

**DIRECȚIA GENERALĂ EDUCAȚIE TINERET ȘI SPORT**

Conferința științifică “ MUNCĂ, TALENT CUTEZANȚĂ ”

**Disciplina, atelierul: Informatica. Aplicații, utilizare**

**Tema :**

**Joc elaborat în mediul de dezvoltare LOGISIM,  
ca aplicație sumativă la cursul  
de STRUCTURA CALCULATORULUI**

Autorul, clasa: Mazur Dinu, clasa a X-a B

Profesor – îndrumător : Sergiu Burlacu

**Chișinău 2016**

## Scopul lucrării

A elabora un joc în baza cunoștințelor acumulate și a explorare potențialul didactic al aplicației LOGISIM în scopul studierii cursului școlar “Structura calculatorului ” în clasa a X-a, profil real.

“

### *Cuprins*

Descrierea lucrării	3
Descrierea fiecărui compartiment al aplicației	4
Generatorul de valori aleatoare GVA	4
Regiștrii de deplasare	6
Generatorul de linii de spații	7
Controller pentru dirijarea obiectului – celulă	8
Contorul pentru numărul de vieți	9
Procesul de investigație științifică individuală	10
Noutatea și abordarea științifică	13
Aplicabilitatea cercetării	13
Anexe elaborate personal în bază de exerciții proprii	15

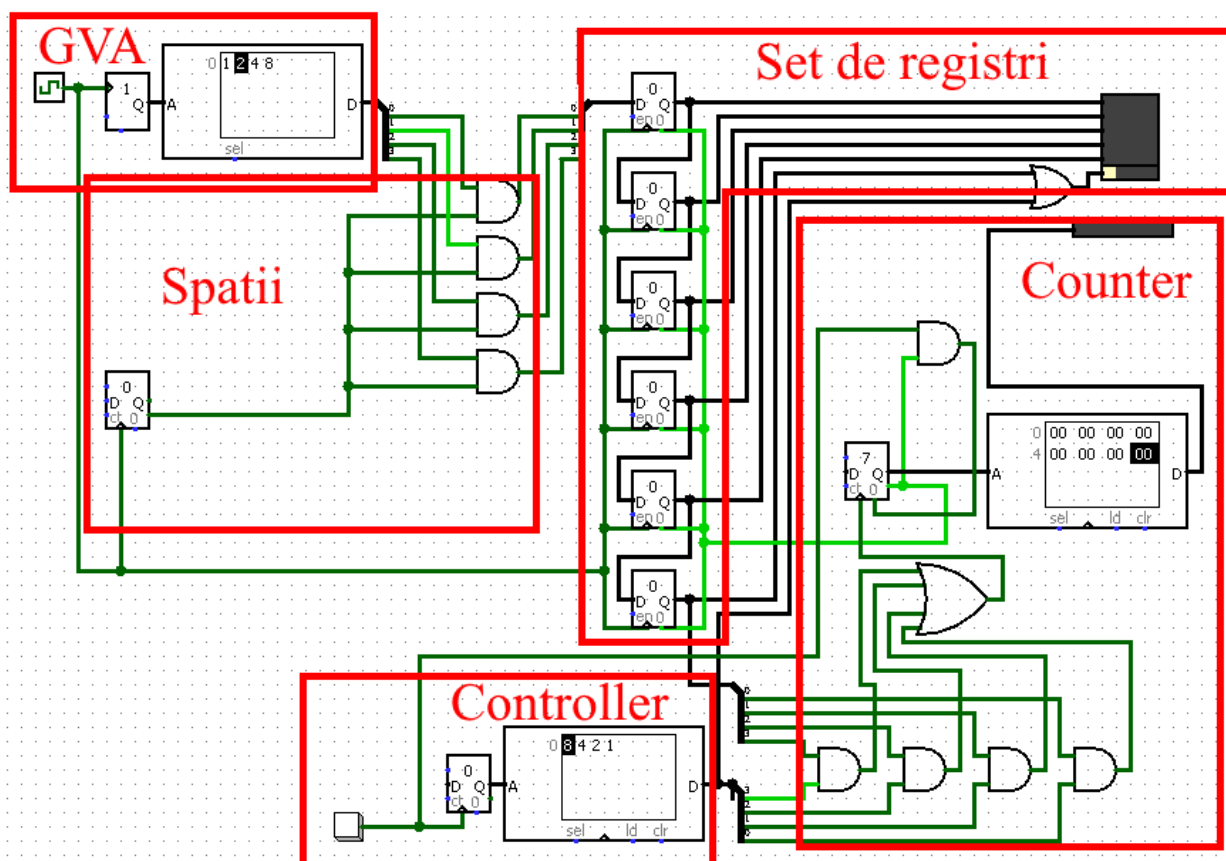
## Descrierea lucrării

Această lucrare reprezintă în esență ei un joc aplicație, care reprezintă un obiect virtual dirijat de către utilizator— o celulă, pe un câmp- panel LED matrice din 1 row x 4 cols, aflat pe ultima linie de jos a ariei de afișare a jocului,

REGULILE JOCULUI va fi nevoie de a evita coliziunea cu obiectele celule generate aleator și care “cad de sus” pe aria unui alt panel-LED matrice din 4cols x 5 rows.

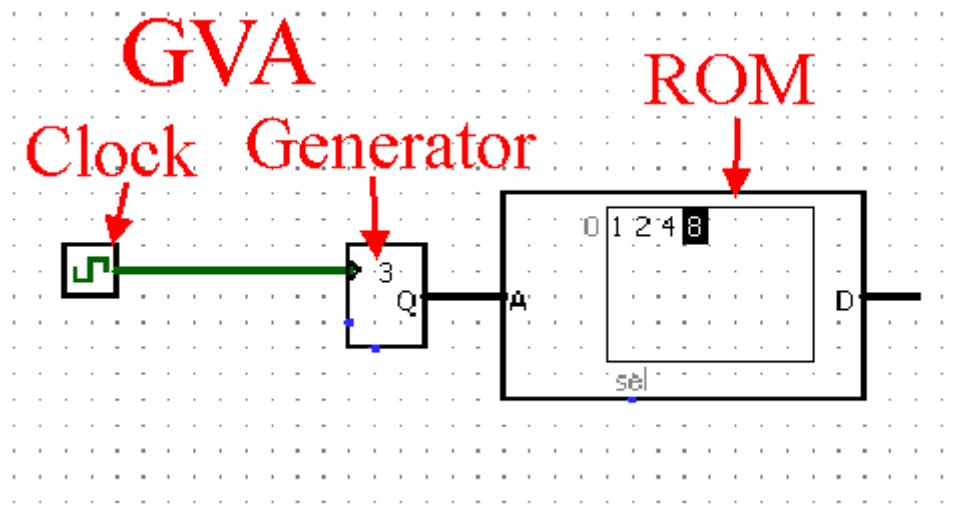
Lucrarea mea este constituită din următoarele 5 compartimente

1. Generatorul de valori aleatoare GVA, cu exprimare hexazecimală 1-2-4-8
2. Generator de linii din spații
3. Set de regiștrii pentru deplasarea paralelă a valorilor binare si proiectarea lor pe panel
4. Controller pentru dirijare a obiectului celulă de către utilizator
5. Contorul pentru numărul de “ vieți ”, care în acest caz, inițial sunt 7



## Descrierea fiecărui compartiment al aplicației

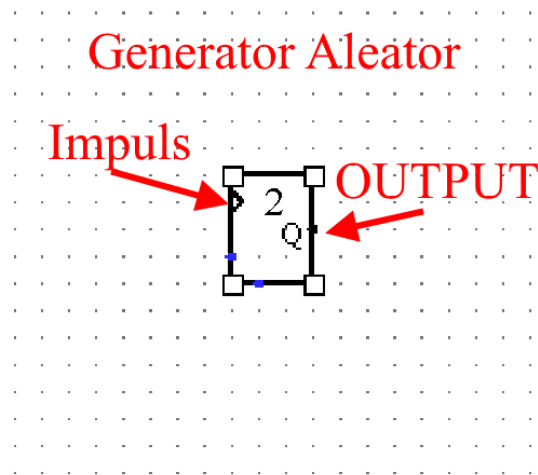
**Generatorul de valori aleatoare GVA, cu exprimare hexazecimală 1-2-4-8 conectat la elementul ROM de memorie**



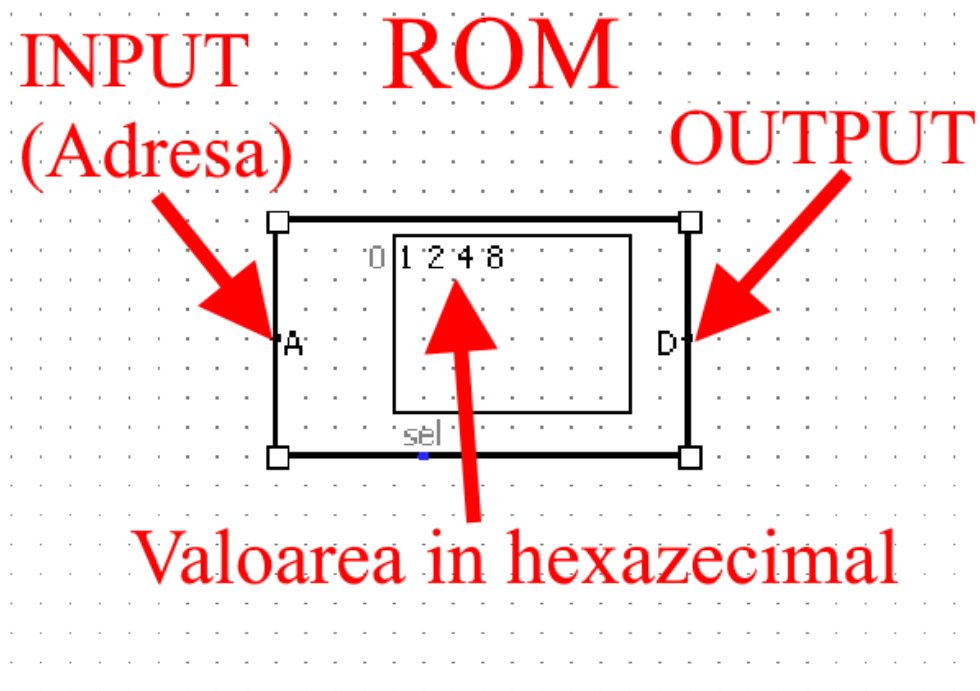
GVA reprezintă un set din următoarele componente cu funcționalitatea lor:

- Generatorul de impulsuri CLOCK (de indicat-se porneste opreste setează), care periodic trimite câte un impuls către generatorul de valori aleatoare.
- Generatorul de valori aleatoare este pe 2 bits, adică în total 4 valori binare, care ulterior vor fi servi pentru selectarea a 4 adrese corespunzătoare din elementul ROM – memorie statică

Bin	Hex
00	0
01	1
10	2
11	3



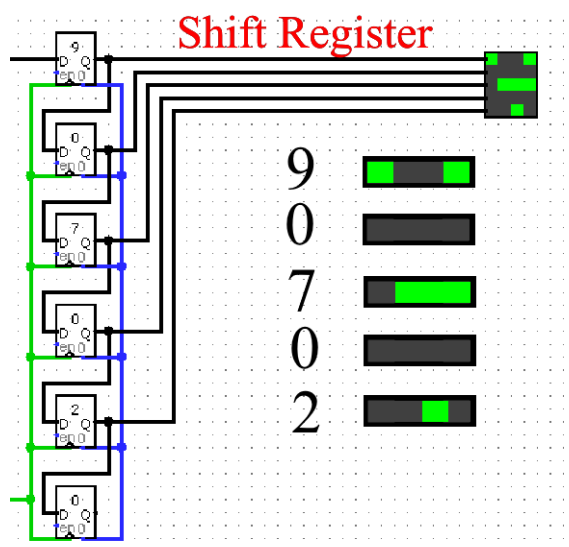
- ROM păstrează informația, afișată hexa în desen 1-2-4-8, dar care în realitate este una binară și o afișează în OUTPUT – eșire marcat cu D, în dependență de valoarea adresei în INPUT - intrare, marcat cu A.



In continuare valoarea binară de la ROM se transmite mai departe spre Setul de regiștri

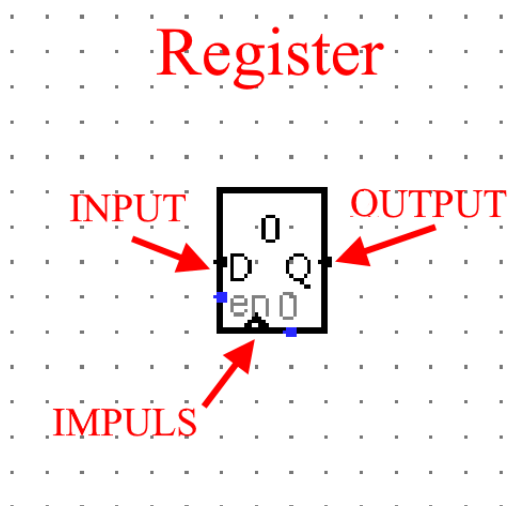
## Set de regiștrii pentru deplasarea paralelă a valorilor

binare și proiectarea lor pe panou



Circuitul răspunde de proiectarea succesivă a valorilor binare, primite de la memoria ROM, pe panelul LED pentru crearea efectului de cădere a obstacolelor.

Registrul pentru deplasarea paralelă se mai numește **shiftregister** eng. Proprietatea principală registrului menționat este de a transfera valoarea binară de la INPUT spre OUTPUT în regim sincron odată recepționarea impulsului de la CLOCK-ul comun al aplicației. La fel, fiecare registru posedă o linie de RESET-are, odată cu venirea impulsului de acțiune, care va determina zerografierea conținutului regiștrilor.



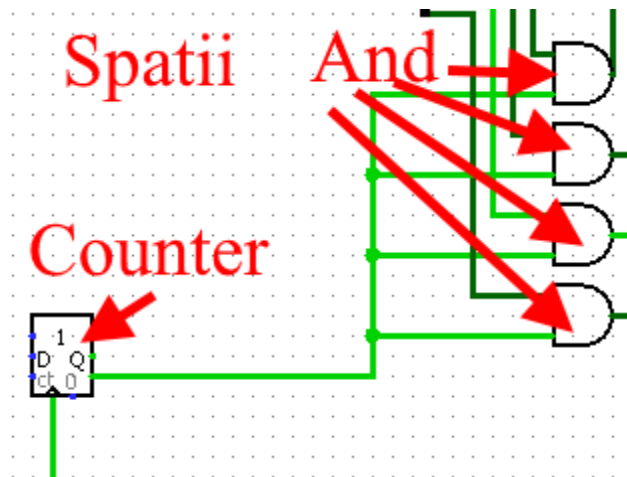
Evident că un circuit din mai mulți regiștri se face capabil să transfere informația de la primul registru spre al doilea, apoi al treilea ș. a. m. d.

În general întreg sistemul precum și setul de regiștri este construit pe 4 biți. Dar există segmente de circuit în care am utilizat 1, 2 sau 7 biți.

## Elementele componente

1. Registere pe 2 biți
2. Panou LED 4cols x 5 rows culoare verde

### Generator de linii de spații



Acest circuit răspunde de crearea spațiilor pentru a simplifica navigarea printre obstacole utilizatorului în procesul jocului.

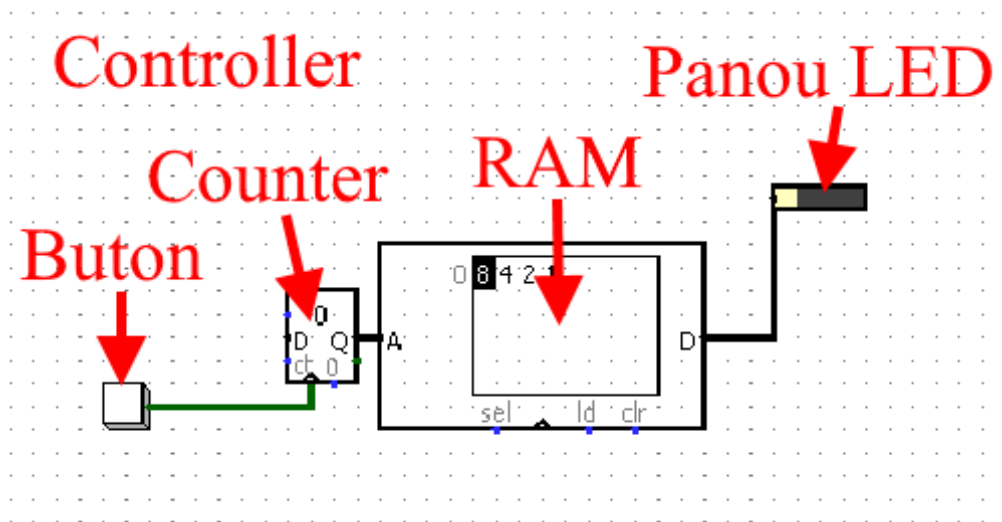
Este alcătuit din 2 elemente

1. Counter pe un bit sau elementul care va împărți la 2 impulsurile venite de la CLOCK-ul general
2. Patru elemente de “ȘI”, care are rolul de a de a înlocui valoarea unei linii de date care vine de la ROM cu o linie de spații

## Funcționalitatea

Counter-ul numără impulsurile primite de la clock. La primirea a 2 impulsuri, el va bloca valoarea care venea de la GVA și pleca spre shift registru. În locul ei el va fi inserată valoarea 0000 a unei linii. Aceasta acțiune va continua până în momentul când counterul va primi un impuls nou venit de la clock. Astfel la căderea obstacolelor pe panou, vor apărea peste una linii cu spații.

### Controller pentru dirijare a obiectului-celulă de către utilizator



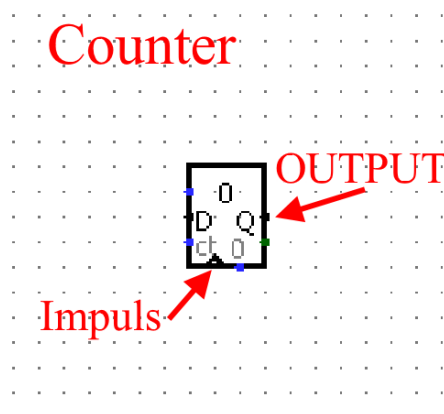
Acest controller este simplu de realizat. El este destinat deplasării a obiectului ghidat de utilizator, numai spre dreapta, ceea ce complică esențial condițiile jocului.

Controller-ul este constituit din următoarele elemente

1. Buton *Button eng*
2. Contor pe 2 biți *Counter eng*
3. Memorie RAM cu 4 adrese
4. Panou LED 1x4 culoare alb-gălbui

Funcționalitatea

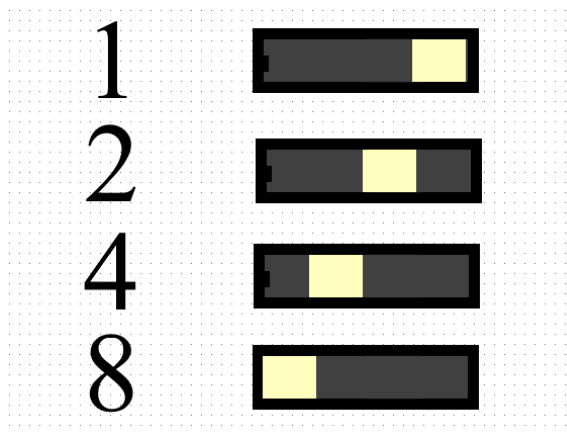
1. Butonul, la apăsare de către user, va genera un impuls, care va nimeri în INPUT-ul controlului
2. Contorul va memora numărul de impulsuri și la rîndul său va genera un OUTPUT binar propriu pentru sistemul de adrese RAM



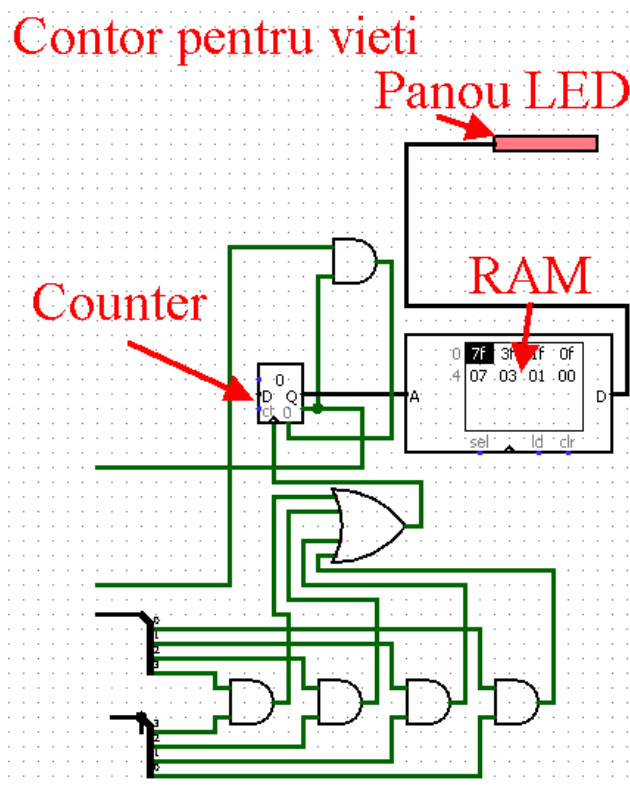
3. RAM, este un element care diferă de ROM, dar cu funcționalitatea apropiată. Cu toate acestea, dacă valorile plasate în adresele RAM se vor pierde la deconectarea sursei de alimentare, pentru ROM aceasta nu e o problemă și respectiv aceste valori nu se pierd.



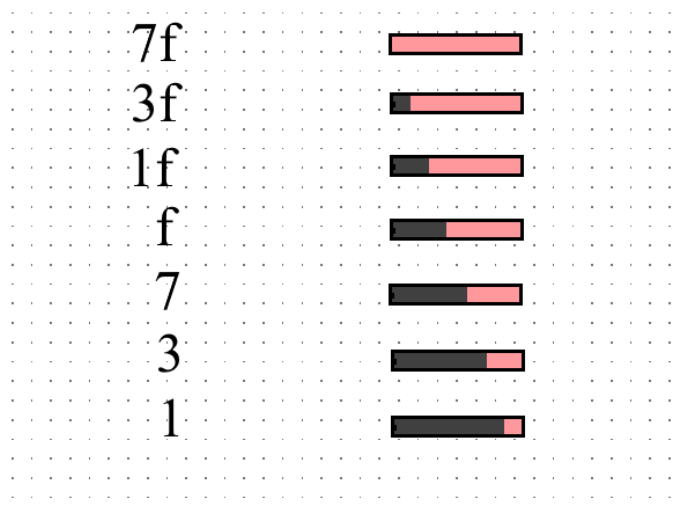
Valorile din adresele RAM vor fi în continuare proiectate pe Panoul LED 1 x 4 de culoare alb-gălbuie.



**Contorul pentru numărul de “vieți”, care în acest caz sunt 7**



Acest circuit este capabil să compare valorile primite din capătul setului de regiștri și valorile de la controller. Așadar primind valori egale el generează o valoare binară care corespunde cu numărul de viețile rămase apoi o afișează pe panoul LED 1x7. Valorile necesare pentru afișare sunt 7f-3f-1f-f-7-3-1-0



Această acțiune se va repeta până în momentul când numărul de corespunderi va fi, în acest caz egal cu 7. La atingerea valorii 7 Panoul LED de menționat deja se va receta 5x4 de la setul de regiștri. Eventual jocul va relua din start la apăsarea butonului de la Controller-ul pentru dirijare.

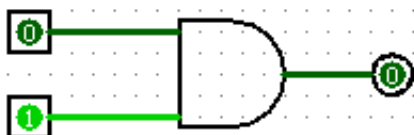
Acest circuit este construit din 3 elemente de baza

1. Contor pe 3 biți
2. Memorie RAM cu 8 adrese si 7 biți
3. Panou LED 1x7 culoare roz

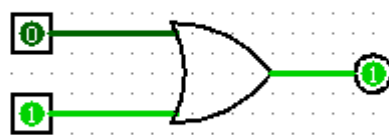
### Procesul de investigație științifică individuală

Pe parcursul orelor de informatică am studiat elementele unui calculator si ne-am învățat sa operăm cu setul de biți. În acest process elevii au învățat multe, dar înca nu s-a creat o imagine clară despre totul. Astfel profesorul a oferit un mediu de dezvoltare vizual pentru elementele studiate care se numește LOGISIM. Primele exerciții păreau la prima vedere, simple,

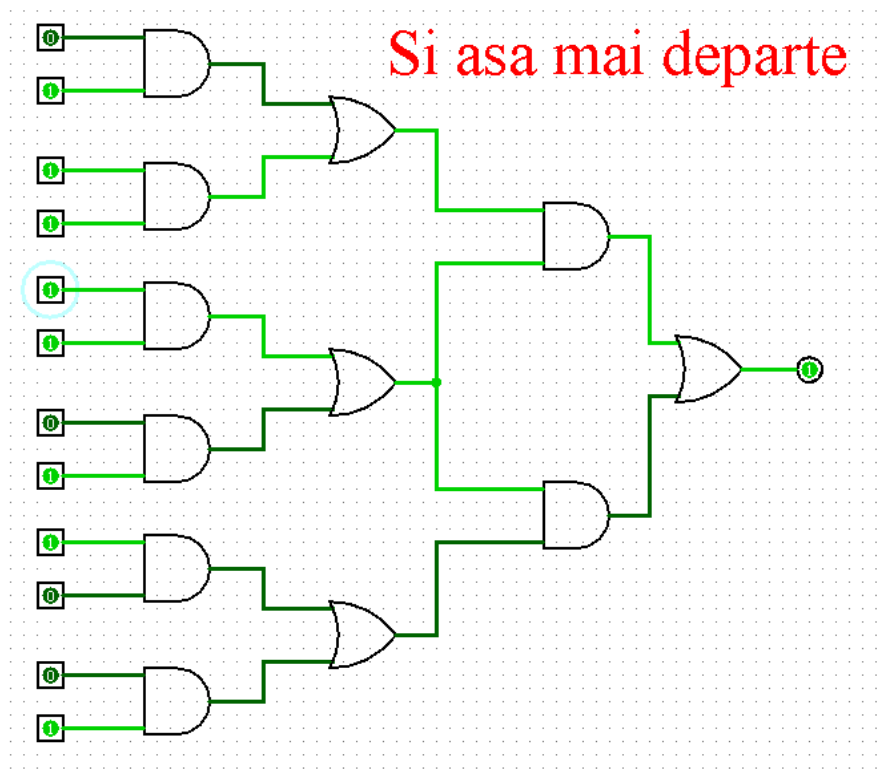
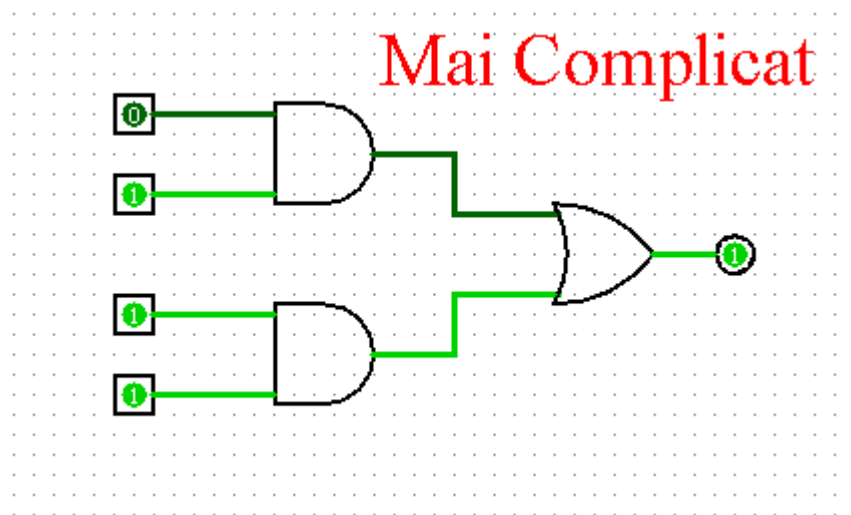
And



OR



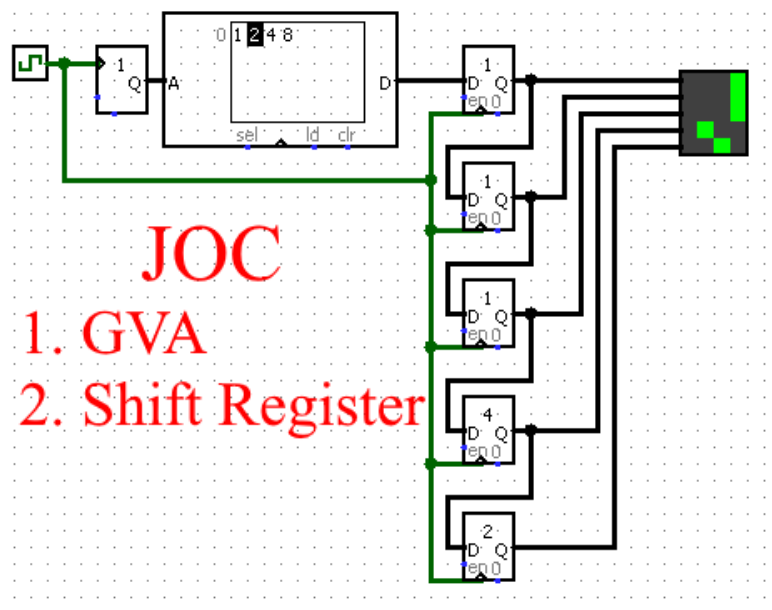
dar din mai multe circuite se primește un circuit complex care este deja mai complicat.



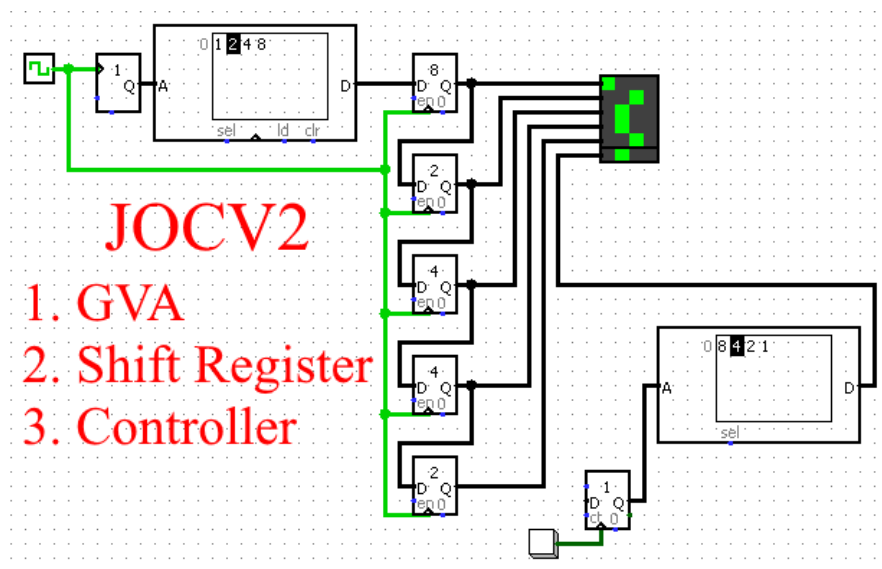
Am început cu o simplă numărătoare binară, începînd să număr pe 4 biți de la 0000 pînă la 1111, am observat că niște valori au doar cîte un unu în rînd (1.2.4.8)

Dec	Bin	Dec	Bin
0	0000	10	1010
• 1	0001	11	1011
• 2	0010	12	1100
3	0011	13	1101
• 4	0100	14	1110
5	0101	15	1111
6	0110		
7	0111		
• 8	1000		
9	1001		

astfel m-am gândit la un concept de joc. Condiția jocului era ocolirea obstacolelor cu o celulă folosind doar 2 butoane (dreapta și stânga). Primul lucru care l-am realizat era căderea obstacolelor dar la controale nu am ajuns.



Cu timpul, în procesul creării jocului au intervenit elevii și profesorul. Noi am trecut la controale pe baza de shiftregistru bidirecțional dar circuitul era de câțeva ori mai mare decât circuitul pentru obstacole și am rămas la controler unidirecțional (dreapta).



În acest proces am aflat despre ROM și RAM. De exemplu memoria RAM pierde valorile la fiecare restart și astfel am înțeles că RAM este capabil să mențină informația atît timp cît este conectată sursa de energie. Am mai observat că într-un circuit din logisim funcționează doar un element de memorie statică ROM. În a 3-a versiune am adăugat contor pentru numărul de vieți și așa cu puține optimizări, la versiunea a 4-a, am finalizat jocul.

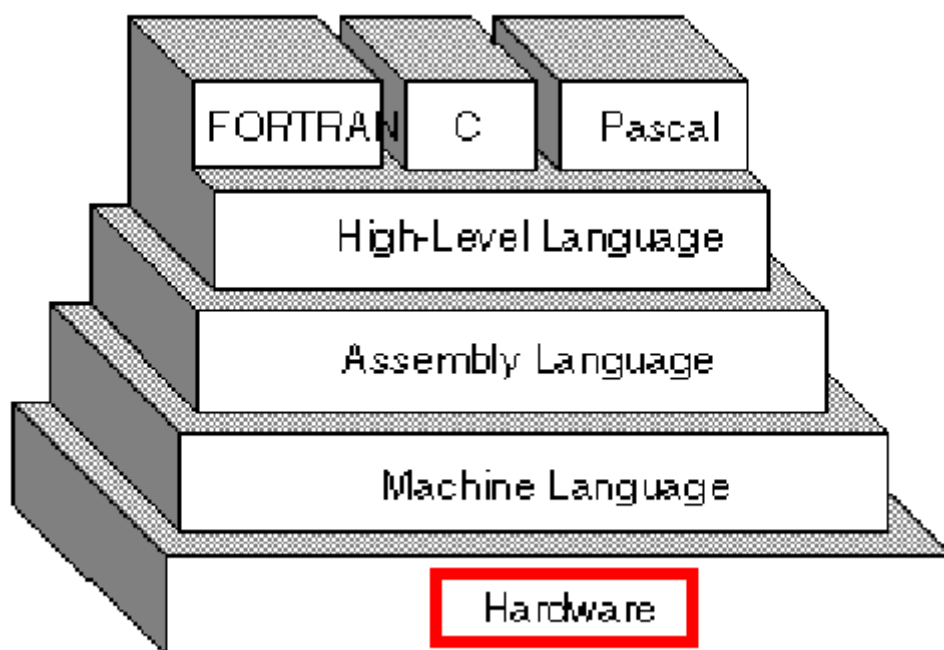
### Noutatea și abordarea științifică

În acest proces de inițiere în domeniul HARDWARE am observat că mediul LOGISIM este un mediu ideal pentru aprofundare. Pentru a putea lucra efectiv în acest mediu este nevoie ca să cunoaștem deja totul despre biți (definiția, numărătoarea binară, operațiile binare) elementele se vor învăța prin metoda încercărilor. Mediul LOGISIM ne oferă aproape tot ceea ce putem face cu elementele reale iar crearea și eventual testarea unor circuite complexe se face mai rapid și nelimitat în resurse. Circuitul care l-am propus se assemblează, se testează, modifică și corectează în LOGISIM. Mai apoi, dacă el funcționează așa cum ne-am propus, el poate fi recreat pe placa pentru machetare și deja pentru o variantă reală. Aceasta nu se compară cu timpul pierdut, dacă bunăoară aceleaș experiențe era să le realizăm în realizate, adică cu ciocanul de lipit în mînă, la crearea variantei finale ale aceluiași circuit. Logisim poate fi de folos atît cît în cursul școlar, ca mediu de dezvoltare, cît și în viața unui lucrător legat de domeniul hardware care poate să-și testeze circuitele ca apoi să fie create în varianta reală.

### Aplicabilitatea cercetării

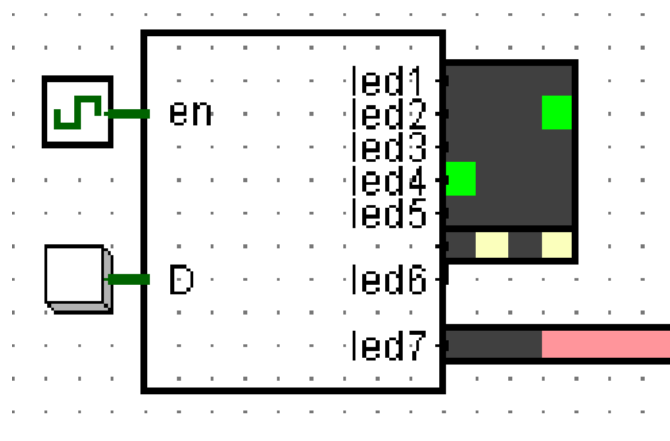
Acest joc combină cunoștințele acumulate în timpul orelor teoretice și celor practice cu aplicația LOGISIM. În mediul logisim putem arăta aplicabilitatea practică a elementelor în vederea construirii unui circuit mai complex. Așadar realizarea unui joc este un bun început pentru cursul de inițiere în domeniul microcontroalelor, procesoarelor și roboticii. De la un simplu joc de fiecare dată cu siguranță încep lucruri mai serioase, așa ca noi idei, proiecte,

etc. Lumea deja s-a întâlnit cu asemenea cazuri și din practicile acumulate a apărut următorul adevăr – jocul este etapa de inițiere a unui proiect mai complex. Pe lângă altele, jocul rămîne să fie o sursă de relaxare și ocupă loc prioritar în industria distracțiilor. LOGISIM-ul este un mediu minunat pentru testarea cunoștințelor și eventual continuarea studiilor și avansarea în acest domeniu. La orice greșeală comisă în LOGISIM, putem să ne oprim, să corectăm greșeala și să continuăm mai departe. Astfel fiecare greșeală ne determină asupra cunoașterii a unor noi “reguli de joc”. Așadar noi nu suntem nevoiți să ne preocupăm de plăci, ciocane de lipit și alte elemente pentru a ne testa ideile proprii. Odată cu cunoașterea elementelor din HARDWARE, cu ușurință putem să continuăm dezvoltarea pe ramura SOFTWARE, care este o ramură suficient de extinsă. Sunt mulți programatori incompetenți deoarece nu s-au întâlnit cu lumea procesoarelor și a microcontroalelor. Dacă am grupa elementele limbajelor de programare și elementele HARDWARE obținem aceea, că domeniul HARDWARE se află la baza piramidei și aceasta este evident.



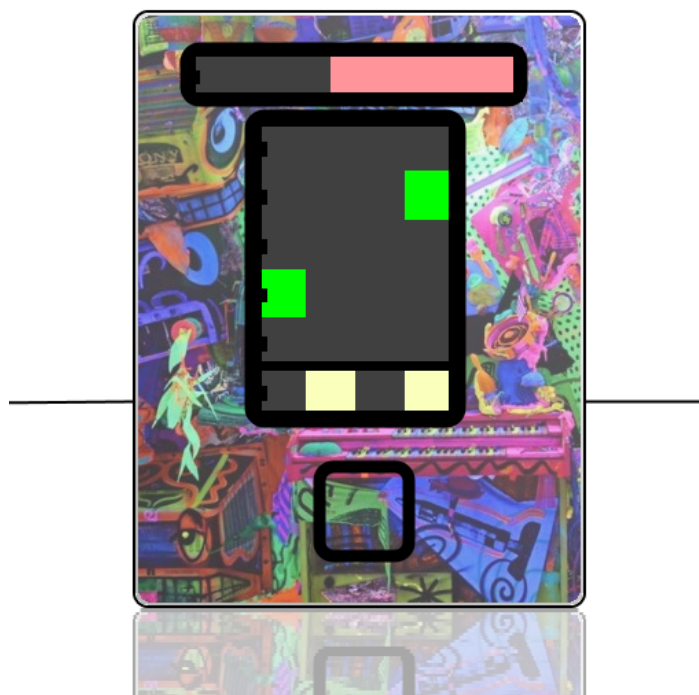
Așadar, mediul LOGISIM oferă un sprijin în ceea ce privește ambele ramuri. Cu toate acestea noi, elevii clasei a x-a, tindem, prin această prezentare, să trezim interesul față de elementele HARDWARE și astfel, în manualele de informatică să apară și compartimentul LOGISIM pe lângă cangur, robochert, ș. a. m. d. Programul nostru este un exemplu bun pentru inițiere în domeniul Hardware. Evident că acest program nu este apogeul dezvoltării, el poate fi optimizat sau să-l modificăm așa ca el să poată realiza o condiție nouă care ne-am propus-o. Noi ne vom aprofunda mai departe creînd circuite mai mari și mai multe, nu numai jocuri, noi nu suntem limitați în idei. Multe școli din țări diferite deja s-au aprofundat în acest domeniu.

## Anexe elaborate personal în bază de exerciții proprii



Logisim e capabil să încadreze un circuit într-un subcircuit. Astfel în desenul de mai sus se conține întreg circuitul jocului lucrării date. Așa ar arăta jocul într-o interpretare de modul confecționat.

# JOC™



# Panou pentru troleibuz

