Problem M3.1: Cache Access-Time & Performance

This problem requires the knowledge of Handout 4 (Cache Implementations) and Lecture 3 (Caches). Please, read these materials before answering the following questions.

Ben is trying to determine the best cache configuration for a new processor. He knows how to build two kinds of caches: direct-mapped caches and 4-way set-associative caches. The goal is to find the better cache configuration with the given building blocks. He wants to know how these two different configurations affect the clock speed and the cache miss-rate, and choose the one that provides better performance in terms of average latency for a load.

Problem M3.1.A — Access Time: Direct-Mapped

Now we want to compute the access time of a direct-mapped cache. We use the implementation shown in Figure H4-A in Handout #4. Assume a 128-KB cache with 8-word (32-byte) cache lines. The address is 32 bits, and the two least significant bits of the address are ignored since a cache access is word-aligned. The data output is also 32 bits, and the MUX selects one word out of the eight words in a cache line. Using the delay equations given in Table M3.1-1, fill in the column for the direct-mapped (DM) cache in the table. *In the equation for the data output driver, 'associativity' refers to the associativity of the cache (1 for direct-mapped caches, A for A-way set-associative caches).*

Component	Delay equation (ps)		DM (ps)	SA (ps)
Decoder	200×(# of index bits) + 1000	Tag		
		Data		
Memory array	$200\times\log_2$ (# of rows) + $200\times\log_2$ (# of bits in a row) + 1000	Tag		
		Data		
Comparator	200×(# of tag bits) + 1000			
N-to-1 MUX	$500\times\log_2 N + 1000$			
Buffer driver	2000			
Data output driver	500×(associativity) + 1000			
Valid output driver	1000			

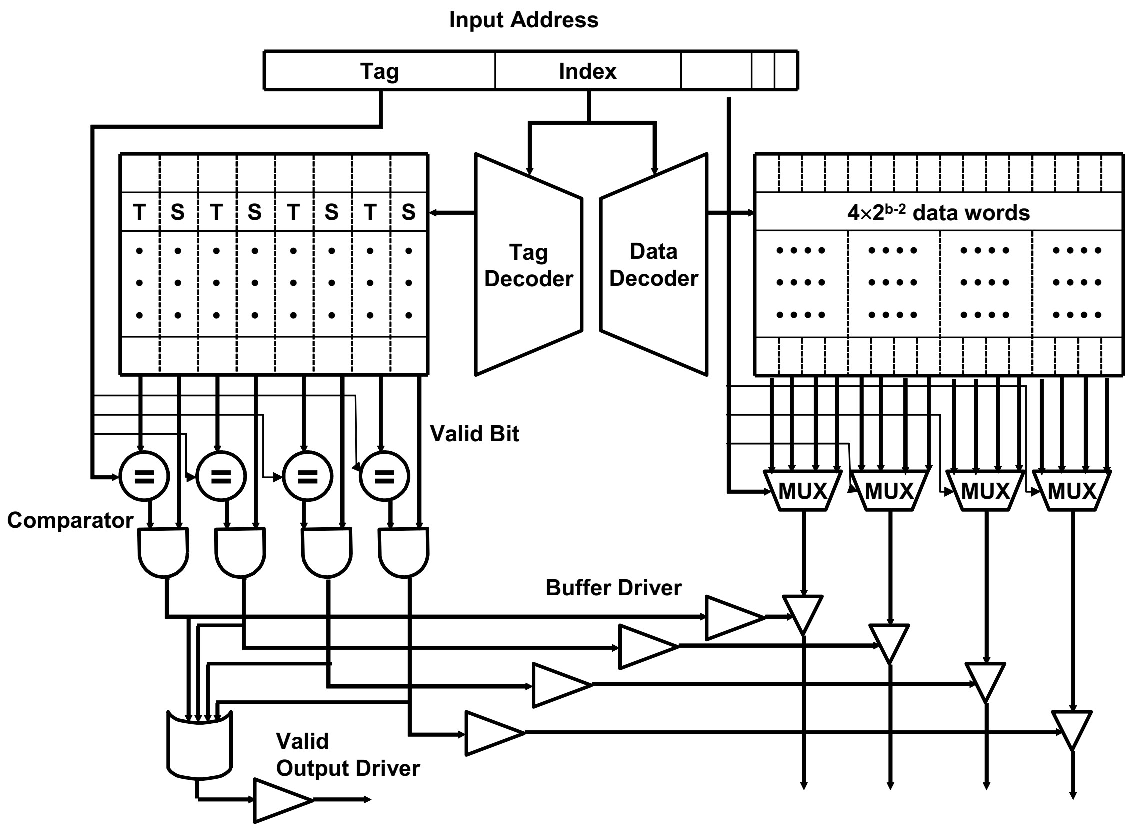
Table M3.1-1: Delay of each Cache Component

What is the critical path of this direct-mapped cache for a cache read? What is the access time of the cache (the delay of the critical path)? To compute the access time, assume that a 2-input gate (AND, OR) delay is 500 ps. If the CPU clock is 150 MHz, how many CPU cycles does a cache access take?

Problem M3.1.B — Access Time: Set-Associative

We also want to investigate the access time of a set-associative cache using the 4-way set-associative cache in Figure H4-B in Handout #4. Assume the total cache size is still 128-KB (each way is 32-KB), a 4-input gate delay is 1000 ps, and all other parameters (such as the input address, cache line, etc.) are the same as part M3.1.A. Compute the delay of each component, and fill in the column for a 4-way set-associative cache in Table M3.1-1.

What is the critical path of the 4-way set-associative cache? What is the access time of the cache (the delay of the critical path)? What is the main reason that the 4-way set-associative cache is slower than the direct-mapped cache? If the CPU clock is 150 MHz, how many CPU cycles does a cache access take?



Problem M3.1.C	Miss-rate analysis

Now Ben is studying the effect of set-associativity on the cache performance. Since he now knows the access time of each configuration, he wants to know the miss-rate of each one. For the miss-rate analysis, Ben is considering two small caches: a direct-mapped cache with 8 lines with 16 bytes/line, and a 4-way set-associative cache of the same size. For the set-associative cache, Ben tries out two replacement policies – least recently used (LRU) and round robin (FIFO).

Ben tests the cache by accessing the following sequence of hexadecimal byte addresses, starting with empty caches. For simplicity, assume that the addresses are only 12 bits. Complete the following tables for the direct-mapped cache and both types of 4-way set-associative caches showing the progression of cache contents as accesses occur (in the tables, 'inv' = invalid, and the column of a particular cache line contains the {tag,index} contents of that line). *You only need to fill in elements in the table when a value changes.*

D-map									
	line in cache								hit?
Address	L0	L1	L2	L3	L4	L5	L6	L7	
110	inv	11	inv	inv	inv	inv	inv	inv	no
136				13					no
202	20								no
1A3									
102									
361									
204									
114									
1A4									
177									
301									
206									
135									

	D-map
Total Misses	
Total Accesses	

4-way									LRU
Address	line in cache								hit?
	Set 0				Set 1				
	way0	**way1**	**Way2**	**way3**	**way0**	**way1**	**way2**	**way3**	
110	inv	Inv	Inv	inv	11	inv	inv	inv	no
136					11	13			no
202	20								no
1A3									
102									
361									
204									
114									
1A4									
177									
301									
206									
135									

	4-way LRU
Total Misses	
Total Accesses	

4-way									FIFO
Address	line in cache								hit?
	Set 0				Set 1				
	way0	**way1**	**way2**	**way3**	**way0**	**way1**	**way2**	**way3**	
110	inv	Inv	Inv	inv	11	inv	inv	inv	no
136						13			no
202	20								no
1A3									
102									
361									
204									
114									
1A4									
177									
301									
206									
135									

	4-way FIFO
Total Misses	
Total Accesses	

Problem M3.1.D — Average Latency

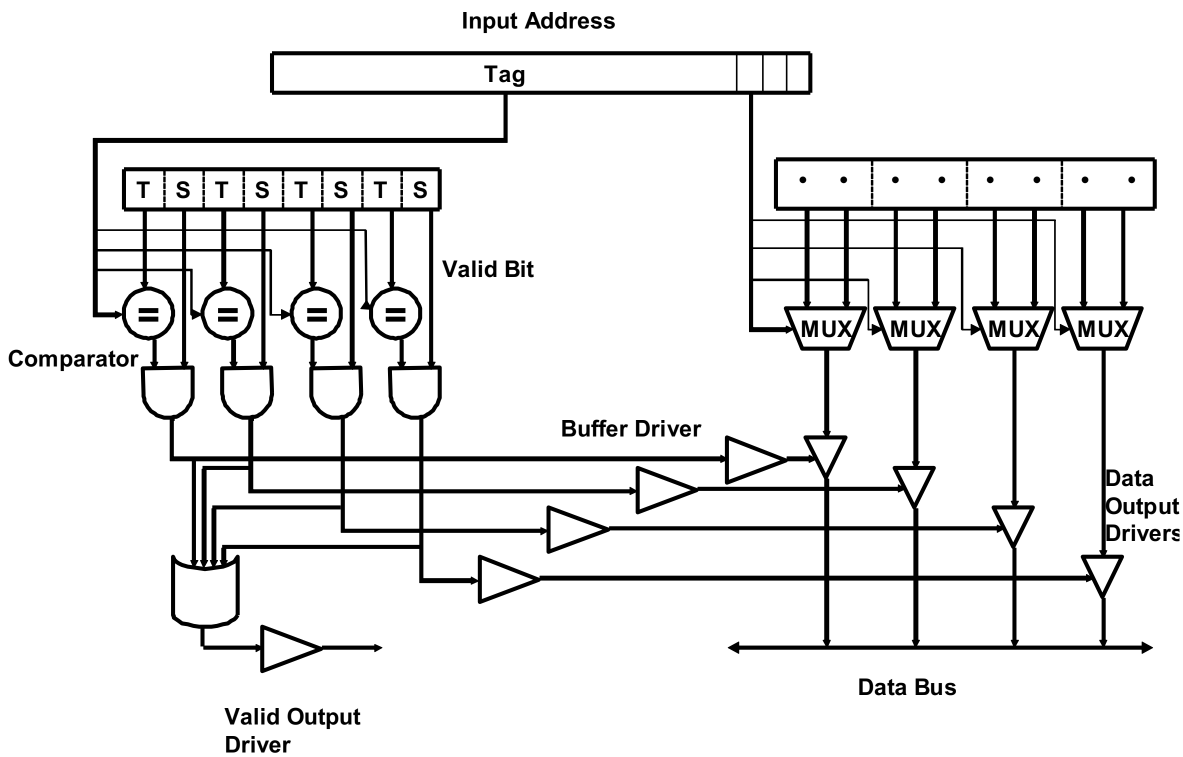
Assume that the results of the above analysis can represent the average miss-rates of the direct-mapped and the 4-way LRU 128-KB caches studied in M3.1.A and M3.1.B. What would be the average memory access latency in CPU cycles for each cache (assume that a cache miss takes 20 cycles)? Which one is better? For the different replacement policies for the set-associative cache, which one has a smaller cache miss rate for the address stream in M3.1.C? Explain why. Is that replacement policy always going to yield better miss rates? If not, give a counter example using an address stream.

Problem M3.2: Victim Cache Evaluation

This problem requires the knowledge of Handout #5 (Victim Cache) and Lecture 3. Please, read these materials before answering the following questions.

Problem M3.2.A — Baseline Cache Design

The diagram below shows a 32-Byte fully associative cache with four 8-Byte cache lines. Each line consists of two 4-Byte words and has an associated tag and two status bits (valid and dirty). The Input Address is 32-bits and the two least significant bits are assumed to be zero. The output of the cache is a 32-bit word.



Please complete Table M3.2-1 below with delays across each element of the cache. Using the data you compute in Table M3.2-1, calculate the critical path delay through this cache (from when the Input Address is set to when both Valid Output Driver and the appropriate Data Output Driver are outputting valid data).

Component	Delay equation (ps)		FA (ps)
Comparator	200×(# of tag bits) + 1000		
N-to-1 MUX	$500 \times \log_2 N + 1000$		
Buffer driver	2000		
AND gate	1000		
OR gate	500		
Data output driver	500×(associativity) + 1000		
Valid output driver	1000		

Table M3.2-1

Critical Path Cache Delay: ________________________

Problem M3.2.B — Victim Cache Behavior

Now we will study the impact of a victim cache on a cache hit rate. Our main L1 cache is a 128 byte, direct mapped cache with 16 bytes per cache line. The cache is word (4-bytes) addressable. The victim cache in Figure H5-A (in Handout #5) is a 32 byte fully associative cache with 16 bytes per cache line, and is also word-addressable. The victim cache uses the first in first out (FIFO) replacement policy.

Please complete Table M3.2-2 on the next page showing a trace of memory accesses. In the table, each entry contains the {tag,index} contents of that line, or "inv", if no data is present. You should only fill in elements in the table when a value changes. For simplicity, the addresses are only 8 bits.

The first 3 lines of the table have been filled in for you.

For your convenience, the address breakdown for access to the main cache is depicted below.

7	6 – 4	3 – 2	1 – 0
TAG	INDEX	WORD SELECT	BYTE SELECT

Problem M3.2.C — Average Memory Access Time

Assume **15%** of memory accesses are resolved in the victim cache. If retrieving data from the victim cache takes **5 cycles** and retrieving data from main memory takes **55 cycles**, by how many cycles does the victim cache improve the average memory access time?

Input Address	Main Cache									Victim Cache		
	L0	L1	L2	L3	L4	L5	L6	L7	Hit?	Way0	Way1	Hit?
	inv	inv	inv	inv	inv	inv	inv	inv	-	inv	inv	-
00	0								N			N
80	8								N	0		N
04	0								N	8		Y
A0												
10												
C0												
18												
20												
8C												
28												
AC												
38												
C4												
3C												
48												
0C												
24												

Table M3.2-2

Problem M3.3: Loop Ordering

This problem requires the knowledge of Lecture 3. Please, read it before answering the following questions.

This problem evaluates the cache performances for different loop orderings. You are asked to consider the following two loops, written in C, which calculate the sum of the entries in a 128 by 64 matrix of 32-bit integers:

Loop A	***Loop B***

`sum = 0;`
`for (i = 0; i < 128; i++)`
`  for (j = 0; j < 64; j++)`
`    sum += A[i][j];`
	`sum = 0;`
	`for (j = 0; j < 64; j++)`
	`  for (i = 0; i < 128; i++)`
	`    sum += A[i][j];`

The matrix `A` is stored contiguously in memory in row-major order. Row major order means that elements in the same row of the matrix are adjacent in memory as shown in the following memory layout:

`A[i][j]` resides in memory location `[4*(64*i + j)]`

Memory Location:

0	4		252	256		4*(64*127+63)
A[0][0]	A[0][1]	...	A[0][63]	A[1][0]	...	A[127][63]

For *Problem M3.3.A* to *Problem M3.3.C*, assume that the caches are initially empty. Also, assume that only accesses to matrix `A` cause memory references and all other necessary variables are stored in registers. Instructions are in a separate instruction cache.

Problem M3.3.A

Consider a 4KB direct-mapped data cache with 8-word (32-byte) cache lines.
Calculate the number of cache misses that will occur when running Loop A.
Calculate the number of cache misses that will occur when running Loop B.

The number of cache misses for Loop A:______________________________

The number of cache misses for Loop B:______________________________

Problem M3.3.B

Consider a direct-mapped data cache with 8-word (32-byte) cache lines. Calculate the minimum number of cache lines required for the data cache if Loop A is to run without any cache misses other than compulsory misses. Calculate the minimum number of cache lines required for the data cache if Loop B is to run without any cache misses other than compulsory misses.

Data-cache size required for Loop A: _____________________________ cache line(s)

Data-cache size required for Loop B: _____________________________ cache line(s)

Problem M3.3.C

Consider a 4KB fully-associative data cache with 8-word (32-byte) cache lines. This data cache uses a first-in/first-out (FIFO) replacement policy.
Calculate the number of cache misses that will occur when running Loop A.
Calculate the number of cache misses that will occur when running Loop B.

The number of cache misses for Loop A:______________________________

The number of cache misses for Loop B:______________________________

Problem M3.4: Cache Parameters

For each of the following statements about making a change to a cache design, circle **True** or **False** and provide a one sentence explanation of your choice. Assume all cache parameters (capacity, associativity, line size) remain fixed except for the single change described in each question. Please provide a one sentence explanation of your answer.

Problem M3.4.A

Doubling the line size halves the number of tags in the cache

True / False

Problem M3.4.B

Doubling the associativity doubles the number of tags in the cache.

True / False

Problem M3.4.C

Doubling cache capacity of a direct-mapped cache usually reduces conflict misses.

True / False

Problem M3.4.D

Doubling cache capacity of a direct-mapped cache usually reduces compulsory misses.

True / False

Problem M3.4.E

Doubling the line size usually reduces compulsory misses.

True / False

Problem M3.5: Microtags

Problem M3.5.A

Explain in one or two sentences why direct-mapped caches have much lower hit latency (as measured in picoseconds) than set-associative caches of the same capacity.

Problem M3.5.B

A 32-bit byte-addressed machine has an 8KB, 4-way set-associative data cache with 32-byte lines. The following figure shows how the address is divided into tag, index and offset fields. Give the number of bits in each field.

tag	Index	offset

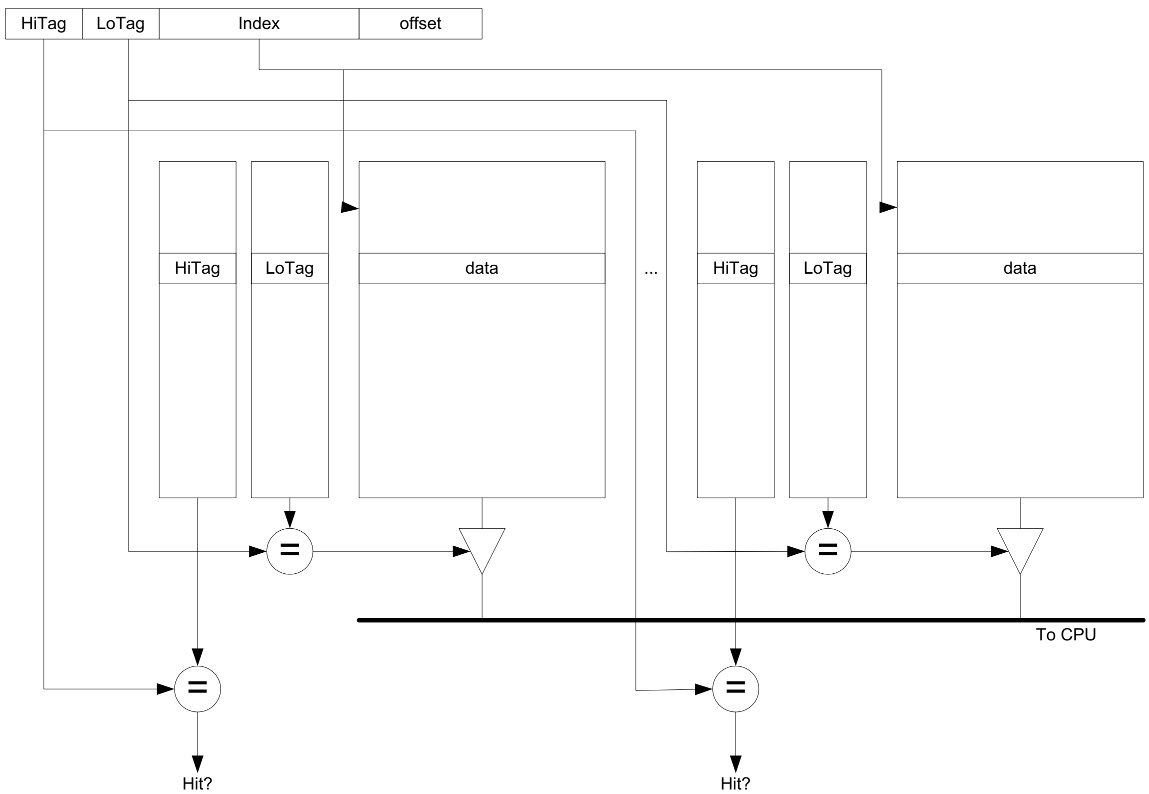
# of bits in the tag: _______________

# of bits in the index: _______________

# of bits in the offset: _______________

Microtags (for questions M3.5.C – M3.5.H)

Several commercial processors (including the UltraSPARC-III and the Pentium-4) reduce the hit latency of a set-associative cache by using only a subset of the tag bits (a "microtag") to select the matching way before speculatively forwarding data to the CPU. The remaining tag bits are checked in a subsequent clock cycle to determine if the access was actually a hit. The figure below illustrates the structure of a cache using this scheme.



Problem M3.5.C

The tag field is sub-divided into a **loTag** field used to select a way and a **hiTag** field used for subsequent hit/miss checks, as shown below.

tag			
hiTag	**loTag**	index	offset

The cache design requires that all lines within a set have unique loTag fields.
In one or two sentences, explain why this is necessary.

Problem M3.5.D

If the **loTag** field is exactly two bits long, will the cache have greater, fewer, or an equal number of conflict misses as a direct-mapped cache of the same capacity? State any assumptions made about replacement policy.

Problem M3.5.E

If the **loTag** field is greater than two bits long, are there any additional constraints on replacement policy beyond those in a conventional 4-way set-associative cache?

Problem M3.5.F

Does this scheme reduce the time required to complete a write to the cache? Explain in one or two sentences.

Problem M3.5.G

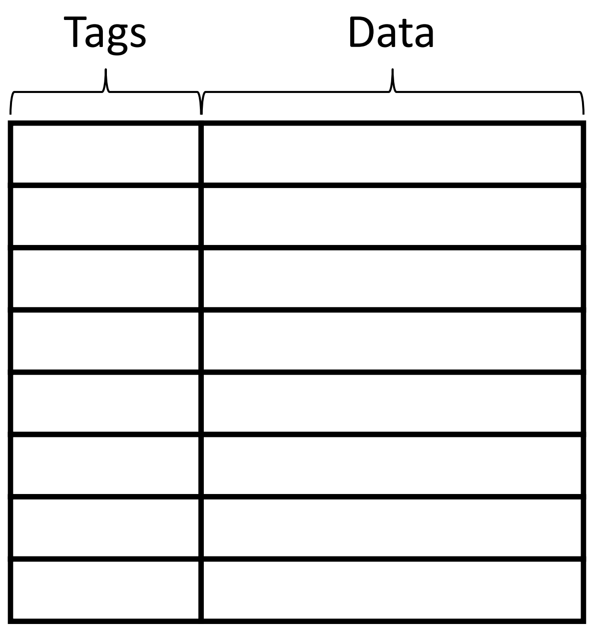
In practice, microtags hold virtual address bits to remove address translation from the critical path, while the full tag check is performed on translated physical addresses. If the **loTag** bits can only hold untranslated bits of the virtual address, what is the largest number of **loTag** bits possible if the machine has a 16KB virtual memory page size? (Assume 8KB 4-way set-associative cache as in Question M3.5.B)

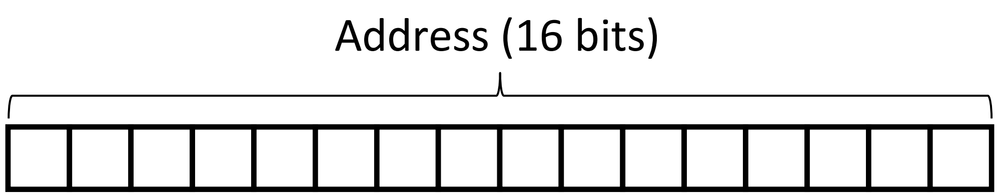
Problem M3.5.H

Describe how microtags can be made much larger, to also include virtual address bits subject to address translation. Your design should not require address translation before speculatively forwarding data to the CPU. Your explanation should describe the replacement policy and any additional state the machine must maintain.

Problem M3.6: Caches (Spring 2014 Quiz 1, Part C)

Your processor has an 8-line level 1 data cache as illustrated below. Suppose that cache lines are 32 bytes (256 bits) and memory addresses are 16 bits, with byte-addressable memory. The cache is indexed by low bits without hashing.





Problem M3.6.A

We're first going to fill in the above diagram with more detail.

Divide the bits of the address according to how they are used to access the cache (tag, index, offset).

What exactly is contained in the cache tags? (Include all bits necessary for correct operation of the cache as discussed in lecture.)

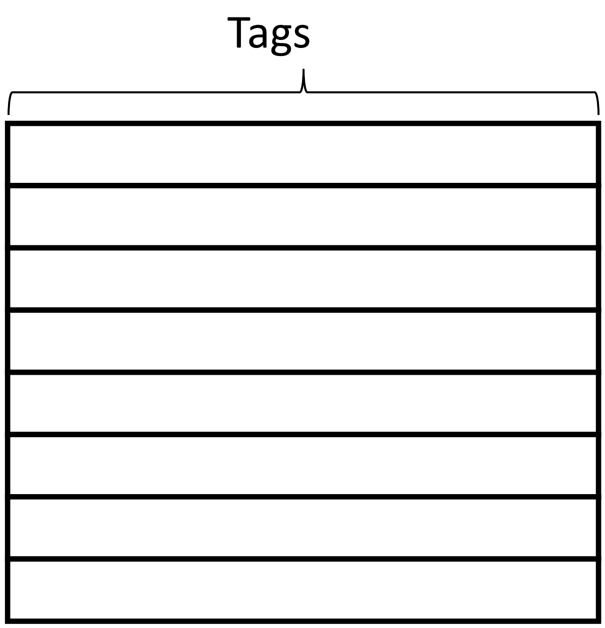
How many bits <u>in total</u> are needed to implement the level 1 data cache?

Problem M3.6.B

Suppose the processor accesses the following data addresses starting with an empty cache:

```
0x0028: 0000 0000 0010 1000
0x102A: 0001 0000 0010 1010
0x9435: 1001 0100 0011 0101
0xEFF4: 1110 1111 1111 0100
0xBEEF: 1011 1110 1110 1111
0x4359: 0100 0011 0101 1001
0x01DE: 0000 0001 1101 1110
0x8075: 1000 0000 0111 0101
0x9427: 1001 0100 0010 0111
```

What would the level 1 data cache tags look like after this sequence? How many hits would there be in the level 1 data cache? *(Don't worry about filling in the Data column – we didn't give you the data!)*



Problem M3.6.C

Suppose that the level 1 data cache has a hit rate of 40% on your application, an access time of a single cycle, and a miss penalty to memory of forty cycles. What is the average memory access time?

You aren't happy with your memory performance, so you decide to add a level two cache. Suppose the level two cache has a hit rate of 50%. What access time must the level two cache have for this to be a good design (ie, reduce AMAT)?