The memory system

load + store instructions

`movl` in x86

20-50% of all instructions

But it's actually much more complicated

CPU "chip"/package
RAM: Random access memory

DRAM → Dynamic (must be refreshed)

SRAM → Static RAM

DRAM cell

-1 if capacitor is charged
0, otherwise (charged)
- capacitor leaks over time
  ↓ refresh the charge
  1 - 100 ms

- DRAM cells are very small
- Goal of DRAM is highest density + lowest cost
SRAM cell 2 NOT gates in a feedback loop
- 6 transistor cell
  - some cells up to 12
  - no need to refresh
  - but still leakage

- Goal with SRAM → Speed or L1 cache low power
  - or registers → L3 cache

Storage devices
disks → still very common
non-volatile memories, but soon primarily flash others?

Disks → spinning platters
- arm with a head that detect magnetic variations
Flash - "floating gate transistor"
- kind of like DRAM, but not a capacitor
- Faster than disk
- Slower than DRAM
- Much denser than DRAM
- Many drawbacks (limited # writes) (read in blocks)
DRAM chip

"super cell" bunch of bits

0 1 2 3

0 1 2 3

row buffer

column access strobe (CAS)

row access strobe (RAS)
Memory controller gets an address

0x8125ca70

Going to get 641 bytes -> ignore bottom 6 bits

\[ \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
\end{array} \]

Row
Column

L > bank/rank/memory controller

DDR2

-7

DRAM DIMM -> Dual inline Memory Module

DDR3 -> common today

DDR4 -> becoming common

6DDR5 -> graphics DRAM

LPDDR 2/3/4 -> phones
Cool new memory technologies
HBM => high bandwidth memory
HMC => hybrid memory cube
Wide IO