COS10004 – Computer Systems

Name: Thanh Tam Vo

ID: 103487596

Submission Lab05

1. Memories:

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

Read-only memory, or ROM, is a kind of memory that serves as a permanent repository for data.

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

Random access memory, or RAM, is a kind of memory that serves as a short-term repository for data. RAM's stored data will be erased when the computer is switched off, unlike ROM, which can retain information when it is turned off.

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

Information is stored in static RAM until the power is turned off. It uses a little amount of power and offers a quick and large silicon area per byte.

Information is stored in dynamic RAM as long as it is updated often enough. It uses less power and has a lower silicon surface area per byte.

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 is typically used in USB thumb drives. Despite the fact that it offers a number of advantages, we shouldn't rely on them excessively because they are somewhat fragile and may be lost if we are not attentive.

2. How many bits are in 1GB of RAM?

1GB has $2^{10} * 2^{10} * 2^{10} = 2^{30}$ bytes

1 byte has 8 bits

So, the number of bits that are required to address all bytes is going to be:

$2^{30} * 8 = 8\ 589\ 934\ 592$ bits

3. Von Neumann and Harvard computing architectures.

The most widely used CPU architecture is von Neumann's design. The fact that data bits and control bits are kept in the same hardware/memory region enables processes to be paused (while data is maintained) so that more important activities can be finished before being resumed.

Harvard architecture is the computer architecture that separates data from instructions. In digital signal processors and PIC controllers, where speed is essential and memory is constrained, control buses and data buses can be of a variety of types and widths.

Differences: Harvard's architecture separates data and instructions while Von Neumann's approach merged them into a single memory space. Although it often functions more quickly and securely than Von Neumann, Harvard design is also more expensive and less expandable.

The Von Neumann architecture is used by the majority of personal computers and general-purpose computers. Microcontrollers and signal processing both use Harvard design.

4. What is cache memory

Cache speeds up and improves the efficiency of accessing, retrieving, and processing by storing frequently used instructions and data in high-speed memory.

5. Interrupt, and four common types.

When the CPU is working on one job and the user suddenly assigns another task, the effect is an interruption.

Types of interrupts:

+ Software Interrupt
+ Hardware Interrupt
+ Level-triggered Interrupt
+ Edge-triggered Interrupt

5.1. Polling

The CPU determines which device needs attention by checking the status or input of all relevant devices using the polling protocol. Polling is simple to set up, but it is rarely used since it takes time and inefficiently wastes CPU cycles.

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

Stacks offer an alternative to memory access that is random (indexed). The ability to mothball, backup, or hibernate a process or task in response to an interrupt or code invocation is made possible by a stack.

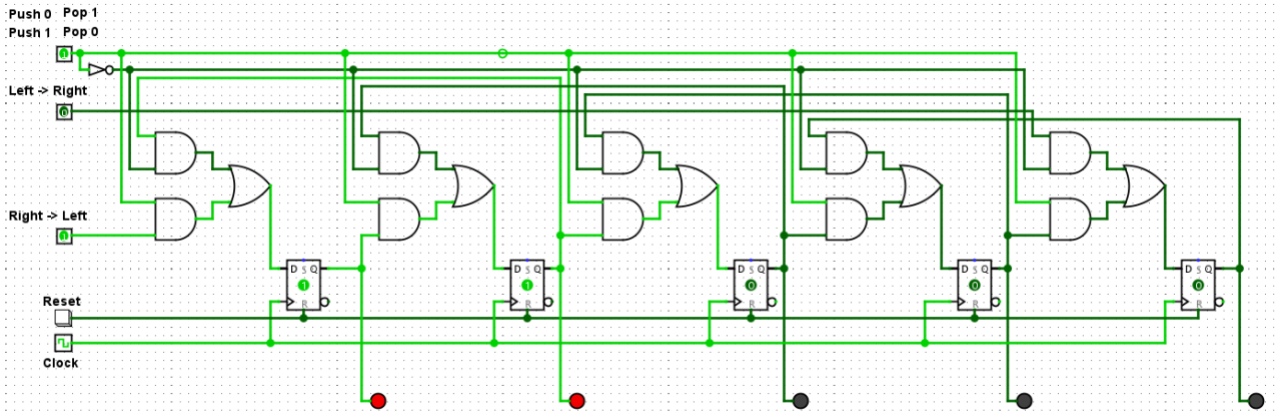
6.1. How are stacks useful for handling interrupts?

Stack is the only possible data structure for safely storing return addresses since Interrupt Handler is a hardware-triggered function call with nesting capabilities.

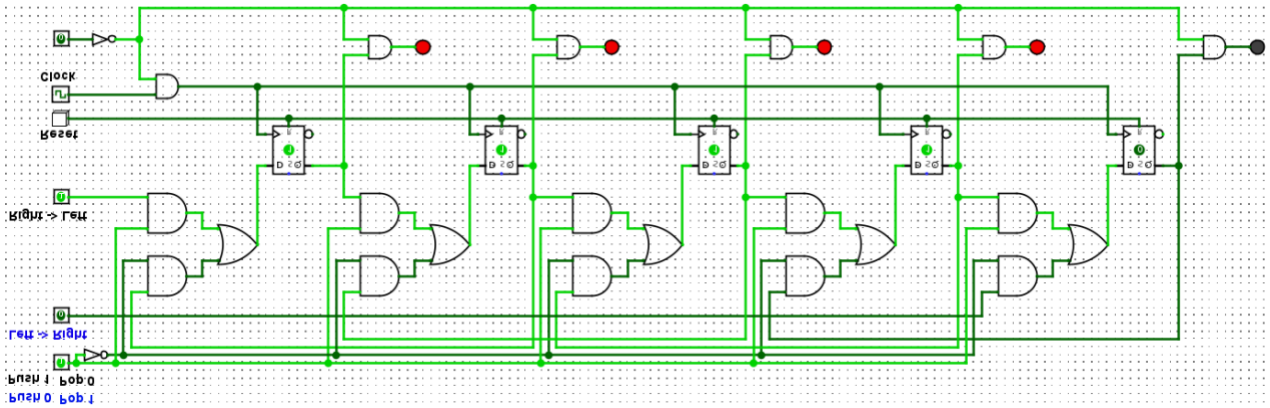
6.2. How are stacks useful in programming?

You may build parsers, functions, evaluate expressions, and do backtracking using stacks.

7. 5-bit deep, 1-bit wide stack



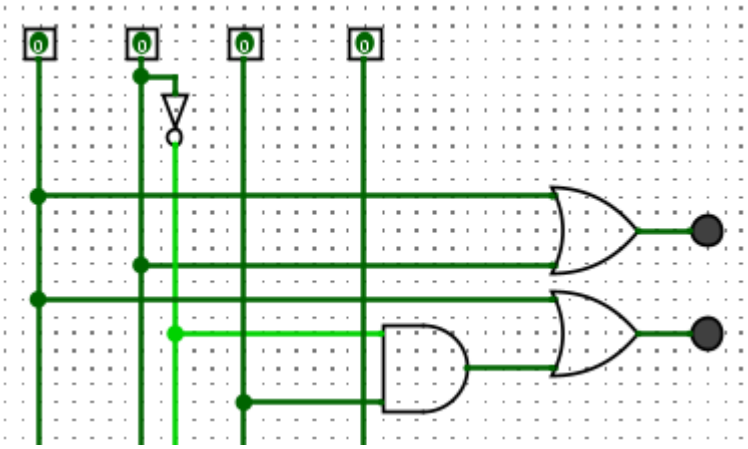
8. Modified circuit



9. Decoder/Encoder

Encoder:

Inputs				Outputs	
D3	D2	D1	D0	Q1	Q0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x



Decoder:

A	B	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	1
1	0	0	0	1	0
1	1	0	0	0	1