1. Direct mapped cache

format:

tag	block	word
2 bits	3 bits	2 bits

words per block = _______________

blocks in cache = _______________

words in memory = _______________

Initial contents of main memory, in hex with 8-bit words:

address (measured in words, written in binary)	???0000- ???0011	???0100- ???0111	???1000- ???1011	???1100- ???1111
000????	40 8e 31 9e	bf ea 46 56	52 fd 31 44	5c e7 5e f6
001????	87 6e d2 82	72 e4 c1 42	b1 28 31 b0	73 f1 b7 84
010????	81 f4 84 2f	86 7a 16 7d	10 e5 27 fd	04 6c 8d c6
011????	e9 5f c6 4c	07 51 8f 56	b4 d0 75 2b	89 57 7d 95
100????	e5 d0 c7 e6	92 3a e6 d3	de a8 16 15	5a 76 92 3a
101????	e9 e2 14 c8	a6 b5 3d 0c	a7 64 69 16	9f 29 19 87
110????	3b 1f 22 4c	a1 8a f4 1b	64 db 78 20	80 6e 2a b3
111????	42 f1 c7 b3	70 26 29 03	b8 ac 7e 4d	96 6d 52 98

memory references:

cycle	address referenced	hit or miss?
0	0010100	
1	0010101	
2	1111001	
3	0101100	
4	1110110	
5	0000101	
6	0000110	
7	1110100	
8	1111110	

cache contents:

block	tag	data	valid
0			0
1			0
2			0
3			0
4			0
5			0
6			0
7			0

2. Fully associative cache

format:

tag	word
5 bits	2 bits

Assume **8 blocks** in cache.

words per block = ________________

words in memory = ________________

Initial contents of main memory, in hex with 8-bit words:

address (measured in words, written in binary)	???0000- ???0011	???0100- ???0111	???1000- ???1011	???1100- ???1111
000????	40 8e 31 9e	bf ea 46 56	52 fd 31 44	5c e7 5e f6
001????	87 6e d2 82	72 e4 c1 42	b1 28 31 b0	73 f1 b7 84
010????	81 f4 84 2f	86 7a 16 7d	10 e5 27 fd	04 6c 8d c6
011????	e9 5f c6 4c	07 51 8f 56	b4 d0 75 2b	89 57 7d 95
100????	e5 d0 c7 e6	92 3a e6 d3	de a8 16 15	5a 76 92 3a
101????	e9 e2 14 c8	a6 b5 3d 0c	a7 64 69 16	9f 29 19 87
110????	3b 1f 22 4c	a1 8a f4 1b	64 db 78 20	80 6e 2a b3
111????	42 f1 c7 b3	70 26 29 03	b8 ac 7e 4d	96 6d 52 98

memory references:

cycle	address referenced	hit or miss?
0	0010100	
1	0010101	
2	1111001	
3	0101100	
4	1110110	
5	0000101	
6	0000110	
7	1110100	
8	1111110	
9	1010010	
10	0011010	
11	0010110	
12	0011100	

cache contents (use LRU eviction):

block	tag	data	valid	last used
0			0	
1			0	
2			0	
3			0	
4			0	
5			0	
6			0	
7			0	

3. N-way set associative cache

format:

tag	set	word
3 bits	2 bits	2 bits

Assume **2-way** set associative cache.

words per block = ________________
blocks in set = ________________
sets in cache = ________________
blocks in cache = ________________
words in memory = ________________

Initial contents of main memory, in hex with 8-bit words:

address (measured in words, written in binary)	???0000- ???0011	???0100- ???0111	???1000- ???1011	???1100- ???1111
000????	40 8e 31 9e	bf ea 46 56	52 fd 31 44	5c e7 5e f6
001????	87 6e d2 82	72 e4 c1 42	b1 28 31 b0	73 f1 b7 84
010????	81 f4 84 2f	86 7a 16 7d	10 e5 27 fd	04 6c 8d c6
011????	e9 5f c6 4c	07 51 8f 56	b4 d0 75 2b	89 57 7d 95
100????	e5 d0 c7 e6	92 3a e6 d3	de a8 16 15	5a 76 92 3a
101????	e9 e2 14 c8	a6 b5 3d 0c	a7 64 69 16	9f 29 19 87
110????	3b 1f 22 4c	a1 8a f4 1b	64 db 78 20	80 6e 2a b3
111????	42 f1 c7 b3	70 26 29 03	b8 ac 7e 4d	96 6d 52 98

memory references:

cycle	address referenced	hit or miss?
0	0010100	
1	0010101	
2	1111001	
3	0101100	
4	1110110	
5	0000101	
6	0000110	
7	1110100	
8	1111110	
9	1010010	
10	0011010	
11	0001110	
12	0010100	

cache contents (use LRU eviction):

set	tag	data	valid	last used
0			0	
			0	
1			0	
			0	
2			0	
			0	
3			0	
			0	