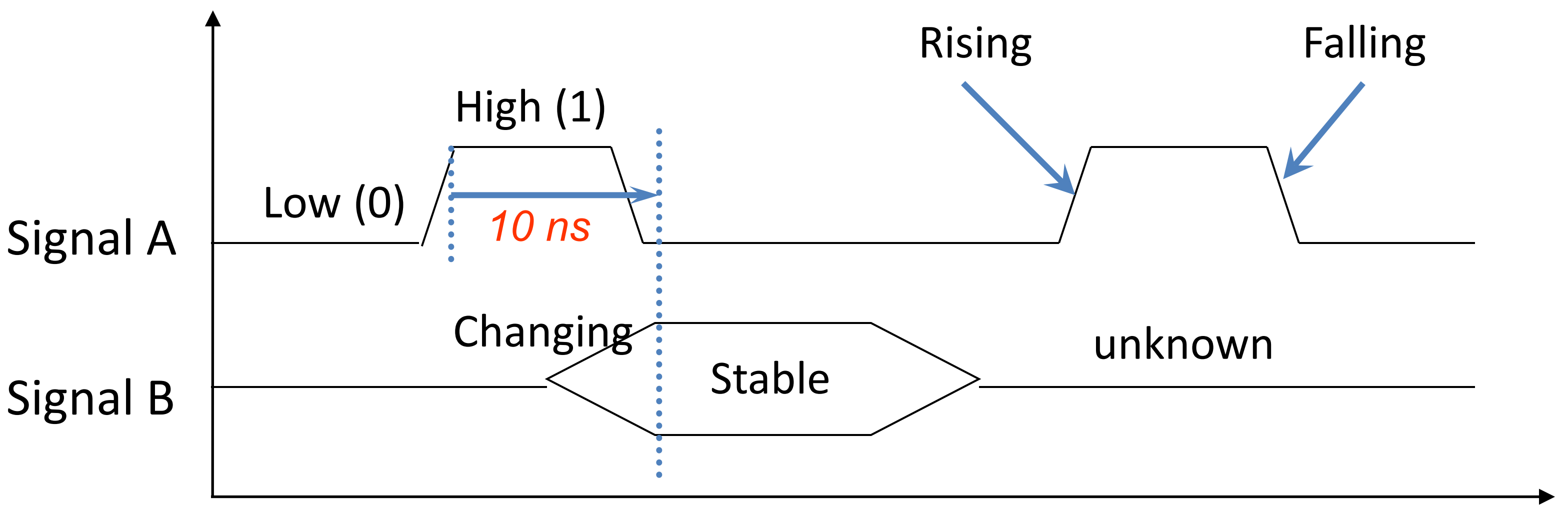
Recap from last class

- The CPU Bus
  - A set of wires and protocols for CPU to communicate with memory and I/O devices.
  - Four-cycle handshake protocol
  - Timing diagram for typical bus access
- Timing diagram syntax:
  - Constant value (0/1), stable, changing, unknown.

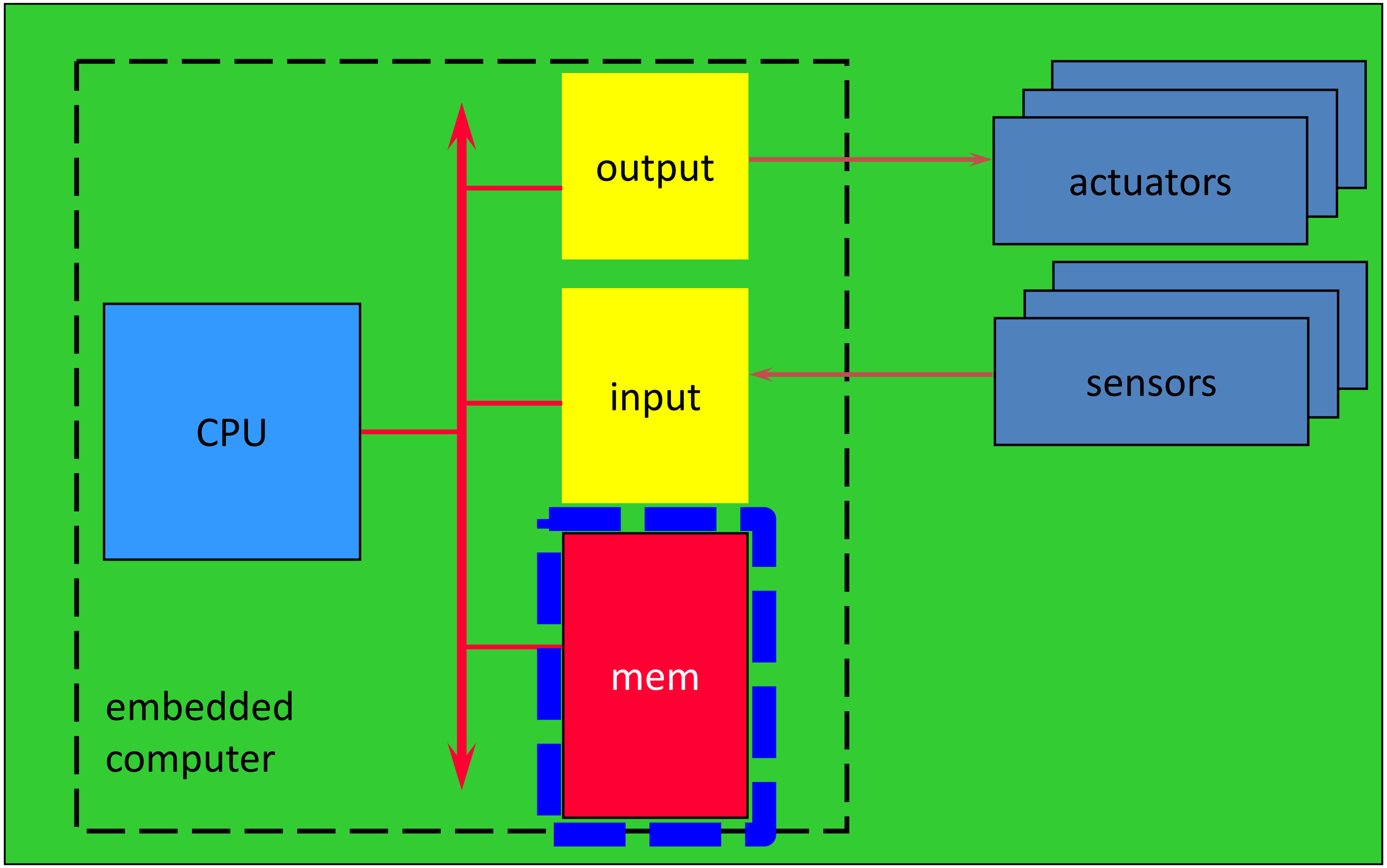


ECE 1175
Embedded Systems Design

Cache and Memory

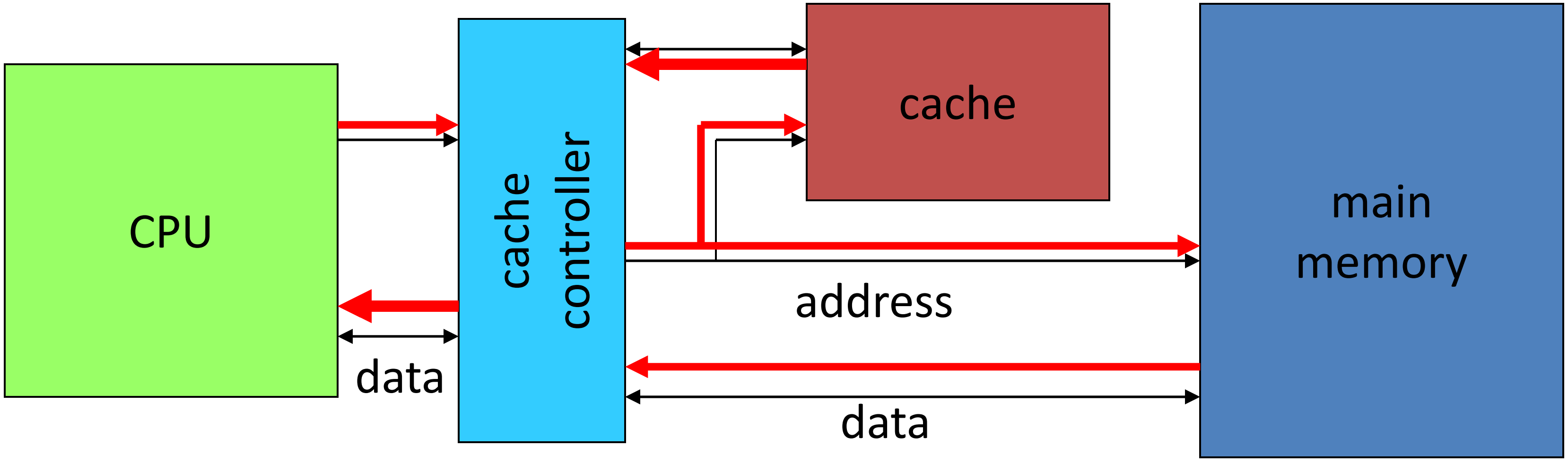
Wei Gao

Embedding A Computer



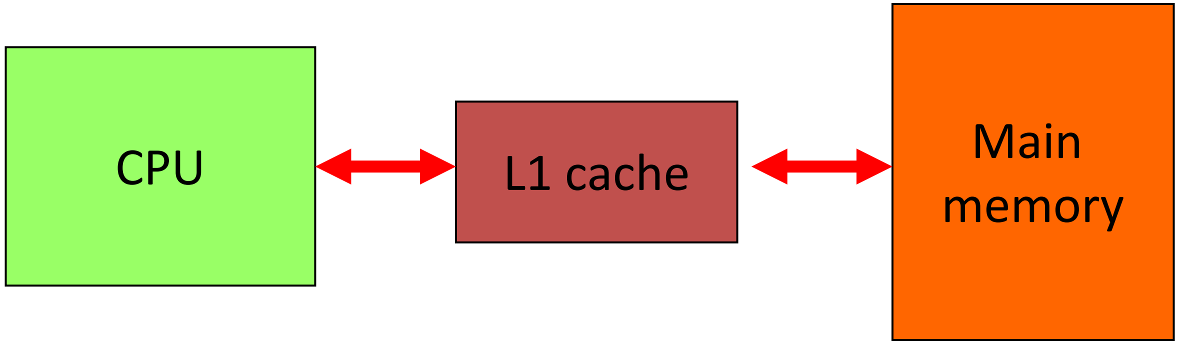
Cache in the Memory System

- Cache controller mediates between CPU and memory system
- Sends a memory request to both cache and main memory
- If requested location is in cache, request to main memory is aborted



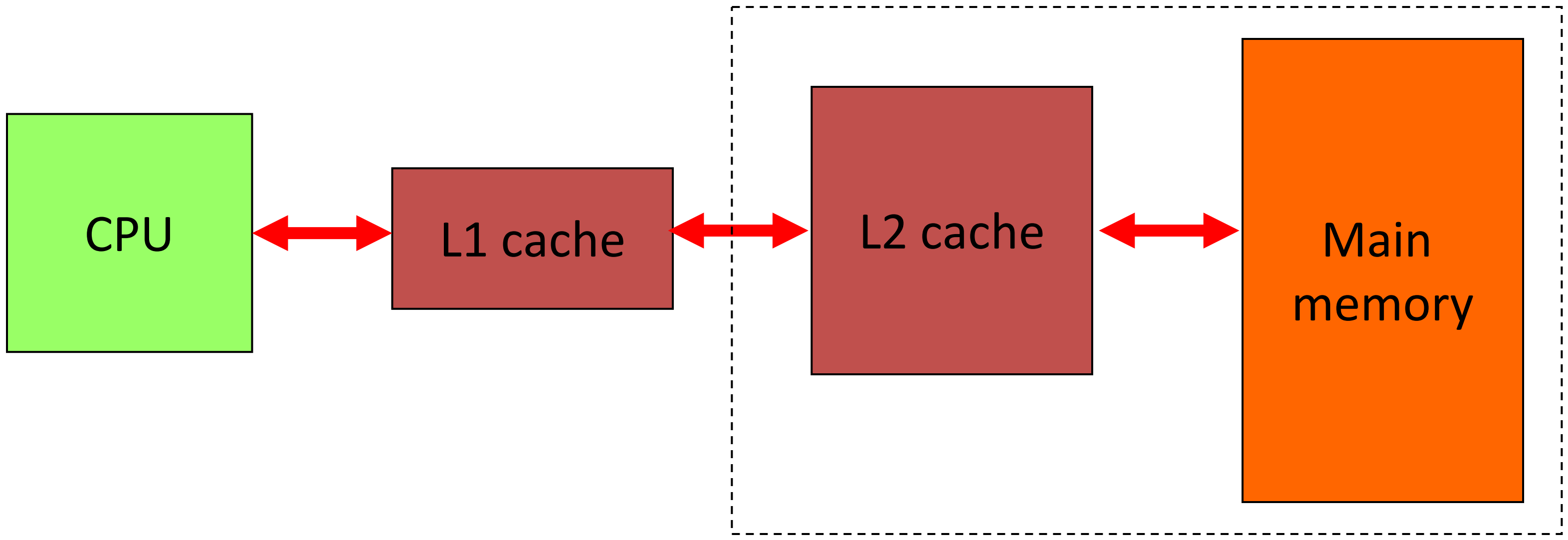
Memory System Performance

- h = cache hit rate: the percentage of cache hits
- t_{cache} = cache access time,
- t_{main} = main memory access time.
- Average memory access time:
  - $t_{av} = ht_{cache} + (1-h)t_{main}$
- Example: t_{cache} = 10ns, t_{main} = 100ns, h = 97%
  - t_{av} = 97%*10ns + (1-97%)*100ns = 12.7ns



Multi-Level Cache Access Time

- h_1 = cache hit rate for L1
- h_2 = cache hit rate for L2
- Average memory access time:
  - $t_{av} = h_1 t_{L1} + (1-h_1)(h_2 t_{L2} + (1-h_2)t_{main})$



Cache Performance Improvement

- To maximize cache hit rate
  - Keep most frequently-accessed memory items in fast cache.
- It is impossible to put everything in small cache
  - Need a good policy to decide which items should be in cache
  - e.g. who should be your favorite 5 people?
    - Nationwide unlimited calls by T-Mobile

Cache Entry Replacement Policies

- Replacement policy: strategy for choosing which cache entry to throw out to make room for a new memory location.
- Two popular strategies:
  - Least-recently used (LRU)
    - Throw out the block that has been used farthest in the past, assuming the chance to use it in the future is small
  - Random
    - Randomly pick one to throw out; requires less hardware

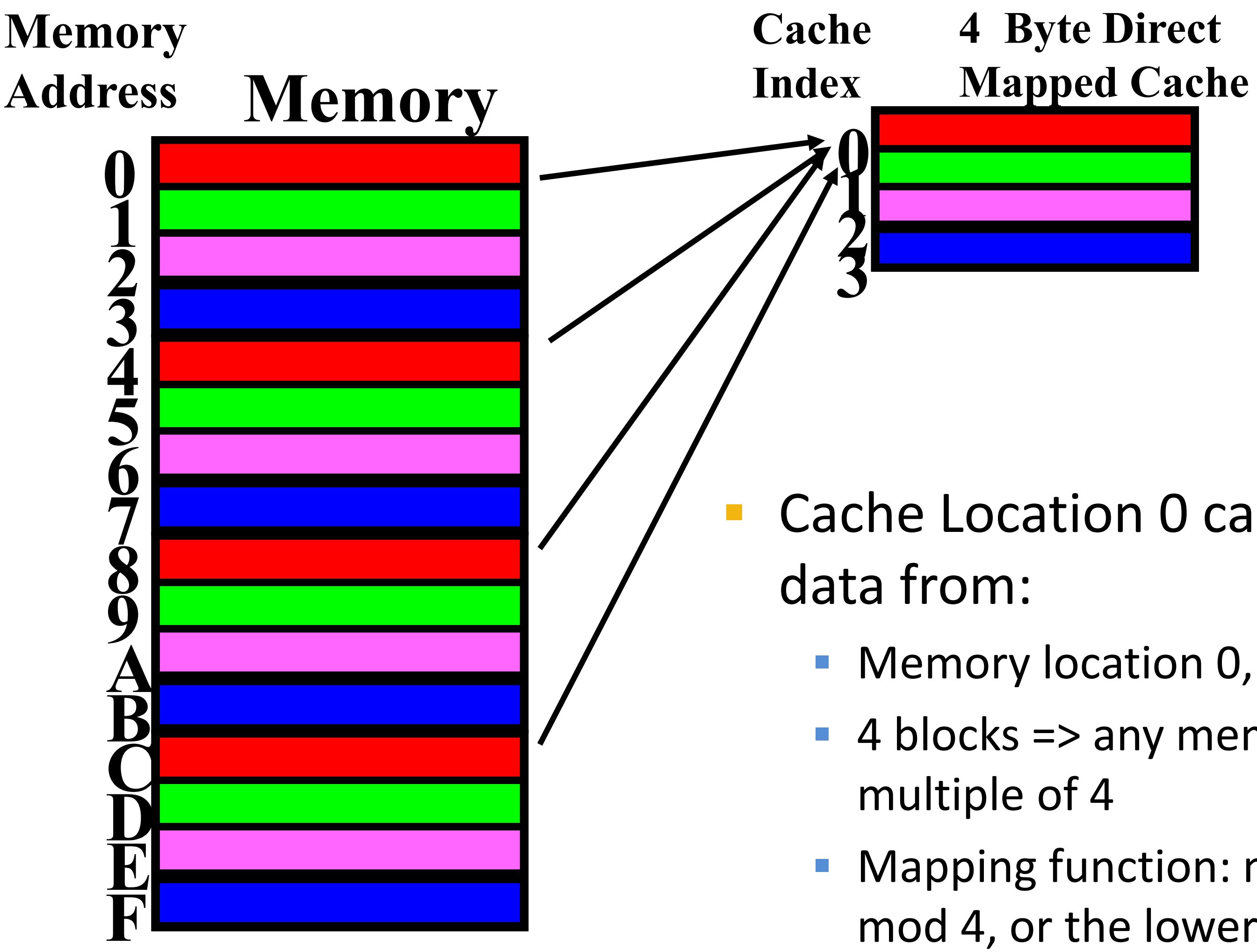
Cache Write Operations

- Cache writes are more complicated than reads
  - Need to update memory as well as cache
- Write-through: immediately copy write to main memory.
  - ✓ Ensures cache and memory are consistent
  - × Additional memory traffic
- Write-back: write to main memory only when location is removed from cache.
  - ✓ Reduces the number of times we write to memory
  - × May cause inconsistency between cache and memory

Cache Organizations

- How should we map memory to cache?
  - Fully-associative: any memory location can be stored anywhere in the cache.
    - Ideal, best cache hit rate but implementation is complex and slow
    - Almost never implemented
  - Direct-mapped: each memory location maps onto exactly one cache entry.
    - Simplest, fastest but least flexible
    - Easy to have conflicts
  - N-way set-associative: each memory location can go into one of n sets.
    - Compromised solution

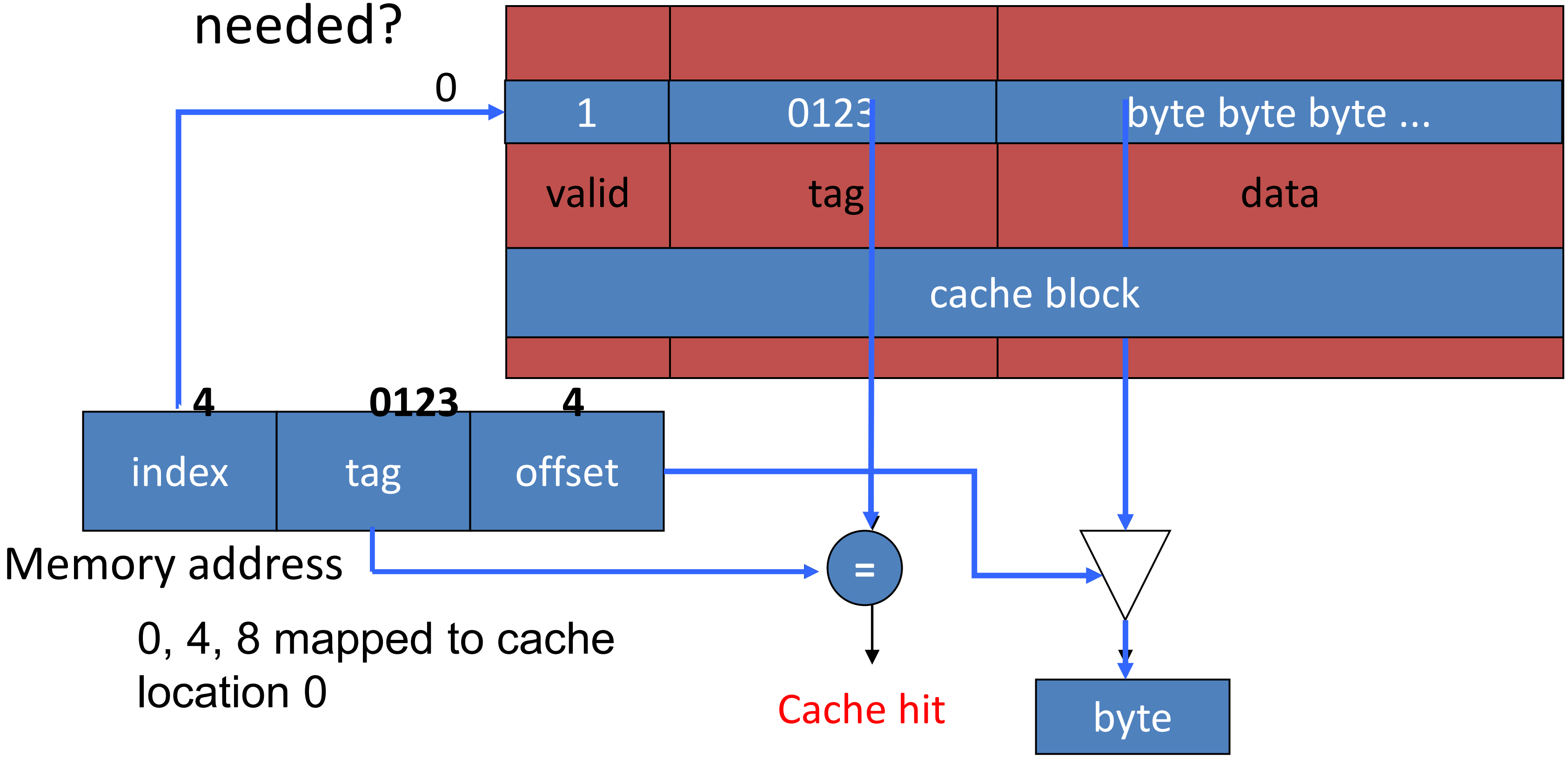
Direct-Mapped Cache



- Cache Location 0 can be occupied by data from:
  - Memory location 0, 4, 8, ...
  - 4 blocks => any memory location that is multiple of 4
  - Mapping function: memory address mod 4, or the lower 2 bits

Direct-Mapped Cache

- Memory address divided to three sections
  - Index: which block to find; tag: compared to the tag used in cache for cache hit; offset: which word in the block is needed?



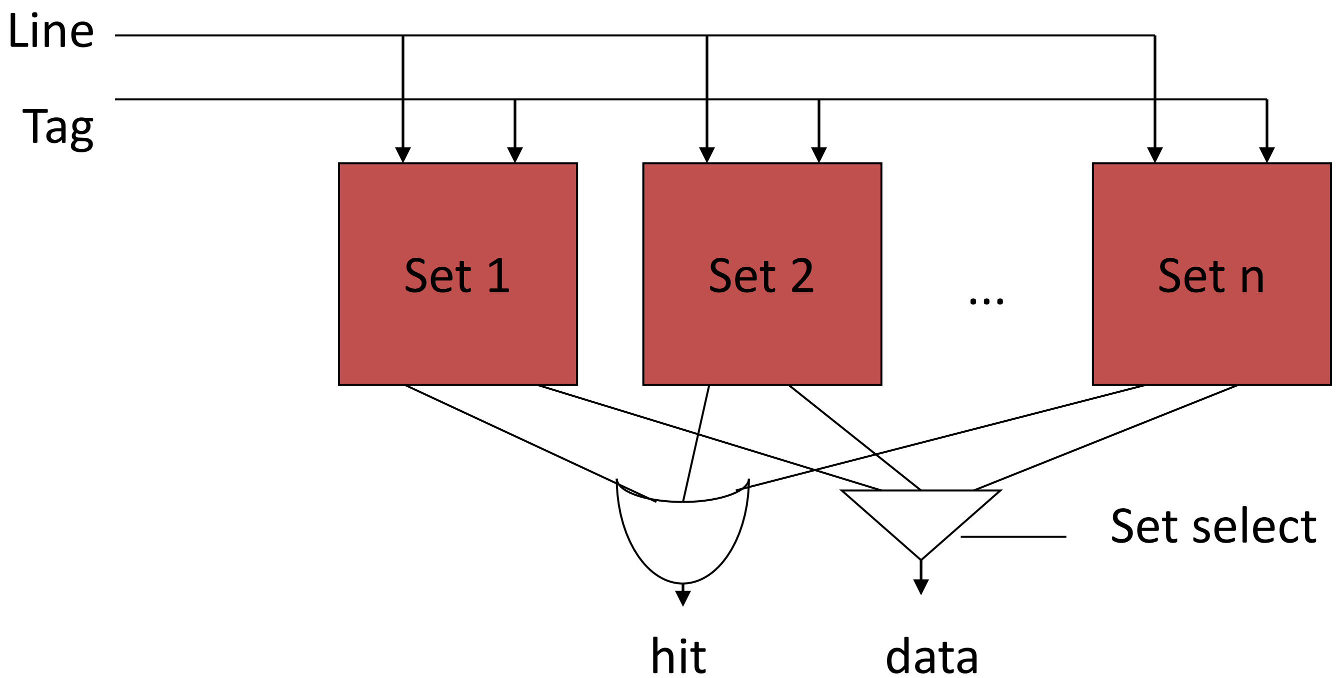
0, 4, 8 mapped to cache location 0

Problems of Direct-Mapped Cache

- Many locations map onto the same cache block.
- Conflict misses are easy to generate:
  - Array a[] uses locations 0, 1, 2, …
    - Mapped to cache 0, 1, 2
  - Array b[] uses locations 1024, 1025, 1026, …
    - Also mapped to cache 0, 1, 2
  - Operation a[i] + b[i] generates conflict misses.

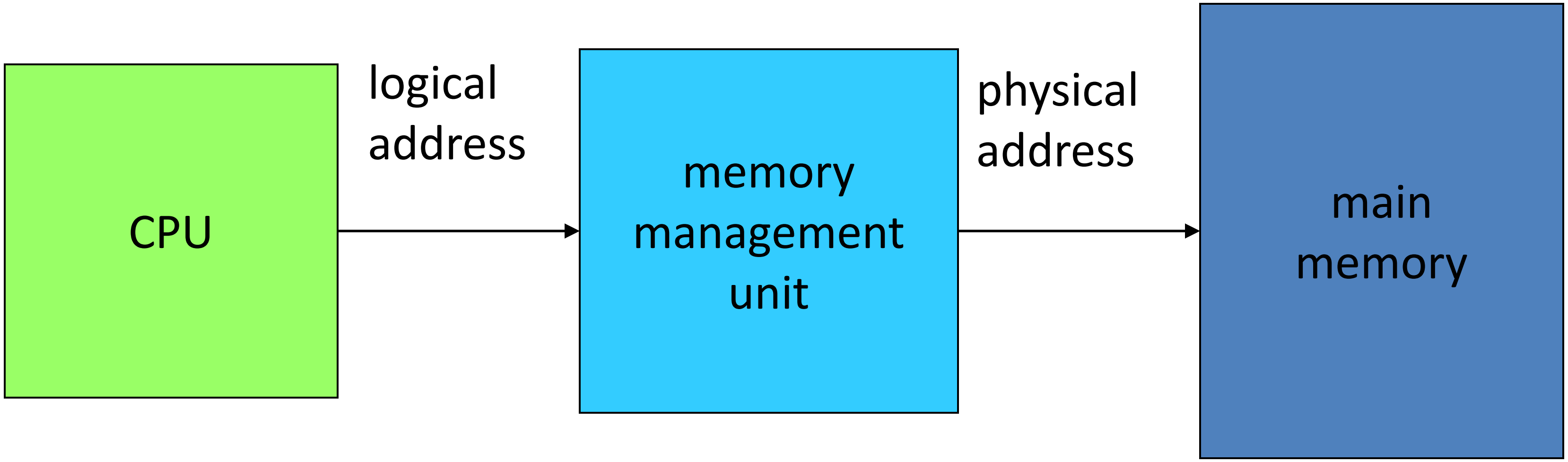
N-Way Set-Associative Cache

- N set of direct-mapped caches
- Each set is implemented with a direct-mapped cache
- A cache request is broadcasted to all sets simultaneously



Memory Management Unit

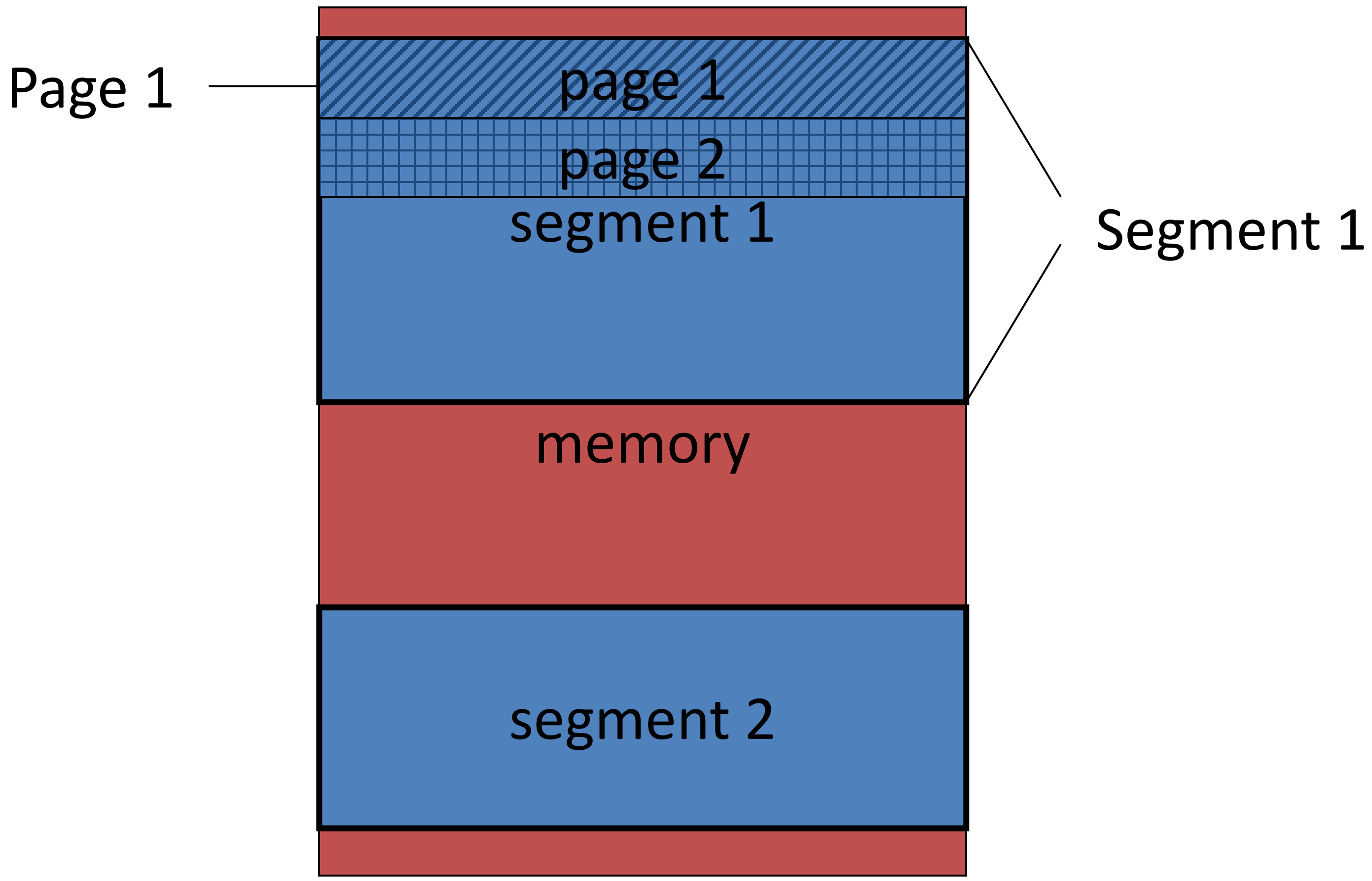
- Memory size is not large enough for all applications?
- Memory management unit (MMU)
  - Provides a larger virtual memory than physical memory
  - Translates logical addresses to physical addresses



Memory Management Tasks

- Allows programs to move in physical memory during execution.
- Allows virtual memory:
  - memory images kept in secondary storage;
  - images returned to main memory on demand during execution.
- Page fault: request for location not resident in memory.

Segments and Pages

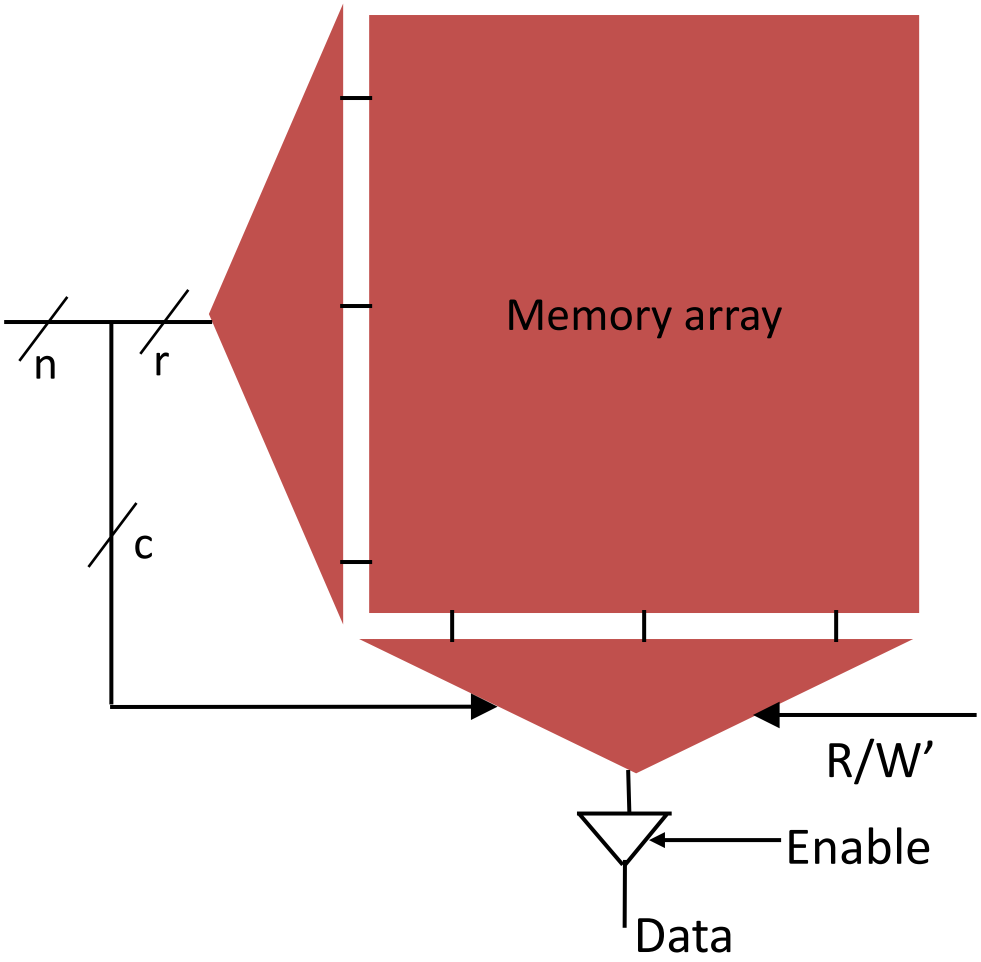


Memory Devices

- Types of memory devices
  - RAM (Random-Access Memory)
    - Address can be read in any order, unlike magnetic disk/tape
    - Usually used for data storage
    - DRAM vs. SRAM.
  - ROM (Read-Only Memory)
    - Usually used or program storage
    - Mask-programmed vs. field-programmable.

Memory Device Organization

- Data stored in a 2-D array of memory cells
- Address split into row and column address
  - n = r + c
- Enable controls the tri-stating of data onto the memory's pins
- R/W controls the direction of data transfer

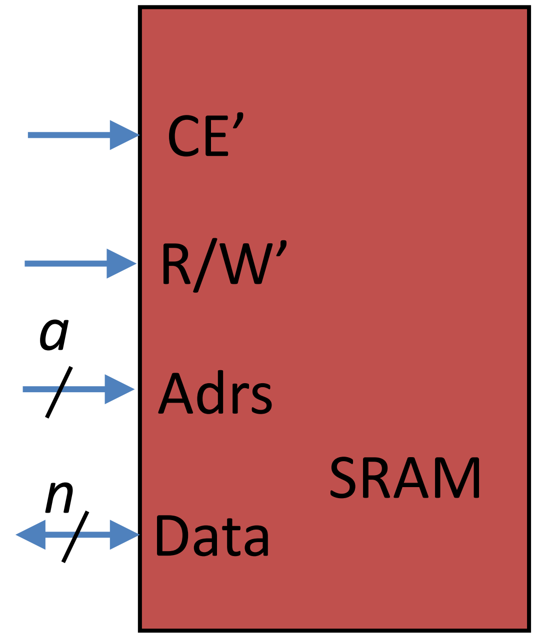


RAM (Random-Access Memory)

- SRAM (Static RAM)
  - Faster, usually used for caches
  - Easier to integrate with logic.
  - Higher power consumption.
- DRAM (Dynamic RAM)
  - Structurally simpler
    - Only1 transistor and 1 capacitor are required per bit, compared with 6 transistors used in SRAM
  - Can reach very high density

Typical Generic SRAM

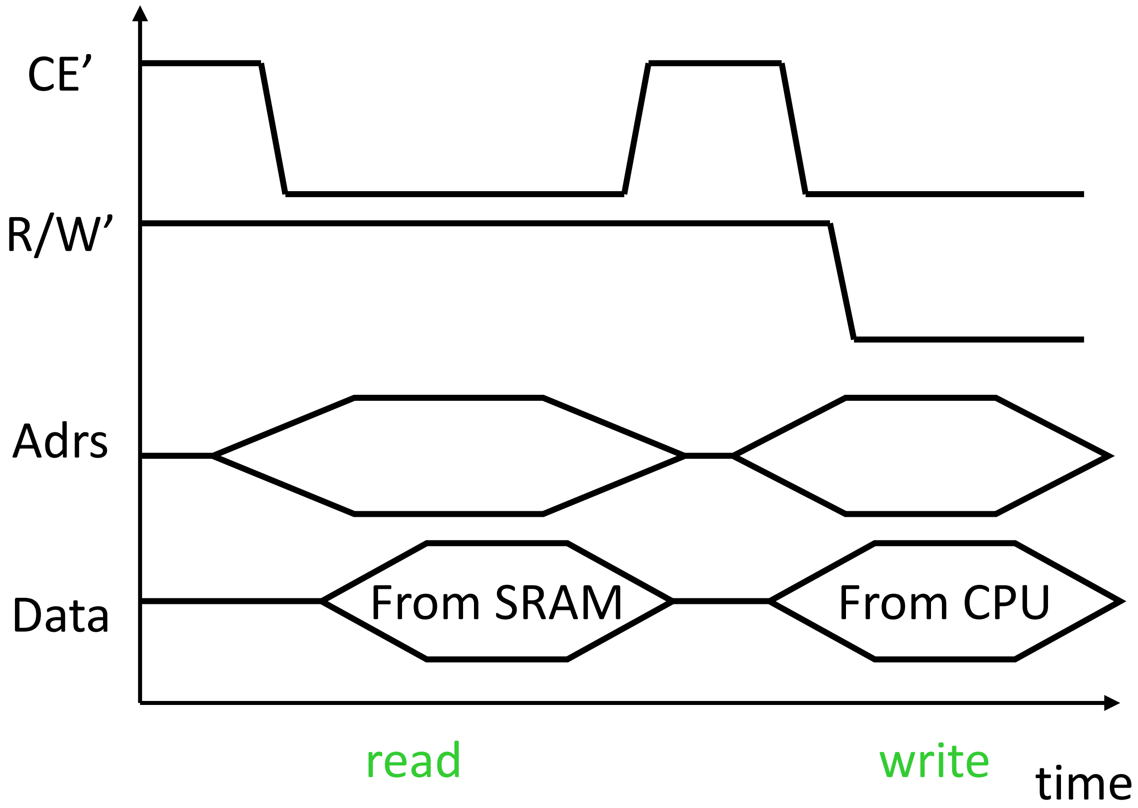
- CE’ is the chip enable input. CE’ = 1, data pins are disabled
- R/W’=1 means the current operation is read; R/W’=0 means write
- Adrs is the address for read or write
- Data is a bundle of signals for data transfer



Generic SRAM Timing

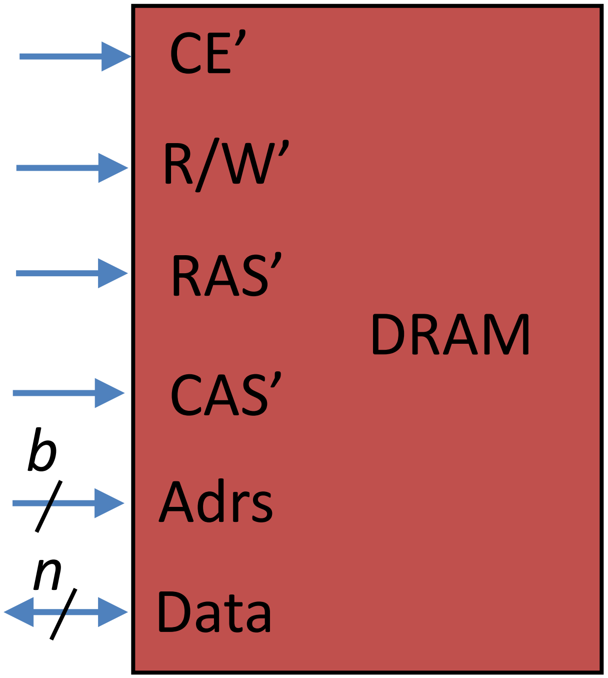
Read operation

- CE's is set to 0 to enable the chip with R/W'=1
- An address is put on the address lines
- After some delay, data appear on the data lines



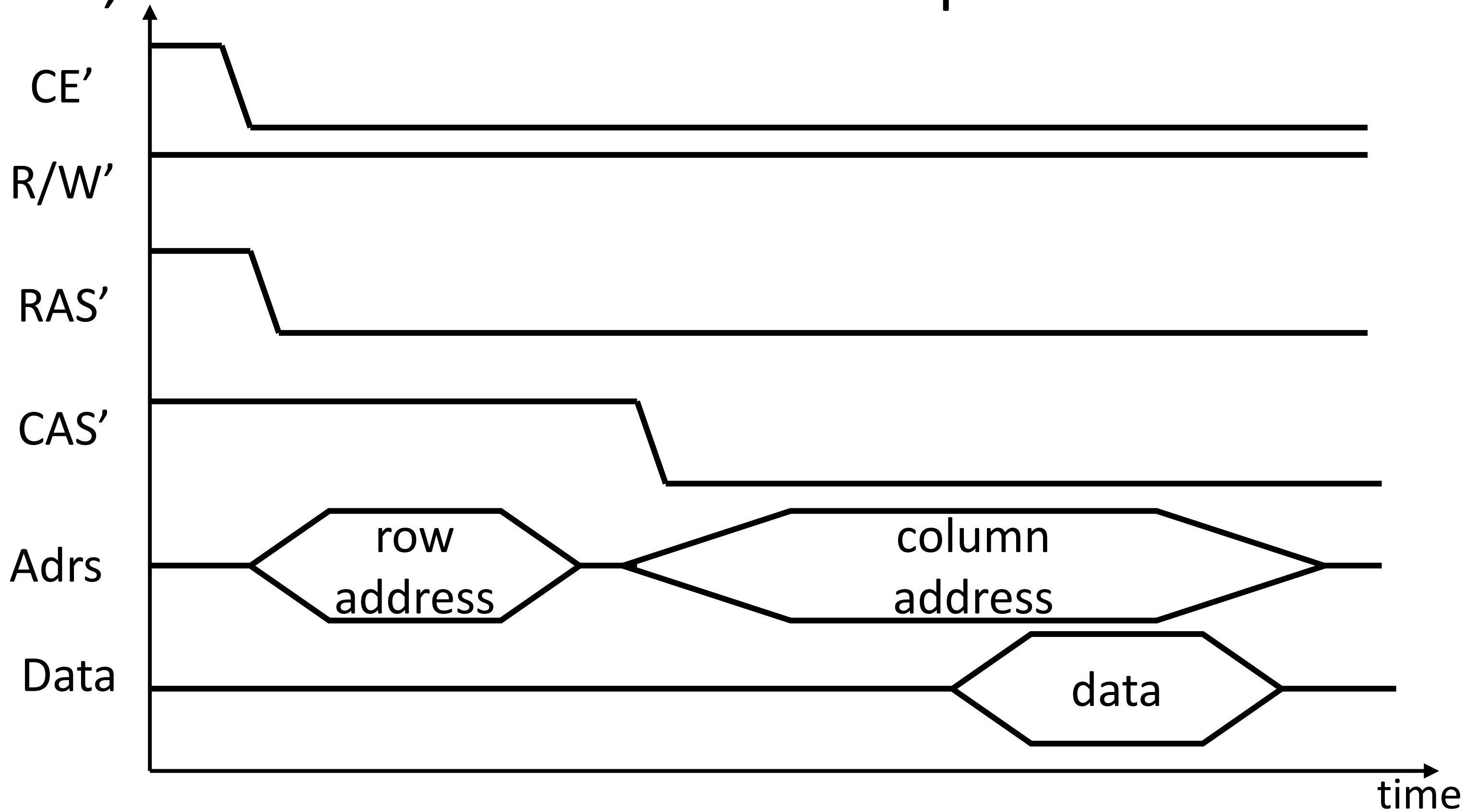
Generic DRAM Device

- The interface of DRAM is more complex
  - To minimize the # of pins
- Address line provides only half of the address
  - (RAS') row address select
  - (CAS') column address select



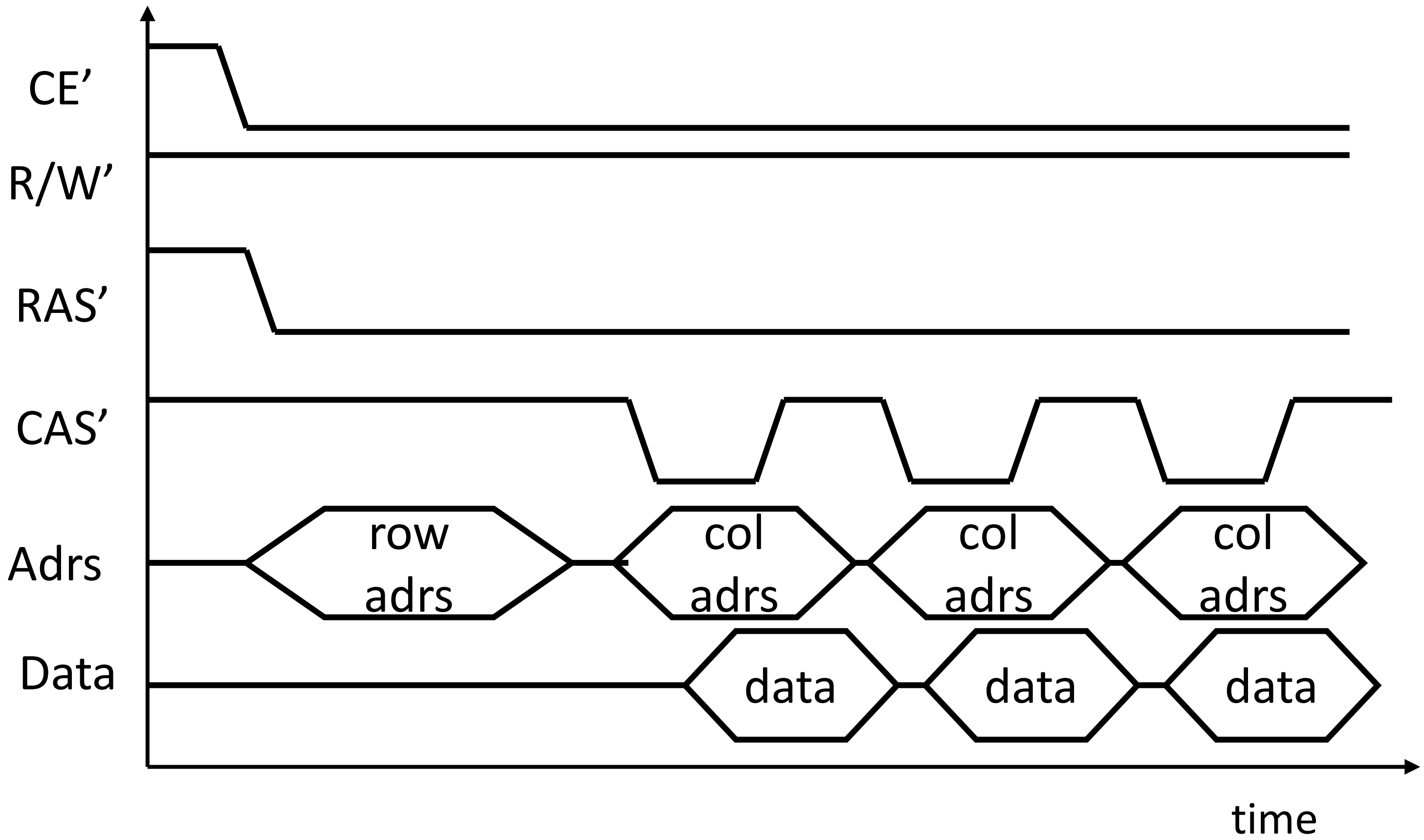
Generic DRAM Timing

- First, RAS’ is set to 0 and row part of address is on the address lines
- Next, CAS’ is set to 0 and column part of address is on



Page Mode Access of DRAM

- Slower than SRAM, how to improve DRAM performance?
- Supply one row address and many column addresses
  - Programs often access several locations in the same memory region



Read-Only Memory (ROM)

- Factory-programmed ROM
  - Not programmable in the lab
  - Also called Mask-programmed ROM
- Field-programmable ROM
  - Programmable once only
    - Cheapest but less flexible (e.g., Antifuse-programmable ROM)
  - Re-programmable ROM
    - UV-erasable PROM
    - Flash PROM
      - Modern form of electrically erasable PROM
      - Reprogrammed inside a typical system, such as Tmotes
      - Can be erased in blocks instead of a whole chip

Summary

- Caches
  - Cache mediates between CPU and memory system
  - Average memory access time
- Cache organizations
  - Direct-mapped cache
  - N-way set-associative
- Memory management: segment/page based
- Memory devices
  - RAM (Random Access Memory) vs. ROM (Read-Only Memory)
  - Memory device organization
  - SRAM (Static RAM) vs. DRAM (Dynamic RAM)

Homework Assignment 2

- On I/O operations
- 4% of your final grade
  - Individual work, no collaboration is allowed
- Two-week turnaround time
  - Due on 10/7 before class
  - Email your work to the TA