ECE 571 – Advanced Microprocessor-Based Design Lecture 17

Vince Weaver

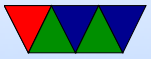
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

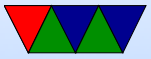
3 April 2018

Announcements

- HW8 is readings

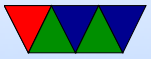


More DRAM



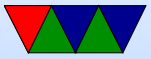
ECC Memory

- 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



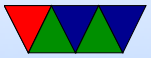
Registered Memory

- 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)
- RDIMM/UDIMM



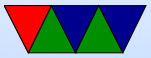
Bandwidth/Latency Issues

- Truly random access? No, burst speed fast, random speed not.
- Is that a problem? Mostly filling cache lines?



Memory Controller

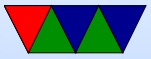
- Can we have full random access to memory? Why not just pass on CPU mem requests unchanged?
- What might have higher priority?
- Why might re-ordering the accesses help performance (back and forth between two pages)



Reducing Refresh

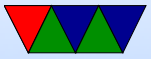
- DRAM Refresh Mechanisms, Penalties, and Trade-Offs by Bhati et al.
- Refresh hurts performance:
  - Memory controller stalls access to memory being refreshed
  - Refresh takes energy (read/write)

    On 32Gb device, up to 20% of energy consumption and 30% of performance

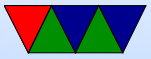


Async vs Sync Refresh

- Traditional refresh rates
  - Async Standard (15.6us)
  - Async Extended (125us)
  - SDRAM - depends on temperature, 7.8us normal temps (less than 85C) 3.9us above
- Traditional mechanism
  - Distributed, spread throughout the time
  - Burst, do it all at once (not SDRAM, just old ASYNC or LPDDR)

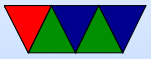


- Auto-refresh
  - ALso CAS-Before RAS refresh
  - No need to send row, RAM has a counter and will walk the next row on each CBR command
  - Modern RAM might do multiple rows
- Hidden refresh – refresh the row you are reading? Not implemented SDRAM

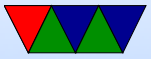


SDRAM Refresh

- Autorefresh (AR)
  - Device brought idle by precharging, then send AR (autorefresh)
  - Has a counter that keeps track of which row it is on, updates on each AR
  - The memory controller needs to send proper number of AR requests
  - LPDDR is a bit more complicated
  - Takes power, as all of SDRAM active while refreshing

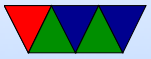


- Self-Refresh (SR)
  - Low-power mode
  - All external access turned off, clocks off, etc.
  - Has simple analog timer that generates clock for sending refresh pulses
  - Takes a few cycles to come out of SR mode
  - LPDDR has extra low-power features in SR mode
    Temperature compensated self-refresh (temp sensor)
    Partial-array self refresh (PASR), only refresh part of memory



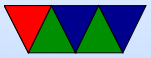
Refresh Timings

- Most SDRAM have 32 or 64ms retention time (tREFW)
- One AR command should issue in interval time (tREFI)
- A DDR3 with tREFI of 7.8us and tREFW of 64ms then 8192 refreshes
- Spec allows delaying refreshes if memory is busy



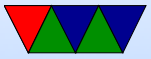
DRAM Retention Time

- Varies per-process, per chip
- Some chips over 1s, but have to handle worst-case scenario

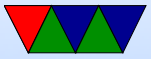


What can be done to improve refresh behavior

- Can you only refresh RAM being used? How do we know if values no longer important? free()? trim() command sort of like on flash drives?
- Probe chip at boot to see what actual retention time is, only refresh at that rate? Does chip behavior change while up?

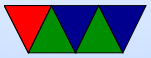


Advanced/Recent DRAM Developments



DDR4 Speed and Timing

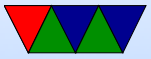
- Higher density, faster speed, lower voltage than DDR3
- 1.2V with 2.5V for "wordline boost" This might be why power measurement cards are harder to get (DDR3 was 1.5V)
- 16 internal banks, up to 8 ranks per DIMM
- Parity on command bus, CRC on data bus
- Data bus inversion? If more power/noise caused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



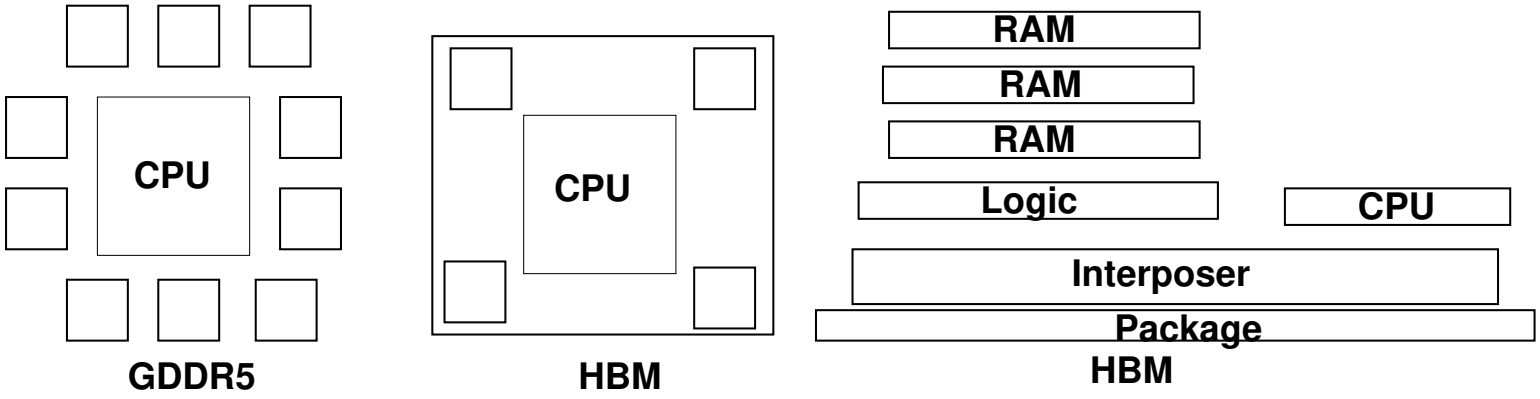
sending lots of 0s, you can set bit and then send them as 1s instead. New package, 288pins vs 240pins,

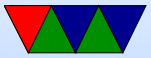
- pins are 0.85mm rather than 1.0mm Slightly curved edge connector so not trying to force all in at once

- 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



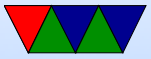
HBM RAM


CPU
GDDR5
CPU
HBM
RAM
RAM
RAM
Logic
CPU
Interposer
Package
HBM

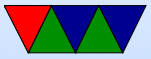


HBM/HBM2 RAM

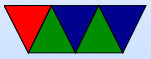
- HBM
  - High bandwidth memory
  - 3d-stacked RAM, stacked right on top of CPU
  - Silicon through VIA
  - Higher bandwidth, two 128-bit channels per die
  - 4096 bit wide bus compared to GDDR5 where you might have 32-bit channel times 16 chips for 512 bit
- HBM2
  - Eight dies per stack, up to 2GT/s



- HBM3/HBM4
  - Specified, doesn't exist yet? HPC?
- In newer GPUs, AMD and NVIDIA. HBM2 in new Nvidia Pascal Tesla P100

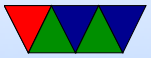


Future



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

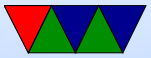


  - Only so many write cycles (thousands) as opposed to billions+ for DRAM
  - High power to erase
  - Often have to erase in large blocks, not bit by bit
  - Wear leveling

- Phase change RAM (see below)

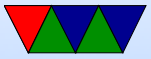
- Memristors (see below)

- Intel/Micron Optane/3D-Xpoint (see below)

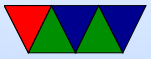


Phase Change RAM

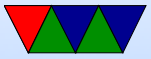
- Material
  - bit of material can be crystalline or amorphous
  - resistance is different based on which
  - need a heater to change shape
  - needs a lot of current to change phase
  - chalcogenide glass – used in CD-Rs
  - heating element change from amorphous (high resistance, 0) to crystalline (low resistance, 1)
  - temp sensitive, values lost when soldering to board



(unlike flash)
  - Newer methods might involve lasers and no phase change?
- Features
  - Faster write performance than flash (slower than DRAM)
  - Can change individual bits (flash need to erase in blocks)
  - can potentially store more than one bit per cell
  - better than flash (takes .1ms to write, write whole blocks at once

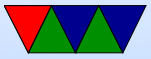


- 100ns (compared to 2ns of DRAM) latency
- Longevity
  - Flash wears out after 5000 writes, PCM millions
  - Flash fades over time. Phase change lasts longer as long as it doesn't get too hot.
  - But also, unlike DRAM, a limit on how many times can be written.



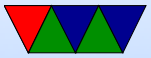
Memristors

- 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



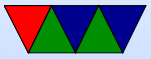
Intel/Micron Optane/3D-Xpoint/QuantX

- 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
  ReRAM (store in changed resistance) but is it phase-change?

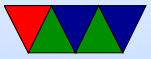


NVRAM Operating System Challenges

- 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
- Challenges: Mapping into memory? No need to copy from disk?

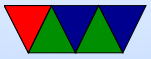


- Problems with NVRAM: caches.
- Memory is there when reboot like it was, but things in caches lost.
- So like with disks, if the cache and memory don't match you're going to have problems trying to pick up the pieces.



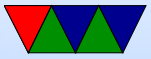
Why not have large SRAM

- 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

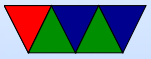


Saving Power/Energy with RAM

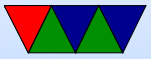
- AVATAR: A Variable retention time aware refresh for DRAM systems by Qureshi et al.
  - JEDEC standard: cell must have 64ms retention time
  - Why refresh bad? Block memory, preventing read/write requests
  - Consume energy (6,28,35)
  - The bigger DRAM gets, more refresh needed
  - predict that in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



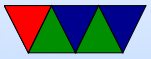
- Multi-rate refresh possible – detect which cells need more and refresh them more often (can be a 4-8x difference)
- VRT (variable retention rate) a problem. Some cells switch back and forth between. So when you probe it might check fine, but then fail later.
- They find that addition of cells stabilized 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

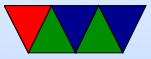


- VRT not necessarily bad – can cause retention to get better!
- Test – use FPGA to talk to 24 different DRAM chips, at controlled temperatures.
  Why do they use an FPGA?
- Actually it's just 3 chips from different vendors, each with 8 chips (for 24)
- Look into ECC. Soft-error rate is 200-5000 FIT/Mbit. 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



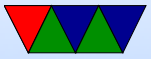
for errors. Also has energy/perf overhead. Energy to refresh DIMM 1.1mJ, energy to scrub 161mJ (150x) but if you scrub every 15 minutes it's a win.

– Use memory system simulator USIMM



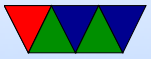
Cryogenic Memory

- Dip DIMMS in liquid nitrogen
- Low power? Faster? Interface with quantum circuits?



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.