

ECE 571 – Advanced Microprocessor-Based Design Lecture 17

Vince Weaver

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

`vincent.weaver@maine.edu`

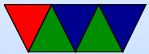
5 November 2019

Announcements

- Return midterm.



From Last Time



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).



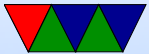
Memory Parameters

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



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)
- 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?



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 (t_{REFW})
- One AR command should issue in interval time (t_{REFI})
- A DDR3 with t_{REFI} of 7.8us and t_{REFW} 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

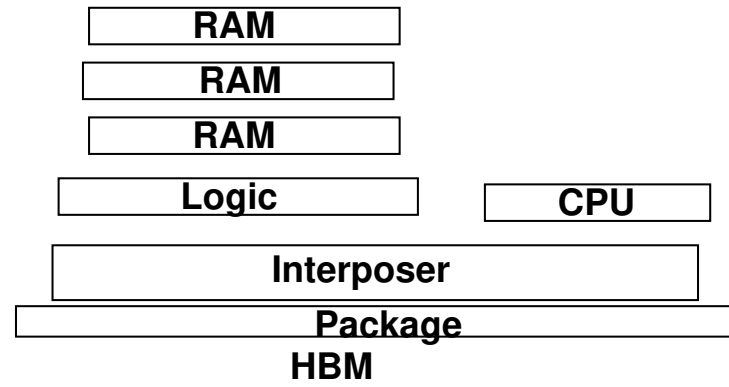
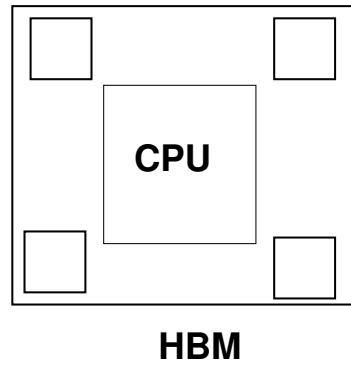
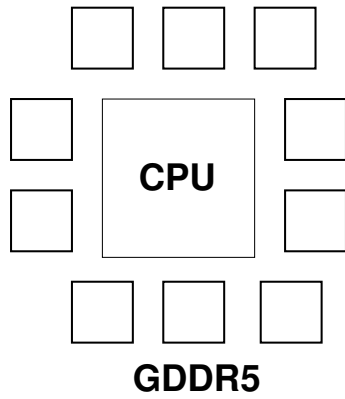


sending lots of 0s, you can set bit and then send them as 1s instead. New package, 288pins vs 240pins,

- 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



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

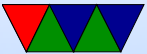


Hybrid Memory Cube

- Samsung and Micron
- vaguely similar to HBM
- discontinued in 2018



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



- 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
- 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

- 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.



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.

