4.22 (Understand computer)

Chip → core → thread

CPU 1 → virtual Memory protection
CPU 1 →

# Multithreading

- Different stacks ($sp) (local variables)
- PCs
- registers
- Multiple threads
- same memory (globals, heap)
- same time

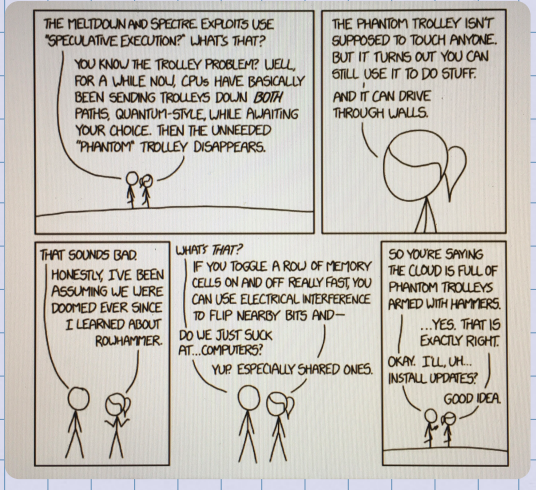
blocking semantic (keyboard interrupt functions) | Multiple ALUs in case someone stalls

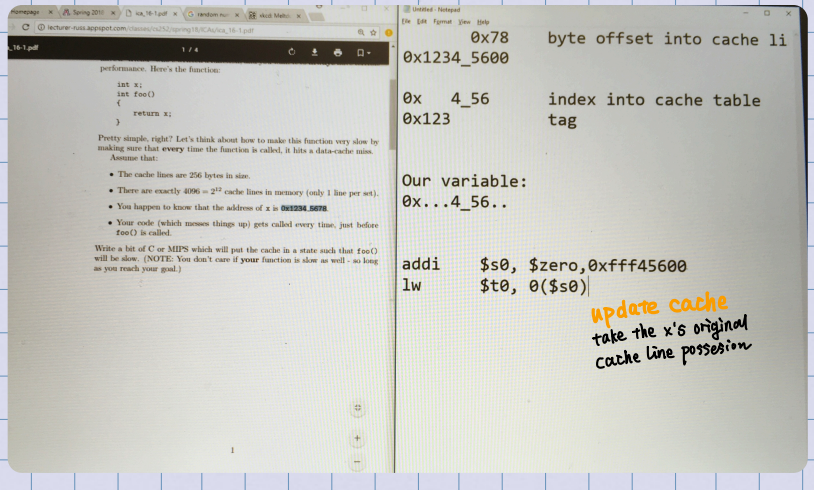
progress (main is thread)
-thread
-thread
-thread
→ race condition
undefined sequence

LOCK - someone is writting back
don't interfere.

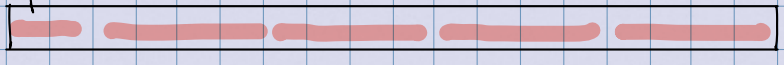
not operated instruction
non-unfill caches

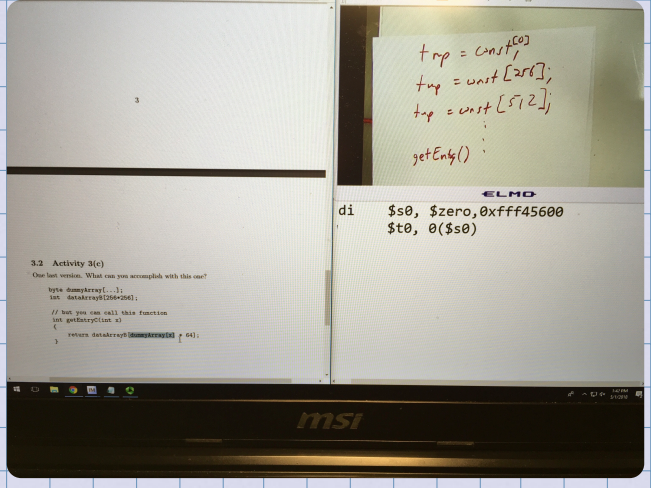
# Spectre & Meltdown




different cache lines




3
3.2 Activity 3(c)
One last version. What can you accomplish with this one?
byte dummyArray[...];
int dataArray8[256*256];
// but you can call this function
int getEntryC(int x)
{
return dataArray8[dummyArray[x] * 64];
}
trap = const[0];
trap = const[256];
trap = const[512];
getEntry()
ELMO
di $s0, $zero,0xfff45600
$t0, 0($s0)
msi