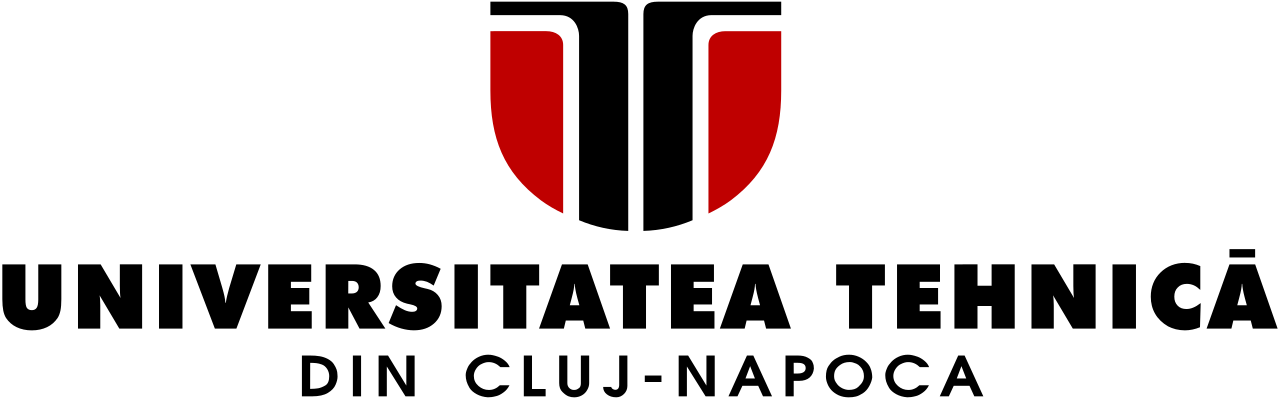
***Universitatea Tehnica din Cluj-Napoca***

***Facultatea de Automatica si Calculatoare***

***Calculatoare si Tehnologia Informatiei***



***Proiect: Controller VGA***

***Studenti: Linaru Petra Rozalia***

***Mudura Ana-Andreea***

***Grupa: 30211***

***Prof.coordonator: Pop Diana Irena***

**CUPRINS**

**1.Specificatia proiectului 2**

1.1 Intrari si iesiri………………………………………………………………………. 3

1.2 Resure………………………………………………………………………………. 4

1.3. Codificarea intrarilor………………………………………………………………. 4

**2.Diagrama block 5**

**3.Unitatea de control si unitatea de executie 6**

3.1 Unitatea de control…………………………………………………………………... 7

3.2 Unitatea de executie………………………………………………………………….. 8

3.3 Schema de detaliu……………………………………………………………………. 9

**4.Justificarea solutiei 14**

**5.Manual de utlizare 15**

**6.Posibilitati de imbunatatire 17**

**CONTROLLER VGA**

**1.Specificatia proiectului**

VGA este acronim pentru Video Graphics Array și este, probabil, cel mai vechi tip de conector pe care încă îl putem întâlni. Acest sistem asigură o metodă simplă de a conecta un sistem prevăzut cu un monitor pentru a afișa diferite informații și imagini.

Controller-ul VGA reprezintă interfața standard de control a monitoarelor ce funcționează în sistem  analog. Totodată, producția industrială a mașinăriilor trebuie să fie extrem de flexibilă pentru a putea îndeplini diferite solicitări. În acest sens, controller-ul VGA, care este un circuit logic care poate controla interfața VGA, poate fi realizat cu ușurintă cu ajutorul tehnologiei FPGA și prezintă o mare flexibilitate.

Interfața are ca și component principal un ceas (Pixel\_Clock) care își va diviza frecvența în patru.Astfel vom sincroniza “liniile” verticale, cele orizontale, precum si semnalele backporch si frontporch cu acest semnal.

Monitorul dispune de 640 de coloane si 480 de rânduri de elemente numite pixeli.Totodată, acesta este controlat de cinci semnale : cele 3 culori principale ce alcătuiesc un pixel(roșu,galben și verde) și pixelii orizontali respectiv verticali ce alcătuiesc cele două axe în reprezentarea imaginii pe ecran. Semnalul de sincronizare verticală (VSync) definește frecvența la care informația este reluată. Semnalul de sincronizare orizontală (Hsync) definește numărul de linii orizontale care sunt afișate la o rată de refresh.

O imagine este afișată pe ecran prin activarea, respectiv dezactivarea elementelor numite pixeli. Monitorul scanează imaginea pe axa orizontală, până traversează intreaga lungime, după care urmează o scurtă perioada de timp în care acesta traversează lungimea înapoi, revine in poziția inițiala și trece pe următorul rând. Acest proces este unul rapid datorită frecvenței cu care are loc fiecare scanare, motiv pentru care noi avem impresia că toți pixelii sunt activați.

Pentru a obține o rezoluție de 640x480 , avem nevoie de un pixel\_clock cu frecvența 25.175 MHz și 60Hz frecvența de refresh.

*Coordonate pe axa orizontală*:

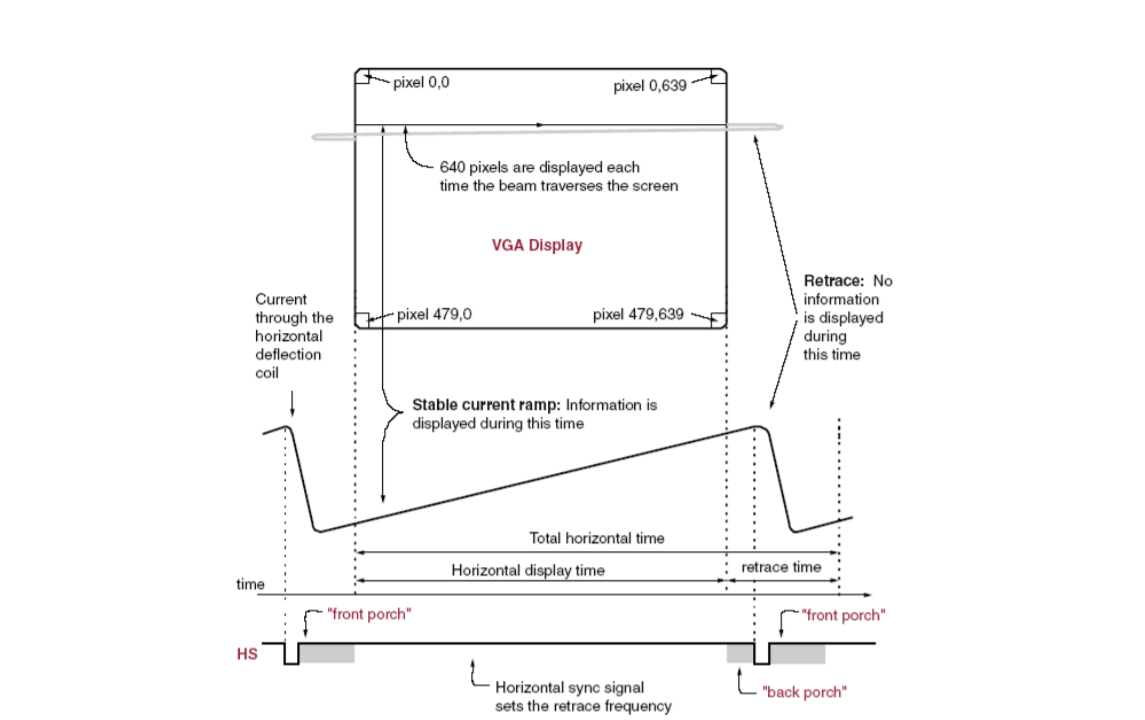
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REZOLUTIE | H\_PIXELI\_ACTIVI | H\_FRONTPORCH | H\_BACKPORCH | H\_SYNC | H\_TOTAL |
| 640x480 | 640 | 16 | 48 | 96 | 800 |

Scurtă descriere : trebuie să obținem un semnal de sincronizare, care să numere în intervalul 0-799. (un numărator pe orizontală)

*Coordonate pe axa verticală*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REZOLUTIE | V\_PIXELI\_ACTIVI | V\_FRONTPORCH | V\_BACKPORCH | V\_SYNC | V\_TOTAL |
| 640x480 | 480 | 10 | 33 | 2 | 525 |

Scurtă descriere : trebuie să obținem un semnal de sincronizare, care să numere în intervalul 0-524. (un numărător pe verticală)



* 1. **Intrări si iesiri**

Intrari :

* *Pixel Clock*
* *Selectia culorii* RGB pe 3 biti biti(ofera utilizatorului posibilitatea de a alege culoarea dorita in timp real)-**3 SWITCHES**
* *Selectia formei* reprezentata pe 2 biti(ofera utilizatorului posibilitatea de a accesa adresa formei dorite din memorie)-**2 SWITCHES**
* *Reset* pe 1 bit**-SWICTH**
* *Enable Move* pe 1 bit-**1 SWITCH(**definim domeniul de miscare)
* *CONTROL\_Debouncers-***4 BUTOANE**(controleaza miscarea imaginii-numara cate cicluri de impulsuri de tact trebuie apasate butoanele)

Iesiri :

* VGA\_RED-4 bit
* VGA\_GREEN-4 bit RGB pe 12 biti
* VGA\_BLUE-4 bit
* H\_SYNC-1 bit
* V\_SYNC-1 bit

**1.2 Resurse**

*Unitatea de control*

Circuit pentru selectia formei si a culorii formei

* *Circuit decodificare culoare* (CLC particular)
* *Memorie Forma* (circuit de selectie a formei)

*Unitatea de executie*

* *Divizor frecventa1*-imparte frecventa impulusului de tact(*numarator modulo 4*)pentru a genera semnalele de VSYNC si HSYNC cu ajutorul numaratoarelor
* *Divizor frecventa*2-imparte frecventa si ajuta la miscarea lina a imaginii pe display

Circuit pentru generarea semnalelor de sincronizare a randurilor si coloanelor

* *Un numarator modulo 799* pentru a obtine semnalul de sincronizare pe orizontala(**H\_SYNC**)
* *Un numarator modulo 524* pentru a obtine semnalul de sincronizate pe verticala(**V\_SYNC**)

Circuit care genereaza pozitia exacta unde se afla controllerul in colorarea matricii( se testeaza daca **H\_SYNC/V\_SYNC** indeplinesc anumite *conditii-*counter de rezolutie)

-zona activa(in care putem afisa) pentru H\_SYNC=1 si V\_SYNC=1

-zona de blanking vertical pentru H\_SYNC=1 si V\_SYNC=0

-zona de blanking orizontal pentru H\_SYNC=0 si V\_SYNC=1

-zona de blanking total pentru H\_SYNC=0 si V\_SYNC=0

Circuit care identifica pe baza unor *conditii* daca pixelul trebuie colorat sau blank(Display\_S)(**daca pixelul curent se afla in frameul formei sau nu**)

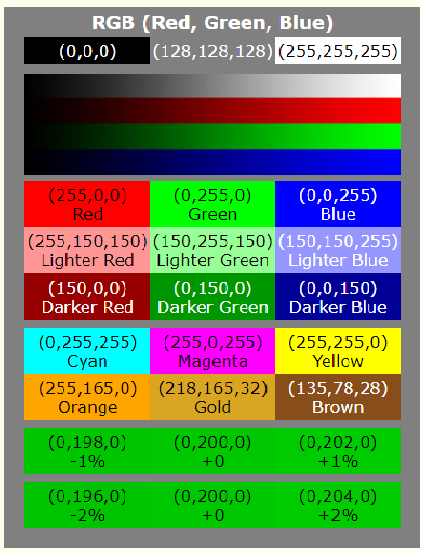
Circuit care verifica conditiile pentru fiecare forma in parte si stabileste daca pixelul este colorat sau nu(Pattern Generator)

Circuit pentru miscarea imaginii pe display(CONTROL\_MOVE)(**in functie de semnalul de *EN\_MOVE, CONTROL\_DEBOUNCERS* si forma generata)**

**1.3 Codificarea intrarilor**

*Codificarea culorilor*

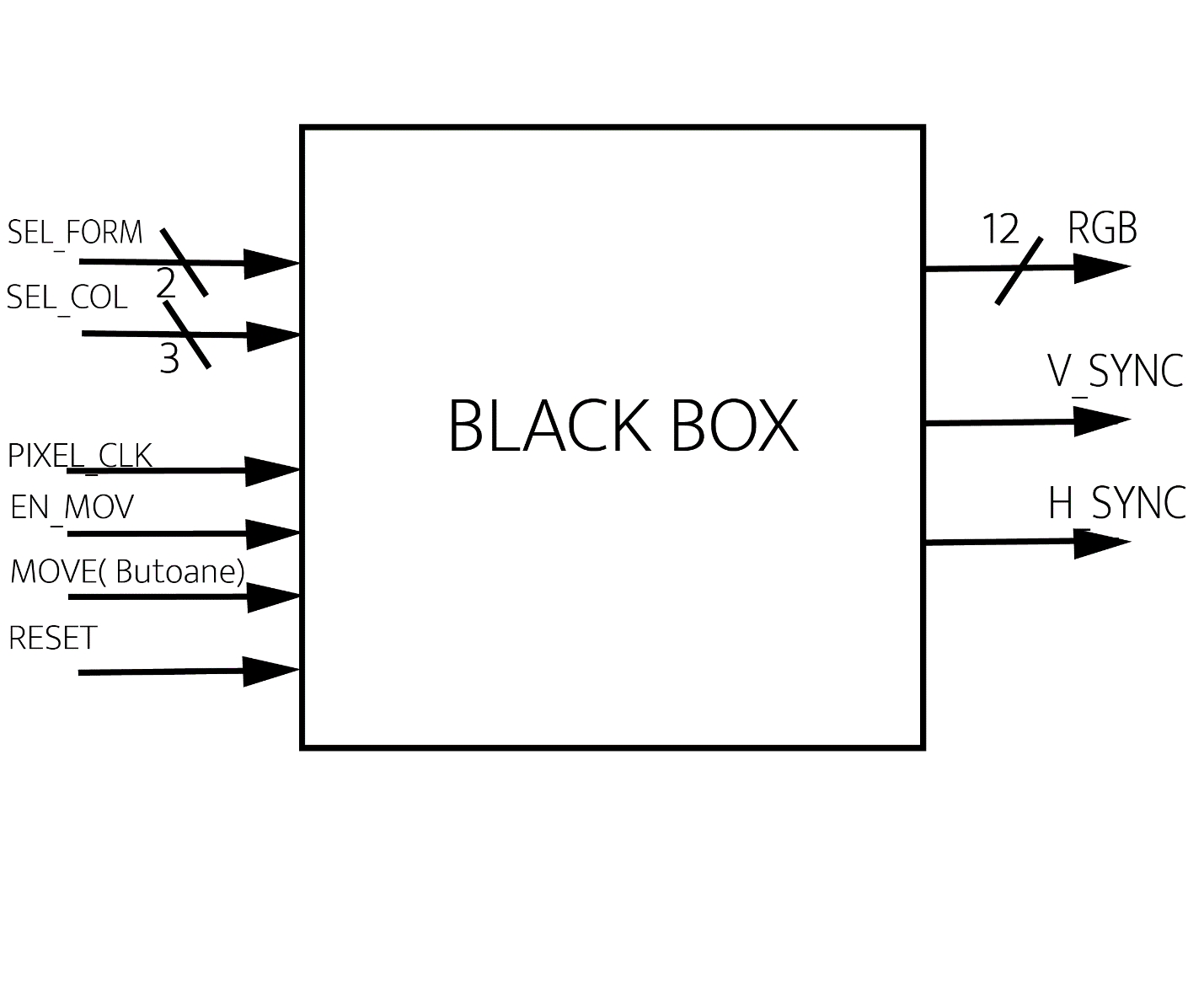
|  |  |  |  |
| --- | --- | --- | --- |
| VGA\_RED | VGA\_GREEN | VGA\_BLUE | CULOARE |
| 1 | 1 | 1 | ALB |
| 1 | 1 | 0 | ROZ |
| 1 | 0 | 1 | VERDE INCHIS |
| 1 | 0 | 0 | ROSU |
| 0 | 1 | 1 | GALBEN |
| 0 | 1 | 0 | VERDE |
| 0 | 0 | 1 | MOV |
| 0 | 0 | 0 | NEGRU |



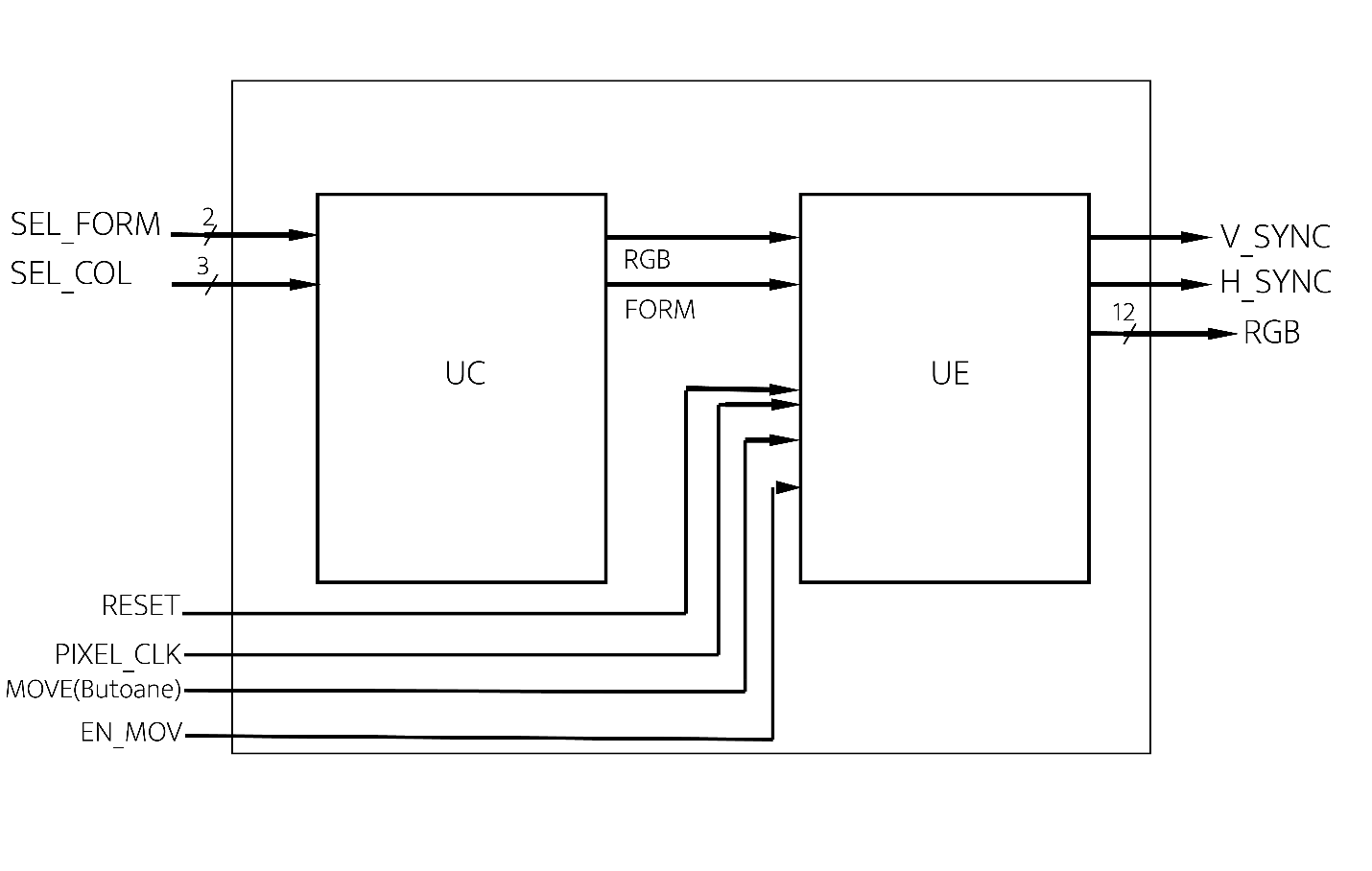
*Codificarea selectiei formelor*

|  |  |  |
| --- | --- | --- |
| SEL\_FORM1 | SEL\_FORM0 | FORMA |
| 0 | 0 | PATRAT |
| 0 | 1 | DREPTUNGHI |
| 1 | 0 | DUNGI VERTICALE |
| 1 | 1 | BROASCA CU BRETON |

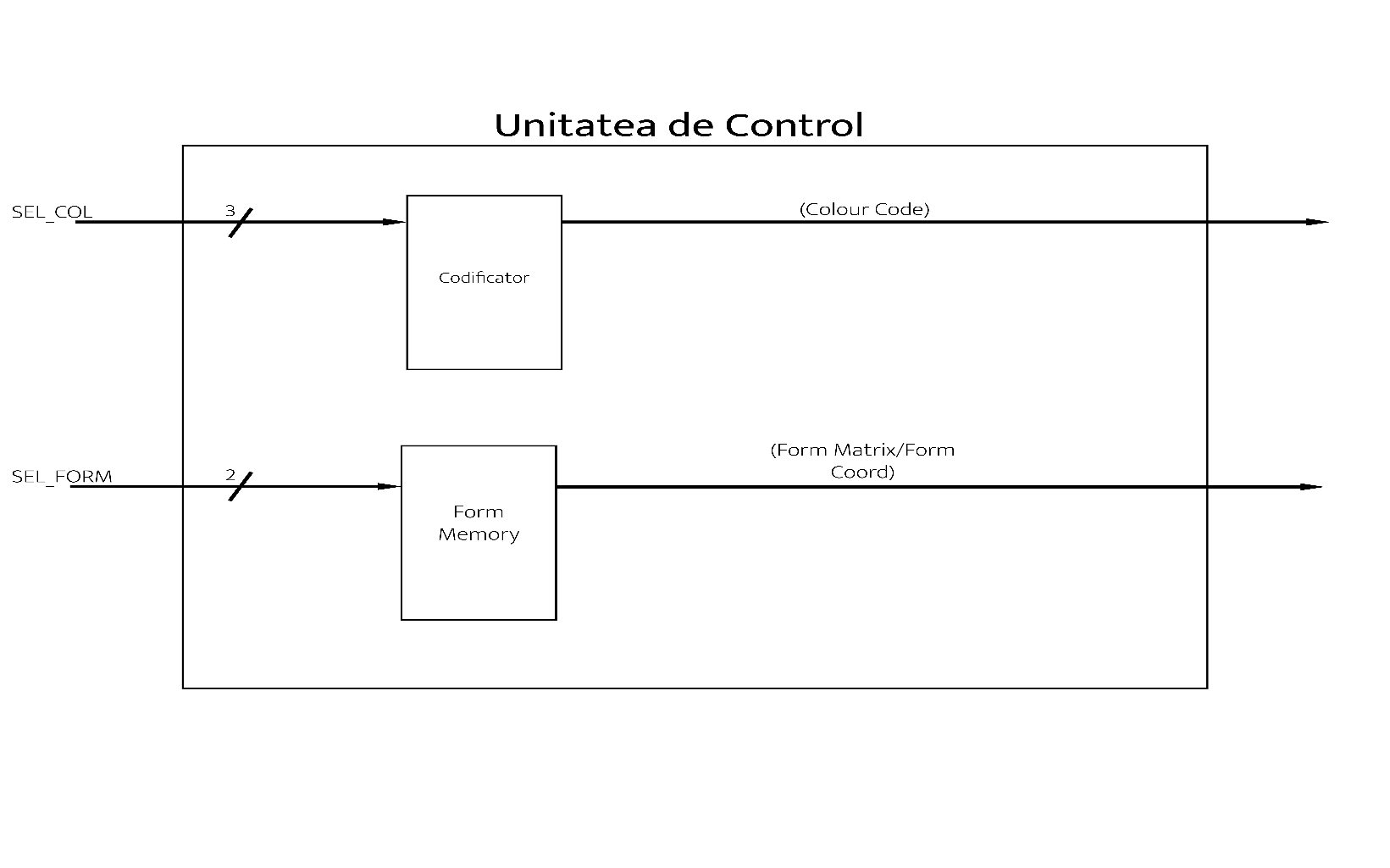
**2.Diagrama Block**

****

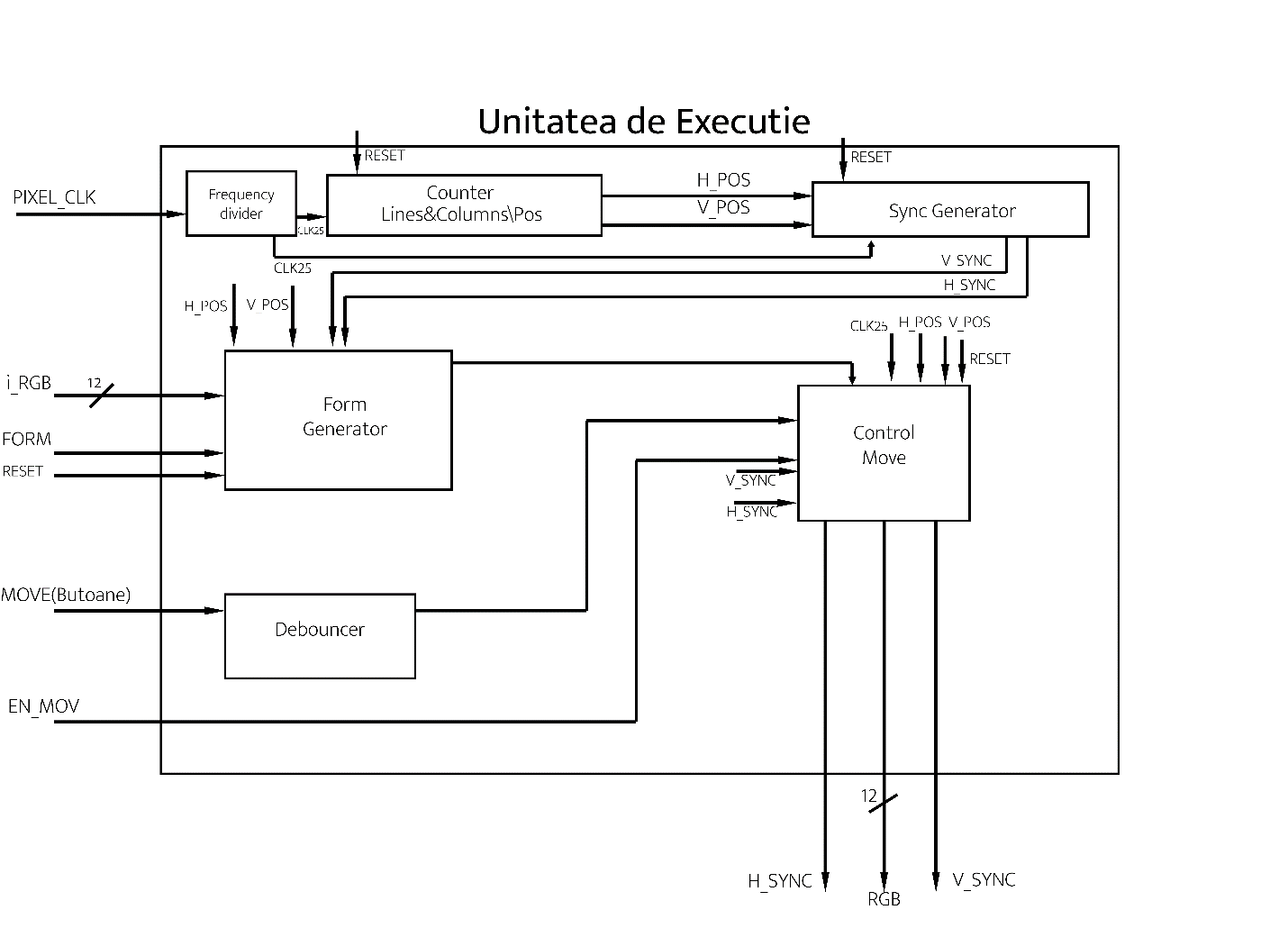
**3.Unitatea de control si unitatea de executie**

****

**3.1 Unitatea de control**



**3.2.Unitatea de executie**



FREQUENCY DIVIDER

DISPLAYS

**3.3.Schema de detaliu**

*1.Decodificator*

Intrari:SEL\_COL

Iesiri: i\_VGA\_R

i\_VGA\_G

i\_VGA\_B

daca(SEL\_COL="000")atunci daca(SEL\_COL="101")atunci

i\_VGA\_R<="0000"; i\_VGA\_R<="0001";

i\_VGA\_G<="0000"; i\_VGA\_G<="0110";

i\_VGA\_B<="0000"; i\_VGA\_B<="0100";

daca(SEL\_COL="001")atunci daca(SEL\_COL="111")atunci

i\_VGA\_R<="1001"; i\_VGA\_R<="1111";

i\_VGA\_G<="0011"; i\_VGA\_G<="1111";

i\_VGA\_B<="1011"; i\_VGA\_B<="1111";

daca(SEL\_COL="010")atunci sfarsit daca

i\_VGA\_R<="0001";

i\_VGA\_G<="1110";

i\_VGA\_B<="0101";

daca(SEL\_COL="100")atunci

i\_VGA\_R<="1000";

i\_VGA\_G<="0100";

i\_VGA\_B<="0000";

daca(SEL\_COL="110")atunci

i\_VGA\_R<="1011";

i\_VGA\_G<="0000";

i\_VGA\_B<="1000";

*2.Memorie*

Intrari: J(formele codificate)

Iesire: F(coordonatele formei)

Daca I=00 atunci

Altfel daca I=01 atunci

Altfel daca I=11 atunci

Altfel daca I=10 atunci

*3.Form Generator*

Intrari: F(coordonatele formei)

POS (pozitia curenta a pixelilor)

Iesire : P(se coloreaza coloreaza forma pizelului sau nu)

Daca forma selectata e **smiley face patrat/dungi orizontale/patrat/tabla de sah** atunci

Daca POS apartine intervalului coresponzator formei atunci

P=RGB selectat de la placa

Altfel

P=RGB corespunzator culorii negre

Sfarsit daca

*4.Bistabil D si Debouncer*

Intrari: D,DEB(indica numarul de cicluri parcurse care controleaza bistabilul)

Iesire: MOVE(activeaza circuitul de control al miscarii)

Daca DEB nu este activ atunci MOVE=D

Altfel MOVE=0 ;

Sfarsit daca

Atlfel MOVE=0 ;

*5.Control move*

Intrari: POS

P

EN\_MOVE

MOVE

Iesiri-(imaginea miscata sau nu)

Daca EN\_MOVE=0 atunci

MOVE=0

Coordonatele imaginii nu se schimba(forma e statica)

Altfel

Daca DEB este inactiv atunci

MOVE=1

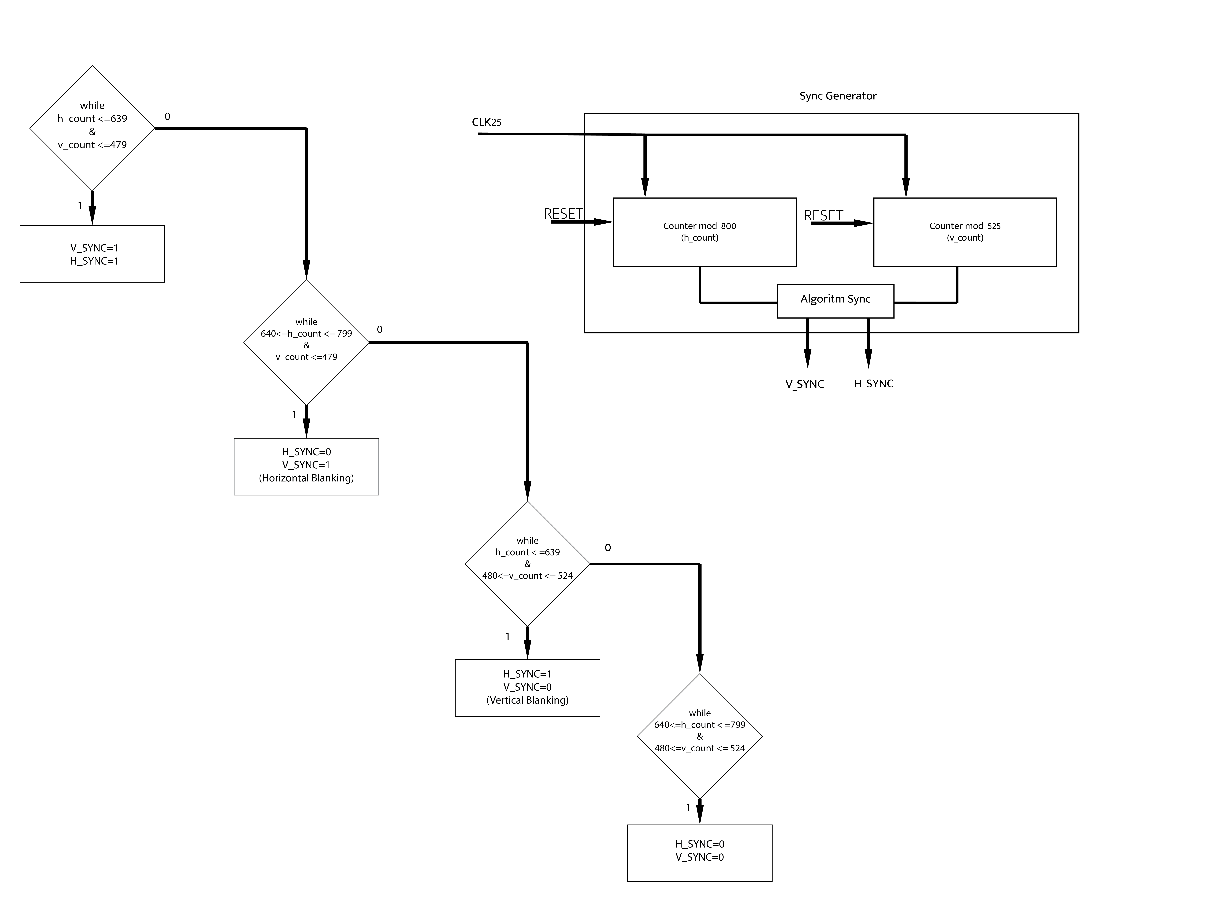
Coordonatele imaginii se schimba(forma se misca pe display)

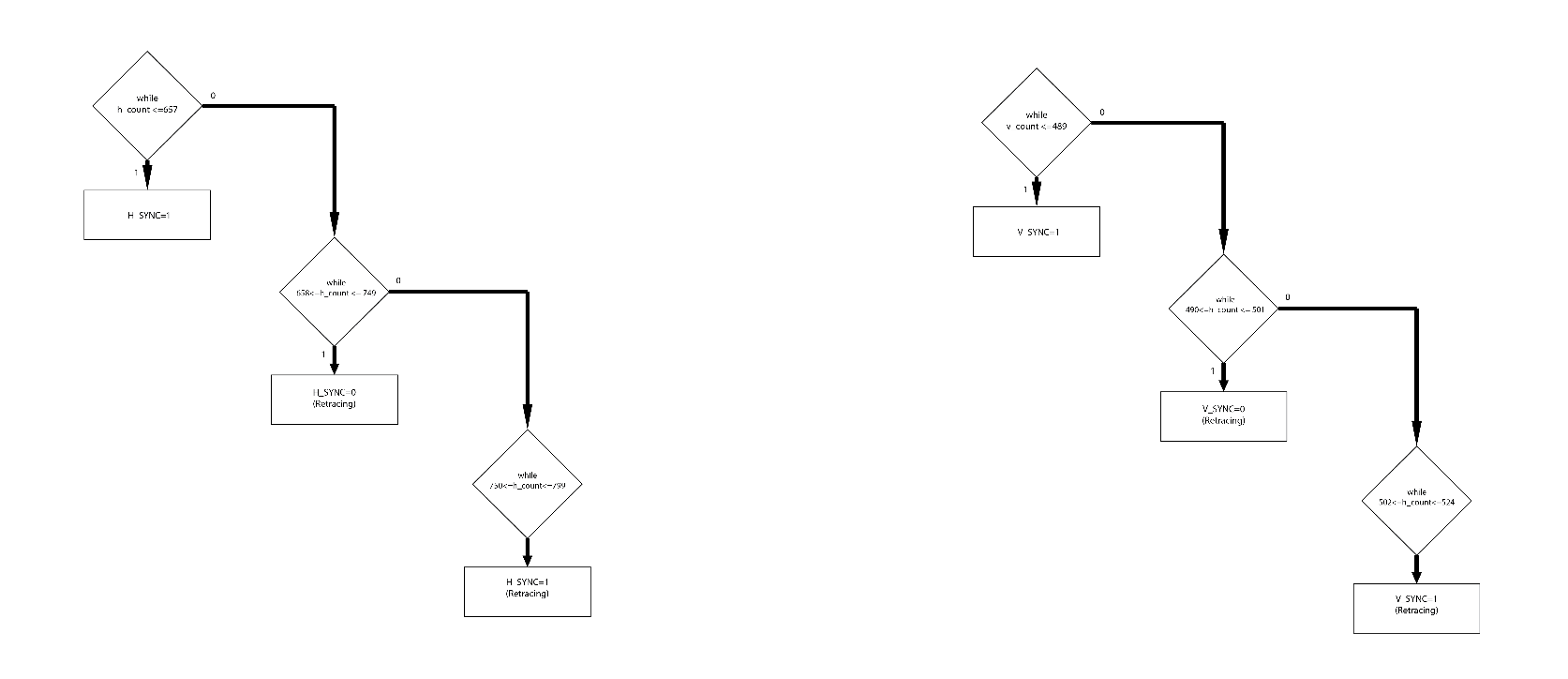
Altfel

Coordonatele imaginii nu se schimba(forma e statica)

Sfarsit daca

Sfarsit daca

 *6.Sync Genereator*



**4.Justificarea solutiei**

# Solutia aleasa reprezinta implementarea unui controller VGA pe placa Basys 3 Artix-7 FPGA de la Digilent.

# Circuitul presupune alegerea unei forme,apoi culoarea ei,ambele decizii sunt puse la dispozitie de la intrerupatoarele placii, designul ales urmand sa fie afisata pe ecran. Posibilitatea de a alege forma si culoarea folosind direct interfata placii a fost aleasa deoarece utilizatorul prin aceasta modalitate poate schimba forma si culoarea in timp real.

# Culorile sunt codificate pe 3 biti,ele urmand sa fie decodificate in rosu,verde si albastru,pentru a reduce complexitatea algoritmului si a volumului datelor de intrare. Aceelasi principiu este aplicat si la selectia formelor,care sunt codificate pe 2 biti care reprezinta adresa unei celule de memorie.

# Divizorul de frecventa este folosit pentru a genera impulsurile de tact pentru cele doua numaratoare ce genereaza semnalele de sincronizare(H\_SYNC si V\_SYNC). Iesirile celor doua numaratoare reprezinta intrarile unui circuit de testare a rezolutiei, unde ele sunt verificate daca indeplinesc anumite conditii. Numaratorul a fost ales pentru a obtine pozitia exacta a pixelului pe ecran.

# Pentru a genera forma a fost ales un circuit care testeaza daca pixelul trebuie sa fie colorat cu culoarea aleasa de la placa sau cu culoarea de background(in proiect este implicit culoarea neagra). Imaginea generata intra intr-un circuit care pune la dispozitie utilizatorului alegerea de a misca forma pe ecran sau nu prin intermediul unor 2 butoane(miscare dreapta-stanga). Cele doua circuite au fost implementate in cadrul proiectului tot deoarece trebuie oferita posibilitatea de alege forma,culoarea ei si statusul de miscare in timp real de la butoanele placii.

# Pentru a controla ciclurile de ceas prin care trec butoanele cand sunt folosite a fost aleasa solutia unor debouncere. Numaratorul are ca si scop parcurgerea unor bucle de numarare, sfarsitul lor semnaland aducerea butonului in starea initiala pentru refolosire.

# Iesirile circuitului de controlare a miscarii sunt afisate pe ecran (forma culorata,statica sau nu).

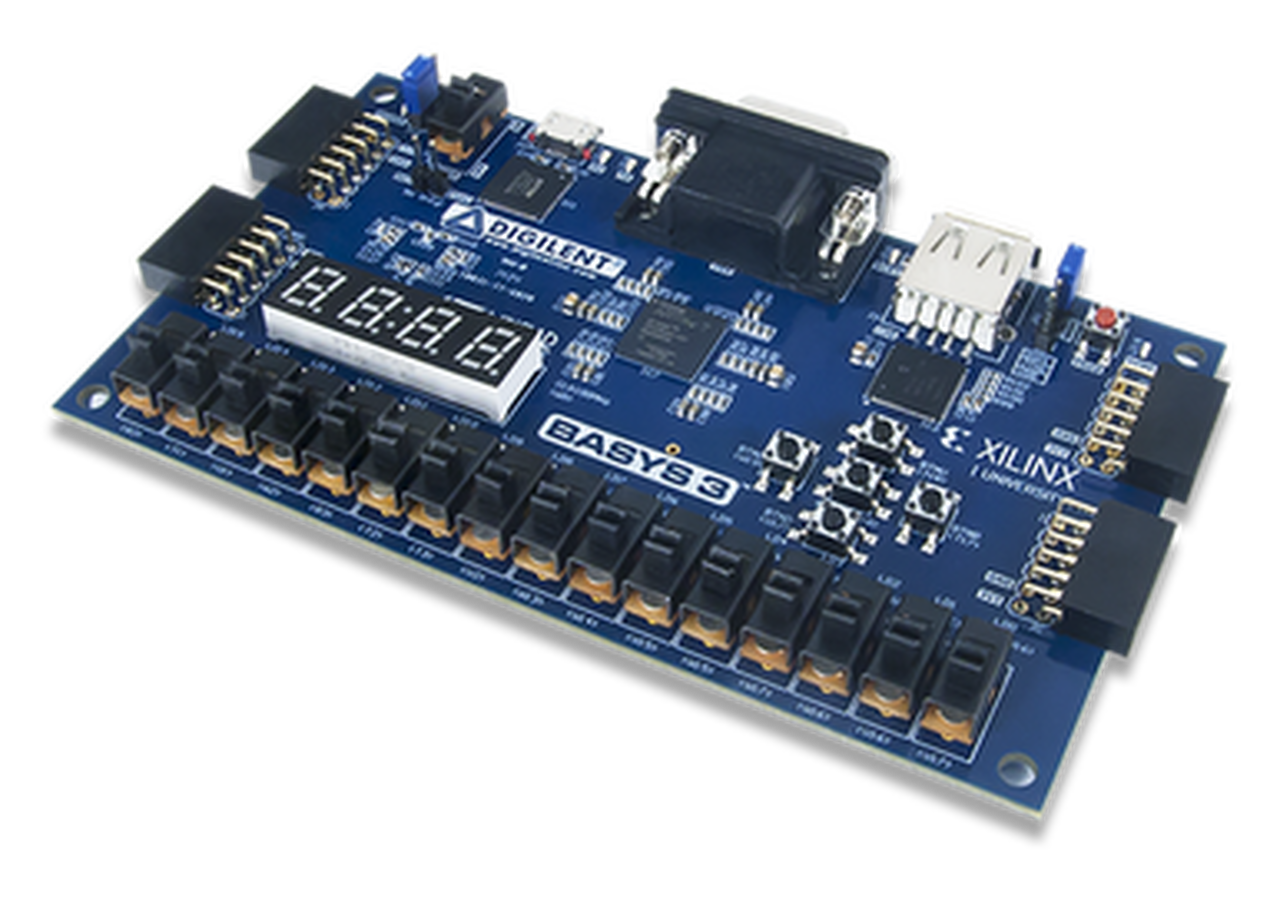
**5.Manual de utilizare**

Controllerul VGA este implementat pe placa Basys 3 Artix-7 FPGA. Placa dispune de 16 switchuri si 5 butoane cu ajutorul carora se poate controla ecranul.

Utilizatorul trebuie sa aiba la dispozitie codificarile culorilor si a formelor pentru a utiliza controllerul. Codificarile reprezinta numere reprezentate in baza 2( 0 sau 1),fiecarui numar ii este asociata o culoare/forma (*tabelele de codificatii pot fi gasite in subcapitolul* **1.3 Codificarea intrarilor** *la pagina 5*)

Intrerupatoarele au doua stari : inchis sau deschis (0 logic-inchis,1 logic-deschis). Pe baza acestor instructiuni se vor introduce de la placa codificarea culorilor si a formelor.

Butoanele au doua stari : apasat sau neapasat (0 logic-neapasat,1 logic-apasat). Pe baza acestor instructiuni se va misca imaginea la stanga,dreapta sus sau jos. Daca butonul1 este apasat imaginea generata se va misca la stanga, iar daca butonul2 este activ la dreapta(nu este posibila activarea celor doua butoane simultan) si asa mai departe.

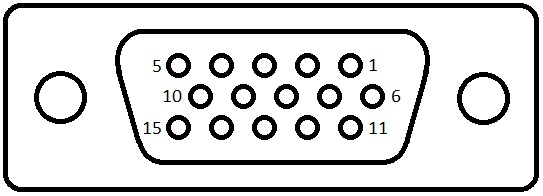
Placa este conectata la monitor prin portul VGA.Utilizatorul controllerului introduce forma dorita de placa prin switchuri,iar in urma alegerii este pusa la dispozitie gama de culori. Pe displayul conectat va fi generata imaginea colorata aleasa si dupa preferinte forma poate sa fie miscata sau nu.

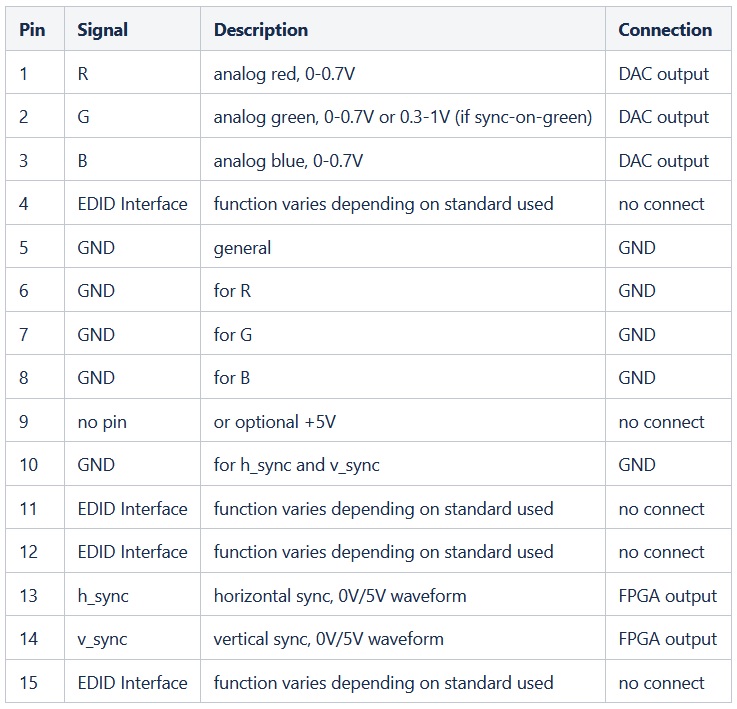
PORT VGA

BUTON

Pentru conectarea placii la monitor se foloseste portul VGA. Mai jos se poate gasi pinii folositi si functionalitatea lor.

SWITCHH





PIN V17,V16,W16-selectie culoare(SEL-COL) W19-BUTON STANGA

PIN W17-RESET T17-BUTON DREAPTA

PIN W15-enable move(EN\_MOVE) U17-BUTON JOS

PIN W13,W14-selectie forma(SEL\_FORM) T18-BUTON SUS

**6. Posibilitati de imbunatatire**

1.Culoarea de background interactiva

Culoarea pe 12 biti poate fi controlata de un block de memorie RAM,iar culoarea de background sa fie controlata de la intrerupatoarele placii. In proiectul curent backgroundul are culoarea implicita negru.

2.Miscarea imaginii

Cu ajutorul semnalelor HPOS si VPOS(redau pozitia curenta a pixelului pe ecran) se pot alege anumite zone de miscare in functie de calcularea anumitor formule si indeplinirea conditiei unei selectii(se prelucreaza un model). In proiectul curent imaginea se poate misca doar stanga-dreapta-sus-jos cu ajutorul butoanelor.

3.Diversificarea imaginii

In functie de modelul ales pentru miscare se pot afisa mai multe design-uri deodata,fiecare sa apartina unei zone de miscare de pe ecran.

**7.Bibliografie**

* Probleme de proiectare logica a sistemelor numerice-Lucia Vacariu,Octavian Cret
* ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT, OAKLAND UNIVERSITY Tutorial: VHDL Coding for FPGAs – VGA Controller
* <https://github.com/dominic-meads/Quartus-Projects/tree/main/VGA_face>
* <https://www.nandland.com/goboard/pong-game-in-fpga-with-go-board-vga.html>
* <https://forum.digikey.com/t/vga-controller-vhdl/12794>
* <https://www.youtube.com/watch?v=WK5FT5RD1sU>
* <https://www.youtube.com/watch?v=7wjTJivsNMM&t=1873s>

Linaru Petra-design desene/implementare cod in vivado/implementare placa

Mudura Ana-documentatie/cod vhdl/resurse proiect