Computer Architecture

Fall, 2025
Week 15
2025.12.8

組別：_____ 簽名：_____

[group 1]

1. **True or False: Addressing for 32-bit addresses**

In a memory hierarchy system consisting of a TLB, a Page Table (in main memory), and a Physically Addressed Cache, specific inclusion properties must hold.

Please analyze the following three scenarios describing the outcome of a single memory access request. For each scenario, state whether it is Possible or Impossible. If you answer "Possible," briefly describe the sequence of events. If you answer "Impossible," explain why based on the relationship between these hardware structures.

1. Scenario A: TLB Hit, but a Page Fault occurs (Page is not in Main Memory).
2. Scenario B: TLB Miss, Page Table Hit, and Cache Hit.
3. Scenario C: Page Fault occurs, but Cache Hit.

Ans:

1.Scenario A: Impossible

A TLB entry is a copy of a valid Page Table Entry (PTE). If a translation is present in the TLB (TLB Hit), it implies the page is currently resident in physical memory (Valid bit = 1). Therefore, you cannot have a TLB Hit for a page that is not in memory.

2. Scenario B: Possible

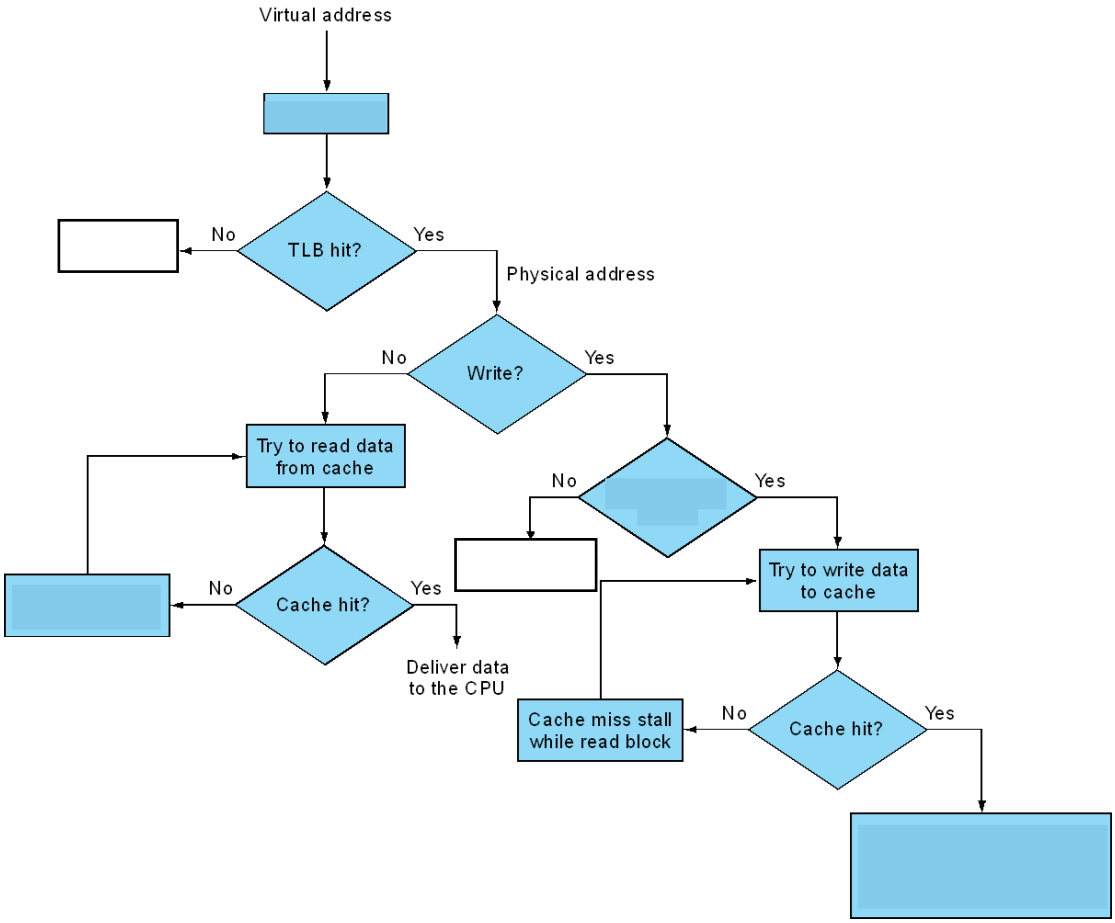
This is a standard occurrence. The translation was not in the TLB (TLB Miss), so the hardware/OS looked up the Page Table. The page was found in the Page Table (Page Table Hit, meaning the page is in memory). After obtaining the physical address, the system checked the Cache and found the data block there (Cache Hit).

3. Scenario C: Impossible

The Cache is a subset of Main Memory. Data can only exist in the cache if the corresponding page is currently resident in Main Memory. If a Page Fault occurs (meaning the page is on the disk and not in memory), the data cannot possibly be valid in the Cache.

[group 2]

2. Question:



1. What is the first step the system takes when the CPU issues a virtual address, and what is the immediate result if this step is successful (TLB Hit)?
2. According to the flowchart, what specific event occurs if the system attempts to perform TLB access but experiences a TLB Miss?
3. During a write operation, what check is performed immediately before attempting to write the data to the cache, and what is the consequence if this check fails?
4. Assume a successful write operation. If the subsequent Cache hit? step results in a Yes, what three actions are performed?
5. If the operation is a read (Write? No), what is the subsequent check performed, and what happens if this check fails (Cache hit? No)?

Ans:

1. TLB access. If TLB hit, the system retrieves the corresponding physical address.
2. TLB miss exception
3. “Write access bit on?” performed. If the bit is No (meaning write access is not allowed) -> write protection exception occurs.
4. 1. Write data into cache 2. Update the dirty bit 3. Put the data and the address into the write buffer
5. If “Cache hit?” fails (No), a cache miss stall while read block data being delivered to the CPU via a read operation.

[group 3]

3. Question:

According to the lecture slides on TLB Miss and page fault, which of the following statements are correct? (Select all that apply.)
(A) When a TLB miss occurs and the page is already in memory, the processor loads the corresponding page-table entry (PTE) from memory and retries the instruction.
(B) A TLB miss may be handled either by hardware or by software.
(C) If the page is not in memory, a page fault occurs, and the operating system fetches the page and updates the page table.
(D) On every TLB miss, the operating system must fetch the page from disk before the instruction can be retried.
(E) When the page is in memory, a TLB miss can be resolved without causing a page fault.

Ans:

(A)、(B)、(C)、(E)

(D) check whether in page table first.

[group 4]

4. Question:

(1) In a virtual memory system, a page fault occurs when the requested virtual page is not found in the main memory.

(2) Increasing page size always reduces the page fault rate.

(3) TLB miss handling always requires a disk access.

(4) Fully associative placement is commonly used for page tables because page fault penalties are extremely large.

Ans:

(1) True. A page fault means the page is not resident in physical memory, so the OS must load it from disk.
(2) False. Larger pages reduce page faults due to spatial locality, but too-large pages cause internal fragmentation and may increase faults.
(3) False. A TLB miss usually only requires reading the page table from memory. A disk access happens only if the page is not in memory (a page fault).
(4) True. Virtual memory uses fully associative placement to minimize conflict misses, since a page fault is extremely expensive.

[group 5]

5. Question:

In the context of memory hierarchy misses, what are the "Three C's," and how does associativity affect them?

Ans:

The "Three C's" classify the sources of cache misses:

- Compulsory misses (Cold start): Occur during the first access to a block.
- Capacity misses: Occur because the cache has a finite size and cannot hold all the blocks needed during execution.
- Conflict misses (Collision): Occur in non-fully associative caches due to competition for entries in a specific set.

Effect of Associativity: Increasing associativity generally decreases conflict misses. However, this improvement comes with a trade-off: higher associativity increases access time and hardware cost.

[group 6]

6. True or false

(A) The page table is stored in the main memory.
(B) It is possible that cache hits, TLB hits, and page table misses.
(C) Using bounds register to limit table size, and add more if exceed is one of the possible solutions to handle huge page table.
(D) We detect the page faults in software and handle them in hardware.

Ans:

(A) (C)
(B) false, it's impossible, not in TLB if page not in memory
(D) false, we detect the page faults in hardware and handle the page faults in software.

[group 8]

7. Question:

When the design of memory hierarchy changes it might have some effect, please fill the following table.

design	effect on miss rate	possible effects
size increase		
associativity increase		
block size increase		

Ans:

design	effect on miss rate	possible effects
size increase	capacity miss decrease	access time increase
associativity increase	conflict miss decrease	access time increase
block size increase	spatial locality increase	miss penalty increase

[group 11]

8. Question:

Determine whether the following statements are true or false. If false, provide an explanation.

a. Page faults occur when the required page is not in physical memory.

b. Demand paging loads all pages of a process into memory at once.

c. FIFO (First-In-First-Out) is a page replacement policy that replaces the oldest page in memory.

d. A process can continue execution even when its required page is not in memory.

Ans:

a. True

b. False. Demand paging only loads pages into memory as they are accessed, reducing memory usage and startup latency. This policy is specifically called the "demand load policy."

c. True

d. False. When a page fault occurs, the process cannot proceed until the operating system loads the required page into memory. Execution resumes only after the fault is handled.

[group 13]

9. True or False:

A. A page fault can take millions of cycles to process (huge miss penalty)
B. Pages should be fairly small (Ex 512GB) to amortize high access time
C. Because disk writes take millions of cycles, write-through is practical and is preferred over write-back.
D. If a page is not present in memory, the PTE stores the physical page number plus status bits like reference/dirty.
E. On a TLB miss when the page is in memory, the system can load the PTE from the page table in memory and retry, and this can be handled in hardware or software.

Ans:

A. True
B. False -> 4KB
C. False (Write-through is impractical use write-back)
D. False (That is the case when the page is present. If not, PTE can refer to swap space on disk)
E. True

[group 14]

10. Question:

Given the table below, calculate the total page table size for a system running 5 applications.

Virtual address size	Page size	Page table entry size
32 bits	4 KB	4 bytes

Ans:

32 – 12 = 20
2^20 * 4 bytes = 4 MB (page table size)
4 MB * 5 = 20 MB

[group 10]

11. True or False

A. The page table is stored in the main memory.

B. n-way and full associativity negate the possibility of conflict misses.

C. Array of page table entries, indexed by page offset.

D. As the TLB miss occurs, if the page is not in the memory, it can be fixed by hardware method.

Ans:

A. True.
B. False. Only for full associativity.
C. False. Array of page table entries, indexed by virtual page number.
D. False. It can only be solved by software method which OS will handle fetching the page and updating the page table.