Dispozitiv de calcul al mediei unui set de numere

Studenţi:
Zelenszky Bianca
Pop Ruxandra Maria

Coordonator Proiect:
Prof.Pop Diana

Universitatea tehnică din Cluj-Napoca
Facultatea de Automatică şi Calculatoare
Secţia Calculatoare şi Tehnologia Informaţiei
Anul 1, Grupa 30216

Capitole

1.Specificaţie proiect

Cerinţă:

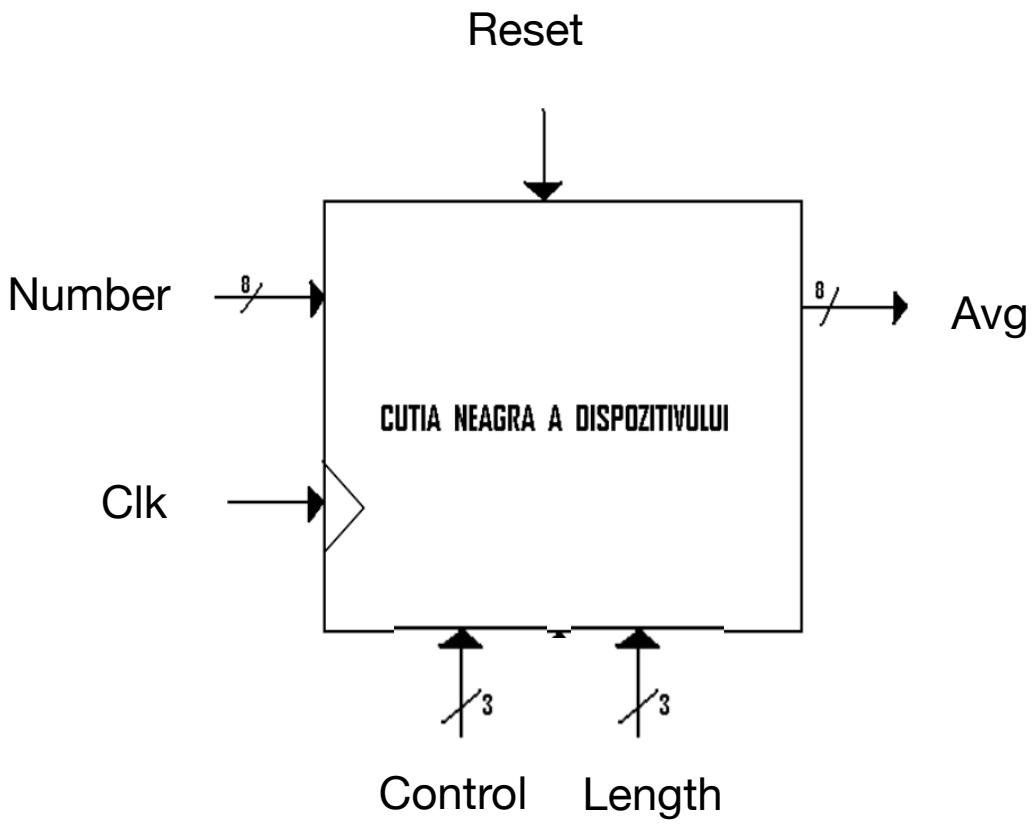
→Dezvoltarea unui dispozitiv de calcul a mediei unui set de numere.

- Design-ul va fi implementat pe Basys3 CPG236ABX1841
- Sistemul va fi dezvolat în VHDL utilizând modelul Active-HDL 6.3 și in Vivado 2019.2

Specificaţii:

→Va fi necesar un sistem de filtrare pentru a perimite sistemul de calcul să realizeze operațiile în timp real, atât afișarea numerelor generate aleator cât și media efectuată până în acel moment.

2.Descriere schemă bloc



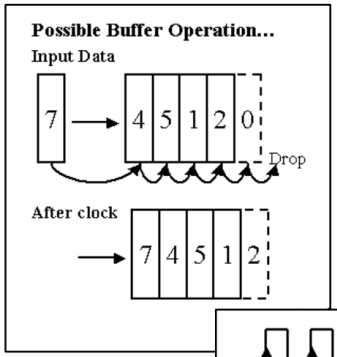
Idei de inceput

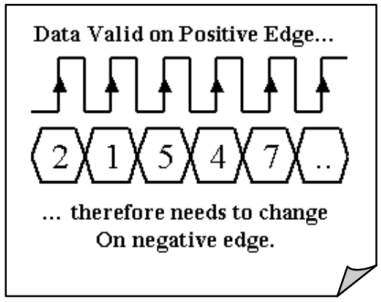
Example Rolling Average Over 4 Samples

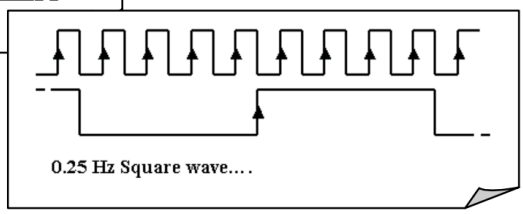
Sequence: 2 1 5 4 7 8 3 9 9 …

Values	Total	Average	Output
2 1 5 4	12	3.00	3
1 5 4 7	17	4.25	4
5 4 7 8	24	6.00	6
4 7 8 3	22	5.50	5 (or 6?)
7 8 3 9	27	6.75	6 (or 7?)
8 3 9 9	29	7.25	7

What happens at start up until buffer is full?



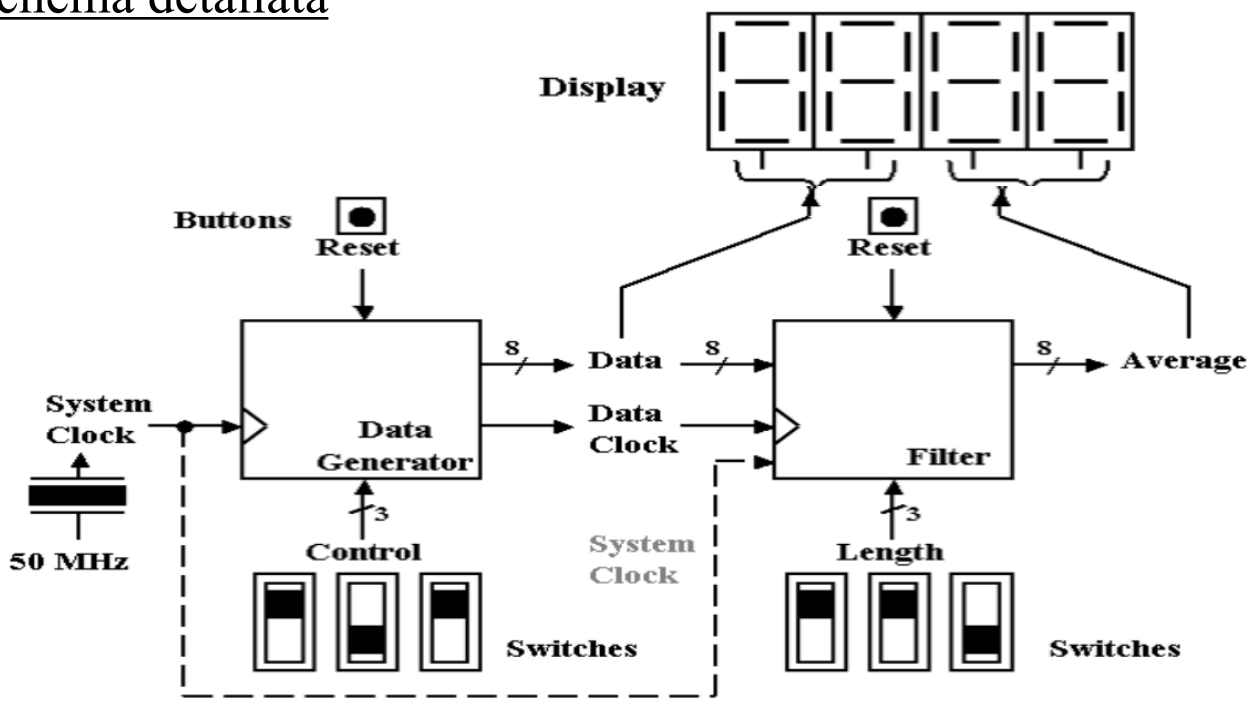




3.Etapele de proiectare

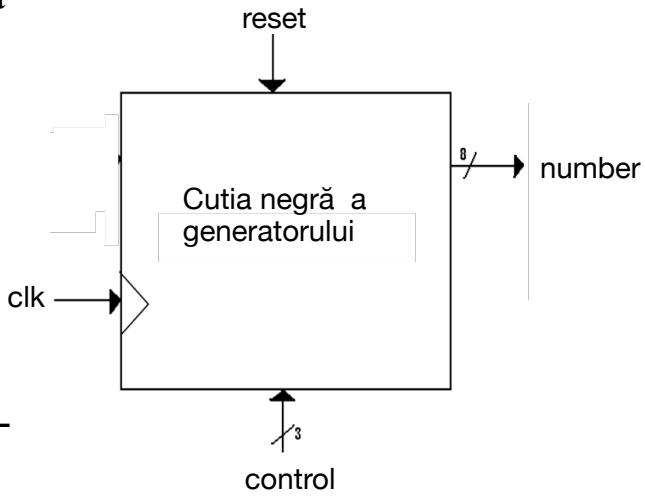
În prima fază , am construit o schema bloc a dispozitivul ,pentru a avea o privire de asamblu asupra proiectului.Pe urmă ,pornind de la aceasta schemă ,am construit-o pe cea detaliată ,identificand astfel componentele de care aveam nevoie pentru a realiza dispozitivul de calcul al mediei. După care ,urmatoarea etapă ,a fost scrierea codului, în Active-HDL ,a fiecarei componente în parte.Ulterior,a trebuit să construim o componentă (mai complexă) “cutie neagră “ ,pentru a lega componentele impreună .Cel mai dificil pas a fost verificarea funcţionari a ceea ce am proiectat ,prin testarea tuturor posibilitatţlor, pentru a ne asigura de faptul ca dizpotivul este implementat corect.Dupa ce ne-am convins ca în simulatorul Active-HDL, proiectul functioneaza comform cerinţelor, a trebuit să-l testăm şi pe placuţa FPGA pentru a vedea că şi acolo rezultatele sunt corecte.

Schema detaliată



1.Generatorul

Modul de funcționare al generatorului în funcție de semnalul de Control (pe 3 biți) este prezentat în tabelul de mai jos.Secvența de 6 numere a primului student este: 0-9-8-1-2-6. .Secvența de 6 numere a celui de al doilea student este: 1-4-6-8-3-7.Pentru generarea numerelor din intervalul 0 – 15 am folosit un registru implementat ca tablou pe 4 biți (variabila aleator_4), și pentru care intrarea serială este un sau-exclusiv între biții 3 și 2. Asemănător am lucrat și pentru generarea numerelor din intervalul 0 – 255 (cu aleator_8, exprimat ca tablou pe 8 biți), numai că sau-exclusiv s-a realizat între biții 7 și 6.



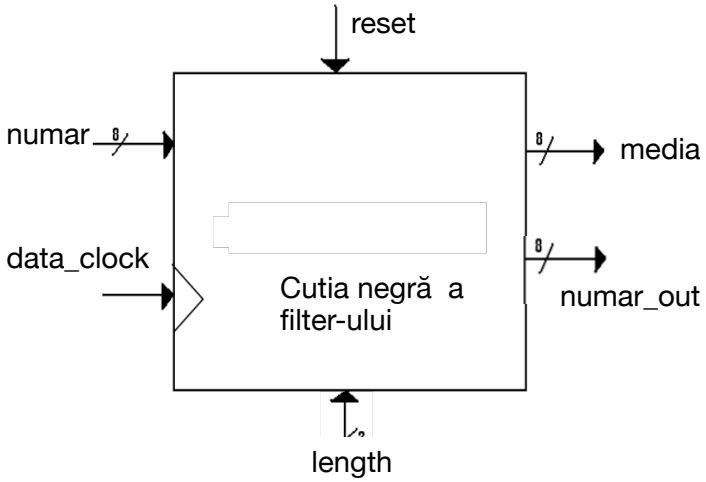
Registru funcţionează sincron motiv pentru care a fost necesară includerea unui semnal de tact.

2.Divizorul de frecvenţă

→Această componentă este folosită cu scopul de a tine ocupat procesorul plăcuței ”dându-i de lucru”.
→Divizorul de frecvenţă încetinește System clk-ul de la 100 MHz la 1 Hz

3.Filter

In funcție de semnalul de Lenght, dispozitivul calculeaza media unui set de numere(2,4,8,12).



Cand length-ul este 100 se calculează suma tot a câte doua numere și fiecare suma în parte este deplasată cu o poziție spre dreapta.

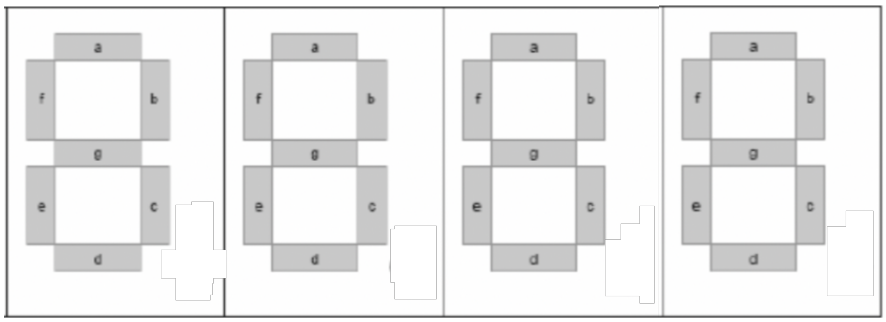
- Cand length-ul este 101 se calculează suma tot a câte patru numere ,folosindu-ne de sumele de la pasul anterior și suma este deplasată cu 2 poziții spre dreapta
- Cand length-ul este 110 se calculează suma tot a câte opt numere ,folosindu-ne de sumele precedente și suma este deplasată cu 3 poziții spre dreapta
- Cand length-ul este 111 se calculează suma tot a câte 16 numere ,folosindu-ne de sumele precedente și suma este deplasată cu 4 poziții spre dreapta

4.Divizorul de frecvență pentru anod

Este o componentă mai complexă,prin intermediul caruie se ține ocupat procesorul plăcuței , pentru ca numerele sa apara in timp real,și se determina ce anozi/catozi se vor folosi.
Am folosit un registru (r) pentru a reține informația de pe led.După care ne-am folosit de un contor(numar),care se incrementeaza in funcție de semnalul de tact, pentru a determina care anod este activ, și totodată pentru a asigna catodului corespunzător anodului (activ),informația de pe led.

5.Display

Este componenta prin intermediul căruia se va afisa pe placuță atat numarul generat cat și media calculată, dupa cum se observa mai jos.



→Se aseamană cu un multiplexor,care primeste ca intrare un vector pe 4 biți number, care reprezintă selecția ,în functie de care se determina iesirea
→După cum se observă din imagine ,afișoarele au cele 7 leduri reprezentând segmentele.

Numerele și media sunt afisate in hexazecimal ,aseamenea reprezentarilor de mai jos:

Cifrele reprezentate :1-15

4.Lista de componente utilizate

Componentele principale care alcătuiesc sistemul sunt: generatorul de numere și filtrul. Pe lângă acestea 2 am mai putea include aici divizorul de frecvență realizat la o secundă (regăsit în schema bloc sub numele de Clock System) cât și divizorul de frecvență folosit pentru a putea afișa pe display ,media și numerele,folosind toate cele 4 afișoare.

Interfața cu utilizatorul este realizată prin intermediul switch-urilor.
Prin intermediul switch-urilor de control, utilizatorul poate să-și aleagă, folosind diferite combinații, următoarele opțiuni:

- Numere aleatoare între 0-15;
- Numere aleatoare între 0-255.
- Secvența de 6 numere a primului student;
- Secvența de 6 numere al doilea student;

Prin intermediul switch-urilor de length, utilizatorul poate să-și aleagă, și aici, folosind tot diferite combinații, următoarele opțiuni:

- Medie pentru 2 numere;
- Medie pentru 4 numere;
- Medie pentru 8 numere;
- Medie pentru 16 numere;

Resetarea sistemului se va face și ea cu ajutorul unui switch.

Generatorul de numere:

→Modul de funcționare al generatorului în funcție de semnalul de Control (pe 3 biți) este prezentat în următorul tabel:

Control Settings:

Off-On-Off(010)	Secvența de 6 numere a primului student
Off-On-On(011)	Secvența de 6 numere a celui de al doilea student
On-On-Off(110)	Secvența pseudoaleatoare din intervalul 0-15
On-On-On(111)	Secvența pseudoaleatoare din intervalul 0-255
Alte combinații	Test Mode(se reduce numarul in starea 0)

Dispozitivul de calcul al mediei:

→Modul de funcționare al dispozitivului în funcție de semnalul de Length (pe 3 biți) este prezentat în următorul tabel:

Length Settings:

On – Off – Off(100)	Medie 2 numere
On – Off – On(101)	Medie 4 numere
On - On – Off(110)	Medie 8 numere
On – On – On(111)	Medie 16 numere
Off – Off – Off(000)	Stop-valoarea curentă

Semnalul de Clock

→la semnalul de Clock se va genera câte un număr din secvența pseudoaleatoare iar generatorul va transmite semnalul de tact dispozitivului de calcul al mediei. “Data Clock” rate:1Hz

Butoanele de reset

→Primul resetează generatorul.

→Al doilea resetează media.

Display

→pe primele doua afişoare se va afişa numărul generat la fiecare pas.

→pe ultimele două se va afişa media calculată.

5.Semnificaţia notaţiilor

Vom explica fiecare notaţie pe care am folosit-o în sursa cu codul VHDL, evidenţiind totodată entitatea din care face parte

→*generator:*

- control-un vector de 3 biți care hotărăște,conform tabelului *Control Settings* cum să fie generarea pseudoaleatorea de numere
- reset-ce resetează generatorul
- clk-semnalul de tact
- number-numarul generat(pe 8 biți)
- aleator_4-variabilă pentru secvența pseudoaleatoare pe 4 biți
- aleator_8-variabilă pentru secvența pseudoaleatoare pe 8 biți
- Aux-variabilă auxiliara pentru reținerea numerelor

→*divizor_generator:*

- clk-semnalul de tact
- reset-ce resetează
- Q-semnalul de clk divizat(din 50 MHz in 1 Hz)
- i-variabilă folosită ca un contor

→*filter:*

- numar-numărul generat de generator
- data_clock-semnalul de tact
- length- un vector de 3 biți care hotărăște,conform tabelului *Length Settings* media a câtor numere să se realizeze;
- reset-resetează media
- medie-media numerelor
- suma2-variabilă ce reține suma a 2 numere
- suma4-variabilă ce reține suma a 4 numere
- suma8-variabilă ce reține suma a 8 numere
- suma16-variabilă ce reține suma a 16 numere
- numar_out-ultimul numar care intră în filtru

→*divizor_anod:*

- clock_anod-semnalul de tact
- led-informația primită de pe leduri
- catod-iesire căruia i se genereaza in funcție de anodul ativ informația din registru
- anod-ieșire pe 4 biți care arată ce anod este activ
- r-variabilă tip registru folosită cu scopul de a reține informația de pe intrari (led)

→*display:*

- Data- o jumatate din media sau numarul generat, care trebuie afisată
- catod -transformara "data" în hexazecimal utilizând codurile corespunzătoare afişorului cu 7 segmente

→*CUTIE_NEAGRA_MEDIE:*

- clk-semnalul de tact(intern)
- reset_g-ce resetează generarea numerelor
- reset_f-ce resetează media
- Control-control-un vector de 3 biți care hotărăște,conform tabelului Control Settings cum să fie generarea pseudoaleatorea de numere
- Length-un vector de 3 biți care hotărăște,conform tabelului Length Settings media a câtor numere să se realizeze;
- Catod-codificarea pe 7 segmente/care pe displey este in hexazecimal;
- Anod-ieșire pe 4 biți care arată ce anod este activ;

→*tipuri_package:*

Aceasta reține toate tipurile de date folosite in proiect.

- student_array-un nou tip de date pentru a putea reține mai ușor secvențele studenților,in variabilele student1și student2
- memory-un nou tip de date (asemenea unei matrici)pentru a putea reține mai ușor numerele care intra in filtru
- sume-un nou tip de date pentru a putea reține toate cele 4 sume
- sum-un nou tip de date pentru a putea reține suma finală
- reg-un nou tip de date pentru a putea reține informația de pe led-uri fragmentate
- data_l-un nou tip de date pentru a putea reține in cutia neagră pentru a putea retine jumatati din data
- ies_c-un nou tip de date pentru a putea reține numerele generate de catod

6.Justificarea soluţiei alese

Am ales această soluție deoarece este ușoară de înțeles și intuitivă, precum și eficientă și simplu de implementat.Pentru implementarea proiectului am utilizat o metoda mai simpla și anume i*mparțirea pe componente,* folosind astfel 5 componente, prezentate mai sus, și a căror schemă de legătură reiese din schema bloc aflată pe pagina … Am folosit aceste componente atât pentru ca soluția să fie mai ușor de implementat, cât și pentru a ilustra modul de funcționare a fiecărei componente în parte.Toate componentele sunt legate in *cutia negră* printr-o descriere structurală .Prin urmărirea soluției alese, un utilizator cu orice nivel de experiență poate să înțeleagă funcționalitatea sistemului de calcul, și poate de asemenea să îmbunătățească sau să modifice codul în cazul în care ar vrea să se folosească de el pentru lucruri mult mai complexe.

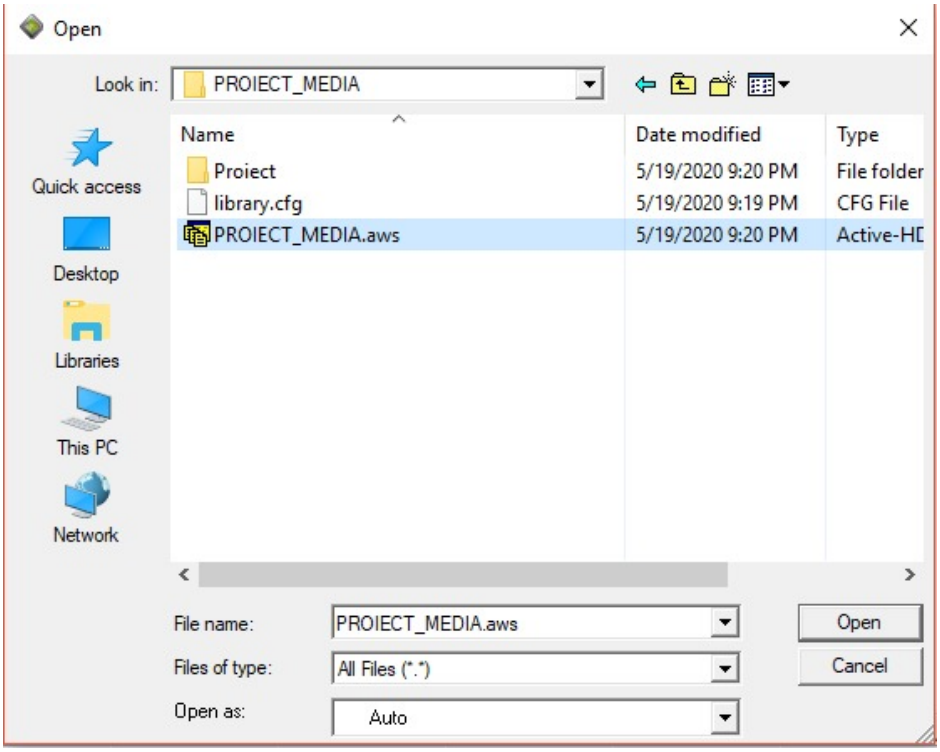
7.Utilizare şi rezultate

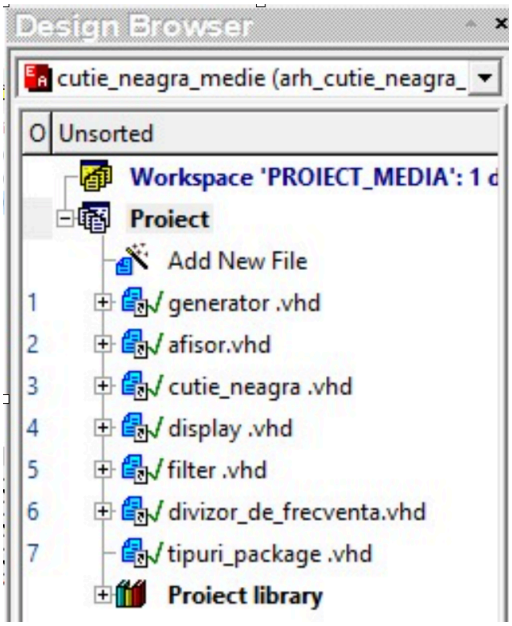
Partea de utilizare în Active HDL

→Pentru acest proiect este nevoie de mediul de proiectare Active-HDL 6.3.

1. Se va intra în programul Active-HDL 6.3.

2.Dupa ce acesta s-a încărcat alegem din meniul File optiunea Open, iar pe ecran se va deschide o fereastă unde avem posibiliatea de a selecat proiectul din directorul unde l-am salvat.

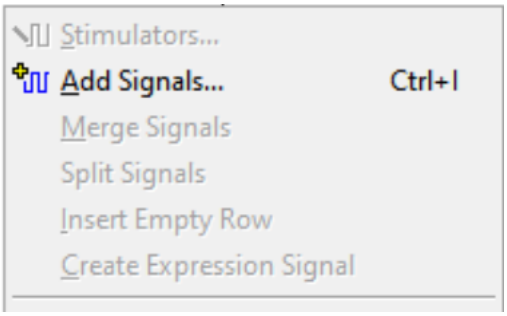




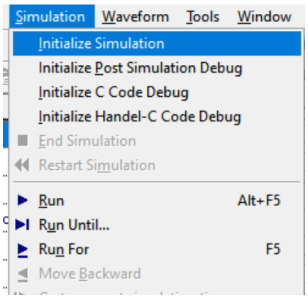
În meniul din partea stânga a ecranului "Design Browser" se pot vedea toate elementele ce compun proiectul sub denumirea data de utilizator.

3. Odata încărcat, proiectul va trebui compilat. Pentru acest lucru din meniul „*Design"* se va selecta *"Compile"* şi se va aştepta compilarea proiectului urmărind în căsuţa de „*Design Browser"* dacă în dreptul numelui fiecărui fişier apare o bifă de culoarea verde. Dacă acest lucru se întâmplă, atunci proiectul este pregătit pentru a fi simulat.

4. Se va crea un waveform selectând *"New Waveform"* din bara de sus.



5. Se vor adauga semnalele, dând click dreapta şi *"Add Signals"*, având însă grijă că în căsuţa de „*Design Browser"* să fie selectată arhitectura *"cutie_neagra"*.

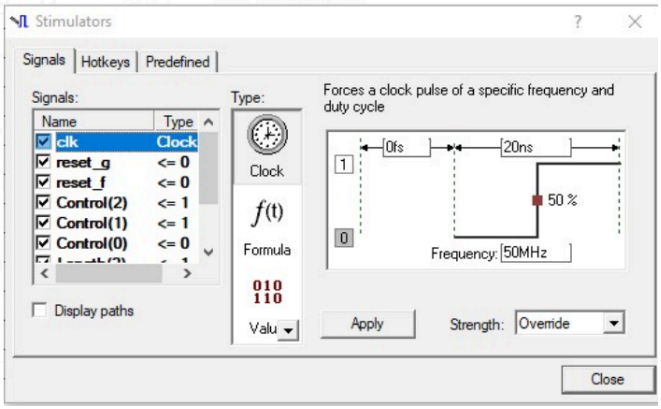


6. Din lista de semnale se vor selecta semnalele de care avem nevoie.

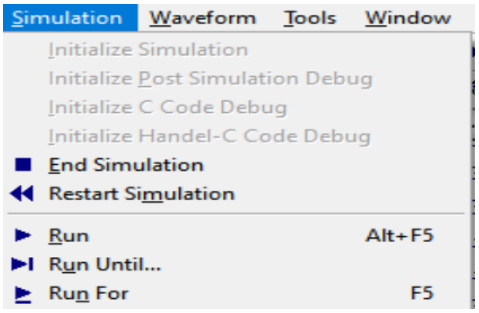
7. Se va iniţializa simularea cu *"Initialize Simulation"* din meniul *„Simulation"* aflat în bara de sus.

8. La toate semnalele de intrare se alocă câte un stimul pentru a simula comportamentul circuitului. Astfel în câmpul *Stimulator* se dă dublu click în dreptul semnalului dorit. Dacă semnalul este pe mai mult de un bit, atunci se vor putea aloca anumite taste pentru fiecare bit al semnalului.

Pentru acest lucru, se va face click pe semnul de plus din dreptul semnalelor şi apoi se va putea face alocarea de stimuli în dreptul fiecărui bit.



9. În lista *Type* se selecteaza tipul Stimulului. De exemplu dacă se selecteaza stimul de tip *Clock* se va genera un semnal periodic cu frecvenţa configurabilă de 50MHz. Dacă se selectează *HotKey* se va permite generarea de semnale aperiodice; starea intrării se schimbă la apăsarea tastei. Tasta este selectabilă. Se apasă *"Apply"* pentru fiecare intrare setată cu un stimul, iar în final "Close".



10. Pentru rularea simulării se apasă butonul de *"Run For"(* echivalentul tastei F5) aflat în bara de sus. Simulatorul va rula un interval de timp selectat din lista de lângă buton.

11. Dacă se doreşte restartarea simularii se va selecta *"Restart Simulation"* din meniul Simulation.

!!!!

Există posibilitatea ca in simulatorul Active-HDL ,proiectul să nu funcționeze corect datorită dizviorului de frecvență care determină apariția in timp real a numerelor pe placuță.

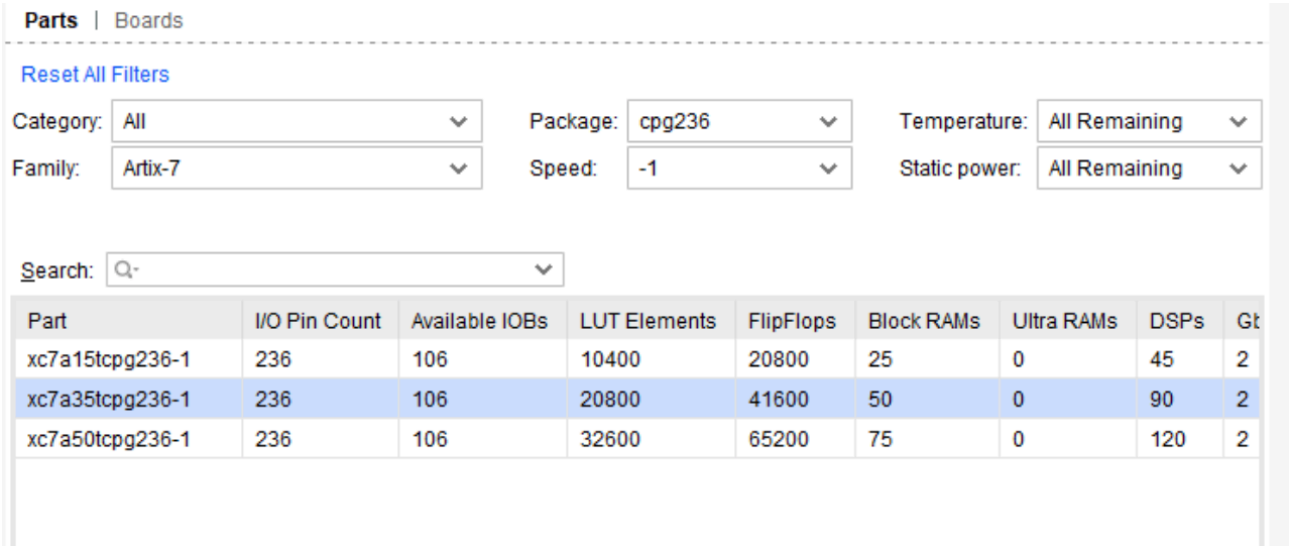
Partea de utilizare în Vivado şi pe plăcuţa FPGA

→*Pentru acest proiect este nevoie de mediul de proiectare Vivado 2019.2.*

1. Se va intra în programul Vivado 2019.2.

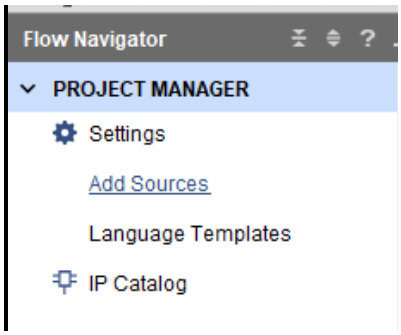
2.Dupa ce acesta s-a încărcat alegem , din *Quick Start* optiunea *Create Project* ,iar pe ecran se va deschide o fereastă unde vom selecta Next, apoi vom denumi proiectul cu numele care dorim. Vom selecta prima casuță RTL Project.

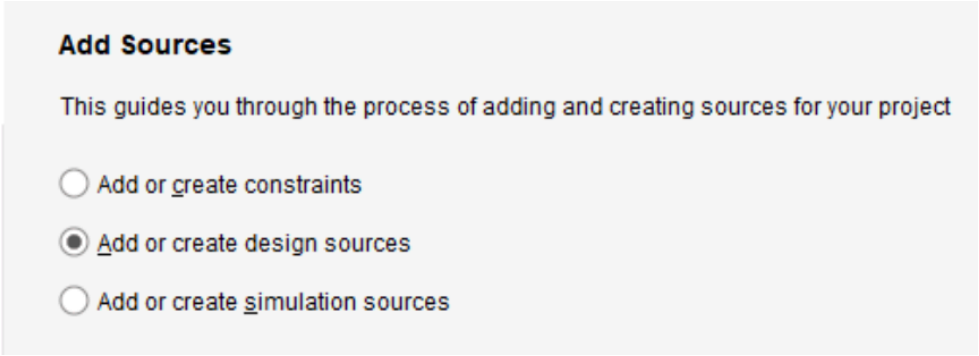
Va apărea o fereastră care cere informaţii despre plăcuţă ,iar noi selectata urmatoarele:



Se va da Next,iar apoi Finish.

3. Acum in partea stângă va scrie *PROJECT MANAGER*, unde vom selecta "*Add Sources"* iar apoi *"Add or create design sources"*. Apasam pe *"Add files " unde* adaugăm sursa cu codul vhdl din folder-ul "PROIECT". Selectăm Next și Finish.





4.Acum vom selecta din nou "Add Sources" ,doar ca de data aceasta vom alege *"Add or create constrints",* unde alegem *"Create File",* denumin lista de constrângeri cum dorim,după care apasam pe Finish. În căsuța din dreapta se va da copy-paste ,pentru pinii de mai jos:

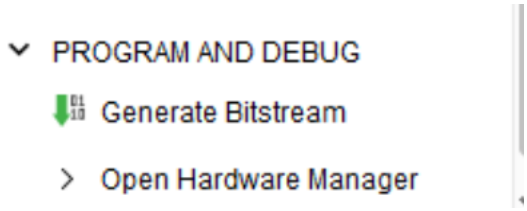
```
#Clock-ul
    set_property PACKAGE_PIN W5 [get_ports clk]
                set_property IOSTANDARD  LVCMOS33 [get_ports clk]
#Controlul
    set_property PACKAGE_PIN V17 [get_ports {Control[0]}]
                set_property IOSTANDARD  LVCMOS33 [get_ports {Control[0]}]
     set_property PACKAGE_PIN V16 [get_ports {Control[1]}]
                set_property IOSTANDARD  LVCMOS33 [get_ports {Control[1]}]
      set_property PACKAGE_PIN W16 [get_ports {Control[2]}]
                 set_property IOSTANDARD  LVCMOS33 [get_ports {Control[2]}]
#Length
       set_property PACKAGE_PIN W17 [get_ports {Length[0]}]
              set_property IOSTANDARD  LVCMOS33 [get_ports {Length[0]}]
       set_property PACKAGE_PIN W15 [get_ports {Length[1]}]
             set_property IOSTANDARD  LVCMOS33 [get_ports {Length[1]}]
        set_property PACKAGE_PIN V15 [get_ports {Length[2]}]
           set_property IOSTANDARD LVCMOS33 [get_ports {Length[2]}]
#Reseturile
        set_property PACKAGE_PIN U14 [get_ports reset_f]
            set_property IOSTANDARD  LVCMOS33 [get_ports reset_f]
        set_property PACKAGE_PIN V14 [get_ports reset_g]
             set_property IOSTANDARD  LVCMOS33 [get_ports reset_g]


#Catozii
         set_property PACKAGE_PIN W7 [get_ports {Catod[7]}]
              set_property IOSTANDARD  LVCMOS33 [get_ports {Catod[7]}]
         set_property PACKAGE_PIN W6 [get_ports {Catod[6]}]
               set_property IOSTANDARD  LVCMOS33 [get_ports {Catod[6]}]
```

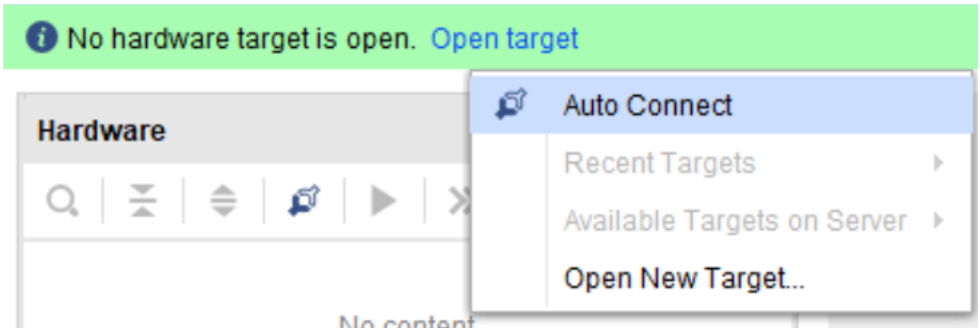
```
            set_property PACKAGE_PIN U8 [get_ports {Catod[5]}]
                 set_property IOSTANDARD  LVCMOS33 [get_ports {Catod[5]}]
            set_property PACKAGE_PIN V8 [get_ports {Catod[4]}]
                set_property IOSTANDARD  LVCMOS33 [get_ports {Catod[4]}]
           set_property PACKAGE_PIN U5 [get_ports {Catod[3]}]
                 set_property IOSTANDARD  LVCMOS33 [get_ports {Catod[3]}]
            set_property PACKAGE_PIN V5 [get_ports {Catod[2]}]
                 set_property IOSTANDARD LVCMOS33 [get_ports {Catod[2]}]
           set_property PACKAGE_PIN U7 [get_ports {Catod[1]}]
                 set_property IOSTANDARD  LVCMOS33 [get_ports {Catod[1]}]
            set_property PACKAGE_PIN V7 [get_ports {Catod[0]}]
                  set_property IOSTANDARD  LVCMOS33 [get_ports {Catod[0]}]


#Anozii
           set_property PACKAGE_PIN U2 [get_ports {Anod[1]}]
                 set_property IOSTANDARD  LVCMOS33 [get_ports {Anod[1]}]
           set_property PACKAGE_PIN U4 [get_ports {Anod[0]}]
                  set_property IOSTANDARD  LVCMOS33 [get_ports {Anod[0]}]
           set_property PACKAGE_PIN V4 [get_ports {Anod[3]}]
                 set_property IOSTANDARD  LVCMOS33 [get_ports {Anod[3]}]
           set_property PACKAGE_PIN W4 [get_ports {Anod[2]}]
                 set_property IOSTANDARD LVCMOS33 [get_ports {Anod[2]}]
```

5.Plăcuţa FPGA se va pune sub tensiune de la întrerupător.

6. Se va da click pe "*Generete Bitstream*", care se află in *PROGRAM AND DEBUG .Şi se asteaptă să se încarce.*

7. Dacă nu există erori, se va da pe *"Open Hardware Manager"*, iar apoi pe *"Open target"* unde se va selecta *"Auto Connect"*.

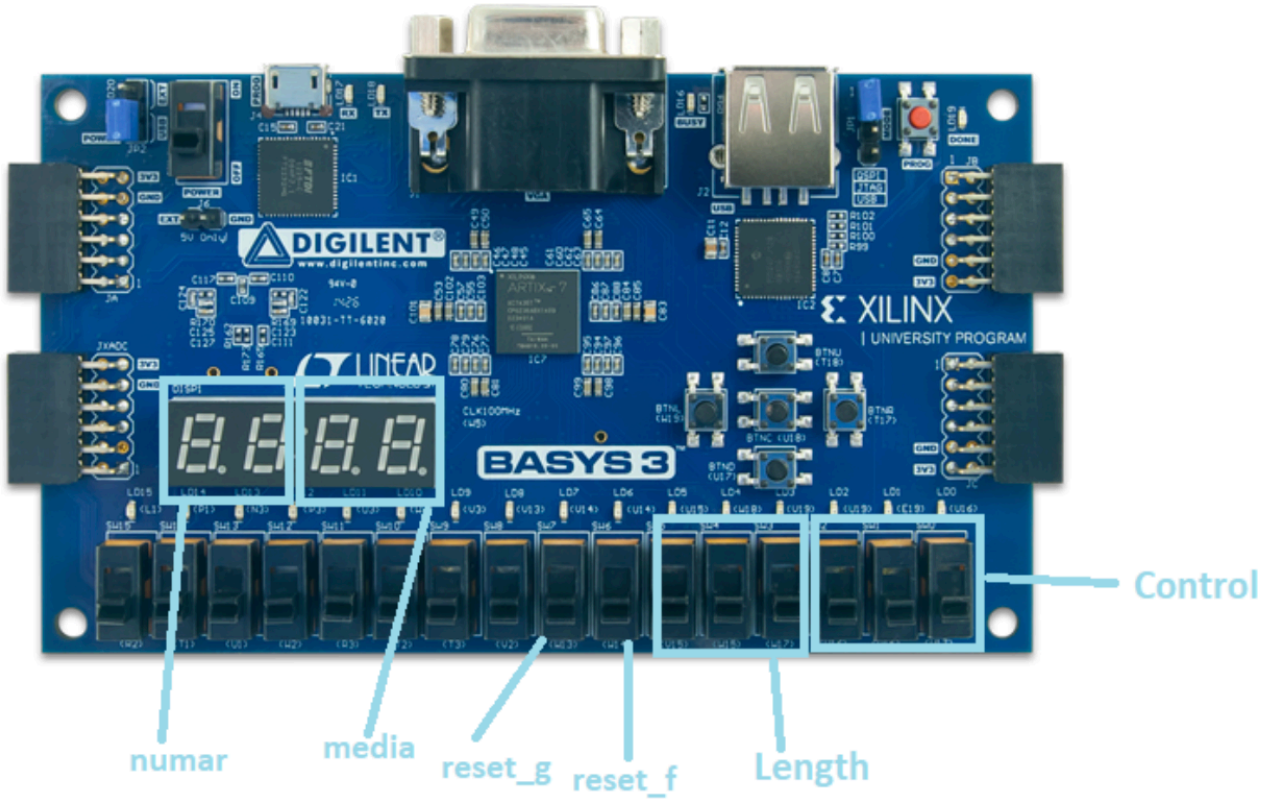




8.Click pe "*Program Device*", se alege sursa cu extensia *bit* .

9. Se face click pe Program şi se aşteapta încărcarea proiectului pe plăcuţa FPGA.

10. Odată încărcat, proiectul este gata pentru a fi testat pe plăcuţa Basys.



8.Posibilităţi de dezvoltare ulterioară

→Dispozitivul de calcul al mediei unui set de numere poate fi îmbunătățit în sensul adăugării unor alte funcții precum calcularea mediei geometrice sau armonice. De asemenea, prin adăugarea altor switch-uri pentru a reprezenta Control și Length pe mai mult de 3 biți, se poate crește cantitatea maximă de numere generate (numerele vor fi mai mari de 8 biți), precum și calcula media a mai mult de 16 numere.

→O altă îmbunătățire ar fi o aproximare mai exactă a mediei, astfel că pentru numerele cu zecimalele mai mici ca 50 să se rotunjească la partea întreagă a numărului, iar cele care au zecimalele mai mari sau egale cu 50 să se rotunjească la partea întreagă a numărului următor.