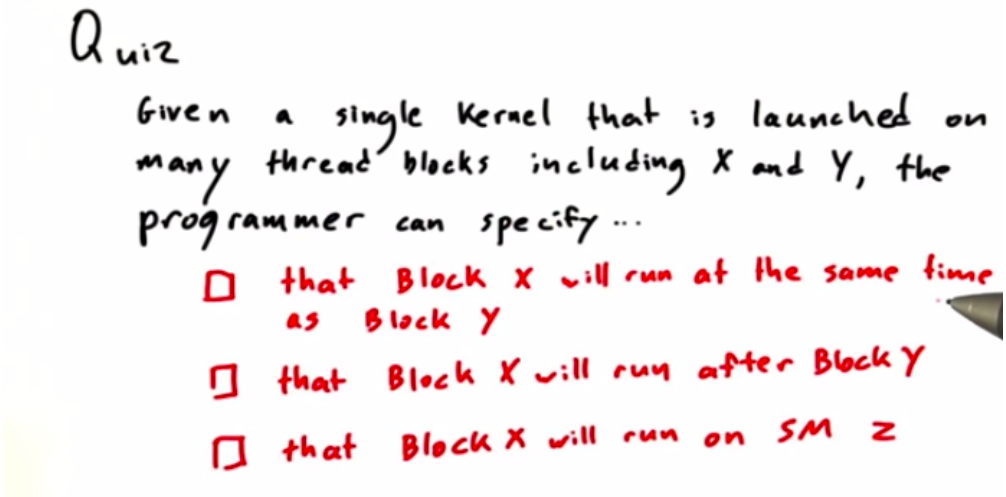
L2-2.13-Thread and Blocks - Quiz



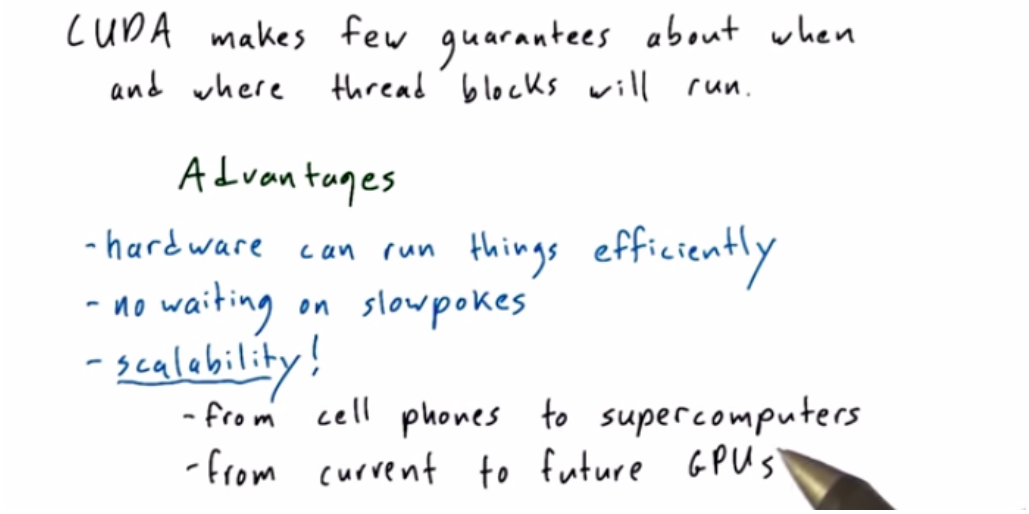
L2-2.14-Another Quiz on Thread Blocks

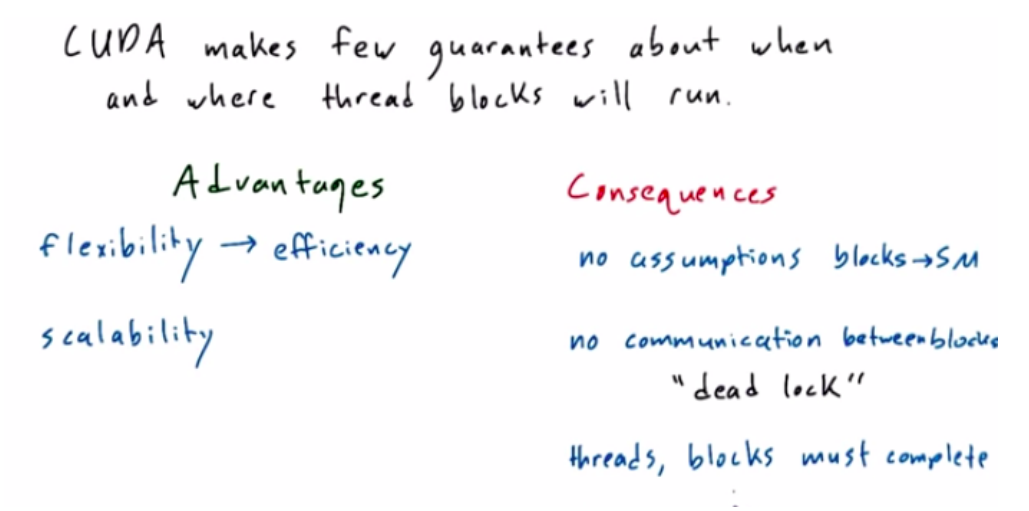


L2-2.15-What Can The Programmer Specify-Quiz

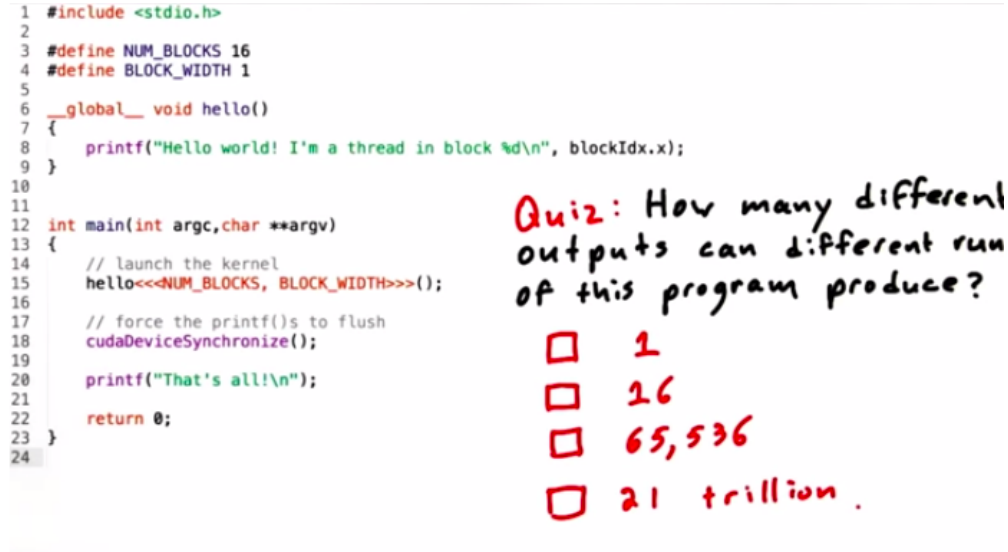


L2-2.16-CUDA Makes Few Guarantees About Thread Block





L2-2.17-A Thread Block Programming Example (-Quiz?)



```
#include <stdio.h>

#define NUM_BLOCKS 16
#define BLOCK_WIDTH 1

__global__ void hello()
{
    printf("Hello world! I'm a thread in block %d\n", blockIdx.x);
}


int main(int argc,char **argv)
{
    // launch the kernel
    hello<<<NUM_BLOCKS, BLOCK_WIDTH>>>();

    // force the printf()s to flush
    cudaDeviceSynchronize();

    printf("That's all!\n");

    return 0;
}
```

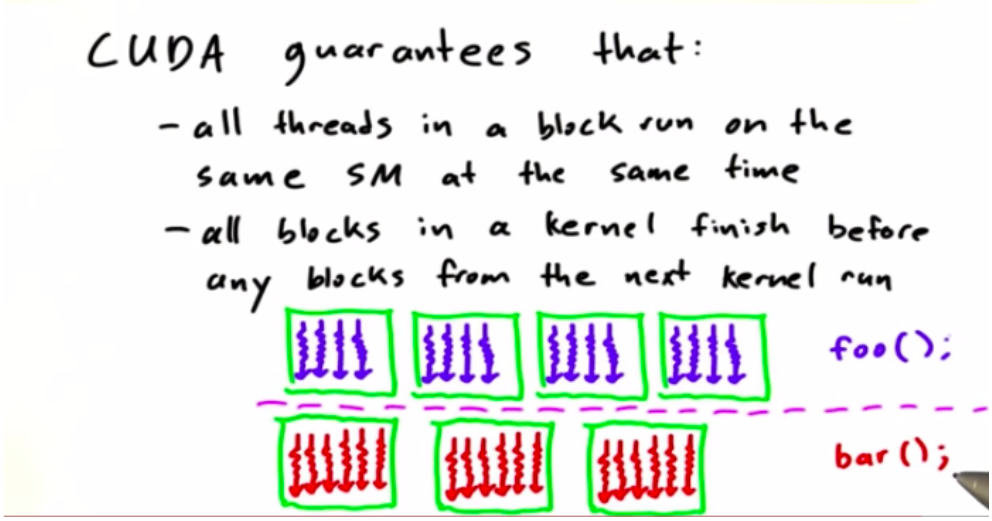
Quiz: How many different outputs can different run of this program produce?

- ☐ 1
- ☐ 16
- ☐ 65,536
- ☐ 21 trillion.

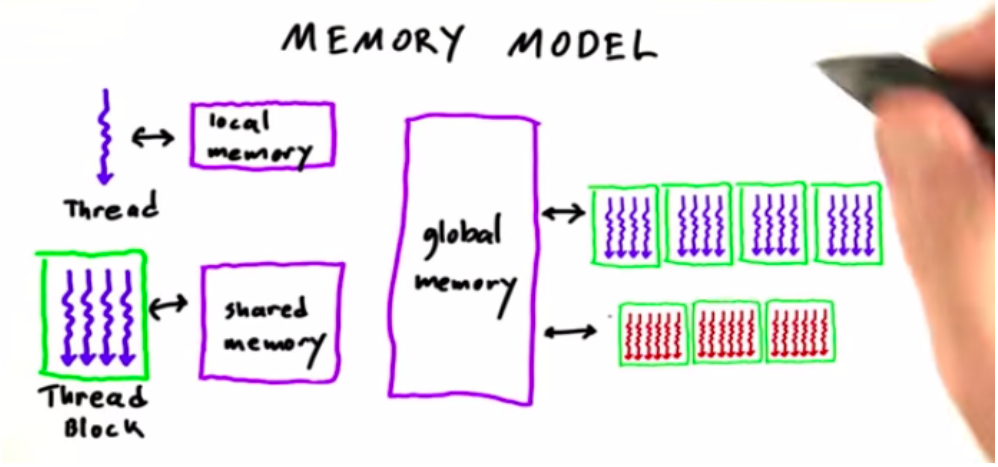
L2-2.18-A Programming Example on Blocks

```
#include <stdio.h>

#define NUM_BLOCKS 16
#define BLOCK_WIDTH 1

__global__ void hello()
{
    printf("Hello world! I'm a thread in block %d\n", blockIdx.x);
}


int main(int argc,char **argv)
{
    // launch the kernel
    hello<<<NUM_BLOCKS, BLOCK_WIDTH>>>();

    // force the printf()s to flush
    cudaDeviceSynchronize();

    printf("That's all!\n");

    return 0;
}
```

L2-2.19-What Does CUDA Guarantee



L2-2.20-GPU Memory Model



[CS344 Intro to Parallel Programming, Lesson 2]

L2-2.21-A Quiz about GPU Memory Model

Quiz

- ☐ All threads from a block can access the same variable in that blocks shared memory
- ☐ Threads from two different blocks can access the same variable in global memory
- ☐ Threads from different blocks have their own copy of local variables in local memory
- ☐ Threads from the same block have their own copy of local variables in local memory

L2-2.22-Synchronization-Barrier

Synchronization

threads can access each other's results through shared and global memory

⇒ they can work together!

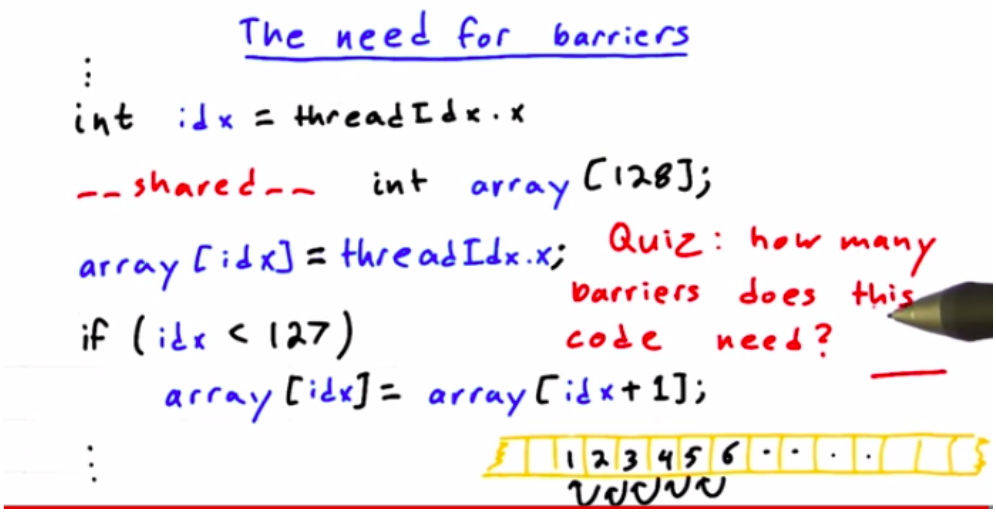
Danger: what if a thread reads a result before another thread writes it?

Threads need to synchronize

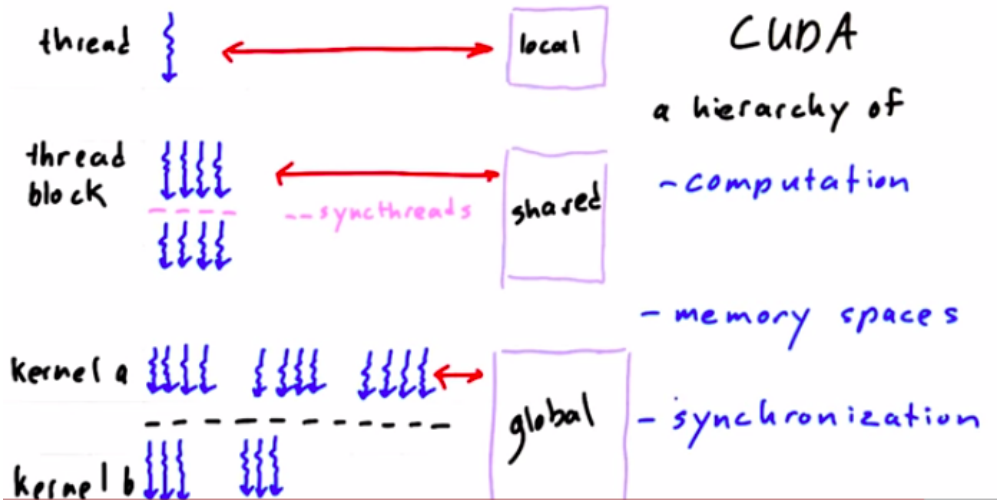
Barrier - point in the program where threads stop and wait.

When all threads have reached the barrier, they can proceed.

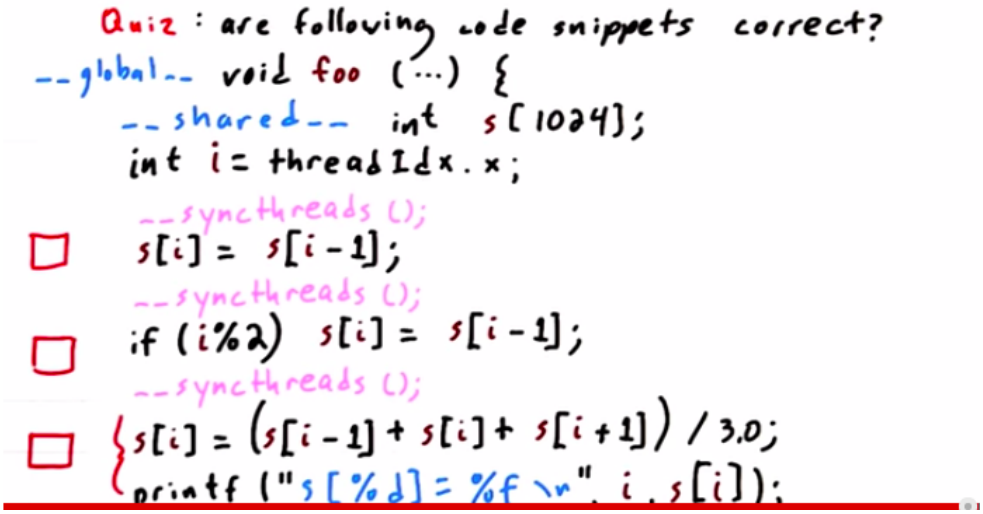
L2-2.23-The Need For Barriers-Quiz



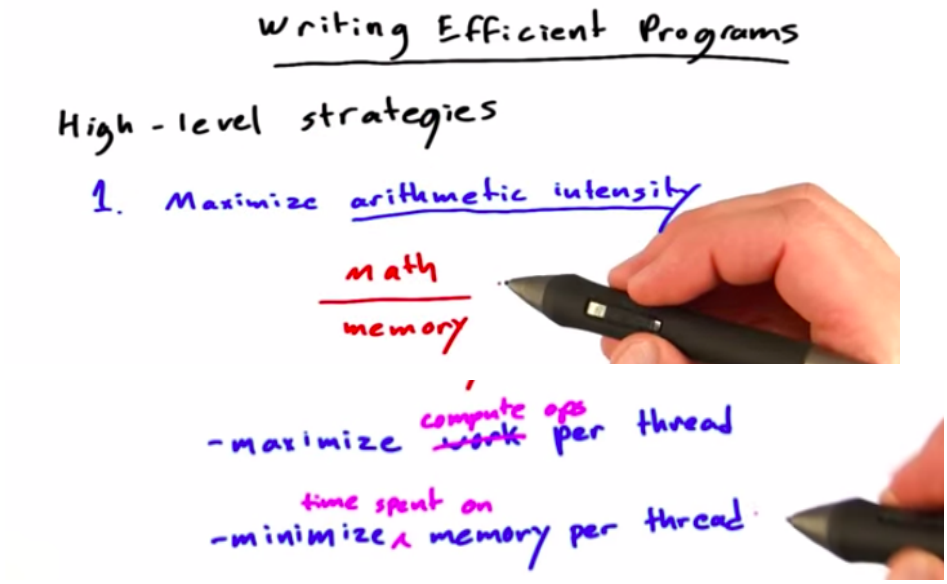
L2-2.24-Programming Model



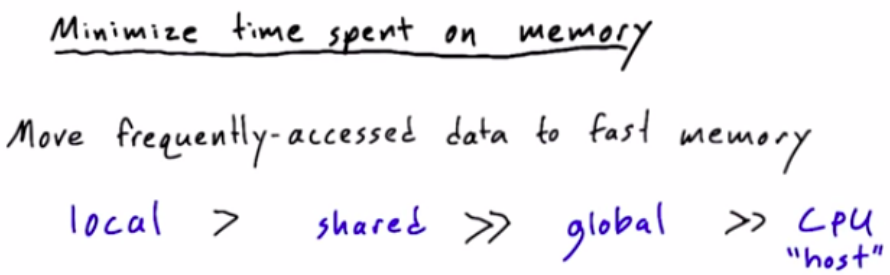
L2-2.25-A Quiz on Synchronization - Quiz



L2-2.26-Writing Efficient Programs



L2-2.27-Minimize Time Spent On Memory

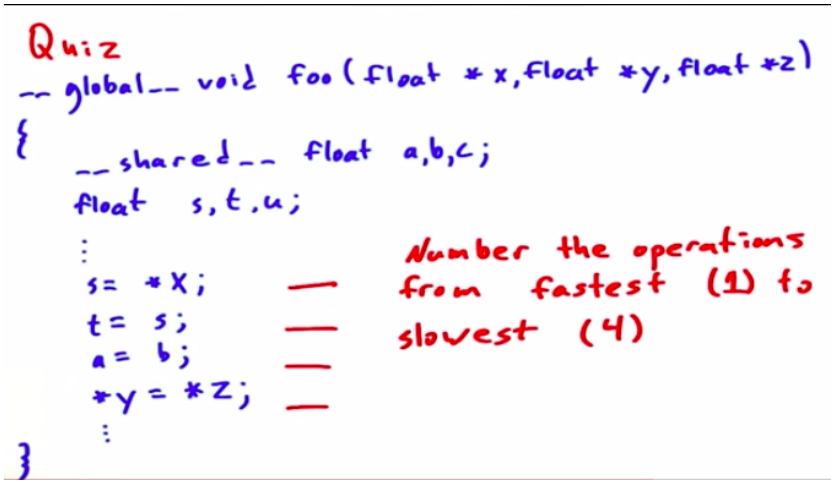


L2-2.28-Global Memory

L2-2.29-Shared Memory : Note Instructor Notes!

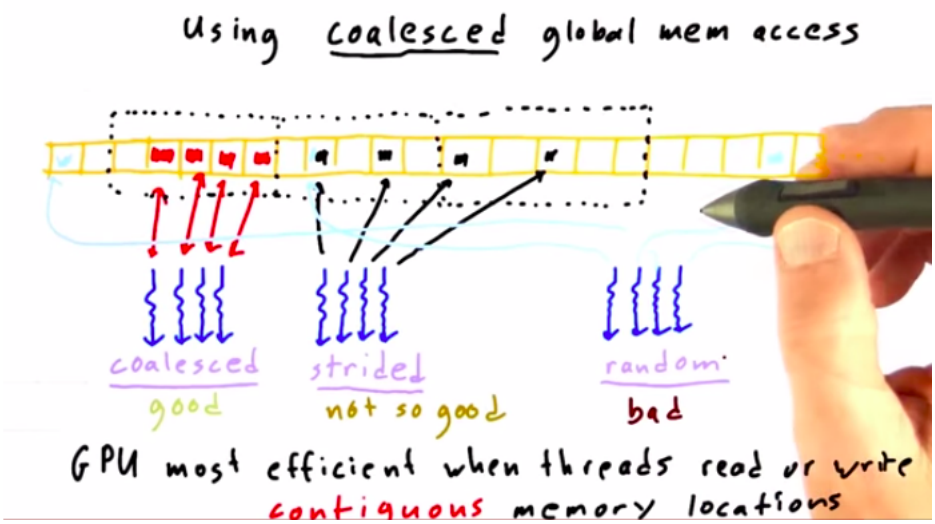
Code Available:
https://github.com/udacity/cs344/blob/master/Unit2%20Code%20Snippets/memory.cu

L2-2.30-Quiz on Memory Access

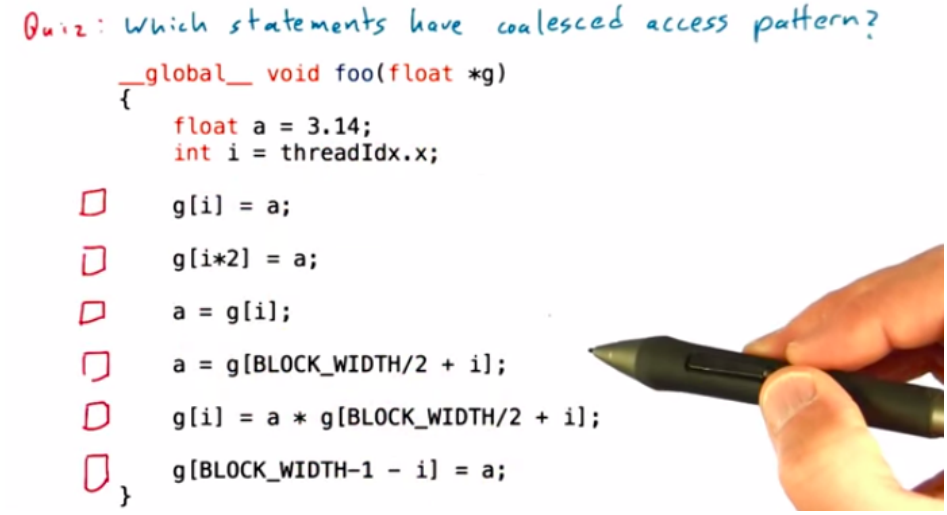


[CS344 Intro to Parallel Programming, Lesson 2]

L2-2.31-Coalesce Memory Access



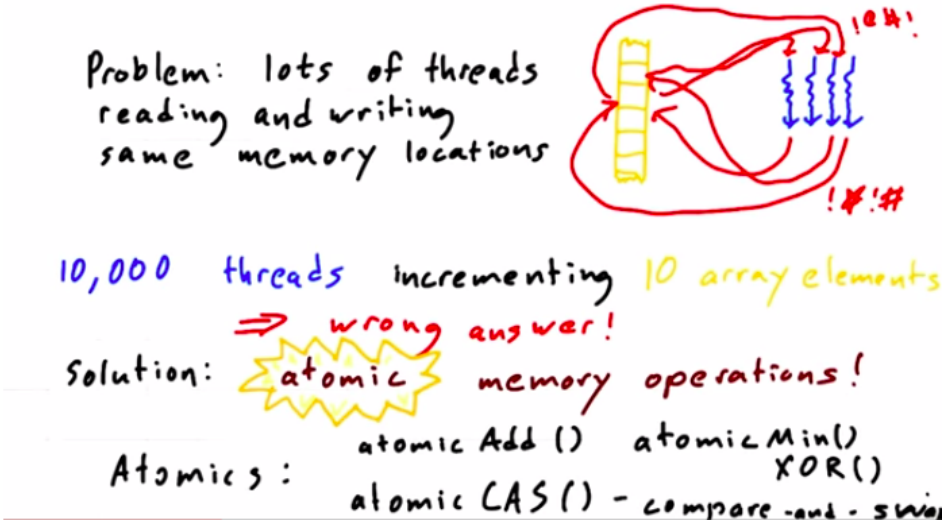
L2-2.32-A Quiz on Coalescing Memory Access



L2-2.33-A related Problem Part 1

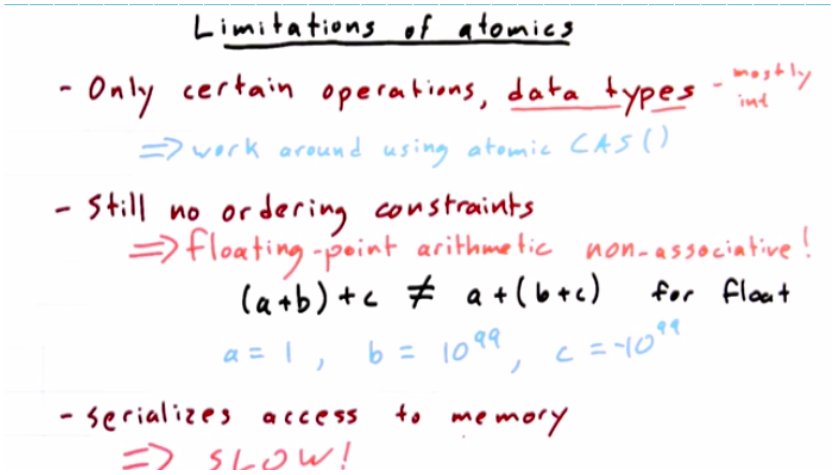
L2-2.34-A related Problem Part 2
code Available:
https://github.com/udacity/cs344/blob/master/Unit2%20Code%20Snippets/atomics.cu

L2-2.35-Atomic Memory Operations



code Available:
https://github.com/udacity/cs344/blob/master/Unit2%20Code%20Snippets/atomics.cu

L2-2.36-Limitations of Atomic Memory Operations

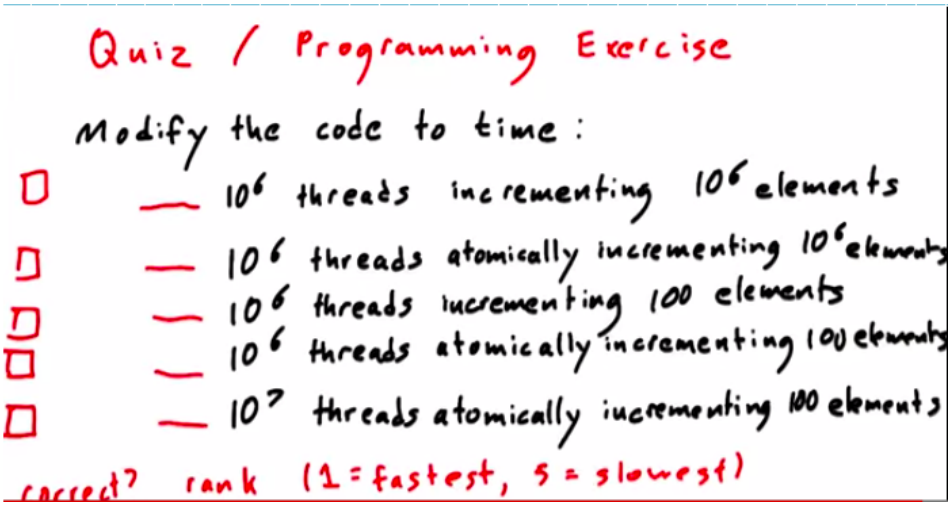


L2-2.37-Timing Atomic Operations

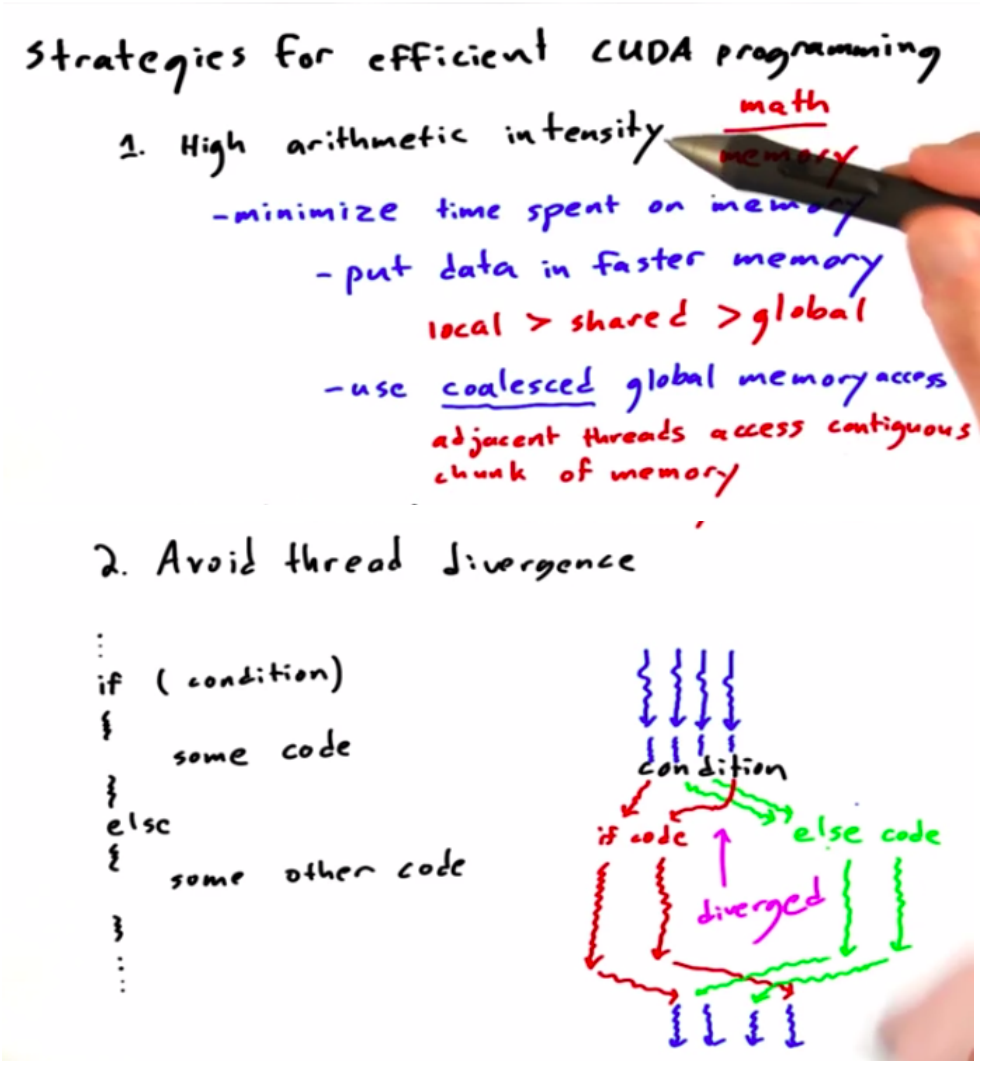
Code Available:
https://github.com/udacity/cs344/blob/master/Unit2%20Code%20Snippets/atomics.cu

L2-2.38-Let Us Time Some Code-Quiz

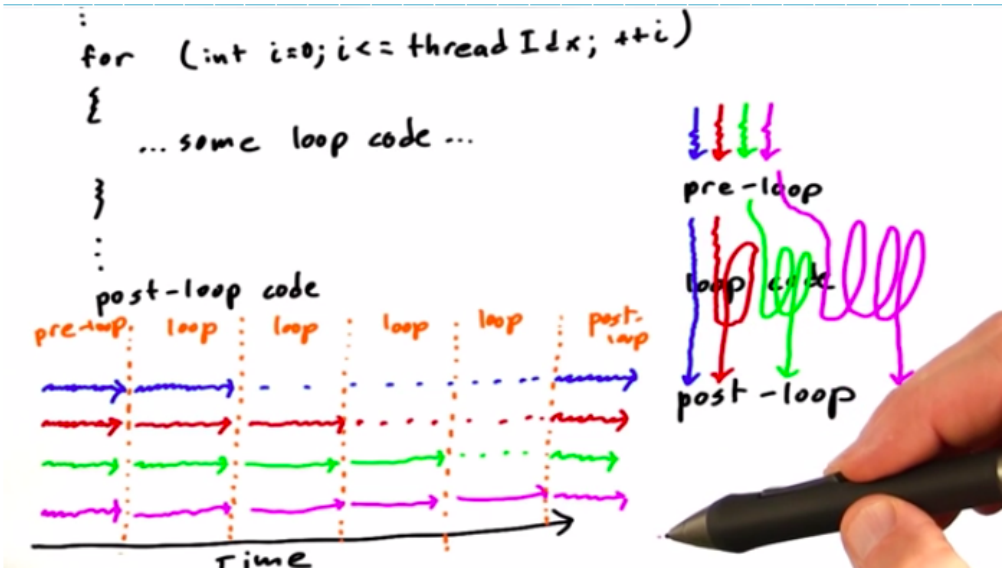
same Code Available:
https://github.com/udacity/cs344/blob/master/Unit2%20Code%20Snippets/atomics.cu



L2-2.39-High Arithmetic Intensity



L2-2.40-Thread Divergence



L2-2.41 Summary of Unit 2

L2-2.43-Problem Set #2

[CS344 Intro to Parallel Programming, Lesson 2]

Problem Set #2

9	1	3	4	8
10	4	6	0	10
2	4	1	2	7
2	3	6		1
8	0	0	3	9

Avg Intensity Value =

$\frac{1}{9}(4 + 10 + 4 + 6 + 2 + 1 + 2 + 3)$

$(W_1 \cdot 4) + (W_2 \cdot 10) + (W_3 \cdot 4) + (W_4 \cdot 6) \ldots$

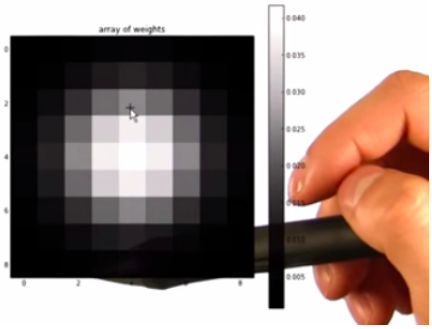
Blur Examples

Weighted

Unweighted.

Weights

.1	.1	.1	.1
.1	.3	.3	.1
.1	.3	.3	.1
.1	.1	.1	.1



What You Need To Do

1. Write the blur kernel
2. Write Kernel to Separate Color Image to R, G, B channels
3. Allocate memory for the filter
4. Set Grid and Block Size.

[CS344 Intro to Parallel Programming, Lesson 2]