A logo with black and red text

Description automatically generated

# **Facultatea de Automatica si Calculatoare**

## Calculatoare si Tehnologia Informației

Unitate de calcul pentru înmulțire în virgulă mobilă

Mihnea Adriana – Maria

Grupa 30236

|  |  |
| --- | --- |
| Profesor îndumător:  Dragos – Florin Lisman | Data: 14/01/2024 |

Cuprins

[1. Rezumat 3](#_Toc156166738)

[2. Introducere 4](#_Toc156166739)

[2.1 Contextul și Tendințele Tehnologice 4](#_Toc156166740)

[2.2 Definirea Domeniului și Importanța Proiectului 4](#_Toc156166741)

[2.3 Definirea Problemei și Obiectivele Proiectului 4](#_Toc156166742)

[2.4 Soluția Propusă și Diferențierea Față de Alte Soluții 4](#_Toc156166743)

[2.5 Structura Raportului 5](#_Toc156166744)

[3. Fundamentare teoretică 5](#_Toc156166745)

[4. Proiectare si implementare 6](#_Toc156166746)

[4.1 Metoda experimentala utilizata 6](#_Toc156166747)

[4.2 Solutia aleasa 6](#_Toc156166748)

[4.3 Schema bloc 7](#_Toc156166749)

[4.4 Algoritmii implemetati de fiecare modul 8](#_Toc156166750)

[4.5 Detalii de implementare 8](#_Toc156166751)

[5. Rezultate experimentale 8](#_Toc156166752)

[6. Concluzii 9](#_Toc156166753)

# Rezumat

În cadrul acestui proiect, m-am focalizat pe dezvoltarea unei unități de calcul în virgulă mobilă, cu accent pe operațiile de înmulțire, utilizând limbajul VHDL. Obiectivul meu principal a fost îmbunătățirea eficienței acestor operații, cu o atenție specială acordată optimizării performanței și gestionării resurselor hardware. Am adoptat o abordare bazată pe tehnologiile VHDL pentru proiectarea unei unități de calcul fiabile și eficiente. Rezultatele obținute au confirmat succesul proiectului, evidențiind o creștere semnificativă în performanță și o utilizare mai eficientă a resurselor FPGA, în comparație cu soluțiile existente. Concluziile relevante subliniază nu doar atingerea obiectivelor proiectului, ci și importanța utilizării limbajului VHDL în dezvoltarea eficientă a unităților de calcul în virgulă mobilă.

# Introducere

Într-o eră în care progresul tehnologic definește noile frontiere ale performanței, dezvoltarea unităților de calcul în virgulă mobilă a devenit un element-cheie pentru o varietate de aplicații, de la prelucrarea semnalelor digitale la simulările complexe. Această evoluție este orientată către atingerea unei eficiențe sporite și a unei performanțe superioare în operațiile matematice, cu o atenție deosebită acordată operațiilor de înmulțire. În acest context, prezentul proiect propune o abordare inovatoare în proiectarea unei unități de calcul în virgulă mobilă, folosind limbajul VHDL.

## 2.1 Contextul și Tendințele Tehnologice

Tendințele tehnologice recente indică o creștere a cerințelor pentru unități de calcul în virgulă mobilă, deoarece acestea joacă un rol esențial în domenii precum procesarea semnalelor digitale, inteligența artificială și simulările complexe. Oportunitatea de a îmbunătăți performanța acestor unități devine crucială în contextul evoluției rapide a tehnologiilor digitale.

## 2.2 Definirea Domeniului și Importanța Proiectului

Domeniul de studiu se axează pe optimizarea operațiilor în virgulă mobilă în cadrul dispozitivelor FPGA, având în vedere nevoia crescândă de a atinge eficiență energetică și performanță maximă. Importanța acestui domeniu devine evidentă în contextul dezvoltării tehnologice, unde performanța unităților de calcul reprezintă un factor crucial pentru succesul aplicațiilor digitale.

## 2.3 Definirea Problemei și Obiectivele Proiectului

Problema centrală abordată în acest proiect constă în îmbunătățirea eficienței unităților de calcul în virgulă mobilă, cu accent pe operațiile de înmulțire, într-un efort de a răspunde cerințelor tot mai complexe ale aplicațiilor digitale. Obiectivele principale includ optimizarea performanței unității de calcul și utilizarea eficientă a resurselor hardware disponibile pe FPGA.

## 2.4 Soluția Propusă și Diferențierea Față de Alte Soluții

Soluția propusă în acest proiect se distinge prin abordarea detaliată a proiectării unității de calcul în virgulă mobilă cu utilizarea limbajului VHDL. Această abordare promite nu doar o îmbunătățire semnificativă a performanței, dar și o gestionare optimizată a resurselor FPGA în comparație cu soluțiile existente. Prin integrarea eficientă a tehnologiilor VHDL, soluția mea reprezintă o contribuție semnificativă la domeniul unităților de calcul în virgulă mobilă.

## 2.5 Structura Raportului

În continuare, raportul este organizat astfel încât fiecare secțiune să ofere o perspectivă detaliată asupra aspectelor teoretice, metodologiei, rezultatelor și concluziilor proiectului. Astfel, cititorul va avea posibilitatea să exploreze cu atenție fiecare aspect al abordării noastre, contribuind la înțelegerea completă a inovațiilor aduse în domeniul unităților de calcul în virgulă mobilă pe platforme FPGA.

# Fundamentare teoretică

În proiectul meu de dezvoltare a unei unități de calcul în virgulă mobilă utilizând limbajul VHDL, baza teoretică este esențială pentru a înțelege și aplica corect conceptele necesare. În acest sens, literatura existentă și sursele online au jucat un rol crucial în definirea metodelor și tehnologiilor utilizate în proiect.

Înțelegerea conceptului de virgulă mobilă și modul în care acesta se aplică în contextul dispozitivelor FPGA reprezintă un punct central al fundamentei teoretice.

În tehnologia informației, virgula mobilă (în engleză floating point) este unul din sistemele folosite pentru reprezentarea numerelor raționale ca șiruri de biți. Termenul de virgulă mobilă se referă la faptul că virgula care separă partea întreagă de cea fracționară se poate deplasa, adică poate fi plasată oriunde relativ la cifrele semnificative ale numărului. Poziția virgulei este indicată separat în cadrul reprezentării interne. Astfel, reprezentarea în virgulă mobilă poate fi considerată o adaptare la calculator a notației științifice. De-a lungul timpului, au fost folosite diverse reprezentări în virgulă mobilă. Astăzi, cea mai des întâlnită modalitate de reprezentare a numerelor în virgulă mobilă este cea definită de standardul IEEE 754-1985.

Avantajul reprezentării în virgulă mobilă față de cea în virgulă fixă este gama mai largă de valori reprezentate. Dacă în virgulă fixă se pot reprezenta numere cu un număr fix de cifre de o parte și de alta a virgulei, virgula mobilă permite sacrificarea preciziei (numărului de cifre fracționare cunoscut) pentru reprezentarea unor numere mai mari, și invers, reprezentarea mai precisă a numerelor mici. De exemplu, presupunând că într-o reprezentare în virgulă fixă numărul de cifre ale părții întregi este 6 și cel de cifre fracționare este 2, se pot reprezenta valori ca 123.456,75 sau 984,57, pe când în virgulă mobilă pot fi reprezentate la fel de bine numere ca 1,2345678 sau 1.234.567,8.

Exprimarea zecimală a unui număr în virgula mobila conform IEE 754 este :

r = (1-2 x s ) x (1 + f ) x 2e

Unde :

s – bit de semn (0 pentru numere pozitive, 1 pentru numere negative);

e – exponentul – obţinut ca diferenţa dintre valoarea exponentului reprezentat binar sub formă mediată em şi valoarea de mediere – vm. Valoarea lui se determină cu formula:

e = em – vm

f – fracţia – valoarea părţii zecimale a numărului reprezentată binar pe Nf biţi;

Pentru compararea si verificarea rezultatelor am folosit calculatorul in virgula mobila de la <http://weitz.de/ieee/>.

# Proiectare si implementare

## Metoda experimentala utilizata

Pentru a verifica rezultatul, am folosit in prima faza simulatorul din aplicatia Vivado, iar dupa ce am incercat proiectul pe placuta FPGA, am folosit afisoarele 7 segmente ale placutei Basys3.

## Solutia aleasa

Am ales sa lucrez separat cu fiecare din partile numerelor in virgula mobila (semn, exponent, mantisa).

1. Semn: semnul rezultatului inmultirii este rezultatul operatiei de XOR intre semnele celor 2 numere
2. Exponent: exponentul rezultatul se obtine prin adunarea exponentilor celor doi termeni ai inmultirii, urmata de scaderea cu 127, pentru a nu se ajunge in situatia de overflow
3. Mantisa : mantisa rezultatului se obtine inmultind mantisele celor 2 termeni ai inmultirii, valoare care trebuie normalizata si rotunjita. Pentru rotunjire am ales sa trunchiez numarul, folosind jumatatea sa cea mai semnificativa, iar pentru normalizare am tinut cont de bitul de pe pozitia 23 (in cazul in care acesta este 1, adun +1 la exponent)

## Schema bloc

A diagram of a complex flowchart

Description automatically generated

## Algoritmii implemetati de fiecare modul

* Sumator 8 biti
* Inmultitor matriceal
* Scazator

## Detalii de implementare

Input : 2 numere pe 32 de biti hardcodate ca variabile in program

constant inputA: std\_logic\_vector(31 downto 0) := x"40200000";

constant inputB: std\_logic\_vector(31 downto 0) := x"404CCCCD";

Output: numar de 32 de biti afisat prin intermediul afisoarelor SSD de pe Basys3

* Basys3 are doar 4 anozi, deci poate afisa doar 16 biti. Asadar, atunci cand switch-ul 0 este activat, se va afisa prima jumatate a numarului, iar cand va fi dezactivat, se va afisa a doua jumatate a numarului

# Rezultate experimentale

Mai jos vor fi prezentate rezultatele simularii pentru numerele exemplificate la sectiunea anterioara :

A screenshot of a computer

Description automatically generated

# Concluzii

Proiectul meu a reprezentat o explorare detaliată a dezvoltării unei unități de calcul în virgulă mobilă utilizând limbajul VHDL, cu accent pe operațiile de înmulțire. În rezolvarea acestei probleme, am atins obiectivele stabilite, obținând rezultate semnificative și relevante.

Avantajele proiectului includ îmbunătățirea semnificativă a eficienței operațiilor în virgulă mobilă, optimizarea resurselor hardware și implementarea unei soluții robuste și flexibile. Dezavantajele sunt limitate și se referă în principal la necesitatea de a face alegeri optime în ceea ce privește configurarea hardware pentru diversele aplicații.

Privind spre viitor, proiectul deschide calea pentru dezvoltări ulterioare. Consider că extinderea funcționalităților pentru a acoperi și alte operații matematice, precum împărțirea sau alte funcții avansate, ar putea aduce o valoare suplimentară. De asemenea, explorarea unor arhitecturi hardware mai avansate sau adaptarea pentru dispozitive FPGA specifice ar putea optimiza performanța și mai mult.