

Universitatea Tehnica din Cluj-Napoca
Catedra de Calculatoare
Titlul proiectului Unitate aritmetica si logica
Student Mitroi Bianca
Numarul grupei 5
Indrumator Gherman Filip
Data 21 decembrie 2022

Cuprins

Contents

1	Introducere	3
1.1	Obiectivele proiectului	3
2	Fundamentare teoretica	3
3	Proiectare si implementare	3
3.1	Principii abordate	3
3.2	Schema bloc	4
3.3	Specificari de utilizare	4
4	Rezultate experimentale	5
4.1	Resurse	5
5	Concluzii	5
6	Rezumat	6

1 Introducere

1.1 Obiectivele proiectului

Modelarea operatiilor aritmetice elementare (+, -, *, /) pe un sistem digital. Se cere sa se proiecteze un sistem care primeste ca date de intrare doua numere fara semn si un operator din cei mentionati anterior si returneaza rezultatul operatiei corespunzatoare dintre cei 2 operanzi.

2 Fundamentare teoretica

- Atat pentru adunare cat si pentru scadere se pot folosi mai multe tehnici, printre care: sumator cu propagarea transportului (elementar daca este pe un singur bit per operand), sumator cu anticiparea transportului (are blocuri suplimentare pentru generare si propagare, pe langa sumatoare elementare), sumator cu salvarea transportului in care suma si transportul pe fiecare bit sunt separate in doi termeni ce poti fi adunati cu un sumator cu anticipare sau cu propagare, sumator zecimal in care operanzii sunt impartiti in grupuri de biti pe care sa se poata reprezenta cate o cifra zecimala, etc.
- Pentru inmultire se poate folosi un arbore Wallace care realizeaza operatia combinational (nu necesita iteratii de calcul), sau orice tehnica ce necesita secventiere care furnizeaza rezultate pariale ale produsului pana ce procesul de calcul se incheie (ex. metoda Booth - care faciliteaza calculul in cazul numerelor negative). Se mai poate folosi si inmultirea matriceala in care in cascada se aduna rezultate pariale pe cate un bit care se grupeaza pe bitii rezultatului dupa ce toti bitii din ambele numere au fost procesati.
- Pentru impartire se pot folosi tehnicile in care se reface restul partial, respectiv in care nu se reface decat atunci cand este necesar (ambele tehnici se realizeaza secvential si imprumuta principii de la inmultirea secventiala).

3 Proiectare si implementare

3.1 Principii abordate

Drept concepte de design al proiectului, am decis sa folosesc urmatoarele notiuni:

- Pentru adunare am folosit un sumator cu anticiparea transportului (in care transportul este calculat cu ajutorul altor semnale de propagare si generare a transportului)
- Pentru scadere am folosit acelasi principiu ca pentru adunare, cu diferenta ca al doilea operand (scazatorul) este mai intai transformat in complement fata de 2 si apoi prelucrat.
- Pentru inmultire am folosit arborele Wallace in care se aduna - in grupuri de cate 3, prin sumatoare cu salvarea transportului - numere care sunt formate din and-uri intre fiecare cifra a primului termen si toate cifrele celui de-al doilea numar. Rezultatele adunarilor sunt adunate recursiv pana se obtine un singur numar care va fi rezultatul inmultirii numerelor initiale.
- Pentru impartire am folosit tehnica in care nu se reface restul partial. In mai multe iteratii (in functie de cat de multi biti stocheaza deimpartitul si impartitorul), din deimpartit se scade sau se aduna impartitorul in functie de un anumit semnal care se schimba (ne spune daca trebuie sa se faca adunare sau scadere). Rezultatul impartirii se pune in felul urmator: restul impartirii este pus in prima jumătate a rezultatului, iar catul in a doua.

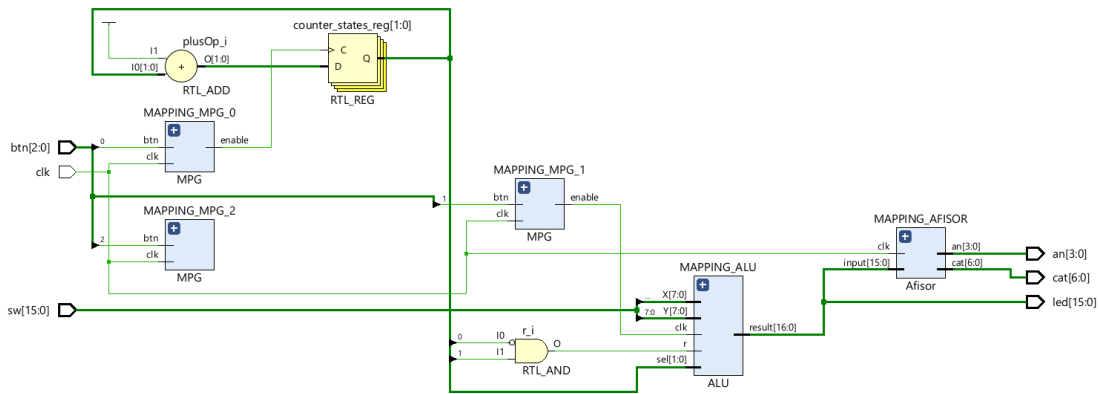


Figure 1: Elaborated design for Main

3.2 Schema bloc

Inputuri: 3 butoane (pe cate un bit), clock (pe un bit) , 16 switch-uri (16 biti)

Outputuri: anozii(4 biti), catozii(7 biti), leduri(16 biti)

Descriere:

- primele 8 switch-uri de la stg la dr - introducerea primului operand
- urmatoarele 8 switchuri - al doilea operand
- butonul din mijloc (btn(0)) - trece prin operatii
- butonul de sus (btn(1)) - clock pentru impartitor
- ledul 1 de la dr la stg - se aprinde daca al doilea operand este 0 si se face operatie de impartire
- ledul 0 de la dr la stg - carry out

Specificatii: Opernazzii sunt pe 8 biti fiecare. Rezultatul furnizat de ALU este pe 17 biti.

Biblioteci folosite:

- use IEEE.NUMERIC STD.ALL;
- use IEEE.std logic signed.ALL;
- use IEEE.std logic unsigned.ALL;
- use IEEE.STD LOGIC 1164.ALL;

3.3 Specificari de utilizare

La prima utilizare, SSD-ul va afisa rezultatul adunarii dintre operanzii deja pusi pe switch-uri in prealabil. La prima apasare de buton central, se va afisa rezultatul scaderii dintre operanzii de pe switch-uri. La urmatoarea apasare se va afisa rezultatul inmultirii operanzilor. Pentru a se afisa impartirea trebuie sa se apese butonul din partea de sus a gruparii de butoane de pe placa Basys 3 pentru a se reseta circuitul de impartire. La urmatoarea apasare de buton central se va trece in modul impartire. Se va afisa 0 sau un alt numar care nu corespunde cu rezultatul final al impartirii. Pentru a se ajunge la rezultat se apasa succesiv butonul de sus pana valoarea de pe SSD nu se mai schimba (se poate apasa cam de 5 ori pentru siguranta). Valoarea afisata este compusa din rest si cat.

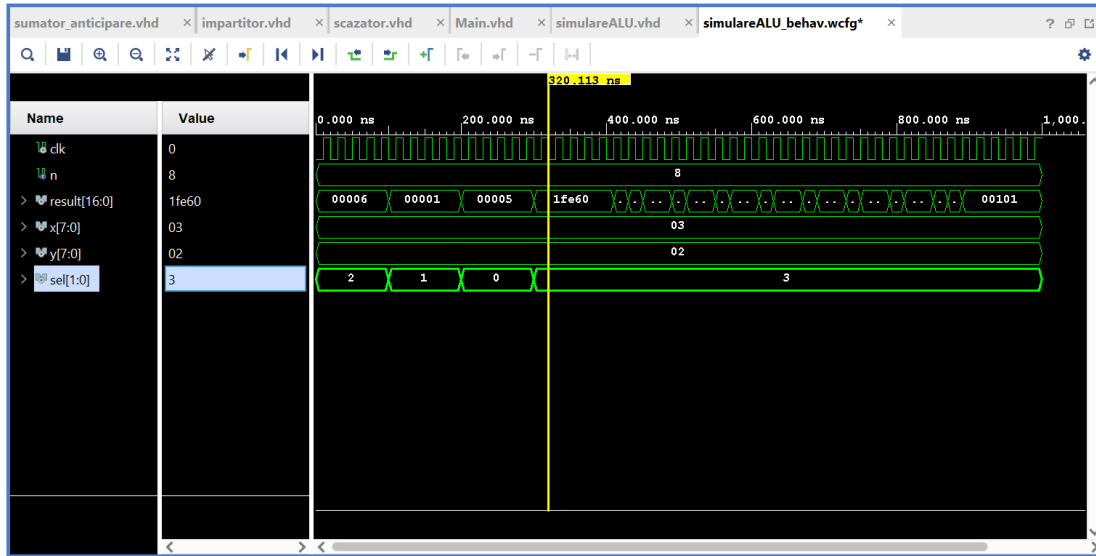


Figure 2: Simulation results

4 Rezultate experimentale

4.1 Resurse

mediu de dezvoltare

limbaj

sistem de operare folosit

componente fizice folosite

fisiere sursa

(cu extensia .vhd)

fisier de simulare (.vhd)

fisier de constrangeri

- Xilinx Vivado 2021.2

- VHDL

- Windows 11

- Laptop

- placuta e dezvoltare Basys 3

- Main (face legatura intre ce furnizeaza ALU si interfata placii)

- Afisor (pentru afisarea rezultatului operatiilor pe SSD-ul placii)

- ALU (inglobeaza componentele in care sunt implementate operatiile)

- sumator anticipare (pentru operatia de adunare)

- scazator (pentru operatia de scadere)

- inmultitor (pentru operatia de inmultire)

- impartitor (pentru operatia de impartire)

- sumator salvare (instantiat de inmultitor)

- sumator mic (sumator pe 2 biti, instantiat in sumator salvare

si sumator anticipare)

- MPG (monopuls generator) (pentru a se putea utiliza butoanele)

- simulareALU (pentru a testa functionalitatile ALU)

- constrangeri.xdc

5 Concluzii

Obiectivele proiectului au fost atinse cu success.

S-au modelat operatiile aritmetice elementare atat pe placuta de dezvoltare, cat si in simulare.

6 Rezumat

Prin implementarea de componente relativ mici ca si complexitate (sumatorul pe 2 biti sau MPG) si instantierea acestora in componente mai mari (in sumator, inmultitor, etc.) s-a reusit implementarea unui intreg sistem de calcul ce deserveste obiectivelor proiectului.

Bibliografie

- [1] <https://users.utcluj.ro/~baruch/ro/pages/cursuri/structura-sistemelor-de-calcul/laborator.php>
- [2] <https://users.utcluj.ro/~baruch/ssc/proiect/Ghid-Proiect.pdf>