

Ministry of Education, Culture, and Research of the Republic of Moldova

Technical University of Moldova

Department of Software Engineering and Automatics

**Report**

Computer Architecture

Laboratory Work No.2

Done by: Lupașcu Felicia, FAF-212

Checked by: Voitcovschi Vladislav

Chisinau 2023

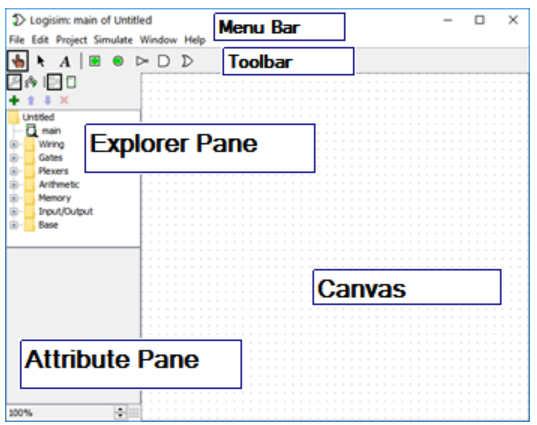
**The purpose of the work:** Each student must perform 5 exercises in the Logisim application. The exercises will start from the number in the catalog and will continue every 5 exercises. My variant is 15: 15,20,25,3,8

**Theoretical considerations:** Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits.

When learning computer architecture and logic circuits, you will need a real-world, graphical example of what you are studying. Text and diagrams only go so far. A helpful tool for designing and simulating logic circuits is Logisim. Because the tool lets you create large circuits from smaller circuits, you can design entire CPUs using Logisim. Furthermore, the tool will run on any computer!

The interface itself is very intuitive and the use of color-coding of wires and elements allows for easy analysis and testing of circuits. You can also save the completed file as an image, or as a .circ file (core to Logisim).

After you have installed Logisim and launched it, take a moment to familiarize yourself with the interface. All of your circuits will be arranged in the canvas as shown in Figure 1.

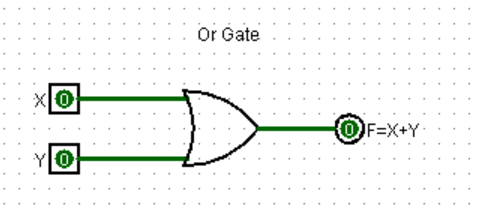


The explorer pane contains all of the elements needed to create circuits (gates, plexers, etc.).In order to add an element from the explorer pane, you select the item in the pane. Next, you click on the canvas to drop the element onto your design. In Figure 2, we've dropped an AND gate onto the canvas.

You'll also notice the attribute pane has now filled up with some options. We can change the direction the gate faces (east, west, north, south), or the number of data bits (the default is 1). As I build circuits, I will be changing the attributes of your elements here.

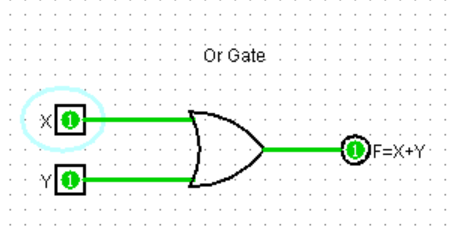
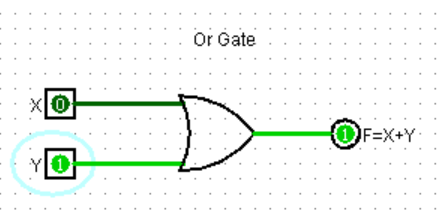
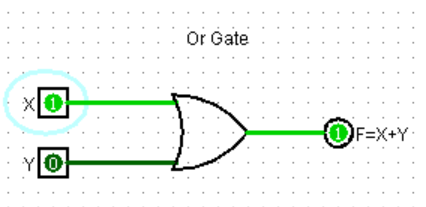
Am varianta numarul 15 si trebuie sa fac urmatoarele exercitii: 15, 20, 25, 3, 8. Voi efectua exercitiile in ordine crescatoare incepand de la exercitiul 3. **Exercițiul 3**. Porți OR: Construiți o poarta OR care să aibă două intrări și să furnizeze un semnal de ieșire "1" atunci când cel puțin una dintre intrări este "1".

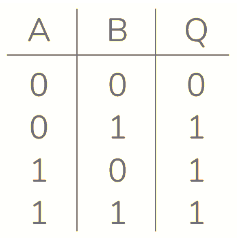
**Exercițiul 3**. Porți OR: Construiți o poarta OR care să aibă două intrări și să furnizeze un semnal de ieșire "1" atunci când cel puțin una dintre intrări este "1".



Exemple:

In exemplele de mai jos sunt reprezentate OR Gates atunci cand cel putin una din intrari este “1”. Din imagini putem observa ca atunci cand cel putin una din intrari este unu, semnalul de iesire, de asemenea, este 1.



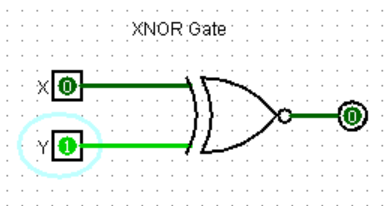
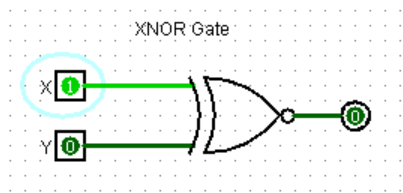


Explicatia functionalitatii circuitului este demonstrata prin acest Truth Table, care arata care sunt valorile de iesire pentru anumite valori de intrare. Schema de circuit furnizată constă dintr-o poartă SAU cu două semnale de intrare, A și B. O poartă SAU este o poartă logică care efectuează disjuncție logică, adică emite un semnal „adevărat” (reprezentat prin 1) dacă cel puțin unul dintre intrările este „adevărat” și un semnal „fals” (reprezentat prin 0) dacă ambele intrări sunt „false”.

In acest circuit am folosit 2 pini pentru valorile de intrare, poarta OR Gate, si un pin pentru valoarea de iesire. Primii 2 pini au fost conectati cu Poarta OR,iar Poarta OR am conectat-o cu al treilea pin.

Prin urmare, funcția acestui circuit este de a determina dacă cel puțin unul dintre semnalele de intrare A sau B este „adevărat” și de a scoate un semnal corespunzător pe ieșirea Y.

**Exercitiul 8** Porți XNOR: Construiți o porți XNOR care să aibă două intrări și să furnizeze un semnal de ieșire "0" atunci când doar o singură intrare este "1".

****

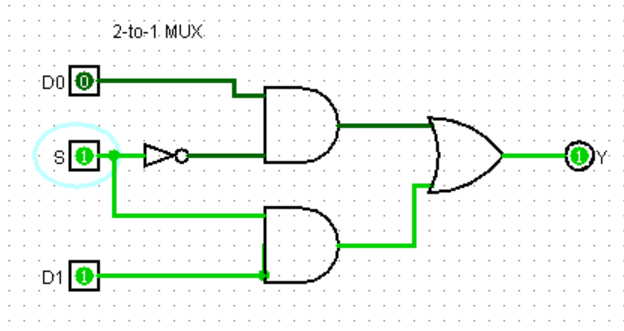
In aceste 2 screen-uri sunt 2 cazuri, cand una dintre portile de intrare = 1 si cand a doua parta de intrare = 1. Demonstrand ca semnalul 0 este furnizat atunci cand doar o singura intrare este = 1.

O poartă XNOR este un tip de poartă logică digitală care primește două intrări și produce o ieșire. Ieșirea porții XNOR este 1 atunci când ambele intrări sunt 1 (adevarat) sau dacă ambele intrări sunt 0 (fals). Daca una dintre intrari este 1 (adevarat) iesirea va fi 0 (fals). Cu alte cuvinte, ambele intrări ar trebui să fie la același nivel logic pentru ca ieșirea să fie 1.

|  |  |  |
| --- | --- | --- |
| X | Y | Output |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

In acest circuit am folosit 2 pini pentru valorile de intrare, poarta XNOR Gate, si un pin pentru valoarea de iesire. Primii 2 pini au fost conectati cu Poarta XNOR,iar Poarta XNOR am conectat-o cu al treilea pin.

**Exercitiul 15** Multiplexor cu 2 intrări: Construiți un multiplexor cu 2 intrări și o intrare de selectare.



Un multiplexor 2-la-1 constă din două intrări D0 și D1, o intrare de selectare S și o ieșire Y. În funcție de semnalul de selectare, ieșirea este conectată la oricare dintre intrări. Deoarece există două semnale de intrare, sunt posibile doar două moduri de a conecta intrările la ieșiri, deci este necesară o singură selectare pentru a efectua aceste operațiuni.

Tabelul de adevăr al multiplexorului 2-la-1 este prezentat mai jos. În funcție de valoarea intrării selectate, intrările, adică D0, D1 sunt produse la ieșiri. Ieșirea este D0 când valoare Selectare este S = 0 și ieșirea este D1 când valoarea Selectare este S = 1.

|  |  |  |  |
| --- | --- | --- | --- |
| S | D0 | D1 | Y |
| 0 | 0 | X | 0 |
| 0 | 1 | X | 1 |
| 1 | X | 0 | 0 |
| 1 | X | 1 | 1 |

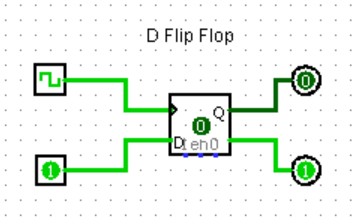
„X” în tabelul de adevăr de mai sus denotă o condiție “Don’t care”. Deci, ignorând condițiile “Don’t care”, putem deriva expresia booleană a unui multiplexor tipic 2 la 1 după cum urmează: 

In acest circuit am folosit 3 pinuri cu valori de intrare. Pinul S este conectat la o poarta NOT, apoi pinul S si D0 sunt conectat la poarta AND GATE. Acelasi lucru si mai jos. Pinul S e conectat la NOT GATE si apoi S si D1 conectati la o alta Poarta AND. Ulterior, am conectat portile AND cu poarta OR. La finalul circuitului este un alt pin “Y” care reprezinta valoarea de iesire.

**Exercitiul 20** Flip-flop D: Construiți un flip-flop D.

Un flip-flop este un tip de element de circuit digital care poate stoca un bit de informație (un singur bit de date) și poate fi utilizat într-o varietate de aplicații, cum ar fi înregistrarea de date, contoare, înregistrare și altele. .

Flip-flop-ul D este numit astfel, deoarece sunt un pin de intrare numit "D" (sau "data") care permite stocarea valorii de intrare. Flip-flop-ul D are, de asemenea, un pin de intrare numit "CLK" (sau "clock") care controlează momentul la care valoarea de intrare va fi stocată în flip-flop. Când semnalul de ceas "CLK" se schimbă, flip-flop-ul D va "captura" valoarea pe pinul "D" și va stoca această valoare la ieșirea "Q" a flip-flop-ului. În același timp, flip-flop-ul D va furniza o valoare complementară "Q̅" (adesea reprezentată prin "not Q") la ceaaltă ieșire.



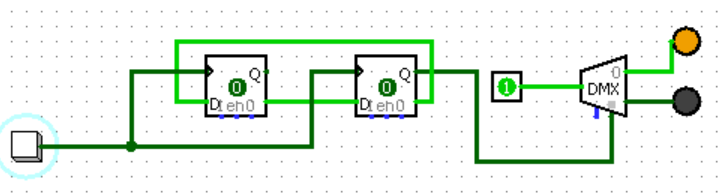
In acest circuit am folosit un Clock, 3 pini, si un component D Flip Flop din Logisim. Flip-flop-ul D este un flip-flop cu două intrări. Intrările sunt intrarea de date (D) și intrarea ceasului (CLK). Ceasul este un impuls de sincronizare generat de echipament pentru a controla operațiunile. Flip-flop-ul D este folosit pentru a stoca date la un moment prestabilit și pentru a le menține până când este nevoie. Acest circuit este uneori numit un flip-flop cu întârziere. Cu alte cuvinte, intrarea datelor este întârziată cu până la un impuls de ceas înainte de a fi văzută la ieșire.

**Exercitiul 25** Circuit pentru comutarea a două LED-uri: Construiți un circuit care să permită comutarea a două LED-uri cu ajutorul unui buton

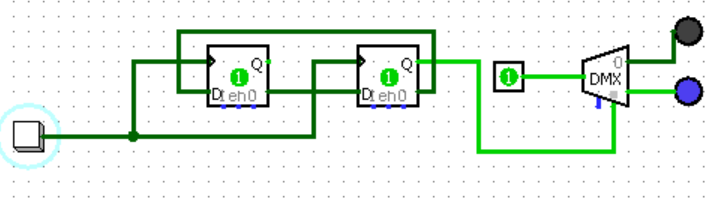
Un LED are un singur pin, o intrare de 1 bit care este folosită pentru a determina dacă LED-ul trebuie afișat colorat (când intrarea este 1) sau întunecat (când intrarea este orice altceva).

Un buton are un singur pin, o ieșire de 1 bit, care este 0, cu excepția cazului în care utilizatorul apasă butonul folosind Poke Tool, când este 1. Când butonul mouse-ului este apăsat, ieșirea componentei va fi 1. La eliberarea butonului mouse-ului, ieșirea revine la 0.

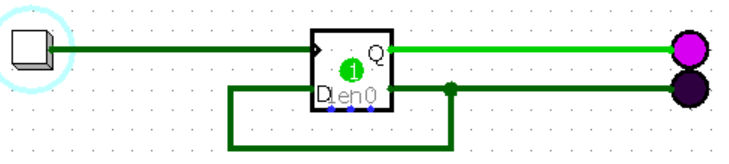
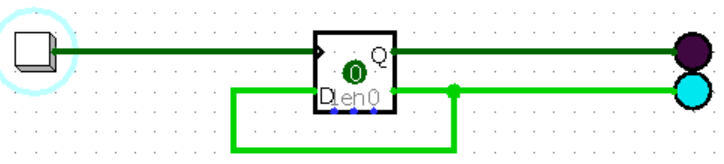
Cazul 1:



Cazul 2:



Pentru a implementa aceasta conditie, am folosit un buton, 2 leduri, 2 flip flop D, un pin si un demultiplexer. Butonul este conectat cu ambele D Flip Flop , si acestea sunt conecctate intraconectate. Si apoi sunt conectate la Demultiplexer, care la randul sau este conectat la leduri si la un pin. Când butonul este apăsat, valorile stocate în flip-flop-uri se schimbă, ceea ce duce la schimbarea stării de ieșire a demultiplexorului. Acest lucru determină aprinderea unui LED și stingerea celuilalt LED. Si astfel are loc si comutarea ledurilor. Anterior am schimbat culorile ledurilor.

**Varianta 2** de implimentare a conditiei: O implimentare simplificata care consta doar dintr-un un buton, 2 leduri, un D flip flop. LED-urile sunt conectate la ieșirea D-FF. Aceasta înseamnă că, atunci când D-FF este în starea 0, primul LED va fi stins, iar al doilea LED va fi aprins. În schimb, atunci când D-FF este în starea 1, primul LED va fi aprins, iar al doilea LED va fi stins. Butonul este conectat la intrarea D-FF. Atunci când butonul este apăsat, semnalul de intrarea D-FF se schimbă, determinând D-FF să schimbe starea. În acest fel, LED-urile se vor aprinde și stinge în funcție de starea D-FF.

**Concluzie** În concluzie, experiența mea de inițiere în aplicația Logisim a fost una foarte interesantă și educativă. Am învățat despre funcționarea elementelor de logică digitală, cum ar fi ușile logice, multiplexarele, flip-flop-urile și am înțeles cum se pot conecta aceste elemente pentru a crea circuite logice mai complexe.

De asemenea, am dobândit cunoștințe practice despre modul în care se pot utiliza LED-urile și butoanele pentru a interacționa cu circuitele digitale, ceea ce mi-a permis să văd cum funcționează într-un mod practic.

În urma acestui laborator, am înțeles că aplicația Logisim poate fi o unealtă puternică pentru proiectarea și simularea circuitelor digitale. Aceasta poate fi folosită pentru a proiecta și a testa circuite logice înainte de a fi implementate într-un mediu hardware real.

În concluzie, am obținut o perspectivă mai profundă asupra funcționării circuitelor digitale și am acum competențele necesare pentru a proiecta și a simula circuite logice simple în aplicația Logisim. Aceste abilități ar putea fi utile pentru proiectarea de proiecte practice sau pentru aprofundarea cunoștințelor despre logica digitală.