Ausnutzung von CPU-Sicherheitslücken

Meltdown

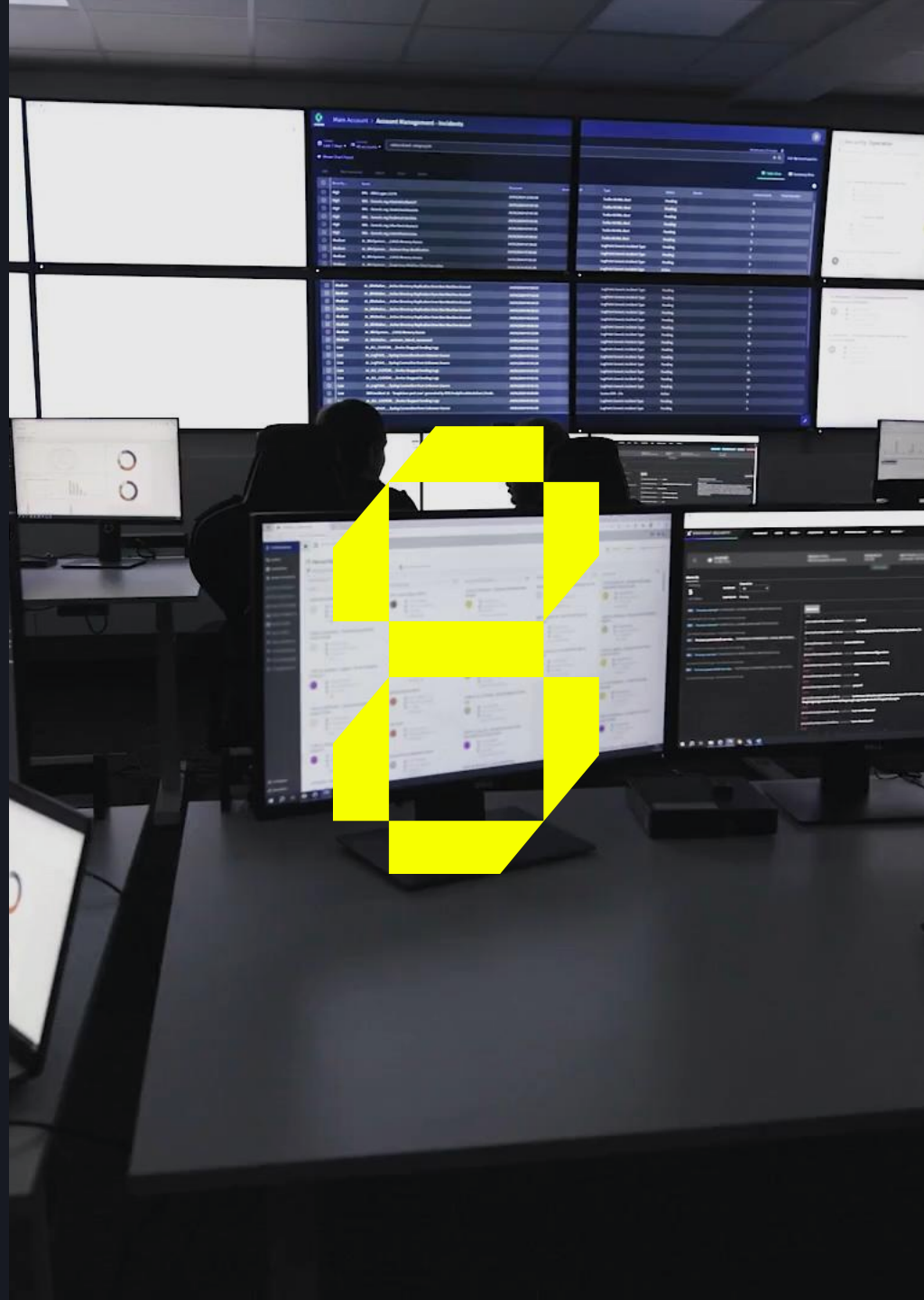
8COM
CYBER SECURITY

TECH SESSION!

Ausnutzung von CPU-Sicherheitslücken
Meltdown

Tobias Kopf
Head of Penetration Testing

8COM
CYBER SECURITY

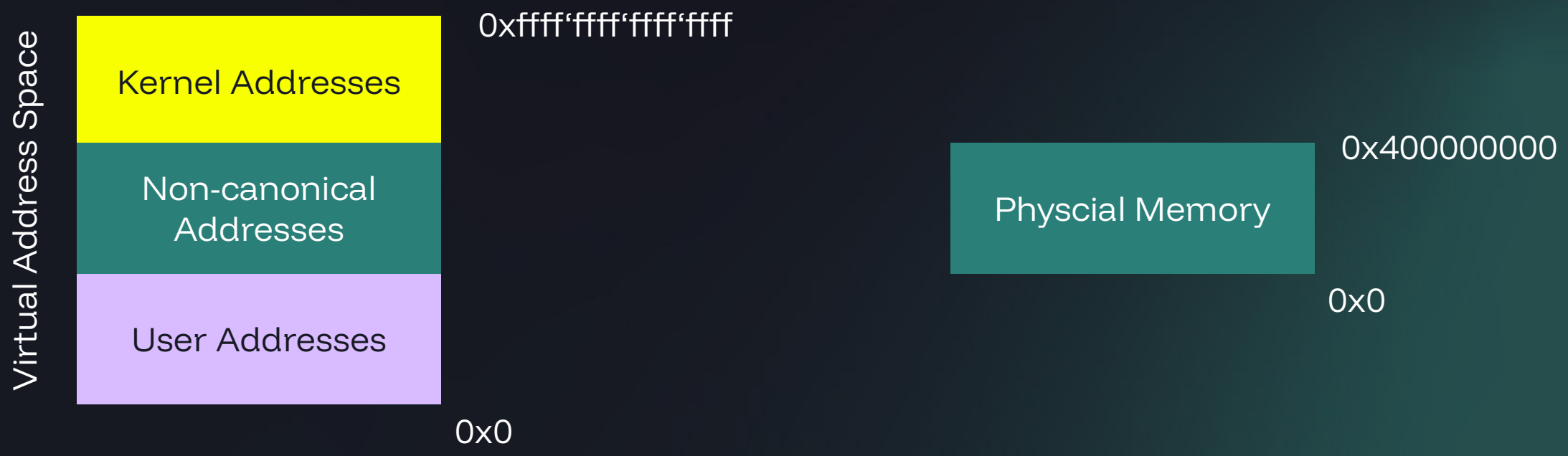


Meltdown

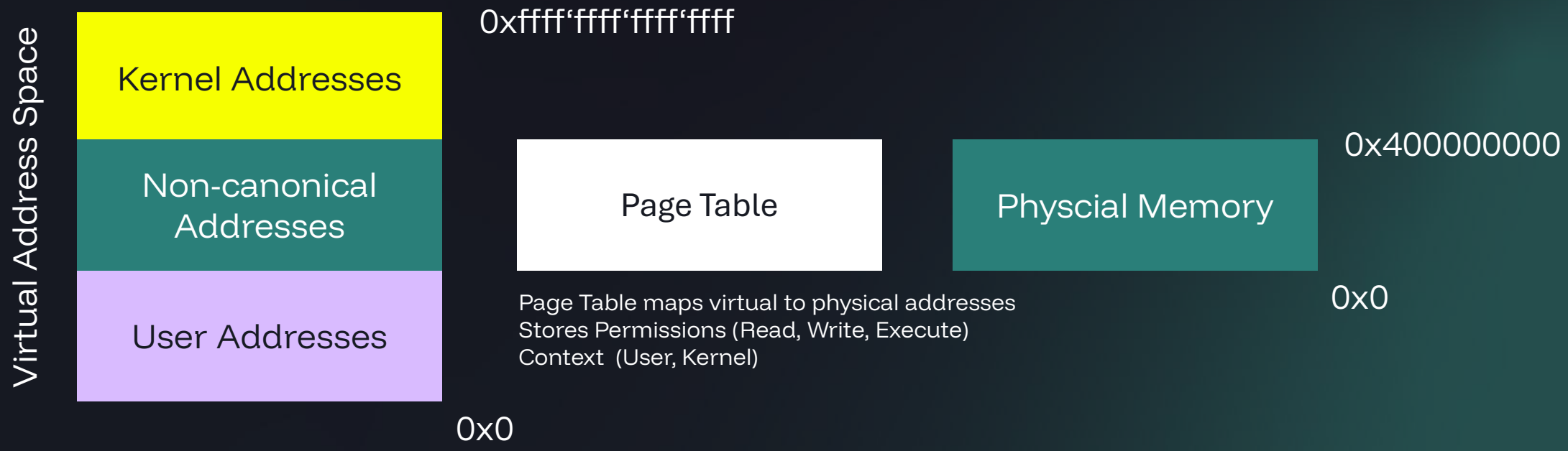
8COM

- CPU vulnerability that breaks fundamental isolation between OS and applications
- Break KASLR, read all memory, dump hashes, etc.
- All Intel and some ARM CPUs are affected
- Mitigation possible via microcode and OS updates (PTI)

Virtual Memory



Virtual Memory



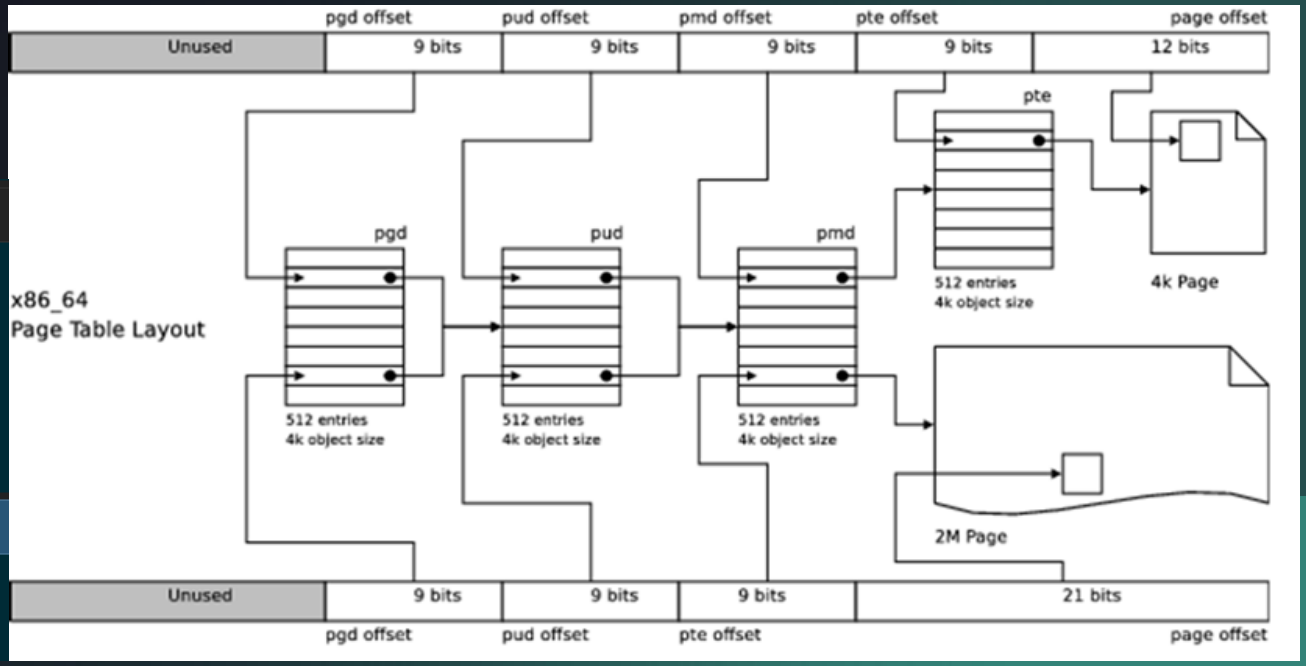
Page Table

8COM

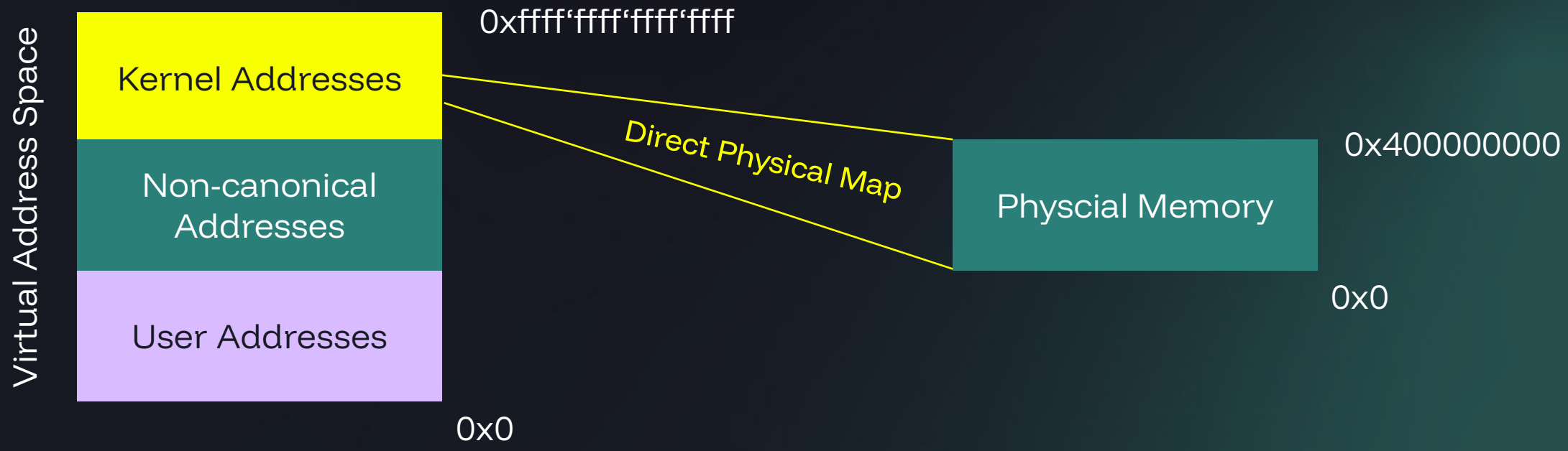
- Isolation of memory between processes and user/kernel
- E.g. User can’t access kernel memory @ 0xffff….
  - Leads to a segmentation fault

```
Terminal
void main() {
  char data = *(char*) 0xffff888000000000;
  printf("%c\n", data);
}
```

```
Terminal
❯ ./segfault
zsh: segmentation fault (core dumped)  ./segfault
```



Direct Physical Map



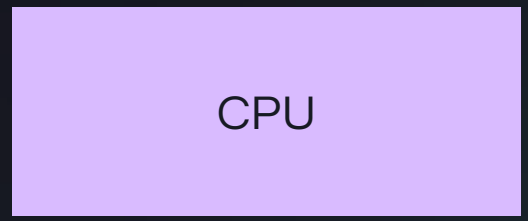
Side-Channel Attack: Flush+Reload

Flush+Reload

8COM

- CPU stores recently accessed variables in cache

```
println!("{var}");
println!("{var}");
```

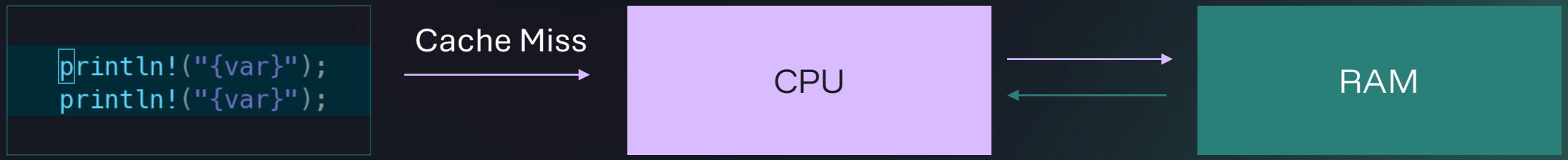


Side-Channel Attack: Flush+Reload

Flush+Reload

8COM

- CPU stores recently accessed variables in cache

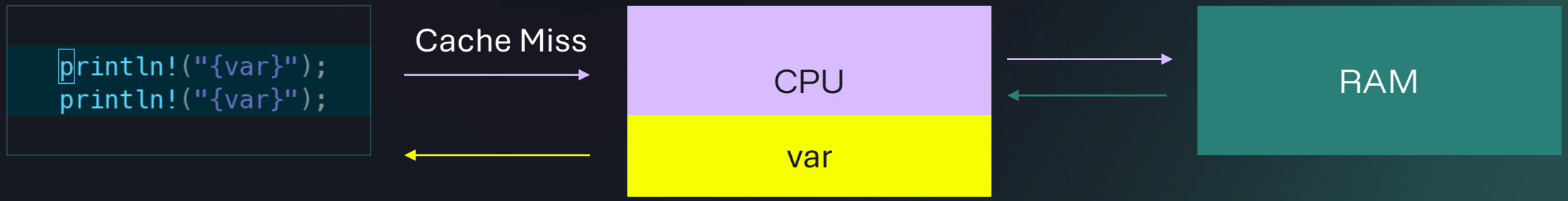


Side-Channel Attack: Flush+Reload

Flush+Reload

8COM

- CPU stores recently accessed variables in cache

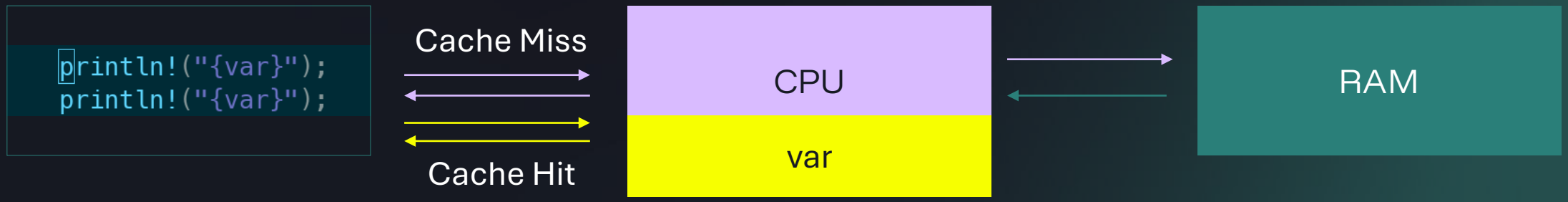


Side-Channel Attack: Flush+Reload

Flush+Reload

8COM

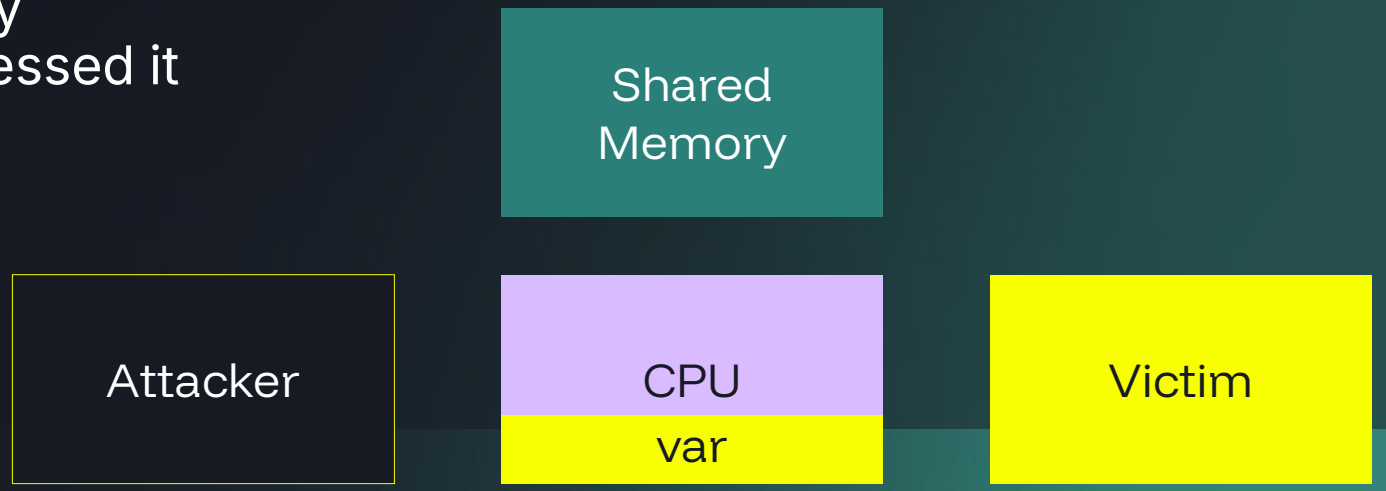
- CPU stores recently accessed variables in cache
- Latency can be measured (4 to 80 cycles for cache hit, 200+ for cache miss)



Side-Channel Attack: Flush+Reload

Flush+Reload

- Attacker and victim have shared memory
- Attacker flushes shared memory out of cache
- Victim may access it
- Attacker can measure latency to check whether victim accessed it



Microarchitecture

8COM

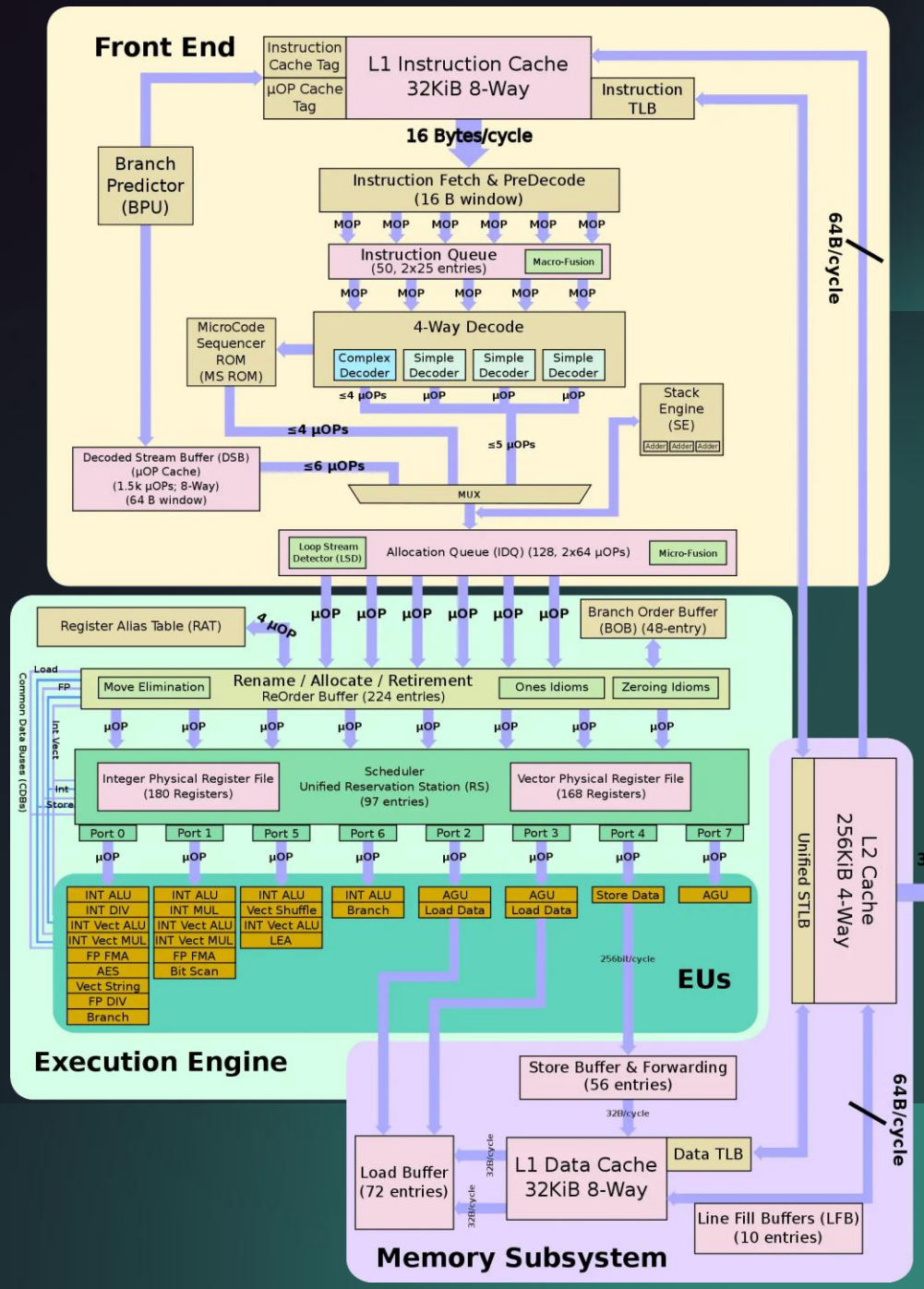
- Implementation of Instruction Set
- Instruction Set may be ARM, x86, MIPS, etc...
- Microcode can be updated to enable/disable features or fix bugs

```
60fa:       e8 d1 f6 ff ff          call   0x57d0
60ff:       ba 05 00 00 00          mov    edx,0x5
6104:       48 8d 35 dd 13 00 00    lea    rsi,[rip+0x13dd]
610b:       31 ff                   xor    edi,edi
610d:       48 89 c5                mov    rbp,rax
6110:       ff 15 5a 3d 00 00       call   QWORD PTR [rip+0x3d5a]
6116:       48 89 c3                mov    rbx,rax
6119:       ff 15 f9 3c 00 00       call   QWORD PTR [rip+0x3cf9]
611f:       48 89 e9                mov    rcx,rbp
6122:       48 89 da                mov    rdx,rbx
```

Example of x86 instructions

Single Skylake CPU Core

- Front End: Fetches Intructions from L1 Instruction Cache & Decodes them into microOps
- Execution Engine: Performs optimizations on microOps (rename, retire, reordering) & sends microOps to ports to execute (Tomasulo's algorithm)



8COM

Out-of-order Execution

8COM

- Reduce pipeline stalling, e.g. perform slow load operation from memory
- inc rsi and add rsi, rcx are independent of rax
  → can be performed while waiting on memory access @ rbx + 8

```
mov rax, [rbx + 8]      # move value from memory @ rbx+8 into rax
inc rsi                 # increase rsi by 1
add rsi, rcx            # rsi = rsi + rcx
```

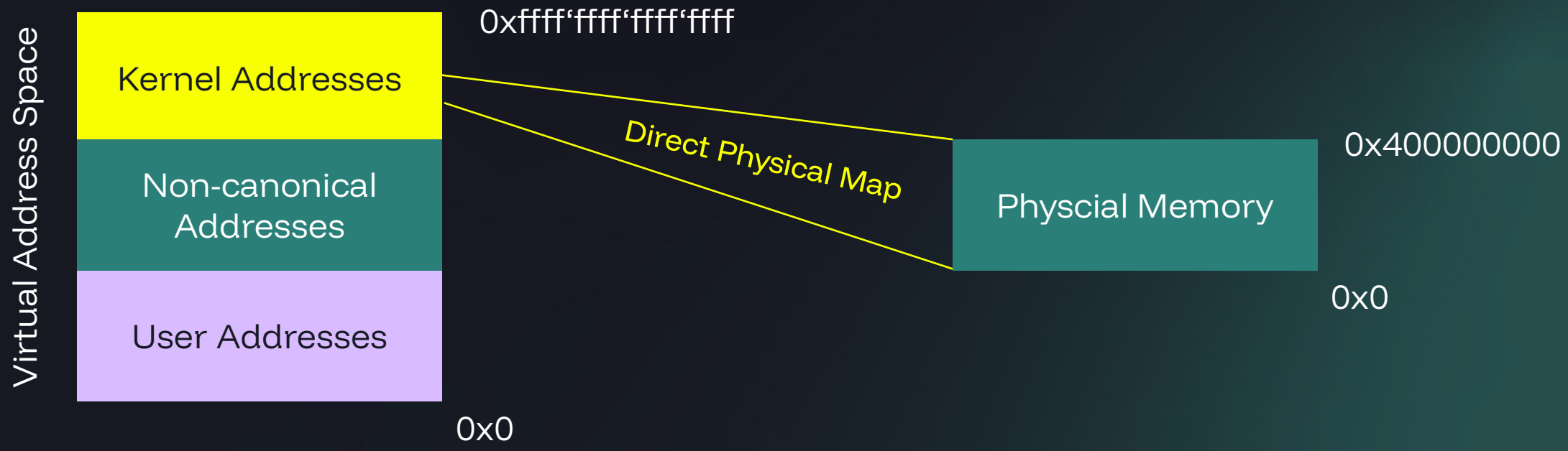
Out-of-order Execution

8COM

- 4: Try to read a byte of kernel memory into register al → Triggers SEGFAULT
- 5: Multiply al by 4096 (Page Size)
- 7: Transient Execution of accessing probe array with rax as offset
- Use Flush+Reload to see which page of probe array was accessed

```
1 ; rcx = kernel address
2 ; rbx = probe array
3 retry:
4 mov al, byte [rcx]
5 shl rax, 0xc
6 jz retry
7 mov rbx, qword [rbx + rax]
```

Direct Physical Map



Demo

8COM

Mitigations

Kernel Page-Table Isolation (KPTI)

- Don‘t map kernel address space and physical address space into process in user mode
- Needs flushing of translation-lookaside buffer (TLB) → Performance impact

Add Jitter to Timing Measurements

- Questionable

Further Resources

8COM

Meltdown Paper

Wikichip Skylake

Sie haben noch Fragen?

Sprechen Sie mich gerne an!

8COM
CYBER SECURITY

Tobias Kopf

Head of Penetration Testing

tobias.kopf@8com.de

+49 171 1522283

www.8com.de

8com.de/newsletter