JOINT INSTITUTE
交大密西根学院

ADAC LAB

# ECE3700J RC
# Cache 2

Presenter: Ruan Renjian 阮仁剑

SJTU

# T/F Questions Related to Performance



T 1. It is possible for a write-back and a write-through cache to perform the same number of writes.

F 2. Cache is about the same size as main memory.

F 3. A cache with very large block sizes must have less compulsory misses than a cache with smaller block sizes.

F 4. A write-through cache system will always write to both cache and memory

F 5. Larger block size can always reduce CUP time.

# Memory Write Through



## Handling Data Writes – Write Through

- On data-write (e.g. sw) hit, could just update the block in cache
  - But then cache and memory would be inconsistent
- Write through: also update the word in memory

- But makes writes take longer time
  - Must wait till the update finishes

## Write Buffer

- Solution to time consuming write through technique (for both hit and miss)
  - Buffer stores data to be written to memory
    - May have one or more entries
  - CPU proceeds to next step, while letting buffer to complete write through
  - Frees buffer when completing write to memory
  - CPU stalls if buffer is full

# Memory Write Back



## Handling Data Writes – Write Back

- Alternative of write through: On data-write hit, just update the block in cache
  - CPU keeps track of whether each block is *dirty* (updated with new values)
- Write a block back to memory
  - Only when a dirty block has to be replaced (on miss)
  - More complex than write through

- Write back sequence
  - Two steps:
    - 1. check match,
    - 2. write data
  - Otherwise, will destroy the mismatch block, and there is no backup copy
  - May use write buffer
    - Writing buffer and checking match simultaneously

# Write Allocation

## Write allocation

For write through on miss:

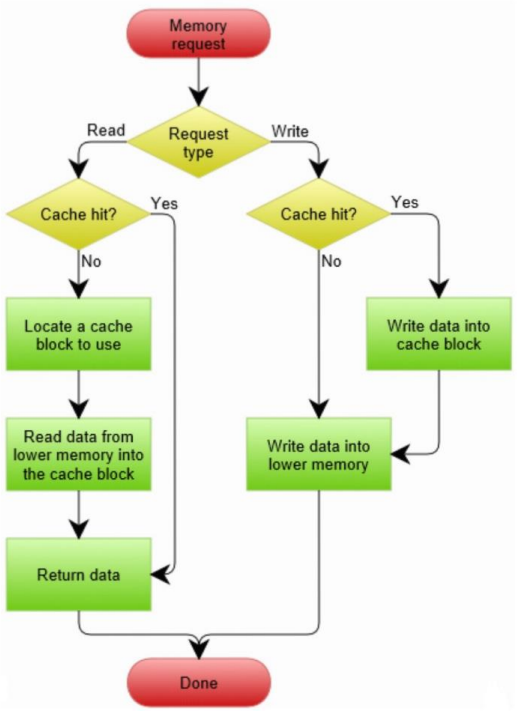
**Write allocate**: Allocate cache block on miss by fetching corresponding memory block. Then modify the cache block and main memory block

**No write allocate**: Write directly to main memory and then fetch the block to cache.

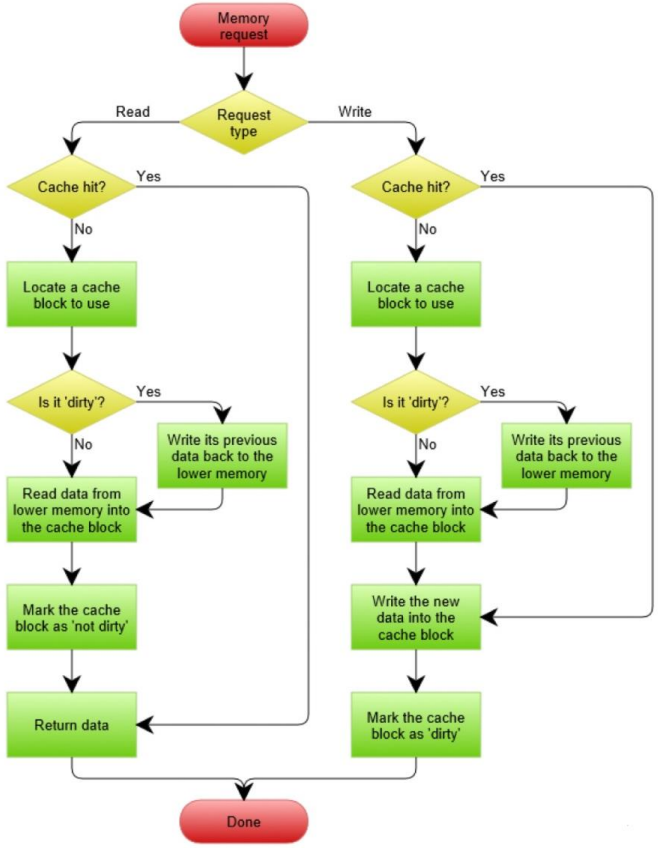
For write back on miss:

Usually use write allocate.

write through with no write allocation




write back with write allocation



# CPU Performance

Components of CPU time = program execution cycles (hit time included) + memory stall cycles

Memory stall cycles = memory-access instruction count * miss rate * miss panelty

Given instruction cache miss rate 1%, data cache miss rate 4%, miss panelty for both 100 cycles, base CPI is 2, and we have 50% memory access instructions. Then what is the total CPI?

total CPI = base CPI + instruction cache total miss cycles + data cache total miss cycles
= 2 + 100% * 1% * 100 + 50% * 4% * 100
= 5