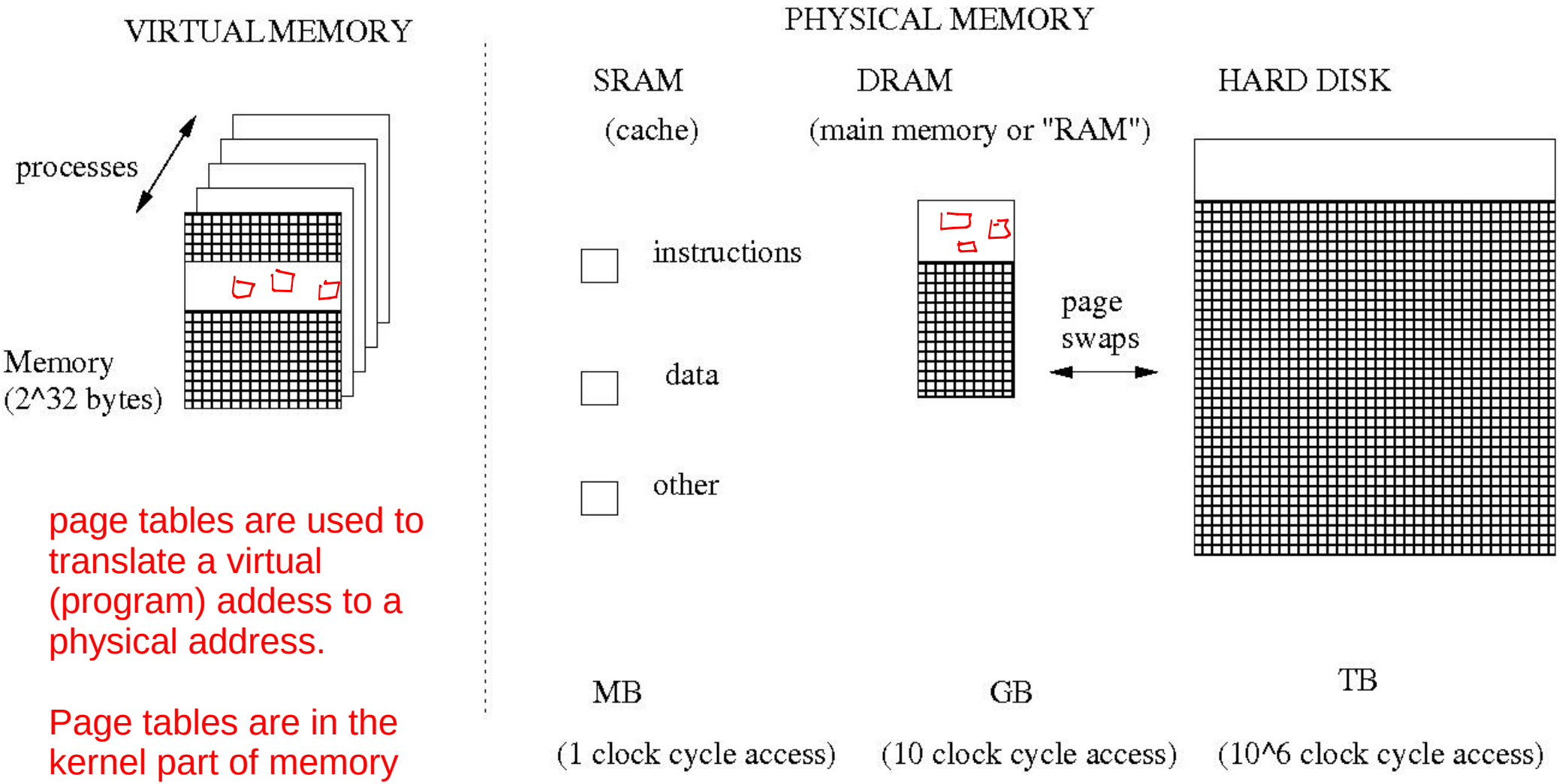
lecture 17

cache 1

- page table cache (TLB)

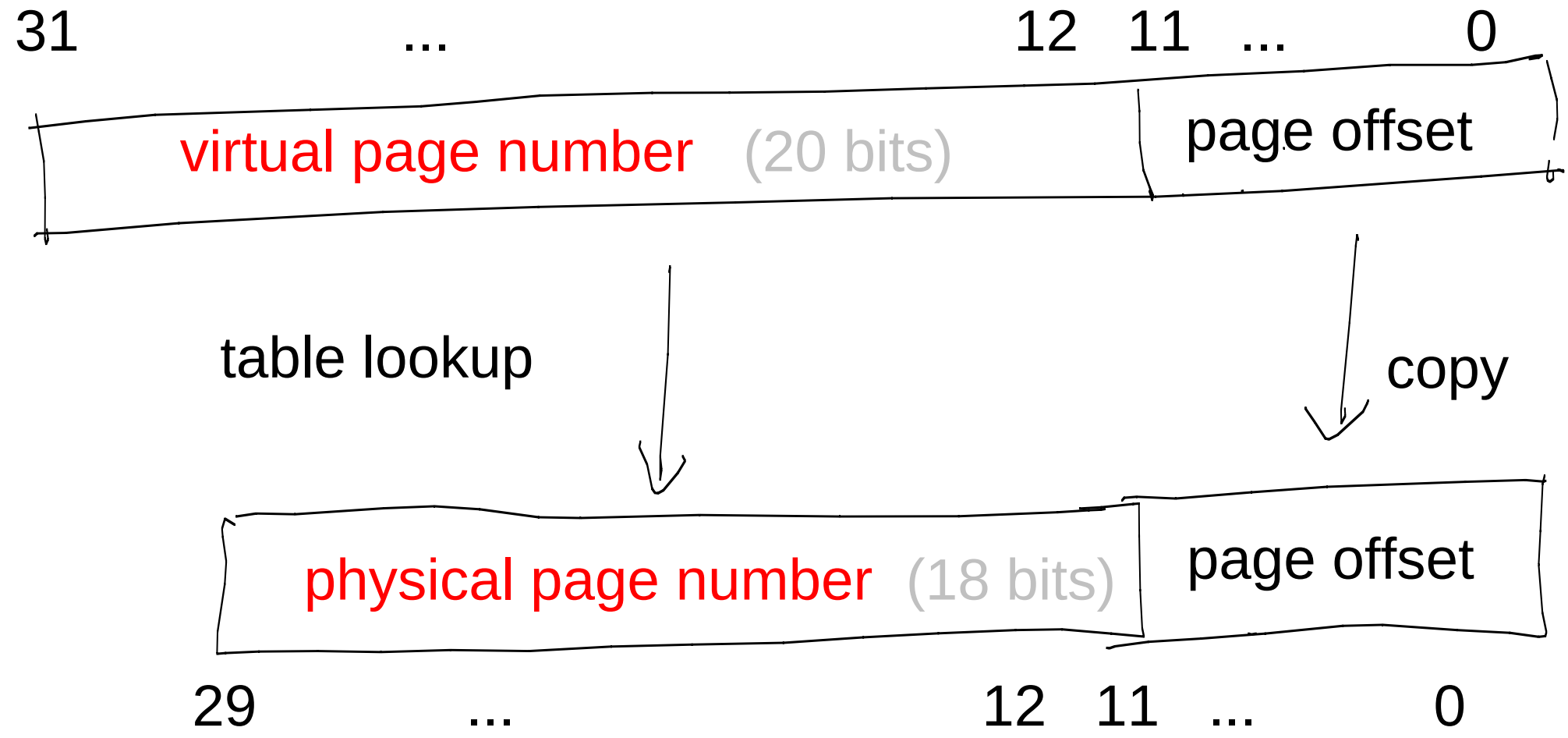
Mon. March 14, 2016

some key ideas from last lecture



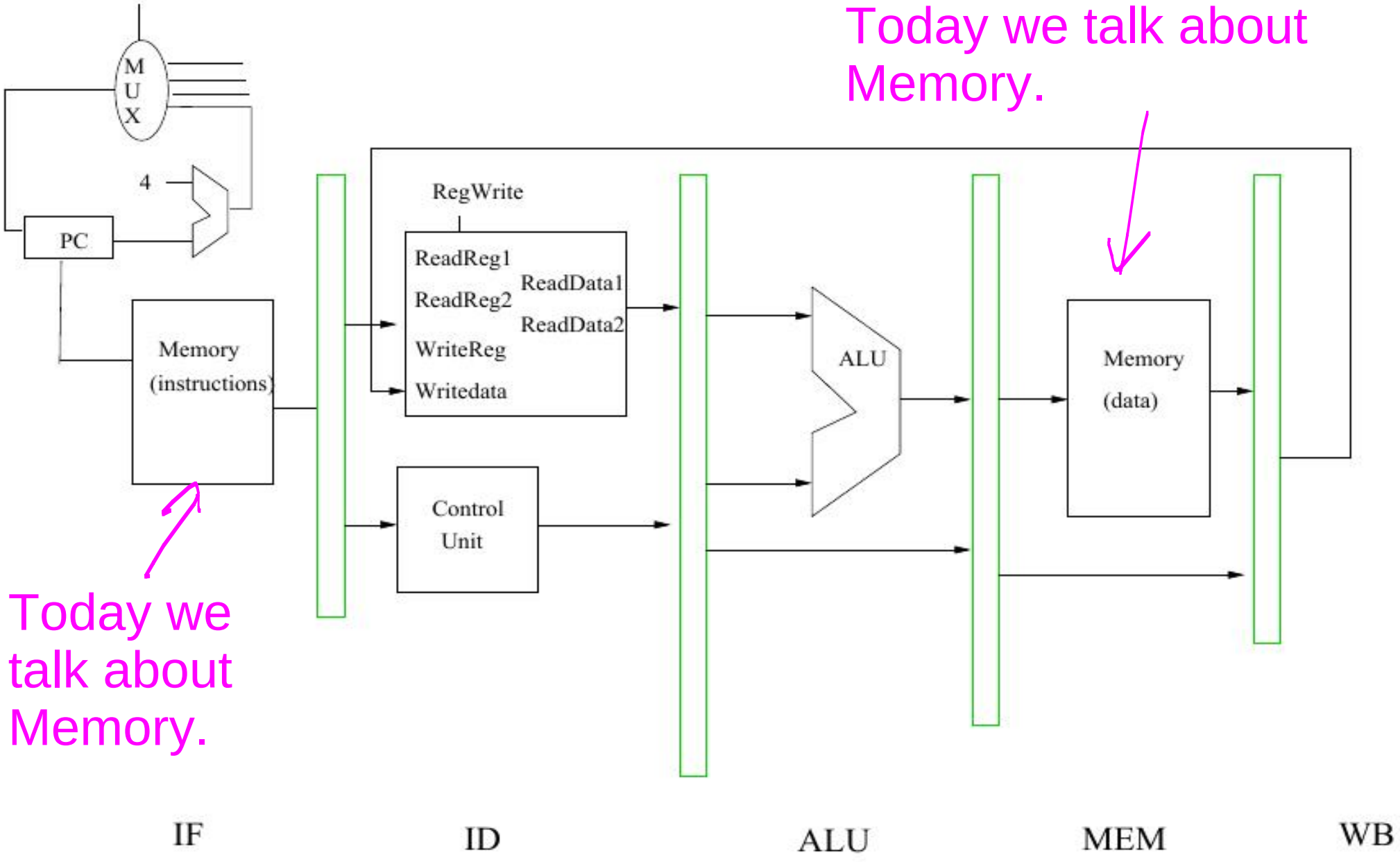
page tables are used to translate a virtual (program) addess to a physical address.

Page tables are in the kernel part of memory

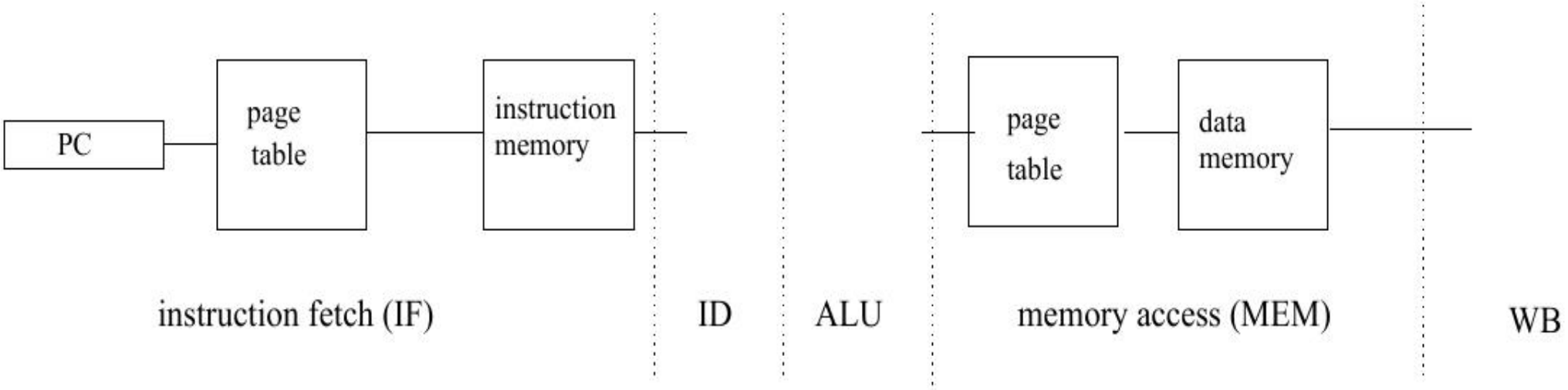


physical address (RAM) e.g. 1 GB = 2^30 bytes

recall MIPS CPU pipeline

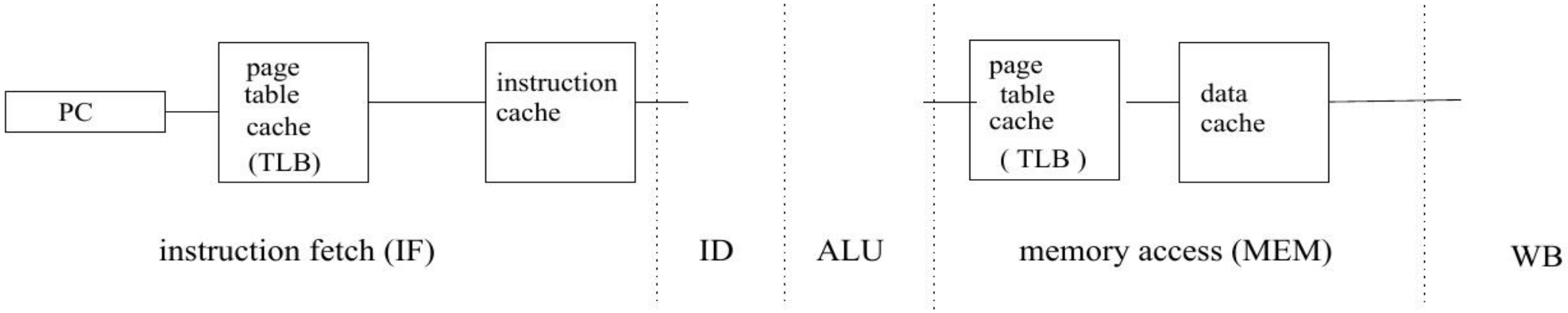


IF and MEM stages (conceptual only)



It is too slow to go always to main memory (RAM) or to disk. Instead, we want to use a fast memory that can be accessed in one clock cycle (SRAM).

implementation with caches (SRAM)



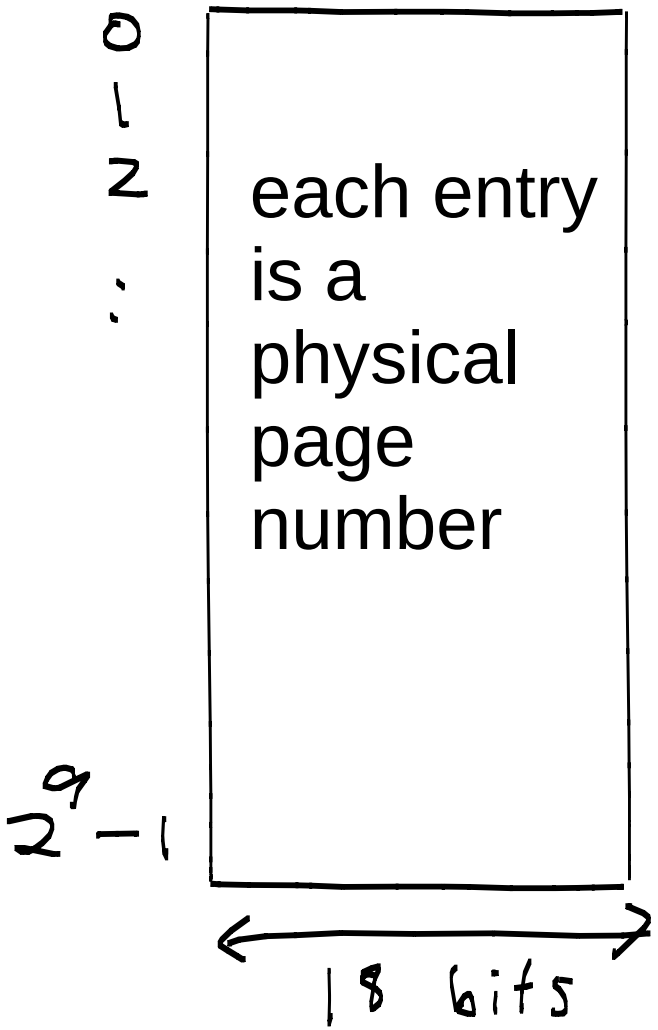
The page table cache is called the "translation lookaside buffer" (TLB). We can think of two TLB's, for data and instructions.

We can assume physical page numbers in TLB are in main memory only (i.e. not on disk, otherwise page swap is needed).

Instruction and data caches are separate.

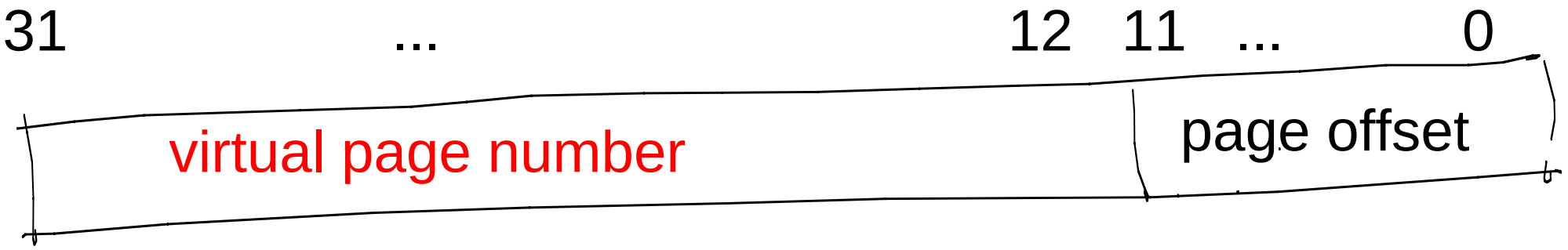
Page table cache
(called the "Translation Lookaside Buffer" or "TLB")

e.g. Suppose the TLB has 2^9 = 512 entries.

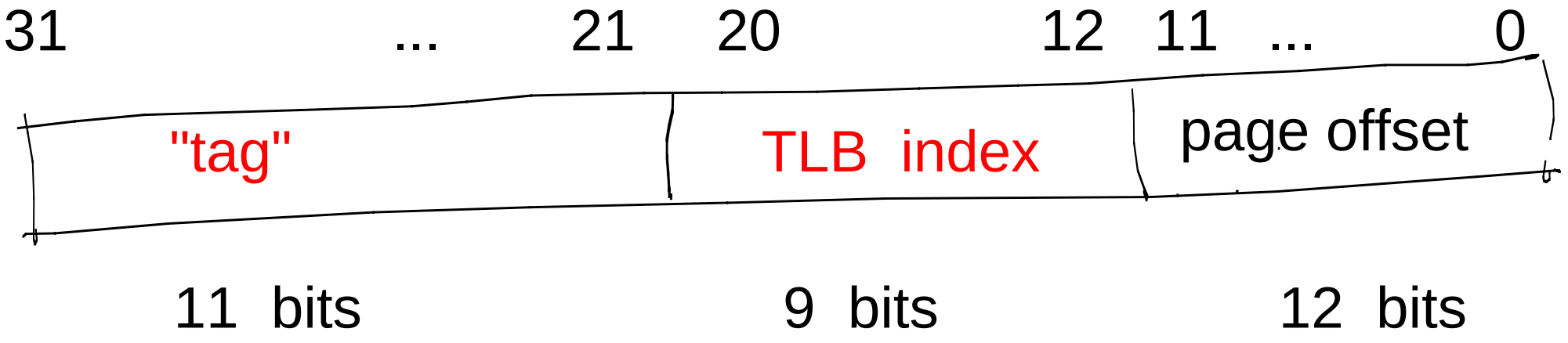


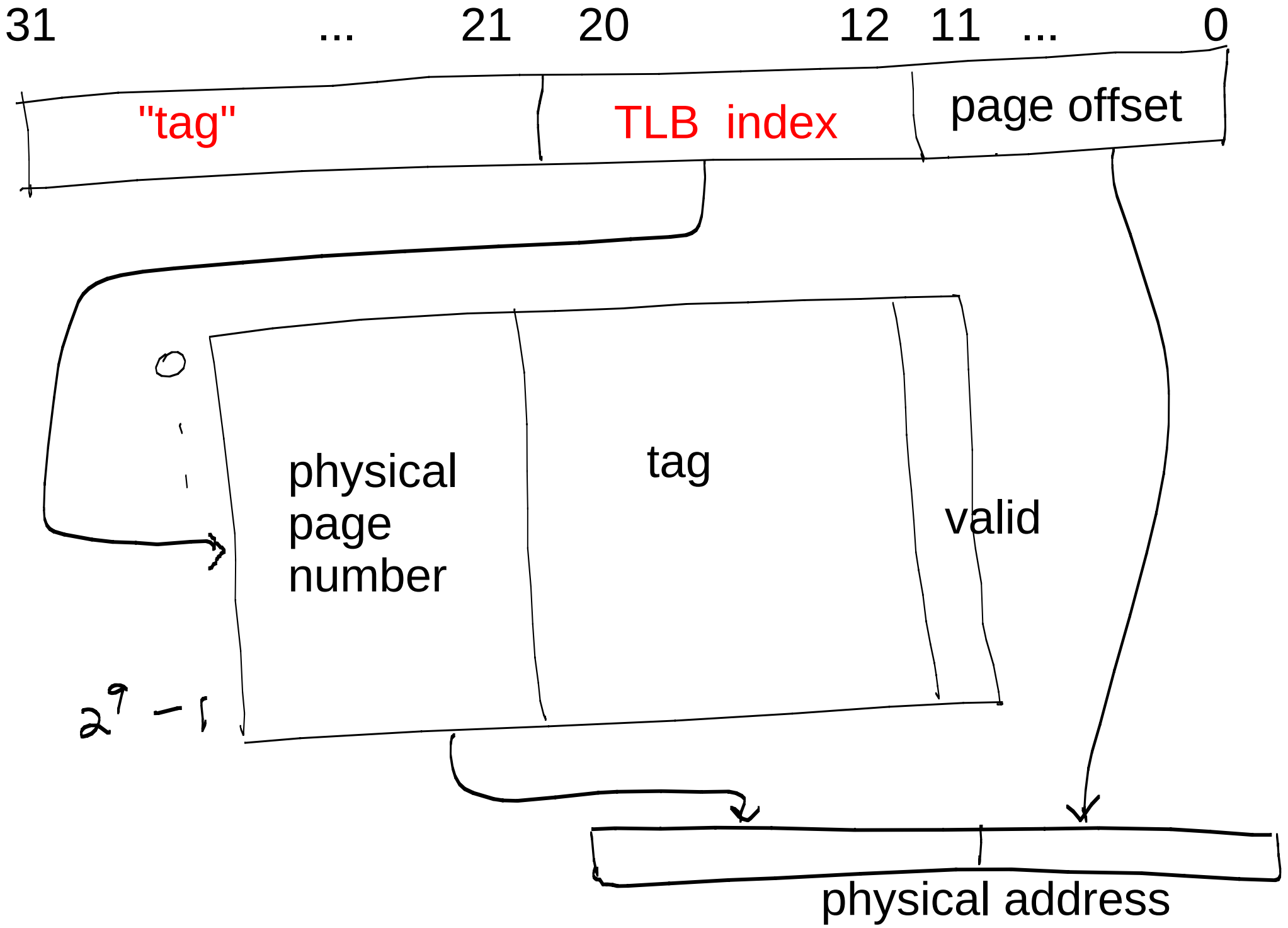
Only a subset of entries from the page table fits into the TLB (2^9 out of 2^20)

How to index and 'recognize' these entries ?



partition VPN into two parts




31
...
21
20
12
11
...
0
"tag"
TLB index
page offset
0
physical page number
tag
valid
$2^9 - 1$
physical address

Example (3 virtual addresses)

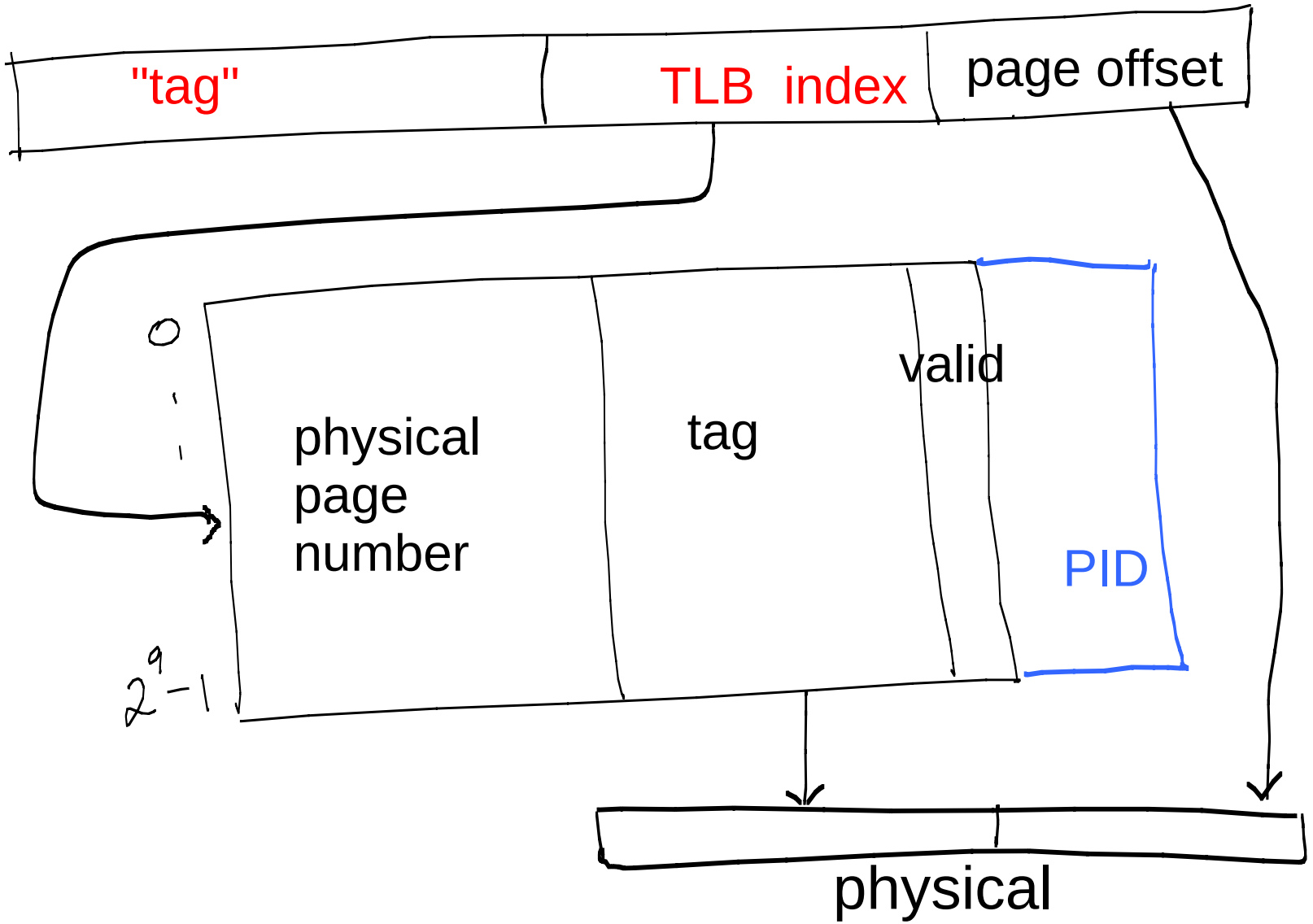
tag (11)	TLB index (9)	page offset (12)
01010100100	001001011	010101111111
01010100100	001001011	001001001001
01000111011	001001011	001001001001

All three map to the same TLB entry.

The first and second are on the same page but correspond to different physical addresses (because the offsets differ)

The second and third have the same page offsets, but are on different pages.

[ASIDE: Different processes can share the TLB, so one typically adds a process identification (PID) field and verify that the translation is indeed for that current process'es page table.]



TLB Hit versus Miss

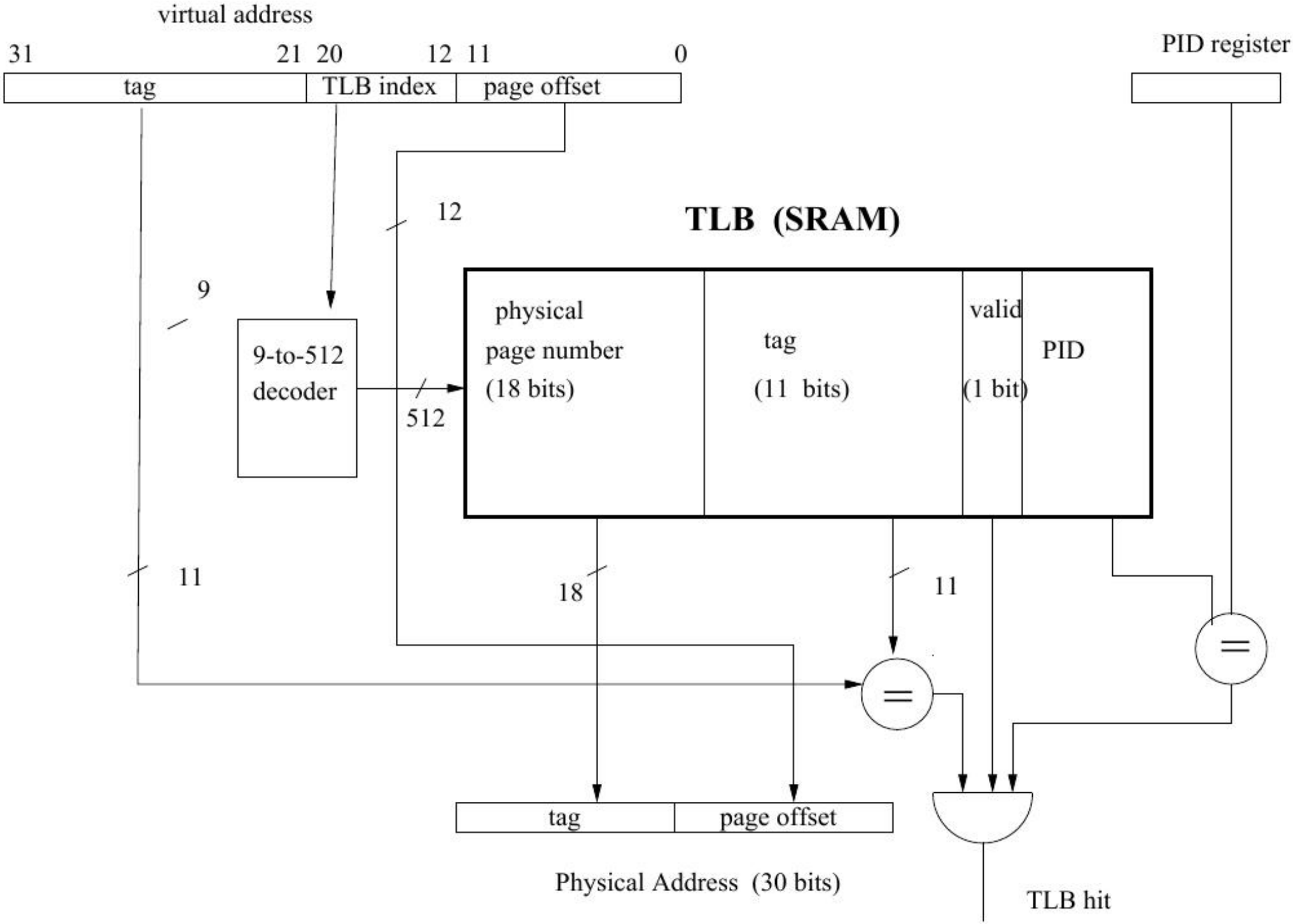
Only a subset of entries from the page table fits into the TLB.

"Hit" = the translation we want is in the TLB

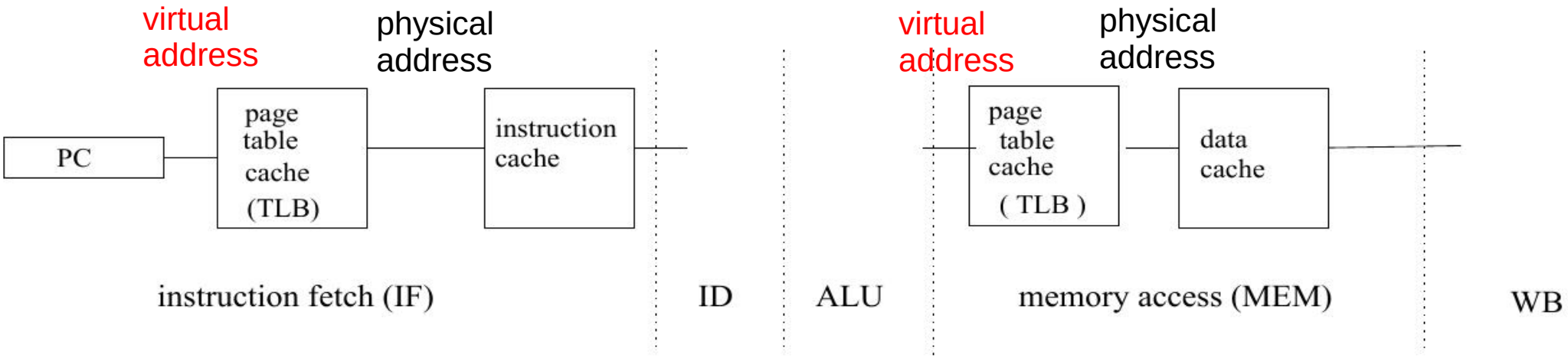
"Miss" = the translation we want is NOT in the TLB
(and need to be brought into the TLB from the page table)

What does the TLB circuit look like? How can we decide if we have a hit or a miss ?

We will discuss TLB misses next lecture (analogous to page fault from last lecture).


virtual address
31
21
20
12
11
0
tag
TLB index
page offset
PID register
TLB (SRAM)
12
9
9-to-512
decoder
512
physical
page number
(18 bits)
tag
(11 bits)
valid
(1 bit)
PID
11
18
11
=
=
=
tag
page offset
Physical Address (30 bits)
TLB hit

Instruction and data caches



We can assume entries in all these caches correspond to pages that are in main memory only

(i.e. otherwise page swap would be needed... more to say about that in future lecture)..