ECE 571 – Advanced Microprocessor-Based Design Lecture 18

Vince Weaver

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

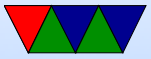
vincent.weaver@maine.edu

11 April 2017

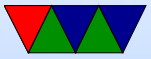
Announcements

- HW9 is a bit last minute
  Read DRAM measurement paper

- Don't forget project update due Thursday.
  (Extended to Friday)
  - Summary of your project
  - How is it going? Still time to change if needed.
  - Will you need any equipment from me? (Power measurement, accounts on machines)

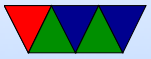


- Related work. Three/Six references related to your work. Ideally academic papers, though websites also OK if you can't find any other reference. Always supposed to do related work before starting project, people often do later.
- See the project handout for an example.
- Whether you are willing to present Tuesday (some extra bonus points)



DDR4 Speed and Timing

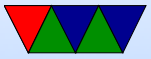
- Higher density, faster speed, lower voltage than DDR3
- 1.2V with 2.5V for “wordline boost” This might be why power measurement cards are harder to get (DDR3 was 1.5V)
- 16 internal banks, up to 8 ranks per DIMM
- Parity on command bus, CRC on data bus
- Data bus inversion? If more power/noise caused by



sending lots of 0s, you can set bit and then send them as 1s instead. New package, 288pins vs 240pins,

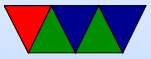
- pins are 0.85mm rather than 1.0mm Slightly curved edge connector so not trying to force all in at once

- Example: DDR4-2400R Memory clock: 300MHz, I/O bus clock 1200MHz, Data rate 2400MT/s, PC4-2400, 19200MB/s (8B or 64 bits per transaction)
  CAS latency around 13ns



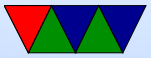
HBM2 RAM

- High bandwidth memory
- 3d-stacked RAM, stacked right on top of CPU
- In newer GPUs, AMD and NVIDIA. HBM2 in new Nvidia Pascal Tesla P100



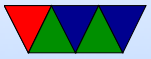
Types of NVRAM

- Core Memory
  - Old days, tiny ferrite cores on wire
  - Low density
- MASK ROM/EPROM/EEPROM
- Battery backed (CMOS) RAM
- FeRAM/Magnetoram – store in magnetic field
- Flash NAND/NOR



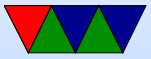
- Only so many write cycles (thousands) as opposed to billions+ for DRAM
- High power to erase
- Often have to erase in large blocks, not bit by bit
- Wear leveling

- Phase change RAM.
  - chalcogenide glass – used in CD-Rs
  - 100ns (compared to 2ns of DRAM) latency
  - heating element change from amorphous (high resistance, 0) to crystaline (low resistance, 1)



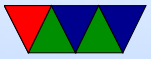
- temp sensitive, values lost when soldering to board (unlike flash)
- better than flash (takes .1ms to write, write whole blocks at once
- Newer methods might involve lasers and no phase change?
- Mapping into memory? No need to copy from disk?
- But also, unlike DRAM, a limit on how many times can be written.

- Memristors

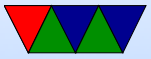


  - resistors, relationship between voltage and current
  - capacitors, relationship between voltage and charge
  - inductors, relationship between current and magnetic flux
  - memristor, relationship between charge and magnetic flux; "remembers" the current that last flowed through it
  - Lot of debate about whether possible. HP working on memristor based NVRAM

- Intel/Micron Optane/3D-Xpoint
  - Faster than flash, more dense than DRAM

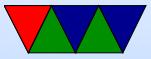


- special slot on motherboard
- 3D grid, not every bit needs a transistor so can be 4x denser than DRAM. Bit addressable.
- Intel very mysterious about exactly how it works



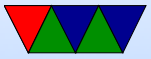
NVRAM Operating System Challenges

- How do you treat it? Like disk? Like RAM?
- Do you still need RAM? What happens when OS crashes?
- Problem with treating like disk is the OS by default caches/ copies disk pages to RAM which is not necessary if the data is already mapped into address space



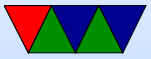
Why not have large SRAM

- 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

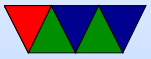


Saving Power/Energy with RAM

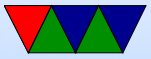
- AVATAR: A Variable retention time aware refresh for DRAM systems by Qureshi et al.
  - JEDEC standard: cell must have 64ms retention time
  - Why refresh bad? Block memory, preventing read/write requests
  - Consume energy (6,28,35)
  - The bigger DRAM gets, more refresh needed
  - predict that in 64Gb chips 50% of Energy will be in refresh



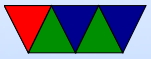
- Multi-rate refresh possible – detect which cells need more and refresh them more often (can be a 4-8x difference)
- VRT (variable retention rate) a problem. Some cells switch back and forth between. So when you probe it might check fine, but then fail later.
- They find that addition of cells stabilized to one new cell/15 mins over time
- Use ECC to catch these errors, though relying on ECC in this case can lead to uncorrectable error every 6 months



- They propose using ECC to adjust the VRT at runtime based on errors that are found
- They find on a 64Gb chip improves perf by 35% and **Energy-Delay** by 55%
- "Refresh-wall"
- Memory controller keeps track of this info
- VRT first reported in 1987. Fluctuations in GIDL (gate-induced drain leakage) presence of "trap" near the gate region
- Intel and Samsung say VRT one of biggest challenge in scaling DRAM

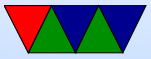


- VRT not necessarily bad – can cause retention to get better!
- Test – use FPGA to talk to 24 different DRAM chips, at controlled temperatures.
  Why do they use an FPGA?
- Actually it's just 3 chips from different vendors, each with 8 chips (for 24)
- Look into ECC. Soft-error rate is 200-5000 FIT/Mbit. Every 3-75 hours for 8GB DIMM. Soft errors happen 54x-2700x lower rate than VRT
- Downside of ECC ... have to scrub memory to check



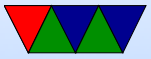
for errors. Also has energy/perf overhead. Energy to refresh DIMM 1.1mJ, energy to scrub 161mJ (150x) but if you scrub every 15 minutes it's a win.

– Use memory system simulator USIMM



Rowhammer

- Been observed for years, adjacent rows discharging can affect nearby rows
- Particularly bad in DDR3 from 2012-2013
- Accessing same row over and over can make voltage fluctuations in nearby rows, causing faster leakage than normal
- Mitigations? Refresh more often? ECC? Refresh nearby lines if a lot of row hammering going on?



- Can cause exploit. Google NaCl disable “cflush” exploit (need to force access to row)
- Can also trigger just with lots of cache misses
- If you can flip bits of kernel/trusted pointers to point to something you control, then you win.