Diagonal Transpose

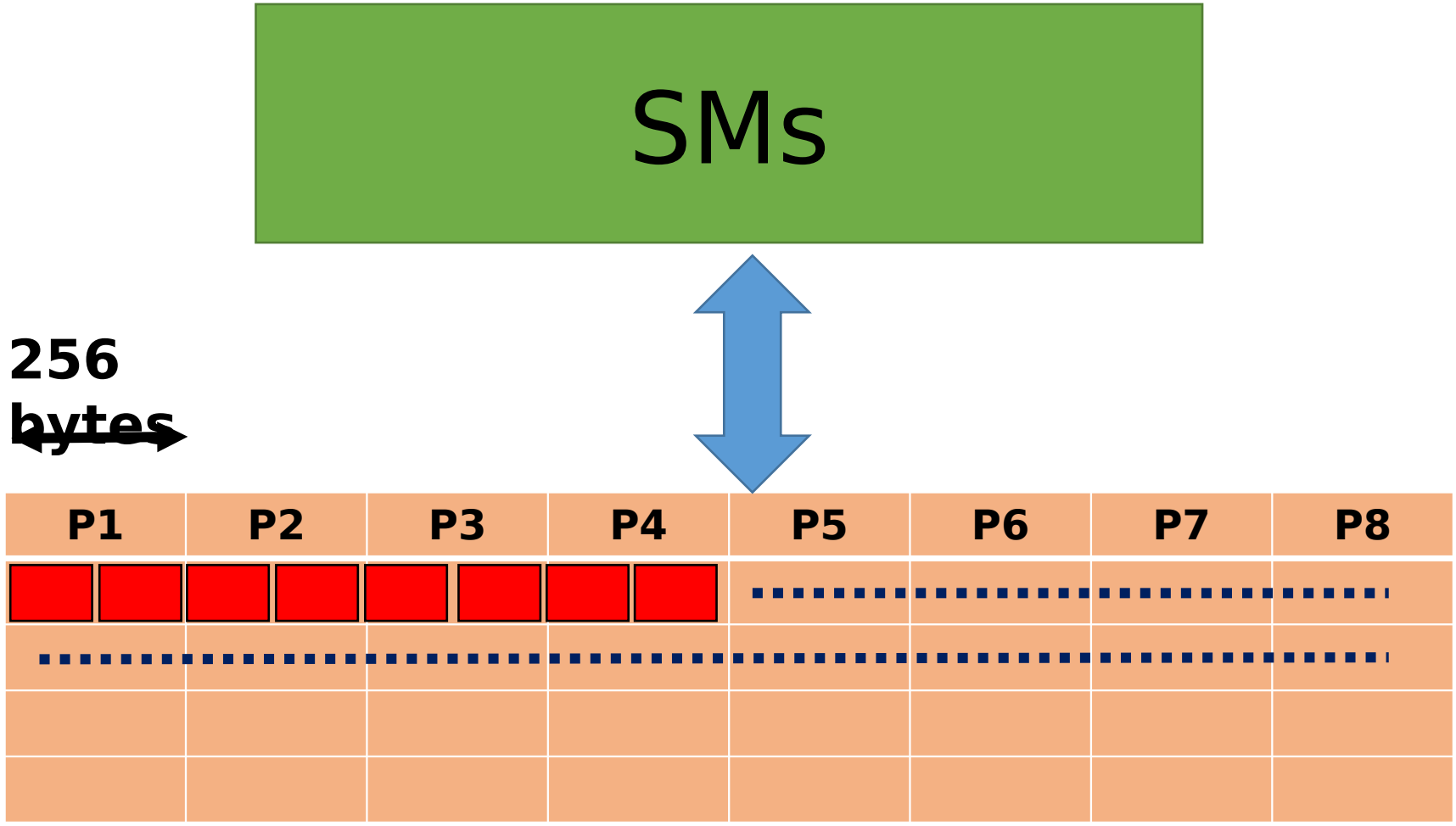
blockIdx.X

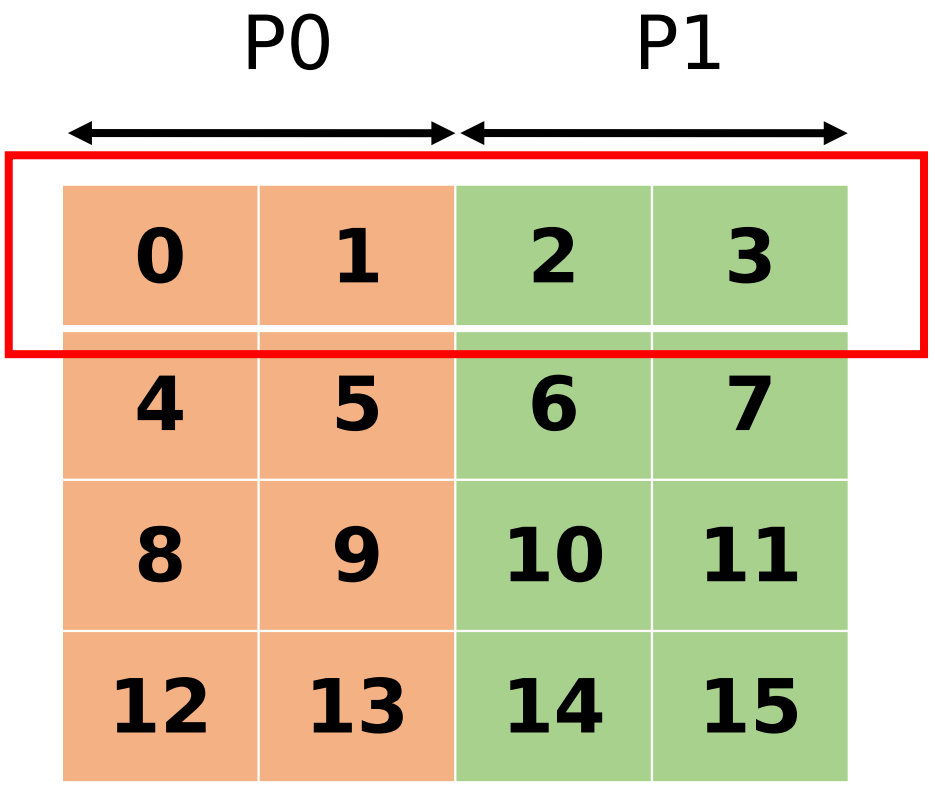
blockIdx.Y

(0,0)	(1,0)	(2,0)	(3,0)
(0,1)	(1,1)	(2,1)	(3,1)
(0,2)	(1,2)	(2,2)	(3,2)
(0,3)	(1,3)	(2,3)	(3,3)

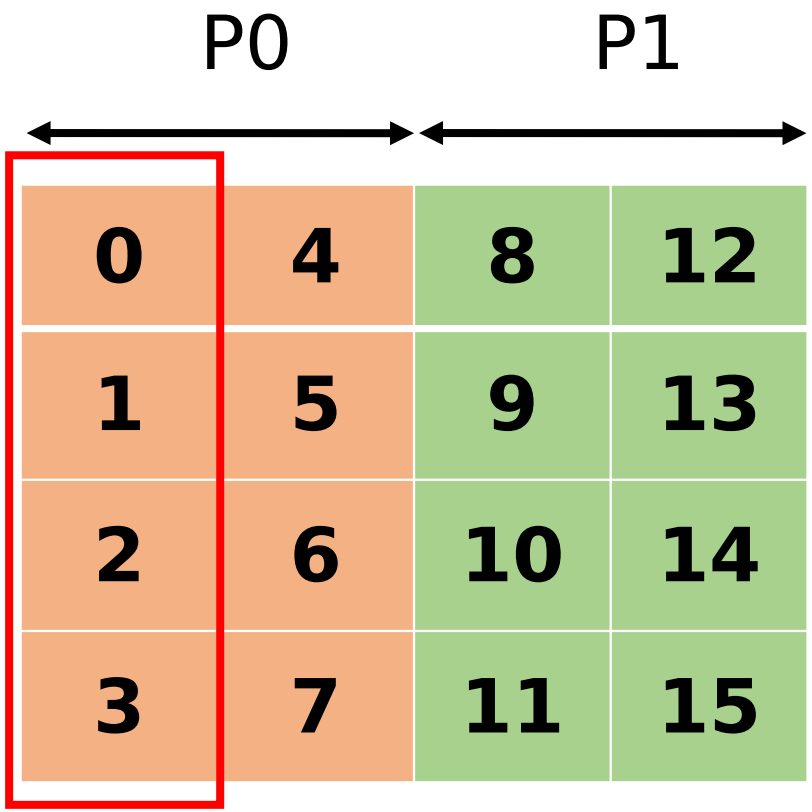
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

bid = blockIdx.Y * gridDim.x + blockIdx.X


SMs
256 bytes
P1
P2
P3
P4
P5
P6
P7
P8



For original matrix



For transpose matrix

Partition camping

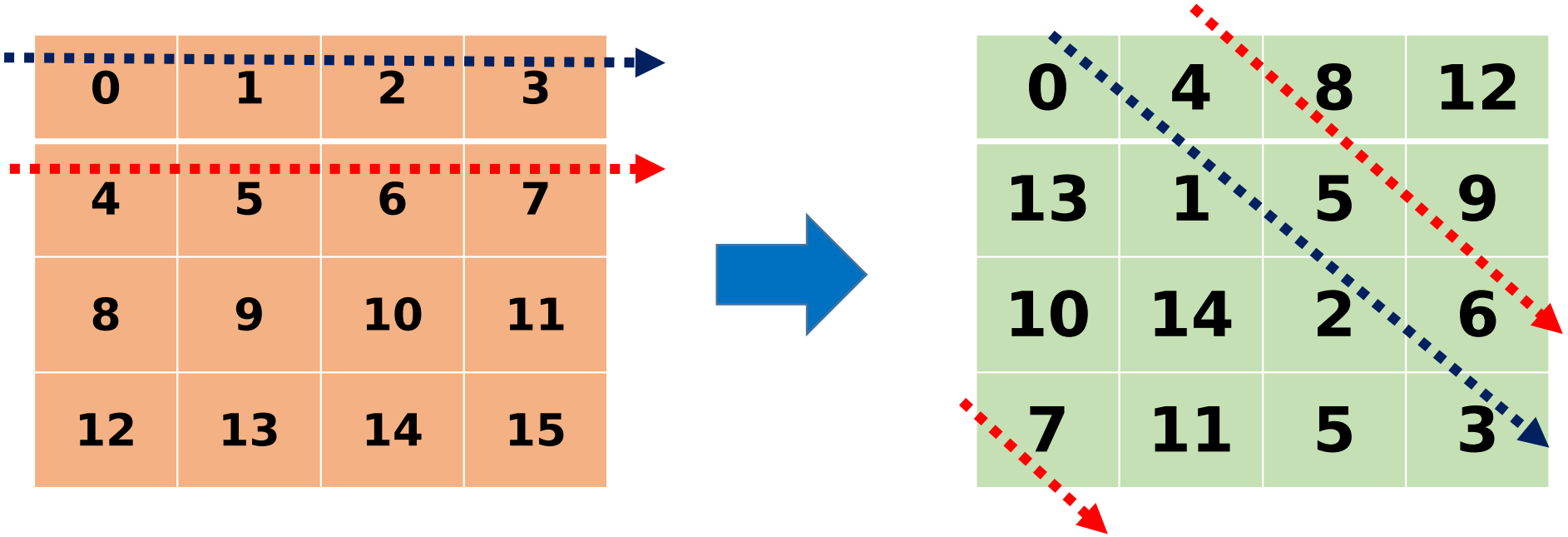
In partition camping, memory requests are queued at some partitions while other partitions remain unused

Solution

We are going to calculate the index values ix and iy, using diagonal coordinate system which will make sure consecutive thread blocks to access nonconsecutive memory blocks.

But with in a single thread block we need threads with that thread block to access consecutive memory address to adhere the coalesced memory access

Thread block and data block mapping using Diagonal coordinate system



```
blk_y = blockIdx.x;

blk_x = ( blockIdx.x + blockIdx.y) %
gridDim.x;


ix = blockDim.x * blockIdx.x+
threadIdx.x;

iy = blockDim.y * blockIdx.y +
threadIdx.y;
```