MICROPROCESSOR SYSTEMS (IAS0430)

Department of Computer Systems
Tallinn University of Technology

22.10.2021

MEMORY HIERARCHY

- **What is computer memory?**

MEMORY HIERARCHY

- **What is computer memory?**
  - Computer memory is any **physical device designed using integrated circuits** for the purpose of **storing and retrieving data and/or instructions** for short term use (**temporarily**) or long term storage (**permanently**).

MEMORY HIERARCHY

- **What is computer memory?**
  - Computer memory is any **physical device designed using integrated circuits** for the purpose of **storing and retrieving data and/or instructions** for short term use (**temporarily**) or long term storage (**permanently**).

  - Computer memory contributes to the circulation of information within a computer system. **Therefore ...**

MEMORY HIERARCHY

- **What is computer memory?**
  - Computer memory is any physical device designed using integrated circuits for the purpose of storing and retrieving data and/or instructions for short term use (temporarily) or long term storage (permanently).
  - Computer memory contributes to the circulation of information within a computer system. Therefore **memory devices and their design affect the overall performance of the system.**

MEMORY HIERARCHY

- **What is computer memory?**
  - Computer memory is any physical device designed using integrated circuits for the purpose of storing and retrieving data and/or instructions for short term use (temporarily) or long term storage (permanently).
  - Computer memory contributes to the circulation of information within a computer system. Therefore **memory devices and their design affect the overall performance of the system.**
  - Types of memory:
    - **Bi-Polar memory:**
      - Uses bi-polar transistors to store information
    - Magnetic Memory:
      - Uses magnetism of material to store information
    - Optical Memory:
      - Uses optical mediums to store information

MEMORY HIERARCHY

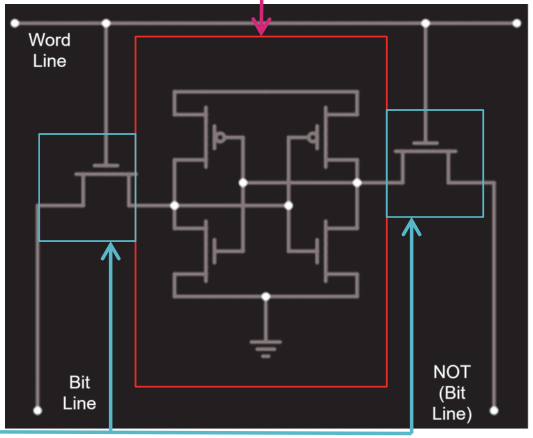
- **Bi-Polar Memory**
  - Called bi-polar because it uses **bi-polar transistors to store information**. Those bi-polar transistors can be replaced with a **MOSFET**.
  - There is two notable memories that use this:
    - **SRAM – Static RAM**
      - Uses a **latch circuit to store a bit** – similar to a flip flop – by trapping a voltage between the MOSFETs to represent the data.

MEMORY HIERARCHY

- **Bi-Polar Memory**
  - Called bi-polar because it uses **bi-polar transistors to store information**. Those bi-polar transistors can be replaced with a **MOSFET**.
  - There is two notable memories that use this:
    - **SRAM – Static RAM**
      - Uses a **latch circuit to store a bit** – similar to a flip flop – by trapping a voltage between the MOSFETs to represent the data.
      - Made out of Six MOSFETs
        - **Four MOSFETs** to store the bit
        - **Two MOSFETs** to control Read/Write accesses to the cell.

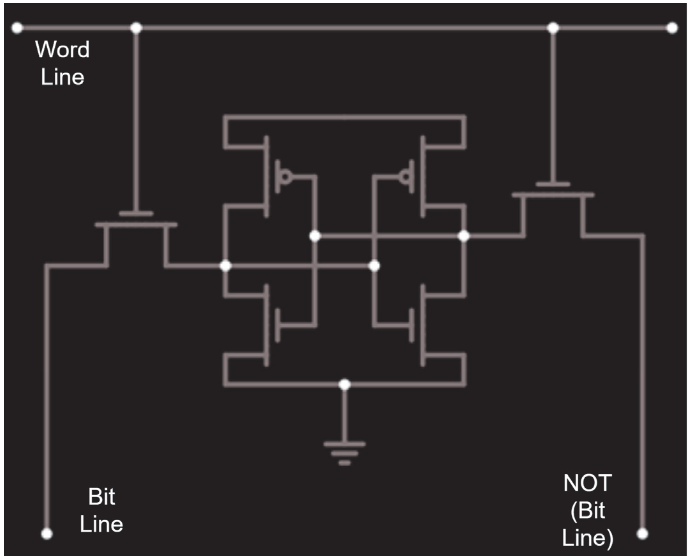
      - SRAM cells trap the data between the 4 MOSFETs mimicking the behavior of a flip-flop

      - Data is retained as long as there is electricity in the circuit



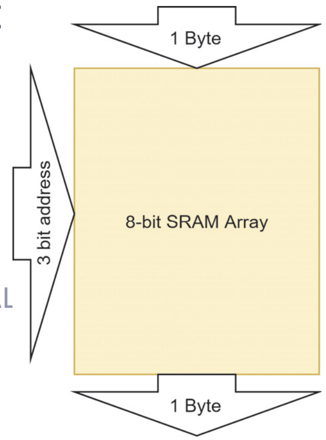
MEMORY HIERARCHY

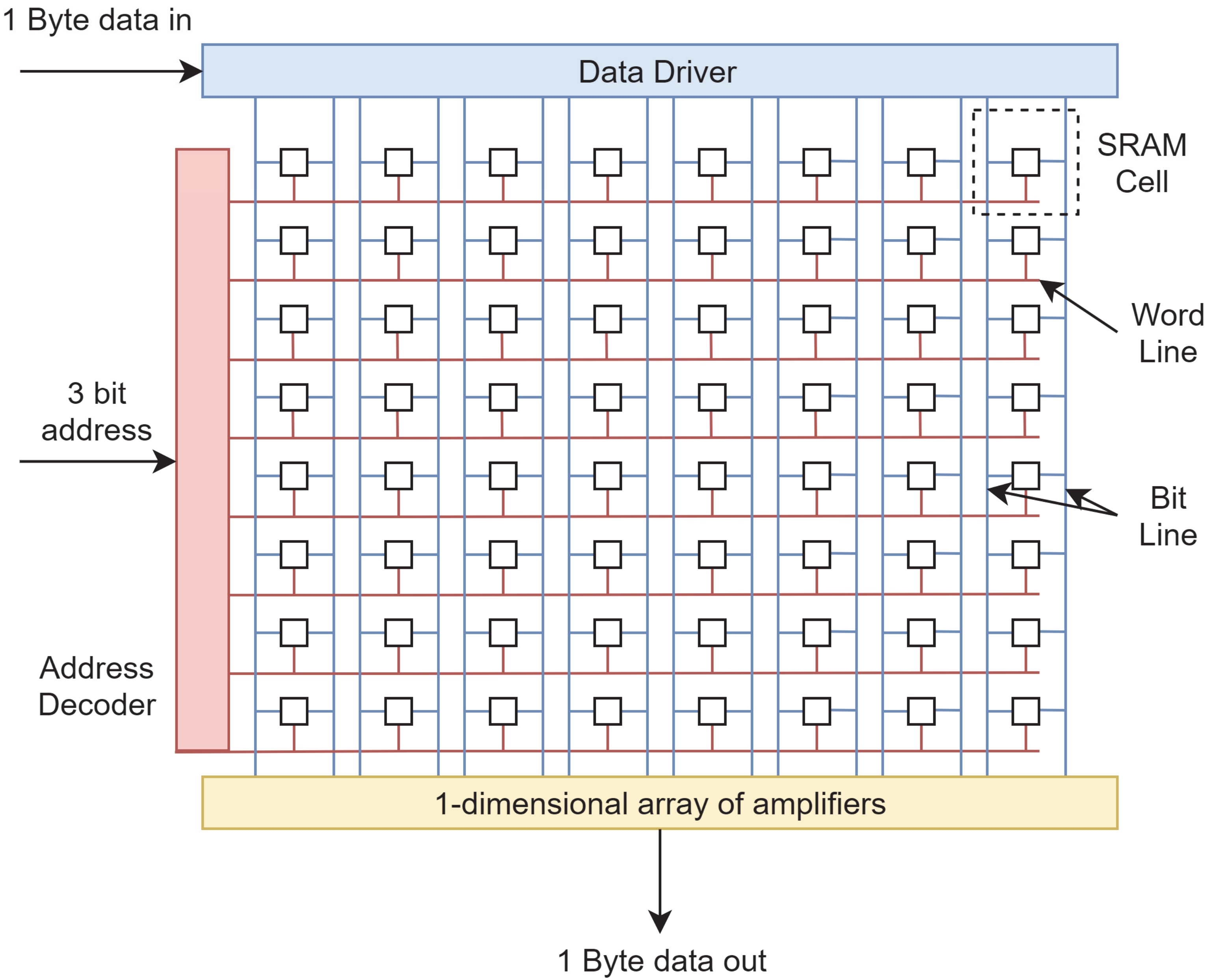
- **Bi-Polar Memory**
  - The SRAM cell is controlled using three inputs:
    - **The Word Line** is used to select which SRAM cell being activated for read and write.
      - Since the word line is connected to the Two MOSFETs controlling the Read/Write accesses to the cell, its value controls if a write/read can occur or not!
    - **The Bit Line** is used to change the value inside the cell. It carries the data needed to written into the cell. This data is used to control which of the four MOSFETs inside the cell are active, allowing to implement a latch within those four MOSFETs.
    - **The NOT (Bit Line)** is the reverse of the Bit Line. This line is crucial as it allows controlling the four MOSFETs storing data. By reversing the value of the input, the different MOSFETs are activated at the same time allowing the latch to reverse value.



MEMORY HIERARCHY

- **Bi-Polar Memory**
- **SRAM ARRAY**
  - Two bit lines for rea and write
  - One word line for array row selection
  - A decoder for row selection
  - A driver for data input
  - Amplifiers for data output





TAL TECH | TAL

MEMORY HIERARCHY

- **Bi-Polar Memory**
  - Called bi-polar because it uses bi-polar transistors to store information. Those bi-polar transistors can be replaced with a MOSFET.
  - There is two notable memories that use this:
    - **SRAM – Static RAM**

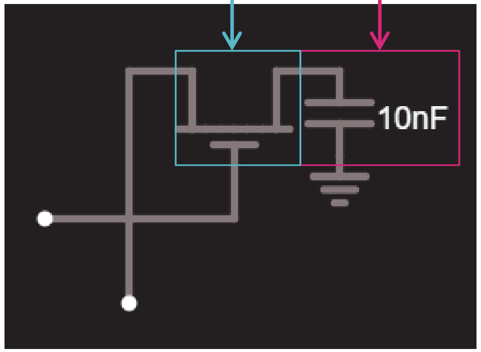
Advantages	Disadvantages
High reliability	Volatile
No refresh needed	High power consumption
High performance	Expensive
Simple access	Low density
	Complex structure

MEMORY HIERARCHY

- **Bi-Polar Memory**
  - Called bi-polar because it uses bi-polar transistors to store information. Those bi-polar transistors can be replaced with a MOSFET.
  - There is two notable memories that use this:
    - **DRAM – Dynamic RAM**
      - Uses a capacitor and a MOSFET (or transistor) to store information as a capacitance charge.

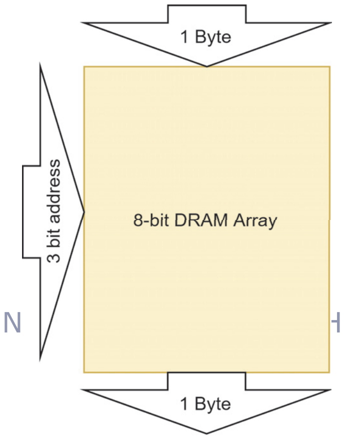
MEMORY HIERARCHY

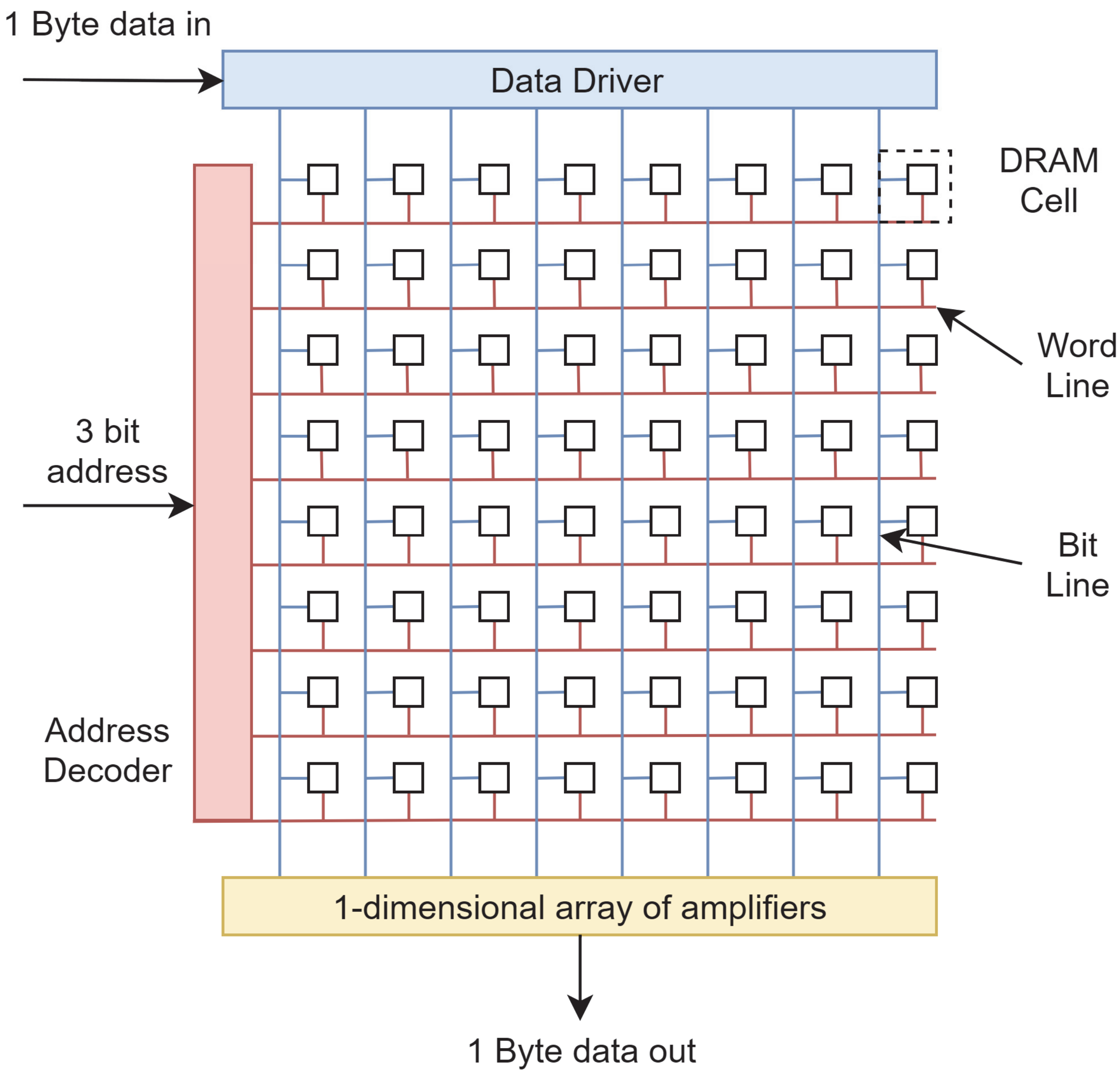
- **Bi-Polar Memory**
  - Called bi-polar because it uses bi-polar transistors to store information. Those bi-polar transistors can be replaced with a MOSFET.
  - There is two notable memories that use this:
    - **DRAM – Dynamic RAM**
      - Uses a capacitor and a MOSFET (or transistor) to store information as a capacitance charge.
        - Capacitor to store information as a charge
          - If **charged**, then 1 – If **not charged**, then 0
        - A MOSFET to control access to the capacitor.
          - Allows charging and discharging the capacitor
      - In **writing**, the MOSFET is set to allow the capacitor to Charge from the voltage coming from a driver.
      - In reading, the capacitor is discharged to the amplifiers. After each read of 1, the capacitors need to be recharged In a process called refreshing!



MEMORY HIERARCHY

- **Bi-Polar Memory**
- **DRAM ARRAY**
  - One bit line for read and write
  - One word line for array row selection
  - A decoder for row selection
  - A driver for data input
  - Amplifiers for data output





MEMORY HIERARCHY

- **Bi-Polar Memory**
  - Called bi-polar because it uses bi-polar transistors to store information. Those bi-polar transistors can be replaced with a MOSFET.
  - There is two notable memories that use this:
    - **DRAM – Dynamic RAM**

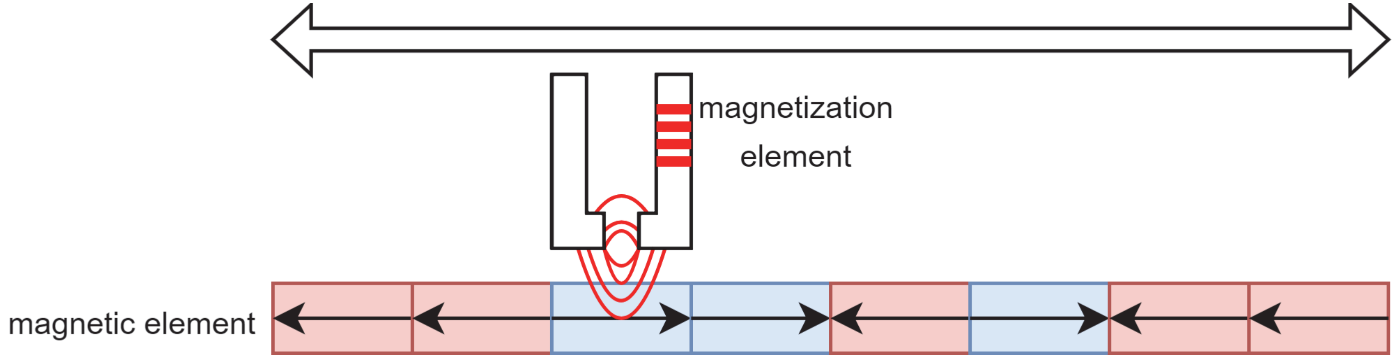
Advantages	Disadvantages
Simple	Volatile
Low Cost	Slow
High density	Refreshing is need
Small area per cell	Refreshing circuitry can become very complex for large arrays
Cheap	High power leakage

MEMORY HIERARCHY

- **What is computer memory?**
  - Computer memory is any physical device designed using integrated circuits for the purpose of storing and retrieving data and/or instructions for short term use (temporarily) or long term storage (permanently).
  - Computer memory contributes to the circulation of information within a computer system. Therefore their performance affects the overall performance of the system.
  - Types of memory:
    - Bi-Polar memory:
      - Uses bi-polar transistors to store information
    - **Magnetic Memory:**
      - Uses magnetism of material to store information
    - Optical Memory:
      - Uses optical mediums to store information

MEMORY HIERARCHY

- **Magnetic Memory:**
  - This form of memory uses magnetizing a medium to store information as **magnetization**.
  - A medium, usually in the form of a rotating disk or a strip, is magnetized. The rotation of the magnetic field on the surface of the medium donates the value of the data being 1 or 0.
  - A magnetization element is moved back and forth above a magnetic element.
    - When the magnetization of the magnetization element is change, the individual cells of the magnetic element change their magnetic rotation.
    - This rotation donates the state of data saved (0 or 1)



MEMORY HIERARCHY

- **Magnetic Memory:**

Advantages	Disadvantages
Non-Volatile	Sensitive surfaces
Low cost	Heat can cause data corruption
Simple structure	Very slow compared to Bi-polar memory
Long data retention	Erosion of data over long time
	Mechanical parts

MEMORY HIERARCHY

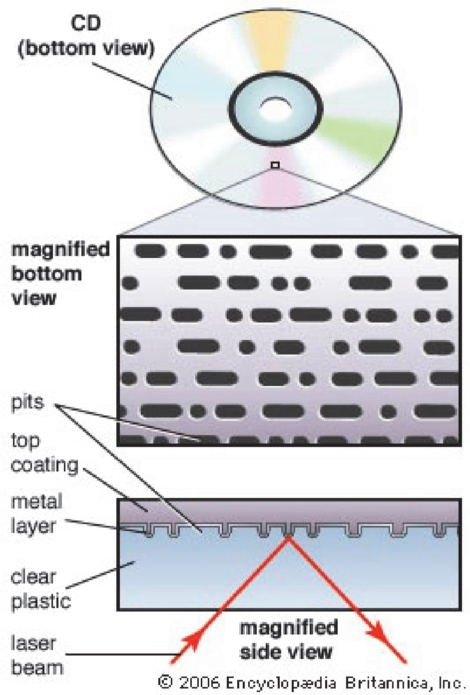
- **What is computer memory?**
  - Computer memory is any physical device designed using integrated circuits for the purpose of storing and retrieving data and/or instructions for short term use (temporarily) or long term storage (permanently).

  - Computer memory contributes to the circulation of information within a computer system. Therefore their performance affects the overall performance of the system.

  - Types of memory:
    - Bi-Polar memory:
      - Uses bi-polar transistors to store information
    - Magnetic Memory:
      - Uses magnetism of material to store information
    - **Optical Memory:**
      - Uses optical mediums to store information

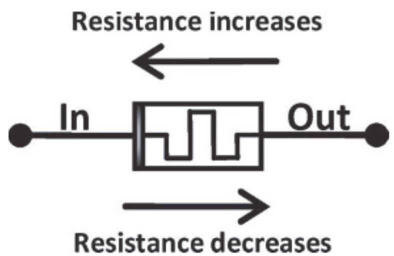
MEMORY HIERARCHY

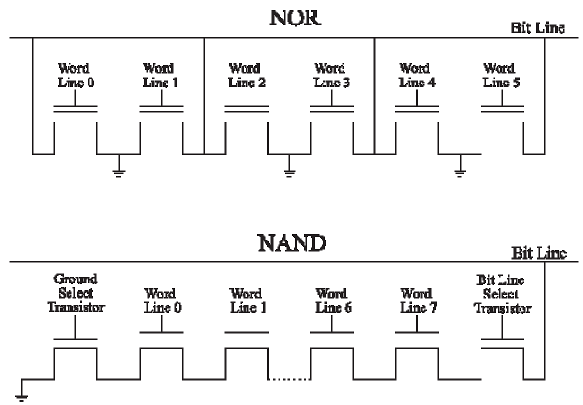
- **Magnetic Memory:**
  - This form of memory uses optical mediums to store data as patterns.
  - A medium, usually in the form of a rotating disk or a strip, is magnetized. The rotation of the magnetic field on the surface of the medium donates the value of the data being 1 or 0.
    - The reflection of the light indicates the type of data being stored onto the medium.



MEMORY HIERARCHY

- **Other memory types:**
  - **Electrically Erasable Programmable Read Only Memory (EEPROM)**
    - Very fast and modern technologies that rely heavily on Flash memory technologies.
    - These require high complexity structures to store data.
      - Examples of this are NAND Flash memory, AND Flash memory, and NOR Flash memory.
      - This memory uses a special type of transistor called the Floating Gate Transistor
      - It is non volatile, meaning that data can be retained without the need for power
  - **Memristors**
    - Memristors are a two terminal devices that take advantage of the relationship between the magnetic flux and charge.
    - They store information as physical state of the material they are made of, not as electrical charge.





*S. Kvatinsky, S. Member, D. Belousov, S. Liman, G. Satat, S. Member, N. Wald, E. G. Friedman, A. Kolodny, S. Member, and U. C. Weiser, "MAGIC — Memristor-Aided Logic", IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 11, pp. 895–899, 2014. DOI: 10. 1109 / TCSII . 2014 .2357292.

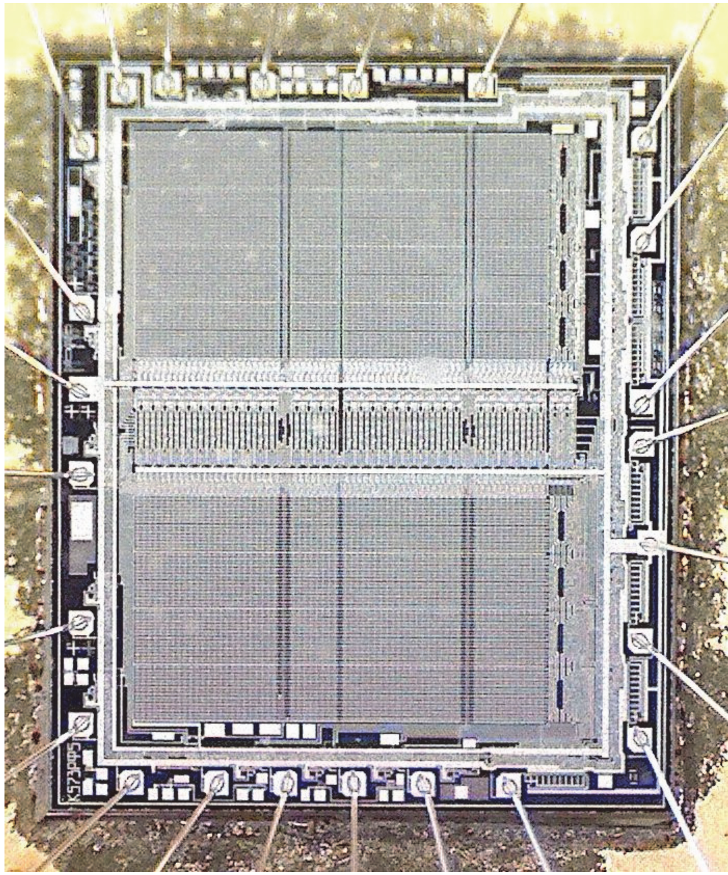
**http://www.iue.tuwien.ac.at/phd/windbacher/node14.html

MEMORY HIERARCHY

- **Memory Devices:**
  - There are many types of memory devices that are used to day….
    - Examples?

MEMORY HIERARCHY

- **Memory Devices:**
  - There are many types of memory devices that are used to day….
    - RAM
    - Registers
    - Flip Flops
    - Caches
    - CD ROMS
    - SSD
    - HDD
    - Flash memory
    - BOOT ROM
    - …… and many others



MEMORY HIERARCHY

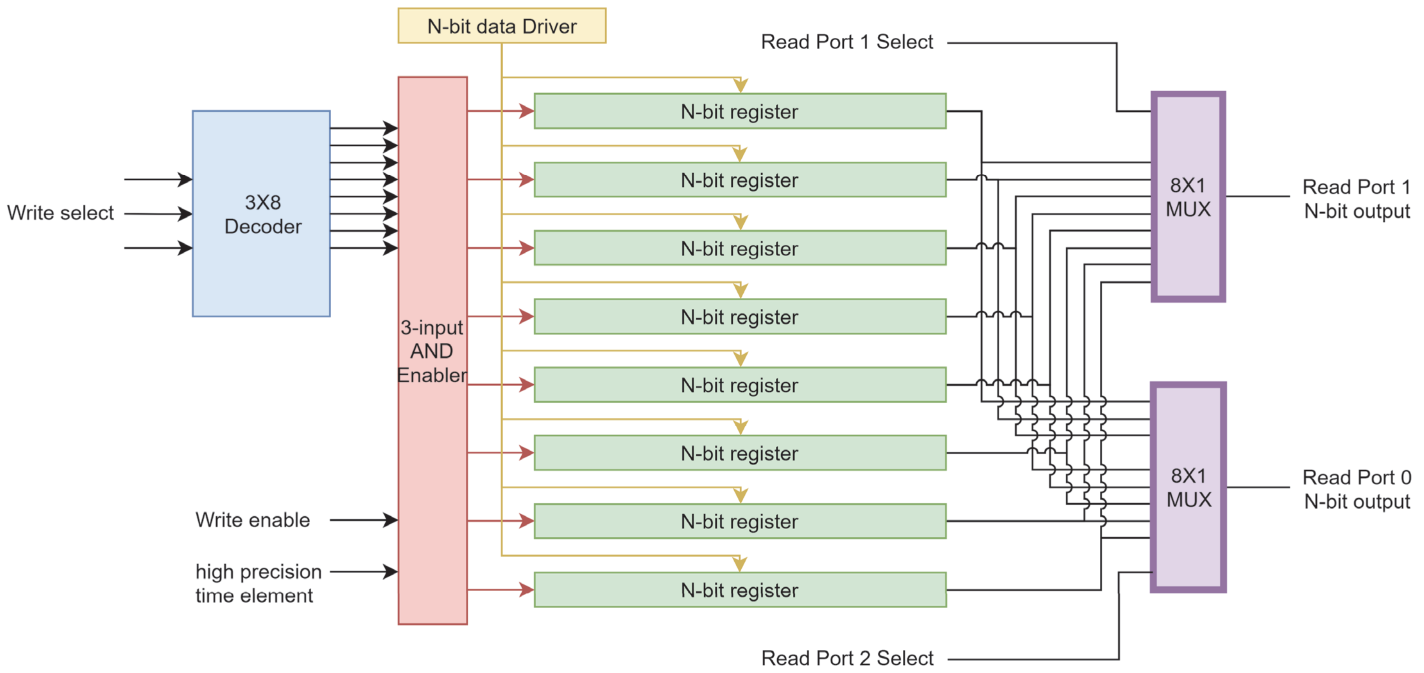
- **Memory Devices:**
  - There are many types of memory devices that are used to day….
    - Today we will be speaking about two of those:
      - Register Files
      - Single Layer caches
    - Next week we will talk about:
      - Multi Level caching
        - Cache addressing
        - WRITE BACK and WRITE THROUGH
      - Large memories
        - RAM
        - HDD
        - SSD – use internal RAM for data caching
        - CD ROM
        - ROM

MEMORY HIERARCHY

- **Memory Devices:**
  - **Register Files:**
    - Those are array of registers located inside the CPU.
    - in **the past** those were independent registers made out of sequences of flip flops.
      - These registers can be specialized:
        - Instruction register, Program counter, ALU accumulator, MSHRs, etc…
      - Can also be general purpose registers used by the CPU for different purposes
    - Today, they are made of **SRAM with multiple ports** were each SRAM register has its own dedicated Write port and 2 Read ports – faster access.
    - The RegFile is the first place the CPU looks for data.
      - If data is found in the registers, it is loaded into the processor and processed.
      - If data is not found on the registers, a cache request is issued.
        - The CPU does not issue a cache request, the register file does.
        - We look for information in the cache.

MEMORY HIERARCHY

- **Register Files:**
  - A write select is decoded to select which register to write the data to.
  - The write enable decides if data needs to be written
  - Read port 1 / 2 select to choose which data goes to which of the two read ports



MEMORY HIERARCHY

- **Memory Devices:**
  - **Caches:**
    - The cache is a **static random access memory designed to operate and to be accesses by the CPU at a much faster rates than the Main Memory**
      - It bridges the speed gap between the Main memory and the CPU.
      - Provide a storage for data that is right near the CPU, increasing the efficiency of the CPU.
      - It is located right inside the CPU. This means that it is very close to the processor.
    - It is the second place where we look for data:
      - If data is found in the cache, this is called a cache **hit**
      - If data is not found in the cache, this is called a cache **Read miss**.
      - If data is being written into the cache in a location where data is already stored, this is called a cache **Write miss**. – we will get to see more types of misses later.
      - If a cache miss occurs, **the data is then requested from the slower main memory**

MEMORY HIERARCHY

- The principle of **cache locality**:
  - Is the principle that relies on the fact that the CPU will access a given set of memory addresses repeatedly.
    - Meaning that the CPU will repeatedly want data stored in certain memory locations more than once during execution.
    - **Spatial locality:** When an address is accessed, it **is highly likely that the CPU wants to access the sequential memory locations located after that address as well**.
      - An example of this is when for-loops and arrays are executed.
      - A sequence of locations that are accessed repeatedly
    - **Temporal locality**: **When an address is accessed, it is highly likely that the CPU wants to access the that address again later**.
      - An example of this is variables.
      - One specific location that is used over and over again.
  - **Locality** can help **determine how the cache should be accessed** and what **methods to use in cache replacement policies** for different programs– remember this for later on…