Computer Organization
CS 231-01

Error Detection and Correction

Dr. William H. Robinson
September 6, 2004

http://eecs.vanderbilt.edu/courses/cs231/

As a reward for attending class

• Prepare for Quiz #1

Topics

“…forasmuch as ye know that your labour is not in vain…”
– 1 Corinthians 15:58

• General announcements
• Cache review
• Error detecting codes
• Error correcting codes
• Hamming algorithm
• Assignment
  – Continue reading Chapter 2 in Tanenbaum

Guest Lecture

• Dr. Robinson will be on travel
  – Monday, September 13, 2004

• Dr. Larry Dowdy will cover the lecture
  – Lecture topics will remain on schedule
  – Responsible for material presented
What is a Cache?

- Small, fast storage used to improve average access time to slow memory.
- Exploits spatial and temporal locality
- In computer architecture, almost everything is a cache!
  - Registers a cache on variables
  - First-level cache a cache on second-level cache
  - Second-level cache a cache on memory
  - Memory a cache on disk (virtual memory)

Why Do Caches Work?

- **Spatial locality** – words in close physical proximity to the word being read will probably be read also.
- **Temporal locality** – a recently read word will probably be read again soon.

Types of Caches

- **Direct-mapped**
  - Cache line only has 1 possible location
  - Simple to implement

- **Set-associative**
  - Cache line has a “set” of possible locations
  - e.g. 4-way set-associative

- **Fully-associative**
  - Cache line can be placed anywhere
  - Costly in H/W to check entire cache for hit

Terminology

- N-bit address
- TAG, LINE, WORD, BYTE
  - **TAG field**
    - Indicates the address tag for comparison
  - **LINE field**
    - Indicates the proper cache entry
  - **WORD field**
    - Indicates which word referenced within a line
  - **BYTE field**
    - Indicates a single byte, but not normally used
Direct-Mapped Cache

Read 0x0A72

<table>
<thead>
<tr>
<th>Entry</th>
<th>Valid</th>
<th>Tag</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>2</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>3</td>
<td>Y</td>
<td>0x0A</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>5</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>6</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>7</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

Mean Access Time

Given
- Cache access time, \( c \)
- Main memory access time, \( m \)
- Hit ratio, \( h \)

\[
\text{mean \_access \_time} = c + (1 - h)m
\]

Example

\[
\text{mean \_access \_time} = 5ns + (1 - 0.85) \times 40ns
\]

Mean access time = 11ns

Example 2

\[
\text{mean \_access \_time} = 5ns + (1 - 0.98) \times 40ns
\]

Mean access time = 5.8ns

Increasing hit ratio gets the performance advantage of cache with the capacity of main memory.
A Small Error?

What if…
• You were to receive $17 million by EFT
• The “17” is encoded $10001_2$ and transmitted
• The received data is $00001_2$

Only one bit was changed, but the impact is quite significant!!!
Where Do Errors Come From?

- Voltage spikes from power supply
- Coupling with nearby signals
- Noise on transmission line
- Radiation
- Manufacturing defect
- Etc.

One Step Further

- Scenario: transmit a yes/no answer
  - Codeword for “yes” = 1111₂
  - Codeword for “no” = 0000₂

- Received codeword is 0010₂
  - Is this valid?
  - What’s the likely answer?

- Closest to 0000₂ codeword
  - Hamming distance equals one
  - Correct the codeword

One Step Further

- Received codeword is 1110₂
  - Is this valid?
  - What’s the likely answer?

- Received codeword is 1010₂
  - Is this valid?
  - What’s the likely answer?

Terminology

- Hamming Distance
  - Number of bit positions in which two codewords differ

- Minimum Distance
  - Minimum Hamming distance between all distinct pairs of codewords

A code with minimum distance \(d_{\text{min}}\) can correct all error patterns of weight less than or equal to \([(d_{\text{min}} - 1)/2]\)
Error Correction

- Scenario: transmit a yes/no answer
  - Codeword for "yes" = $1111_2$
  - Codeword for "no" = $0000_2$

- Minimum distance equals 4
  - Error correction less than or equal to $((d_{\text{min}} - 1)/2)$

- Therefore our code corrects all errors $\leq 1.5$

- Do we need the 4th bit for the same capability?

Summary

- Data can be encoded to detect or correct errors

- Minimum Hamming distance determines the error detection/correction capability of a code