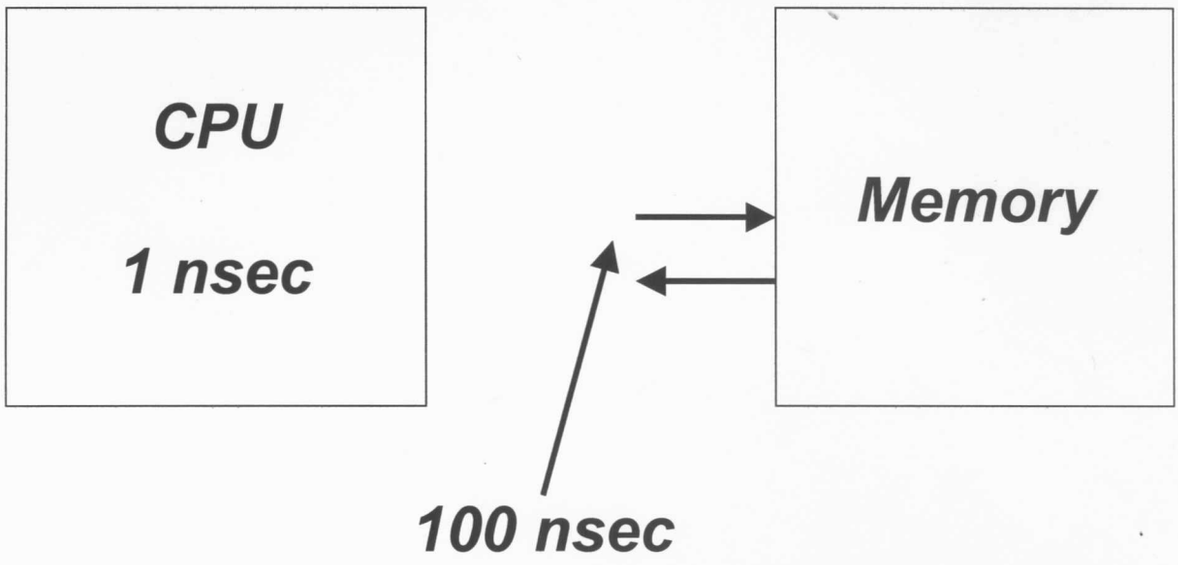
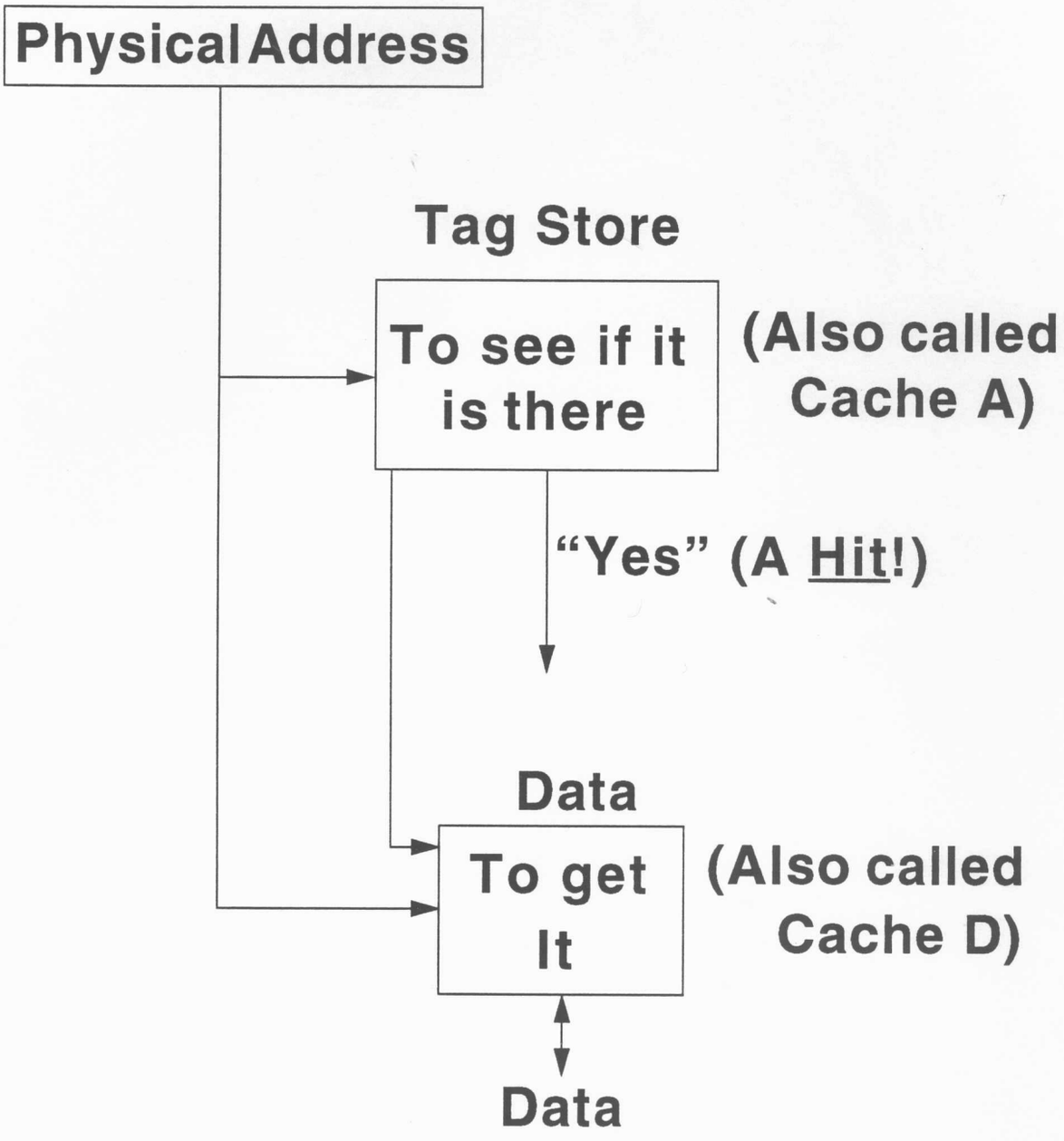
Cache Memory

WHY?

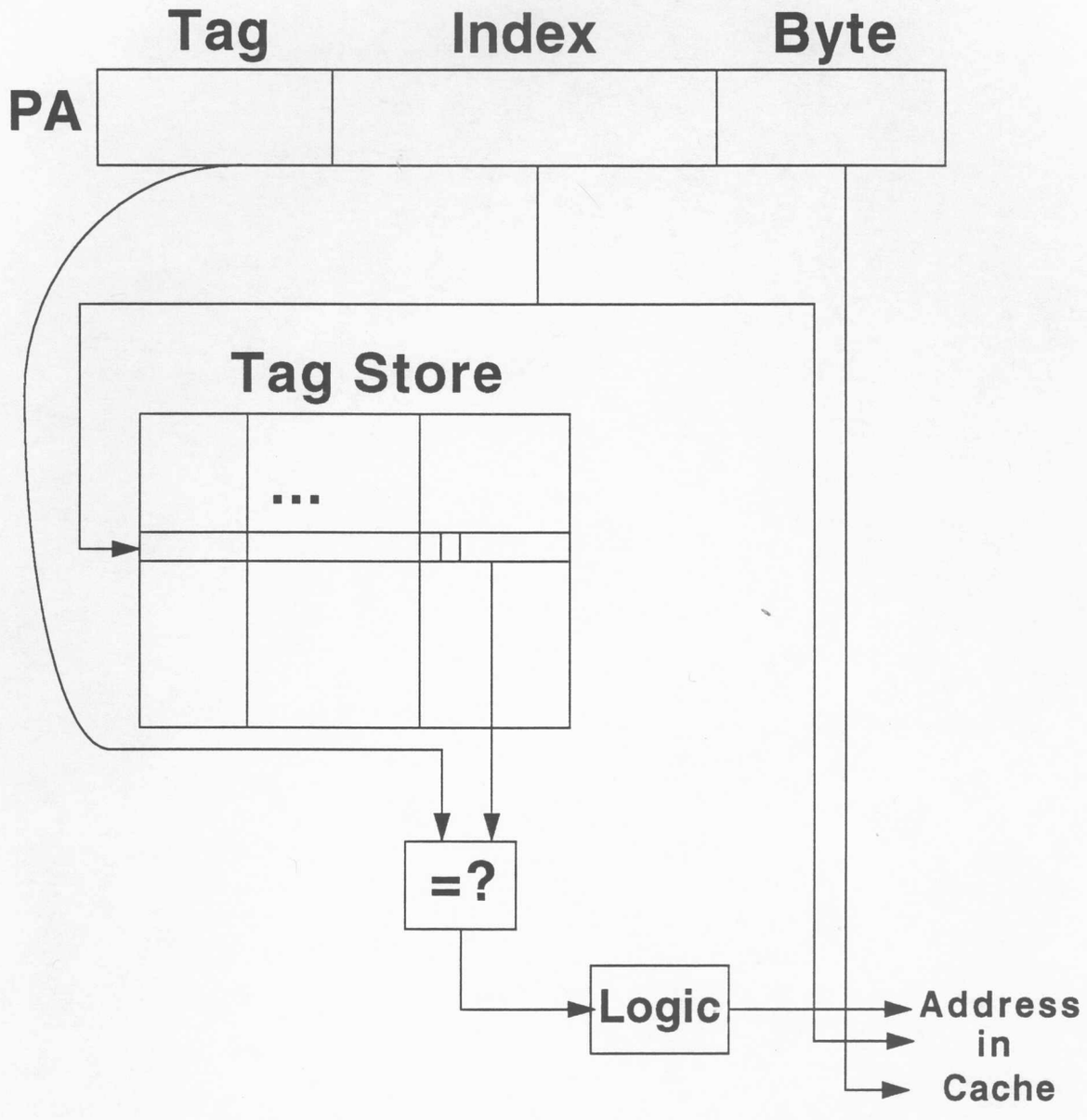


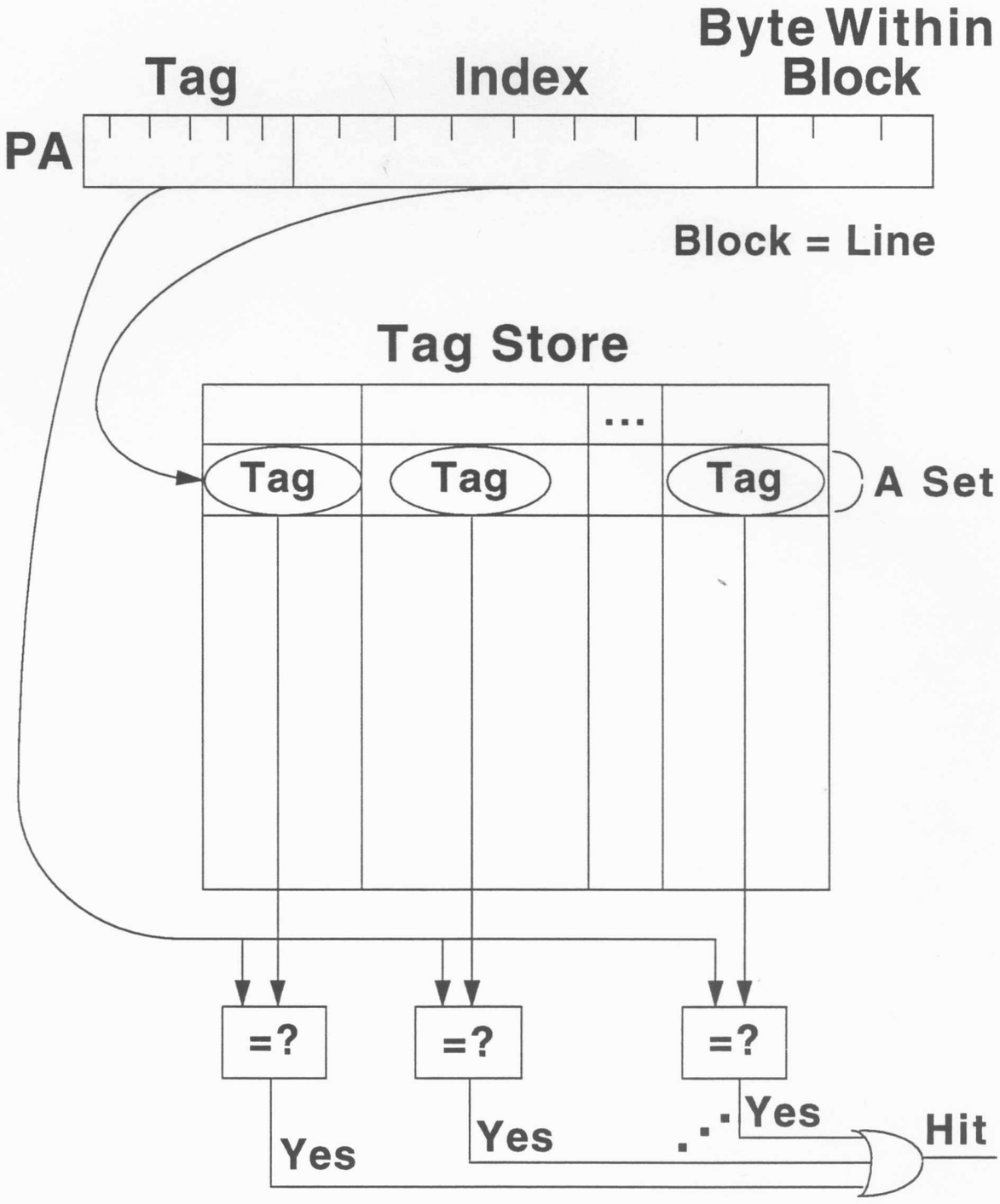
The Abstraction



Note: Hit Ratio =

$$\frac{\text{Hits}}{\text{Hits + Misses}}$$

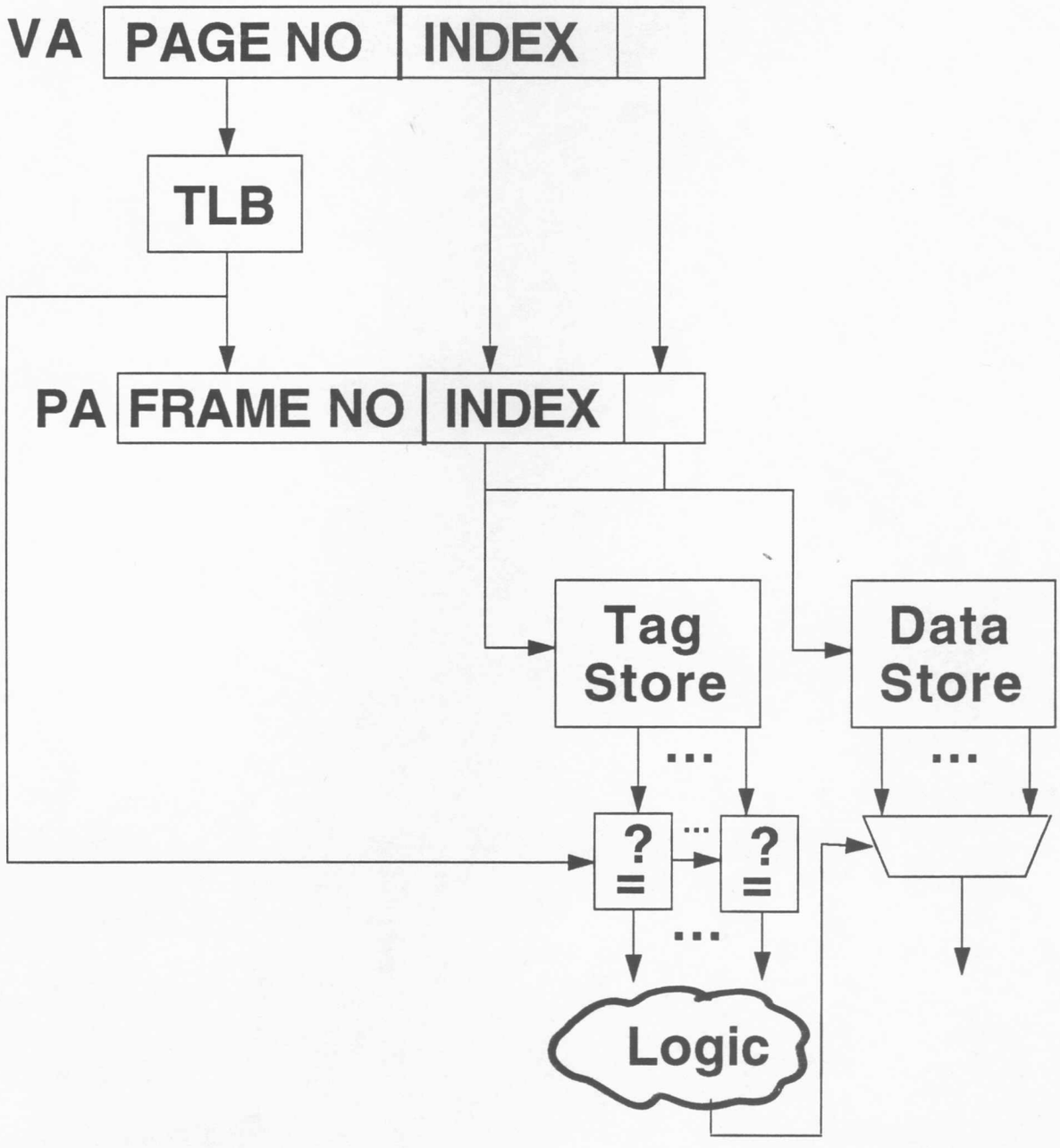

Tag
Index
Byte
PA
Tag Store
...
=?
Logic
Address
in
Cache


Byte Within
Block
Tag
Index
PA
Block = Line
Tag Store
...
Tag
Tag
Tag
A Set
=?
=?
=?
Yes
Yes
Yes
Hit

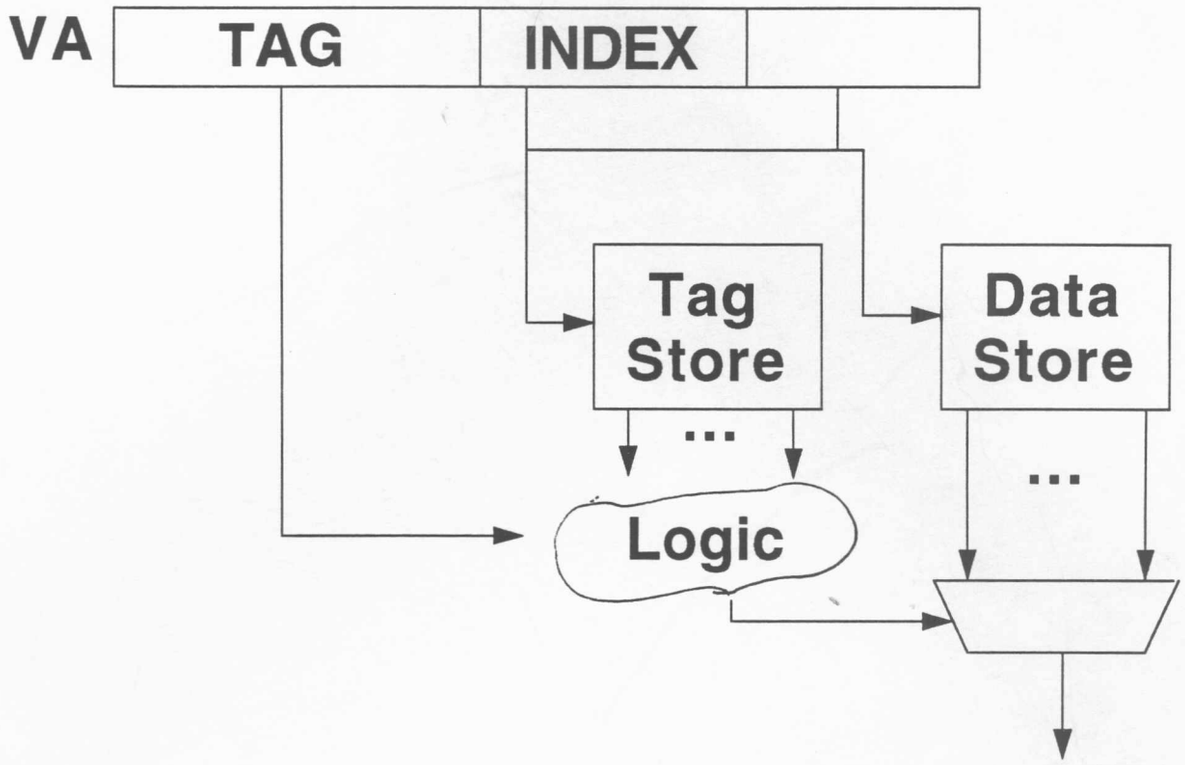
Characteristics

* **Set Associative (Set Size)**
  - **Fully Associative**
  - **Direct Mapped**
* **Write Back, Write Through**
* **Replacement Algorithm**
  - **LRU**
  - **FIFO**
  - **Random**
* **Instructions/Data**
* **Superviser/User**
* **Virtual/Physical**

Virtual/Physical



Virtual/Physical

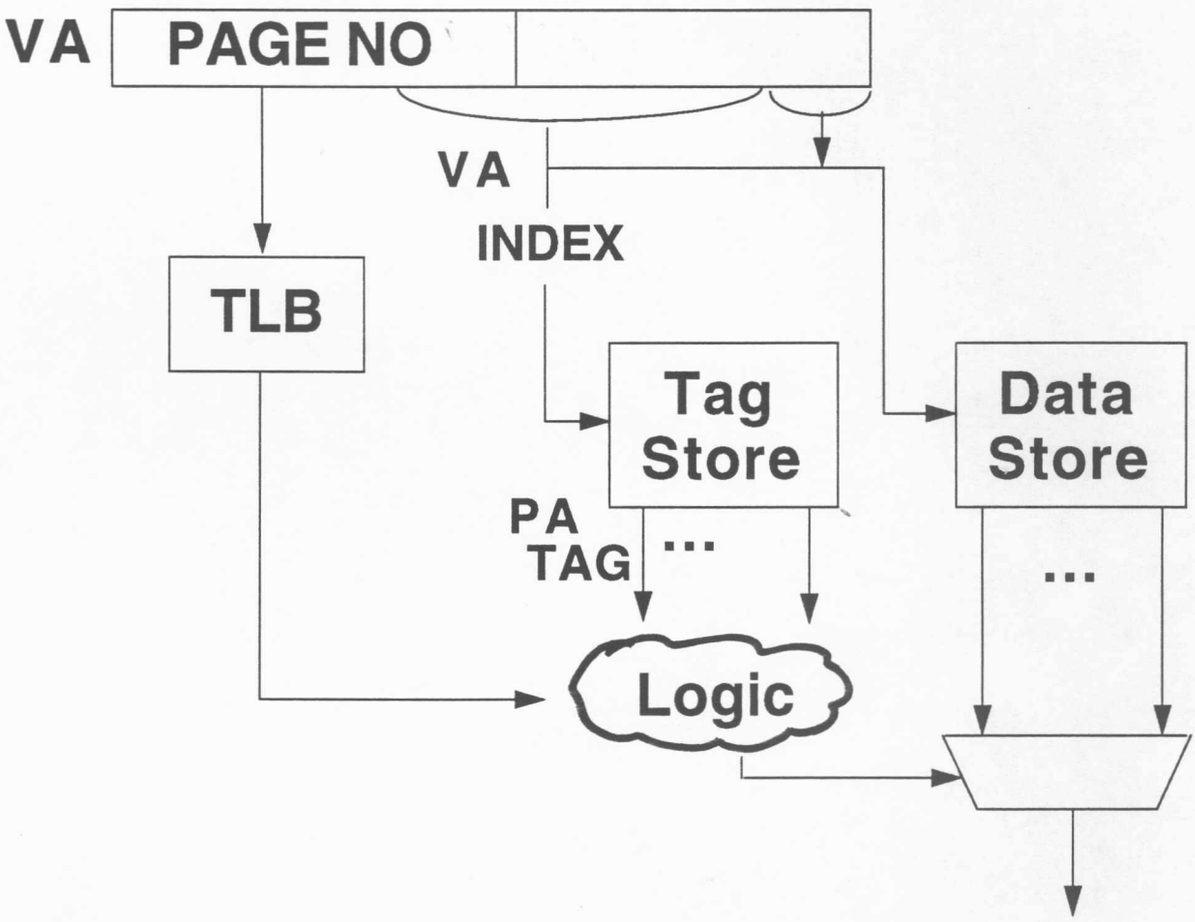


Plus: Don't wait for translation

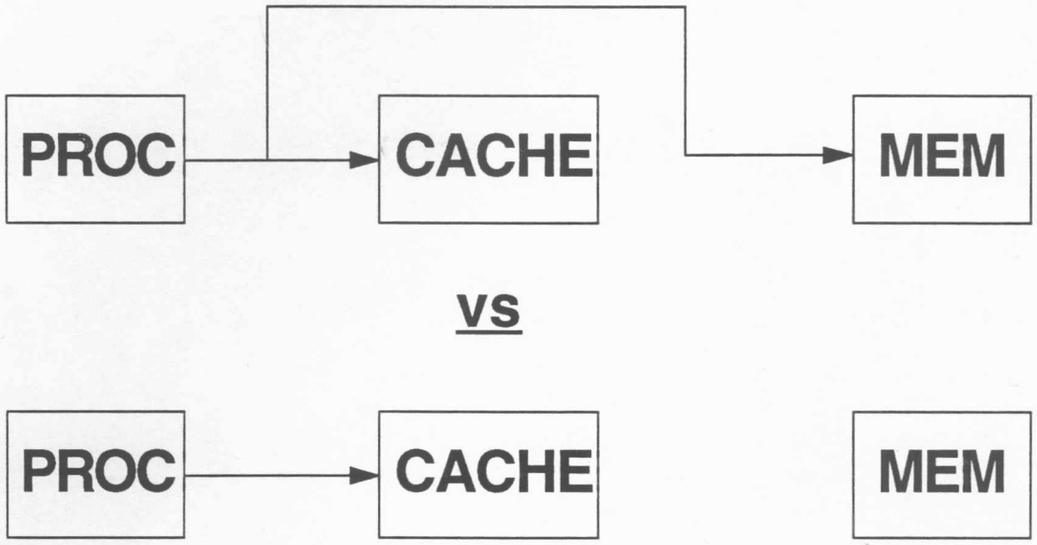
Minuses: Same VA refers to Two Things

One PA Stored in Two Places

Virtual/Physical



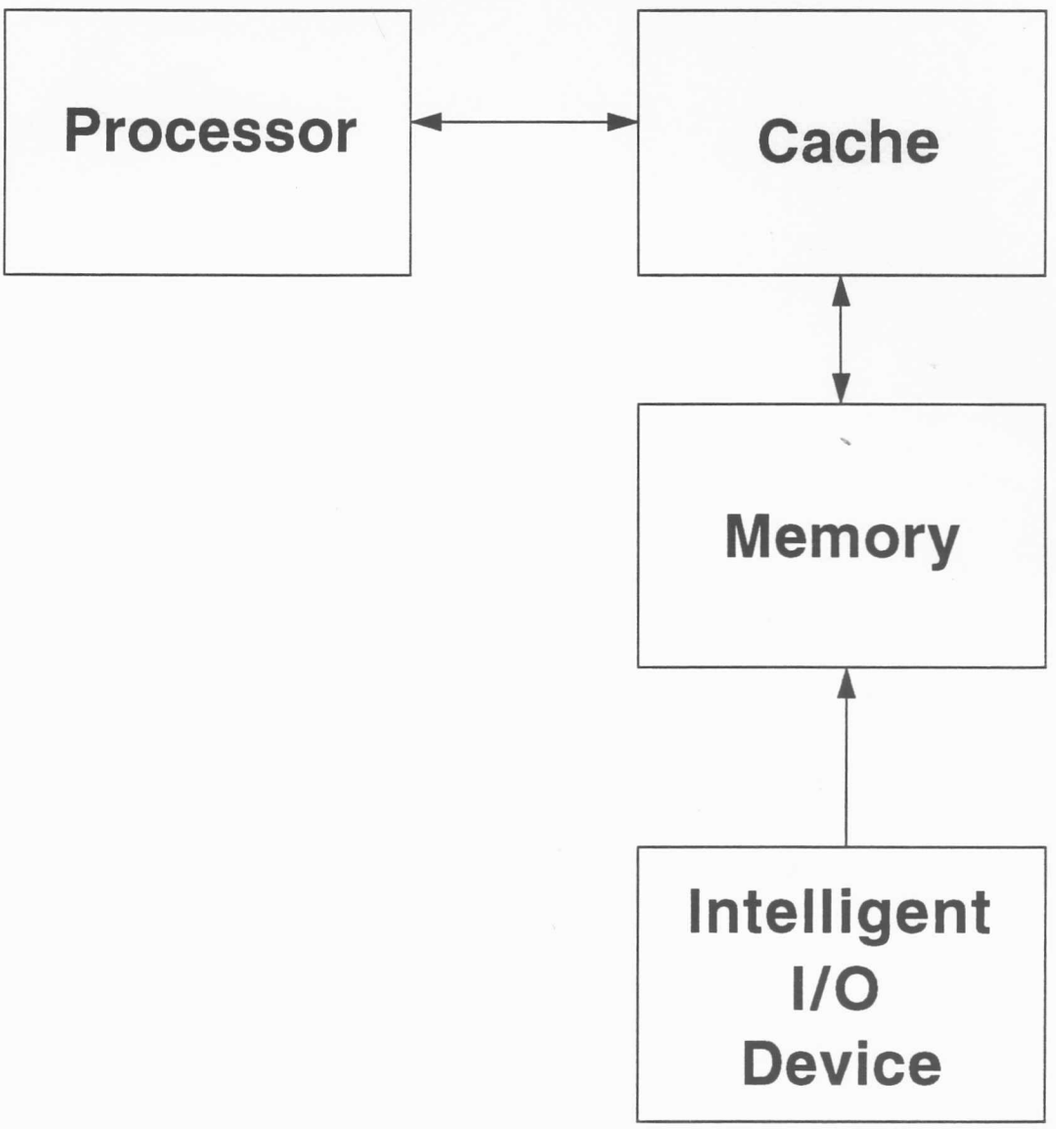
Write Through/Write Back



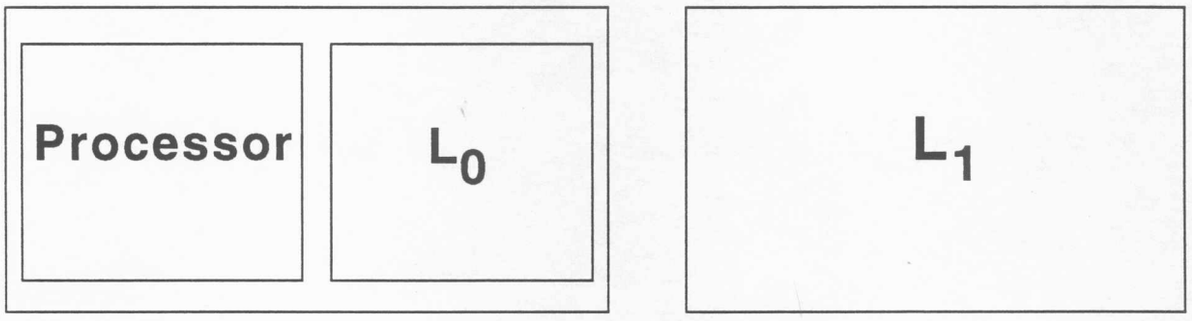
Issues

* **Simplicity of Design**
* **Bus Traffic**
* **Application Environment (Stack Frame)**
* **Allocate on Write Miss**
  - **Sector Cache**

Uniprocessor Cache Consistency

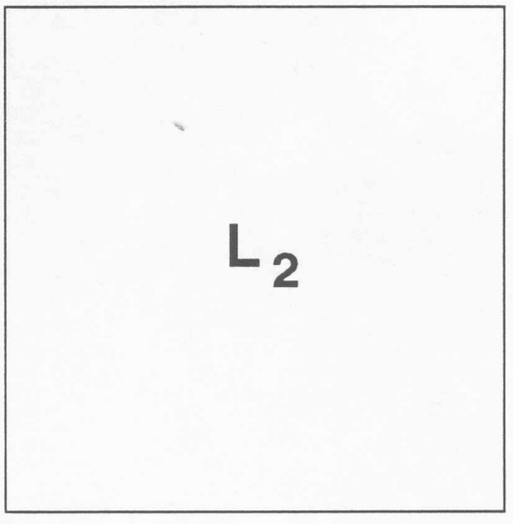


Multiple Levels



5 nsec

20 nsec



150 nsec

* Today's Trend:

  Make Cache Visible

  Prefetch Instructions

* Inclusion Property