1) **Virtual memory is an imaginary memory.** It is an alternate set of memory addresses. It allows programmers to use a very large range of memory for stored data. TLB is a special cache keeping address translations so that a memory access rarely requires a second access to translate the data. Advantages of VM include translation, protection, and sharing. Block replacement policies include LRU, FIFO, and random. DMA gives external device ability to write memory directly: much lower overhead than having processor request one word at a time.

2) **The original motivation for using virtual memory was "compatibility". What does that mean in this context? What are other motivations for using virtual memory?**
Compatibility in this context means the ability to transparently run the same piece of (un-recompiled) software on different machines with different amounts of physical memory. This compatibility freed the application writer from worrying about how much physical memory a machine had. The motivation for using virtual memory today have mainly to do with multiprogramming, the ability to interleave the execution of multiple programs simultaneously on one processor. Virtual memory provides protection, relocation, fast startup, sharing and communication, and the ability to memory map peripheral devices.

3) **Bus is a shared communication link.** It is used to connect multiple subsystems. Buses are important technique for building large scale systems. Buses have advantages like versatility and low cost. On the other hand, they generate a communication bottleneck and their maximum speed is limited by the length of the bus and the number of devices on the bus. Types of buses are processor-memory bus, I/O bus, backplane bus.

4) **TLB misses can be handled in software, or hardware can be used to handle them.** We can expect software to be slower due to the overhead of a context switch to the handler code, but the sophistication of the replacement algorithm can be higher for software and a wider variety of virtual memory organizations can be readily accommodated. Hardware should be faster, but less flexible. Floating-point programs often traverse large data structures and thus more often reference a large number of pages. It is thus more likely that the TLB will experience a higher rate of capacity misses.

5) **What is the principle behind RAID? What is RAID level 0 and why is it widely used.**
A set of physical disk drives viewed by the OS as a single logical drive. Replace large-capacity disks with multiple smaller-capacity drives to improve the I/O performance (at lower price). Data are distributed across physical drives in a way that enables simultaneous access to data from multiple drives. . Redundant disk capacity is used to compensate for the increase in the probability of failure due to multiple drives. RAID level 0 refers to no redundancy and it is normal disk usage. Data are stripped across the available disks. Many personal computers and other systems do not use RAID and so fall under RAID level 0.

6) **What is the reason for using combination of first- and second- level caches rather than using the same chip area for a larger first-level cache?**
The ultimate metric for cache performance is average access time: *tavg = thit + miss-rate* tmiss*. Multiple levels of cache are used because not all of the performance factors can be optimized in a single cache. Specifically, with *tmiss* (memory latency) given, it is difficult to design a cache which is both fast in the common case (a hit) and minimizes the costly uncommon case by having a low miss rate. These two design goals are achieved using two caches. The first level cache minimizes the hit time; therefore it is usually small with a low-associativity. The second level cache minimizes the miss rate; it is usually large with large blocks and a higher associativity.

7) **Length of a bit in a 1-meter copper wire on a 1-Gbps, where the speed of propagation is 2.3x10^8 m/s:** The width of a bit on a 1-Gbps link is 1/109(bit/sec) = 10^{-9} sec/bit. Length of a bit = 10^{-9} (sec/bit) x 2.3x10^8 (meter/sec) = 0.23 meter/bit. One meter cable can have 1/0.23 = 4 bits.

8) **What is the average time to write or read a 32 KB block to a magnetic disk with the following properties? Rotational speed: 7,200 RPM; Transfer rate: 45 MB per second.; The advertised average seek time is 5 ms and it is three times longer than the measured seek time.; Controller overhead: 0.25 ms.**
Average Seek Time (measured) = 5ms/3. Rotational latency: 0.5 * 1/(7200/60) / 1000 = 4.167ms
5/3ms + 4.167ms + 32KB/45MB + 0.25ms = 6.8ms.

9) **Consider a virtual memory system with the following properties: 64-bit virtual byte address; 16-KB pages; 32-bit physical byte address. What is the total size of page table on this machine, assuming that the valid, protection, dirty, and use bits take a total of 4 bits and that all the virtual pages are in use. Please list and explain ways to implement this virtual memory system.**

Page size is 16KB (2^{14}). So, 14 bits are required for page offset. Size of virtual page number = 64 – 14 = 50 bits. Size of physical page number = 32 – 14 = 18 bits. The number of entries of a page table is equal to the number of pages in the virtual address space. (2^{64} bytes / 16 KB) = (2^{64} bytes / $2^4 \times 2^{10}$ bytes) = 2^{50}. The width of each entry is the number of control bits plus the number of bits for physical page number: 4 + 18 = 22 bits. So, the page table contains (22×2^{50}) bits.

10) **Consider a virtual memory system with the following properties: 40-bit virtual byte address; 16-KB pages; 36-bit physical byte address. What is the total size of page table, assuming that the valid, protection, dirty, and use bits take a total of 4 bits and that all the virtual pages are in use?** (26×2^{26}) bits.

11) **The memory architecture of a machine X has the following characteristics: Virtual Address 54 bits; Page Size 16 K bytes; PTE Size 4 bytes. Assume that there are 8 bits reserved for the operating system functions (protection, replacement, valid, modified, and Hit/Miss- All overhead bits) other than required by the hardware translation algorithm. Derive the largest physical memory size (in bytes) allowed by this PTE format. Make sure you consider all the fields required by the translation algorithm.**

Since each Page Table element has 4 bytes (32 bits) and the page size is 16K bytes:
(1) we need $\log_2(16*2^{10})$, i.e., 14 bits to hold the page offset
(2) we need 32 – 8 (used for protection) = 24 bits for page number
The largest physical memory size is $2^{(14+24)}$ bytes = 2^{38} bytes = 256 GB

How large (in bytes) is the page table?

The page table is indexed by the virtual page number which uses 54 – 14 = 40 bits. The number of entries are therefore 2^{40}. Each PTE has 4 bytes. So the total size of the page table is 2^{42} bytes which is 4 terabytes.

Assuming 1 application exists in the system and the maximum physical memory is devoted to the process, how much physical space (in bytes) is there for the application's data and code.

The application's page table has an entry for every physical page that exists on the system, which means the page table size is $2^{24}*4$ bytes. This leaves the remaining physical space to the process: 2^{38} - 2^{26} bytes

12) **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 = 2^8, then Offset = 8; Number of blocks in each direct mapped cache = 32/4 = 2^3, 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.

13) **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.

14) **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 = $\log_2$ (Sets/cache) = $\log_2$ (S/(AB)) = $\log_2$ (S/A) – b

15) Please formulate the average disk access time and define each term. What do disks spend most of their time? What is the average time to write or read a 32 KB block to magnetic disks with the following properties? The disk rotates at 7,200 RPM and the transfer rate is 40 MB per second. The advertised average seek time is 4 ms and it is it is three times longer than the measured seek time. Assume that the controller overhead is 0.2 ms.

Ave. disk access time is the average time of between initiating a request and obtaining the first data character. Ave. disk access time is equal to average seek time + average rotational delay + transfer time + controller overhead. Controller time is the overhead the controller imposes in performing I/O access. The time for the requested sector to rotate under the head is the rotational delay. Transfer time is the time it takes to transfer of a block of bits. The time to move the arm to the desired track is called seek time. Disks spend most of their time waiting for the head to get over the data. ADAT = (4/3) + (0.5/7,200RPM) + (32KB/40MB/sec) + 0.2 ms = 1.33 + 4.17 + 0.8 + 0.2 = 6.5 ms.

16) What is the average time to read 1KB data from a disk? Assuming advertised seek time is 6ms, transfer rate is 20MB/sec, the disk rotates 6000 RPM, and the controller overhead is 0.1 ms. 7.15ms.

17) The table reflects reasonable up-to-date information. However, most of these technologies/standards are currently under continuous improvements.

	IDE/Ultra ATA	SCSI	PCI	PCI-X
Data Width	16 bits	8 bits (regular) 16 bits (Wide)	32 bits	64 bits (current) 32 bits (originally)
Clock Rate	Up to 100Mhz (started at 25mhz)	Up to 160Mhz (started at 5mhz)	33mhz and 66mhz	66, 100, 133mhz.
# of Bus Masters	1	Multiple	Multiple	Multiple
Bandwidth Peak	200MB/s	320MB/s	533MB/s	1066MB/s
Clocking	Asynchronous (old) Synchronous (latest)	Synchronous	Synchronous	Synchronous
Standard	ANSI standard ATA-1 through ATA-8	ANSI standard SCSI-1 through SCSI-3	PCI, PCI 1.0, 2.2, 2.3, and 3.0	PCI-X, PCI-X 1.0, and 2.0

18) Assume a 64KB cache with four-word block size (a word is 4 bytes) and a 32-bit address. If a block has 28 tag bits, what is the type of this cache?

a) Direct mapped c) 2-way set associative c) Fully associative

19) RAID 3 is unsuited to transactional processing because each read involves activity at all disks. In RAID 4 and 5 reads only involve activity at one disk. The disadvantages of RAID 3 are mitigated when long sequential reads are common, but performance never exceeds RAID 5. For this reason, RAID 3 has been all but abandoned commercially.

20) Term definitions:

- Cache • Virtual memory • Block replacement policies • AMAT and its components • Hit time • Miss penalty • Miss rate • Write policies • Set associative caches • Page table • RAID •••

- **Suppose we have two different I/O systems A and B. A has data transfer rate: 5KB/s and has access delay: 5 sec. While B has data transfer rate: 3 KB/s and has access delay: 4 sec. Now we have a 3M I/O request, taking performance into consideration, which I/O system will you use? What about for a 3KB request?**

3M request
Case 1: t (in sec) = 5 + (3 * 1024 * 1024) / (5 * 1024) = 5 + 614.4 = 619.4
Case 2: t (in sec) = 4 + (3 * 1024 * 1024) / (3 * 1024) = 4 + 1024 = 1028 So system 1 will be chosen.
3K request
Case 1: t (in sec) = 5 + (3 * 1024) / (5 * 1024) = 5.6
Case 2: t (in sec) = 4 + (3 * 1024) / (3 * 1024) = 5 So system 2 will be chosen.

21) What is the bottleneck in the following system setup, the CPU, memory bus, or the disk set?

- **The user program continuously performs reads of 64KB blocks, and requires 2 million cycles to process each block.**
- **The operating system requires 1 million cycles of overhead for each I/O operation.**
- **The clock rate is 3GHz.**
- **The maximum sustained transfer rate of the memory bus is 640MB/sec**
- **The read/write bandwidth of the disk controller and the disk drives is 64MB/sec, disk average seek plus rotational latency is 9ms.**
- **There are 20 disks attached to the bus each with its own controller. (Assume that each disk can be controlled independently and ignore disk conflicts.)**

Amount of time CPU takes to process each 64 KB block = $((2 * 10^6)/ (3 * 10^9)) * 10^3 = 0.67$ ms
Amount of time spent in memory transfer for 64 KB block = $((64 * 2^{10})/(640*2^{20}))*1000 = 0.097$ ms
Amount of time spent in I/O transfer for a 64 KB block = seek time + rot. delay + transfer time + controller overhead = $9 + ((64*2^{10})/(64*2^{20}))*10^3 + ((10^6 * 3)/10^9)*10^3 = 12.97$ ms. The main bottleneck is I/O in the above system.

22) You purchased an Acme computer with the following features:

- **95% of all memory accesses are found in the cache**
- **Each cache block is 2 words, and the whole block is read on any miss**
- **The processor sends references to its cache at the rate of 10^9 words per second**
- **25% of those references are writes**
- **Assume that the memory system can support 10^9 words per second, reads or writes**
- **The bus reads or writes a single word at a time (the memory system cannot read or write two words at once)**
- **Assume that any one time, 30% of the blocks in the cache have been modified**
- **The cache uses write allocate on a write miss**

You are considering adding a peripheral to the system, and you want to know how much of the memory system bandwidth is already used. Calculate the percentage of the memory system bandwidth used on the average if the cache is Write Through. Be sure to state your assumptions.

Write Through				
Access Type	Access Hits Cache?	Frequency	Memory Accesses Generated	
Read	Yes	95% x 75% = 71.25%	0	The cache contains the data in question➔ no need to generate a memory system access.
Read	No	5% x 75% = 3.75%	2	The cache fills the appropriate cache block from memory ➔ requires two memory system reads as each block is two words.
Write	Yes	95% x 25% = 23.75%	1	The cache must generate a memory access to update the word in memory being written to cache.
Write	No	5% x 25% = 1.25%	1	Write No Allocate: The word of data is written to main memory only (1 access). If it were 'write allocate' cache: First, fill the appropriate cache block from memory (2 memory accesses). Then, write the word of data to main memory (1 access). ➔ 3 accesses.

AccessesAvg = (71.25% x 0) + (23.75% x 1) + (3.75% x 2) + (1.25% x 1) = 0.325
Bandwidth Used = 0.325 x $(10^9)/10^9$ = 32.5%

23) Fill in the blanks:

- Three types of cache misses are.
- CPU time for a program can be expressed as Instruction Count * (__________________ + Misses Per Instruction * Miss Penalty) * Clock Cycle Time.
- One advantage of write-update protocol could be
- One advantage of write-invalidate protocol could be
- A solution to reduce hit time:

- A solution to reduce miss rate:
- A solution to reduce miss penalty:
- A solution for higher bandwidth for main memory:
- A strategy to select the block to replace on a cache miss:
- An advantage of bus-based I/O interconnection can be:
- A disadvantage of bus-based I/O interconnection can be:
- An alternative I/O interconnection to bus-based interconnection:
- When I/O components are considered we need alternative performance measures besides CPU time. The number of tasks completed per unit time is called

24) True or False.

- False Increasing the size of a cache results in lower miss rates and higher performance.
- False For a given capacity and block size, a set-associative cache implementation will typically have a lower hit time than a direct-mapped implementation.
- False For a given capacity and block size, a set-associative cache implementation will typically have a lower miss penalty than a direct-mapped implementation.
- True 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.
- False Asynchronous buses cannot be long due to clock skew restrictions.
- False ‘Polling’ is always a better mechanism than ‘interrupts’ for handling I/O operations on a network interface card.
- False In a write-through cache, a read miss can cause a write to the lower memory level.
- F SRAMs must be refreshed periodically to prevent loss of information.
- F Magnetic disks are volatile storage devices.
- T An asynchronous bus is not clocked.
- F Victim caches decrease miss penalty while they increase miss rate.
- T/F RAID 5 uses more check disks than RAID 3 for the same number of data disks.
- T Direct-mapped cache of size N has same miss rate as 2-way set associative of size N/2.
- 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.
- T/F 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.
- T/F Increasing the size of a cache results in lower miss rates and higher performance.
- 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.
- T/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.
- 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.
- T Asynchronous buses cannot be long due to clock skew restrictions.
- T/F ‘Polling’ is always a better mechanism than ‘interrupts’ for handling I/O operations on a network interface card.
- F In a write-through cache, a read miss can cause a write to the lower memory level.
- F RAID 5 can recover from a two-disk failure.
- F ATM has a variable message size and Ethernet has a fixed message size.
- F Loop-carried dependencies can be completely eliminated by a hardware mechanism at run-time.
- F The multi-cycle data path is always faster than the single-cycle data path.
- F Loops with intra-iteration dependencies can be executed in parallel.
- T Software Pipelining can be used on superscalar processors.
- False It is possible to eliminate all the stalls due to data dependencies with a special hardware mechanism.
- True Conflict misses do not occur in fully set-associative cache memories.

- True MIPS (million instructions per second) can be defined as $\frac{instruction\ count}{CPU\ time\ in\ seconds} \times \frac{1}{10^6}$.
- True For a given capacity and block size, a set-associative cache implementation will typically have a higher hit time than a direct-mapped implementation.
- False Page tables in virtual memory systems are placed on a special cache memory.
- False "Y is p% faster than X" can be defined as: $\frac{time\ of\ Y}{time\ of\ X} = 1 + \frac{p}{100}$
- False Virtually addressable caches would have less hit time than physically addressable cache memories.
- True TLB misses in a virtual memory system can occur and can be handled by software.
- False In a write-back cache, a read miss always causes a write to the lower memory level.
- False Branch history tables typically eliminate more stall cycles than branch target buffers.
- False Both DRAM and SRAM must be refreshed periodically using a dummy read/write operation.
- True Memory interleaving is a technique for reducing memory access time through increased bandwidth utilization of the data bus.

25) Which cache write mechanism allows outdated data in the main memory? Write back

26) In what pipeline stage is the branch target buffer checked? Fetch Fetch or Decode

27) What needs to be stored in a branch target buffer to eliminate the branch penalty for an unconditional branch? Instruction at branch target or Address of branch target

28) Assume a 64 KB cache with 4-word block size (a word is 4 bytes) and a 32-bit address. If a block has 28 tag bits, what is the type of this cache? Fully associative

29) An embedded system has a response time of 10 ms, 50 percent of which is spent on its processor. How much faster should this processor be to reduce the overall response time to 8 ms? 1.66

30) A design team is to choose between a pipelined or nonpipelined implementation for a simple, single-issue processor. Which implementation will be faster for a program that is composed of 20% ALU instructions, 10% control instructions, and 70% memory instructions? Here are some design parameters for the two versions:

Parameter	Pipelined version	Nonpipelined version
Clock rate	500 MHz	350 MHz
CPI for ALU instructions	1	1
CPI for control instructions	2	1
CPI for memory instructions	2.7	1

31) Consider a system with a two-level cache having the following characteristics: L1 cache has an access time of 1 clock cycle with an average hit rate of 95%; L2 cache has an access time of 5 clock cycles (after L1 miss) with an average miss rate of 10%. If L2 misses take 40 clock cycles, what would be the average memory access time in clock cycles? If the clock rate is 1 Gz, what would it be in seconds? 1.45 ns

32) Prior to the early 1980s, machines were built with more and more complex instruction sets. Why has there been a move to RISC machines away from complex instruction machines? Several possible answers: simplicity, simple design, lower CPI, low power, low cost, …

33) What are the two characteristics of program memory accesses that caches exploit? Temporal and spatial

34) What are the three types of cache misses? Compulsory, capacity, conflict

35) What is a Branch-Target Buffer? How does it differ from a Branch-History Table? What would be their sizes in a typical processor?

36) Given the following code:

```
for (i=2;i<100;i=i+1)
    {
        a[i] = b[i] + a[i];         /* S1 */
        c[i-1] = a[i] + d[i];       /* S2 */
        a[i-1] = 2 * b[i];          /* S3 */
        b[i+1] = 2 * b[i];          /* S4 */
    }
```

a. List all the dependencies by their types (TD: true-data, AD: anti-data, OD: output-data dependencies).

b. Show how Software Pipelining can exploit parallelism in this code to its fullest potential. How many functional units would be sufficient to achieve the fastest execution time?

c. Is this a parallel loop? If not, can it be parallelized? How?

37) Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved via forwarding? Which **dependencies** are data hazards that will cause a stall?

```
S1:     add $3, $4, $6
S2:     sub $5, $3, $2
S3:     lw $7, 100($5)
S4:     add $8, $7, $2
```

38) We have a program core consisting of four conditional branches. The program core will be executed thousands of times. Below are the outcomes of each branch for one execution of the program core (T for taken, N for not taken).

Branch 1: T – T –T
Branch 2: N – N – N – N
Branch 3: T – N – T – N –T
Branch 4: T – T – T – N – T

Assume the behavior of each branch remains the same for each program core execution. For dynamic schemes, assume each branch has its own prediction buffer and each buffer initialized to the same state before each execution. List the predictions for the following branch predication schemes:

a. Always taken
b. Always not taken
c. 1-bit predictor, initialized to predict taken

What are the **prediction accuracies**?

	Always taken	Always not taken	1-bit predictor, initialized to predict taken
T – T –T	100%	0%	100%
N – N – N – N	0%	100%	75%
T – N – T – N –T	60%	40%	20%
T – T – T – N – T	80%	20%	60%