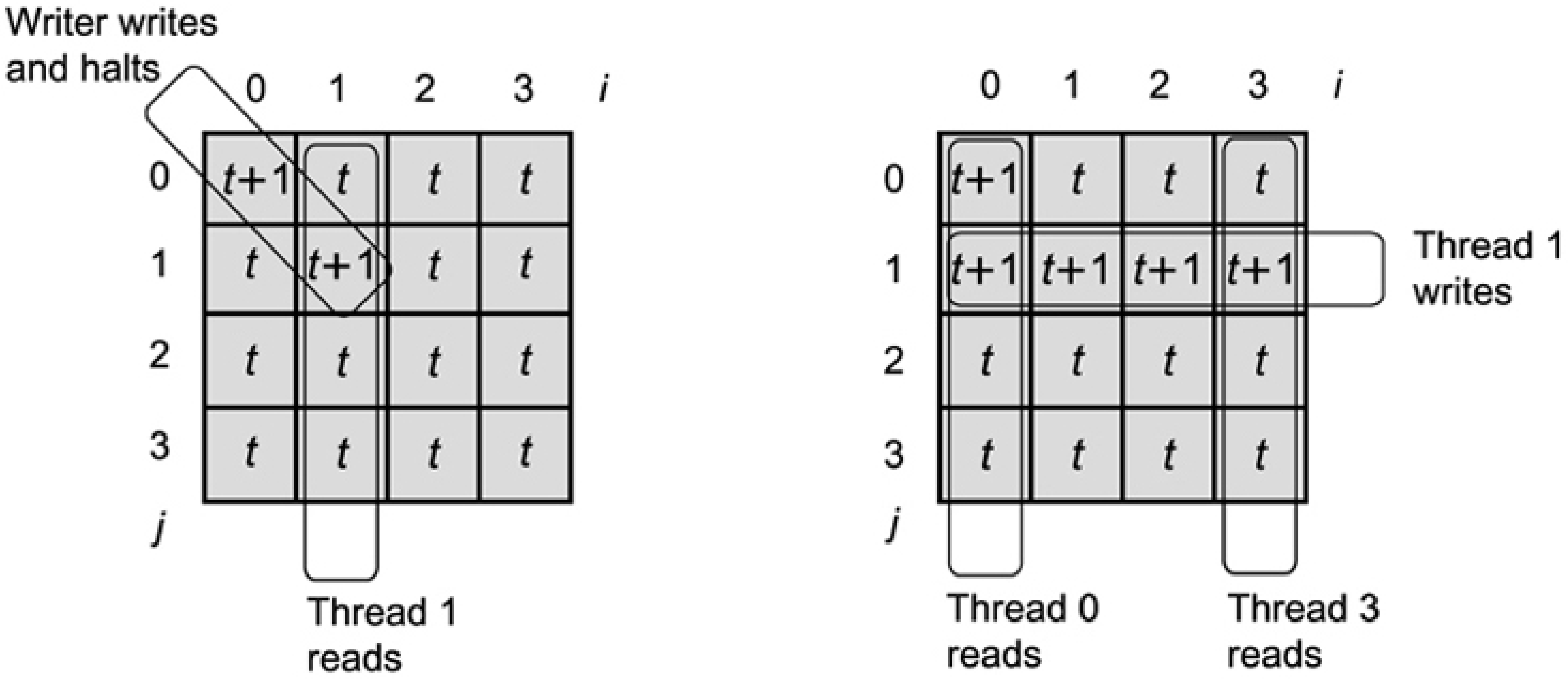

j \ i	0	1	2	3
0	*t*+1	*t*	*t*	*t*
1	*t*	*t*+1	*t*	*t*
2	*t*	*t*	*t*	*t*
3	*t*	*t*	*t*	*t*

j \ i	0	1	2	3
0	*t*+1	*t*	*t*	*t*
1	*t*+1	*t*+1	*t*+1	*t*+1
2	*t*	*t*	*t*	*t*
3	*t*	*t*	*t*	*t*

FIGURE 4.13 An execution of the atomic MRSW register. Each reader thread has an index between 0 and 3, and we refer to each thread by its index. Here, the writer writes a new value with timestamp *t* + 1 to locations a_table[0][0] and a_table[1][1] and then halts. Then, thread 1 reads its corresponding column a_table[*i*][1] for all *i*, and writes its corresponding row a_table[1][*i*] for all *i*, returning the new value with timestamp *t* + 1. Threads 0 and 3 both read completely after thread 1's read. Thread 0 reads a_table[0][0] with value *t* + 1. Thread 3 cannot read the new value with timestamp *t* + 1 because the writer has yet to write a_table[3][3]. Nevertheless, it reads a_table[1][3] and returns the correct value with timestamp *t* + 1 that was read by the earlier thread 1.