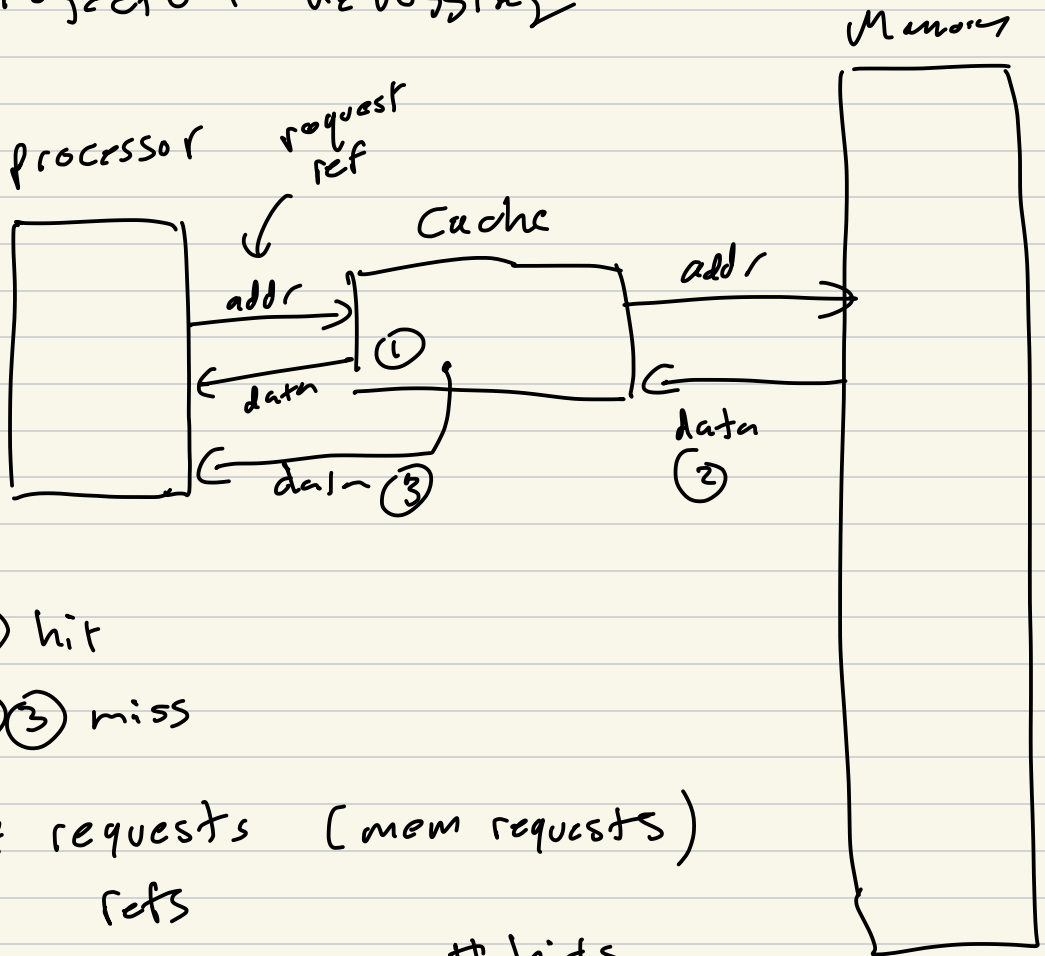


CS315-02 Cache Simulation

Project 04 debugging



$$\text{hit rate} = \frac{\# \text{ hits}}{\# \text{ reqs}}$$

$$\text{miss rate} = \frac{\# \text{ misses}}{\# \text{ reqs}}$$

$$\text{hit rate} = 1 - \text{miss rate}$$

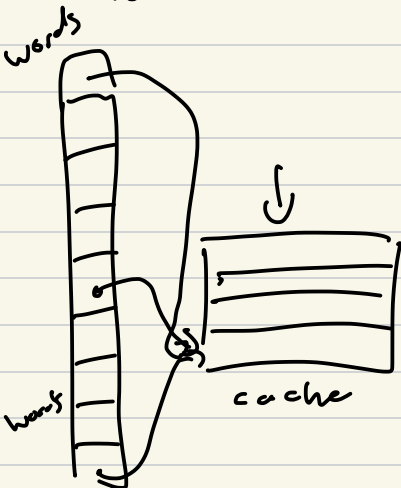
Direct Mapped Cache



addr (byte) assume addr is word aligned

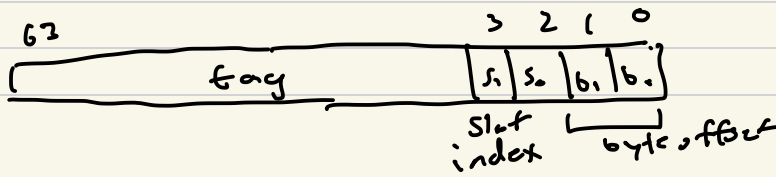
$$\text{addr_word} = \frac{\text{addr}}{4} \quad (\text{byte})$$

$$\text{slot_index} = \text{addr_word} \% 4$$



$$\begin{array}{ll} \text{addr_word} & \\ 4 & 4 \% 4 = 0 \\ 12 & 12 \% 4 = 0 \end{array}$$

byte addr word addr



$slot_index = (addr \gg 2) \& 0b11$

Direct Mapped Cache Pseudo Code

$dm_lookup(addr)$

$tag = addr \gg 4;$

$index_mask = 0b11;$

$slot_index = (addr \gg 2) \& index_mask$

$slot = cache[slot_index]$

$if (slot.valid == 1 \& \& slot.tag == tag) \{$
 // hit

 return slot.data

$\} else \{$

 // miss

$slot.data = *(uint32_t *)addr;$

$slot.tag = tag$

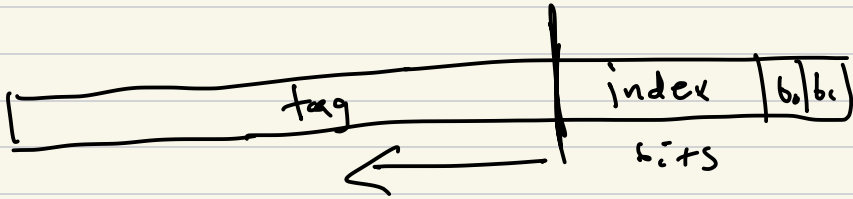
$slot.valid = 1$

 return slot.data

$\}$

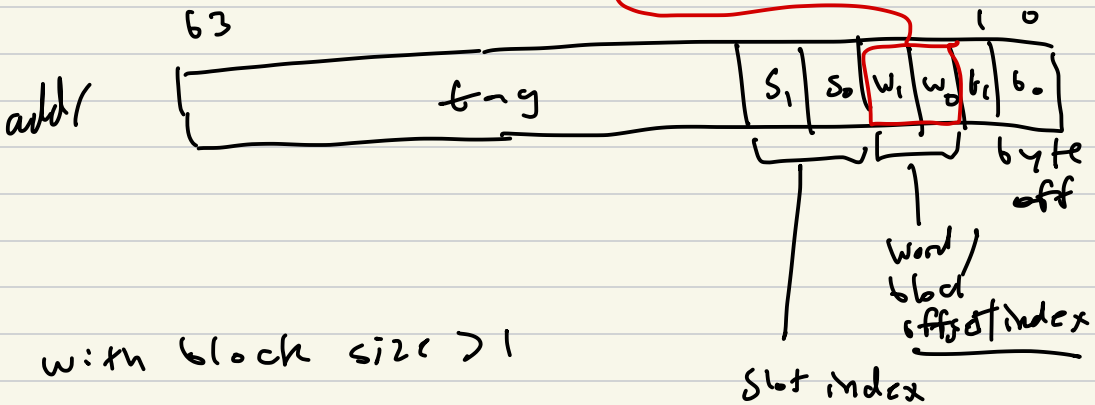
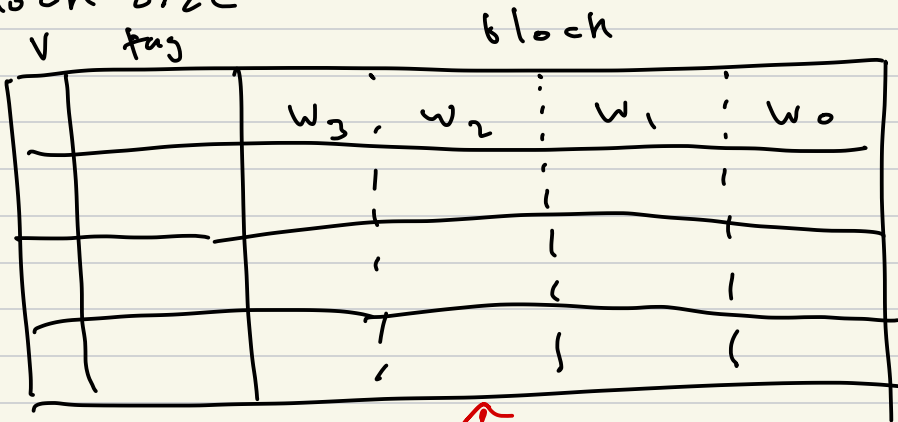
of Slots

4	slots	(2 bits)
8	slots	(3 bits)
16	slots	(4 bits)
128	slots	(7 bits)



Block size = 4 words

Block size



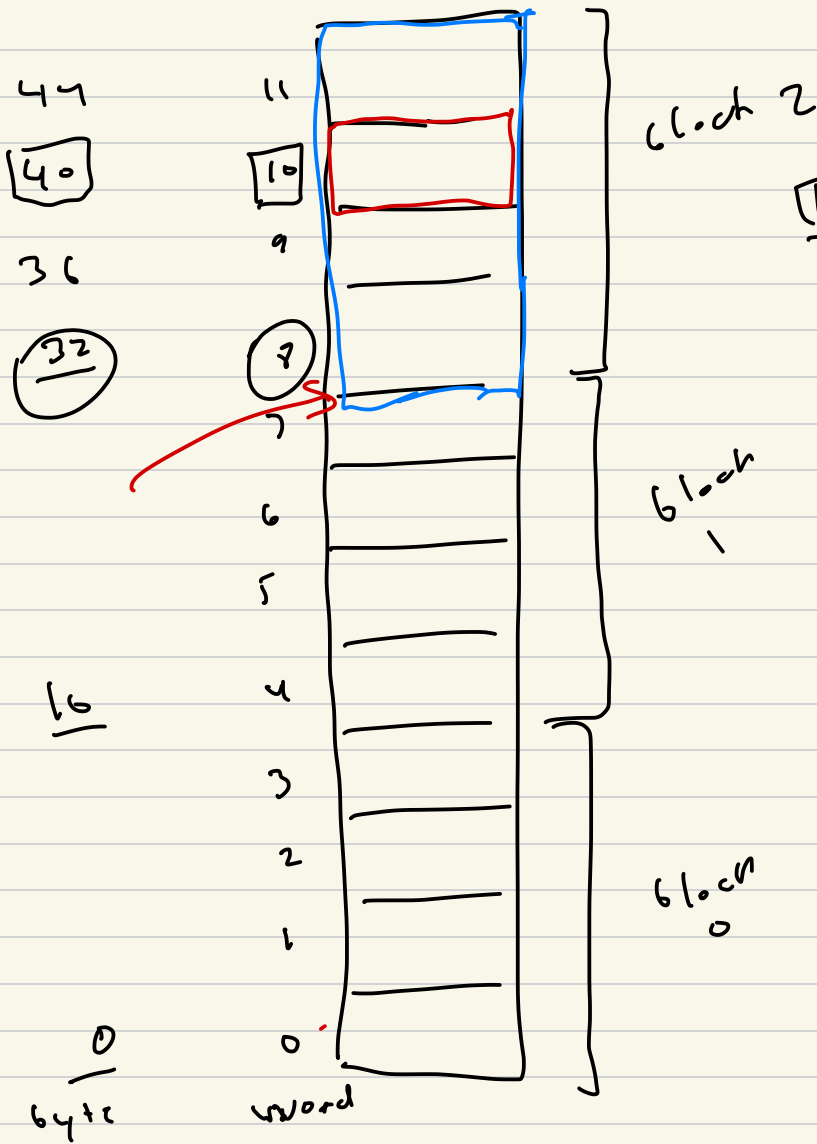
with block size > 1

1) on a hit

get word from slot using the block offset

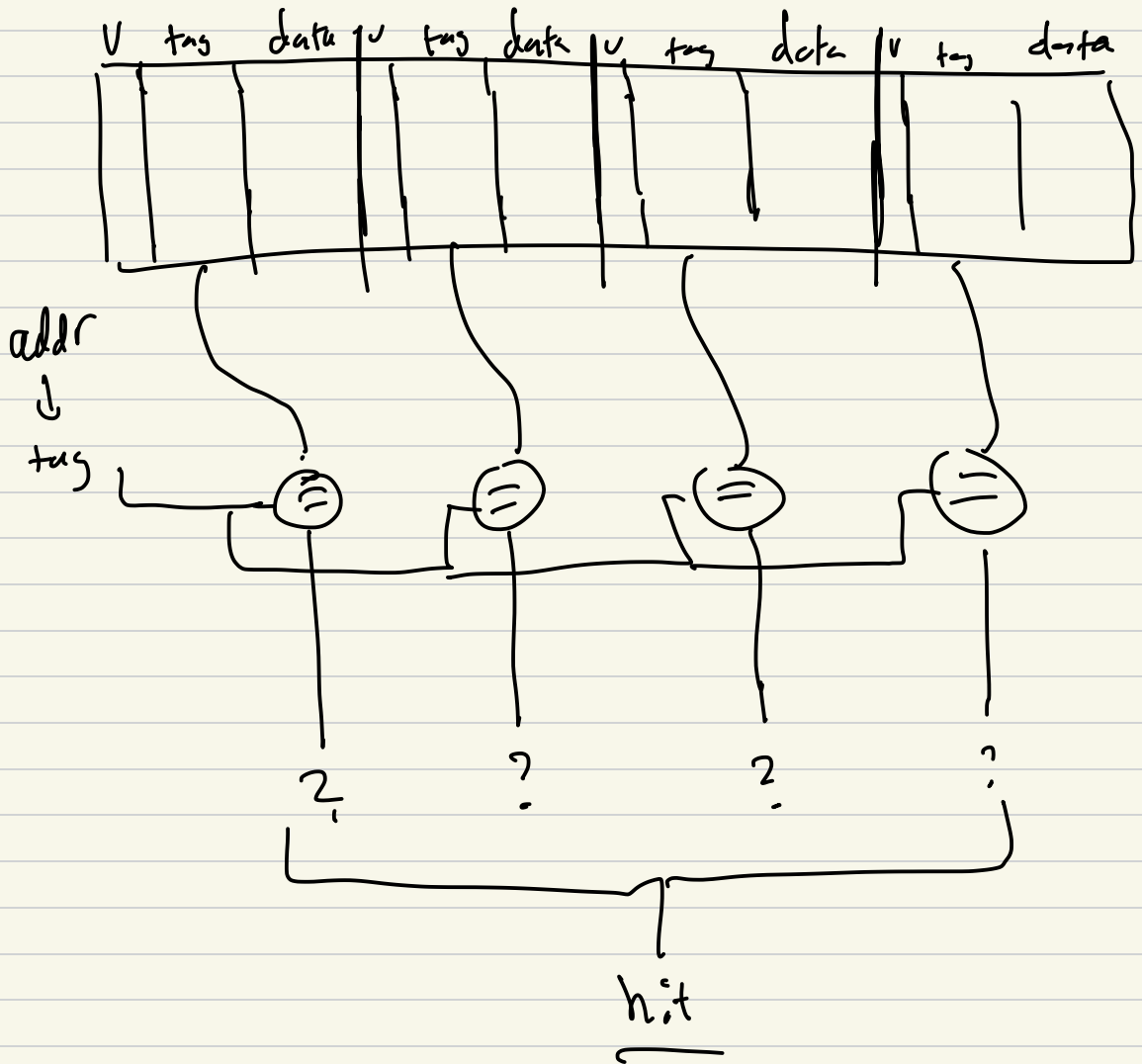
2) on a miss

find block-start-addr from addr
get N(4) words from memory

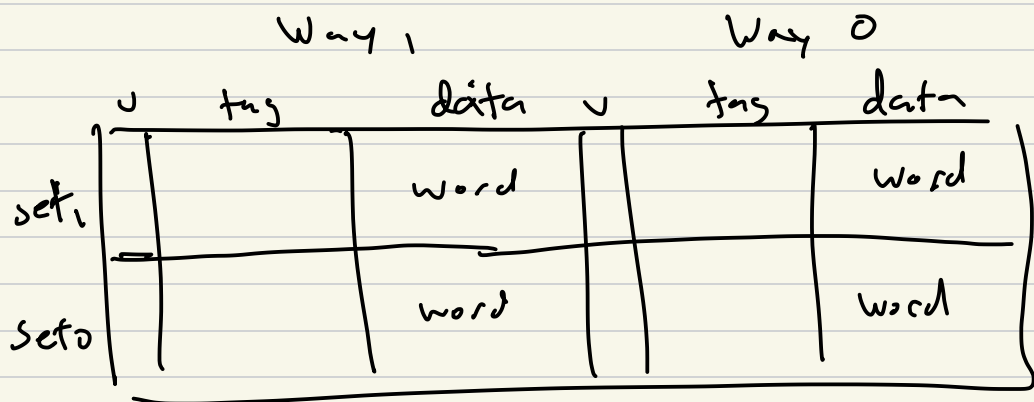


32	0	0	1	0	0	0	0	0	44	0	0	1	0	1	1	0	0
36	0	0	1	0	0	1	0	0	40	0	0	1	0	1	0	0	0
40	0	0	1	0	1	0	0	0	36	0	0	1	0	0	1	0	0
44	0	0	1	0	1	1	0	0	32	0	0	1	0	0	0	0	0

Fully Associative Cache



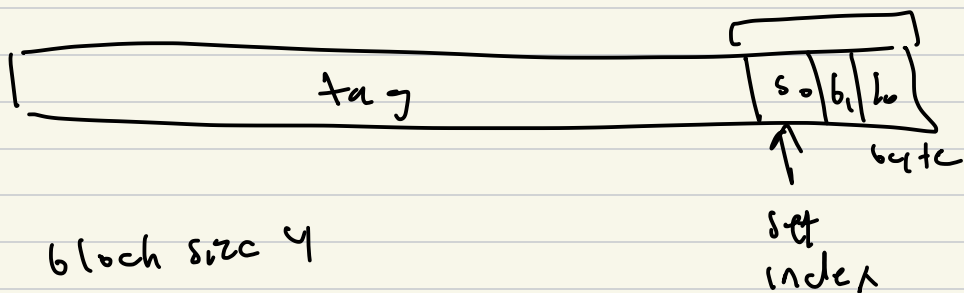
Set Associative Cache



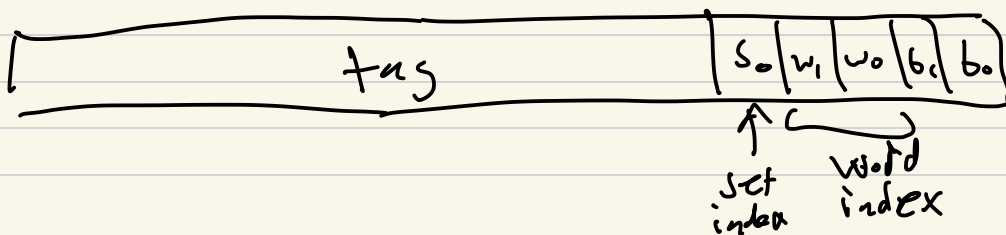
n-way set associative

2-way

addr



addr block size 4



Set Associative Pseudo Code

Cache slots

sa_lookup(addr)

```

num_refs += 1;
num_ways = 2;
tag = addr >> 3;
set_index = (addr >> 2) & 0b1;
set_base = set_index * 2;

```

```

for (i = 0; i < num_ways; i++) {
    slot = cache[set_base + i];
    if (slot.valid == 1 & slot.tag == tag) {
        // hit
        slot.timestamp = num_refs;
        return slot.data;
    }
}

```

// miss

```

slot = find_free_in_set(cache,
                        set_base)

```

```

slot.data = (uint32_t) addr;

```

```

slot.tag = tag;

```

```

slot.timestamp = num_refs;

```

```

return slot.data;

```

