Computer Architecture
ELEC3441

Lecture 8 – Memory (1)

Dr. Hayden Kwok-Hay So

Department of Electrical and Electronic Engineering

CPU vs Memory

Memory

• store (sw, sh, sb)

CPU

• instruction fetch
• load (lw, lh, lb)

Processor-DRAM Gap (latency)

Requirements for an ideal system:
• Large capacity
• High performance
• Low latency
• High bandwidth

Reality:
• Capacity, Latency, Bandwidth often counteract each other

Solution:
• Use different types of memory to build a memory hierarchy that performs well on average

Memory Challenges

3GHz uP, 100ns latency → 300 cycles to get 1st data

~60%/year (2x every 1.5 yr)

~7%/year (2x every ~10 yr)

Gap growing 50%/yr


1 10 100 1000 10,000 100,000

Year

normalized performance

Processor

Memory

~60%/year (2x every 1.5 yr)

~7%/year (2x every ~10 yr)
Agenda
- Memory Technologies (Review)
  - SRAM, DRAM
- Memory Hierarchy
  - Cache
  - Virtual memory
- (optional) Advanced memory hierarchy techniques

Types of Memory
- ROM: read only memory
- RAM: random access memory
- Volatile: Content may disappear w/o power
- Non-Volatile: Content remains w/o power

Early Memory Examples
- Punched cards, From early 1700s through Jaquard Loom, Babbage, and then IBM
- Punched paper tape instruction stream in Harvard Mk 1
- Core memory, invent late 40s/early 50s at MIT
- DEC PDP-8/E Core Memory Board, 4K words x 12 bits, (1968)
Semiconductor Memory

- Semiconductor memory began to be competitive in early 1970s
  - Intel formed to exploit market for semiconductor memory
  - Early semiconductor memory was Static RAM (SRAM). SRAM cell internals similar to a latch (cross-coupled inverters).

- First commercial Dynamic RAM (DRAM) was Intel 1103
  - 1Kbit of storage on single chip
  - charge on a capacitor used to hold value

*Semiconductor memory quickly replaced core in ‘70s*

---

**DRAM**

- Dynamic Random Access Memory
- Data stored in a single capacitor
- Read/write access through 1 transistor
  - Sometimes referred as a 1T cell
- Read is destructive
  - Data stored in capacitor is lost after read
  - write back data after read
- Data lost over time due to charge leakage
  - refresh needed
  - Most modern DRAM have auto refresh capability

---

**One-Transistor Dynamic RAM [Dennard, IBM]**

- 1-T DRAM Cell
- Storage capacitor (FET gate, trench, stack)
- TIN top electrode ($V_{REF}$)
- Ta$_2$O$_x$ dielectric
- W bottom electrode
- Poly word line
- Access transistor
**Trench Capacitor**
- Maximize capacitance through vertical trench into the substrate

**DRAM Architecture**
- Bits stored in 2-dimensional arrays on chip
- Modern chips have around 4-8 logical banks on each chip
  - each logical bank physically implemented as many smaller arrays

**Read Operation**
1. Precharge bitlines to Vdd/2
2. Enable target word line
   - 1: V_bitline ↑
   - 0: V_bitline ↓
3. Sense Amps sense changes in bitline voltage
   - Latch results
4. Select desired column from output latch
DRAM Operation (1)

- 3 basic operations: Row Access, Column Access, Precharge

Row Access
- Select rows depending on address
- Each row can have up to 10s of Kb
- Sense amps sense very small change in bitline level
- Voltage change is small because charge stored in a small cap is shared with long bitline
- Sense amps restore full swing level + restore data in cell (refresh)

DRAM Operation (2)

- Column Access:
  - Select the desired bits out of the entire row
  - Usually select just a small portion (4, 8, 16, or 32 bits)
  - On Read: send the data out of package
  - On Write:
    - write the design data in the sense amp latches
    - let sense amp “refresh” array with new data + original data form unchanged bits

Precharge:
- Precharge bitline for next operation

Performance

- Latency vs. Bandwidth
- Each step (Precharge, RAS, CAS) takes 15-20ns in modern DRAM
- Getting first bit takes very long ➔ high latency
- But since entire row of bits are sensed, subsequent column data can be send out of package at high bandwidth
  - Various burst mode access
  - Modern SDRAM (Synchronous DRAM) has very high bandwidth output to send data out as soon as possible
  - e.g. Double Data Rate (DDR) interface
SRAM Overview

- Static Random Access Memory
- Data stored is persist as long as power is supplied
- Design mostly based on standard digital circuit technology
  - e.g. No exotic capacitors like DRAM
- Simple read/write interface
  - no complex command sequence
- Challenge:
  - Capacity

SRAM

- 1971 state of the art.
- Intel 2102, a 1kb, 1 MHz static RAM chip with 6000 nFETs transistors in a 10 μm process.

SRAM Cell Design

- The most common SRAM cell follows the design of a simple cross-coupled inverter

```
Q
Q'
```

- READ: simply take value from Q or Q', non-destructive
- Need additional logic to write into the cross-coupled inverter

SRAM Write

- Write data by “forcing” values through the 2 access transistors
  - Set B and B’ to desired value
  - Turn on write enable (WE) momentarily
  - Disable WE keeps the new value in Q and Q’
Typical 6T SRAM Cell
- Each inverter can be implemented using 2 transistors
- Entire SRAM cell uses 6 transistors

Each SRAM cell is 6-10x larger than 1 DRAM cell

DRAM vs SRAM
- DRAM has 6x-10x density advantage
- SRAM has low access latency
  - No complex command
- SRAM has deterministic latency
  - No refresh
- SRAM can be fabricated in the same logic IC process
  - Can be built together with the CPU in the same die.

Next Time…
Providing capacity of DRAM and low latency of SRAM:

Memory Cache and Hierarchy

Acknowledgements
- These slides contain material developed and copyright by:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)
  - John Lazzaro (UCB)

- MIT material derived from course 6.823
- UCB material derived from course CS152, CS252