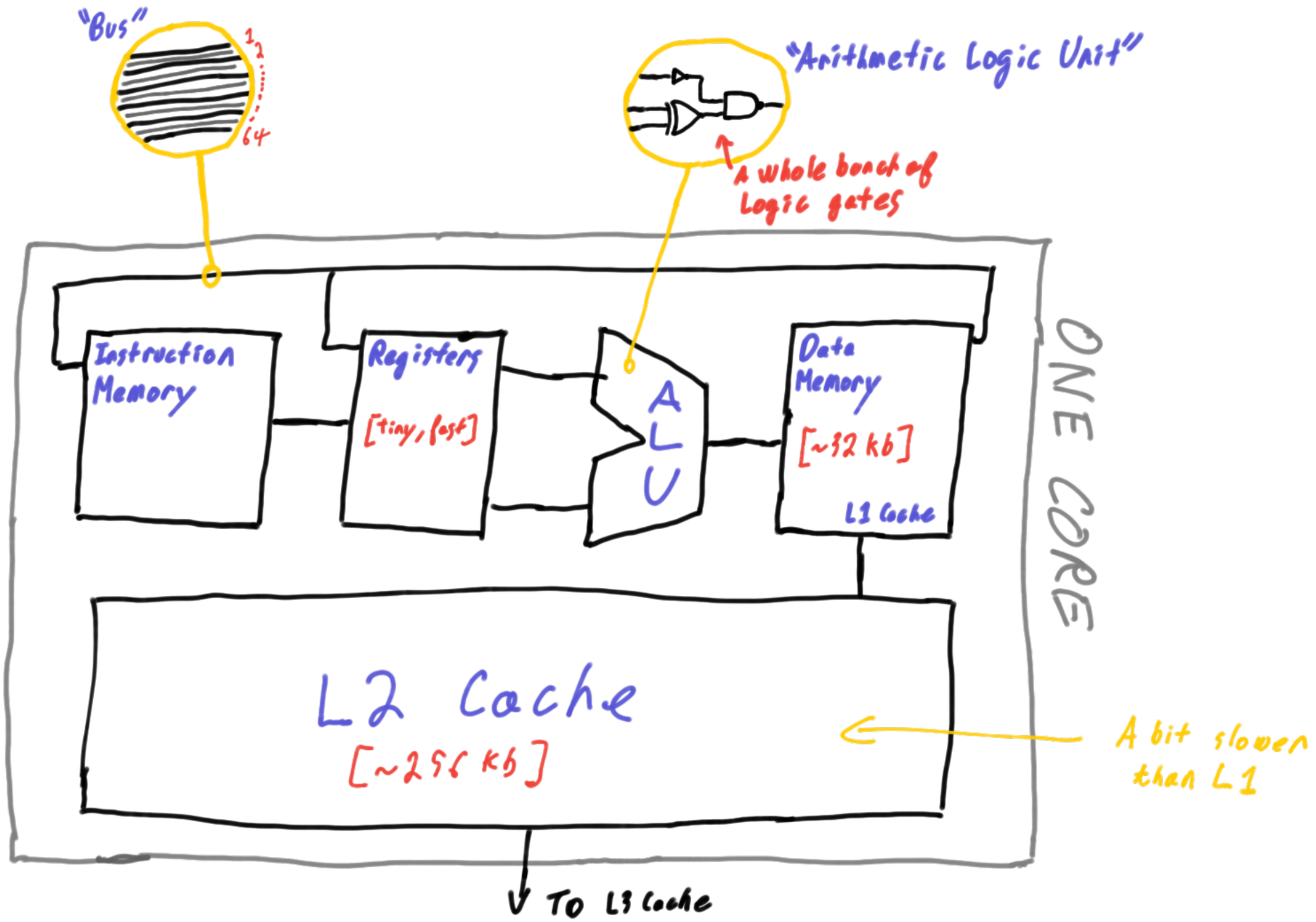
Technical Computing for the Earth Sciences, Lecture 10:

Introduction to Parallel Programming

EARS 80.03

CPU basics


"Bus"
1
2
64
"Arithmetic Logic Unit"
A whole bunch of Logic gates
Instruction Memory
Registers
[tiny, fast]
A
L
U
Data Memory
[~32 kb]
L1 Cache
CORE
L2 Cache
[~256 kb]
A bit slower than L1
TO L3 Cache

CPU basics

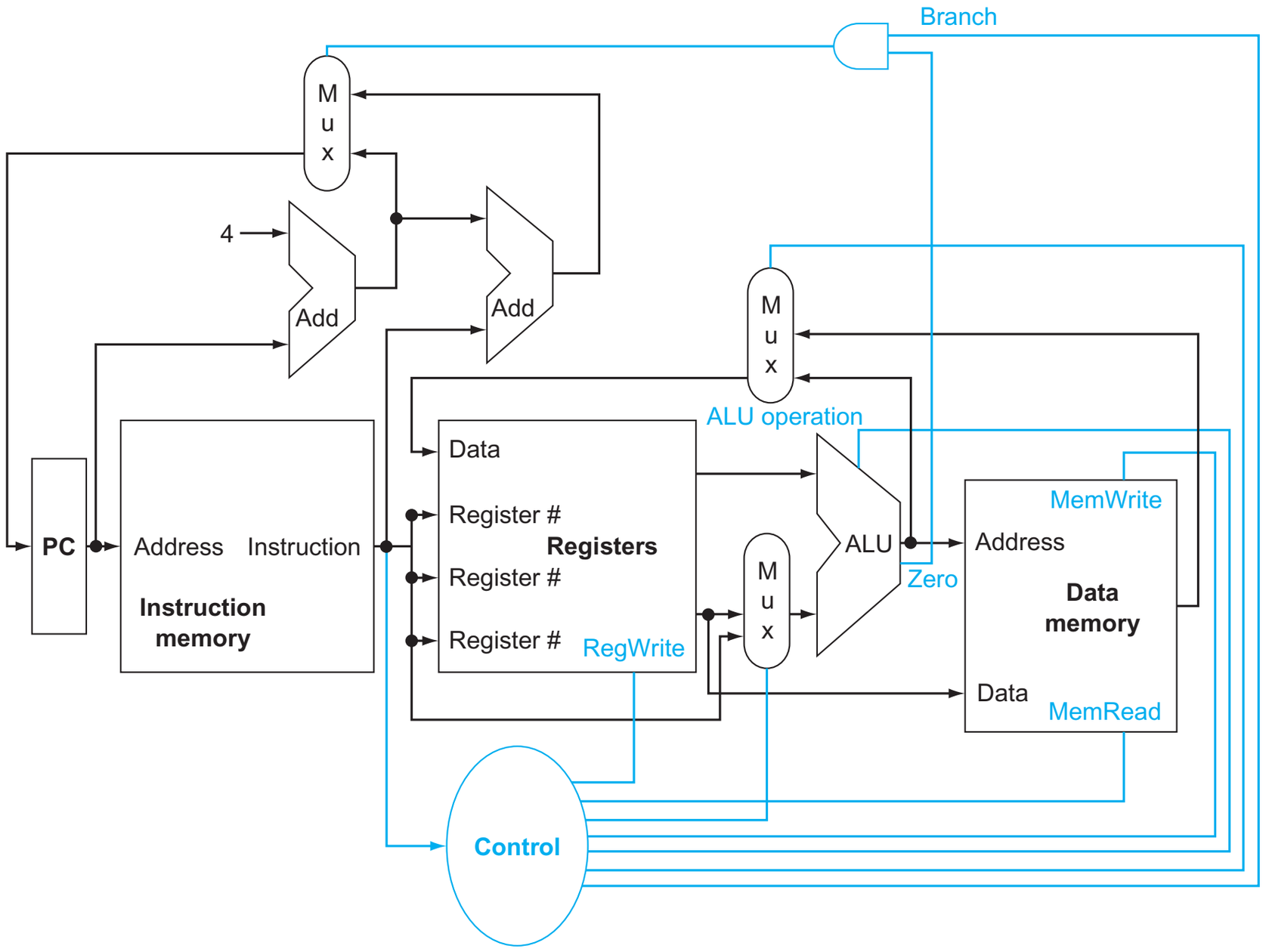
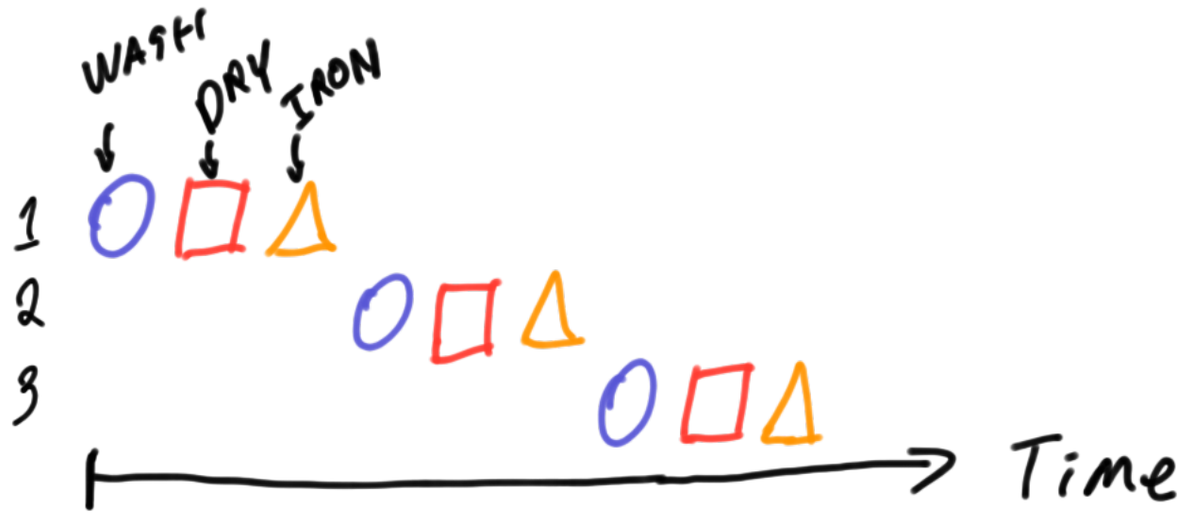


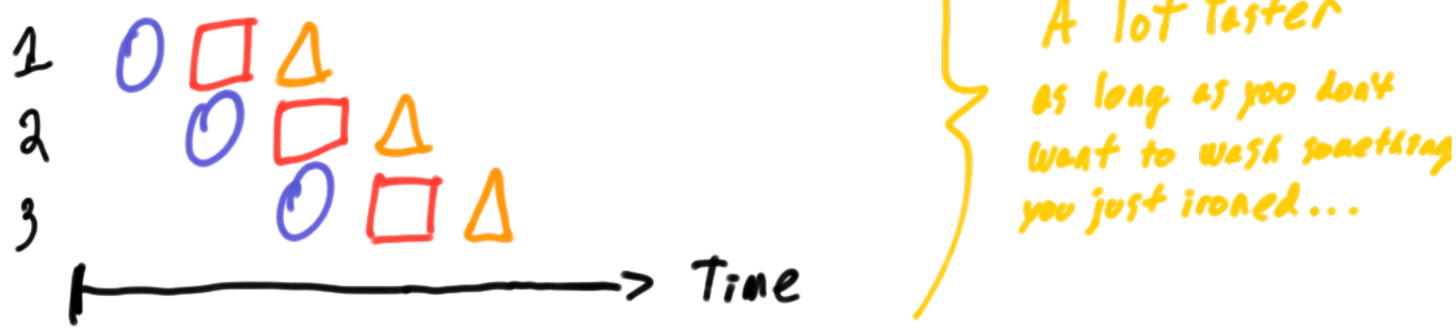
FIGURE 4.2 The basic implementation of the MIPS subset, including the necessary multiplexors and control
The top multiplexor ("Mux") controls what value replaces the PC (PC + 4 or the branch destination address); the multiplexor is con
by the gate that "ANDs" together the Zero output of the ALU and a control signal that indicates that the instruction is a branch. The r
multiplexor, whose output returns to the register file, is used to steer the output of the ALU (in the case of an arithmetic-logical instructi
the output of the data memory (in the case of a load) for writing into the register file. Finally, the bottommost multiplexor is used to dete
whether the second ALU input is from the registers (for an arithmetic-logical instruction or a branch) or from the offset field of the instr
(for a load or store). The added control lines are straightforward and determine the operation performed at the ALU, whether the data m
should read or write, and whether the registers should perform a write operation. The control lines are shown in color to make them ea
see.

Patterson & Hennessy, *Computer Organization and Design*, 5th ed.

Pipelining



VS



Pipelining

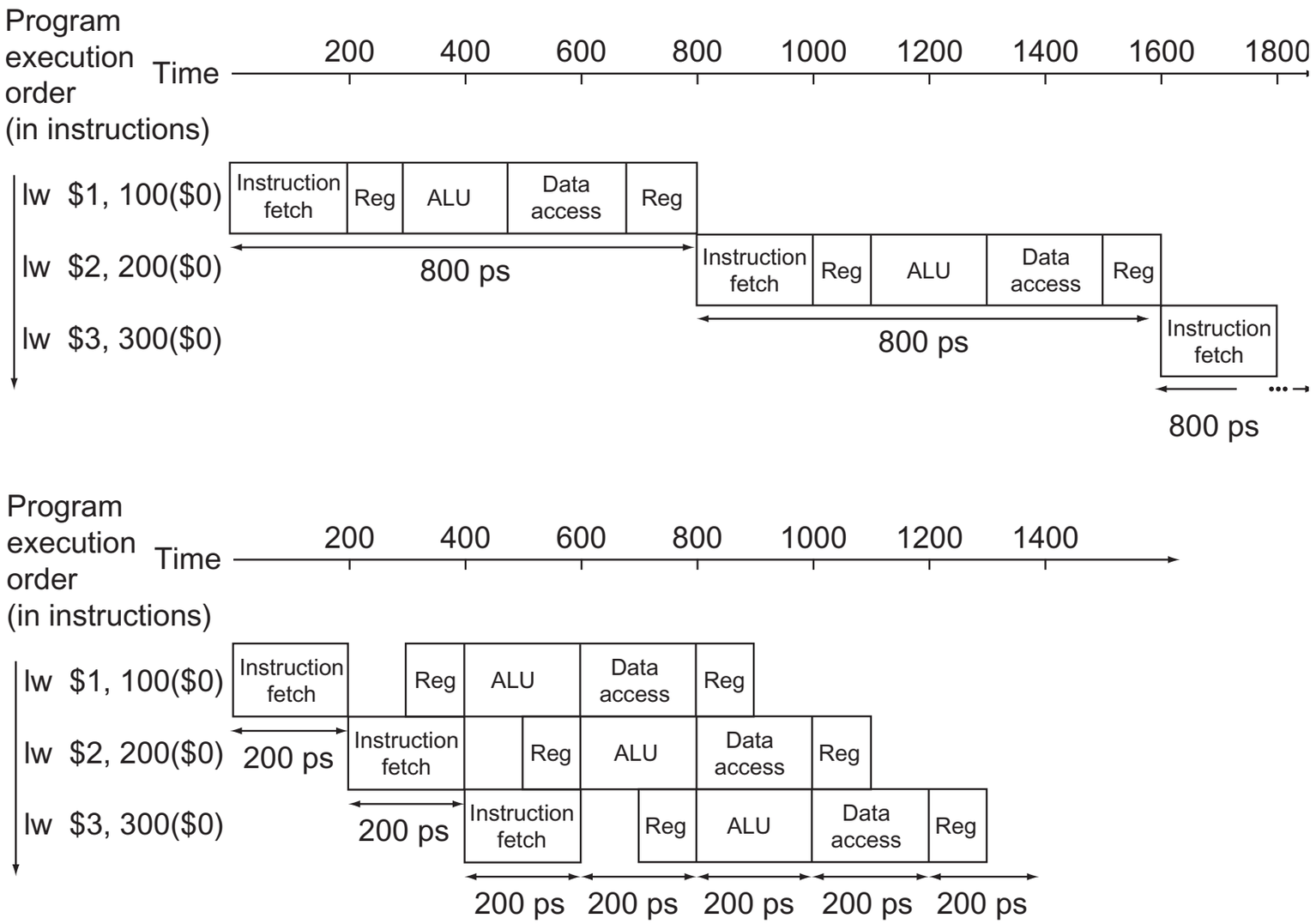
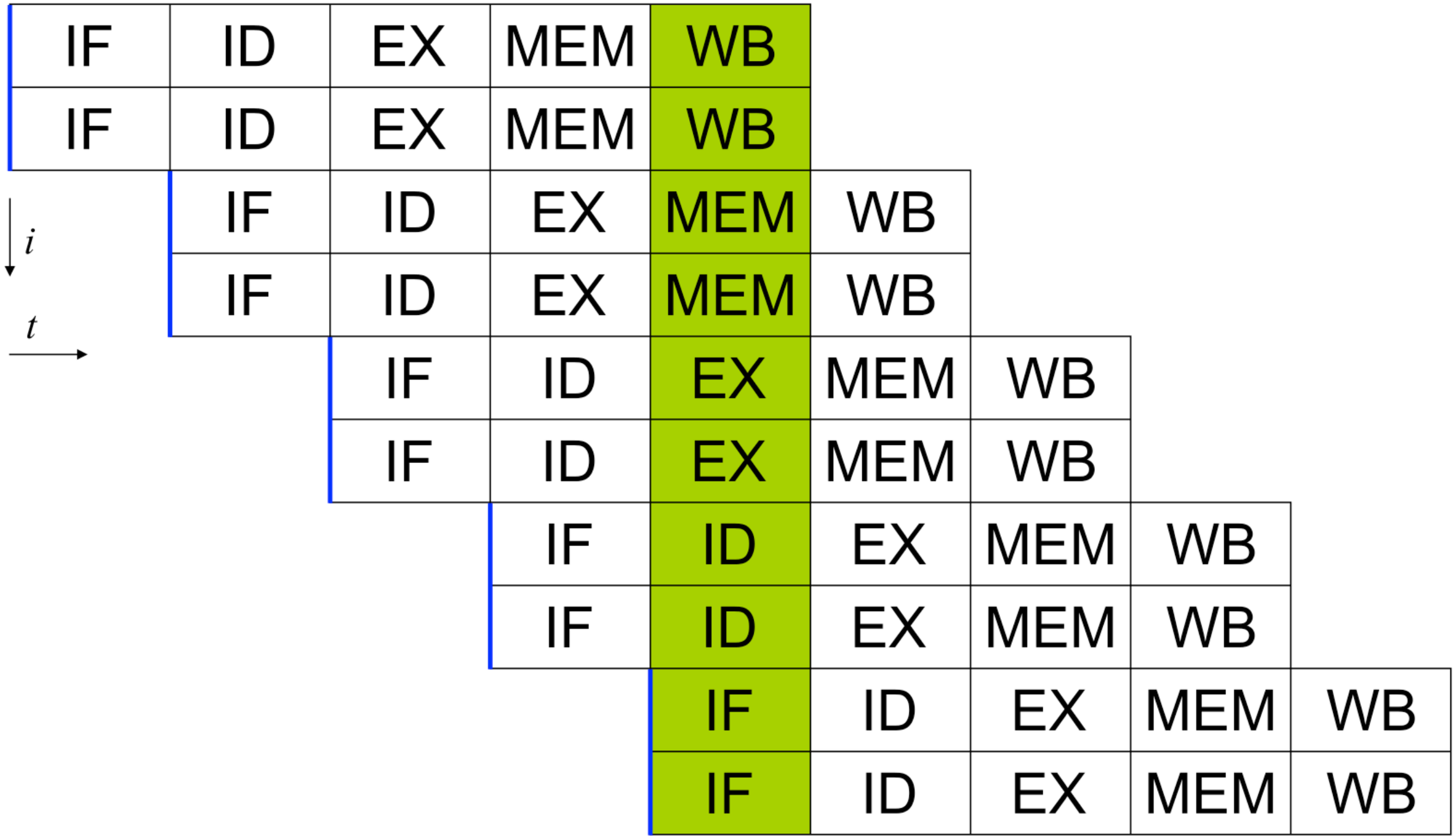
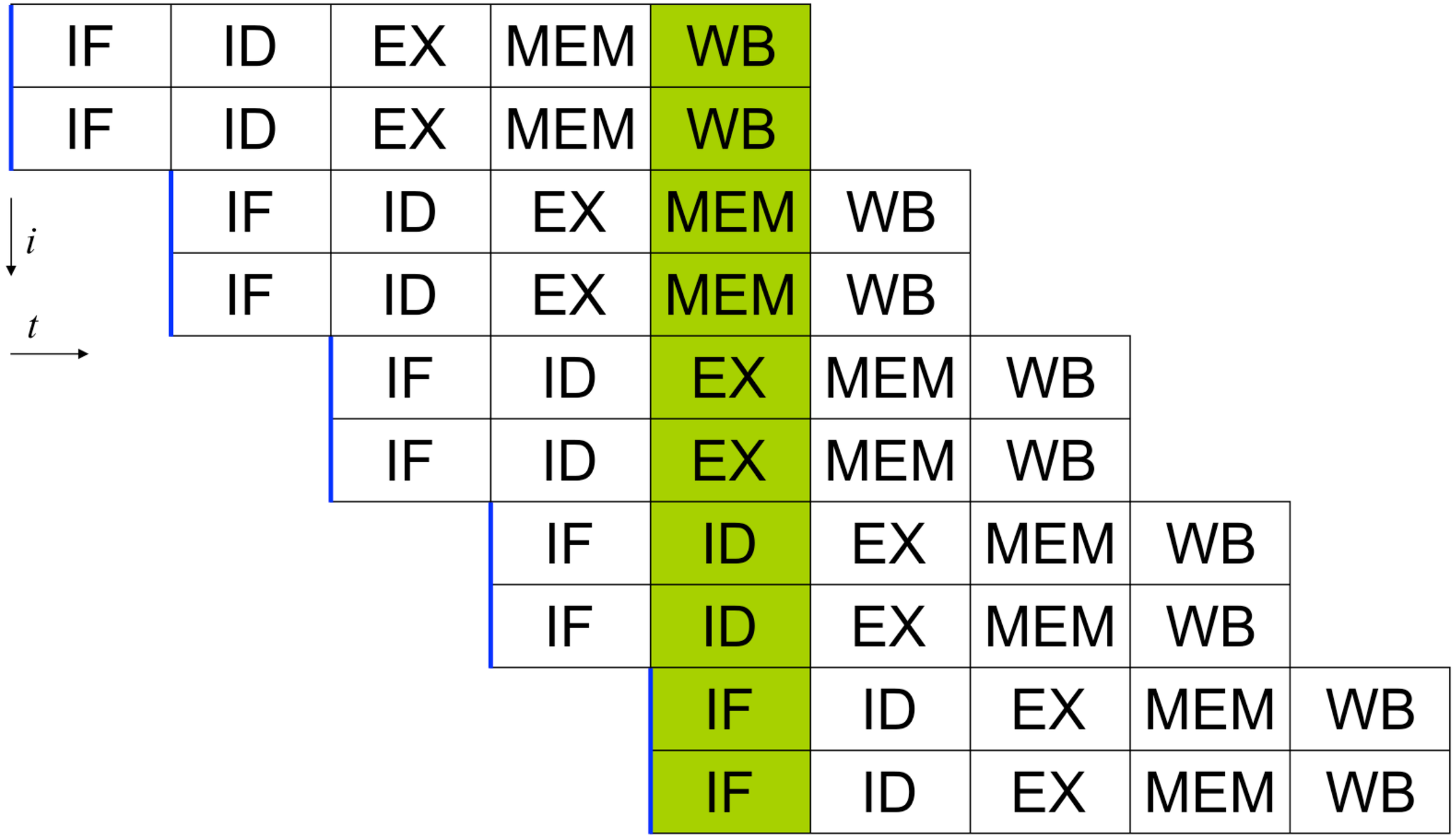


FIGURE 4.27 Single-cycle, nonpipelined execution in top versus pipelined execution in bottom. Both use the same hardware components, whose time is listed in Figure 4.26. In this case, we see a fourfold speed-up on average time between instructions, from 800 ps down to 200 ps. Compare this figure to Figure 4.25. For the laundry, we assumed all stages were equal. If the dryer were slowest, then the dryer stage would set the stage time. The pipeline stage times of a computer are also limited by the slowest resource: either the ALU operation or the memory access. We assume the write to the register file occurs in the first half of the clock cycle and the read from the register file occurs in the second half. We use this assumption throughout this chapter.

Actually…

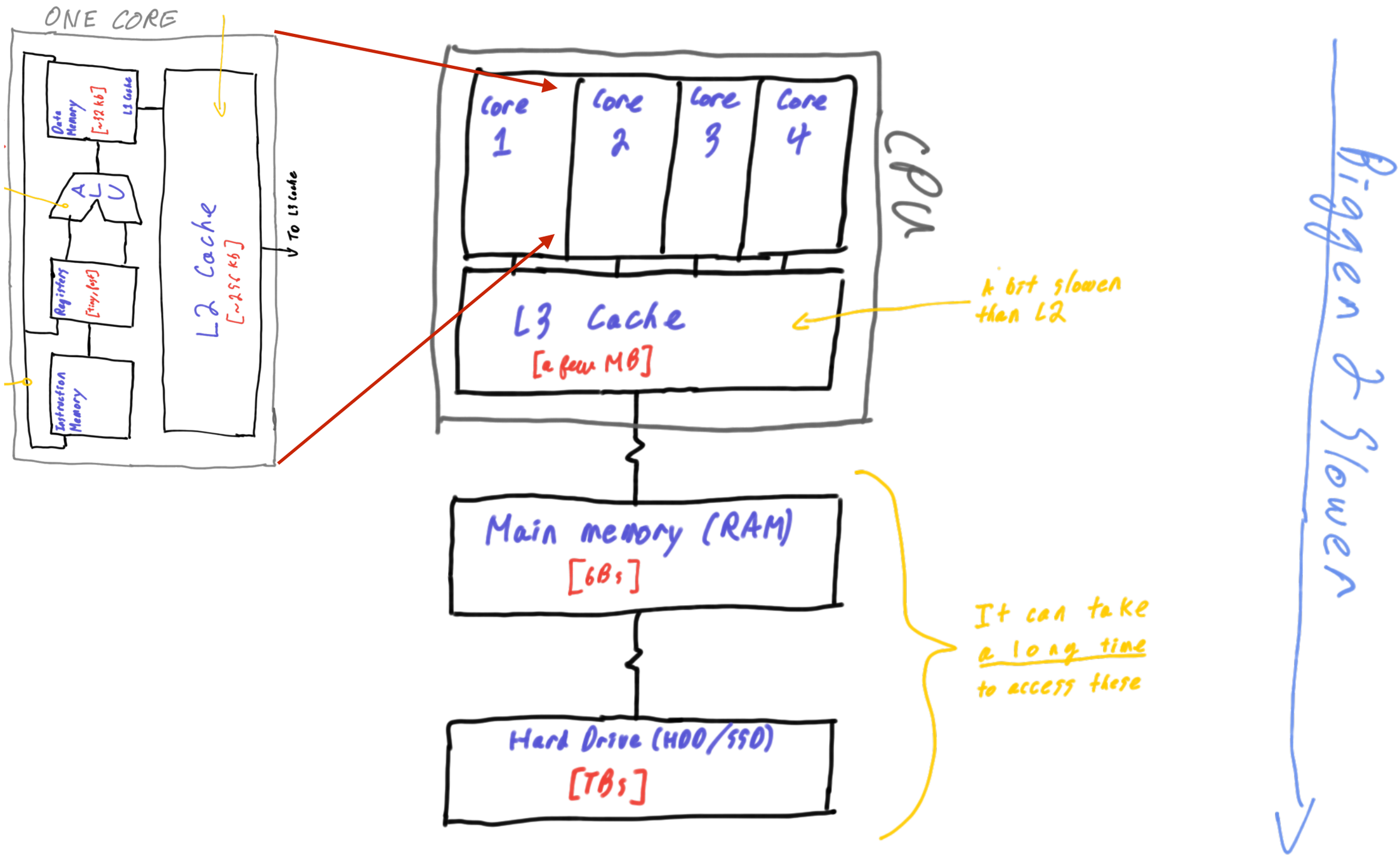

IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
i
t
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB

Actually…

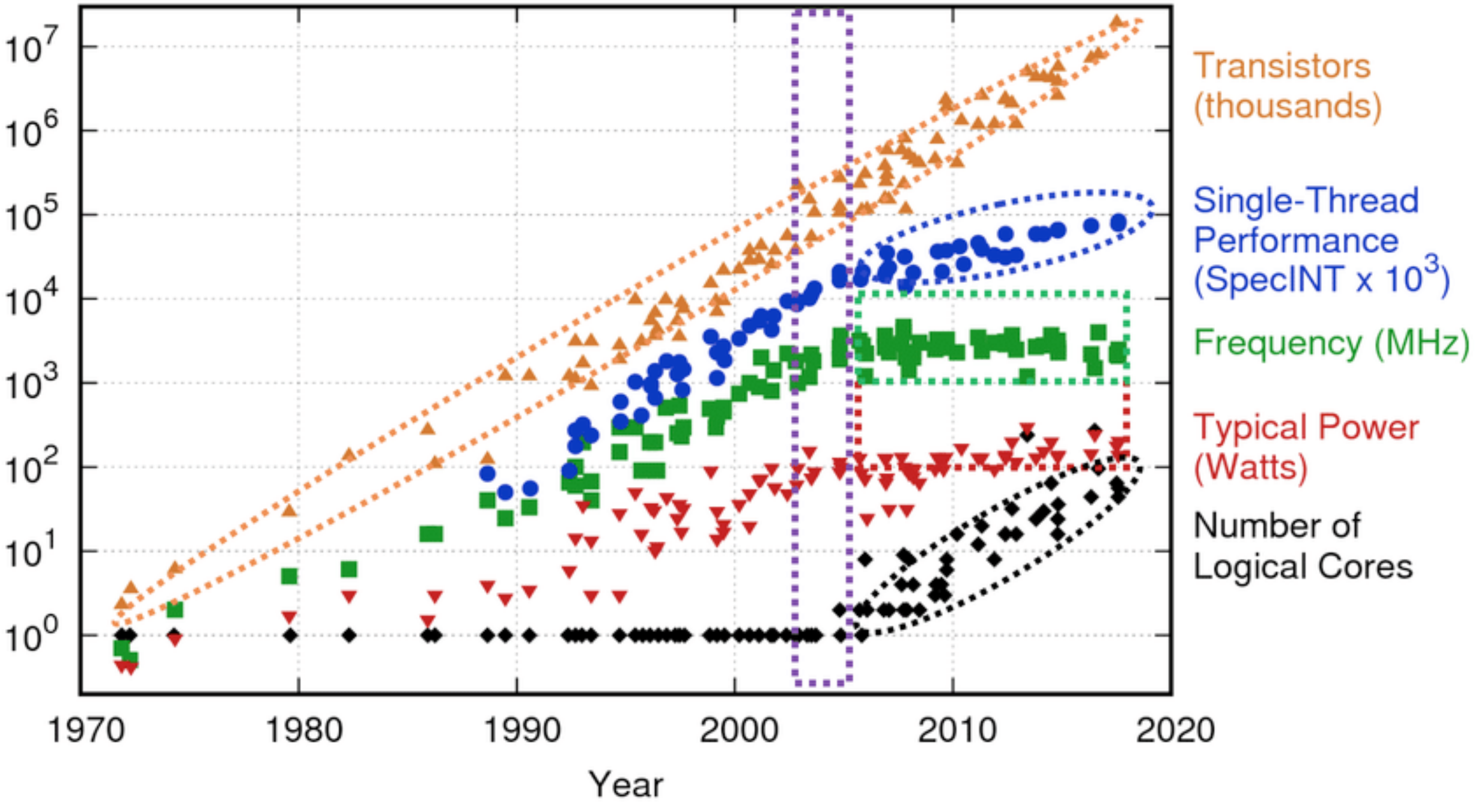


Solution: LoopVectorization.jl and @avx

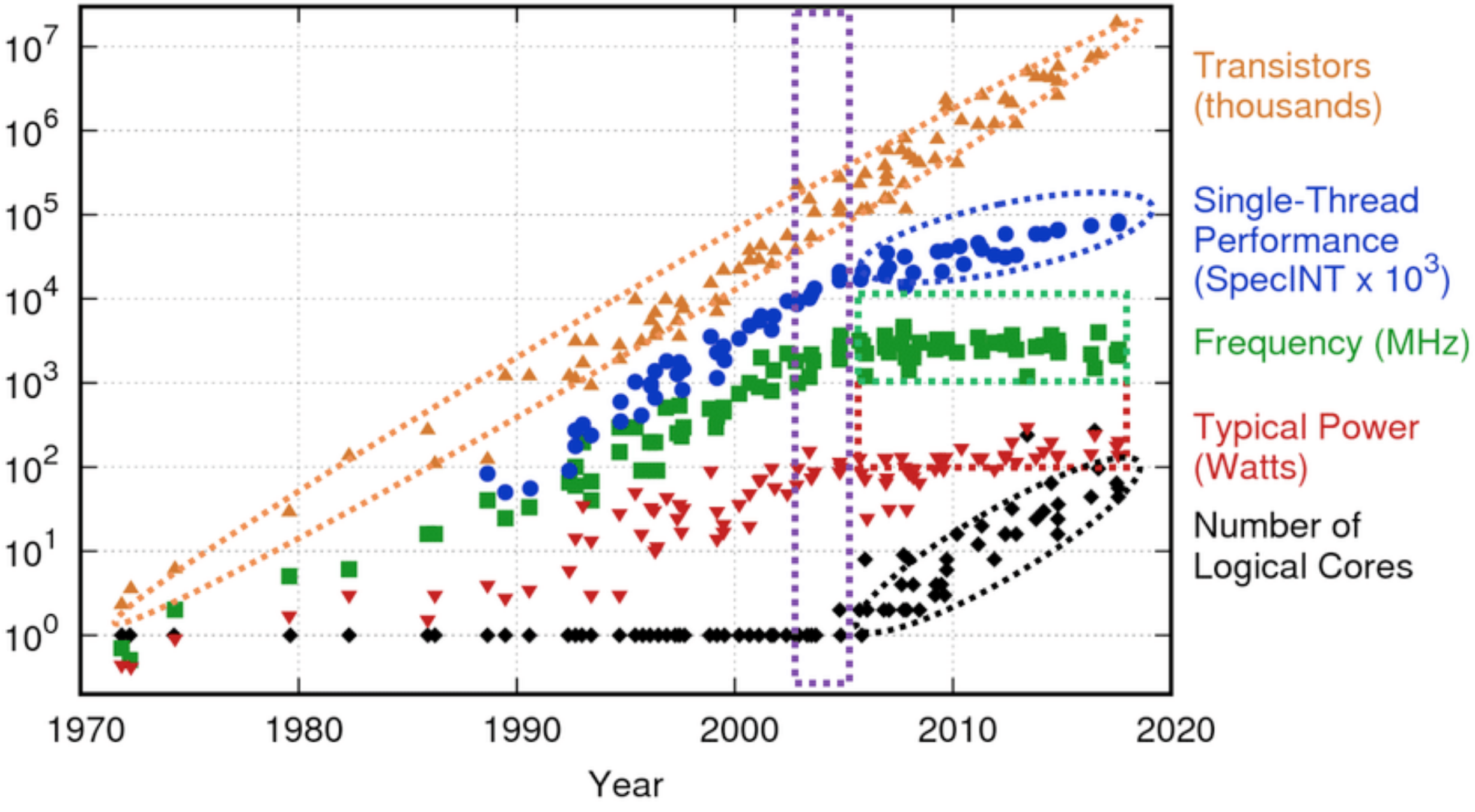
Meanwhile, there are now multiple cores in a CPU



Meanwhile, there are now multiple cores in a CPU



Meanwhile, there are now multiple cores in a CPU



Solution: Shared-memory parallel programming
e.g. Base.Threads.@threads, etc.

Latency

```
"Latency numbers every programmer should know"

L1 cache reference                           0.5 ns
Branch mispredict                            5   ns
L2 cache reference                           7   ns                      14x L1 cache
Main memory reference                      100   ns                     200x L1 cache
Read 1 MB sequentially from memory     250,000   ns       250 us
Read 1 MB sequentially from SSD*     1,000,000   ns     1,000 us     1 ms
Disk seek                           10,000,000   ns    10,000 us    10 ms
Read 1 MB sequentially from disk    20,000,000   ns    20,000 us    20 ms
```