Faculty of Science, Engineering and Technology

Computer Systems

Week 5

Student Name: Lau Ngoc Quyen

Student ID: 104198996

Overview

In this laboratory session we start look at memory, encoders and stacks.

Purpose: To consolidate your knowledge of Memory and Stacks

Task:

Time: This lab is due by the start of your week 6 lab.

Assessment: This lab is worth 1% (up to a maximum of 5%) of your assessment for this unit, and only if demonstrated to your lab demonstrator in the week it is due.

Resources:

- This week's lecture videos

Submission Details

You must submit the following files to Canvas:

- A document containing all required work as described below.

Instructions

Theory (Memory, Architectures, Interrupts and Stacks)

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

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

   - **ROM (Read-Only Memory)**

   - **Function**: Stores fixed data that cannot be rewritten. Data is not lost when power is turned off.

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

   - **RAM (Random Access Memory)** is a type of computer memory that can be read from and written to. It allows computers and devices to store and quickly access data.
   - **Difference from ROM**
   - **Read and Write Capability**: RAM can read and write data, whereas ROM (Read-Only Memory) can only read data.
   - **Volatility**: Data in RAM is lost when power is turned off, while data in ROM is non-volatile and remains even without power.
   - **Size and Capacity**: RAM typically has a larger capacity than ROM, ranging from 1GB to several tens of GB.

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

   - **Static RAM (SRAM)** is faster and doesn't need to be refreshed, while **Dynamic RAM (DRAM)** is slower and requires constant refreshing to retain data.

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

   - **USB thumb drives** use flash memory, which is not ideal for critical data storage because it has limited write cycles and can degrade over time.

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 ?

   - **A 1GB RAM** (1024 MB) system needs 30 bits to address every byte.

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 is each designed to achieve ?
   - **Von Neumann architecture** uses the same memory for data and instructions, while **Harvard architecture** separates them for faster performance.
4. What is cache memory and what is its primary role ?
   - **Cache memory** stores frequently accessed data to speed up access times.
5. Explain the concept of an interrupt, and list four common types.
   - **An interrupt** is a signal to the CPU to stop and handle an event. Common types: hardware, software, timer, and I/O interrupts.

   5.1. Polling is an alternative to interrupts ? Briefly explain polling and why it is not commonly used.
   - **Polling** is checking the device status repeatedly, making it inefficient compared to interrupts.
6. Explain the general concept of a stack - how do they work, and what is their primary purpose.
   - **A stack** is a data structure that follows Last-In, First-Out (LIFO) for storing and retrieving data.

   6.1. How are stacks useful for handling interrupts ?
   - **Stacks temporarily** store program states during interrupts, allowing the system to resume normal operation after handling the interrupt.

   6.2. How are stacks useful in programming ?

   - **Stacks** manage function calls, local variables, and control flow in programs.

Provide all the answers to the above questions in your submission document.

Practical - Stacks of Stacks !

7. Start Logisim and open a new canvas
8. Review the lecture slides on building a stack at the top of this lab sheet. We are going to build a 5-bit deep, 1-bit wide stack.
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.

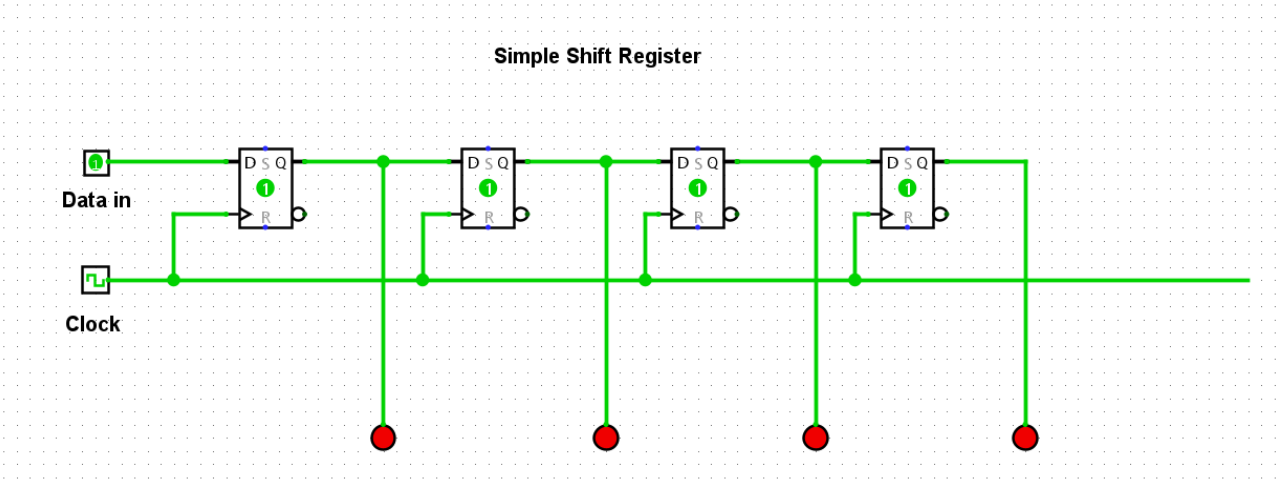

Image 1: *Simple Shift Register*

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.

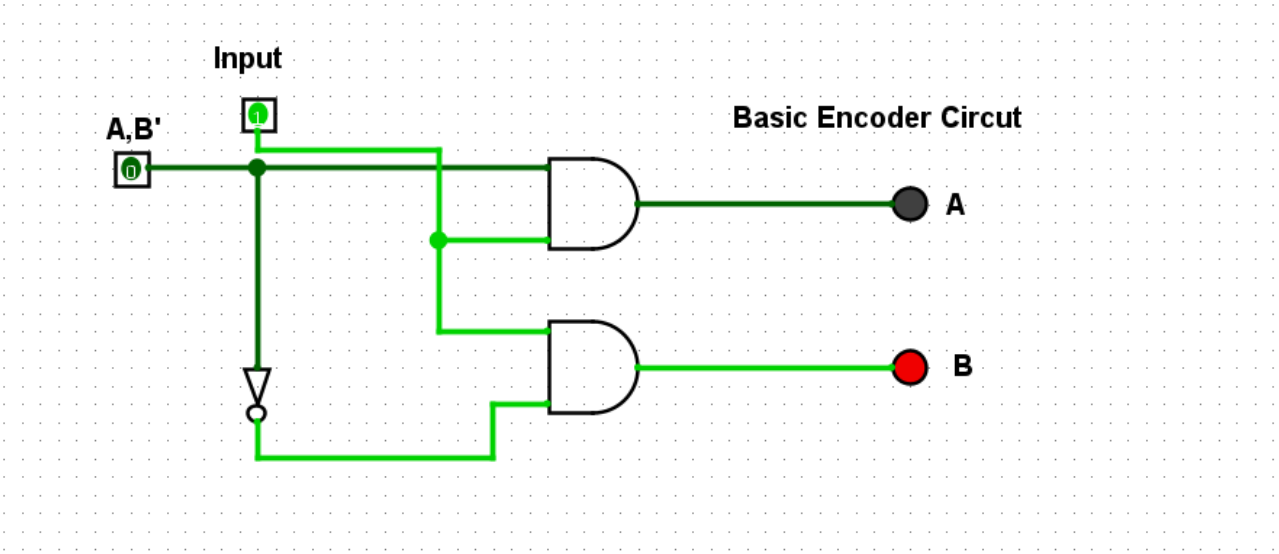

Image 2: *Two-bit encoder*

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.

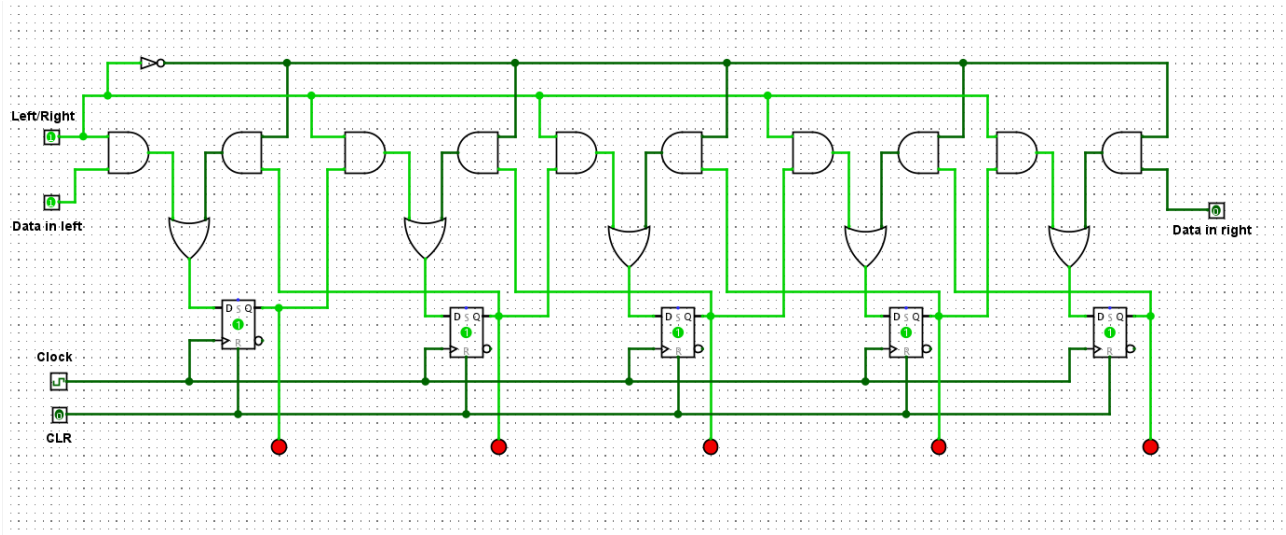

Image 3: 5-bit deep, 1-bit wide stack.

Export your circuit as an image and include it in your submission document. Demonstrate your working stack to your lab demonstrator.

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).

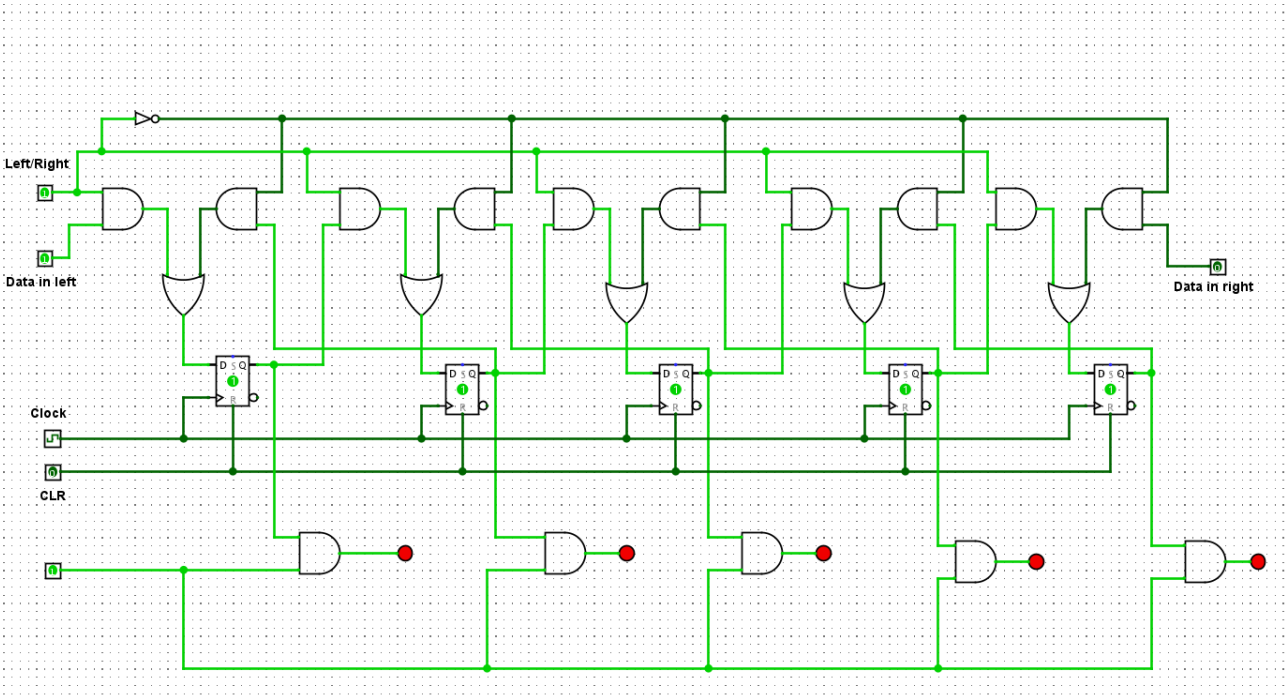

Image 4: 5-bit deep, 1-bit wide stack ***in parallel*** (Toggle on)

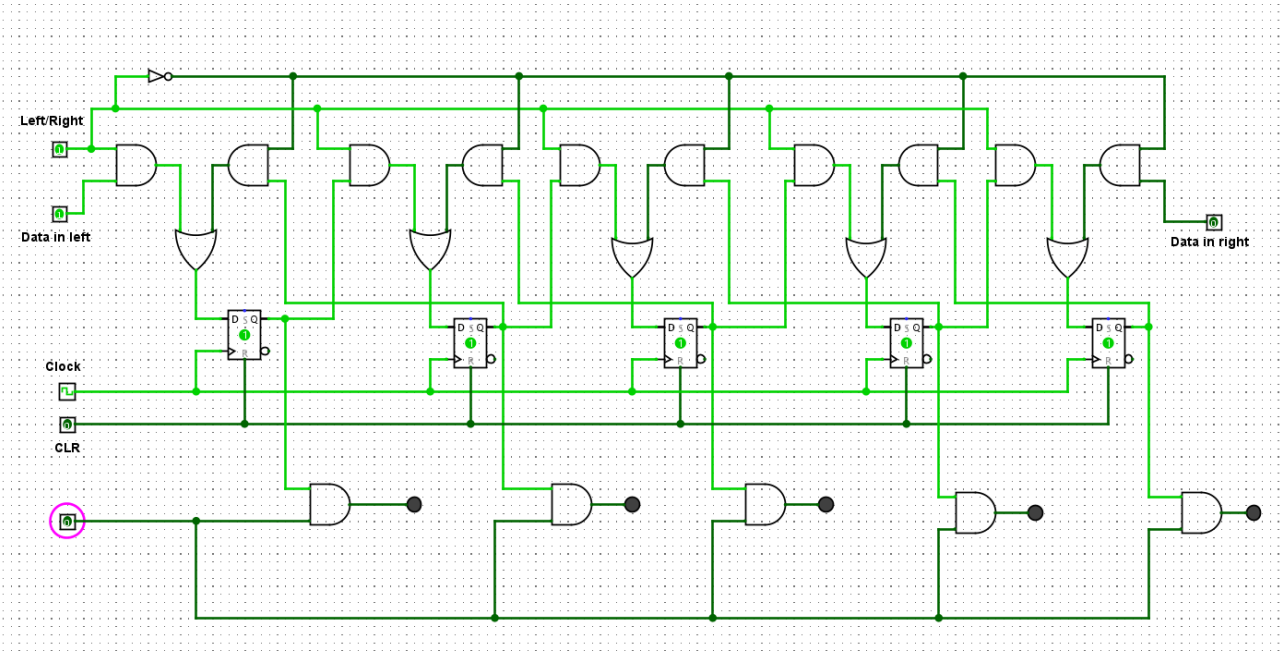

Image 5: 5-bit deep, 1-bit wide stack ***in parallel*** (Toggle off)

Export your circuit as an image and include it in your submission document. Demonstrate your working stack to your lab demonstrator.

When complete:

■ Submit your answers (screen shots, etc) in a single document using **Canvas**

■ Show your lab demonstrator your working circuits in class (you must do this to get the 1%). Your lab demonstrator may request you to resubmit if issues exist.