Direct - mapped cache

memory w/ 4 bit address: 2^4 bytes of data

B = 2: 2 bytes / block

S = 4: 4 sets

E = 1: 1 block/set

$B = 2^1$ → 1 bit offset $b = 1$

4 bits → 2 middle bits $s = 2$

most impt bit $t = 1$

t=1	s=2	b=1
x	x x	x

	V	Tag	Block
Set 0	~~1~~ 1	~~0~~ 1	~~M[0-1]~~ M[8-9]
Set 1			
Set 2			
Set 3	1	0	M[6-7]

first go, cold miss. nothing there
now there

miss b/c conflicts w/ memory address 8

address trace

1. 0 → $[0\ \underline{00}\ 0_2]$ miss cold
   ↳ s=0
2. 1 → $[0\ \underline{00}\ 1_2]$ hit
3. 7 → $[0\ 11\ 1_2]$ miss cold
   tag
4. 8 → $[1\ 00\ 0_2]$ miss cold
   same as 0 & 1 address
5. 0 → $[0\ \underline{00}\ 0_2]$ miss conflict

2 way associative → 2 sets w/ 2 blocks

now only need 1 bit to describe set

t=2	s=1	b=1
x x	x	x

M=4 bit addresses B= 2bytes/block S= 2sets E= 2 blocks/set

Address trace (reads, one byte per read)

0	$0\,0\,\underline{0}\,0_2$	① miss	cold
1	$0\,0\,\underline{0}\,1_2$	② hit	
7	$0\,1\,\underline{1}\,1_2$	③ miss	
8	$1\,0\,\underline{0}\,0_2$	④ miss	
0	$0\,0\,\underline{0}\,0_2$	⑤ hit	

	V	Tag	Block
Set 0	0 1 ①②⑤	00	M[0-1]
	0 1 ④	01	M[8-9]
Set 1	0 1 ③	01	M[6-7]
	0		

Associativity vs Missing

More cache misses
simpler lookups

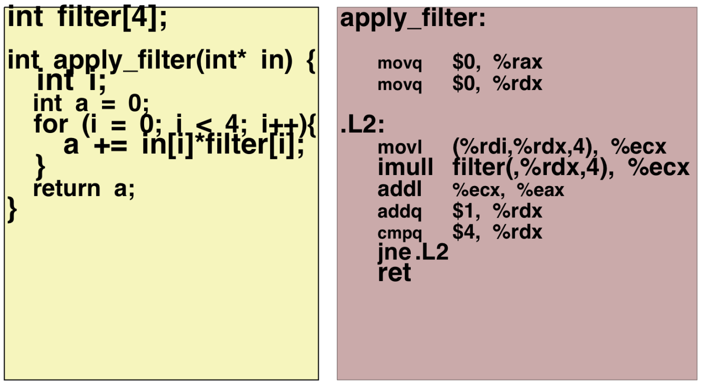
less cache misses
more complex lookups

1 set

full set

① address trace

② simulate cache misses/hits



```
int filter[4];

int apply_filter(int* in) {
   int i;
   int a = 0;
   for (i = 0; i < 4; i++){
      a += in[i]*filter[i];
   }
   return a;
}
```

```
apply_filter:

    movq    $0, %rax
    movq    $0, %rdx
.L2:
    movl    (%rdi,%rdx,4), %ecx
    imull   filter(,%rdx,4), %ecx
    addl    %ecx, %eax
    addq    $1, %rdx
    cmpq    $4, %rdx
    jne .L2
    ret
```

Slides have good ex.