



Universitatea Tehnica din Cluj-Napoca
Facultatea de Automatica și Calculatoare
Specializarea Calculatoare

DISPOZITIV DE CALCUL AL MEDIEI UNUI SET DE NUMERE

Mozsi Denis-Alex

Suciu Sonia-Maria

Grupa 30212

Cuprins

I. Tema proiectului

.....

II. Schema bloc

.....

III. Lista componentelor

.....

IV. Semnificația pinilor

.....

V. Implementare

.....

1. Generator de numere (aleatorii sau introduse manual)

.....

2. Filtru / Validare date de intrare

.....

3.Display

.....

VI. Intrări. Ieșiri. Semnale interne

.....

VII. Justificarea soluției alese

.....

VIII. Manual de utilizare

.....

1. Deschiderea proiectului

.....

2. Xilinx ISE și placa FPGA Basys 3

.....

IX. Posibilități de dezvoltare și îmbunătățire

.....

I. Tema proiectului

Sarcina proiectului este de a dezvolta un sistem simplu de procesare a semnalului care să calculeze **media mobilă** (rolling average) a unui flux de date paralel pe 8 biți, ca exercițiu practic de proiectare digitală.

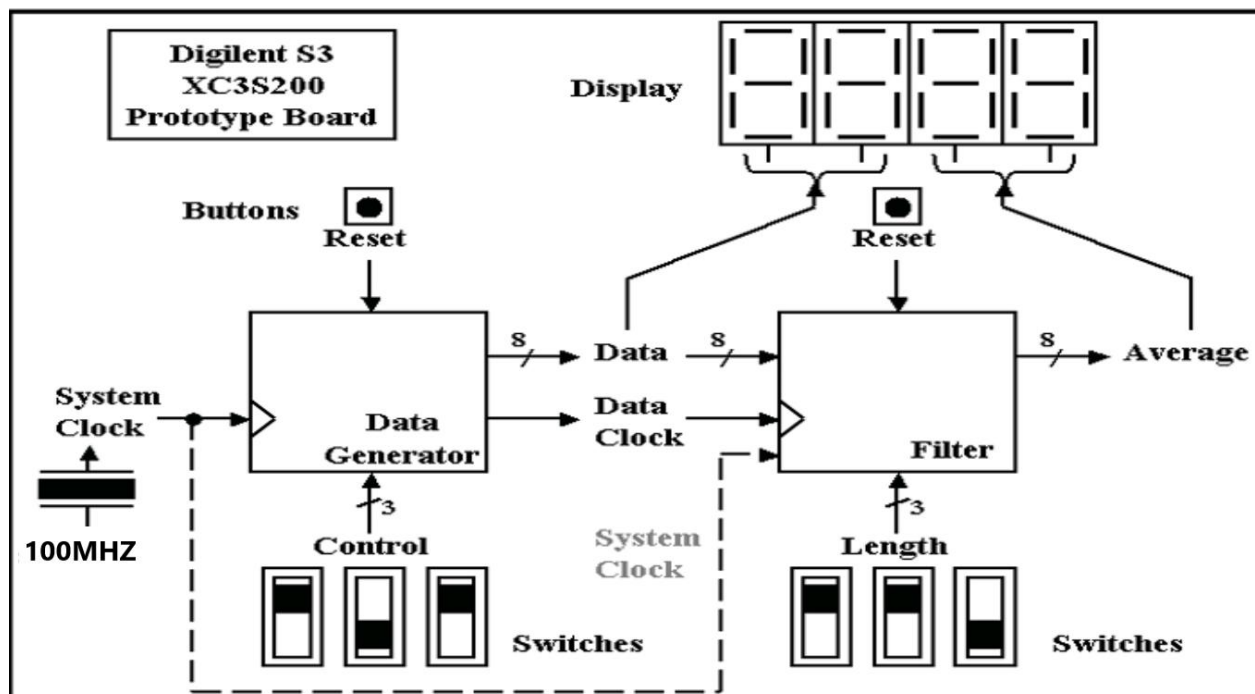
Proiectul va fi implementat pe placa de dezvoltare **Basys 3** cu FPGA **Xilinx Artix-7**, care permite testarea și demonstrarea funcționării corecte a sistemului în timp real.

Sistemul va fi dezvoltat utilizând limbajul **VHDL**, în mediul de proiectare **Xilinx ISE WebPack versiunea 6.3**, iar pentru verificarea funcționalității logice se vor utiliza unelte de simulare **ModelSim**.

Obiectivul este realizarea unui model VHDL al unui **filtru digital** care calculează **media** unui set de date primite în paralel. Acest sistem va fi completat de un **generator de date**, intern sau controlat prin intrări externe. Sistemul trebuie să funcționeze în **timp real**, furnizând valoarea medie sincronizat cu rata de intrare a datelor.

Pentru demonstrarea funcționării corecte, vor fi utilizate **butoanele**, **switch-urile** și **afișajele cu 7 segmente** de pe placa **Basys 3**.

II. Schema bloc



III. Lista componentelor

Componentele care formează sistemul sunt:

- **Generator de date**
- **Filtru**
- **Afișoare**

IV. Semnificația pinilor

Interfața cu utilizatorul se realizează cu ajutorul **switch-urilor (comutatoarelor)**. Prin intermediul **comutatoarelor de control**, utilizatorul poate selecta, prin combinații diferite, una dintre următoarele opțiuni:

- **Test**
- **Square wave**
- **Secvență introdusă de primul student;**
- **Secvență introdusă de al doilea student;**
- **Numere pseudo-aleatorii în intervalul 0–15;**
- **Numere pseudo-aleatorii în intervalul 0–255.**

Folosind **comutatoarele de control pentru lungime (length)**, utilizatorul poate selecta, tot prin combinații diferite, una dintre următoarele opțiuni:

- **Stop – Păstrează valoarea curentă;**
- **Media a 2 valori;**
- **Media a 4 valori;**
- **Media a 8 valori;**
- **Media a 16 valori.**

V. Implementare

Componentele utilizate pentru implementare sunt următoarele:

- Generator de date
- Filtru
- Afișare

1. Generator de date

Test: Modul pentru generarea constantei zero

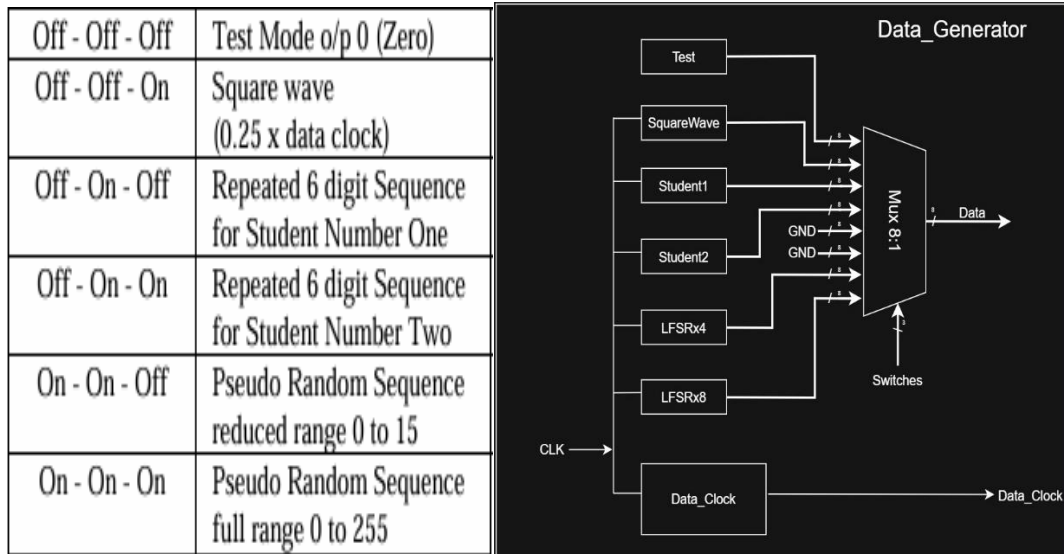
SquareWave: Modulul generează un semnal pătrat, alternând periodic între valorile 0 și 255. Frecvența semnalului generat este fixată la un sfert (0.25 HZ) din frecvența ceasului de 1HZ.

Student1 și **Student2:** Modulele emit secvențe predefinite unice, compuse fiecare din 6 valori specifice. Fiecare secvență este distinctă și asociată unui anumit student, utilizate în identificarea sau testarea personalizată a funcționării sistemului.

LFSRx4 și **LFSRx8:** Module linear feedback shift register (LFSR) sunt generatoare de numere pseudo-aleatorii bazate pe registre de deplasare cu reacție liniară (LFSR). Varianta LFSRx4 utilizează un registru pe 4 biți, generând o secvență de până la 15 valori distincte, iar LFSRx8 utilizează un registru pe 8 biți, generând o secvență extinsă de până la 255 valori distincte.

Data_Clock: Modulul generează semnalul folosit pentru sincronizarea datelor la ieșire. Semnalul este obținut prin divizarea frecvenței semnalului de ceas principal de 100MHZ furnizat de FPGA. Astfel, se asigură o frecvență redusă de 1HZ și stabilă, necesară pentru sincronizarea corectă și fiabilă a datelor transmise.

Mux8: Multiplexor pe 8 intrări pentru selectarea sursei de date conform switch-urilor

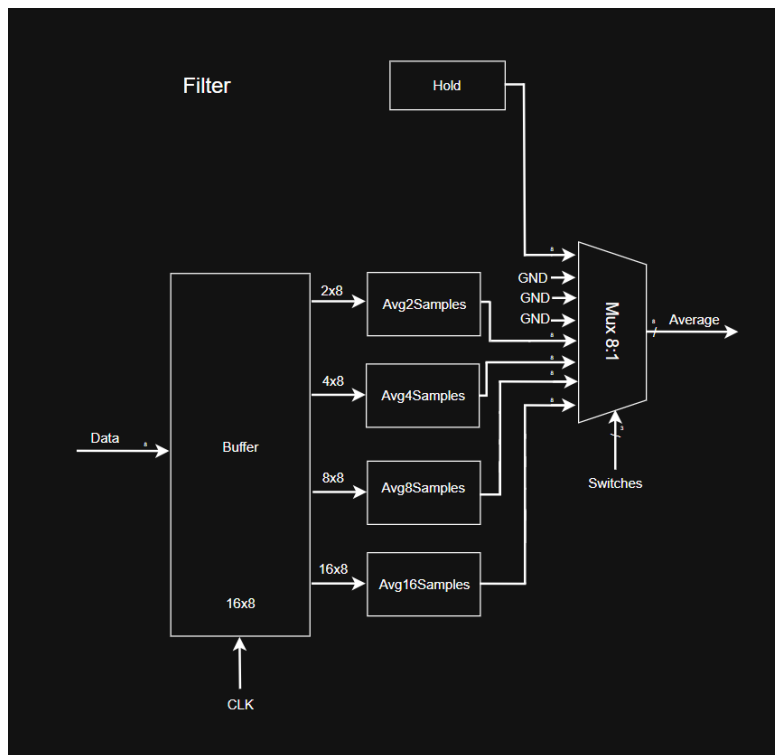


2. Filtru

Modulul Filter integrează următoarele componente:

- **Buff:** Modulul stochează temporar datele primite de la Data Generator. Când memoria internă a acestuia se umple complet, emite semnalul full(full2,full4,full8,full16), indicând că datele sunt gata pentru prelucrare. Ieșirea furnizează datele stocate către modulele care calculează media.
- **Avg2Samples, Avg4Samples, Avg8Samples, Avg16Samples:** Module realizează calculul mediei unui set fix de valori din buffer. La recepția semnalului full, fiecare modul calculează media ultimelor 2, 4, 8 sau 16 valori, primite de la ieșirea bufferului , în funcție de modul ales pentru operare.
- **Hold:** Modul reține și menține la ieșire rezultatul ultimei medii calculate, fiind util pentru verificări și depanare.
- **Mux8:** Un multiplexor care selectează rezultatul corect (media calculată) pe baza valorii din Switches și îl trimite pe ieșirea average.

Off - Off - Off	Stop - Hold Value
On - Off - Off	2 Sample Average
On - Off - On	4 Sample Average
On - On - Off	8 Sample Average
On - On - On	16 Sample Average



3. Afișare

Afișajul este alcătuit din patru cifre pe 7 segmente și are rolul de a prezenta vizual atât valoarea curentă, cât și media rezultată în urma filtrării. Cele patru cifre sunt împărțite în două grupe:

- primele două cifre afișează valoarea curentă generată;
- ultimele două cifre afișează media calculată de filtru.

Afișajul funcționează prin comutarea rapidă între cele patru cifre, astfel încât toate par aprinse simultan pentru ochiul uman. Comutarea se face periodic, cu o frecvență redusă față de ceasul principal, pentru a asigura o afișare stabilă. Fiecare cifră afișează un caracter în format hexazecimal (0–F), corespunzător valorii pe 4 biți primită.

VI. Intrări, ieșiri, Semnale Interne

Pentru generatorul de date am folosit : clk: semnalul de ceas principal al sistemului, reset: reseteaza asincron datele introduse până la acel moment , switches: vector pe 3 biți care determină modul de operare, data : datele de ieșire care urmează să fie introduse în filtru , data_clk: modul care generează semnalul data_clk pentru sincronizarea ieșirilor.

Switches- un vector pe 3 biți care stabilește modul de operare, adică câte valori sunt folosite pentru calculul mediei (2, 4, 8 sau 16). Ieșirea principală a sistemului este average, care



Afișorul are rolul de a afișa, pe un display cu 4 cifre cu anod comun, două valori de 8 biți: numărul generat și media calculată în urma procesului de filtrare. Cele două valori sunt concatenate într-un vector de 16 biți și afișate în format hexazecimal: primele două cifre ale display-ului arată numărul generat, iar ultimele două cifre arată media.

Semnalul clk determină ritmul de actualizare a cifrelor, iar reset reinitializează contorul intern. Afișajul funcționează pe principiul multiplexării: este activată pe rând câte o singură cifră, la o frecvență suficient de mare încât toate să pară aprinse simultan. Semnalul **ANOD** selectează cifra activă, iar **CATOD** controlează segmentele aprinse pentru a forma caracterul corespunzător



VII. Justificarea Soluției Alese

Structura modulară a proiectului a fost aleasă pentru a permite o dezvoltare clară, organizată și ușor de testat. Fiecare funcționalitate – generarea datelor, calculul mediei, afișarea și controlul – a fost implementată ca modul separat, ceea ce a dus la o arhitectură scalabilă și ușor de întreținut.

Pentru generarea datelor, am optat pentru registre LFSR (Linear Feedback Shift Register), deoarece oferă o metodă eficientă din punct de vedere hardware pentru simularea unor valori variabile, fără a necesita memorie suplimentară sau logică complexă.

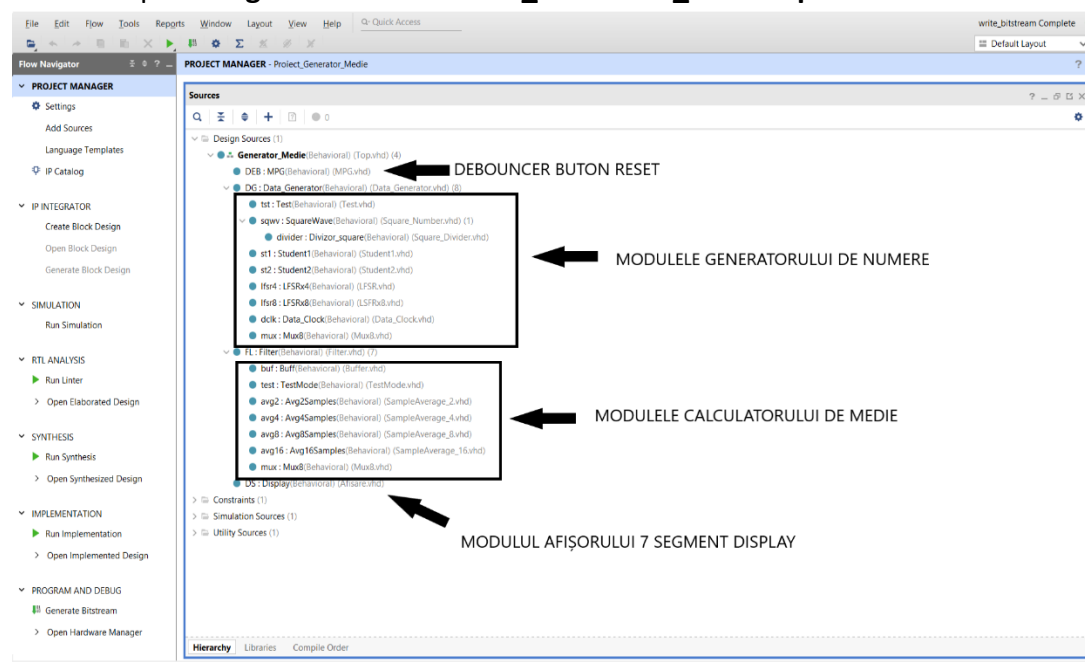
Calculul mediei a fost împărțit în patru module distincte (pentru 2, 4, 8 și 16 mostre), ceea ce a simplificat procesul de selecție și a permis o comutare rapidă între moduri cu ajutorul comutatoarelor (Switches).

Pentru afișare, s-a folosit display-ul cu 4 cifre cu 7 segmente, deja integrat pe placa Basys 3, valorificând resursele existente ale plăcii. Multiplexarea afișajului a permis utilizarea eficientă a pinilor FPGA și o prezentare clară a informațiilor (datele generate + media calculată).

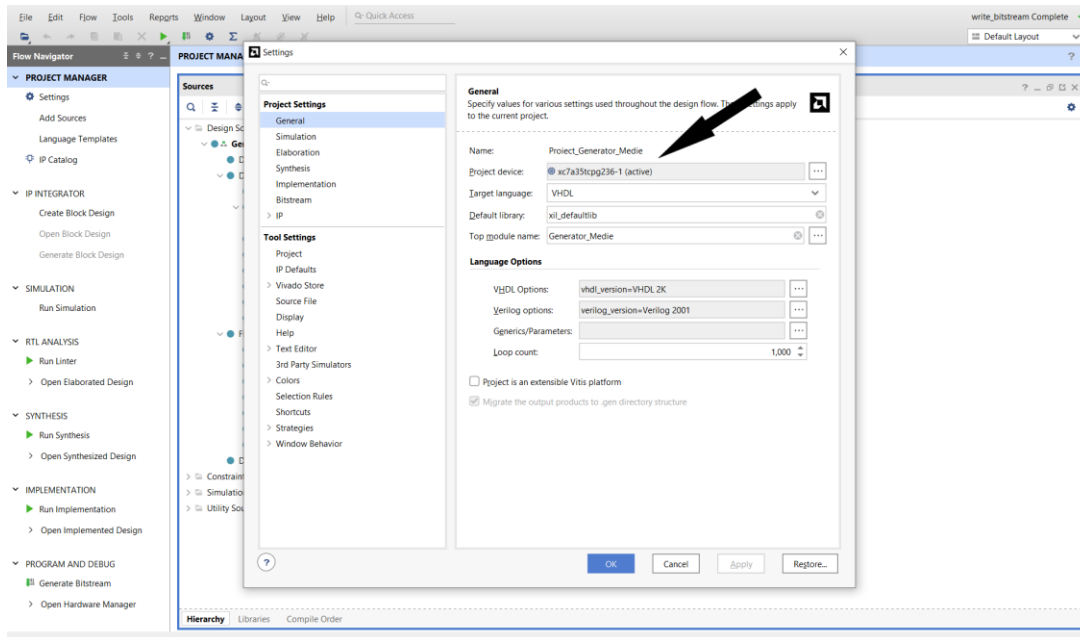
VIII. Manualul Utilizatorului

1. Deschiderea proiectului

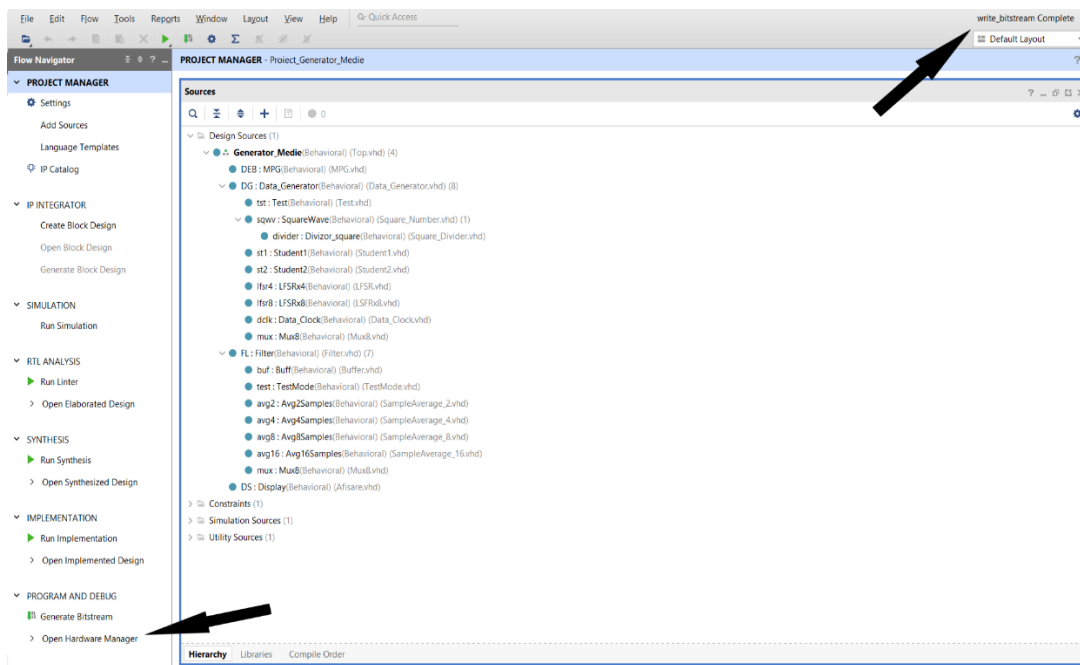
Intrăm în fișierul în care se află Generatorul de Medie, deschidem fișierul cu VIVADO după care dăm click pe: **Design Sources -> Proiect_Generator_Medie.xpr**

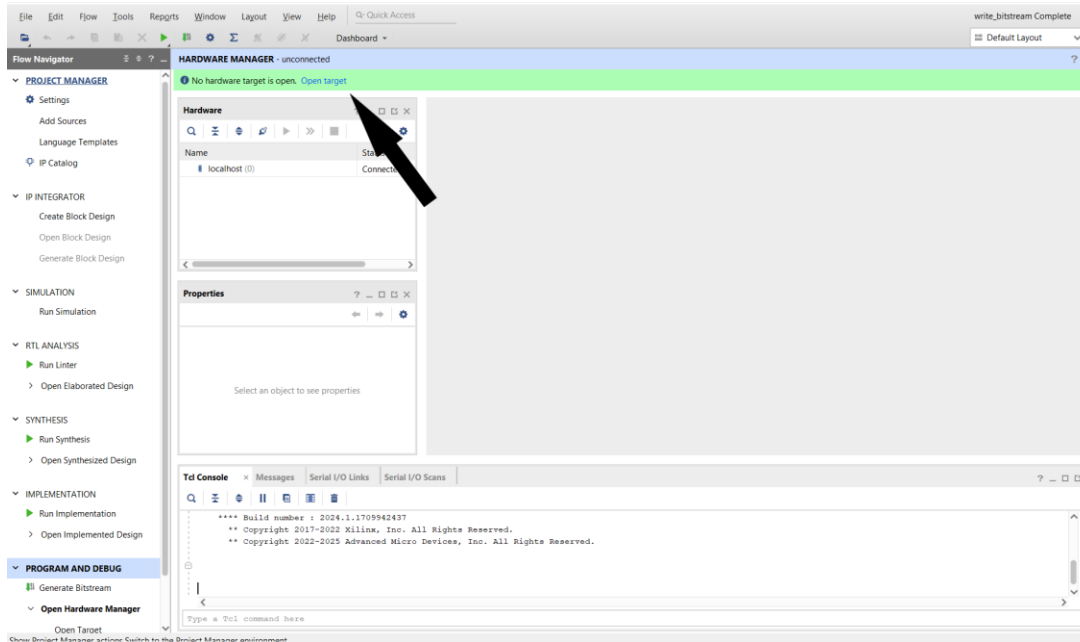


2. Selectarea plăcuței Basys 3

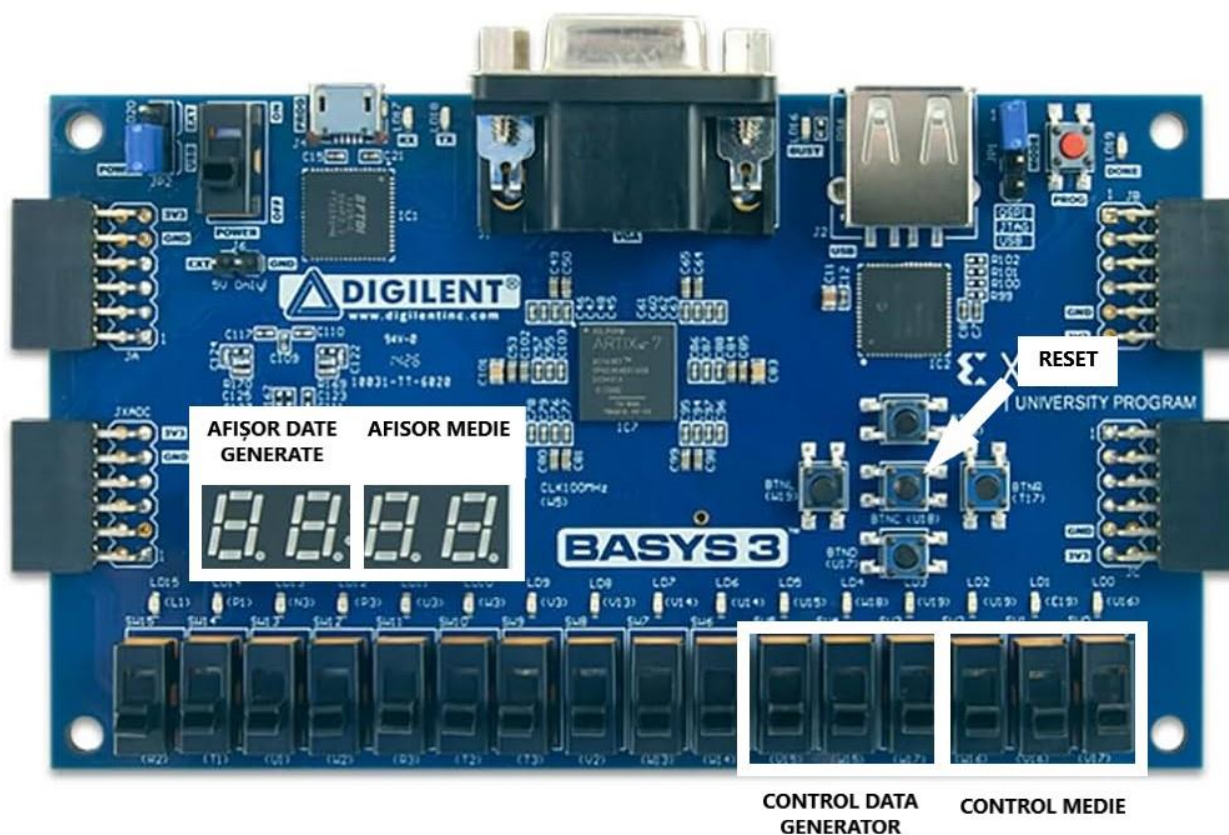


3. Generare Bitstream și încărcare pe placa FPGA





4. Arhitectura fizică a proiectului pe placa FPGA



IX. Posibilități de dezvoltare și îmbunătățire

Proiectul prezentat poate fi îmbunătățit și extins în mai multe direcții:

- **Afișare în format zecimal** – În prezent, valorile sunt afișate în format hexazecimal. Se poate implementa un modul de conversie binar–zecimal (BCD) pentru o afișare mai intuitivă.
- **Selecție dinamică a dimensiunii mediei** – În loc de un selector fix (Switches), s-ar putea permite alegerea numărului de mostre pentru medie direct din software sau printr-un protocol de comunicare.
- **Adăugarea unui mod de vizualizare grafică** – Se poate conecta un modul VGA sau o interfață serială pentru a vizualiza grafic evoluția valorilor mediate.
- **Creșterea complexității generatorului de date** – În locul LFSR-ului simplu, se poate integra un generator pseudo-aleator mai avansat, cu seed variabil sau chiar cu intrare externă.
- **Adăugarea unei memorii de stocare** – Pentru a păstra un istoric al valorilor generate și mediate, se poate integra o memorie RAM internă sau externă.
- **Optimizarea consumului de resurse** – Codul VHDL poate fi analizat și refactorizat pentru a reduce numărul de registre sau logica combinatorie utilizată.