1. A direct-mapped cache consists of eight blocks. Main memory contains 4K blocks of eight words each. Access time for the cache is 22ns and the time required to fill a cache slot from main memory is 300ns. (This time allows us to determine the block is missing and bring it into cache.) Assume a request is always started in parallel to both cache and to main memory (so if it is not found in cache, we do not have to add this cache search time to the memory access). If a block is missing from cache, the entire block is brought into the cache and the access is restarted. Initially, the cache is empty.
2. Show the main memory address format that allows us to map addresses from main memory to cache. Be sure to include the fields as well as their sizes.
3. Compute the hit ratio for a program that loops 4 times from location 0 to 6710 in memory.
4. Compute the effective access time for this program.

Answer:

1. Show the main memory address format that allows us to map addresses from main memory to cache. Be sure to include the fields as well as their sizes

4K = 212 blocks

8 word each = 23 bits

=>main memory = block x words each = 212 x 23 = 215 words

We have 15 bits in address

Cache contain = 23 blocks = 3 bits

Each blocks have 23 word = 3 bits

Tag = 15 – 3 – 3 = 9 bits

1. Compute the hit ratio

Loops 4 times from location 0 to 67

Block 0 = 0 to 7

Block 1 = 8 to 15

Block 2 = 16 to 23

Block 3 = 24 to 31

Block 4 = 32 to 39

Block 5 = 40 to 47

Block 6 = 48 to 55

Block 7 = 56 to 63

Block 8 = 64 to 67

Block 0: 4 misses, 28 hits

Block 1: 1 miss, 31 hits

Block 2: 1 miss, 31 hits

Block 3: 1 miss, 31 hits

Block 4: 1 miss, 31 hits

Block 5: 1 miss, 31 hits

Block 6: 1 miss, 31 hits

Block 7: 1 miss, 31 hits

Block 8: 4 misses, 12 hits

Total miss = 15

Total hit = 257

Hit ratio = 257/ (15 + 257) = 94.49%

1. Effective access time

EAT= (Hit Ratio×Hit Time) + (Miss Ratio×Miss Time)

Miss Ratio = 1 - Hit Ratio = 1 - 0.875 = 0.125

EAT=(0.875×22ns)+(0.125×322ns) = 59.5ns

6. Given a virtual memory system with a TLB, a cache, and a page table, assume the

following:

• A TLB hit requires 5ns.

• A cache hit requires 12ns.

• A memory reference requires 25ns.

• A disk reference requires 200ms (this includes updating the page table, cache, and TLB).

• The TLB hit ratio is 90%.

• The cache hit rate is 98%.

• The page fault rate is .001%.

• On a TLB or cache miss, the time required for access includes a TLB and/or cache

update, but the access is not restarted.

• On a page fault, the page is fetched from disk, all updates are performed, but the access

is restarted.

• All references are sequential (no overlap, nothing done in parallel).

For each of the following, indicate whether or not it is possible. If it is possible, specify the

time required for accessing the requested data.

a) TLB hit, cache hit

b) TLB miss, page table hit, cache hit

c) TLB miss, page table hit, cache miss

d) TLB miss, page table miss, cache hit

e) TLB miss, page table miss

Write down the equation to calculate the effective access time.

Solution:

1. TLB hit, cache hit

Time = TLB hit + cache hit = 5 + 12 = 17 ns, possible

1. TLB miss, page table hit, cache hit

Time = TLB miss + page table hit + cache hit = 5 + 25 + 12 = 42ns, possible

1. TLB miss, page table hit, cache miss

Time = 5 + 25 + 12 = 42ns, possible

1. TLB miss, page table miss, cache hit

It is impossible, we get a table miss we cannot go cache.

1. TLB miss, page table miss

First, we can access the disk and update the memory and restore the instruction

Time = TLB miss + page table miss + disk access + TLB access + page table access

= 5ns + 25ns + 200ms + 5ns + 12ns = 200.015ms

5. A system implements a paged virtual address space for each process using a one level page table. The maximum size of virtual address space is 16MB. The page table for the running process includes the following valid entries (the → notation indicates that a virtual page maps to the given page frame, that is, it is located in that frame):

Virtual page 2 → Page frame 4 Virtual page 4 → Page frame 9

Virtual page 1 → Page frame 2 Virtual page 3 → Page frame 16

Virtual page 0→ Page frame 1

The page size is 1024 byte and the maximum physical memory size of the machine is 2MB.

a). How many bits are required for each virtual address?

b). How many bits are required for each physical address?

c). What is the maximum number of entries in a page table?

d). To which physical address will the virtual address 152410 translate?

e). Which virtual address will translate to physical address 102410?

Answer:

a). Require for each virtual address:

\_ The max of address space is

⮚ 16MB = = = 24 bits

Therefore: There are 24 bits are required for each virtual address.

b). Require for each physical address:

⮚ Physical address = 2MB = = 21 bits

Therefore: There are 21 bits are required for each physical address.

c). The maximum number of entries in a page table:

⮚ Max number of entries page = = = 14 bits

Therefore: 14 bits is the maximum number of entries in a page table.

d). Physical address will the virtual address 152410 translate:

\_ Virtual page 1 → Page frame 2

⮚Page 0 = 0 →

⮚Page 1 = 1024 → 2047

e). Virtual address will translate to physical address 102410

\_ Virtual address → Physical address 102410

⮚ Virtual page 0 → Page frame 1