Ministerul Educaţiei și Cercetării al Republicii Moldova Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică

COMPUTER ARCHITECTURE

Laboratory work 3:

Simulating digital logic circuits using Logisim

Elaborated:
st. gr. FAF-213 Konjevic Alexandra

Verified:
asist. Univ Vladislav Voitcovschi

Chişinău – 2023

Objective:

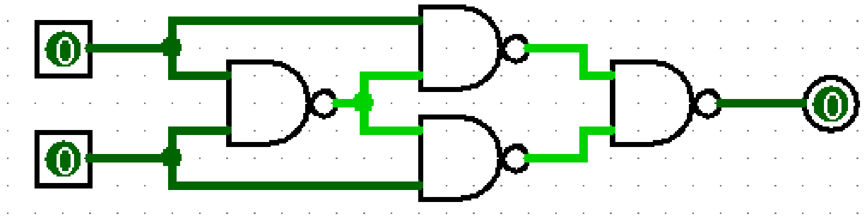
Implement various circuits using a digital logic simulator.

Introduction:

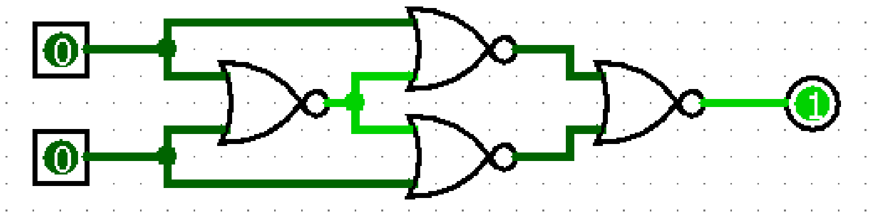
With the use of a graphical user interface, users of the robust digital logic simulator and design tool known as Logisim may construct and simulate digital circuits. A user-friendly environment is provided by this free, open-source program for creating and testing digital logic circuits. Complex circuits that include logic gates, arithmetic circuits, sequential circuits, and other digital components can be built using logic simulation. Using a drag-and-drop interface, users of Logisim may design, edit, and simulate circuits. The software comes with a library of digital components that can be used to construct circuits, but users can also utilize the software's built-in capabilities to design their own unique components.

Tasks:

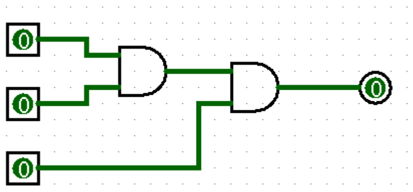
1E. XOR using NAND



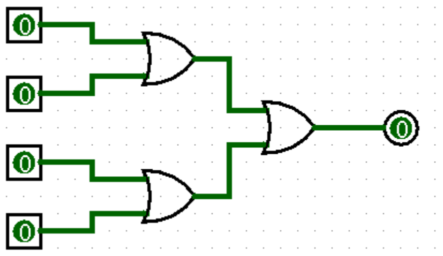
2E. XNOR using NOR



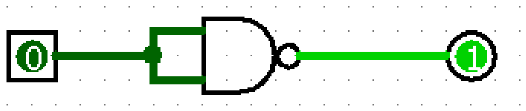
3E. AND with three inputs, using AND



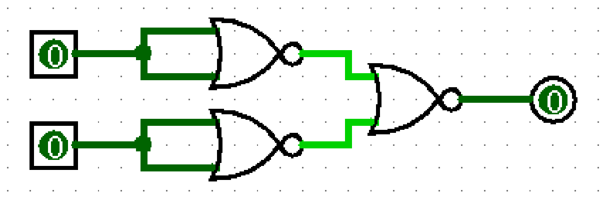
4E. OR with four inputs using OR



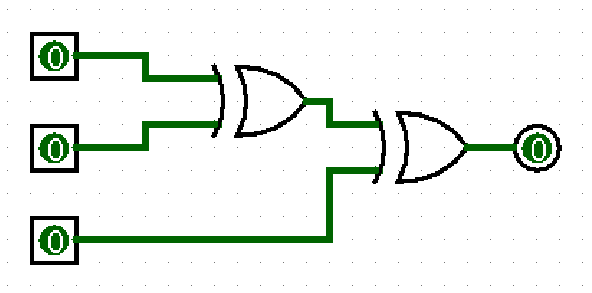
5E. NOT with two inputs using NAND



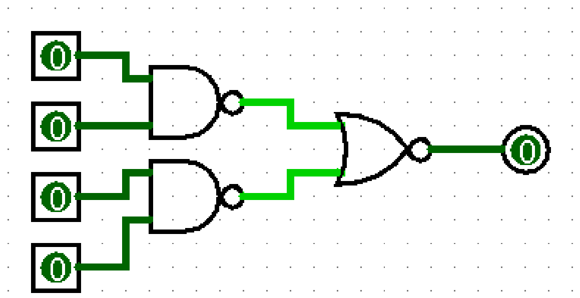
6E. AND using NOR



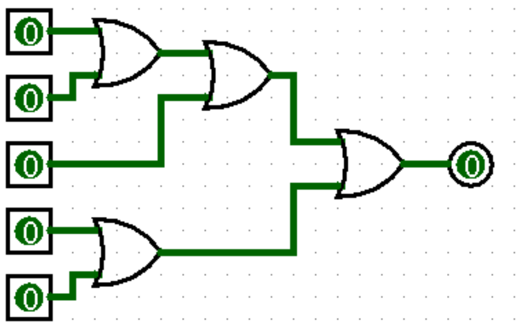
8E. XOR with three inputs using XOR



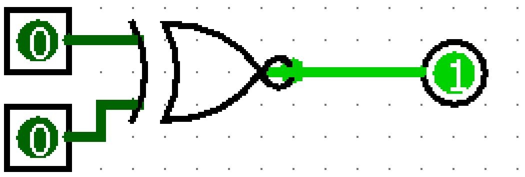
10E. AND with four inputs using NAND



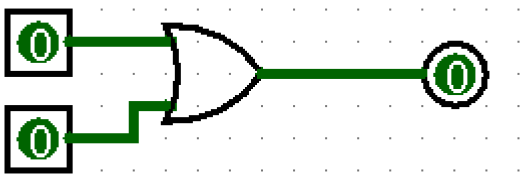
11E. OR with five inputs using OR



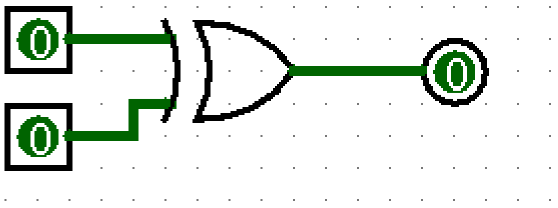
1M. Active output if the two inputs are equal



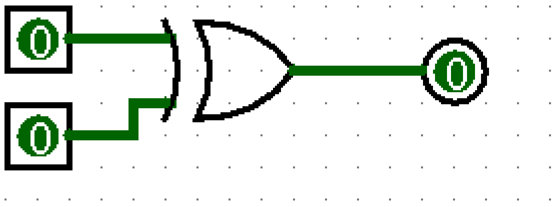
2M. Active output if at least one of the inputs is 1



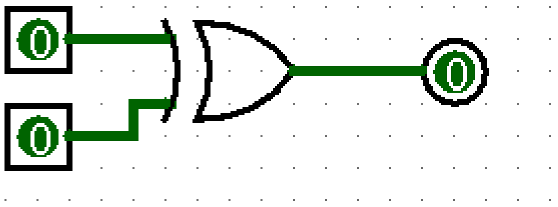
3M. Active output if the two inputs are different



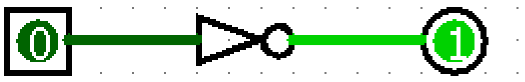
4M. Active output if one input is 1 and another is 0



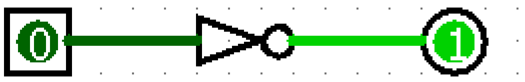
5M. Active output if one input is 0 and another is 1



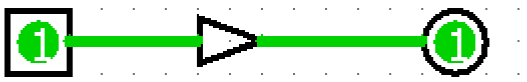
8M. Active output if the input is negated



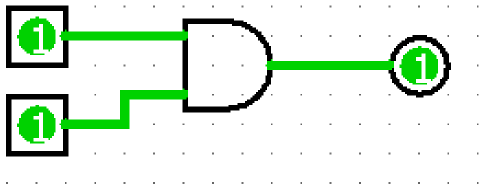
9M. Active output if the input is null



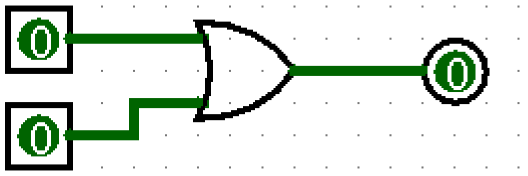
10M. Active output if the input is not null



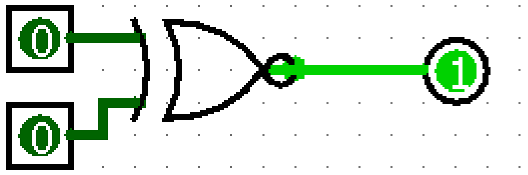
11M. Active output only if both inputs are 1



12M. Active output only if both inputs are 1

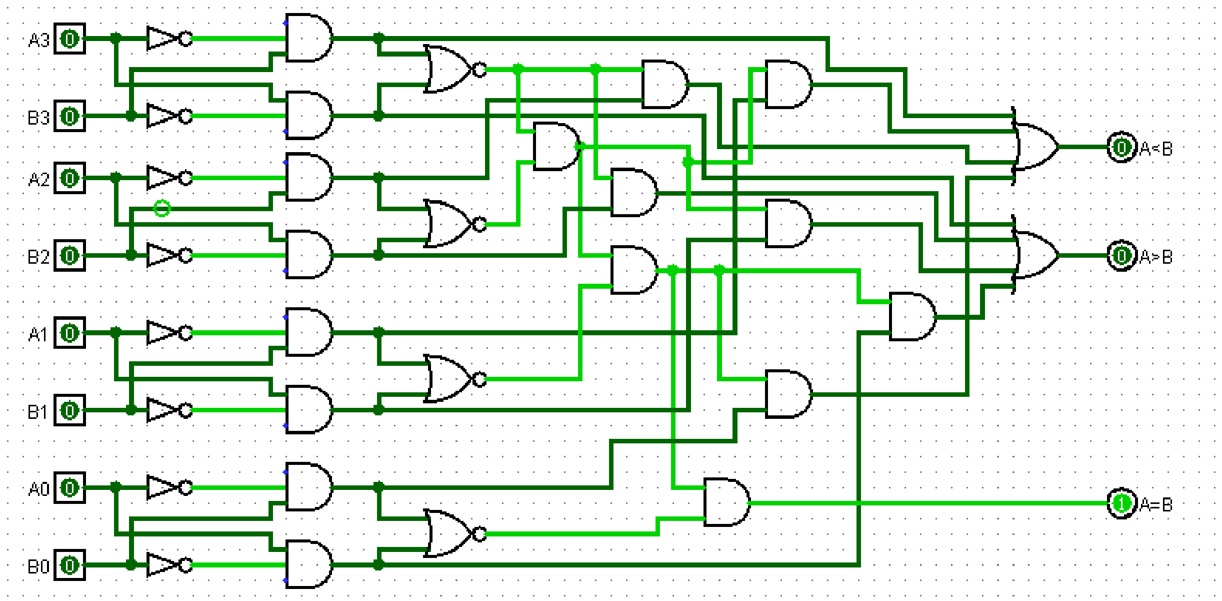


18M. Active output only if both inputs are 1 or both are 0, but not if one is 1 and another is 0

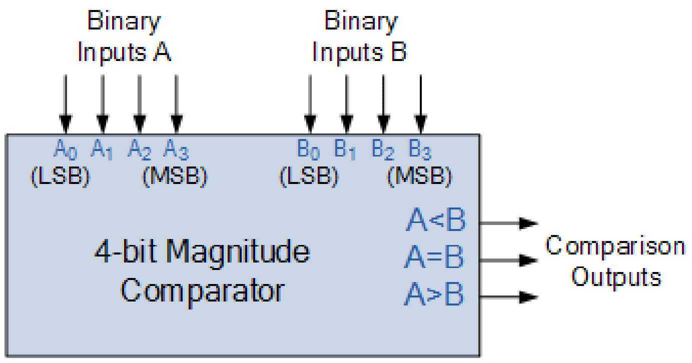


1H. 4 bit comparator using logic gates.

A comparator used to compare two binary numbers each of four bits is called a 4-bit magnitude comparator. It consists of eight inputs each for two four-bit numbers and three outputs to generate less than, equal to, and greater than between two binary numbers.

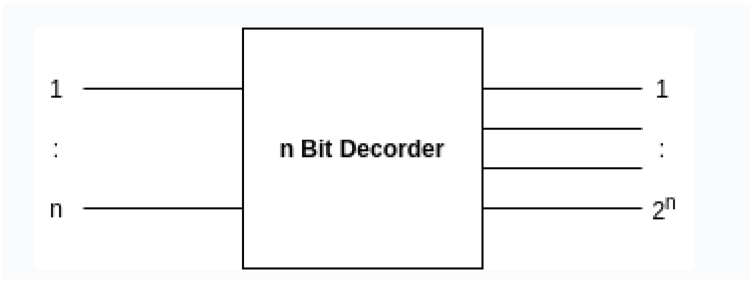


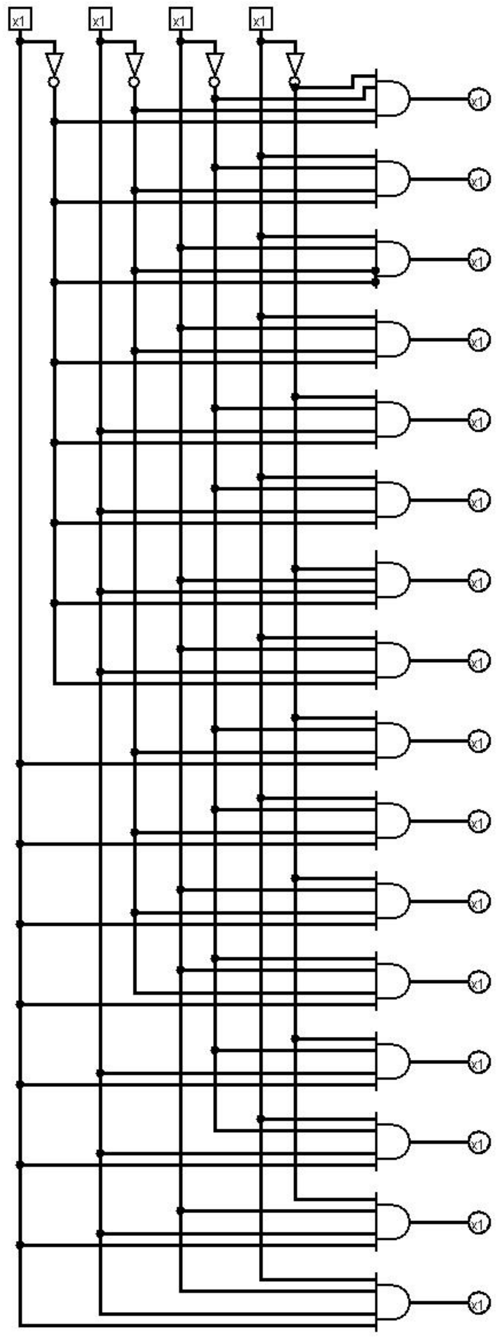
Digital Magnitude Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals present at their input terminals and produce an output depending upon the condition of those inputs. Here, two 4-bit words ("nibbles") are compared to each other to produce the relevant output with one word connected to inputs A and the other to be compared against connected to input B as shown below.



2H. 4 bit decoder with 16 outputs

Decoders are the combinational circuits that detect the presence of some code on its input and indicate the presence of that code by a specified output. Generally, a decoder has n input lines and 2^n output lines.

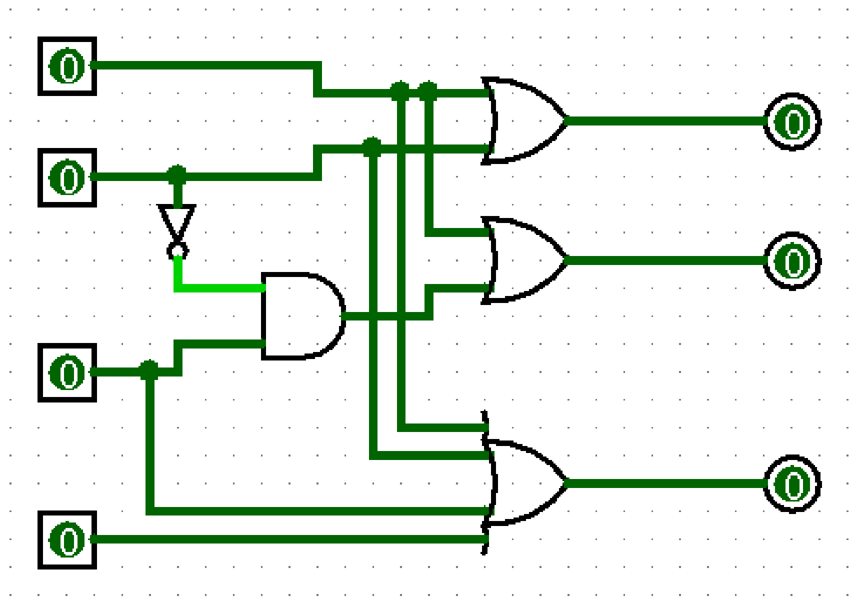




3H. 4 bit priority encoder

A 4-bit priority encoder (also sometimes called a priority decoder). This circuit basically converts the 4-bit input into a binary representation. If the input n is active, all lower inputs (n-1 … 0) are ignored. The following circuit diagram contains two 2-input OR gates, one 4-input OR gate, one 2input AND gate & an inverter. Here AND gate &

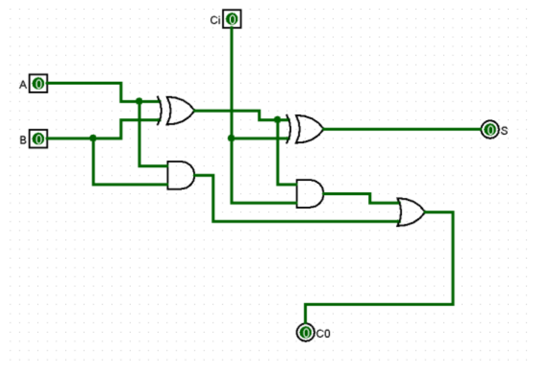
inverter combination are used for producing a valid code at the outputs, even when multiple inputs are equal to '1' at the same time. Hence, this circuit encodes the four inputs with two bits based on the priority assigned to each input.



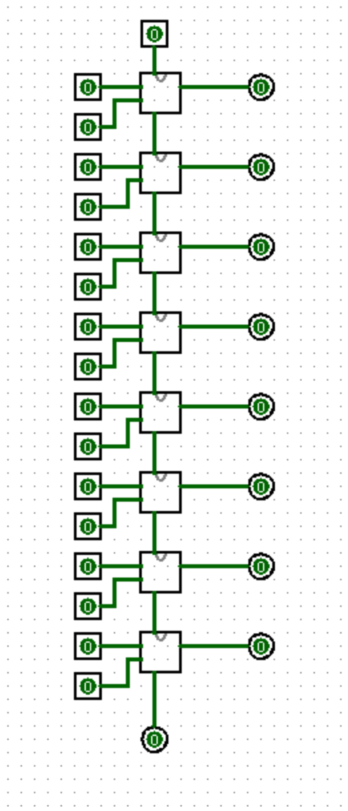
4H. 8 bit full-adder

Full Adder is the adder that adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to another.

First of all, I implemented a full-adder circuit:

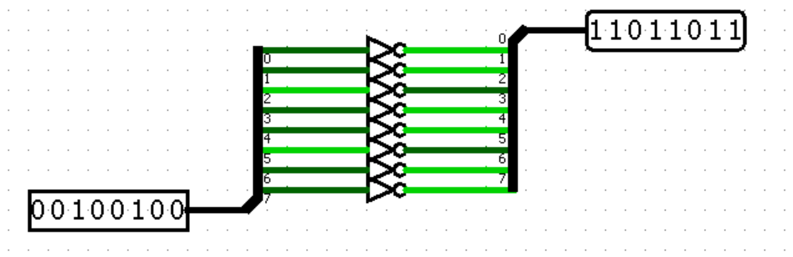


And after that, I used this custom circuit to create an 8 bit full-adder:



11H. Circuit that negates an 8 bit number

The following circuit uses a separator to split a 8 bit number into 8 bits, negates each of them with a not gate, and unions them into a new, negated number.



Conclusion:

In conclusion, the laboratory work provided an invaluable chance to get first-hand experience with digital logic circuits and the modeling program Logisim. I successfully used a variety of various structures throughout the project, including comparator, full-adder, encoder, decoder etcs.

The chance to experiment with various circuit designs and combinations was one of the laboratory work's most important advantages. This gave me the opportunity to witness firsthand how even little modifications to the circuit may significantly alter its behavior. I was able to improve my designs and maximize their performance via trial and error, which is an essential ability in the field of digital logic.