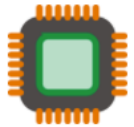


CASPER

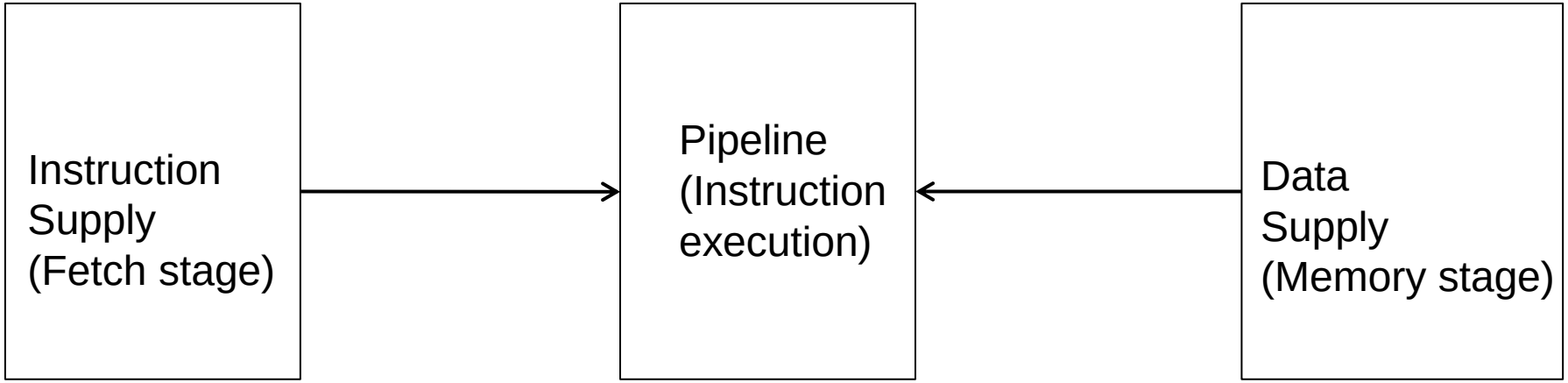
CS305: Computer Architecture

Memory Hierarchy

https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html

https://www.cse.iitb.ac.in/~biswa/

The Ideal World



- Zero-cycle latency
- Infinite capacity
- Perfect control flow

- Zero-cycle latency
- Infinite capacity
- Infinite bandwidth

World of Memory Hierarchy: Why?

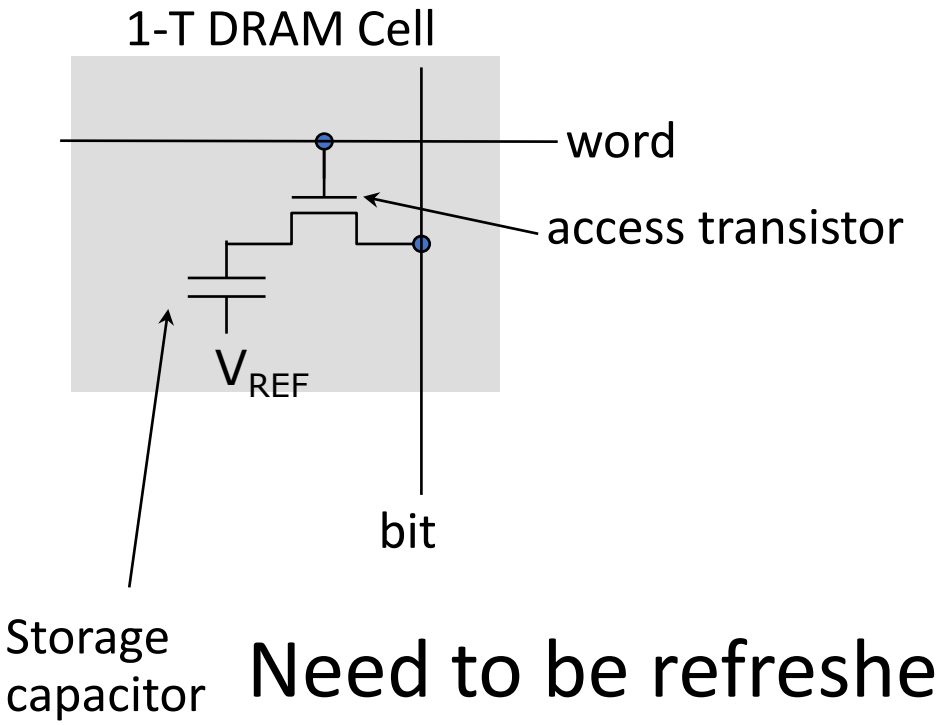
Before that What is memory?

Semiconductor Memory

- Semiconductor memory began to be competitive in early 1970s
  - Intel formed to exploit market for semiconductor memory
  - Early semiconductor memory was Static RAM (SRAM). SRAM cell internals similar to a latch (cross-coupled inverters).

- First commercial Dynamic RAM (DRAM) was Intel 1103
  - 1Kbit of storage on single chip
  - charge on a capacitor used to hold value

One transistor DRAM



Denser

Value kept in one cell is as per the charge in the capacitor

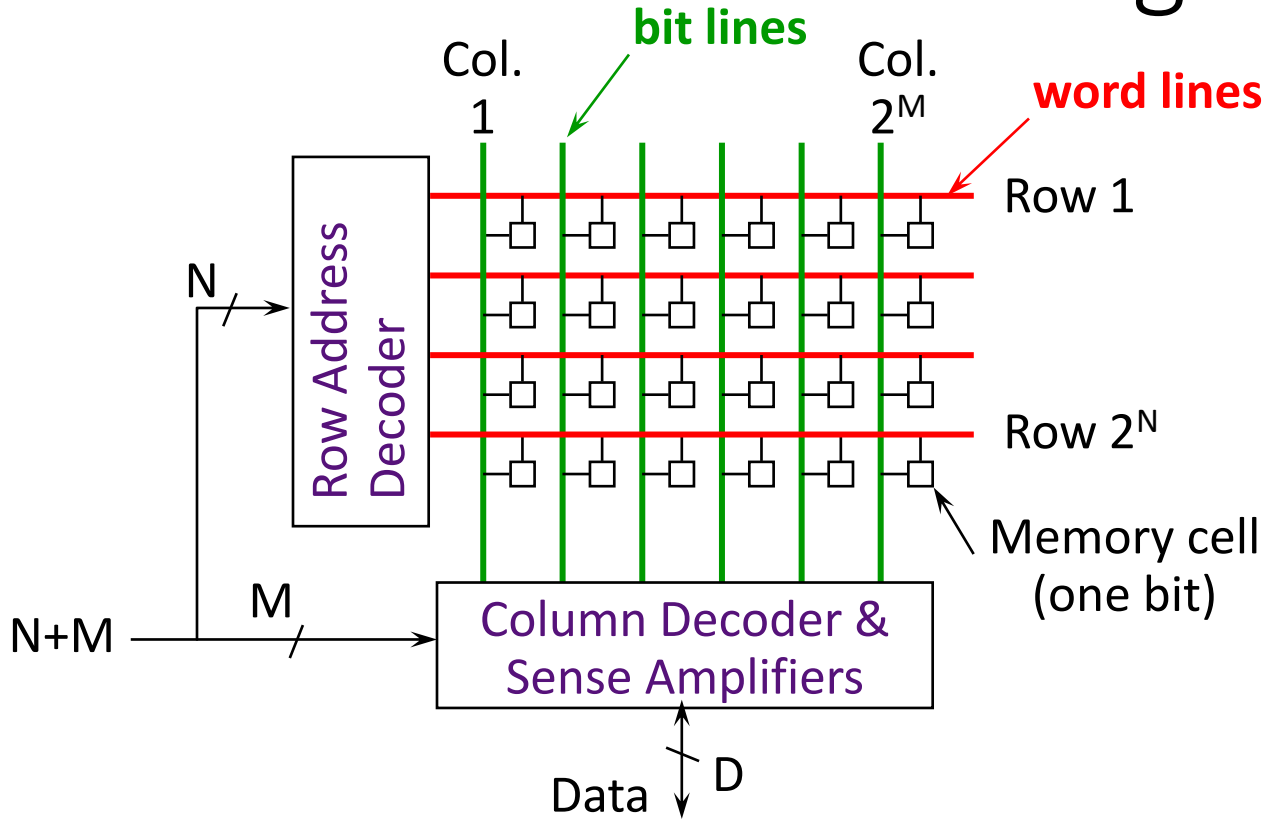
Need to be refreshed periodically to maintain the charge

So dynamic RAM (DRAM)

DRAM

- Dynamic random-access memory
- Capacitor charge state indicates stored value
  - Whether the capacitor is charged or discharged indicates storage of 1 or 0
  - 1 capacitor
  - 1 access transistor
- Capacitor leaks
  - DRAM cell loses charge over time
  - DRAM cell needs to be refreshed

10K feet view on DRAM organization



Bits stored in 2-dimensional arrays on chip

SRAMs (6T to 8T)

Static RAMs. No need of refresh as no capacitor.

Faster access (no capacitor)

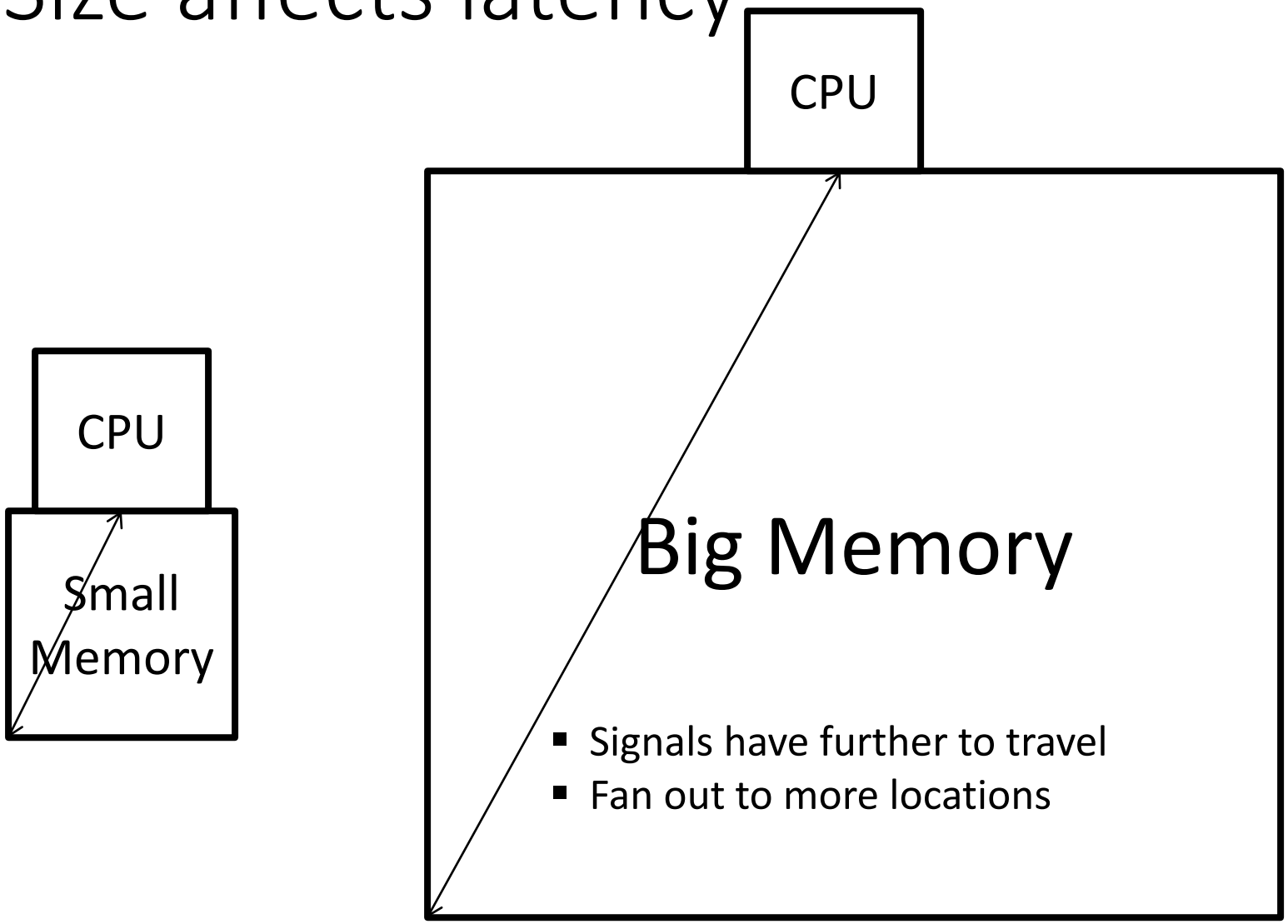
Density low as 6T: 1 bit compared to 1T: 1bit in DRAM

Costly than DRAM

The Problem

- Bigger is slower
  - SRAM, 512 Bytes, sub-nanosec
  - SRAM, KByte~MByte, ~nanosec
  - DRAM, Gigabyte, ~50 nanosec
  - Hard Disk, Terabyte, ~10 millisec
- Faster is more expensive (dollars and chip area)
  - SRAM, < 1000$ per GB
  - DRAM, < 20$ per GB
  - Hard Disk < 0.01$ per GB
  - These sample values scale with time
- Other technologies have their place as well
  - Flash memory, NVRAM, MRAM etc

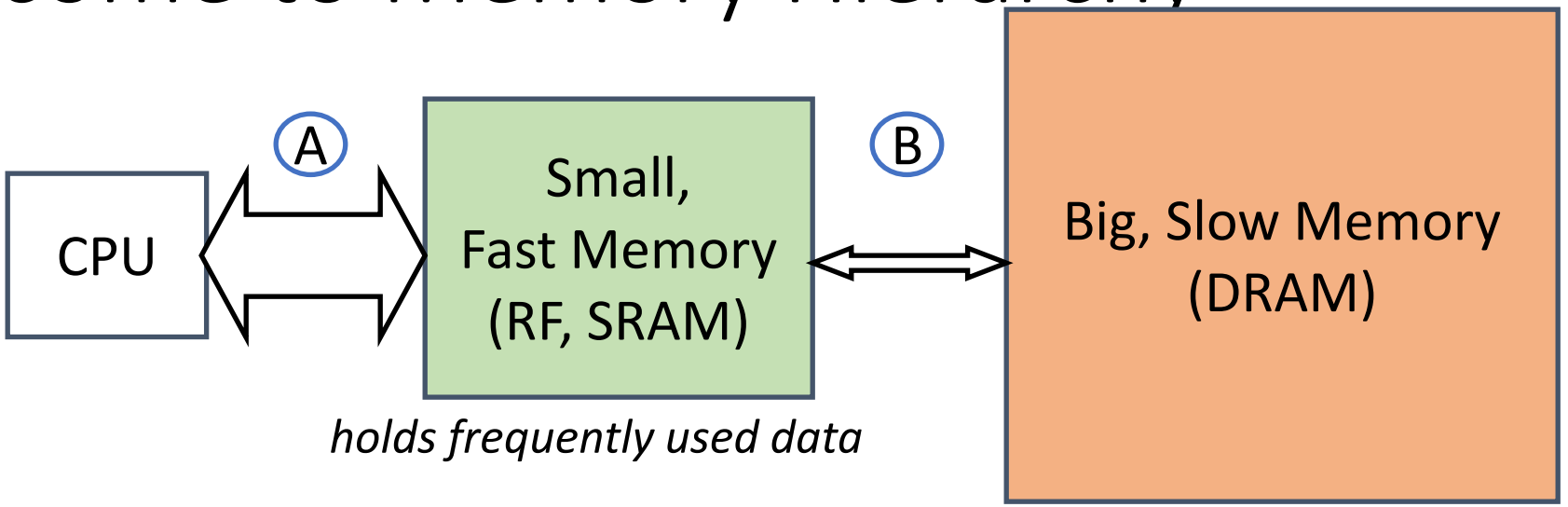
Size affects latency



Why Memory Hierarchy

- We want both fast and large
- But we cannot achieve both with a single level of memory
- Idea: Have multiple levels of storage (progressively bigger and slower as the levels are farther from the processor) and ensure most of the data the processor needs is kept in the fast(er) level(s)

Welcome to Memory Hierarchy



- *capacity*: Register << SRAM (Cache) << DRAM
- *latency*: Register << SRAM (Cache) << DRAM
- *bandwidth:* on-chip >> off-chip

On a data access:

if data $\in$ fast memory $\Rightarrow$ low latency access *Cache*

if data $\notin$ fast memory $\Rightarrow$ high latency access *DRAM*

Asante