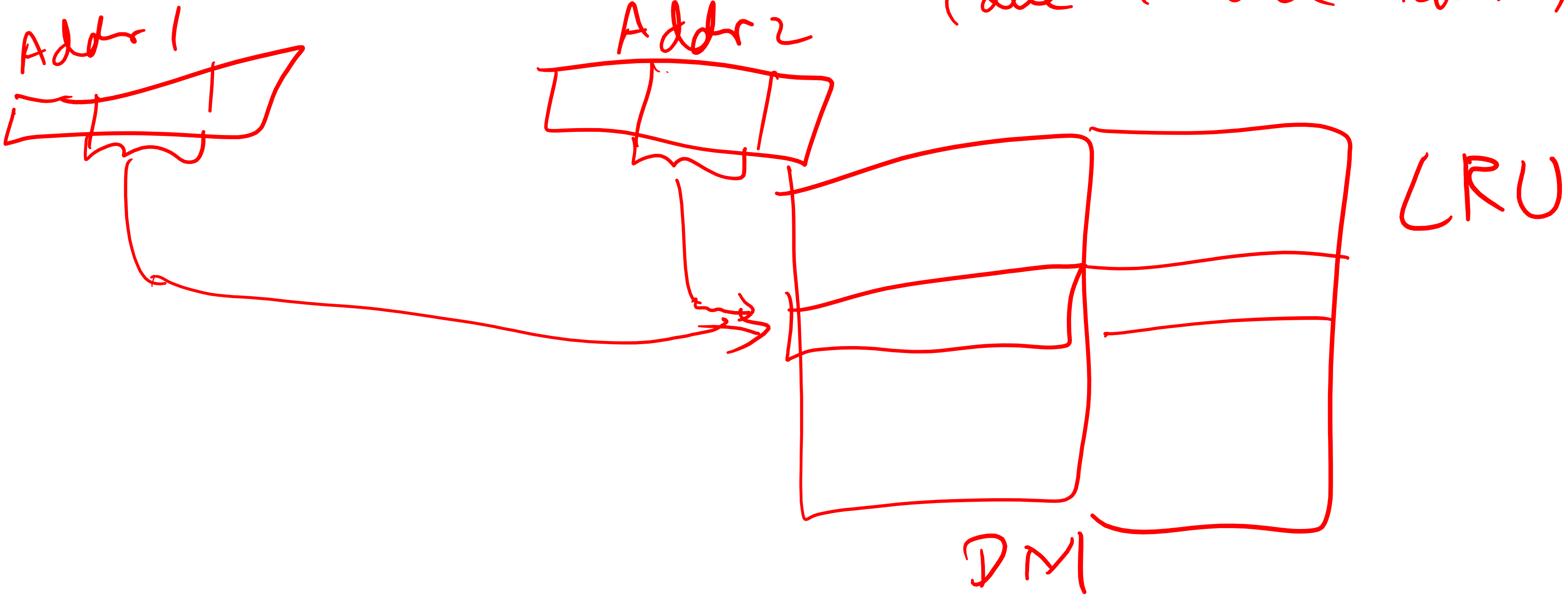
# Lecture 24: Memory, Security

- Today's topics:
  - Main memory system
  - Hardware security intro

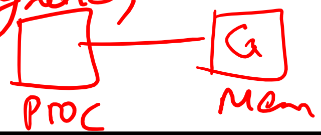
HW 9 due tomorrow

HW 10 posted Thursday
(due a week later)



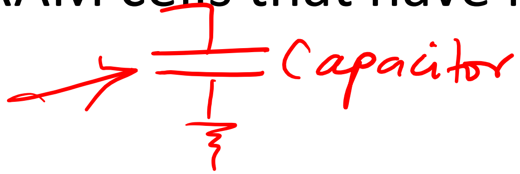
Reg (Latches)  Caches (SRAM)  Mem (DRAM)  Mem (NVM)  SSD (Flash)  HDD (magnetic)

# Off-Chip DRAM Main Memory



- Main memory is stored in DRAM cells that have much higher storage density

volatile

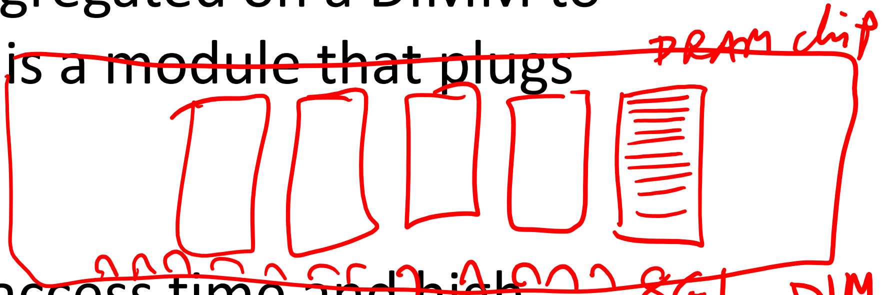


- DRAM cells lose their state over time – must be refreshed periodically, hence the name *Dynamic*

64 ms

Dual

- A number of DRAM chips are aggregated on a DIMM to provide high capacity – a DIMM is a module that plugs into a bus on the motherboard



8Gb DIMM
=1GB

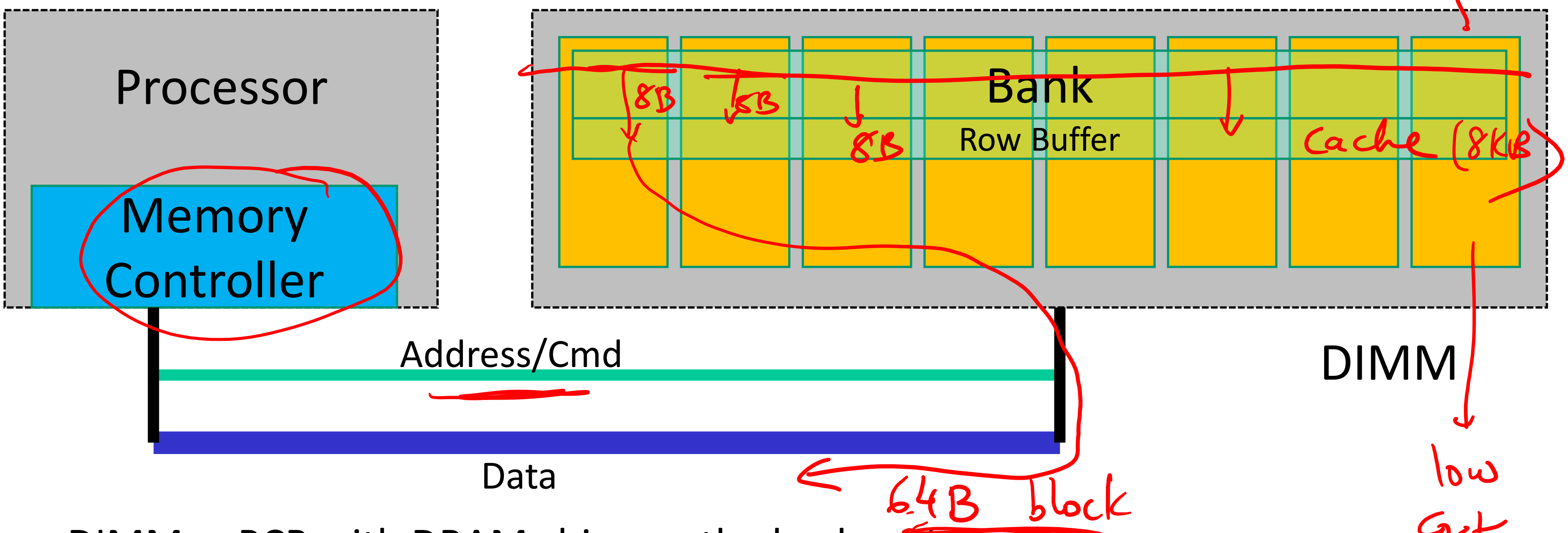
- DRAM access suffers from long access time and high energy overhead

300 cycles

NVM → Optane

DDR standard

# Memory Architecture



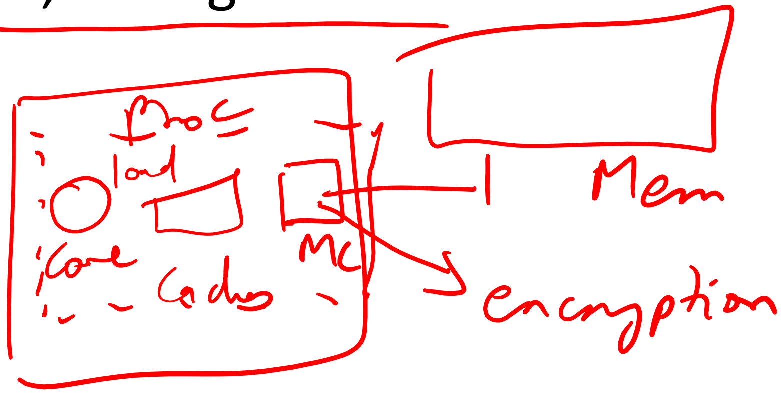
- DIMM: a PCB with DRAM chips on the back and front
- The memory system is itself organized into ranks and banks; each bank can process a transaction in parallel
- Each bank has a row buffer that retains the last row touched in a bank (it’s like a cache in the memory system that exploits spatial locality) (row buffer hits have a lower latency than a row buffer miss)

# Hardware Security

- Software security: key management, buffer overflow, etc.
- Hardware security: hardware-enforced permission checks, authentication/encryption, etc.
- Information leakage, side channels, timing channels
- Meltdown, Spectre, SGX

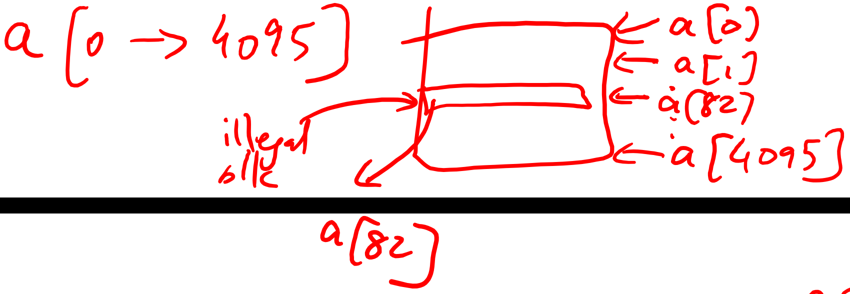
Intel

Threat Model



OS could be malicious

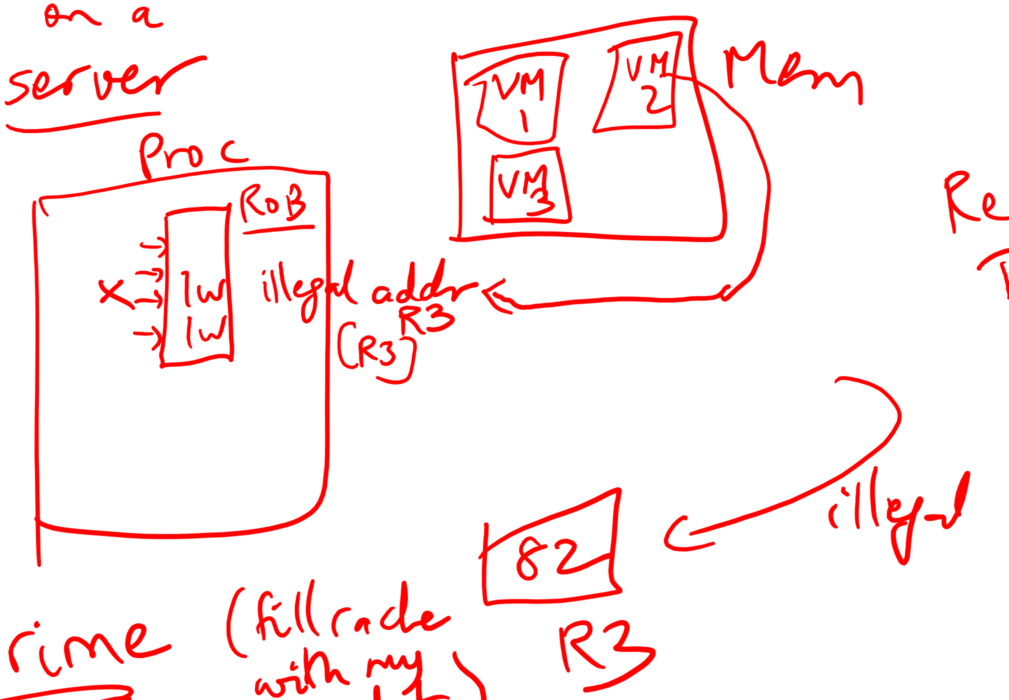
# Meltdown

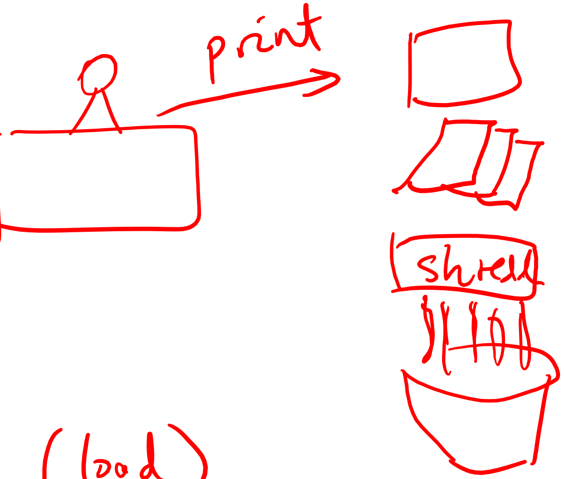


Attacker

on a

server

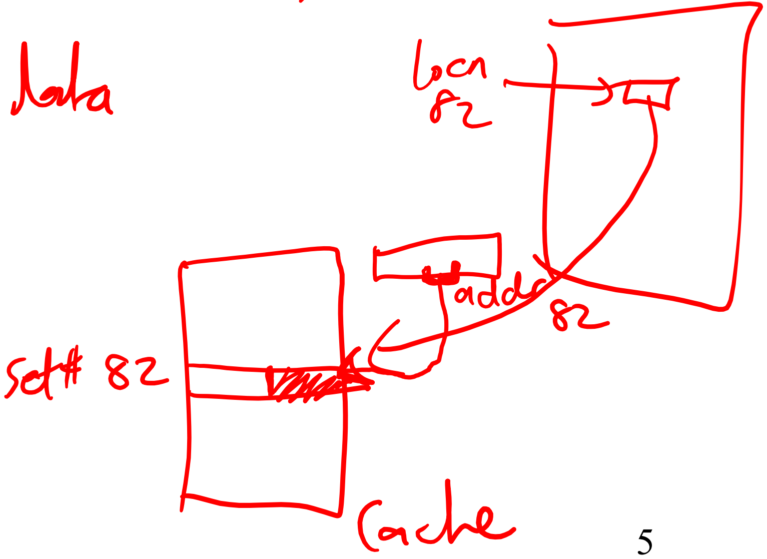




Reg

TaxD

(load)



prime (fill cache with my own data)

execute

lw R3 <- illegal addr

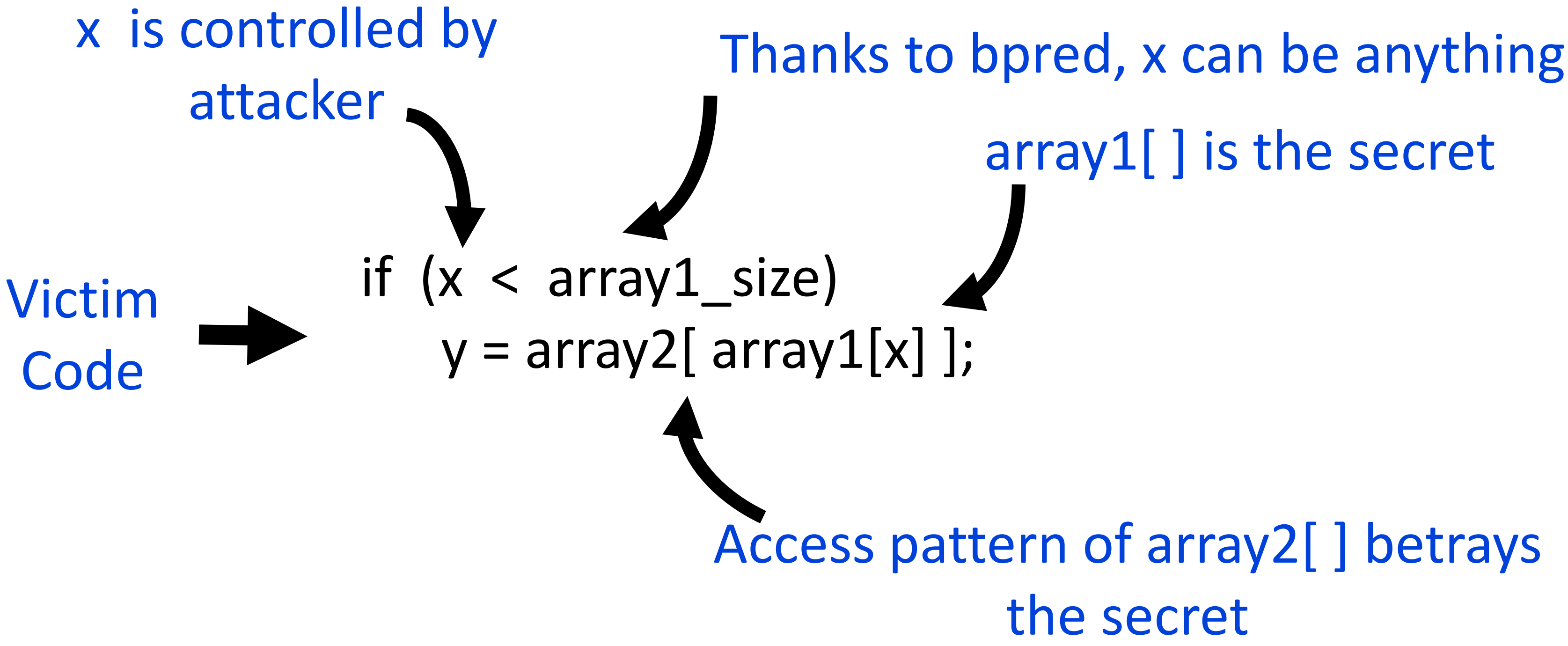
lw [R3]

Probe

lw

[R3]

# Spectre: Variant 1



x is controlled by attacker

Thanks to bpred, x can be anything

array1[ ] is the secret

Victim Code

```
if (x < array1_size)
    y = array2[ array1[x] ];
```

Access pattern of array2[ ] betrays the secret

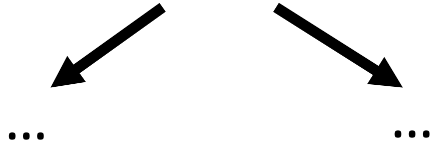
# Spectre: Variant 2

Victim code

R1 ← (from attacker)

R2 ← some secret

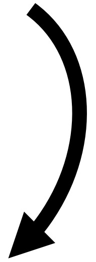
Label0: if (…)



…  …

Attacker code

Label0: if (1)



Label1: …

Victim code

```
Label1:
        lw [R2]
```