CS773-2022-Autumn: Computer Architecture for Performance and Security

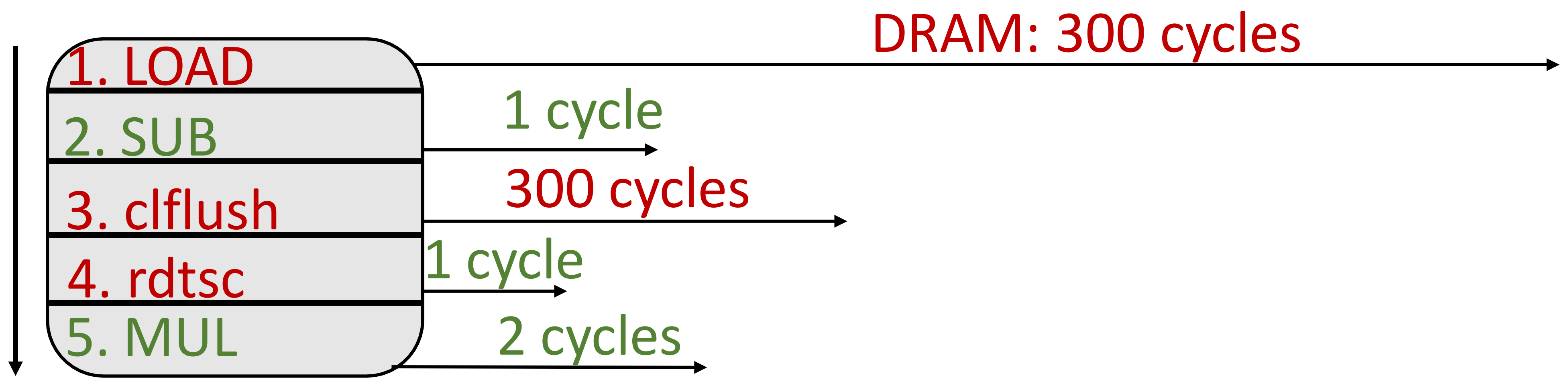
Lecture 6: More on flush attacks

https://www.cse.iitb.ac.in/~biswa/

Phones on silence, please

Thank You

The O3 effect (See the F+R code)



Out-of-order execution ☹
(Multiple fetch in one cycle)

rdtsc

```
uint64_t rdtsc_nofence() {
uint64_t a, d;
asm volatile ("rdtsc" : "=a" (a), "=d" (d));
a = (d<<32) | a;
return a;
}
```

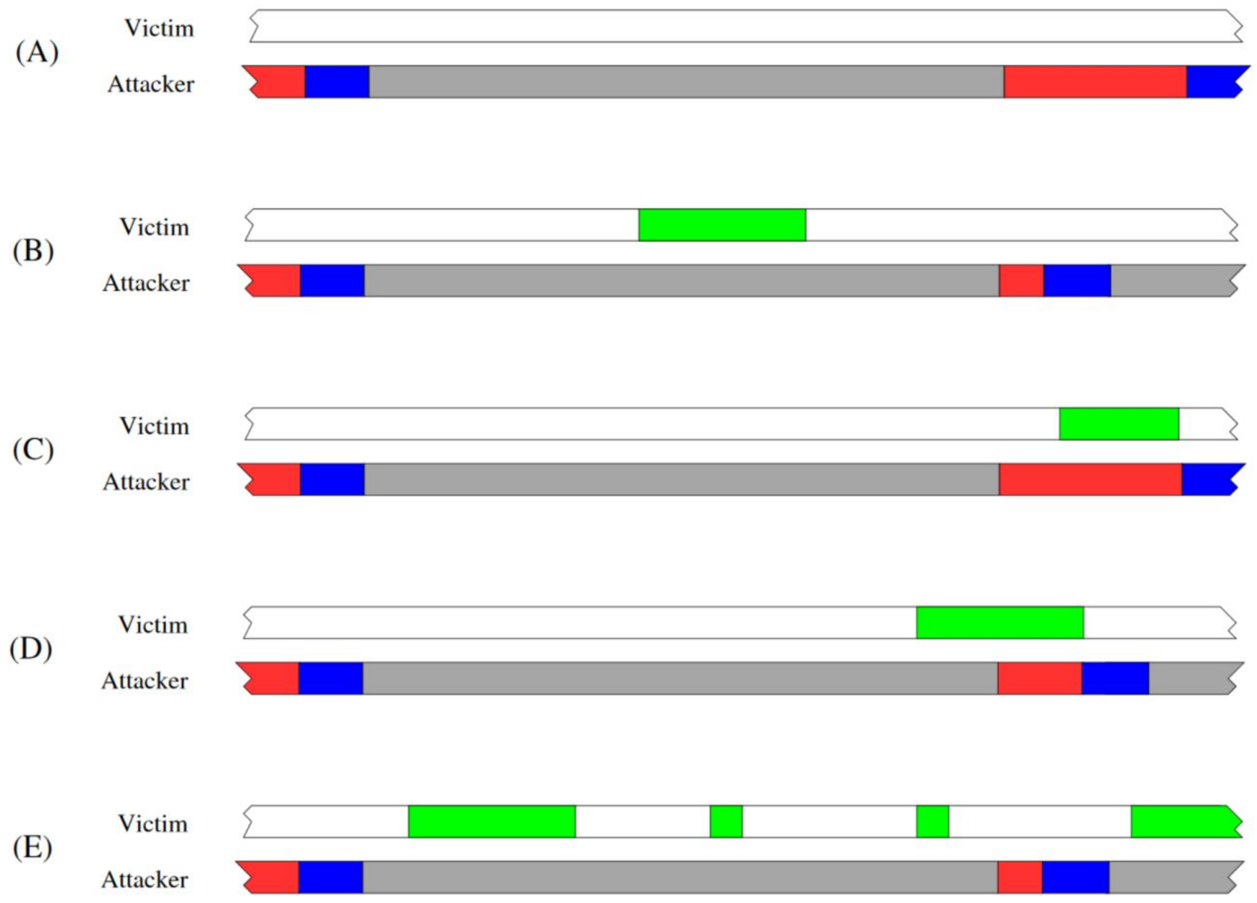
asm: To include assembly code in C/C++
rdtsc registers the clock ticks passed since the last reset, 64 bit values stored as 32-bits in EDX (upper half) and EAX (lower half)
volatile: makes sure compiler does not do anything stupid ☺

rdtsc with fence

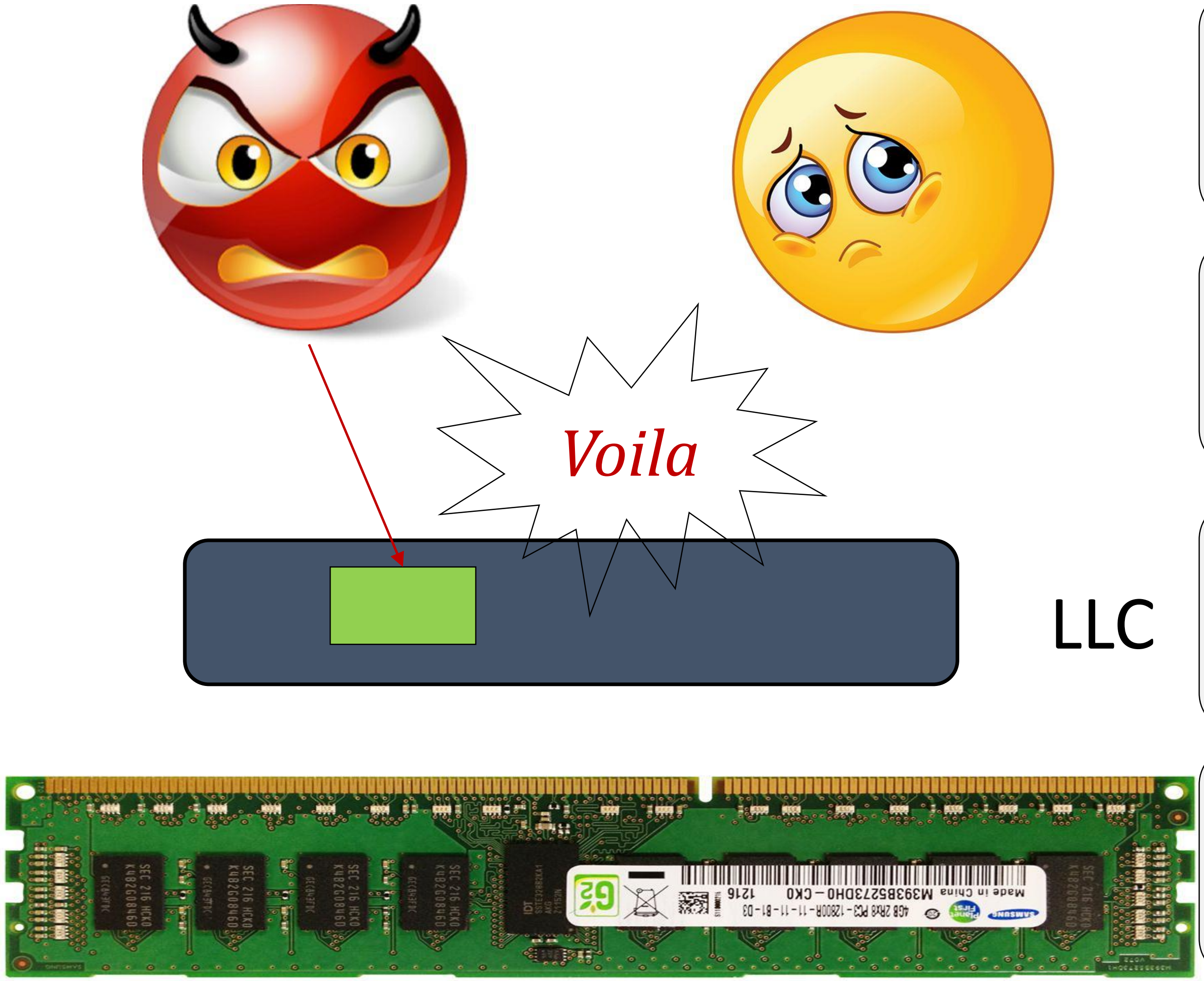
```
uint64_t rdtsc() {
uint64_t a, d;
asm volatile ("mfence");
asm volatile ("rdtsc" : "=a" (a), "=d" (d));
a = (d<<32) | a;
asm volatile ("mfence");
return a;
}
```

There are lfence, sfence too.

The notion of Time (an agreement in covert channel)



Flush&Flush attack [how does it work]



Step 0:Spy *maps* the shared library, shared in the cache

Step 1:Spy *flushes* the cache block

Step 2: Victim *reloads* the cache block

Step 3: Spy *flushes* the cache block again

Job of an attacker

CALIBRATION; FOR LATENCY THRESHOLD

FIND OUT ADDRESSES OF INTEREST

BITS OF INTEREST

CASPER

How good is the attacker

Bandwidth
synchronous/asynchronous

Accuracy (error rate)

Stealthy and agility

CASPER

In-class discussion time