Recitation 2: Cache Attacks

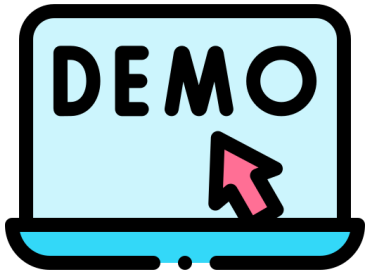
Mengjia Yan

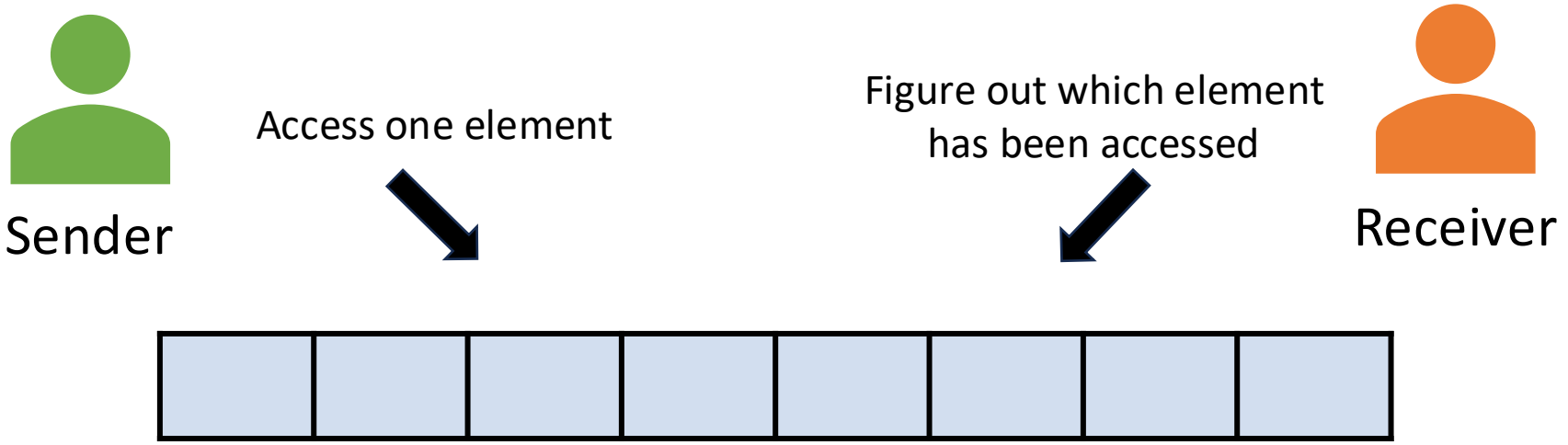
Spring 2026

Agenda

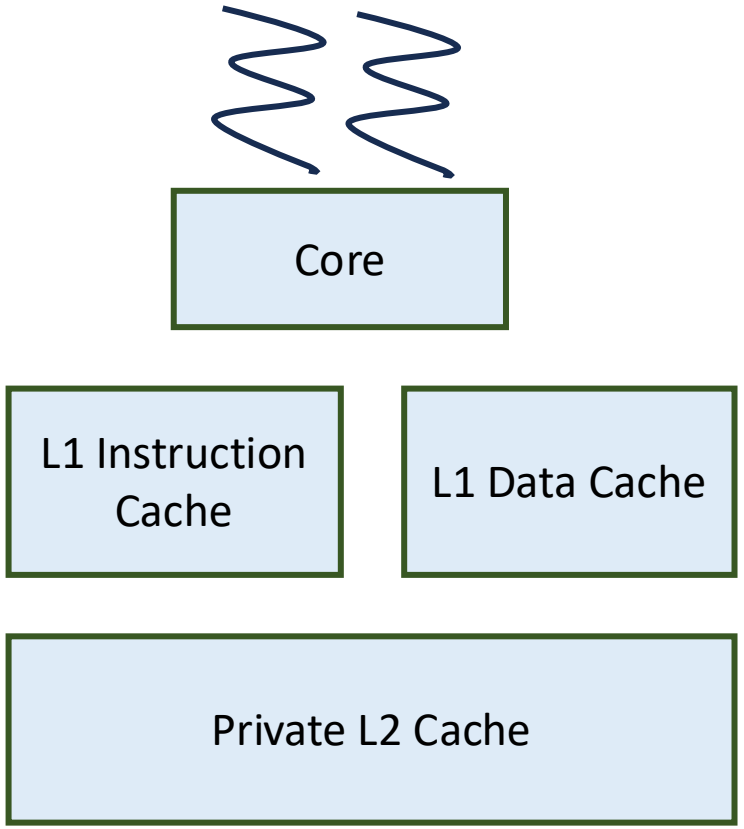
- Tour of the unicorn/dobby machines
- Explanation for the TLB-flush demo

Recall the Flush+Reload Demo





Core and Private Caches



Simultaneous Multithreading (**SMT**) = Hyperthreading

- Running more than 1 threads in one physical core

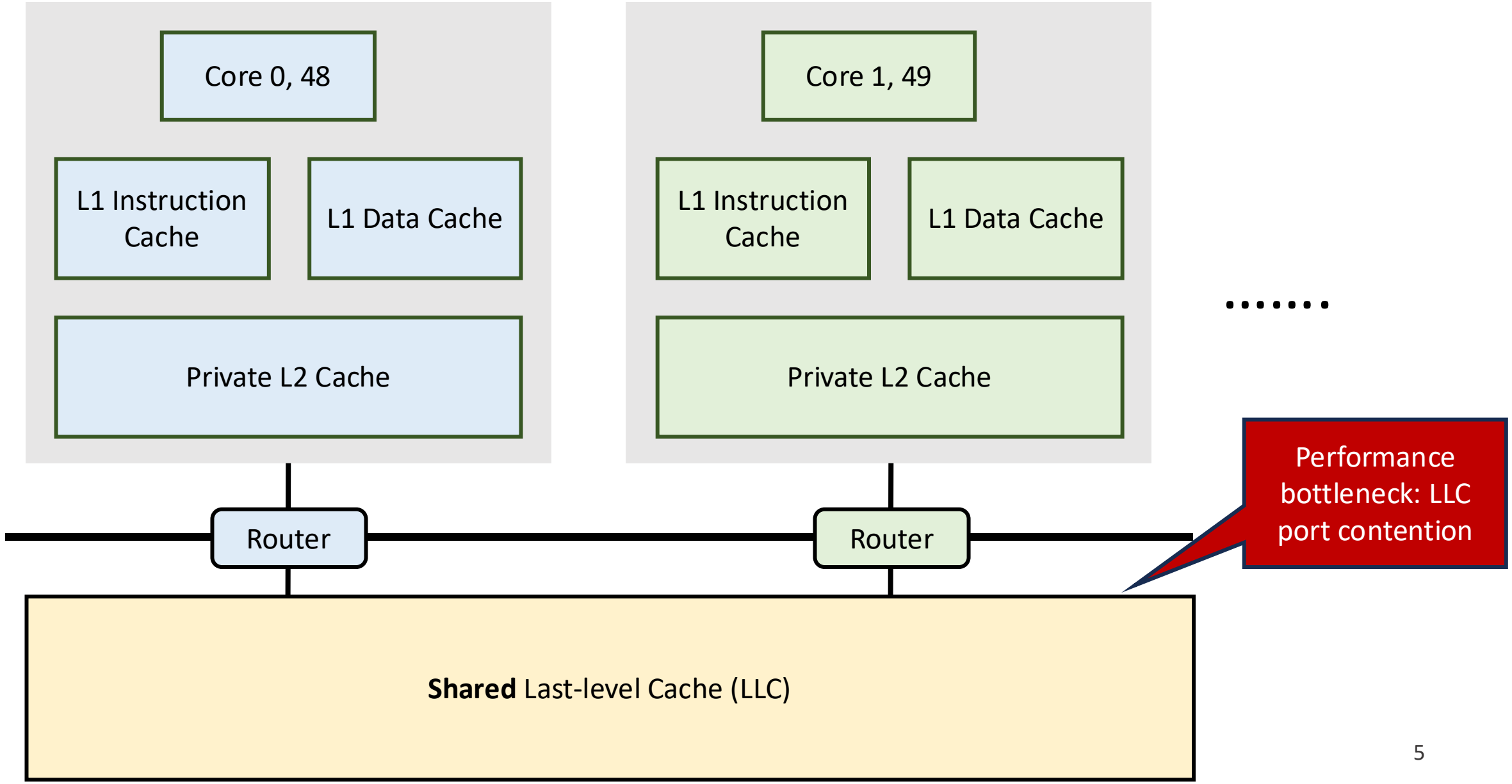
Inclusive cache:

- Any L1 cache block has a duplicated copy in the L2

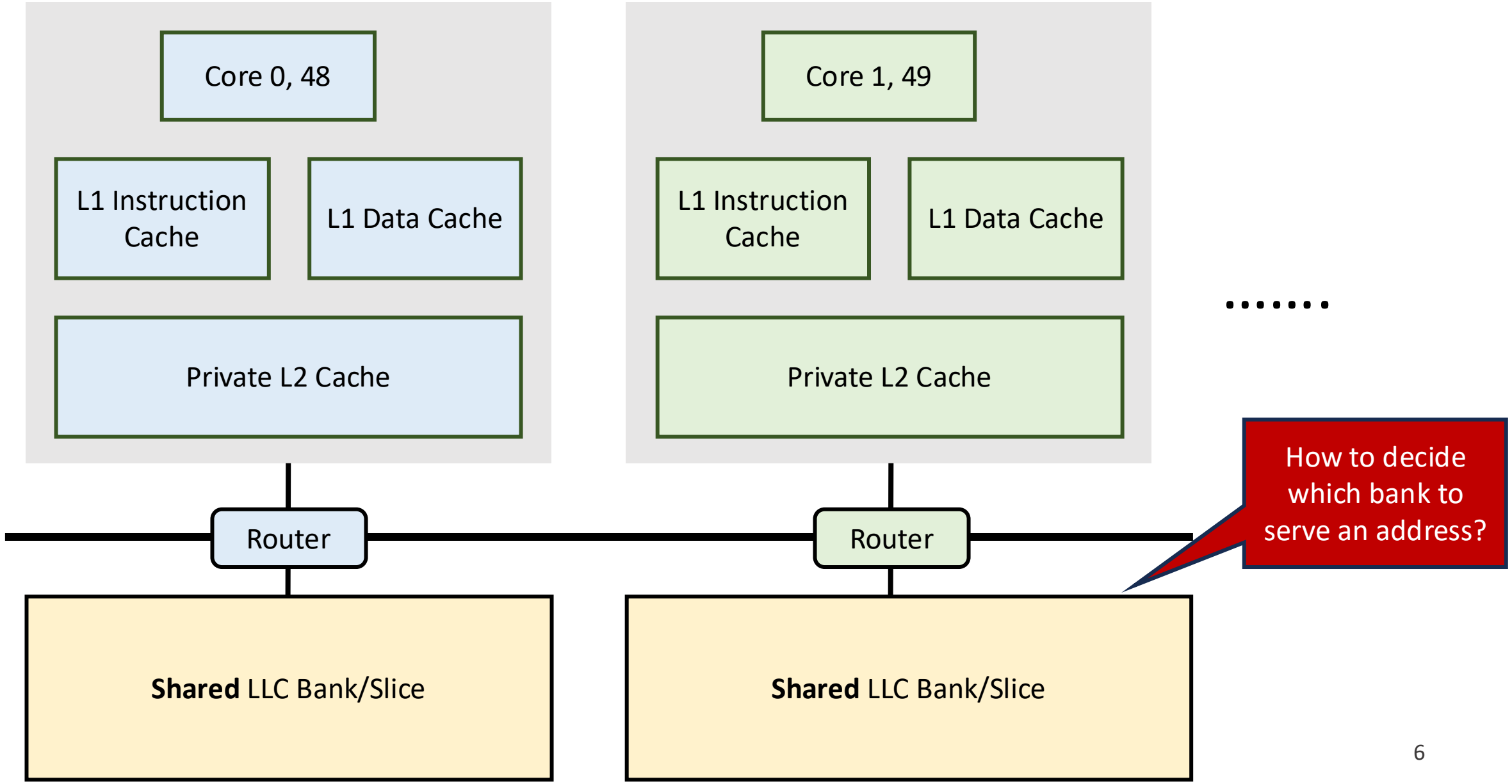
Cache operations:

- Lookup
  - A cache miss if missing L1 and L2
- Insertion
  - **Q:** To fulfill a cache miss, we insert to L1, L2, or both?
- Eviction
  - **Q:** What will happen if we have L1-D conflicts?
  - **Q:** what will happen if we have L2 conflicts?

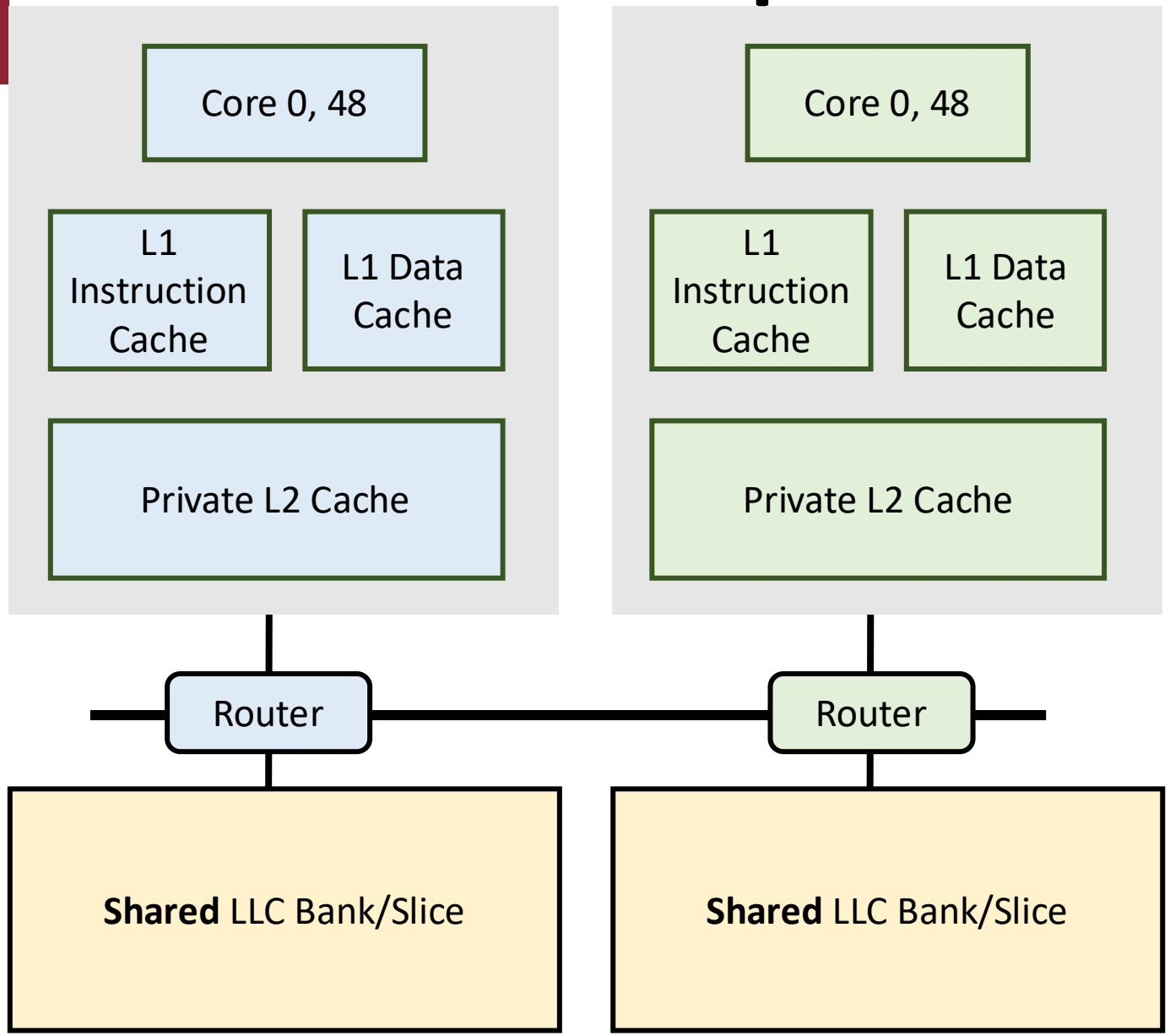
Network-on-Chip and Shared Caches



Network-on-Chip and Shared Caches



Shared Cache Operations



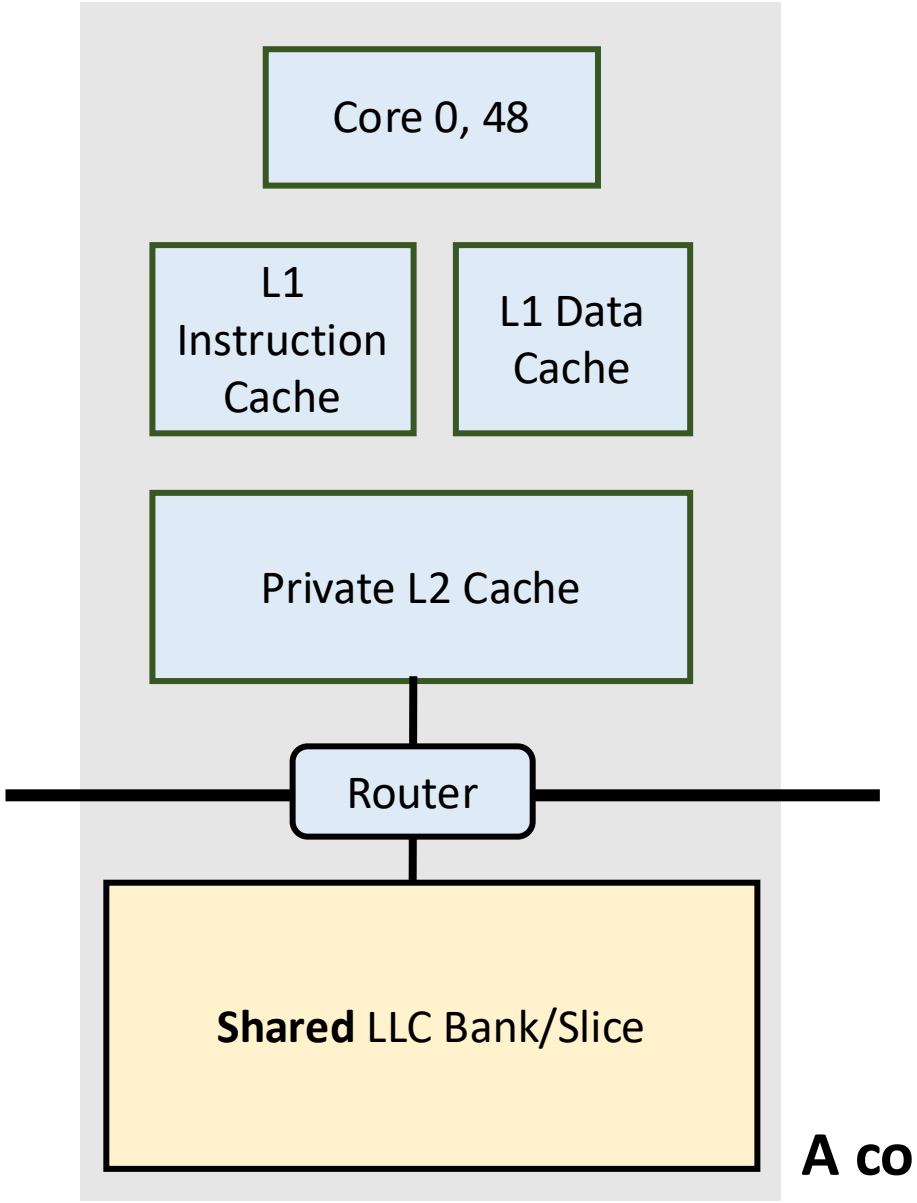
Non-inclusive cache:

- An L2 cache block **may or may not** have a duplicated copy in the L3

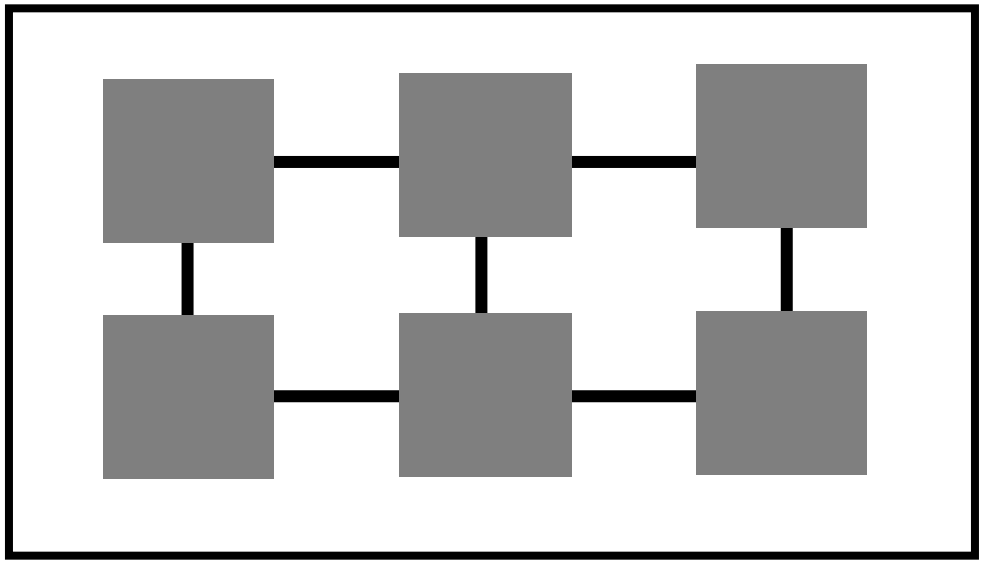
Cache operations:

- Lookup
  - **Q:** Do we have a cache miss if we have an L3 miss?
- Insertion
  - **Q:** To fulfill a cache miss, we insert to L1, L2, L3 or all of them?
- Eviction
  - **Q:** What will happen if we have an L2 conflict?
  - **Q:** what will happen if we have an L3 conflict?

Composing A Chip

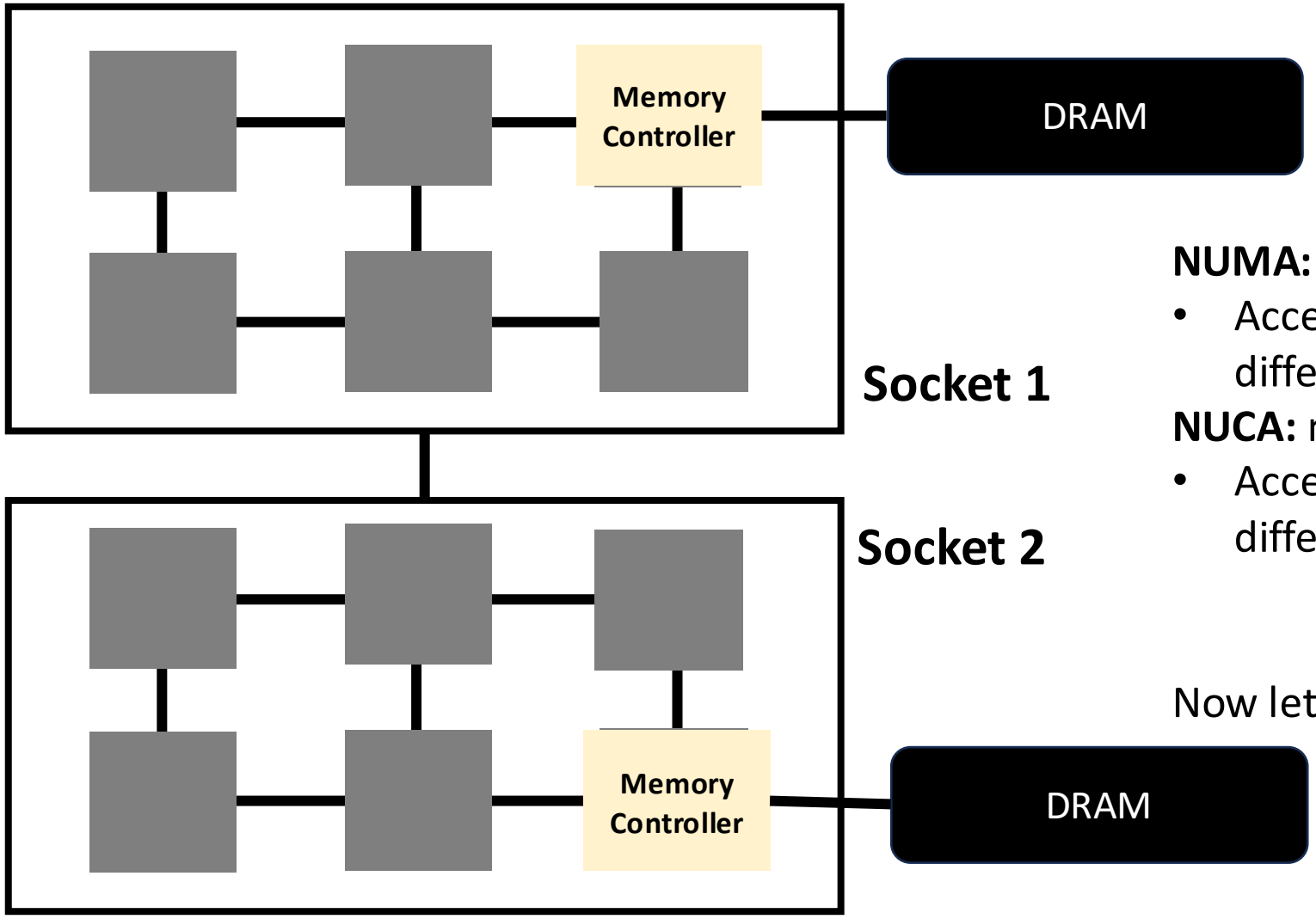


A compute tile



A chip/socket

Multi-socket Machine



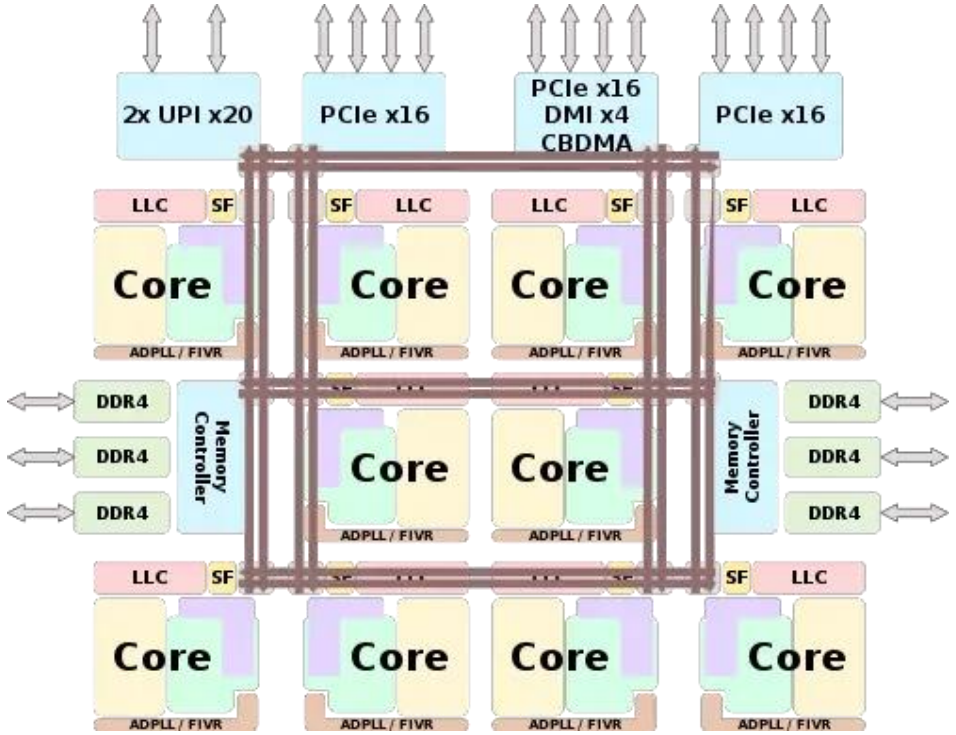
NUMA: non-uniform memory access

- Accesses to different memory banks take different latencies

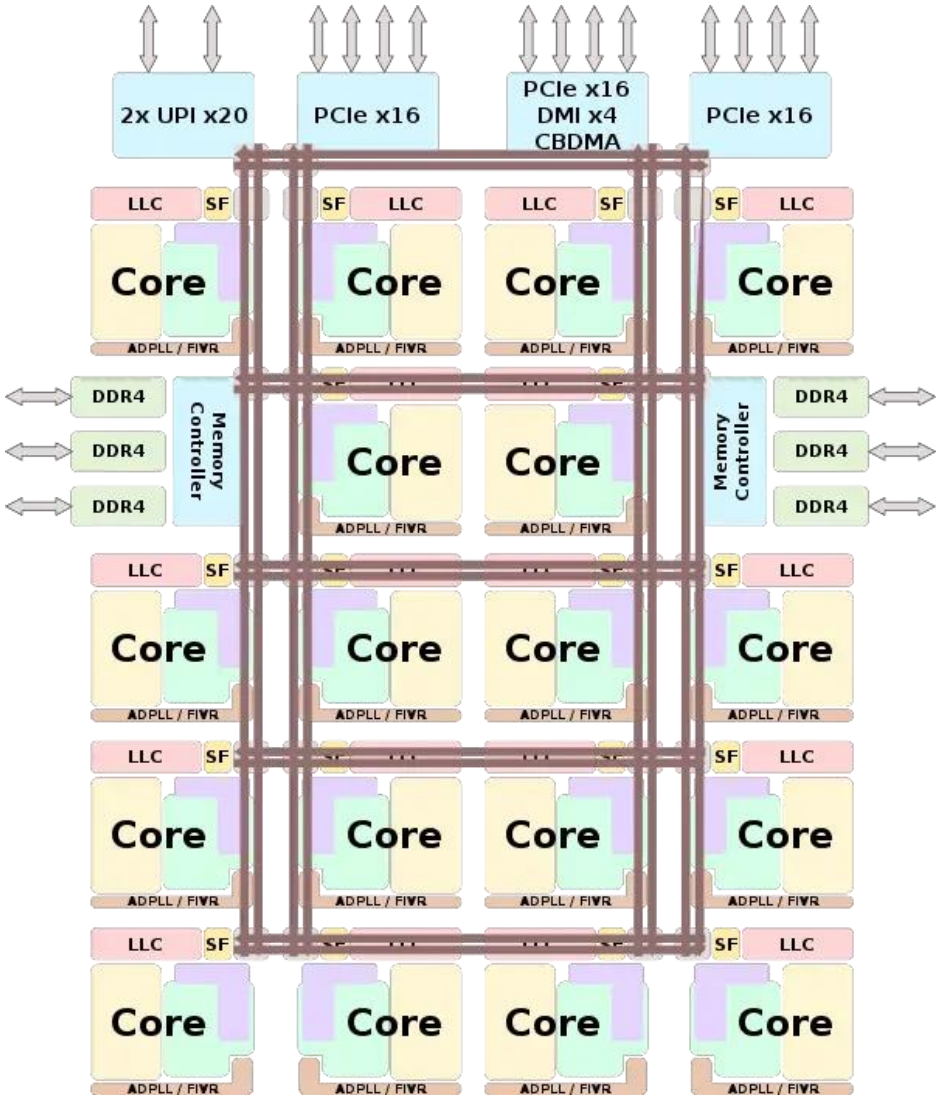
NUCA: non-uniform cache access

- Accesses to different LLC banks take different latencies

Now let's explain the flush+reload attack.

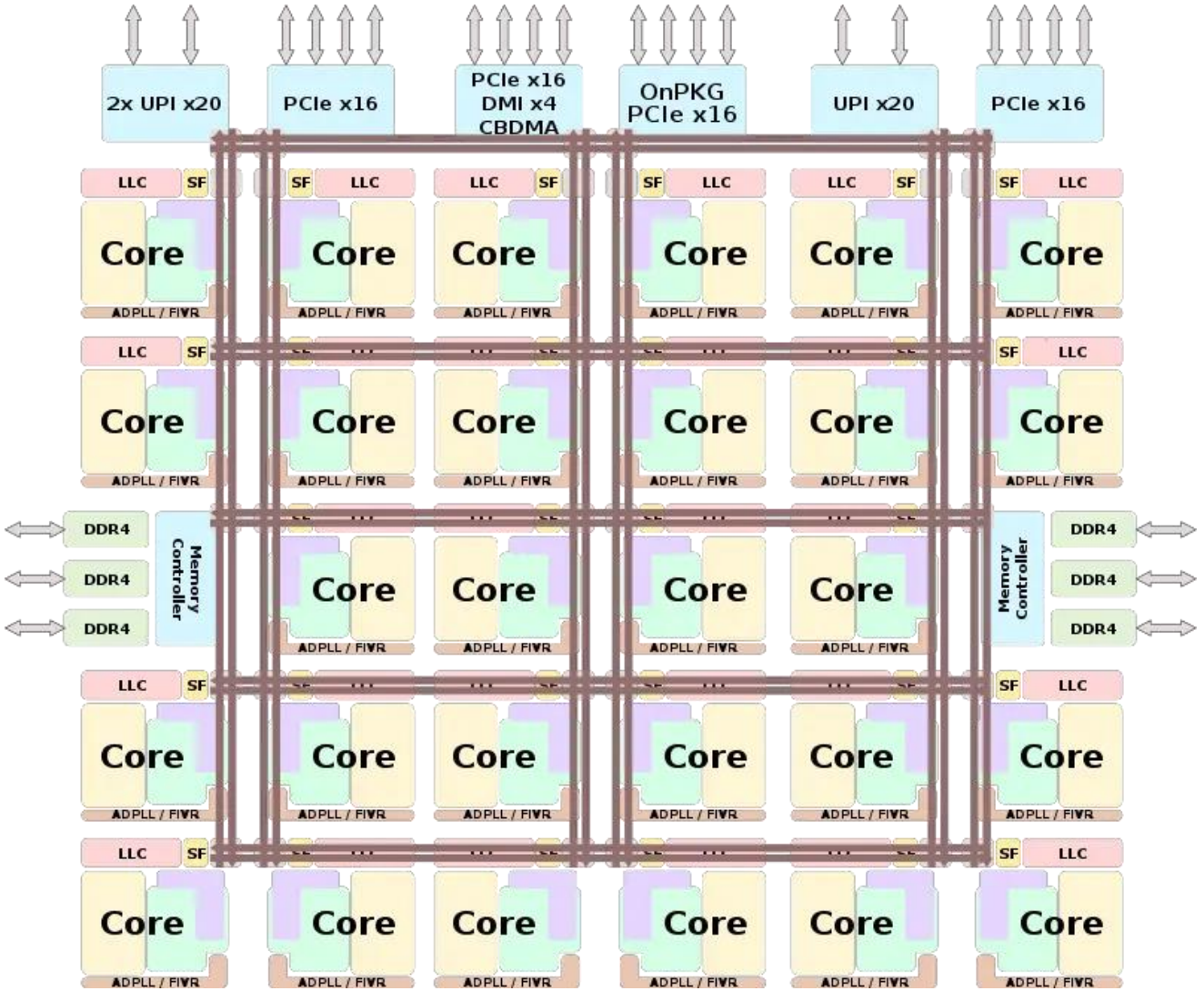


10-core (4x3)



18-core (4x5)

Pictures from wikichip



28-core (6x5)

Unicorn and dobby both have 2 sockets.

In theory: we have 28*2*2 = 112 cores

Why the cap is 96?

Pictures from wikichip

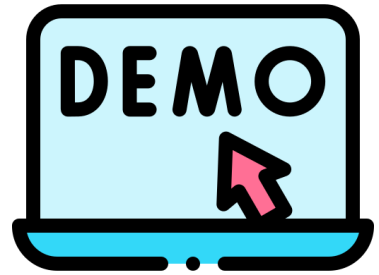
(0, 0) cpu 0 slice 0	**(0, 1)** cpu 1 slice 4	**(0, 2)** cpu 15 slice 9	**(0, 3)** cpu 16 slice 13	**(0, 4)** cpu 17 slice 17	**(0, 5)** cpu 12 slice 22
(1, 0) IMC 0	**(1, 1)** cpu 14 slice 5	**(1, 2)** cpu 9 slice 10	**(1, 3)** cpu 10 slice 14	**(1, 4)** cpu 11 slice 18	**(1, 5)** IMC 1
(2, 0) cpu 13 slice 1	**(2, 1)** cpu 8 slice 6	**(2, 2)** cpu 20 slice 11	**(2, 3)** cpu 21 slice 15	**(2, 4)** cpu 22 slice 19	**(2, 5)** cpu 23 slice 23
(3, 0) cpu 7 slice 2	**(3, 1)** cpu 19 slice 7	**(3, 2)** cpu 3 slice 12	**(3, 3)** X	**(3, 4)** cpu 5 slice 20	**(3, 5)** cpu 6 slice 24
(4, 0) slice 3	**(4, 1)** cpu 2 slice 8	**(4, 2)** X	**(4, 3)** cpu 4 slice 16	**(4, 4)** cpu 18 slice 21	**(4, 5)** slice 25

Figure 2: An example tile layout of an Intel Cascade Lake processor with 24 active cores and 26 active LLC slices.

Dai et al; Don't Mesh Around: Side-Channel Attacks and Mitigations on Mesh Interconnects; USENIX'22

Next:
Transient Execution Attacks

The Heartbeat Demo from Lecture 2



- Sender: send a heartbeat every 5 seconds

```
while(1) {
    allocate a buffer;
    sleep(5);
    free the buffer;
}
```

- Receiver: sample system status every 1 second

```
allocate a buffer;
while(1) {
    latency = time(access the buffer);
    report latency;
    sleep(1);
}
```