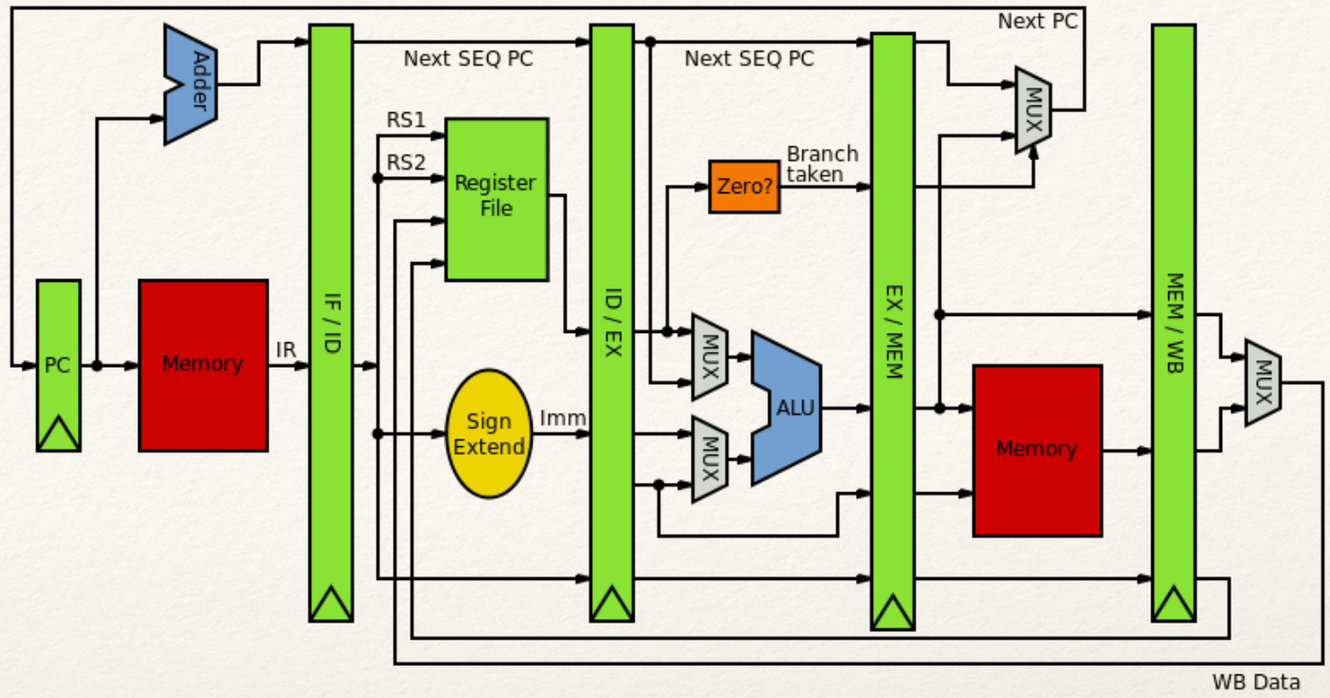


COMP2120 Computer Organization

Internal Memory

Chapter 5

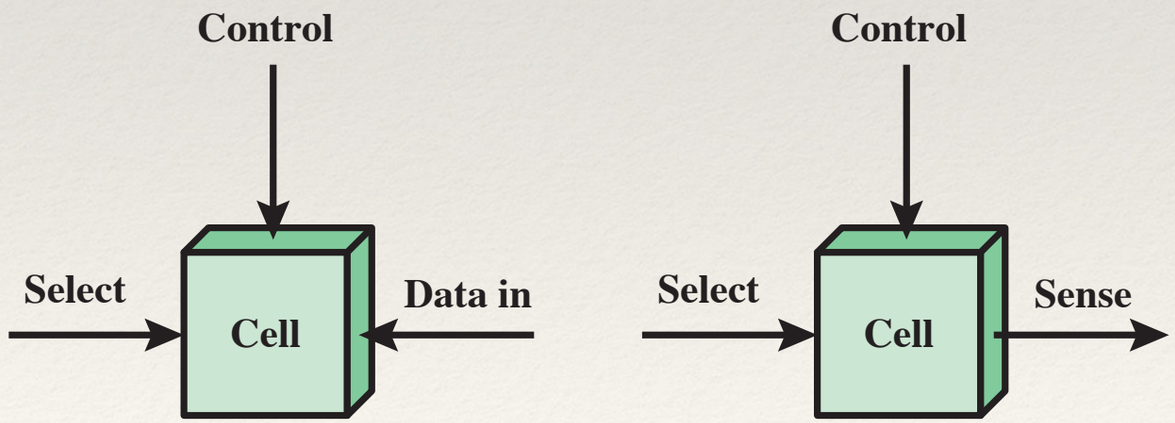
Dr. Ronald H.Y. Chung

THE UNIVERSITY OF HONG KONG

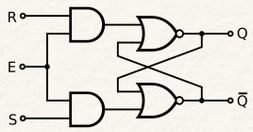
DEPARTMENT OF
COMPUTER SCIENCE

Memory Cell Operation

- The basic element of a semiconductor memory is the memory cell.
- All semiconductor memory cells share certain properties:
  - They exhibit two stable (or semistable) states, which can be used to represent binary 1 and 0.
  - They are capable of being written into (at least once), to set the state.
  - They are capable of being read to sense the state.



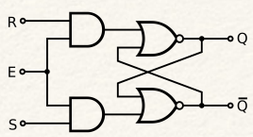
Memory Cell Operation



Types of Memory

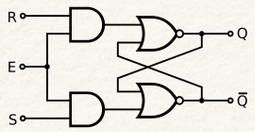
- The most common memory is referred to as random-access memory (RAM)
- One distinguishing characteristic of memory that is designated as RAM is that it is possible both to read data from the memory and to write new data into the memory easily and rapidly
- Another distinguishing characteristic of RAM is that it is volatile
- If the power is interrupted, then the data are lost.

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		



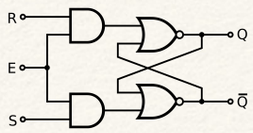
SRAM vs DRAM

- Both volatile
  - Power must be continuously supplied to the memory to preserve the bit values
- Dynamic cell
  - Simpler to build, smaller
  - More dense (smaller cells = more cells per unit area)
  - Less expensive
  - Requires the supporting refresh circuitry
  - Tend to be favored for large memory requirements
  - Used for main memory
- Static
  - Faster
  - Used for cache memory (both on and off chip)



Read Only Memory (ROM)

- Contains a permanent pattern of data that cannot be changed or added to
- No power source is required to maintain the bit values in memory
- Data or program is permanently in main memory and never needs to be loaded from a secondary storage device
- Data is actually wired into the chip as part of the fabrication process
  - Disadvantages of this:
    - No room for error, if one bit is wrong the whole batch of ROMs must be thrown out
    - Data insertion step includes a relatively large fixed cost
- Programmable ROM (PROM)
  - Less expensive alternative
  - Nonvolatile and may be written into only once
  - Writing process is performed electrically and may be performed by supplier or customer at a time later than the original chip fabrication
  - Special equipment is required for the writing process
  - Provides flexibility and convenience
  - Attractive for high volume production runs



Read-Mostly Memory

EPROM

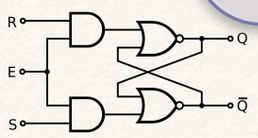
- Erasable programmable read-only memory
- Erasure process can be performed repeatedly
- More expensive than PROM but it has the advantage of the multiple update capability

EEPROM

- Electrically erasable programmable read-only memory
- Can be written into at any time without erasing prior contents
- Combines the advantage of non-volatility with the flexibility of being updatable in place
- More expensive than EPROM

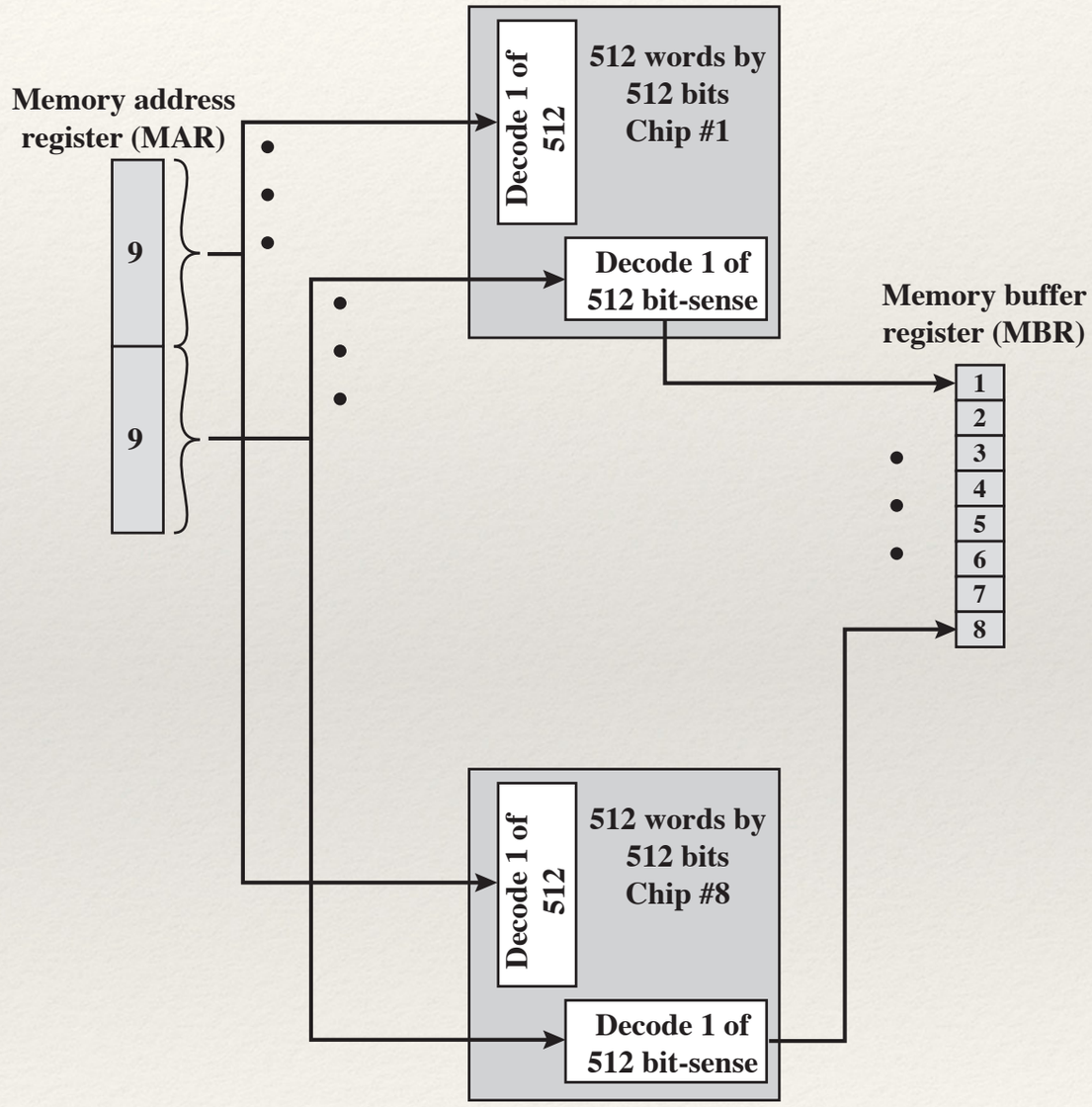
Flash Memory

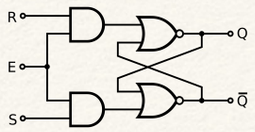
- Intermediate between EPROM and EEPROM in both cost and functionality
- Uses an electrical erasing technology, does not provide byte-level erasure
- Microchip is organized so that a section of memory cells are erased in a single action or “flash”



256-KByte Memory Organisation

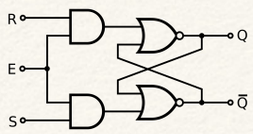
- If a RAM chip contains only 1 bit per word, then clearly we will need at least a number of chips equal to the number of bits per word
- As an example, the figure on the right shows how a memory module consisting of 256K 8-bit words could be organised
  - For 256K words, an 18-bit address is needed and is supplied to the module from some external source (e.g., the address lines of a bus to which the module is attached)
  - The address is presented to eight 256K * 1-bit chips, each of which provides the input/ output of 1 bit.





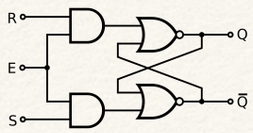
Location and Capacity of Memory

- Location
  - Refers to whether memory is internal and external to the computer
  - Internal memory is often equated with ***main memory***
  - Processor requires its own local memory, in the form of ***registers***
  - Cache is another form of internal memory
  - External memory consists of peripheral storage devices that are accessible to the processor via I/O controllers
- Capacity
  - Memory is typically expressed in terms of bytes (MBytes/GBytes)
  - CPU addresses memory by word (32 bits or 64 bits)
    - Usually memory is byte addressable, i.e. each address is 1 byte
    - Hence 1 32-bit word will occupy 4 addresses



Unit of Transfer of Memory Systems

- Unit of transfer
  - For internal memory the unit of transfer is equal to the number of electrical lines into and out of the memory module
  - Memory is a bottle neck for faster performance, because it is much slower than the CPU
  - CPU reads memory word by word
- Nowadays, CPU do not read directly from main memory, but from cache memory instead
  - Unit of transfer between main memory and cache memory will be based on block (e.g. 4KByte memory per block)



Access Method

Sequential access

- Memory is organized into units of data called records
- Access must be made in a specific linear sequence
- Access time is variable

Direct access

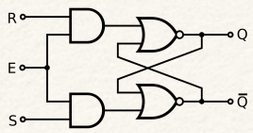
- Involves a shared read-write mechanism
- Individual blocks or records have a unique address based on physical location
- Access time is variable

Random access

- Each addressable location in memory has a unique, physically wired-in addressing mechanism
- The time to access a given location is independent of the sequence of prior accesses and is constant
- Any location can be selected at random and directly addressed and accessed
- Main memory and some cache systems are random access

Associative

- A word is retrieved based on a portion of its contents rather than its address
- Each location has its own addressing mechanism and retrieval time is constant independent of location or prior access patterns
- Cache memories may employ associative access



Performance

Three performance parameters are used:

Access time (latency)

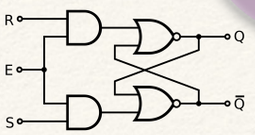
- For random-access memory it is the time it takes to perform a read or write operation
- For non-random-access memory it is the time it takes to position the read-write mechanism at the desired location

Memory cycle time

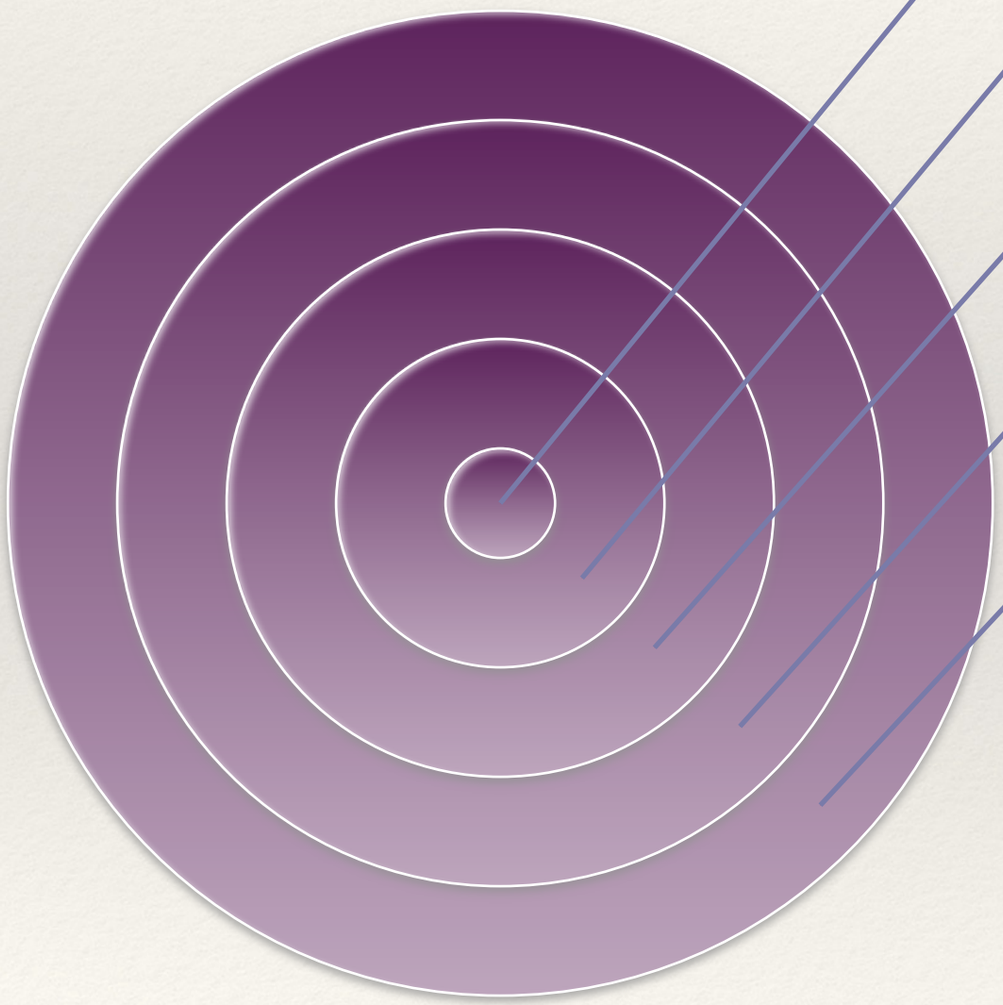
- Access time plus any additional time required before second access can commence
- Additional time may be required for transients to die out on signal lines or to regenerate data if they are read destructively
- Concerned with the system bus, not the processor

Transfer rate

- The rate at which data can be transferred into or out of a memory unit
- For random-access memory it is equal to 1/(cycle time)



Interleaved Memory



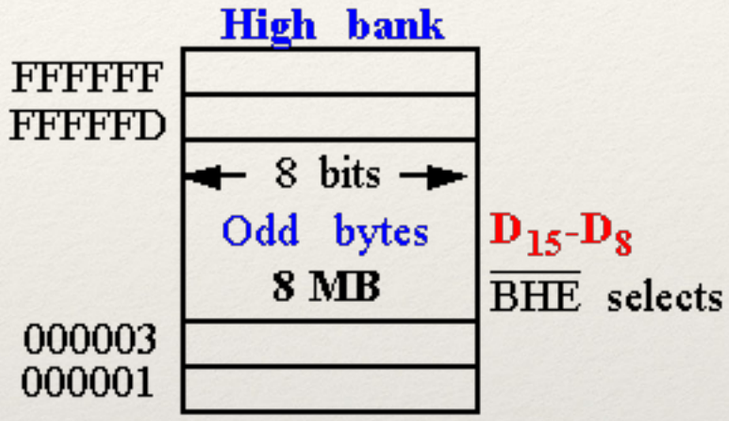
Composed of a collection of DRAM chips

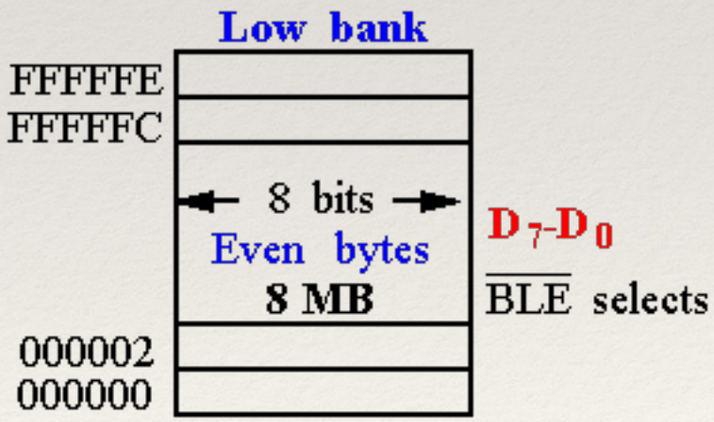
Grouped together to form a *memory bank*

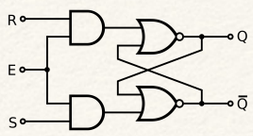
Each bank is independently able to service a memory read or write request

K banks can service *K* requests simultaneously, increasing memory read or write rates by a factor of *K*

If consecutive words of memory are stored in different banks, the transfer of a block of memory is speeded up

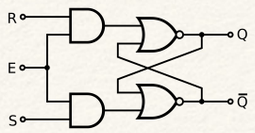






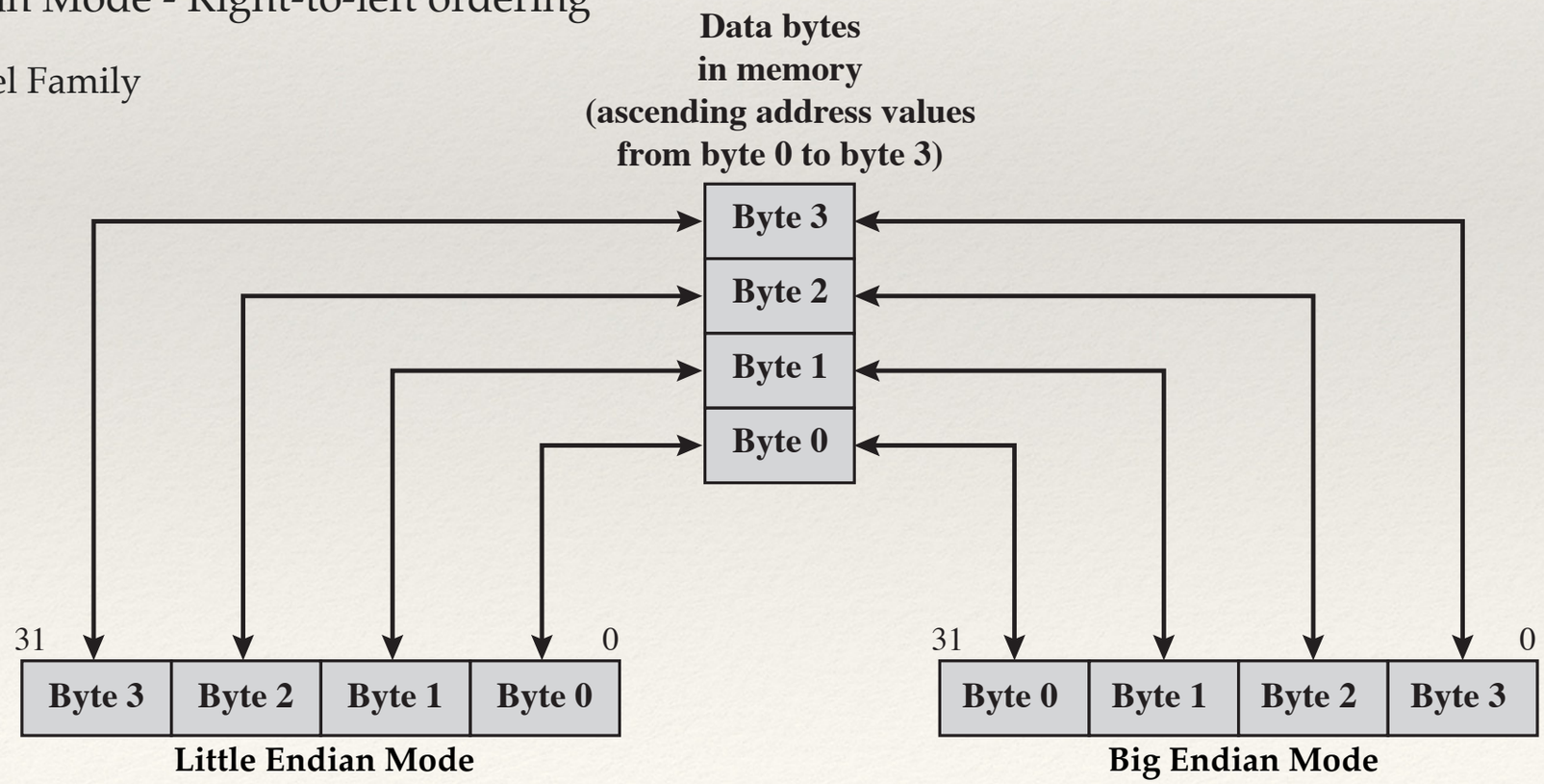
Physical Type and Characteristics

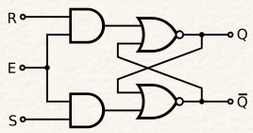
- The most common forms are:
  - Semiconductor memory
  - Magnetic surface memory
  - Optical
  - Magneto-optical
- Several physical characteristics of data storage are important:
  - Volatile memory
    - Information decays naturally or is lost when electrical power is switched off
  - Nonvolatile memory
    - Once recorded, information remains without deterioration until deliberately changed
    - No electrical power is needed to retain information
  - Magnetic-surface memories
    - Are nonvolatile
  - Semiconductor memory
    - May be either volatile or nonvolatile
  - Nonerasable memory
    - Cannot be altered, except by destroying the storage unit
    - Semiconductor memory of this type is known as read-only memory (ROM)
- For random-access memory the organization is a key design issue
  - Organization refers to the physical arrangement of bits to form words



Memory Organization

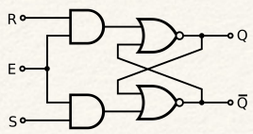
- Memory Byte Ordering for multiple byte data
  - e.g. Integer, floating point numbers
  - The bytes in a word can be numbered from left-to-right or right-to-left
  - Big Endian Mode - Left-to-right ordering (numbering begins at the “big”
    - e.g. IBM mainframes and SPARC
  - Little Endian Mode - Right-to-left ordering
    - e.g. Intel Family





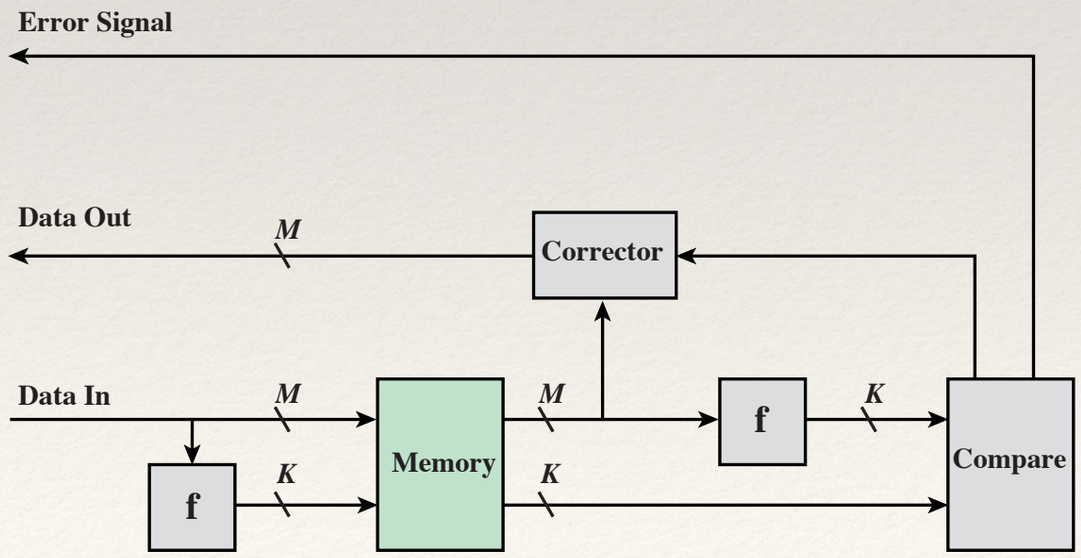
Error Detection & Correction

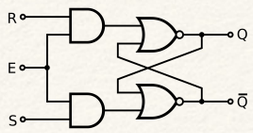
- Hard Failure
  - Permanent physical defect
  - Memory cell or cells affected cannot reliably store data but become stuck at 0 or 1 or switch erratically between 0 and 1
  - Can be caused by:
    - Harsh environmental abuse
    - Manufacturing defects
    - Wear
- Soft Error
  - Random, non-destructive event that alters the contents of one or more memory cells
  - No permanent damage to memory
  - Can be caused by:
    - Power supply problems (e.g. voltage spikes during lightning)
    - Alpha particles



Error-Correcting Code Function

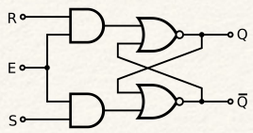
- When data are to be written into memory, a calculation, depicted as a function *f*, is performed on the data to produce a code. Both the code and the data are stored.
  - if an *M*-bit word of data is to be stored and the code is of length *K* bits, then the
  - actual size of the stored word is *M* + *K* bits.
- When the previously stored word is read out, the code is used to detect and possibly correct errors
  - No errors are detected
    - The fetched data bits are sent out.
  - An error is detected, and it is possible to correct the error
    - The data bits plus error correction bits are fed into a corrector, which produces a corrected set of *M* bits to be sent out.
  - • An error is detected, but it is not possible to correct it
    - This condition is reported





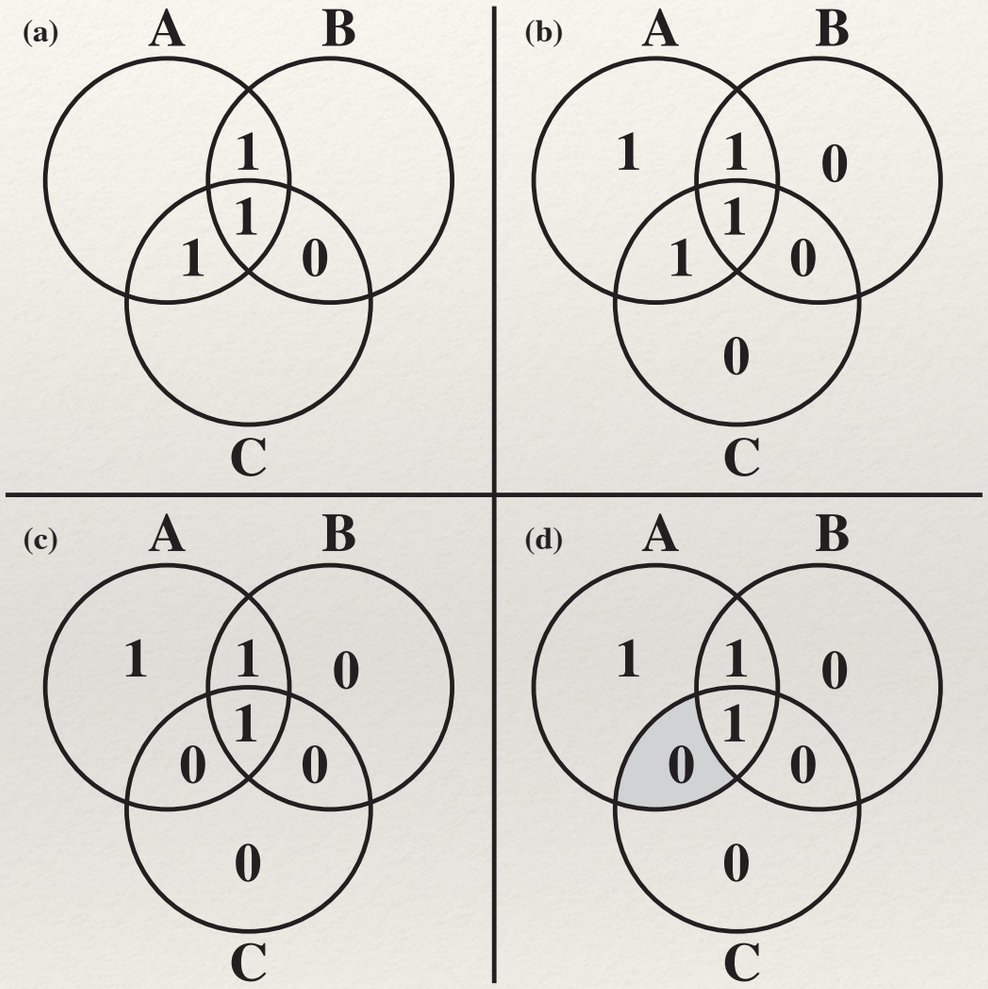
Error Detection

- A single **parity** bit is attached to the bit pattern
- The parity bit is chosen such that the number of 1's in the bit pattern is even (***even parity***) or odd (***odd parity***)
- Example: (parity in parenthesis)
  - Even parity
    - 11010011(1)
  - Odd parity
    - 11010011(0)
- It can detect 1-bit error, because if one of the bit is wrong, (i.e. changing from 1 to 0 or vice versa), the number of 1's will change from odd to even or from even to odd
- Single parity allows error detection (1 bit error) only, to correct an error, more bits are required
  - e.g. Hamming code

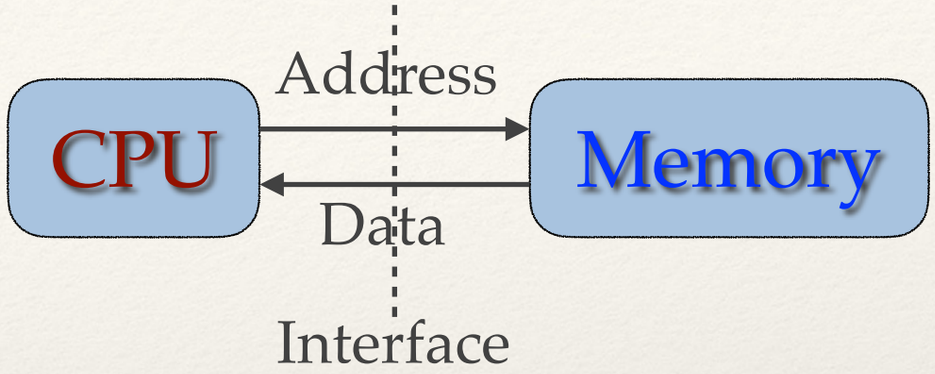


Hamming Code Example

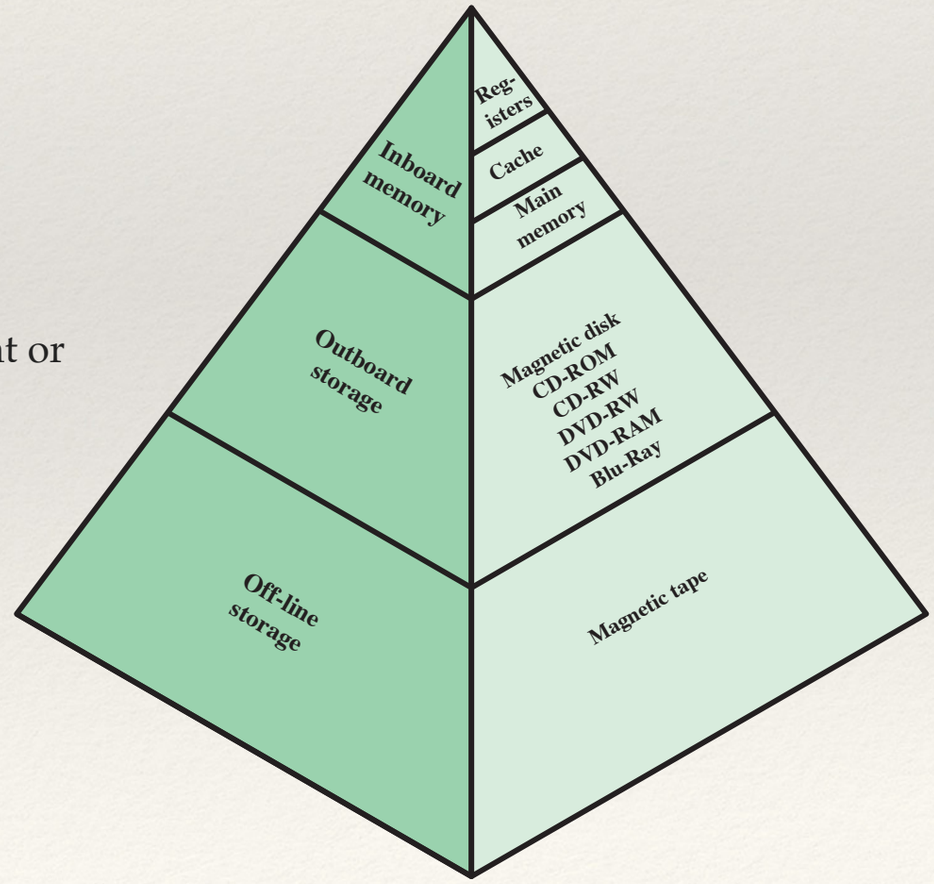
- Assume $M = 4$, $K = 3$
- The 4 data bits are assigned into the inner compartments
- The remaining compartments are filled with what are called parity bits (assume even parity)
- If an error changes one of the data bits
  - For example, the error bit induces parity discrepancies in A & C
  - Observing that only one of the seven compartments is in A and C but not B, and thus the error can therefore be corrected by changing that bit.

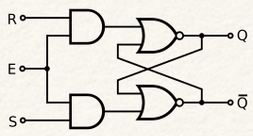


Memory Hierarchy



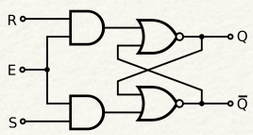
- Memory Hierarchy
  - The CPU reads memory by giving address and wait for data
  - It does not care what happens on the other side of the interface
    - Cache Memory, Virtual Memory (e.g. Hard Disk) can be added
- Design constraints on a computer's memory can be summed up by three questions:
  - How much, how fast, how expensive
- There is a trade-off among capacity, access time, and cost
  - Faster access time, greater cost per bit
  - Greater capacity, smaller cost per bit
  - Greater capacity, slower access time
- The way out of the memory dilemma is not to rely on a single memory component or technology, but to employ a memory hierarchy
  - From top to bottom:
    - Decreasing cost per bit
    - Increasing capacity
    - Increasing access time
    - Decreasing frequency of access of the memory by the processor





Principle of Locality

- Memory reference tends to be localised
  - Data in the vicinity of a referenced word are likely to be referenced in the near future
    - Code reference is local, obviously
    - Data reference is less local, but is still so, e.g. small arrays, and blocks of local variables in a subprogram
    - When data is references, it will tend to be referenced again in the near future
    - Thus, future access can be from higher level, e.g. cache memory, instead of lower level



Chapter 5 - Summary

- Semiconductor main memory
  - Organization
  - DRAM and SRAM
  - Types of ROM
  - Module organization
  - Interleaved memory
- Error correction
- Skipped Topics
  - Semiconductor memory chip logic (Subtopic under 5.1)
  - Semiconductor memory chip packaging (Subtopic under 5.1)
  - DDR DRAM (5.3)
  - Flash memory (5.4)
  - Newer nonvolatile solid-state memory technologies (5.5)