Memory Devices

EE2222 Computer Interfacing and Microprocessors

Partially based on
Memory Devices and Interfacing – (Chapter 9) by Dr. Costas Kyriacou and Dr. Konstantinos Tatas
Microprocessors and Interfacing by Dr. Saeid Nooshabadi

Basic Concepts

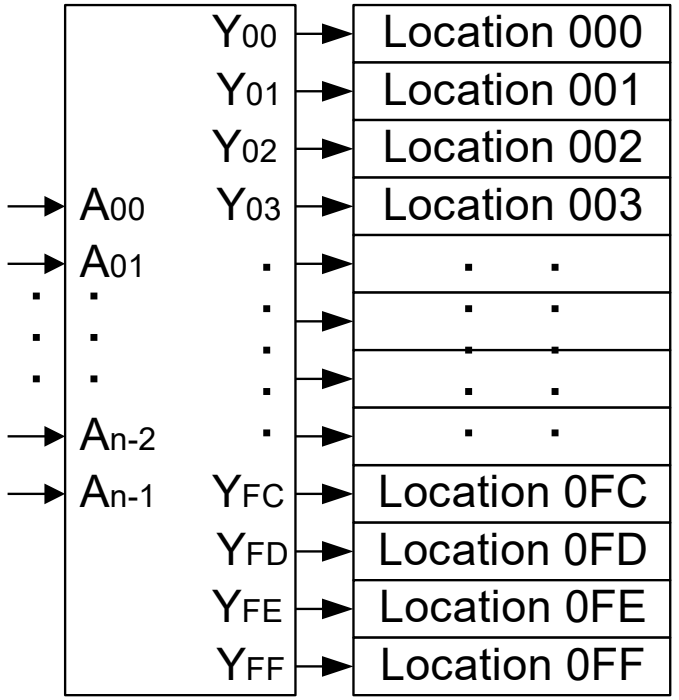
- A memory device can be viewed as a *single column table*.
  - Table index (row number) refers to the address of the memory.
  - Table entries refer to the memory contents or data.
  - Each table entry is referred as a memory location or as a word.
- Both the memory address and the memory contents are binary numbers, expressed in most cases in Hex format.
- The size of a memory device is specified as the number of memory locations X width or word size (in bits).
  - For example a 1K X 8 memory device has 1024 memory locations, with a width of 8 bits.
- A memory device or memory chip must have three types of lines or connections: **Address, Data, and Control**.

Memory Address		Memory Contents
Binary	Hex	
00-0000-0000	000	10011001
00-0000-0001	001	00111000
00-0000-0010	002	11001001
00-0000-0011	003	00111011
⋮	⋮	⋮
11-1111-1100	3FC	01101000
11-1111-1101	3FD	10111001
11-1111-1110	3FE	00110100
11-1111-1111	3FF	00011000

1024 X 8 (or 1KX8) Memory

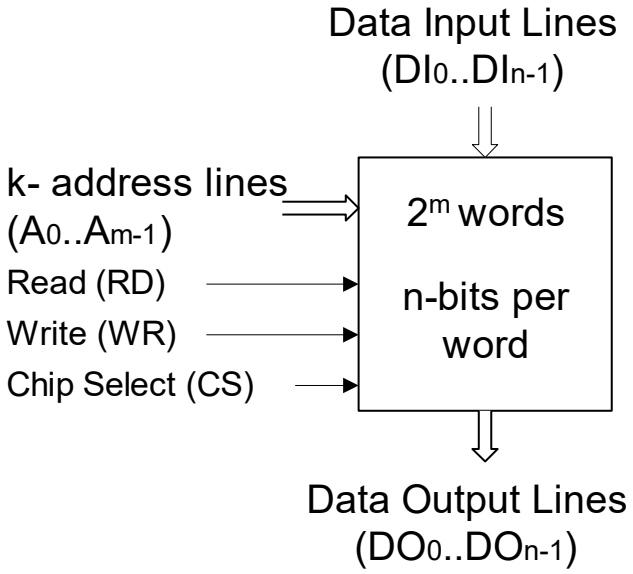
Address Lines

- The input lines that select a memory location within the memory device.
- Decoders are used, inside the memory chip, to select a specific location
- The number of address pins on a memory chip specifies the number of memory locations.
  - If a memory chip has 13 address pins (A0..A12), then it has: 213 = 23 X 210 = 8K locations.
  - If a memory chip has 4K locations, then it should have N pins: 2N = 4K = 22 X 210 = 212 $\rightarrow$ N=12 address pins (A0..A11)

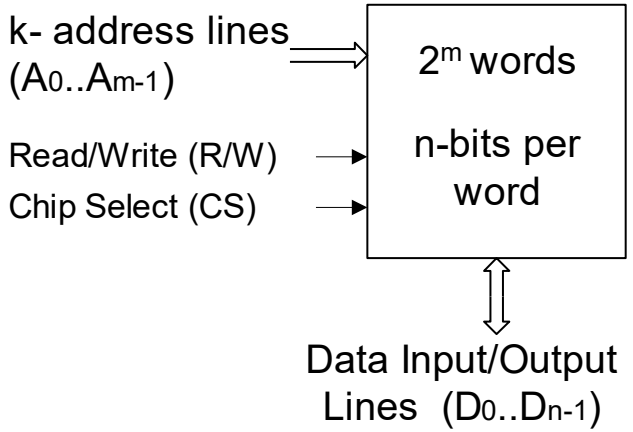


Data Lines

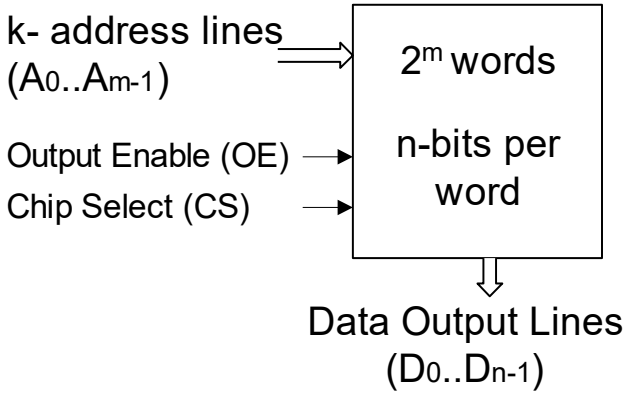
- All memory devices have a set of data output pins (for ROM devices), or input/output pins (for RAM devices).
  - Most RAM chips have common bi-directional I/O connections.
  - Most memory devices have 1, 8 or 16 data lines.



(2m X n) RAM with separate I/P and O/P Data lines



(2m X n) RAM with common I/P and O/P Data lines



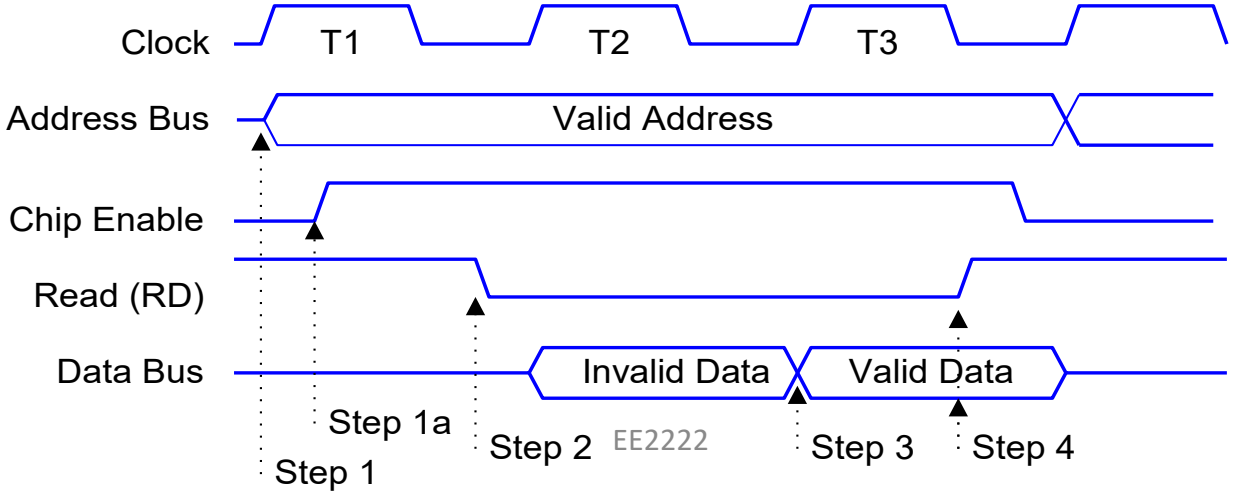
(2m X n) ROM with only O/P Data lines

Control Lines

- **Enable Connections:**
  - All memory devices have at least one **Chip Select (CS)** or **Chip Enable (CE)** input, used to select or enable the memory device.
    - If a device is not selected or enabled then no data can be read from, or written into it.
    - The CS or CE input is usually controlled by the microprocessor through the higher address lines via an address decoding circuit.
- **Control Connections:**
  - RAM chips have two control input signals that specify the type of memory operation: the **Read (RD)** and the **Write (WR)** signals.
    - Some RAM chips have a common Read/Write (R/W) signal.
  - ROM chips can perform only memory read operations, thus there is no need for a Write (WR) signal.
    - In most real ROM devices the Read signal is called the **Output Enable (OE)** signal.

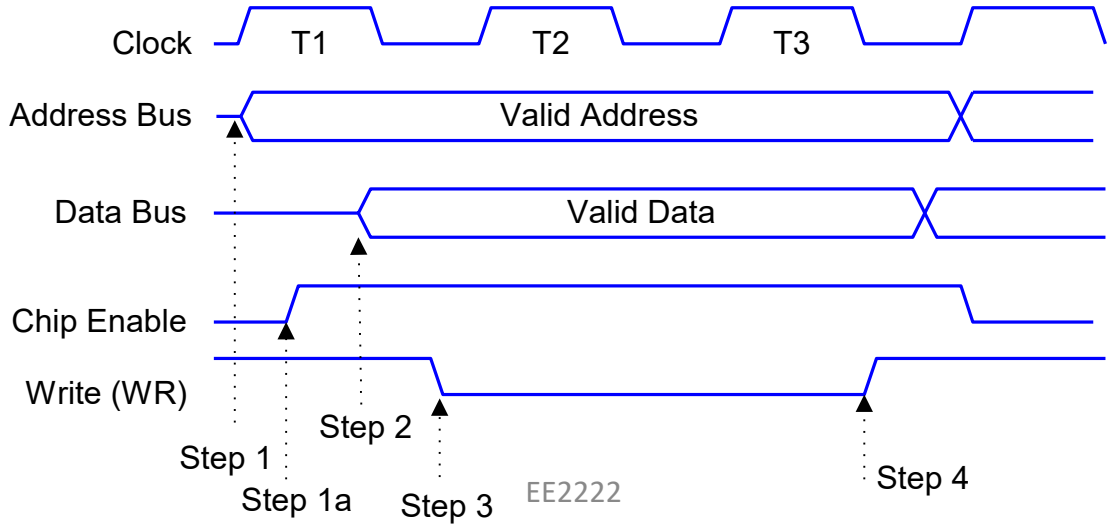
Memory Read Operations

- A memory read operation is carried out in the following steps:
  - The processor loads on the Address bus the address of the memory location to be read **(Step 1)**.
    - Some of the address lines select the memory devices that owns the memory location to be read **(Step 1a)**, while the rest point to the required memory location within the memory device.
  - The processor activates the Read (RD) signal **(Step 2)**.
  - The selected memory device loads on the data bus the content of the memory location specified by the address bus **(Step 3)**.
  - The processor reads the data from the data bus, and resets the RD signal **(Step 4)**.



Memory Write Operations

- A memory write operation is carried out in the following steps:
  - The processor loads on the Address bus the address of the memory location **(Step 1)**.
    - Some of the address lines select the memory devices that owns the memory location to be written **(Step 1a)**, while the rest point to the required memory location within the memory device.
  - The processor loads on the data bus the data to be written **(Step 2)**.
  - The processor activates the Write (WR) signal **(Step 3)**.
  - The data at the data bus is stored in the memory location specified by the address bus **(Step 4)**.



Types of Semiconductor Memory Devices

- **Read Only Memory (ROM)**
- A memory device that maintains its data permanently (or until the device is reprogrammed).
  - Non-volatile: It maintains its data even without power supply.
- Used to store
  - Programs such as the BIOS.
  - Data such as look tables
    - e.g. the bit pattern of the characters in a dot matrix printer.
- A ROM device can be
  - Masked ROM (Programmed by the manufacturer)
  - Programmable ROM (can be program-erased-reprogrammed many times

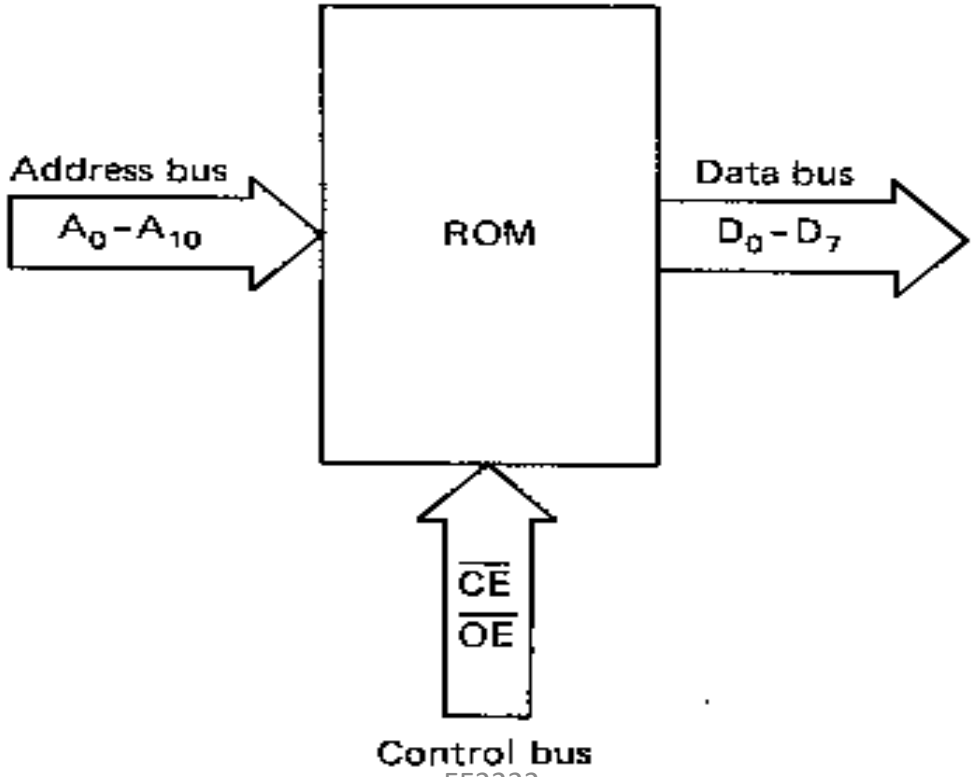
- **Random Access Memory (RAM)**
- A memory device that can be read and written.
  - Volatile: It looses its data when the power supply is switched-off
  - When the supply is switched-on it contains random data
- Used to store
  - User programs that are loaded from a secondary memory (disk)
  - Temporary data used by programs such as variables and arrays.
- A RAM device can be
  - Static
  - dynamic

Exercise

- There is a BIOS in your computer, what kind of memory is it?

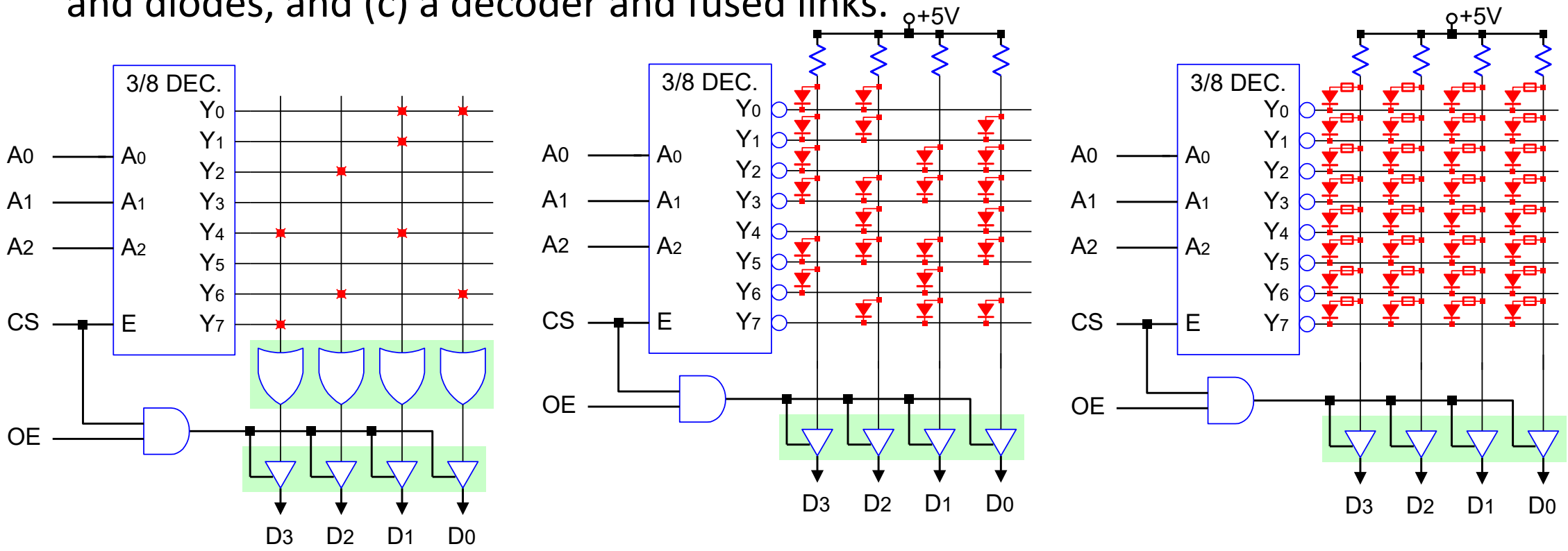
Block diagram of a ROM

- ROM interface – address input, data output, /CE – chip enable, /OE – output enable (for READ operation)



A Read Only Memory Example

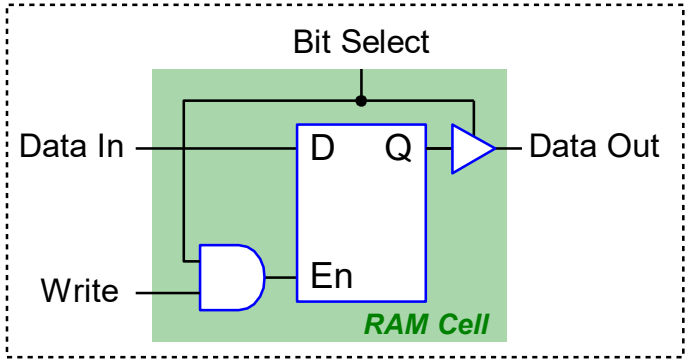
- Implementation of an 8X4 ROM using (a) a decoder and OR-gates, (b) a decoder and diodes, and (c) a decoder and fused links.



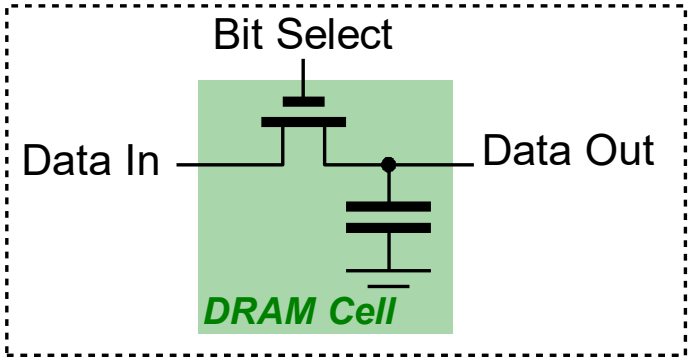
Address	000	001	010	011	100	101	110	111
Data	0011	0010	0100	0011	1010	0000	0101	1000

RAM Cells

- **Static RAM (SRAM):**
  - The basic element of a static RAM cell is the D-Latch.
  - Data remains stored in the cell until it is intentionally modified.
  - SRAM is fast (Access time: 1ns).
  - SRAM needs more space on the semiconductor chip than DRAM.
    - SRAM more expensive than DRAM
    - SRAM needs more space than DRAM
  - SRAM consumes power only when accessed.
  - SRAM is used as a Cache

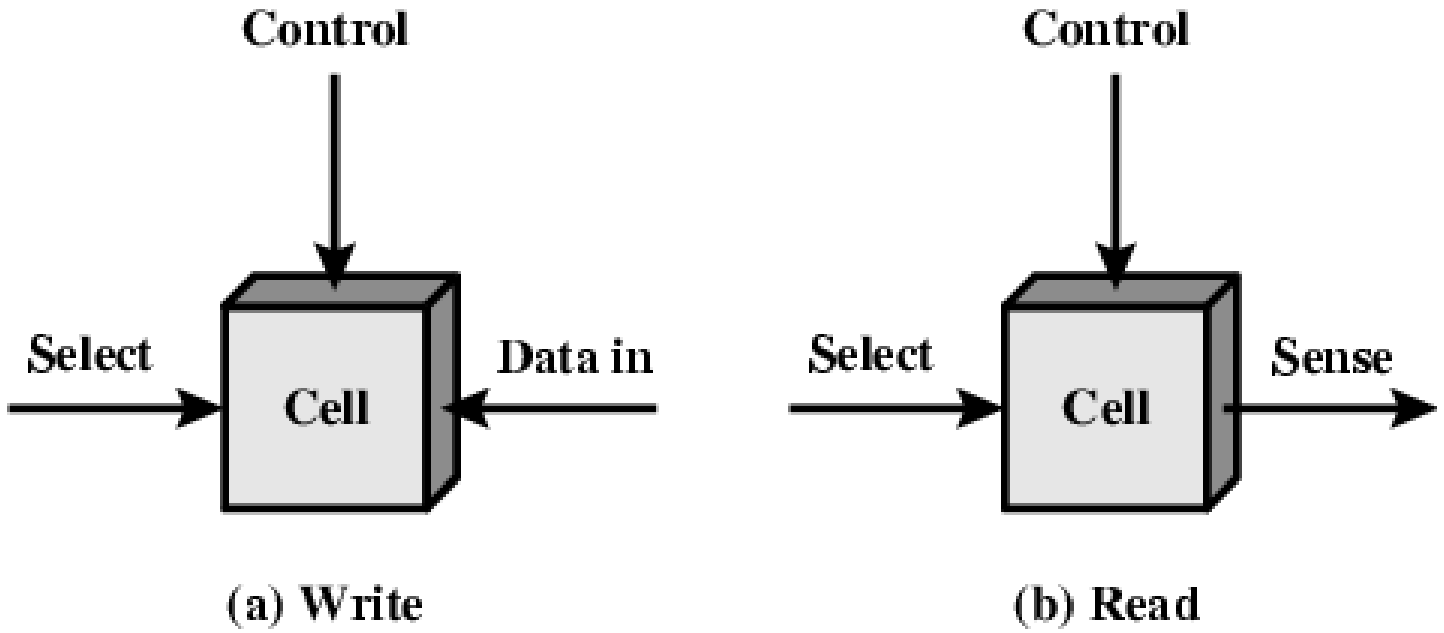


- **Dynamic RAM (DRAM):**
  - DRAM stores data in the form of electric charges in capacitors.
  - Charges leak out, thus need to refresh data every few ms.
  - DRAM is slow (Access time: 60ns).
  - DRAM needs less space on the semiconductor chip than SRAM.
    - DRAM less expensive than SRAM
    - DRAM needs less space than SRAM
  - DRAM needs to be refreshed
  - DRAM is used as the main memory



Memory Cell Operation

- Select terminal select the cell for read and write
- The control terminal indicated read or write

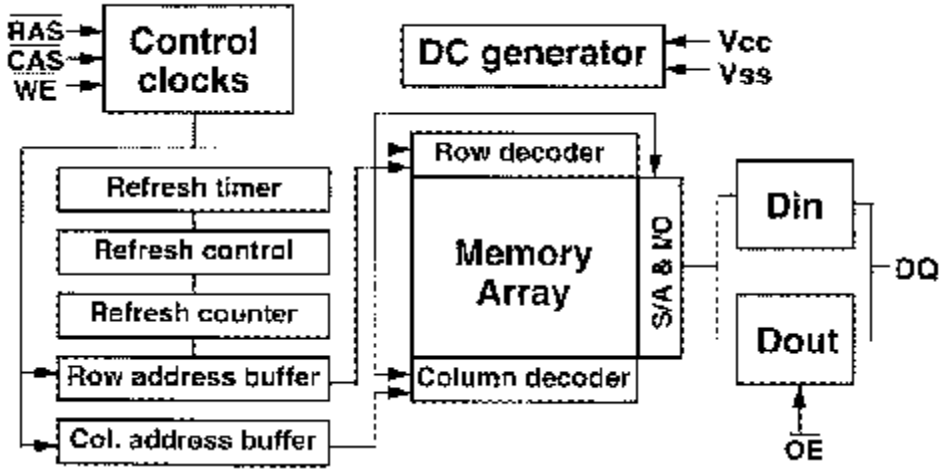


Static RAM (SRAM)

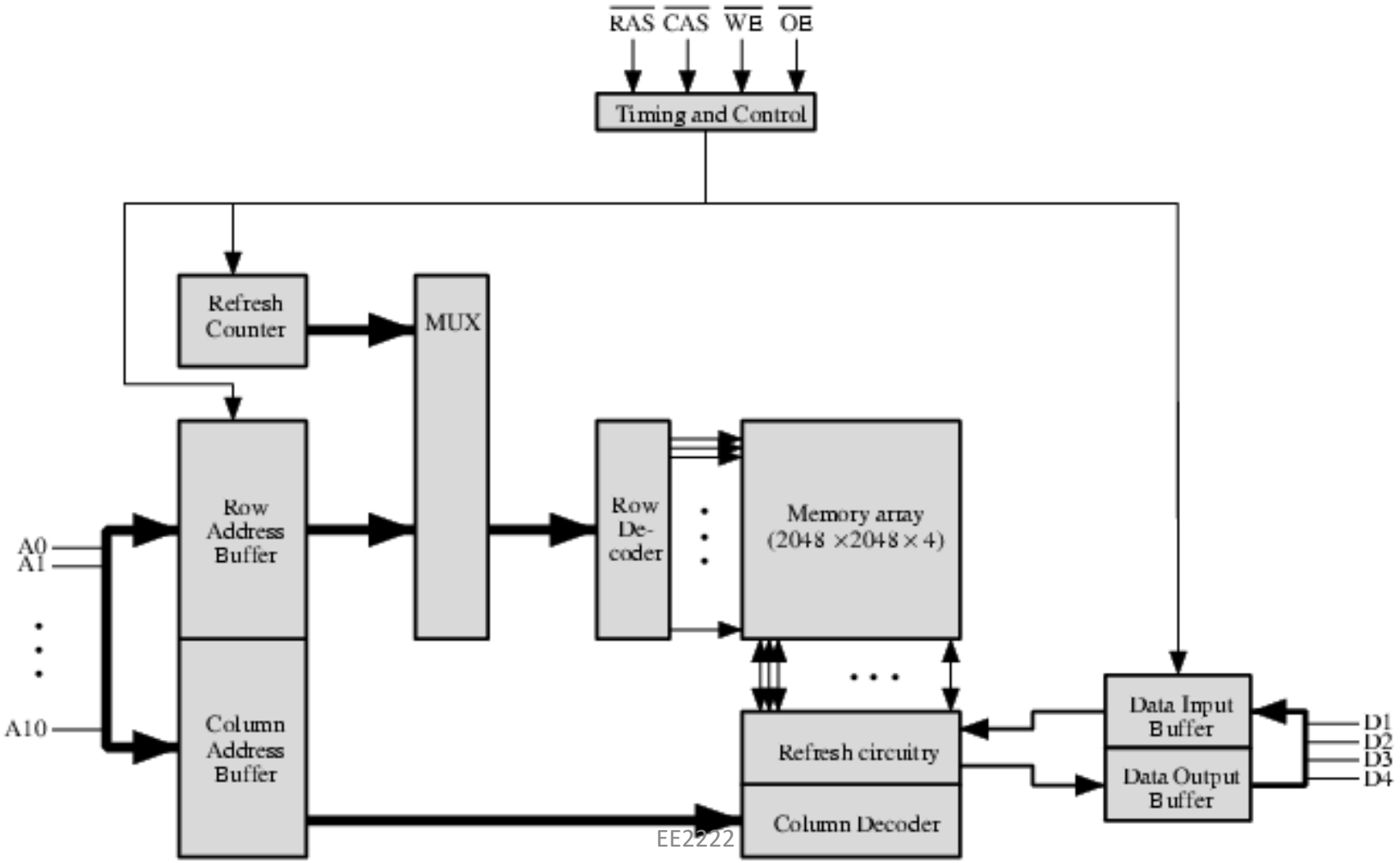
- Bits stored as on/off switches (e.g. flip flops)
- No charges to leak
- No refreshing needed when powered
- More complex construction
- More expensive
- Does not need refresh circuits
- Faster
- e.g. Cache

Dynamic RAM (DRAM)

- DRAM requires refreshing every 2 to 4 ms .
- Refreshing occurs automatically during a read or write.
- Internal circuitry takes care of refreshing cells that are not accessed over this interval.
  - For a 256K X 1 DRAM with 256 rows, a refresh must occur every 15.6us (4ms/256).
  - For the 8086, a read or write occurs every 800ns .
  - This allows 19 memory reads/writes per refresh or 5% of the time.
- DRAM technologies
  - EDO DRAM, SDRAM, DRDRAM, DDR DRAM
- A DRAM CONTROLLER is required for using DRAM



Typical 16 Mb DRAM (4M x 4)



Synchronous Dynamic RAM (SDRAM)

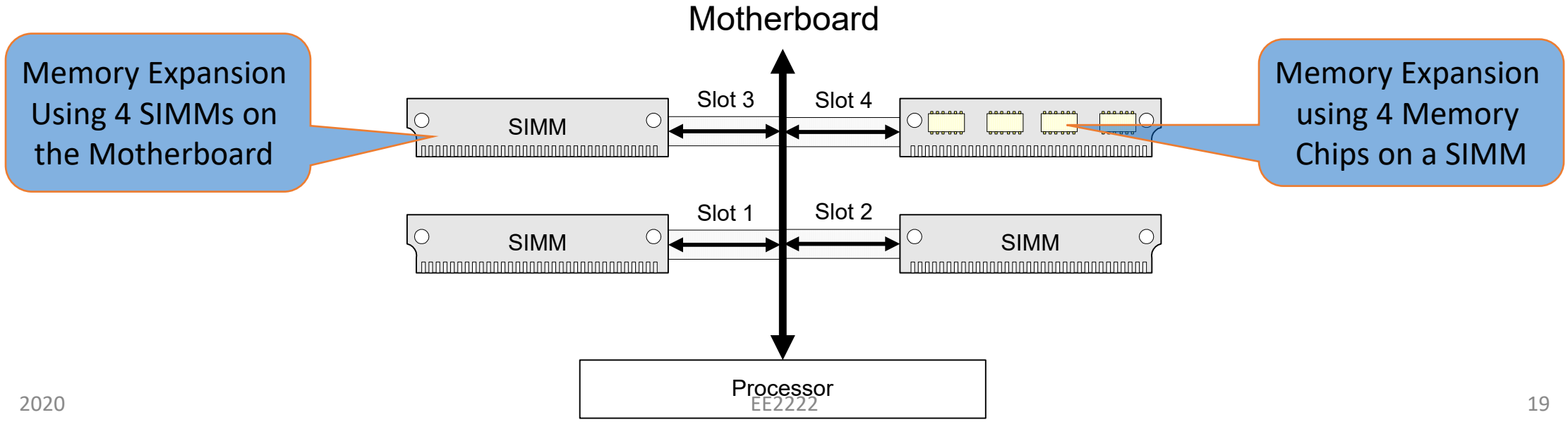
- In a SDRAM, the control signals are synchronized with the system bus clock and therefore with the microprocessor
- It allows *pipelined* read/write operations

Double Data Rate (DDR) DRAM

- An SDRAM type of memory, where data are transferred on both the rising and the falling clock edge, effectively doubling the transfer rate without increasing the clock frequency
- DDR-200 means a transfer rate of 200 million transfers per second, at a clock rate of 100 MHz
- DDR1 up to 400 MHz
- DDR2 and later standards allow higher clock frequencies

Semiconductor Memory Expansion

- The size of memory devices is usually less than the memory requirements of a computer system.
- In all computers, more than one memory devices are combined together to form the main memory of the system.



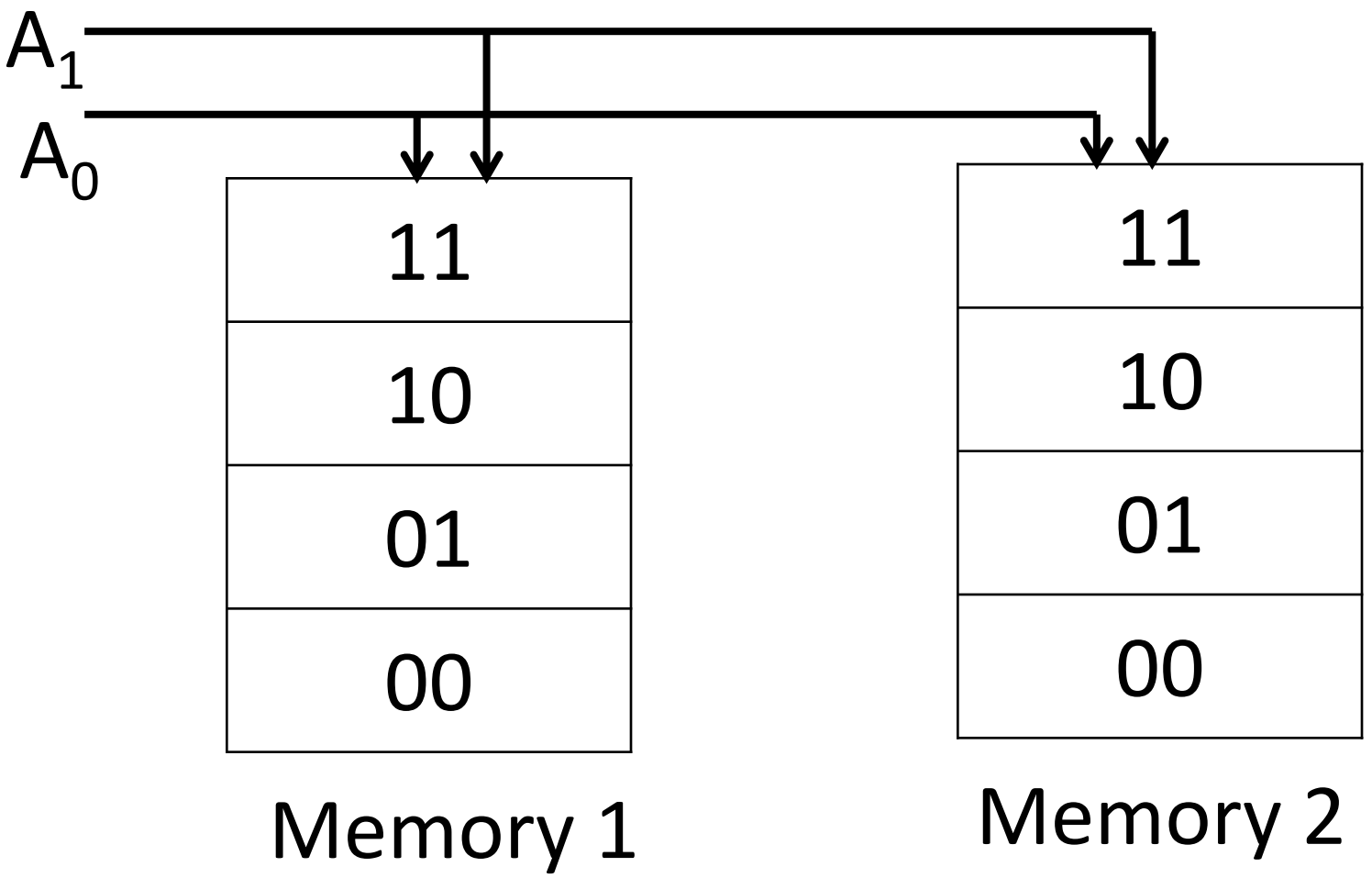
Address Decoding

- Address decoding is required because many memory chips are used by a computer system
- At each memory read/write only a number of chips is used
- Decoding mechanism is used to guarantee that the proper chips are selected
- Certainly, capacity of modern memory device is large (GB!!) so decoding may not be necessary but if you consider the development of SSD then decoding will become necessary if a SSD is 250G then you still need to use more than 1 memory device

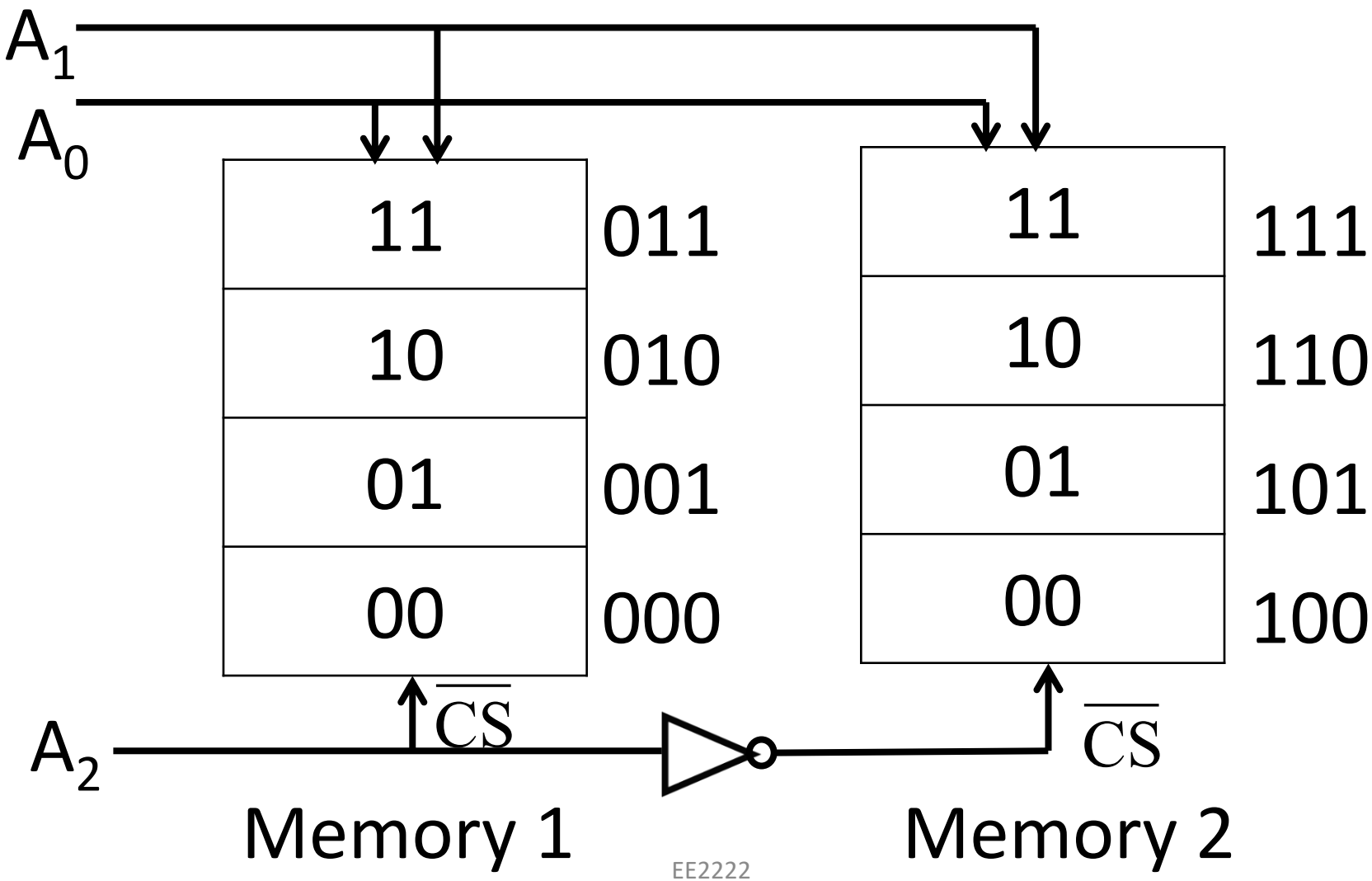
Address decoding

- To design, first determine the number of chips required
- Then determine how many address lines are needed for the decoding purpose
- Example if 4 chips are used then you need 2 address lines for decoding

Interface with two memory chips



Interface with two memory chips



Errors and Error Detection and Correction

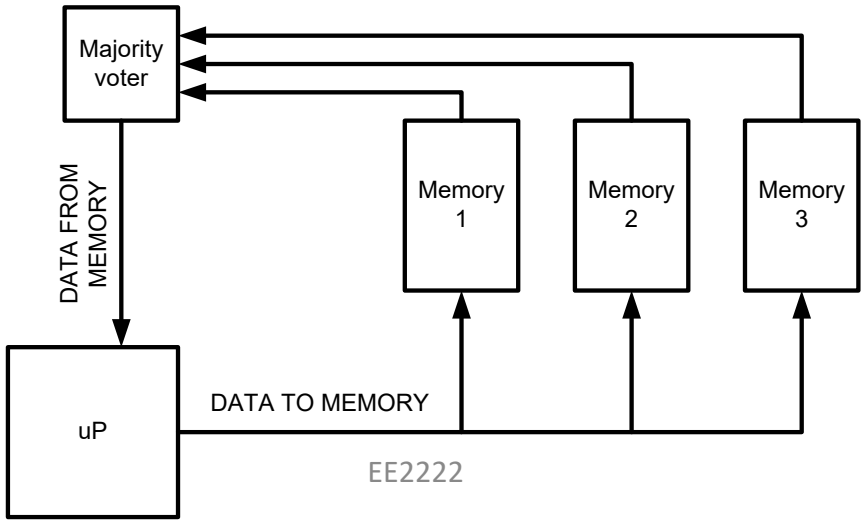
- Electrical or magnetic interference inside a computer system as well as cosmic radiation can cause a single bit of DRAM to spontaneously flip to the opposite state. (“soft“ errors)
- As the components on DRAM chips get smaller while operating voltages continue to fall, DRAM chips may be:
  - affected by such radiation more frequently since lower energy particles will be able to change a memory cell's state.
  - or since smaller cells make smaller targets individual cells may be less susceptible to such effects
- A reasonable rule of thumb is to expect one bit error, per month, per gigabyte of memory
- Systems often use error detection and correction methods to identify and possibly correct soft errors
  - repetition schemes
  - parity schemes (74AS280)
  - cyclic redundancy checks
  - Hamming distance based checks (74LS636)

Error Detection: Parity

- A *parity bit* is a bit added to a fixed number of data bits to ensure that the total number of ‘1’s is either odd (odd parity) or even (even parity)
- Therefore, if the number of data bit ‘1’s is odd in an even parity scheme, the parity bit is ‘1’, otherwise it is ‘0’
- Likewise, if the number of data bit ‘1’s is even in an odd parity scheme, the parity bit is ‘1’, otherwise it is ‘0’
- The parity bit is transmitted with the data, and checked by the receiver
- Advantages:
  - Only one bit overhead
  - Simple digital circuit implementation
- Disadvantages:
  - Cannot correct errors, only detect them
  - Only detects an odd number of errors

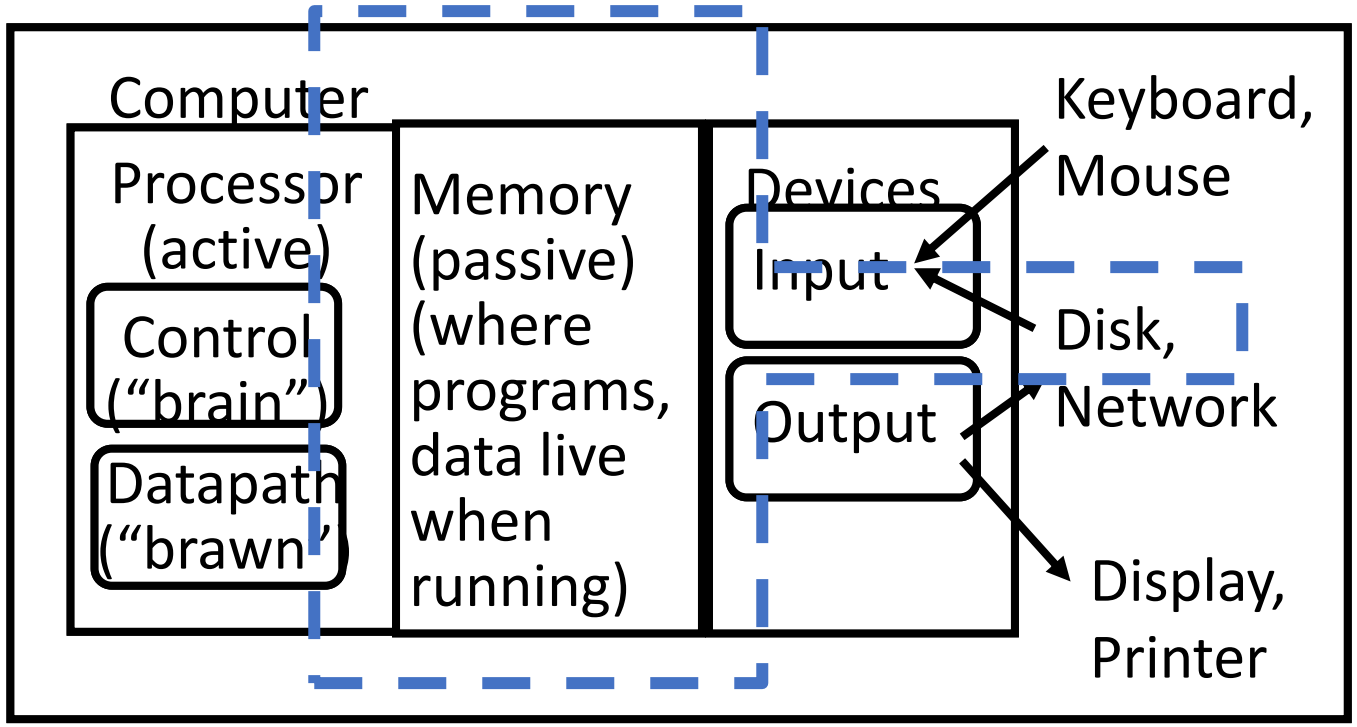
Error Correction: Repetition and Majority Voting

- Data are saved (copied) in three different memory elements
- During a memory read, all three memories are accessed and majority voting circuitry decides the final output.
- Advantages: the possibility of soft errors is practically eliminated
- Disadvantages: Triple(!) memory space is required, and there is a performance and area overhead caused by the majority voting circuitry



Memory Hierarchy

- Purpose:
  - Faster access to large memory from processor

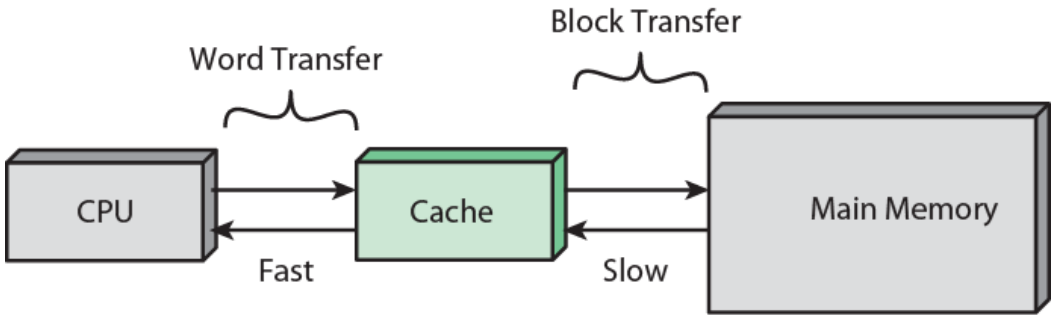


Memory Hierarchy Basis

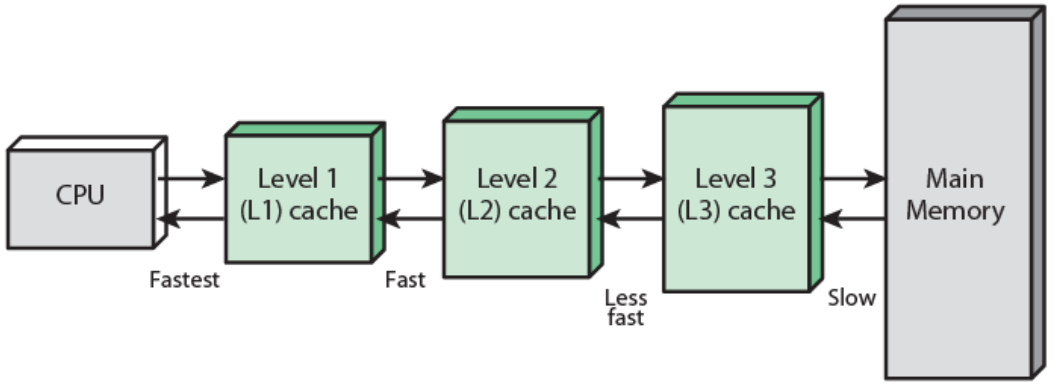
- Disk contains everything.
- When Processor needs something, bring it into to all higher levels of memory.
- On-chip Memory/Cache contains copies of data in memory that are being used.
- Memory contains copies of data on disk that are being used.
- Entire idea is based on **Temporal Locality**
  - if we use it now, we'll want to use it again soon (a Big Idea)

Cache Design

- How do we organize cache?
- Where does each memory address map to? (Remember that cache is subset of memory, so multiple memory addresses map to the same cache location.) (Books from many shelves are on the same table)
- How do we know which elements are in cache?
- How do we quickly locate them?

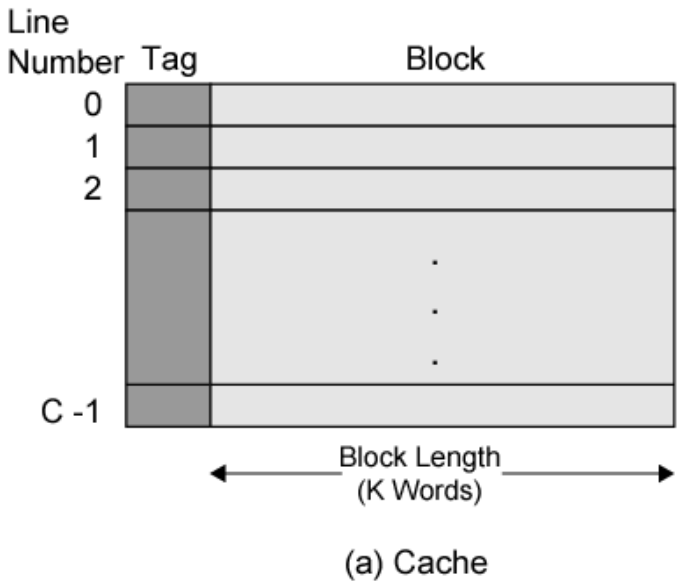


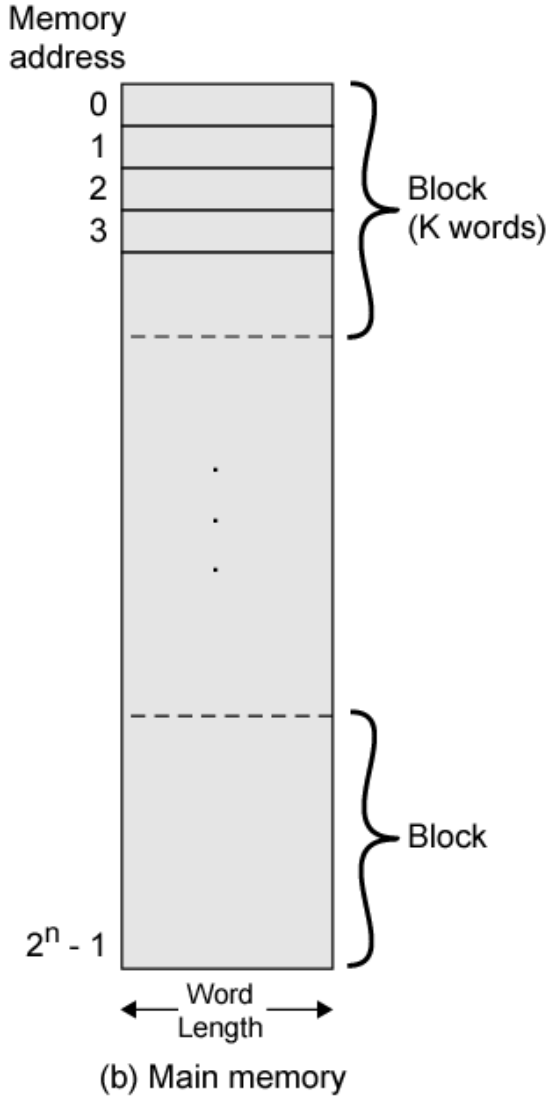
(a) Single cache



(b) Three-level cache organization

Cache/Main Memory Structure





$M = 2^n / K$ blocks in RAM

$C << M$ blocks in cache

M, numober of blocks in ram

C, number of lines in cache

Cache operation

- CPU requests contents of memory location
- Check cache for this data
- If present, get from cache (fast) *(cache hit)*
- If not present, read required block from main memory to cache *(cache miss)*
- Then deliver from cache to CPU
- Cache includes tags to identify which block of main memory is in each cache slot

Summary

- Memory organisation
- Memory operations
- Memory devices
  - ROM
  - RAM
    - SRAM
    - DRAM
      - SDRAM
      - DDRDRAM
- Addressing
- Cache memory