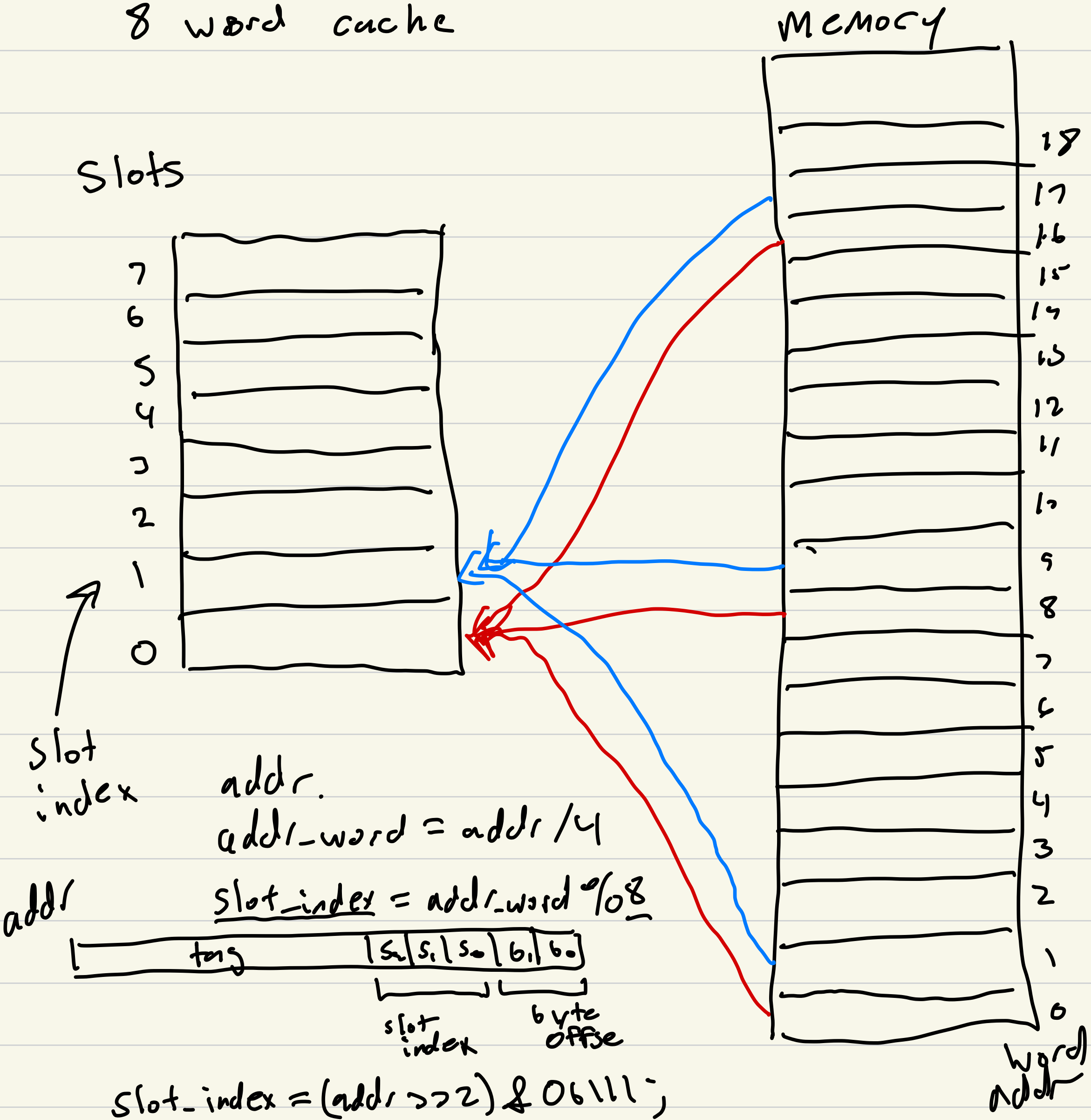
CS 315-02 Project04 Midterm

Cache Direct mapped

1 word per slot (block size is 1)

8 word cache



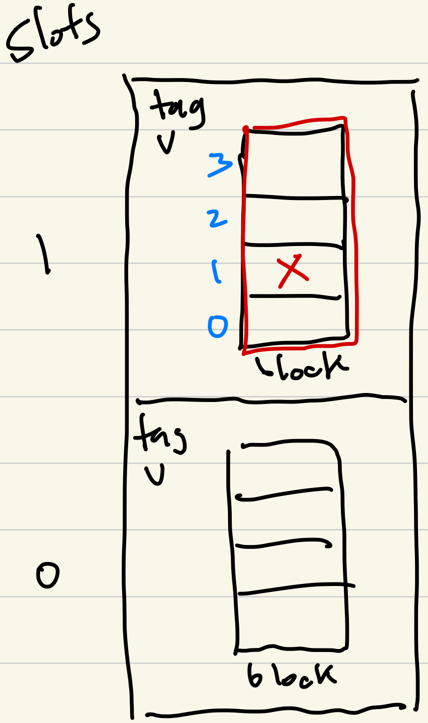
addr.

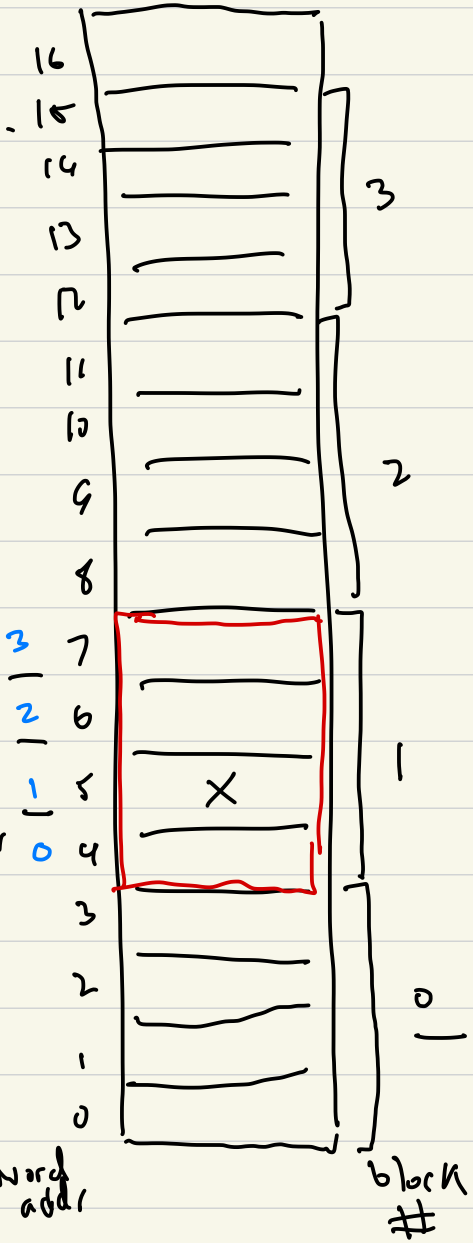
addr_word = addr / 4

slot_index = addr_word % 8

slot_index = (addr >> 2) & 0b111;

Direct Mapped Block Size of 4
8 words



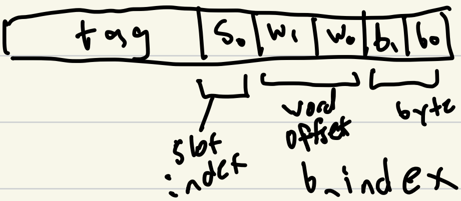


addr

addr_word = addr / 4 bytes per word

addr_block = addr_word / 4 words per block

slot_index = addr_block % 2 # of slots



```
slot_index = (addr >> 4) & 0b1
```

Hit

slot_index is slot to look at
Look at valid bit
and tags

```
b_index = add_word % 4
b_index = (addr >> 2) & 0b11;

data = slot.block[b_index];
```

Miss

Need to bring in entire block

Determine the block_base

```
block_base = addr_word - b_index;
block_base_byte = block_base * 4
loop i 0 to 4
slot.block[i] = *((uint32_t*) (block_base_byte + i*4))

slot.block[i] = *(((uint32_t*) block_base_byte) + i)

return slot.block[b_index]
```

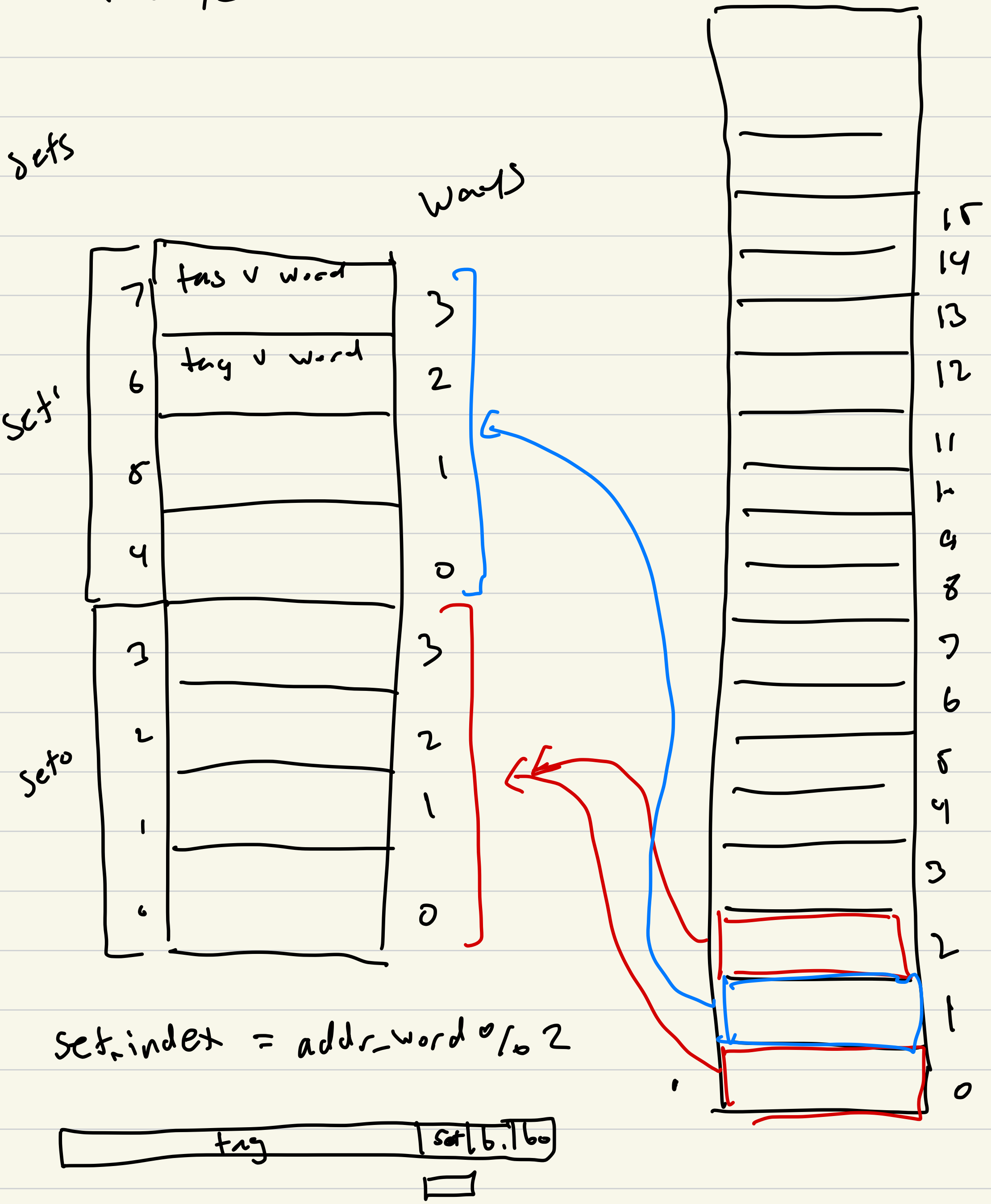
```
int arr[5)
int *p;

p = &arr[0];

p = p+1  = &arr[1];
```

Set associative Cache

4 ways 8 word

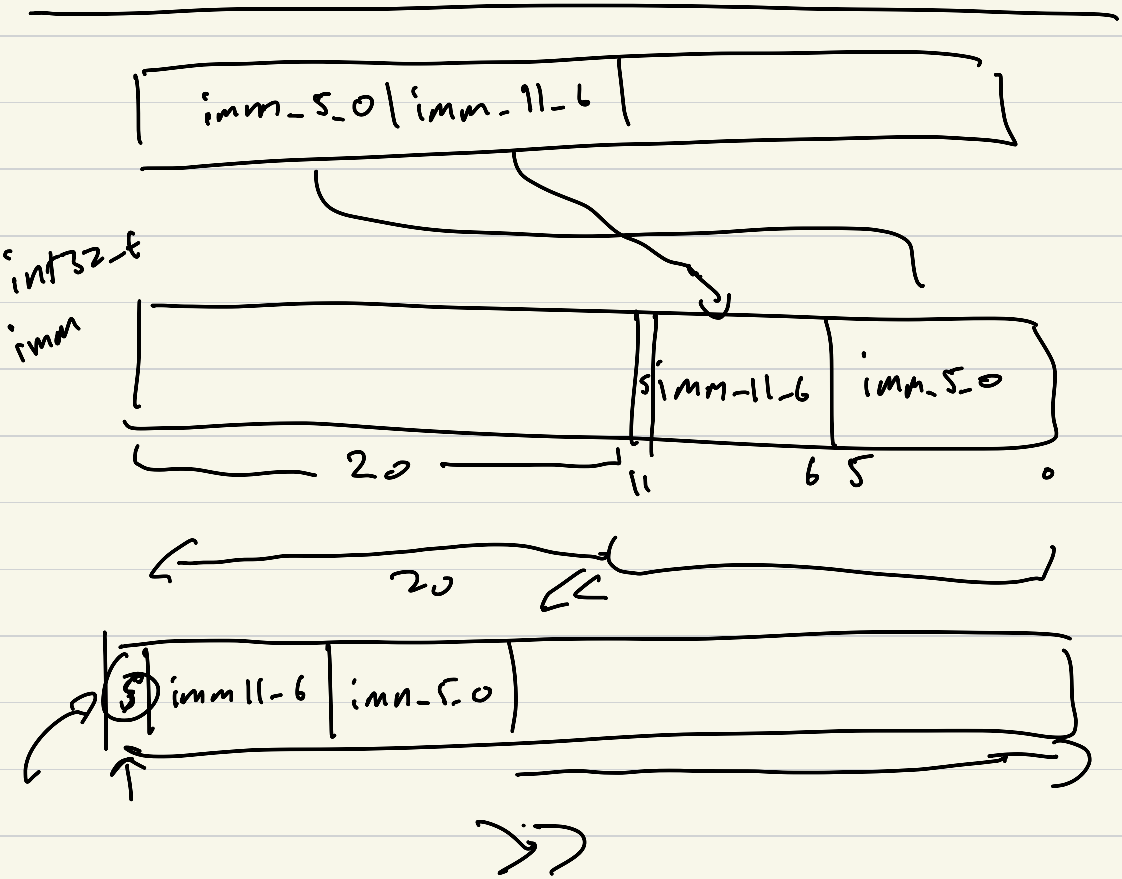


set_index = addr_word % 2

Lookup

set_index

set_base = set_index * 4



ASR

SSA