Faculty of Science, Engineering and Technology

stacks.



Computer Systems

Week 5

# Overview

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

**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 answer to the following questions (using your own words):
   1. What is ROM and what is its primary purpose?

From my perspective, ROM (Read Only Memory) Is a type of computer memory that stores permanent data and instructions for a device to start up or regenerate each time it is turned on.

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

|  |  |  |
| --- | --- | --- |
| **Characteristics** | **RAM (Random Access Memory)** | **ROM (Read Only Memory)** |
| Data persistence | Volatile, data lost when power off | Non-volatile, data remains when power off |
| Data writing | Data can be written, read, and erased repeatedly | Data written during manufacturing, cannot be changed |
| Access | Fast, random access to stored data | Sequential access, typically for firmware/software |
| Purpose | Temporary storage of data and program instructions | Permanent storage of firmware or low-level system software |

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

|  |  |  |
| --- | --- | --- |
| **Characteristics** | **Static RAM (SRAM)** | **Dynamics RAM (DRAM)** |
| Storage method | Stores data in a static flip-flop circut | Stores data in capacitors |
| Power consumption | Low | High due to refreshes |
| Speed | Faster access times | Slower access time due to refreshes |
| Cost | More expensive | Less expensive |
| Density | Less density | More density |
| Reliability | More reliable, less prone to data loss | Less reliable, more prone to data loss |
| Usage | Often used in cache memory | Often used in main memory |

* 1. What type of memory is typically used in USB thumb drives? Why shouldn’t we rely on this for critical data storage?
* The type of memory is typically used in USB thumb drives is the Flash Memory – EEPROM (Electrically Erasable Programmable Read Only Memory).
* We shouldn’t rely on this for critical data storage due to its limited lifespan, potential data corruption, and risk of physical damage or loss.

1. 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?

- Total number of bytes:

1 GB = 1024 MB

1 MB = 1024 KB

1 KB = 1024 bytes

* 1 GB = = 1,073,741,824 bytes

- Required Number of bits: Using the formula >total number of bits required => >1,073,741,824 bytes => x>21,073,741,824 => x30

So 30 bits are needed to address all bytes in the system’s RAM, but the actual number of bits used for memory addressing might be 32 bits.

1. 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?

- The Von Neumann computing architecture is the most common CPU architecture. It consists of a CPU, memory, and I/O devices, and operates using a fetch-decode-execute cycle. The architecture allows flexibility, efficiency, and scalability but is limited due to its bottleneck and sequential processing.

- The Harvard computing architecture is another one where instructions and data are kept separate. It runs faster (generally) and more secure (generally) then Von Neumann, but more expensive and less extensible. The Harvard architecture is used in PIC controllers, digital signal processors where memory is scarce, and speed is important.

- Differences between Von Neumann and Harvard computing architectures.

|  |  |  |
| --- | --- | --- |
| Characteristics | Von Neumann architecture | Harvard architecture |
| Memory | Single memory space for instructions and data | Separate memory spaces for instructions and data |
| Instruction Fetch | CPU fetches instructions and data from a single memory space | CPU fetches instructions from instruction memory and data from data memory |
| Data Access | CPU accesses data from the same memory space as instructions | CPU accesses data from a separate memory space |
| Performance | Limited by the Von Neumann bottleneck | Can achieve higher performance due to simultaneous access to instructions and data |
| Design Goals | General-purpose computing, flexibility, and scalability | High-performance computing, specialized applications, and digital signal processing |
| Examples | Personal computers, laptops, and mobile devices | Digital signal processors, microcontrollers, and some specialized computers |

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

Cache memory Is a small, fast mrmoty that stores frequently used data or instructions, reducing memory access time ans increasing system performance. Its primary role is to speed up data access by providing quick access to frequently used data and instructions.

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

* - An interrupt is a signal to the processor that an event has occurred and requires immediate attention. It is a way for hardware or software to interrupt the normal execution of a program and request the processor to perform a specific task or service.
* Four common types of interrupts:

+ Hardware Interrupts

+ Software Interrupts

+ Timer Interrupts

+ I/O Interrupts

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

- Polling is a technique where the processor continuously checks the status of a device or a condition to see if itneeds attention. The processor repeatedly asks the device or condition it it is ready or if an event has oocured and if so, it takes action.

- Polling is not conmmonly used because it has several disadvantages:

+ Inefficient use of processor time

+ Slow response time

+ Increased power consumption

1. 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 the Last-In-First\_Out (LIFO) principle, where elements are added and removed from the top. It works by:

+ Pushing: Adding elements to the top of the stack

+ Popping: Removing elements from the top of the stack

* Primary purpose: The primary purpose of a stack is to manage function calls, store temporary data, and implement recursive algorithms. It allows for efficient use of memory and hlpes to prevent data corruption.
  1. How are stacks useful for handling interrupts?

- Stacks are useful for handling interrupts because they provide a way to:

+ Save context: When an interrupt occurs, the current state of the processor (registers, program counter, etc.) Is pushed onto the stack, allowing the processor to save its context.

+ Restore context: When the interrupt Is handled, the saved context is popped from the stack, restoring the processor to its previous state.

* 1. How are stacks useful in programming?

Stacks are useful in programming for providing manage function calls, implementing recursive functions, parsing expressions, evaluating postfix notation, and implementing undo/redo functionality. Therefore, stacks provide a way to manage memory, store temporary data, and implement complex algorithms.

## Provide all the answers to the above questions in your submission document. Practical - Stacks of Stacks !

1. Start Logisim and open a new canvas
2. 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.
3. 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.
4. For your shift register to work as a stack, it needs to be bi-directional. This means the in- put to any Flip Flop could come from two places - the left or the right. In lectures we dis- cussed 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.

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

A diagram of a circuit

Description automatically generated

## Export your circuit as an image and include it in your submission document. Demon- strate your working stack to your lab demonstrator.

1. Modify your stack so that it has the option to read out its contents ***in parallel*** to a sepa- rate 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 re- tain the last state read in (and should have LEDs connected to each Flip Flop out showing its state).

A diagram of a circuit

Description automatically generated

## Export your circuit as an image and include it in your submission document. Demon- strate 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.