Guerrilla 5: FL, Cache

July 21, 2016

Floating Points

Floating Points

Caches

IEEE Floating Points

31	30 … 23	22 … 0
S	Exponent	Mantissa

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

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

bias = $2^{\#\text{Exp bits} - 1} - 1$ (for above: $2^{8-1} - 1 = 127$)

IEEE Floating Points

Exponent	Mantissa	Meaning
0x00	0	± 0
0x00	non-zero	± Denorm Num
0x01 – 0xFE	anything	± Norm Num
0xFF	0	± ∞
0xFF	non-zero	NaN

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 = log2(Blocksize)
- Index bits = log2(number of sets) = log2($\frac{\#Blocks}{N}$) = log2(cachesize) - log2(N) - $\#OffsetBits$
- Tag Bits = $AddressLength - \#OffsetBits - \#IndexBits$

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