# SPECTRE: SPECULATIVE EXECUTION

A

**D**
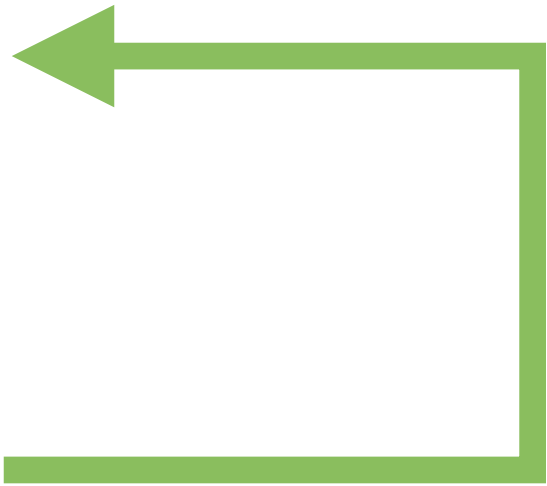
The CPU has learned that Counter *probably* is > 0

# Reading Counter from memory is very slow

The CPU *speculatively* executes        to improve performance

**3**

Counter
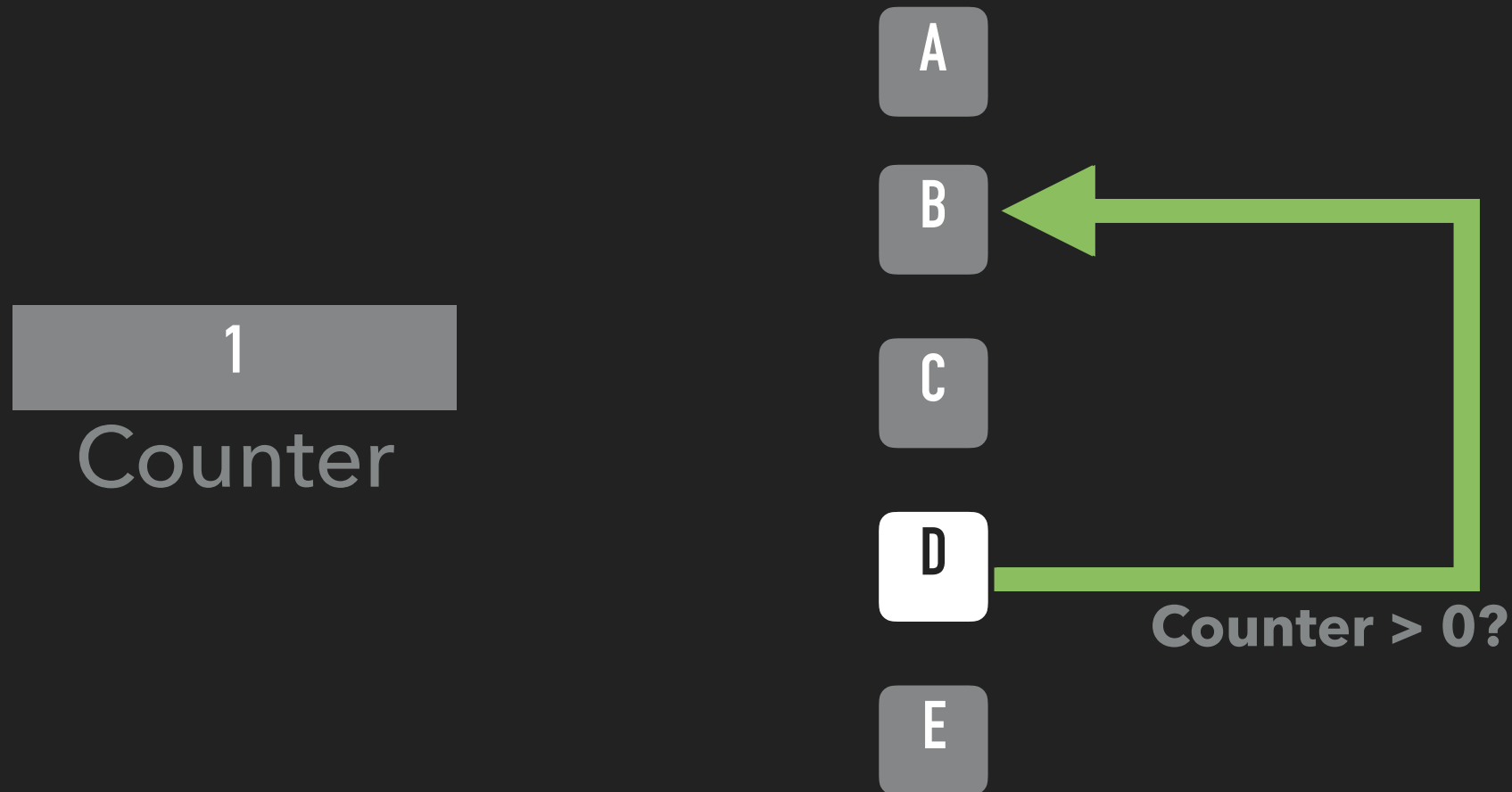
**Counter > 0?**

B C

# SPECTRE: SPECULATIVE EXECUTION

**A**
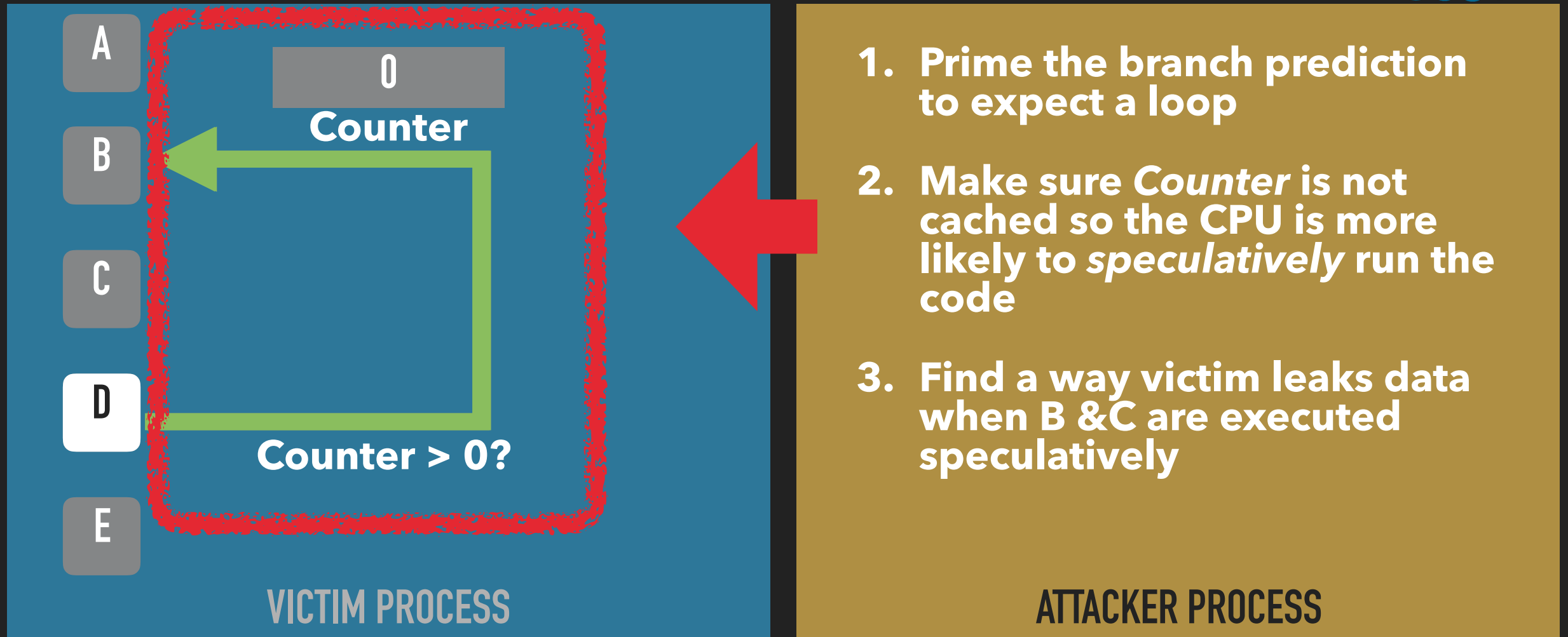
**B**

**1**

Counter

**C**

**D**

Counter > 0?

**E**

The CPU has learned that Counter *probably* is > 0

Reading Counter from memory is very slow

The CPU *speculatively* executes **B** **C** to improve performance

# SPECTRE: SPECULATIVE EXECUTION

**A**

**0**

**Counter**

**B**

**C**

**D**

**Counter > 0?**

**E**

VICTIM PROCESS

1. **Prime the branch prediction to expect a loop**

2. **Make sure *Counter* is not cached so the CPU is more likely to *speculatively* run the code**

3. **Find a way victim leaks data when B &C are executed speculatively**

ATTACKER PROCESS

Attacker can influence the CPUs branch prediction of victim.

Making the victim *speculatively* execute "wrong" code.

E.g. loop even when Counter is == 0.