

Quiz

Was gilt für die folgende Instruktionssequenz?

```
add rbx, rdx  
adc rax, rcx
```

☐

Es werden die beiden 128-bit Zahlen `rax:rbx` und `rcx:rdx` addiert

☐

`adc` addiert zusätzlich das Carry-Flag (CF) zum Ergebnis

☐

Das Ergebnis von `adc rax, rcx` wird von `add rbx, rdx` beeinflusst

☐

Das Ergebnis von `adc rax, rcx` wird nicht von `add rbx, rdx` beeinflusst

Quiz

Was macht die Instruktion `xchg rax, rax` ?

☐

Nichts; `nop` ist ein alias dafür

☐

Sie setzt nur Flags

☐

Sie setzt die oberen 32 Bit von `rax` auf 0

☐

Die Instruktion ist nicht valide, da mindestens ein Operand eine Speicherreferenz sein muss

Quiz

Was gilt für die Instruktion `cmovl eax, ecx` ?

☐

Es wird `eax` auf `ecx` gesetzt aber nur wenn `eax` kleiner als `ecx` ist

☐

Es wird `eax` auf `ecx` gesetzt aber nur wenn `ecx` kleiner als `eax` ist

☐

Es wird `eax` auf `ecx` gesetzt wenn die `SF` \neq `OF` anzeigen

☐

Es werden die oberen 32 Bits von `rax` auf 0 gesetzt

Quiz

Was macht die Instruktion `popcnt rax, rcx` ?

☐

Sie setzt `rcx` auf 0

☐

Sie zählt die gesetzten Bits in `rcx` und schreibt das Ergebnis in `rax`

☐

Sie nimmt `rcx` viele 64-Bit Elemente von Stack und schreibt das letzte in `rax`

☐

Sie setzt das Zero-Flag (ZF) gdw. das Ergebnis der Operation 0 ist

Quiz

Was kann passieren, wenn `popcnt rax, rcx` ausgeführt wird, obwohl die Instruktion auf der genutzten CPU nicht verfügbar ist?

☐

`rax` wird auf 0 gesetzt

☐

Die Exception "Invalid Opcode" (`#UD`) wird ausgelöst

☐

Das `cpuid`-Flag wird gesetzt

☐

Die Instruktion wird als `nop` ausgeführt

Quiz

Wie kann man rax (64-bit) auf rdx:rax (128-bit) sign-extenden?

☐

`movsx rdx, rax`

☐

`cqo rdx, rax`

☐

`cqo`

☐

`mov rdx, rax`
`sar rdx, 63`