Occupancy

- Previous video: "Parameters required to compare between different Architectures."
  - Memory bandwidth and instructions throughput.
- The real issue is addressing only the theoretical occupancy.

Occupancy

- Occupancy is a measure of the utilization of the resources in a GPU.
- Occupancy is categorized into two distinct types:
  - Theoretical occupancy: the ideal case.
  - Achieved occupancy: the actual usage of the GPU's resources

Occupancy

- The theoretical occupancy calculation:

$$\frac{warp\ used\ in\ a\ kernel}{\max warps\ per\ SM}$$

- For RTX 3090:
  - Maximum threads/SM are : 1536.
  - Maximum warps/SM are : 48 warps.

Occupancy

- The theoretical occupancy calculation:

$$\frac{warp\ used\ in\ a\ kernel}{\max warps\ per\ SM}$$

- Example 1:
  - Assuming a kernel utilizes 48 warps.
- Example 2:
  - Assuming a kernel utilizes 16 warps.

Occupancy

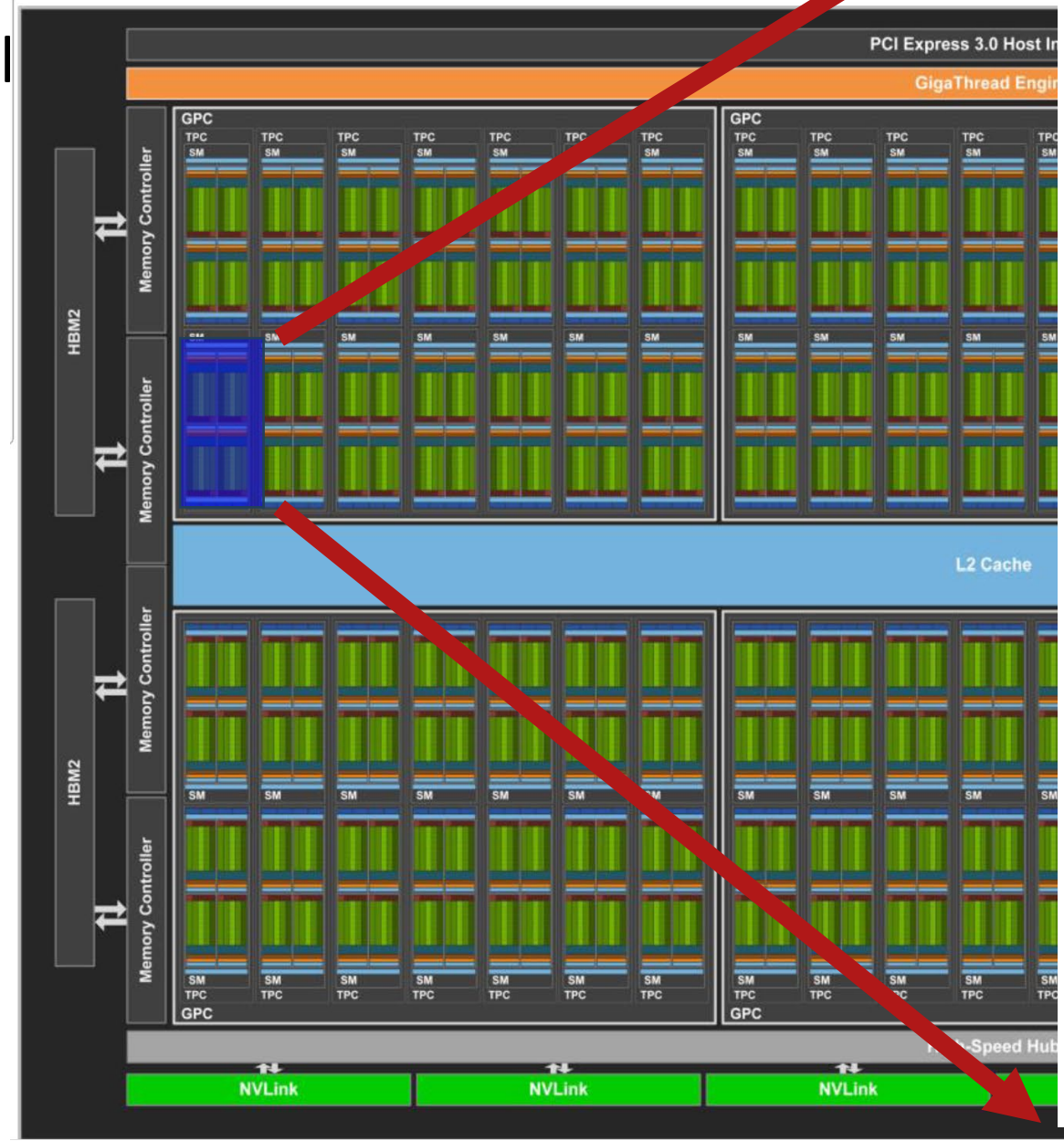
- The theoretical occupancy summary:

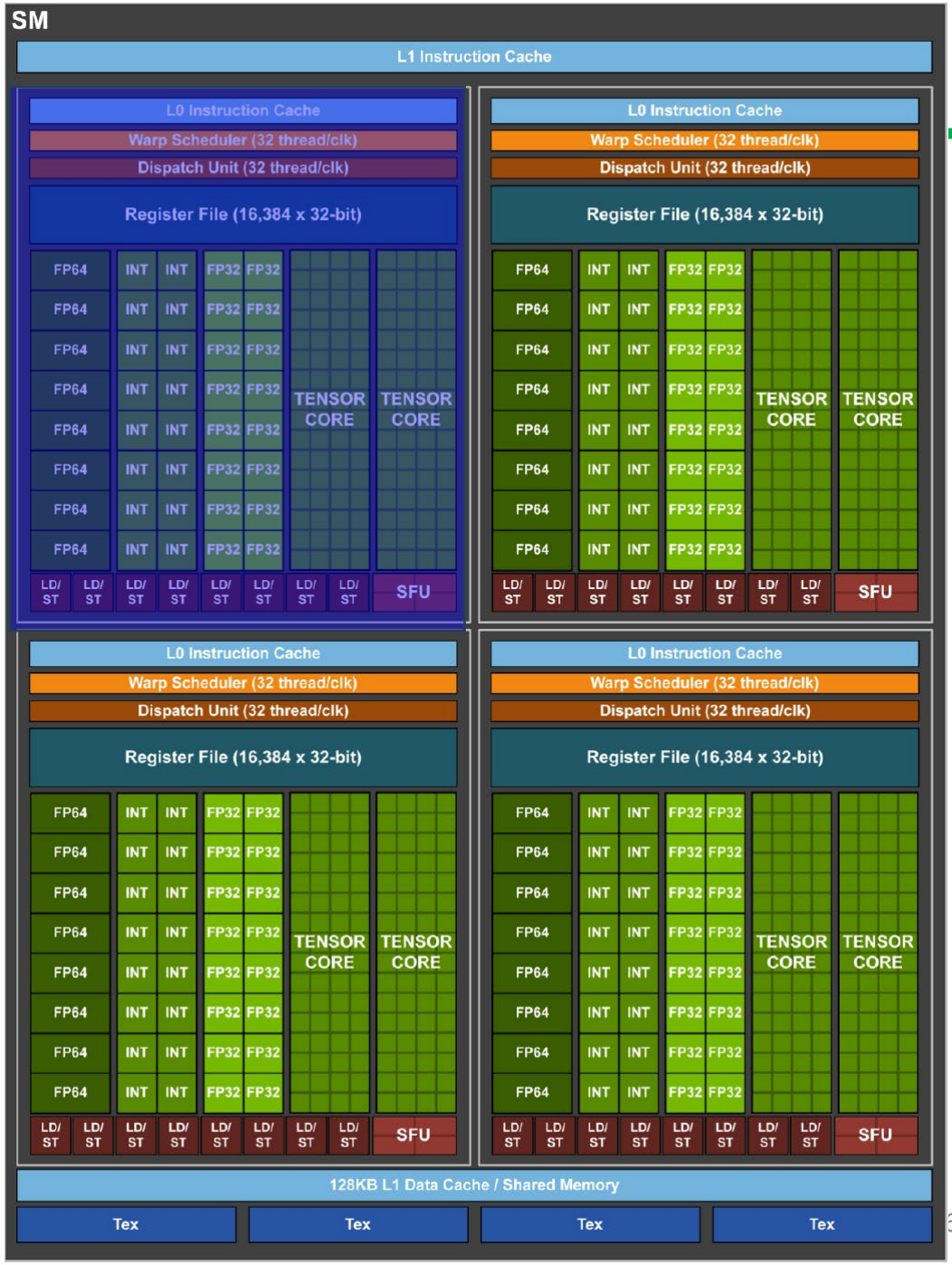
$$\frac{warp\ used\ in\ a\ kernel}{\max warps\ per\ SM}$$

  - Refers to the ideal circumstances.
  - Optimal conditions where there are enough independent tasks.
    - Without being bottlenecked by memory, computation, dependency.

Occupancy

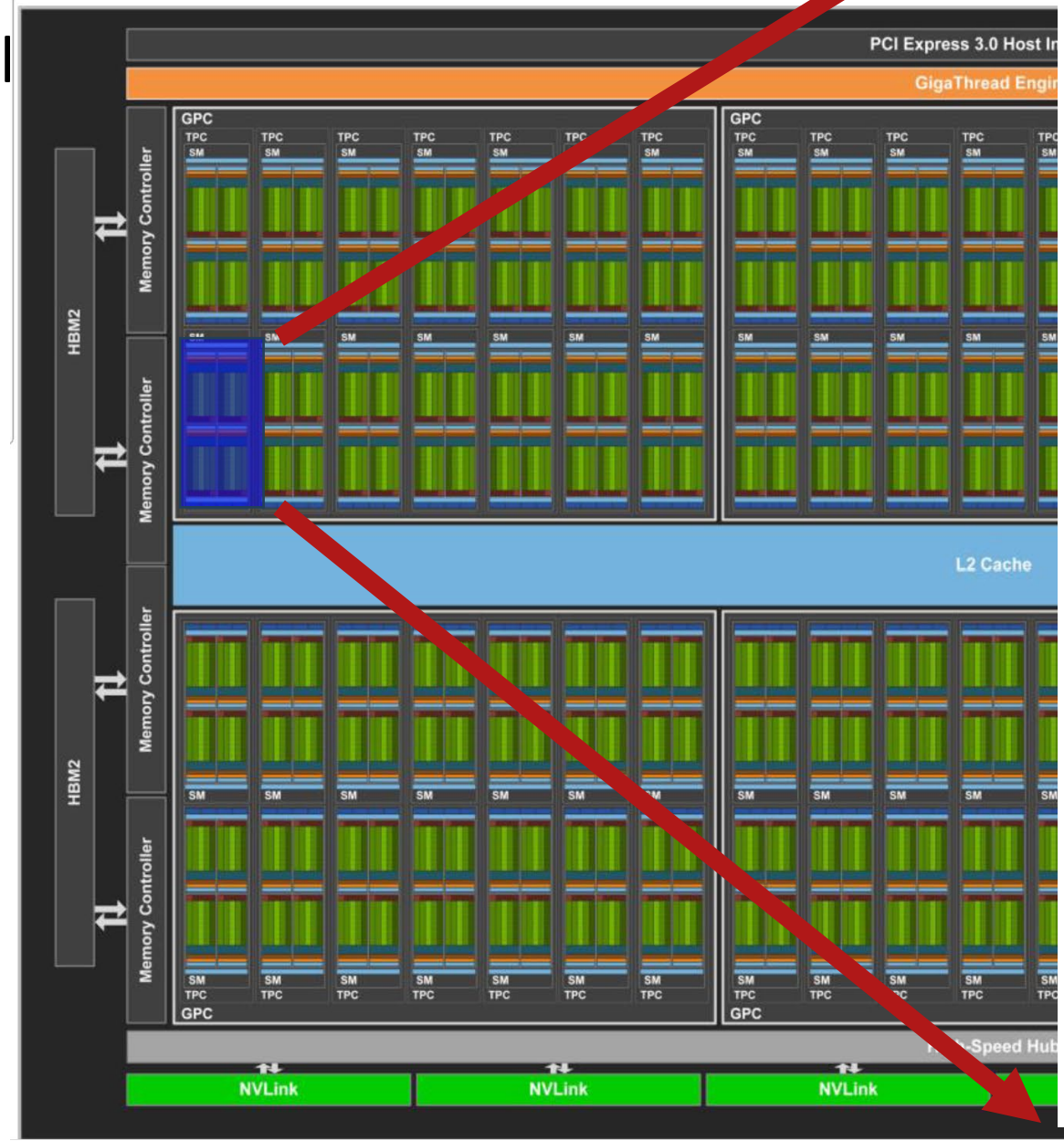
- Th

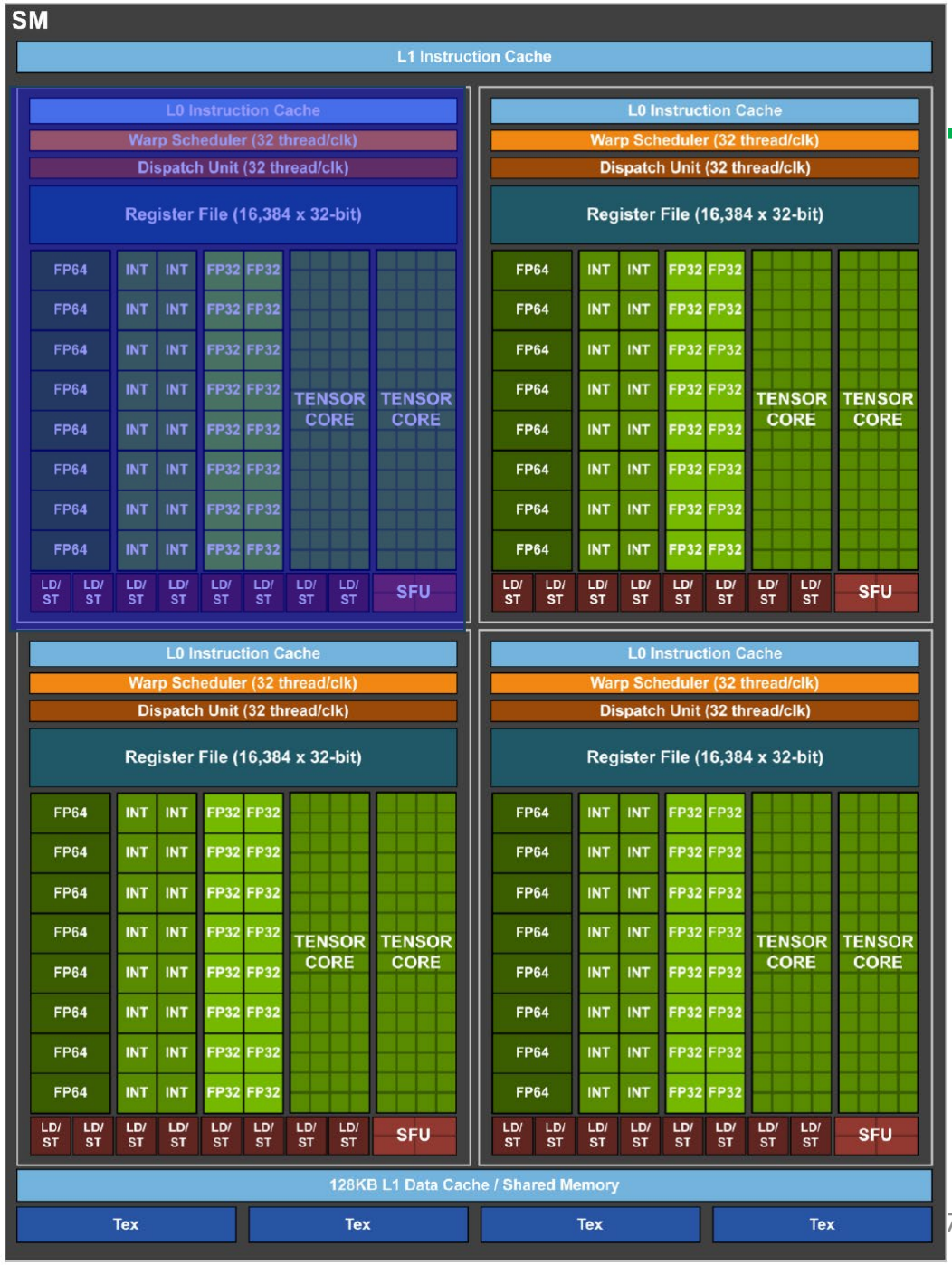




Occupancy

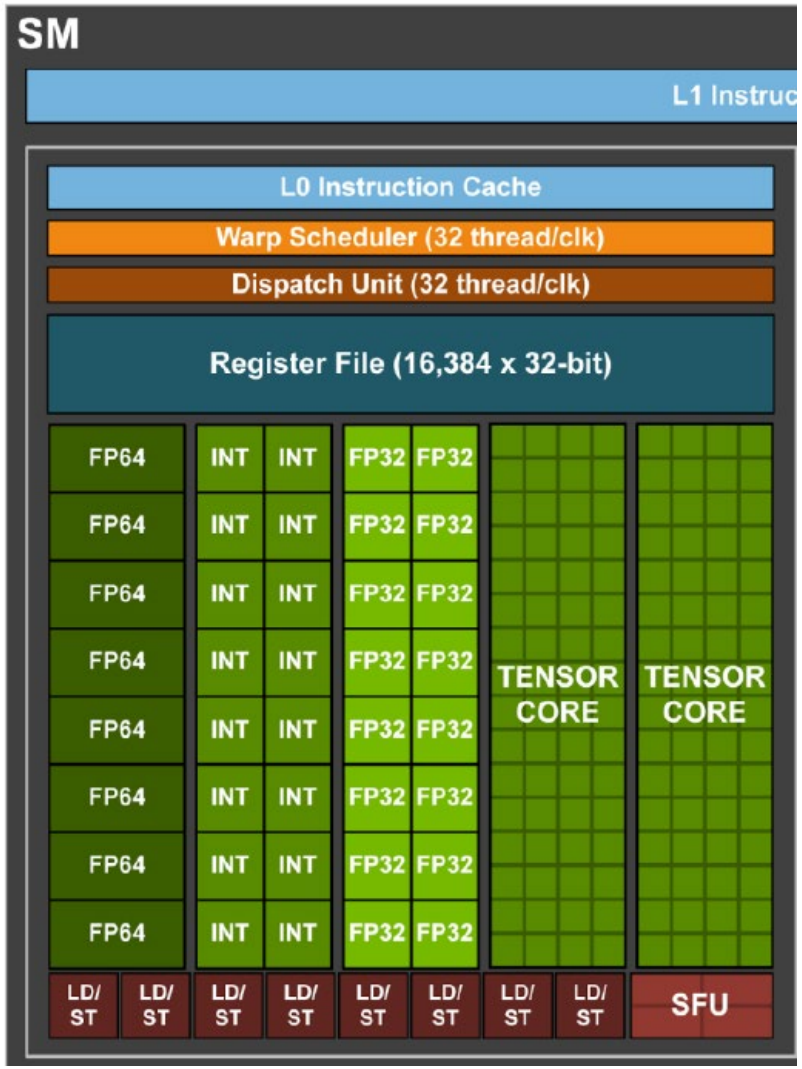
- Th

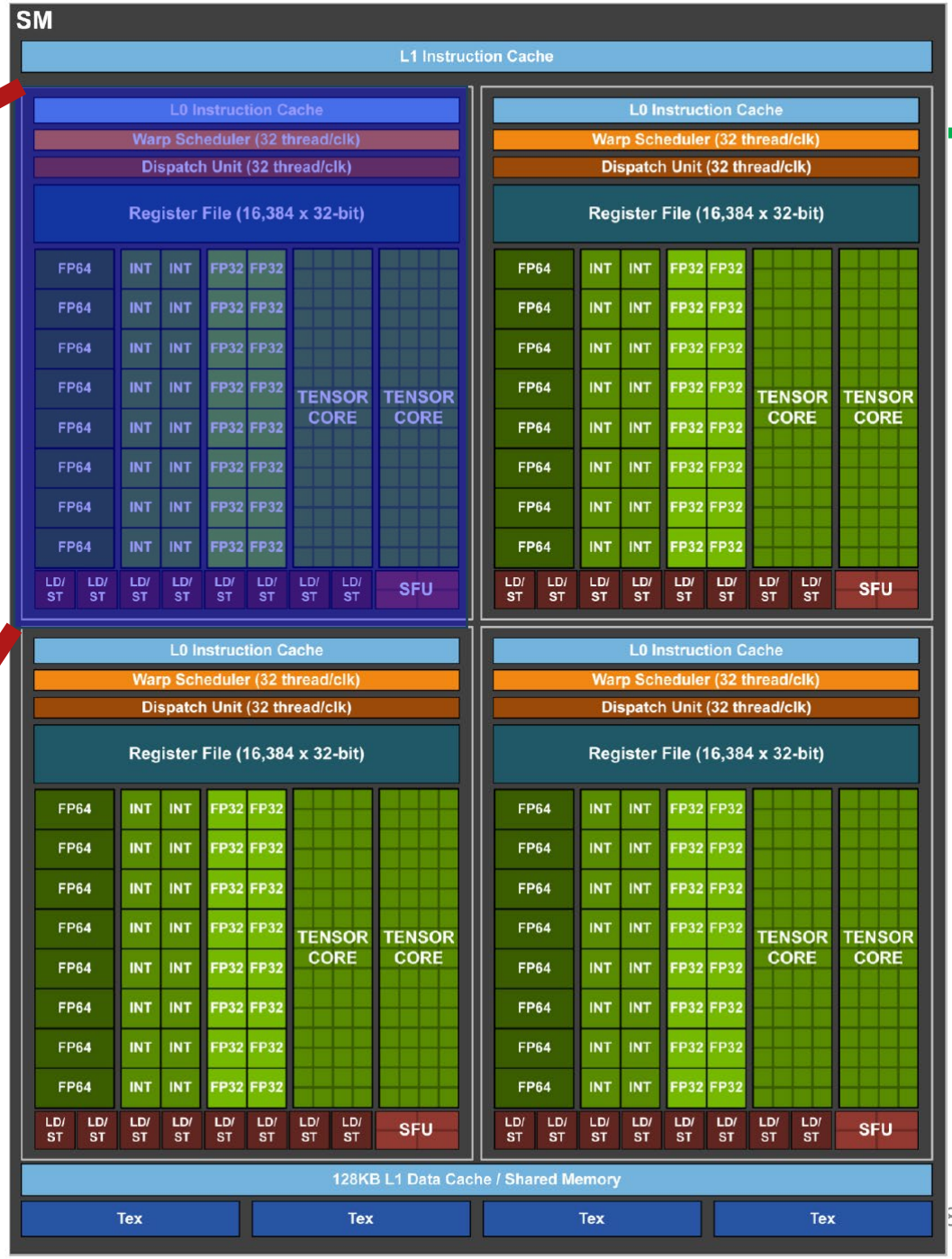




Occupancy

- The achieved occupancy :



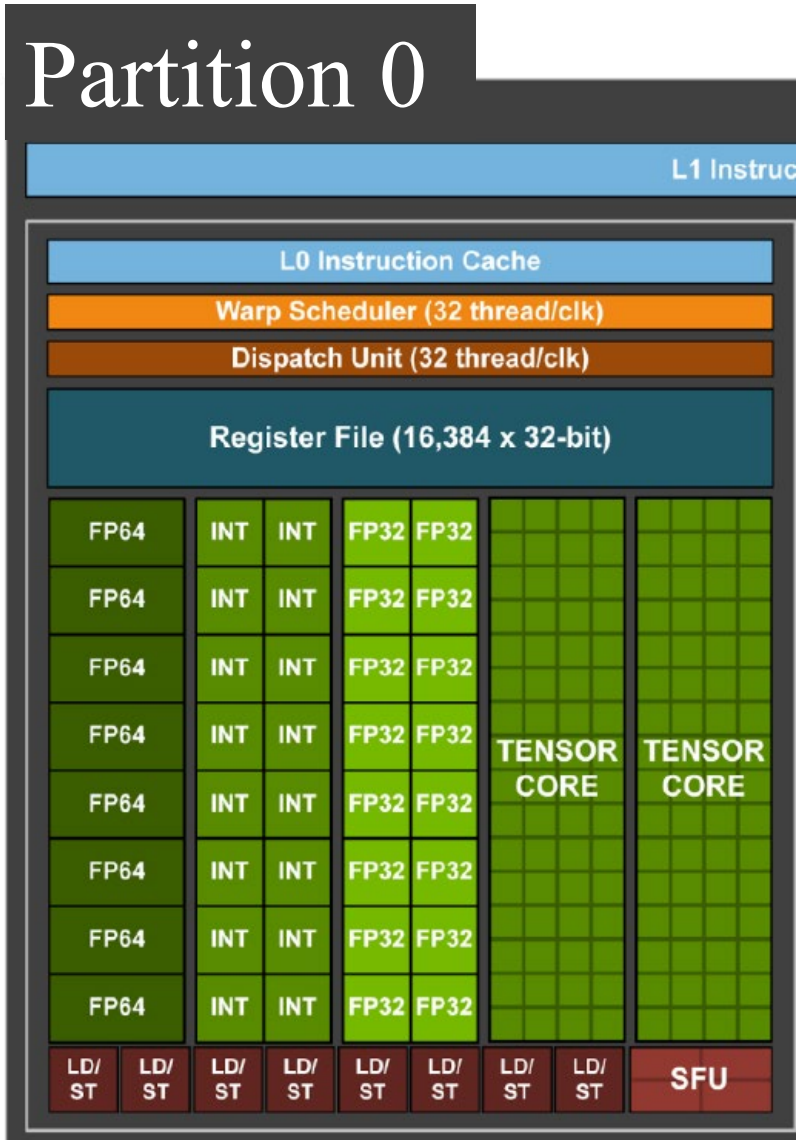


Occupancy

- The achieved occupancy:
  - 1 block/SM. Block size = 512 threads.
  - Total warps/SM = 16 warps.
  - Theoretical occupancy = 100%.
- There are examples when warps are not ready
  - Warp is awaiting a value from memory.
  - During this wait, other warps may be scheduled for execution.
- If all warps encounter significant memory requests
  - Cycles where no warp is ready for execution.
  - **Stall cycles**, which occur due to these **stalled warps**.

The achieved occupancy : scenario 2, memory request

- Assume:
  - 1 block/SM. Block size = 512 threads.
  - Total warps/SM = 16 warps.
  - Theoretical occupancy = 100%.
  - The achieved occupancy is = ?!



Assume regular balance workload for all SMs and partitions.

Occupancy and hiding latency: scenario 1, no memory or dependency

cycle	warp
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

Warp 0

```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  IMAD R14162 R14162 R12370
5  FFMA R33 R31 R32 R136
6  IADD3 R11 P12624 R23 R24
```

Warp 1

```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  IMAD R14162 R14162 R12370
5  FFMA R33 R31 R32 R136
6  IADD3 R11 P12624 R23 R24
```

Warp 2

```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  IMAD R14162 R14162 R12370
5  FFMA R33 R31 R32 R136
6  IADD3 R11 P12624 R23 R24
```

Warp 3

```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  IMAD R14162 R14162 R12370
5  FFMA R33 R31 R32 R136
6  IADD3 R11 P12624 R23 R24
```

Remember: warp=32 threads execute same instruction on different data.

Occupancy and hiding latency : scenario 2, memory request

cycle	warp
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

Warp 0

```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  LDG.E.SYS R3289426 R0
5  IMAD R14162 R14162 R12370
6  FFMA R33 R31 R32 R136
7  IADD3 R11 P12624 R23 R24
8  STG.E.SYS R0 R3354962
```

Warp 1

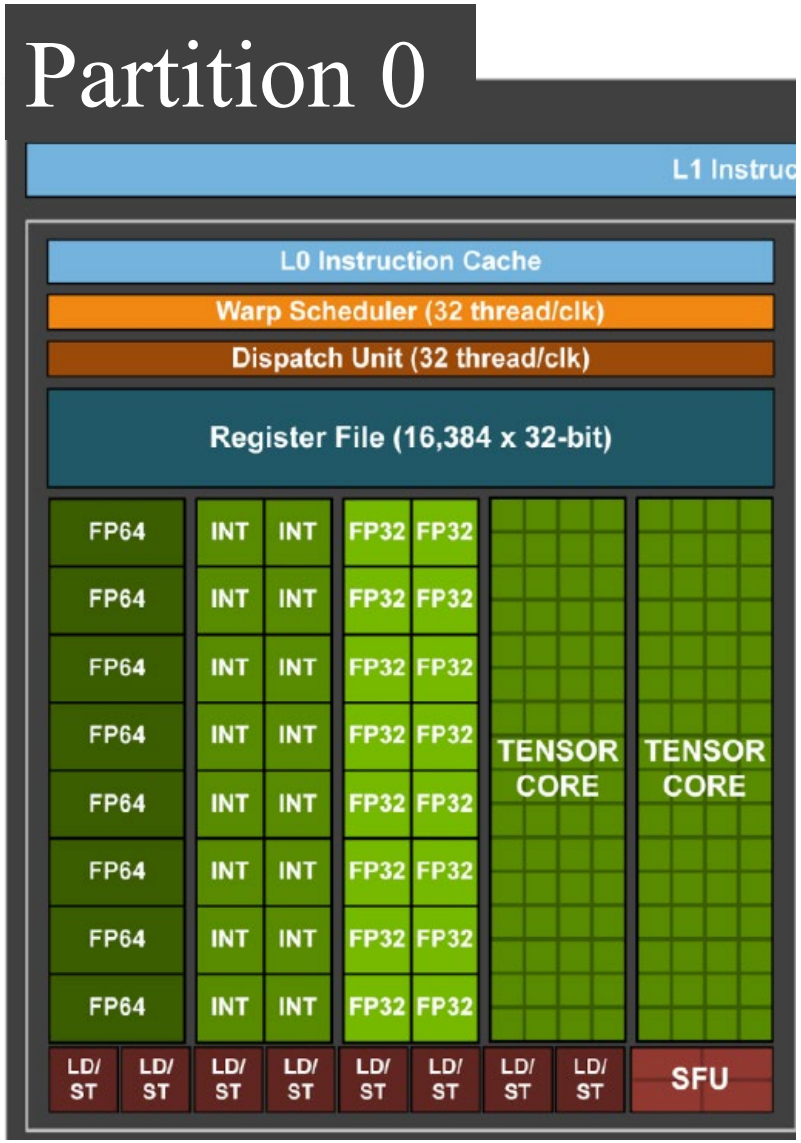
```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  LDG.E.SYS R3289426 R0
5  IMAD R14162 R14162 R12370
6  FFMA R33 R31 R32 R136
7  IADD3 R11 P12624 R23 R24
8  STG.E.SYS R0 R3354962
```

Warp 2

```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  LDG.E.SYS R3289426 R0
5  IMAD R14162 R14162 R12370
6  FFMA R33 R31 R32 R136
7  IADD3 R11 P12624 R23 R24
8  STG.E.SYS R0 R3354962
```

Warp 3

```
1  FMUL R3 R4 R5
2  FMUL R6 R7 R8
3  ISETP.GE.AND P12 P21 R13 P215
4  LDG.E.SYS R3289426 R0
5  IMAD R14162 R14162 R12370
6  FFMA R33 R31 R32 R136
7  IADD3 R11 P12624 R23 R24
8  STG.E.SYS R0 R3354962
```



Remember: warp=32 threads execute same instruction on different data.

Occupancy

- This computation is replicated for each of the 4 partitions within an SM.
- The average of all petitions is calculated. (occupancy/SM).
- This methodology is applied across all SMs.
- Overall achieved occupancy value is determined by averaging the values across all SMs.

The summary:

- Surprisingly, high occupancy doesn't always equate to high performance.
  - Indicate that a significant portion of the GPU's resources are being utilized.
- Identifying and understanding occupancy can help us pinpoint performance issues.
- Low occupancy, on the other hand, suggests that there's a bottleneck preventing the GPU from being fully utilized.
- I want to say” For further information, I recommend visiting this webpage.

https://docs.nvidia.com/gameworks/content/developertools/desktop/analysis/report/cudaexperiments/kernellevel/achievedoccupancy.htm

The summary:

- The theoretical depends on the number of warps and the maximum warps.
- The achieved occupancy depends on other factors
  - Memory requests, instructions dependencies
- Each scheduler in each partition attempts to issue instructions from a warp on each clock cycle.
- To hide latencies: each scheduler must have at least one warp eligible to issue an instruction every clock cycle.

Occupancy

- Occupancy calculator (excel file)