SECURITY SESSION

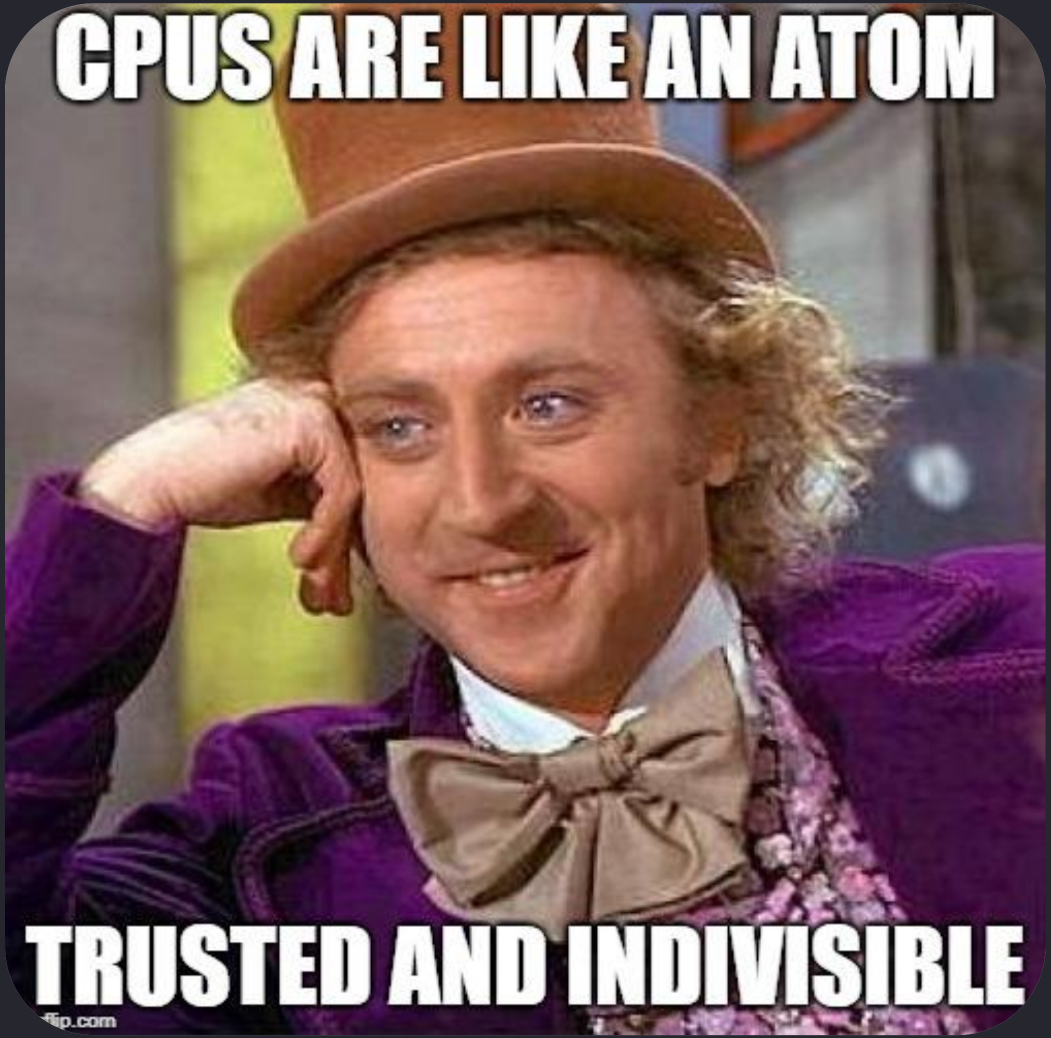
Wintel Hell 2

Melting point

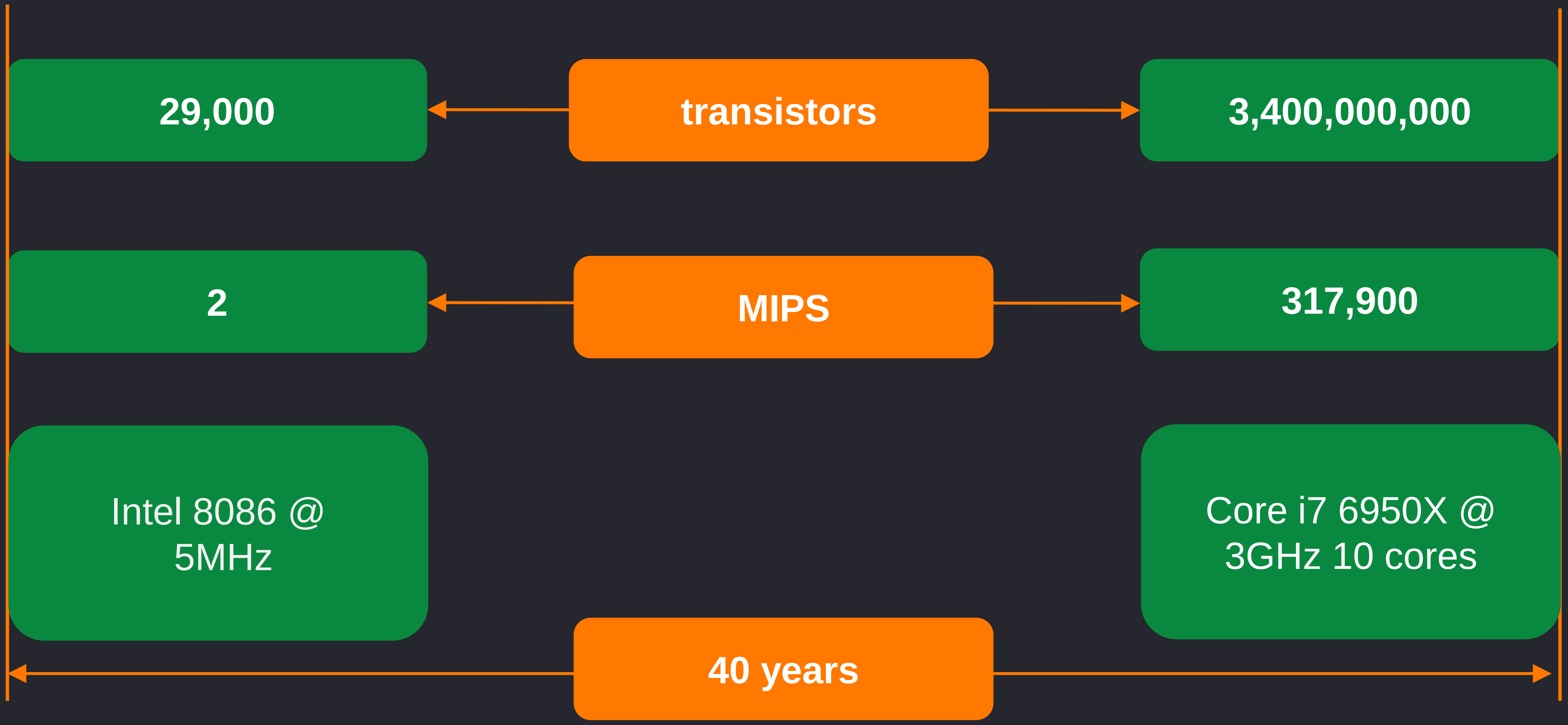
Martin Hron, research @ avast

avast

Matter of trust



Matter of trust : evolution



29,000

transistors

3,400,000,000

2

MIPS

317,900

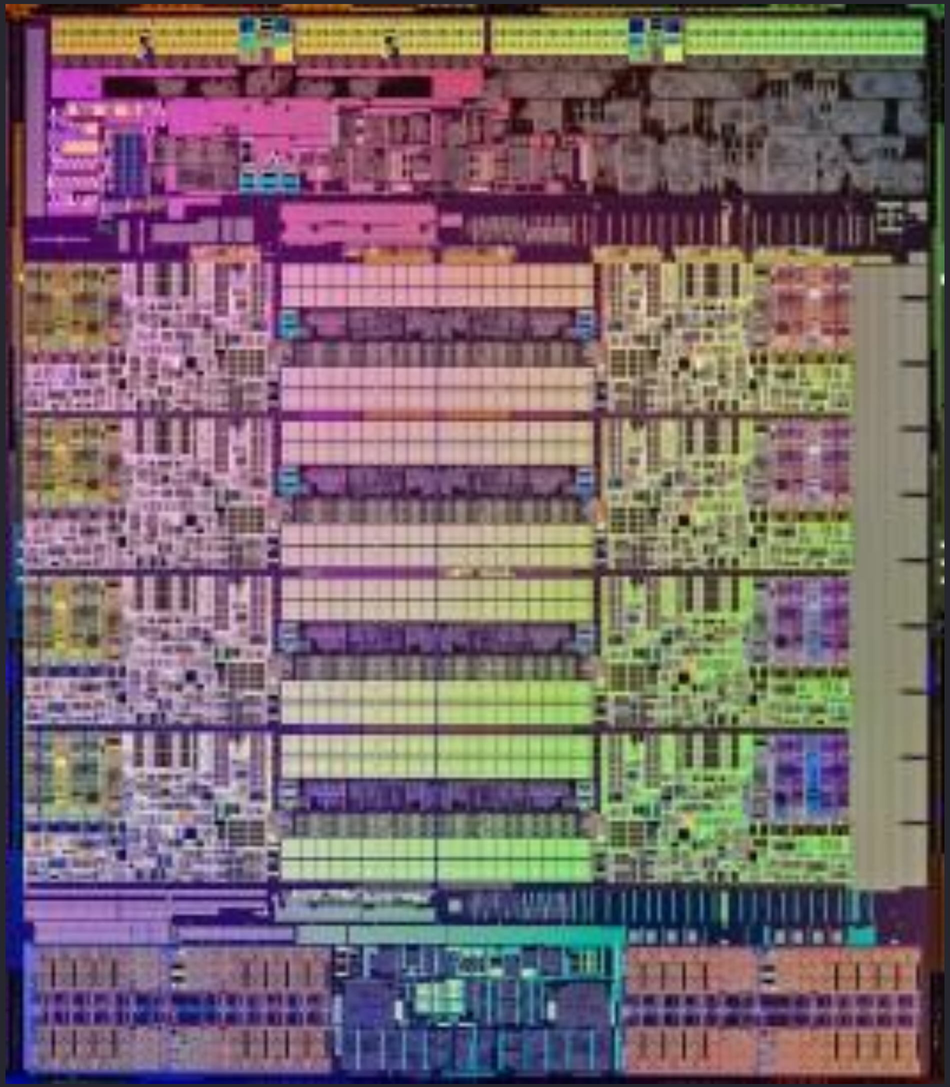
Intel 8086 @ 5MHz

Core i7 6950X @ 3GHz 10 cores

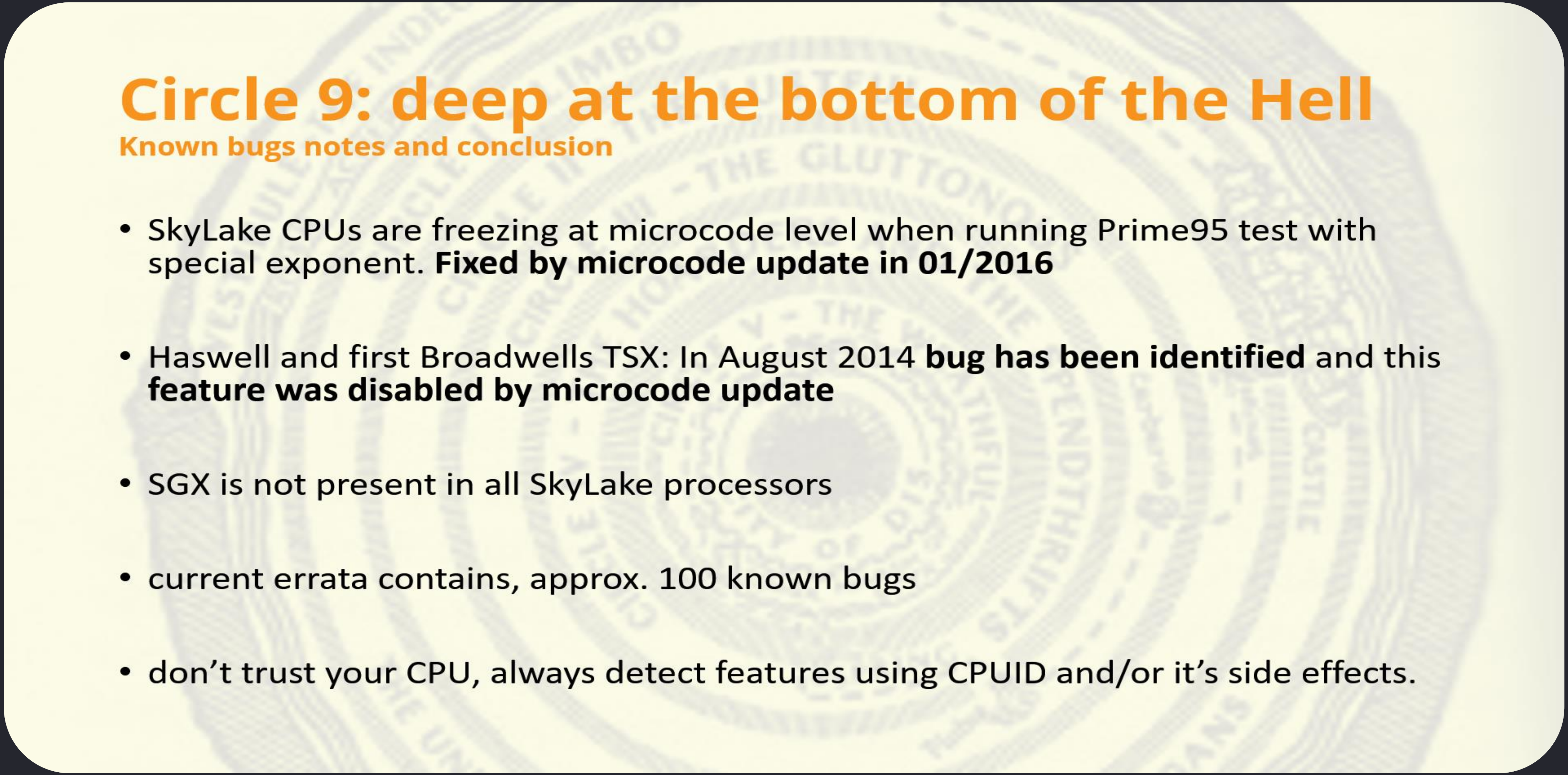
40 years

Matter of trust

- Do you trust your CPU?
- Complex beasts with microcode, simultaneous processing and state machines
- Do you read CPU manual when doing low level stuff?
- Have you ever read errata?



Matter of trust : flashback : Security Session 2016

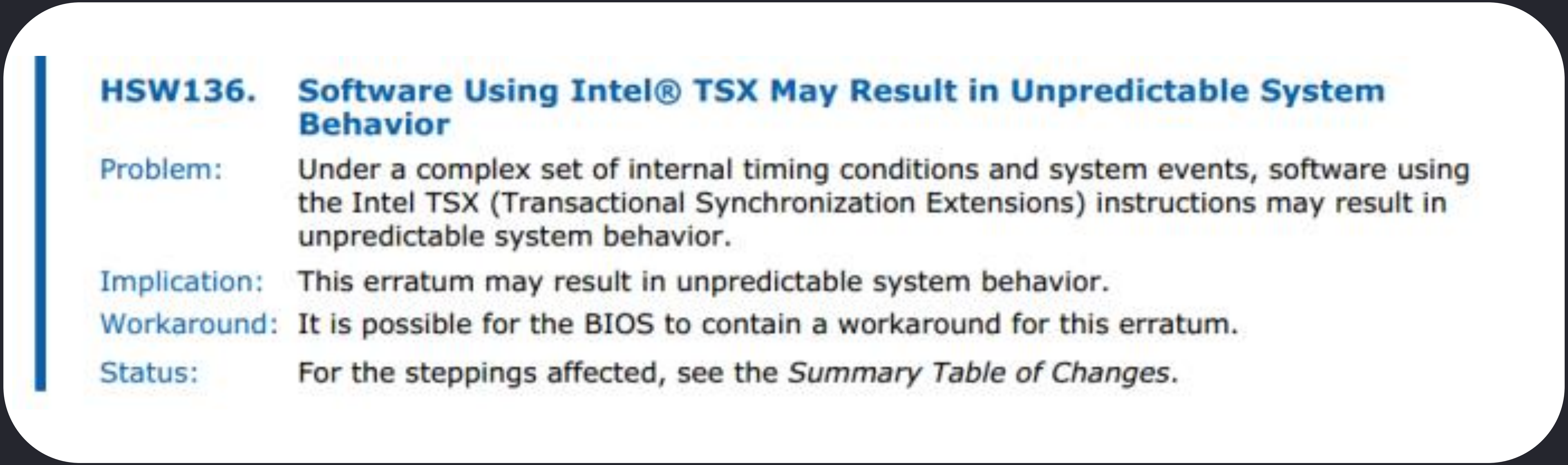


Circle 9: deep at the bottom of the Hell

Known bugs notes and conclusion

- SkyLake CPUs are freezing at microcode level when running Prime95 test with special exponent. **Fixed by microcode update in 01/2016**
- Haswell and first Broadwells TSX: In August 2014 **bug has been identified** and this **feature was disabled by microcode update**
- SGX is not present in all SkyLake processors
- current errata contains, approx. 100 known bugs
- don’t trust your CPU, always detect features using CPUID and/or it’s side effects.

Matter of trust



HSW136. Software Using Intel® TSX May Result in Unpredictable System Behavior

Problem: Under a complex set of internal timing conditions and system events, software using the Intel TSX (Transactional Synchronization Extensions) instructions may result in unpredictable system behavior.

Implication: This erratum may result in unpredictable system behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*.

HASWELL errata has 173 items on 6 pages

Most of them have NO FIX

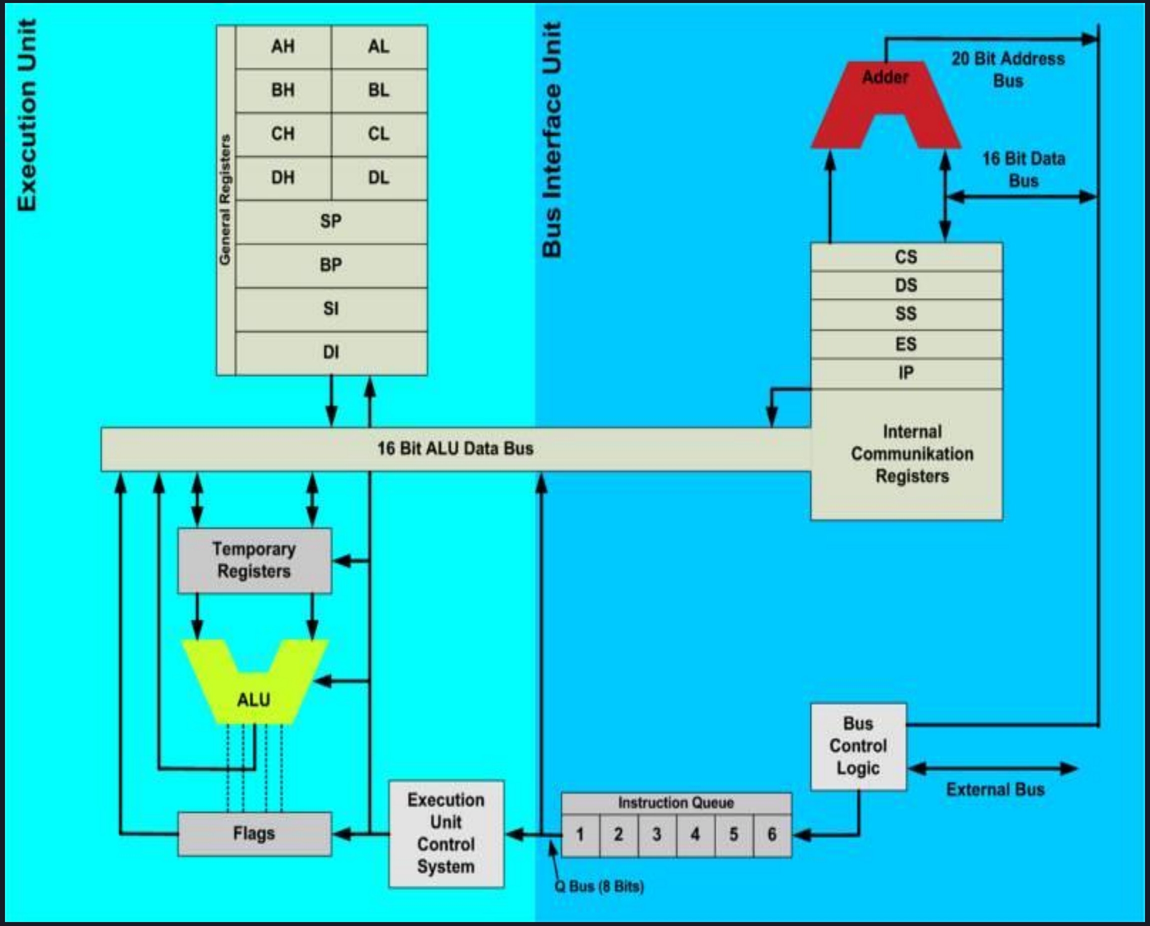
Matter of trust

RARE CONDITIONS ARE RARE

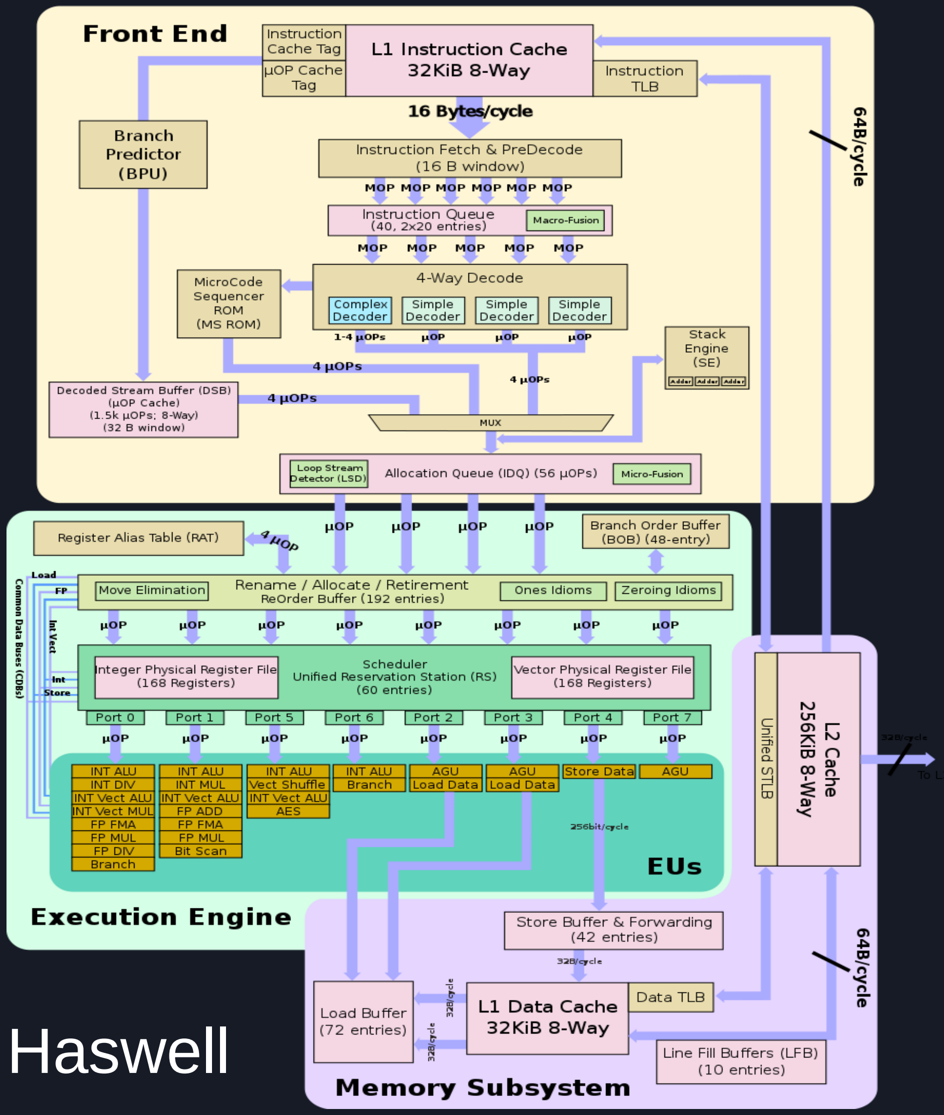
UNTIL YOU LEARN

HOW TO REPRODUCE THEM

Matter of trust : evolution



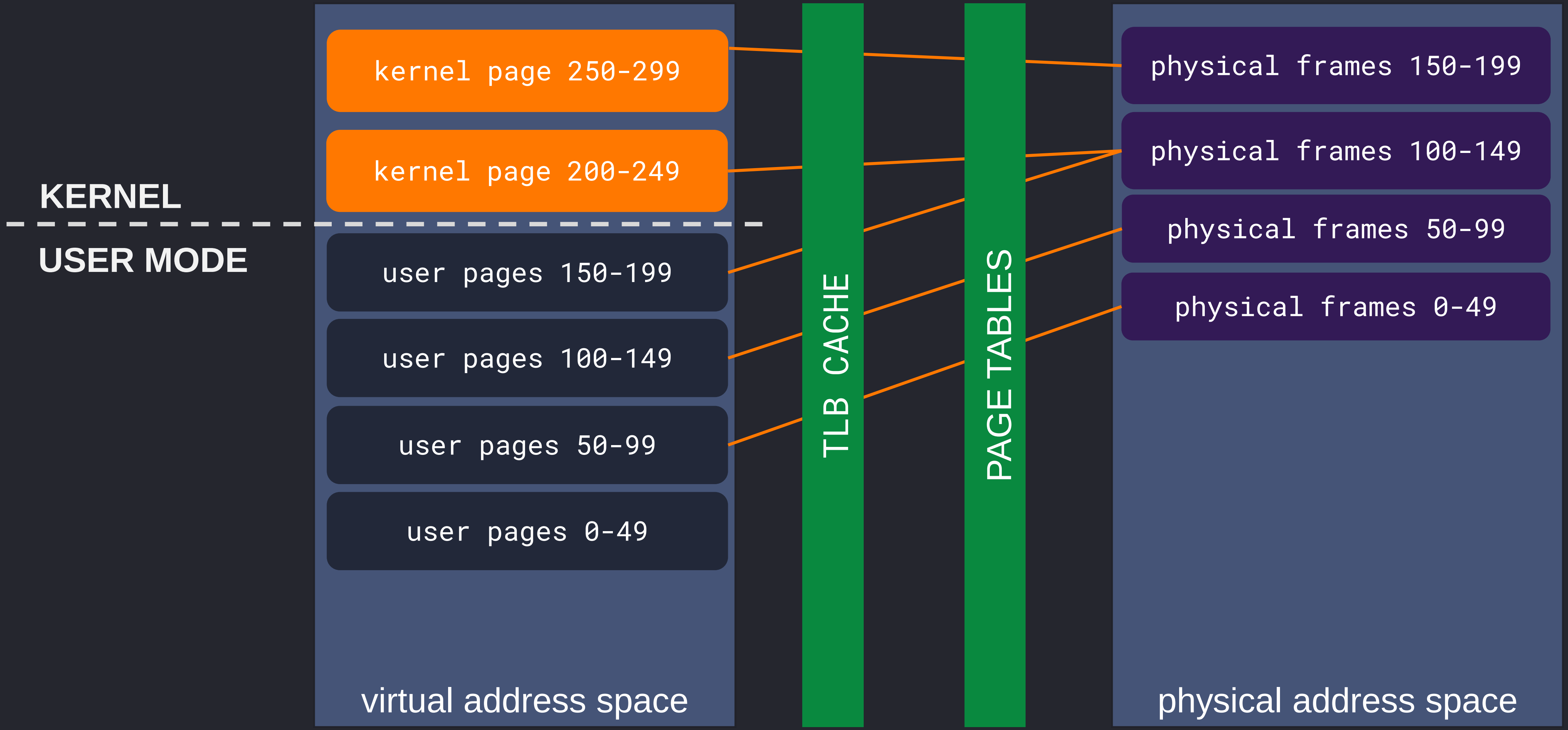
Intel 8086



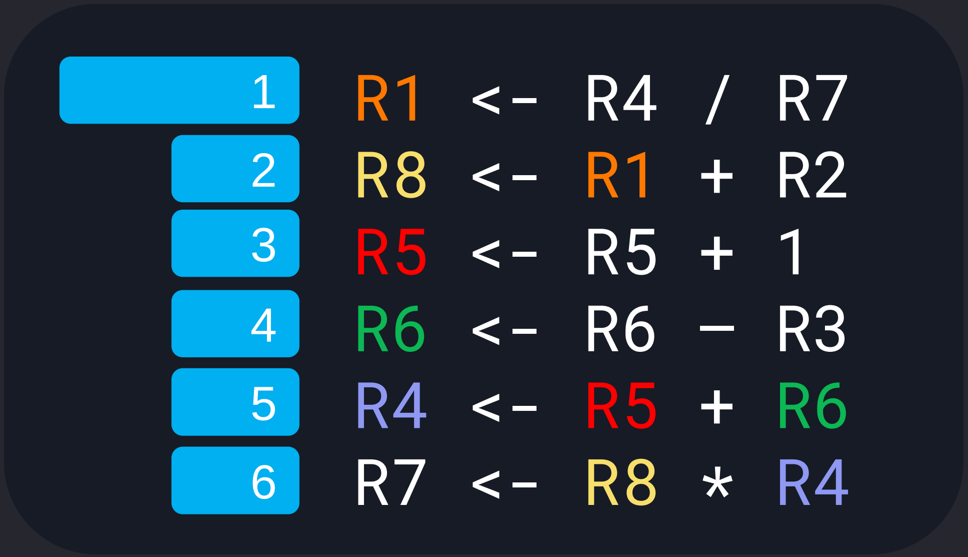
Haswell

BASICS

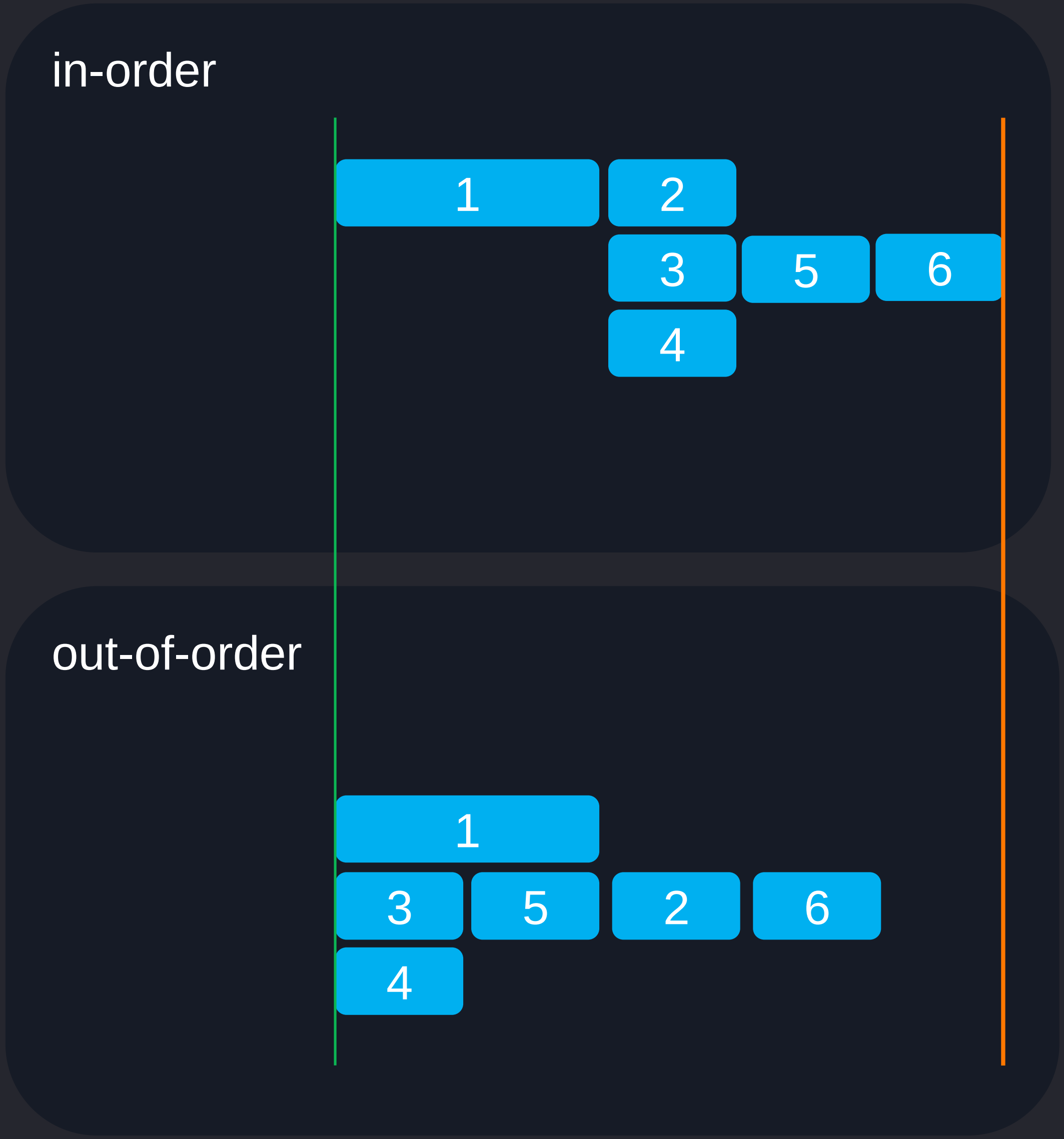
Basics : virtual memory

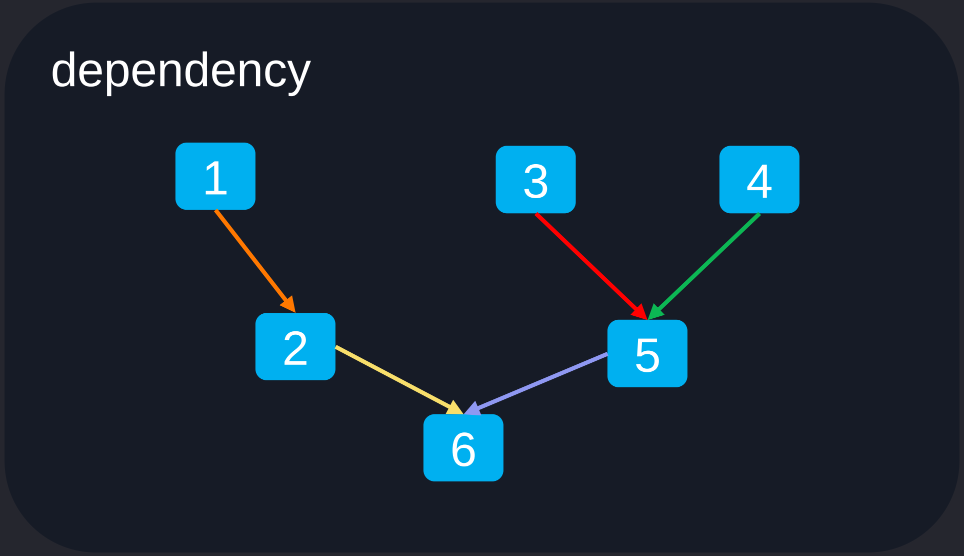


Basics: In-order / out-of-order execution

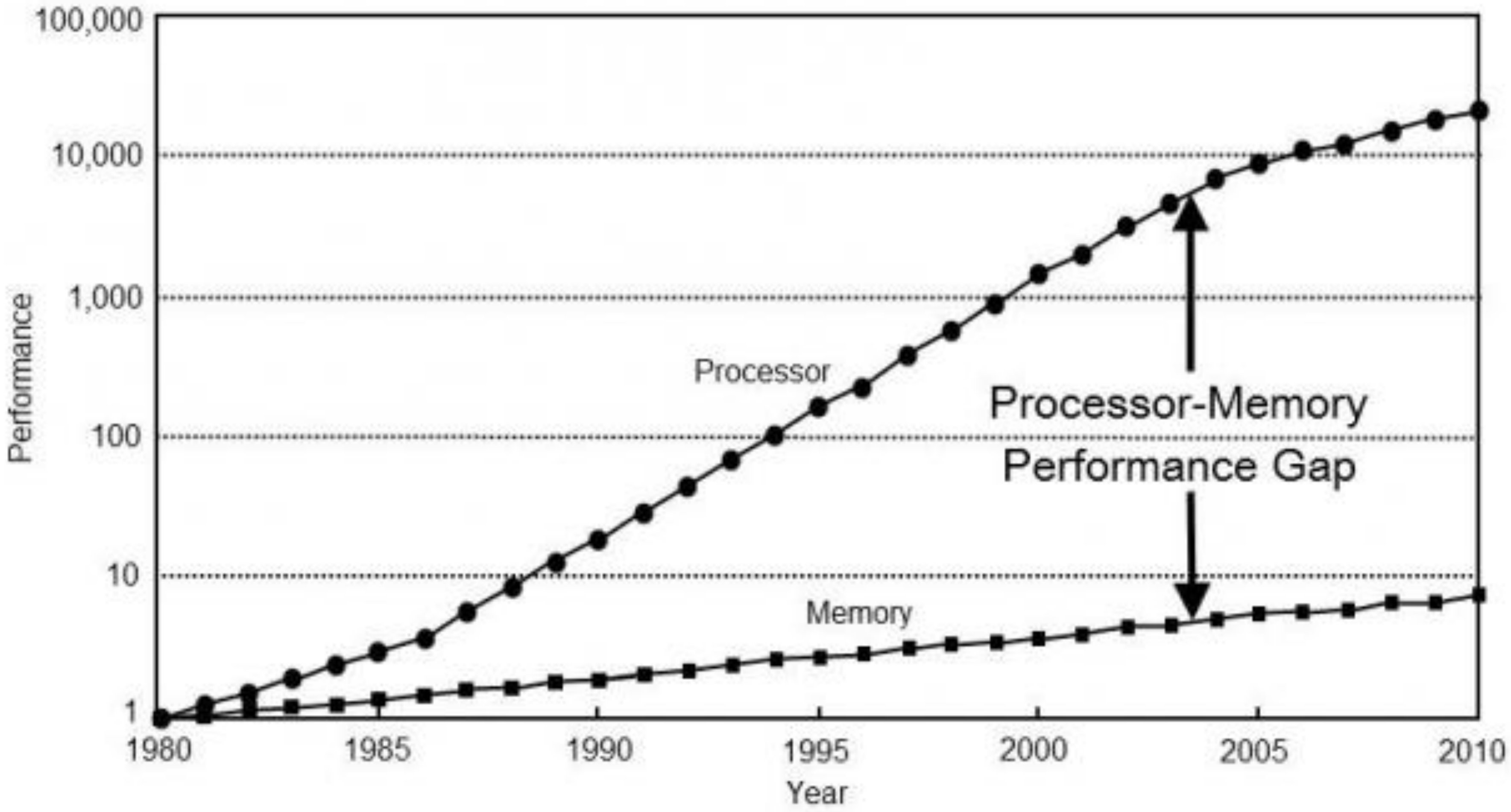


```
R1 <- R4 / R7
R8 <- R1 + R2
R5 <- R5 + 1
R6 <- R6 - R3
R4 <- R5 + R6
R7 <- R8 * R4
```





Basics : speculative and out-of order execution : reason



Discovered bugs

CVE-2017-5754

ROGUE DATA CACHE LOAD

CVE-2017-5753

BOUND S CHECK BYPASS

CVE-2017-5715

BRANCH TARGET INJECTION

CVE-2017-5754

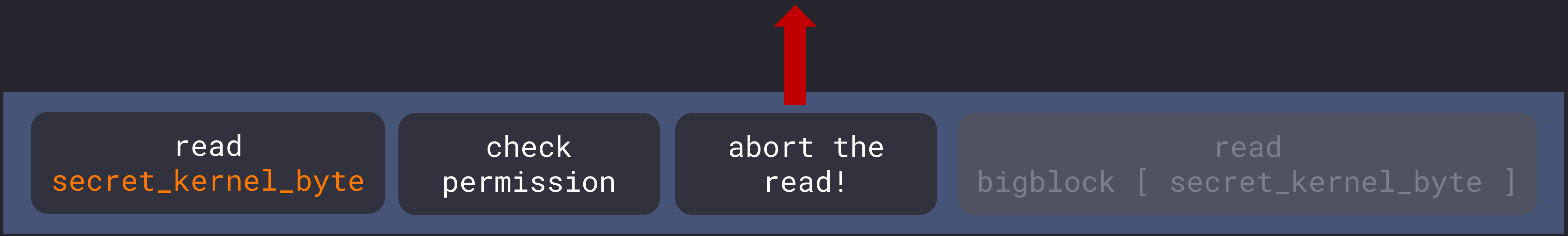
Meltdown : prerequisites

- Intel CPU, some IBM Power and ARM CPUs affected
- Big array of 256 x 4K items - bigblock
- Kernel address kernel_addr from which we want to read secret_kernel_byte

```
MOV RBX, BIGBLOCK
XOR EAX, EAX
MOV AL, BYTE PTR [ KERNEL_ADDR ]
SHL RAX, 12
MOV AL, [ RAX + RBX ]
```

Meltdown : in order execution, single pipeline

```
MOV RBX, BIGBLOCK
XOR EAX, EAX
MOV AL, BYTE PTR [ KERNEL_ADDR ]
SHL RAX, 12
MOV AL, [ RAX + RBX ]
```



read secret_kernel_byte

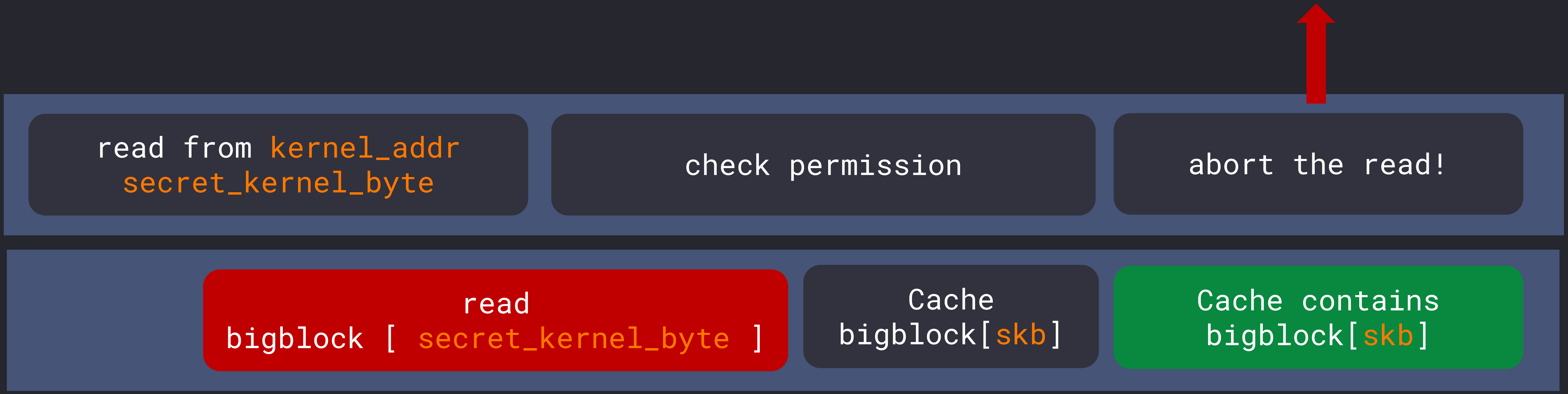
check permission

abort the read!

read bigblock [secret_kernel_byte]

Meltdown : out-of-order execution

```
MOV RBX, BIGBLOCK
XOR EAX, EAX
MOV AL, BYTE PTR [ KERNEL_ADDR ]
SHL RAX, 12
MOV AL, [ RAX + RBX ]
```



read from kernel_addr secret_kernel_byte

check permission

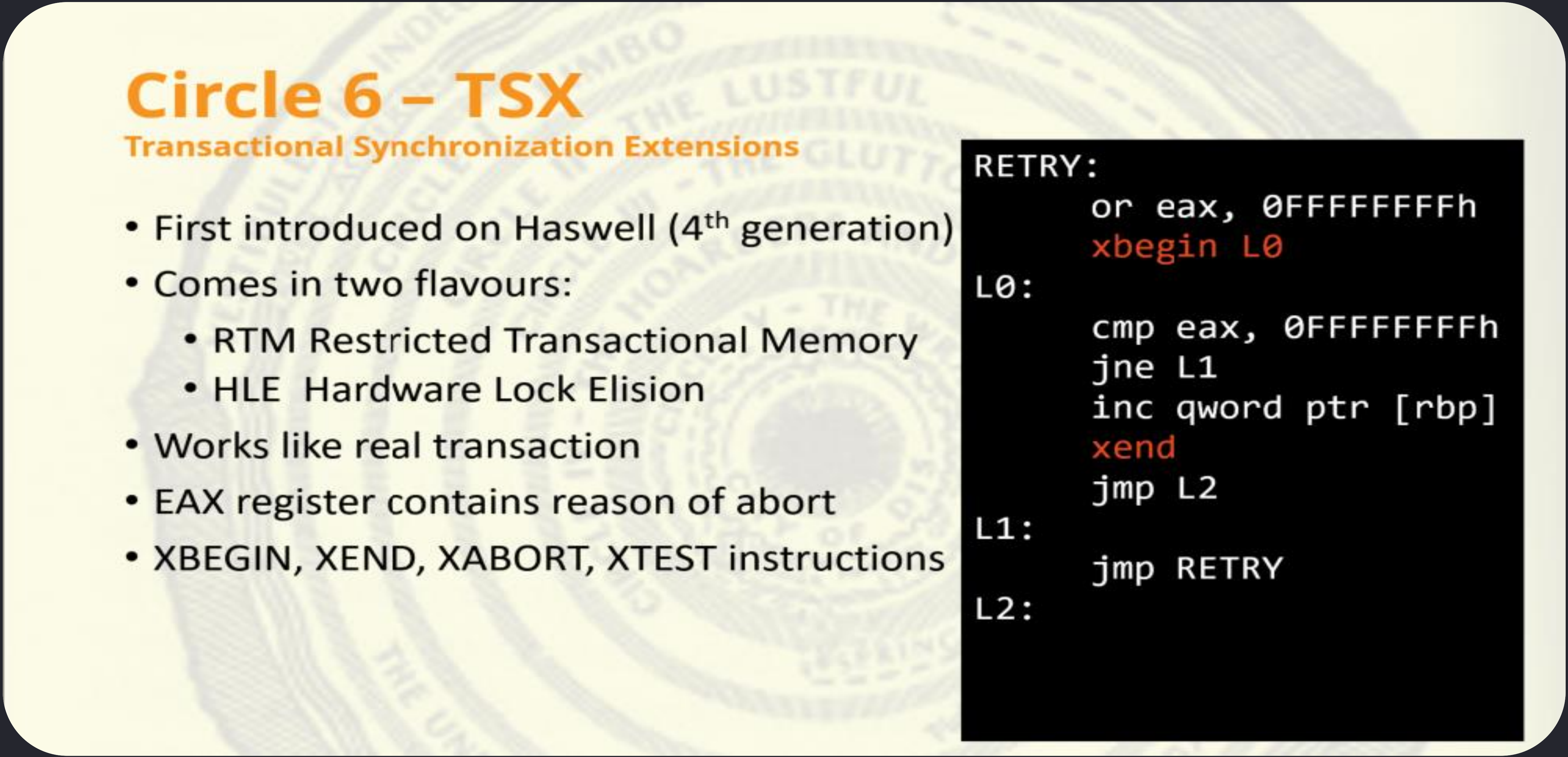
abort the read!

read bigblock [secret_kernel_byte]

Cache bigblock[skb]

Cache contains bigblock[skb]

FlashBack: TSX – Security Session 2016



Circle 6 – TSX

Transactional Synchronization Extensions

- First introduced on Haswell (4th generation)
- Comes in two flavours:
  - RTM Restricted Transactional Memory
  - HLE Hardware Lock Elision
- Works like real transaction
- EAX register contains reason of abort
- XBEGIN, XEND, XABORT, XTEST instructions

```
RETRY:
        or eax, 0FFFFFFFFh
        xbegin L0
L0:
        cmp eax, 0FFFFFFFFh
        jne L1
        inc qword ptr [rbp]
        xend
        jmp L2
L1:
        jmp RETRY
L2:
```

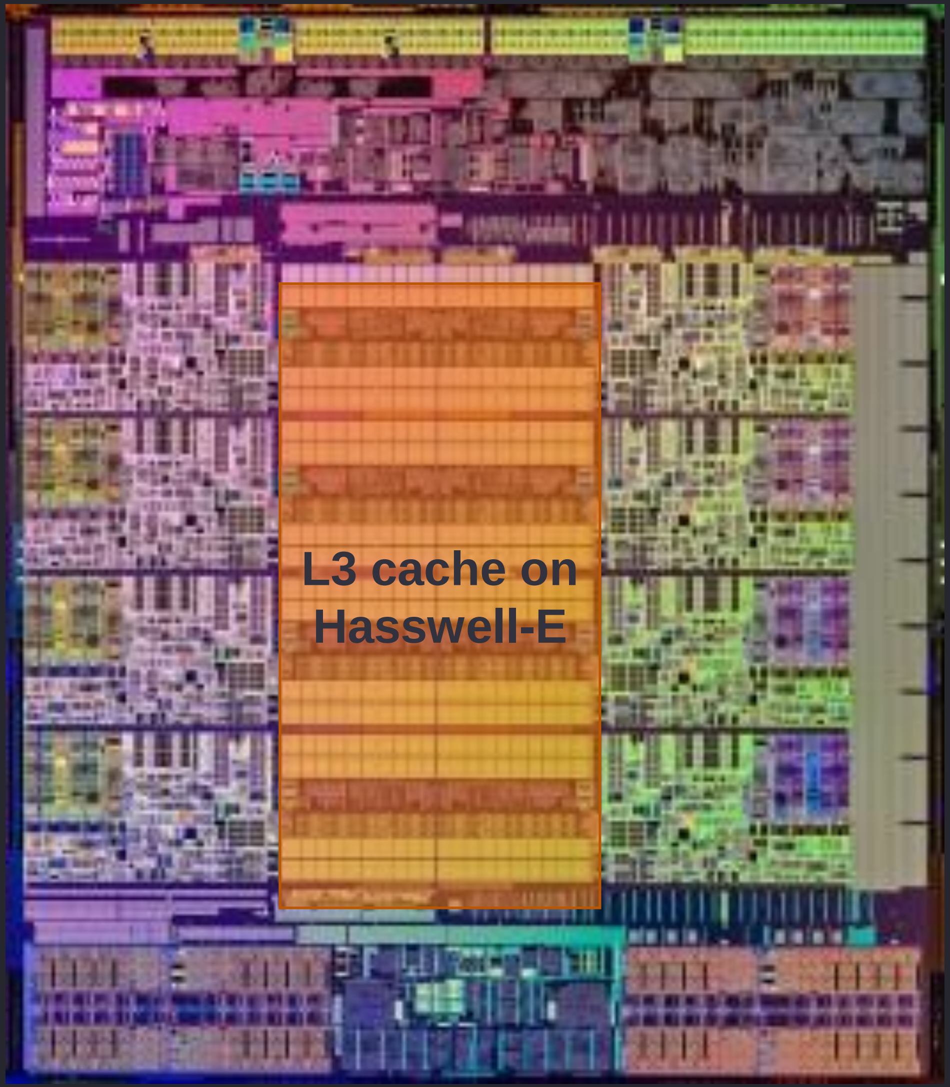
Meltdown : TSX to inhibit exception

```
      XBEGIN L1
      MOV    RBX, BIGBLOCK
      XOR    EAX, EAX
      MOV    AL, BYTE PTR [ KERNEL_ADDR ]
      SHL    RAX, 12
      MOV    AL, [ RAX + RBX ]
      XEND
L1:
```

Side channel attack

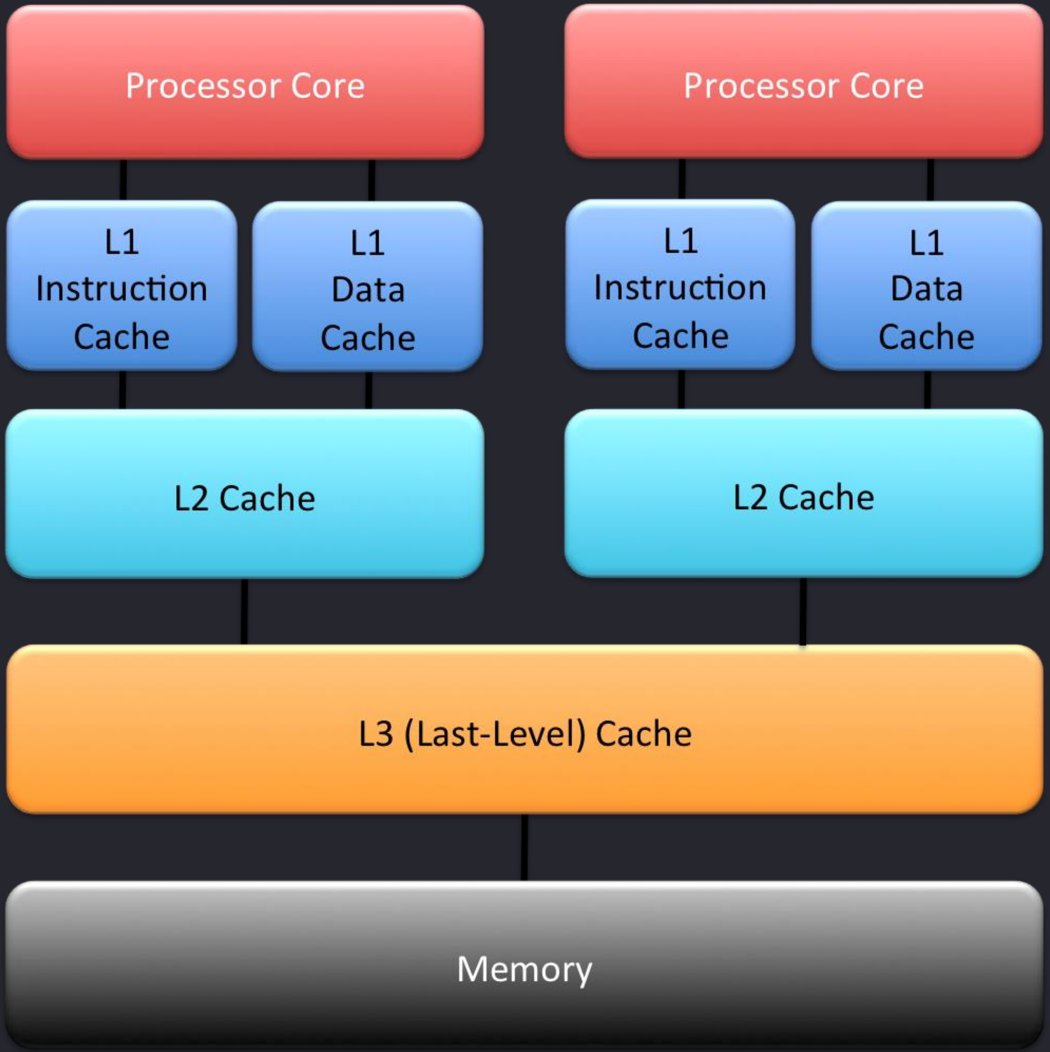
In computer security, a side-channel attack is any attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself .

Wikipedia

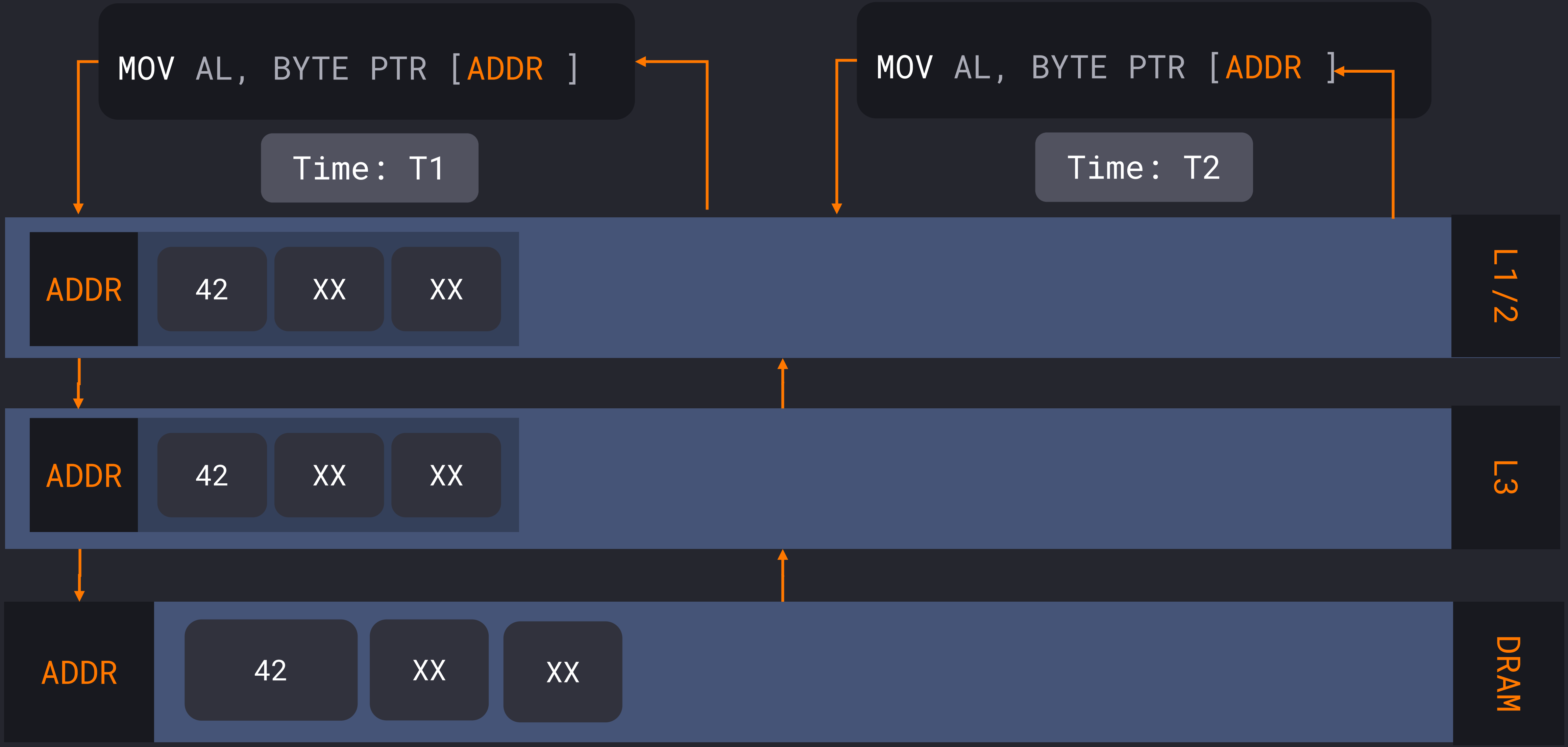


Side channel attack : story of caches

- L1/L2/L3 Caches
- To lower memory – CPU speed gap
- There is no instruction that can read specific cache line or tell you if there is data inside cache
- However you can flush specific VA address (or cache line) from cache by CLFLUSH instruction
- How could we learn what is inside cache?



Intermezzo : everything is about right timing



```
MOV AL, BYTE PTR [ADDR ]
```

Time: T1

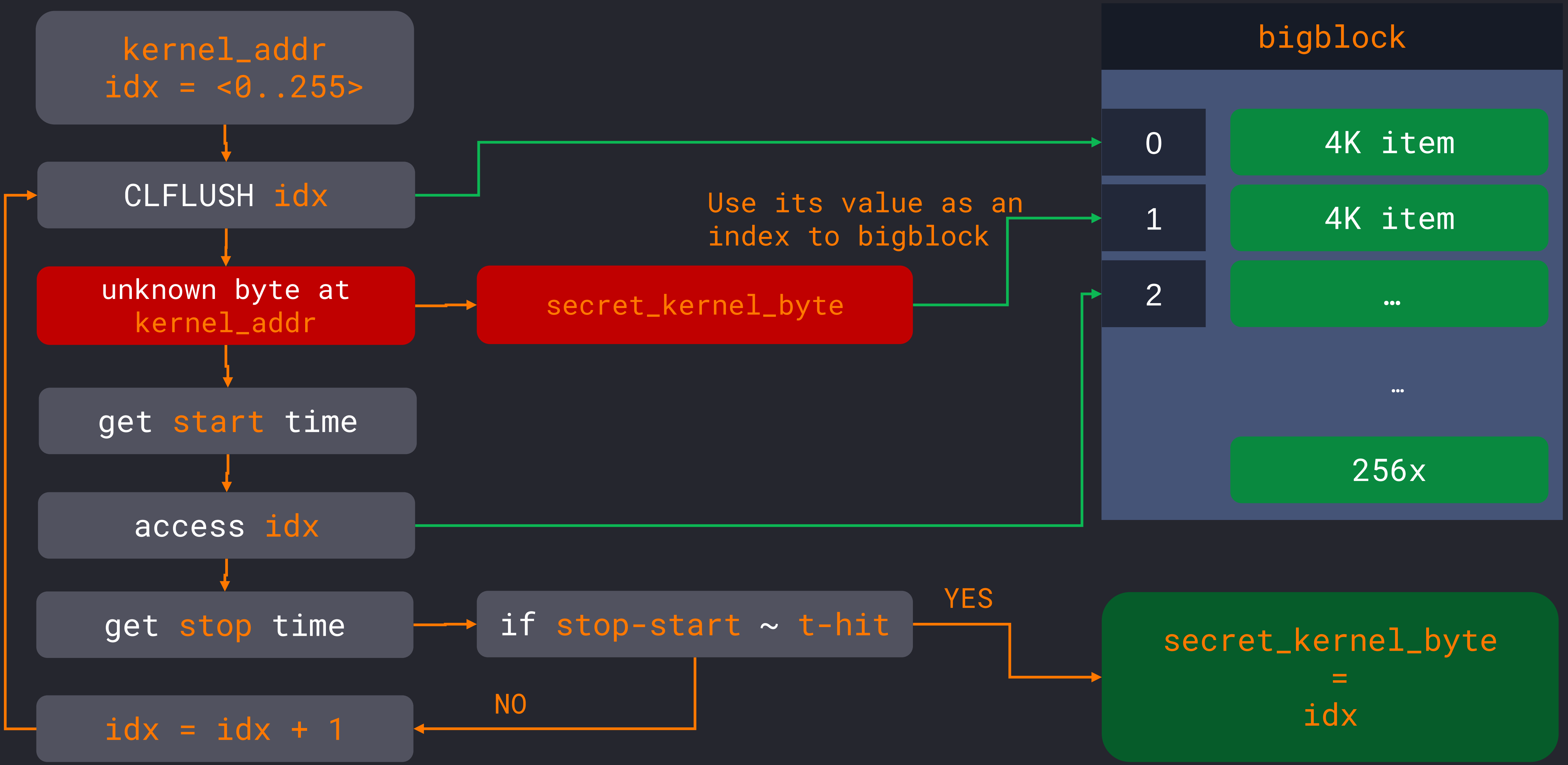
```
MOV AL, BYTE PTR [ADDR ]
```

Time: T2

Intermezzo : everything is about right timing

`Time t-miss: T1` > `Time t-hit: T2`

Side channel attack : final plan



CVE-2017-5715
CVE-2017-5753

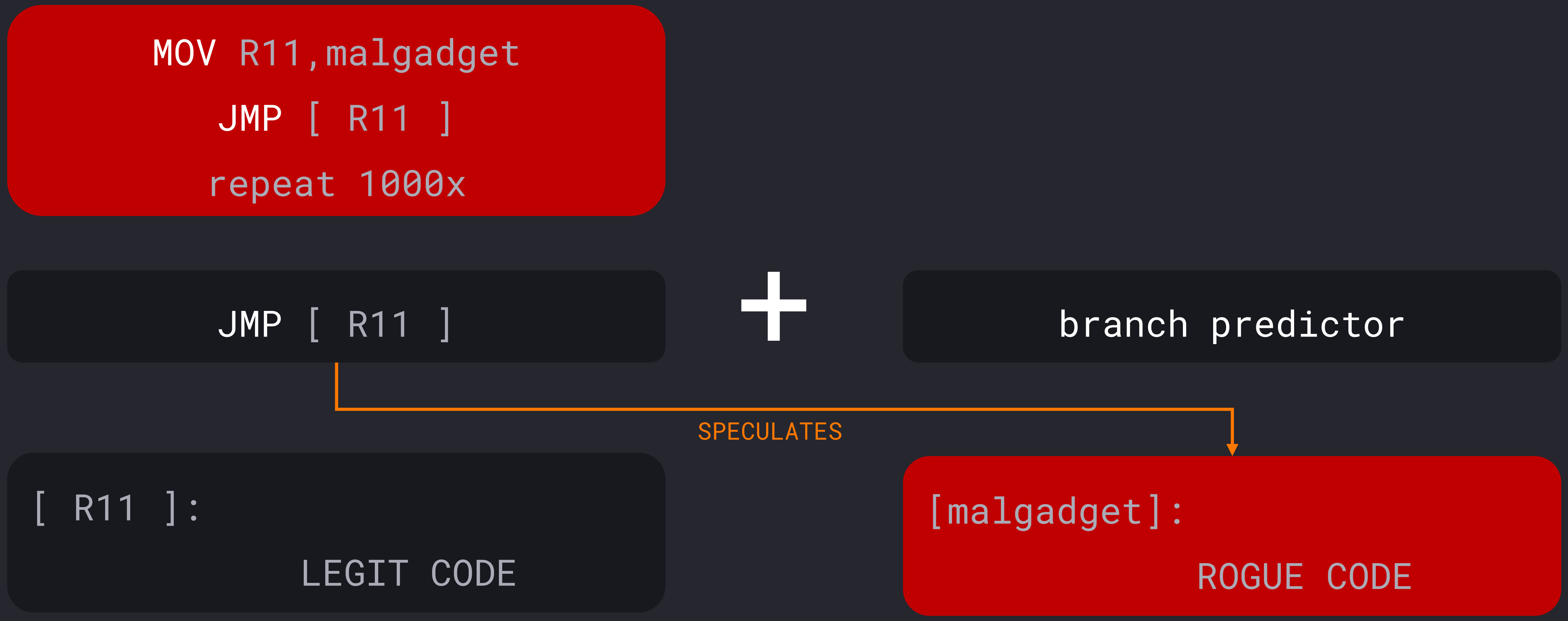
Spectre – bounds check bypass

- Leverages speculative execution to access memory out of bounds of an array
- If `array1_size` is not in cache, CPU has time enough to speculate and speculatively executes inner part of the condition no matter what is the value of `x`
- That loads item from `array2` to cache based on value of `array1[x]` where `x` can be anything

```
if (x < array1_size) {
    junk &= array2[array1[x] * 512];
}
```

Spectre – indirect branch target injection

- Manipulating branch predictor to speculative execute gadget of interest
- Training branch predictor to jump to your desired target
- The CPU before the legit code is executed speculatively runs your `malgadget`



```
MOV R11,malgadget
JMP [ R11 ]
repeat 1000x
```

```
JMP [ R11 ]
```

+

```
branch predictor
```

SPECULATES

```
[ R11 ]:
          LEGIT CODE
```

```
[malgadget]:
          ROGUE CODE
```

WORKAROUNDS

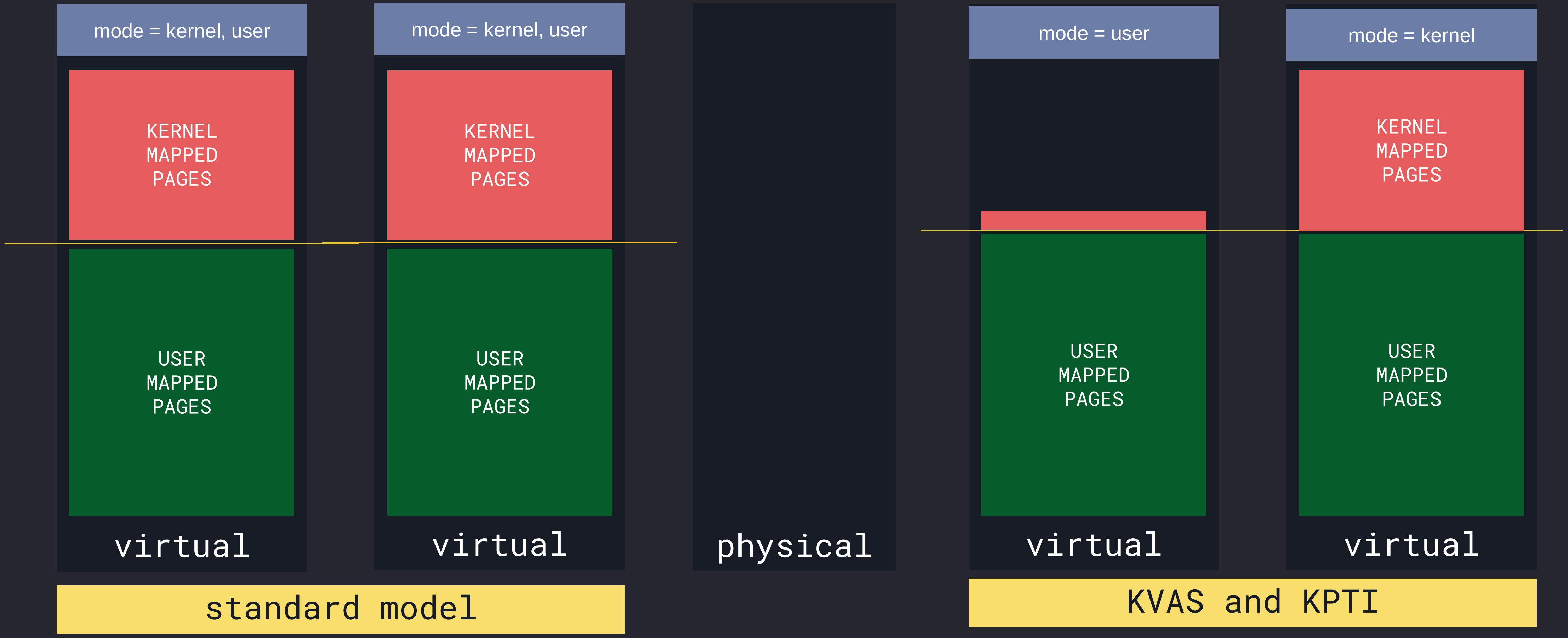
FIXES

Workarounds

- Not easy to fix, the only proper fix is on silicon
- In case of Spectre, it is an architectural problem
- Meltdown – easy to fix , workarounds for most OSes available
- Intel deploying microcode updates, but not all CPUs and not all variants of Spectre can be easily treated
- Running 286 on 16MHz? You are safe :D

Workarounds : meltdown : KVAS and KPTI

- Same concept , complete isolation of kernel and user space mappings
- Induces some performance cost as page tables have to be switched and TLB flushed



Workarounds : meltdown : KVAS and KPTI

- Best solution against Meltdown, brings performance overhead up to 30%
- Bigger performance hit with higher rate of syscalls and need of switching to privileged mode.
- On modern CPUs with support of ASID or PCID (Address Space IDs or Processor-Context ID). Performance can be nearly “native” because TLB can be flushed only per address space

Workarounds : Spectre : retpoline

- A Branch Target Injection mitigation invented by Google using RSB
- Software based -> needs recompilation
- Upcoming support in gcc `-mindirect-branch=thunk-extern`

```
JMP [ R11 ]
```

```
        CALL LOAD
CAPTURE:
        PAUSE
        JMP CAPTURE
LOAD:
        MOV [ RSP ], R11
        RET
```

Workarounds : microcode updates

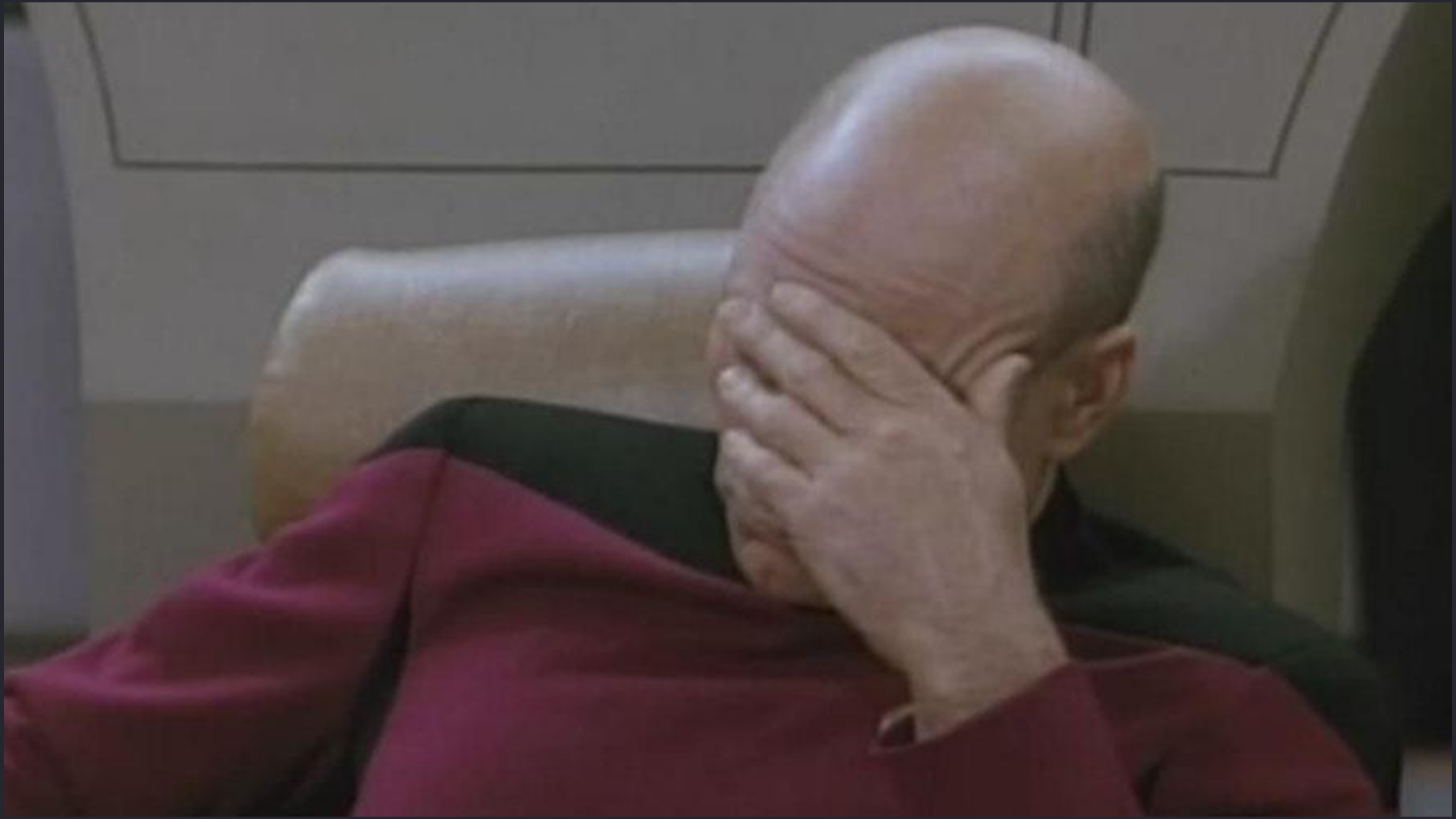
- CPUID AX=0x7, return RDX.26 to indicate presence of this feature
  - IA32_SPEC_CTRL (0x48) and IA32_PRED_CMD (0x49)
  - IA32_SPEC_CTRL, bit0 – Indirect Branch Restricted Speculation (IBRS)
  - IA32_PRED_CMD, bit0 – Indirect Branch Prediction Barrier (IBPB)

- **IBRS - Indirect Branch Restricted Speculation**
  If IBRS is set, near returns and near indirect jumps/calls will not allow their predicted target address to be controlled by code that executed in a less privileged prediction mode before the IBRS mode was last written with a value of 1
- **STIBP** **-** Single Thread Indirect Branch Predictors (RDX.27).
  Stops sharing predictor cache between physical threads.
- **IBPB** - Indirect Branch Predictor Barrier
  Setting of IBPB ensures that earlier code's behavior does not control later indirect branch predictions. It is used when context switching to new untrusted address space.

They do literally insane things. They do things that do not make sense ... The patches do things that are not sane.
WHAT THE F*CK IS GOING ON?

Linus Torvalds

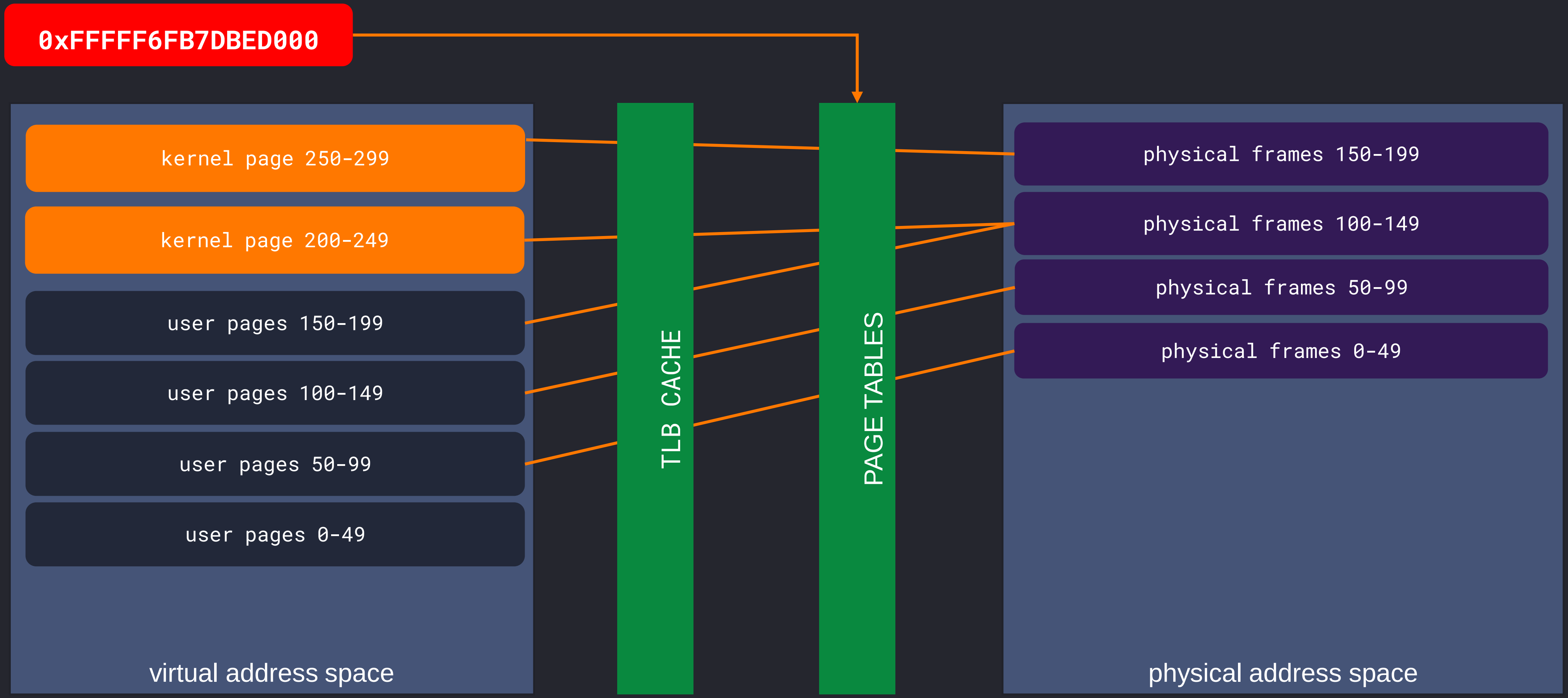
Fix that has fixed a bug and now has to be fixed



Page tables accessible from user mode

- Affected version Windows 7 x64, Windows 2008R2
- CVE-2018-1038
- Delivered via updates 2018-01, 2018-02
- Self referenced base entry for PML4 mapped and accessible in user mode
- Allows to read whole physical memory without any special privileges

Page tables accessible from user mode



Conclusion

- This will haunt us for a long time
- Every system is as safe as it’s weakest part
- Hardware is not an exception
- CPUs neither
- Race for performance may induce unexpected behaviours
- There are more versions of these bugs: SGXSpectre

Thank you for your attention!

Ask and I'll try to answer….

Martin Hron

@thinkcz

hron@avast.com , martin@hron.eu