Chapter 5
Internal Memory

Computer Organization and Architecture

Semiconductor main memory
• Early computers used doughnut shaped ferromagnetic loops called cores for each bit
• Main memory was often referred to as “core” memory or just “core”
• Term persists: e.g. a core dump
• Semiconductors are almost universal today

Memory Cells
• Properties:
  – Exhibit two stable or semi-stable states representing 1 and 0
  – Capable of being written to at least once to set state
  – Capable of being read to sense the state

Memory Cell Operation
• Select line selects cell for operation specified by control line
• Control line has read or write signal
• Data/Sense line
• Most details below this level are beyond the scope of this course

Semiconductor Memory Types

<table>
<thead>
<tr>
<th>Memory Type</th>
<th>Category</th>
<th>Erase</th>
<th>Write Mechanism</th>
<th>Volatility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random-access memory</td>
<td>Read-write</td>
<td>Electrically, laser</td>
<td>Electrically</td>
<td>Non-volatile</td>
</tr>
<tr>
<td>Read-only memory</td>
<td>Read-only</td>
<td>Not applicable</td>
<td>Stable</td>
<td>Volatile</td>
</tr>
<tr>
<td>Erasable PROM</td>
<td>Read-only</td>
<td>UV light, infrared</td>
<td>Photolytically</td>
<td>Non-volatile</td>
</tr>
<tr>
<td>Electrically-erasable PROM</td>
<td>Read-only</td>
<td>UV light, infrared</td>
<td>Photolytically</td>
<td>Non-volatile</td>
</tr>
<tr>
<td>Flash memory</td>
<td>Read-only</td>
<td>Electrically</td>
<td>Non-volatile</td>
<td>Volatile</td>
</tr>
</tbody>
</table>

Semiconductor Memory
• RAM (Random Access Memory)
  – Misnamed as all semiconductor memory is “random access”
    - Time required to access any address is constant and does not depend on previous address accessed
  – Read/Write
  – Non-volatile
  – Temporary storage
• Two technologies:
  – Dynamic RAM: analog device, uses capacitor to store charge
  – Static RAM: digital device, uses flip-flop logic gates to store state
Dynamic RAM (DRAM)
- Bits stored as charge in capacitors
- But charges leak, need refreshing even when powered
- Simpler construction than static RAM (SRAM)
- Slower, but smaller per bit and less expensive than SRAM
- Used for main memory
- Essentially analog rather than digital
  - Level of charge determines value

Dynamic RAM Structure
- Address line
- Transistor
- Storage capacitor
- Bit line
- Ground

DRAM Operation
- Address line active when bit read or written
  - Transistor switch closed (current flows)
- Write
  - Voltage is applied to bit line
    - High for 1 low for 0
  - Then address line is activated
    - Transistor allows current to flow; transfers charge to capacitor
- Read
  - Address line is activated
    - Transistor allows current to flow; transfers charge from capacitor to bit line
  - Bit line fed to sense amplifier
    - Compares with reference value to determine 0 or 1
    - Capacitor charge must be restored to complete the read operation

Static RAM
- Bits stored as on/off switches (flip-flops)
- No charges to leak
- No refresh needed when powered
- More complex construction - 6 transistors
- Larger and more expensive per bit, but faster than DRAM
- Used for cache
- True digital device
  - Uses flip-flops

Static RAM Structure
- Transistor arrangement gives stable logic state
- State 1
  - C1, C2 high, C low
  - T1 T4 off, T2 T3 on
- State 0
  - C2 high, C1 low
    - T2 T3 off, T1 T4 on
  - Address line transistors T5, T6 form a switch
- Write - apply value to B & complement to B
- Read - value is on line B

Static RAM Operation
SRAM v DRAM
- Both are volatile
  - Power needed to preserve data
- Dynamic cell
  - Simpler to build and smaller than SRAM
  - Therefore more dense and less expensive
  - Needs refresh circuitry
  - Favored for larger memory units
- Static cells
  - Faster than DRAM
  - More expensive to build
  - Favored for cache memory

Acronym Soup
- So what is SDRAM?
  - Synchronous DRAM
- SDR SDRAM “Single Data Rate”
- DDR SDRAM “Double Data Rate”
  - DDR2, DDR3 double and quadruple r/w unit, DDR4 in development
- RDRAM Rambus DRAM (more later)
- Many more acronyms abound
- See A Fast Guide to RAM
  http://whatis.techtarget.com/definition/0,,sid9_gci523855,00.htm
- More on SDRAM later

Read Only Memory (ROM)
- Permanent, nonvolatile storage
- Typical usage:
  - Microprogramming (see later)
  - Library subroutines
  - Systems programs (BIOS)
  - Function tables

Types of ROM
- Data written during manufacture
  - Actually wired into the chip
  - Large fixed cost, expensive for small runs
- Programmable (once)
  - PROM is electrically programmed after manufacture
  - Needs special equipment to program

Read-mostly memories
- Read “mostly” memories can be rewritten
- Erasable Programmable (EPROM)
  - Optical erase of entire chip by UV
  - Can take 20 minutes to erase
  - Only one transistor per bit
- Electrically Erasable (EEPROM)
  - Takes much longer to write than read (several hundred microseconds)
  - Can rewrite single bytes
  - Less dense than EPROM

Flash memory
- Provides block electrical erase but not byte level
- High-density, only one transistor per bit
- Fast read speeds, but not as good as DRAM
- Erase (very slow) sets blocks to all 1’s
- After a block is erased, 0’s can be written to individual bits
  - But large block size for erase makes them considerably faster than other EEPROMs
  - Typical block (page) sizes are 512, 2048 and 4096
Flash memory
- Flash memory is not durable
- Mostly limited to ~50K to ~100K erase cycles
  - Recently extended upwards to ~1M cycles
- Flash drivers have to manage bad blocks in a fashion similar to disk drivers

Chip Logic
- Trade-offs in chip design among speed, capacity and cost
- Key issue is number of bits that can be written simultaneously
  - One extreme: physical arrangement of memory cells same as logical arrangement of words in memory. 16Mbit chip is 1M 16-bit words
  - Other extreme: one bit per chip, 16M memory uses 16 1-bit chips; with bit 1 of each word in chip 1 etc.

Organization in detail
- A 16Mbit (2 MByte) chip can be also organized as a 2048 x 2048 x 4bit array
  - Reduces number of address pins
    - Multiplex row address and column address
    - 11 pins to address (2^11=2048)
    - Adding one more pin doubles range of values so x4 capacity

Typical 16 Mbit DRAM (4M x 4)

Refreshing
- Refresh circuit included on chip
  - Disable chip
  - Count through rows
  - Read & Write back
- Refresh takes time
- Slows down apparent performance

Packaging

(a) 8 Mbit EPROM
(b) 16 Mbit DRAM
1 MB EPROM Packaging

- Organized as 1M 8 bit words; 32 pins
- A0 - A19 address pins (20 bit address)
- D0 - D7 are data out
- Power supply Vcc and ground Vss
- Chip enable pin CE
  - indicates whether address is valid for this chip; there may be several chips
  - CE pins are connected to H.O. bits of address lines on bus (> A19)
- Programming voltage pin Vpp used in write operations

16MBit DRAM

- Organized as 4Mx6 bits
- Data pins D1-D4 are input/output
- WE (Write enable) and OE (output enable) determine if R or W
- Because DRAM is accessed by row, then by column, 11 address pins A0-A10 are multiplexed
- RAS (row address select) and CAS (column address select) pins
- 2 supply Vcc and ground pins Vss
- One No-Connect NC to make even number of pins

256kByte Module Organisation

1MByte Module Organisation

Error Detection and Correction

- Hard Failure
  - Permanent defect
- Soft Error
  - Random, non-destructive
  - No permanent damage to memory
- A single parity bit can be used to detect (most) errors in a word
- Parity bit test can fail to detect errors when there is more than bit error
- Hamming codes can be used to detect AND correct errors

Error Correcting Code Function
Hamming Codes

• Developed at Bell Labs by Richard Hamming - late 1940’s

• Key to Hamming codes is to add extra parity bits in such a way that each bit in a word is checked by a unique combination of parity bits

• Parity bits themselves are included in parity computations - we can tell if there was simply an error in a parity bit

How many bits?

• For memory of M bits we need K check bits, where

\[ 2^K - 1 \geq M + K \]

<table>
<thead>
<tr>
<th>Data Bits</th>
<th>Check Bits</th>
<th>% Increase</th>
<th>Check Bits</th>
<th>% Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
<td>50</td>
<td>5</td>
<td>62.5</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>31.25</td>
<td>6</td>
<td>37.5</td>
</tr>
<tr>
<td>32</td>
<td>6</td>
<td>18.75</td>
<td>7</td>
<td>21.875</td>
</tr>
<tr>
<td>64</td>
<td>7</td>
<td>10.94</td>
<td>8</td>
<td>12.5</td>
</tr>
<tr>
<td>128</td>
<td>8</td>
<td>6.25</td>
<td>9</td>
<td>7.65</td>
</tr>
<tr>
<td>256</td>
<td>9</td>
<td>3.52</td>
<td>10</td>
<td>3.91</td>
</tr>
</tbody>
</table>

History of Hamming Codes

• Richard Hamming worked on a Bell Model V electro-mechanical computer

• Input was on punch cards, and the reader was not reliable

• On weekdays it would flash lights to alert the operator of an error. On weekends it would just abort the job

• Richard Hamming unfortunately worked on weekends and became rather annoyed at having to re-run all of his jobs, so he invented error-correcting codes!

Error Correction

• Data read from memory does not match parity computation for A and C

• Only one bit is in A and C but not B

• We will correct that bit to a 1

Pattern for Check Bits

• \( C_n \): skip \( n \) bits, check \( n \) bits, skip \( n \) bits, check \( n \) bits, skip \( n \) bits, ...

• Bits are numbered from 1 (not zero)

• Check bits are placed in every bit position that is a power of 2

  - \( c_1 = 1 \), \( c_2 = 2 \), \( c_3 = 4 \), \( c_4 = 8 \), \( c_5 = 16 \), ...

  - \( 7 = 1 + 2 + 4 \), so pos 7 checked by \( c_1, c_2, c_3 \)

  - \( 9 = 1 + 8 \), so pos 9 checked by \( c_1, c_4 \)
Computing a Hamming Code

- Note that each check bit also checks itself!
- Example: A byte of data: 10100010
  
  Create the data word, leaving spaces for the parity bits:
  
  _ _ 1 _ 01 1 _ 0 0 1 0

  Calculate the parity for each check bit (? represents the bit position being set; treat as 0):
  
  - Position 1 checks bits 1,3,5,7,9,11:
    1.1.011_0010. Odd parity so c1 = -1:
    1.1_0111_0010
  
  - Position 2 checks bits 2,3,6,7,10,11:
    111.0111_0010. Even parity so c2 = 0:
    101.1111_0010

  - Position 4 checks bits 4,5,6,7,12:
    101.0011_0010. Odd parity so c3 = -1:
    10100111_0010

  - Position 8 checks bits 8,9,10,11,12:
    10100111_0010. Odd parity so c4 = -1:
    101001110010

  Coded word: 101001110010

Finding the error

- We had 101001110010
- Suppose mem read = 101001110110
- Compute check bits:
  
  _ _ 1 _ 01 1 _ 0 0 1 0
  
  1 2 3 4 5 6 7 8 9 10 11 12
  
  c1 [3,5,7,9,11] < -1 (OK)
  c2 [3,6,7,10,11] < 1 (but c2 = 0)
  c3 [5,6,7,12] < -0 (OK)
  c4 [9,10,11,12] < 0 (but c4 = 1)

  Error: c2 in pos 2, c4 in pos 8. 8+2=10
  Correct bit 10: 101001110010

Error in check bit

- We had 101001110010
- Suppose mem read = 101001100010
- Compute check bits:
  
  _ _ 1 _ 01 1 _ 0 0 1 0
  
  1 2 3 4 5 6 7 8 9 10 11 12
  
  c1 [3,5,7,9,11] < -1 (OK)
  c2 [3,6,7,10,11] < 0 (OK)
  c3 [5,6,7,12] < 0 (OK)
  c4 [9,10,11,12] < 1 (but c4 = 0)

  Error: only c4 in pos 8.

  No data bits were affected; check bit itself is the error

SEC-DED Coding

- Hamming codes as described above are capable only of correcting single-bit errors
  - A double-bit error cannot be distinguished from a single-bit error in a different code
  - Correction will add a third error!
- Adding a single parity bit over the entire M+K bits allows double-bit errors to be detected (but not corrected)
  - Hence “Single Error Correction, Double Error Detection”
- A common implementation uses 8 check bits per 64 bits of memory
  - Same overhead as older 9-bit parity check DRAM

Advanced DRAM Organization

- Memory access is a bottleneck (the “von Neumann bottleneck”) in a high-performance system
- Basic DRAM same since first RAM chips
- SRAM cache is one line of attack
  - Expensive
  - Diminishing returns as more cache is added

DRAM Enhancements

- DDR-DRAM: Double-data-rate DRAM
  - DDR-SDRAM now available
- SDRAM: Synchronous DRAM
- RDRAM: Rambus DRAM
- CDRAM: Cache DRAM
- DDR = Double Data Rate
  - Data sent on both rising and falling edges of clock signal

<table>
<thead>
<tr>
<th></th>
<th>Clock Frequency (MHz)</th>
<th>Transfer Rate (GB/s)</th>
<th>Access Time (ns)</th>
<th>Pin Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDRAM</td>
<td>260</td>
<td>3.3</td>
<td>17.5</td>
<td>184</td>
</tr>
<tr>
<td>DDR</td>
<td>260</td>
<td>1.8</td>
<td>25.5</td>
<td>164</td>
</tr>
<tr>
<td>DDR-SDRAM</td>
<td>260</td>
<td>1.8</td>
<td>15.5</td>
<td>164</td>
</tr>
</tbody>
</table>
Synchronous DRAM (SDRAM)

- Access is synchronized with an external clock
  - Very fast (18 ns access time, 1.3 Gbps at 166 MHz bus speed)
- Request on address bus latched in the SDRAM
- Some time later RAM responds with data
  - (CPU waits for conventional DRAM)
  - Since SDRAM moves data in time with system clock, CPU knows when data will be ready
  - CPU and bus do not have to wait; can do something else
- Burst mode allows SDRAM to set up stream of data and fire it out in block
- DDR-SDRAM sends data twice per clock cycle (leading & trailing edge)

SDRAM latency is about the same asynchronous DRAM

- Speed comes from ability to interleave operations between banks of chips
- Note that SDRAM has a “mode register”
  - Specifies things like burst length and read timing

Typical SDRAM Organization

- A 512 MB SDRAM DIMM might be made of 8 or 9 SDRAM chips, each containing 512 Mbit of storage, and each one contributing 8 bits to the DIMM’s 64- or 72-bit width.
- A typical 512 Mbit SDRAM chip internally contains 4 independent 16 Mbyte banks. Each bank is an array of 8192 rows of 16384 bits each. A bank is either idle, active, or changing from one to the other.
- An active command activates an idle bank. It takes a 2-bit bank address (BAD-BA1) and a 13-bit row address (AD-A12), and reads that row into the bank’s array of 16384 sense amplifiers. This is also known as ‘opening’ the row. This operation has the side effect of refreshing that row.
- Once the row has been activated or ‘opened’, read and write commands are possible. Each command requires a column address, but because each chip works on 8 bits at a time, there are 2048 possible column addresses, needing only 11 address lines (AD-A11).
- (Source: http://en.wikipedia.org/wiki/SDRAM)

RAMBUS

- Intel attempted to adopt RDRAM for Pentium & Itanium; provided commitments and funding 1997-2000
  - Market share never exceeded 5%
  - RAMBUS still is involved in several nasty legal battles; primary income is from patent licensing
  - Also alleges DDR mfrs fixed prices to suppress RDRAM
- Vertical package - all pins on one side
- Data exchange over 28 wires < cm long
- Bus addresses up to 320 RDRAM chips at 1.6Gbps
- Asynchronous block protocol
  - 480ns access time
  - Then 1.6 Gbps
- Can require heat-sinks
**RAMBUS Diagram**

**DDR SDRAM**
- SDRAM can only send data once per clock
- Double-data-rate SDRAM can send data twice per clock cycle
  - Rising edge and falling edge
- Fast, 12.5ns access time, 3.2Gbs at 200MHz bus speed

**DDR SDRAM Read Timing**

**DDR2 and DDR3**
- Operate at twice or more the bus speed of DDR SDRAM
- Higher latency than DDR SDRAM means worse performance at same bus speed
- Both have larger prefetch buffers

<table>
<thead>
<tr>
<th></th>
<th>DDR1</th>
<th>DDR2</th>
<th>DDR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh buffer</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Voltage level (V)</td>
<td>2.5</td>
<td>1.8</td>
<td>1.5</td>
</tr>
<tr>
<td>Front side bus data rate (MHz)</td>
<td>200, 246, 333, 400</td>
<td>400, 533, 667, 800</td>
<td>800, 1066, 1333, 1600</td>
</tr>
</tbody>
</table>

**SDR/DDR/DDR2 Comparison**

**Cache DRAM**
- Mitsubishi
- Integrates small SRAM cache (16 kb) onto generic DRAM chip
- Used as true cache
  - 64-bit lines
  - Effective for ordinary random access
- To support serial access of block of data
  - E.g., refresh bit-mapped screen
  - CDRAM can prefetch data from DRAM into SRAM buffer
  - Subsequent accesses solely to SRAM