

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.3

Done by: Lupașcu Felicia, FAF-212

Checked by: Voitcovschi Vladislav

Chisinau 2023

**Scopul laboratorului:** Laboratorul 3 reprezinta un ansamblu de exercitii in LogiSim.

Fiecare nivel de exercitii are ponderea sa.

* Easy – 0.2 p
* Medium – 0.4 p
* Hard – 0.8 p

Necesita a fi executate exercitii din toate 3 nivele, iar scopul acestui laborator este de a acumula 10 puncte.

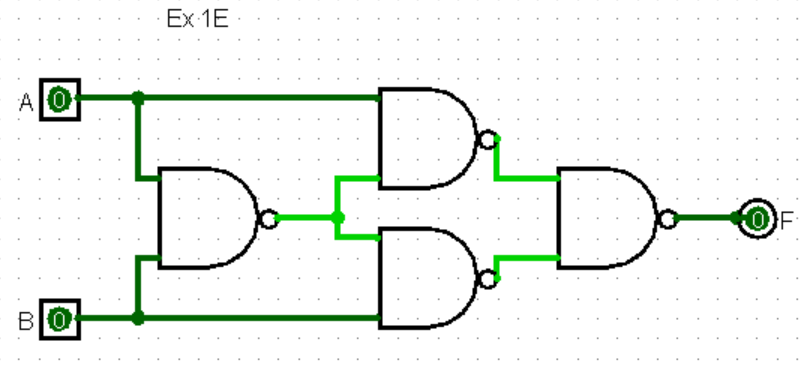
Pentru acest laborator am ales sa execut exercitii de la toate nivelele:

**EASY** : 1E, 2E, 5E, 6E, 10E

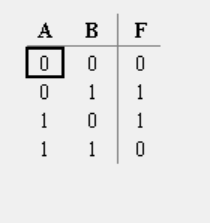
**MEDIUM**: 1M, 2M, 3M , 4M, 5M, 6M, 7M, 8M, 9M, 10M, 11M, 12M, 13M, 14M, 15M, 16M, 17M, 18M, 19M

**HARD**: 1H, 2H, 11H

**Exercițiul 1E**. 1. Implementați o poartă XOR cu două intrări utilizate poarta NAND.

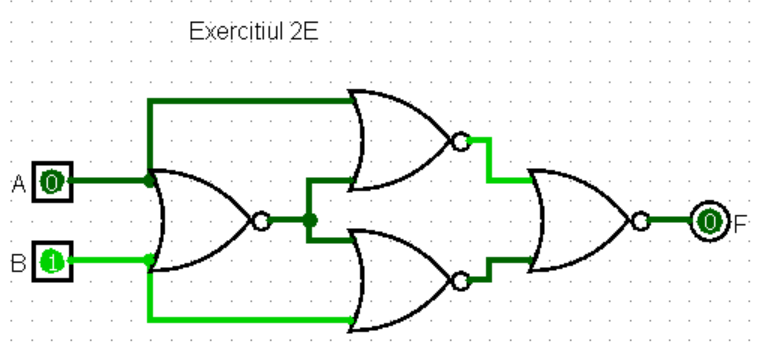


Truth Table generat de Logisim pentru acest circuit.



Astfel, F este ieșirea porții NAND4, care este conectată la ieșirile porților NAND2 și NAND3. Acestea din urmă primesc semnalele de intrare de la porțile NAND1, NAND2 și NAND3, care la rândul lor au semnalele de intrare A și B. Iesirea este activata atunci cand valorile initiale difera una de cealalta, iar atunci cand sunt egale, outputul este 0.

**Exercitiul 2E** Implementați o poartă XNOR cu două intrări folosind poarta NOR.

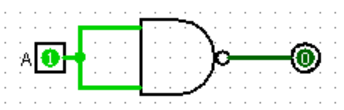


Circuitul descris anterior implementează o poartă XNOR cu 2 intrări folosind 4 porți NOR.



Poarta XNOR este o poartă logică cu două intrări care produce o ieșire "adevărat" (1) atunci când ambele intrări sunt egale (adicatea ambii biti sunt fie ambii 1, fie ambii 0), și o ieșire "fals" (0) în caz contrar.Pentru a implementa această funcționalitate cu 4 porți NOR, circuitul se bazează pe proprietatea dualității logice care afirmă că, dacă înlocuim fiecare poartă logică cu poarta sa duală și inversăm toate intrările și ieșirile, vom obține aceeași funcționalitate logică.

**Exercitiul 5E** Implementați o poartă NOT cu două intrări folosind poarta NAND.

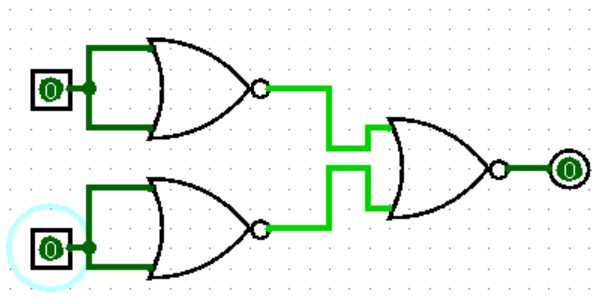


NAND este o poartă logică având două sau mai multe terminale de intrare și oferă o ieșire în funcție de combinația semnalelor de intrare. NAND reprezintă NOT + AND, adică produce o ieșire care este inversarea sau complimentul operațiunii logice AND.

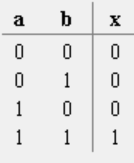


Din figură, este clar că pentru a realiza poarta NOT folosind poarta NAND, trebuie să unim cele două terminale de intrare ale porții NAND pentru a forma un singur terminal de intrare a porții NOT, iar ieșirea porții NOT este luată de la terminalul de ieșire al porții NAND. In acest table este definite functionalitatea circuitului pentru valori difertie ale inputului:

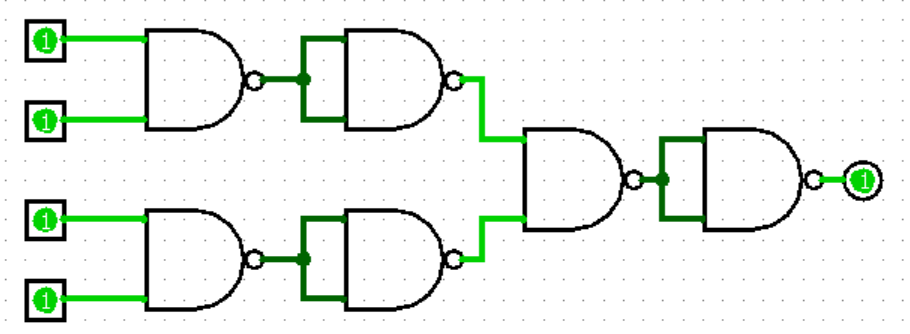
**Exercitiul 6E** Implementați o poartă AND cu două intrări folosind poarta NOR.

****

Porțile NOR logice sunt disponibile folosind circuite digitale pentru a produce funcția logică dorită și primesc un simbol a cărui formă este aceea a unei porți SAU standard cu un cerc, uneori numită „bulă de inversare” la ieșire pentru a reprezenta simbolul porții NOT cu funcţionarea logică a porţii NOR.

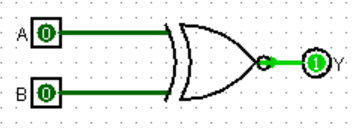


**Exercitiul 10E**  Implementați o poartă AND cu patru intrări folosind poarta NAND

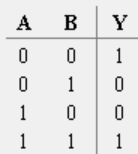
****

Pentru acest circuit , pentru a crea poarta AND cu 4 intrari folosing poarta NAND am utilizat 6 porti NAND si 4 pini de intrare. Comutand valorile, observam ca poarta de iesire devine activa doar atunci cand toate 4 intrari sunt 1. In celelalte cazuri iesirea este egala cu 0.

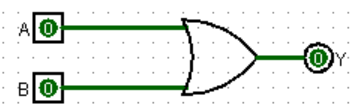
**Exercitiul 1M** Implementați un circuit logic care va activa ieșirea dacă două intrări sunt egale.



Pentru implimentarea acestui circuit am folosit 3 pinuri (2 de intrare si 1 de iesire) si poarta logica XNOR, care activeaza iesirea (valoarea 1) atunci cand intrarile sunt egale. Acesta este tabelul de adevar generat de Logisim care demonstreaza ca atunci cand intrarile sunt 0,0 sau 1,1 iesirea este activata.



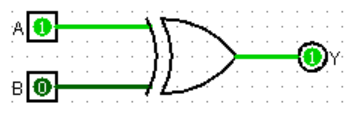
**Exercitiul 2M** Implementați un circuit logic care va activa ieșirea dacă cel puțin una dintre două intrări este 1.



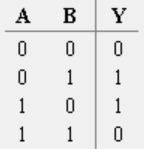
Pentru implimentarea acestui circuit am folosit 3 pinuri si poarta logica OR cu 2 intrari. SAU poarta efectuează o operație logică OR care înseamnă că ieșirea este 1 dacă cel puțin una dintre intrări este 1. La fel ca și poarta AND o poartă OR poate avea, de asemenea, două sau mai multe numere de intrări, dar numai o ieșire. Numai dacă toate intrările sunt 0, ieșirea este neactivată sau 0 și în toate celelalte condiții de intrare ieșirea va fi activată sau 1.



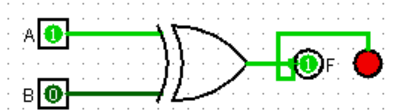
**Exercitiul 3M** Implementați un circuit logic care va activa ieșirea dacă două intrări sunt diferite.



Poarta XOR este o poartă logica digitală care implementează operația de disjuncție exclusivă; abrevierea XOR provine din engleză de la expresia „exclusive OR” („SAU” exclusiv). Ea se comportă conform tabelului de adevăr alăturat. Un rezultat 1 apare numai dacă una și numai una dintre intrările porții e 1. Tabelul de adevar ofera o reprezentare si a celorlalte cazuri.



**Exercitiul 4M** Implementați un circuit logic care va activa ieșirea dacă una dintre două intrări este 1 și cealaltă este 0.

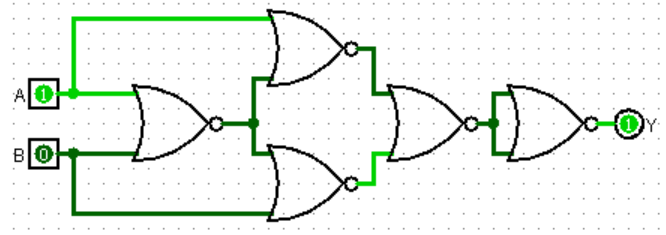


Pentru aceasta conditie am avut nevoie de o poarta XOR. Am adaugat un led, care ar putea fi o alternativa pentru reprezentarea activarii portii de iesire. Tabelul de adevar ofera o reprezentare si a celorlalte cazuri.

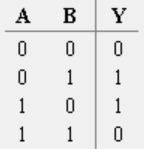
Expresia: ~A B + A ~B



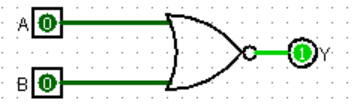
**Exercitiul 5M** Implementați un circuit logic care va activa ieșirea dacă una dintre două intrări este 0 și cealaltă este 1.



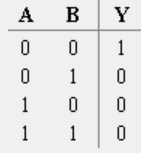
Pentru acest exercitiu am avut nevoie de o poarta XOR. Am creat aceasta poarta folosind 5 porti NOR cu 2 inputs. Respectiv, am reprezentat aceasta poarta intr-un mod mai complex, insa care are acelasi rezultat. Poarta se va activa doar cand o intrare este 1 si cealalta 0.



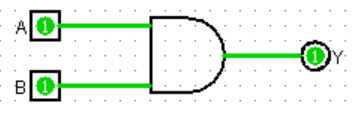
**Exercitiul 6M** Implementați un circuit logic si va activa ieșirea dacă ambele intrări sunt 0.



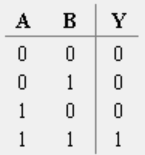
Pentru acest circuit am folosit 3 pini (2 intrare, 1 iesire) si Poarta NOR. Poarta logică NOR este o combinație de poartă logică digitală OR și o poartă inversoare sau NOT care sunt conectate în serie. Poarta inclusivă NOR (NOT-OR) are o ieșire care este în mod normal la nivelul logic "1" și merge "LOW" la nivelul logic "0" atunci când **oricare**dintre intrările sale este la nivel logic "1". Poarta l**ogică NOR**este forma inversă sau “complementară“ a porții OR inclusiv.



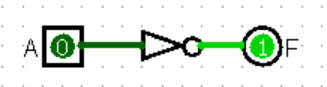
**Exercitiul 7M** Implementați un circuit logic va activa ieșirea dacă ambele intrări sunt 1.



Pentru acest circuit am folosit 3 pini (2 intrare, 1 iesire) si Poarta AND. O poartă logică AND este un tip de poartă logică digitală a cărei ieșire merge HIGH la un nivel logic 1 când toate intrările sale sunt HIGH. Starea de ieșire a unei "porți logice AND" este "LOW" când **oricare** dintre intrările sale este la un nivel logic "0". Cu alte cuvinte, pentru o poartă logică AND, orice intrare LOW va da o ieșire LOW.



**Exercitiul 8M** Implementați un circuit logic care va activa ieșirea dacă intrarea este negată.

****

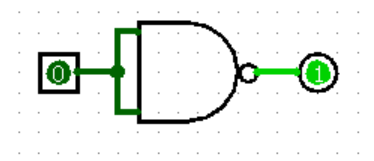
Poarta logică NOT este cea mai de bază dintre toate porțile logice și este adesea menționată ca un Buffer Inversor sau pur și simplu un inversor. Ea este un dispozitiv cu o singură intrare care are un nivel de ieșire normal la nivelul logic "1" și merge "LOW" la un nivel logic "0" când intrarea este la nivelul logic "1", cu alte cuvinte "inversează" (complementează) semnalul său de intrare. Ieșirea de la o poartă NOT este "HIGH" atunci când intrarea sa este la nivelul logic "0", oferindu-ne expresia booleană: Ᾱ = F.

**Exercitiul 9M** Implementați un circuit logic care va activa ieșirea dacă intrarea este nulă.

****

Poarta care activeaza iesirea cand intrarea este nula este poarta NOT- un dispozitiv cu o singură intrare care are un nivel de ieșire normal la nivelul logic "1" și merge "LOW" la un nivel logic "0" când intrarea sa este la nivelul logic "1", cu alte cuvinte "inversează "(complementează) semnalul său de intrare. Ieșirea de la o poartă NOT este "HIGH" atunci când intrarea sa este la nivelul logic "0", oferindu-ne expresia booleană: Ᾱ = Y.

Deci, putem defini funcționarea unei porți logice digitale NOT cu o intrare ca fiind:"Dacă A nu este adevărat (NOT TRUE), atunci Q este adevărat (TRUE)".

O alta varianta de rezolvare a acestui exercitiu : 

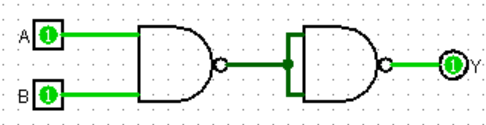
**Exercitiul 10M** Implementați un circuit logic care va activa ieșirea dacă intrarea este nenulă.



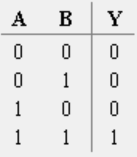
Pentru acest circuit am folosit 2 porti AND pentru a obtine un circuit care va active iesirea atunci cand intrarea este nenula (adica egala cu 1). Poarta standard NOT este dată de un simbol a cărui formă este de un triunghi orientat spre dreapta cu un cerculeț la capătul său. Acest cerc este cunoscut ca o „bulă de inversare“.



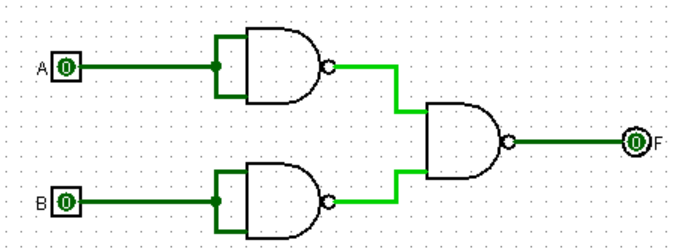
**Exercitiul 11M** Implementați un circuit logic care va activa ieșirea dacă ambele intrări sunt 1, dar nu și dacă sunt ambele 0.



Pentru acest circuit aveam nevoie de o poarta AND, dar pentru a complexa circuitul, am creat o poarta AND din 2 porti NAND, care este echivalenta. O poartă NAND este o poartă logică care este opusă unei porți logice AND. Este o combinație de porți AND și NU și este o poartă logică frecvent utilizată. O poartă logică AND este un tip de poartă logică digitală a cărei ieșire merge HIGH la un nivel logic 1 când toate intrările sale sunt HIGH.

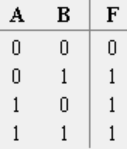


**Exercitiul 12M** Implementați un circuit logic care va activa ieșirea dacă cel puțin una dintre două intrări este 1, dar nu și dacă ambele sunt 0.

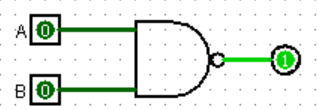


In general pentru a indeplini conditiile acestui exercitii avem nevoie de o poarta OR. Am creat aceasta poarta OR cu ajutorul a 3 porti NAND. O poartă logică OR (SAU) este un tip de poarta logică digitală a cărei ieșire merge HIGH la un nivel logic 1 când una sau mai multe intrări sunt HIGH

Ieșirea Q a unei "porți logice OR" este "LOW" când toate intrările sunt la un nivel logic "0". Cu alte cuvinte, pentru o poartă logică OR, orice intrare "HIGH" va da o ieșire "HIGH", nivel logic "1".

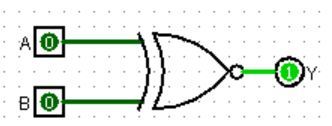


**Exercitiul 13M** Implementați un circuit logic care va activa ieșirea dacă cel puțin una dintre două intrări este 0, dar nu și dacă ambele sunt 1.



Pentru acest circuit si indeplinerea conditiei exercitiului am folosit 3 pini (2 pini de intrare, unul de iesire) si o poarta NAND.Poarta logică NAND este o combinație de poartă logică digitală AND și poartă NOT conectate între ele în serie. Poarta NAND (NOT - AND) are o ieșire care este în mod normal la nivelul logic "1" și este "LOW" la nivelul logic "0" atunci când toate intrările sunt la nivel logic "1". Poarta logică NAND este forma inversă sau „complementară“ a porții AND.

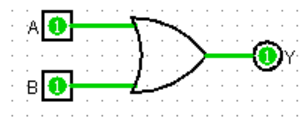
**Exercitiul 14M** Implementați un circuit logic care va activa ieșirea dacă ambele intrări sunt 0 sau ambele sunt 1, dar nu și dacă una dintre ele este 0 și cealaltă este 1.



Pentru acest circuit am folosit 3 pini si o poarta XNOR. Poarta XNOR este o poartă logică digitală care implementează negația operației de disjuncție exclusivă (XOR), în modul prezentat în tabelul de adevăr alăturat. Versiunea cu două intrări reprezintă egalitatea logică. O ieșire cu nivel înalt (1) rezultă dacă ambele intrări sunt pe același nivel. Dacă una (și nu ambele intrări) este cu nivel înalt (1), atunci ieșirea va fi pe nivel jos (0).

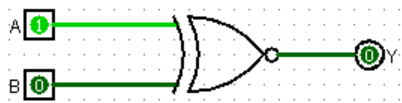


**Exercitiul 15** Implementați un circuit logic care va activa ieșirea dacă ambele intrări sunt 1 și cel puțin una dintre ele este 0.

****

O poartă logică OR (SAU) este un tip de poarta logică digitală a cărei ieșire merge HIGH la un nivel logic 1 când una sau mai multe intrări sunt HIGH. Ieșirea Q a unei "porți logice OR" este "LOW" când **toate** intrările sunt la un nivel logic "0". Cu alte cuvinte, pentru o poartă logică OR, orice intrare "HIGH" va da o ieșire "HIGH", nivel logic "1".

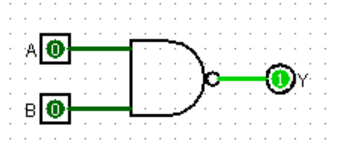
**Exercitiul 16M** Implementați un circuit logic care va activa ieșirea dacă ambele intrări sunt 0 sau ambele sunt 1, dar nu și dacă una dintre ele este 1 și cealaltă este 0.



Pentru acest circuit am folosit 3 pini si o poarta XNOR. Poarta XNOR este o poartă logică digitală care implementează negația operației de disjuncție exclusivă (XOR), în modul prezentat în tabelul de adevăr alăturat. Versiunea cu două intrări reprezintă egalitatea logică. O ieșire cu nivel înalt (1) rezultă dacă ambele intrări sunt pe același nivel. Dacă una (și nu ambele intrări) este cu nivel înalt (1), atunci ieșirea va fi pe nivel jos (0).

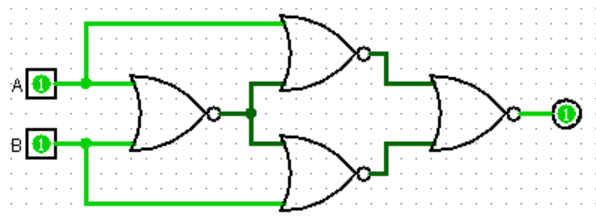


**Exercitiul 17M** Implementați un circuit logic care va activa ieșirea dacă ambele intrări sunt 0 și cel puțin una dintre ele este 1.

****

Pentru acest circuit am folosit poarta NAND. Poarta logică NAND este o combinație de poartă logică digitală AND și poartă NOT conectate între ele în serie. Poarta NAND (NOT - AND) are o ieșire care este în mod normal la nivelul logic "1" și este "LOW" la nivelul logic "0" atunci când toate intrările sunt la nivel logic "1". Poarta logică NAND este forma inversă sau „complementară“ a porții AND

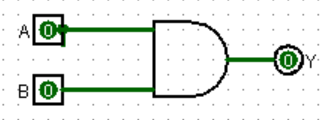
**Exercitiul 18M** Implementați un circuit logic care va activa ieșirea dacă ambele intrări sunt 1 sau ambele sunt 0, dar nu și dacă una dintre ele este 0 și cealaltă este 1.



In general, pentru acest circuit aveam nevoie de o poarta XNOR. Pentru un circuit mai complex, am creat poarta XNOR prin intermediul a 4 porti NOR. Poarta inclusivă NOR (NOT-OR) are o ieșire care este în mod normal la nivelul logic "1" și merge "LOW" la nivelul logic "0" atunci când oricare dintre intrările sale este la nivel logic "1". Poarta XNOR este o poartă logică digitală care implementează negația operației de disjuncție exclusivă (XOR), în modul prezentat în tabelul de adevăr alăturat.



**Exercitiul 19M** Implementați un circuit logic care va activa ieșirea dacă ambele intrări sunt 1 și nu există 0 între ele.

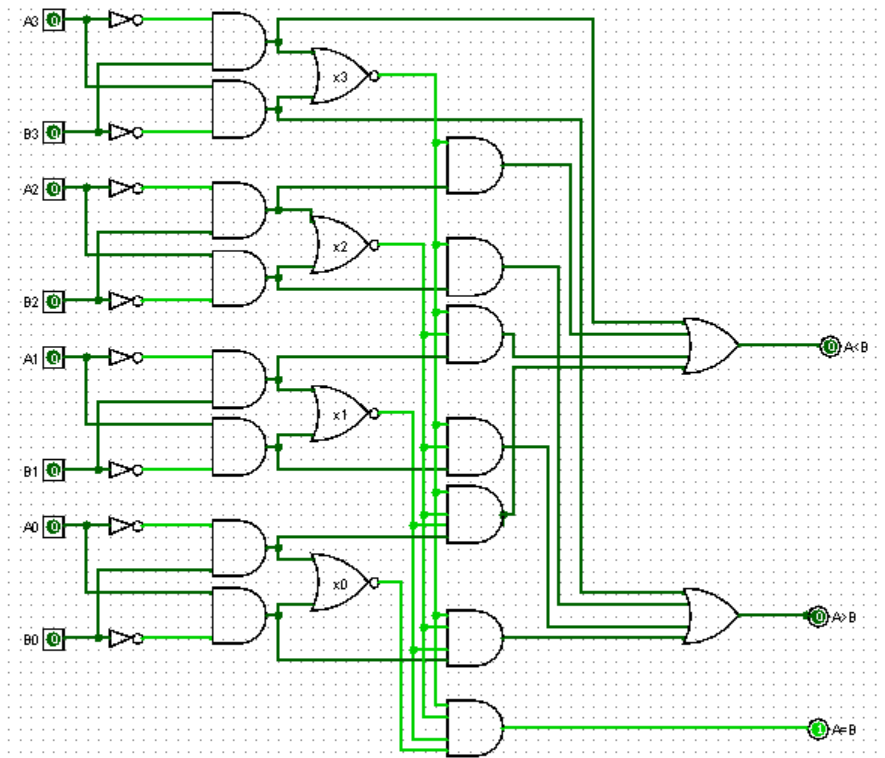


Pentru acest exercitiu avem nevoie de o poarta AND. O poartă logică AND este un tip de poartă logică digitală a cărei ieșire merge HIGH la un nivel logic 1 când toate intrările sale sunt HIGH. Starea de ieșire a unei "porți logice AND" este "LOW" când oricare dintre intrările sale este la un nivel logic "0". Cu alte cuvinte, pentru o poartă logică AND, orice intrare LOW va da o ieșire LOW

**Exercitiul 1H** Implementați un comparator cu 4 biți folosind porți logice.

Input:A=A3 A2 A1 A0 B=B3 B2 B1 B0

Expresia pentru AB:

****

Pentru implimentarea acestui exercitiu am folosit pini de intrare care au fost conectati toti cu porti NOT. Apoi din fiecare pereche de pini, unul a fost conectat la o intrare a portii AND, iar la cealalta intrare a portii NAND s-a conectat iesirea portii NOT. Apoi fiecare pereche de porti AND se conecteaza la o poarta XOR, ulterior la porti AND si OR.

Într-un comparator pe 4 biți, ***condiția A>B*** poate fi posibilă în următoarele patru cazuri:

Dacă A3 = 1 și B3 = 0

Dacă A3 = B3 și A2 = 1 și B2 = 0

Dacă A3 = B3, A2 = B2 și A1 = 1 și B1 = 0

Dacă A3 = B3, A2 = B2, A1 = B1 și A0 = 1 și B0 = 0

În mod similar, ***condiția pentru A<B*** poate fi posibilă în următoarele patru cazuri:

Dacă A3 = 0 și B3 = 1

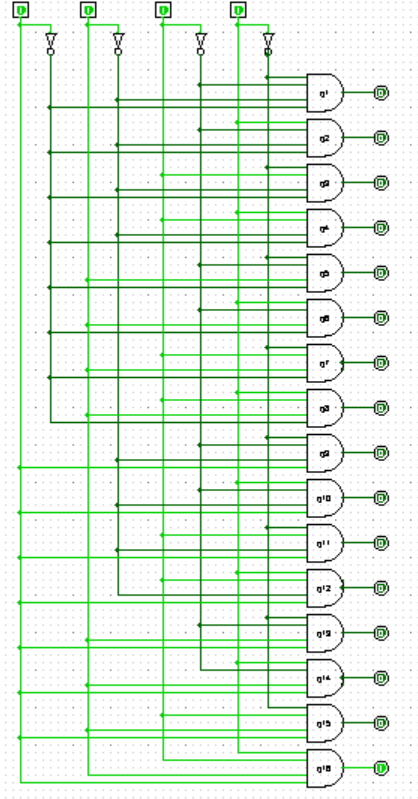
Dacă A3 = B3 și A2 = 0 și B2 = 1

Dacă A3 = B3, A2 = B2 și A1 = 0 și B1 = 1

Dacă A3 = B3, A2 = B2, A1 = B1 și A0 = 0 și B0 = 1

***Condiția A=B*** este posibilă numai atunci când toți biții individuali ai unui număr coincid exact cu biții corespunzători altui număr.

**Exercitiul 2H** Implementați un decodor cu 4 biți și 16 ieșiri

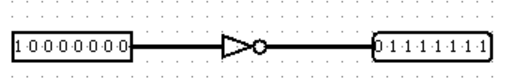


Un decodor pe 4 biți are 4 linii de intrare și 16 linii de ieșire. Adică sunt necesare 16 porți de decodare pentru a decoda toate combinațiile posibile de patru biți. Pentru orice cod dat de intrare, una dintre cele șaisprezece ieșiri devine HIGH

In acest exercitii pentru a implimenta un decoder am folosit 4 pini de intrare care au fost conectati la poarta NOT si ulterior au fost conectati cu porti AND. Am folosit 16 porti AND si repsectiv 16 pini de iesire.

Exemplu: valoarea inputului 1111, atunci circuitul va activa iesirea pentru a 16-a iesire.

**Exercitiul 11H** Implementați un circuit care realizează operația de negare a unui număr de 8 biți.



Operatia de negare reprezinta inversarea valorii. Daca este 1, se va transforma in 0, si daca este 0 va deveni 1.

Diferite verificari :





**Concluzie** În final, pot spune că experiența mea cu Logisim în laboratorul 3 a fost foarte utila. Am dobândit cunoștințe despre modul în care elementele de logică digitală, cum ar fi ușile logice, multiplexarele, flip-flop-urile, ceasurile și altele, funcționează și am înțeles cum acestea pot fi conectate pentru a crea circuite logice mai complexe.

În urma laboratorului nr.3, am realizat că aplicația Logisim poate fi o unealtă puternică pentru simularea și proiectarea circuitelor digitale. Prin aceste aplicații, putem proiecta și testa circuitul logic într-un mediu virtual, înainte de a fi implementat într-un mediu hardware real.

Prin urmare, am dobândit o mai bună înțelegere a modului de funcționare a circuitelor digitale și acum deținem abilitățile necesare pentru a proiecta și a simula circuite logice simple în Logisim. Aceste competențe pot fi extrem de utile pentru a realiza proiecte practice sau pentru aprofundarea cunoștințelor despre logica digitală.