Review Questions

1. F Increasing the size of a cache results in lower miss rates and higher performance.
2. F For a given capacity and block size, a set-associative cache implementation will typically have a lower hit time than a direct-mapped implementation.
3. T Memory buses are usually picked based on the speed whereas the I/O buses are primarily adopted based on the compatibility (industry standards) and cost.
4. F Asynchronous buses cannot be long due to clock skew restrictions.
5. F In a write-through cache, a read miss can cause a write to the lower memory level.
6. F SRAMs must be refreshed periodically to prevent loss of information.
7. F Magnetic disks are volatile storage devices.
8. T An asynchronous bus is not clocked.
9. F Victim caches decrease miss penalty while they increase miss rate.
10. T Direct-mapped cache of size N has same miss rate as 2-way set associative of size N/2.
11. F The main difference between DRAM (the technology used to implement main memory) and SRAM (the technology used to implement caches) is that DRAM is optimized for access speed while SRAM is optimized for density.
12. T If the I/O devices are connected to the CPU through main memory busses, the performance of the processor may decrease since I/O commands could interfere with CPU memory accesses.
13. T Memory buses are usually picked based on the speed whereas the I/O buses are primarily adopted based on the compatibility (industry standards) and cost.
14. F In a write-through cache, a read miss can cause a write to the lower memory level.
15. T Conflict misses do not occur in fully set-associative cache memories.
16. F Virtually addressable caches would have less hit time than physically addressable cache memories.
17. T TLB misses in a virtual memory system can occur and can be handled by software.
18. F In a write-back cache, a read miss always causes a write to the lower memory level.
19. T Memory interleaving is a technique for reducing memory access time through increased bandwidth utilization of the data bus.
20. F Increasing the size of a cache results in lower miss rates and higher performance.
21. F For a given capacity and block size, a set-associative cache implementation will typically have a lower miss penalty than a direct-mapped implementation.
22. T Memory buses are usually picked based on the speed whereas the I/O buses are primarily adopted based on the compatibility (industry standards) and cost.

23. Consider a memory system with a two-level cache with the following characteristics: The miss penalty from L2 cache to the main memory is 100 clock cycles, the hit time of the L2 cache is 10 clock cycles, the hit time of L1 is 1 clock cycle, and both cache memories have an average hit rate of 90%. What is the average memory access time?

    a. 12 clock cycles

    b. 34 clock cycles

    c. 30 clock cycles

    d. None of the above

24. Assume a 64KB cache with 16-byte block size and a 32-bit physical address. If a block has 16 tag bits, what is the type of this cache?

    a. Direct mapped

    b. 2-way set associative

    c. Fully associative

    d. None of the above

25. For the following 64-bit memory address references, identify the binary word address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. List whether each reference is a hit or a miss, assuming the cache is initially empty.

Word Address	Binary Address	Tag	Index	Hit/Miss
0x03	0000 0011	0	3	M
0x02	0000 0010	0	2	M
0xba	1011 1010	b	a	M

26. For the following 64-bit memory address references, identify the binary word address, the tag, the index, and the offset given a direct-mapped cache with two-word blocks and a total size of eight blocks.

Word Address	Binary Address	Tag	Index	Offset	Hit/Miss
0x03	0000 001 1	0	1	1	M
0x02	0000 001 0	0	1	0	H
0xba	1011 101 0	b	5	0	M

27. Calculate the total number of bits required to implement a 32 KiB cache with two-word blocks. Assume that this cache is byte addressable, and that addresses and words are 64 bits.

    Each word is (64/8=) 8 bytes; each block contains two words; each block contains 16 = 2^4B.

    The cache contains 32KiB = 2^15 bytes of data. Thus, it has 2^15/2^4 = 2^11 lines of data.

    Each 64-bit address is divided into:

    (1) a 3-bit word off set, (2) a 1-bit block off set, (3) an 11-bit index (because there are 2^11 lines), and (4) a 49-bit tag (64 − 3 – 1 − 11 = 49).

    The cache is composed of: 2^15 * 8 bits of data + 2^11*49 bits of tag + 2^11*1 valid bits

    ➔ 364,544 bits. Total size is 364,544 bits = 45,568 bytes.

    A cache is named according to the amount of data it contains (i.e., a 4 KiB cache can hold 4 KiB of data); however, caches also require SRAM to store metadata such as tags & valid bits.

28. For a direct-mapped cache design with a 64-bit address, the following bits of the address are used to access the cache.

Tag	Index	Offset
63–10	9–5	4–0

    Each cache block consists of four 8-byte words (=32 bytes). ➔ The total off set is 5 bits.

    Three of those 5 bits is the word offset (the offset into an 8-byte word).

    The remaining two bits are the block offset. Two bits allows us to enumerate 2^2 = 4 words.

    There are five index bits. This tells us there are 2^5 = 32 lines in the cache.

29. Given the below parameters, calculate the maximum possible page table size for a system running five processes.

Virtual Address Size	Page Size	Page Table Entry Size
32 bits	8 KiB	4 bytes

    The tag size is 32–log2 (8192) = 32–13 = 19 bits. All five page tables would require 5 × (2^19 × 4) bytes = 10 MB.

30. A four-set associative cache-memory system consists of 32 blocks of cache. Each block holds 256 bytes of data. Each block includes a control field with two LRU bits, a valid bit and a dirty bit. The address space is 4 gigabytes. Calculate the total size of the cache, including all control, data and tag bits.

    Total $ size = Data + Control bits + Tag;

    For each block: Control bits = LRU bits + Valid bit + Dirty bit = 2 + 1 + 1 = 4 b

    Address = Tag + Index + Offset; 256 = 28, then Offset = 8;

    # of blocks in each direct mapped cache = 32/4 = 23, Index = 3; Tag = 32 – 8 – 3 = 21 bits.

    Total $ size = 32 blocks x (4 bits + 21 bits + 256 x 8 bits) = 66,336 bits = 8,292 B.

31. Consider a system with a two-level cache having the following characteristics:

    L1 Cache: Physically addressed; L1 hit time is 2 clock cycles; L1 average miss rate is 0.15

    L2 Cache • Physically addressed; L2 hit time is 5 clocks (after L1 miss); L2 average miss rate is 0.05

    If L2 miss takes 50 clock cycles, compute the average memory access time in clocks for the given 2-level cache system.

    AMAT = HT1 + MR1 x (HT2 + MR2 x MP2) = 2 + 0.15 x (5 + 0.05 x 50) = 3.125 clock cycles.

32. Assume that a computer's address size is k bits, the cache size is S bytes, the block size is B bytes, and the cache is A-way set-associative. Suppose that B = 2b. Please find the followings in terms of S, B, A, b, and k: the number of sets in the cache and the number of index bits in the address.

    Address size is k bits, cache size is S bytes/cache, block size is B = 2b bytes/block, and associativity is A blocks/set. The number of sets in the cache (X) can be can be defined, as follows:

    X = Sets/Cache = (Bytes/cache)/[(Blocks/set) x (Bytes/block)] = S/(AB)

    The number of address bits needed to index a particular set of the cache can be found, as follows:

    Cache set index bits = log2 (Sets/cache) = log2 (S/(AB)) = log2 (S/A) – b

33. The following list provides parameters of a virtual memory system.

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size (byte)
43	16 GiB	4 KiB	4

For a single-level page table:

Worst case: 2^(43 − 12) = 2^31 entries (# of pages)

➔ Requiring 2^(31) × 4 bytes = 2^33 = 8 GB.

Using a multi-level page table can reduce the physical memory consumption of page tables by only keeping active PTEs in physical memory. How many levels of page tables will be needed if the segment tables (the upper-level page tables) are allowed to be of unlimited size? How many memory references are needed for address translation if missing in TLB?

With only two levels, the designer can select the size of each page table segment. In a multi-level scheme, reading a PTE requires an access to each level of the table.

34. Assume that main memory accesses take 70 ns and that 36% of all instructions access data memory. The following table shows data for L1 caches attached to each of two processors, P1 and P2.

	L1 Size	L1 Miss Rate	L1 Hit Time
P1	2 KiB	8.0%	0.66ns
P2	4 KiB	6.0%	0.90 ns

Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?

P1	1.515 GHz
P2	1.11 GHz

What is the Average Memory Access Time for P1 and P2 (in cycles)?

P1	6.31 ns	9.56 cycles
P2	5.11 ns	5.68 cycles

For **P1** all memory accesses require at least one cycle (to access L1).

8% of memory accesses additionally require a 70 ns access to main memory.

This is 70/0.66 = 106.06 cycles. However, we can't divide cycles; therefore, we must round up to 107 cycles.

Thus, the Average Memory Access time is 1 + 0.08*107 = **9.56 cycles**, or **6.31 ps.**

For **P2**, a main memory access takes 70 ns. This is 70/0.66 = 77.78 cycles.

Because we can't divide cycles, we must round up to 78 cycles. Thus the Average Memory Access time is 1 + 0.06*78 = **5.68 cycles**, or **6.11 ps**.

35. Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster? (When we say a "base CPI of 1.0", we mean that instructions complete in one cycle, unless either the instruction access or the data access causes a cache miss.)

P1	12.64 CPI	8.34 ns per inst
P2	7.36 CPI	6.63 ns per inst

For P1, every instruction requires at least one cycle. In addition, 8% of all instructions miss in the instruction cache and incur a 107-cycle delay.

Furthermore, 36% of the instructions are data accesses.

8% of these 36% are cache misses, which adds an additional 107 cycles.

1 + 0.08*107 + 0.36*0.08*107 = **12.64.** With a clock cycle of 0.66 ps, each instruction requires **8.34 ns**.

Using the same logic, we can see that P2 has a CPI of **7.36** and an average of only **6.63 ns/instruction**.