24-cache-p3-notes

Caching: P3

Agenda:

- 0. Re-Orienting
- 1. The Memory Mountain
- 2. Cache-Friendly Code

Reading: ...on up to 6.5 and 6.6, too

slides.pdf (https://ssl.cs.dartmouth.edu/~sws/cs51-s15/24-cache-p3/slides.pdf)

---

0. Re-Orienting

Computation...all the way down to the laws of physics.

Mapping high-level languages to ISAs

Building datapaths out of combo logic (with some registers too)

Then using sequential logic to get the datapath implementing the ISA

But what about making it **faster?**.

Two main techniques:

- Caching
- Pipelining

---

1. The Memory Mountain

(Not to be confused with the Memory Pyramid :)

6.6.1.

- working set
- read throughput/read bandwidth
- size
- stride

Then:

- "slopes of spatial locality" (visible if you keep the working set fixed)
- "ridges of temporal locality" (visible if you keep the stride fixed)

Discussion in 6.6.1, e.g.:

- "falling off the back of the ridge... for large strides in small working set sizes, these overheads are not amortized"
- "flat ridge line for strides of 1 and 2.... hardware prefetching"

---

2. Cache-Friendly Code

Sec 6.5.

(And crazy fine-tuning of tables/data structures)