

**Multiplicator de numere complexe**

**Student:**

**Gheorghe Ștefan**

**BRAȘOV, 2020**

Cuprins

[1 Prezentare generală 2](#_Toc36630162)

[2 Implementare cu un singur modul de multiplicare 2](#_Toc36630163)

[2.1 Arhitectură 2](#_Toc36630164)

[2.2 Logica de control 3](#_Toc36630165)

[2.3 Scenarii de test și forme de undă 4](#_Toc36630166)

[3 Implementare cu două module de multiplicare 5](#_Toc36630167)

[3.1 Arhitectură 5](#_Toc36630168)

[3.2 Logica de control 6](#_Toc36630169)

[3.3 Forme de undă obținute 7](#_Toc36630170)

[4 Implementare cu patru module de multiplicare 7](#_Toc36630171)

[4.1 Arhitectură 7](#_Toc36630172)

[4.2 Logica de control 8](#_Toc36630173)

[4.3 Forme de undă obținute 9](#_Toc36630174)

# Prezentare generală

Circuitul implementat realizează înmulțirea a două numere complexe reprezentate sub forma algebrică.

Părțile reale și imaginare ale operanzilor sunt numere întregi reprezentate pe 8 biți, în complement față de 2.

În acest document sunt prezentate trei variante de implementare, fiecare cu un grad diferit de paralelism.

# Implementare cu un singur modul de multiplicare

## Arhitectură

În Figura 1 este prezentată arhitectura modulului. Tabelul 1 prezintă interfețele modulului, semnalele interne și semnificația acestora.



Figura 1 : Arhitectura modulului implementat

Tabel 1 : Prezentarea interfețelor și semnalelor interne.

|  |  |  |
| --- | --- | --- |
| Denumire | Tip | Explicatie |
| op\_val | I | Semnalizează faptul că operanzii sunt gata de preluat de către modul. |
| sw\_rst | I | Reset sincron al modulului. |
| op\_ready | O | Semnalizează faptul că modulul este pregătit pentru a primi noi operanzi. |
| op\_data | I | Datele de intrare ale modulului. |
| res\_ready | I | Modulul Master este gata de a primi rezultatul. |
| res\_val | O | Rezultatul multiplicarii este pregătit pentru a fi transmis. |
| res\_data | O | Datele de ieșire ale modului. |
| op\_1\_sel | Intern | Semnalul de selecție pentru primul operand al modulului de multiplicare. |
| op\_2\_sel | Intern | Semnalul de selecție pentru al doilea operand al modulului de multiplicare. |
| res\_reg\_sel | Intern | Semnalul de selecție a registrului în care va fi stocat rezultatul multiplicării curente. |
| compute\_enable | Intern | Semnal de enable pentru realizarea adunării și scăderii finale. |

## Logica de control

În Figura 2 este prezentat graful de tranziții al modulului implementat. Tabelul 2 conține o scurtă explicație a fiecărei stări în parte.

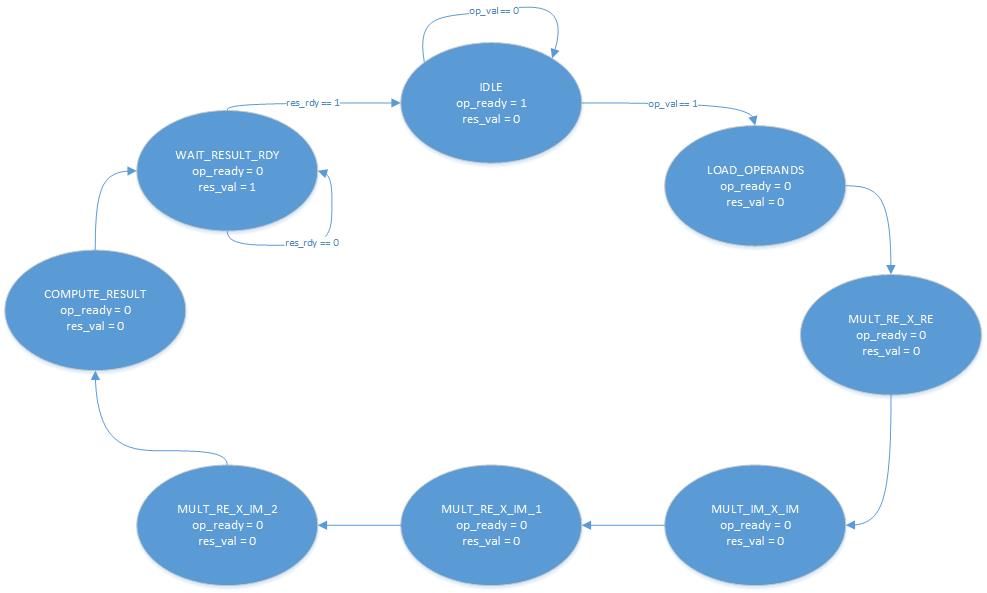


Figura 2: Graful de tranziție a stărilor

Tabel 2 : Explicarea stărilor modulului.

|  |  |
| --- | --- |
| Denumire | Explicație |
| IDLE | Modulul este în așteptare de noi operanzi, op\_ready este 1. |
| LOAD\_OPERANDS | Operanzii sunt încărcați în registrele interne. |
| MULT\_RE\_X\_RE | Se înmulțesc părțile reale ale fiecărui operand și se stochează în registrul corespunzător. |
| MULT\_IM\_X\_IM | Se înmulțesc părțile imaginare ale fiecărui operand și se stochează în registrul corespunzător. |
| MULT\_RE\_X\_IM\_1 | Se înmulțește partea reală a primului operand cu partea imaginară a celui de-al doilea operand și se stochează în registrul corespunzător. |
| MULT\_RE\_X\_IM\_2 | Se înmulțește partea imaginară a primului operand cu partea reală a celui de-al doilea operand și se stochează în registrul corespunzător. |
| COMPUTE\_RESULT | Se calculează adunarea și scăderea finale. |
| WAIT\_RESULT\_RDY | Se așteaptă semnalul de res\_ready, res\_val este activ. |

## Scenarii de test și forme de undă

Pentru testarea modulului au fost implementate mai multe scenarii de test. Acestea sunt prezentate mai jos, în codul pentru modulul complex\_nr\_mult\_tb. Acest modul, împreună cu monitor\_complex\_multiplier, vor fi folosite pentru simularea și verificarea funcționalității tuturor modulelor implementate în cadrul acestui proiect.

Fiecare scenariu de test are un task asociat. Din modulul test\_environment se poate alege ce scenariu va fi rulat. Proiectantul are posibilitatea de a selecta operanzii, de a trimite valori aleatoare, de a trimite valori extreme sau de a efectua mai multe calcule consecutiv.

Modulul monitor\_complex\_multiplier are rolul de a verifica automat dacă operațiile efectuate sunt corecte.

Formele de undă obținute în urma simulării primului scenariu de test cu valorile preluate din specificațiile proiectului sunt prezentate în Figura 3.

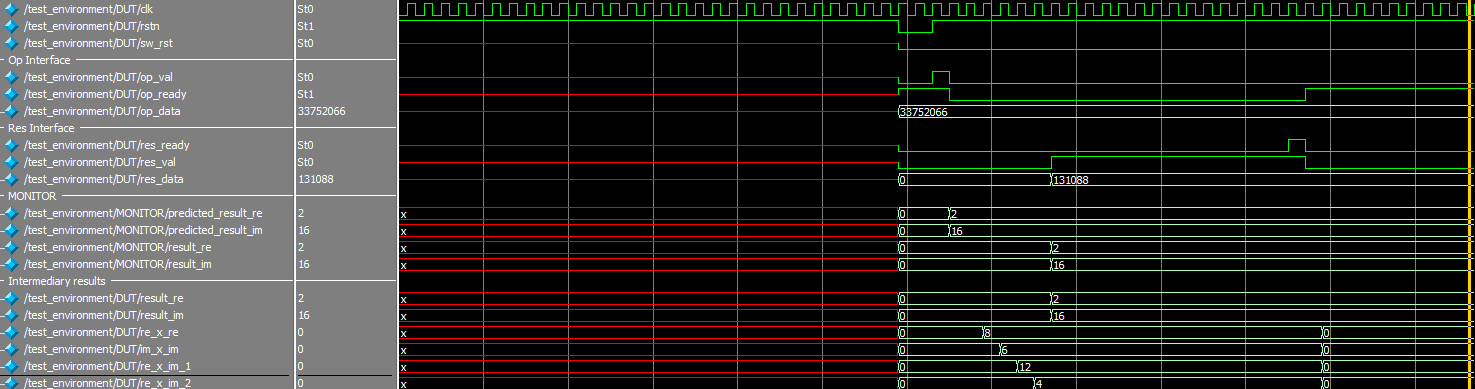


Figura 3 : Forme de undă obținute

# Implementare cu două module de multiplicare

## Arhitectură

În Figura 4 este prezentată arhitectura modulului. Tabelul 3 prezintă semnalele interne ale modulului și semnificația acestora. Interfața acestuia este aceeași cu implementarea cu un singur modul de multiplicare, interfață prezentată în Tabelul 2.



Figura 4 : Arhitectura modulului implementat

Tabel 3 : Prezentarea semnalelor interne.

|  |  |  |
| --- | --- | --- |
| Denumire | Tip | Explicatie |
| mult\_1\_op\_1\_sel | Intern | Semnal de selecție pentru primul operand al multiplicatorului 1. |
| mult\_1\_op\_2\_sel | Intern | Semnal de selecție pentru al doilea operand al multiplicatorului 1. |
| mult\_2\_op\_1\_sel | Intern | Semnal de selecție pentru primul operand al multiplicatorului 2. |
| mult\_2\_op\_2\_sel | Intern | Semnal de selecție pentru al doilea operand al multiplicatorului 2. |
| mult\_1\_result \_sel | Intern | Semnal de selecție pentru registrul de stocare al rezultatului de la ieșirea multiplicatorului 1. |
| mult\_2\_result \_sel | Intern | Semnal de selecție pentru registrul de stocare al rezultatului de la ieșirea multiplicatorului 2. |
| compute\_enable | Intern | Semnal de enable pentru realizarea adunării și scăderii finale. |

## Logica de control

În Figura 5 este prezentat graful de tranziții al modulului implementat. Tabelul 4 conține o scurtă explicație a fiecărei stări în parte.



Figura 5: Graful de tranziție a stărilor

Tabel 4 : Explicarea stărilor modulului.

|  |  |
| --- | --- |
| Denumire | Explicație |
| IDLE | Modulul este în așteptare de noi operanzi, op\_ready este 1. |
| LOAD\_OPERANDS | Operanzii sunt încărcați în registrele interne. |
| FIRST\_STAGE\_MULTIPLY | Se înmulțesc părțile reale ale fiecărui operand și părțile imaginare între ele. |
| SCND\_STAGE\_MULTIPLY | Se calculează valorile pentru adunare. |
| COMPUTE\_RESULT | Se calculează adunarea și scăderea finale. |
| WAIT\_RESULT\_RDY | Se așteaptă semnalul de res\_ready, res\_val este activ. |

## Forme de undă obținute



Figura 6: Forme de undă obținute

# Implementare cu patru module de multiplicare

## Arhitectură

În Figura 7 este prezentată arhitectura modulului. Tabelul 5 prezintă semnalele interne ale modulului și semnificația acestora. Interfața acestuia este aceeași cu implementarea cu un singur modul de multiplicare, interfață prezentată în Tabelul 2.



Figura 7 : Arhitectura modulului implementat

Tabel 5 : Prezentarea semnalelor interne.

|  |  |  |
| --- | --- | --- |
| Denumire | Tip | Explicatie |
| compute\_enable | Intern | Semnal de enable pentru realizarea adunării și scăderii finale. |

## Logica de control

În Figura 8 este prezentat graful de tranziții al modulului implementat. Tabelul 5 conține o scurtă explicație a fiecărei stări în parte.



Figura 8: Graful de tranziție a stărilor

Tabel 6 : Explicarea stărilor modulului.

|  |  |
| --- | --- |
| Denumire | Explicație |
| IDLE | Modulul este în așteptare de noi operanzi, op\_ready este 1. |
| COMPUTE\_RESULT | Se calculează adunarea și scăderea. |
| WAIT\_RESULT\_RDY | Se așteaptă semnalul de res\_ready, res\_val este activ. |

## Forme de undă obținute

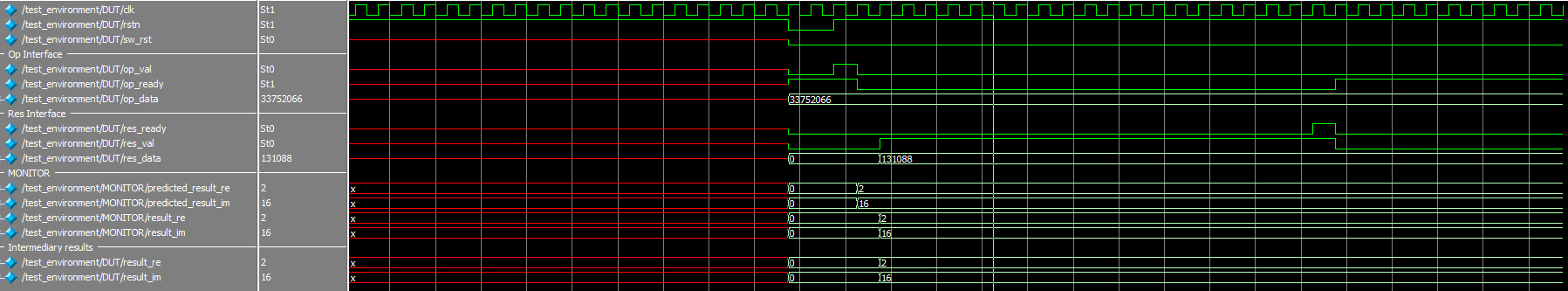


Figura 8: Forme de undă obținute