Advanced Caches → Memory → Storage

Topics:
- SRAM
- DRAM

Memory

This lesson discusses the basic 1 bit DRAM and SRAM. Then expands to discuss the organization of a large scale memory system, including row and column reads and writes. Memory controllers and their role in DRAM is also explained.

Memory Technology: SRAM and DRAM

The RANDOM ACCESS part in both names simply refers to the fact that we can access any memory location by address without needing to go through all the memory locations. So random access is as opposed to sequential access like a tape, where you have to actually scan through the whole tape to get somewhere

6T = 6 Transistor Bit Cell

1T = 1 Transistor (+ Capacitor) Bit Cell

SRAM = static random access memory. Static = retains data when power is on.

DRAM = dynamic random access memory. Dynamic = loses data unless refreshed. (even if connected to a power source!)

DRAM has "Destructive Reads" (the capacitor drains through the Bitline when the Wordline is selected)

SRAM requires more transistors than DRAM, but is faster.

SRAM is faster and simpler (no refreshing) than DRAM, but takes more transistors = costly.

Note that both of these types of memory will lose data when the power is not supplied.
However, even with power, DRAM is "leaky" and needs to be continuously refreshed.

Memory Chip Organization

Memory is organized and accessed using Row Decoders and Column Decoders.

So a row can be written and read at the same time, this is called fast page mode.

On Read, we read the whole Row/wordline out and then select the SINGLE BIT we want from that row.
On Writes, we first READ THE WHOLE ROW, update the bit(s) of interest in the row, and then PUSH the new, updated ROW back to the wordline.

Using the fast page mode, memory can be read in a more efficient order.

Fast-Page Mode - The "Row Buffer" is like a Cache, and Fast-Page Mode is similar to the Write-Back Cache Policy on that Row Buffer. Nothing to do with Virtual Mem Pages.
- Opening Up A "Page" = Trigger Row Decoder, Read into Sense Amplifier, store in Row Buffer.
- Closing the "Page" = Sense Amplifier "writes" the Row Buffer contents back to the Wordline

Open the "Page" -> Multiple Reads/Writes -> Close the Page (Always need to close the page, even for (destructive) reads)

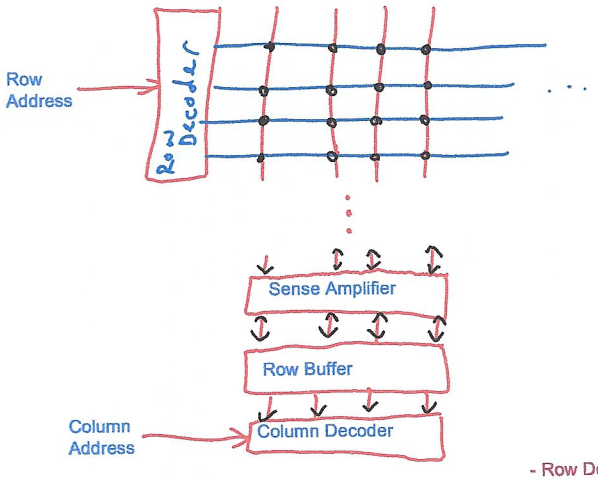
Connecting DRAM to the Processor

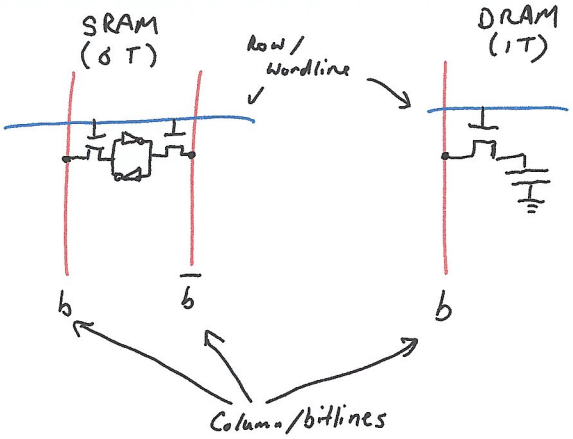
DRAM is accessed to the processor through the front-side bus using a memory controller.

~~The downside of this is the DRAM must be much more standardized and inflexible to work with the on-chip memory controller.~~

Old Architecture = CPU Chip (contains Core + Caches) -> Front-Size Bus -> Memory Controller -> Memory Channels -> DRAMs
- Both the Front-Side Bus and the Memory Channel added "travel" latency to the data transfer from DRAM to the Last Level Cache.

New Architecture = CPU Chip (contains Core + Caches + Memory Controller) -> Memory Channels -> DRAMs
- This is much faster, but requires the DRAMs to adhere to a much stricter standard interface in order to communicate with the smaller, on-chip Memory Controller





For SRAM:
- On reads, we read the different between b & b' to determine the direction (which gives us 0 or 1).
- On write, set both b and b' to "overpower" the inverter circuit and set the value.

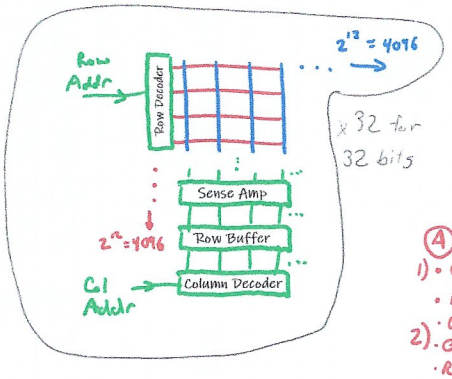
- Row Decoder - Given Row Address, decides which Wordline (blue) to activate.

- Bit Lines (red) are long and are affected by physics. Use 1 "Sense Amplifier" to mitigate.
  - For DRAM (which has Destructive Reads), the Sense Amplifier must Rewrite After Read.

- The Amplified Bitline Output goes into a "Row Buffer" that stores the whole outputted Row.

- Row Buffer feeds data to the "Column Decoder" - selects 1 Bit from the Entire Row.

- Want to read more than one bit? Replicate the structure to select more bits.



DRAM ACCESS SCHEDULING QUIZ

- DRAM HAS 32 1-BIT ARRAYS
- EACH ARRAY IS 16 MBIT $2^{12} \times 2^{12}$

- CACHE MISSES FOR

1. FOO FOO
2. E00 F00
3. F00 E04
4. E04 F00
5. F00 E00
6. F00 123
7. 123 F00

- PAGE OPEN 10ns
- READ FROM ROW BUF 2ns
- PAGE CLOSE 5ns

(A) IN THIS ORDER 119 ns

(B) BEST ORDER 74 ns

(A)

1) • Open Page (Row Addr = F00)
   • Read
2) • Close Page
   • Open Page (Row Addr = E00)
   • Read
3) • Close Page
   • Open Page (Row Addr = F00)
   • Read
4) • Close Page
   • Open Page (Row Addr = E04)
   • Read
5) • Close Page
   • Open Page (Row Addr = E00)
   • Read
6) • Close Page
   • Open Page (Row Addr = F00)
   • Read
7) • Close Page
   • Open Page (Row Addr = 123)
   • Read
   • Close Page

So for each memory access in this order, we had to open → read → close

$7 \times (10ns + 2ns + 5ns) = 119$ ns

(B) Group accesses by Row Address

For example,

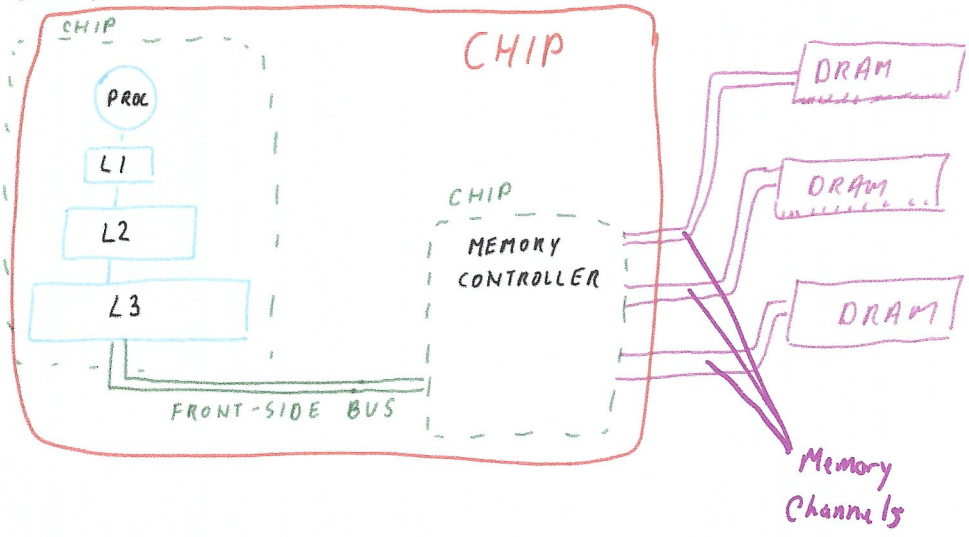
1), 3), 6) } F00
2), 5) } E00
4) – E04
7) – 123

- Open Page (Row Addr = F00)
  - Read (1) (Col Addr = F00)
  - Read (3) (Col Addr = E04)
  - Read (6) (Col Addr = 123)
- Close Page
- Open Page (Row Addr = E00)
  - Read (2) (Col Addr = F00)
  - Read (5) (Col Addr = E00)
- Close Page
- Open Page (Row Addr = E04)
  - Read (4) (Col Addr = F00)
- Close Page
- Open Page (Row Addr = 123)
  - Read (7) (Col Addr = F00)
- Close Page

(# Opens)(10ns) + (# Reads)(2ns) + (# Closes)(5ns)

(4)(10ns) + (7)(2ns) + (4)(5ns) = 74 ns

CONNECTING DRAM TO THE PROCESSOR



MEMORY REFRESH QUIZ

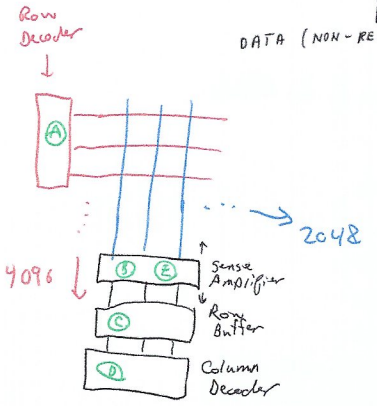
- MEMORY HAS 4096 ROWS, 2048 COLUMNS
- REFRESH PERIOD IS 500 μs ← Every cell needs to be refreshed within 500 us before it loses data.
- READ TIMING:
  - A. 4 ns TO SELECT A ROW
  - B. 10 ns FOR SENSE AMP TO GET BIT VALUES
  - C. 2 ns TO PUT DATA IN ROW BUFFER
  - D. 4 ns FOR COLUMN DECODER
  - E. 11 ns TO WRITE DATA FROM SENSE AMP TO MEMORY ROW ← OVERLAP
- THIS MEMORY CAN SUPPORT

31808000

DATA (NON-REFRESH) READS PER SECOND

nano = 10^{-9}

(μ) micro = 10^{-6}



$t = 500\mu$ MAX b/w Refreshes

Read time

A → B → C → D → 4ns → 10ns → 2ns → 4ns

↘ E → 11ns

(E & C → D occur in parallel)

So a read-then-write operation takes

4+10+11 = 25 ns = 0.025 μs

If we have 4096 rows, each of which need to be refreshed at least every 500 μs (each of which will take 0.025 μs).

So for a single row, how many times is it getting "Refreshed"?

$\frac{1\text{ s}}{500\,\mu\text{s}}$ = 2000 refreshes per row, per second.

Since the Rows are Refreshed Sequentially (rows can overlap), then we can calculate the total # of Refreshes for the entire memory unit

$\left(\frac{2000 \text{ refreshes per second}}{1 \text{ row}}\right)(4096 \text{ rows}) = 8192000$ refreshes per second

How much time will this take?

$(8192000 \text{ refreshes})\left(\frac{0.025\,\mu\text{s}}{1 \text{ refresh}}\right) = 0.2048$ seconds (total time spent on all Refreshes in 1 second)

How much time is left in that second for normal memory accesses?

1 − 0.2048 = 0.7952 seconds

How many memory accesses can we fit in that time?

$\frac{0.7952 \text{ seconds}}{0.025\,\mu\text{s}} = 31808000$