Andreea Georgiana Ionas

UTCN

 Grupa : 30217

Proiectarea sistemelor numerice.

Reclama publicitara cu animatii multiple

RECLAMA PUBLICITARA CU ANIMATII MULTIPLE

Faceți parte dintr-o echipa care are de proiectat o reclama publicitara cu animatii multiple, conceput special pentru organizatiile studentesti.

Reclama publicitara trebuie sa aiba un comprtament deosebit de simplu : afisajul 7 segmente va afisa reclama aleasa de utilizator(4 regimuri de functionare):

1. Afisare de la dreapta la stanga.
2. Curgere text.
3. Afisare litera cu litera.
4. Palpaire text.

Reclama trebuie sa functioneze dupa cum urmeaza:

* Starea initiala este prima stare a reclamei publicitare.
* Cand utilizatorule activeza unul dintre cele 4 regimuri de afisare (switch), se va afisa un text care se comporta conform regimului ales de utilizator.
* In momentul in care switch-ul ales este dezactivat, atunci textul va disparea de pe ecran, si se revine la starea initiala.

Table of Contents

[CAPITOLUL 1 : PROIECTARE 3](#_Toc104219119)

[1.1 Cutia Neagra 3](#_Toc104219120)

[1.1.1 Maparea intrarilor si iesirilor cutiei mari pe cele doua componente UC si UE. 5](#_Toc104219121)

[1.1.2 Determinarea resurselor (UE) 6](#_Toc104219122)

[1.1.3 Schema bloc a primei descompuneri. 8](#_Toc104219123)

[1.2 Reprezentarea UC prin diagrama de stari (organigrama). 8](#_Toc104219124)

[Diagrama de stari reprezinta partea de control, partea decizionala din orice algoritm. 8](#_Toc104219125)

[1.3 Justificarea solutiei alese 10](#_Toc104219126)

[1.4 Manual de utilizare si intretinere 10](#_Toc104219127)

[1.5 Posibilitati de dezvoltare ulterioara 15](#_Toc104219128)

# 

# CAPITOLUL 1 : PROIECTARE

## Cutia Neagra

Shape

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Denumire | Intrare/ Iesire | Nr.biti | Descriere |
| Clock | in | 1 | Semnal de tact activ pe front ascendent. |
| Switch | in | 4 | Vor exista 4 regimuri de funcionare fiecare corspunzand unui switch. La selectia unui switch se activeaza regimul specific. |
| Anozi | out | 4 | Semanle de control pentru afisaj 7 segmente. |
| Catozi | out | 8 |

Cutia neagră a sistemului trebuie descompusa mai departe pentru a putea găsi componenteimplementabile. Vom face o descompuneretop-downaproblemei pana cand ajungem lacircuite cunoscute, iar apoi vom implementabottom-up.Prima descompunere a oricărui sistem este una în care vom diferenta intrelogica de controldin sistem șiresursele sistemului. Logica de controleste reprezentata de Unitatea de Controliar resursele sunt reprezentate de Unitatea de Execuție. Orice algoritm poate fi descompus inacest fel.

A picture containing diagram

Description automatically generated

### Maparea intrarilor si iesirilor cutiei mari pe cele doua componente UC si UE.

Diagram, schematic

Description automatically generated

Putem împărți atat intrările cat și ieșirile în 2 categorii:de date si de control. Aceasta separareeste esențială la început:

* ***Intrări de date***: ---.
* ***Intrari de control***: switch pentru selectarea moduli de functionare , clock.
* ***Ieșiri de date***: anozii si catozii (afisaj 7 segmente).

### Determinarea resurselor (UE)

Pentru a stabili mai departe legăturile dintre UC și UE trebuie mai intaisa identificamresursele pe baza cărora luăm decizii.Aceste resursepot sa genereze semnale cătreunitatea de control și pot fi controlate de UC prin semnale de Enable sau Reset.

Orice informație pe baza careia se ia decizii trebuie sa vina de la o resursa care genereazaacea informație și o transmite mai departe UC.Resursele pot ficircuite simple,care pot fi implementatedirect (numarator, registru etc)sauresurse complexe(algoritm de dat rest, algoritmde inmultire, etc).

Aceste resursecomplexe pot apărea în prima descompunere cu cutii negre cărora trebuie sa le stabilim intrari si iesiri, dar ulterior trebuie descompuse mai departe (de obicei tot în UC și UE) pana candajungem la circuite cunoscute.

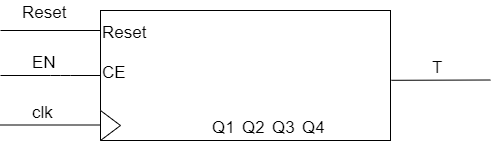
***RESURSE:***

*1.Memorie ROM*

Aceasta este o memorir ROM. Pentru a functiona reclama vom aveae nevoie de 4 astfel de memorii. Pentru a retine animatiile.

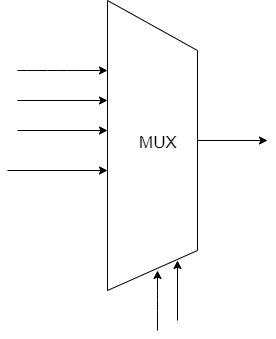
*2.Numarator*

Pentru acesta reclama vom avea nevoie de 2 numaratoare. Unul dintre ele care va parcurge liniile din memorii. Iar cel de-al doilea numarator este folosit ca divizor de fragventa.

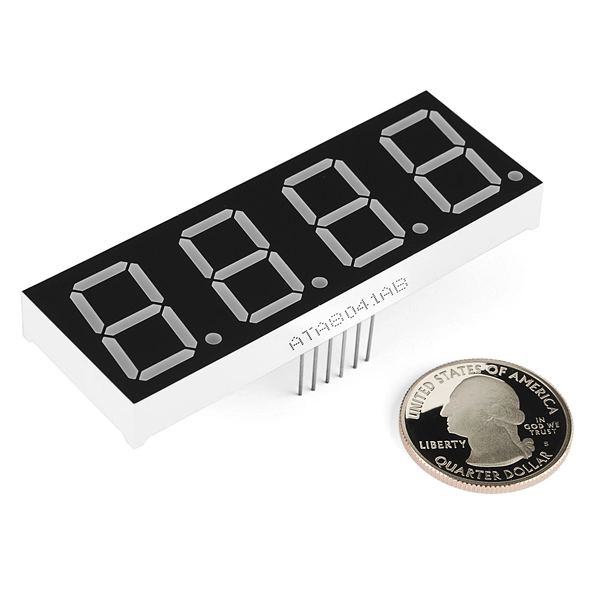


*3.Multiplexor*

Pentru acesta reclama vom avea nevoie de 2 multiplexoare. Pentru a selecte display-urile pe care sa afisam.



*4.Afisaj 7 segmente*



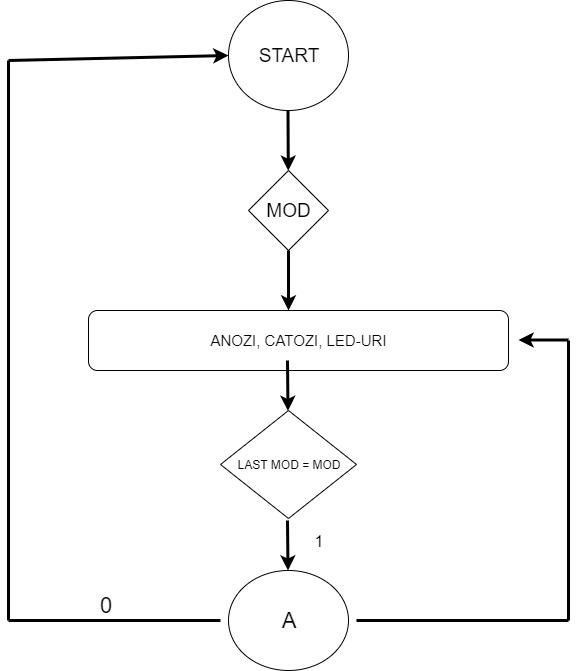
### Schema bloc a primei descompuneri.

Diagram, schematic

Description automatically generated

## Reprezentarea UC prin diagrama de stari (organigrama).

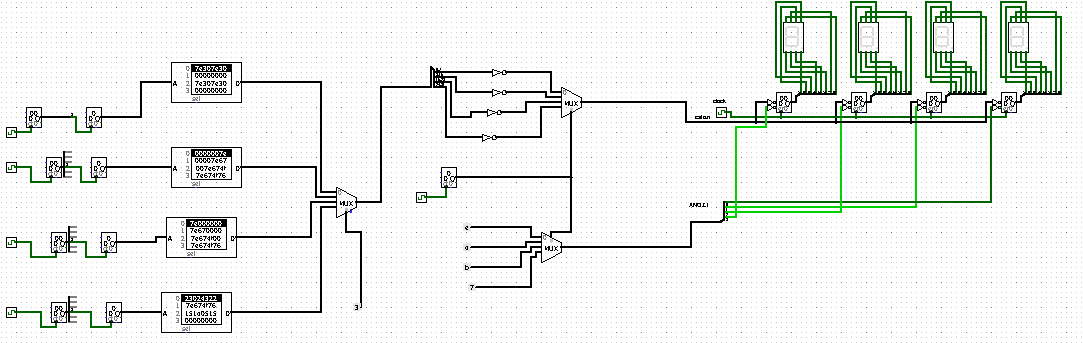
### Diagrama de stari reprezinta partea de control, partea decizionala din orice algoritm.



## Justificarea solutiei alese

Am ales aceasta solutie deoarece este una simpla si eficienta.

La inceput am implementat un numarartor pe 27 biti pentru a alege frecventa cu care sa se itereze cadrele din memorie. In acest proiect avem implementate 4 memorii ROM, una pentru fiecare animatie. Outputrile memoriilor le legam la un mux 4:1, cu selectia data de switchurile de pe placuta FPGA. Dupa ce s-a selectat animatia dorita, outputul mux-ului ne va indica catozii.



## Manual de utilizare si intretinere

Pentru a lucra la acest proiect am folosit Vivado Design Suite.

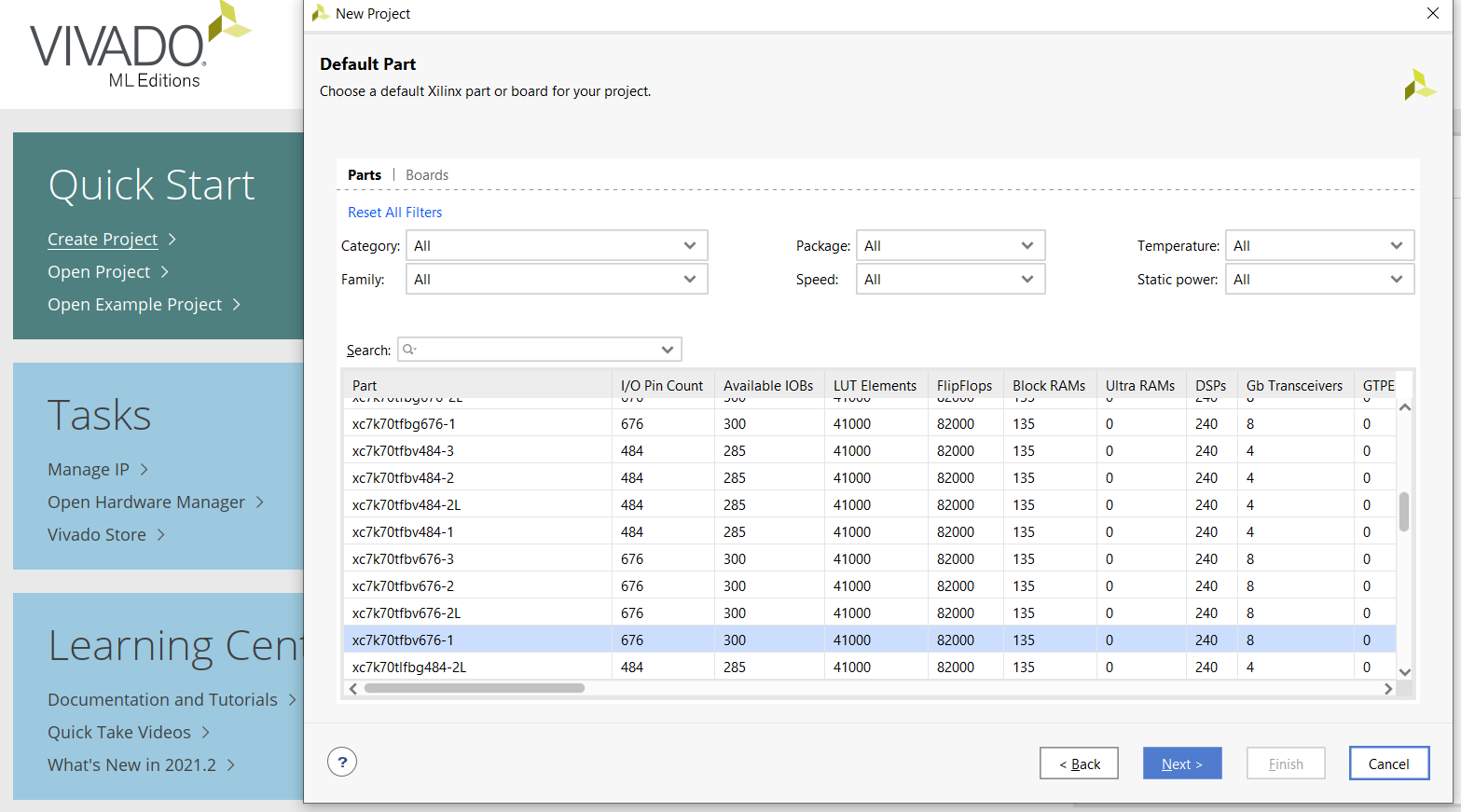
Vivado Design Suite este o suita de software produsa de Xilinx pentru sinteza si analiza modelelor de limbaj de descriere hardware (HDL) , inlocuind Xilinx ISE cu functii suplimentare pentru dezvoltarea sistemului pe un cip si sinteza la nivel inalt.

Pentru a crea un proiect nou in Vivado, se da dublu click pe pictograma de la Vivado, iar dupa ce pe ecran apare interfata de la Vivado, se da click pe ”Create Project”:

Graphical user interface, application

Description automatically generated

Apoi se selecteaza numele proiectului si specificatiile de la placuta FPGA pe care se va implementa proiectul.

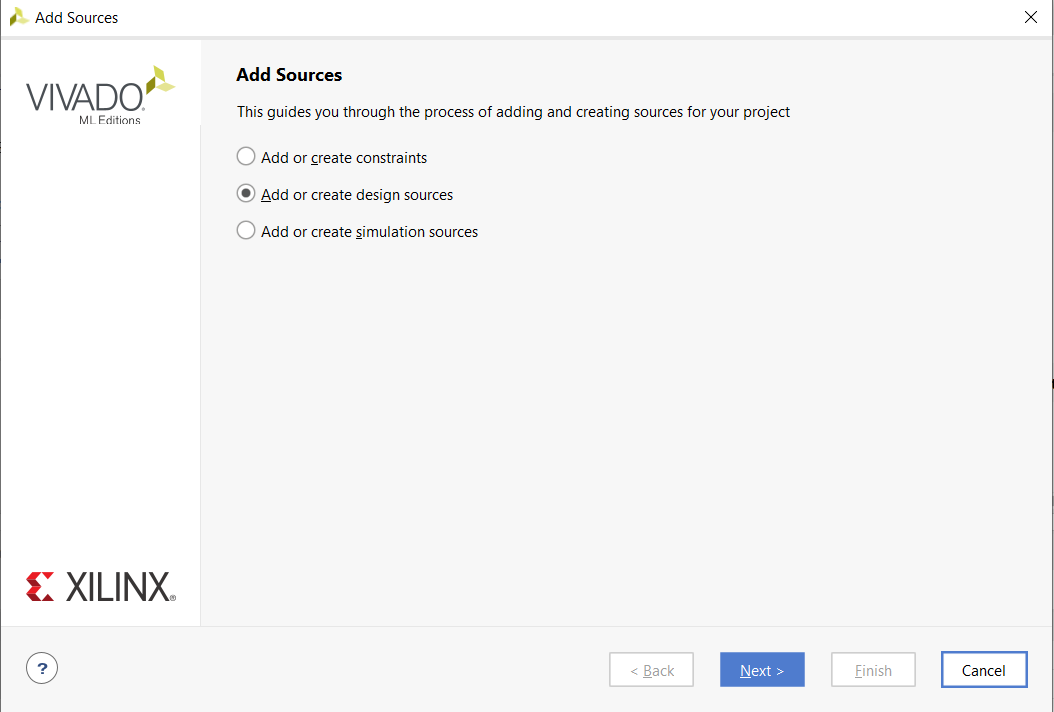


Pentru a crea un proiect se da click pe butonul cu simbolul ”+” :

A screenshot of a computer

Description automatically generated

Se selecteaza ”Create design sources”, apoi se da click pe ”Create File” , unde se alege numele proiectului si datele de intrare si iesire:



Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

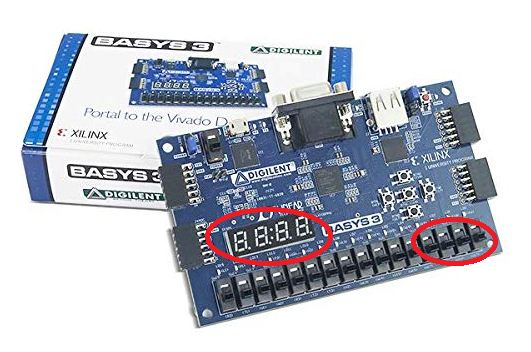
Description automatically generated

In cele din urma se da click pe proiectul creat si va aparea o fereastra unde se poate scrie codul VHDL pentru circuitul dorit:

Graphical user interface, text, application

Description automatically generated

Programul finit va fi apoi implementat pe placuata FPGA.



Switch-uri corespunzatoare animatiilor.

Afisaj



Pentru intretinerea placutei FPGA este de ajuns sa o pastrati la o temperatura intre 0 si 85 grade Celsius.

Placuta trebuie ferita de socuri, lichide si alimente.

## Posibilitati de dezvoltare ulterioara

Ulterior, la reclama s-ar putea adauga mai multe animatii, pe langa cele realizate deja.

Spre exemplu, s-ar putea implememta afisarea textului de la stanga la dreapta sau afisarea textului care urca pe afisor.

Ambele exemple s-ar putea implementa relativ usor, dar va fi nevoie de mai multe memorii, iar mux-urile ar trebui sa isi mareasca capacitatea, la fel si selectia, adica mai multe switch-uri, pentru a selecta animatiile.

Bibliografie

* <https://www.xilinx.com/products/design-tools/vivado.html>
* <https://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf>
* <https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>
* <https://www.jameco.com/Jameco/workshop/TechTip/working-with-seven-segment-displays.html>