Faculty of Science, Engineering and Technology

Computer Systems

Week 5

Overview

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

Purpose: To consolidate your knowledge of Memory, Encoders and Stacks

Task:

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

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

Resources:

- Encoders tutorials
  - Encoders and Decoders (to come)
- Hardware Stacks:
  - Video to come (to come)

Submission Details

You must submit the following files to Canvas:

- A document containing all required work as described below.

SWINBURNE UNIVERSITY OF TECHNOLOGY

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 stands for **Read-Only Memory**, it can be used to store the data permanently and provide storage for BIOS or anywhere that a small but permanent storage solution is needed.

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

   RAM stands for Random Access Memory; it is a memory that stores data temporarily and will be erased after the computer is turned off. The main difference is that ROM is non-volatile, and RAM is not. More importantly, because of ROM's static nature it will take more time to access ROM information as the system will have to copy that data into RAM to allow us to read it.

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

   SRAM requires constant power supply, which means this type of memory consumes more power, however DRAM offer reduced power consumption, due to the fact that the information is stored in the capacitor. But SRAM has lower access time, therefore faster than DRAM.

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

   A USB thumb drive or a flash drive, like the name suggests these drives used the flash memory, this technology is long obsolete as it has limited capacity and it is slow for both reading or writing. Therefore, it should not be used for critical data storage

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 ?**

   Considering that 1GB is equal to 2^30 bytes, we will need 2^30 * 8 = 8 589 934 592 (bits) to address all bytes in the system's RAM.

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 ?**

   - Von Neumann architecture is built around the stored-program concept where instructions and data share the same memory space and transfer bus.

   - Harvard architecture is an architecture where instructions and data are stored separately and are transferred via different buses.

   - Differences: *Von Neumann* utilizes the same memory space for instructions and data while Harvard uses a separate storage. The same is for data/instructions transfer where *Von Neumann* uses a common bus while *Harvard* uses a separate bus for the two.

   - *Von Neuman architecture* is used mainly in general purpose computers and personal computers. *Harvard architecture* is used in micro controllers and signal processing.

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

   Cache memory is a temporary storage unit that stores frequently used data/instructions in high-speed memory, providing faster and more efficient data retrieval or processing.

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

   An interrupt occurs when the CPU is processing a task when it is suddenly stopped by another task requested by the user.

   Four interrupt types:
   - Clock Interrupt
   - Hardware Interrupt
   - Network Interrupt
   - Exception Interrupt

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

   Polling is a protocol where the CPU checks the state or input of all devices to determine which one has problems. Polling is not widely used as it is time-consuming and wasteful of resources for CPU

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

   A stack is an abstract data type which stores and performs tasks following an ordered, linear sequence. Stack follows the principle of *Last In – First Out*, or LIFO, where the last element inserted into the stack (the element on the top of the stack) would be the first element to be removed/performed. Stack acts as a simple yet effective way to store or recall data.

   6.1. **How are stacks useful for handling interrupts ?**

   Interrupt Handler is a hardware-triggered function call that has the ability to nest, which would mean that the only viable data structure to correctly store return addresses is stack.

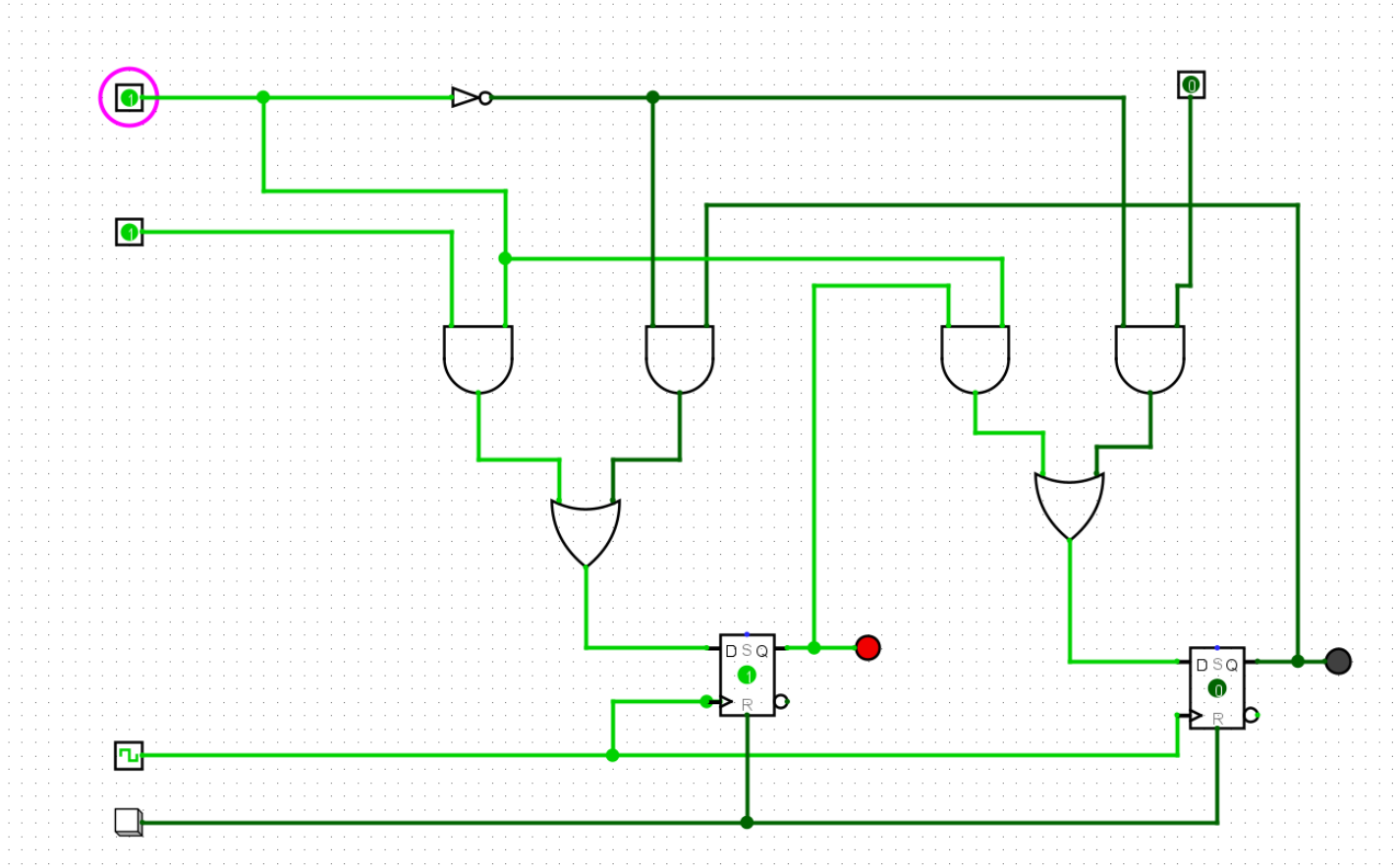
   6.2. **How are stacks useful in programming ?**

   Stacks can be used to implement functions, parsers, evaluate expressions or backtracking.

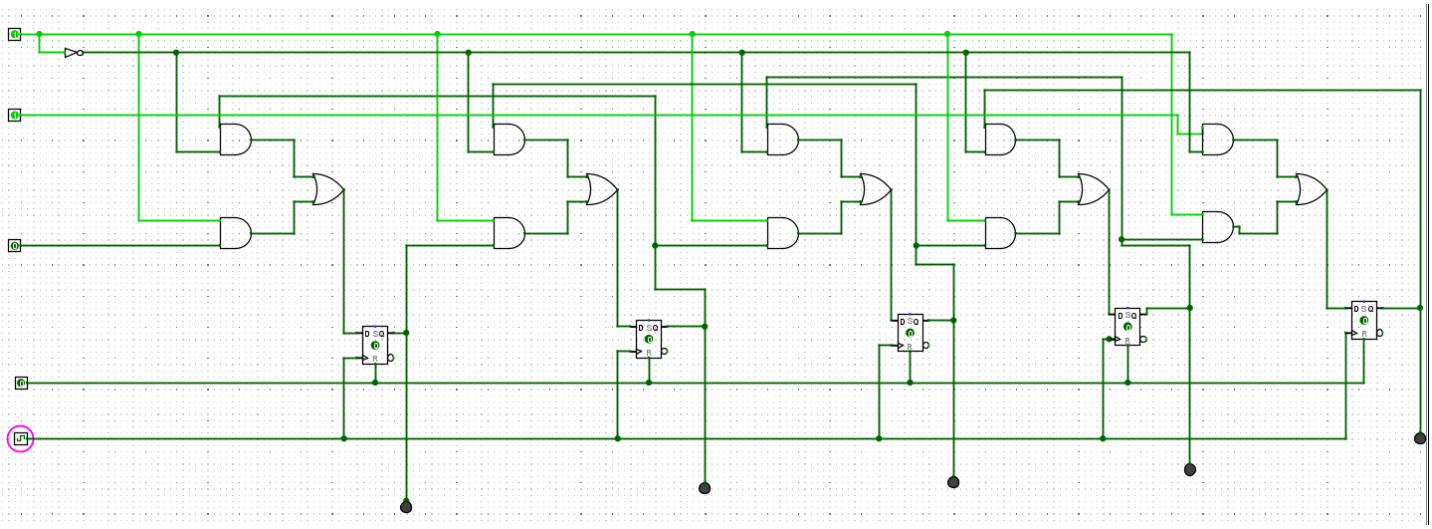
Provide all the answers to the above questions in your submission document.

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.
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.



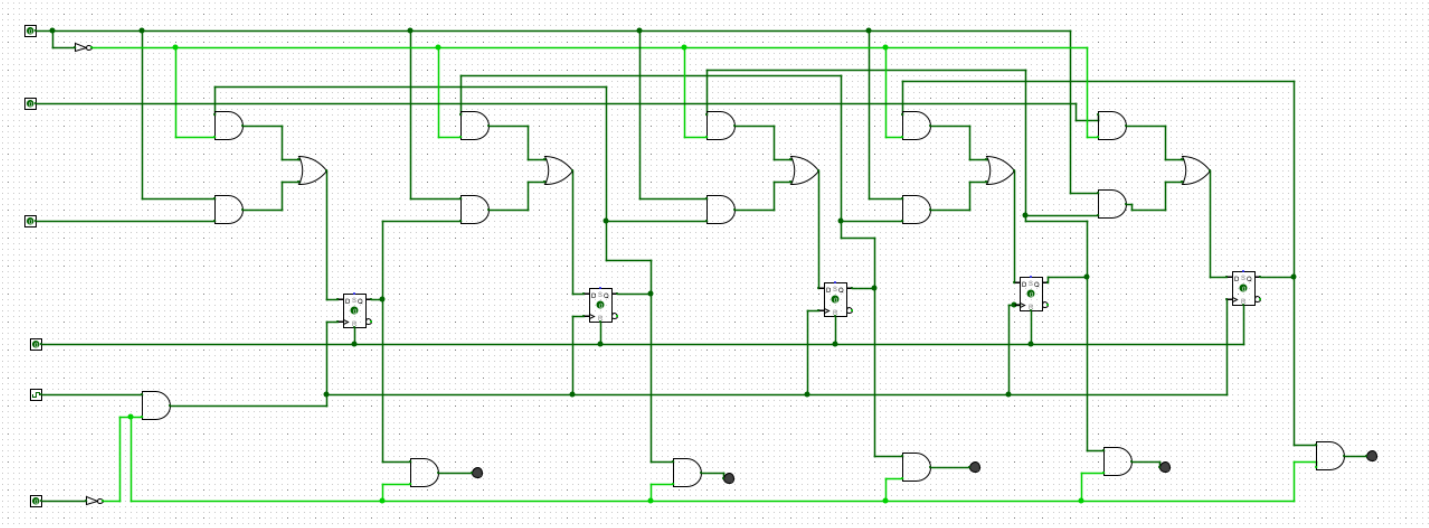
11. Now incorporate your encoder above to allow bi-directional shifting of your stack. Your stack should:
    11.1. push and pop bits onto and off the stack, using clock pulses and a direction toggle switch
    11.2. show the state of each Flip Flop using LEDs.



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).

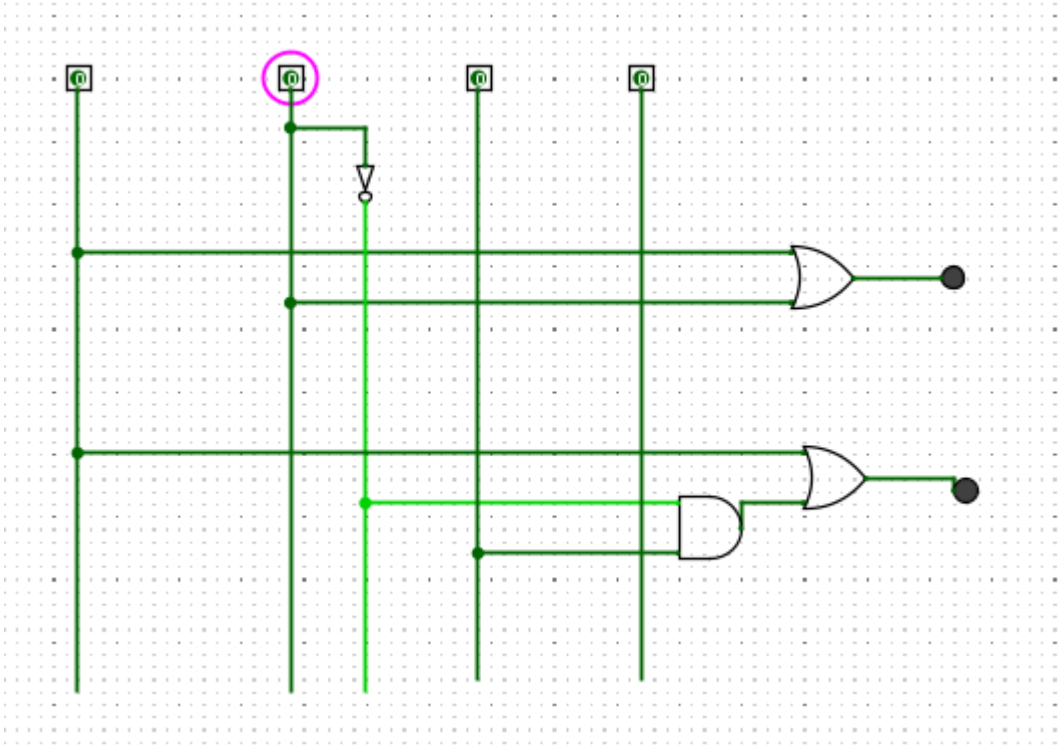
Export your circuit as an image and include it in your submission document.
Demonstrate your working stack to your lab demonstrator.



Practical – Encoders and Decoders

13. Review the lecture slides and build a 4-to-2 priority encoder. Simulate the circuit with all possible combinations of 4 input values and observe the corresponding output values. Write the results in form of a truth table.

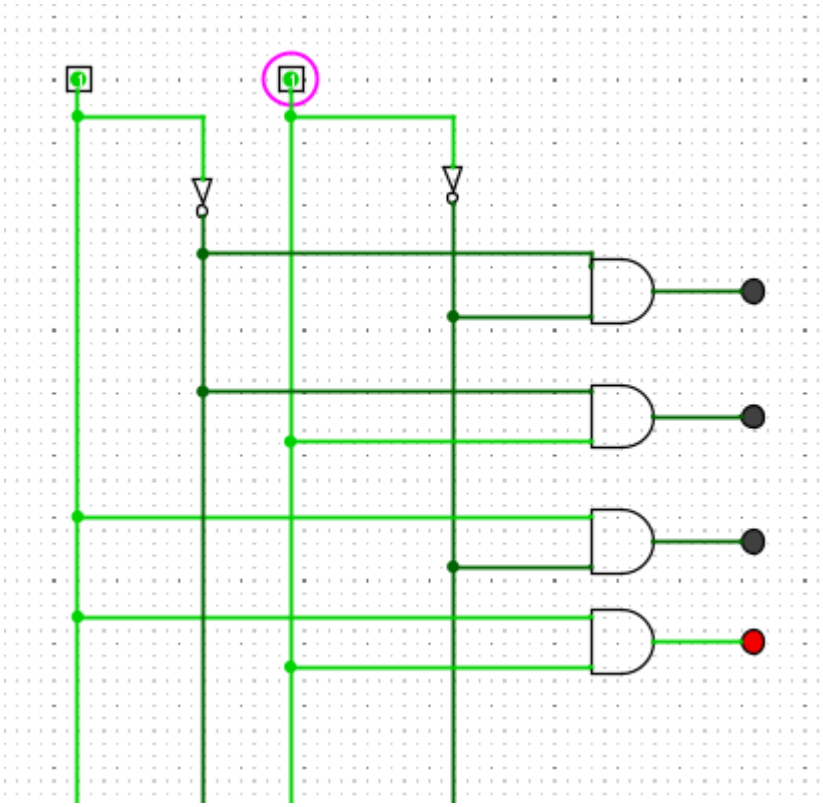
D3	D2	D1	D0	Q1	Q0
0	0	0	**1**	0	0
0	0	**1**	X	0	**1**
0	**1**	X	X	**1**	0
1	X	X	X	**1**	**1**
0	0	0	0	X	X



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

14. Now wire up a 2-to-4 binary decoder and simulate the circuit with all possible combinations of 2 input values and observe the corresponding output values. Write the results in form of a truth table.

Input1	Input0	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Export your circuit as an image and include it in your submission document.
Demonstrate your working circuit 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.