ECE 571 – Advanced Microprocessor-Based Design Lecture 15

Vince Weaver

http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

29 November 2019

Announcements

- HW7 was assigned
- Midterm next class!
- Useful readings:
  - "A performance and power comparison of modern high-speed DRAM arch" from MEMSYS 2018
  - "DRAM Refresh Mechanisms, Penalties, and Trade-offs" Bhati, Chang, Chishti, Lu and Jacob. IEEE transactions on Computers, 2016.

- Note that ECE598 (NASA class) with be MW 2-3:15
- Note from last class – modern motherboards have 100MHz clock and use PLL to multiply the frequency

Midterm Review

Closed book/laptop/phone but can have front of one 8.5x11 piece of paper worth of notes if you want.

1. Performance/Benchmarking
   - Be familiar with the general idea of performance counters and interpreting perf results.
   - Benchmark choice: it should match what you plan to do with the computer.
   - Know a little about the difference between integer benchmarks and floating point (integer have

more random/ unpredictable behavior with lots of conditionals; floating point are often regular looped strides over large arrays or data sets)

- Be familiar with concept of skid.

2. Power
   - Know the CMOS Power equation
   - Energy, Energy Delay, Energy Delay Squared
   - Idle Power Question
3. Branch Prediction
   - Static vs Dynamic

- 2-bit up/down counter
- Looking at some simple C constructs say expected branch predict rate

4. Cache

- Given some parameters (size, way, blocksize, addr space) be able to calculate number of bits in tag, index, and offset.
- Know why caches are used, that they exploit temporal and spatial locality, and know the tradeoffs (speed vs nondeterminism)

- Be at least familiar with the types of cache misses (cold, conflict, capacity)
- Know difference between writeback and write-through
- Be able to work a few simple steps in a cache example (like in HW#5)

5. Prefetch
   - Why have prefetchers?
   - Common prefetch patterns?

6. Virtual Memory
   - General concept of VM

- Benefits of VM?

  Memory Protection, each program has own address space, allows having more memory than physical memory, demand paging, copy-on-write for fork, less memory fragmentation, etc.
- Why is TLB behavior important?

  Depending on cache config:

  worst case: (VIVT) every memory access looked up in TLB best case: (PIPT) every cache miss looked up in TLB

Static RAM (SRAM) Review

- Used on chip: caches, registers, etc. Made in same process as CPU
- 6 transistors (or 4 plus hard-to make resistors with high static power)
- Cross-coupled inverters
  - For read, precharge both bitlines. Raise wordline.
  - Lots of capacitance so hard to swing whole way, so sense amp which amplified the small voltage shift
  - For write, set bitline and not-bitline, set wordline.

Overpowers inverters

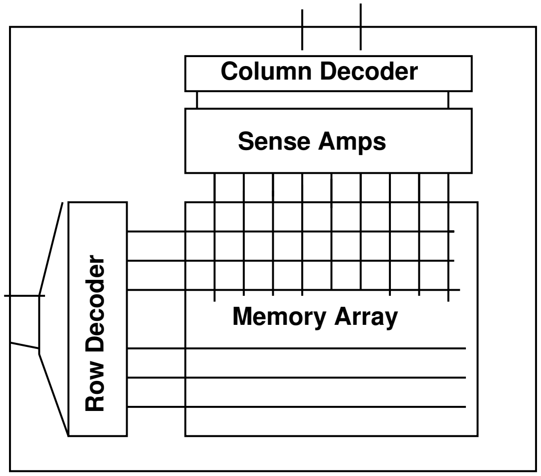
- Clocked or no, clocked saves power
- Bitlines might be braided to avoid noise

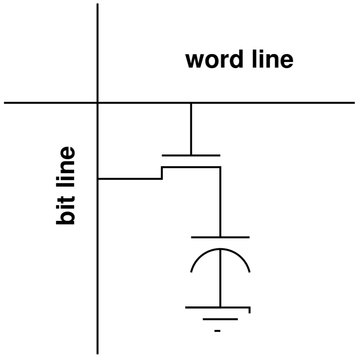
Why not have large SRAM

- SRAM is low power at low frequencies but takes more at high frequencies
- It is harder to make large SRAMs with long wires
- It is a lot more expensive while less dense (Also DRAM benefits from the huge volume of chips made)
- Leakage for large data structures
- Price: 8MB for $163

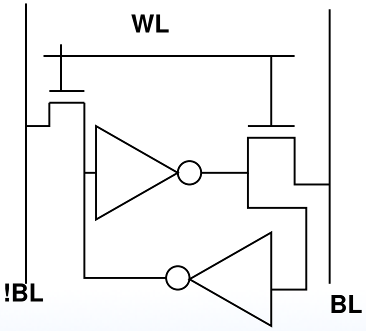
Diagram

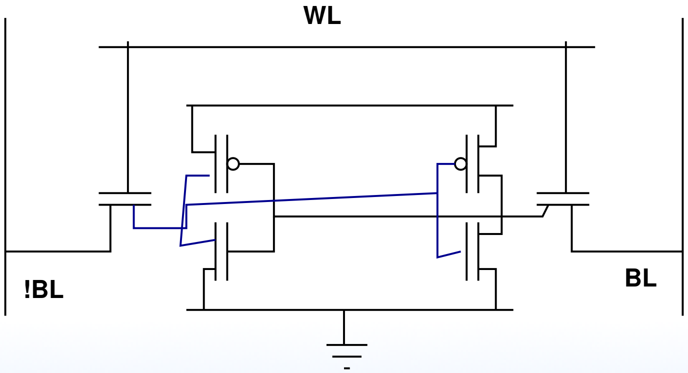
DRAM





SRAM



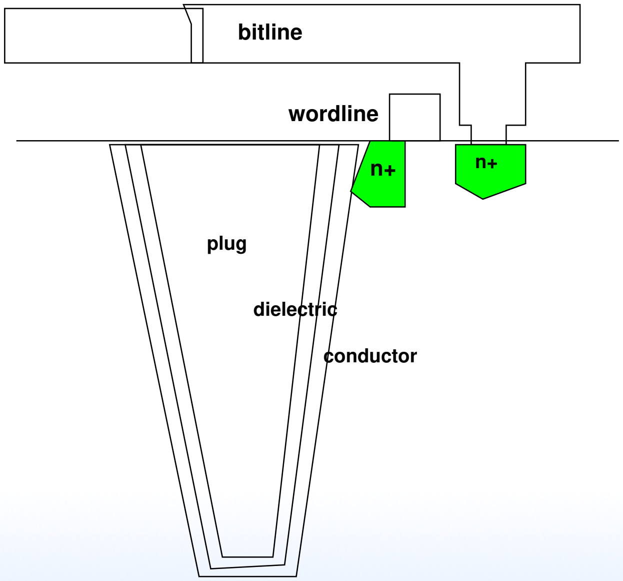


DRAM

- Single transistor/capacitor pair. (can improve behavior with more transistors, but then less dense)
- In 90nm process, 30fF capacitor, leakage in transistor 1fA. Can hold charge from milliseconds to seconds.
- DRAMs gradually lose charge, need to be refreshed.
- Need to be conservative. Refresh every 32 or 64ms (if 8192 rows, then 64ms/8192 is 7.8us)
- DRAM read is destructive, always have to write back

Low Level

- Planar (old)
- Trench Capacitors (transistors above)
- Stacked Capacitors (transistors below)



SIMMs/DIMMS

- How many chips on DIMM? 8? 9?
  9 usually means ECC/parity
- Chips x1 x4 x8 bits, how many get output at a time. Grouped together called a "bank"
- Banks can mask latency, sort of like pipelining. If takes 10ns to respond, interleave the request.
- DIMM can have independent "ranks" (1 or 2 per DIMM), each with banks, each with arrays

- Layout, multiple mem controllers, each with multiple channels, each with ranks, banks, arrays

- Has SPD “serial presence detect” chip that holds RAM timings and info. Controlled by smbus (i2c)

- SODIMM – smaller form factor for laptops “small outline”

Refresh

- Need to read out each row, then write it back. every 32 to 64ms

- Old days; the CPU had to do this. Slow
  Digression: what the Apple II does

- Newer chips have “auto refresh”

Low-Level Memory Bus

- JEDEC-style. address/command bus, data bus, chip select

- Row address sends to decoder, activates transistor

- Transistor turns on and is discharged down the column rows to the sense amplifier which amplifies

- The sense amplifier is first “pre-charged” to a value halfway between 0 and 1. When the transistors are enabled the very small voltage swing is amplified.

- This goes to column mux, where only the bits we care about are taken through

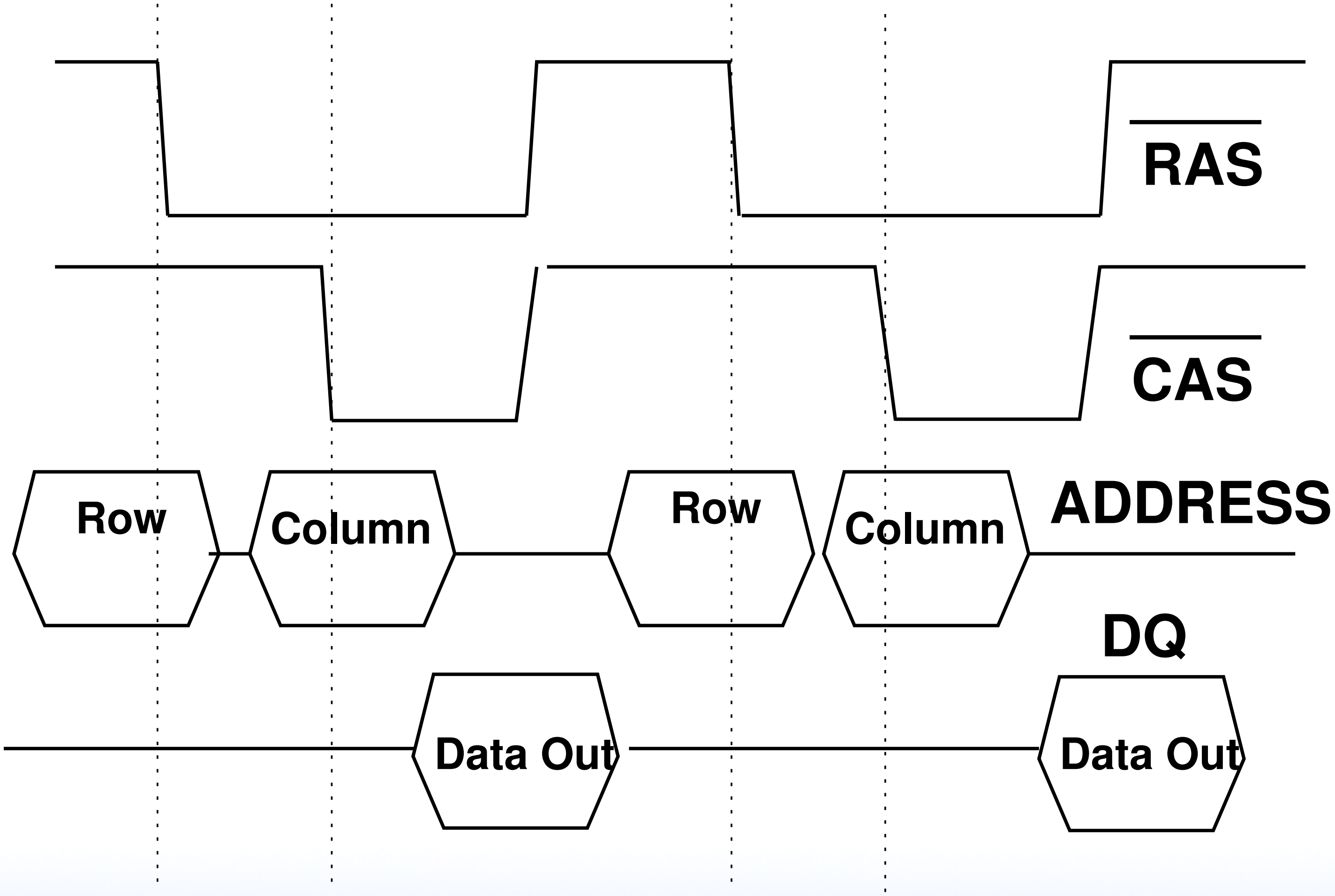
Memory Access

- CPU wants value at address (cache miss?)
- Passed to memory controller
- Memory controller breaks into rank, bank, and row/column
- Proper bitlines are pre-charged
- Row is activated, then $\overline{RAS}$, row address strobe, is signaled, which sends all the bits in a row to the sense

amp. can take tens of ns.

- Then the desired column bits are read. The $\overline{CAS}$ column address strobe sent.
- Again takes tens of ns, then passes back to memory controller.
- Unlike SRAM, have separate CAS and RAS? Why? Original DRAM had low pincount.
- Also a clock signal goes along. If it drives the device it's synchronous (SDRAM) otherwise asynchronous

Async DRAM Timing Diagram



Memory Controller

- Formerly on the northbridge
- Now usually on same die as CPU

Advances in Memory Technology

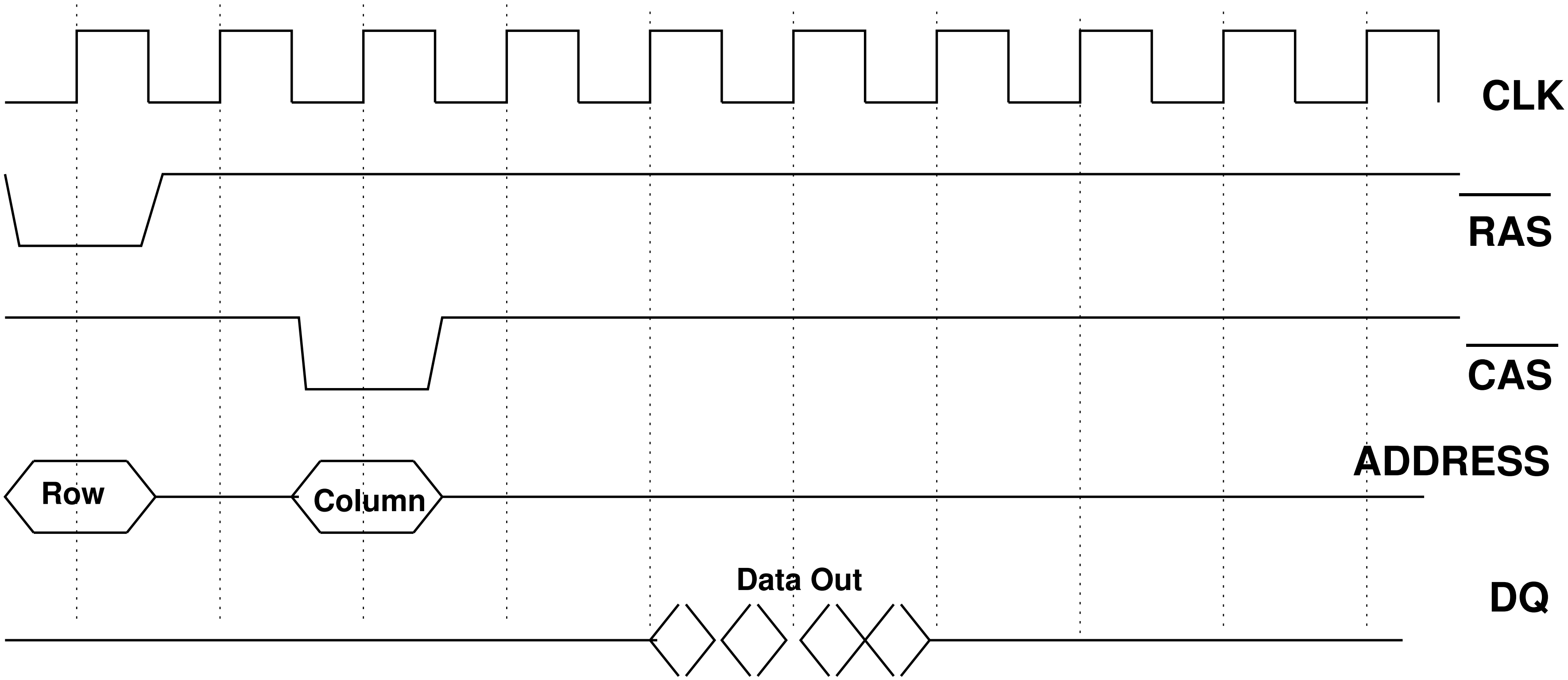
In general the actual bit array stays same, only interface changes up.

- Clocked
- Asynchronous
- Fast page mode (FPM) – row can remain active across multiple CAS.
- Extended Data Out (EDO) – like FPM, but buffer

"caches" a whole page of output if the CAS value the same.

- Burst Extended Data Out (BEDO) – has a counter and automatically will return consecutive values from a page
- Synchronous (SDRAM) – drives internal circuitry from clock rather than outside RAS and CAS lines. Previously the commands could happen at any time. Less skew.

DDR Timing Diagram



Historical Memory Types

- SDRAM – 3.3V
- DDR – transfer and both rising and falling edge of clock 2.5V. Adds DLL to keep clocks in sync (but burns power)
- DDR2 – runs internal bus half the speed of data bus. 4 data transfers per external clock. memory clock rate * 2 (for bus clock multiplier) * 2 (for dual rate) * 64 (number of bits transferred) / 8 (number of bits/byte) so at 100MHz, gives transfer rate of 3200MB/s. not pin

compatible with DDR3. 1.8 or 2.5V

- DDR3 – internal doubles again. Up to 6400MB/s, up to 8gigabit dimms. 1.5V or 1.35V

- DDR3L - low voltage, 1.35V (not same as LPDDR3)

- DDR4 – recently released. 1.2V , 1600 to 3200MHz. 2133MT/s, parity on command/address busses, crc on data busses.

- DDR4L – 1.05V

- DDR5 – just announced
- GDDR2 – graphics card, but actually halfway between DDR and DDR2 technology wise
- GDDR3 – like DDR2 with a few other features. lower voltage, higher frequency, signal to clear bits
- GDDR4 – based on DDR3, replaced quickly by GDDR5
- GDDR5 – based on DDR3
- LPDDR2/LPDDR3/LPDDR4 – as with GDDR, not

necessarily a lot in common with DDR2/DDR3/DDR4 – NOTE TO FUTURE write up something on how it works

More obscure Memory Types

- RAMBUS RDRAM – narrow bus, fewer pins, could in theory drive faster. Almost like network packets. Only one byte at time, 9 pins?
- FB-DIMM – from intel. Mem controller chip on each dimm. High power. Requires heat sink? Point to point. If multiple DIMMs, have to be routed through each controller in a row.
- VCDRAM/ESDRAM – adds SRAM cache to the DRAM

- 1T-SRAM – DRAM in an SRAM-compatible package, optimized for speed

Memory Latencies, Labeling

- DDR400 $=$ 3200MB/s max, so PC3200
- DDR2-800 $=$ 6400MB/s max, so PC2-6400
- DDR2 5-5-5-15
  - C_L CAS latency
  - T_{RCD} row address to column address delay
  - T_{RP} row precharge time
  - T_{RAS} row active time
  - CMD (optional), command time
- DDR3 7-7-7-20 (DDR3-1066) and 8-8-8-24 (DDR3-1333).

Memory Parameters

You might be able to set this in BIOS

- Burst mode – select row, column, then send consecutive addresses. Same initial latency to setup but lower average latency.
- CAS latency – how many cycles it takes to return data after CAS.
- Dual channel – two channels (two 64-bit channels to memory). Require having DIMMs in pairs