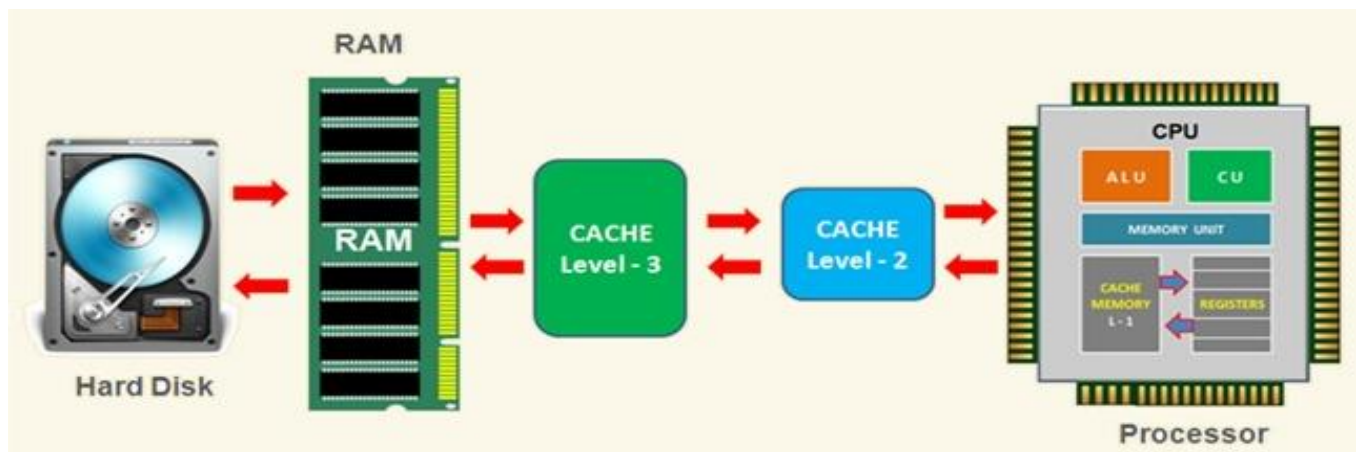


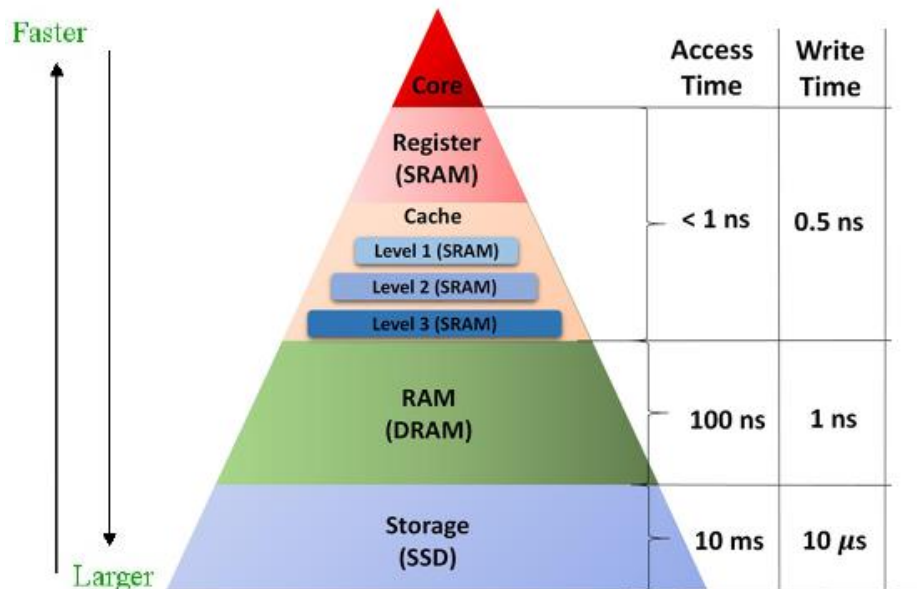
1. What is Hierarchy of Memories?

Programmers want memory to be fast, large, and cheap. Question may be asked whether we need a very fast and very large memory, since programs access a small proportion of their address space at any time! Architects have found that they can address these conflicting demands with a hierarchy of memories, with the fastest, smallest, and most expensive memory per bit at the top of the hierarchy and the slowest, largest, and cheapest per bit at the bottom. **To optimize cost and speed, combination of memory devices are used. Smaller amount of expensive but fast memory is used close to the processor whereas large amounts of cheaper but slower memory is used farther from the processor.** Hierarchy of memories give the programmer the illusion that main memory is nearly as fast as the top of the hierarchy and nearly as big and cheap as the bottom of the hierarchy. **Due to hierarchy, to CPU, it would appear** as fast as most expensive memory and as big as the cheapest.

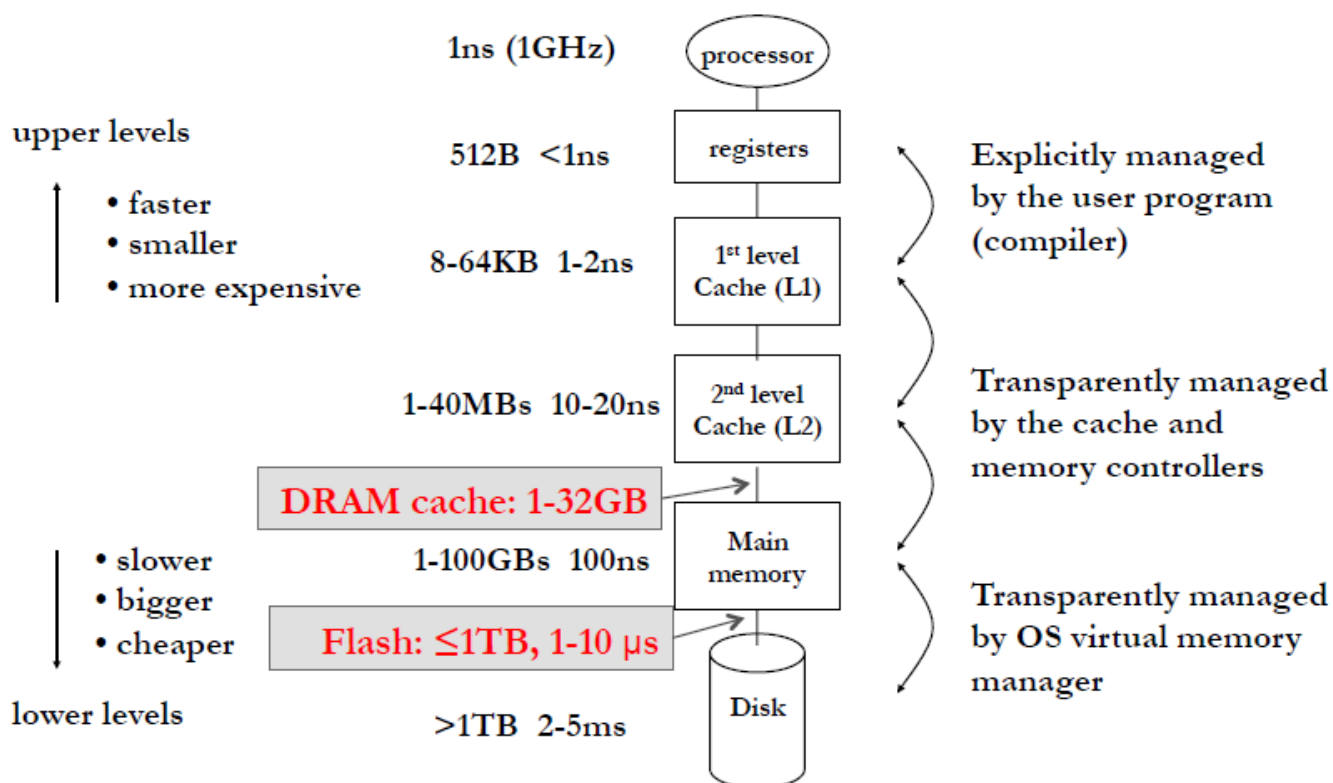
“Ideally one would desire an indefinitely large memory capacity such that any particular ... word would be immediately available... we are ... forced to recognize the possibility of constructing a hierarchy of memories, each of which has greater capacity than the preceding but which is less quickly accessible.”

A. W. Burks, H. H. Goldstine, and J. von Neumann - 1946



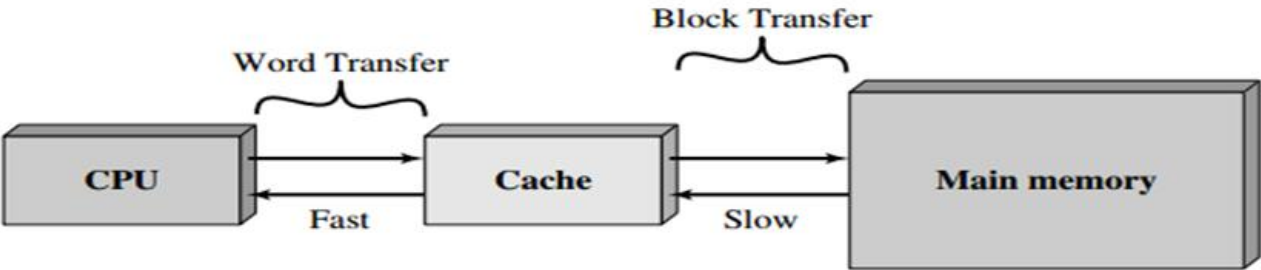
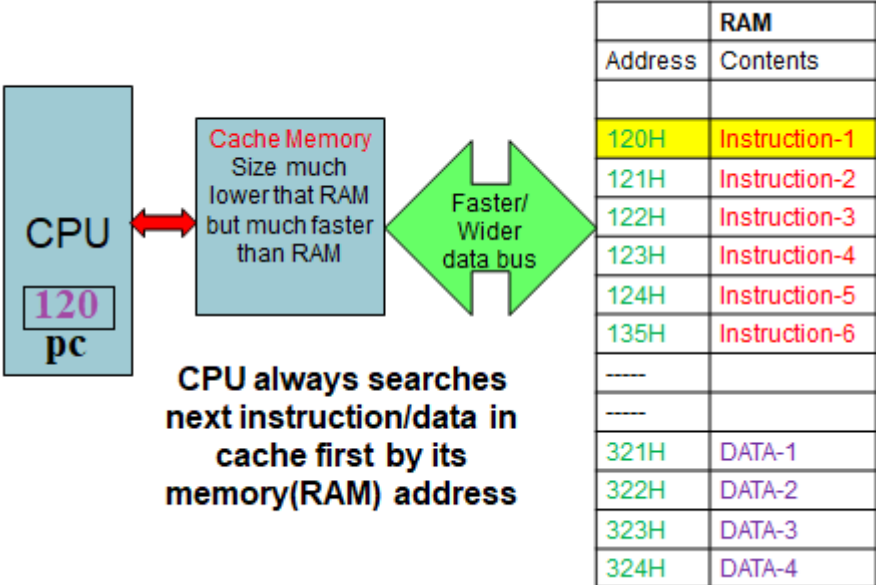


The fastest, smallest, and most expensive memory per bit at the top of the hierarchy and the slowest, largest, and cheapest per bit at the bottom.

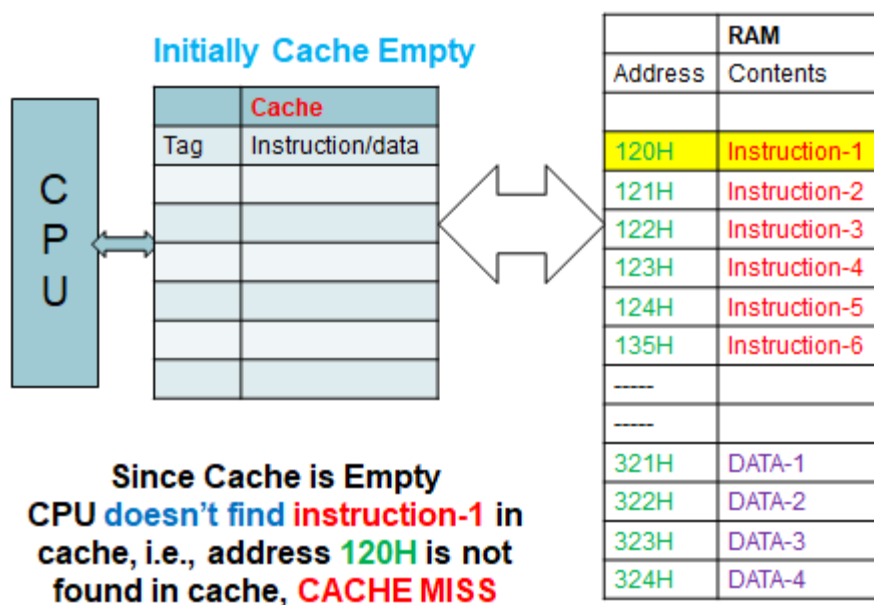
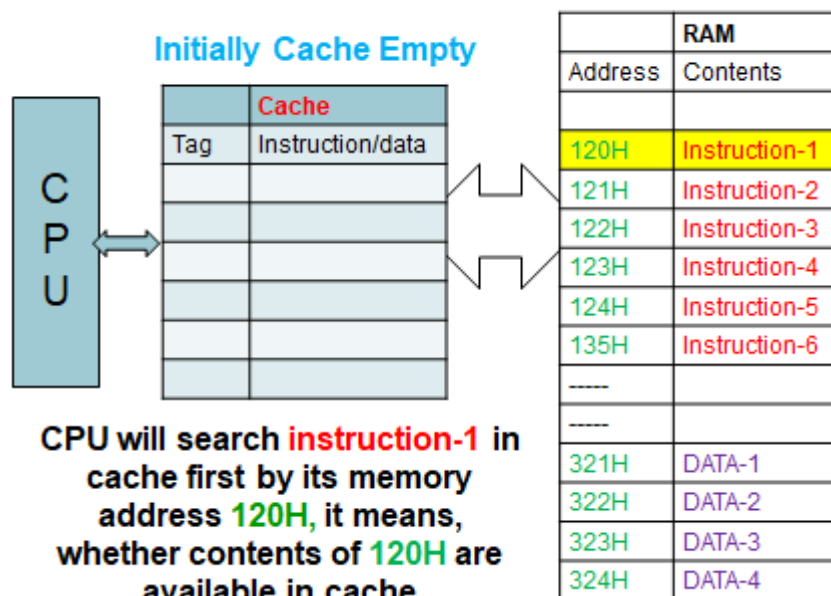


2. What is principle of locality? What is locality of reference? Define temporal locality and spatial locality.

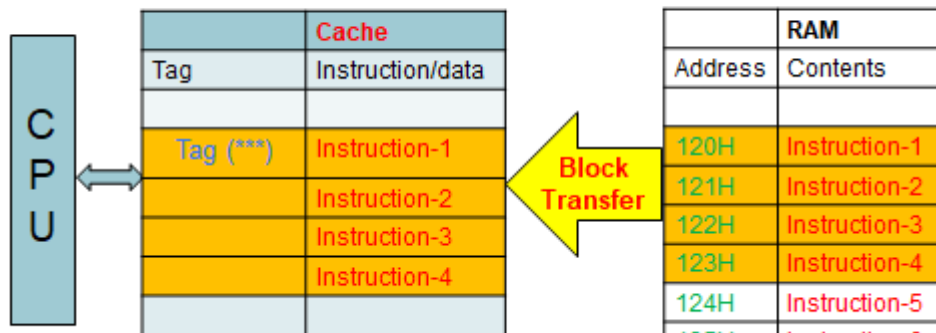
- Programs access a small proportion of their address space at any time
- Temporal locality
 - Items accessed recently are likely to be accessed again soon
 - instructions in a loop
- Spatial locality
 - Items near those accessed recently are likely to be accessed soon
 - sequential instruction access, array data

3.	<p>Cache Memory reduces frequency of access to RAM while CPU is running a program</p> <p>Reduces average memory access time of a program</p> <p>Reduces program execution time</p> <p>Caches give the programmer the illusion that main memory is nearly as fast as the top of the hierarchy and nearly as big and cheap as the bottom of the hierarchy.</p> 																												
4.	<p>How does CPU use cache memory? Explain briefly</p> <p>List the steps in sequential order, how CPU access/uses cache memory while it runs a program.</p> <p>Program (Instructions and data) are loaded into RAM. Cache memory is usually much smaller than the size of RAM. So only a fraction of information (Instruction and Data) of RAM can be copied/transferred/stored into cache memory. In a computer having cache memory, CPU is designed to search next instruction or data it requires in cache. If it is found, it is called cache hit and the CPU will read instruction/data from cache. If the instruction or data the CPU is searching for is not found in cache, it is called cache miss. In the event of cache miss, the CPU will access RAM and a number (block) of instructions/data including the one the CPU is searching for will be copied/transferred from RAM to cache memory following the spatial locality of reference. The size of the block is only few bytes, i.e., 4B/8B/16B/32B so on. The block transfer from RAM to cache in the event of cache miss increases the probability of reading following instructions from cache instead of reading from RAM.</p>  <table border="1" data-bbox="820 1451 1104 2033"> <thead> <tr> <th colspan="2">RAM</th> </tr> <tr> <th>Address</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>120H</td> <td>Instruction-1</td> </tr> <tr> <td>121H</td> <td>Instruction-2</td> </tr> <tr> <td>122H</td> <td>Instruction-3</td> </tr> <tr> <td>123H</td> <td>Instruction-4</td> </tr> <tr> <td>124H</td> <td>Instruction-5</td> </tr> <tr> <td>135H</td> <td>Instruction-6</td> </tr> <tr> <td>----</td> <td></td> </tr> <tr> <td>----</td> <td></td> </tr> <tr> <td>321H</td> <td>DATA-1</td> </tr> <tr> <td>322H</td> <td>DATA-2</td> </tr> <tr> <td>323H</td> <td>DATA-3</td> </tr> <tr> <td>324H</td> <td>DATA-4</td> </tr> </tbody> </table>	RAM		Address	Contents	120H	Instruction-1	121H	Instruction-2	122H	Instruction-3	123H	Instruction-4	124H	Instruction-5	135H	Instruction-6	----		----		321H	DATA-1	322H	DATA-2	323H	DATA-3	324H	DATA-4
RAM																													
Address	Contents																												
120H	Instruction-1																												
121H	Instruction-2																												
122H	Instruction-3																												
123H	Instruction-4																												
124H	Instruction-5																												
135H	Instruction-6																												

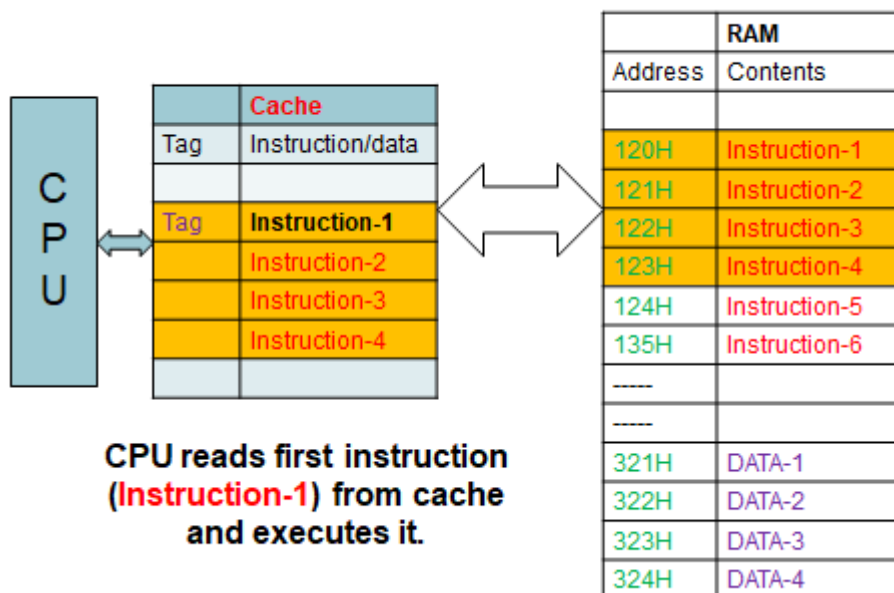
321H	DATA-1																												
322H	DATA-2																												
323H	DATA-3																												
324H	DATA-4																												

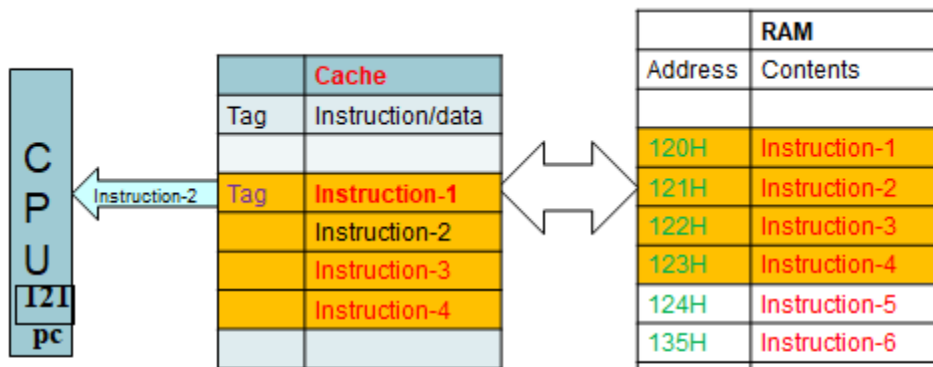


Cache Memory reduces frequency of access to RAM by using the concept of Locality of Reference

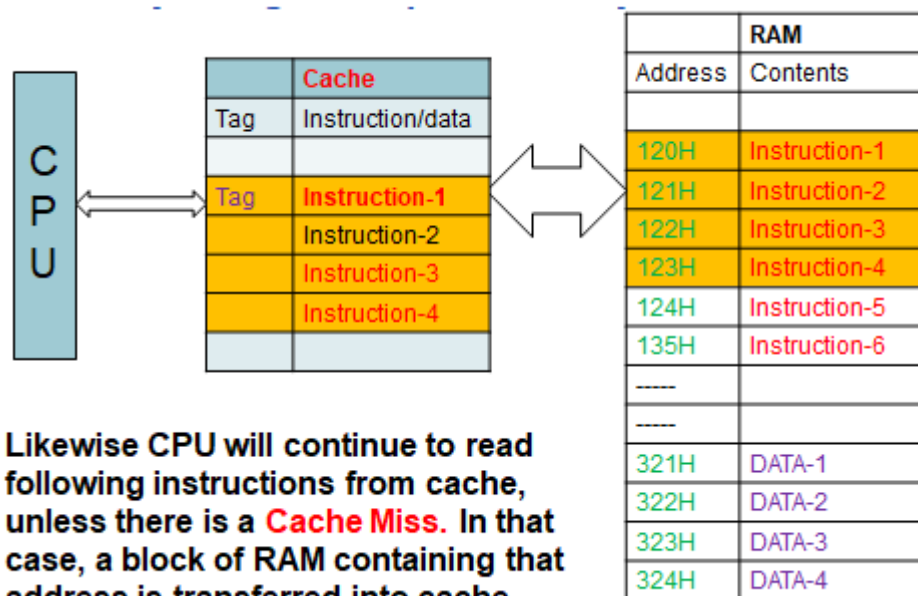


A **block of RAM** containing that instruction/address is transferred into cache. Cache also keeps info(**Tag**) which block of RAM is transferred/available in cache. **Block Transfer** supports Temporal & Spatial locality of reference





Likewise CPU will search next instruction, **instruction-2** in cache first by its memory address **121H**, and finds it, called **CACHE HIT**. CPU simply reads **Instruction-2** from cache, no need to access RAM!



Likewise CPU will continue to read following instructions from cache, unless there is a **Cache Miss**. In that case, a block of RAM containing that address is transferred into cache.

5. Define Hit rate, Hit time, Miss rate, miss penalty, average memory access time, and memory stall cycles.

- Hit Time (HT): The hit time is how long it takes data to be sent from the cache to the processor. This is usually fast, on the order of 1-3 clock cycles.
- Miss Penalty (MP): The miss penalty is the time to copy data from main memory to the cache. This often requires dozens of clock cycles (at least). The miss rate is the percentage of misses.
- Miss Rate (MR): $1 - \text{Hit Ratio}$
- The average memory access time, or AMAT, can then be computed.

$$\text{AMAT} = \text{Hit time} + (\text{Miss rate} \times \text{Miss penalty})$$

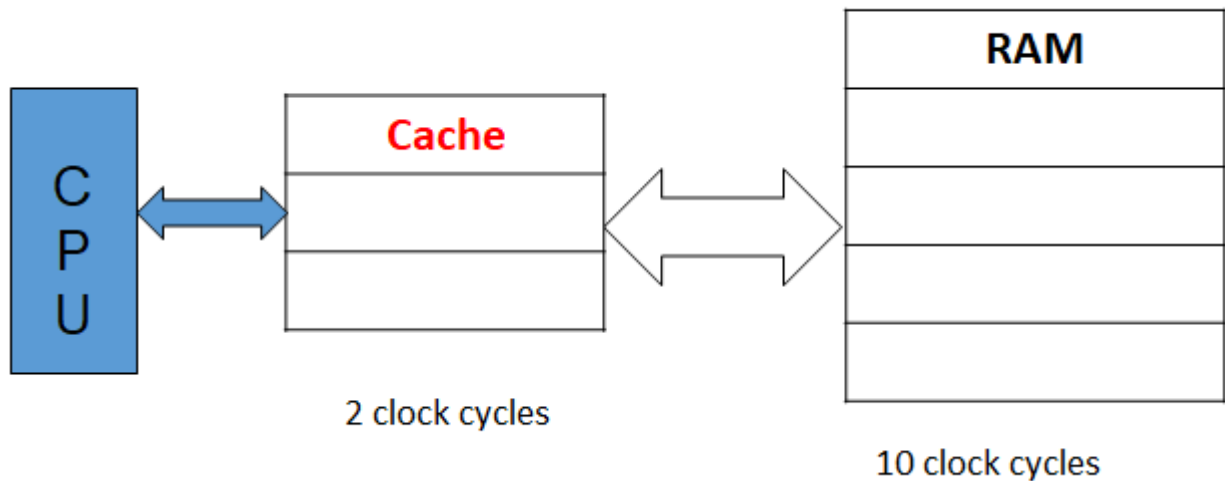
This is just averaging the amount of time for cache hits and the amount of time for cache misses

6. What is cache miss? What is the consequence of cache miss? Explain.

When CPU is searching any instruction or data in cache and if it is not found, called cache miss. In the event of cache miss, the CPU will access RAM and a number (block) of instructions/data including the one the CPU is searching for will be copied/transferred from RAM to cache memory following the spatial locality of

	<p>reference. The size of the block is only few bytes, i.e., 4B/8B/16B/32B so on. The block transfer from RAM to cache in the event of cache miss increases the probability of reading following instructions from cache instead of reading from RAM.</p> <p>In the event of cache miss, a new block is always copied/transferred from RAM to cache. If the cache is full and there is a cache miss, a new block will be copied/transferred from RAM to cache by replacing a block transferred earlier.</p>
7.	<p>What are the performance measures of cache memory?</p> <ul style="list-style-type: none"> • Hit Time (HT): The hit time is how long it takes data to be sent from the cache to the processor. This is usually fast, on the order of 1-3 clock cycles. • Miss Penalty (MP): The miss penalty is the time to copy data from main memory to the cache. This often requires dozens of clock cycles (at least). The miss rate is the percentage of misses. • Miss Rate (MR): $1 - \text{Hit Ratio}$ <p>The average memory access time, or AMAT, can then be computed.</p> <p>$\text{AMAT} = \text{Hit time} + (\text{Miss rate} \times \text{Miss penalty})$</p> <p>This is just averaging the amount of time for cache hits and the amount of time for cache misses</p>
8.	<p>What is the justification of block transfer from RAM to cache in the event of cache miss? Explain</p> <p>Instructions in users programs usually follow principle of spatial locality. It means that instruction from nearby RAM locations is likely to be read next. So in the event of cache miss, a block of instruction, including the one the CPU is searching for is copied from RAM to cache so that the CPU can read following few instructions from cache instead of accessing RAM again.</p>
9.	<p>Parameters that matter:</p> <p>Hit Time (HT)</p> <p>Miss Rate (MR)</p> <p>Miss Penalty (MP)</p> <p>$\text{AMAT} = \text{Hit Time} + \text{Miss Rate} \times \text{Miss Penalty}$</p> <p>Suppose a program (all instructions are register based), initially loaded into RAM. The Hit ratio of Cache is 80%. Calculate average access time.</p> <p>Solution: $\text{AMAT} = \text{Hit Time} + \text{Miss Rate} \times \text{Miss Penalty}$</p> <p>Average Memory Access Time: $5\text{ns} + 0.2 \times 30\text{ns} = 11\text{ns}$</p> <div data-bbox="858 1196 1493 1431" data-label="Diagram"> <pre> graph LR CPU[CPU] <--> Access time 5ns Cache[Cache] Cache <--> Access time 30ns RAM[RAM] </pre> </div>
10.	<p>If a memory system consists of a single cache with an access time of 20 ns and a hit rate of 0.92, and a main memory with an access time of 60 ns, what is the average memory access time (AMAT) of this system?</p> <p>$\text{AMAT} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$</p> <p>$\text{AMAT} = 20 + (0.08 \times 60) = 24.8 \text{ ns}$</p>

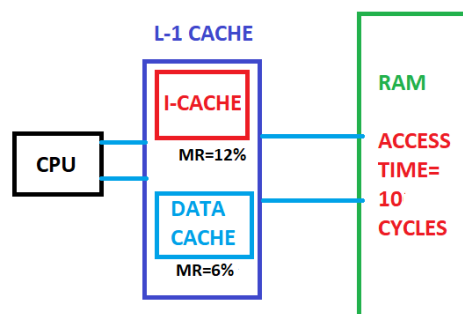
11.	<p>If Hit ratio of Cache is 80%, Hit Time (HT) = 5ns and Miss Penalty (MP) = 30ns, calculate AMAT.</p> <p>Miss Rate = $1 - 0.8 = 0.2$</p> <p>AMAT = Hit Time + Miss Rate \times Miss Penalty</p> <p>$= 5ns + 0.2 \times 30ns = 11ns$</p>
12.	<p>Hit time is also important for performance Average memory access time (AMAT)</p> <p>AMAT = Hit time + Miss rate \times Miss penalty</p> <p>Example</p> <p>CPU with 1ns clock, hit time = 1 cycle, miss penalty = 20 cycles, cache miss rate = 5%</p> <p>AMAT = $1 + 0.05 \times 20 = 2ns$</p> <p>2 cycles per instruction</p>
13.	<p>Processor clock cycle: 200 ns, Miss Penalty of 50 clock cycles, Miss Ratio of 0.02 misses/instruction, and Hit time of 1 clock cycle</p> <p>AMAT = Hit time + Miss rate \times Miss penalty</p> <p>$= 1 + 0.02 \times 50 = 2 \text{ clock cycles} = 400 \text{ ns}$</p> <p>Which improvement would be best?</p> <p>A) 190 ns clock:</p> <p>AMAT = $1 + 0.02 \times 50 = 2 \text{ clock cycles} = 380 \text{ ns}$</p> <p>B) MP (Miss Penalty) of 40 clock cycles:</p> <p>AMAT = $1 + 0.02 \times 40 = 1.8 \text{ clock cycles} = 3600 \text{ ns}$</p> <p>C) MR(Miss Ratio) of 0.015 misses/instruction:</p> <p>AMAT = $1 + 0.015 \times 50 = 1.75 \text{ clock cycles} = 350 \text{ ns}$</p>
14.	<p>A machine has a base CPI of 2 clock cycles. Measurements obtained show that the instruction miss rate is 12% and the data miss rate is 6%, and that on average, 30% of all instructions contain one data reference. The miss penalty for the cache is 10 cycles. What is the total CPI?</p> <p>Solution:</p> <p>Please note that Base CPI means the clock cycles required to read from cache memory. You can also say that this is hit time in CPU clock cycles, i.e., access time of Cache memory in CPU clock cycles.</p>



$$\begin{aligned} \text{Average CPI} &= 2.0 + \text{instruction miss cycles} + \text{data miss cycles} \\ &= 2.0 + 0.12 \times 10 + 0.30 \times 0.06 \times 10 = 2.0 + 1.2 + 0.18 = 3.38 \end{aligned}$$

15. Machine has a base CPI of 1 clock cycles. Measurements obtained show that the instruction miss rate is 15% and the data miss rate is 6%, and that on average, 40% of all instructions contain one data reference. The miss penalty for the cache is 20 cycles. What is the total CPI?

Solution:



$$\text{Average CPI} = \text{Base CPI} + \text{instruction miss cycles} + \text{data miss cycles}$$

I-cache miss rate = 15%

D-cache miss rate = 6%

Miss penalty = 20 cycles

Base CPI (ideal cache) = 1

Load & stores are 40% of instructions

Miss cycles per instruction

$$\text{I-cache: } 1 \times 0.15 \times 20 = 3$$

$$\text{D-cache: } 1 \times 0.40 \times 0.06 \times 20 = 0.48$$

$$\text{Actual CPI} = 1 + 3 + 0.48 = 4.48$$

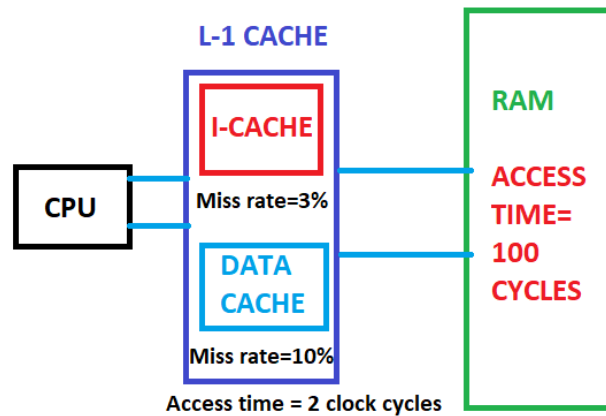
16. Suppose our processor has separate L1 instruction cache and data cache. Our CPI-base is 2 clock cycles, whereas memory accesses take 100 cycles. Our Instruction cache miss rate is 3% while our Data cache miss rate is 10%. 40% of our instructions are loads or stores.

a. What is our processor's CPI stall?

Average CPI = CPI base + L1 inst miss cycles + L1 data miss cycles

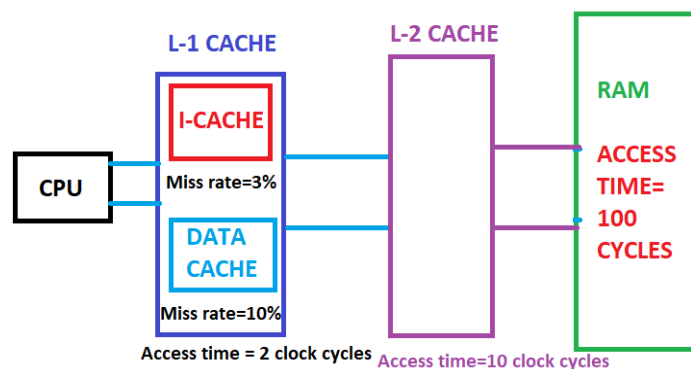
$$= 2 + 1 \times 0.03 \times 100 + 0.4 \times 0.1 \times 100$$

$$= 2 + 0.07 \times 100 = 9 \text{ cycles}$$



To improve the performance our processor, we add a unified L2 cache between the L1 caches and memory. Our L2 cache has a hit time of 10 cycles and a global miss rate of 2%.

b. What is our new Average CPI?



CPI stall = CPI base + L1 inst miss cycles + L1 data miss cycles + L2 inst miss cycles + L2 data miss cycles

$$= 2 + (1 \times 0.03 \times 10) + (0.4 \times 0.1 \times 10) + (1 \times 0.02 \times 100) + (0.4 \times 0.02 \times 100)$$

$$= 2 + (1 \times 0.03 \times 10) + (1 \times 0.02 \times 100) + (0.4 \times 0.1 \times 10) + (0.4 \times 0.02 \times 100)$$

$$= L1HT + (L1MR \times L2HT + L1MR \times L2MR \times L2MP) \text{ for instruction} + (L1MR \times L2HT + L1MR \times L2MR \times L2MP) \text{ for data}$$

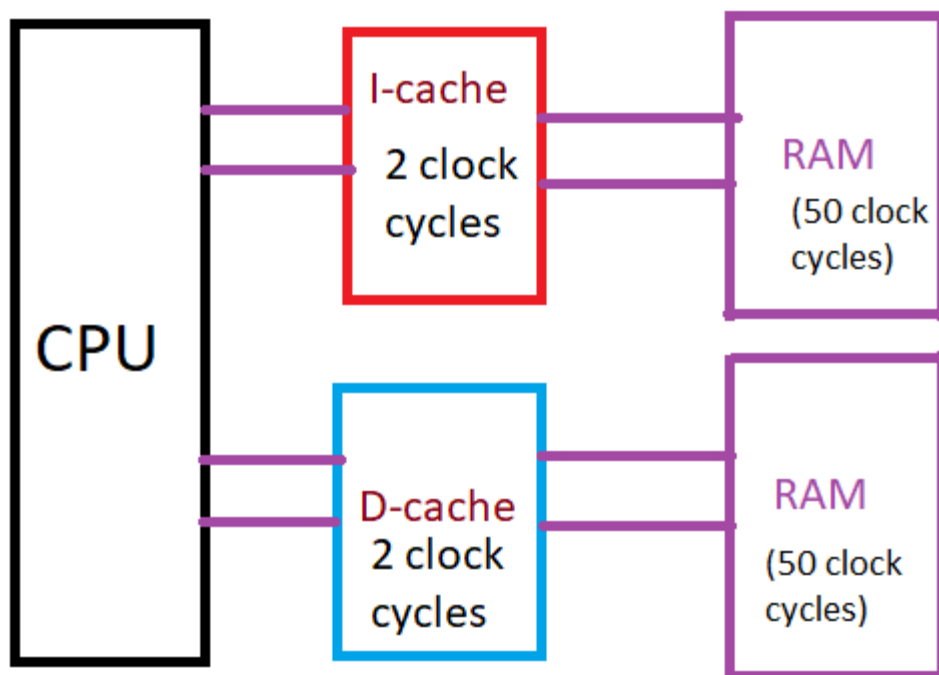
$$= 2 + 0.3 + 0.4 + 2 + 0.8$$

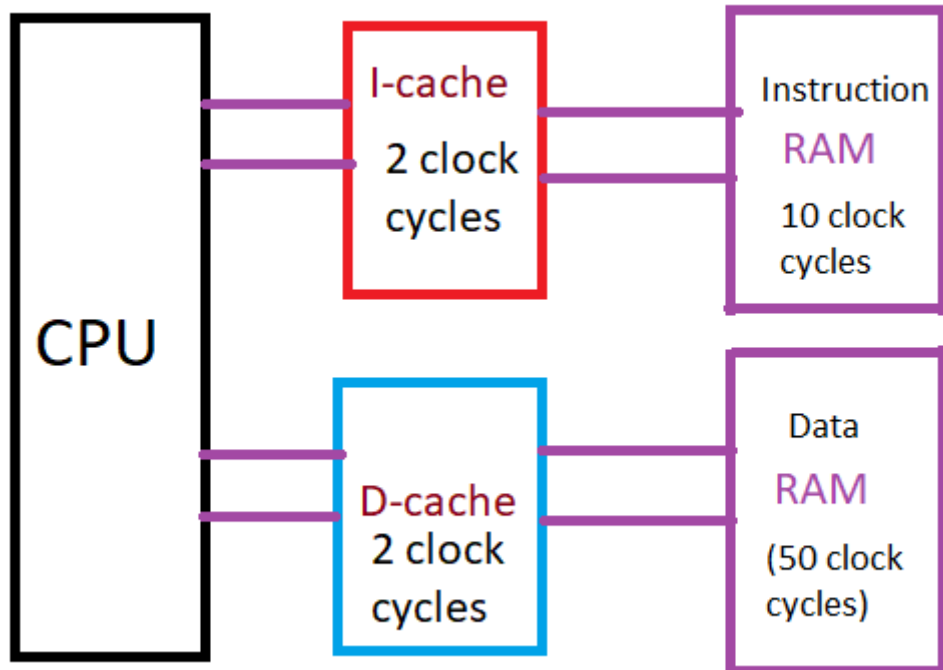
$$= 2 + 3.5 = 5.5 \text{ cycles}$$

$$= [100 \times 2 + (100 \times 0.03 \times 10) + (100 \times 0.4 \times 0.1 \times 10) + (100 \times 0.02 \times 100) + (100 \times 0.4 \times 0.02 \times 100)] / 140$$

17.

- Assume
 - I-cache miss rate 3%.
 - D-cache miss rate 5%.
 - 40% of instructions reference data.
 - Miss penalty of 50 cycles.
 - Base CPI is 2.
- What is the CPI including the misses? Average CPI
- How much slower is the machine when misses are taken into account? Average CPI/base CPI
- Redo the above if the I-miss penalty is reduced to 10 (D-miss still 50)
- With I-miss penalty back to 50, what is performance if CPU (and the caches) are 100 times faster





- 18.
- Assume
 - 5% I-cache misses.
 - 10% D-cache misses.
 - 1/3 of the instructions access data.
 - What is the CPI if the miss penalty is 12?
 - What is the CPI if miss penalty is 24 clock)?

19. I-cache miss rate = 2%
 D-cache miss rate = 4%
 Miss penalty = 100 cycles
 Base CPI (ideal cache) = 2
 Load & stores are 36% of instructions
 Solution:

Miss cycles per instruction

I-cache: $0.02 \times 100 = 2$

D-cache: $0.36 \times 0.04 \times 100 = 1.44$

Actual CPI = $2 + 2 + 1.44 = 5.44$

Ideal CPI = Base CPI = 2

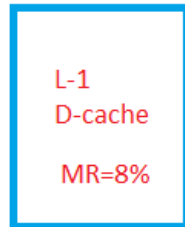
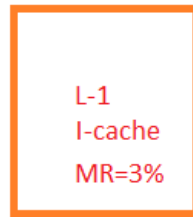
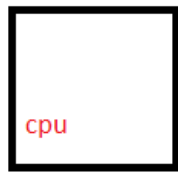
Speed up = $5.44/2 = 2.72$

Ideal CPU is 2.72 times faster

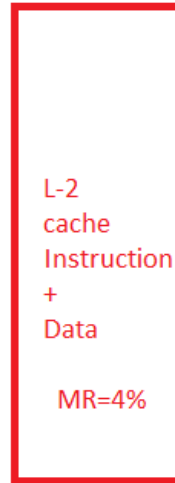
20.	<p>CPU with 1ns clock, hit time = 1 cycle, miss penalty = 20 cycles, I-cache miss rate = 5%</p> <p>Calculate: Average memory access time (AMAT)</p> <p>Solution:</p> <p>AMAT = Hit time + Miss rate \times Miss penalty</p> <p>AMAT = 1 + 0.05 \times 20 = 2ns</p> <p>2 cycles per instruction</p>
21.	<p>■ Assume that:</p> <ul style="list-style-type: none"> □ Instruction miss rate %2 □ Data miss rate %4 □ CPI is 2 (without any memory stalls) □ Miss penalty 40 cycles □ %36 of instructions are load/store <p>■ Determine how much faster a machine would run with a perfect cache that never missed.</p> <p>Instruction miss cycles = $I \times 0.02 \times 40 = 0.80 I$ (I is # of instructions)</p> <p>Data miss cycles = $I \times 0.36 \times 0.04 \times 40 = 0.58 I$</p> <p>Total memory stall cycles = $0.80 I + 0.58 I = 1.38 I$</p> <p>$CPI_{stall} = 2 + 1.38 = 3.38$</p> $\frac{\text{CPU time with stalls}}{\text{CPU time with perfect cache}} = \frac{I \times CPI_{stall} \times \text{Clock cycle}}{I \times CPI_{perfect} \times \text{Clock cycle}} = \frac{3.38}{2} = 1.69$
22.	<p>Assume</p> <p>I-cache miss rate 3%.</p> <p>D-cache miss rate 5%.</p> <p>40% of instructions reference data.</p> <p>miss penalty of 50 cycles.</p> <p>Base CPI is 2.</p> <p>What is the CPI including the misses?</p> <p>Average CPI = $2 + 0.03 \times 50 + 0.05 \times 0.4 \times 50 = 4.5$ clock cycles</p> <p>How much slower is the machine when misses are taken into account?</p> <p>$4.5/2 = 2.25$</p> <p>Redo the above if the I-miss penalty is reduced to 10 (D-miss still 50)</p> <p>I-cache miss rate 3%.</p> <p>D-cache miss rate 5%.</p> <p>40% of instructions reference data.</p> <p>I-miss penalty of 10 cycles, D-miss penalty of 50 cycles</p>

	<p>Base CPI is 2.</p> <p>With I-miss penalty back to 50, what is performance if CPU (and the caches) are 100 times faster</p>
23.	<p>Assume</p> <p>5% I-cache misses.</p> <p>10% D-cache misses.</p> <p>1/3 of the instructions access data.</p> <p>The CPI = 4 if the miss penalty is 0. A 0miss penalty is not realistic of course.</p> <p>What is the CPI if the miss penalty is 12?</p> <p>What is the CPI if we upgrade to a double speed cpu+cache, but keep a single speed memory (i.e., a 24 clock miss penalty)?</p> <p>How much faster is the double speed machine? It would be double speed if the miss penalty were 0 or if there was a 0% miss rate.</p>

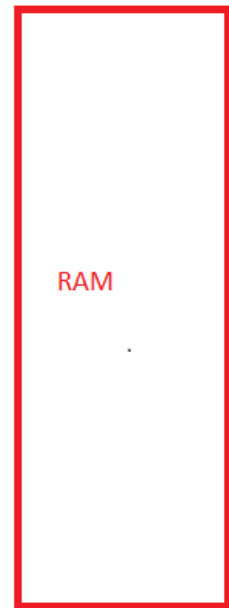
	<p>Problem:</p> <p>HT(L1) = 2ns ;</p> <p>HT(L2) = 10ns ;</p> <p>Miss Ratio(L1) = 6%,</p> <p>Miss Ratio (L2) = 2%</p> <p>RAM Access time = 100ns.</p> <p>Calculate average memory access time.</p> <p>If Average Memory Access Time = 3.358 ns, calculate what percent of instruction CPU reads from L1 cache memory?</p> <p>Miss rate of L-1 = x%</p> <p>Hit ratio of L-1 = (100 – x)</p>								
	<p>Assume:</p> <table border="1"> <tr> <td>L1 I-cache miss rate 3%</td><td>L1 D-cache miss rate 8%</td></tr> <tr> <td>30% of instructions reference data</td><td>L2 miss rate 4%</td></tr> <tr> <td>L2 time of 10 clock cycles</td><td>Memory access time 90 clock cycles</td></tr> <tr> <td>Base CPI of 2.5</td><td>CPU Clock rate 3GHz</td></tr> </table> <p>Which of the following implementation will be better?</p> <ol style="list-style-type: none"> A faster L2 of 6 clock cycles A larger L2 of miss rate 3% 	L1 I-cache miss rate 3%	L1 D-cache miss rate 8%	30% of instructions reference data	L2 miss rate 4%	L2 time of 10 clock cycles	Memory access time 90 clock cycles	Base CPI of 2.5	CPU Clock rate 3GHz
L1 I-cache miss rate 3%	L1 D-cache miss rate 8%								
30% of instructions reference data	L2 miss rate 4%								
L2 time of 10 clock cycles	Memory access time 90 clock cycles								
Base CPI of 2.5	CPU Clock rate 3GHz								



HT: L-1: 2.5 clock cycles



HT=10 clock cycles



MP = 90 clock cycles

Average CPI: base CPI + L-1 Instruction miss cycles + L-1 Data Miss cycles + L-2 instruction miss cycles + L-2 Data Miss cycles

base CPI = 2.5

L-1 Instruction miss cycles = Instruction count x Instruction MR x L-1 Miss Penalty = $1 \times 0.03 \times 10$

L-1 data miss cycles = Data access count x data MR x L-1 Miss Penalty = $(1 \times 0.3) \times 0.08 \times 10$

L-2 Instruction miss cycles = Instruction count x Instruction MR x L-2 Miss Penalty = $1 \times 0.04 \times 90$

L-2 data miss cycles = data access count x data MR x L-2 Miss Penalty = $(1 \times 0.3) \times 0.04 \times 90$

Average CPI =

Case-a

L1 I-cache miss rate 3%	L1 D-cache miss rate 8%
30% of instructions reference data	L2 miss rate 4%
L2 time of 6 clock cycles	Memory access time 90 clock cycles
Base CPI of 2.5	CPU Clock rate 3GHz

Average CPI=

Case-b

L1 I-cache miss rate 3%	L1 D-cache miss rate 8%
30% of instructions reference data	L2 miss rate 3%
L2 time of 10 clock cycles	Memory access time 90 clock cycles
Base CPI of 2.5	CPU Clock rate 3GHz

Average CPI =

Which of the following implementation will be better?

- A faster L2 of 6 clock cycles
- A larger L2 of miss rate 3%

Which of the following implementation will be better?

- a. A faster L2 of 6 clock cycles
- b. A larger L2 of miss rate 3%

Assume:

L1 I-cache miss rate 3%	L1 D-cache miss rate 8%
40% of instructions reference data	L2 miss rate 4%
L2 time of 10 clock cycles	Memory access time 80 clock cycles
Base CPI of 2	CPU Clock rate 4GHz

Which of the following implementation will be better?

- a. A faster L2 of 5 clock cycles
- b. A larger L2 of miss rate 2%