MEMORY HIERARCHY DESIGN

unlimited amount of fast memory. -> very expensive
memory hierarchy.
↳ which takes advantage of locality.
temporal locality , spatial locality.

several levels, each smaller, faster and more expensive per byte than next lower level..

data contained in a lower level are a superset of next higher level - inclusion principle.

because highend processor have multiple, cores, the bw requirements are greater than for single core.

intel i7 6700 can generate 2 data memory references per core each clock cycle. clock rate 4.2GHz generate a peak of 32.8 billion 64-bit data memory refs.

in addition a peak instruction demand of about 12.8 billion 128-bit instruction reps.
total peak demand bw of 409.6 GB/s.

this can be achieved by by multiporting and pipllining the caches. Bw for DRAM main memory using 2 channels is only 8%. of the demand bandwidth.

memory designers have focussed on optimizing average memory access time.

depends upon

- cache acess time
- miss rate
- miss penalty.

power has become a major consideration, in highend microprocessors, ~~ter~~ there may be 60MiB of on chip cache.

significant power.

we have leakage - static power - and we have active ~~d~~power - dynamic power - when performing read and write operations.

memory on storage.

serious problem in PMD.
CPU is less aggressive, and power budget may be 20-50 times smaller

basics of memory hierarchy

block (or line) spatial locality.
each block -tag.

we have direct mapped cache.

fully associate cache

set associative cache

(block address) MOD (no. of sets in cache).

write through cache and write back, both strategies can use write buffer.

miss rate is fraction of cache accesses that result in a miss ~~the~~.

3 Cs model for misses into 3 categories

- compulsory miss

  the very first access to a block cannot be in the cache

- capacity miss.

  if the cache cannot contain all the blocks needed during execution of a program, capacity misses will occur.

- conflict miss.

  if the block placement strategy is not fully associative, conflict misses will occur

4th C when we have multithreading / multiple cores.

coherence.

avg. mem access time

= hit time + miss rate × miss penalty.

BASIC OPTIMIZATIONS (6)

1. large block size to reduce miss rate.

increases miss penalty

2. bigger caches reduce miss rate

Potentially longer hit time of the larger cache memory + power + cost.

3. higher associativity to reduces ~~misses~~ rate.

at the cost of increased hit time + power.

4. multi-level caches to reduce miss penalty.

$\text{hit time}_{L1} + \text{missrate}_{L1} (\text{hit time}_{L2} + \text{missrate}_{L2} * \text{miss penalty}_{L2})$

5. giving priority to read misses over write to reduce miss penalty.
   (writes can be put in write buffer)

6. avoiding address translation during indexing of the cache to reduce hit time.

Keywords link wiki

Date 1.4.2020
Page 1

energy consumption is also a consideration, many processo take atleast 2 TLB out of critical path.

- access wait prediction to reduce wait time.

- merging write buffers to reduce miss penalty.

- compiler optimization to reduce miss rate.

✓ - compiler controlled prefetching to reduce miss penalty or miss rate.

- using HBM to extend memory hierarchy.

MEMORY TECHNOLOGY AND OPTIMIZATIONS.

phase charg memory - memristor

WAREHOUSE SCALE COMPUTERS