0x40188: 0100 0000 0001 1000 1000

0xA0104: 1010 0000 0001 0000 0100

Writing data to caches

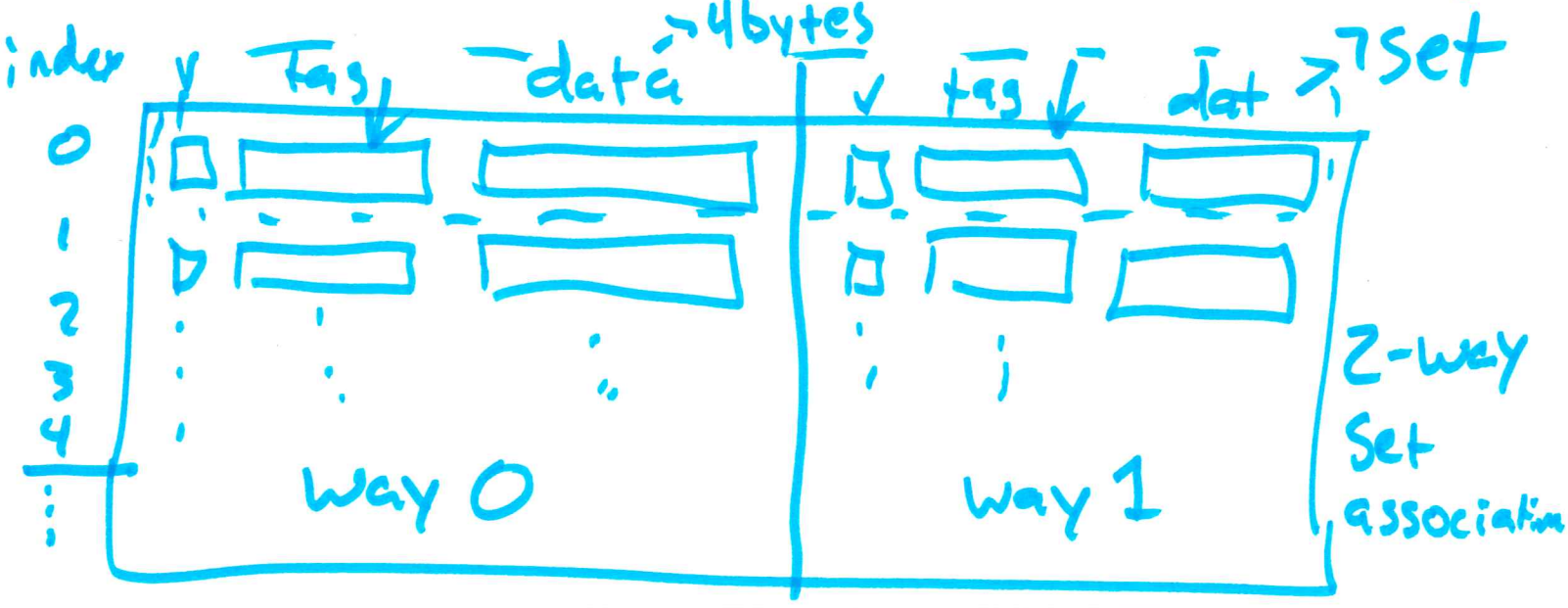
2 policies: write-through

write-back

Problem w/ direct mapped caches:

Lots of conflict misses

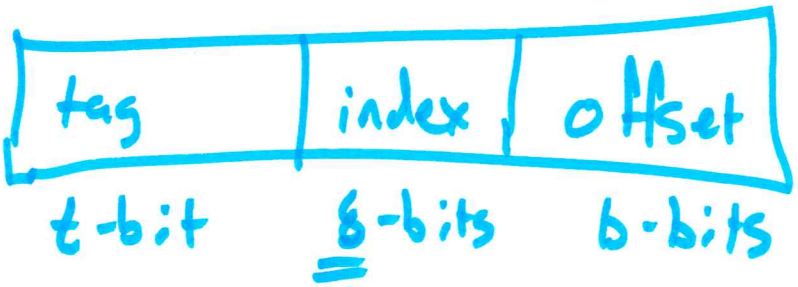
Solve this with set-associative caches



0xA0104: 1010 0000 0001 0000 01 00

0x40184: 0100 0000 0001 1000 01 00

General cache organization!



sets → $2^s = S$

E → lines or blocks per set (E-way SA)

B → block size (bytes) $= 2^b$

M → total physical address bits $m = t + s + b$

$= \log_2(M)$

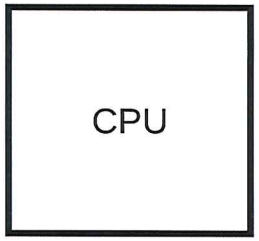
M size of main memory → 2^m

$s \rightarrow \log_2(S)$ # of bits for index

$b = \log_2(B)$ " " offset

$t = m - s - b$ " " tag

C cache size $= B \cdot E \cdot S$




```
movl 8(%ebp), %ecx
top:
movb (%ecx,%eax), %al
testb %al, %al
je end
incl %eax
jmp top
end:
movl %eax, 8(%ebp)
```

Address stream

0x40188
0xA0104
0xA0105
0xA0106
0xA0107
0xA0108
0xA0109
0xA010A
0xA010B
0xA010C
0xA010D
0xA010E
0xA010F
0x40188

Cache

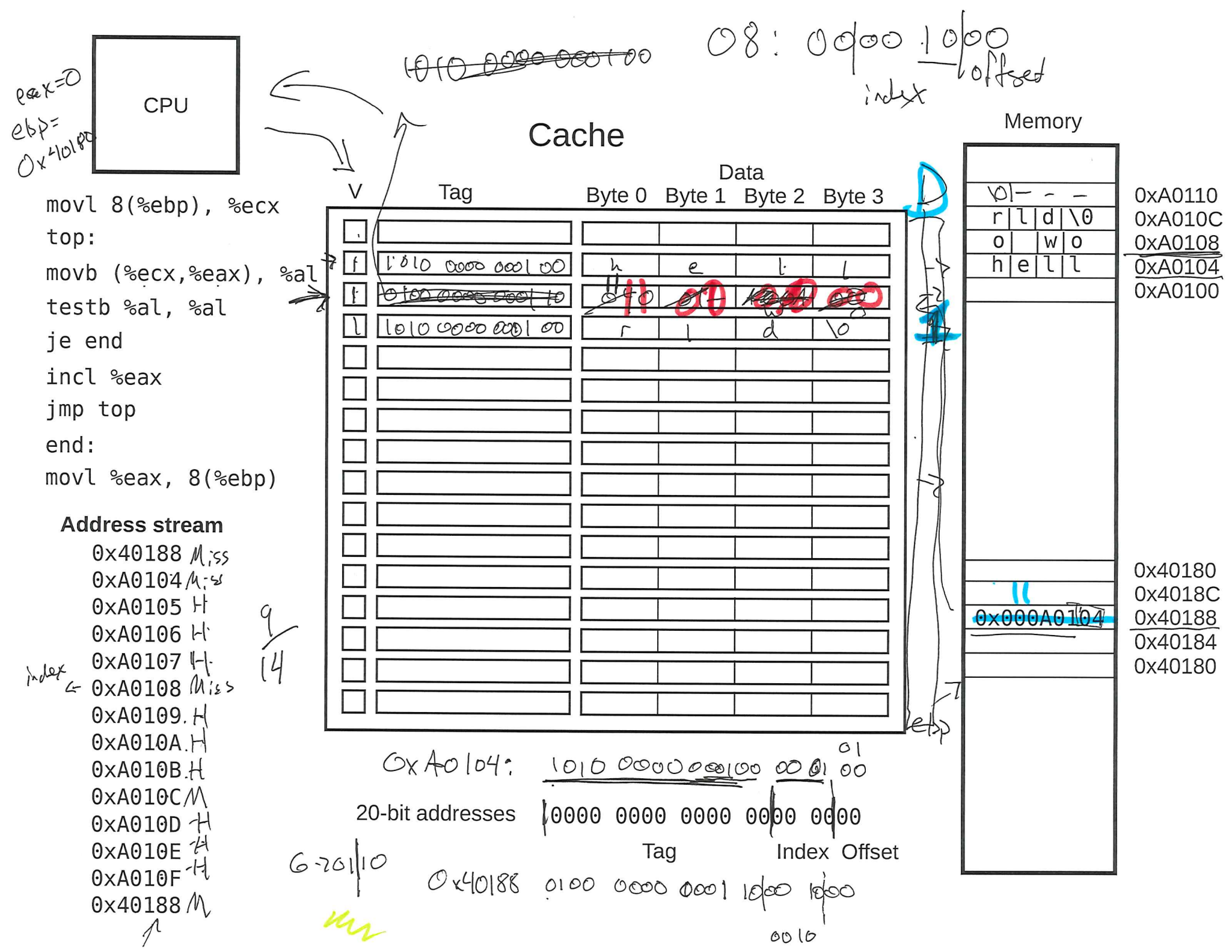
V	Tag	Data: Byte 0	Byte 1	Byte 2	Byte 3
1	1010 0000 0001 00	h	e	l	l
1	1010 0000 0001 00	o	_	w	o

20-bit addresses 0000 0000 0000 0000 0000

Tag | Index | Offset

Memory

	0xA0110
r | l | d | \0	0xA010C
o | _ | w | o	0xA0108
h | e | l | l	0xA0104
	0xA0100
	0x40180
	0x4018C
0x000A0104	0x40188
	0x40184
	0x40180


CPU
eax=0
ebp=
0x40188
Cache
Memory
V
Tag
Data
Byte 0
Byte 1
Byte 2
Byte 3
movl 8(%ebp), %ecx
top:
movb (%ecx,%eax), %al
testb %al, %al
je end
incl %eax
jmp top
end:
movl %eax, 8(%ebp)
Address stream
0x40188 Miss
0xA0104 Miss
0xA0105 H
0xA0106 H
0xA0107 H
0xA0108 Miss
0xA0109 H
0xA010A H
0xA010B H
0xA010C M
0xA010D H
0xA010E H
0xA010F H
0x40188 M
0xA0110
0xA010C
0xA0108
0xA0104
0xA0100
0x40180
0x4018C
0x40188
0x40184
0x40180
0x000A0104
ebp
0xA0104:
20-bit addresses
Tag
Index
Offset
0x40188