1. Memory technology & Hierarchy

RAM types

Advances in Computer Architecture
Andy D. Pimentel
“Memory wall” = divergence between CPU and RAM speed

We can increase bandwidth by introducing concurrency in memory access (e.g. through pipelining accesses)

- but this requires regular access patterns
- random accesses to main memory can cause severe performance degradation
Conflicting requirements in a memory system: we want both large and fast.

Electronic systems have higher latencies as they increase in size.

- Speed of light is approximately 1 ns for 30 cms.
- N.b. 1 ns is 3 clock cycles in a state of the art processor.
- Pins, wires, connectors etc. all add resistance and capacitance which delay signals significantly.

A memory hierarchy attempts to make a large slow memory appear fast by buffering data in smaller faster memories close to the processor.
Memory hierarchy - issues

- Can make memories faster but this requires more power
- Need to drive long wires on chip across memory array – to do this fast needs more power
- Memory performance is a compromise between power and performance
- (As is processor performance today)
RAM cell designs

DRAM cell

SRAM cell
Components - DRAM

- DRAM - is a very dense form of RAM - it is volatile
  - access is destructive & data must be re-written
  - charge also leaks from the capacitor which stores the data so data must be refreshed periodically ("dynamic" RAM)

- Typical DRAM chip characteristics
  - 256-1024 Mbit and 2-800MHz cycle
  - Uses two cycle Row/Column addressing (RAS/CAS)
  - two-stage access and requirement to rewrite data contribute to slow cycle time
  - but good design can help for regular accesses
DRAM - RAS/CAS addressing

Step 1: Row Address Select
Step 2: Column Address Select (select bit)

Refresh: read and write back a whole row
DRAM refresh

Voltage for 1
Threshold voltage
Voltage for 0

1 Written
Refreshed
Refreshed
Refreshed

Time

10s of ms before needing refresh cycle

0 Stored

Advances in Computer Architecture, Andy D. Pimentel
Components - SRAM

SRAM - is less dense but faster than DRAM

- uses four transistors to store data and two transistors to access it - access is non-destructive
- data is stable while the RAM has power connected ("static" RAM)

Typical SRAM chip characteristics

- ~64Mbit and 100MHz-1GHz cycle
  - although SRAM cycle times are similar to DRAM, SRAM is true random access memory
    DRAM can only read consecutive bits at the cycle rate, therefore DRAM has a much larger latency time
- SRAM is also used for memory on the processor chip
  - registers and cache both use SRAM technology

the smaller the memory the "faster" it operates: lower latency
New technology

Memristor - a circuit element in which the resistance is a function of the history of the current through the device.

Memristor theory was formulated and named by Leon Chua in 1971.

HP announced they teamed up with Hynix to produce a commercial product dubbed "ReRam" August 2010.
http://www.reuters.com/article/idUS254583059320100901

Has the potential to be dense and non-volatile.
**Bandwidth vs. Latency**

**Memory latency** is the time delay required to obtain a specific item of data.

This is larger in DRAM than in SRAM:
- SRAM can access any bit each cycle.
- DRAM is restricted to bits in the same row, CAS cycles.

**Memory Bandwidth** is the rate at which data can be accessed (e.g. bits per second):
- Bandwidth unit is normally $1/\text{cycle time}$.
- This rate can be improved by concurrent access.
Improving DRAM bandwidth

Using locality to get maximum bandwidth

- **One RAS multiple CAS e.g.**
  - Fast page mode DRAM
  - (xtended) (ata) (utput) RAMs

- **Burst-mode DRAMs**
  - Burst EDO RAM
  - (ynchronous)DRAM

By improving the interface

- **DDR SDRAM** and **RAMBUS**
Example - Synchronous DRAM

SDRAM changed the memory interface from asynchronous to synchronous and uses a form of pipelining – will return to this concept later

**DDR SDRAM** - double data rate uses transfers on both rising and falling clock edges
Different SDRAM technologies

Double Data Rate (DDR) SDRAM is advantageous for systems that require higher bandwidth than can be obtained using SDRAM. Basically, DDR SDRAM doubles the transfer rate without increasing the frequency of the memory clock. This section describes three generations of DDR SDRAM technology.

**DDR-1**

To develop the first generation of DDR SDRAM (DDR-1), designers made enhancements to the SDRAM core to increase the data rate. These enhancements include prefetching, double transition clocking, strobe-based data bus, and SSTL_2 low voltage signaling. At 400 MHz, DDR increases memory bandwidth to 3.2 GB/s, which is 400 percent more than original SDRAM.

**Prefetching**

In SDRAM, no bit is transferred from the memory cell array to the input/output (I/O) buffer or data queue (DQ). The I/O buffer releases one bit to the bus per pin and clock cycle (on the rising edge of the clock signal). To double the data rate, DDR SDRAM uses a technique called prefetching to transfer two bits from the memory cell array to the I/O buffer in two separate pipelines. Then the I/O buffer releases the bits in the order of the queue on the same output line. This is known as 2n-prefetch architecture because the two data bits are fetched from the memory cell array before they are released to the bus in a time multiplexed manner.
This is an interface improvement using a pipelined bus interface sometimes called a split-transaction.

- Bus comprises row and column address line + 18 bits of data
- 3 transactions on bus simultaneously (RAS/CAS/Data)
- High clock rate (400MHz) with data transfers on both edges

Note that neither technique (SDRAM or RAMBUS) can improve the latency to access a single item of data.
Flash memory

Source lines

Bit lines

Word lines

Control gate

Floating gate

Source

p substrate

n-
n+

Drain
Memory wall – solutions

The most common solution to the memory wall is to cache data

- requires locality of access or memory reuse
- compiler optimisations can help to localise data

Can also design banked memory systems to provide high bandwidth to random memory locations

- Some access patterns will still break the memory

Can design processors that tolerate high-latency memory accesses

- "don’t wait do something else"

- Requires concurrency in instruction execution
Summary

- Need for new dense + fast technology
- Most common solution: cache data
  - Requires locality of access, or memory reuse
- Design processors that tolerate high-latency memory accesses – “don’t wait do something else”
  - Trend: hardware multithreading in current and future chips
Summary

Limits of RAM components

Memory wall

Caches & hw multithreading

New components (?)

solutions

cause

problem