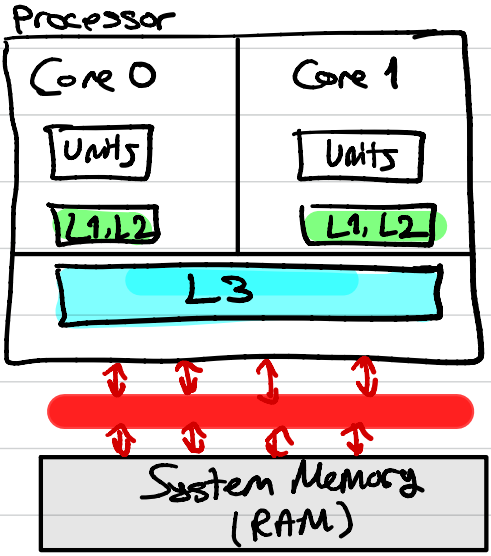
Synchronization

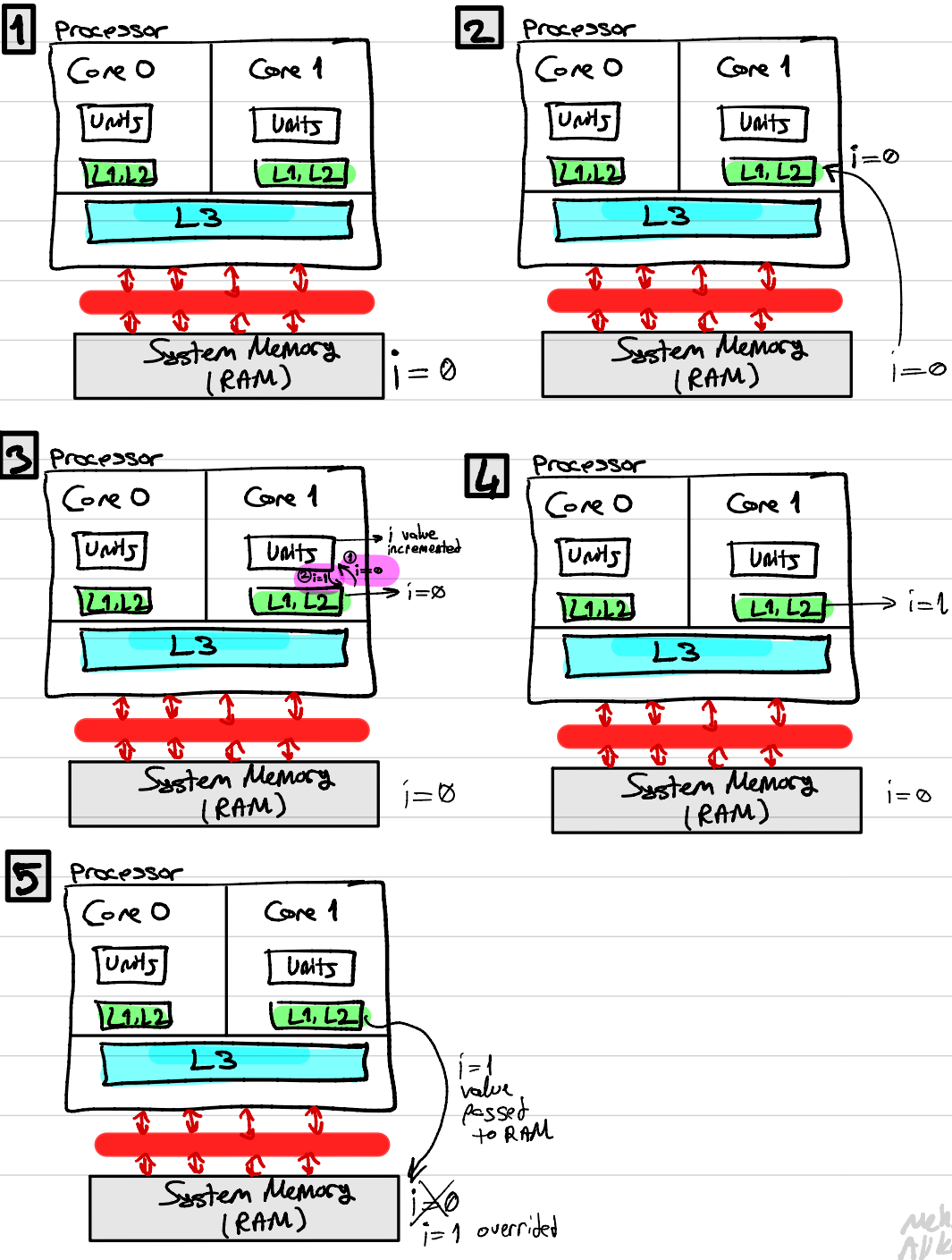
Processor Cores with Cache



L1, L2, L3 are Core Caches. Holds Data.
L3 is the shared cache.
Core 0 L1 and L2 will only hold data related to Core 0.

- int i = 0; → if i value is only used by Core 0, then this value will be copied to Core 0 L1 or L2. If i was used by Both Core 0 and Core 1, then it would be copied to L3.

```
int i=0;
i++;
```

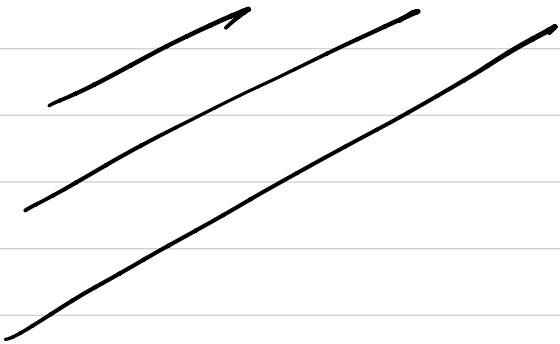


Note: Notice That on 4th step the value of i is different on Core cache (L) and RAM. if i value is being worked on only one thread there is no problem. i value is not shared between threads.

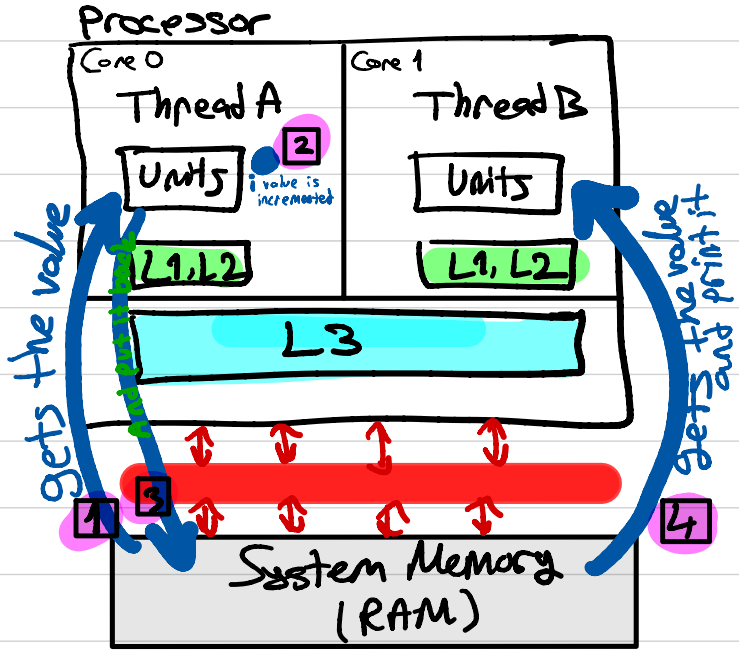
2 problematic Scenerios:

2 threads works on same data.

1. Thread A writes on data and At the same Time Thread B reads Data
2. Both of the threads tries to write Data

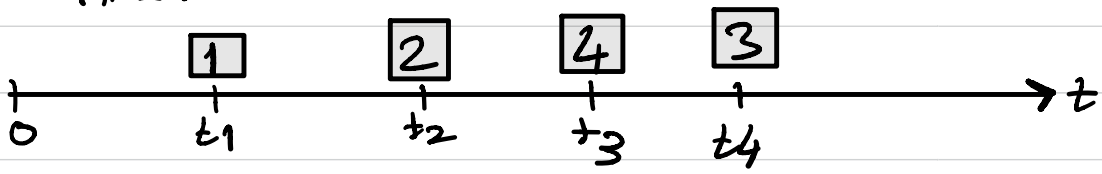


-1 thread is writing, 1 is reading



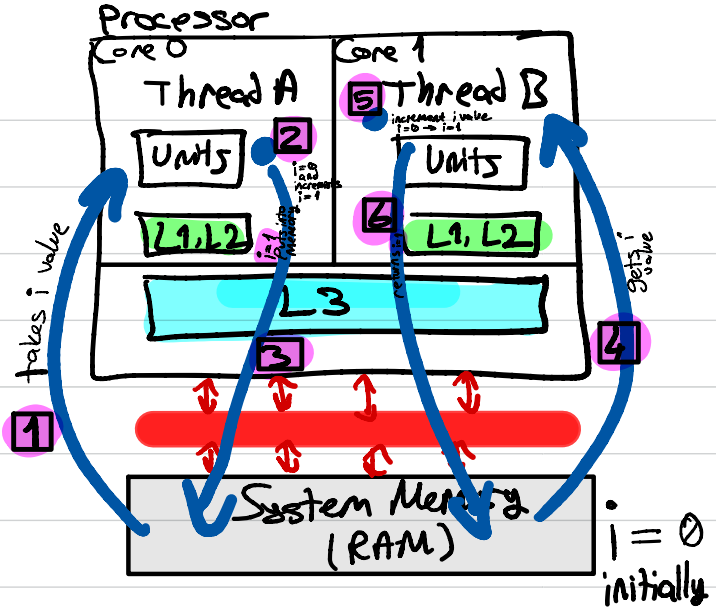
i=0 initially

Time Line

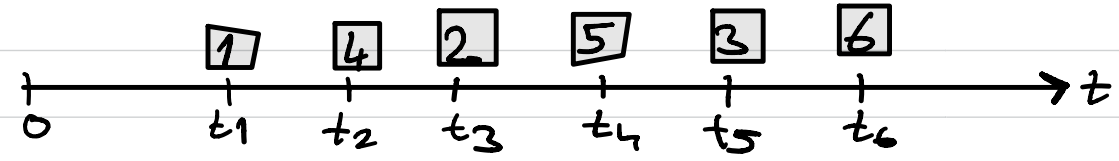


	RAM	Core 0 L1,L2	Core 1 L1,L2
t1 :	0	0	-
t2 :	0	1	-
t3 :	0	1	0
t4 :	1	1	0

2 threads tries to increment same variable and end up with not incrementing it.

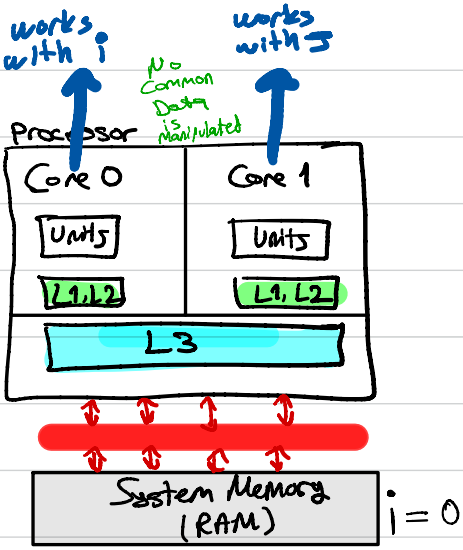


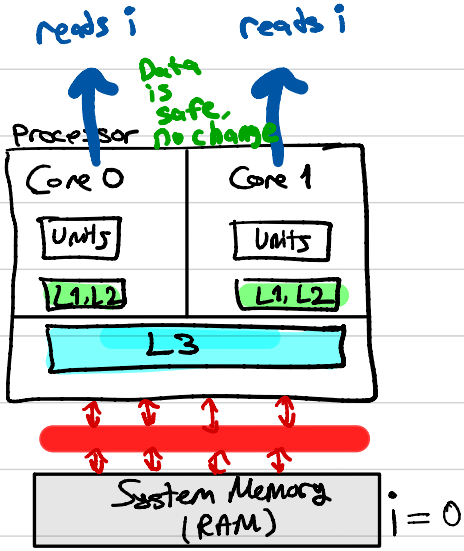
Time Line



	RAM	Core 0 L1,L2	Core 1 L1,L2
t1:	0	0	—
t2:	0	0	0
t3:	0	1	0
t4:	0	1	1
t5:	1	1	1
t6:	1	1	1

- No Problem





- Problem

Both cores reads and writes (details above pages)