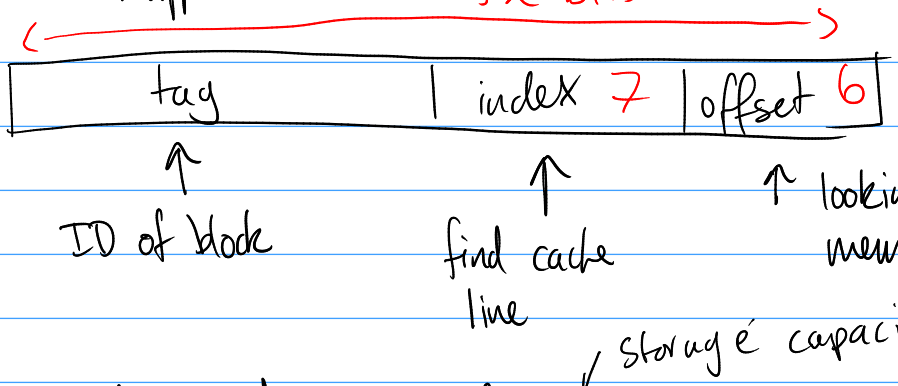
Direct Mapped Cache



Q: - Direct Mapped Cache of size (storage capacity) 8192 bytes
- Cache line (mem block size) has a size of 64 bytes

a) How many cache lines are in my cache?

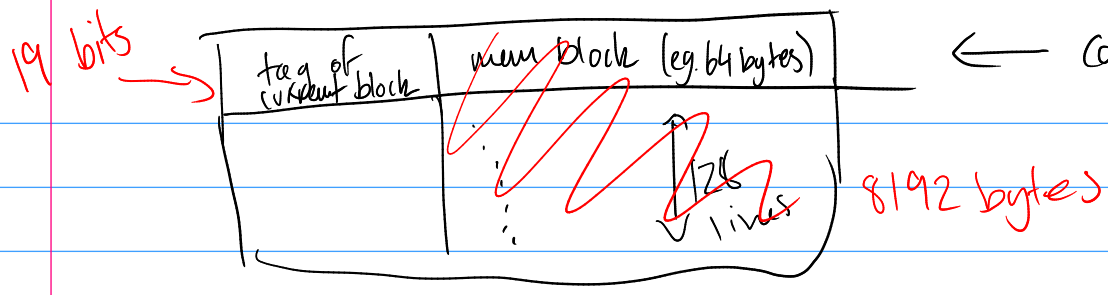
$$\#\text{ cache lines} = \frac{\text{size of cache}}{\text{line size}} = \frac{8192}{64} = 128 \text{ lines}$$

b) What are the size (in bits) of: Assuming 32 bit system
- the offset (block offset) field?
- the index field?
- the tag?

offset → 64 byte blocks, each byte has offset between 0 and 63, this can be shown in binary using 6 bits

index → we have 128 lines (see Qa), each line is therefore between 0 – 127, this can be shown in binary using 7 bits

Tag → remaining bits → 32 – offset(6) – index(7) = 19



c) How much memory (actual bytes on the silicone) to create this cache?

Actual size of cache = Storage capacity + meta data (storing the current tag for each cache line)

Storage capacity = 8192 bytes

We know the tag is 19 bits & we have 128 cache lines

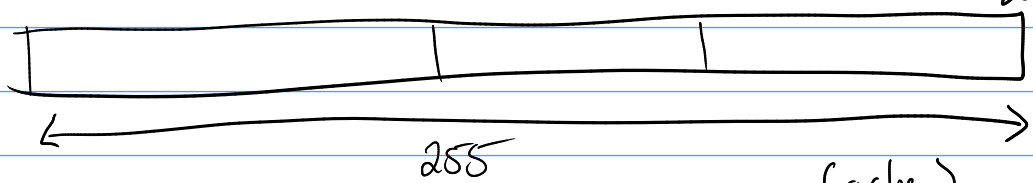
∴ we need 19 bits × 128 cache lines of meta data

= 2432 bits = 304 bytes

Actual cach size = 8192 + 304 bytes

= 8496 bytes

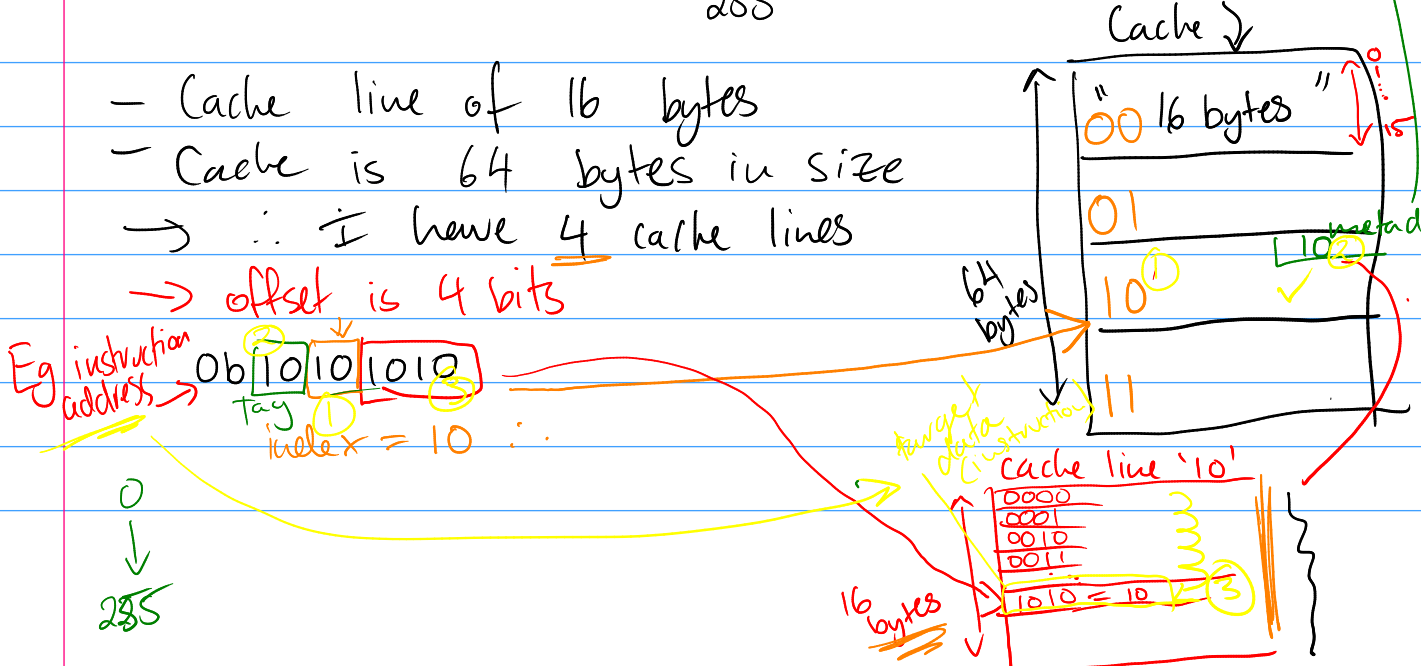
Eg. 8 bit microcontroller, memory addr between 0 → (2^8-1) 255



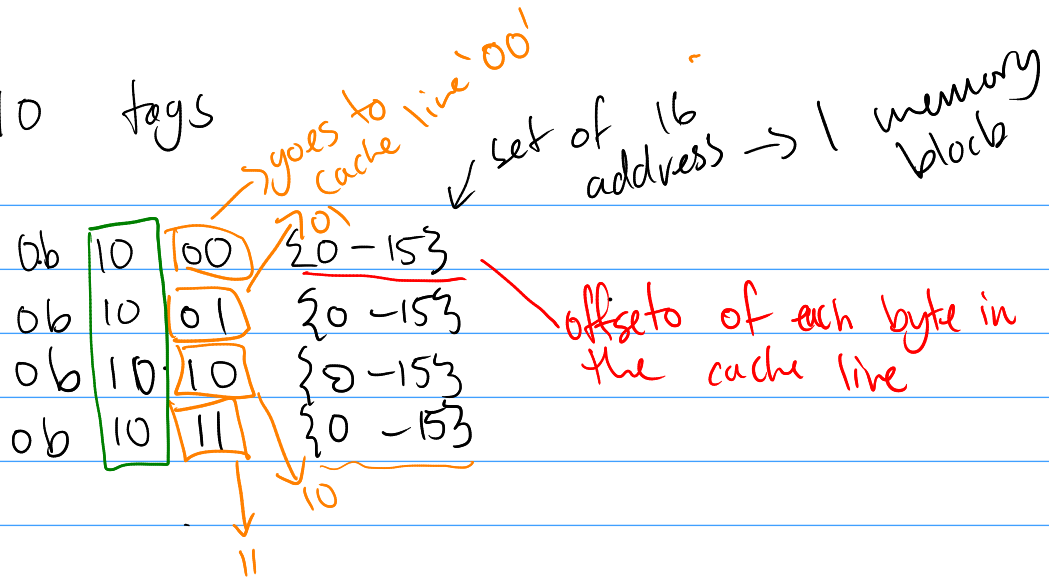
- Cache line of 16 bytes
- Cache is 64 bytes in size

→ ∴ I have 4 cache lines

→ offset is 4 bits



10 tags

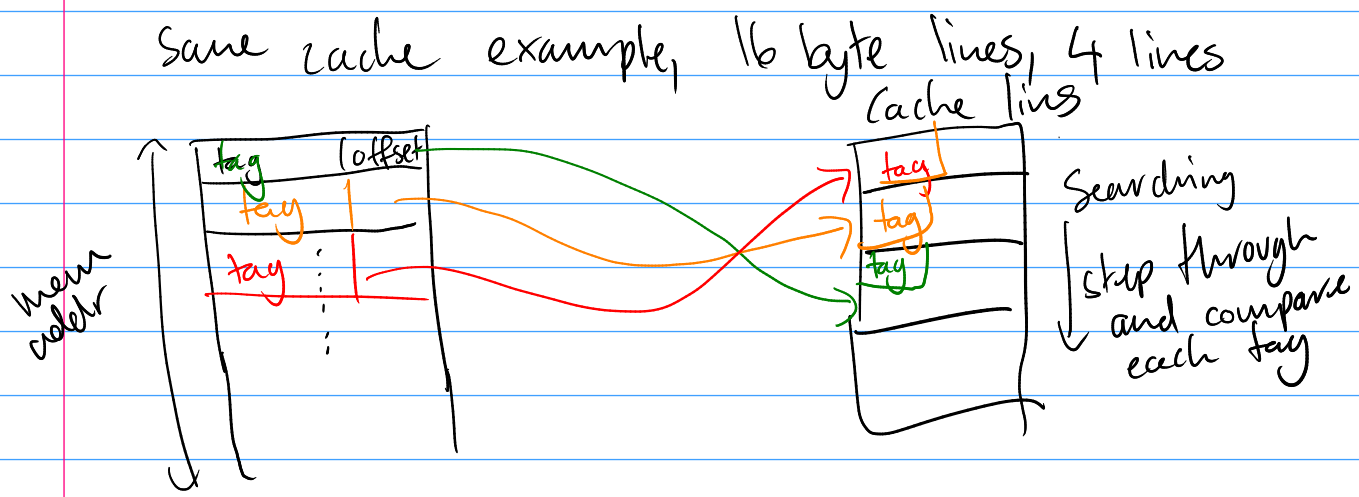


for eg. cache line '00' will only ever have 1 block with tag 10, ie. address 0b1000 {0-15}

Fully Associative cache

We dont have an index

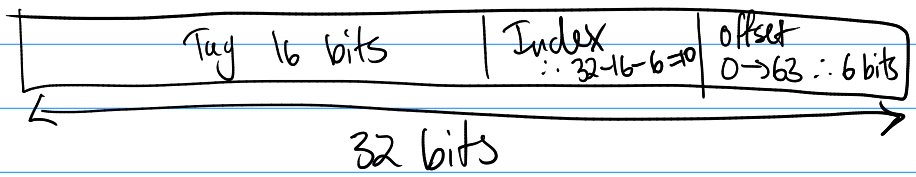
Same cache example, 16 byte lines, 4 lines



13. Assume you have a direct mapped cache with a cache line size of 64 bytes, a tag of 16bits and 32bit addresses.
   a) How many 32bit instructions can be stored in the cache?
   b) How many cache lines are there in total?
   c) What is the total amount of memory required for implementing such a cache (not counting the valid bit)?

/4P

Address:



Index = 10 bits = 2^{10} possibilities
= 1024 cache lines

a) We need to know cache size because answer is cache size / 32 bit instructions (4 bytes)

Cache size = 1024 lines × 64 byte cache line size
= 65536 bytes

This means we can store $\frac{65536}{4}$ 32 bit instructions (4 bytes)

∴ We can store 16,384 instructions in our cache!

b)

c) Tag is 16 bits ∴ we need 16 bits of meta-data for each cache line

∴ Total meta-data = 16 × 1024 = 16384 bits

= 2048 Bytes of meta data

∴ Total memory = cache capacity + memory needed for meta-data
= 65536 + 2048
= 67584 bytes

4.1 Assume a direct mapped instruction cache:
If 10 Bits are used for the index and the cache capacity is 16Kb (1kb = 1024 Bytes).
How much additional memory is needed in the cache to store the tags? 5 points

32 bit system

↳ how large is our meta data in bytes?

b = bits
B = bytes

Tag	Index 10 bits	offset 4 bits

block/line size = Size of cache / number of lines → 16×1024 = 16384 B

↓

cache line size = 16384 / 1024 = 16B

we have 10 bits telling us our index

∴ we have 2^{10} cache lines
= 1024 cache lines

∴ offset is 4 bits

32 bit address ∴ Tag bits = 32 − 10(index) − 4(offset)
= 18 bits for tag

∴ Each cache line requires 18 bits of meta-data
∴ The cache requires a total of 1024 × 18 bits = 18432 b
= 2304 B

4.2 Now consider a different cache that can store 65536 Bytes of user data and has a index size of 8 bits.
a) How many cache lines are there in total?
b) How big is a single cache line?
c) How big is the Tag 6 points

Tag ?	index 8bits	offset (k) 8bits

a) # cache lines = $2^{\text{index bits}}$
= 2^8 = 256 cache lines

b) Size of cache line = cache size / no # of lines = 65536 / 256 = 256B

c) no # of offset bits (k) → cache line size = 2^k
$256 = 2^k$ ∴ k = 8 → offset is 8 bits

∴ Tag is 16 bits