

Лекция 4

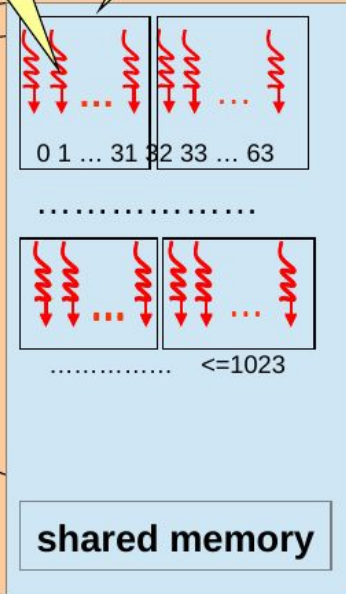
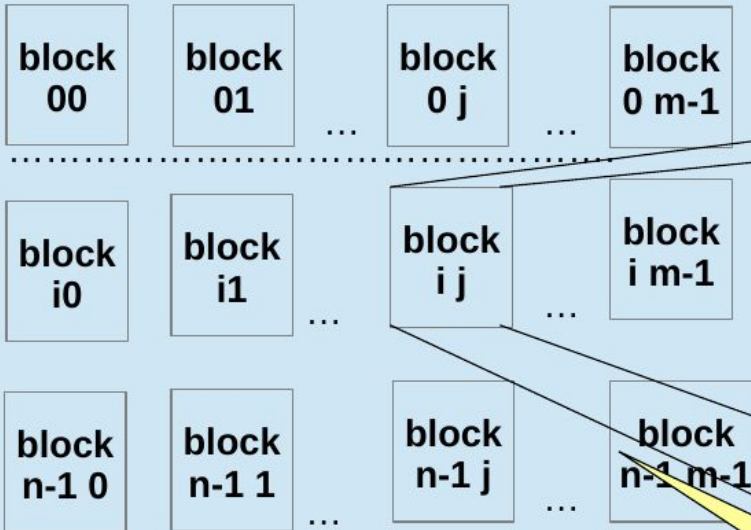
- Объединение нитей в блоки и варпы.
- Оптимальная конфигурация нитей.
- Иерархия памяти.

device

grid

thread

warp



$n \times m \leq 65536$

Global memory

копирование device -> host

```
#include <stdio.h>
#include <stdlib.h>

global void gShowIdx() {
int idx = blockIdx.x * blockDim.x + threadIdx.x;
int warp_idx = threadIdx.x / warpSize;
int lane_idx = threadIdx.x % warpSize;
printf(" %5d\t%5d\t %2d\t%2d\n",
        idx, blockIdx.x, warp_idx, lane_idx);
}
```

```
int main(int argc, char** argv){
    if(argc < 3){
        fprintf(stderr,
            "USAGE: <prog> <threads_per_block> <num_of_blocks>\n");
        return -1;
    }

    int threads_per_block=atoi(argv[1]);
    int num_of_blocks=atoi(argv[2]);

    gShowIdx<<<num_of_blocks, threads_per_block>>>();
    cudaDeviceSynchronize();

    return 0;
}
```

```
./lab4a 64 2 | sort -g -k1,1 -k2,2  
-k4,4 > stat.txt
```

<i>idx</i>	<i>blk</i>	<i>wrp</i>	<i>lane</i>
------------	------------	------------	-------------

0	0	0	0
---	---	---	---

1	0	0	1
---	---	---	---

2	0	0	2
---	---	---	---

30	0	0	30
----	---	---	----

31	0	0	31
----	---	---	----

32	0	1	0
----	---	---	---

33	0	1	1
----	---	---	---

34	0	1	2
----	---	---	---

62	0	1	30
----	---	---	----

63	0	1	31
----	---	---	----

64	1	0	0
----	---	---	---

65	1	0	1
----	---	---	---

95	1	0	31
----	---	---	----

96	1	1	0
----	---	---	---

97	1	1	1
----	---	---	---

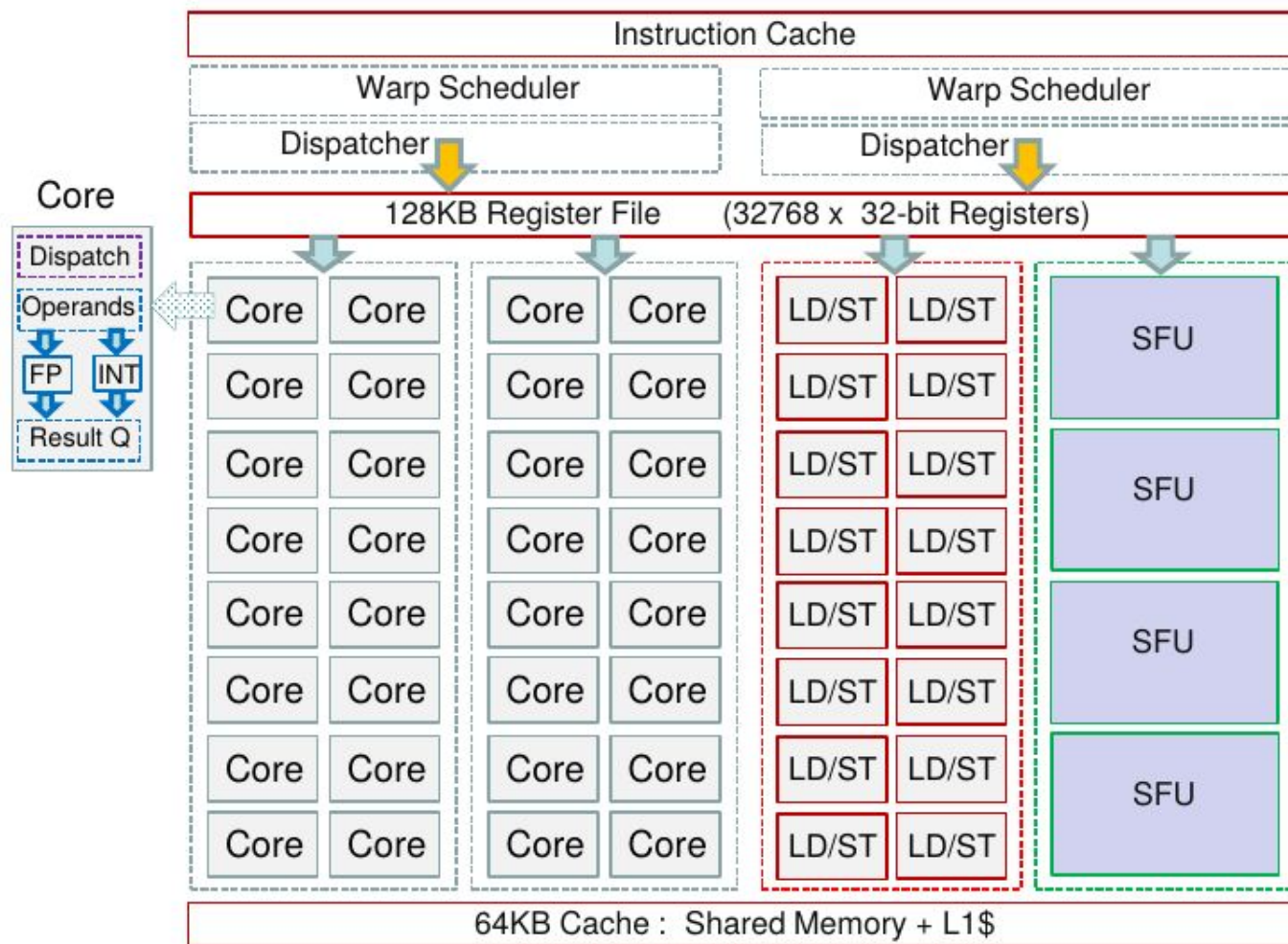
127	1	1	31
-----	---	---	----

Оптимальное количество нитей в блоках (сокращение латентности)

Преимущества
перекрывания/многозадачности:



Кол-во варпов $\% 32 == 0 \ \&\& \ \text{Кол-во варпов} / 32 > 1$
Кол-во блоков \geq Кол-во мультипроцессоров



```
Detected 1 CUDA Capable device(s)
Device 0: "NVIDIA GeForce RTX 2060"
  CUDA Driver Version / Runtime Version      12.0 / 11.1
  CUDA Capability Major/Minor version number: 7.5
.....
  (30) Multiprocessors, ( 64) CUDA Cores/MP: 1920 CUDA Cores
.....
  Memory Clock rate:                        7001 Mhz
  Memory Bus Width:                         192-bit
  L2 Cache Size:                            3145728 bytes
.....
  Total amount of constant memory:           65536 bytes
  Total amount of shared memory per block:   49152 bytes
  Total shared memory per multiprocessor:    65536 bytes
  Total number of registers available per block: 65536
  Warp size:                                32
  Maximum number of threads per multiprocessor: 1024
  Maximum number of threads per block:      1024
```



```
Detected 2 CUDA Capable device(s)
Device 0: "GeForce GTX 1050"
  CUDA Driver Version / Runtime Version          9.1 / 9.1
  CUDA Capability Major/Minor version number:    6.1
.....
( 5) Multiprocessors, (128) CUDA Cores/MP:      640 CUDA Cores
.....
Memory Clock rate:                             3504 Mhz
Memory Bus Width:                              128-bit
L2 Cache Size:                                 1048576 bytes
.....
Total amount of constant memory:                65536 bytes
Total amount of shared memory per block:        49152 bytes
Total number of registers available per block:  65536
Warp size:                                      32
Maximum number of threads per multiprocessor:  2048
Maximum number of threads per block:            1024
.....
```

<https://docs.nvidia.com/cuda/archive/11.2.0/cuda-c-programming-guide/index.html#compute-capabilities>

Table 15. Technical Specifications per Compute Capability

	Compute Capability												
Technical Specifications	3.5	3.7	5.0	5.2	5.3	6.0	6.1	6.2	7.0	7.2	7.5	8.0	8.6
Warp size	32												
Maximum number of resident blocks per SM	16		32								16	32	16
Maximum number of resident warps per SM	64										32	64	48
Maximum number of resident threads per SM	2048										1024	2048	1536
Number of 32-bit registers per SM	64 K	128 K	64 K										
Maximum number of 32-bit registers per thread block	64 K				32 K	64 K		32 K			64 K		
Maximum number of 32-bit registers per thread	255												

```
global void gSum(int* a, int *b){  
    int i=threadIdx.x+blockIdx.x*blockDim.x;  
    a[i]+=b[i];  
}
```

```
int N=1<<atoi(argv[1]);  
int num_threads=atoi(argv[2]);  
int num_blocks=N/num_threads;
```

```
gSum<<<num_blocks, num_threads>>>(a,b);  
cudaDeviceSynchronize();  
CUDA_CHECK_RETURN(cudaGetLastError());
```

```
/Lab4 # ncu --target-processes all -k gSum ./lab4 20 32
```

gSum(int *, int *), 2023-Feb-20 17:05:19, Context 1, Stream 7

Section: GPU Speed Of Light Throughput

Duration **usecond** **69.38**

Section: Occupancy

Theoretical Occupancy % **50**
Achieved Occupancy % **33.63**

..... . .

```
/Lab4 # ncu --target-processes all -k gSum ./lab4 20 64
```

gSum(int *, int *), 2023-Feb-20 17:05:19, Context 1, Stream 7

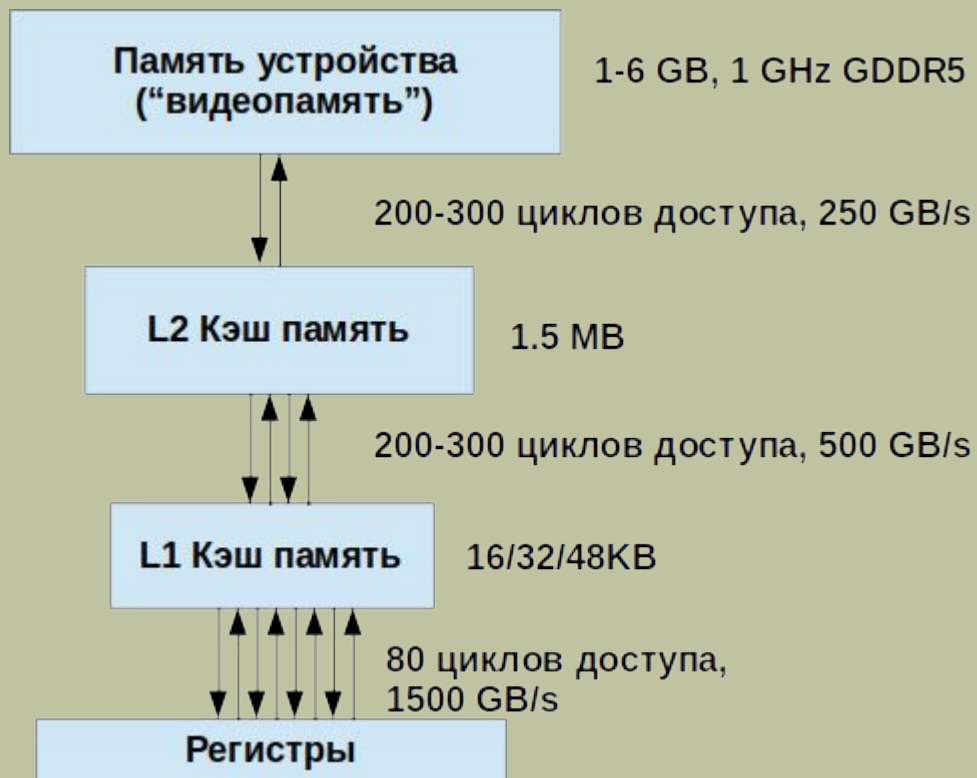
Section: GPU Speed Of Light Throughput

Duration **usecond** **42.78**

Section: Occupancy

Theoretical Occupancy % **100**
Achieved Occupancy % **79.17**

..... . .



Device

DRAM

Local

Global

Constant

Texture

To Host

GPU

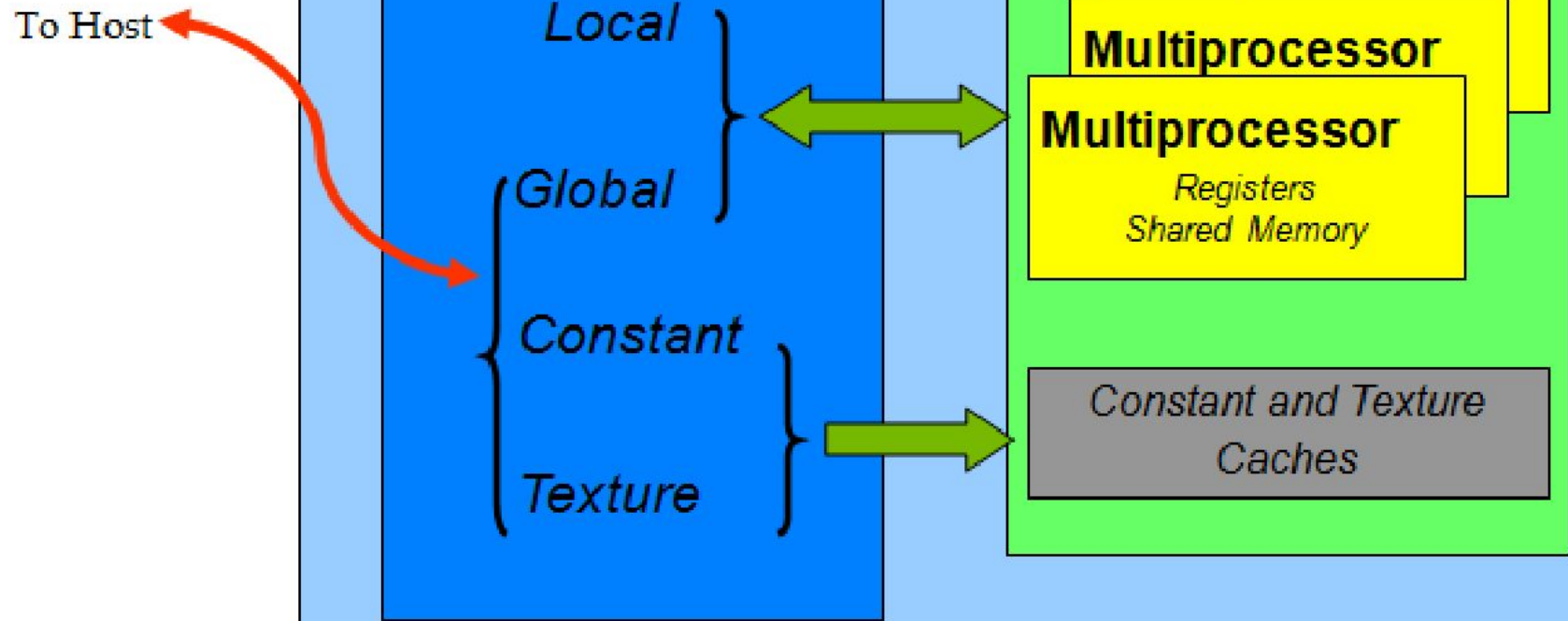
Multiprocessor

Multiprocessor

Multiprocessor

Registers
Shared Memory

Constant and Texture
Caches



```
#define DUMMY_LENGTH 102
```

```
__global__ void gInit(int* a, int* b){  
    int i=threadIdx.x+blockIdx.x*blockDim.x;  
    int dummy[DUMMY_LENGTH];  
    for(int j=0;j<DUMMY_LENGTH;j++)  
        dummy[j]=j%2+1;  
  
    a[i]=dummy[DUMMY_LENGTH-2]*i;  
    b[i]=dummy[DUMMY_LENGTH-1]*i+1;  
}
```

быдло-
код 🤪


```
/PGP-2023> nvprof -m local_memory_overhead ./lab4 20 32
```

Invocations	Metric Name	Metric Description	Min	Max	Avg
Device "GeForce GTX 1050 (0)"					
Kernel: gInit(int*, int*)					
1	local_memory_overhead	Local Memory Overhead	98.11%	98.11%	98.11%
Kernel: gSum(int*, int*)					
1	local_memory_overhead	Local Memory Overhead	0.00%	0.00%	0.00%

Type	Time (%)	Time	Calls	Min	Max
			Avg		Name
GPU activities:	83.15%	4.3196ms	1		
			4.3196ms	4.3196ms	4.3196ms
					gInit(int*, int*)
	4.67%	242.60us	1		
			242.60us	242.60us	242.60us
					gSum(int*, int*)

```
#define DUMMY_LENGTH 101
```

```
/PGP-2023> nvprof -m local_memory_overhead ./lab4 20 32
```

Invocations	Metric Name	Metric Description	Min	Max	Avg
Device "GeForce GTX 1050 (0)"					
Kernel: gInit(int*, int*)					
1	local_memory_overhead	Local Memory Overhead	0.00%	0.00%	0.00%
Kernel: gSum(int*, int*)					
1	local_memory_overhead	Local Memory Overhead	0.00%	0.00%	0.00%

Type	Time (%)	Time	Calls	Min	Max
			Avg	Name	
GPU activities:	16.67%	173.67us	1		
			173.67us	173.67us	173.67us
				gInit(int*, int*)	
	23.26%	242.25us	1		
			242.25us	242.25us	242.25us
				gSum(int*, int*)	