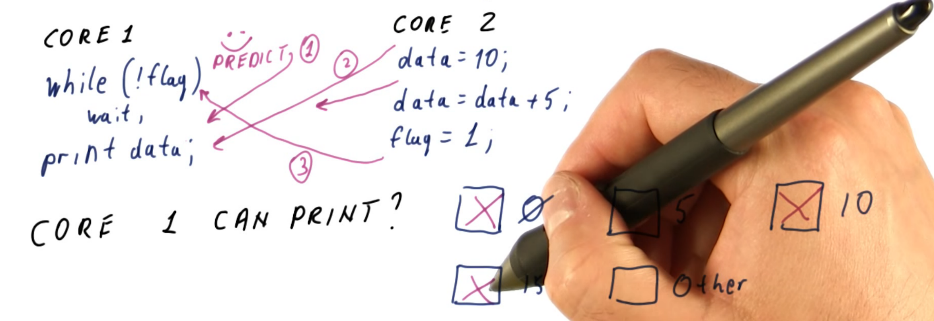
## Memory Consistency

* Coherence: define the order of access for shared data (same address)
* Consistency: define the order of access for different address



Since there is branch prediction in core 1, could pre-fetch data value.

Sequential Consistency:

The access result from each processor should follow program order. Have to delay until all previous access is completed. (Memory Level Parallelism MLP as 1) -> bad for performance

1. Detect for other processors when doing reorder (monitor coherence traffic)
2. MSYNC instruction (need to complete previous order and then to following)

Put MSYNC after Acquire Lock, and Put MSYNC before Release Lock 图片包含 文字

描述已自动生成

Data Race: RD->WR, WR->RD, WR->WR

1. Sequential consistency ensures the program order (no reorder)
2. Weak consistency ensures synchronization accesses never reordered (code before acquire and release can be reordered, but must complete when it comes to acquire or release)
3. Release consistency distinguish between acquire and release (code before release can be reorder, but must complete when it comes to release)

图片包含 屏幕截图

描述已自动生成