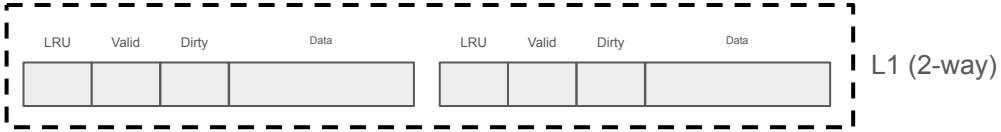
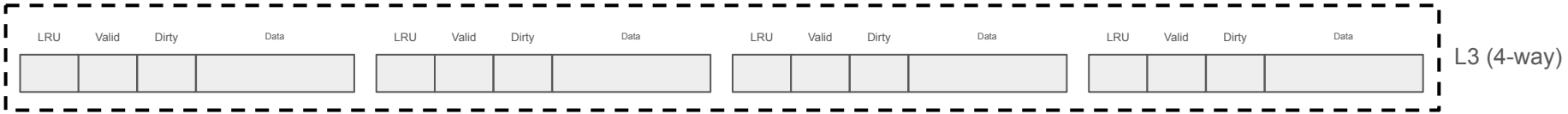
Cache Access Path

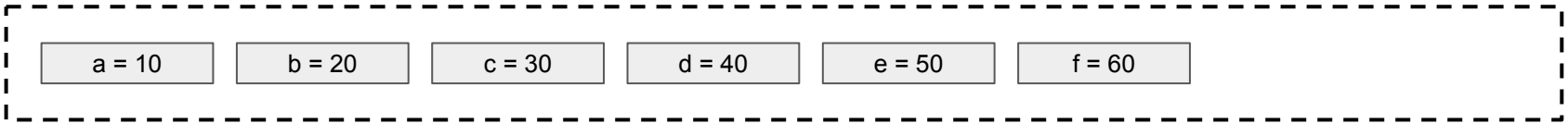
Spring 2025 ICS Team

Cache Structure

Cache Line Structure

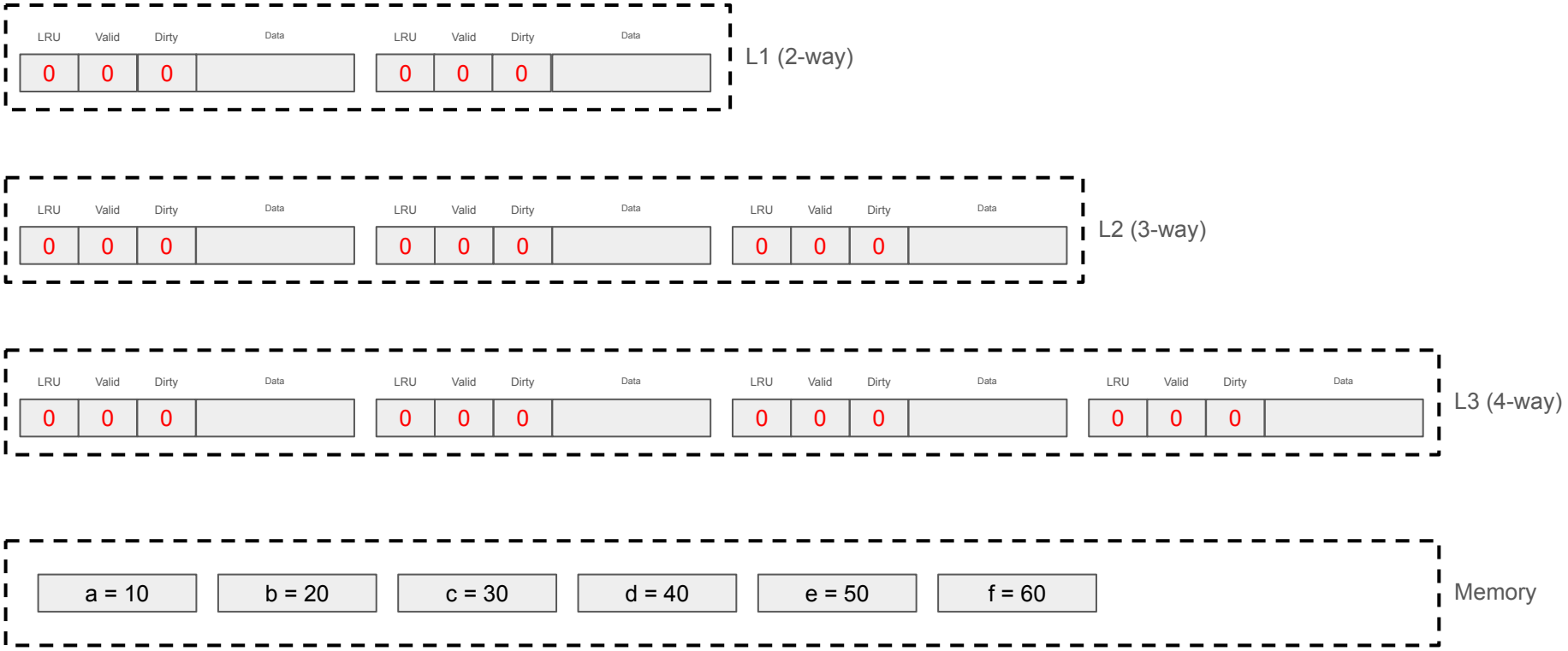






Initialize

Tick 0

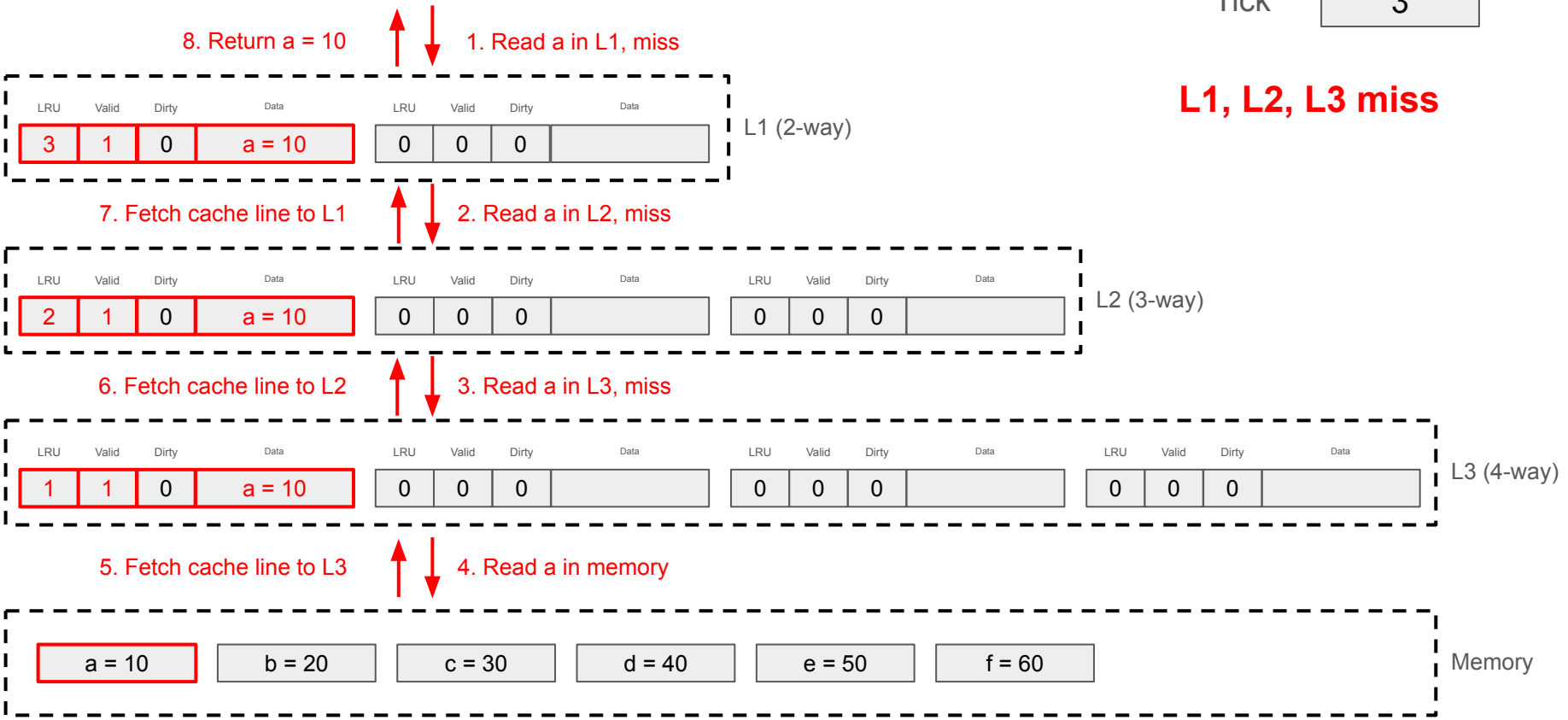


Cache line affected

Read a

Tick 3

L1, L2, L3 miss

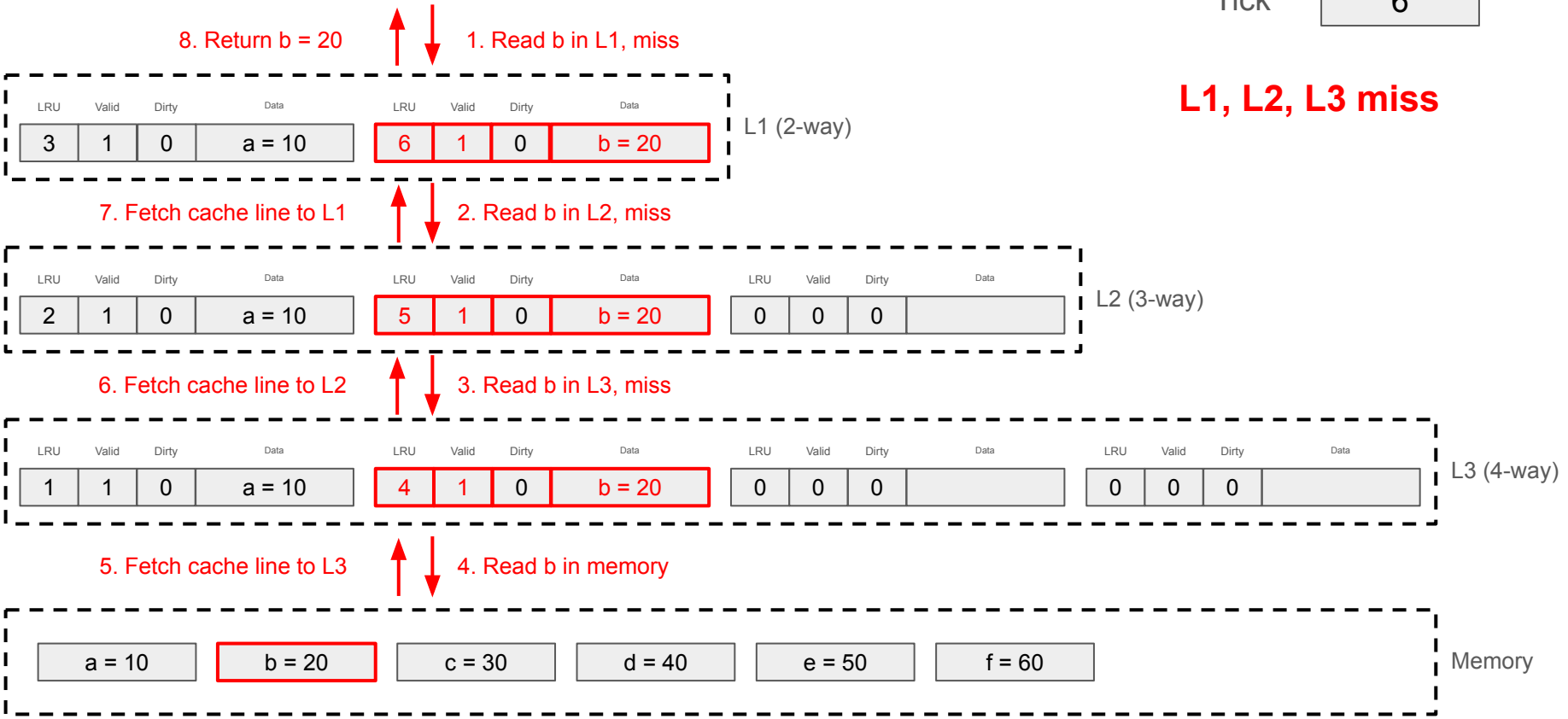


Cache line affected

Read b

Tick 6

L1, L2, L3 miss



Cache line affected

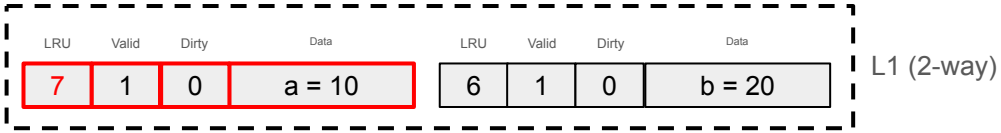
Read a

Tick 7

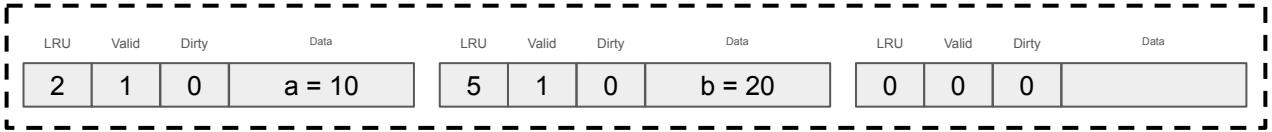
2. Return a = 10

1. Read a in L1, hit

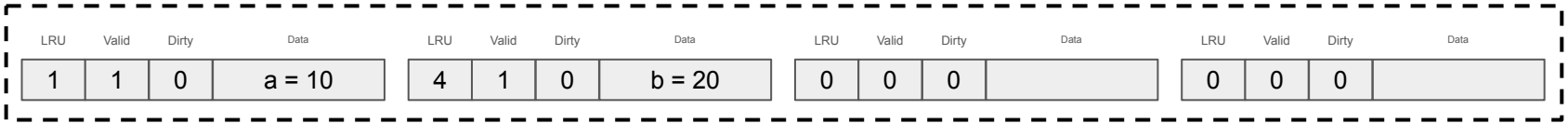
L1 hit



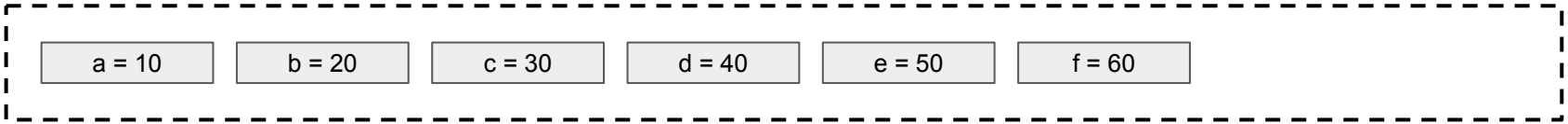
L1 (2-way)



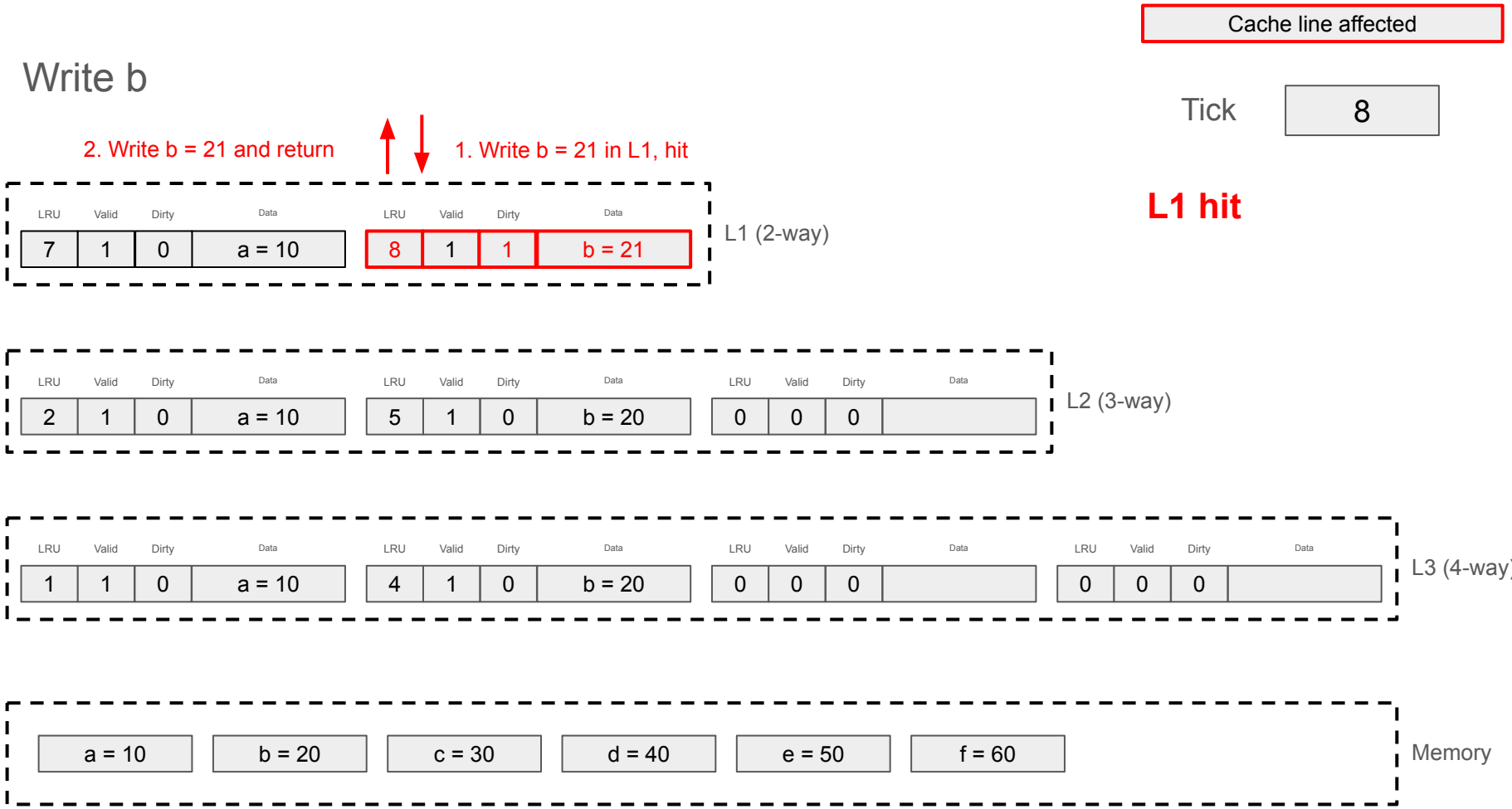
L2 (3-way)

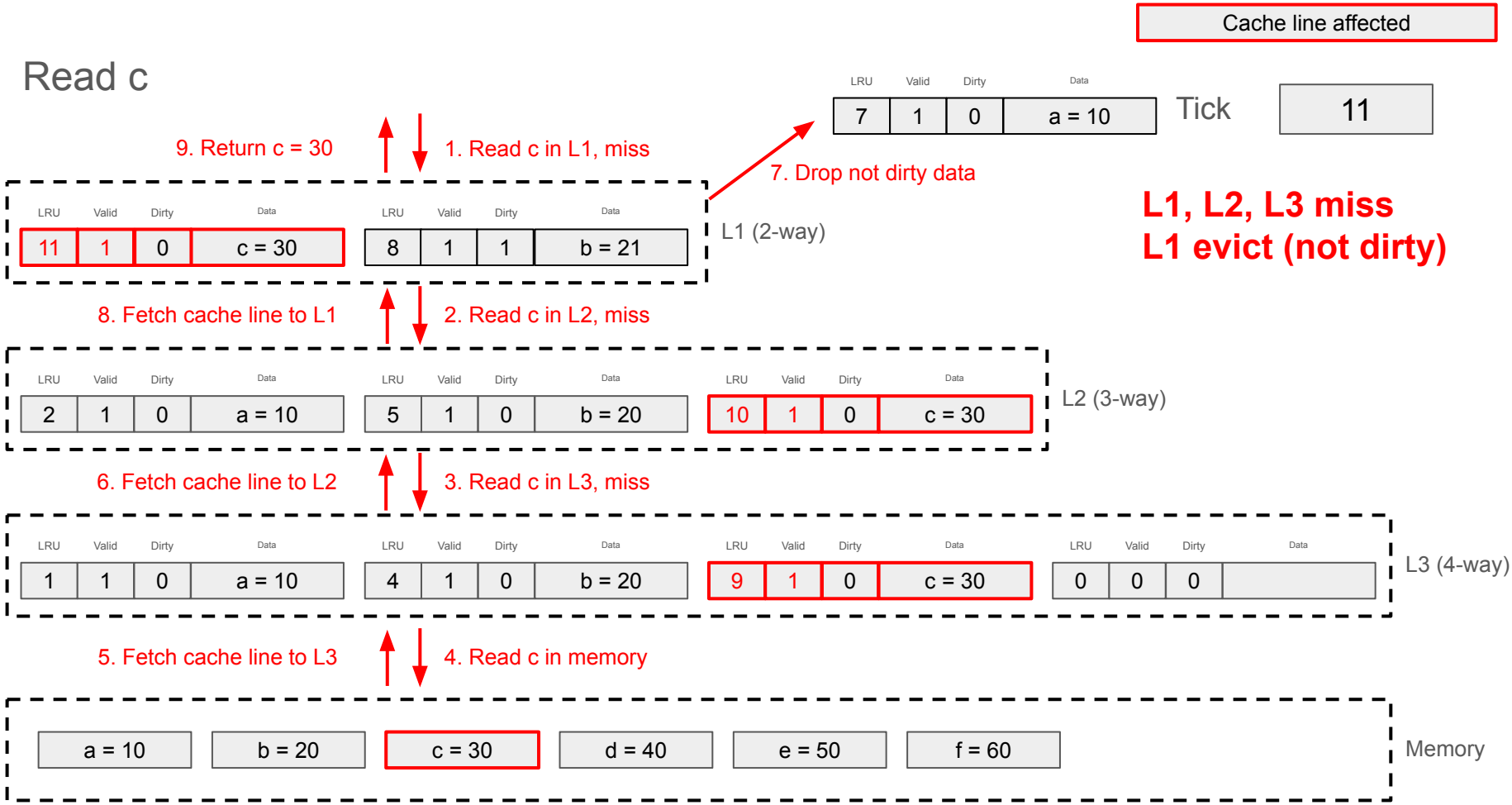


L3 (4-way)



Memory


Cache line affected
Write b
Tick
8
2. Write b = 21 and return
1. Write b = 21 in L1, hit
L1 hit
LRU Valid Dirty Data
7 1 0 a = 10
8 1 1 b = 21
L1 (2-way)
LRU Valid Dirty Data
2 1 0 a = 10
5 1 0 b = 20
0 0 0
L2 (3-way)
LRU Valid Dirty Data
1 1 0 a = 10
4 1 0 b = 20
0 0 0
0 0 0
L3 (4-way)
a = 10
b = 20
c = 30
d = 40
e = 50
f = 60
Memory

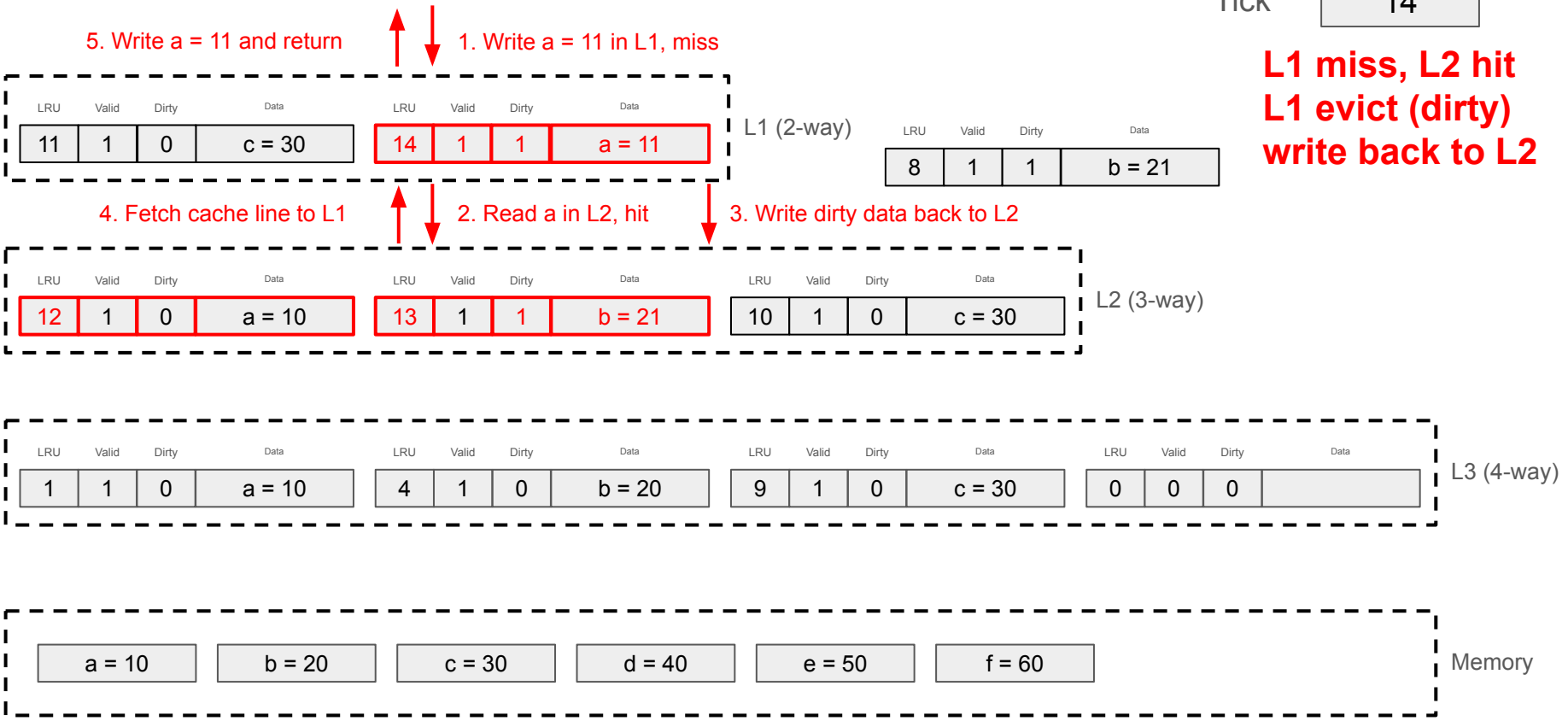

Cache line affected
Read c
LRU Valid Dirty Data
7 1 0 a = 10
Tick
11
9. Return c = 30
1. Read c in L1, miss
7. Drop not dirty data
LRU Valid Dirty Data
11 1 0 c = 30
8 1 1 b = 21
L1 (2-way)
L1, L2, L3 miss
L1 evict (not dirty)
8. Fetch cache line to L1
2. Read c in L2, miss
LRU Valid Dirty Data
2 1 0 a = 10
5 1 0 b = 20
10 1 0 c = 30
L2 (3-way)
6. Fetch cache line to L2
3. Read c in L3, miss
LRU Valid Dirty Data
1 1 0 a = 10
4 1 0 b = 20
9 1 0 c = 30
0 0 0
L3 (4-way)
5. Fetch cache line to L3
4. Read c in memory
a = 10
b = 20
c = 30
d = 40
e = 50
f = 60
Memory

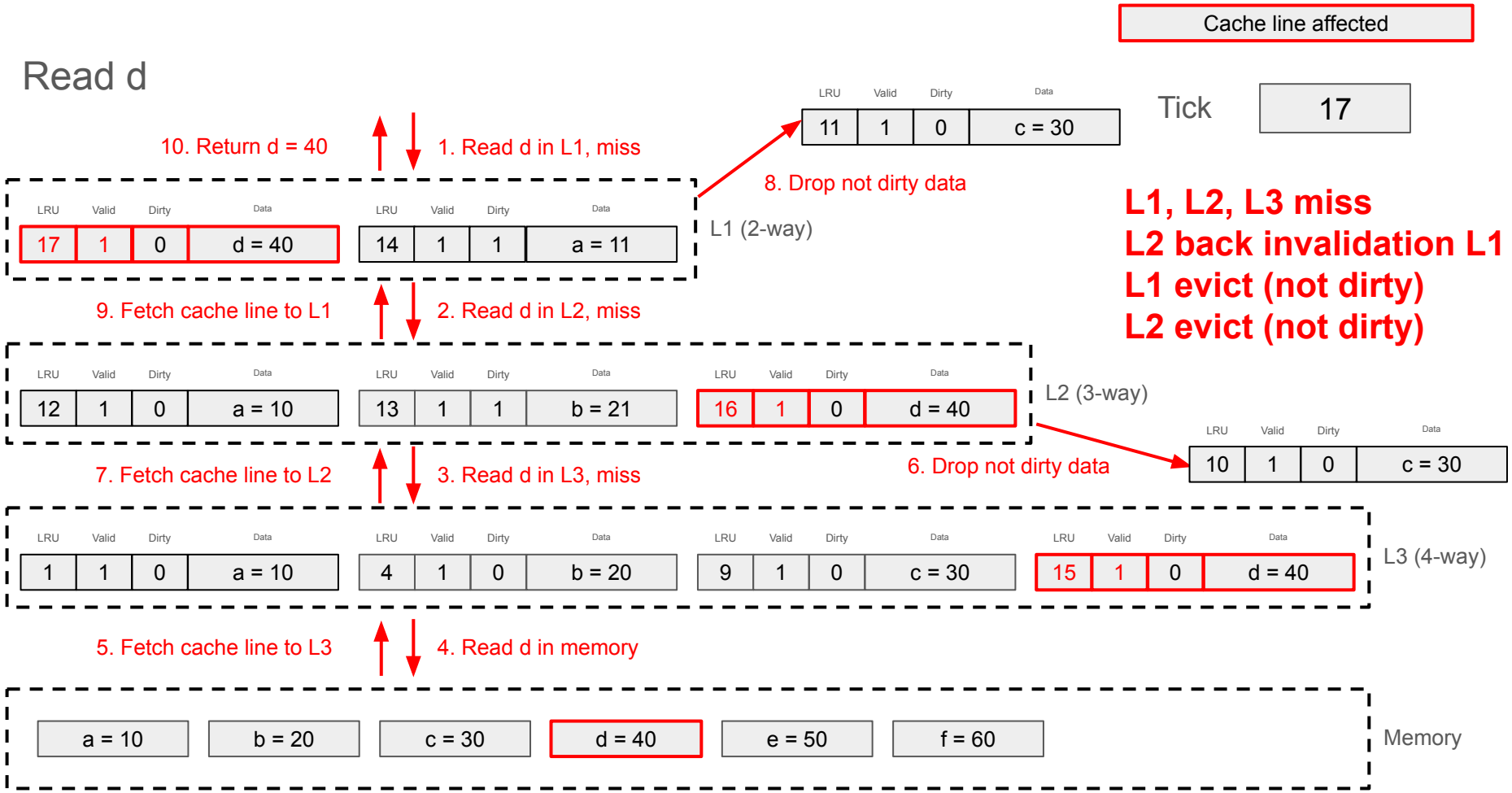
Cache line affected

Write a

Tick 14

L1 miss, L2 hit
L1 evict (dirty)
write back to L2



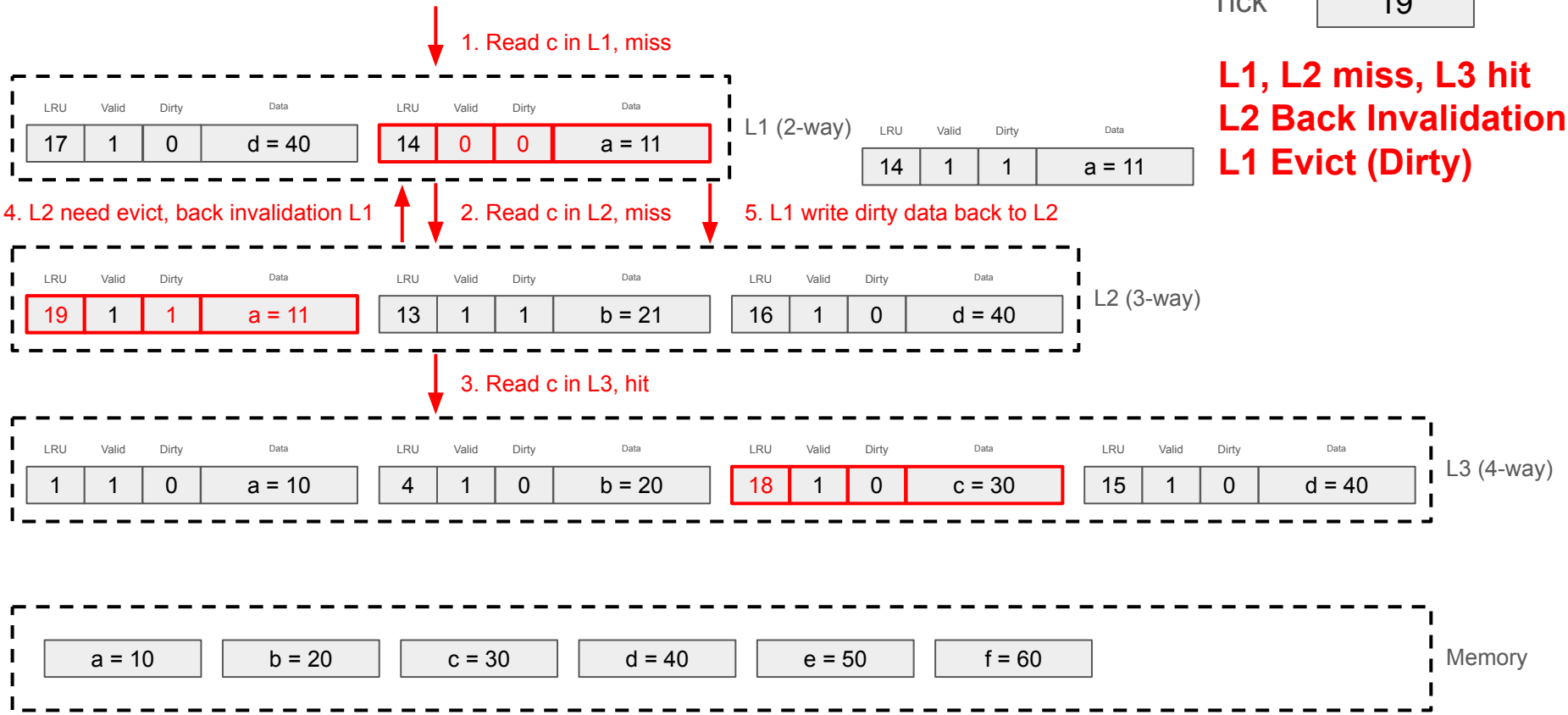

Cache line affected
Read d
Tick
17
LRU Valid Dirty Data
11 1 0 c = 30
10. Return d = 40
1. Read d in L1, miss
8. Drop not dirty data
LRU Valid Dirty Data
17 1 0 d = 40
14 1 1 a = 11
L1 (2-way)
L1, L2, L3 miss
L2 back invalidation L1
L1 evict (not dirty)
L2 evict (not dirty)
9. Fetch cache line to L1
2. Read d in L2, miss
LRU Valid Dirty Data
12 1 0 a = 10
13 1 1 b = 21
16 1 0 d = 40
L2 (3-way)
6. Drop not dirty data
LRU Valid Dirty Data
10 1 0 c = 30
7. Fetch cache line to L2
3. Read d in L3, miss
LRU Valid Dirty Data
1 1 0 a = 10
4 1 0 b = 20
9 1 0 c = 30
15 1 0 d = 40
L3 (4-way)
5. Fetch cache line to L3
4. Read d in memory
a = 10
b = 20
c = 30
d = 40
e = 50
f = 60
Memory

Read c (cont'd)

Cache line affected

Tick 19

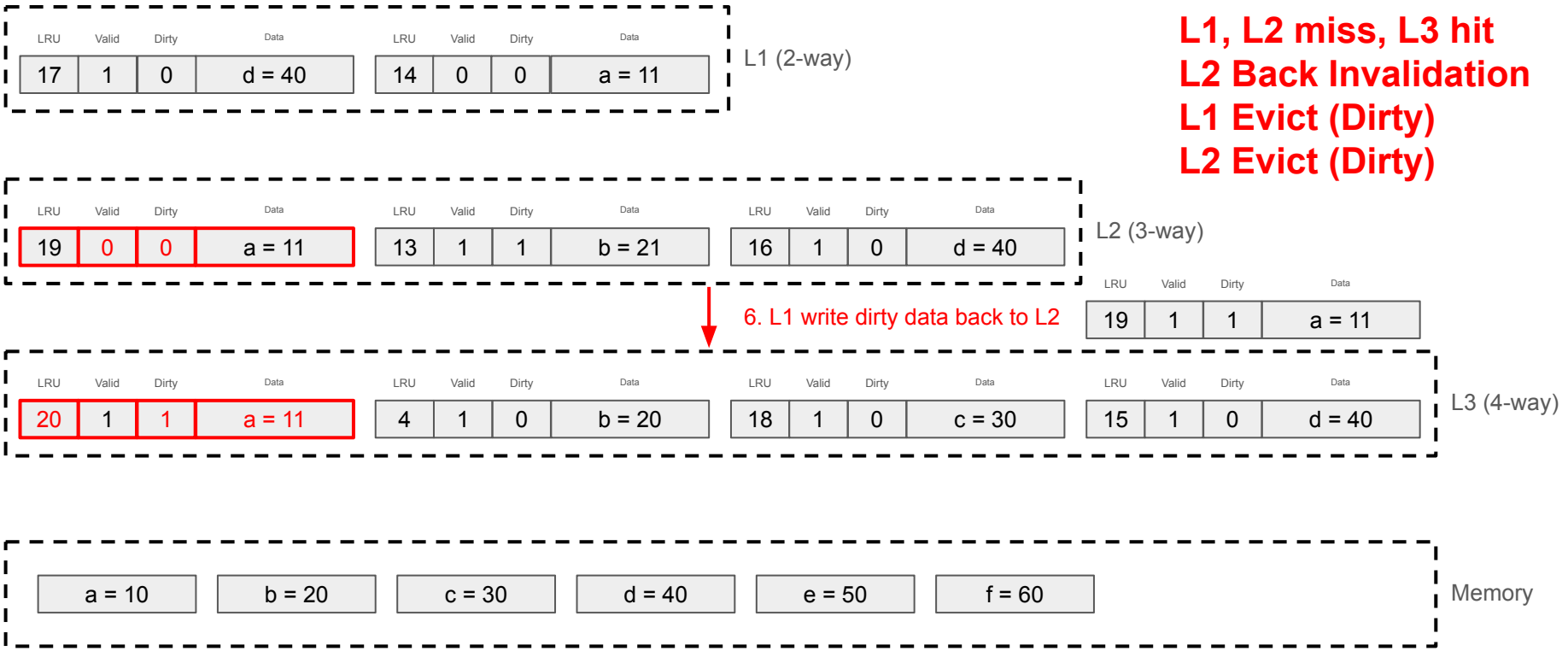
L1, L2 miss, L3 hit
L2 Back Invalidation
L1 Evict (Dirty)

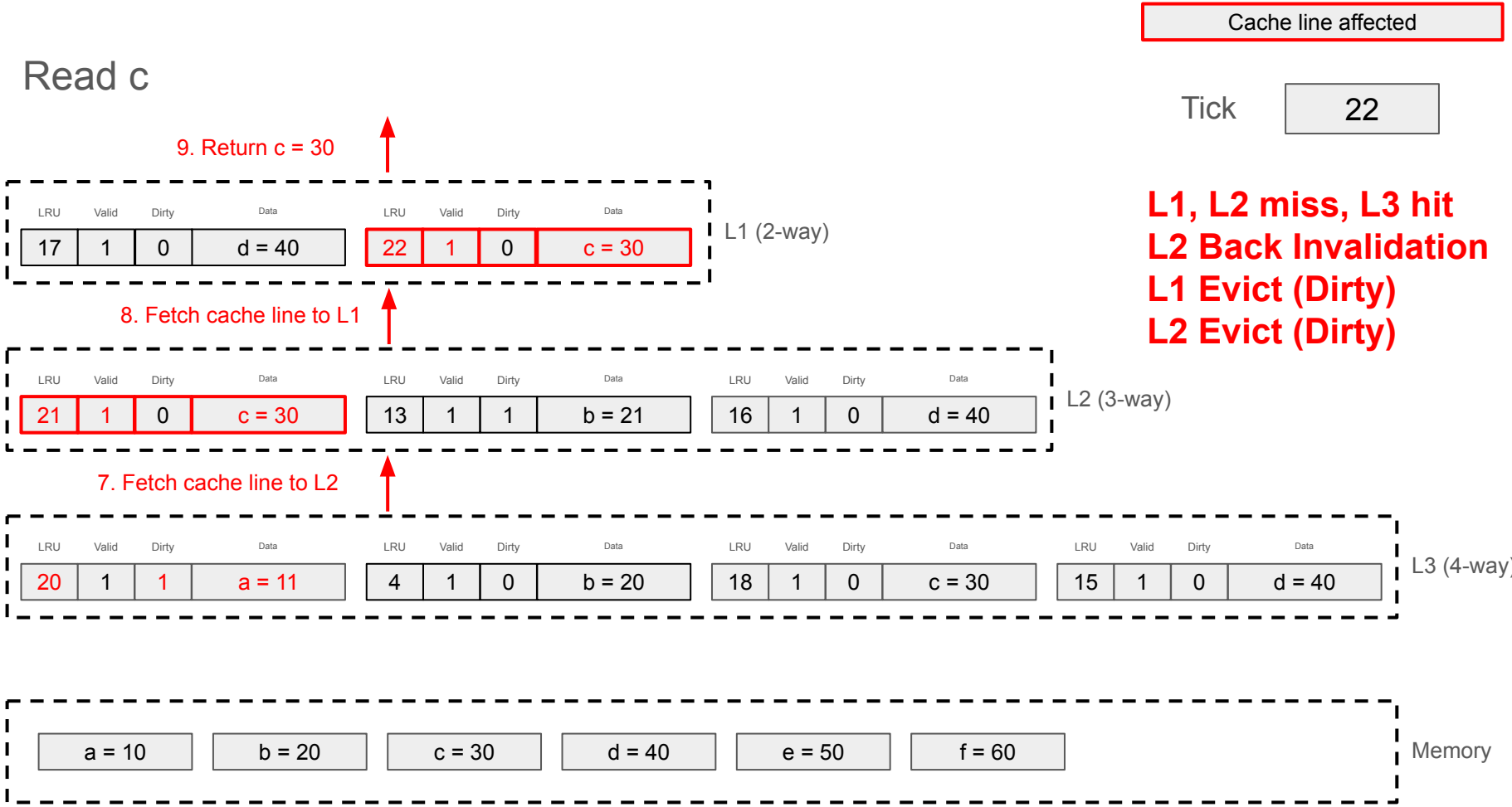


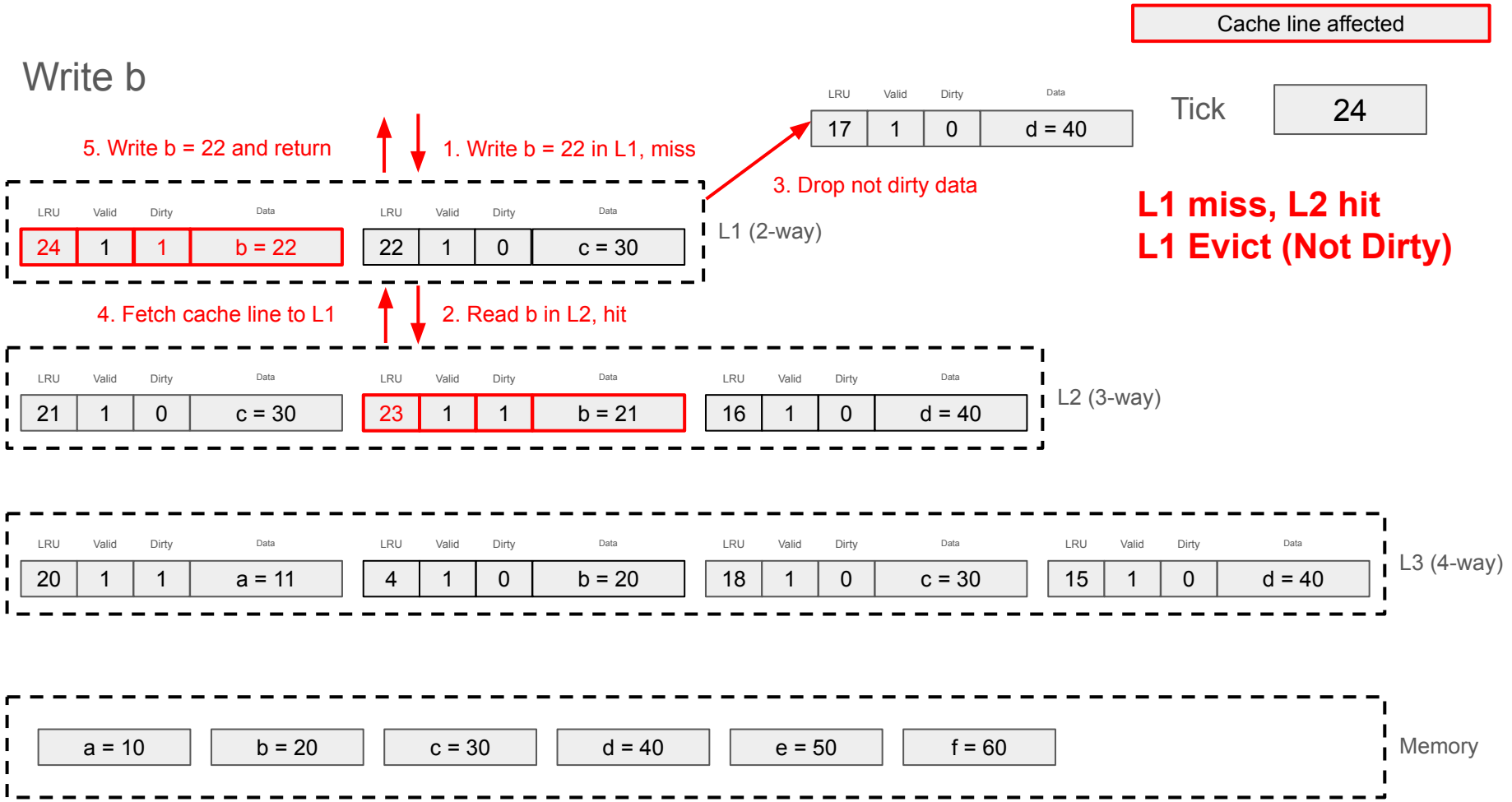
Read c (cont'd)

Cache line affected

Tick 20




Cache line affected
Read c
Tick
22
9. Return c = 30
LRU
Valid
Dirty
Data
17
1
0
d = 40
22
1
0
c = 30
L1 (2-way)
L1, L2 miss, L3 hit
L2 Back Invalidation
L1 Evict (Dirty)
L2 Evict (Dirty)
8. Fetch cache line to L1
21
1
0
c = 30
13
1
1
b = 21
16
1
0
d = 40
L2 (3-way)
7. Fetch cache line to L2
20
1
1
a = 11
4
1
0
b = 20
18
1
0
c = 30
15
1
0
d = 40
L3 (4-way)
a = 10
b = 20
c = 30
d = 40
e = 50
f = 60
Memory


Cache line affected
Write b
Tick
24
LRU
Valid
Dirty
Data
17
1
0
d = 40
5. Write b = 22 and return
1. Write b = 22 in L1, miss
3. Drop not dirty data
24
1
1
b = 22
22
1
0
c = 30
L1 (2-way)
L1 miss, L2 hit
L1 Evict (Not Dirty)
4. Fetch cache line to L1
2. Read b in L2, hit
21
1
0
c = 30
23
1
1
b = 21
16
1
0
d = 40
L2 (3-way)
20
1
1
a = 11
4
1
0
b = 20
18
1
0
c = 30
15
1
0
d = 40
L3 (4-way)
a = 10
b = 20
c = 30
d = 40
e = 50
f = 60
Memory

Cache line affected

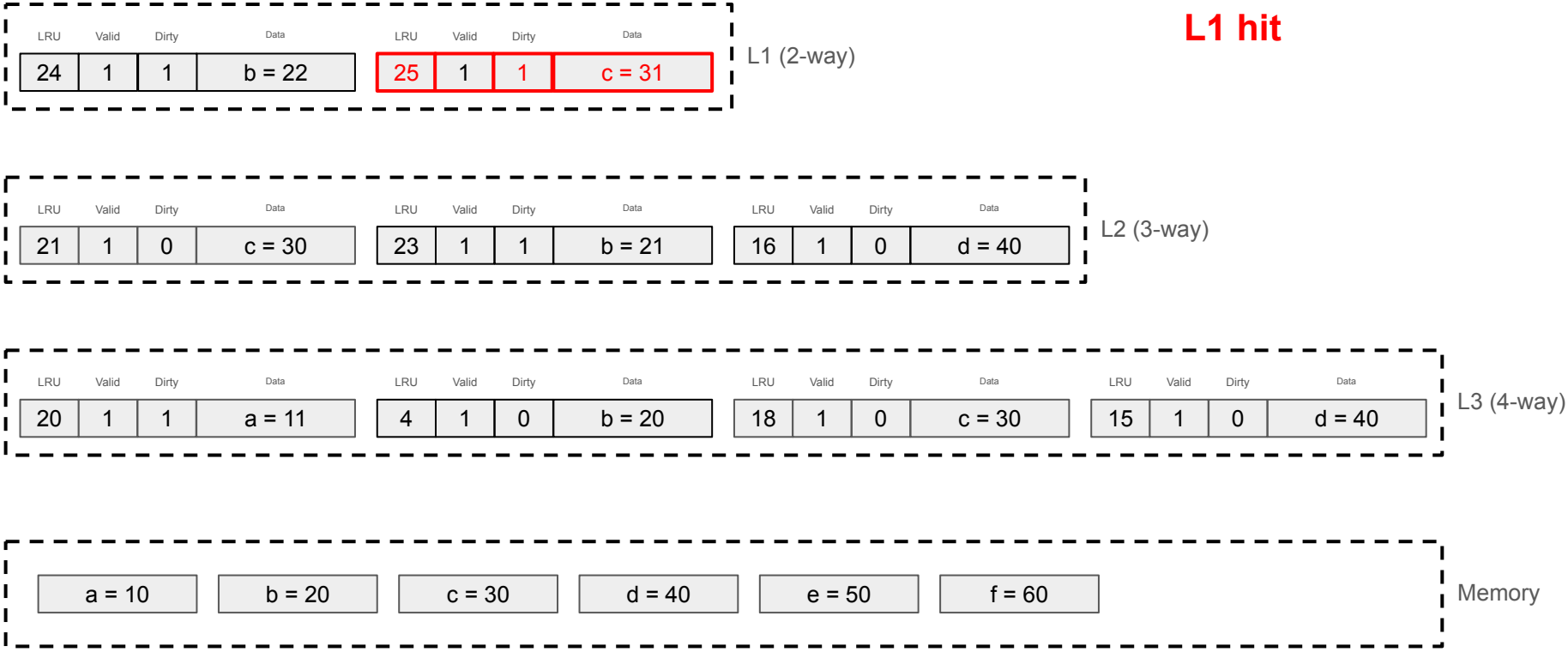
Write c

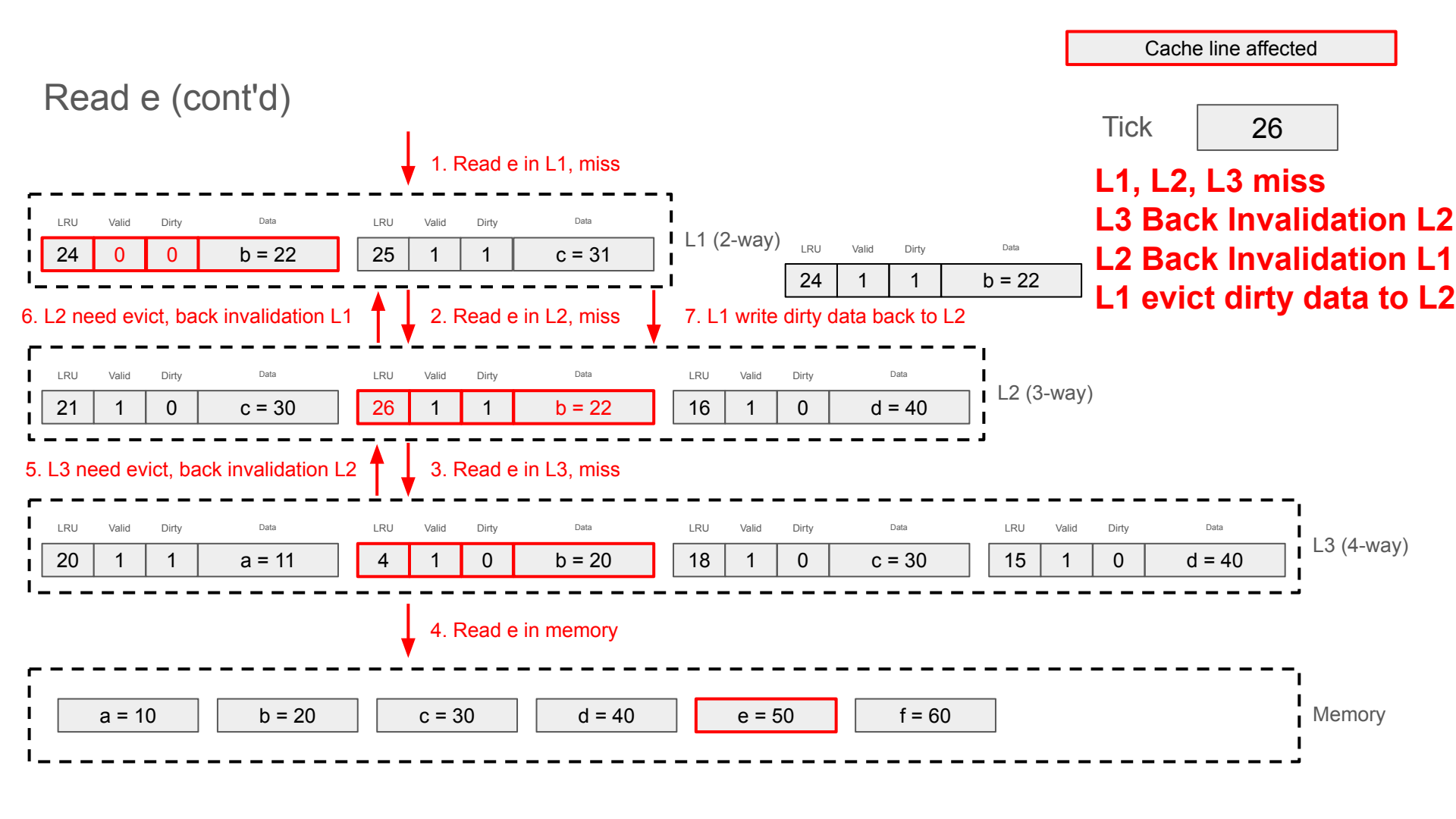
Tick 25

3. Write c = 31 and return

1. Write c = 31 in L1, hit

L1 hit



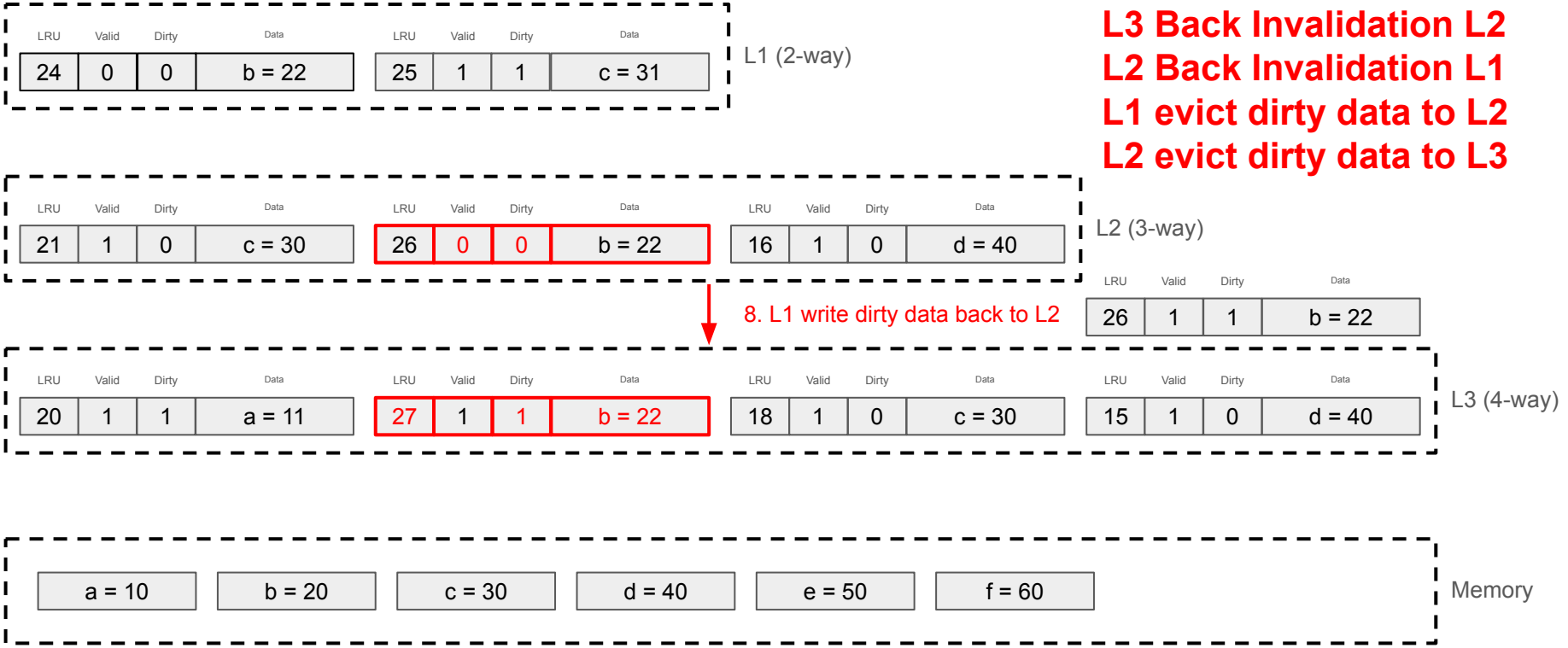

Cache line affected
Read e (cont'd)
Tick
26
1. Read e in L1, miss
L1, L2, L3 miss
L3 Back Invalidation L2
L2 Back Invalidation L1
L1 evict dirty data to L2
LRU Valid Dirty Data
24 0 0 b = 22
25 1 1 c = 31
L1 (2-way)
24 1 1 b = 22
6. L2 need evict, back invalidation L1
2. Read e in L2, miss
7. L1 write dirty data back to L2
21 1 0 c = 30
26 1 1 b = 22
16 1 0 d = 40
L2 (3-way)
5. L3 need evict, back invalidation L2
3. Read e in L3, miss
20 1 1 a = 11
4 1 0 b = 20
18 1 0 c = 30
15 1 0 d = 40
L3 (4-way)
4. Read e in memory
a = 10
b = 20
c = 30
d = 40
e = 50
f = 60
Memory

Read e (cont'd)

Cache line affected

Tick 27

L1, L2, L3 miss
L3 Back Invalidation L2
L2 Back Invalidation L1
L1 evict dirty data to L2
L2 evict dirty data to L3

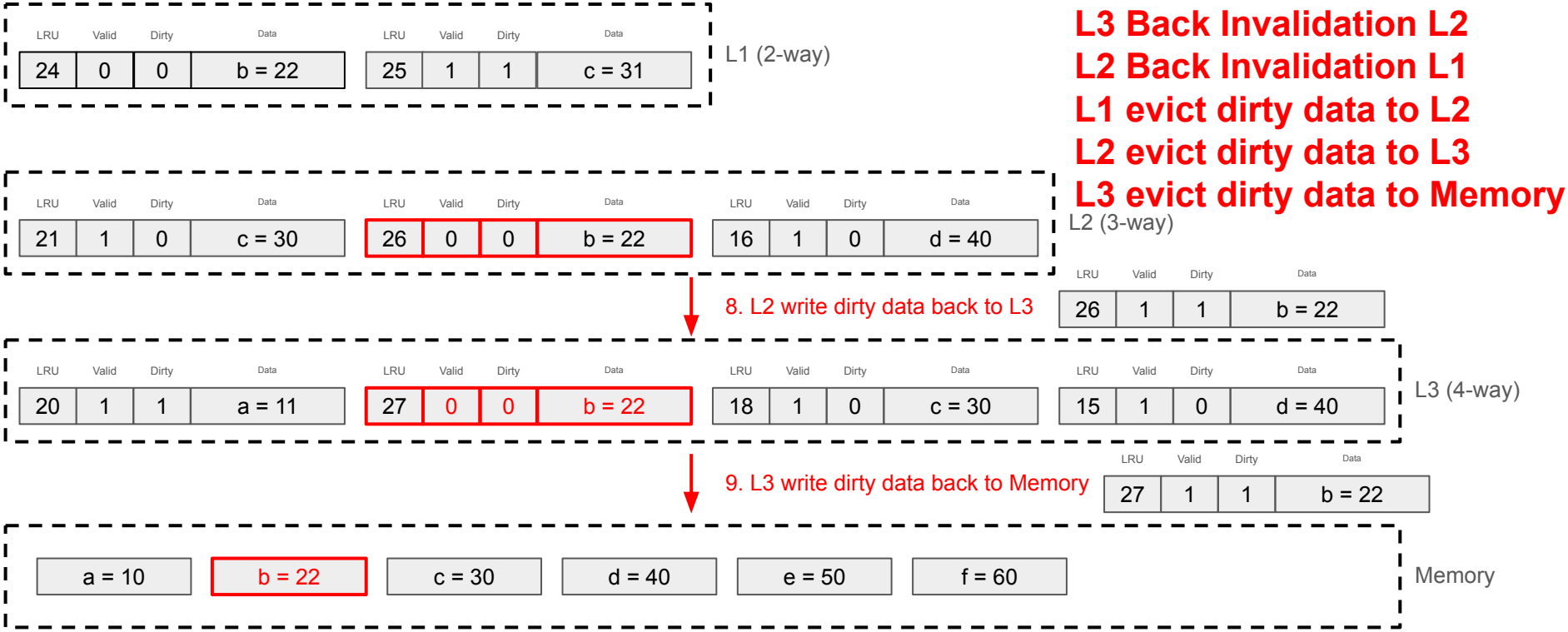


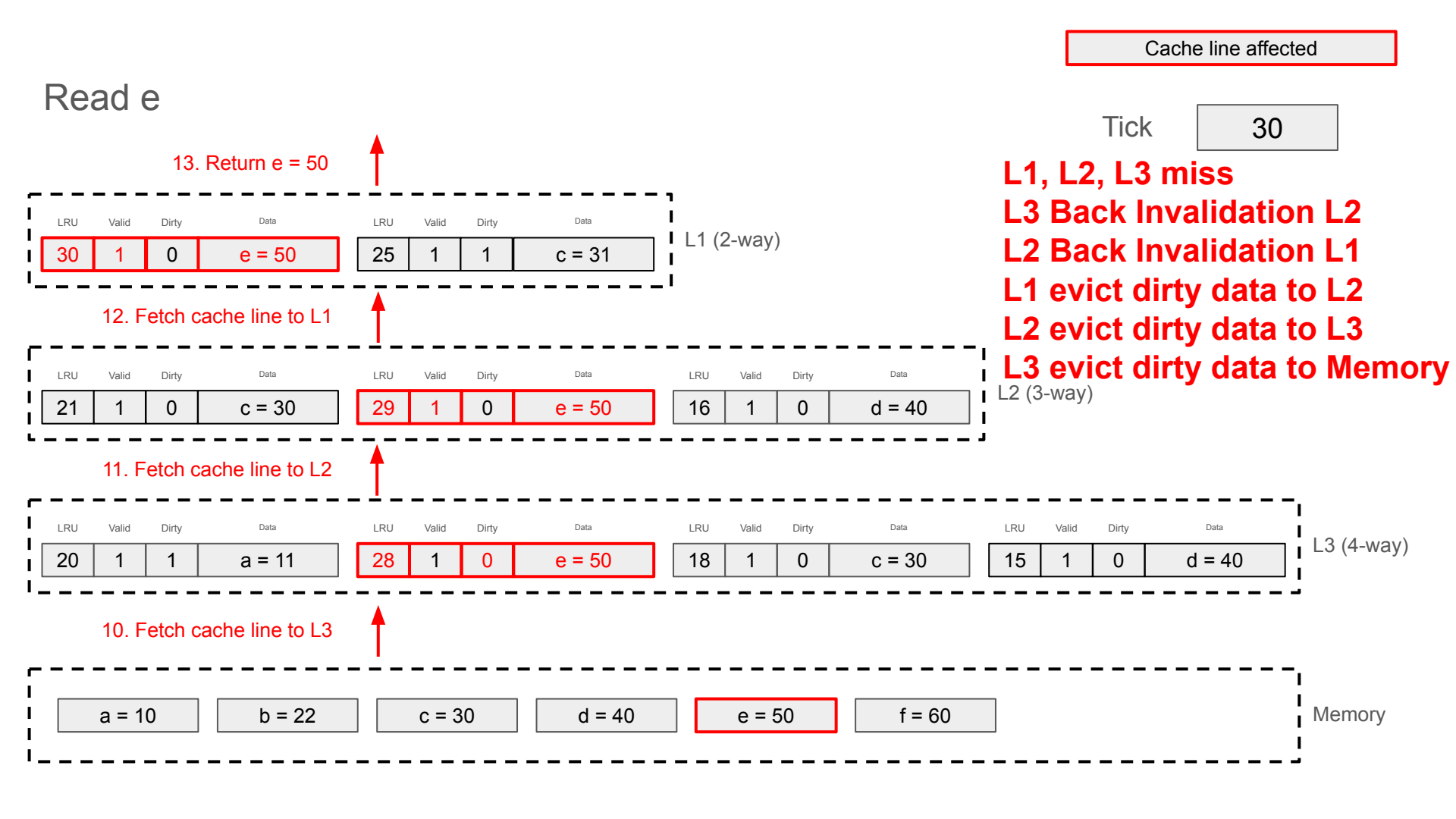
Read e (cont'd)

Cache line affected

Tick 27

L1, L2, L3 miss
L3 Back Invalidation L2
L2 Back Invalidation L1
L1 evict dirty data to L2
L2 evict dirty data to L3
L3 evict dirty data to Memory



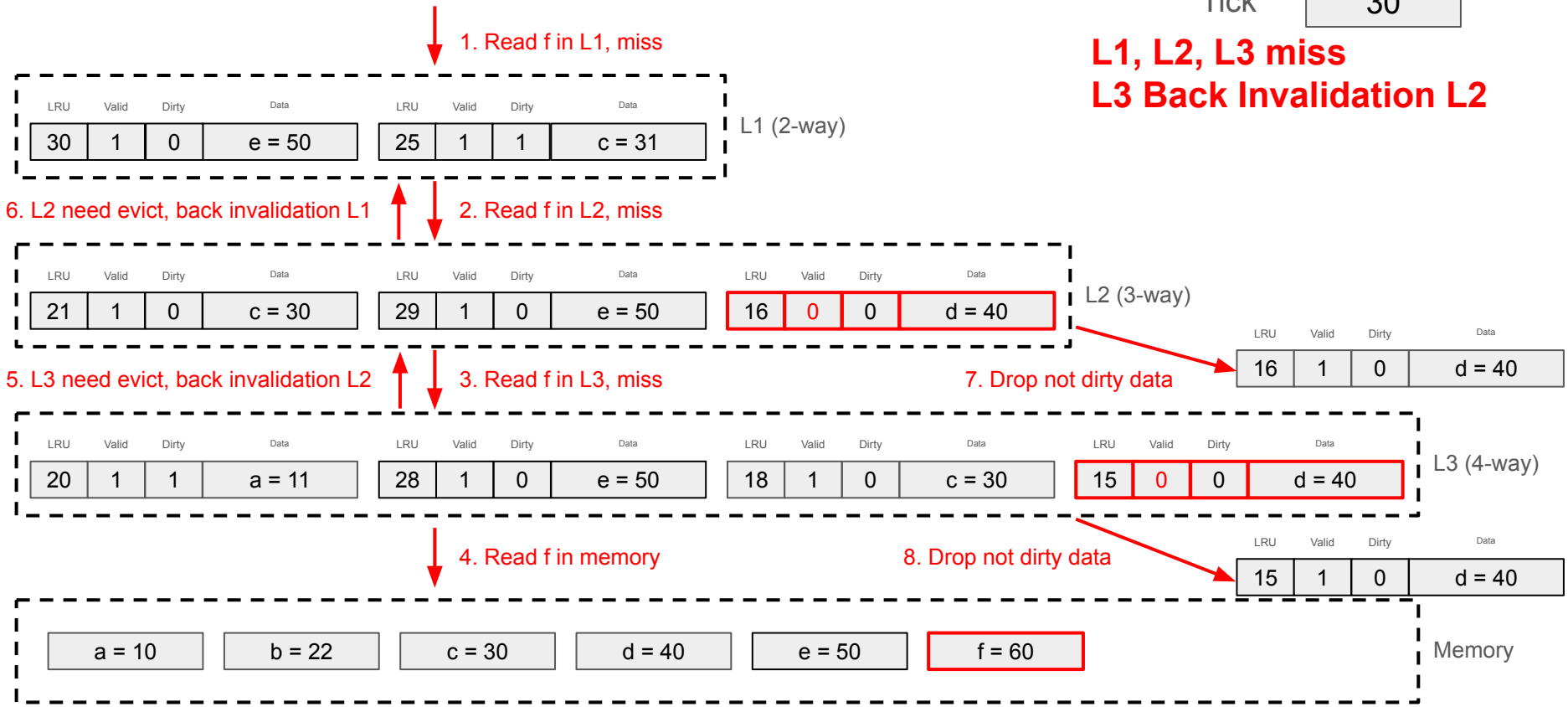

Cache line affected
Read e
Tick
30
L1, L2, L3 miss
L3 Back Invalidation L2
L2 Back Invalidation L1
L1 evict dirty data to L2
L2 evict dirty data to L3
L3 evict dirty data to Memory
13. Return e = 50
LRU Valid Dirty Data
30 1 0 e = 50
25 1 1 c = 31
L1 (2-way)
12. Fetch cache line to L1
LRU Valid Dirty Data
21 1 0 c = 30
29 1 0 e = 50
16 1 0 d = 40
L2 (3-way)
11. Fetch cache line to L2
LRU Valid Dirty Data
20 1 1 a = 11
28 1 0 e = 50
18 1 0 c = 30
15 1 0 d = 40
L3 (4-way)
10. Fetch cache line to L3
a = 10
b = 22
c = 30
d = 40
e = 50
f = 60
Memory

Read f (cont'd)

Cache line affected

Tick 30

L1, L2, L3 miss
L3 Back Invalidation L2

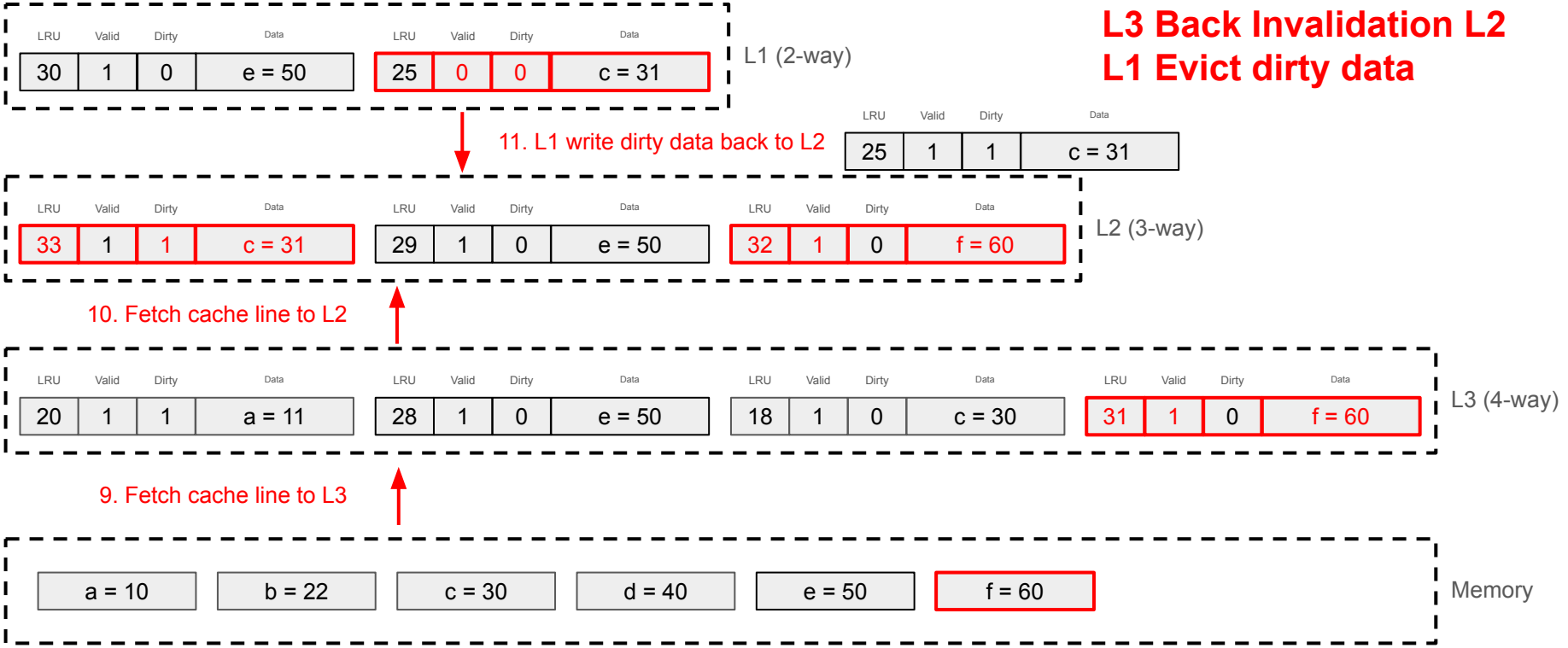


Cache line affected

Read f (cont'd)

Tick 30

L1, L2, L3 miss
L3 Back Invalidation L2
L1 Evict dirty data



Cache line affected

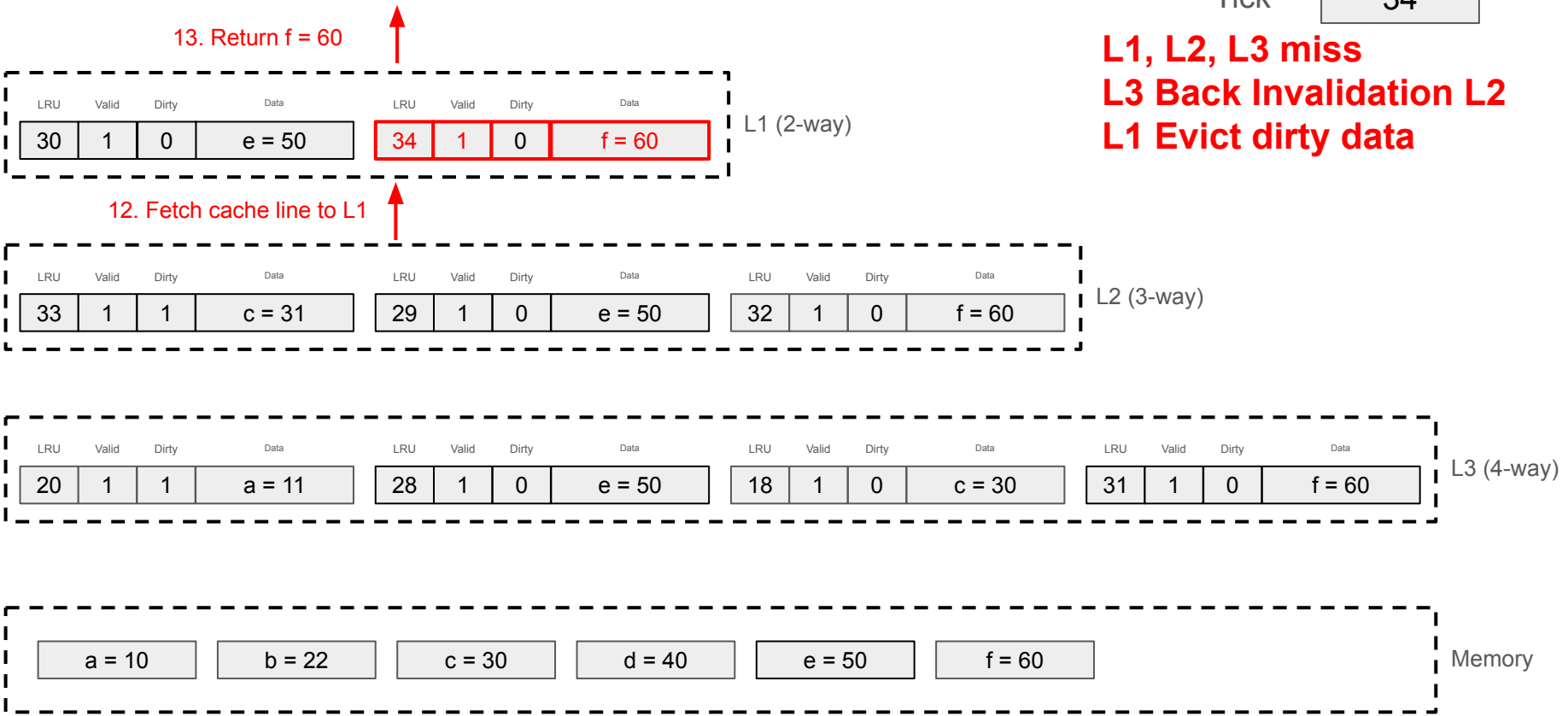
Read f

Tick 34

L1, L2, L3 miss
L3 Back Invalidation L2
L1 Evict dirty data

13. Return f = 60

12. Fetch cache line to L1

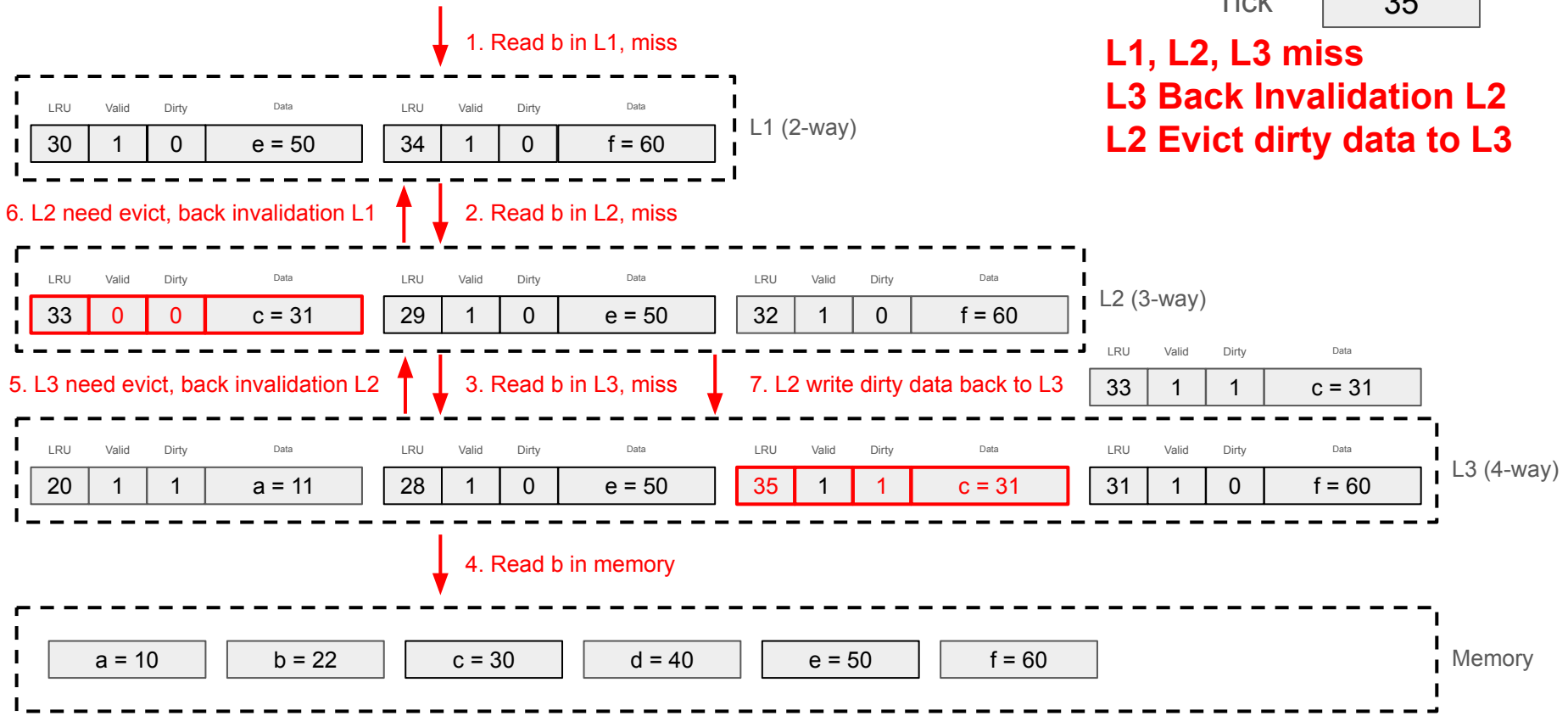


Read b (cont'd)

Cache line affected

Tick 35

L1, L2, L3 miss
L3 Back Invalidation L2
L2 Evict dirty data to L3

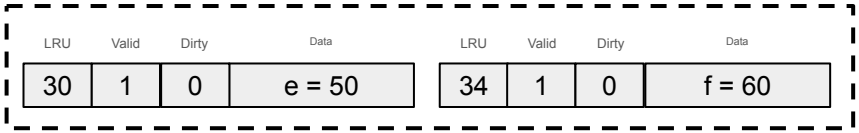


Read b (cont'd)

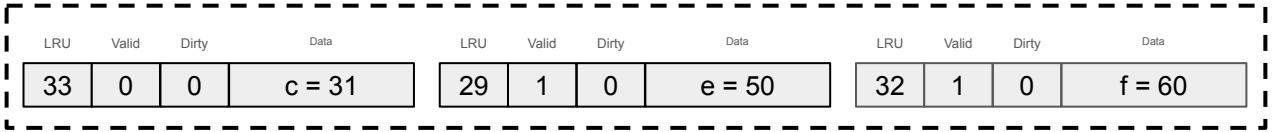
Cache line affected

Tick 34

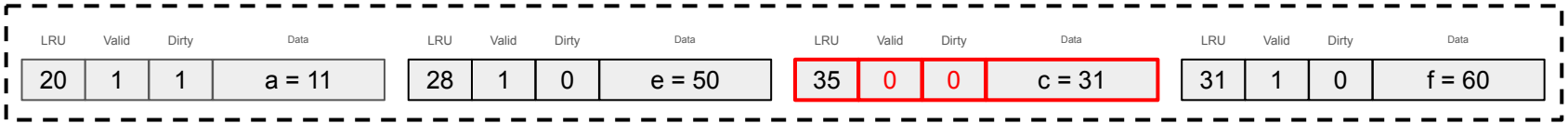
L1, L2, L3 miss
L3 Back Invalidation L2
L2 Evict dirty data to L3
L3 Evict dirty data to memory



L1 (2-way)

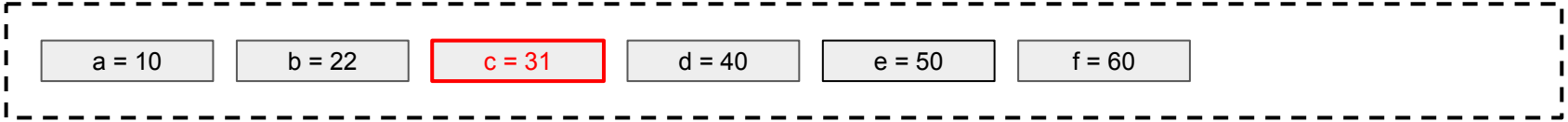


L2 (3-way)

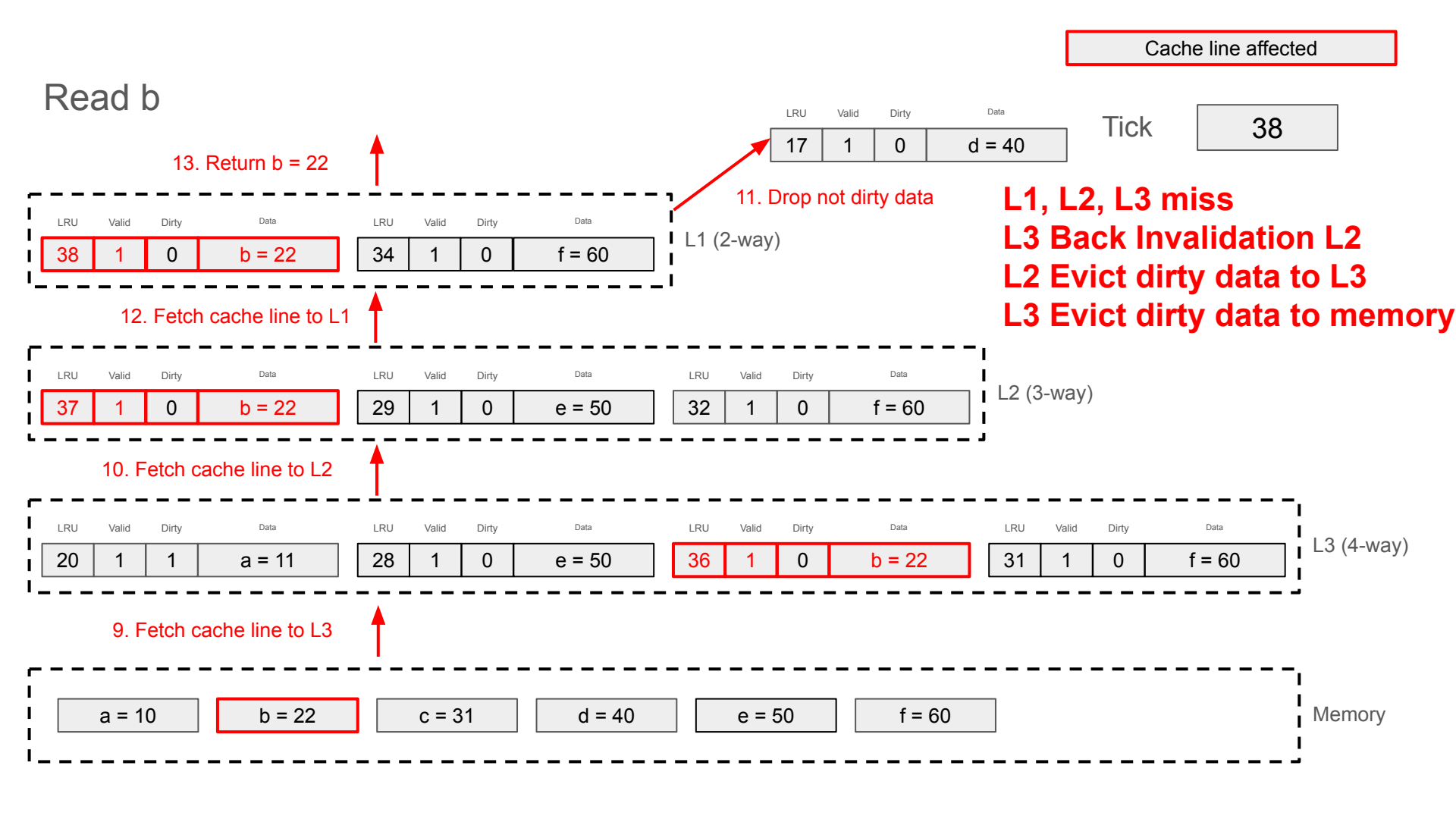


L3 (4-way)

8. L3 write dirty data back to Memory



Memory


Cache line affected
Read b
Tick
38
LRU
Valid
Dirty
Data
17
1
0
d = 40
13. Return b = 22
11. Drop not dirty data
38
1
0
b = 22
34
1
0
f = 60
L1 (2-way)
L1, L2, L3 miss
L3 Back Invalidation L2
L2 Evict dirty data to L3
L3 Evict dirty data to memory
12. Fetch cache line to L1
37
1
0
b = 22
29
1
0
e = 50
32
1
0
f = 60
L2 (3-way)
10. Fetch cache line to L2
20
1
1
a = 11
28
1
0
e = 50
36
1
0
b = 22
31
1
0
f = 60
L3 (4-way)
9. Fetch cache line to L3
a = 10
b = 22
c = 31
d = 40
e = 50
f = 60
Memory

Read d (cont'd)

Cache line affected

Tick 38

L1, L2, L3 miss
L3 Evict dirty data to memory

1. Read b in L1, miss

L1 (2-way)

LRU	Valid	Dirty	Data	LRU	Valid	Dirty	Data
38	1	0	b = 22	34	1	0	f = 60

2. Read b in L2, miss

L2 (3-way)

LRU	Valid	Dirty	Data	LRU	Valid	Dirty	Data	LRU	Valid	Dirty	Data
37	1	0	b = 22	29	1	0	e = 50	32	1	0	f = 60

3. Read b in L3, miss

L3 (4-way)

LRU	Valid	Dirty	Data	LRU	Valid	Dirty	Data	LRU	Valid	Dirty	Data	LRU	Valid	Dirty	Data
20	0	0	a = 11	28	1	0	e = 50	36	1	0	b = 22	31	1	0	f = 60

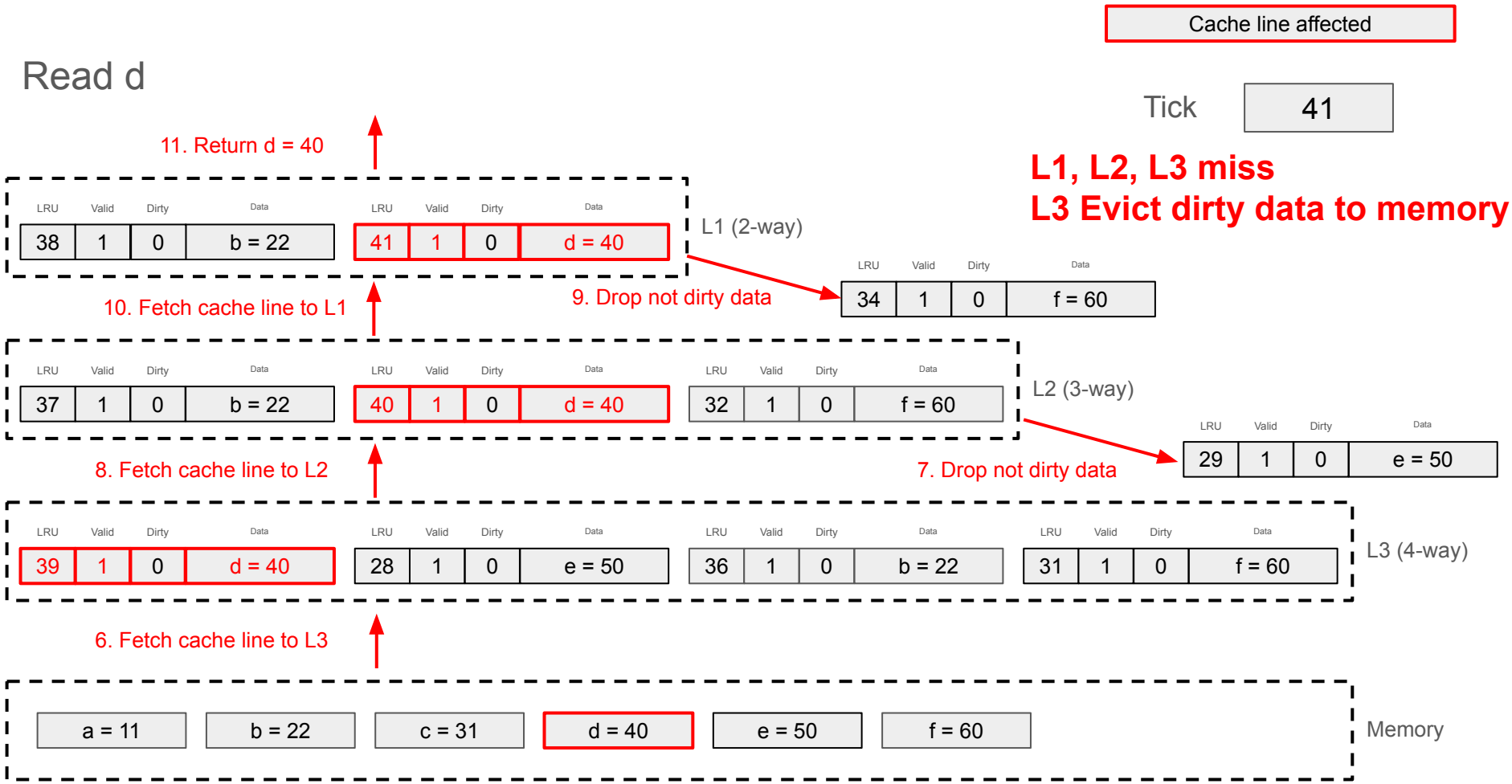
4. Read b in memory

5. L3 write dirty data back to Memory

LRU	Valid	Dirty	Data
20	1	1	a = 11

Memory

a = 11	b = 22	c = 31	d = 40	e = 50	f = 60


Cache line affected
Read d
Tick
41
L1, L2, L3 miss
L3 Evict dirty data to memory
11. Return d = 40
LRU
Valid
Dirty
Data
38
1
0
b = 22
41
1
0
d = 40
L1 (2-way)
9. Drop not dirty data
34
1
0
f = 60
10. Fetch cache line to L1
37
40
32
L2 (3-way)
29
e = 50
7. Drop not dirty data
8. Fetch cache line to L2
39
28
36
31
L3 (4-way)
6. Fetch cache line to L3
a = 11
b = 22
c = 31
d = 40
e = 50
f = 60
Memory