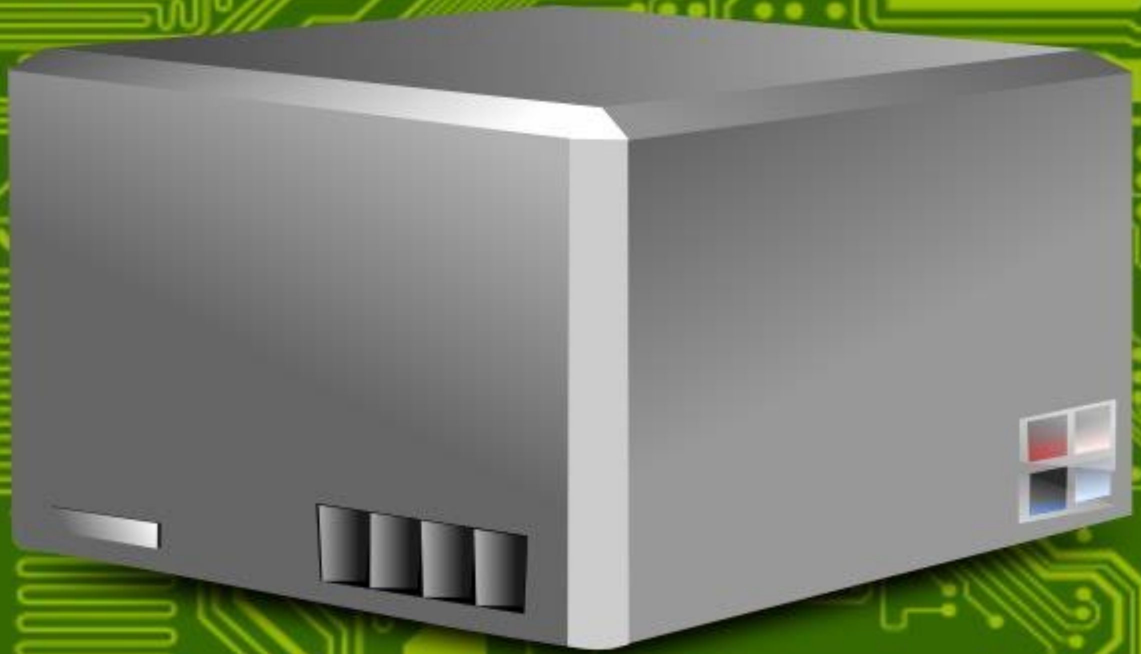
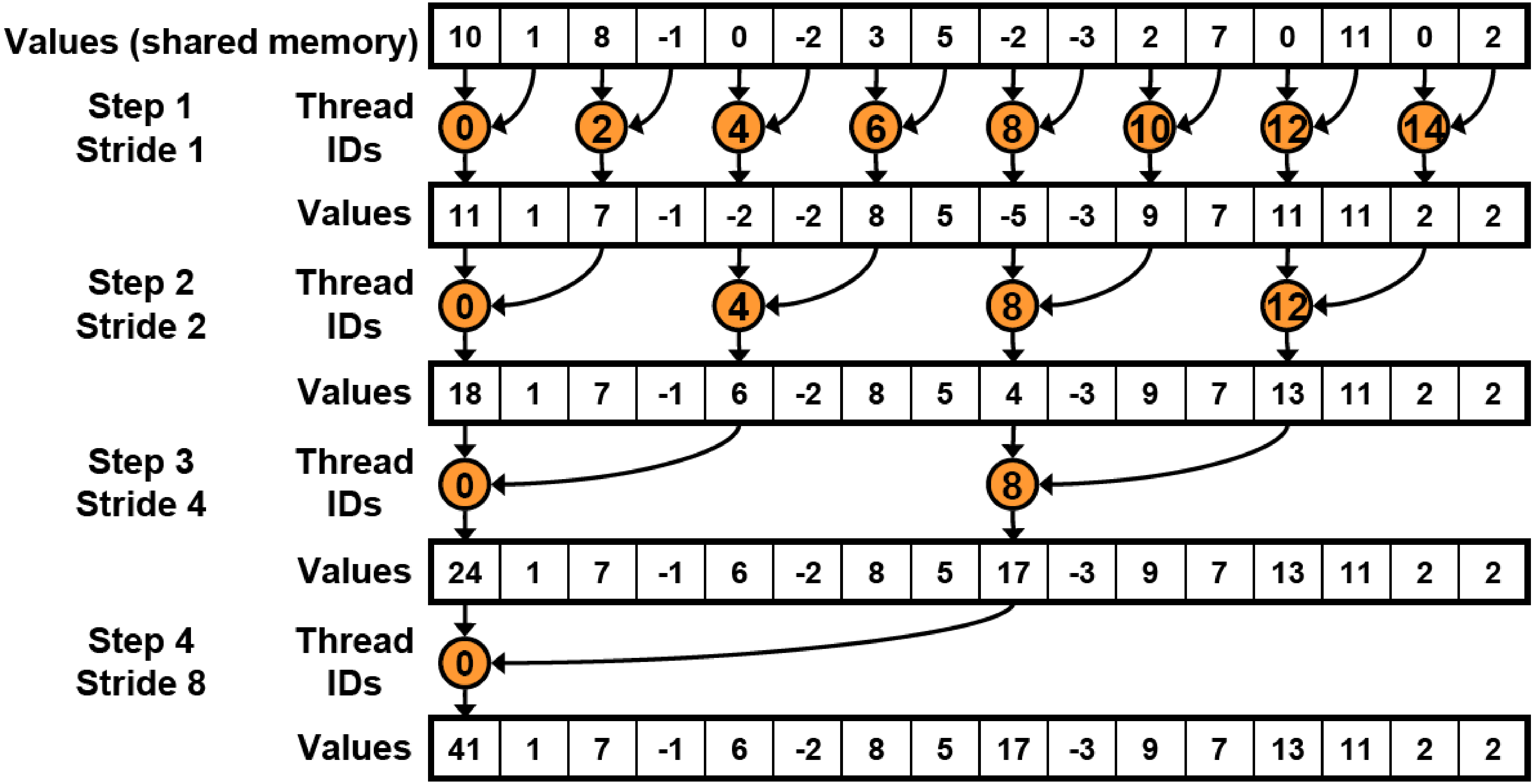
GPU Computing



规约算法

Hui Liu
Email: hui.sc.liu@gmail.com

Reduction Steps



Values (shared memory)		10	1	8	-1	0	-2	3	5	-2	-3	2	7	0	11	0	2
Step 1 Stride 1	Thread IDs	0		2		4		6		8		10		12		14	
	Values	11	1	7	-1	-2	-2	8	5	-5	-3	9	7	11	11	2	2
Step 2 Stride 2	Thread IDs	0				4				8				12			
	Values	18	1	7	-1	6	-2	8	5	4	-3	9	7	13	11	2	2
Step 3 Stride 4	Thread IDs	0								8							
	Values	24	1	7	-1	6	-2	8	5	17	-3	9	7	13	11	2	2
Step 4 Stride 8	Thread IDs	0															
	Values	41	1	7	-1	6	-2	8	5	17	-3	9	7	13	11	2	2

Reduction #2: Interleaved Addr./non-divergent branching

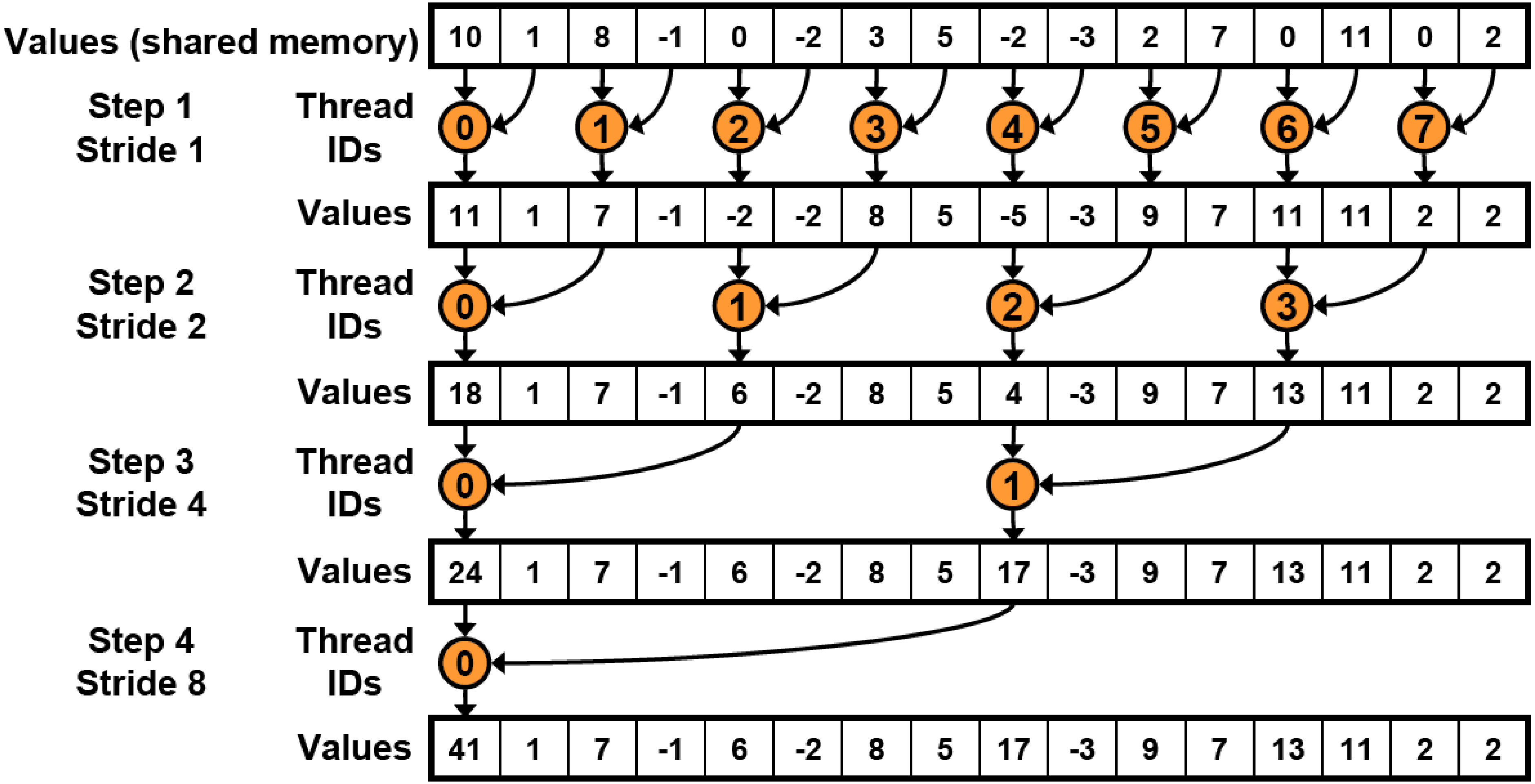
- Replace the divergent branching code:

```
// do reduction in shared mem
for (unsigned int s=1; s < blockDim.x; s *= 2) {
        if (tid % (2*s) == 0) {
                sdata[tid] += sdata[tid + s];
        }
        __syncthreads();
}
```

- With strided index and non-divergent branch

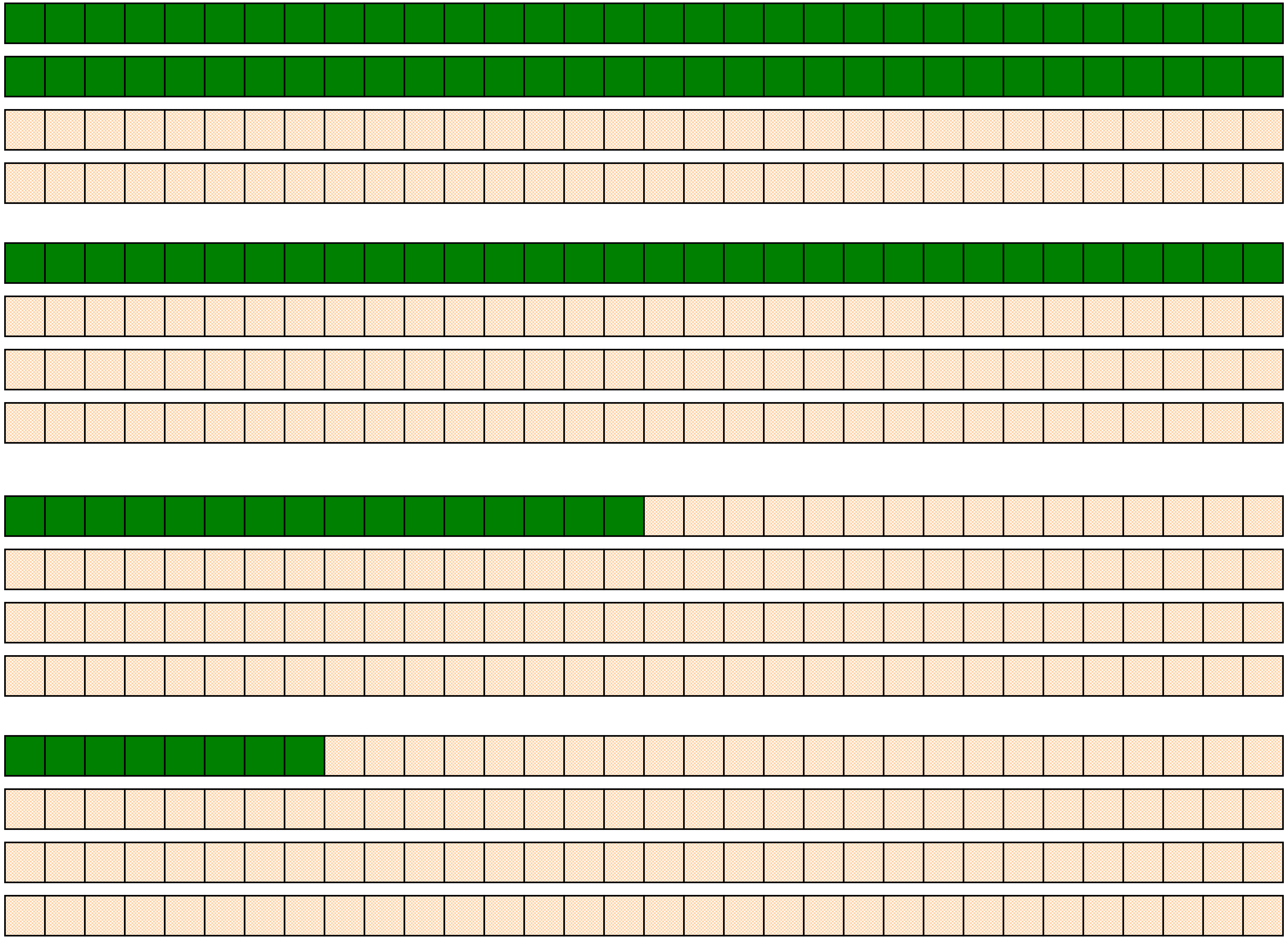
```
// do reduction in shared mem
for (unsigned int s=1; s < blockDim.x; s *= 2) {
        int index  = 2 * s * tid;

        if (index < blockDim.x / s) {
                sdata[index] += sdata[index + s];
        }
        __syncthreads();
}
```

Reduction #2: Access Pattern



Values (shared memory)	10	1	8	-1	0	-2	3	5	-2	-3	2	7	0	11	0	2
Step 1 Stride 1 — Thread IDs	0		1		2		3		4		5		6		7	
Values	11	1	7	-1	-2	-2	8	5	-5	-3	9	7	11	11	2	2
Step 2 Stride 2 — Thread IDs	0				1				2				3			
Values	18	1	7	-1	6	-2	8	5	4	-3	9	7	13	11	2	2
Step 3 Stride 4 — Thread IDs	0								1							
Values	24	1	7	-1	6	-2	8	5	17	-3	9	7	13	11	2	2
Step 4 Stride 8 — Thread IDs	0															
Values	41	1	7	-1	6	-2	8	5	17	-3	9	7	13	11	2	2

Reduction #2: Warp control flow

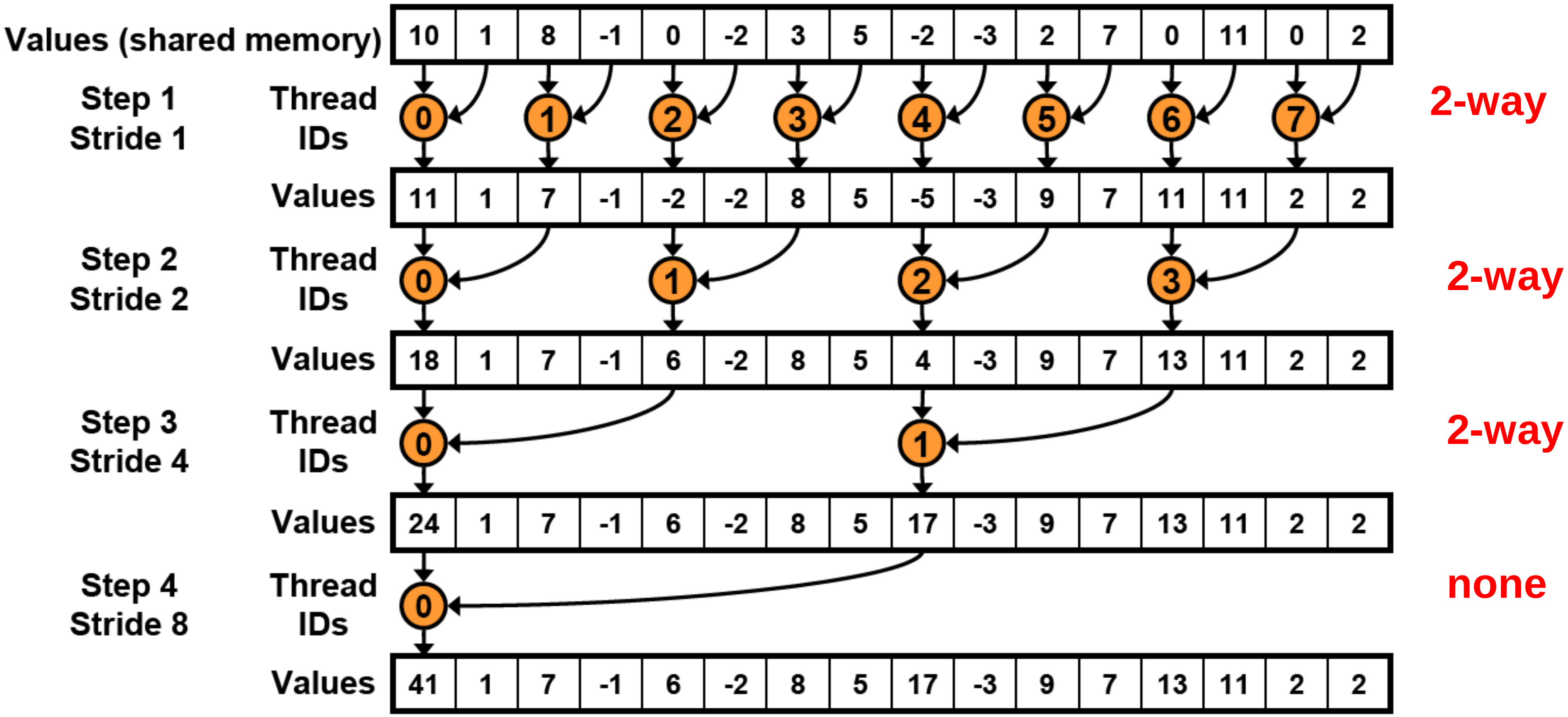


4 warps/step

Performance for 4M element reduction

	Time (2^{22} ints)	Bandwidth	Step Speedup	Cumulative Speedup
Kernel 1: interleaved addressing with divergent branching	**8.054 ms**	**2.083 GB/s**		
Kernel 2: interleaved addressing non-divergent branching	**3.456 ms**	**4.854 GB/s**	**2.33x**	**2.33x**

Reduction #2: Problem



Values (shared memory)		10	1	8	-1	0	-2	3	5	-2	-3	2	7	0	11	0	2	
Step 1 Stride 1	Thread IDs	0		1		2		3		4		5		6		7		2-way
	Values	11	1	7	-1	-2	-2	8	5	-5	-3	9	7	11	11	2	2	
Step 2 Stride 2	Thread IDs	0				1				2				3				2-way
	Values	18	1	7	-1	6	-2	8	5	4	-3	9	7	13	11	2	2	
Step 3 Stride 4	Thread IDs	0								1								2-way
	Values	24	1	7	-1	6	-2	8	5	17	-3	9	7	13	11	2	2	
Step 4 Stride 8	Thread IDs	0																none
	Values	41	1	7	-1	6	-2	8	5	17	-3	9	7	13	11	2	2	

2-way bank conflicts at every step

Recall there are more than 16 threads

To see the conflicts see what happens with 128 threads

THANK YOU