# 308-273
# Caches, Part II

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

# Review

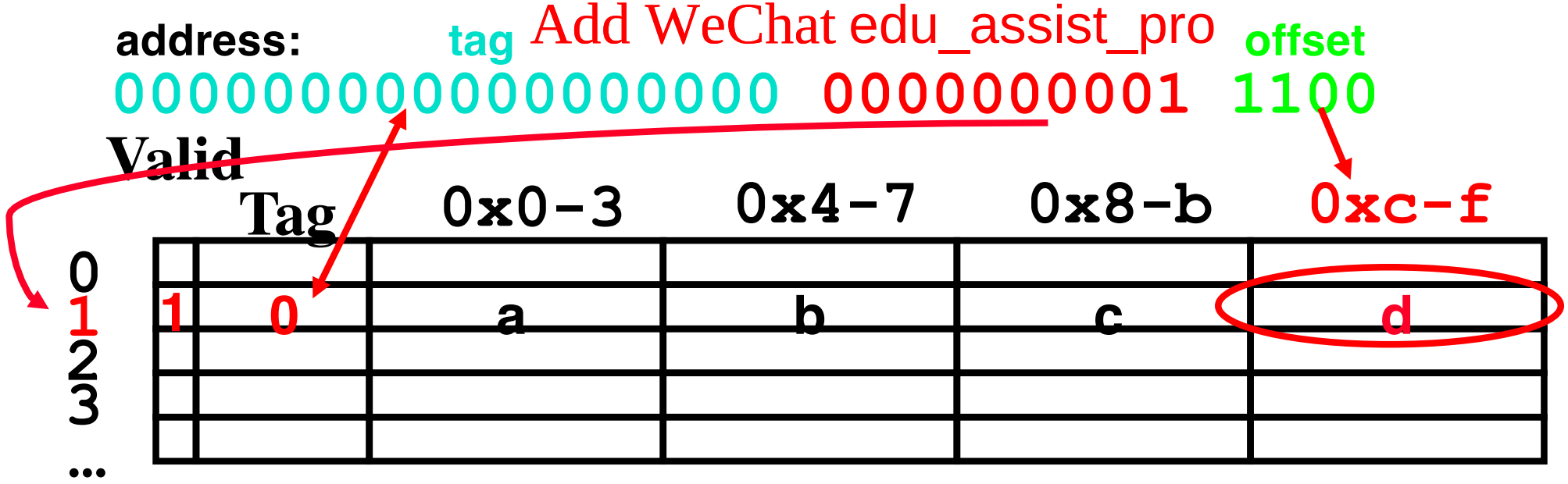
- We would like to have the capacity of disk at the speed of the processor: unfortunately this is not feasible.
- So we create a memory hierarchy:
  - each suc vel contains "most us lower level
  - exploits temporal lo
  - do the common case fast, worry less about the exceptions (design principle of MIPS)
- Locality of reference is a Big Idea

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Big Idea Review (1/2)

- **Mechanism for transparent movement of data among levels of a storage hierarchy**
  - **set of address/value bindings**
  - **address => index to set of candidates**
  - **compare desired address with tag**
  - **service hit or miss**
    - **load n on miss**

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro



# Outline

- Block Size Tradeoff
- Types of Cache Misses
- Fully Associative Cache
- N-Way Ass
- Block Replacement
- Multilevel Caches (if time)
- Cache write policy (if time)

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Block Size Tradeoff (1/3)

- Benefits of Larger Block Size
  - Spatial Locality: if we access a given word, we're likely to access other nearby words soon (Another Big Idea)
  - Very applic -Program Concept: ven instruction, it's likel 'll execute the next few as well
  - Works nicely in sequential array accesses too

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Block Size Tradeoff (2/3)

- Drawbacks of Larger Block Size
  - Larger block size means larger miss penalty
    - on a miss, takes longer time to load a new block from next level
  - If block s ive to cache size, then ther its
    - Result: miss rate g
- In general, minimize Average Access Time

  = Hit Time + Miss Penalty x Miss Rate

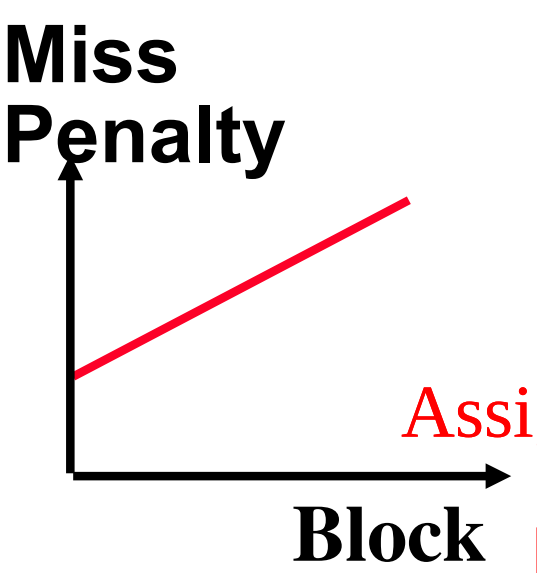
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

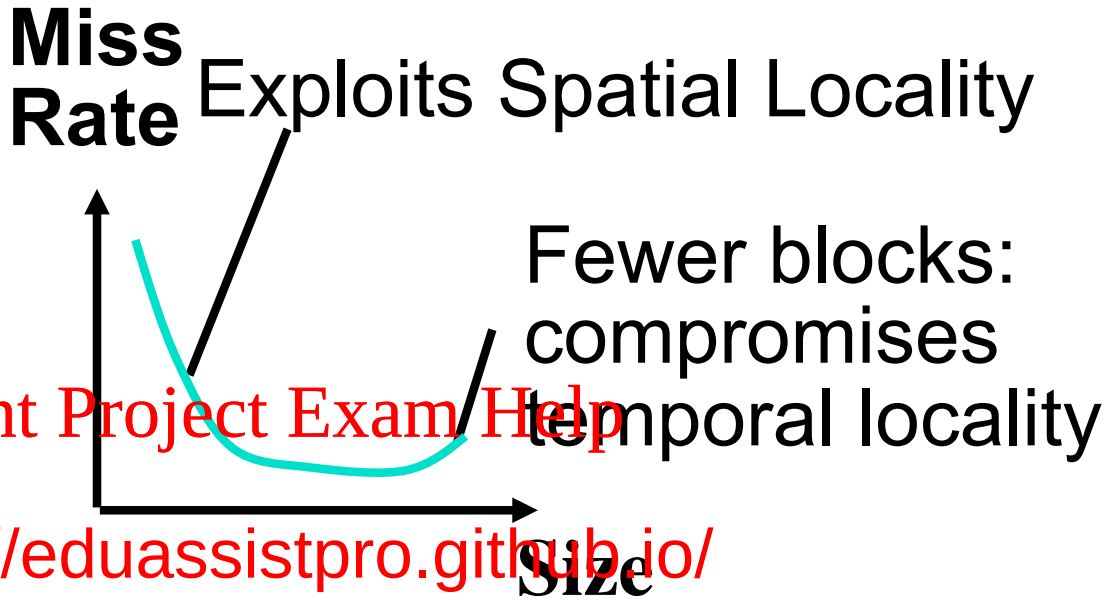
# Block Size Tradeoff (3/3)

- **Hit Time** = time to find and retrieve data from current level cache
- **Miss Penalty** = average time to retrieve data on a current level miss (includes the possib n successive levels of m y)
- **Hit Rate** = % of requ at are found in current level cache
- **Miss Rate** = 1 - Hit Rate

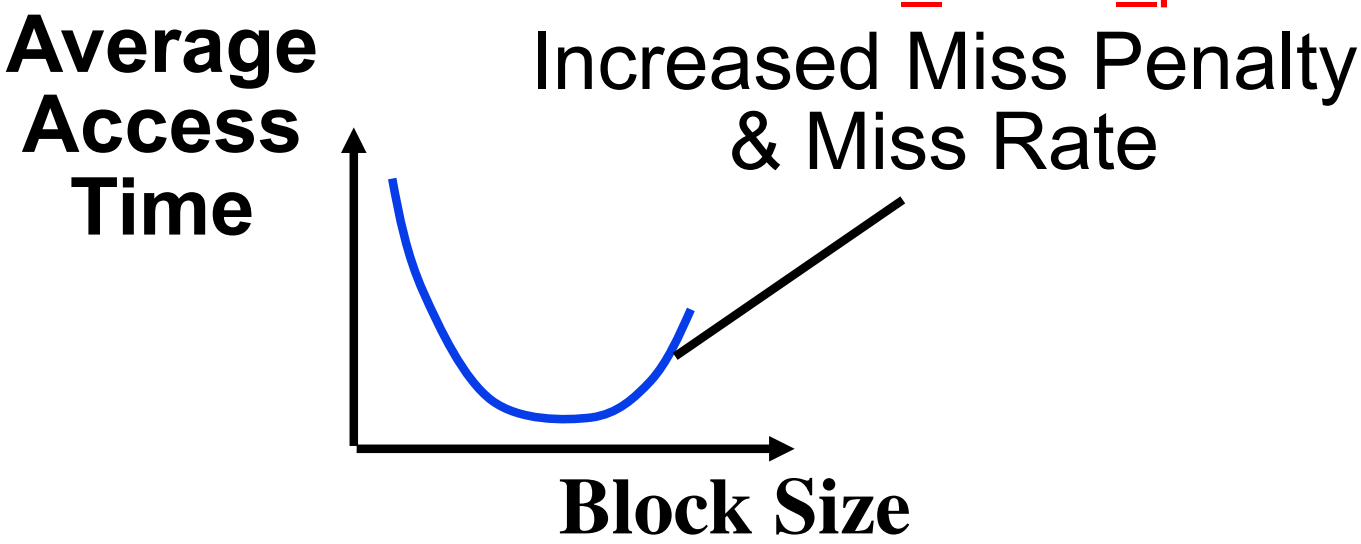
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Block Size Tradeoff Conclusions




Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro



# Types of Cache Misses (1/2)

- **Compulsory Misses**
  - occur when a program is first started
  - cache does not contain any of that program's data yet, so misses are bound to occur
  - can't be avoided ea on't focus on these in this cou

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Types of Cache Misses (2/2)

- **Conflict Misses**
  - **miss that occurs because two distinct memory addresses map to the same cache location**
  - **two bloc to map to the same loc verwriting each other**
  - **big problem in direct-mapped caches**
  - **how do we lessen the effect of these?**

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Dealing with Conflict Misses

- Solution 1: Make the cache size bigger
  - relatively expensive
- Solution 2: Multiple distinct blocks can fit in the same Cache Index?

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

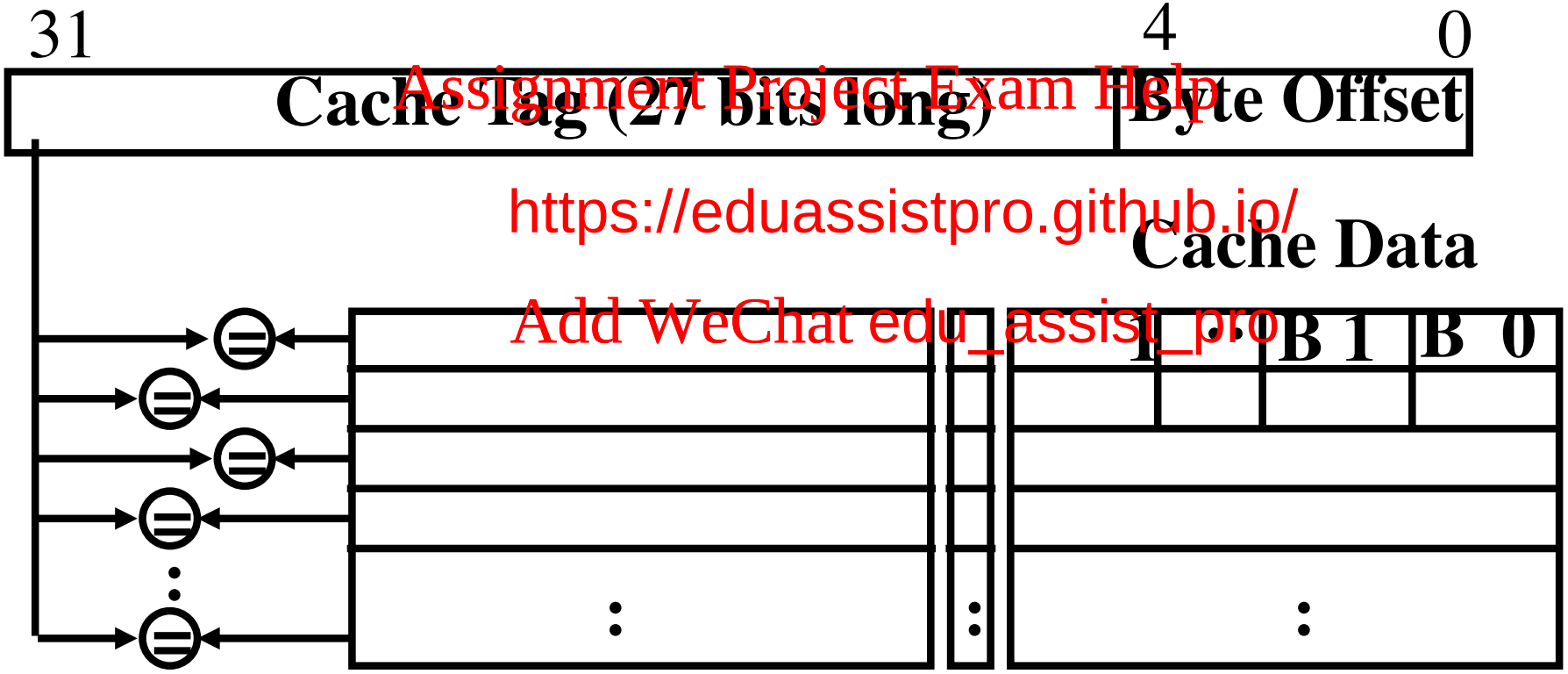
# Fully Associative Cache (1/3)

- Memory address fields:
  - Tag: same as before
  - Offset: same as before
  - Index: non-existent
- What does
  - any block can go an n the cache
  - must compare with all tags in entire cache to see if data is there

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Fully Associative Cache (2/3)

- **Fully Associative Cache (e.g., 32 B block)**
  - **compare tags in parallel**




Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Fully Associative Cache (3/3)

- Benefit of Fully Assoc Cache
  - no Conflict Misses (since data can go anywhere)
- Drawback
  - need har single entry: if we h cache with 4B entri comparators: very expensive
- Small fully associative cache may be feasible

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Cache for every KB of data in ed 16K

# Third Type of Cache Miss

- **Capacity Misses**
  - **miss that occurs because the cache has a limited size**
  - **miss that would not occur if we increase the size**
- **This is the primary type of miss for Fully Associate cac**

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# N-Way Set Associative Cache (1/4)

- Memory address fields:
  - Tag: same as before
  - Offset: same as before
  - Index: po ect "row" (called a
- So what's the differ
  - each set contains multiple blocks
  - once we've found correct set, must compare with all tags in that set to find our data

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# N-Way Set Associative Cache (2/4)

- Summary:
  - cache is direct-mapped with respect to sets
  - each set is fully associative
  - basically N direct-mapped caches, each of which is full iative. Each has its own valid bit a_

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# N-Way Set Associative Cache (3/4)

- Given memory address:
  - Find correct set using Index value.
  - Compare Tag with all Tag values in the determined set.
  - If a matc otherwise a miss.
  - Finally, use the offs s usual to find the desired data within the desired block.

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# N-Way Set Associative Cache (4/4)

- What's so great about this?
  - even a 2-way set assoc cache avoids a lot of conflict misses
  - hardware cost isn't that bad: only need N compara
- In fact, for blocks,
  - it's Direct-Mapped if y set assoc (1 block per set)
  - it's Fully Assoc if it's M-way set assoc (M blocks per set)
  - so these two are just special cases of the more general set associative design

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Block Replacement Policy (1/2)

- Direct-Mapped Cache: index completely specifies which position a block can go in on a miss
- N-Way Set Assoc (N > 1): index specifies a [illegible] any positio[illegible] an occupy on a miss
- Fully Associative: bl[illegible] be written into any position (there is no index)
- Question: if we have the choice, where should we write an incoming block?

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Block Replacement Policy (2/2)

- Solution!
- If there are any locations with valid bit off (empty), then usually write the new block into
- If all possi eady have a valid block, we mus replacement policy by which we determine which block gets "cached out" on a miss.

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Block Replacement Policy: LRU

- LRU (Least Recently Used)
  - Idea: cache out block which has been accessed (read or write) least recently
  - Pro: temporal locality => recent past use implies li fact, this is a very effe
  - Con: with 2-way set asy to keep track (one LRU bit); with 4-way or greater, requires complicated hardware and much time to keep track of this

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Block Replacement Example

- We have a 2-way set associative cache with a four word *total* capacity and one word blocks. We perform the following word accesses (ignore bytes for this problem):

Assignment Project Exam Help

0, 2, 0,

https://eduassistpro.github.io/

How many hits and misses will there for the LRU block replacement policy?

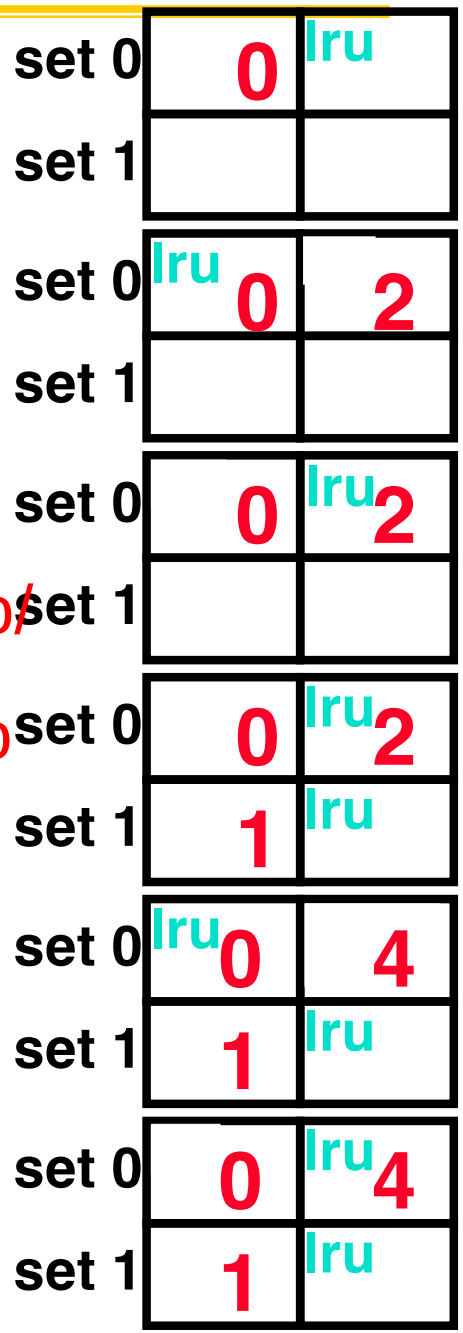
Add WeChat edu_assist_pro

Hint: treat addresses as TAG + INDEX

# Block Replacement Example: LRU

- Addresses 0, 2, 0, 1, 4, 0, ...
  - 0: miss, bring into set 0 (loc 0)
  - 2: miss, bring into set 0 (loc 1)
  - 1: miss, bring into s 0)
- 4: miss, bring into set 0 (loc 1, replace 2)
  - 0: hit

Assignment Project Exam Help
hit
https://eduassistpro.github.io/
Add WeChat edu_assist_pro



# Ways to reduce miss rate

- Larger cache
  - limited by cost and technology
  - hit time of first level cache < cycle time
- More plac to put each block of m tivity
  - fully-associative
    - any block any line
  - k-way set associated
    - k places for each block
    - direct map: k=1

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Big Idea

- How do we chose between options of associativity, block size, replacement policy?
- Design against a performance model
  - Minimize: Hit Time + Miss Rate x Miss Penalty
  - influenced by technology and program behavior

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Example

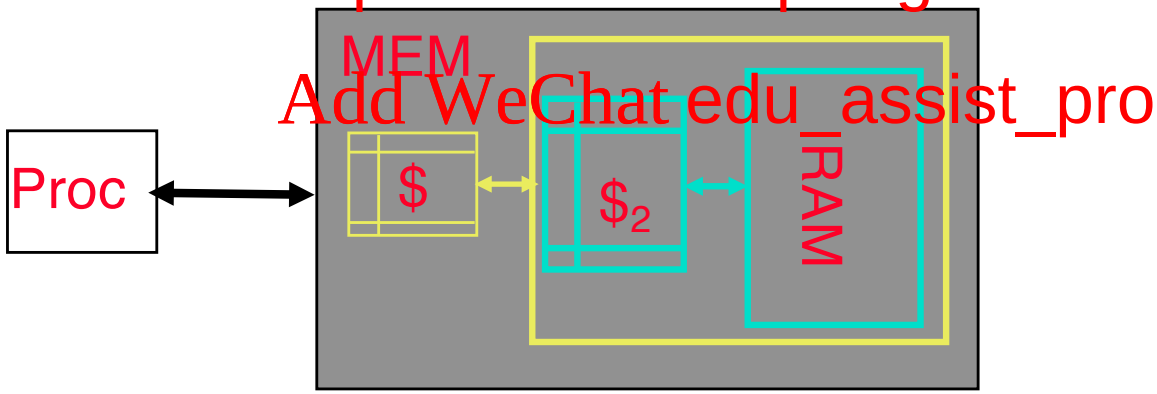
- Assume
  - Hit Time = 1 cycle
  - Miss rate = 5%
  - Miss pen
- Avg mem a 05 x 20 ycle

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Improving Miss Penalty

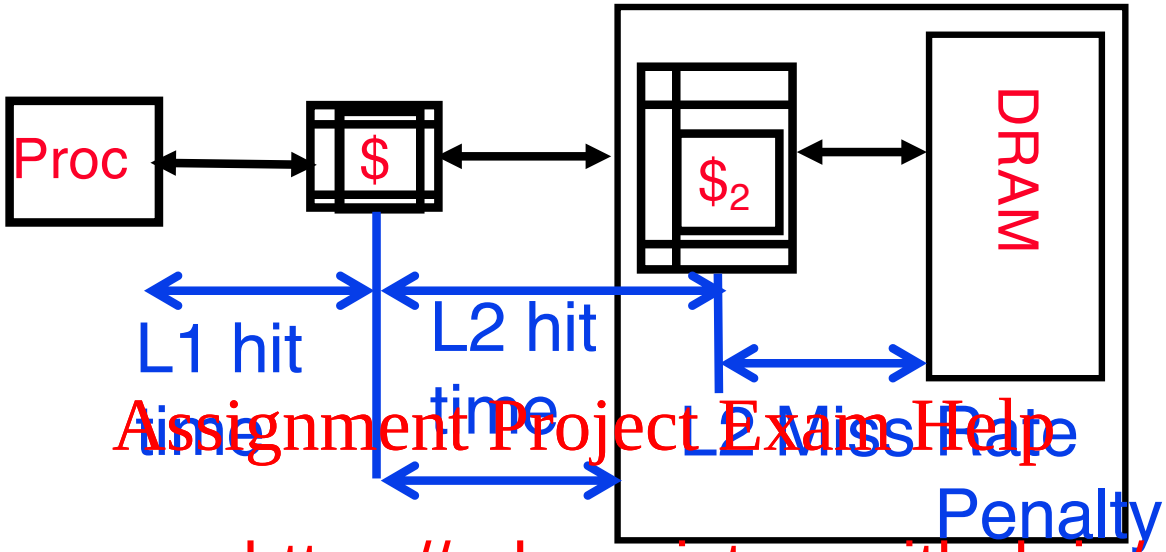
- When caches first became popular, Miss Penalty ~ 10 processor clock cycles
- Today 1000 MHz Processor (1 ns per clock cycle) and 100 ns to go to DRAM ⇒ 100 processor clock cycles!

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro




Solution: another cache between memory and the processor cache: Second Level (L2) Cache

# Analyzing Multi-level cache hierarchy




Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

**Avg Mem Access Time =**
**L1 Hit Time + L1 Miss Rate * L1 Miss Penalty**

**L1 Miss Penalty = L2 Hit Time + L2 Miss Rate**
*** L2 Miss Penalty**

**Avg Mem Access Time =**
**L1 Hit Time + L1 Miss Rate * (L2 Hit Time +**
**L2 Miss Rate * L2 Miss Penalty)**

# Typical Scale

- L1
  - size: tens of KB
  - hit time: complete in one clock cycle
  - miss rates: 1-5%
- L2:
  - size: hun
  - hit time: few clock c
  - miss rates: 10-20%
- L2 miss rate is fraction of L1 misses that also miss in L2
  - why so high?

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# Example: without L2 cache

- Assume
  - L1 Hit Time = 1 cycle
  - L1 Miss rate = 5%
  - L1 Miss Pe
- Avg mem ac 5 x 100

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

6 cycles

# Example with L2 cache

- Assume
  - L1 Hit Time = 1 cycle
  - L1 Miss rate = 5%
  - L2 Hit Tim
  - L2 Miss ra isses that miss)
  - L2 Miss Penalty = 100
- L1 miss penalty = 5 + 0.15 * 100 = 20
- Avg mem access time = 1 + 0.05 x 20
  = 2 cycle
- 3x faster with L2 cache

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# What to do on a write hit?

- **Write-through**
  - update the word in cache block and corresponding word in memory
- **Write-back**
  - update w k
  - allow memory word tale”

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

=> add ‘dirty’ bit to each line indicating that memory needs to be updated when block is replaced

=> OS flushes cache before I/O !!!

- Performance trade-offs?

# "And in conclusion…" (1/2)

- Caches are NOT mandatory:
  - Processor performs arithmetic
  - Memory stores data
  - Caches simply make data transfers go faster
- Each level of memory hierarchy is just a subset of next higher level
- Caches speed up due to **temporal locality**: store data used recently
- Block size > 1 word speeds up due to **spatial locality**: store words adjacent to the ones used recently

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

# "And in conclusion…" (2/2)

- Cache design choices:
  - size of cache: speed v. capacity
  - direct-mapped v. associative
  - for N-way set assoc: choice of N
  - block repl
  - 2nd level cache?
  - Write through v. write back?
- Use performance model to pick between choices, depending on programs, technology, budget, ...

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro