

# Chapter 09: Caches

## Lesson 05:

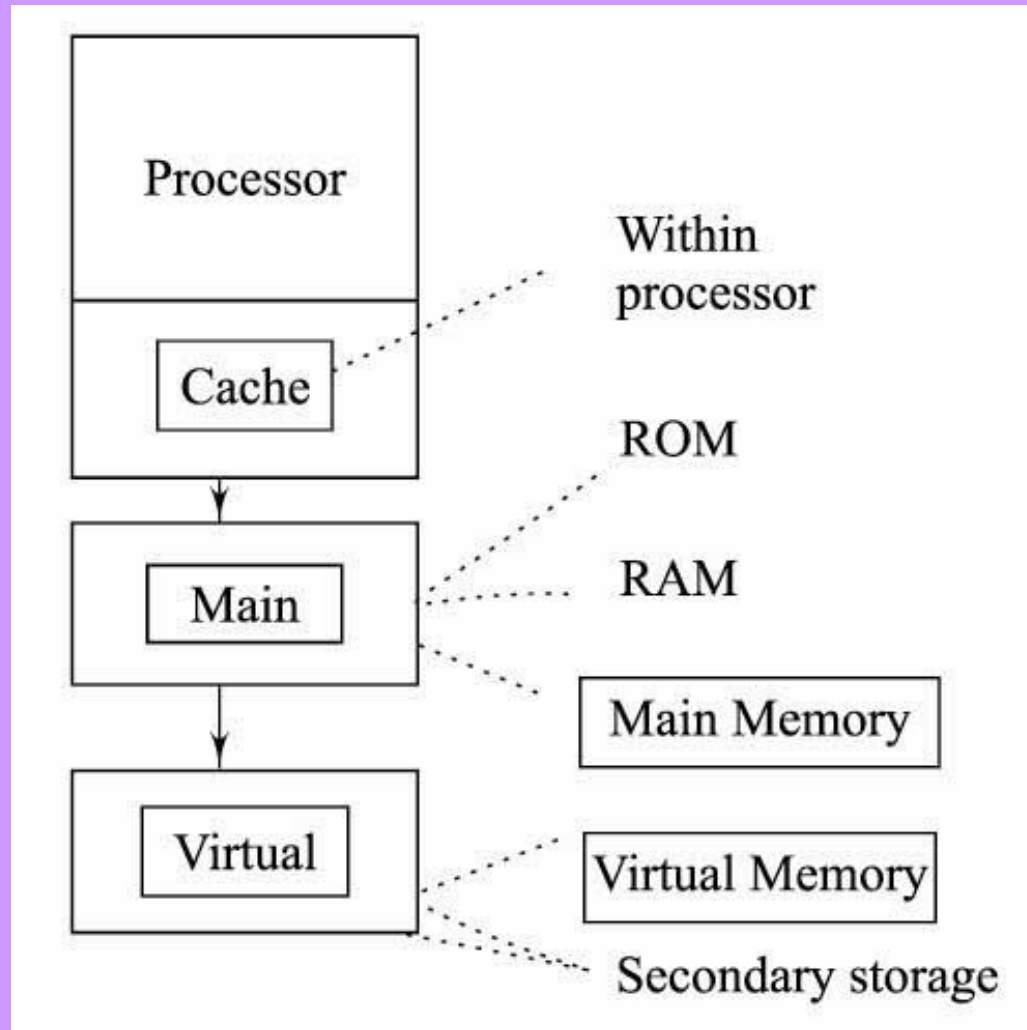
### Write through and write back policy

# Objective

- Comparisons between the write back and write through caches

# Write Back vs. write through

# Memory levels



# Write Through

- Store operation— If a cache is write-through, stored values are written into the level (cache) and sent to the next level down (main memory) as the is executed

# Write-back

- Store operation— Write-back levels keep stored data in the level, and when a block that has been written is evicted from the level, the contents of the block are written back (copied) into the next level down
- Write-back levels keep stored data in the cache, and when a block that has been written is evicted from the level, the contents of the block are written back (copied) into the main memory down

# Write through

- Ensures that the contents of the level and the next level down are always the same
- Write-back levels keep stored data in the level, the next level down become the same after the contents of the block are written back (copied) into the next level down

# Cache write policy



# Cache Write Policy

- Caches may be implemented as either write-back or write-through systems, and both approaches have their advantages

# Write Through Advantage

- It is not necessary to record which lines have been written
- Because the data in a write-through cache is always consistent with the contents of the next level
- Evicting a line can be done by writing the new line over the old one, reducing the time to bring a line into the cache

# Write back advantage

- Only write their contents back when a line is evicted
- If a given line receives multiple store requests while it is in the cache, waiting until the line is evicted can significantly reduce the number of writes sent to the next level of the cache hierarchy

# Cache write back policy

# Write back Hardware

- Tracks whether or not each line has been written since it was fetched
- Also, when a dirty line (one that has been written since it was brought into the cache) must be evicted from the cache to make room for an incoming line
- Write-back caches require that the incoming line wait for the outgoing line to be written back, increasing the delay until the incoming line is available

# Write-back caches when including the write buffers

- Temporary storage for lines that are being written back to avoid the delay

# Cache write back performance

# Write-back caches Performance

- Higher performance than write-through caches
- Cache lines that are written at least once tend to be written enough times to make writing the entire cache line back once when it is evicted, more efficient than sending each write to the next level of the hierarchy as it occurs



# Write Back Effect

- Even more significant if the next level of the hierarchy implemented in page-mode DRAM
- Page mode to reduce the time to write the line back to the next level when a block write occurs

# Example for non-page mode DRAM

- Assume— The level below a cache in the memory hierarchy implemented out of non-page-mode DRAM that requires 60 ns to read or write a word of data
- Assume— cache lines are 8 lines long
- Find how many times does the average line have to be written (counting only lines that are written at least once) before a write-back cache is more efficient than a write-through cache

# Solution

- In a write-back cache, each dirty 8-word line is written back once for every time it is brought into the cache, taking 480 ns ( $8 \times 60$  ns)
- In a write-through cache, each store operation requires that its data be written through to the next level, taking 60ns
- Therefore, if the average line that gets written at least once gets written more than 8 times, the write-back cache will be more efficient

# Solution for page mode DRAM

- Assume— a first-word access time of 60 ns, and then 10 ns/word thereafter
- Writing back a line takes  $60 \text{ ns} + 7 \times 10 \text{ ns} = 130 \text{ ns}$ , enough time for  $130/60 = 2.17$  single-word memory operations
- Therefore, if the average line that is written at least once is written more than 2.17 times, the write-back cache will be more efficient.

# Summary

# We learnt

- Write through and write back caches and their comparison
- Write-back caches higher performance
- Cache lines that are written at least once tend to be written enough times to make writing the entire cache line back once when it is evicted, more efficient than sending each write to the next level of the hierarchy as it occurs

End of Lesson 05 on  
**Write through and write back policy**