****

**Dispozitiv de calculare a mediei unui set de numere**

* **Proiect PSN -**

**CUPRINS**

**Îndrumător:**

**Noema Maier**

**Proiect realizat de:**

**Saca Victor – Valentin**

**Țâbulac Daria - Mara**

**1. Specificații……………………………………………...3**

**2. Proiectare……………………………………………….3**

**2.1. Schema block – black-box..…………….3**

**2.2. Resurse………………………………………….4**

**2.3. Organigrama………………………………..13**

**2.4. Schema detaliată a circuitului……….15**

**3. Plăcuța utilizată……………………………………..16**

**4. Instrucțiuni de utilizare………………………….17**

**5. Justificarea soluției alese……………………….19**

**6. Posibilități de dezvoltare ulterioare……….20**

**7. Bibliografie…………………………………………….21**

**DISPOZITIV DE CALCULARE A MEDIEI**

**UNUI SET DE NUMERE**

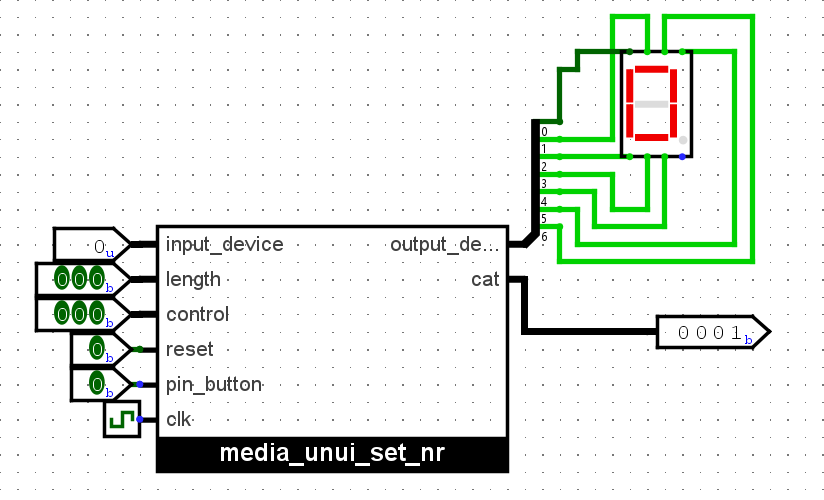
1. **SPECIFICAȚII**

Acest proiect prezintă arhitectura internă a unui dispozitiv de calculare a mediei unui set de numere care funcționează astfel: se selectează modalitatea de introducere a numerelor a căror medie dorim să o calculăm, apoi in funcție de modalitatea aleasă, se află media numerelor și se afișează pe SSD-uri. După ce aceasta este afișată, se apasă butonul de restart care permite calcularea unui alt set de numere.

Modalitățile de introducere a numerelor sunt: citirea lor din 2 memorii ROM, la alegere, generarea aleatoare a unor numere din intervalul 0-15 sau din intervalul 0-255, sau introducerea lor în binar, de la switch-uri.

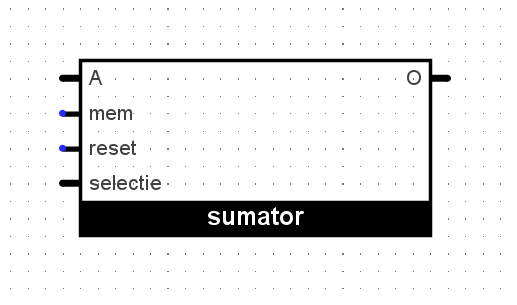
1. **Proiectare**

**2.1 SCHEMA BLOC – BLACK-BOX**

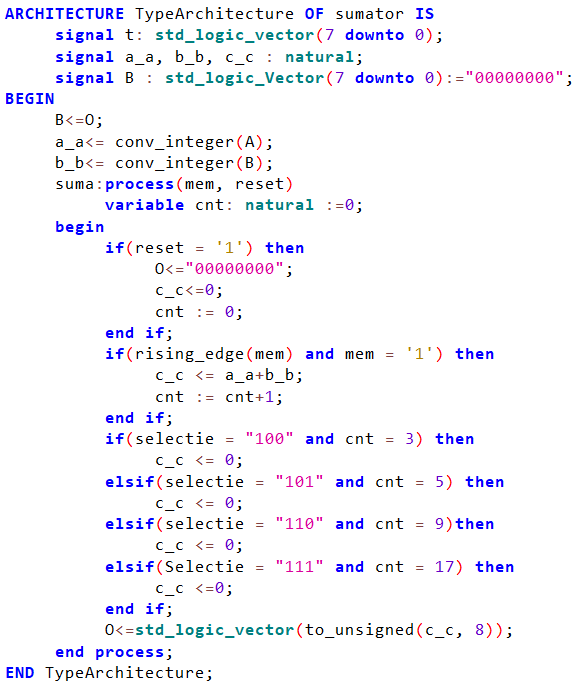


* 1. **RESURSE**

1. Sumator

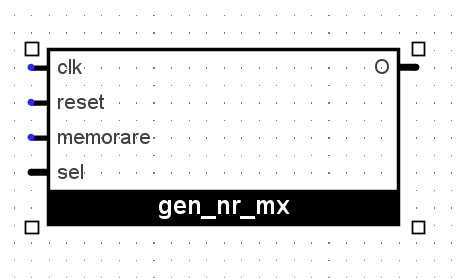


Arhitectura:

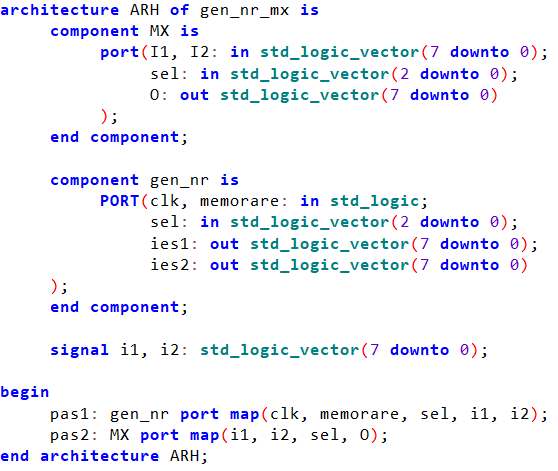


În funcție de selecție, sumatorul va face media a 2 (“100”), 4 (“101”), 8 (“110”) sau 16 (“111”) numere. Totodata, in numărător se țin minte câte nr au fost adăugate la sumă, iar atunci când numărul este depășit, sumatorul se “resetează”, ieșirea primind valoarea 0.

1. Generator de numere pe 4 sau 8 biți

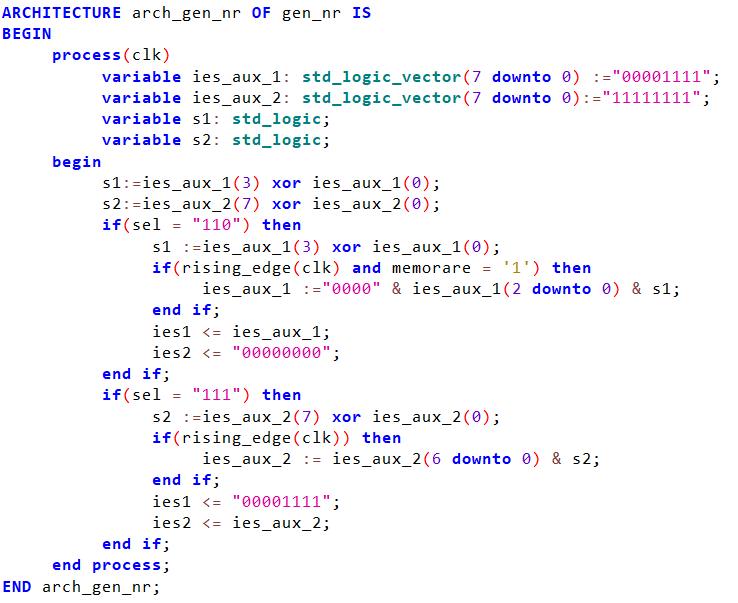


Arhitectura:

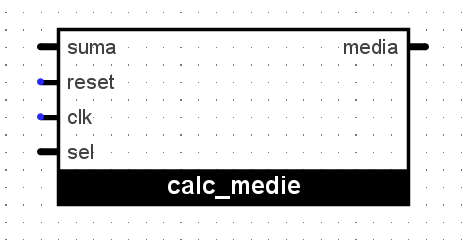


În această componentă se folosesc 2 genratoare de numere pseudoaleatoare și un MUX. Unul dintre generatoare generează numere în bucla 0 – 15, iar celălalt generează numere în bucla 0 – 255. Generarea lor se realizeaza prin efectuarea operației xor între primul și ultimul bit al numărului introdus. MUX-ul se folosește pentru a alege în ce buclă generăm numerele.

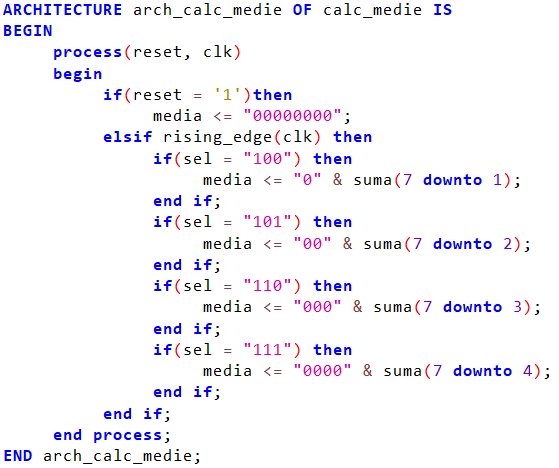
Arhitectura generatorului care evidențiază regula de obținere a numerelor:



1. Registru de shiftare pentru medie



Arhitectura:

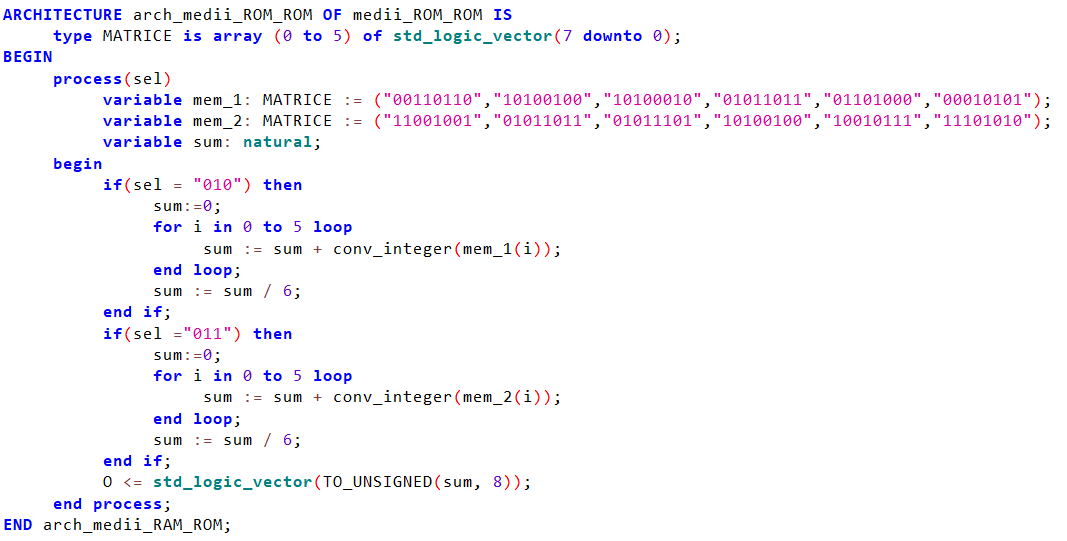


Selecția sel ne spune media a câtor numere trebuie calculată. Deci în loc să se efectueze împărțire prin scădere repetată, suma introdusă se va shifta cu 1, 2, 3 sau 4 biți pentru a obține media numerelor.

1. Mediile ROM

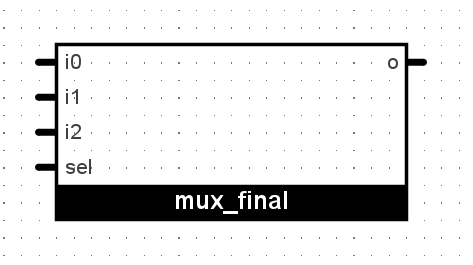


Arhitectura:

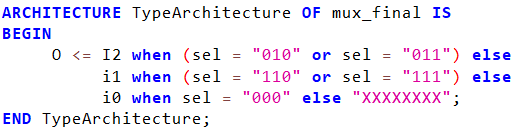


Cu ajutorul selecției alegem din ce memorie extragem numerele a căror medie dorim să o calculăm, iar pe ieșire se afișază media acelor numere. Selecția “010” calculează media numerelor din prima memorie, iar selecția “011” calculează media numerelor din a doua memorie. Prima memorie conține numerele 54, 164, 162, 91, 104 și 21, iar media lor va fi 99. În a doua medie se găsesc numerele 201, 91, 93, 164, 151 și 234, iar media lor este 155.

1. MUX final

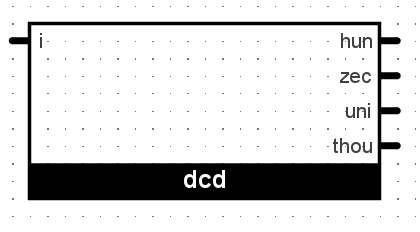


Arhitectura:

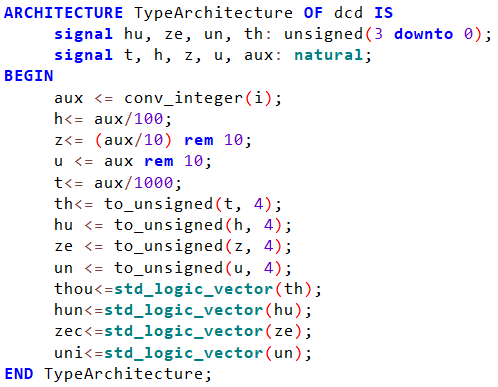


Este, de fapt, funcția control settings, care alege modul de calculare a mediei, dintre cele 2 memorii, cele 2 generatoare sau introducerea numărului de la switch-uri. “000” înseamnă introducerea numerelor de către utilizator prin switch-uri, “010” sau “011” înseamnă media numerelor generate aleator si “110” sau “111” înseamnă media numerelor din memorii.

1. DCD

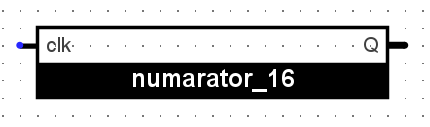


Arhitectura DCD-ului:

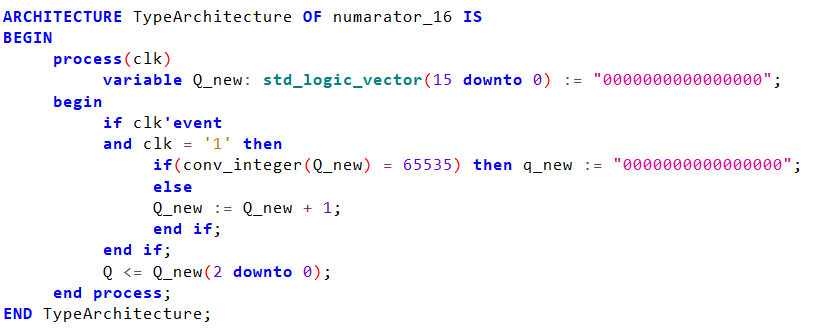


Decodificatorul DCD transformă numerele din binar în zecimal, pe ieșire afișând numărul împărțit în unități, zeci, sute și mii.

1. Numărător

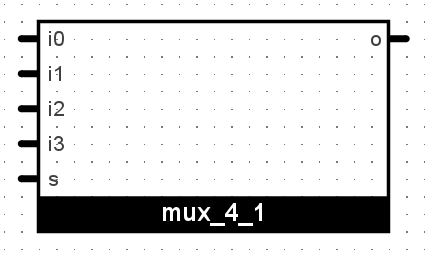


Arhitectura:

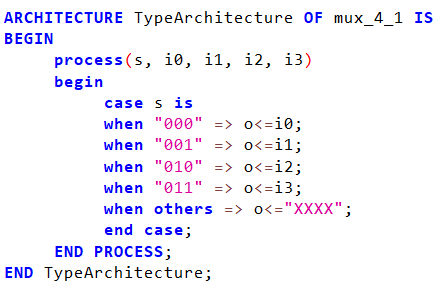


Acest numărător pe 16 biți afișază pe ieșire biții 15 și 14, care vor fi folosiți ca selecție pentru cele doua MUX-uri de 4:1.

1. MUX 4:1



Arhitectura:

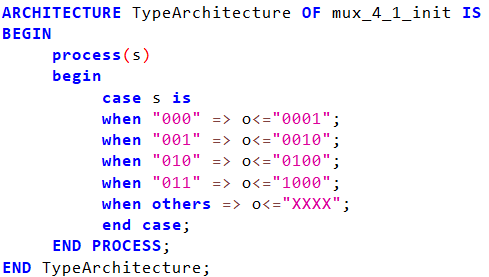


Acest MUX, în funcție de selecția s, afișază, pe rând, unitățile, zecile, sutele și miile mediei calculate de dispozitiv.

1. MUX 4:1

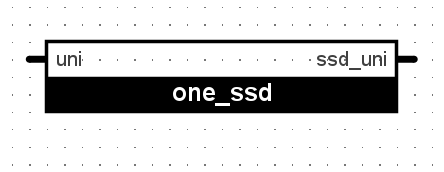


Arhitectura:

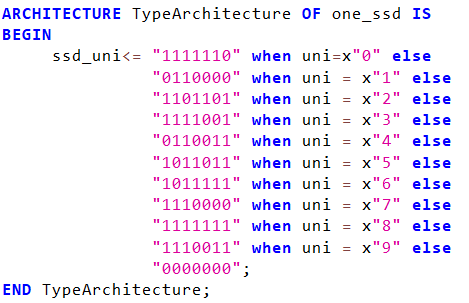


Acest MUX alege pe care catod de pe placă se va afișa numărul ales anterior.

1. SSD



Arhitectura:



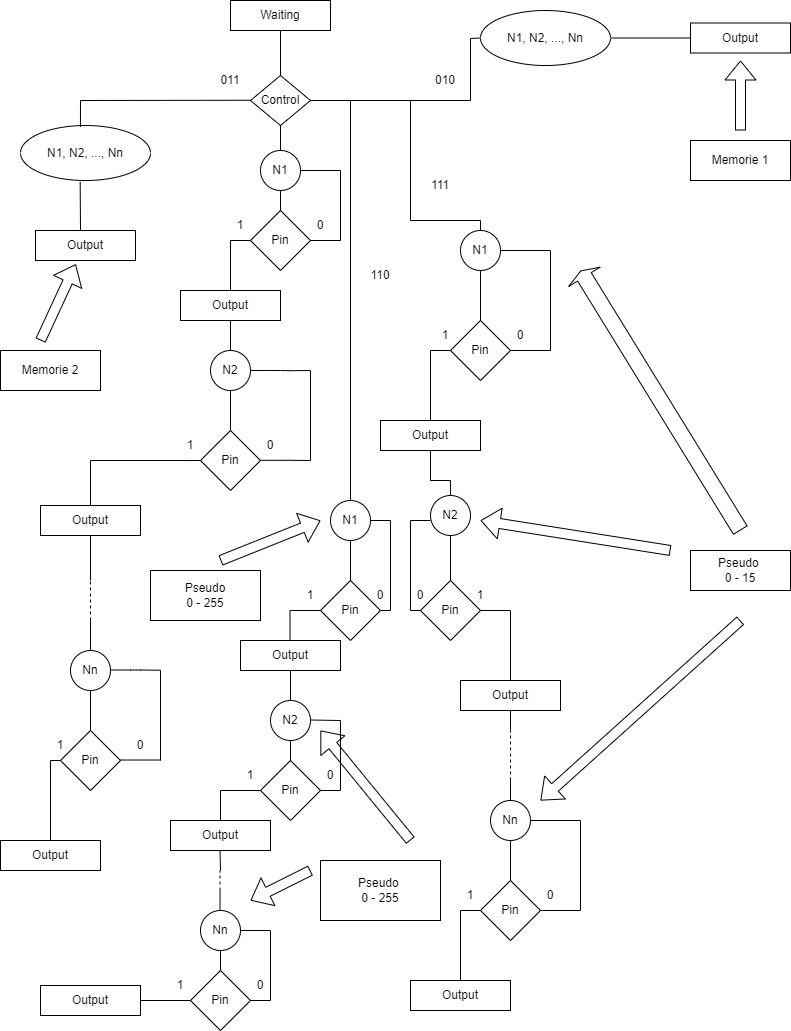
În această parte se afișează pe SSD-ul ales, cifra corespunzătoare acelui moment din calcularea mediei.

**2.3 ORGANIGRAMA**

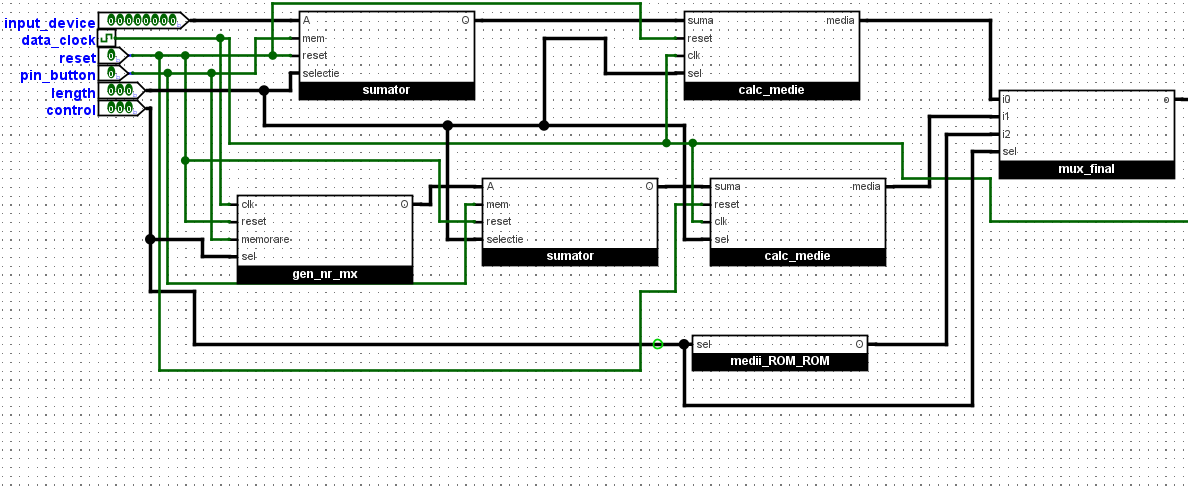
Diagrama de stări nu este un flow-chart, ci reprezintă partea de control, partea decizională din orice algoritm, și ea poate fi apoi implementată direct în VHDL dacă e facută corect.

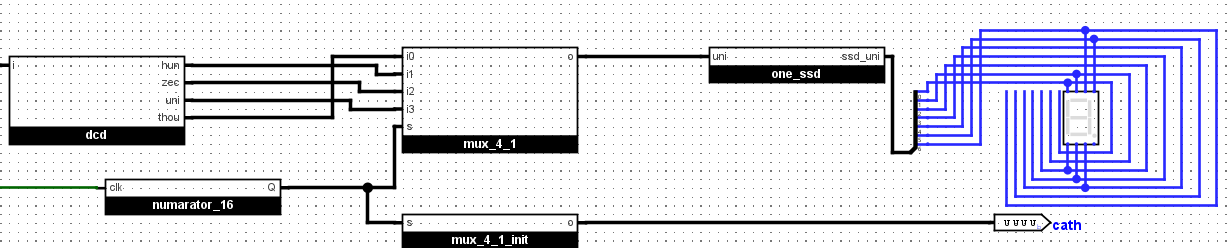
* **Stările** sunt reprezentate prin . O stare reprezinta un moment de timp (o perioada).
* **Deciziile** luate în fiecare stare sunt reprezentate prin romb. `
* **Ieșirile** generate în fiecare stare sunt reprezentate prin . În interiorul dreptunghiului se enumera ieșirile care sunt adevărate în acel moment.



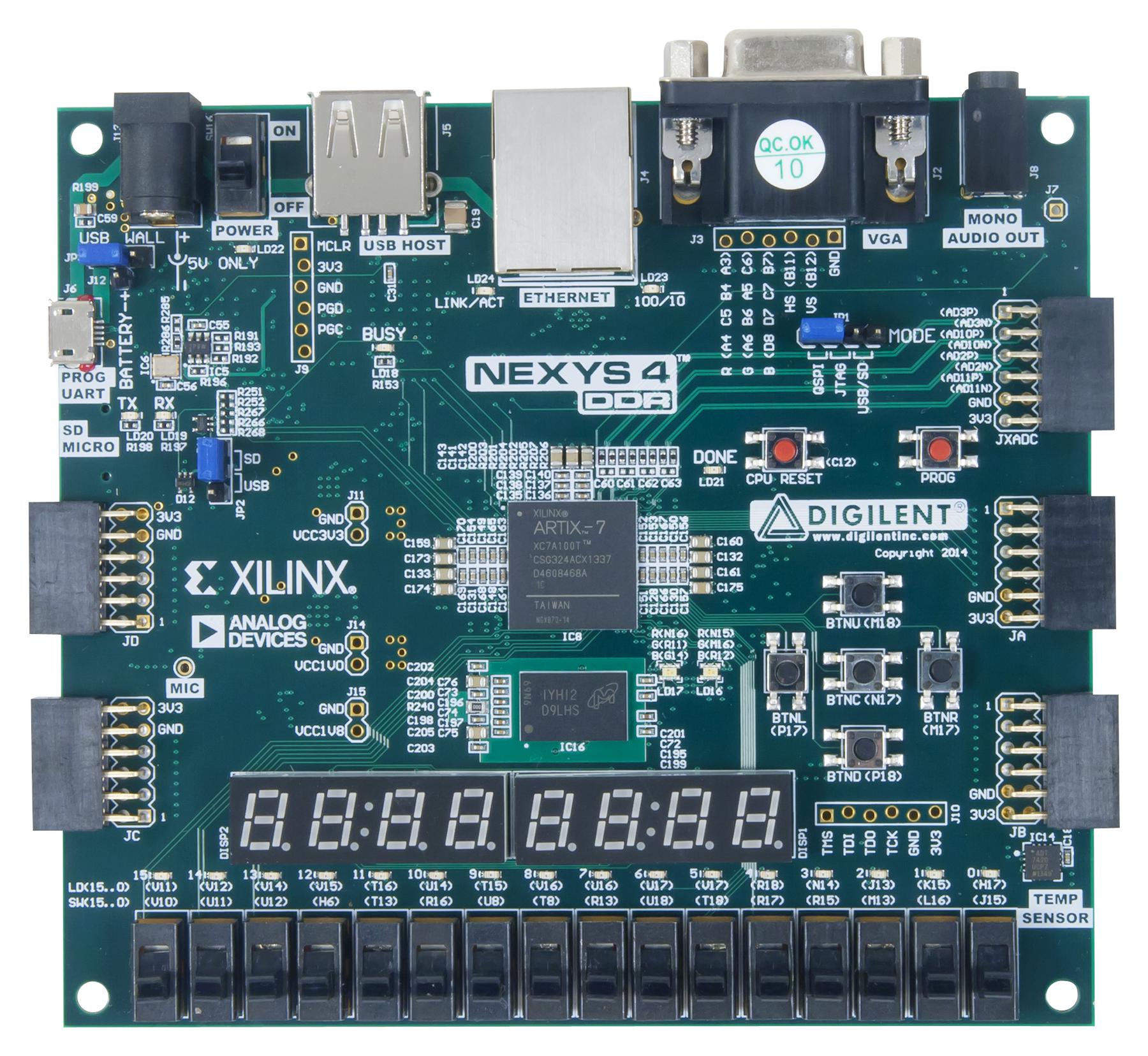


**2.4 SCHEMA DETALIATĂ A CIRCUITULUI**

****

****

1. **PLĂCUȚA UTILIZATĂ (Nexys 4 DDR)**



**6**

**5**

**4**

**2**

**3**

**7**

**1**

1 – 4 SSD-uri pentru afișarea mediei

2 – 8 switch-uri de la care se introduc numerele

3 – 3 switch-uri pentru Control Settings

4 – 3 switch-uri pentru Length

5 – switch de memorare ( pin\_button )

6 – switch de reset ( Reset )

7 – clock-ul plăcii

1. **INSTRUCȚIUNI DE UTILIZARE**

Deși proiectul funcționează și fără această operație, înainte de alegerea modului de calculare a mediei, este de preferat să se apese butonul de reset. După efectuarea acestei acțiuni, se aleg Control Settings și Length dorite.

Length este o selecție pe 3 biți care, în funcție de valoarea dată de utilizator, stabilește dacă se calculează media a 2 numere (selecția “100”), a 4 numere (selecția “101”), a 8 numere (selecția “110”) sau a 16 numere (selecția “111”).

Control Settings are 5 posibilități de alegere, fiind, de asemenea, o selecție pe 3 biți.

Selecția “000” permite utilizatorului să introducă de la switch-uri numerele a căror medie dorește să o calculeze. După introducerea fiecăruia trebuie apăsat butonul pin\_button pentru a se reține numărul introdus. Pe SSD-uri se vor afișa și mediile intermediare. În funcție de valoarea aleasă pentru Length, adică de câte numere trebuie introduce, dacă val maxima (2, 4, 8 sau 16) este depășită, atunci media va deveni 0 în mod automat, dispozitivul “resetându-se” singur.

Pentru selecția “010” se va afișa pe SSD numărul 99, adică media numerelor din prima memorie ROM, valoare acestora fiind 54, 164, 162, 91, 104 și 21. Media lor va fi calculate odată cu citirea lor din memorie.

Pentru selecția “011” pe SSD se va afișa numărul 155, care reprezintă media numerelor din cea de-a doua memorie ROM, ce conține valorile 201, 91, 93, 164, 151 și 234. Această medie va fi de asemenea calculată odată cu citirea valorilor din memorie.

Dacă selecția are valoarea “110” se vor genera numere aleator și acestea vor avea valori în bucla 0 – 15. După fiecare număr generat trebuie apăsat butonul pin\_button pentru ca numărul să fie reținut.

Valoarea “111” a selecției înseamnă calcularea mediei unor numere generate aleator în bucla 0 – 255. La fel, după generarea fiecărei valori trebuie apăsat butonul pin\_button pentru ca numărul să fie memorat.

Pentru ambele generatoare de numere, fie în bucla 0 – 15 , fie în bucla 0 – 255, se va ține cont de valoarea selectată pe Length, adică 2, 4, 8 sau 16, și se va calcula media introduce până la valoarea maximă. Dacă aceasta va fi depășită, atunci se va afișa valoarea 0, la fel ca și în cazul introducerii numerelor de la switch-uri.

1. **JUSTIFICAREA SOLUȚIEI ALESE**

Am ales acest mod de implementare, deoarece ni s-a părut ușor de înțeles și de pus în aplicare. Am ales să nu salvăm numerele într-o memorie pentru a nu folosi atât de multe resurse. Din același motiv am ales ca în memorii să calculăm suma numerelor salvate odată cu citirea lor. Împărțirea sumei obținute în oricare din cazurile alese se efectuează prin shiftare, și nu prin scăderi repetate sau împărțire directă(ALU), deoarece este un mod mai rapid de a efectua operația, având în vedere că putem avea doar 2, 4, 8 sau 16. În concluzie, prin aceste feature-uri am vrut să minimizăm resursele utilizate.

1. **POSIBILITĂȚI DE DEZVOLTARE ULTERIOARĂ**

Dispozitivul poate fi dezvoltat astfel încât pe afișoare să apară doar media finală, fără mediile intermediare. În ceea ce privește precizia calculului, în viitor, media ar putea fi calculată cu o precizie mai mare, astfel încât pe afișoare să se poată vedea rezultatul exact, în virgulă mobilă. De asemenea, s-ar putea mări capacitatea memoriilor și range-ul în care se poate calcula media numerelor introduse.

1. **BIBLIOGRAFIE**

* Lucrările de laborator
* <https://ro.farnell.com/digilent/410-292/dev-board-artix-7-fpga-nexys-ddr/dp/2490174>
* <https://app.diagrams.net/>
* Documentația proiectului de pe teams
* [1.Divizor Frecventa.mp4](https://didatec.sharepoint.com/:v:/r/sites/PSN2023-SeriaB/Class%20Materials/Proiect/1.Divizor%20Frecventa.mp4?csf=1&web=1&e=dRpU6g)
* [2.Controler afisaje 7 segmente.mp4](https://didatec.sharepoint.com/:v:/r/sites/PSN2023-SeriaB/Class%20Materials/Proiect/2.Controler%20afisaje%207%20segmente.mp4?csf=1&web=1&e=KsDojz)
* [Exemplu proiect PSN.docx](https://didatec.sharepoint.com/:w:/r/sites/PSN2023-SeriaB/Shared%20Documents/General/Exemplu%20proiect%20PSN.docx?d=w025a47eaca374f30b24ee49a828819ba&csf=1&web=1&e=WStPsH)