DRAM
Overview & Devices

Reference: "Memory Systems: Cache, DRAM, Disk
Bruce Jacob, Spencer Ng, & David Wang
Today's material & any uncredited diagram came from chapters 7 & 8

Intended CS7810 Plan

• 5 weeks on memory systems
  • 3 weeks on DRAM
    » overview & devices
    » system signalling
    » system organization & access protocols
    » DRAM memory controllers
    » FBDIMM and BoB approaches
    » memory system design Analysis
  • 1 week on disk
    » physical and data layers
    » design trade-offs, interfaces, & futures
• 1 week on NVRAM or URAM new technologies
  » got moved up to the first week
    • FLASH - the current dominant technology
    • NVRAM/URAM options on the horizon
Memory & Storage are Different

• Memory hierarchy (7810 focus now)
  ▪ multiple levels of cache
  ▪ main memory
    » for now it’s DRAM
  ▪ disk

• Storage (additional component next year?)
  ▪ logical view: file system and backups
  ▪ physical view: disks & tape for now
  ▪ operational view: huge complexity
    » reliability, availability, serviceability (a.k.a. RAS)
    » migration issues, cabling, cooling & power, interconnect
    » today's datacenters often have more processors in the storage subsystem than in the compute subsystem

• What should Google care more about: storage or compute?
  ▪ food for thought

Key Item to Remember

• It is easy to predict SRAM behavior
  ▪ even though discrete DRAM may well disappear in this decade
    » since cache buses (BSBs) are almost extinct now

• Hard to predict DRAM behavior
  ▪ probabilistic resource availability
  ▪ performance depends on controller and device model
    » small controller differences show up as big performance differences

• Disk performance is probabilistic as well

• Plus
  ▪ lots of intermediate buffers, prefetch, ... issues as well
Typical PC

Orthogonal address to save pins & cost

Simplified DRAM

Sense amps now combined with row buffer
It’s All about Mats

- DRAM devices come in several flavors
  - interface & speed: we’ll deal with these later
  - width
    - x4 & x8 are highest density die
      - used in price sensitive applications like PC’s
    - x16 & x32
      - higher per bit cost used in high performance systems
- DRAM chip = lot’s of memory arrays (mats)
  - mats operate under several regimes
    - unison
      - each access targets one bit/mat
        - x4 accesses 4 mats
    - independent
      - mats organized as subsets to create banks
      - concurrent bank access is the idea
      - intra-bank mats operate in unison
    - interleaved banks

Mat & Width Organization
Slow Mat Problem

- Mat access is slow
  - high-C word and bit lines
    » bigger = slower
    » C for wire is linear in length at same width
    » Cgate is linear with size of row or column in the mat
- Interleave to speed up
  - mid-60’s hack used on IBM 360/91 and Seymour’s CDC 6600
    » essentially a form of pipelining
  - If interface is n times faster than mat latency interleave n banks
    » should be able to make things arbitrarily fast
      » in theory yes - in practice no
        - constraints: jitter, signal integrity, power
  - multiple on-die banks
    » may be internally or externally controlled

Ranks & Banks vs. DRAMs & DIMMs
JEDEC Interface

address width depends on DRAM capacity
control: RAS, CAS, Oenable, CLKenable, etc.

Chip select goes to every DRAM in a rank
Separate select per rank - 2 per DIMM common

64 bits typical wider in high-end systems

See any problems on the horizon with this model?

Memory Controller Issues

- **DRAM control is tricky**
  - CPU prioritizes memory accesses
    - transaction requests send to Mem_Ctl
  - Mem_Ctl
    - translates transaction into the appropriately timed command sequence
      - transactions are different
        - open bank then it's just a CAS
        - no open bank then Activate, PRE, RAS, CAS
        - wrong open bank then write-back and then ACT, PRE, RAS, CAS
        - lots of timing issues
      - result: latency varies
        - often the command sequence can be stalled or even restarted
        - refresh controller always wins
    - now moving onto the CPU die
      - multi-core and multi-mem_ctl involves a lot of issues
**DRAM Evolution**

- **Not that important**
  - naming conventions vary by vendor to some extent
    » Clocked - treat DRAM as a really slow SRAM
    » Asynch DRAM - access and wait
      - still clocked but the timing provided by the command lines
    » Fast Page Mode
      - add latches to the sense amps to form row buffer
    » EDO
      - add latches to output drivers so data stays valid
    » P/BEDO
      - add counter to cycle through successive width sized nibbles
  - SDRAM - mid 90's - the bulk of the action now
    » clock now controls row select circuits as well
    » DDRx variants still SDRAM just higher bandwidth

**Simple SDRAM Timing**

Note: pipelining possibilities
DRAM Conundrum

• Cost/bit vs. Latency
  • widening memory gap
    » CPUs faster by 58%/yr
    » DRAM faster by 7%/yr - now going even flatter
    » multi-core makes this problem much worse
  • current industry trend
    » minimize cost through density improvements
    » so we’re stuck with long latency
• Focus on improving throughput
  • enter DDRx and Rambus

RDRAM Throughput Idea

• System interface solution rather than a fundamental technology
  • narrow, split request-response bus
    » addr, data, ctl, selects all mux’d on the same bus
    » uses the DDR model
    » initially 1 byte wide and ran at 250 MHz
      • limited concurrency so redesigned ==> concurrent RDRAM
  » C-RDRAM
    • looks a lot more like a JEDEC interface now
    • simplifies transaction scheduling
    • shares a row buffer between adjacent banks
      - limits open bank options but saves on cost

![Diagram](image)
Evolving RDRAM

- Widen data & address + pump up the clock

C-RDRAM

Other RAMBUS Wrinkles

- Packaging
  - pins on one side of die so chips can be inserted rather than just DIMMs
    - not clear if anyone bought into this
- Interesting dual clock timing model in the patent
  - usage is still TBD
- Variable request response latency
  - Mem_Ctl can specify
  - usage: variable packet sizes & higher utilization of the narrow bus
Mainstream Throughput Idea: DDRx

- **Use both clock edges**
  - DDR transfers 2 bits per cycle per lane
    - DDR2 transfers 4
    - DDRn transfers 2^n
    - signal integrity and power limit clock speeds
      - particularly on long FR4 wire traces
- **Also add source synchronous clocking - enter DQS**
  - timing variance creates synchronization issues
    - DDR device uses DLL/PLL to synch with Mem_Ctl master clock
      - note skew depends on where the DIMM sits in the chain
      - need to latch in the center of the data “eye”
  - other sources of timing uncertainty
    - manufacturing variation, temperature, Miller side-wall effect, trace length
    - delay proportional to RC
    - power proportional to CV^2

Optimizing for Latency

- **Virtual channel memory (VCDRAM)**
  - add SRAM cache for segments
    - manage by mem_ctlr
    - adds prefetch and restore commands to the mix
    - latency better if cache hit - worse on a miss

- **Enhanced SDRAM (ESDRAM)**
  - put latch before column mux rather than after as in EDO
    - expensive since it’s a row wide rather than a column wide
    - allows
      - overlap of row precharge without closing existing row
      - allows a write-around option which may be useful in write-back cache models
Optimizing for Latency (cont’d)

• MoSys 1T-SRAM (current low latency winner)
  • catchy name but it’s still 1T+1C
  • wraps SRAM interface around DRAM core
  • large number of small independent banks => lower latency
    » increased control circuitry => reduced density => increased cost
  • niche market in game systems
• Reduced Latency DRAM (RLDRAM)
  • has no DIMM specification
  • SRAM like interface Raddr and Caddr on different pins
• Fast Cycle DRAM (FCRAM)
  • breaks row into subarrays - smaller is faster
  • moves some Caddr bits to Raddr
    » does have a DIMM spec
    » faster clock and higher bandwidth - obvious limits on the horizon

Disturbing Trend

• DIMM capacity going up
  • process improvements yield more bits/die
• DRAM channel speed going up
  • DDRn
• # of DIMMs per channel going DOWN!!
  • SDR - 8 DIMMs/channel
  • DDR - 4 DIMMs/channel
  • DDR2 - 2 DIMMs/channel
  • DDR3 - 1 DIMM/channel and higher latency
    » isn’t this a lower bound?
    » adding channels is expensive in CPU pins
      • remember mem_ctl is on chip now and for good reason
• Why?
  » stub electronics problem on a JEDEC broadcast bus
  » gets worse if bus speed increases - it’s the di/dt thing
• Problem essence
  » not enough memory capacity per socket
  » huge server problem today
**Intel’s FB-DIMM Compromise**

Traditional (JEDEC) Organization  
Graphics-Card Organization  
Fully Buffered DIMM Organization

- Move to point to point signalling and add an ASIC
  - AGP already uses this tactic to run higher bus speeds

**FB-Dimm Problems?**

- There are many
  - daisy chain causes varied response time
  - bit lane retiming additional latency problem
- Already considered a 1-trick pony
- Enter BoB - Buffer on Board - the new Intel hack
  - use a tree rather than a daisy chain for 4x DDR3
  - BoB placement
    - motherboard or on a memory card riser
  - problem - another buffer stage in the memory hierarchy
    - OK if prefetch strategy is working for you
- AMD has/had? a similar variant
  - Socket 3 Memory Extender (G3MX) micro-buffer
    - effort now seems to have been cancelled
DRAM Systems Issues 1

- **Architecture and scaling**
  - DDRn causes $2^n$ prefetching
    - I/O side faster but mat side is wider
    - Implies wider cache lines
    - We know the issues involved
  - Timing fundamentally limited by signal integrity issues
    - Lots can be done here but impact is cost/bit increase

- **Pins vs. protocol**
  - Pin count has large cost adder
  - Use them more efficiently $\Rightarrow$ protocol change
    - JEDEC moves slowly

---

DRAM Systems Issues 2

- **Power and heat**
  - The biggest concern now and in the future most likely
    - Early DIMMs consumed about 1W
    - FB-DIMMs now at 10W

- **Servers**
  - Goal
    - 3x more channels and 6x more DIMMs per channel
  - Looks like 250 W per socket just for memory
    - Huge problem now
  - Definite time for a rethink
    - Problem
      - Industry momentum
      - Standards
      - DRAM commodity $\Rightarrow$ super low margins
        - Rethinking is a costly proposition
Slight Change of Focus

• Very brief device technology overview
  • background for what comes later
• Key issues
  • leaky devices
  • process differences
  • refresh requirements
  • how to build that pesky capacitor

64 Mbit FPM DRAM (4096x1024x16)
DRAM Cell

Logical View

Trench implementation now primarily used in eDRAM

stacked implementation mainstream DRAM processes

Leakage & Refresh

• Transistors are not ideal switches
  • leakage currents in DRAM processes are minimized
    » but not to 0
  • leakage currents increase as Tsize goes down
    » tricky balance of Vth, Vdd, and process
    » additional increase with temperature
  • industry target - refresh every 32 - 64 ms
**Folded vs. Open Bit-Line**

Folded: \(8F^2\) - 2 bit line per cell

6\(F^2\) version shipped by Micron using MIM (metal insulator metal C) in 2004

Open: \(6F^2\) - 1 bit line per cell

---

**Issues**

- **Open**
  - requires dummy array segments at mat edge
    - balance C characteristics of bit-line pairs
  - more noise susceptibility
  - combine to dilute the cell size advantage

- **Folded**
  - differential sense amps have better common-mode noise rejection properties
    - e.g. alpha particle or neutron spike shows up on both sides
  - current industry focus
    - new folding strategies show up regularly in circuits venues
Sense Amps

- Small stored charge requires high sensitive amps
  - use differential model
    - reference voltage precharged to half-way mark
    - then look at which way the charge goes to determine value
      - noise margins must exist and trick is to keep them small
      - problematic as devices shrink

- Roles
  - 1: basic sense value
  - 2: restore due to the destructive read
    - 2 variants in play
      - restore instantly or restore on row close
  - 3: act as a temporary storage element (row buffer)
    - how temporary depends on restore choice

Sense Amp Operation

- Diagram showing various stages of sense amp operation.
Decoders & Redundancy

- Defects occur and yields have to be high
  - rules of a low margin business
- Redundant rows, columns, and decoders
  - fuses are used to isolate defective components
  - appearance is of a fully functional mat
  - fuse set
    » burn in, test and then fuse set
Packaging, Performance, Cost

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>process (nm)</td>
<td>90</td>
<td>65</td>
<td>45</td>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td>CPU pin count</td>
<td>2263</td>
<td>3012</td>
<td>4009</td>
<td>5335</td>
<td>7100</td>
</tr>
<tr>
<td>cents/pin</td>
<td>1.88</td>
<td>1.61</td>
<td>1.68</td>
<td>1.44</td>
<td>1.22</td>
</tr>
</tbody>
</table>

| DRAM pin count | 48-160 | 48-160 | 62-208 | 81-270 | 105-351 |
| cents/pin     | 0.34-1.39 | 0.27-0.84 | 0.22-0.34 | 0.19-0.39 | 0.19-0.33 |

Pressure runs wild!!

DRAM vs. Logic Process
Hybrid Processes Coming

- IBM was the pioneer
  - start with logic process
  - add extra layers to create high-C DRAM cells
    » multiple oxide thicknesses
    » fast leaky transistors
    » slow less-leaky transistors
    » enables eDRAM
    » also helps with power issues
      » leakage is a big deal
      » only use fast transistors on the critical CPU path
      » use slow T's for non-critical path and memory blocks
- Current usage in transition
  - from high-performance SoC's to mainstream CPU