|  |  |
| --- | --- |
| Basys3 Artix-7 FPGA Board - RobotShop  Reclama cu animatii multiple | Indrumator  Diana Irena Pop  Darius - Razvan Radu  Student la UNIVERSITATEA TEHNICA DIN “CLUJ-NAPOCA”, An I, Grupa 5 |

Contents

[**Capitol 1: Specificatii proiect:** 2](#_Toc72171207)

[**1.1** **Scopul proiectului:** 2](#_Toc72171208)

[**1.2** **Considerente teoretice:** 3](#_Toc72171209)

[**1.3 Codificarea alfabetului pentru afisarea pe 7 segmente**: 4](#_Toc72171210)

[**1.4 Moduri de functionare (animatii posibile):** 6](#_Toc72171211)

[**Capitol 2: Proiectare** 7](#_Toc72171212)

[**2.1 Schema block** 7](#_Toc72171213)

[**2.2 Descompunere in U.C. si U.E.** 9](#_Toc72171214)

[**2.2.1 Continutul memoriilor ROM:** 10](#_Toc72171215)

[**2.2.2 Schema de detaliu UE:** 16](#_Toc72171216)

[**Capitol 3: Justificarea solutiei alese** 17](#_Toc72171217)

[**Capitol 4: Manualul de utilizare si intretinere** 18](#_Toc72171218)

[**Capitol 5: Posibilitati de dezvoltare ulterioara** 19](#_Toc72171219)

[**Bibliografie:** 20](#_Toc72171220)

# **Capitol 1: Specificatii proiect:**

## **Scopul proiectului:**

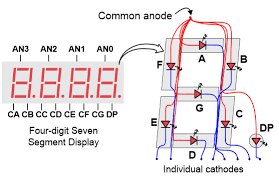
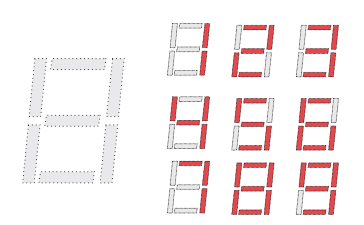
Vom proiecta o reclama publicitara cu animatii multiple. Se vor folosi afisoare pe 7 segmente. Textul de afişat va fi format din simboluri ale unui alfabet disponibil. Reclama va avea mai multe regimuri de funcţionare (minimum 4) ce vor putea fi selectate de către utilizator, de la comutatoarele plăcuţei cu FPGA. Se va folosi oscilatorul de cuarţ încorporat în plăcuţa cu FPGA (semnalul de clock respectiv va trebui desigur să fie divizat). Exemple de regimuri de funcţionare: „curgerea” scrisului de la dreapta spre stânga, pâlpâire, afişaj literă cu literă etc.

Deoarece pe un afişaj cu 7 segmente nu se pot reprezenta toate literele, se va crea un alfabet maximal şi mesajele vor fi compuse din simbolurile acelui alfabet. Mesajul va fi conţinut într-o memorie pentru a putea fi uşor de schimbat.

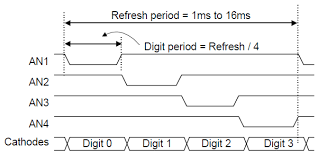
## **Considerente teoretice:**

Placa ce o avem la dispozitie contine doua afisoare cu 7-segmente, ce contin fiecare cate 4 anoade comune. Fiecare din cele 4 cifre e compusa din 7 segmente aratanjate in „forma 8”, cu cate un LED in fiecare segment, ce este responsabil cu iluminarea acelui segment din afisor.

Anodurile celor 7 LED-uri sunt grupate strans intr-un singur „anod comun”, dar catodurile LED-urilor raman separate. Semnalele anodurilor comune sunt disponibile ca semnale de intrare cu „digit enable” pe afișorul cu 4 cifre. Catodurile de segmente similare de pe toate cele patru afișaje sunt conectați în șapte noduri de circuit etichetate de la CA la CG (astfel, de exemplu, cele patru catoade „D” din cele patru cifre sunt grupați împreună într-un singur nod de circuit numit „CD”). Aceste șapte semnale catodice sunt disponibile ca intrări pe afișorul cu 4 cifre. Această schemă de conectare a semnalului creează un afișaj multiplexat, în care semnalele catodice sunt comune tuturor cifrelor, dar pot ilumina doar segmentele cifrei al căror semnal corespunzător anodic este afirmat.



Pentru ca fiecare din cele 4 cifre ale afisorului sa apara luminoasa si iluminata continuu, toate cele 4 cifre ar trebui sa aiba o perioada de refresh intre fiecare 1 - 16ms, la o frecventa de refresh intre 1KHz to 60Hz.



## **1.3 Codificarea alfabetului pentru afisarea pe 7 segmente**:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Digit** | **a** | **b** | **c** | **d** | **e** | **f** | **g** |  | **Digit** | **a** | **b** | **c** | **d** | **e** | **f** | **g** |
| A | **on** | **on** | **on** | off | **on** | **on** | **on** |  | N | **on** | **on** | **on** | off | **on** | **on** | off |
| B | **on** | **on** | **on** | **on** | **on** | **on** | **on** |  | O | **on** | **on** | **on** | **on** | **on** | **on** | off |
| C | **on** | off | off | **on** | **on** | **on** | off |  | P | **on** | **on** | off | off | **on** | **on** | **on** |
| D | **on** | **on** | **on** | **on** | **on** | off | off |  | Q | **on** | **on** | off | **on** | **on** | **on** | off |
| E | **on** | off | off | **on** | **on** | **on** | **on** |  | R | **on** | **on** | off | **on** | **on** | **on** | **on** |
| F | **on** | off | off | off | **on** | **on** | **on** |  | S | **on** | off | **on** | **on** | off | **on** | **on** |
| G | **on** | off | **on** | **on** | **on** | **on** | off |  | T | **on** | off | off | off | **on** | **on** | off |
| H | off | **on** | **on** | off | **on** | **on** | **on** |  | U | off | **on** | **on** | **on** | **on** | **on** | off |
| I | off | off | off | off | **on** | **on** | off |  | V | off | **on** | **on** | **on** | off | **on** | off |
| J | off | **on** | **on** | **on** | **on** | off | off |  | W | off | **on** | off | **on** | off | **on** | **on** |
| K | off | **on** | off | **on** | **on** | **on** | **on** |  | X | **on** | off | off | **on** | off | off | **on** |
| L | off | off | off | **on** | **on** | **on** | off |  | Y | off | **on** | **on** | **on** | off | **on** | **on** |
| M | **on** | off | **on** | off | **on** | off | **on** |  | Z | **on** | **on** | off | **on** | **on** | off | **on** |

Literele din alfabetul englez vor fi codificate pe 7 biti astfel, iar in imagine se poate vedea cum va arata fiecare litera afisata pe anod:

A = 0001000

B = 0000000

C = 0110001

D = 0000011

E = 0110000

F = 0111000

G = 0100001

H = 1001000

I = 1111001

J = 1000011

K = 1010000

L = 1110001

M = 0101010

N = 0001001

O = 0000001

P = 0011000

Q = 0010001

R = 0010000

S = 0100100

T = 0111001

U = 0000001

V = 1000101

W = 1010100

X = 0110110

Y = 1000100

Z = 0010010

## **1.4 Moduri de functionare (animatii posibile):**

Ca intrare de control pentru a selecta una din cele 4 moduri de afisare va fi intrarea SEL pe 3 biti, iar cuvantul ce va fi animat va fi UTCN:

000 – palpaire

001 – mersul cuvantului de la stanga la dreapta

010 – fiecare litera va aparea pe rand pana la formarea cuvantului

011 – se vor afisa toate literele cuvantului pe rand pe intregul afisor, apoi intreg cuvantul

100 – “curgere” de sus in jos

101 – “curgere” pe diagonala

# **Capitol 2: Proiectare**

## **2.1 Schema block**

Intrari:

Sel:

- care determina modul de afisare al reclamei (unul din cele 4 animatii)

- pe 3 biti, de tip SW

En:

- intrare de enable, cat timp este active aceasta intrare circuitul functioneaza.

- pe 1 bit (1 = active, 0 = inactiv), de tip SW.

Reset:

- pentru a reseta afisorul.

- pe 1 bit, de tip buton

CLK: - care este generat de generatorul cu cristale de cuart.

Iesiri:

Litera:

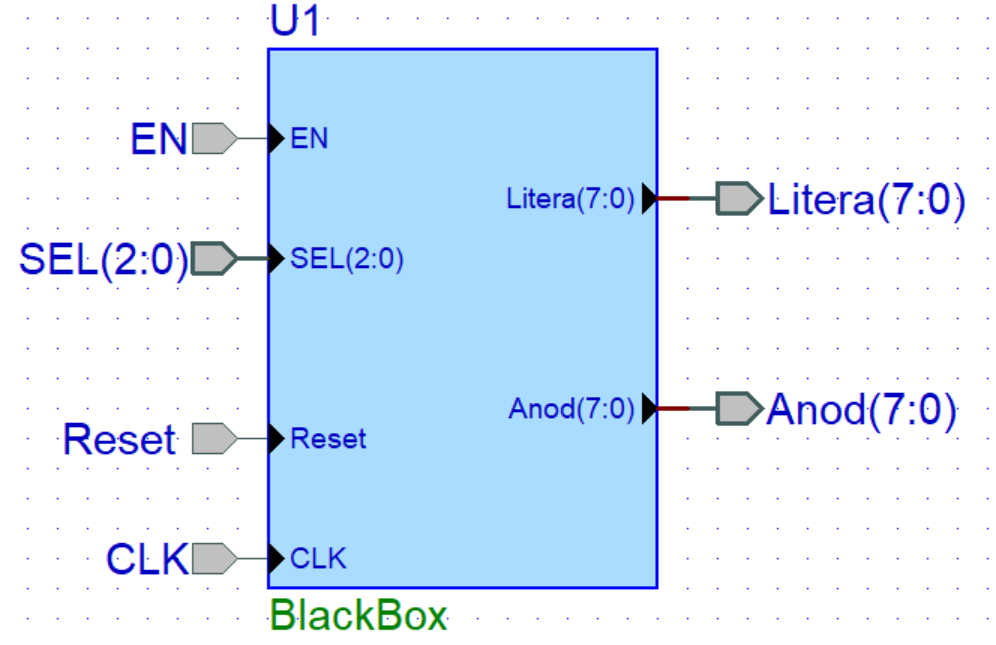
- pentru a afisa una dintre literele din cuvant-ul memorat

- pe 8 biti (avem la dispozitie un afisor pe 7 segm)

Anod:

- pentru a detecta pe care din cele 8 anoduri sa afisam litera

- pe 8 biti (sunt 8 posibile anoduri)



## **2.2 Descompunere in U.C. si U.E.**

**In Unitatea de Control (U.C.) vom avea:**

- un DMUX 1:8 => pentru a selecta animatia ce o dorim a fi afisata.

- va avea ca intrare En, pt En = 1 sistemul functioneaza, pentru En = 0 acesta nu va functiona.

- selectia va fi intrarea Sel, pe 3 biti pentru a selecta 1 din cele 5 animatii.

- un DMUX 1:8 => pentru a selecta anodul pe care se va afisa litera.

- un MUX 8:1 (pe 8 biti) => pentru a selecta litera ce va iesi pentru a fi afisata.

- ce va primi ca intrari, iesirile memoriilor ce contin animatiile

- ca selectii va primi Mod pentru a afisa litera din animatia selectata

- iesirea muxului va fi iesirea “Litera”, ce contine litera ce va fi afisata pe afisor

- un numarator ce va avea rol de divizor de frecventa.

Acesta va diviza frecventa CLK de pe placa de 100mhz

Vom folosi un numerator pe 27 de biti (pentru un refresh rate de 95hz si pentru un delay de 1,33s). Si va functiona astfel: in memorii avem animatiile impartite in mai multe grupuri de cate 8 adrese (acestea formeaza un cuvant afisat complet pe afisoare). Fiecare grup reprezinta un afisaj complet pe placa. Deci prin urmare ultimii 3 biti ai adresei vor fi pentru determinarea carui caracter sa il trimitem la afisoare si totodata pentru a determina pe care anod va fi afisat. Acesti 3 biti de adresa vor fi conectati la iesirile Q18 Q19 Q20 ale numaratorului pentru a avea un refresh rate de aproximativ 95hz. Iar pentru a avea un delay de aproximativ 1,33 s intre afisarile complete ale animatiilor vom conecta ultimii biti (3, 2 sau 1 depinde de animatie) de adresa cu iesirile Q24 Q25 Q26 (depinde de caz) ale numaratorului.

**In Unitatea de Executie (U.E.) vom avea:**

- 5 memorii ce vor contine animatiile ce urmeaza a fi afisate.

- 1 memorii ROM 16x8

- 3 memorii ROM 32x8

- 2 memorii ROM 64x8

### **2.2.1 Continutul memoriilor ROM:**

Pentru MOD ul de functionare 000, vom stoca animatia “palpaire” intr-o memorie ROM 16x8 ce va avea harta de mai jos. Caracterele continute in memorie vor fi codificate pe 7 biti in fucntie de alfabetul ales.

|  |  |
| --- | --- |
| ADRESA | CONTINUT |
| 0 | U |
| 1 | T |
| 2 | C |
| 3 | N |
| 4 | - |
| 5 | - |
| 6 | - |
| 7 | - |
| 8 | - |
| 9 | - |
| 10 | - |
| 11 | - |
| 12 | - |
| 13 | - |
| 14 | - |
| 15 | - |

Pentru MOD ul de functionare 001, vom stoca animatia “mersul cuvantului de la stanga la dreapta” intr-o memorie ROM 64x8 ce va avea harta de mai jos. Caracterele continute in memorie vor fi codificate pe 7 biti in fucntie de alfabetul ales.

|  |  |  |  |
| --- | --- | --- | --- |
| ADRESA | CONTINUT | ADRESA | CONTINUT |
| 0 | U | 32 | - |
| 1 | T | 33 | - |
| 2 | C | 34 | - |
| 3 | N | 35 | - |
| 4 | - | 36 | U |
| 5 | - | 37 | T |
| 6 | - | 38 | C |
| 7 | - | 39 | N |
| 8 | - | 40 | N |
| 9 | U | 41 | - |
| 10 | T | 42 | - |
| 11 | C | 43 | - |
| 12 | N | 44 | - |
| 13 | - | 45 | U |
| 14 | - | 46 | T |
| 15 | - | 47 | C |
| 16 | - | 48 | C |
| 17 | - | 49 | N |
| 18 | U | 50 | - |
| 19 | T | 51 | - |
| 20 | C | 52 | - |
| 21 | N | 53 | - |
| 22 | - | 54 | U |
| 23 | - | 55 | T |
| 24 | - | 56 | T |
| 25 | - | 57 | C |
| 26 | - | 58 | N |
| 27 | U | 59 | - |
| 28 | T | 60 | - |
| 29 | C | 61 | - |
| 30 | N | 62 | - |
| 31 | - | 63 | U |

Pentru MOD ul de functionare 010, vom stoca animatia “fiecare litera va aparea pe rand pana la formarea cuvantului” intr-o memorie ROM 32x8 ce va avea harta de mai jos. Caracterele continute in memorie vor fi codificate pe 7 biti in fucntie de alfabetul ales.

|  |  |
| --- | --- |
| ADRESA | CONTINUT |
| 0 | U |
| 1 | - |
| 2 | - |
| 3 | - |
| 4 | - |
| 5 | - |
| 6 | - |
| 7 | - |
| 8 | U |
| 9 | T |
| 10 | - |
| 11 | - |
| 12 | - |
| 13 | - |
| 14 | - |
| 15 | - |
| 16 | U |
| 17 | T |
| 18 | C |
| 19 | - |
| 20 | - |
| 21 | - |
| 22 | - |
| 23 | - |
| 24 | U |
| 25 | T |
| 26 | C |
| 27 | N |
| 28 | - |
| 29 | - |
| 30 | - |
| 31 | - |

Pentru MOD ul de functionare 011, vom stoca animatia “se vor afisa toate literele cuvantului pe rand pe intregul afisor, apoi intreg cuvantul” intr-o memorie ROM 32x8 ce va avea harta de mai jos. Caracterele continute in memorie vor fi codificate pe 7 biti in fucntie de alfabetul ales.

|  |  |
| --- | --- |
| ADRESA | CONTINUT |
| 0 | U |
| 1 | U |
| 2 | U |
| 3 | U |
| 4 | U |
| 5 | U |
| 6 | U |
| 7 | U |
| 8 | T |
| 9 | T |
| 10 | T |
| 11 | T |
| 12 | T |
| 13 | T |
| 14 | T |
| 15 | T |
| 16 | C |
| 17 | C |
| 18 | C |
| 19 | C |
| 20 | C |
| 21 | C |
| 22 | C |
| 23 | C |
| 24 | N |
| 25 | N |
| 26 | N |
| 27 | N |
| 28 | N |
| 29 | N |
| 30 | N |
| 31 | N |

Pentru MOD ul de functionare 100, vom stoca animatia “curge de sus in jos” intr-o memorie ROM 32x8 ce va avea harta de mai jos. Caracterele continute in memorie vor fi codificate pe 7 biti in fucntie de alfabetul ales.

Se va afisa pe 5 nivele: 1 – catozii A, 2 – catozii F,B, 3 – catozii G, 4 – catozii C, E, 5 – catozii D

|  |  |
| --- | --- |
| ADRESA | CONTINUT |
| 0 | 1111111 |
| 1 | 0111111 |
| 2 | 0111111 |
| 3 | 0111111 |
| 4 | - |
| 5 | - |
| 6 | - |
| 7 | - |
| 8 | 1011101 |
| 9 | 0111101 |
| 10 | 0111101 |
| 11 | 0011101 |
| 12 | - |
| 13 | - |
| 14 | - |
| 15 | - |
| 16 | 1001001 |
| 17 | 0111001 |
| 18 | 0111001 |
| 19 | 0001001 |
| 20 | - |
| 21 | - |
| 22 | - |
| 23 | - |
| 24 | U |
| 25 | T |
| 26 | C |
| 27 | N |
| 28 | - |
| 29 | - |
| 30 | - |
| 31 | - |

Pentru MOD ul de functionare 101, vom stoca animatia “curgere pe diagonala” intr-o memorie ROM 64x8 ce va avea harta de mai jos. Caracterele continute in memorie vor fi codificate pe 7 biti in fucntie de alfabetul ales.

Se va realiza pe 5 nivele: 1- catozii A,F,B(prim afisor), 2 – prima litera si catozii A,F,B(al doilea afisor), 3 – primele 2 litere si catozii A,F,B(3 afisor), 4 – primele 3 litere si catozii A,F,B(4 afisor) si 5 – toate cele 4 litere (cuvant complet)

|  |  |  |  |
| --- | --- | --- | --- |
| ADRESA | CONTINUT | ADRESA | CONTINUT |
| 0 | 1011101 | 32 | U |
| 1 | - | 33 | T |
| 2 | - | 34 | C |
| 3 | - | 35 | N |
| 4 | - | 36 | - |
| 5 | - | 37 | - |
| 6 | - | 38 | - |
| 7 | - | 39 | - |
| 8 | U | 40 | - |
| 9 | 0111101 | 41 | - |
| 10 | - | 42 | - |
| 11 | - | 43 | - |
| 12 | - | 44 | - |
| 13 | - | 45 | - |
| 14 | - | 46 | - |
| 15 | - | 47 | - |
| 16 | U | 48 | - |
| 17 | T | 49 | - |
| 18 | 0111101 | 50 | - |
| 19 | - | 51 | - |
| 20 | - | 52 | - |
| 21 | - | 53 | - |
| 22 | - | 54 | - |
| 23 | - | 55 | - |
| 24 | U | 56 | - |
| 25 | T | 57 | - |
| 26 | C | 58 | - |
| 27 | 1011101 | 59 | - |
| 28 | - | 60 | - |
| 29 | - | 61 | - |
| 30 | - | 62 | - |
| 31 | - | 63 | - |

### **2.2.2 Schema de detaliu UE:**

# **Capitol 3: Justificarea solutiei alese**

Pentru realizarea proiectului am incercat sa fiu cat mai sugestiv posibil in denumirea semnalelor cat si a tipurilor definite. De asemenea am optat pentru o solutie cat mai modulara pentru a-mi fi usor de testat fiecare componenta in caz de erori. Componentele folosite sunt generice, MUX, DEMUX, NUMARATOR, MEMORII etc. Si acest lucru il face usor de inteles.

Am ales o implementare cu un singur numarator pentru ca mi s-a parut mai eficienta decat definirea a doua numaratoare pentru unele animatii, chiar daca initial a fost mai complicat alesul iesirilor cu frecventele potrivite.

De asemenea implementarea cu memorii este destul de usoara de realizat, dar in acelasi timp as fi putut opta pentru o solutie mai eficienta, ce nu necesita memorie inutil. (unele memorii nu au nimic la multe adrese).

In concluzie am ales o solutie ce mi s-a parut optima si usor de implementat, si am incercat sa scriu codul astfel incat sa fie usor de citit si de inteles.

# **Capitol 4: Manualul de utilizare si intretinere**

Pentru inceput, pentru a porni orice fel de reclama utilizatorul trebuie sa activeze primul switch din partea dreapta a placii.

Apoi va activa/dezactiva primele 3 switch uri din partea stanga pentru a porni animatiile dupa cum urmeaza:

Daca cele trei switch uri sunt pornite simultan animatia va fi clipire

Daca se activeaza doar al treilea switch animatia va fi mersul cuvantului de la stanga la dreapta

Daca se activeaza doar switch ul 2 fiecare litera va aparea pe rand pana la formarea cuvantului

Daca se activeaza simultan si switch ul 2 si 3, se va porni animatia in care fiecare litera clipeste pe rand.

Daca este activat doar switch ul 1, va aparea animatia cu curgerea de sus in jos.

Si in final daca Switch urile 1 si 3 sunt activate simultan se va porni animatia “curgere pe diagonala”.

Tinem sa specificam ca daca se activeaza toate 3 switch urile sau doar switch urile 1 si 2 simultan, nu va aparea nicio animatie, ecranul ramanand gol.

# **Capitol 5: Posibilitati de dezvoltare ulterioara**

Exista cateva modalitati de imbunatatire atat a experientei utilizatorului cat si a functionalitatii.

Pentru inceput, putem opta pentru o varianta in care fiecare animatie este pusa pe cate un switch pentru a fi mai comod la utilizare.

De asemenea putem adauga mult mai multe animatii sau cuvinte de lungimi mai mari(sau mai complexe, expresii din mai multe cuvinte, numere, caractere speciale etc).

Din punct de vedere al proiectarii am optat pentru o implementare cu un singur numerator (atat ca divizor cat si ca numerator de “scene”), dar am folosit multa memorie inutil. Se poate imbunatati prin folosirea mai optima a memoriei.

# **Bibliografie:**

1. Suport de curs PL
2. Suport de curs PSN
3. Documentatia pusa de dispozitie pentru intelegerea functionarii placutelor FPGA si a afisoarelor cu 7 segmente.
4. Pozele de pe google images