Cache Memory

WHY?

CPU

1 nsec

Memory

100 nsec
The Abstraction

Physical Address

Tag Store

To see if it is there

"Yes" (A Hit!)

Data

To get It

Data

(Also called Cache A)

(Also called Cache D)

Note: Hit Ratio = \[
\frac{\text{Hits}}{\text{Hits} + \text{Misses}}
\]
Characteristics

* Set Associative (Set Size)
  - Fully Associative
  - Direct Mapped

* Write Back, Write Through

* Replacement Algorithm
  - LRU
  - FIFO
  - Random

* Instructions/Data

* Supervisor/User

* Virtual/Physical
Virtual/Physical

VA  TAG  INDEX

Tag Store  Data Store

Logic

Plus:  Don’t wait for translation

Minuses:  Same VA refers to Two Things
           One PA Stored in Two Places
Write Through/Write Back

\[
\text{PROC} \rightarrow \text{CACHE} \rightarrow \text{MEM} \quad \text{vs} \quad \text{PROC} \rightarrow \text{CACHE} \rightarrow \text{MEM}
\]

**Issues**

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

Processor  

Cache

Memory

Intelligent I/O Device
Multiple Levels

Processor  $L_0$  $L_1$

5 nsec  20 nsec

* Today's Trend:
  Make Cache Visible
  Prefetch Instructions

* Inclusion Property

$L_2$

150 nsec

Memory