R1 (0, 4)
R2 (2, 7)

pos: 111 010 100 000

R2.y R2.x R1.y R1.x

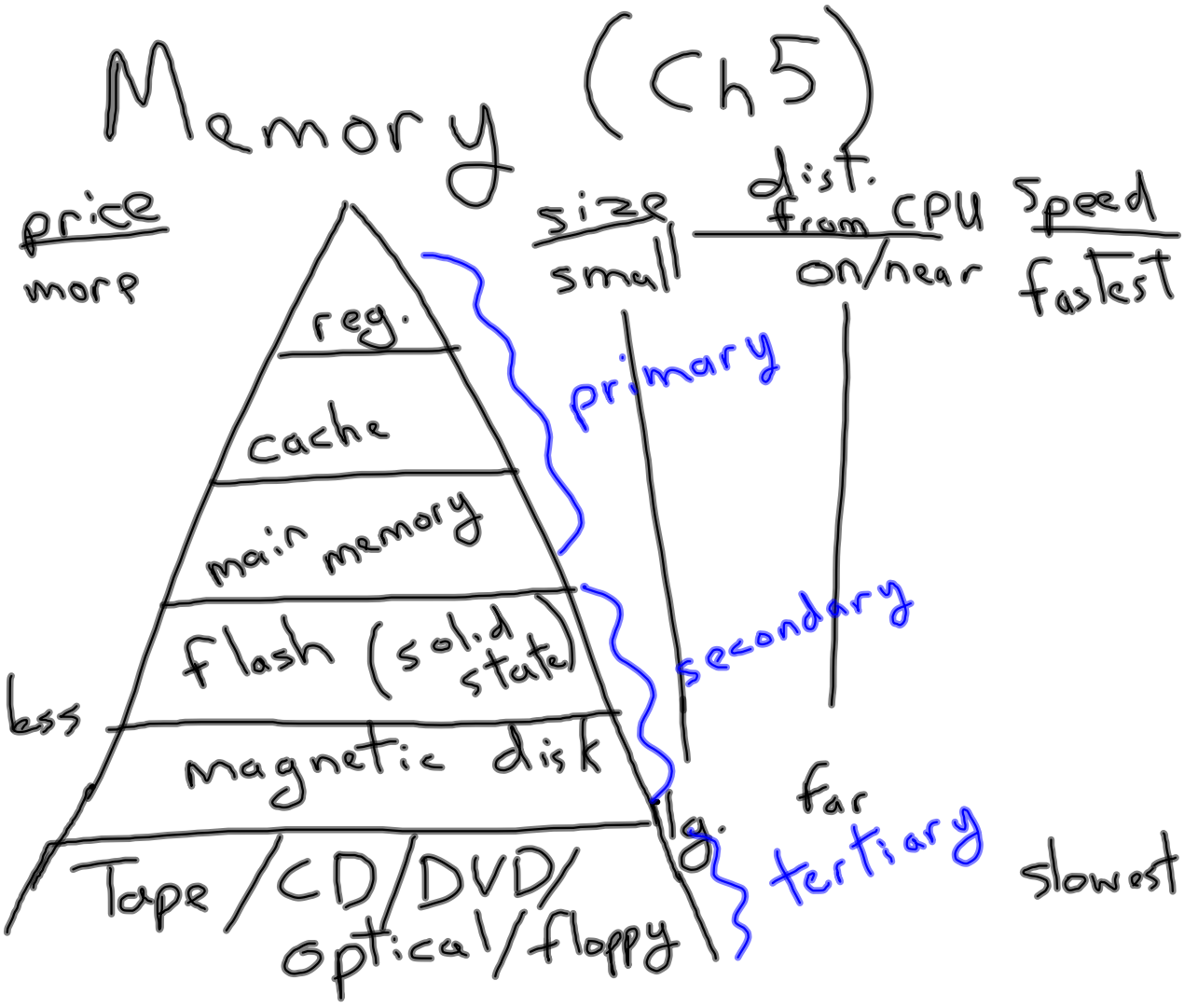
000 000 000 111 07

x1 = pos & 07

y1 = (pos & 070) >> 3

000 000 111 000

070
0x38
56


Memory (Ch5)
price
more
size
small
dist. from CPU
on/near
speed
fastest
reg.
cache
main memory
flash (sol.d state)
magnetic disk
Tape / CD / DVD / optical / floppy
less
lg.
far
primary
secondary
tertiary
slowest

Memory Access

Random

Direct

Sequential

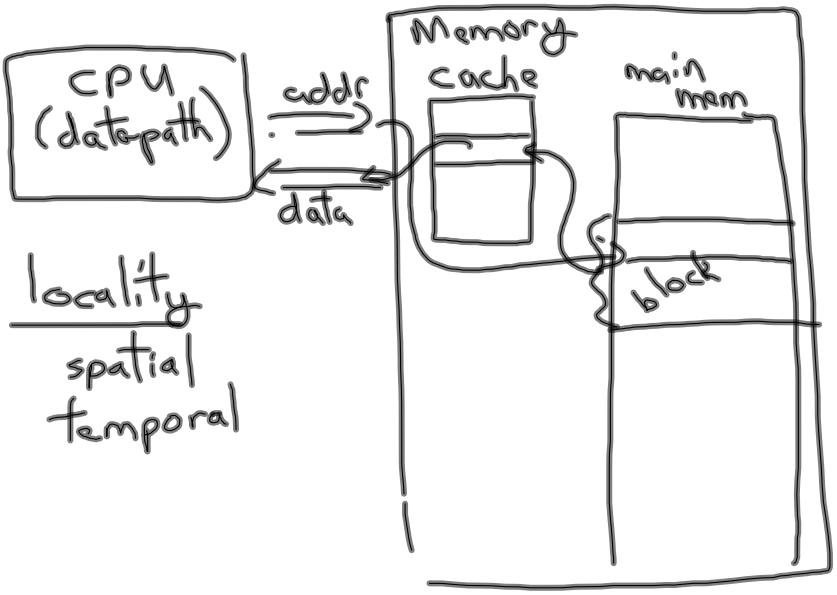
RAM

Static RAM - flip flops
(SRAM)

Dynamic RAM
(DRAM)
- refresh

ROM

chip	rewrite	write
ROM	no	no
PROM	no	once
EPROM	yes	yes
EEPROM	yes (byte)	yes
FLASH	yes (block)	yes



locality
- spatial
- temporal

cache hit
↳ found the addr in the cache

cache miss
-> get block from main memory (put in cache)
-> access cache

---

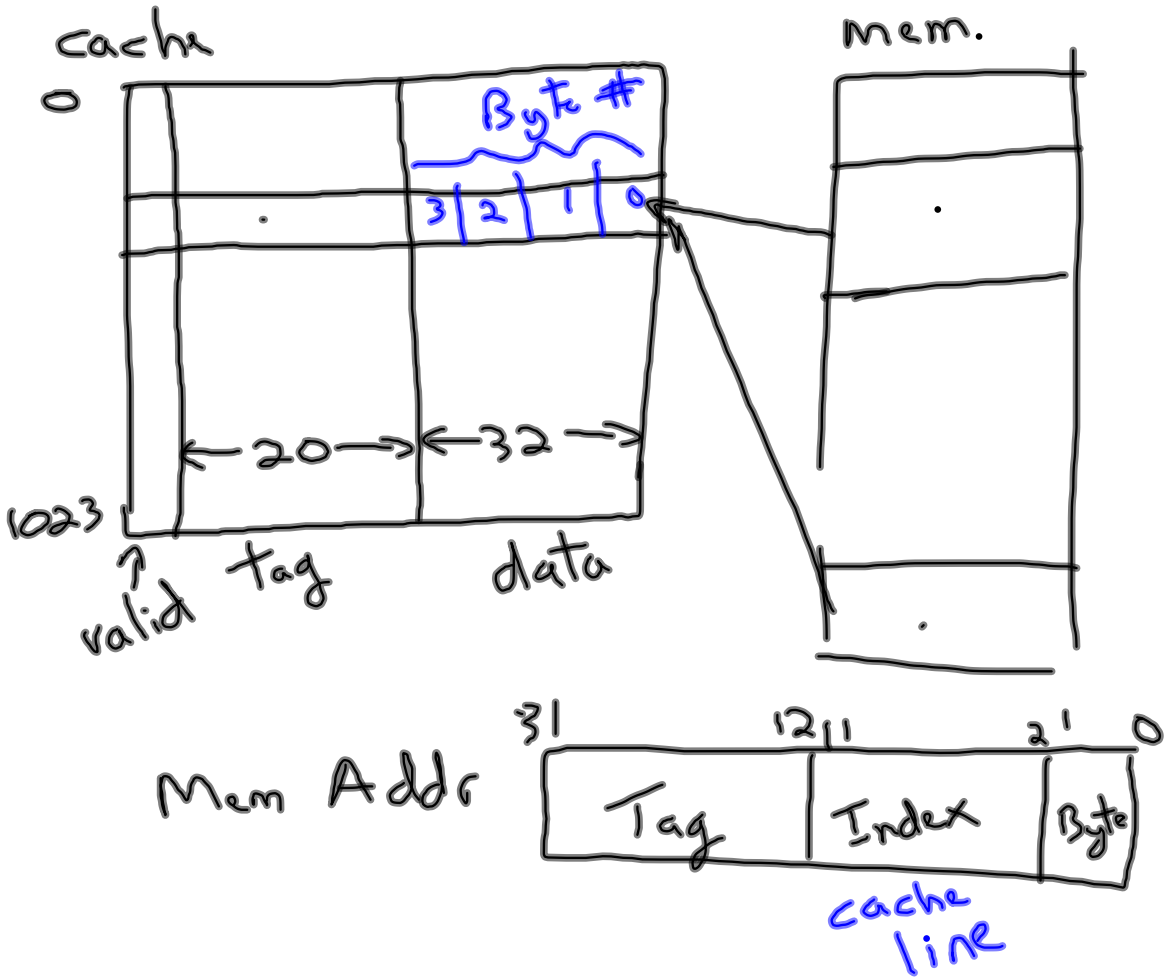
write policies

write through
- write value to main mem. when changed in cache

write back
- when block is going to be replaced

Direct Mapped Cache

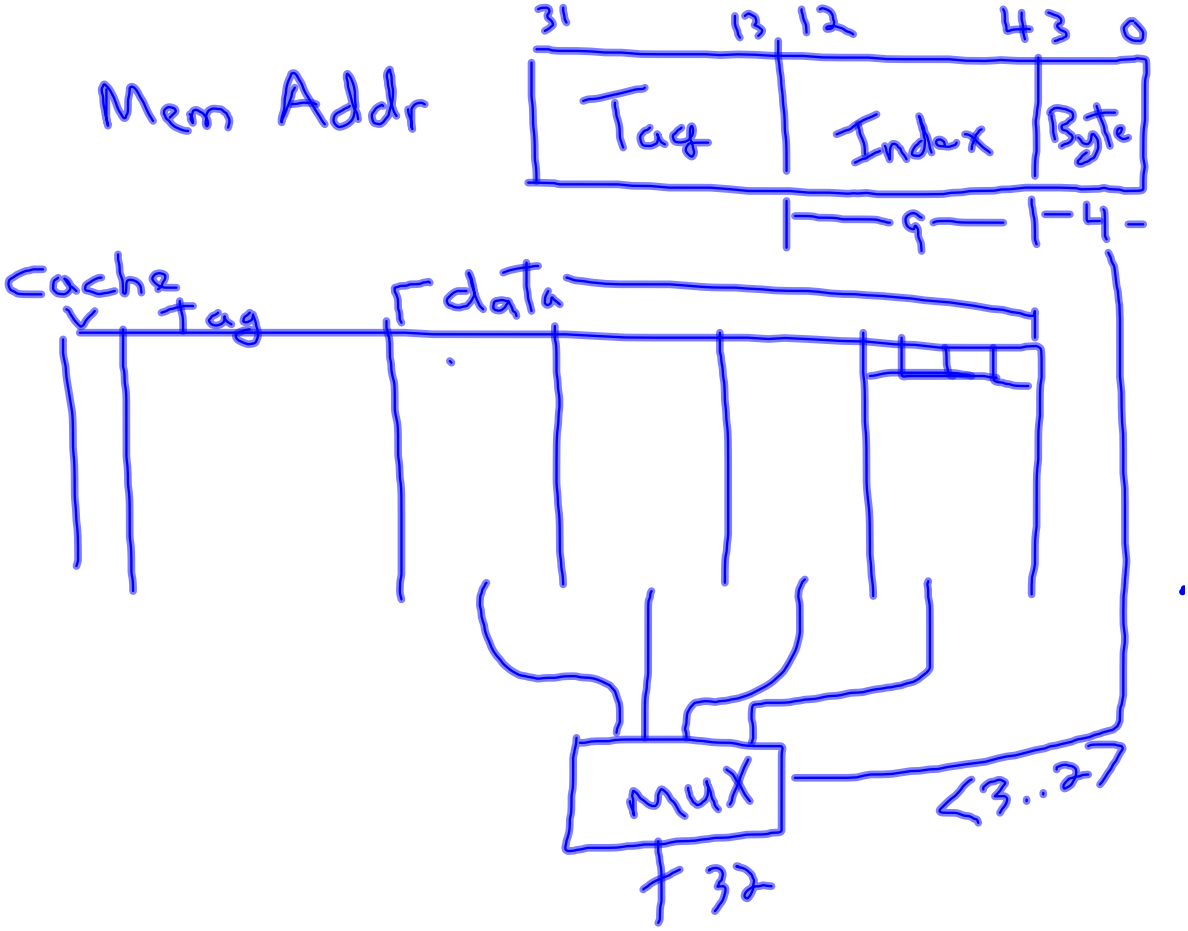
cache index = mem. addr % cache lines



Memory Size: 2^{32}

Cache: 512 lines

Block size: 4 words (16 Bytes)



Set-Associative