Guerrilla 5: FL, Cache

July 21, 2016
Floating Points

Floating Points

Caches
IEEE Floating Points

Normalized: \((-1)^S \times 2^{\text{Exponent} - \text{bias}} \times 1.\text{Mantissa}\)

Denorm: \((-1)^S \times 2^{-\text{bias} + 1} \times 0.\text{Mantissa}\)

\[
\text{bias} = 2^{\#\text{Exp bits} - 1} - 1 \quad \text{(for above: } 2^{8-1} - 1 = 127)\]
### IEEE Floating Points

<table>
<thead>
<tr>
<th>Exponent</th>
<th>Mantissa</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>0</td>
<td>± 0</td>
</tr>
<tr>
<td>0x00</td>
<td>non-zero</td>
<td>± Denorm Num</td>
</tr>
<tr>
<td>0x01 – 0xFE</td>
<td>anything</td>
<td>± Norm Num</td>
</tr>
<tr>
<td>0xFF</td>
<td>0</td>
<td>± ∞</td>
</tr>
<tr>
<td>0xFF</td>
<td>non-zero</td>
<td>NaN</td>
</tr>
</tbody>
</table>
Why cache?

- Spacial locality: Memory accesses over time tend to be near each other
- Temporal locality: Tend to access a few group of addresses frequently for a given workload
Config. for Caches

- Cache size = amount of memory data that the cache can hold (does not include tag bits, valid etc.)
- Blocksize = size of a cache block (unit of transfer from cache and memory)
- N way = associativity (how many data blocks can be in the cache for the same set/index) # = how many data blocks need to be checked for a cache access
How many bits?

- Offset bits = $\log_2(\text{Blocksize})$
- Index bits = $\log_2(\text{number of sets}) = \log_2\left(\frac{\text{Blocks}}{N}\right) = \log_2(\text{cachesize}) - \log_2(N) - \#\text{OffsetBits}$
- Tag Bits = $\text{AddressLength} - \#\text{OffsetBits} - \#\text{IndexBits}$
Write Policy

- Cache write hit policy:
  - Write-back (requires dirty bit): Write to cache
  - Write-through: Write to cache and data memory

- Cache write miss policy:
  - Write allocate: Write in memory and put in cache
  - No write allocate: Write in memory don’t put in cache