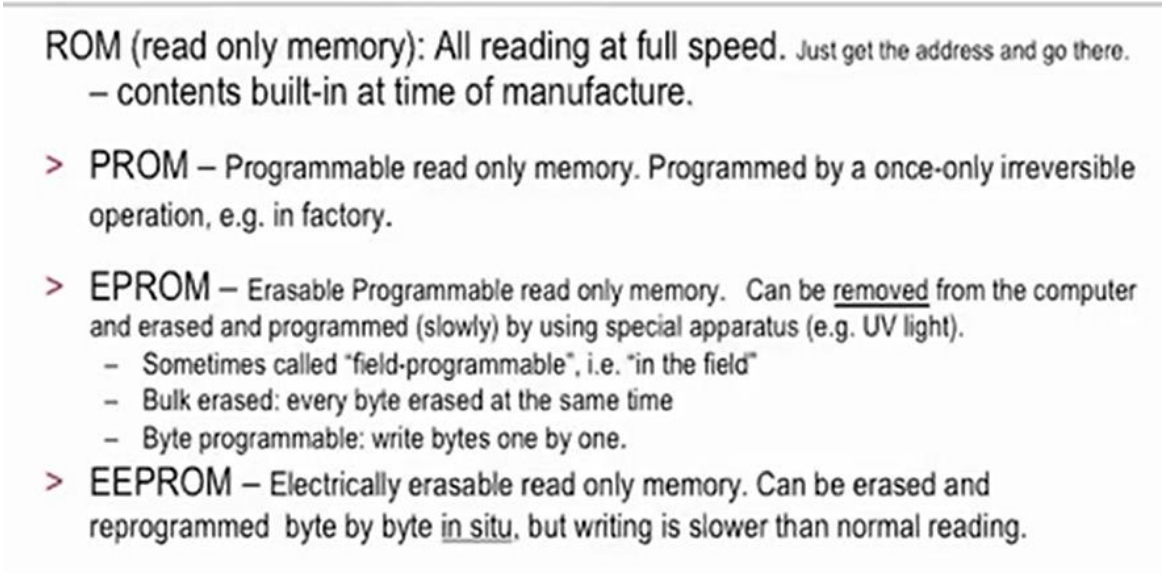
1. Review the lecture slides on types of memory and provide a short answer to the following questions (using your own words):

1.1. What is ROM and what is its primary purpose?

- **ROM (Read only memory):** a type of memory that store data permanently and cannot be changed. Data storing in ROM cannot be changed and will be retained even when the computer is powered off.
- **Purpose:**
  + Optimize reading-only actions.
  + Storing BIOS – instruction for computer right after turning on.
  + Storing data that needs to be permanently retained and accessible, even when power is turned off or the device is restarted.



1.2. What is RAM and how is it different from ROM ?

- **RAM (Random Access memory):** a type of memory that store data that the computer's processor (CPU) needs to access quickly while it's actively running programs and performing tasks.
- **Differences with ROM:**
  + Data storing in RAM can be modify and can be easily read and write frequently.
  + Data storing in ROM are not intended to change frequently and are only intended to Read.
  + Data storing in RAM cannot be retained after the computer turned off.
  + Data storing in ROM can be retained after the computer turned off.

1.3 What is the difference between static RAM and dynamics RAM ?

- **Static RAM:** retain information until power removed. Fast, larger area of silicon per byte, requiring constant power.
- **Dynamic RAM:** retain information as long as the contents are refreshed frequently enough. Smaller area of silicon per byte, low power requirement.

1.4 What type of memory is typically used in USB thumb drives? Why shouldn't we rely on this for critical data storage?

- **Flash memory (EEPROM)** are typically used in USB thumb drives.
- Because charge stored between insulator, in order to writes, the electrons must pierce through barrier layer and physically damaging it. Therefore, it has a finite life and could be unusable after a certain number of writes.

2. Consider a computer with 1GB RAM (1024 MB). Given memory addressing is for each byte, how many bits are needed to address all bytes in the system's RAM?

- 1gb = 1024 MB = 1024*1024 kb = 1024^3 byte = 1024^3 * 8 bits

3. Give a brief description of the Von Neumann and Harvard computing architectures. What are the fundamental differences between the two and for what is each designed to achieve?

- **The Von Neumann architecture:** is fundamental computing architectures which *control bits and data bits (data and instruction) are stored a* ***common memory space/hardware***.
- **Harvard Computing architecture:** is fundamental computing architectures which *control bits and data bits (data and instruction) are stored a* ***separate memory space/hardware***.
- The Von Neumann architecture prioritizes flexibility in programming, support a wide range of tasks, handling multiple tasks, handling interruptions (one programs need immediate attention while another process is undergoing). **Aiming for multiple tasking CPU.**
- The Harvard Computing architecture prioritizes flexibility of hardware and spaces, performance optimization, parallelism, immune to buffer overflow attack. **Aiming for single tasking CPU where memory is scarce, and speed is important.**

4. What is cache memory and what is its primary role?

- **Cache memory** is a type and of high-speed volatile computer memory providing a temporary storage area for data instructions.
- **The primary role** of cache memory is to provide a temporary storage area for data and instructions that the CPU uses frequently or is likely to use soon.

5. Explain the concept of an interrupt, and list four common types.

- Interrupts is a response to a signal that needs attention from the software.
- Instead of going and asking each device if they need attention, the devices will trigger an interrupt if they need attention.

Four Common types of interrupts:

- Hardware Interrupts (keyboards, mouse…).
- Software interrupts
- Clock interrupts.
- I/O interrupts.

5.1. Polling is an alternative to interrupts. Briefly explain polling and why it is not commonly used.

- Polling is an alternative to interrupts. The concept of polling is that the CPU is going to ask if each device is needing attention, checking states of each device, …
- Because:
    - It can't utilize CPU resources and wasting time checking hardware which doing nothing.
    - Wasting time and freezing computer while polling.

6. Explain the general concept of a stack - how do they work, and what is their primary purpose.

- Stacks if a way of organizing, storing, and addressing data, operate in the First in – Last Out principle.

6.1. How are stacks useful for handling interrupts?

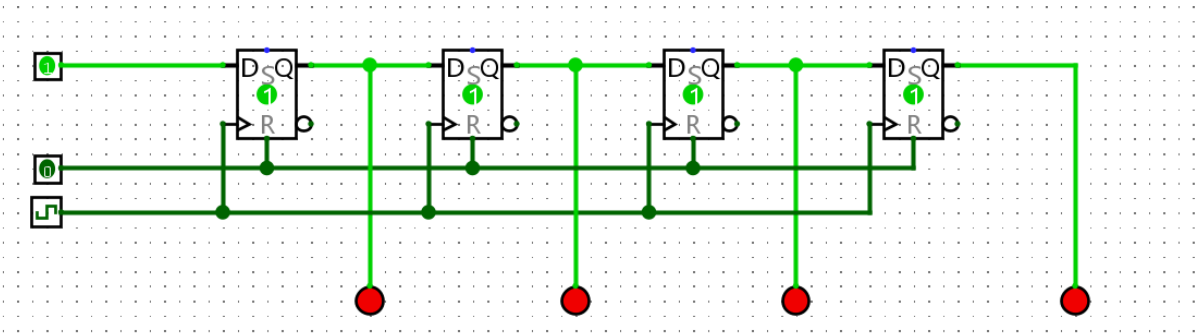
- A stack allows us to mothball/backup/hibernate a process task while dealing with another tasks.
    - Push the instructions/data into the stacks.
    - Dealing with the interruption.
    - Pop the stored data/instructions back off the stack.

6.2. How are stacks useful in programming?

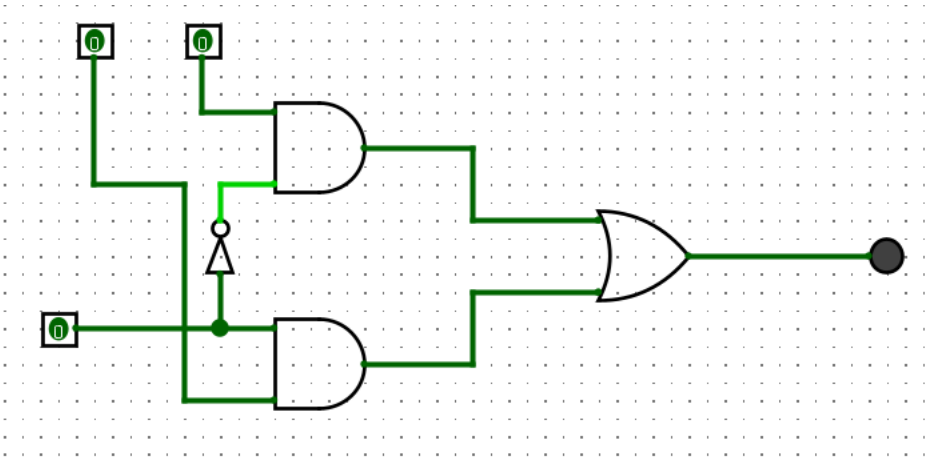
- Stack allows storing procedure/function while programming.

Practical - Stacks of Stacks!

9. Start by building a simple shift register that moves bits from one flip flop to the next each clock pulse. For this you will need a "Data In" pin which sets the next bit to be pushed to the stack, and a clock to invoke the shifting.



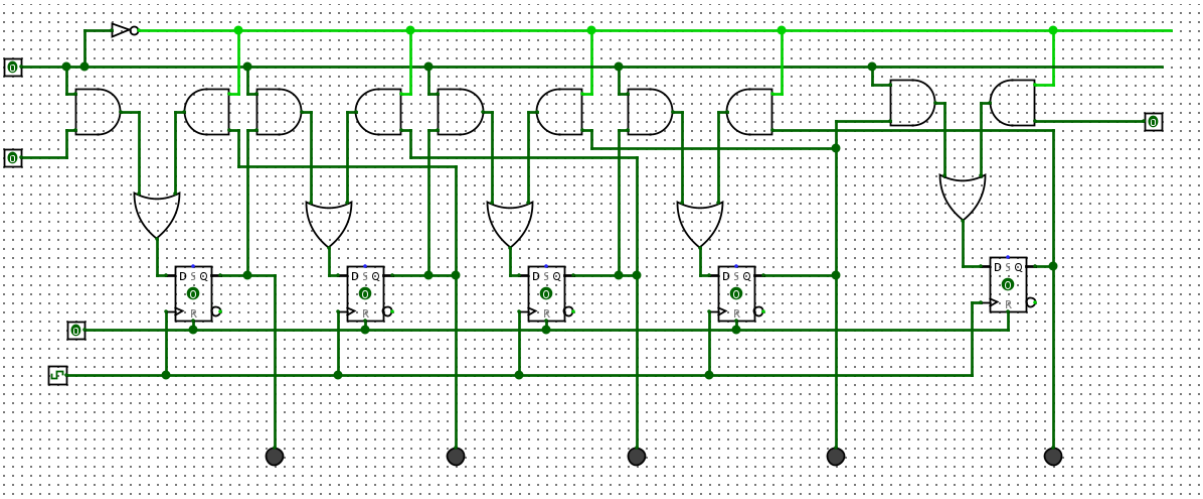
10. For your shift register to work as a stack, it needs to be bi-directional. This means the input to any Flip Flop could come from two places - the left or the right. In lectures we discussed a simple "encoder" circuit that selects which of two data inputs is allowed through, based on a third selection bit. Design the logic for this 2-bit encoder and demonstrate it to your lab demonstrator.



11. Now incorporate your encoder above to allow bi-directional shifting of your stack. Your stack should:

11.1. push and pop bits onto and off the stack, using clock pulses and a direction toggle switch.

11.2. show the state of each Flip Flop using LEDs.



12. Modify your stack so that it has the option to read out its contents in parallel to a separate register of D Flip Flops. This should only occur when a "stack dump" toggle switch (i.e., pin) is enabled. When the toggle is disabled, the register of D Flip Flops should retain the last state read in (and should have LEDs connected to each Flip Flop out showing its state).