



**UNIVERSITATEA
TEHNICĂ
CLUJ-NAPOCA**

- Animație text -

-FPGA Nexys4-

Proiect realizat de Șular Ștefan

Facultatea de automatică și calculatoare

Anul I de studiu

Grupa 30213

Indrumător: Dragoș Florin Lișman

Cuprins

Cuprins.....	2
1. Specificatiile proiectului	4
1.1 Cerinta.....	4
1.2 Placuta FPGA.....	4
1.3 Alfabetul utilizat	6
2. Prezentarea generală a proiectului	7
2.1 Black box	7
2.2 Schema logica	8
3. Componentele	9
3.1 Prezentare generală	9
3.2 Message_ROM	10
3.3 Decodificator Litera -> Seven-Segment	10
3.4 Animatii	11
3.4.1. Animatie 1	11
3.4.2. Animatie 2.....	11
3.4.3. Animatie 3.....	12
3.4.4. Animatie 4.....	13
3.5 Clock Divider.....	13
3.6 Multiplexor animatii	14
3.7 Display	14
4. Notatii.....	15
4.1 Semnale Interne	15

5.	De ce aceasta metoda de rezolvare?.....	15
6.	Dezvoltări ulterioare	16

1. Specificatiile proiectului

1.1 Cerinta

Să se proiecteze o reclamă publicitară cu animații multiple. Se vor folosi afișajele cu 7 segmente. Textul de afișat va fi format din simboluri ale unui alfabet disponibil. Reclama va avea mai multe regimuri de funcționare (minimum 4) ce vor putea fi selectate de către utilizator, de la comutatoarele plăcuței cu FPGA. Se va folosi oscilatorul de cuarț încorporat în plăcuța cu FPGA (semnalul de clock respectiv va trebui desigur să fie divizat). Exemple de regimuri de funcționare: „curgerea” scrisului de la dreapta spre stânga, pâlpâire, afișaj literă cu literă etc.

Deoarece pe un afișaj cu 7 segmente nu se pot reprezenta toate literele, se va crea un alfabet maximal și mesajele vor fi compuse din simbolurile acelui alfabet. Mesajul va fi conținut într-o memorie pentru a putea fi ușor de schimbat. Proiectul va fi realizat de 1 student.

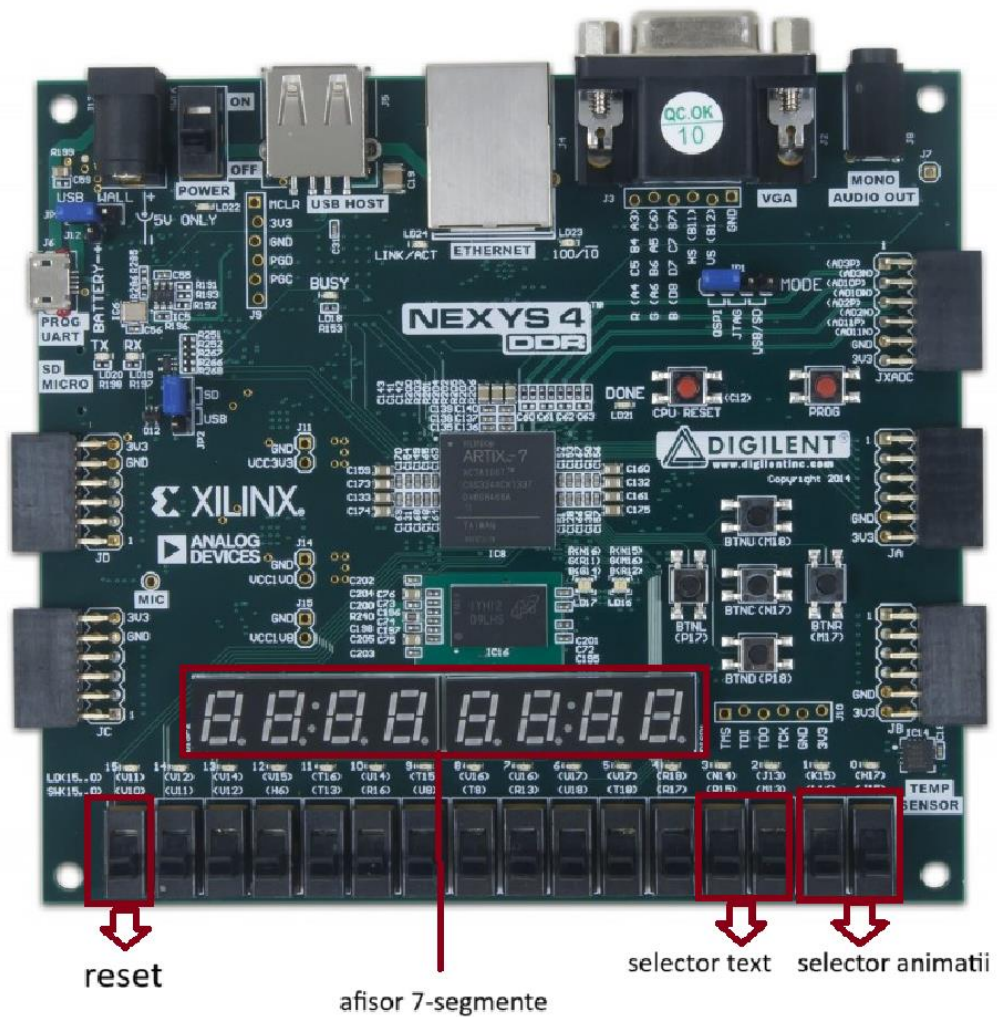
1.2 Placuta FPGA

Acest proiect, este realizat pentru placa Nexys 4, datorită afișajului cu 7 segmente mai lat, care permite introducerea unor mesaje mai lungi.

Placa Nexys 4 este o platformă completă și gata de utilizat pentru dezvoltarea circuitelor digitale, bazată pe cel mai nou FPGA Artix-7™ de la Xilinx. Cu un FPGA de capacitate mare (codul Xilinx: XC7A100T-1CSG324C), memorii externe generoase și o colecție de porturi USB, Ethernet și altele, Nexys 4 poate găzdui proiecte variind de la circuite combinaționale introductive până la procesoare încorporate complexe.

Mai multe periferice integrate, inclusiv un accelerometru, senzor de temperatură, microfon digital MEMs, amplificator pentru difuzor și multe dispozitive de intrare/ieșire, permit utilizarea plăcii Nexys 4 într-o gamă largă de proiecte, fără a fi nevoie de alte componente suplimentare.

Câteva specificații ale plăcii FPGA vor fi prezentate în secțiunea următoare, alături de denumirile pinilor folosiți și butoanele și switch-urile de intrare utilizate.



În imaginea precedentă, componentele utilizate au fost evidențiate. În continuare, vor fi prezentate imagini mai detaliate care vor arăta denumirea și funcția fiecărei componente în parte.

V10 – Switch-ul pentru reset

L16, J15 – Switch-ul pentru selectare animatie

R15, M13 – Switch-ul pentru selectare text

































7-Segmente Display - Afiseaza textul cu multiple animatii



În plus, generatorul de cuarț integrat în placa FPGA generează un semnal de ceas cu o frecvență de 100 MHz.

1.3 Alfabetul utilizat

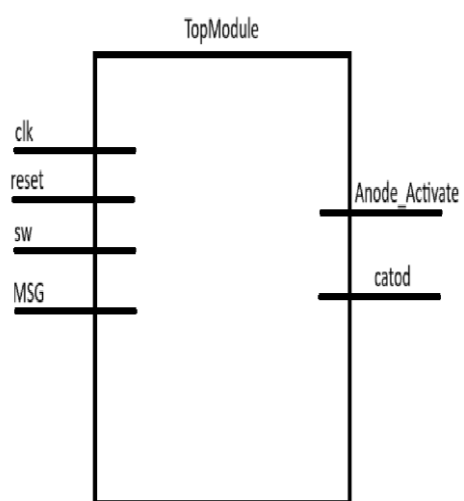
Având în vedere că nu toate literele pot fi reprezentate pe afișajul cu 7 segmente, am conceput un model pentru anumite litere din alfabetului englez si numere. În imaginile următoare, literele de la A la Z vor fi reprezentate în ordine, cat si numere de la 0 la 9. De asemenea caracterul spatiu va fii folosit ca o setare de tip default.

									
A	B	C	D	E	F	G	H	I	L
									
n	O	P	S	T	U	V	Y	Z	Space
									
0	1	2	3	4	5	6	7	8	9

2. Prezentarea generală a proiectului

2.1 Black box

În imaginea următoare va fi prezentată o schemă simplificată (black box) a proiectului, împreună cu semnificația pinilor utilizați.



CLK -> Ceasul plăcii funcționează la o frecvență de 100MHz

RESET -> Resetează toate componentele

SW -> Selectează animația

MSG -> Selectează mesajul

În imaginea următoare va fi prezentat un fragment de cod care reprezintă entitatea descrisă anterior:

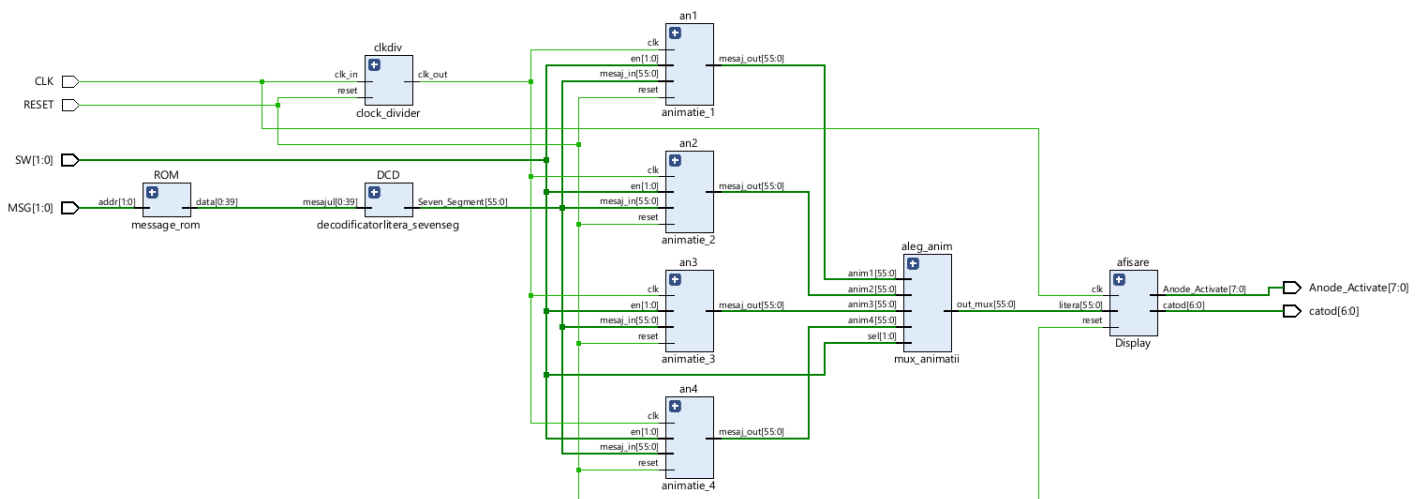
```

entity TopModule is
  Port (
    CLK : in  STD_LOGIC;
    RESET : in  STD_LOGIC; -- reset
    SW : in  STD_LOGIC_VECTOR(1 downto 0); -- selector animatii
    MSG : in  STD_LOGIC_VECTOR(1 downto 0); -- selector mesaj
    Anode_Activate : out STD_LOGIC_VECTOR(7 downto 0);
    catod : out STD_LOGIC_VECTOR(6 downto 0)
  );
end TopModule;
  
```

2.2 Schema logica

Componenta principală a întregului proiect este **MainAdv** (care a fost prezentată anterior). În continuare, este atașată o imagine ce conține diagrama bloc a acestei componente.

Această componentă leagă între ele mai multe subcomponente, care vor fi prezentate în capitolele următoare.



Componentele utilizate și prezentate în imagine sunt:

- ❖ Message_ROM
- ❖ Decodificator Litera -> Sevent-Segment
- ❖ Clock Divider
- ❖ Animatie 1
- ❖ Animatie 2
- ❖ Animatie 3
- ❖ Animatie 4
- ❖ Multiplexor Animatii
- ❖ Display

3. Componentele

3.1 Prezentare generală

Această secțiune oferă o imagine de ansamblu asupra structurii și funcționării proiectului. Sunt descrise principalele module utilizate, modul în care acestea interacționează între ele și scopul general al arhitecturii implementate. Se evidențiază, de asemenea, abordarea utilizată pentru descrierea componentelor (structurală sau comportamentală), precum și organizarea generală a proiectului la nivel de componente.

Pentru o înțelegere mai clară a componentelor utilizate, precum și pentru o perspectivă mai bună asupra tuturor subcomponentelor, în continuare este atașată o imagine a arborelui de componente al proiectului.

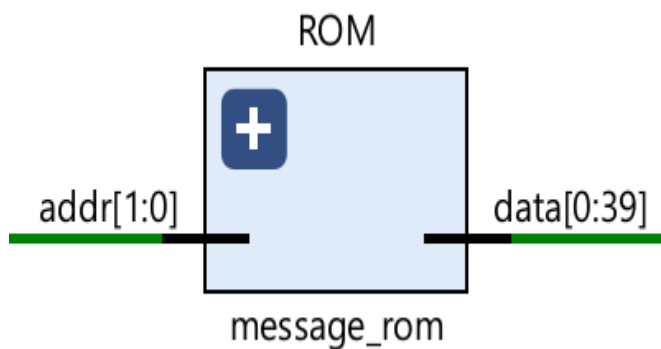
Design Sources (1)

TopModule(Behavioral) (Main.vhd) (9)

- an1 : animatie_1(Behavioral) (animatie_1.vhd)
- an2 : animatie_2(Behavioral) (animatie_2.vhd)
- an3 : animatie_3(Behavioral) (animatie_3.vhd)
- an4 : animatie_4(Behavioral) (animatie_4.vhd)
- ROM : message_rom(Behavioral) (ROM_mesaj.vhd)
- DCD : decodificatorlitera_sevenseg(Behavioral) (DCD.vhd)
- clkdiv : clock_divider(Behavioral) (CLK_divider.vhd)
- aleg_anim : mux_animatii(Behavioral) (mux_animatii.vhd)
- afisare : Display(Behavioral) (7segment_afisare.vhd)

3.2 Message_ROM

Message_ROM are rolul de a contine mesajul pe care dorim sa il afisam pe seven-segment, acest mesaj fiind unul hard-codat.

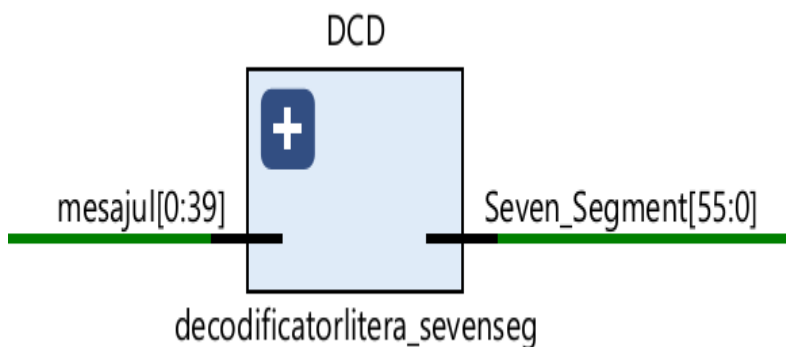


ADDR -> reprezintă adresa sau indexul în memorie unde se găsește mesaj-ul

DATA -> este valoarea asociată acelei adrese, care va iesi, adică mesaj-ul hard-codat

3.3 Decodificator Litera -> Seven-Segment

Decodificator Litera -> Seven-Segment are rolul de a converti mesajul primit de la Message_ROM intr un cod de afisare pentru 7-segment. Putem spune ca acest decodificator are rolul unui dictionar, intru cat acesta primeste mesajul intr un cod cunoscut de catre utilizator si il converteste in cod-ul cunoscut de catre masina.



MESAJUL -> este mesajul primit de la Message_ROM

SEVEN_SEGMENT -> mesajul decodificat

3.4 Animatii

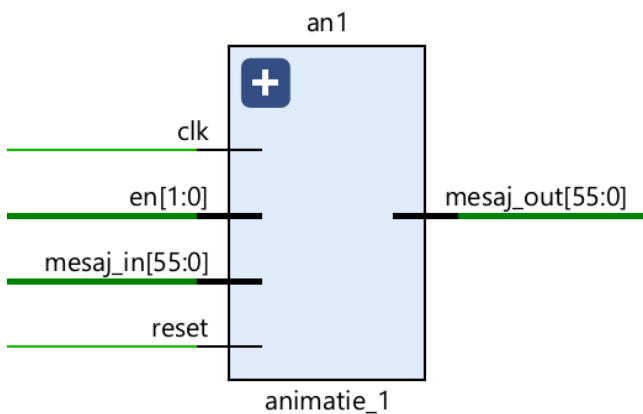
Animatiile pe care le-am ales pentru afisare sunt: afisarea normala, blinking, scroll left to right si typing. Modul in care functioneaza animatiile este in felul urmator. In functie de ce animatie este activata, animatia va modifica codul mesajului de afisare.

3.4.1. Animatie 1

Animatia 1 consta in modul de afisare normala, unde mesajul selectat se va afisa exact la fel ca in poza atasata:



Componenta Animatie_1:



CLK -> ceasul care functioneaza la o frecventa de 1MHz

EN -> porneste animatia

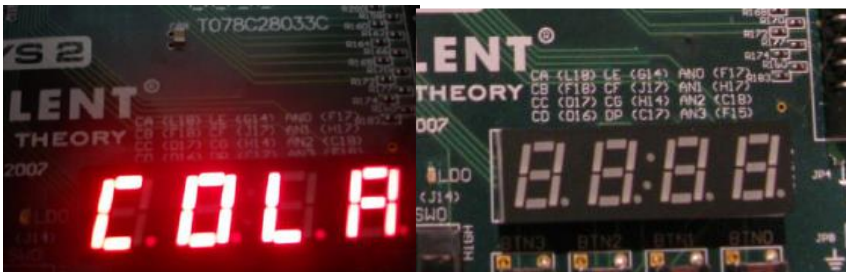
MESAJ_IN -> mesajul decodificat pentru seven-segment

RESET -> reseteaza animatia

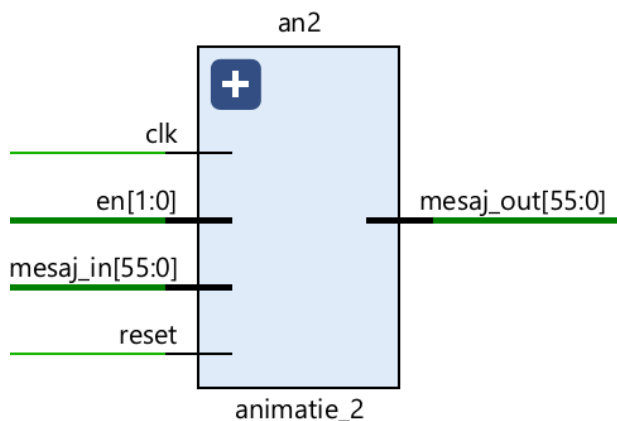
MESAJ_OUT -> mesajul care v a fii afisat pe seven-segment

3.4.2. Animatie 2

Animatia 2 consta in modul de afisare blinking, unde mesajul selectat se va afisa exact la fel ca in pozele atasate:



Modul de functionare consta in odata la schimbarea de ceas, mesaj_out v-a lua valoarea mesajului_in sau others => '1' (adica spatiu)



CLK -> ceasul care functioneaza la o frecventa de 1MHz

EN -> porneste animatia

MESAJ_IN -> mesajul decodificat pentru seven-segment

RESET -> reseteaza animatia

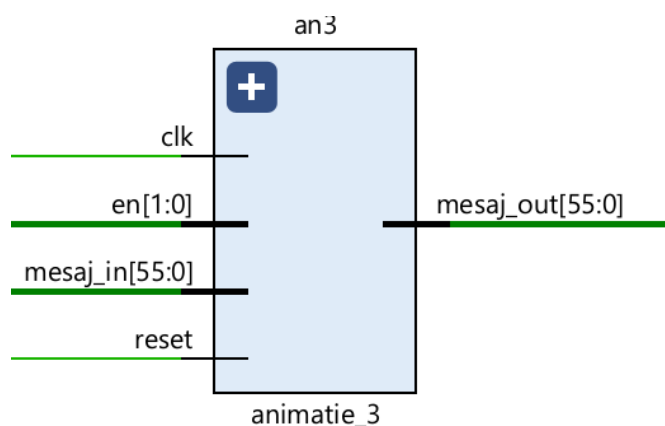
MESAJ_OUT -> mesajul care v a fii afisat pe seven-segment

3.4.3. Animatie 3

Animatia 3 consta in modul de afisare scroll left to right, unde mesajul selectat se va afisa exact la fel ca in pozele atasate:



Modul de functionare consta in odata la schimbarea de ceas, mesaj_out v-a concatena de la o memorie cate un cuvant pana la formarea intregului cuvant.



CLK -> ceasul care functioneaza la o frecventa de 1MHz

EN -> porneste animatia

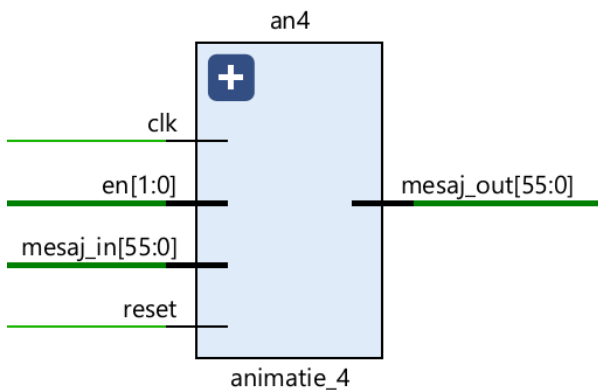
MESAJ_IN -> mesajul decodificat pentru seven-segment

RESET -> reseteaza animatia

MESAJ_OUT -> mesajul care v a fii afisat pe seven-segment

3.4.4. Animatie 4

Animatia 4 consta in modul de afisare typing, unde mesajul selectat se va afisa litera cu litera.



CLK -> ceasul care functioneaza la o frecventa de 1MHz

EN -> porneste animatia

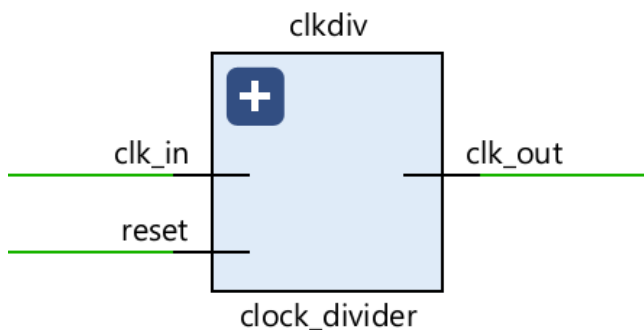
MESAJ_IN -> mesajul decodificat pentru seven-segment

RESET -> reseteaza animatia

MESAJ_OUT -> mesajul care v a fii afisat pe seven-segment

3.5 Clock Divider

Clock divider-ul care transformă frecvența ceasului de la 100 MHz la 1 MHz are rolul de a reduce viteza cu care sunt executate procesele pentru animatii.



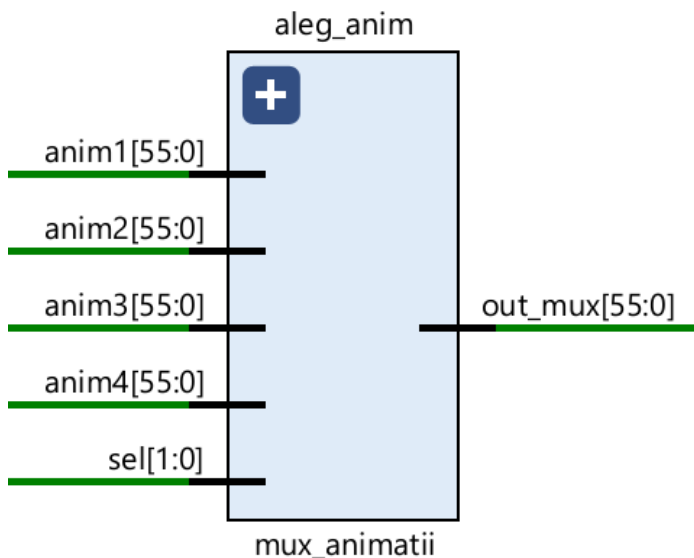
CLK_IN -> ceasul placii care functioneaza la o frecventa de 100MHz

RESET -> reseteaza componenta

CLK_OUT -> ceasul care este modificat la o frecventa de 1MHz

3.6 Multiplexor animatii

Multiplexor-ul de animatii are rolul de a selecta in functie de switchurile de pe placa animatia



ANIM1 -> mesajul modificat de catre animatia 1

ANIM2 -> mesajul modificat de catre animatia 2

ANIM3 -> mesajul modificat de catre animatia 3

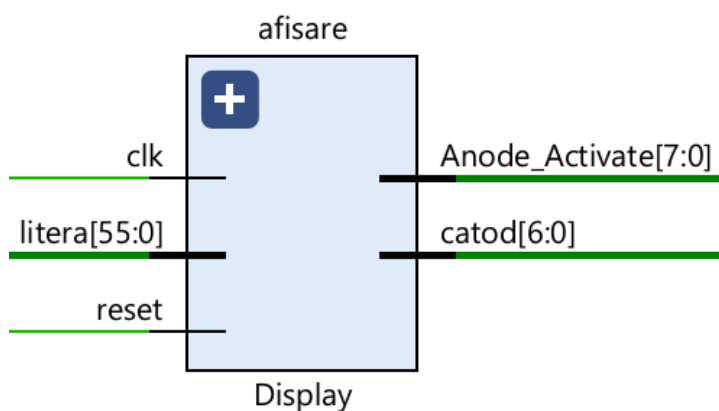
ANIM4 -> mesajul modificat de catre animatia 4

SEL -> mesajul pe care sa il transmita mai departe

Out_mux -> mesajul selectat

3.7 Display

Display are rolul de a gestiona modul de afişare a literelor pe display-urile cu 7 segmente, asigurând conversia codurilor binare ale caracterelor în semnalele specifice fiecărui segment, astfel încât literele să fie reprezentate corect și clar pe ecran.



CLK -> ceasul placii care functioneaza la o frecventa de 100MHz

LITERA -> mesaj-ul ce se v-a dispune pe 7-segment

RESET -> reseta componenta

ANODE_ACTIVATE -> activare display 7-segment

CATOD -> afisare pe 7-segment

4. Notatii

4.1 Semnale Interne

În ceea ce privește semnalele interne utilizate în proiect, pentru o mai bună înțelegere a denumirilor și funcțiilor acestora, mai jos este atașat un fragment din arhitectura **MainAdv**.

```
-- Semnale interne
signal clkmod : STD_LOGIC; -- ceasul modificat
signal anim1 : STD_LOGIC_VECTOR(55 downto 0); -- mesajul modificat de catre anim1
signal anim2 : STD_LOGIC_VECTOR(55 downto 0); -- mesajul modificat de catre anim2
signal anim3 : STD_LOGIC_VECTOR(55 downto 0); -- mesajul modificat de catre anim3
signal anim4 : STD_LOGIC_VECTOR(55 downto 0); -- mesajul modificat de catre anim4
signal animselect : STD_LOGIC_VECTOR(55 downto 0); -- animatia selectata
signal msgselect : STD_LOGIC_VECTOR(0 to 39); -- mesajul selectat
signal msgdcd : STD_LOGIC_VECTOR(55 downto 0); -- mesajul decodificat
```

În acesta poza, fiecare semnal declarat are un comentariu care descrie utilizarea sa în cadrul proiectului.

5. De ce aceasta metoda de rezolvare?

Deși numărul mare de componente poate părea copleșitor la prima vedere, modularitatea designului face ca algoritmi și funcționarea generală să fie mai ușor de urmărit.

Am ales această abordare pentru a rezolva problema deoarece am putut testa fiecare componentă separat, pentru a mă asigura că funcționează corect. Apoi, după ce fiecare componentă a fost proiectată și testată, le-am conectat între ele cu dificultate minimă.

6. Dezvoltări ulterioare

Pentru viitor, cea mai eficientă și utilă direcție de dezvoltare ar fi extinderea memoriei de litere pentru a cuprinde toate caracterele ce se pot afișa pe 7 segmente, cât și îmbunătățirea modului de selectare a mesajului, astfel încât utilizatorul să poată alege, cu ajutorul unor butoane, exact litera pe care dorește să o afișeze. Acest sistem ar înlocui introducerea bazată pe coduri binare, oferind o interacțiune mult mai intuitivă și accesibilă, chiar și pentru utilizatorii fără experiență tehnică.

Odata implementate aceste îmbunătățiri, se poate extinde posibilitatea de introducere a textului prin intermediul unor dispozitive periferice precum tastatura sau telefonul mobil. Un pas important în dezvoltarea din punct de vedere practic al acestui proiect ar fi schimbarea mediului de afișare, mesajele urmând a fi afișate pe panouri publicitare.

7. Bibliografie

<https://www.fypsolutions.com/fpga/fpga-vhdl-7-segment-multiplexing/>
<https://www.fpga4student.com/2017/09/vhdl-code-for-seven-segment-display.html>
https://en.wikipedia.org/wiki/Seven-segment_display