Cache Simulations on Paper. The following table allows you to simulate the behavior of a cache on paper. The first 4 columns show you the contents of the cache at the time of a particular cache access. We are showing the **address** of each byte that lives in a particular cache block. A real cache would store a tag and data here, but for simulation purposes, the address itself is sufficient.

The **outcome** should say whether this access was a hit or a miss. If it was a miss, specify whether it was *cold, capacity, or conflict.* The next row should begin with the *new* contents of the cache as a result of the access on the previous line. Given that current
state of the cache, determine the outcome of the next access. If the contents of a particular set does not change, feel free to leave that field empty on the next line down.

This cache is a direct-mapped, 8B cache with 4B blocks. This cache was filled via access to addresses in the order: 0000, 0100.

Cache contents (prior to access)		Address of Cache Access	Outcome of Cache Access
Set 0	**Set 1**		
0000 | 0001 | 0010 | 0011	0100 | 0101 | 0110 | 0111	1100	*Cold miss*
	1100 | 1101 | 1110 | 1111	1010	**cold miss**
1000 | 1001 | 1010 | 1011		1011	**Hit!**
		0011	**capacity miss**
0000 | 0001 | 0010 | 0011		0101	**capacity miss**
	0100 | 0101 | 0110 | 0111	1100	**capacity miss**
	1100 | 1101 | 1110 | 1111	1001	**capacity miss**

Repeat this process a third time for a 2-way set associative 8B cache with 2B blocks. Use LRU replacement policy and assume that Way 1 of each set was most recently used.

Cache contents (prior to access)				Address of Cache Access	Outcome of Cache Access
Set0, Way0	Set0, Way1	Set1, Way0	Set1, Way1		
0000|0001	0100|0101	0010|0011	0110|0111	1100	*Cold miss*
1100|1101				1010	cold miss
		1010|1011		1011	Hit!
				0011	capacity miss
			0010|0011	0101	Hit!
				1100	Hit!
				1001	cold miss

This cache is a direct-mapped, 8B cache with 2B blocks. Addresses are 4 bits.
This cache was filled via access to addresses in the order: 0000, 0010, 0100, 0110.

Cache contents (prior to access)				address of cache access	outcome of cache access
Set 00	Set 01	Set 10	Set 11		
0000|0001	0010|0011	0100|0101	0110|0111	1100	cold miss
		1100|1101		1010	cold miss
	1010|1011			1011	Hit
				0011	capacity miss
	0010|0011			0101	capacity miss
		0100|0101		1100	conflict miss
		1100|1101		1001	cold miss

0) 0000 | 0001
1) 0010 | 0011 0-3 are the initial accesses that populate the cache
2) 0100 | 0101
3) 0110 | 0111

Now the question starts:

4) **1100** | 1101 cold miss
5) **1010** | 1011 cold miss
1010 | **1011** hit (this is a repeat from #5, not distinct)
6) 0010 | **0011** capacity miss – we haven't seen this since time #1
7) 0100 | **0101** capacity miss – we haven't seen this since time #2
8 **1100** | 1101 conflict miss – there are 4 cache lines in this cache, and we brought this line in 4 ago – it would still be there if F.A.
9 1000 | 1001 cold miss

Virtual Memory

The following problem concerns the way virtual addresses are translated into physical addresses. The memory is byte addressable.Memory accesses are to **1-byte words** (not 4-byte words).Virtual addresses are 16 bits wide.Physical addresses are 13 bits wide.The page size is 512 bytes.The TLB is 8-way set associative with 16 total entries.The cache is 2-way set associative, with a 4 byte line size and 16 total lines.

In the following tables, **all numbers are given in hexadecimal**. The contents of the TLB, the page table for the first 32 pages, and the cache are as follows:

TLB			
Index	Tag	PPN	Valid
0	09	4	1
	12	2	1
	10	0	1
	08	5	1
	05	7	1
	13	1	0
	10	3	0
	18	3	0
1	04	1	0
	0C	1	0
	12	0	0
	08	1	0
	06	7	0
	03	1	0
	07	5	0
	02	2	0

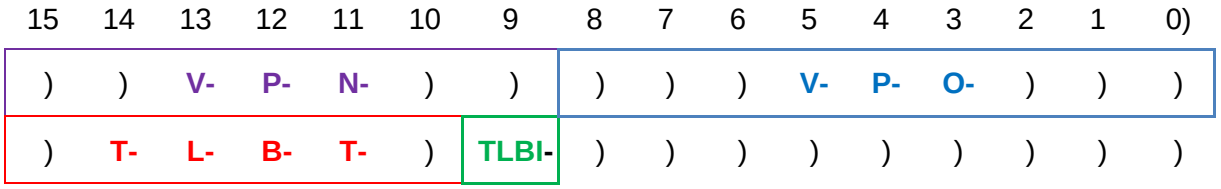
Page Table					
VPN	PPN	Valid	VPN	PPN	Valid
00	6	1	10	0	1
01	5	0	11	5	0
02	3	1	12	2	1
03	4	1	13	4	0
04	2	0	14	6	0
05	7	1	15	2	0
06	1	0	16	4	0
07	3	0	17	6	0
08	5	1	18	1	1
09	4	0	19	2	0
0A	3	0	1A	5	0
0B	2	0	1B	7	0
0C	5	0	1C	6	0
0D	6	0	1D	2	0
0E	1	1	1E	3	0
0F	0	0	1F	1	0

2-way Set Associative Cache												
Index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	19	1	99	11	23	11	00	0	99	11	23	11
1	15	0	4F	22	EC	11	2F	1	55	59	0B	41
2	1B	1	00	02	04	08	0B	1	01	03	05	07
3	06	0	84	06	B2	9C	12	0	84	06	B2	9C
4	07	0	43	6D	8F	09	05	0	43	6D	8F	09
5	0D	1	36	32	00	78	1E	1	A1	B2	C4	DE
6	11	0	A2	37	68	31	00	1	BB	77	33	00
7	16	1	11	C2	11	33	1E	1	00	C0	0F	00

(a) The box below shows the format of a virtual address. Indicate (by labeling the diagram) the fields (if they exist) that would be used to determine the following: (If a field doesn't exist, don't draw it on the diagram.)

VPO The virtual page offset
VPN The virtual page number
TLBI The TLB index
TLBT The TLB tag

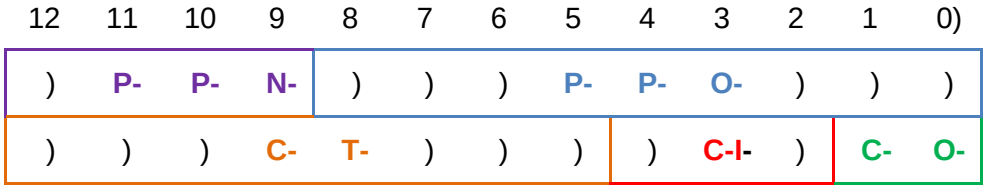
The page size is 512 bytes ✎ Page offset = 9 bits
The remaining bits are the VPN = 7 bits
The TLB uses the VPN to perform a lookup. The TLB has 2 rows ✎ 1 index bit
The remaining bits are the tag bits.



(b) The box below shows the format of a physical address. Indicate (by labeling the diagram) the fields that would be used to determine the following:

PPO The physical page offset
PPN The physical page number
CO The block offset within the cache line
CI The cache index
CT The cache tag

Page offset never changes, stays 9 bits.
The remaining bits are the PPN = 4 bits
Cache has 4 byte line size ✎ 2 bit offset
8 cache entries ✎ 3 index bits
Remaining bits are tag.



For the given virtual address, indicate the TLB entry accessed, the physical address, and the cache byte value returned **in hex**. Indicate whether the TLB misses, whether a page fault occurs, and whether a cache miss occurs.
If there is a cache miss, enter "-" for "Cache Byte returned". If there is a page fault, enter "-" for "PPN" and leave parts C and D blank.

Virtual address: **`1DDE`**

(c) Virtual address format (one bit per box)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0)
0-	**0-**	**0-**	**1-**	**1-**	**1-**	**0-**	**1-**	**1-**	**1-**	**0-**	**1-**	**1-**	**1-**	**1-**	**0-**

(d) Address translation

Parameter	Value
VPN	**0x0E**
TLB index	**0x0**
TLB tag	**0x07**
TLB Hit? (Y?N)	**N**
Page Fault? (Y?N)	**N**
PPN	**0x1**

(e) Physical address format (one bit per box)

12	11	10	9	8	7	6	5	4	3	2	1	0)
0-	**0-**	**0-**	**1-**	**1-**	**1-**	**1-**	**0-**	**1-**	**1-**	**1-**	**1-**	**0-**

(f) Physical memory reference

Parameter	Value
Byte offset	**0x2**
Cache index	**0x7**
Cache tag	**0x1E**
Cache Hit? (Y?N)	**Y**
Cache Byte Returned	**0x0F**

Repeat this process with a new address

Virtual address: `0BDE`

(g) Virtual address format (one bit per box)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0)
0-	0-	0-	0-	1-	0-	1-	1-	1-	1-	0-	1-	1-	1-	1-	0-

(h) Address translation

Parameter	Value
VPN	0x05
TLB index	0x1
TLB tag	0x02
TLB Hit? (Y?N)	N
Page Fault? (Y?N)	N
PPN	0x7

(i) Physical address format (one bit per box)

12	11	10	9	8	7	6	5	4	3	2	1	0)
0-	1-	1-	1-	1-	1-	1-	0-	1-	1-	1-	1-	0-

(j) Physical memory reference

Parameter	Value
Byte offset	0x2
Cache index	0x7
Cache tag	0x7E
Cache Hit? (Y?N)	N
Cache Byte Returned	--