Resource partitioning and latency hiding

The local execution context of a warp mainly consists of the following resources:

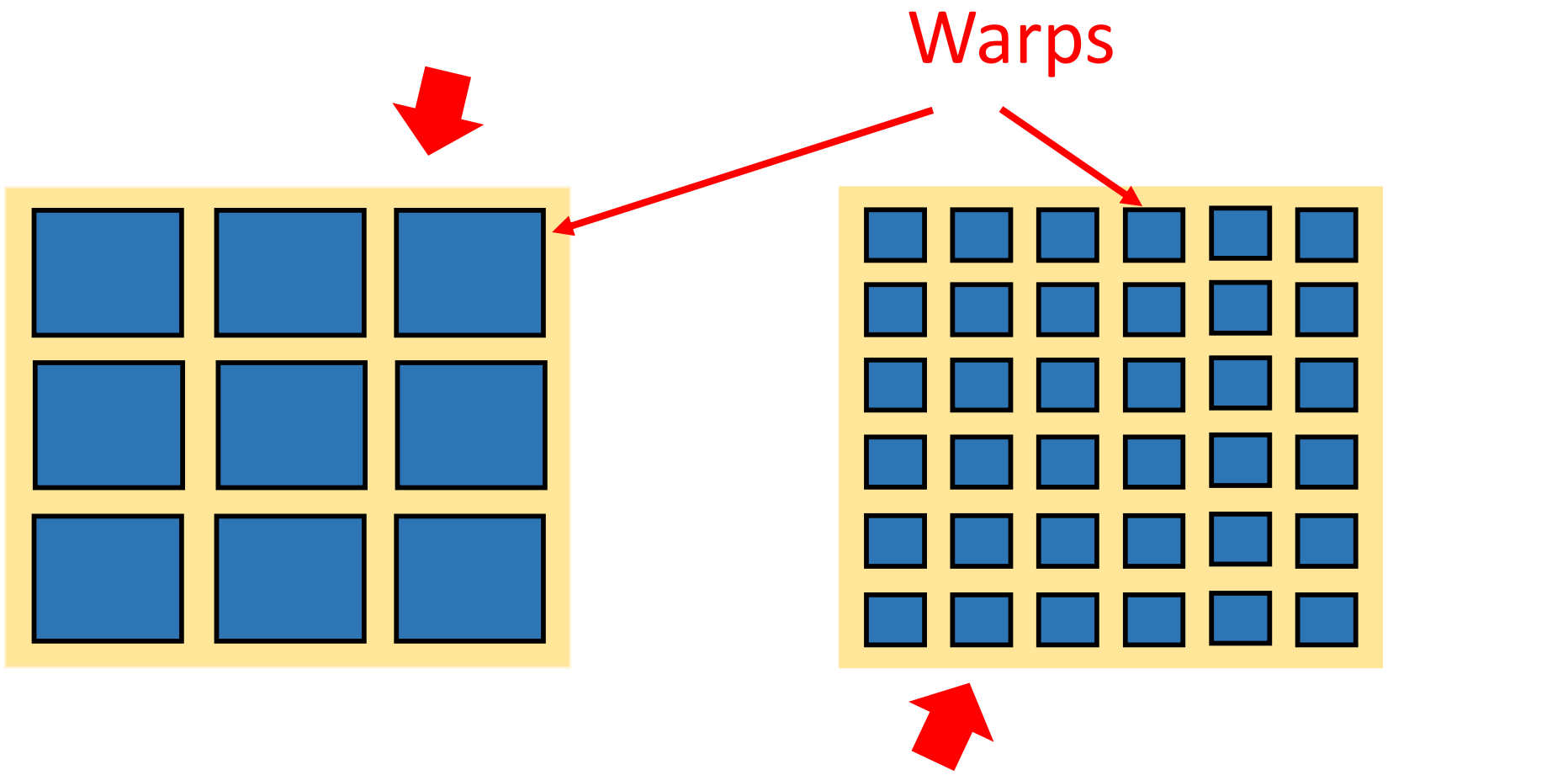
- *Program counters*
- *Registers*
- *Shared memory*

The execution context of each warp processed by a SM is maintained on-chip during the entire lifetime of the warp. Therefore, switching from one execution context to another has no cost

- Registers and shared memory can be directly controlled by the programmer.

- set of 32-bit registers stored in a register file that are partitioned among threads, and a fixed amount of shared memory that is partitioned among thread blocks.

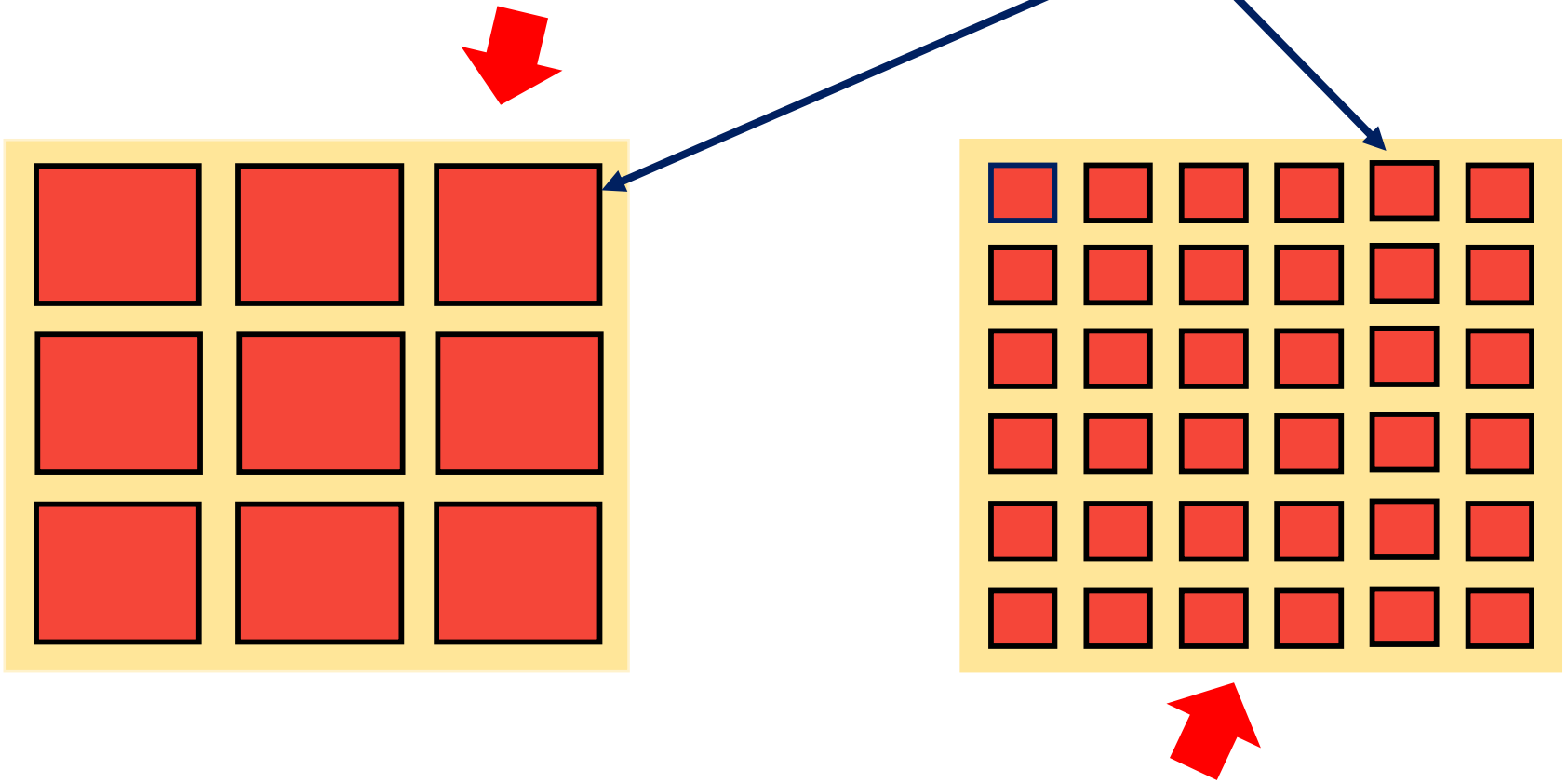
Fewer warps with more register per thread



More warps with fewer register per thread

fewer blocks with more shared memory per block

Thread blocks



More blocks with less shared memory per block

Technical spec	3	3.5	5	6	7
Max concurrent block per SM	16	16	32	32	32
Max concurrent warps per SM	64	64	64	64	64
No register per SM	64K	64K	64K	64K	64K
Max Register per thread	63	255	255	255	255
Shared memory per SM	48K	48K	64K	64K	96K

Warp categories in SM

Active blocks/warps

Resources have been allocated

- Selected warp — *actively executing*
- Stalled warp — *not ready for execution*
- Eligible warp — *ready for execution but not currently executing*

Conditions to be a eligible warps

- 32 CUDA cores should free for execution
- All arguments to the current instruction for that warp should ready