CS 465 – Final Review Questions

Fall 2021

Prof. Daniel Menasce

Question

- What are the types of hazards in a datapath and how each of them can be mitigated?

Answer

- What are the types of hazards in a datapath and how each of them can be mitigated?
  - Data hazard (forwarding)
  - Structural hazard (more functional units; e.g., separate instruction and data caches)
  - Control hazard (branch prediction, determining branch outcome at the ID stage).

Question

- State and explain some of the methods used to deal with branch prediction.

Answer

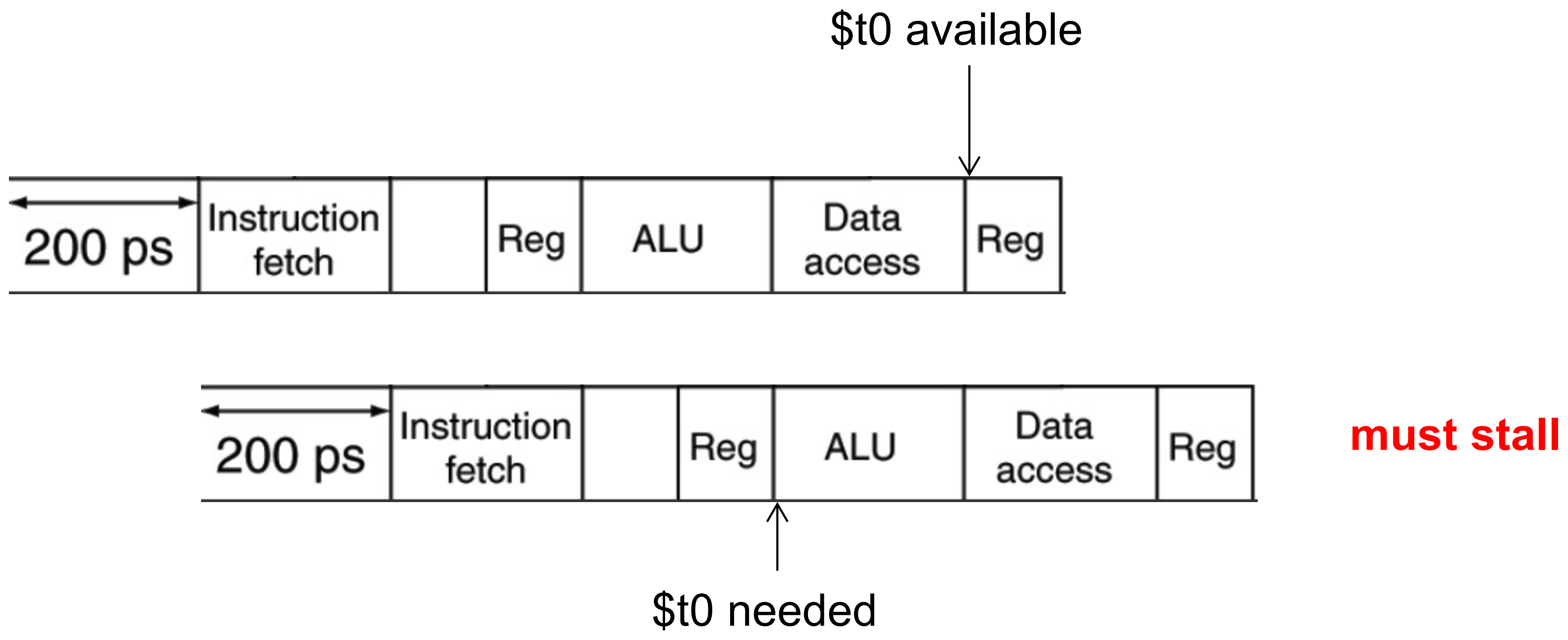
- State and explain some of the methods used to deal with branch prediction.
  - Static (assume taken/not taken) and dynamic (1-bit, 2-bit)

Does the following code sequence must stall, can avoid stalls using forwarding only, or can execute without stalling or forwarding?

- lw $t0, 0($t0)
- add $t1, $t0, $t0

Does the following code sequence must stall, can avoid stalls using forwarding only, or can execute without stalling or forwarding?

- lw $t0, 0($t0)
- add $t1, $t0, $t0



Question

- What is the purpose of the pipeline registers?
- What are the names given to these registers?
- Explain how information stored in these registers changes along the pipeline
- How are these registers used to detect data hazards?

Question

- What is the purpose of the pipeline registers?

Answers

- What is the purpose of the pipeline registers?
  - To store the information needed by each instruction to execute each stage of the pipeline

Question

- What are the names given to these registers?

Answers

- What are the names given to these registers?
  - IF/ID, ID/EX, EX/MEM, and MEM/WB

Answers

- Explain how information stored in these registers changes along the pipeline

.

Answers

- Explain how information stored in these registers changes along the pipeline
  - Information from the control unit is first stored in the ID/EX register and is divided EX, Mem, WB. At each subsequent stage, Mem and WB are passed long and then only WB.

Answers

- How are these registers used to detect data hazards?

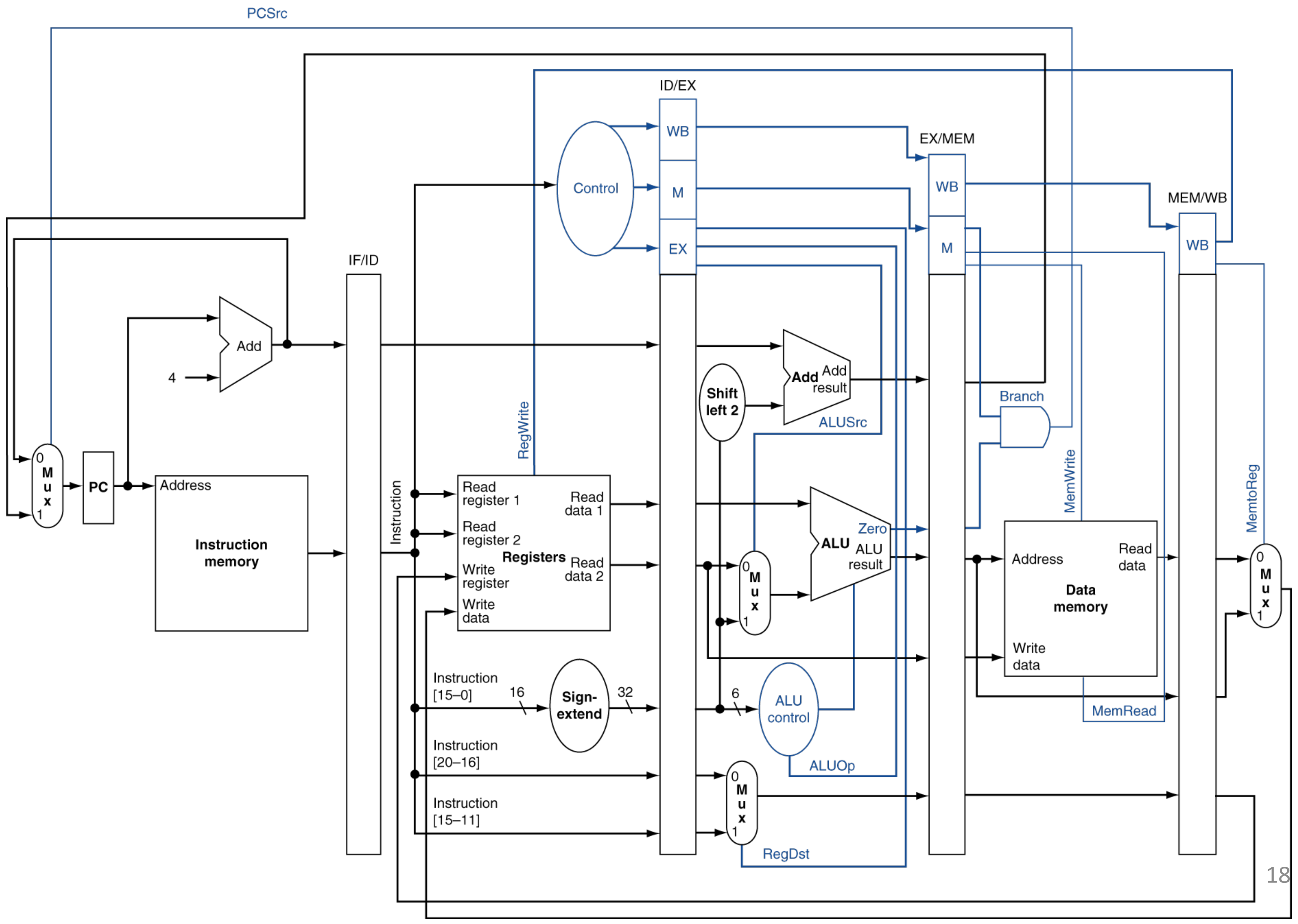
Answers

- How are these registers used to detect data hazards?
  - By comparing the Rd with the Rt/Rs register numbers across pipeline registers.

Answers

- What is the purpose of the pipeline registers?
  - To store the information needed by each instruction to execute each stage of the pipeline
- What are the names given to these registers?
  - IF/ID, ID/EX, EX/MEM, and MEM/WB
- Explain how information stored in these registers changes along the pipeline
  - Information from the control unit is first stored in the ID/EX register and is divided EX, Mem, WB. At each subsequent stage, Mem and WB are passed long and then only WB.
- How are these registers used to detect data hazards?
  - By comparing the Rd with the Rt/Rs register numbers across pipeline registers.

Explain This



Question

- Which of these elements influence the execution time of a program and why?
  - The level 1 cache hit rate
  - The hit rate to the TLB
  - The number of instructions handled per stage
  - The page fault rate
  - The ISA
  - The clock cycle duration

Answer ALL

- Which of these elements influence the execution time of a program?
  - The level 1 cache hit rate: reduces access to RAM
  - The hit rate to the TLB: reduces access to Page Table, which is in RAM
  - The number of instructions handled per stage: increases the throughput
  - The page fault rate: page faults require access to the paging disk
  - The ISA: influences number of instructions executed by a program, compiler design, and ISA implementation
  - The clock cycle duration: proportional to CPU time

Question

- What are all the levels of a memory system hierarchy?
- How do size, access time, and cost/bit vary along the hierarchy?
- What property of programs is important for the performance of the memory system hierarchy?

Answer

- What are all the levels of a memory system hierarchy?
  - Registers, caches, main memory, secondary storage

Answer

- How do size, access time, and cost/bit vary along the hierarchy?
  – Size and access time increase going down and cost/bit decreases going down

Answer

- What property of programs is important for the performance of the memory system hierarchy?
  - Space and temporal locality

Question

- A computer system takes 1,000 days to fail on average and the time to repair is on average one day. What is its availability?

Answer

- A computer system takes 1,000 days to fail on average and the time to repair is on average one day. What is its availability?
- Availability = MTTF / (MTTF + MTTR) =
  - 1,000 / (1,000 + 1) = 0.999

Question

- Explain the Principle of Locality and discuss where it is applied and what it achieves.

Answer

- Explain the Principle of Locality and discuss where it is applied and what it achieves.
- Spatial locality: items close to one just accessed are more likely to be accessed again

.

Answer

- Explain the Principle of Locality and discuss where it is applied and what it achieves.

- Temporal locality: items accessed recently are more likely to be accessed again soon.

.

Question

- What are the types of cache organization?
- Explain for each: block search, block placement, block replacement?
- Explain the two cache hit on write policies: write-through and write back (include write buffer)
- Explain approaches used for cache miss on write policies

Question

- What are the types of cache organization?

Answer

- What are the types of cache organization?
  - Direct mapped (DM), set associative (SA), fully associative (FA)

Question

- Explain for each: block search, block placement, block replacement?

Answer

- Explain for each: block search, block placement, block replacement?
  - DM: search one cache entry only; SA: search all entries in a set; FA: search all cache entries
  - DM: at a specific cache entry; SA: any empty location in a set; FA: any empty location in the cache
  - DM: only one option; SA: LRU/Random within set; FA: LRU/Random within the cache

Question

- Explain the two cache hit on write policies:

Answer

- Explain the two cache hit on write policies: write-through and write back (includes write buffer)

Question

- Explain approaches used for cache miss on write policies

Answer

- Explain approaches used for cache miss on write policies
  - Allocate on miss (fetch block and overwrite portion of block) and write-around (do not fetch block)

Question

- Consider that a physical address is 32 bits and that a direct mapped cache has 1024 entries. Each block in the cache has four 32-bit words. How many bits are used for the tag, cache index, and byte offset in the address?

Answer

- Consider that a physical address is 32 bits and that a direct mapped cache has 1024 entries. Each block in the cache has four 32-bit words. How many bits are used for the tag, cache index, and byte offset in the address?
- Cache index: 10 bits are needed to address 2^10 entries
- Byte offset: each block has 4 * 4 = 16 bytes. So, need 4 bits to address 2^4 byes.
- Tag: 32 – (10+4) = 18 bits

Question

- Explain the three sources of misses known as the 3 C’s: compulsory, capacity, and conflict.

Answer

- Explain the three sources of misses known as the 3 C’s: compulsory, capacity, and conflict.
- Compulsory: happens at the first reference to a block (cold start)
- Capacity:
- Conflict:

Answer

- Explain the three sources of misses known as the 3 C’s: compulsory, capacity, and conflict.
- Compulsory: happens at the first reference to a block (cold start)
- Capacity: a previously cached entry is evicted from the cache to make room for another
- Conflict:

Answer

- Explain the three sources of misses known as the 3 C’s: compulsory, capacity, and conflict.
- Compulsory: happens at the first reference to a block (cold start)
- Capacity: a previously cached entry is evicted from the cache to make room for another
- Conflict: various blocks can only be placed in one particular location (entry or set).

Question

- A direct mapped cache has 128 entries and each entry holds one block of 64 bits. What is the cache entry for a block that is stored in memory starting at byte 1152?

Answer

- A direct mapped cache has 128 entries and each entry holds one block of 64 bits. What is the cache entry for a block that is stored in memory starting at byte 1152?
- Size of a block = 64 bits / (8 bits/byte) =

  8 bytes
- The number of block starting at byte 1152 is 1152 / 8 = 144.
- Cache index = 144 mod 128 = 16

Question

- A 4-way set associative cache has 64 sets and each entry holds four blocks of 64 bits. What is the set entry for a block that is stored in memory starting at byte 1152?

Answer

- A 4-way set associative cache has 64 sets and each entry holds four blocks of 64 bits. What is the set entry for a block that is stored in memory starting at byte 1152?

- Number of the block starting at byte 1152 = 1152 /8 = 144.
- Set number where block 144 should be = 144 mod 64 = 16

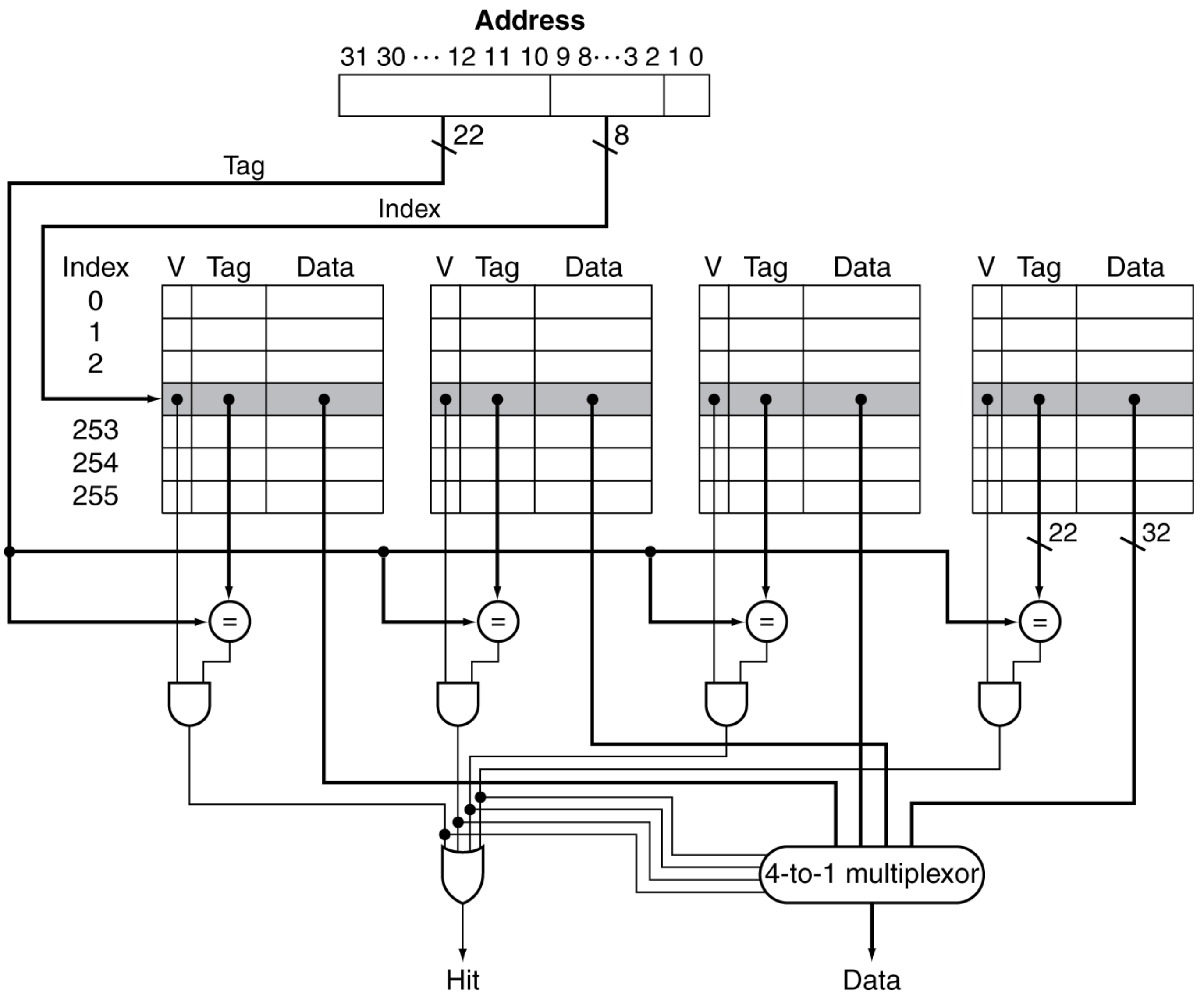
Question

- Which of the following are true?
  - A. All cores share a single L1 cache
  - B. All cores share a single L2 cache
  - C. Each core has its own page table register
  - D. All cores share a single L3 cache
  - E. The TLB is stored in main memory
  - F. The page table needs to be accessed for every memory access

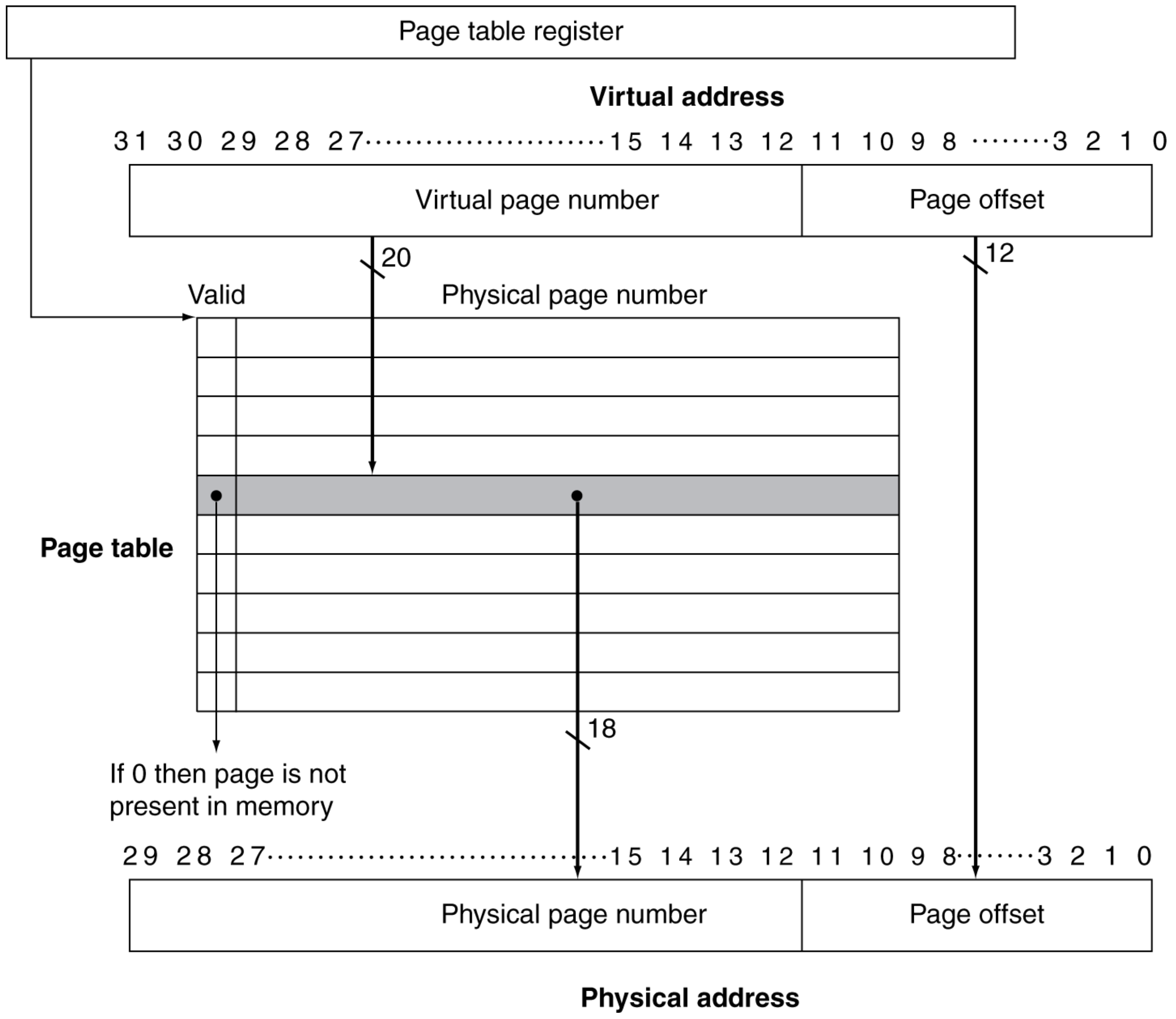
Answer: C and D

- Which of the following are true?
  - A. All cores share a single L1 cache
  - B. All cores share a single L2 cache
  - C. Each core has its own page table register
  - D. All cores share a single L3 cache
  - E. The TLB is stored in main memory
  - F. The page table needs to be accessed for every memory access

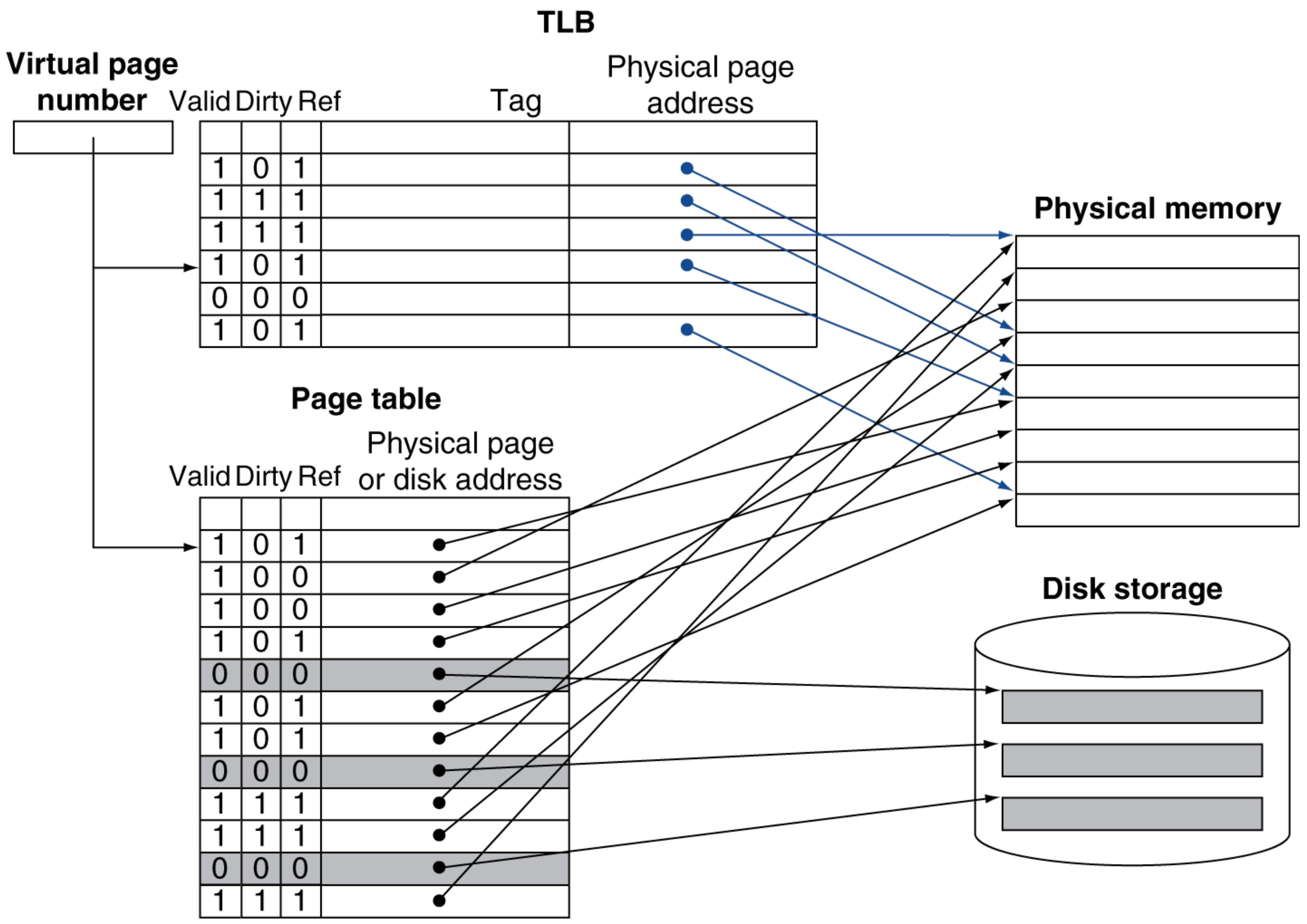
Explain This



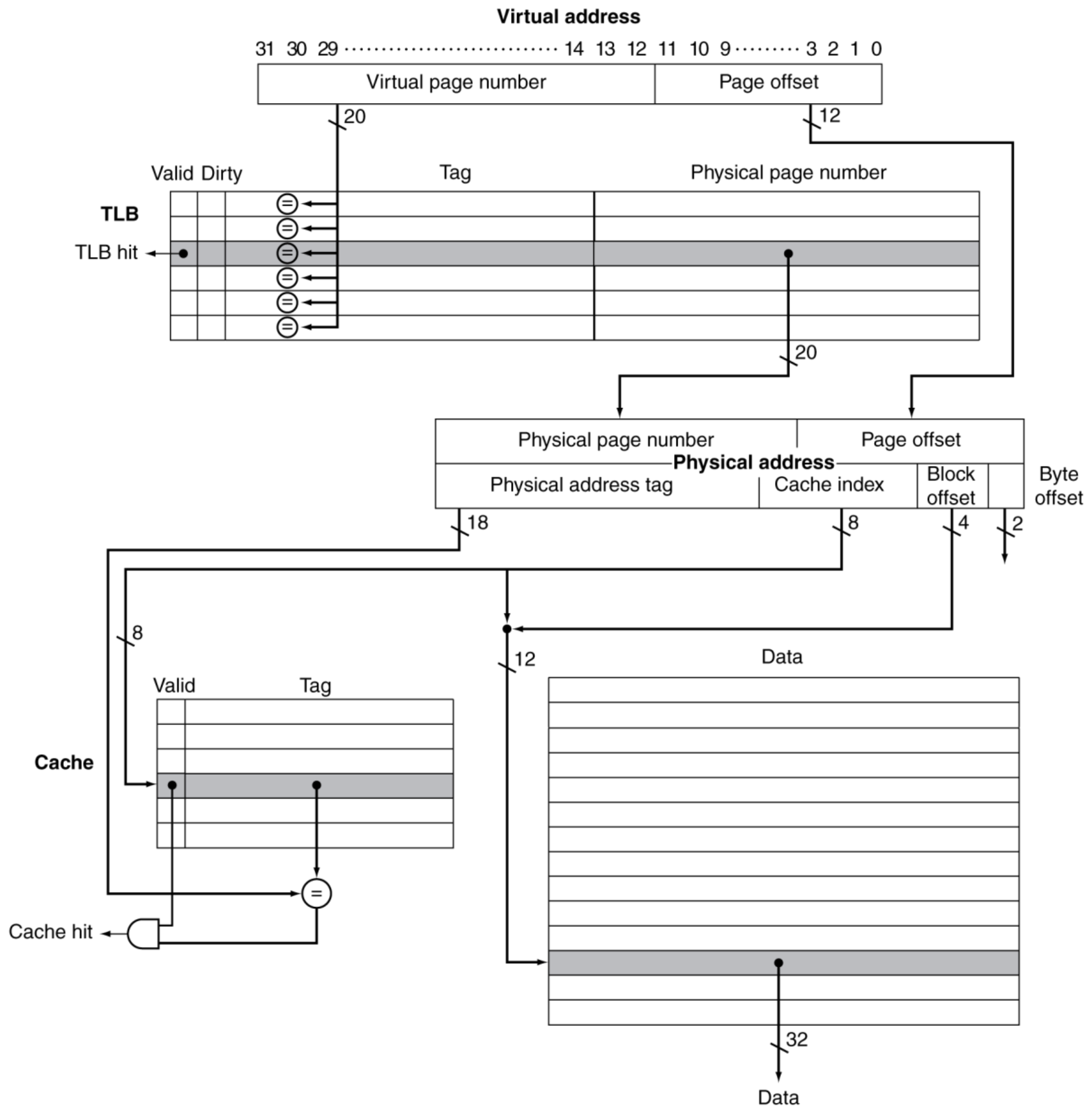
Explain This



Explain This



Explain This



Question

Consider a 4-way set associative cache with blocks of 4 words of 4 bytes each. The cache size is 256 K bytes. How many bits in the address are used for the tag, cache index, and block/word/byte?

Answer

Consider a 4-way set associative cache with blocks of 4 words of 4 bytes each. The cache size is 256 K bytes. How many bits in the address are used for the tag, cache index, and block/word/byte?

Each block has 16 bytes.

Each set has 4 blocks. So, each set has 64 bytes

The number of sets in the cache is the size of the cache divided by the size of a set => $2^8 * 2^{10} / 2^6 = 2^{12}$

Thus, we need 12 bits to index a set a the cache

Index= 12 bits

Block has 16 bytes: need 4 bits to address a byte in a block

Tag = 32 – (12+4) = 16 bits

Question

- A machine supports virtual memory and page sizes are 4 Kbytes long. A physical address is 32 bits long. How many page frames are there in main memory?

Answer

- A machine supports virtual memory and page sizes are 4 Kbytes long. A physical address is 32 bits long. How many page frames are there in main memory?
  - $2^{32} / 2^{12} = 2^{20} = 1024 * 1024$ page frames

Question

- A machine supports virtual memory and page sizes are 4 Kbytes long. A physical address is 32 bits long. How many page frames are there in main memory?

- Suppose that a virtual address is 34 bits long.
  - What is the size of the virtual address space in pages?

Answer and Question

- A machine supports virtual memory and page sizes are 4 Kbytes long. A physical address is 32 bits long. How many page frames are there in main memory?
  - $2^{32} / 2^{12} = 2^{20} = 1024 * 1024$ page frames
- Suppose that a virtual address is 34 bits long.
  - What is the size of the virtual address space in pages?
    - $2^{34} / 2^{12} = 2^{22}$ virtual pages
  - How many entries are there in the page table?

Answer and Question

- A machine supports virtual memory and page sizes are 4 Kbytes long. A physical address is 32 bits long. How many page frames are there in main memory?
  - $2^{32} / 2^{12} = 2^{20} = 1024 * 1024$ page frames
- Suppose that a virtual address is 34 bits long.
  - What is the size of the virtual address space in pages?
    - $2^{34} / 2^{12} = 2^{22}$ virtual pages
  - How many entries are there in the page table?
    - 2^{22}
  - Estimate the size of the page table in MB.

Answer

- A machine supports virtual memory and page sizes are 4 Kbytes long. A physical address is 32 bits long. How many page frames are there in main memory?
  - $2^{32} / 2^{12} = 2^{20} = 1024 * 1024$ page frames
- Suppose that a virtual address is 34 bits long.
  - What is the size of the virtual address space in pages?
    - $2^{34} / 2^{12} = 2^{22}$ virtual pages
  - How many entries are there in the page table?
    - 2^{22}
  - Estimate the size of the page table in MB.
    - $(2^{22} * (20 + 1 + 1 + 1))$ bits $= 2^{22} * (23/8) / 2^{20}$ MB = 11.5 MB

Question

- Consider the following table.

Memory element	Access time	Hit rate	Miss Penalty
L1 Cache	1 cycle	85%	5 cycles
L2 cache	5 cycles	99%	100 cycles
Memory	100 cycles	100%	

  – Compute the Average Memory Access Time in cycles

Answer

- Consider the following table.

Memory element	Access time	Hit rate	Miss Penalty
L1 Cache	1 cycle	85%	5 cycles
L2 cache	5 cycles	99%	100 cycles
Memory	100 cycles	100%	

  – Compute the Average Memory Access Time in cycles
    - 1 + 0.15 *(5 + 0.01 * 100) = 1.9 cycles

Question

- How is write-on-hit handled in virtual memory (write-through or write-back) and why?

Answer

- How is write-on-hit handled in virtual memory (write-through or write-back) and why?
  - Write back because it takes too long to write to disk (approximately one million cycles)

Question

- What is the typical page replacement policy used in virtual memory and how it is implemented?

Answer

- What is the typical page replacement policy used in virtual memory and how it is implemented?
  - LRU, implemented using the Reference Bit and Dirty bit

Question

- Consider a VM system that uses an approximate LRU replacement scheme for main memory whose Page Table is

	Valid bit	Reference bit	Dirty bit	Page frame number	Disk address
0	1	0	0	2	----
1	0	1	1	----	PF:3654
2	1	1	1	1	---
3	1	1	0	0	---
4	1	1	1	3	---

- Does a reference to virtual page 1 cause a page fault?

Answer

- Consider a VM system that uses an approximate LRU replacement scheme for main memory whose content is

	Valid bit	Reference bit	Dirty bit	Page frame number	Disk address
0	1	0	0	2	----
1	0	1	1	----	PF:3654
2	1	1	1	1	---
3	1	1	0	0	---
4	1	1	1	3	---

- Does a reference to virtual page 1 cause a page fault? Yes

Question

- Consider a VM system that uses an approximate LRU replacement scheme for main memory whose Page Table is

	Valid bit	Reference bit	Dirty bit	Page frame number	Disk address
0	1	0	0	2	----
1	0	1	1	----	PF:3654
2	1	1	1	1	---
3	1	1	0	0	---
4	1	1	1	3	---

- If main memory only has space for 4 pages, which page would you replace to bring virtual page 1 into memory?

Answer

- Consider a VM system that uses an approximate LRU replacement scheme for main memory whose content is

	Valid bit	Reference bit	Dirty bit	Page frame number	Disk address
0	1	0	0	2	----
1	0	1	1	----	PF:3654
2	1	1	1	1	---
3	1	1	0	0	---
4	1	1	1	3	---

- If main memory only has space for 4 pages, which page would you replace to bring virtual page 1 into memory?
  – Virtual page 3 because the reference bit is 1 but has not been modified (dirty bit is zero) or virtual page 0 because its reference bit is zero.

Question

- A program has to execute **A** arithmetic operations that take **t** seconds each. Ten percent of them cannot be executed in parallel. Give an expression for the program's execution time when executed on **p** processors. What is the minimum possible execution time of this program?

Answer

- A program has to execute **A** arithmetic operations that take **t** seconds each. Ten percent of them cannot be executed in parallel. Give an expression for the program's execution time when executed on **p** processors. What is the minimum possible execution time of this program?
- Execution time = 0.1 A * t + 0.9 * A * t / p
- Minimum execution = 0.1 A * t (limit when p -> ∞)