ECE 571 – Advanced Microprocessor-Based Design Lecture 18

Vince Weaver

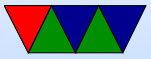
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

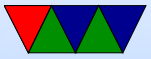
3 November 2022

Announcements

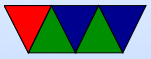
- Homework #7 Due
- HW#8 will be sent out soon
- Useful readings:
  - "A performance and power comparison of modern high-speed DRAM arch" from MEMSYS 2018
  - "DRAM Refresh Mechanisms, Penalties, and Trade-offs" Bhati, Chang, Chishti, Lu and Jacob. IEEE transactions on Computers, 2016.



- https://semiengineering.com/dram-thermal-issue

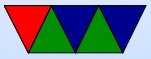


HW#6 Review

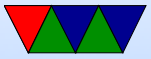


HW#6 – Intel Prefetch Background

- Intel/x86 sw prefetch
  - prefetcht0 – prefetch all levels
  - prefetcht1 – prefetch 2nd and 3rd level
  - prefetcht2 – same as prefetcht1
  - prefetchnta – non-temporal (when would you use this)?
  - prefetch – (3dnow) L1
  - prefetchw – (3dnow) prefetch, plan to write (MESI)
  - movnti – non-temporal move (don't cache)
- Intel hw prefetch



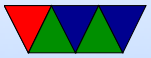
- Can disable all individually (chicken switch?)
- DCU + IP – fetch to L1
- Spatial (next line) – L2
- Stream – L3 (and L2 if L2 not too busy)



HW#6 – SW Prefetch Numbers

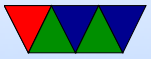
- objdump --disassemble-all ./bzip2 | grep prefetch | wc -l
  0 of them
  objdump --disassemble-all ./bzip2.swprefetch | grep prefetch
  ./bzip2.swprefetch: file format elf64-x86-64
  401397: 0f 18 0b prefetcht0 (%rbx)
  401e8c: 0f 18 88 a0 00 00 00 prefetcht0 0xa0(%rax)
  ... * 5 lines

- objdump --disassemble-all ./equake_l | grep prefetc
  (nothing)
  objdump --disassemble-all ./equake_l.swprefetch | grep prefetch
  ./equake_l.swprefetch: file format elf64-x86-64
  40192a: 0f 18 0a prefetcht0 (%rdx)
  401b8d: 0f 18 4d 48 prefetcht0 0x48(%rbp)
  401be8: 0f 18 4d 48 prefetcht0 0x48(%rbp)

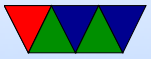


```
  402036:       0f 18 09                prefetcht0 (%rcx)
  40479a:       0f 18 4d 00             prefetcht0 0x0(%rbp)
  4047a9:       0f 18 0b                prefetcht0 (%rbx)
11 of them
```

- C library has 293 sw prefetch instructions (most likely in inline assembly for memcpy and the like?)
- Automated by C compiler was all prefetcht0
- Glibc hand-optimized had all cominations, including prefetchnta (for memset or similar?)
- Can use `addr2line` to try to find out where the compiler



is inserting SW prefetch, though for equake didn't seem to work



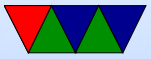
HW#6 – Results

- BZIP

		l2-cache-misses	prefetches	time
1a:	bzip2:	36%	166M	3.8s
2a:	SW prefetch:	34%	170M	3.7s
5a:	HWdisable	45%	63k	4.0s
5a:	HWdisable+Sw	45%	60k	4.0s

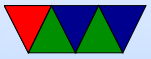
- Equake

		l2-cache-misses	prefetches	time
3a:	equake_l:	20%	49B	27s
4a:	equale_l swpref	20%	49B	27s
5a:	hwdisable	70%	9.2M	63s
5a:	hwdisable swpref	70%	9.1M	59s

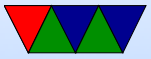


- Summary: disabling prefetch hurt, dramatically so on equake.
  Unclear what exactly the prefetch perf counter is measuring
  Enabling SW prefetch does not seem to do much, even with HW prefetch disabled.

- Why? Lots of possible reasons. compiler bug. hardware bug. hardware engineers not enable SW prefetch (is it incorrect to ignore?) other.



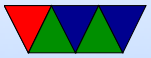
More on Refresh



Reducing Refresh

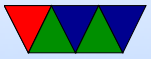
- DRAM Refresh Mechanisms, Penalties, and Trade-Offs by Bhati et al.
- Refresh hurts performance:
  - Memory controller stalls access to memory being refreshed
  - Refresh takes energy (read/write)

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

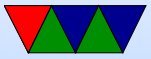


Async vs Sync Refresh

- 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)

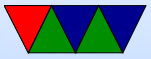


- Auto-refresh
  - Also CAS-Before RAS refresh
  - No need to send row, RAM has a counter and will walk the next row on each CBR command
  - Modern RAM might do multiple rows
- Hidden refresh – refresh the row you are reading? Not implemented SDRAM

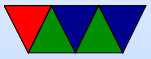


SDRAM Refresh

- 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

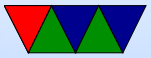


- Self-Refresh (SR)
  - Low-power mode
  - All external access turned off, clocks off, etc.
  - Has simple analog timer that generates clock for sending refresh pulses
  - Takes a few cycles to come out of SR mode
  - LPDDR has extra low-power features in SR mode
    Temperature compensated self-refresh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



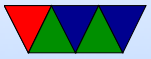
Refresh Timings

- 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



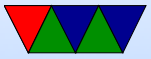
DRAM Retention Time

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



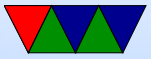
What can be done to improve refresh behavior

- 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?

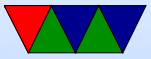


Refresh-related Security Issues – Rowhammer

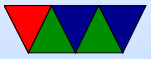
- Been observed for years, adjacent rows discharging can affect nearby rows
- Particularly bad in DDR3 from 2012-2013
- Accessing same row over and over can make voltage fluctuations in nearby rows, causing faster leakage than normal



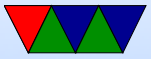
- Mitigations? Refresh more often? ECC? Refresh nearby lines if a lot of row hammering going on?
- Can cause exploit. Google NaCl disable "cflush" exploit (need to force access to row)
- Can also trigger just with lots of cache misses
- If you can flip bits of kernel/trusted pointers to point to something you control, then you win.



Advanced/Recent DRAM Developments

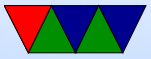


Future



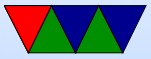
CXL Interconnect

- Memory attached RAM?
- https://www.theregister.com/2022/08/04/cxl_time
  now/



NVRAM

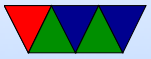
- Core Memory
  - Old days, tiny ferrite cores on wire
  - Low density
- MASK ROM/EPROM/EEPROM
- Battery backed (CMOS) RAM
- FeRAM/Magnetoram – store in magnetic field



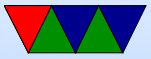
- MRAM (magnetic RAM), Spin-transfer-torque (STT-MRAM)

- 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

- Millipede memory, tiny bumps, MEMS devices to read

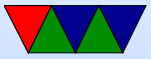


- Phase change RAM (see below)
- Memristors (see below)
- Intel/Micron Optane/3D-Xpoint (see below)



Phase Change RAM

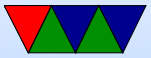
- Material
  - bit of material can be crystalline or amorphous
  - resistance is different based on which
  - need a heater to change shape
  - needs a lot of current to change phase
  - chalcogenide glass – used in CD-Rs
  - heating element change from amorphous (high resistance, 0) to crystalline (low resistance, 1)
  - Amorphous if you heat and quench, crystal if cook a



while

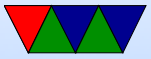
- temp sensitive, values lost when soldering to board (unlike flash)
- Newer methods might involve lasers and no phase change?

- 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



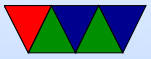
- better than flash (takes .1ms to write, write whole blocks at once
- 100ns (compared to 2ns of DRAM) latency

- 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.
- Can you buy phase change ram?
  - Micron sold from 2012-2014? Not much demand



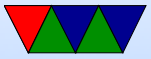
Memristors

- 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



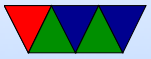
Intel/Micron Optane/3D-Xpoint/QuantX

- Note: Intel finally cancelled Optane in 2022
- 3D-Crosspoint (intel) QuantX (Micron)
- Faster than flash, more dense than DRAM
- Can get it in an SSD (so no special hardware needed)
- Also as special slot on motherboard (or even DIMM)
- 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-



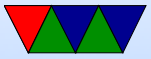
change?

- Intel denies everything
- ReRAM works by having a dielectric layer and blasting channels through it.
- Can you buy Optane?
  April 24th 2019? Special M.2 slot on Gen7 (Kaby lake? motherboards)
  For now, 16GB and 32GB modules, using like a cache of your hard disk.

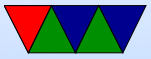


NVRAM Operating System Challenges

- 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?

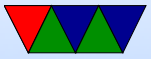


- Problems with NVRAM: caches.
- Memory is there when reboot like it was, but things in caches lost.
- So like with disks, if the cache and memory don't match you're going to have problems trying to pick up the pieces.

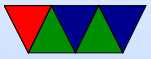


Saving Power/Energy with RAM

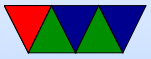
- AVATAR: A Variable retention time aware refresh for DRAM systems by Qureshi et al.
  - JEDEC standard: cell must have 64ms retention time
  - Why refresh bad? Block memory, preventing read/write requests
  - Consume energy (6,28,35)
  - The bigger DRAM gets, more refresh needed
  - predict that in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



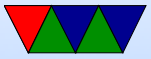
- Multi-rate refresh possible – detect which cells need more and refresh them more often (can be a 4-8x difference)
- VRT (variable retention rate) a problem. Some cells switch back and forth between. So when you probe it might check fine, but then fail later.
- They find that addition of cells stabilized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

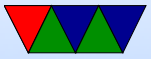


- VRT not necessarily bad – can cause retention to get better!
- Test – use FPGA to talk to 24 different DRAM chips, at controlled temperatures.
  Why do they use an FPGA?
- Actually it's just 3 chips from different vendors, each with 8 chips (for 24)
- Look into ECC. Soft-error rate is 200-5000 FIT/Mbit.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



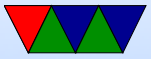
for errors. Also has energy/perf overhead. Energy to refresh DIMM 1.1mJ, energy to scrub 161mJ (150x) but if you scrub every 15 minutes it's a win.

– Use memory system simulator USIMM

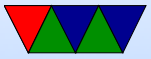


Cryogenic Memory

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

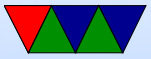


DRAM – Power Saving



DRAM – Mobile DRAM

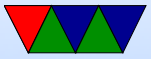
- From Micron: "TN-46-12: Mobile DRAM Power-Saving Features", 2009
- Temperature-Compensated Self Refresh (TCSR) – Auto adjust refresh timings based on temperature
- Partial Array Self Refresh (PASR) – only refresh parts of RAM that have data in them
- Deep Power Down (DPD) – enable turning off the voltage generators when maintaining DRAM not needed
- Has equations for estimating power usage



DRAM – Elsewhere

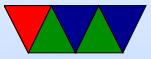
- Tom's Hardware. 2010. "How Much Power Does Low-Voltage DDR3 Memory Really Save?" Using low-voltage (1.25 or 1.35 rather than 1.5) DDR3 DRAM can reduce power by 0.5-1W. Slower performance settings, but not really noticeable.

- Linus Torvalds Rant from 2007: DRAM Energy not a prime concern. Just don't use FBDIMMs if you want low-power.



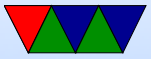
DRAM – Recent Academic

- "Rethinking DRAM Power Modes for Energy Proportionality", Malladi et al, Micro 2012.
  - DRAM spends lots of time idle, but latency is so high for wakeup it cannot utilize powerdown modes
  - Reference 25% of data-center energy usage is DRAM?
  - Use LPDDR2 trades bandwidth for efficiency
  - Current modes involve turning off DLLs (Delay-locked loops?) which are slow to turn on again, 700ns+
  - some background on DRAM operation

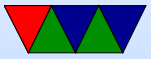


- Low-power mode sounds good, but then it takes 512 memory cycles of power to re-start (a lot of energy)
- Propose MemBLAZE. Moves clock generation out of DIMM and into memory controller, allowing fast wakeup

- "Towards Energy-Proportional Datacenter Memory with Mobile DRAM", Malladi et al, ISCA 2012.
  - Look at using LPDDR2 in servers rather than DDR3.
  - DDR3 often in "Active-idle" as many workloads do not allow sleep.



- "A Predictor-based Power-Saving Policy for DRAM Memories", Thomas et al, EuroMicro 2012.
  - Use a history based predictor to pick when to powerdown.
  - Say up to 20% of mobile devices and 25% of data center is DRAM
- "Rethinking DRAM Design and Organization for Energy-Constrained Multi-Cores", Udipi et al., ISCA 2010
  - DRAMs "overfetch" which hurts energy
- "A Comprehensive Approach to DRAM Power



Management", Hur and Lin, HPCA2008.

- Throttling and Power Shifting – slowing down to fit in power budget
- Put DRAMs in low power mode – available commercially but no one seems to use this yet
- Simulate for Power5 and DDR2-533
- Modify the memory controller