

***Facultate : Automatica si Calculatoare***

***Specializare : Calculatoare si Tehnologia Informatiei***

***Disciplina : Structura sistemelor de calcul***

***Indrumator : Lisman Florin***

***Proiect : Calculator de buzunar***

***Student : Ionas Andreea-Georgiana***

***Grupa : 30237***

# ***C******uprins***

Contents

[***Cuprins*** 2](#_Toc156168731)

[***Specificatia proiectului*** 3](#_Toc156168732)

[***Fundamente teoretice*** 3](#_Toc156168733)

[**Proiectare si implementare** 6](#_Toc156168734)

[***Rezultate Experimentale*** 10](#_Toc156168735)

[***Concluzii*** 12](#_Toc156168736)

[***Bibliografie*** 14](#_Toc156168737)

# ***Specificatia proiectului***

***Cerinta***:

Sa se proiecteze un calculator de buzunar care sa efectueze operatiile aritmetice fundamentale cum ar fi adunarea, scaderea, inmultirea si impartirea. Operanzii sunt reprezentati pe 8 biti, fara semn. Pentru a afisa rezultatul se va folosi afisorul SSD de pe placa.

Componenetele placii folosite la acest proiect sunt :

* Afisorul 7 segmente
* Switches (16)
* Butoane (4)

Pentru realizarea acestui proiect s-a folosit limabjul VHDL si programul Vivado.

# ***Fundamente teoretice***

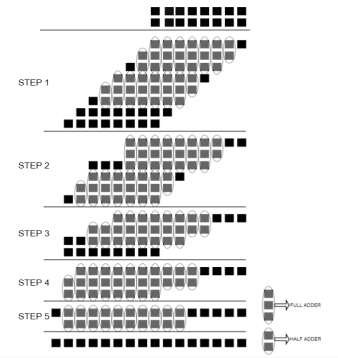
Algoritmii utilizati in implemenatrea metodelor de calcul vor fi descrise in detaliu si insotite de cod in [Proiectare si Implementare](#_Proiectare_si_implementare.).

Algoritmul pentru adunarea numerelor este unul de baza, sunt inlantuite(cascadate) mai multe sumatoare complete unde transportul out al unui sumator devine transportul in al urmatorului sumator.

Sumatorul are ca intrari 2 numere pe 8 biti, un semnal de carry in si un semnal de clk iar ca iesiri avem suma pe 16 biti si un semnal de carry out.

Algoritmul utilizat la scaderea numerelor este asemanator cu cel de adunare doar ca adunarea nu se realizeaza intre cele 2 numere ci intre primul numar si complementul fata de 2 al celui de-al doilea numar.

Algoritmul folosit la inmultirea celor 2 numere este numit si Arborele Wallace. Arborele Wallace este o metoda eficienta de a realiza inmultirea binara utilizand operatii in paralel. Aceasta tehnica implica impartirea numerelor binare in grupuri si efectuarea inmultirii pentru fiecare grup în paralel. Rezultatele partiale sunt apoi adunate pentru a obtine rezultatul final.



Algoritmul de impartire a celor 2 numere se bazeaza pe scaderea repetata a acestora. Algoritmul implementat utilizeaza un scazator complet pe 8 biti si un scazator complet pe 1 bit.

Acest algoritm functioneaza pe baza unei organigrame. In primul rand se initializeaza cele doua numere, catul si restul. Apoi se verifica daca cel de al doilea numar este egal cu zero, daca este zero nu se va efectua impartirea, in caz contrar se trece la urmatoarea stare. In cazul in care restul este mai mare decat impartitorul se opreste impartirea, in caz contrar se continua. In ultima stare se adauga la cat 1, iar la rest se atribuie valoarea obtinuta.

# 

# **Proiectare si implementare**

Calculatorul primeste ca intrari 2 numere reprezentate in binar pe 8 biti, fara semn. Rezultatul este un numar pe 16 biti care va fi afisat in hexazecimal.

Calculatorul este format din 3 principale componente fiecare avand un rol esential : calcul, conversie si afisare.

* ***Unitatea aritmetica - logica (ALU)***

Aceasta componenta este responsabila pentru partea de calcul. Primeste 2 numere pe 8 biti si un semnal de selectie pe 3 biti care indica operatia care trebuie efectuata. Rezultatul este pe 16 biti.

ALU este la randul sau alcatuita din 4 componente :

* Sumator
* Scazator
* Inmultitor
* Impartitor

Toate operatiile se efectueaza in acelasi timp, dar se afiseaza doar rezultatul indicat de semnalul de selectie.

**Adunarea:**

* Este realizata de un Carry Adder care consta in inlantuirea mai multor sumatoare complete, unde carry out-ul unui sumator devine carry in-ul urmatorului.

A screenshot of a computer code

Description automatically generated

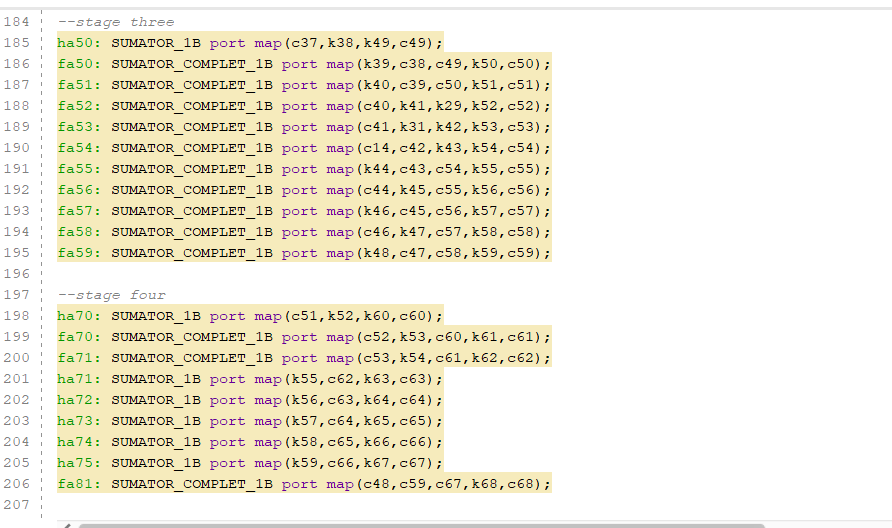
**Scaderea:**

* A screenshot of a computer code

  Description automatically generatedSe bazeaza pe acelasi principu ca adunarea, sunt inlantuite mai multe sumatoare complete unde carry out-ul unui sumator devine carry in-ul urmatorului. Principala diferenta este ca adunarea nu se face intre cele 2 numere ci se face intre primul numar si complementul celui de-al doilea numar.

**Inmultirea:**

* Arborele Wallace este un arbore de adunare in care partial produsele sunt generate prin „impartirea” inmultitorilor în grupuri si apoi adunate în mod succesiv. Aceasta permite efectuarea operatiilor in paralel, ceea ce face arborele Wallace eficient pentru implementari hardware de inmultire.



**Impartirea:**

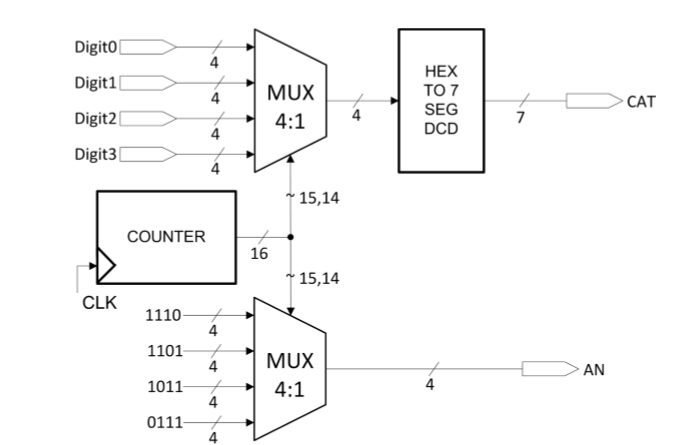
* Impartitorul este realizat conform unei organigame (scadere repetata) . Se verifica daca cel de-al doile numar este zero. Se fac scaderi repetate pana cand restul devine mai mare decat impartitorul.

A screenshot of a computer program

Description automatically generated

* ***Conversia si Afisarea :***

Aceaste componente realizeaza conversia rezultatului dat de ALU din binar in hexazecimal si afisarea acestuia pe SSD. Aceste 2 componente lucreaza in stransa legatura conform schemei alaturate.



# ***Rezultate Experimentale***

Am efectuat mai multe simulari pentru diferite seturi de numere pentru a verifica corectitudinea algoritmilor implementati.

* A = 10111110 (BE)
* B = 01000010 (42)

A screenshot of a computer

Description automatically generated

* A = 01110111 (77)
* A screenshot of a computer

  Description automatically generatedB = 00001101 (0D)
* A = 11000110 (C6)
* B = 00110101 (35)

A screenshot of a computer

Description automatically generated

# ***Concluzii***

Realizarea unui calculator de buzunar pentru operatii binare în VHDL este un proiect complex, dar educativ. O atentie deosebita la detalii, o proiectare modulara si o testare riguroasa sunt esentiale pentru succesul proiectului. În ciuda provocarilor, implementarea unui calculator de buzunar în VHDL ofera o oportunitate valoroasa pentru intelegerea mai profunda a limbajului VHDL si a conceptelor hardware asociate.

Principalele avantaje a unui calculator de buzunare sunt:

* Flexibilitate: Codul VHDL ofera flexibilitate si reutilizabilitate, permitand integrarea usoara a modulelor și extinderea functionalitatii calculatorului de buzunar.
* Operatii în Paralel: Inmultirea si impartirea pot fi realizate eficient în paralel, ceea ce poate duce la performante in implementarile hardware.
* Modularitate: Proiectul poate fi structurat în module independente pentru fiecare operatie, facilitand dezvoltarea, testarea si depanarea.

Viitoare posibilitati de dezvoltare a proiectului :  
 Adaugarea mai multor operatii precum ridicarea la putere, radical.

Testare mai riguroasa, adaugarea mai multor numere de test.

Posibilitatea introducerii numerelor de la o tastatura nu de la switch.

Manual de utilizare:

Se introduc datele de la switchurile placutei, primele 8 switch-uri reprezinta primul numar iar urmatoarele 8 cel de al doilea numar.

Se apasa butonul specific operatiei dorite.

Pe afisor va aparea rezultatul obtinut, in functie de operatia dorita.

# ***Bibliografie***

* <https://vlsiverify.com/verilog/verilog-codes/wallace-tree-multiplier>
* <https://www.ijvdcs.org/uploads/546213IJVDCS9984-85.pdf>
* <https://circuitfever.com/ripple-carry-adder>
* <https://github.com/ioana32/Pocket-Calculator>
* <https://eee.guc.edu.eg/Courses/Electronics/ELCT201%20Digital%20Logic%20Design/Tutorials/Tutorial5Solution-ELCT201.pdf>
* <https://users.utcluj.ro/~baruch/media/ssc/curs/SSC-Adunare.pdf>