



**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII
AL REPUBLICII MOLDOVA Universitatea Tehnică a
Moldovei Facultatea Calculatoare, Informatică și
Microelectronică Departamentul Inginerie Software și
Automatică**

Copta Adrian | FAF-223

Raport

Lucrare de laborator n.2

Arhitecturi de calculatoare

Verificat:

Voitcovschi Vladislav *asist.univ*

1. Scopul lucrării:

Fiecare student/a trebuie sa efectueze cate 5 exerciții în aplicația Logisim. Exercițiile vor începe de la numărul din catalog și vor continua din 5 în 5 exerciții.

2. Introducere:

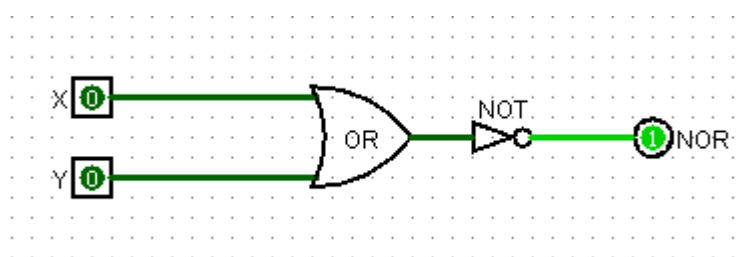
Logisim este o unealtă educațională folosită pentru proiectarea și simularea circuitelor digitale. Cu o interfață simplă în bara de instrumente și simularea circuitelor pe măsură ce sunt construite, este suficient de accesibil pentru a facilita învățarea celor mai fundamentale concepte legate de circuitele logice. Datorită capacității de a construi circuite mai mari din subcircuite mai mici și de a trasa fascicule de fire cu o singură mișcare a mouse-ului. Logisim poate fi folosit (și este folosit) pentru a proiecta și simula întregi unități centrale (CPU-uri) în scopuri educaționale.

Studentii de la colegii și universități din întreaga lume utilizează Logisim în diverse scopuri, printre care:

- Un modul în cursurile generale de informatică
- O unitate în cursurile de organizare a calculatoarelor la nivelul al doilea de studiu
- Pe parcursul unui semestru întreg în cursurile superioare de arhitectură a calculatoarelor

3. Exerciții executate:

Exercițiul 5: Porți NOR: Construiți o poartă NOR care să aibă două intrări și să furnizeze un semnal de ieșire "0" atunci când cel puțin una dintre intrări este "1".



Folostind o poarta OR care primeste doua intrati: X si Y, și o poarta NOT am comutat o poartă NOR care furnizează un semnal de ieșire "0" atunci când cel puțin una dintre intrări este "1".

Combinational Analysis

File Edit Project Simulate Window Help

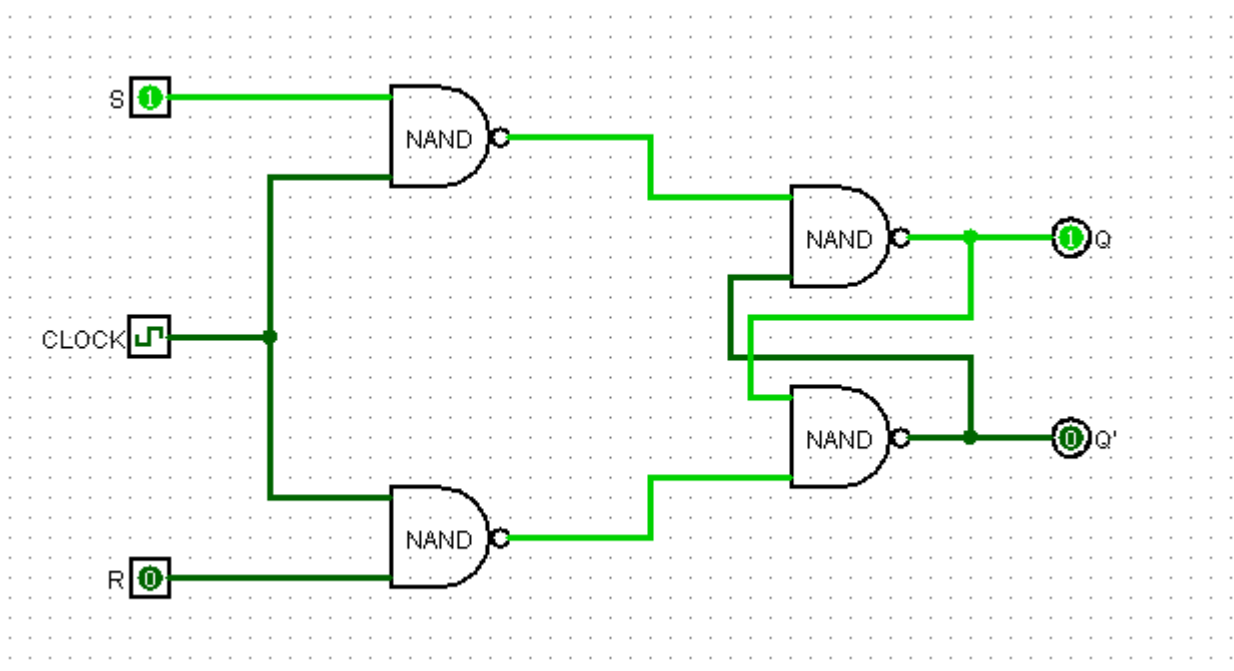
Inputs Outputs **Table** Expression Minimized

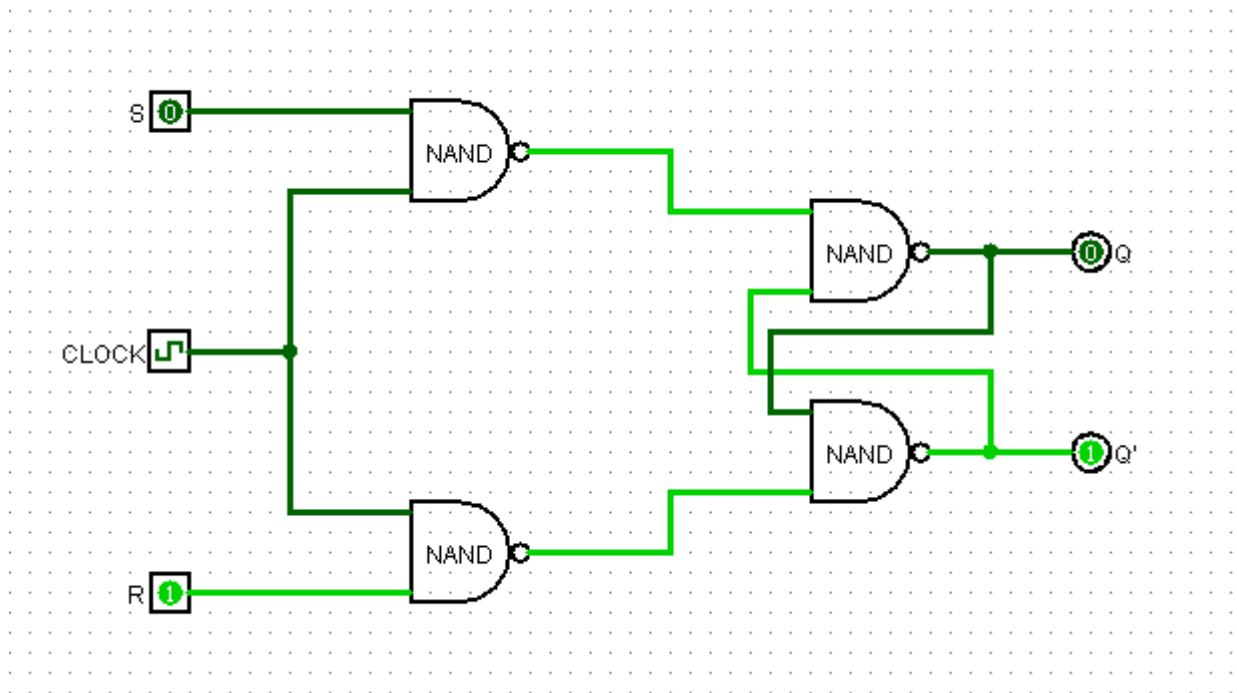
X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Build Circuit

Iar pentru a verifica dacă circuitul nostru intradevar este o poartă NOR, accesam: >Project >Analyze Circuit >Table, și respectiv observam ca poarta noastra furnizeze un semnal de ieșire "0" mereu când cel puțin una dintre intrările X sau Y este "1".

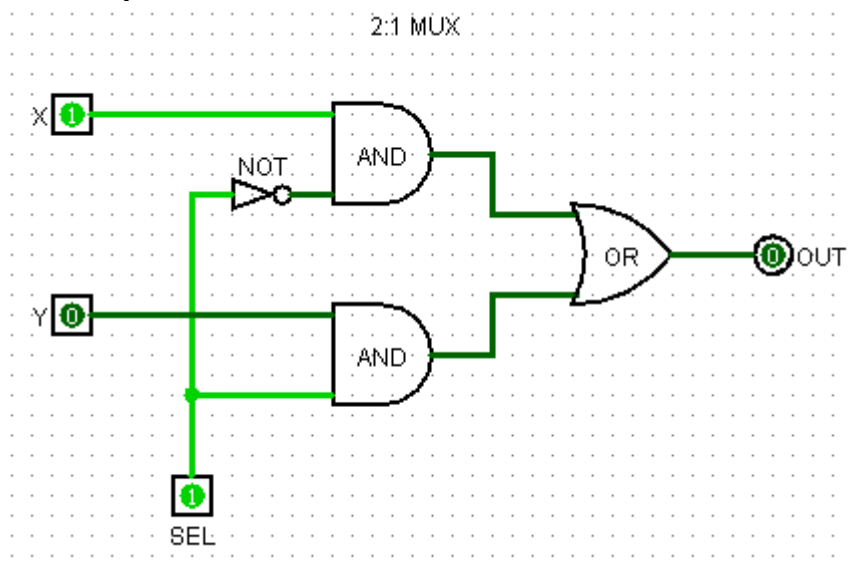
Exercitiul 10: Flip-flop SR: Construiți un flip-flop SR care să rețină starea de ieșire atunci când sunt furnizate semnale de setare și resetare.

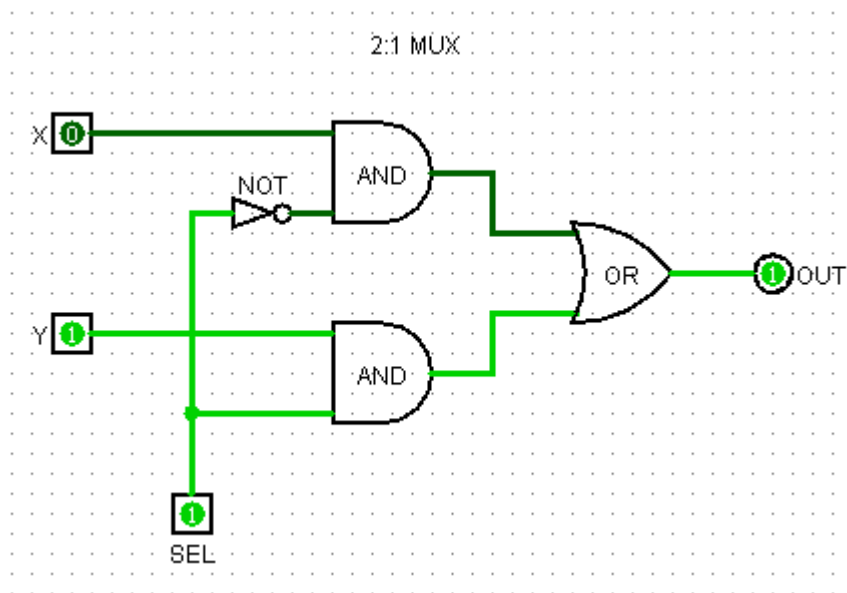




Acest Flip-flop SR folosește un SR NAND latch. Porțile NAND suplimentare inversează și mai mult intrările, astfel încât un latch SR devine un latch SR cu poartă. Când Setul este egal cu 1, la comutarea ceasului, Q de asemenea va fi 1, iar la setarea resetului 'R' cu valoare 1 și 'S' cu 0, Q' va afișa valoarea 1.

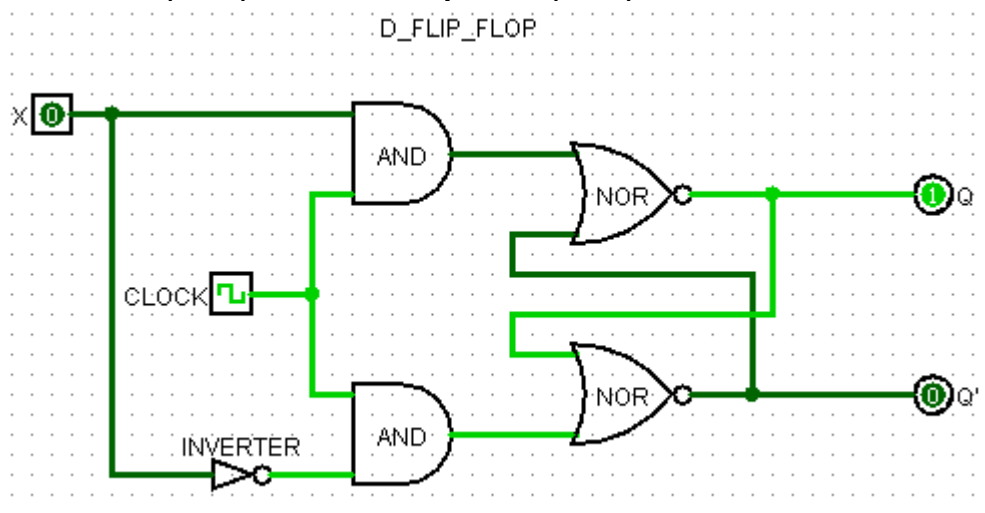
Exercițiul 15: Multiplexor cu 2 intrări: Construiți un multiplexor cu 2 intrări și o intrare de selecție.





În multiplexorul 2×1 , există doar două intrări, și anume, X și Y, 1 linie de selecție, adică SEL și ieșiri simple, adică OUT. Pe baza combinației de intrări care sunt prezente la linia de selecție SEL, una dintre aceste 2 intrări va fi conectată la ieșire.

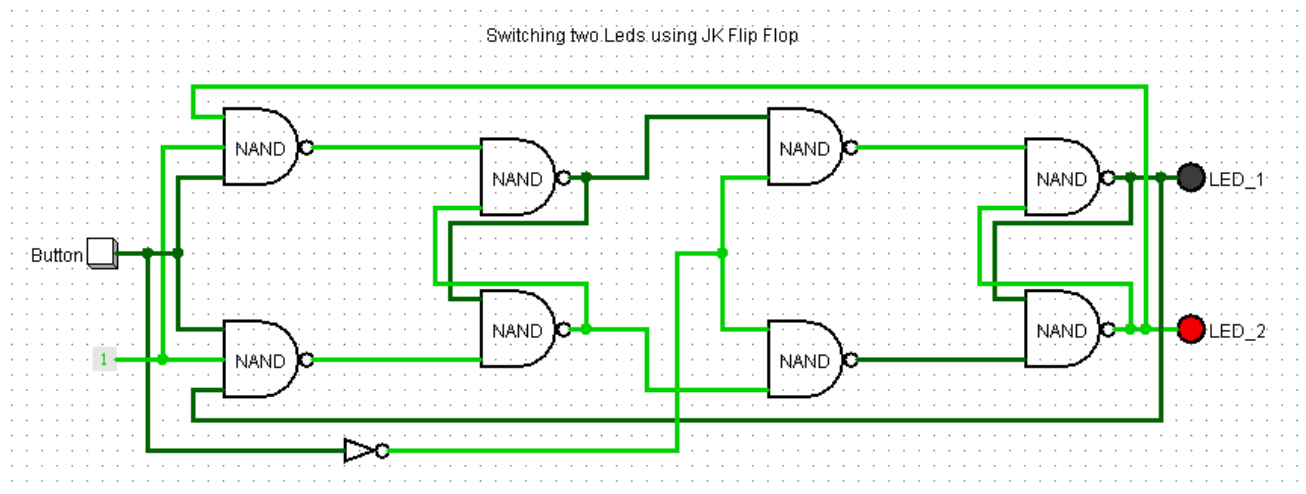
Exercițiul 20: Flip-flop D: Construiți un flip-flop D.



Flip-flop D constă dintr-o singură intrare X și două ieșiri (Q și Q'). Funcționarea de bază a Flip-flop D este după cum urmează:

1. Când semnalul de ceas este scăzut, flip-flop-ul își menține starea curentă și ignoră intrarea D.
2. Când semnalul de ceas este ridicat, flip-flop-ul prelevează și stochează intrarea D.
3. Valoarea care a fost introdusă anterior în intrarea D este reflectată la ieșirea Q a flip-flop-ului.
 - Dacă $D = 0$, atunci Q va fi 0.
 - Dacă $D = 1$, atunci Q va fi 1.
4. Ieșirea Q' a flip-flopului este completată de ieșirea Q.
 - Dacă $Q = 0$, atunci Q' va fi 1.
 - Dacă $Q = 1$, atunci Q' va fi 0.

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



Pentru a comuta doua leduri folosind un buton, am construit un JK Flip Flop, care primește ca intrare 'J' și 'K' o constanta '1'. În urma verificarii acestui circuit, am constatat eroarea "Oscillation apparent".

Pentru a preveni aceste glitch am modificat circuitul nostru după configurația "The Master-Slave Configuration" (circuitul din imagine), care cunoscută și sub denumirea de "Configurația Master-Slave," este folosită pentru a preveni problema "glitch-urilor" care pot apărea în timpul tranzițiilor rapide ale semnalului de intrare. Această configurație utilizează două JK Flip-Flop-uri conectate în serie, unde primul flip-flop este "Master," iar al doilea este "Slave."

Master Flip-Flop (Primul Flip-Flop): Acesta răspunde la tranziția semnalului de intrare (J, K, care sunt mereu '1'). Când semnalul de intrare se modifică, acest flip-flop stochează temporar noua stare și devine "Master" pentru o perioadă scurtă de timp. În timp ce primul flip-flop este în modul "Master," intrările sale (J, K) sunt blocate, iar ieșirile nu se schimbă.

Slave Flip-Flop (Al Doilea Flip-Flop): În timpul perioadei când primul flip-flop este "Master," al doilea flip-flop este în modul "Slave." În această perioadă, al doilea flip-flop nu răspunde la schimbările semnalului de intrare. Odată ce primul flip-flop și-a terminat operația și a devenit "Slave," al doilea flip-flop preia noul stadiu și devine "Master" pentru o altă tranziție.