

-Catedra de Calculatoare-

CIRCUITE DE INMULTIRE ZECIMALA

Nume: Kiraly Nicole Elena

Grupa: 30237

Nume indrumator proiect: Dragos Florin Lisman

Data: 13.01.2024

Cuprins

1. Rezumat ...........................................................................................3

2. Introducere.......................................................................................4

3. Fundamentare teoretică...................................................................6

3.1 „Repeated-addition method”

3.2 „Nine-Multiplies-of-multiplicand method”

4. Proiectare şi implementare...............................................................7

4.1 Arhitectura sistemului

4.2 Metoda 1-Metoda adunarii repetate

4.3 Metoda 2-Metoda celor 9 multipli ai deinmultitului

4.4 Manual de utilizare

5. Rezultate experimentale.................................................................17

6. Concluzii..........................................................................................20

7. Bibliografie......................................................................................21

8. Anexe..............................................................................................21

1. **REZUMAT**

Recent, aritmetica zecimală a câștigat teren în lumea hardware, oferind o alternativă interesantă la metodele tradiționale bazate pe binar pentru calcule. În cadrul proiectului nostru, ne-am concentrat pe crearea și punerea în practică a trei metode diferite pentru înmulțirea numerelor zecimale de patru cifre. Am integrat aceste metode într-un sistem digital, le-am comparat și am subliniat cum pot ele fi folosite în tehnologiile actuale.

Pentru a duce la bun sfârșit aceste sarcini, am folosit limbajul VHDL pentru a descrie structura circuitelor, lucrând în mediul Vivado Design Suite. Acesta ne-a ajutat să testăm, să depistăm erori, să facem sinteza circuitelor și să analizăm rezultatele și rapoartele. Am implementat circuitele pentru fiecare metodă și am creat un sistem care le include și este ușor de folosit.

Am testat rezultatele mai întâi în simulatorul din Vivado Design Suite, apoi pe placa de dezvoltare FPGA Nexys 4 DDR, unde am implementat efectiv circuitul. Acest proiect a necesitat o înțelegere solidă a tehnicilor de descriere hardware și a algoritmilor folosiți, și m-a ajutat să-mi aprofundez cunoștințele și să-mi dezvolt abilitățile în proiectarea și implementarea unor circuite hardware complexe. Aceste competențe sunt super utile și aplicabile în domenii financiare, economice și tehnologice actuale.

1. **INTRODUCERE**

Aritmetica zecimala este preferata in majoritatea mediilor deoarece este cea mai utilizata, fiind cunoscuta de majoritatea lumii. Desi calculatorul se foloseste de aritmetica binara, operand cu 0 si 1, efectueaza si operatii zecimale , utilizand logica binara. Este important sa putem folosi sistemul de numeratie zecimal deoarece numerele in virgula mobila reprezentate in format binar pot duce la erori, si nu precizie, existand domenii unde precizia reprezinta un factor important.

In acest context, dispozitivele hardware care suporta aritmetica zecimala au devenit din ce în ce mai importante, datorita cerintei crescute de putere de procesare. In prezent, aceste unitati hardware sunt o componenta esentiala a procesoarelor general-purpose, permitand efectuarea rapida si eficienta a operatiilor zecimale complexe, care în trecut erau realizate prin algoritmi hardware mai lenti și iterativi.

Tema proiectului este implementarea unui circuit de calcul capabil sa efectueze inmultirea zecimala a 4 numere reprezentate in BCD. Voi implementa 2 dintre metodele de a realiza inmultirea zecimala, in final utilizatorul alegandu-si metoda pe care vrea sa o urmeze. In realizarea acestui proiect se va utiliza limbajul de descriere hardware VHDL, implementarea se va realiza pe o placa de dezvltare FPGA, unde utilizatorul va putea observa numerele introduse precum si rezultatul produsului.

Solutia propusa urmareste descrierea tehnicilor de inmultire „The Repeated-Addition”, si „The Nine-Multiplies-of- Multiplicand”.

Metoda „The Repeated-Addition” este cea mai usoara, dar si cea mai lenta, dar ofera un rezultat corect a inmultirii.

A doua metoda se foloseste de cei 9 multiplii ai de inmultitului, acestia trebuie stocati in registri, ceea ce duce la un cost ridicat si un consum de timp in plus.

In continuarea lucrarii, se vor descrie in detaliu fundamentele teoretice pentru metodele alese in realizarea aritmeticii zecimale, apoi ne vom indrepta atentia spre proiectarea si implementarea celor 2 metode, vom trece sa verificam rezultatele obtinute. In ultimul capitol se va realiza un sumar al lucrarii, iar mai apoi in rezumat vom completa un rezumat al lucrarii.

1. **FUNDAMENTARE TEORETICA**

In prezent, principalele metode observate de implementare a inmultirii zecimale sunt urmatoarele:

* „Repeated-addition method”
* „Nine-multiplies-of-multiplicand method”
* „The Right-and-Left-Hand Components method”
* „Doubling-and-halving method”
  1. **„Repeated-addition method”**

Metoda adunarii repetate este cea mai usoara metoda, precum si cea mai lenta, dar ofera o solutie corecta. Aceasta se plimba prin toate cifrele inmultitorului si calculeaza produsele partiale prin adunarile repetate ale deinmultitului si inmultitorului pana cand nu mai sunt cifre, adica un counter devine 0 prin decrementare. Produsul partial se stocheaza intr-un grup de registri si se shifteaza spre dreapta cand se trece la urmatoarea cifra a inmultitorului.

* 1. **„Nine-multiplies-of-multiplicand method”**

In realizarea acestei metode trebuie generati cei 9 multiplii ai deinmultitului si stocarea lor in registrii astfel incat vor fi folositi dupa. Ne vom focusa atentia pe observarea fiecarei cifre a inmultitorului astfel incat in functie de aceasta este ales registrul cu multiplul corespunzator si apoi adunat la produsul partial, apoi se shifteaza spre dreapta registrul cu produsul partial si se trece la urmatoarea cifra a inmultitorului.

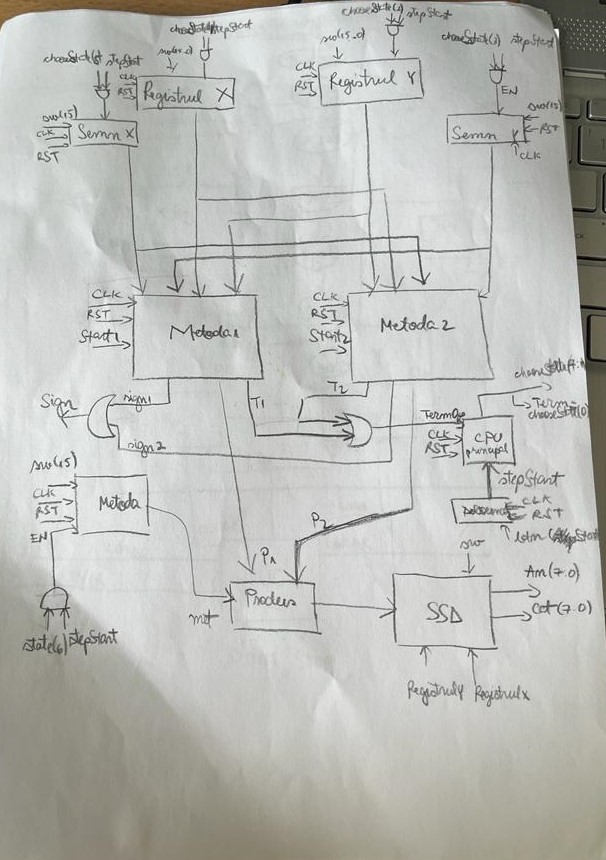
1. **PROIECTARE SI IMPLEMENTARE**

In continuare vom incepe sa ne indreptam atentia spre proiectarea si implementarea celor 2 metode de inmultire: metoda adunarii repetate si metoda celor 9 multiplii ai deinmultitului.

Am ales sa implementez cele 2 metode deoarece prima metoda e cea pe care o invatam de cand suntem mici, este o metoda usoara si prin folosirea unei metode lente si pe una mai rapida putem observa diferentele in implementarile lor. In plus, metoda celor 9 multipli este una interesanta dar costisitoare, insa nu e greu de implementat.

Aceste metode sunt implementate cu ajutorului limbajului de descriere hardware VHDL si in simulator sunt prezentate rezultatele dupa realizarea metodelor folosite.

* 1. **Arhitectura sistemului**



In figura de mai sus este prezentata arhitectura generala a sistemului, inglobarea celor 2 metode pe care le vom implementa si afisarea pe SSD a valorilor x si y si a produsului.

Deoarece o cifra nu poate fi mai mare decat 9 am adaugat si flaguri de eroare care verifica daca cifra este mai mare ca noua.

ANOD(4)

X(16b)

INMULTITOR

ZECIMAL

CATOD(7)

Y(16b)

CLK

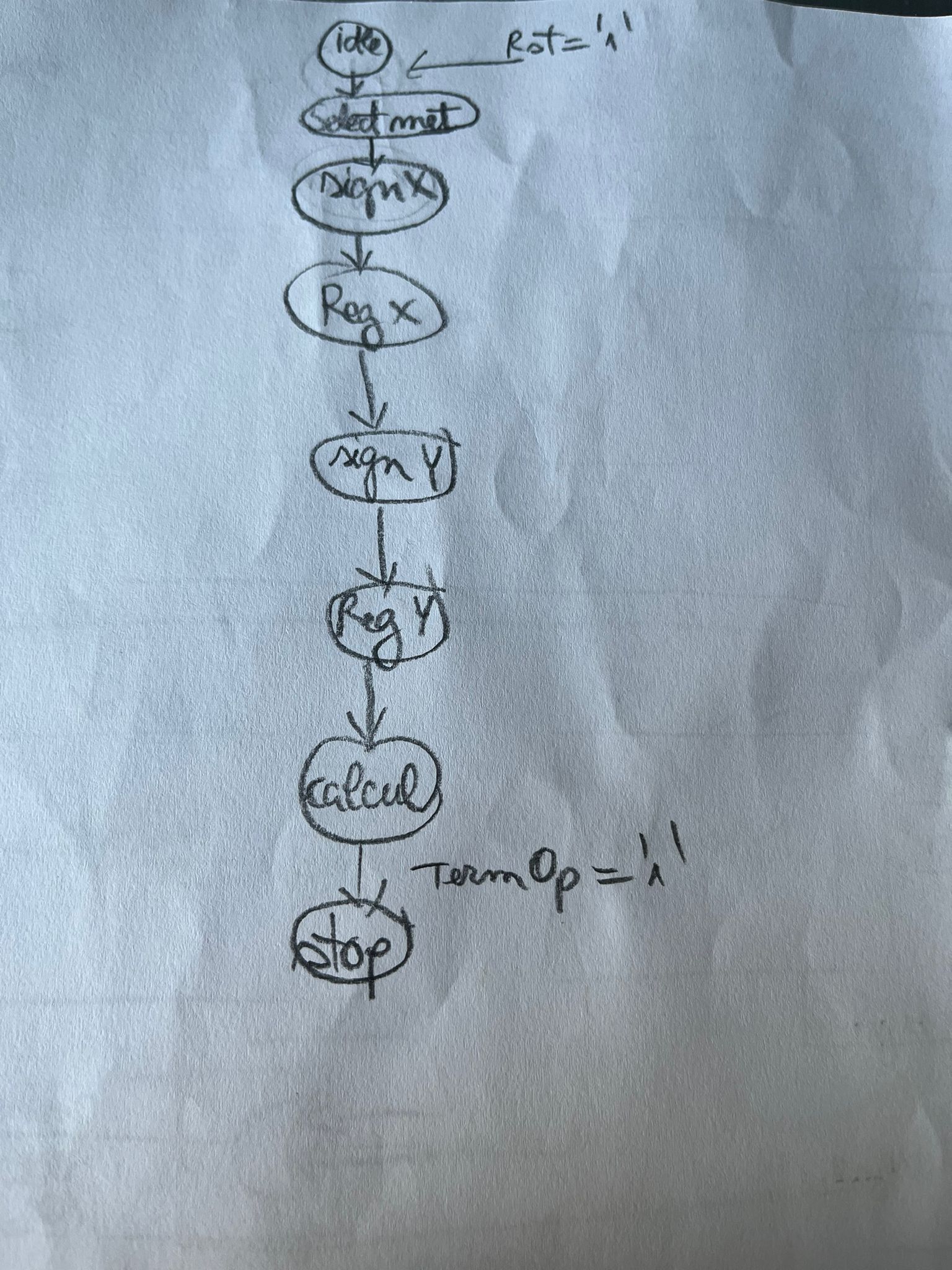
RST

Metoda

Schema bloc pentru acest sistem o putem observa in figura de mai sus. X si y sunt operanzii, clk pentru semnalul de ceas, rst pentru readucerea la 0 a operanzilor si selectarea metodei pe care o vom folosi. Rezultatul se afiseaza cu ajutorul anozilor si catozilor.

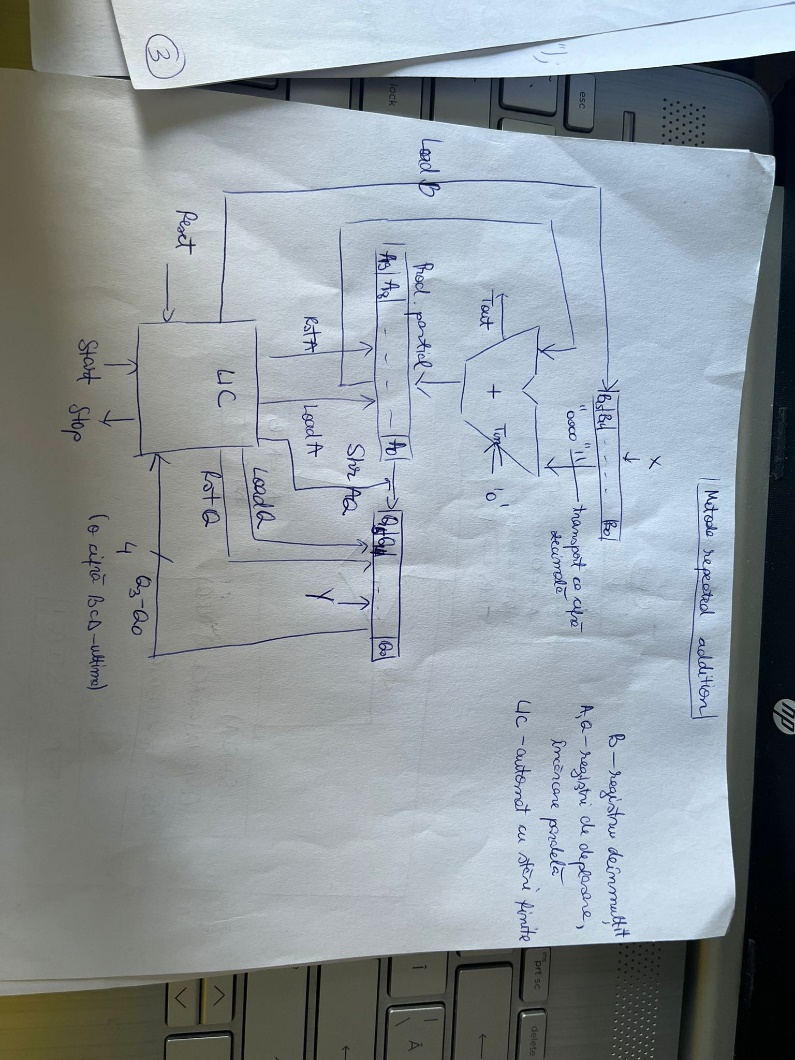
Unitatea de comanda a sistemului este un FSM, aceasta fiind responsabila pentru generarea semnalelor de care avem nevoie pentru activarea registrilor si circuitelor corespunzatoare.

* IDLE: starea de inceput
* SELMET: selecteaza metoda dorita
* INTRSIGNX: introducem semnul deinmultitului
* INTRX: introducem deinmultitul
* INTRSIGNY: introducem semnul inmultitorului
* INTRY: introducem inmultitorul
* CALC: in aceasta stare se calculeaza rezultatul
* STOP: starea finala a sistemului



* 1. **Metoda 1-Metoda adunarii repetate**

Schema metodei adunarii repetate este adaptata dupa circuitul de inmultire prin metoda Booth insa adaptata conditiilor noastre. Aceasta este prezentata in schema urmatoare:

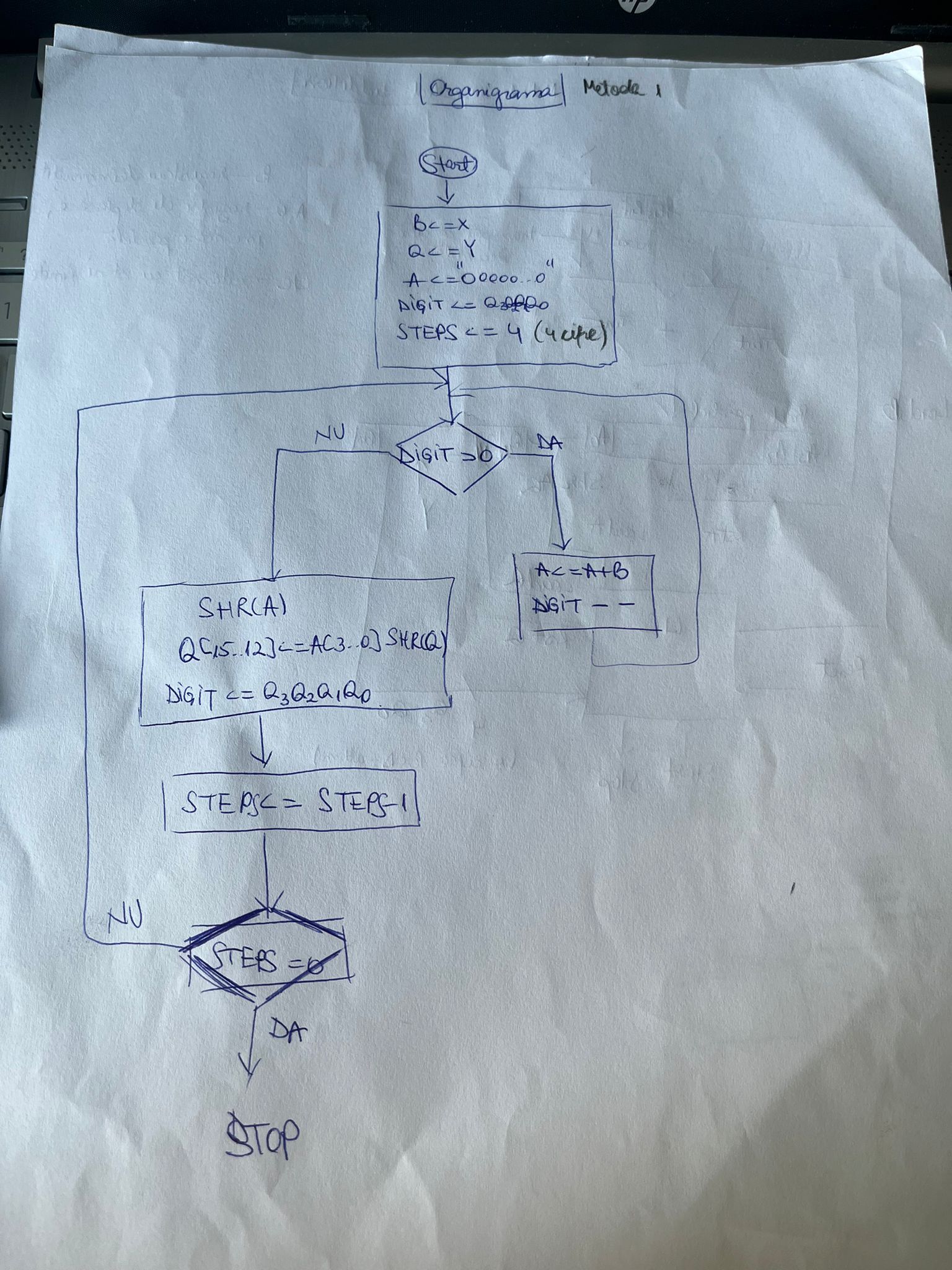


* B - registrul folosit pentru stocaarea lui X
* Q – registrul utilizat pentru incarcarea lui Y, initial
* AQ – utilizat pentru stocarea produsul partial
* Sumator zecimal pentru numere de 5 cifre pentru calculul produsului partial
* UC – unitatea de comanda-FSM care genereaza semnale pentru incarcare paralele, shiftare, terminare operatie, rst, start.

Algoritm folosit:

* Examinarea cifrelor inmultitorului, incepand din dreapta
* Registrul acumulator A aduna x de un nr de ori cat e cifra inmultitorului
* Shiftarea registrului AQ pentru analizarea urmatoarei cifrei a inmultitorului
* Se efectueaza pana cand nu mai sunt cifre ale inmultitorului

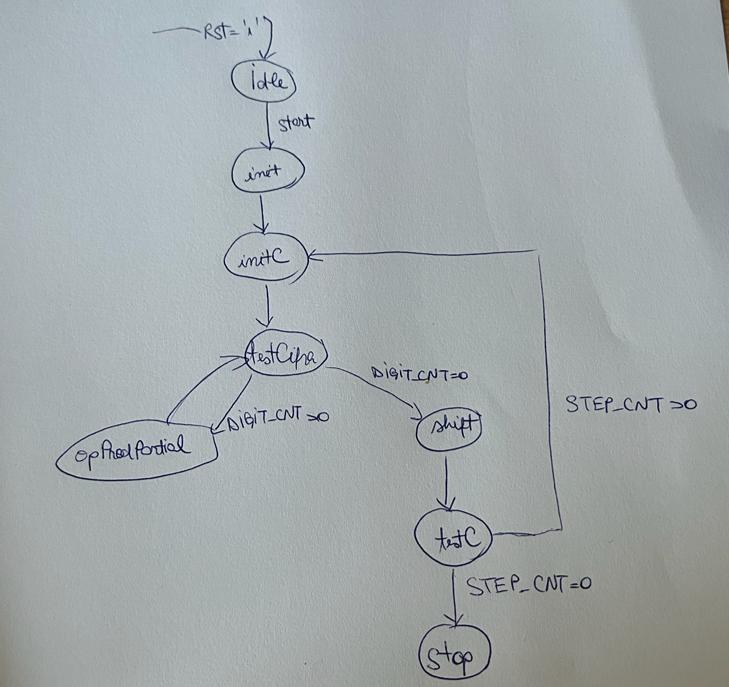
Organigrama:



Semnalul STEP\_CNT (STEPS in desen) este utilizat pentru decrementarea numarului de cifre ale inmultitorului(4 cifre) in timp ce se parcurge cifra unitatilor, zecimalor etc.

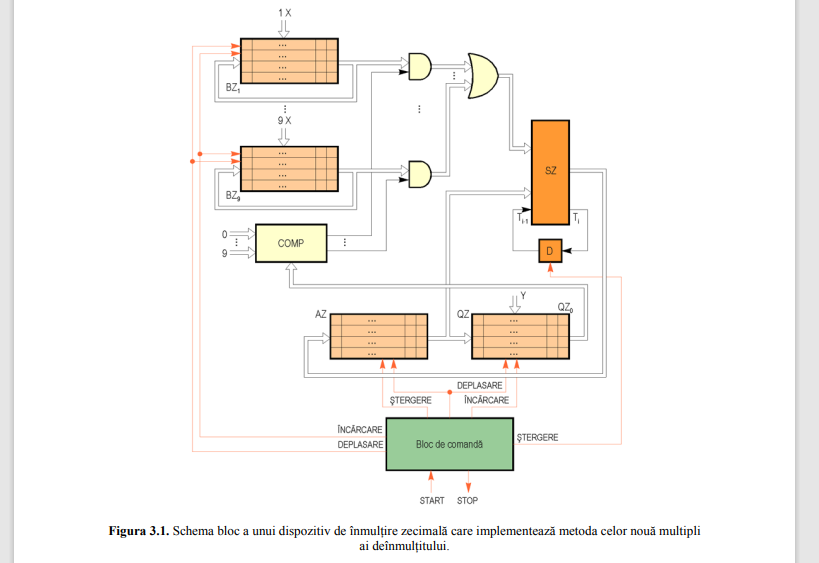
Semnalul DIGIT\_CNT (DIGIT in desen) este utilizat pentru decrementarea numarului de cate ori vom aduna la suma x, primind ca valoare ultima cifra a inmultitorului Y.

FSM-unitate de comanda si control



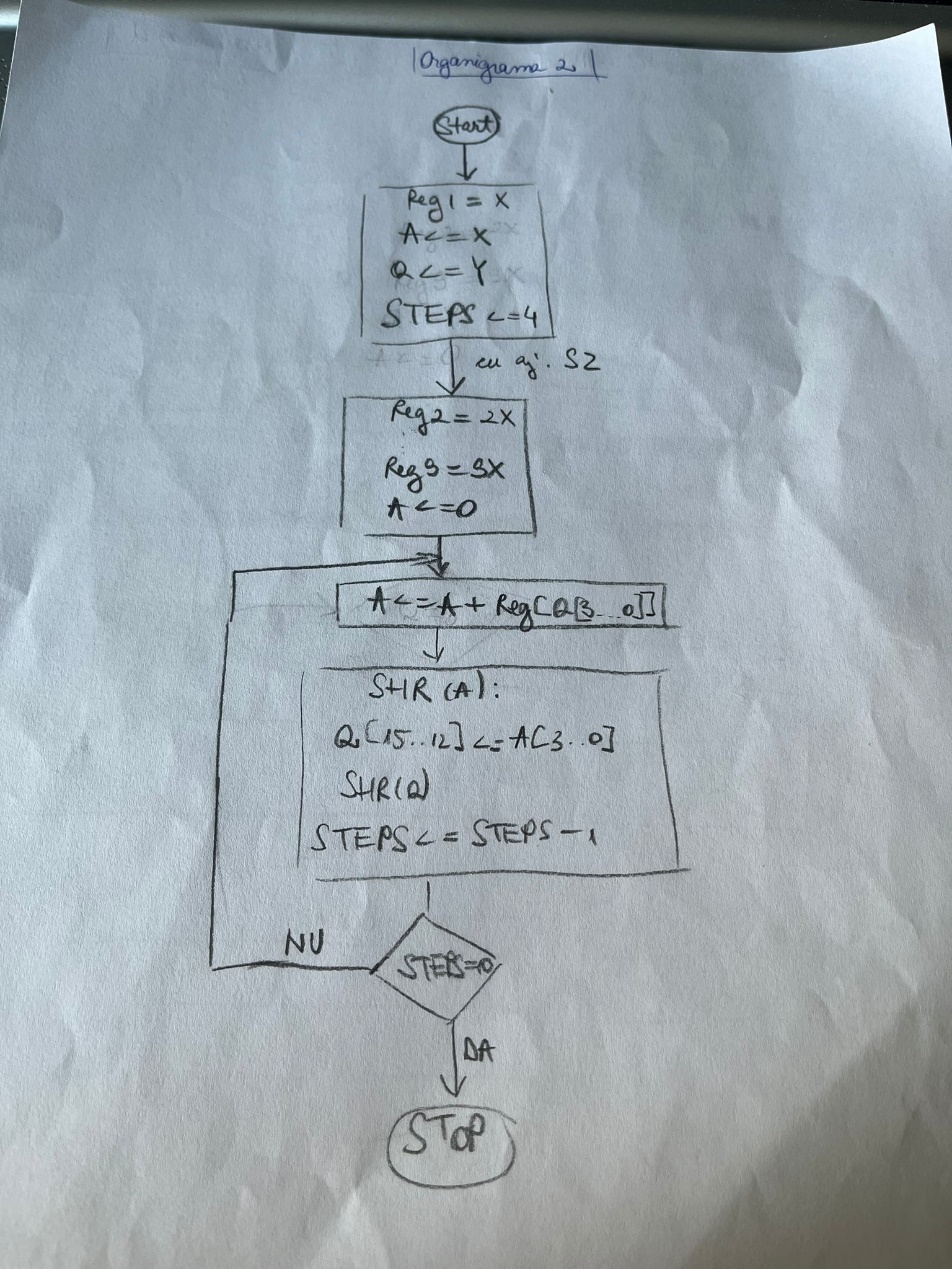
* Idle-starea initiala
* Init-se incarca registrii si bistabilele cu valorile initiale
* InitC-se initializeaza contorul pentru nr de adunari ce trebuie efectuate
* testCifra- se testeaza numarul de adunari care mai trebuie efectuate
* opProdPartial- starea in care se aduna deinmultitul la produsul partial existent
* shift- shiftarea AQ
* testC-verificare nr de cifre ale inmultitorului
* stop- stare finala dupa terminare operatie
  1. **Metoda 2-Metoda celor 9 multipli ai deinmultitului**

Schema metodei celor 9 multipli ai deinmultitului presupune generarea celor 9 multipli la inceput. Acestia sunt stocati in registrii speciali de cate 20 de biti, 5 cifre zecimale. Pe langa acestia mai exista sumatorul zecimal care aduna de fiecare data una dintre valorile din acesti registrii, in functie de comparatorul care compara ultima valoare (Q0) a inmultitorului cu registrul corespunzator al deinmultitului si aduna la produsul partial acea valoare din registru, apoi se shifteaza registrul AQ si se ocupa de urmatoarea cifra.

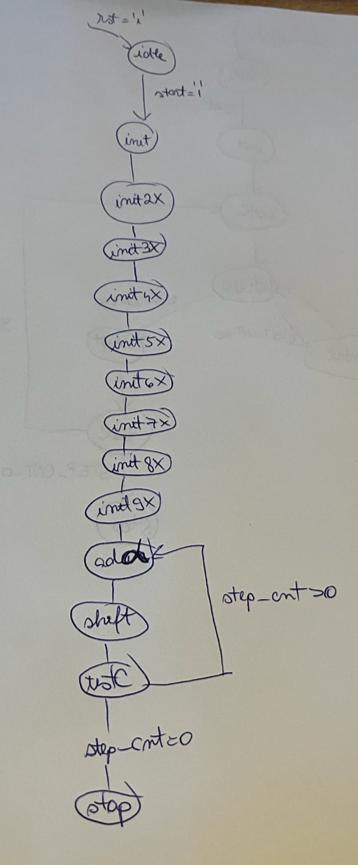
**

* AZ- grup de registri pentru stocarea produsului partial
* QZ- grup de registri pentru stocarea inmultitorului
* BZ1..BZ9- stocarea celor 9 multiplii ai deinmultitului
* SZ- sumator zecimal pentru calculul produsului partial
* D- latch pentru salvarea transportului
* COMP- compararea cifrei curente a inmultitorului cu indexul registrilor multiplicanti
* UC- FSM cu semnalele de incarcare, shift dreapta, stop, start

Organigrama sistemului:



FSM- unitatea de comanda si control



* Idle- starea initiala
* Init- se incarca reg Q si in reg A se salveaza x
* Init2x..init9x-se vor salva multiplii lui x
* Add- adunarea valorii din reg corespunzator ultimii cifre a inmultitorului din reg Q cu valoarea din registrul A; salvarea transportului in reg A, prima pozitie.
* Shift-shift AQ dreapta
* TestC-verificare nr de cifre ale inmultitoului
* Stop-stare finala
  1. **Manual de utilizare:**

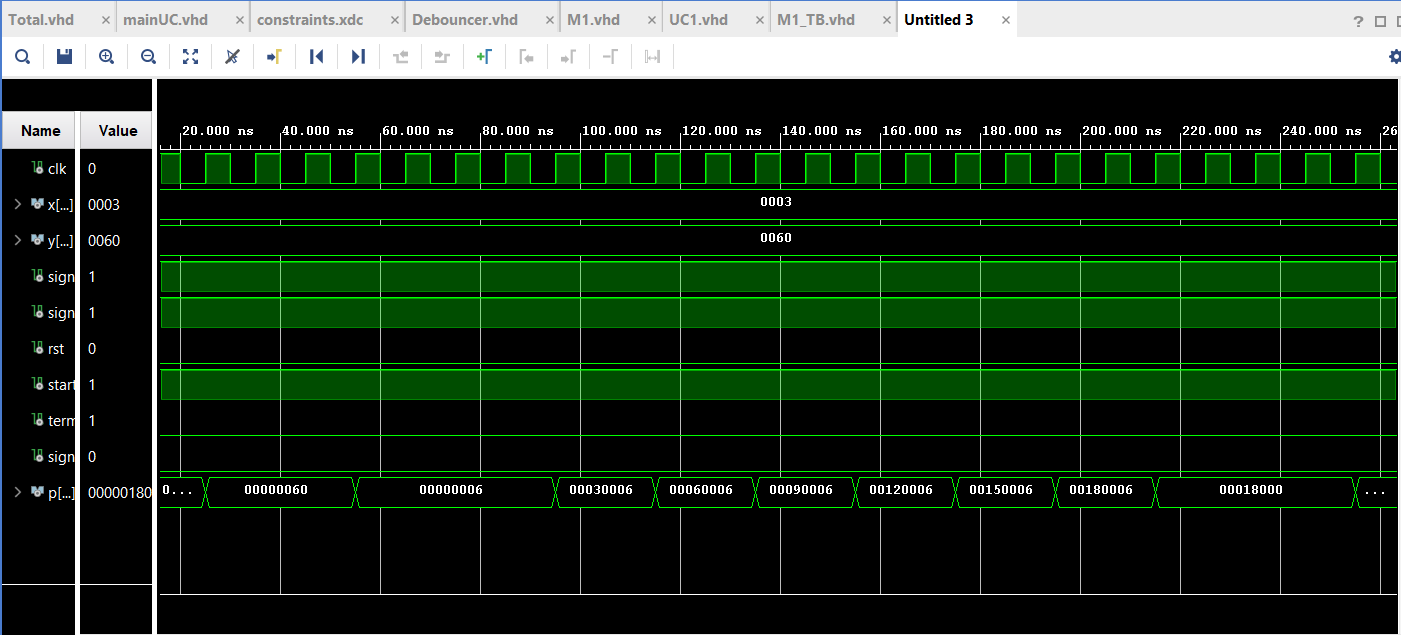
Se incarca codul pe placuta Nexys4, apoi avem butoanele de jos si dreapta. Butonul de jos se foloseste pentru resetare, iar butonul din dreapta pentru trecerea intre stari.

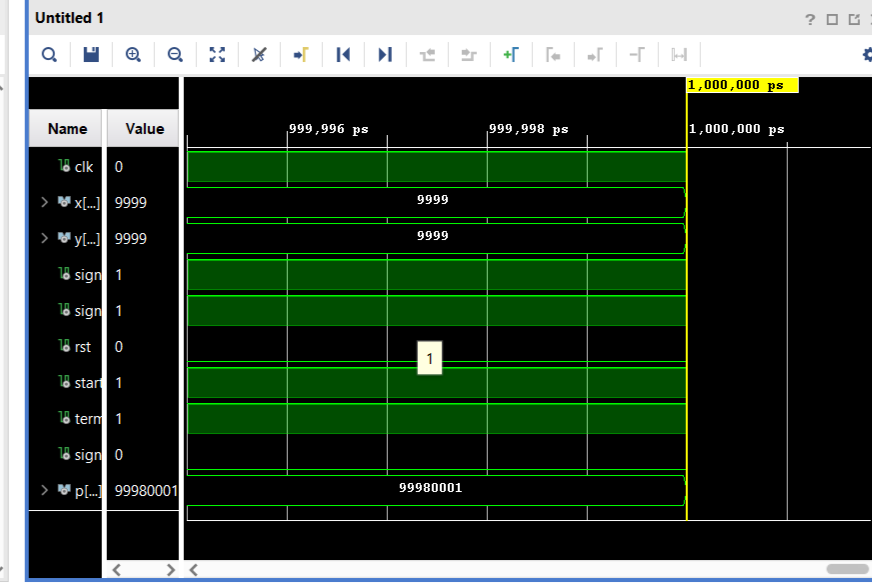
Initial avem starea idle, apoi se selecteaza metoda. Urmeaza sa introducem semnul lui x, apoi deinmultitul x, dupa care semnul lui y si inmultitorul ca y, apoi se realizeaza calculul si se afiseaza produsul sau concatenarea dintre x si y in functie de switchul 0. Primele 4 switchuri sunt folosite pentru cifra miilor, urmatoarele 4 pentru cifra sute, urmatoarele 4 pentru zeci si ultimele 4 pentru cifra unitatilor. Semnul se introduce prin ridicarea sau lasarea switch-ului 15.

1. **REZULTATE EXPERIMENTALE**

Proiectarea si simularea sistemului au fost realizate prin intermediul mediului de dezvoltare Vivado, unde am simulat rezultatele si le-am verificat. Mai jos voi atasa unele dintre rezultate pentru a ne asigura ca functioneaza cum trebuie:

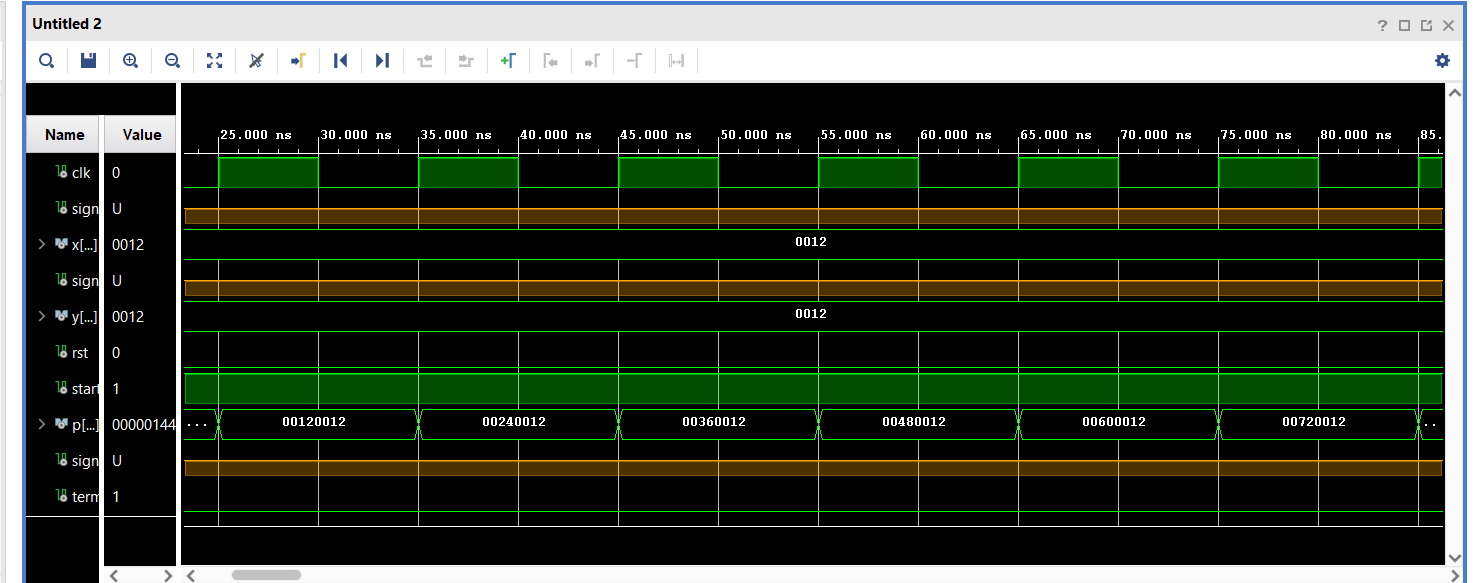
* 1. Metoda1



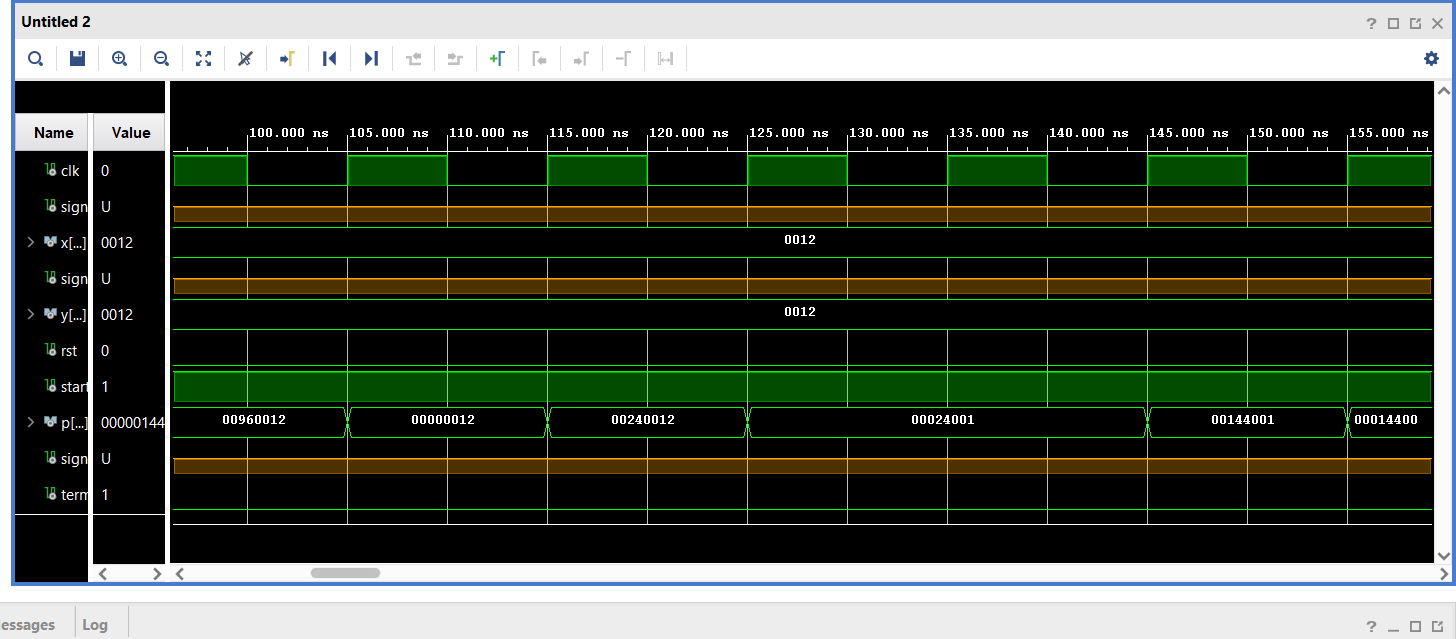


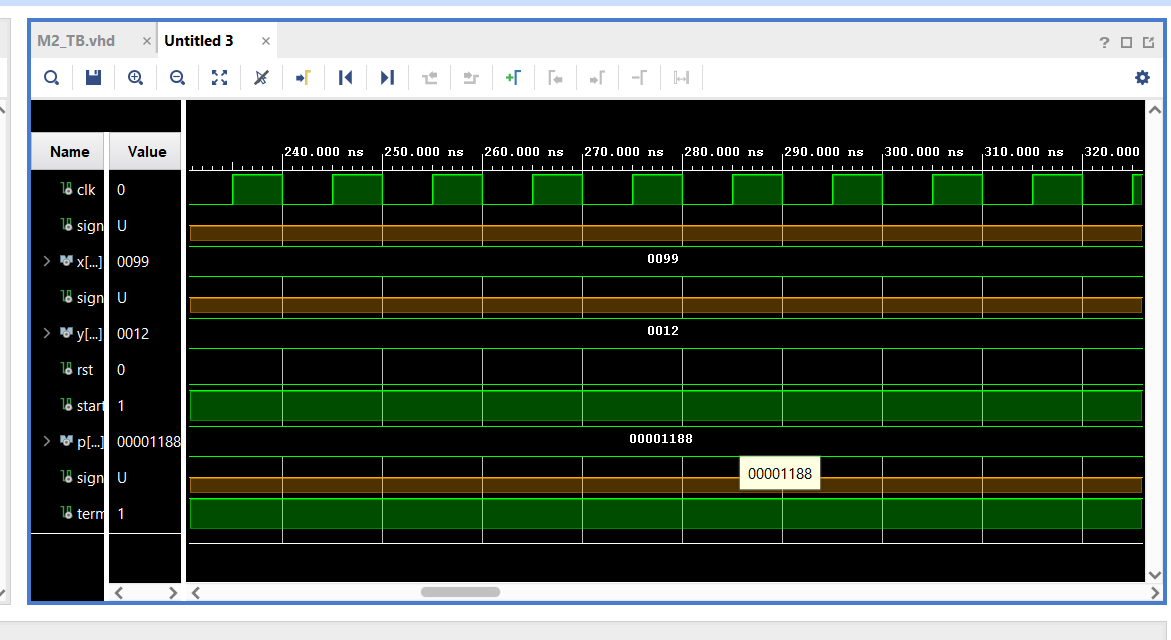
* 1. Metoda2

Generare multiplii



Calculare rezultat





S-au verificat rezultatele pe cazuri mici, dar si cu numere maxime si rezultatul este unul corect, verificarea tranzitiilor observandu-se si in pozele atasate mai sus, metodele functionand precum este prezentat in capitolul proiectare si implementare.

1. **CONCLUZII**

In urma realizarii acestui proiect, am invatat 2 metode de implementare a inmultirii zecimale: „Metoda adunarii repetate” si „Metoda celor 9 multiplii ai deinmultitului”. A fost impelementat pe placuta Nexys4 in cadrul laboratorului de SSC, utilizatorul fiind capabil sa aleaga ce metoda vrea sa foloseasca.

Metodele sunt implementate cu ajutorul limbajului de descriere hardware VHDL, in mediul Vivado. Unitatile de comanda dezvoltate au fost introduse in sectiunea de proiectare si implementare. Sumatorul zecimal a fost realizat cu ajutorul a mai multor sumatoare de 4 biti prin cascadarea lor.

Pe langa aceasta, in fisierul final se tine cont si de erorile ce pot aparea, o cifra nu poate fi mai mare de 9.

Acest proiect oferă un beneficiu semnificativ prin dezvoltarea abilității de a explica funcționarea unui sistem hardware complex. Un punct central este identificarea și elaborarea unităților de control care asigură funcționarea eficientă a circuitelor. De asemenea, proiectul implică aprofundarea înțelegerii algoritmilor hardware specifici, cu un accent pe algoritmii de înmulțire.

În plus, relevanța practică a acestui proiect este amplificată de legătura sa cu un sector în continuă expansiune, cu aplicații extinse în domenii precum economia, finanțele și tehnologia. Această inițiativă are potențialul de a minimiza erorile de calcul produse de algoritmii care folosesc sistemul binar. Prin urmare, contribuie direct la eficientizarea proceselor în domenii unde acuratețea datelor este crucială

1. **BIBLIOGRAFIE**

* [**https://www.tutorialspoint.com/8085-program-to-multiply-two-2-digit-bcd-numbers**](https://www.tutorialspoint.com/8085-program-to-multiply-two-2-digit-bcd-numbers)
* [**https://stackoverflow.com/questions/33986747/error-in-bcd-multiplication**](https://stackoverflow.com/questions/33986747/error-in-bcd-multiplication)
* [**https://stackoverflow.com/questions/59045345/multiplier-via-repeated-addition**](https://stackoverflow.com/questions/59045345/multiplier-via-repeated-addition)
* <https://www.khanacademy.org/math/cc-third-grade-math/intro-to-multiplication/imp-multiplication-intro/v/multiplication-as-repeated-addition>
* Laboratoarele de pe teams: SSC\_OLD\_LAB.PDF, L4-5,L6-7

1. **ANEXE**

* [**https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual?redirect=1**](https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual?redirect=1)