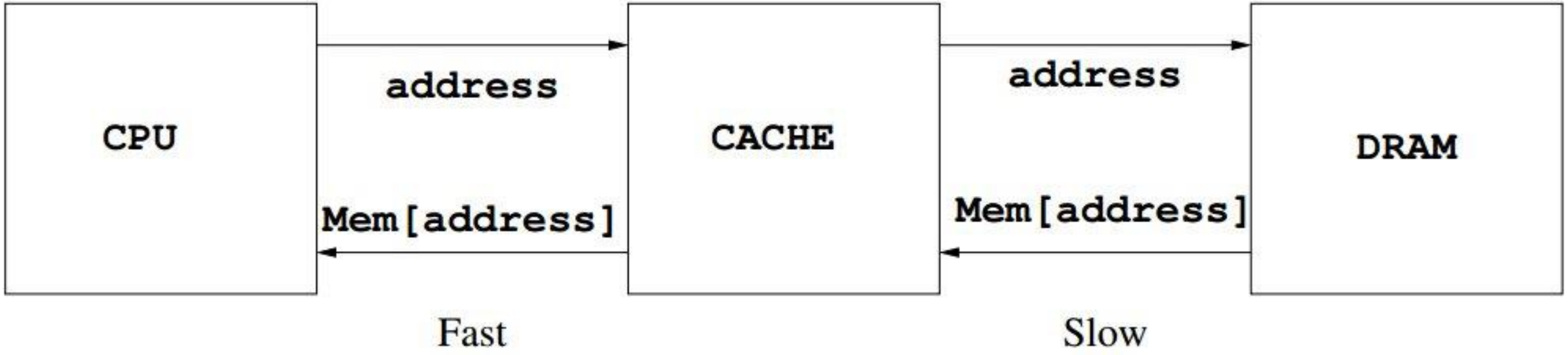
Tutorial 9 : cache memory

Why use a cache ?

- **Main memory** (VRAM/DRAM) **is slow** !
- To deal with this, the β-machine speed is reduced to match the memory read and write speed
- To make the machine faster, one can use a intermediate smaller and faster memory between the processor and the main memory: **a cache**.
- The cache associates memory addresses with their values (taken from the main memory)



Basic working principle

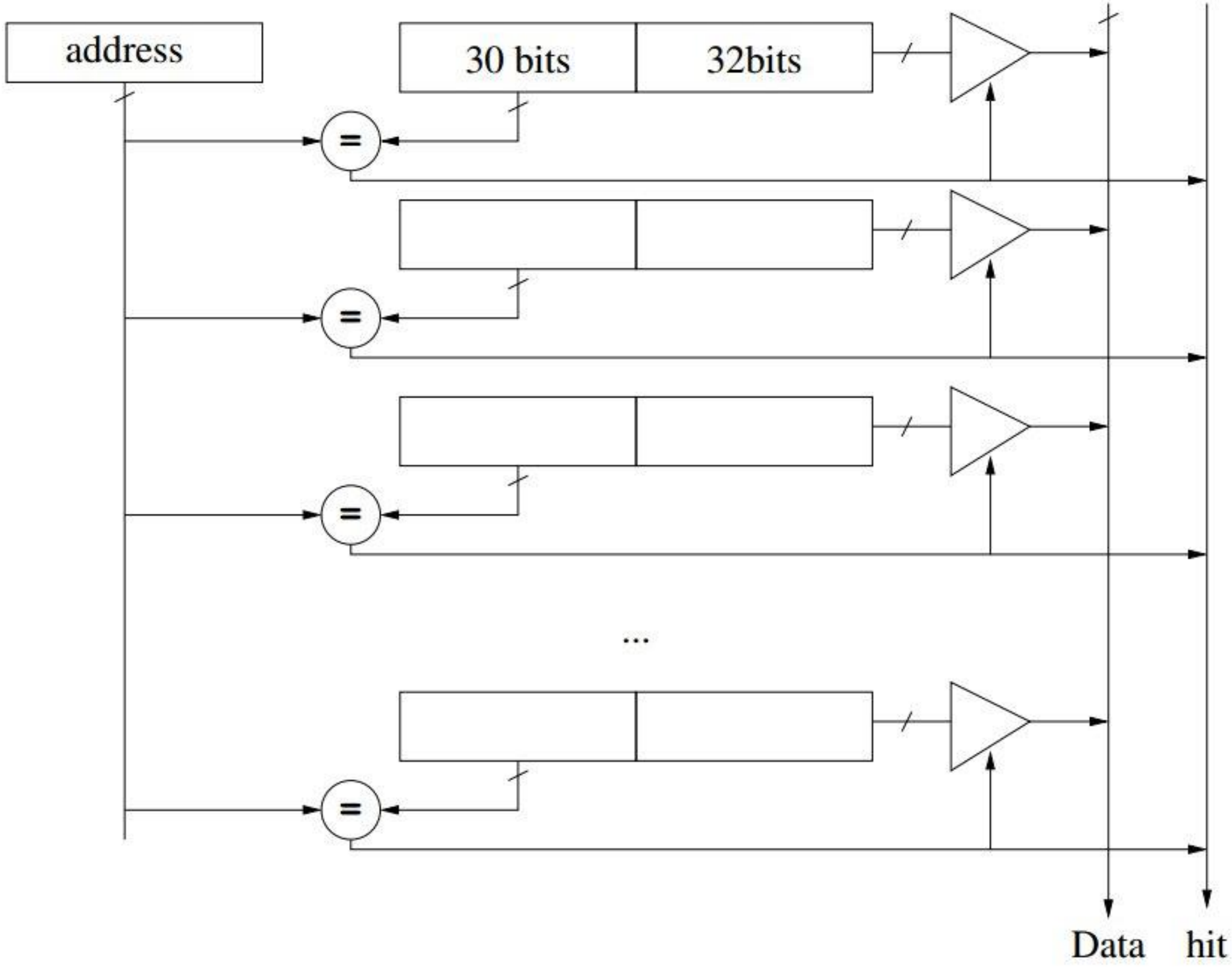
- Reading a value from memory in presence of a cache is simple:
    1. Check whether the cache memory contains the address
    2. If it does, read the associated value from the cache
    3. Otherwise, save the value in the cache and return it

- This usually works because memory accesses are not random. They follow the subsequent principles:
    - **Temporal locality principle**
    - **Spatial locality principle**

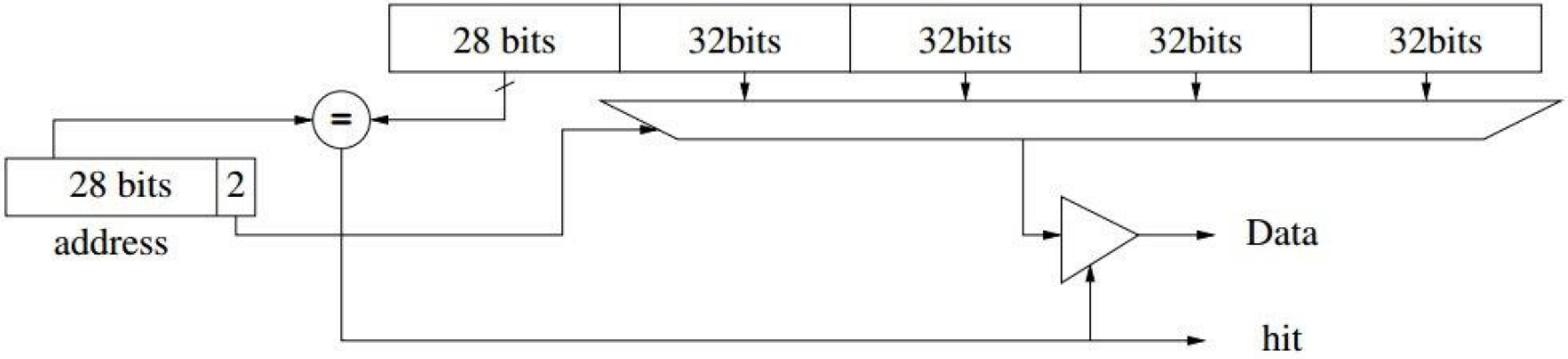
Cache memory variants

- Totally associative cache
- Totally associative cache in blocks
- Direct mapped cache
- Set associative cache

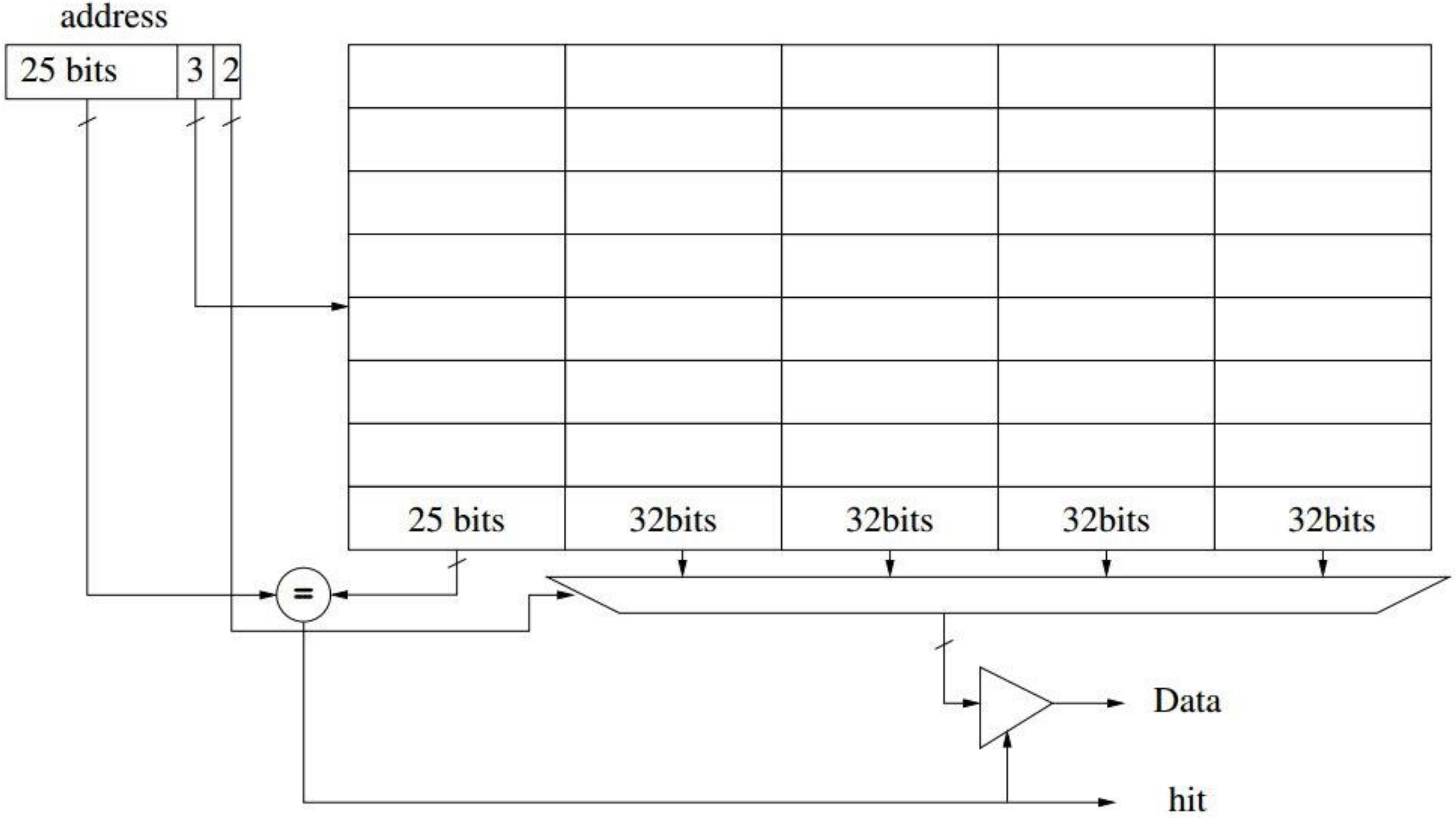
Totally associative cache



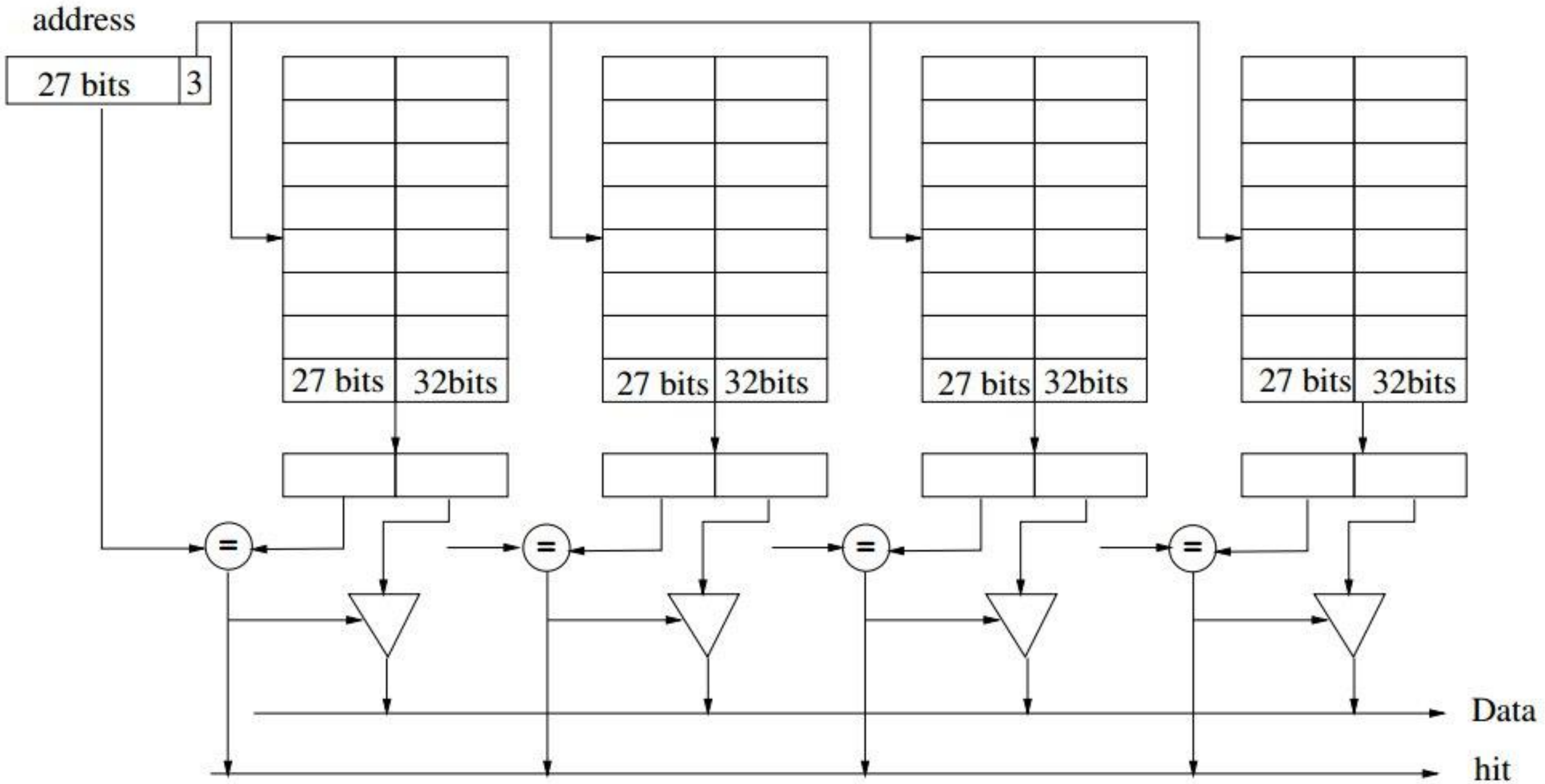
Totally associative cache in blocks



Direct mapped cache (in blocks)



Set associative cache


address
27 bits
3
27 bits
32bits
27 bits
32bits
27 bits
32bits
27 bits
32bits
=
=
=
=
Data
hit