Cache Conflicts

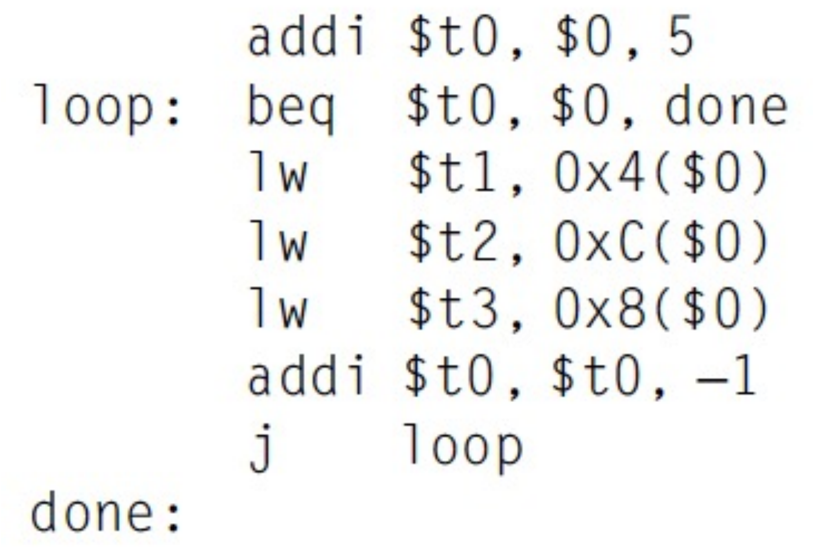
Cache Conflict

When two recently accessed addresses map to the same block, a conflict occurs.

Most recently used address evicts the previous one from the block.

Two addresses that map to the same set always cause a conflict in direct mapping.

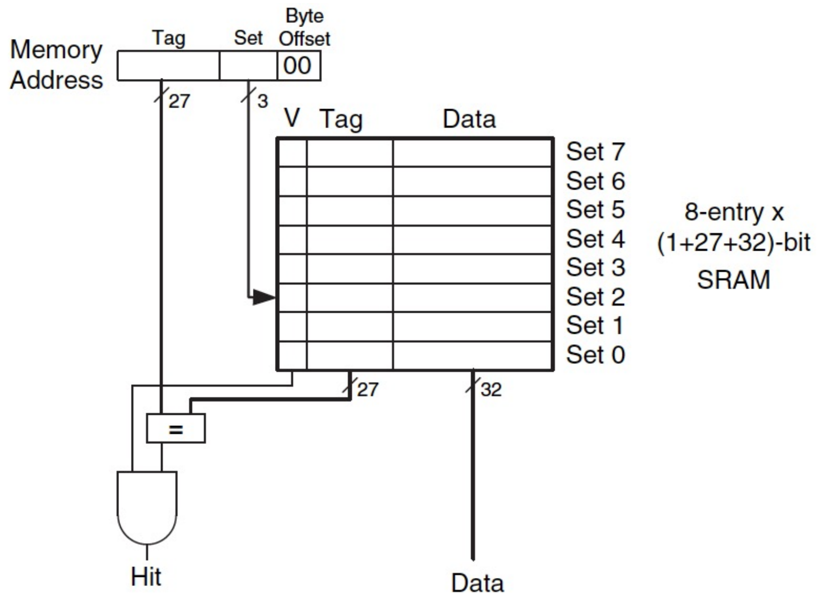
Use the 8-entry cache to see how the cache works given the following loop.

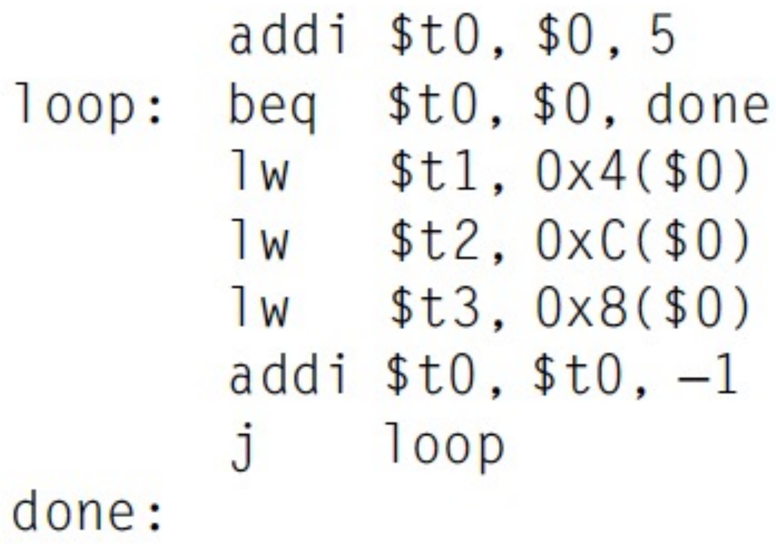


```
      addi $t0, $0, 5
loop: beq  $t0, $0, done
      lw   $t1, 0x4($0)
      lw   $t2, 0xC($0)
      lw   $t3, 0x8($0)
      addi $t0, $t0, -1
      j    loop
done:
```

What does the cache look like at the end of the iterations?

What is the miss rate?





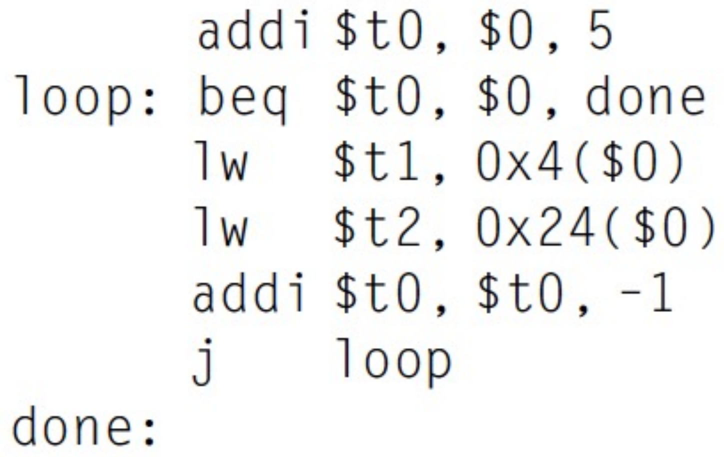
```
      addi $t0, $0, 5
loop: beq  $t0, $0, done
      lw   $t1, 0x4($0)
      lw   $t2, 0xC($0)
      lw   $t3, 0x8($0)
      addi $t0, $t0, -1
      j    loop
done:
```

V	Tag	Data

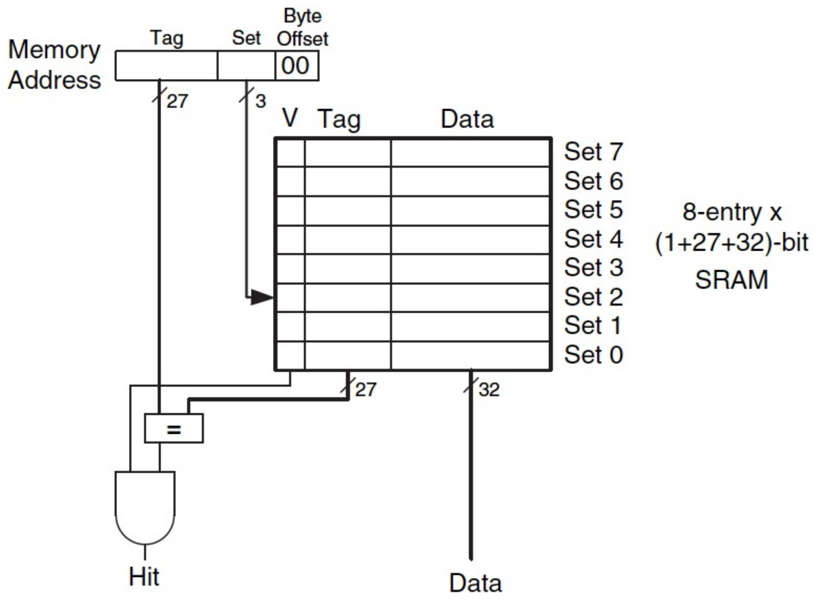
Use the 8-entry cache to see how the cache works given the following loop.

What does the cache look like at the end of the iterations?

What is the miss rate?



```
       addi $t0, $0, 5
loop:  beq  $t0, $0, done
       lw   $t1, 0x4($0)
       lw   $t2, 0x24($0)
       addi $t0, $t0, -1
       j    loop
done:
```


Memory Address
Tag
Set
Byte Offset
00
27
3
V
Tag
Data
Set 7
Set 6
Set 5
Set 4
Set 3
Set 2
Set 1
Set 0
8-entry x (1+27+32)-bit SRAM
27
32
=
Hit
Data