Name:SOUNG VICHKA
ID: e20240672
Group: I2-GIC-2B

TP8

Exercise 1

Suppose we have a computer that uses a memory address word size of 8 bits. This computer has a 16-byte cache with 4 bytes per block. The computer accesses a number of memory locations throughout the course of running a program.

Suppose this computer uses direct-mapped cache. The format of a memory address as seen by the cache is shown here:

Tag	Block	Word
4 bits	2 bits	2 bits

The system accesses memory addresses (in hex) in this exact order: 6E, B9, 17, E0, 4E, 4F, 50, 91, A8, A9, AB, AD, 93, and 94. The memory addresses of the first four accesses have been loaded into the cache blocks as shown below.

Initial Cache State:

Block 0 (Tag: 1110)

Tag Contents	Cache Contents (address)
1110	E0
1110	E1
1110	E2
1110	E3

Block 1 (Tag: 0001)

Tag Contents	Cache Contents (address)
0001	14
0001	15
0001	16
0001	17

Block 2 (Tag: 1011)

Tag Contents	Cache Contents (address)
1011	B8
1011	B9
1011	BA
1011	BB

Block 3 (Tag: 0110)

Tag Contents	Cache Contents (address)
0110	6C
0110	6D
0110	6E
0110	6F

a) What is the hit ratio for the entire memory reference sequence given above?
b) What memory blocks will be in the cache after the last address has been accessed?

Answer:

a) Hit Ratio for the Memory Reference Sequence

First, we convert the memory addresses from hexadecimal to binary:

Hex Address	Binary
6E	0110 1110
B9	1011 1001
17	0001 0111
E0	1110 0000
4E	0100 1110
4F	0100 1111
50	0101 0000
91	1001 0001
A8	1010 1000
A9	1010 1001
AB	1010 1011

AD	1010 1101
93	1001 0011
94	1001 0100

Next, we analyze which address results in a hit or miss:

Address (Hex)	Binary	Tag	Block	Word	Result
6E	0110 1110	0110	11	10	Miss
B9	1011 1001	1011	10	01	Miss
17	0001 0111	0001	01	11	Miss
E0	1110 0000	1110	00	00	Miss
4E	0100 1110	0100	11	10	Miss
4F	0100 1111	0100	11	11	Hit
50	0101 0000	0101	00	00	Miss
91	1001 0001	1001	00	01	Miss
A8	1010 1000	1010	10	00	Miss
A9	1010 1001	1010	10	01	Hit
AB	1010 1011	1010	10	11	Hit
AD	1010 1101	1010	11	01	Miss
93	1001 0011	1001	00	11	Hit
94	1001 0100	1001	01	00	Miss

Total hits = 4 (addresses: 4F, A9, AB, 93)
Total accesses = 14
Hit Ratio = Total Hits / Total Accesses = 4 / 14 ≈ 0.2857 ≈ 28.57%

b) Memory Blocks in Cache after Last Access

After all 14 memory accesses, the final state of the cache is:

Block 0 (Tag: 1001 - from address 90-93)

Tag Contents	Cache Contents (address)
1001	90

1001	91
1001	92
1001	93

Block 1 (Tag: 1001 - from address 94-97)

Tag Contents	Cache Contents (address)
1001	94
1001	95
1001	96
1001	97

Block 2 (Tag: 1010 - from address A8-AB)

Tag Contents	Cache Contents (address)
1010	A8
1010	A9
1010	AA
1010	AB

Block 3 (Tag: 1010 - from address AC-AF)

Tag Contents	Cache Contents (address)
1010	AC
1010	AD
1010	AE
1010	AF

Exercise 2

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 16 MB. The page table for the running process includes the following valid entries:

- 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 bytes and the maximum physical memory size of the machine is 2 MB.

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 1524_{10} translate?
e) Which virtual address will translate to physical address 1024_{10}?

Answer:

a) Bits Required for Each Virtual Address

- Virtual address space = 16 MB = 16 × 2^{20} bytes = 2^{24} bytes
- Number of bits required to address 16 MB = $\log_2(2^{24})$ = 24 bits

b) Bits Required for Each Physical Address

- Physical memory size = 2 MB = 2 × 2^{20} bytes = 2^{21} bytes
- Number of bits required to address 2 MB = $\log_2(2^{21})$ = 21 bits

c) Maximum Number of Entries in a Page Table

- Page size = 1024 bytes = 2^{10} bytes
- Total virtual address bits = 24 bits
- Number of pages = Total virtual memory size / Page size = 2^{24} / 2^{10} = 2^{14} pages

Therefore, the maximum number of entries in a page table = 2^{14} = 16,384 entries

d) Physical Address for Virtual Address 1524_{10}

- Virtual page number = ⌊1524 / 1024⌋ = 1
- Offset within the page = 1524 − 1024 = 500
- From the page table: Virtual page 1 → Page frame 2
- Physical frame size = page size = 1024 bytes
- Physical address = (Page frame number × Page size) + Offset
  = (2 × 1024) + 500 = 2048 + 500 = 2548

Thus, the virtual address 1524_{10} translates to physical address 2548.

e) Virtual Address for Physical Address 1024_{10}

- Physical frame number = ⌊1024 / 1024⌋ = 1
- Offset = 1024 − 1024 = 0
- Find which virtual page maps to physical frame 1: Virtual page 0 → Page frame 1
- Virtual address = (Virtual page number × Page size) + Offset
  = (0 × 1024) + 0 = 0

Thus, the physical address 1024_{10} translates to virtual address 0.

Exercise 3

Given a virtual memory system with a TLB, a cache, and a page table, assume the following:

- A TLB hit requires 5 ns
- A cache hit requires 12 ns
- A memory reference requires 25 ns
- A disk reference requires 200 ms (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 0.001%
- 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.

Answer:

a) TLB hit, cache hit

Possible — This is the best-case scenario.

- TLB hit: 5 ns
- Cache hit: 12 ns

  Total Time = TLB hit + Cache hit = 5 + 12 = 17 ns

b) TLB miss, page table hit, cache hit

Possible.

- TLB miss (check): 5 ns
- Page table lookup in memory: 25 ns
- Cache hit (data retrieval): 12 ns

  Total Time = TLB check + Page table lookup + Cache hit = 5 + 25 + 12 = 42 ns

c) TLB miss, page table hit, cache miss

Possible.

- TLB miss (check): 5 ns

- Page table lookup: 25 ns
- Cache miss + memory reference for data: 12 ns + 25 ns

  Total Time = TLB check + Page table lookup + Cache miss + Memory reference = 5 + 25 + 12 + 25 = 67 ns

d) TLB miss, page table miss, cache hit

Not Possible.

If the page table misses (page fault), the page must be fetched from disk. The data cannot be in the cache since it was not yet present in memory. Therefore, a cache hit cannot occur simultaneously with a page table miss.

e) TLB miss, page table miss (Page Fault)

Possible — This is the page fault case.

- TLB miss (check): 5 ns
- Page table lookup: 25 ns
- Page fault — disk access: 200 ms = 200,000,000 ns
- After page fault, access restarts (the restart repeats the full access)

  Total Time = TLB check + Page table lookup + Disk access = 5 + 25 + 200,000,000 = 200,000,030 ns

Effective Access Time (EAT) Equation

The Effective Access Time is computed as a weighted sum over all possible memory access scenarios:

EAT = P(TLB hit) × Time(TLB hit) + P(TLB miss) × Time(TLB miss)

Where:

- Time(TLB hit) = P(cache hit) × Time(cache hit) + P(cache miss) × Time(cache miss)
  - Time(cache hit) = TLB hit time + cache access time = 5 + 12 = 17 ns
  - Time(cache miss) = TLB hit time + cache miss time + memory access time = 5 + 12 + 25 = 42 ns

- Time(TLB miss) = P(page table hit) × Time(page table hit) + P(page table miss) × Time(page table miss)
  - Time(page table hit) = P(cache hit) × Time(cache hit) + P(cache miss) × Time(cache miss)
  - Time(cache hit) = TLB miss + page table + cache hit = 5 + 25 + 12 = 42 ns
  - Time(cache miss) = TLB miss + page table + cache miss + memory = 5 + 25 + 12 + 25 = 67 ns
  - Time(page table miss) = Page Fault = TLB miss + page table lookup + disk fetch

Substituting the given values (TLB hit ratio = 90% = 0.9, cache hit rate = 98% = 0.98, page fault rate = 0.001% = 0.00001):

Time(TLB hit) = (0.98 × 17) + (0.02 × 42) = 16.66 + 0.84 = 17.5 ns

Time(page table hit cache hit) = (0.98 × 42) + (0.02 × 67) = 41.16 + 1.34 = 42.5 ns

Time(TLB miss) = (1 − 0.00001) × 42.5 + 0.00001 × 200,000,030 ≈ 42.5 + 2000 ≈ 2042.5 ns

EAT = 0.9 × 17.5 + 0.1 × 2042.5 = 15.75 + 204.25 ≈ 220 ns