# 250P: Computer Systems Architecture

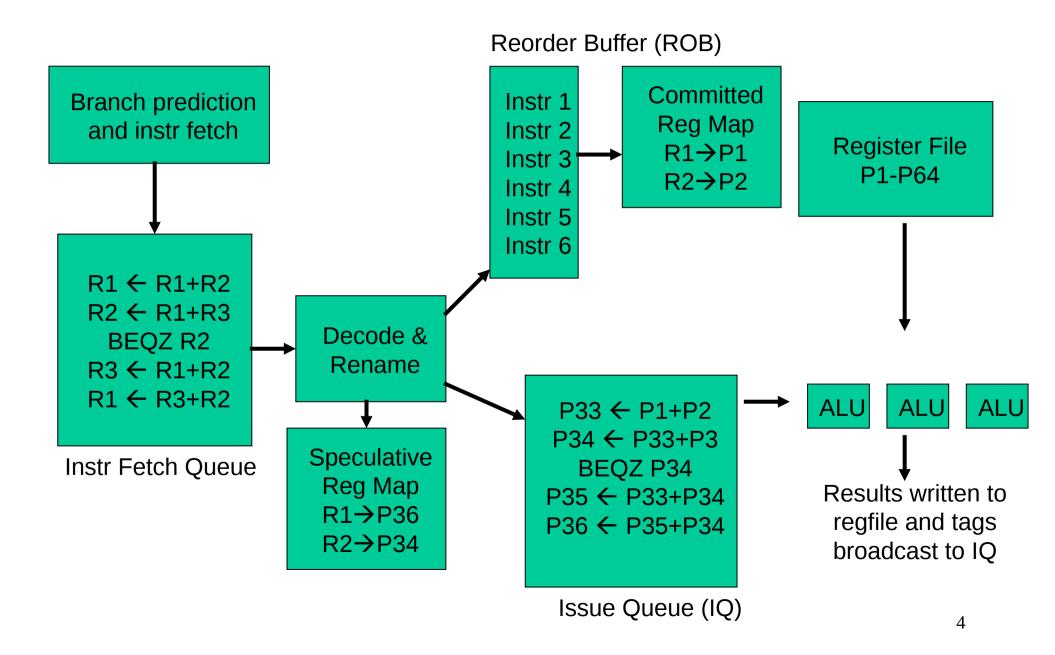
# Lecture 15: Side channel attacks Meltdown and Spectre

Anton Burtsev March, 2019

#### Meltdown

# Page tables and protection

### The Alpha 21264 Out-of-Order Implementation



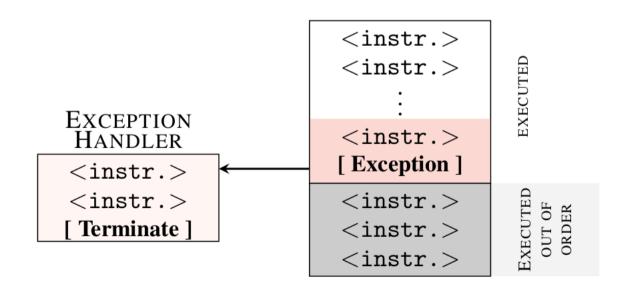
#### ITLB L1 Instruction Cache Instruction Fetch & PreDecode Branch Frontend Predictor Instruction Queue 4-Way Decode μOP Cache μOPs μΟΡ μΟΡ μΟΡ MUX Allocation Queue μΟΡ μΟΡ μΟΡ $\mu$ OP CDB Reorder buffer μΟΡ μΟΡ μΟΡ μΟΡ μΟΡ μΟΡ μΟΡ **Execution Engine** Scheduler μΟΡ $\mu$ OP Load data Store data ALU, FMA, ... ALU, Branch ALU, Vect, **Execution Units** Subsystem Load Buffer Store Buffer Memory STLB DTLB← L1 Data Cache L2 Cache

# Skylake (simplified)

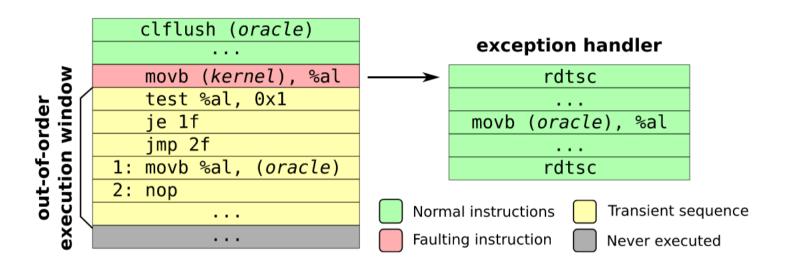
# Exceptions and speculation

```
raise_exception();
// the line below is never reached
captage access(probe_array[data * 4096]);
```

Listing 1: A toy example to illustrate side-effects of outof-order execution.



# Exceptions and speculation



#### Cache access time

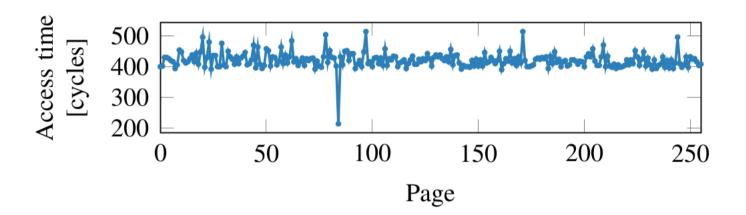


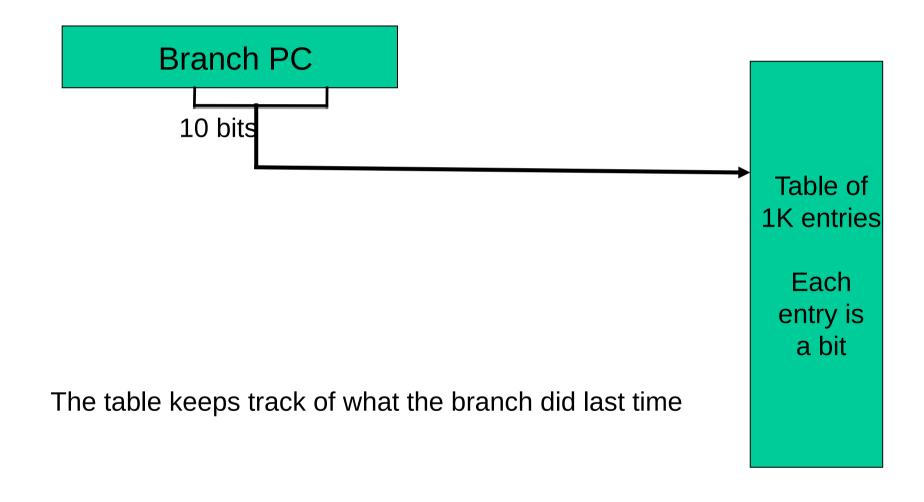
Figure 4: Even if a memory location is only accessed during out-of-order execution, it remains cached. Iterating over the 256 pages of probe\_array shows one cache hit, exactly on the page that was accessed during the out-of-order execution.

### Spectre

#### 1-Bit Bimodal Prediction

- For each branch, keep track of what happened last time and use that outcome as the prediction
- What are prediction accuracies for branches 1 and 2 below:

#### Bimodal 1-Bit Predictor



# Gadget

```
if (x < array1_size)
  y = array2[array1[x] * 4096];</pre>
```

Thank you!

# Exceptions and speculation

