ECE 571 – Advanced Microprocessor-Based Design Lecture 24

Vince Weaver

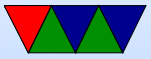
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

28 October 2020

Announcements

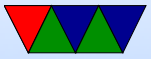
- Midterm on Friday
- Project will be posted
- Useful readings:
  - "A performance and power comparison of modern high-speed DRAM arch" from MEMSYS 2018
  - "DRAM Refresh Mechanisms, Penalties, and Trade-offs" Bhati, Chang, Chishti, Lu and Jacob. IEEE transactions on Computers, 2016.



Midterm Review

Will be given online/remote during class time. Open notes/book, please no collaborating with other people.

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

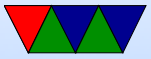


more random/ unpredictable behavior with lots of conditionals; floating point are often regular looped strides over large arrays or data sets)

- Be familiar with concept of skid.

2. Power
   - Know the CMOS Power equation (V^2, f)
   - Energy, Energy Delay, Energy Delay Squared
   - Idle Power Question

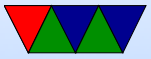
3. Branch Prediction
   - Static vs Dynamic



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

4. Cache

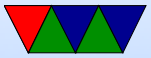
- Given some parameters (size, way, blocksize, addr space) be able to calculate number of bits in tag, index, and offset.
- Know why caches are used, that they exploit temporal and spatial locality, and know the tradeoffs (speed vs nondeterminism)



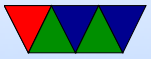
- Be at least familiar with the types of cache misses (cold, conflict, capacity)
- Know difference between writeback and write-through
- Be able to work a few simple steps in a cache example (like in HW#5)

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

6. Virtual Memory
   - General concept of VM

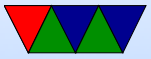


- Benefits of VM?
  Memory Protection, each program has own address space, allows having more memory than physical memory, demand paging, copy-on-write for fork, less memory fragmentation, etc.
- Why is TLB behavior important?
  Depending on cache config:
  worst case: (VIVT) every memory access looked up in TLB best case: (PIPT) every cache miss looked up in TLB



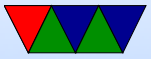
Memory Packaging Question

- DIMM – dual inline memory module
- Why dual? Replaced SIMMs
- SIMM had pins on both side but just duplicated signal
- SIMM also 32-bit, when modern systems moved to 64-bit bus (P5 pentium) you needed to have SIMMs in pairs
- DIMMs 64-bit memory bus and you only needed to add one module at a time



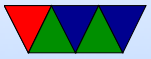
DDR

- transfer and both rising and falling edge of clock
- 2.5V
- Adds DLL to keep clocks in sync (but burns power)



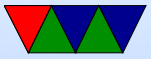
DDR2

- 2003
- 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



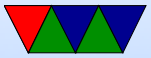
DDR3

- 2007
- internal doubles again
- Up to 6400MB/s, up to 16GB DIMMs.
- 1.5V or 1.35V



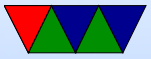
DDR4

- 2014
- I don't *think* things are doubled again, but it apparently somehow multiplexes for higher bandwidth
- 1.2V, 2.5V auxiliary wordline boost
- 1600 to 3200MHz. 2133MT/s
- Up to 64GB DIMMs
- parity on command/address busses, crc on data busses.
- Data bus inversion
- Pins closer toegher



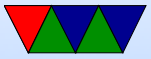
DDR5

- 2020
- Doubled bandwidth over DDR4
- 1.1V



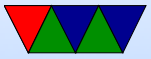
GDDR

- Despite similar name, not related to same DDR version
- 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



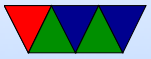
LPDDR

- Despite similar name, not related to same DDR version
- LP-DDR
- LP-DDR2 – low power states, 1.2V, different bus
- LP-DDR3 – higher data rate
- LP-DDR4 – change from 10-bit DDR to 6-bit SDR bus
- LP-DDR4X – I/o voltage 0.6V
- LP-DDR5 – (2019) 6.4Gbit/s/pin, differential clocks



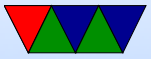
DDRL

- DDR3L - low voltage, 1.35V (not same as LPDDR3)
  DDR3U (ultra-low voltage) 1.25V
- DDR4L – does not exist?

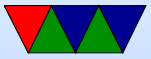


More obscure Memory Types

- 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,

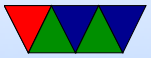


optimized for speed



Memory Latencies, Labeling

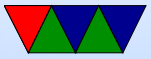
- DDR400 = 3200MB/s max, so PC3200
- DDR2-800 = 6400MB/s max, so PC2-6400
- DDR2 5-5-5-15
  - C_L CAS latency
  - T_{RCD} row address to column address delay
  - T_{RP} row precharge time
  - T_{RAS} row active time
  - CMD (optional), command time
- DDR3 7-7-7-20 (DDR3-1066) and 8-8-8-24 (DDR3-1333).



Memory Parameters

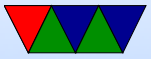
You might be able to set this in BIOS

- 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



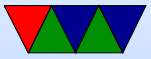
ECC Memory

- There's debate about how many errors can happen, anywhere from 10^{-10} error/bit*h (roughly one bit error per hour per gigabyte of memory) to 10^{-17} error/bit*h (roughly one bit error per millennium per gigabyte of memory
- Google did a study and they found more toward the high end
- Would you notice if you had a bit flipped?
- Scrubbing – only notice a flip once you read out a value



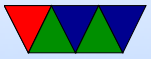
Registered Memory

- Registered vs Unregistered
- Registered has a buffer on board. More expensive but can have more DIMMs on a channel
- Registered may be slower (if it buffers for a cycle)
- Registered uses more power, and might be faster (even with the extra latency)
- RDIMM/UDIMM



Bandwidth/Latency Issues

- Is RAM truly random access these days?
- No, burst speed fast, random speed not.
- Is that a problem? How is RAM accessed these days? Mostly filling cache lines?



Memory Controller

- Do we even want full random access to memory?
  Should we just pass on CPU mem requests unchanged?
- What might have higher priority?
  Are some accesses more important? (regular vs prefetches)
- Why might re-ordering the accesses help performance (it's bad to switch back and forth between two pages)
- Can you improve mem performance with a better mem controller?