



**Proiect realizat de Popovici Eusebiu-
Ionut si Rus Ionel
Grupa 30214**

Tema proiectului: Controller VGA

Continutul lucrării:

Capitolul I: PROIECTARE	2
1.1 Schema bloc	3
1.2 Unitatea de control si unitatea de executie	3
1.2.2 Resurse:	4
1.2.3 Schema bloc a primei descompuneri.....	6
1.2.4 Schema detaliata a resurselor	6
1.2.5 Organigrama	8
Capitolul II: Descrierea detaliata a proiectului.....	9
2.1 Descriere pentru fiecare iesire si intrare:	10
2.2 Timpii necesari pentru desenare pe orizontala si verticala(rezolutia 640x480):	11
Capitolul III: Manual de utilizare	13
3.1 Simulare in ACTIVE-HDL	13
3.2 Simulare in VIVADO 2022.1	16
3.3 Pentru acest proiect s-a folosit placa BASYS-3. 	18
Capitolul IV: Posibilitati de dezvoltare ulterioara.....	18
Capitolul V: Rezultate	19
Capitolul VI: Bibliografie.....	20

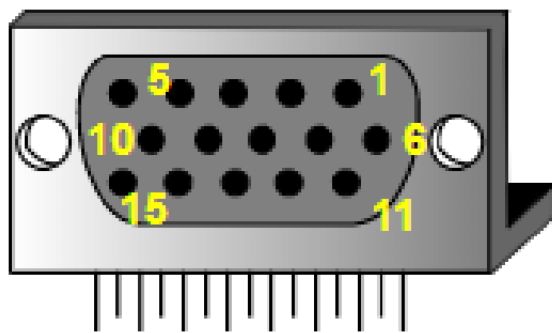
Controller VGA

Folosind placute cu FPGA sa se implementeze un controller VGA. Trebuie afisate 4 imagini diferite, selectia fiind citita de la butoanele placii. Imaginile trebuie să demonstreze abilitatea de selectie a culorilor (minim 4 culori diferite). De asemenea, pozitionarea imaginilor pe ecran trebuie sa fie controlabila pe doua axe cu ajutorul butoanelor. Sugestii de imagini: patrat, dungi verticale, dungi orizontale, triunghi, cerc etc. Timpii necesari diferitelor rezolutii se pot gasi, de exemplu, in tabelul 2-6 din documentatia placii cu FPGA XUPV2P_ User_ Guide.pdf, la pag. 37-38. Diagramele de functionare si explicatii se gasesc in manualele de referinta ale placilor cu FPGA. Proiectul va fi realizat de 2 studenti.

VGA Controller

VGA (Video Graphics Array) se refera specific la hardware introdus prima data de IBM PS/2 computer in 1987. Cu o utilizare globala, acum se refera la standardul de display analog DE-15 Connector.

O sa ne concentram doar la cele 5 semnale din cei 15 pini in acest proiect. Aceste semnale sunt rosu, verde, albastru, HS si VS. Rosu, verde si albastru sunt 3 semnale analoge care specifica culoarea unui punct pe ecran.

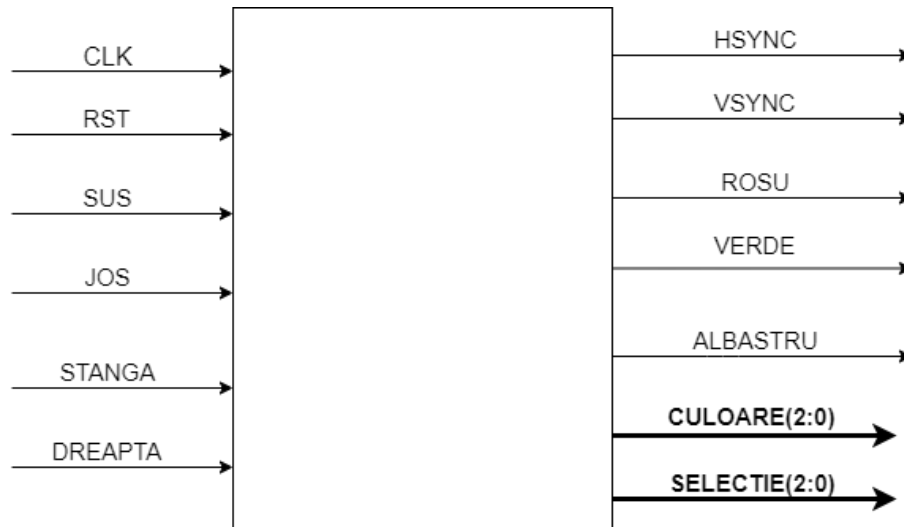


Pin 1: Red	Pin 5: GND
Pin 2: Grn	Pin 6: Red GND
Pin 3: Blue	Pin 7: Grn GND
Pin 13: HS	Pin 8: Blu GND
Pin 14: VS	Pin 10: Sync GND

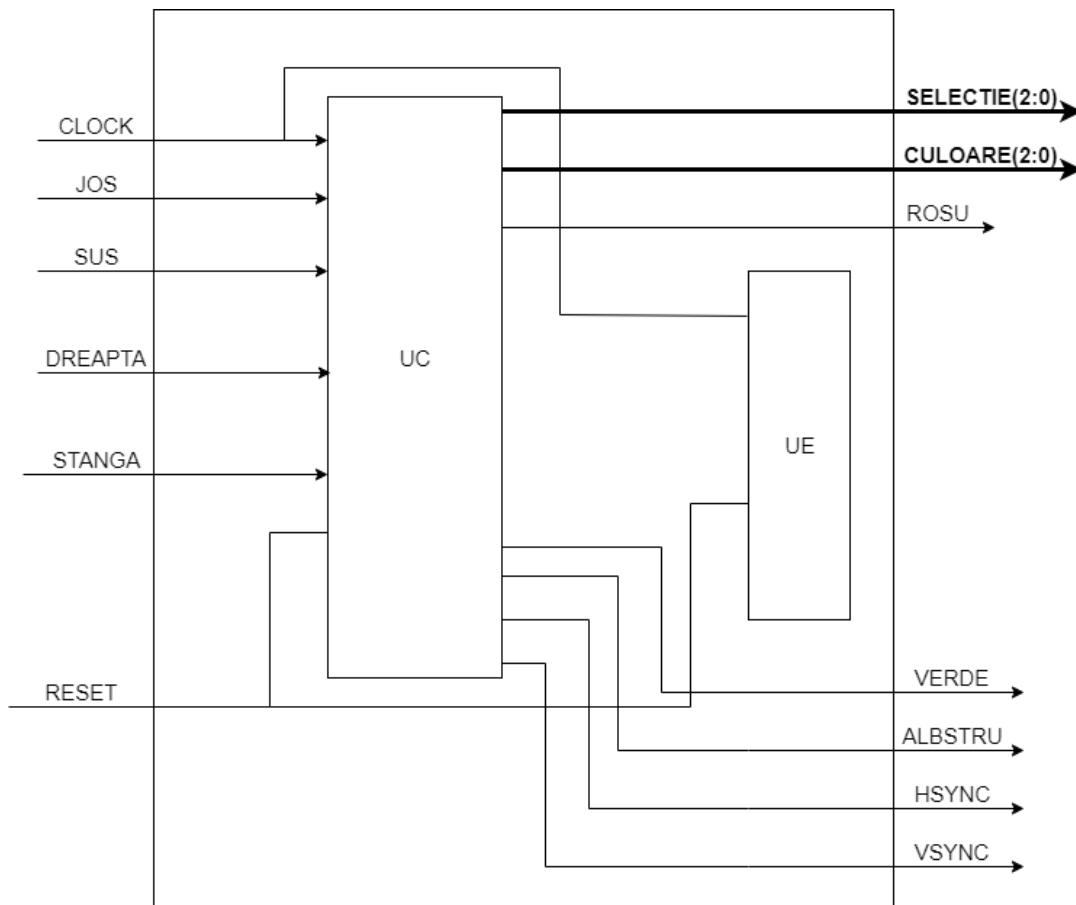
Figure 1. VGA connector.

Capitolul I: PROIECTARE

1.1 Schema bloc



1.2 Unitatea de control si unitatea de executie

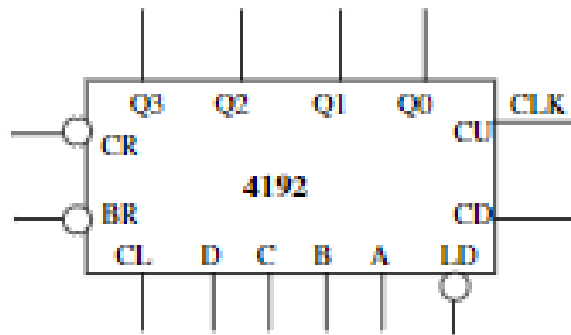


1.2.2 Resurse:

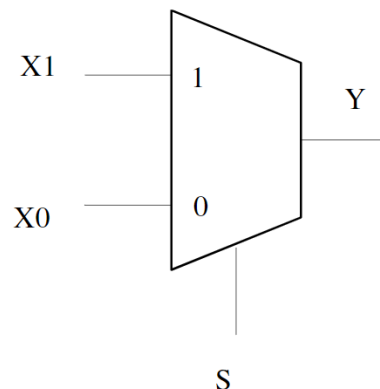
Pentru acest proiect, vom folosi urmatoarele resurse:

- Divizor frecventa
- Numaratoare
- Comparatoare
- Multiplexoare

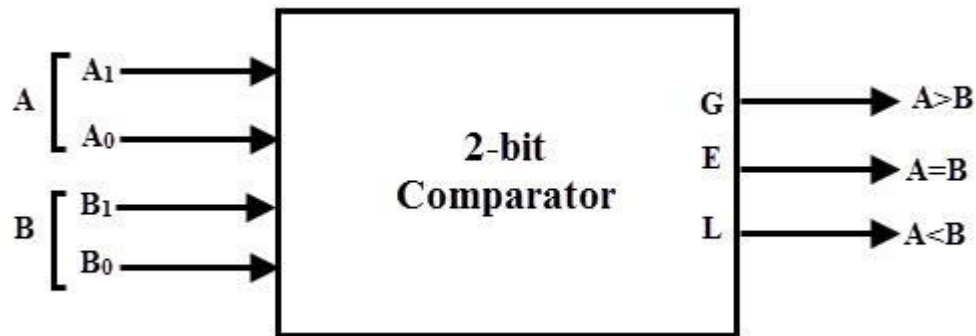
1. Numaratoare: le folosim pentru a parcurge ecranul, pixel cu pixel, in intregime.



2. Multiplexoare: le folosim pentru a selecta culorile si formele geometrice care dorim sa apara pe ecran, in functie de selectia acestora.



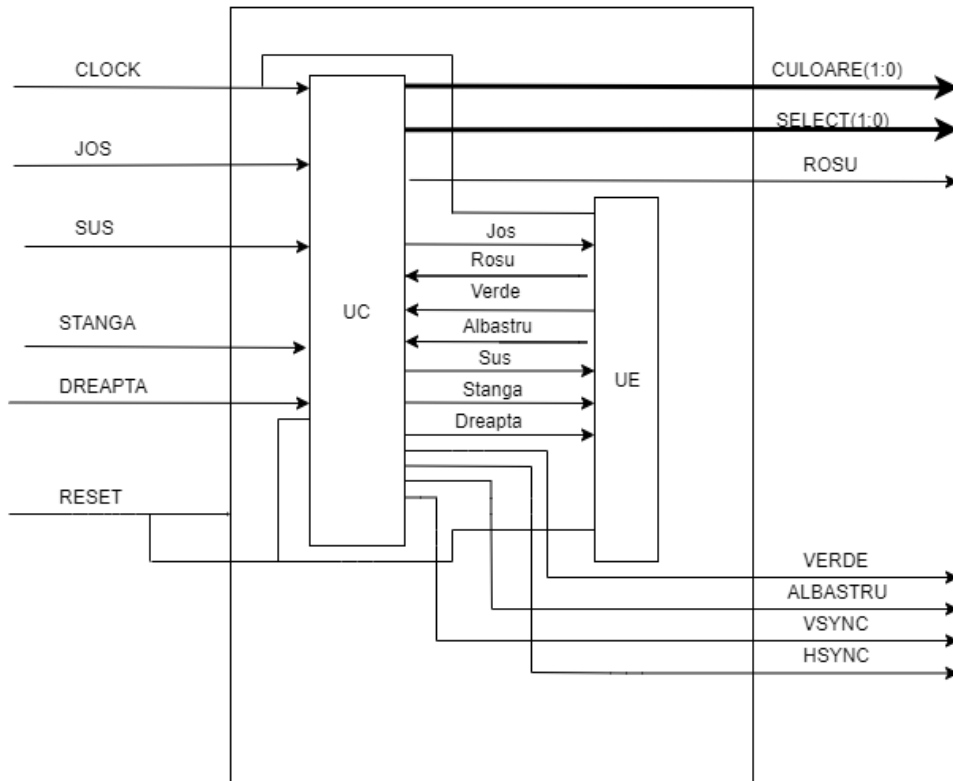
3. Comparatoare: le folosim ca sa putem sti cand putem desena pe ecran(cand suntem in partea vizibila a sa) si ca sa activam HSYNC-ul si VSYNC-ul.



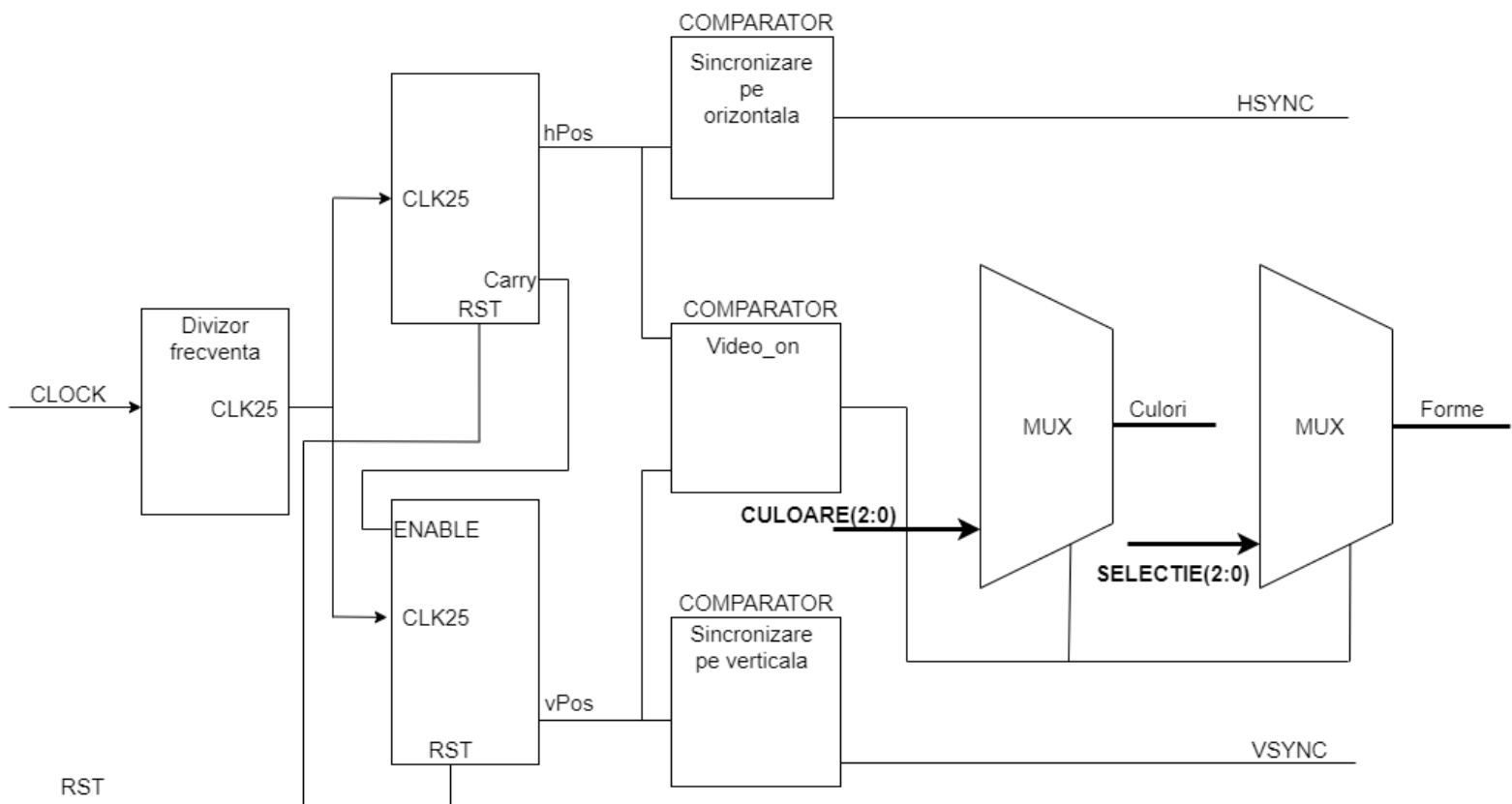
4.Divizor de frecventa: il folosim ca sa divizam clock-ul placutei de 100 mHZ in 50 de mHZ ,iar dupa aceea din 50 de mHZ sa il facem de 25 de mHZ,frecventa de care avem nevoie pe rezolutia 640x480.



1.2.3 Schema bloc a primei descompunerii



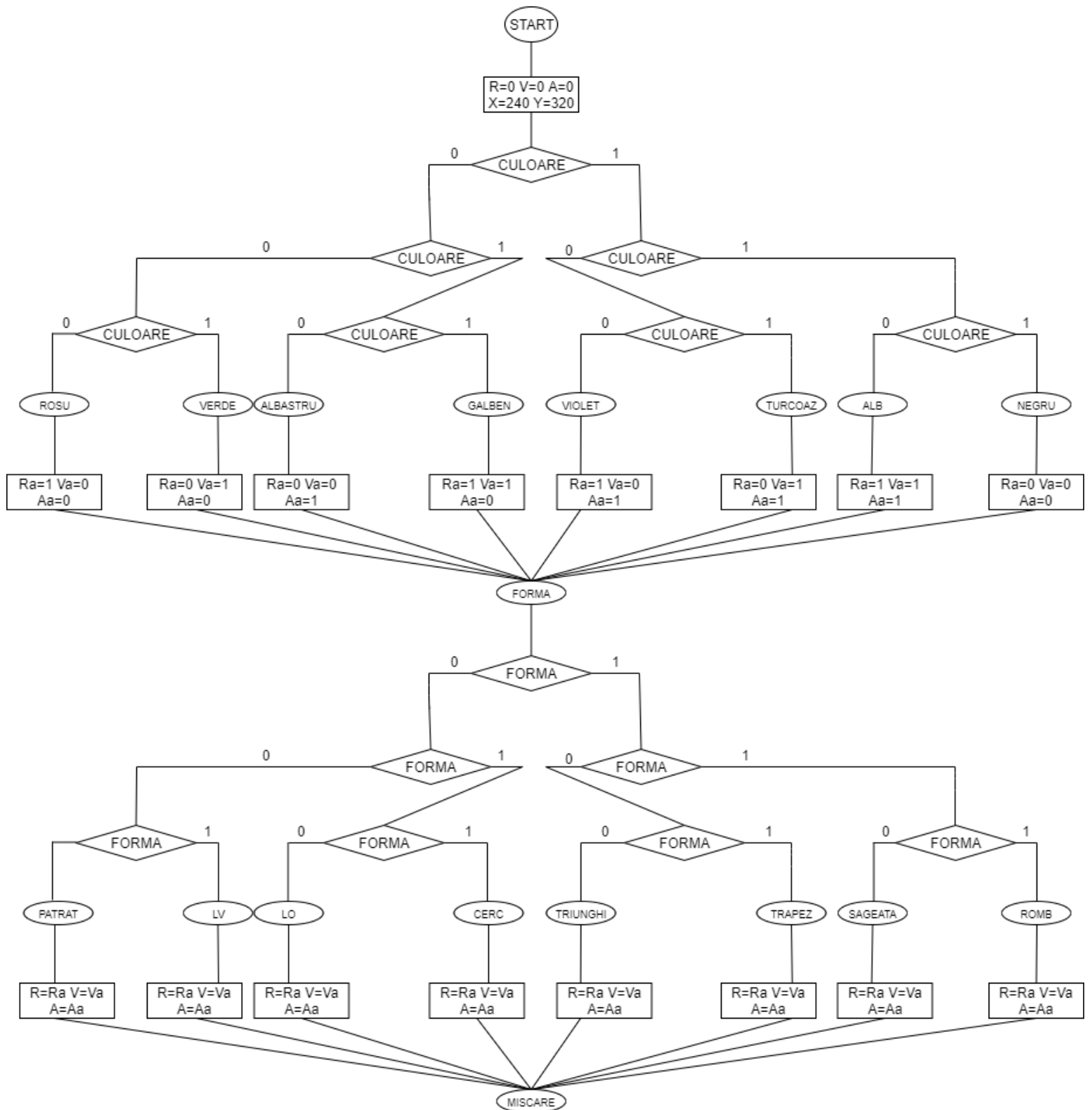
1.2.4 Schema detaliata a resurselor

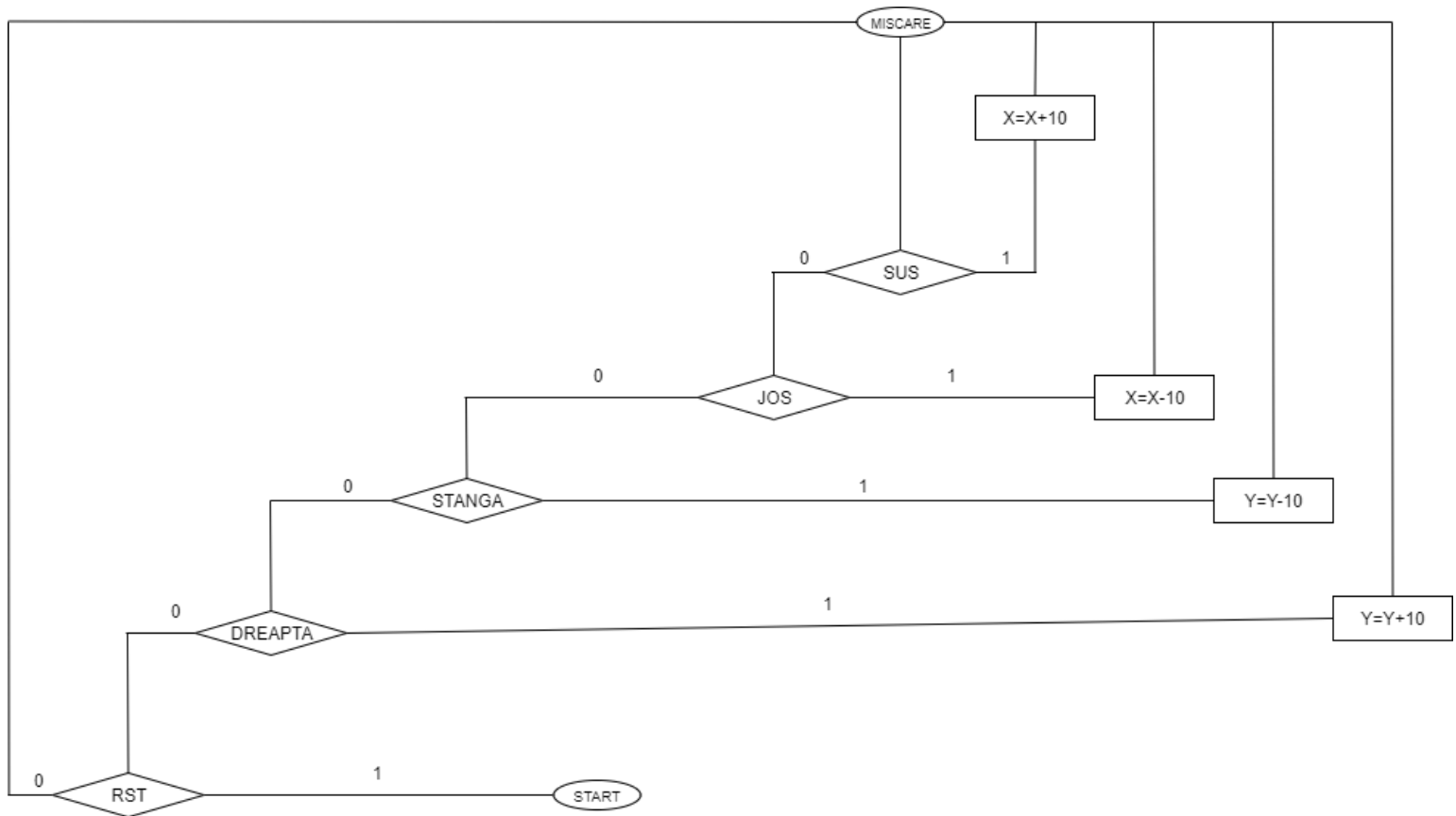


In aceasta schema detaliata,putem observa cum sunt componentele conectate intre ele.Pentru inceput am folosit divizorul de frecventa pentru a ne modifica frecventa clock-ului initial in 25 Mhz. Dupa aceea incepem sa generam HSYNC-ul si VSYNC-ul ca mai apoi ,cand acestea sunt activate,folosim semnalul videoOn pentru a sti momentele in care putem desena formele cerute pe ecran.Prin acest proces vom putea de asemenea sa mutam formele geometrice pe 2 axe,in 4 directii(sus,jos,stanga si dreapta) .

Daca putem desena,abia dupa aceea,bazat pe SELECTIE(2:0) vom desena cele 8 figuri diferite, dar vom putea si sa schimbam culorile acestora,cu ajutorul unei selectii Culoare pe 3 biti.

1.2.5 Organigrama





Capitolul II: Descrierea detaliata a proiectului

- **Intrari:** -CLOCK
 - SUS
 - JOS
 - STANGA
 - DREAPTA
 - RESET
 - SELECTIE(2:0)
 - CULOARE(2:0)

- **Iesiri:** -ROSU
-VERDE
-ALBASTRU
-HSYNC
-VSYNC
-ANOD
-CATOD

2.1 Descriere pentru fiecare iesire si intrare:

1. Culoare(2:0)-putem selecta 8 culori diferite:

- Rosu("000")
- Verde("001")
- Albastru("010")
- Galben("011")
- Violet("100")
- Turcoaz("101")
- Alb("110")
- Negru("111")

2. Selectie(2:0)-putem selecta 8 forme diferite:

- Patrat("000")
- Linie verticala("001")
- Linie orizontala("010")
- Cerc("011")

- Triunghi(“100”)
- Trapez(“101”)
- Sageata(“110”)
- Romb(“111”)

3.Clock – un semnal de tact pe baza caruia toate componentele lucreaza.

4.Sus/Jos – mutam figura apasand butoanele asociate axei y sus si jos.

5.Dreapta/stanga – mutam figura apasand butoanele asociate axei x dreapta si stanga.

6.Iesirile Rosu/Verde/Albastru – iesirile care decid culorile fiecarui pixel la fiecare pas.

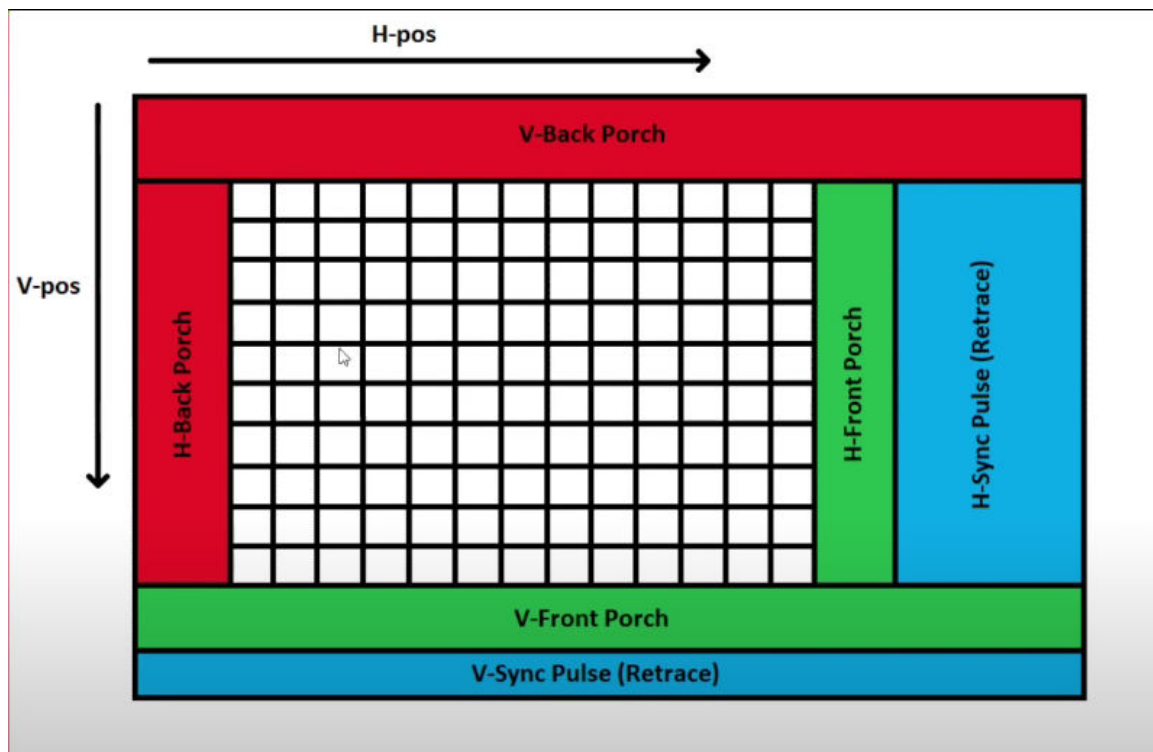
7. Anod/Catod – anodul decide ce afisor se va aprinde din cele 4,iar catodul controleaza cele 7 segmente ale afisorului.

2.2 Timpii necesari pentru desenare pe orizontala si verticala(rezolutia 640x480):

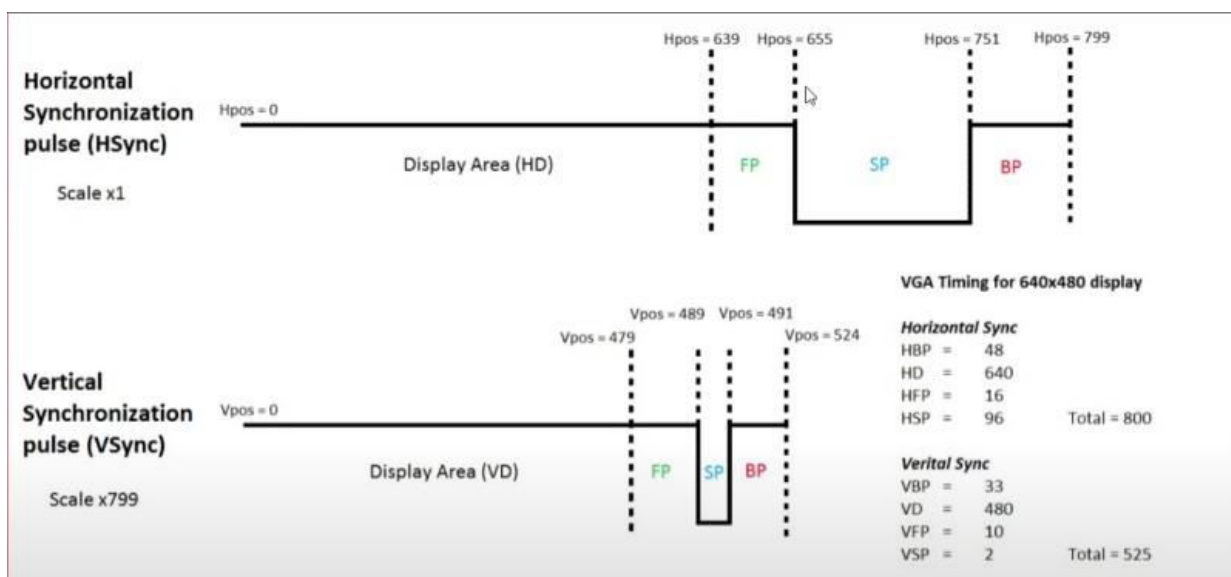
Output Format	Pixel Clock	DCM Settings		Verilog Horizontal Timing Parameters				
				H Active	H FP	H Synch	H BP	H Total
	MHz	M	D	Pixels	Pixels	Pixels	Pixels	Pixels
640 x 480 @ 60 Hz	25.00	1	4	640	16	96	48	800

Output Format	Pixel Clock	DCM Settings		Verilog Vertical Timing Parameters				
				V Active	V FP	V Synch	V BP	V Total
	MHz	M	D	Pixels	Pixels	Pixels	Pixels	Pixels
640 x 480 @ 60 Hz	25.00	1	4	480	9	2	29	520

In urmatoare poza putem observa cum este de fapt impartit ecranul. Acesta este alcatuit din mai multe componente (H/V-Back Porch, H/V-Front Porch, H/V-Synch) care reprezinta zonele colorate, si parte active a ecranului in care putem desena (partea alba):



In aceasta imagine putem observa modificarea HSYNC si VSYNC-ului in functie de pozitie (unde e si unde nu e activ).

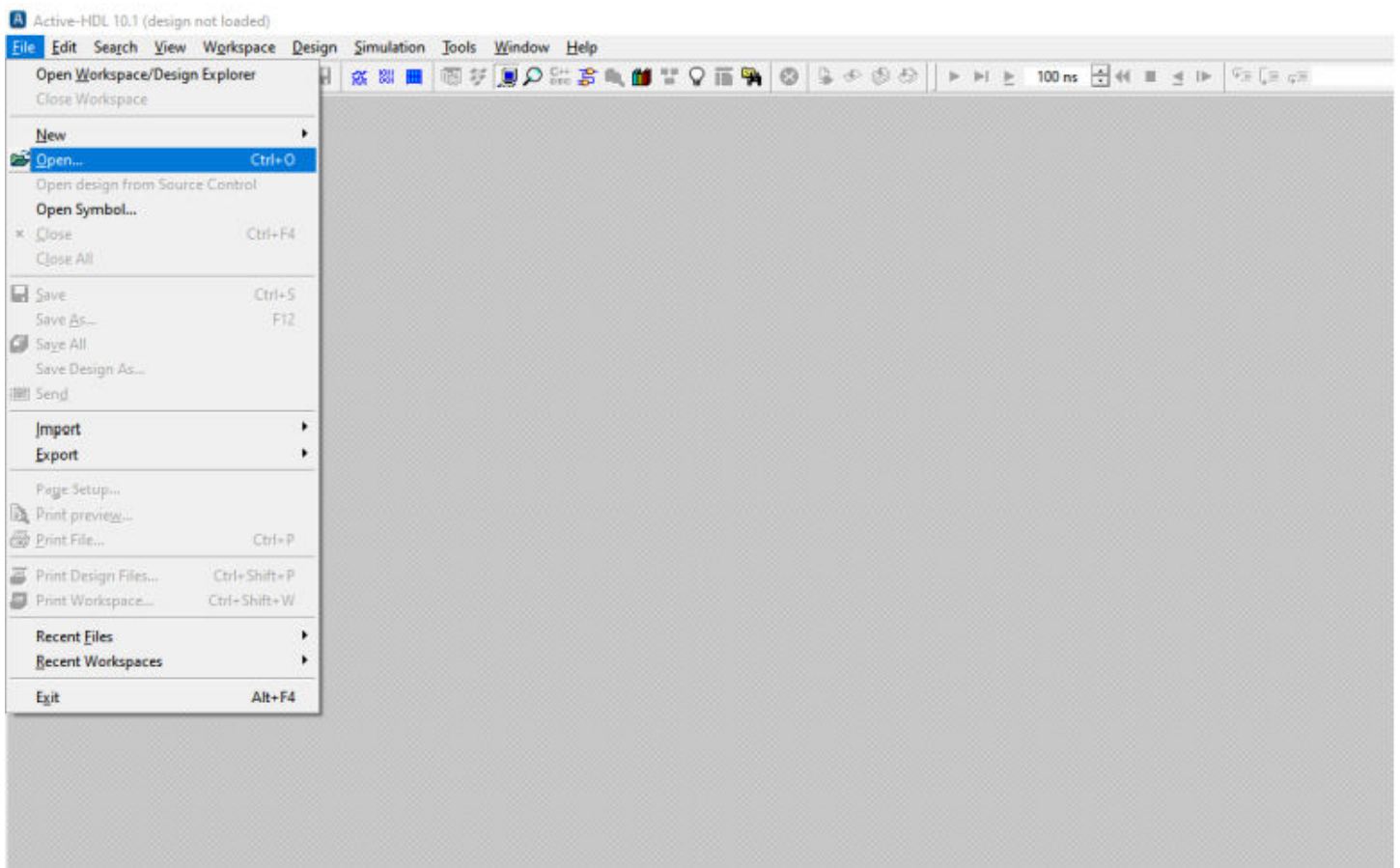


Capitolul III: Manual de utilizare

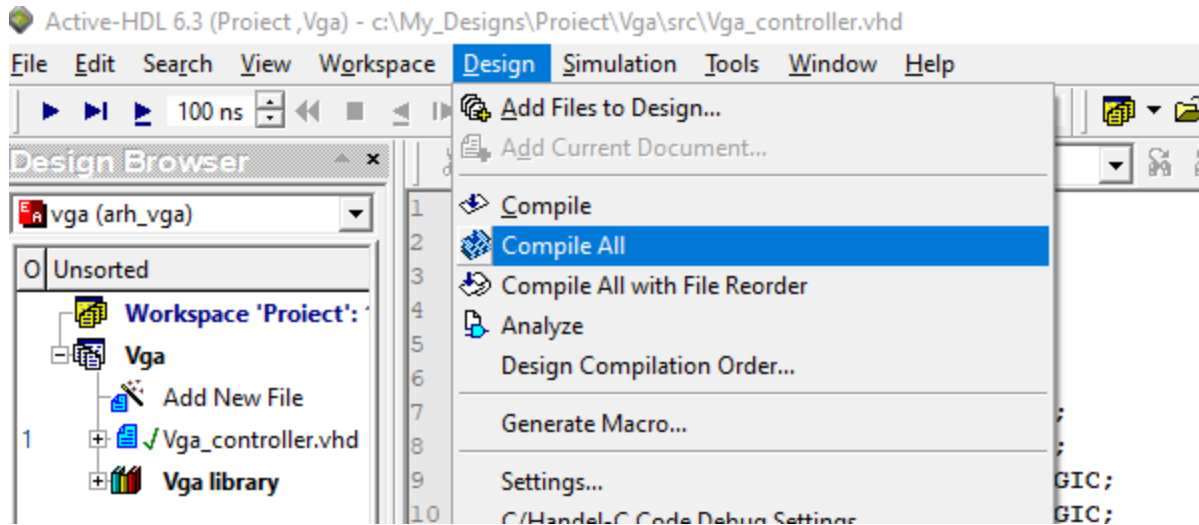
3.1 Simulare in ACTIVE-HDL

Ca sa putem simula VGA Controller-ul trebuie sa executam urmatoorii pasi.

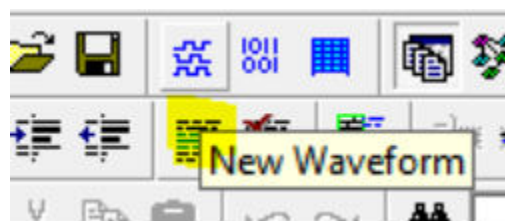
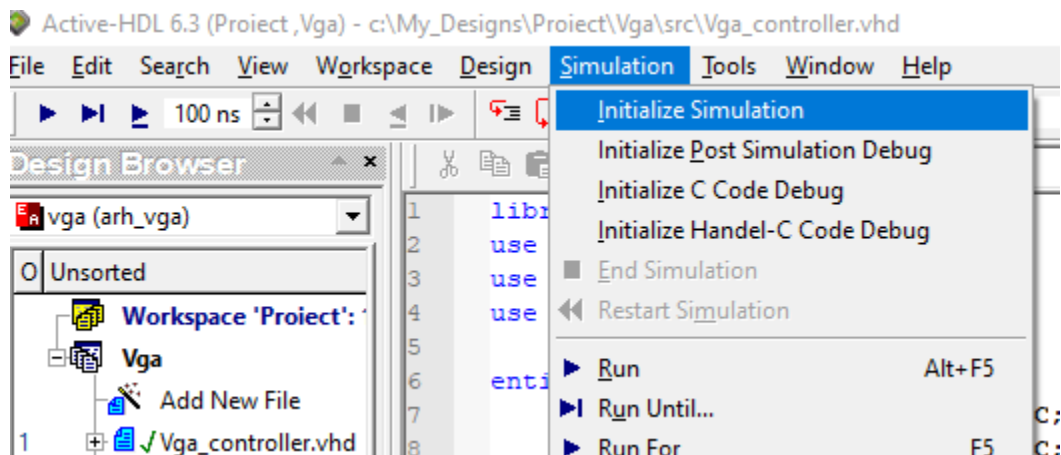
Primul pas: Dupa ce pornim aplicatia vom porni proiectul din **FILE – Open:**



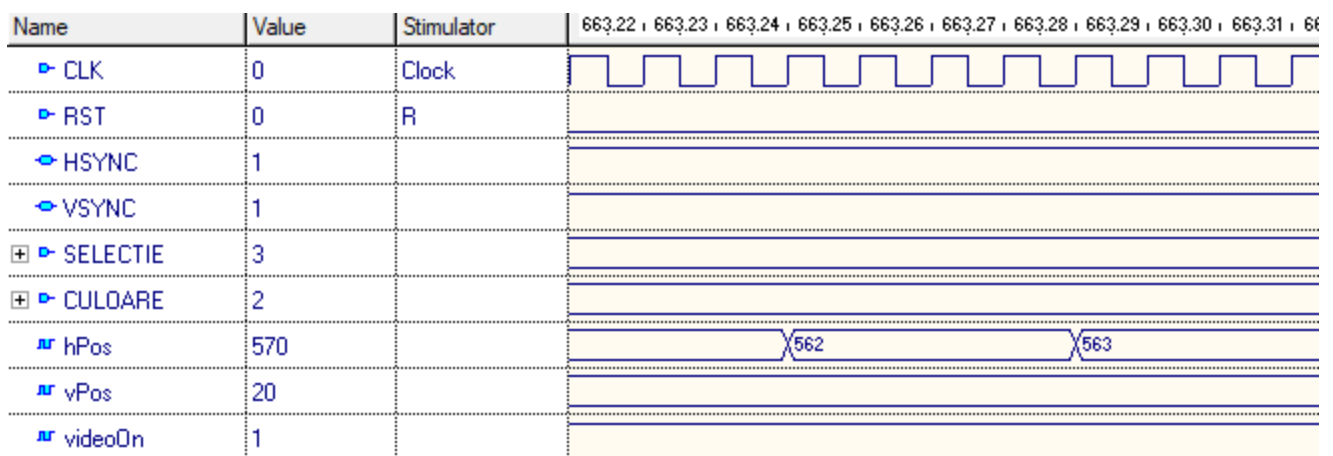
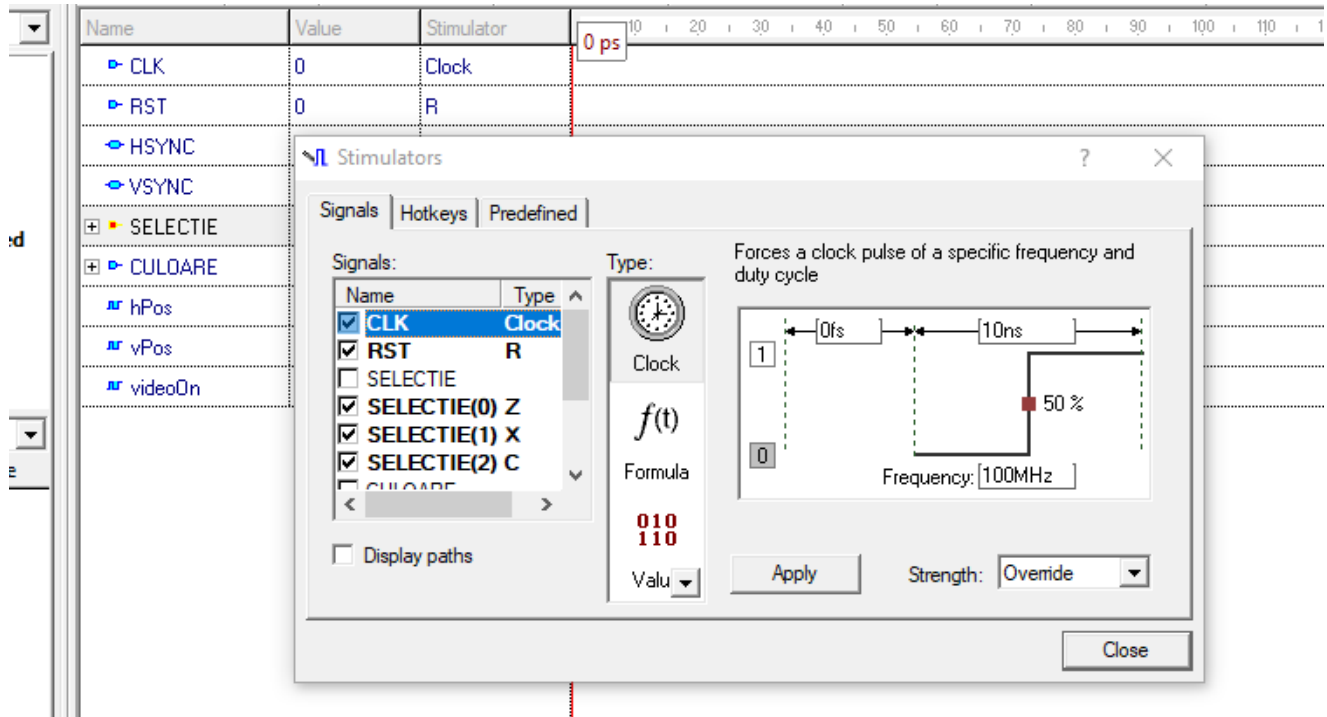
Al doilea pas: acesta consta in compilarea codului. Inainte de initializa simularea,trebuie sa compilam : **Design – Compile all**



Al treilea pas: dupa ce compilam,initializam simularea din meniul **Simulation – Initialize simulation**,iar,dupa aceea,trebuie sa apasam pe iconita **New Waveform** pentru a observa cum merge simularea in timp real.



Al patrulea pas: in waveform selectam toate semnalele de care avem nevoie(Click dreapta – Add Signals).Dupa aceea apasam pe fiecare semnal in parte si ii oferim stimulatorul necesar.De exemplu pentru CLK ii oferim stimulator de tip Clock, pentru RST un hotkey etc.



3.2 Simulare in VIVADO 2022.1

Pentru testarea pe placuta s-a folosit Vivado 2022

1. Pentru inceput vom crea un proiect nou apasand: **Create project.**

2. Dam next la urmatorii pasi.

3. Alegem unde vrem sa salvam proiectul.

4. La **Project Type** selectam **RTL PROJECT.**

5. La sectiunea **Add Sources** ne asiguram ca selectam **Target language: VHDL** si **Simulator language: VHDL.**

6. Dam next la urmatorii pasi pana ajungem la **Default part**, unde vom selecta setarile placutei: **-Family -> Artix-7**

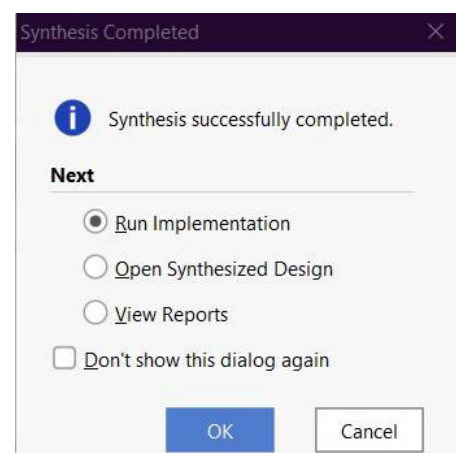
- Package -> cpg236

- Speed -> -1

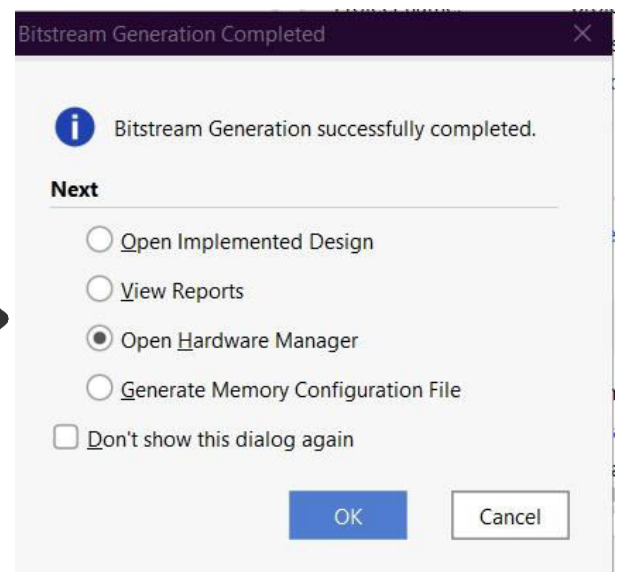
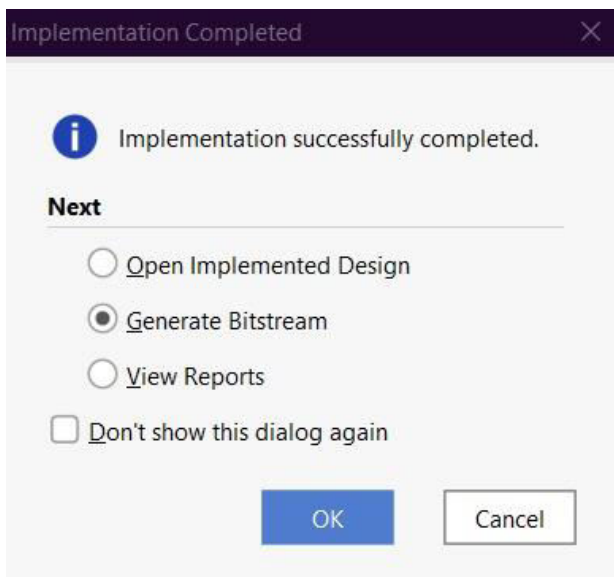
- xc7a35tcpg236-1

7. Dupa aceea dam next si finish si ni se va deschide noul proiect.

8. Urmatorul pas consta in sintetizarea proiectului apasand **Run Synthesis**, iar daca aceasta se efectueaza corect, urmeaza sa apasam **Run Implementation.**

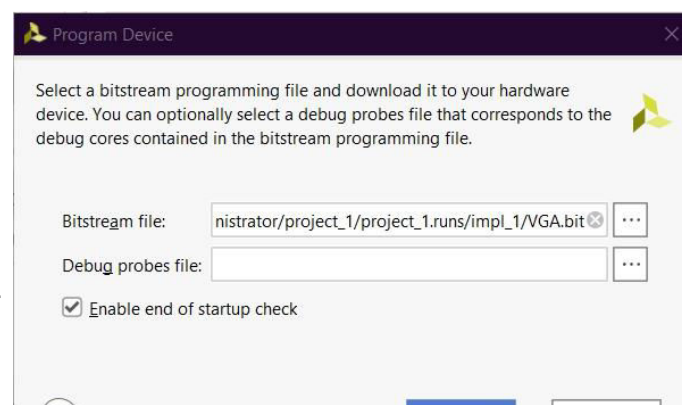
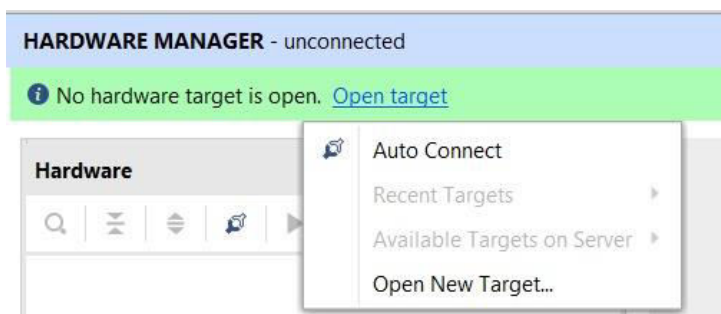


9. Dupa acest pas, ni se va deschide un TAB de unde selectam **Generate Bitstream** si apasam OK, iar imediat dupa vom apasa pe **Open Hardware Manager**.

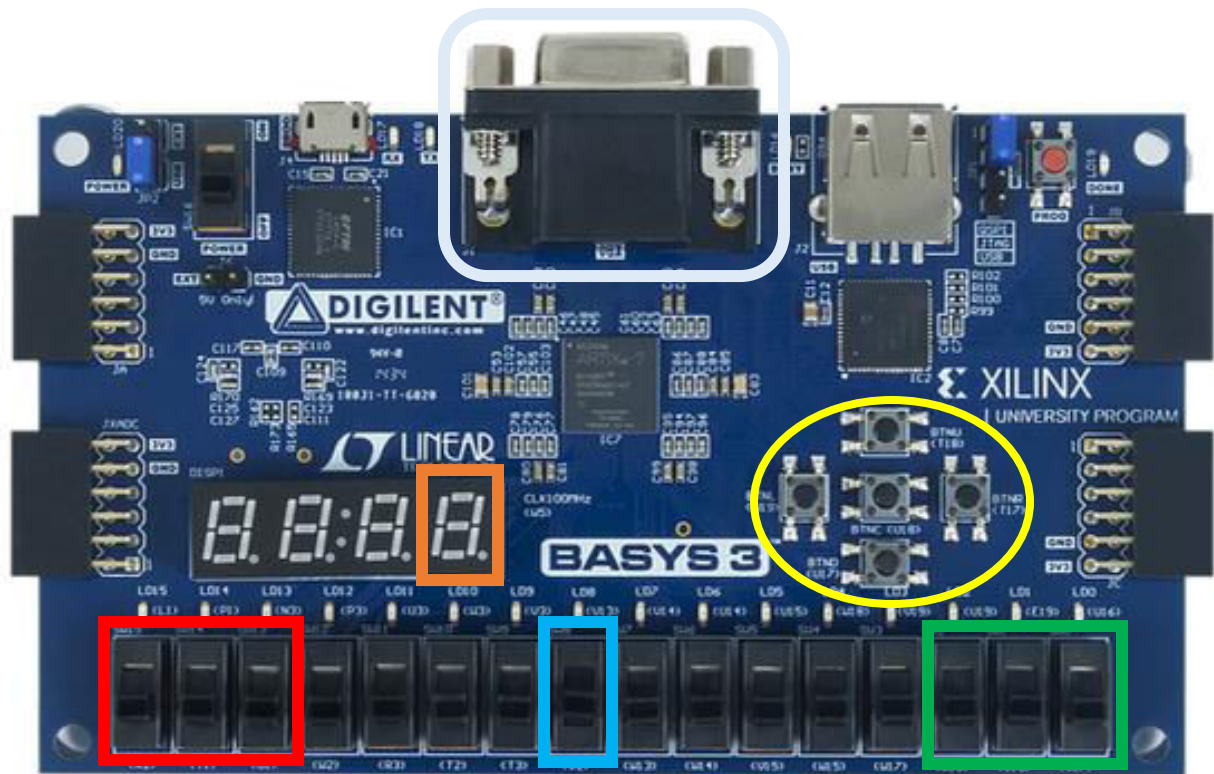


10. In final apasam pe **Open target**, dupa care selectam **Program Device**, lucru care ne obliga sa alegem sursa pe care vrem sa o rulam.

11. Dupa selectia sursei codul va fi rulat si atunci vom putea testa proiectul.



3.3 Pentru acest proiect s-a folosit placuta BASYS-3.

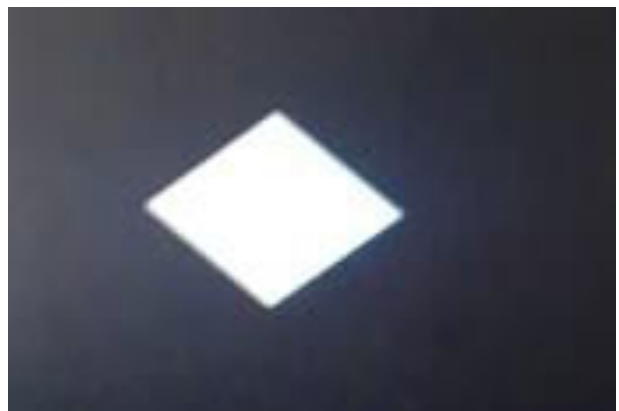
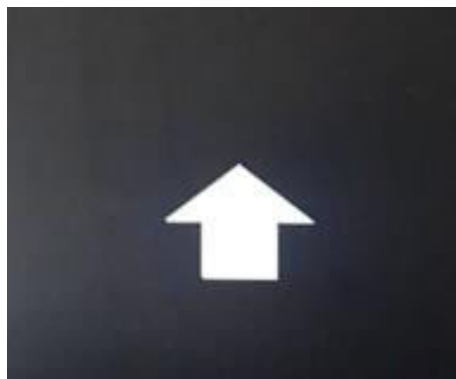
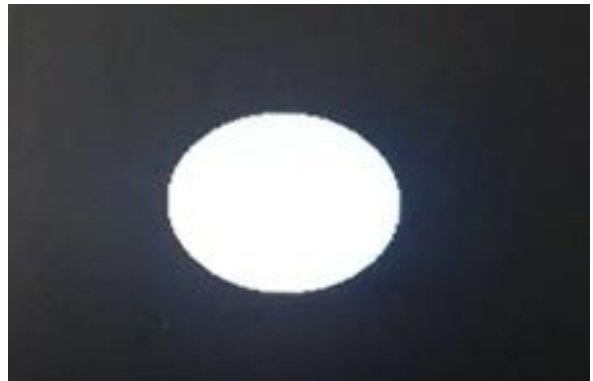


Switchurile din chenarul rosu reprezinta selectarea culorii, cele din chenarul verde, selectarea figurii si switchul din chenarul albastru este folosit pentru RESET. In cercul galben avem butoanele, folosite la deplasarea figurii pe monitor. De asemenea, chenarul portocaliu este afisorul 7-segmente folosit pentru a afisa numarul selectiei, iar chenarul albastru deschis este portul de VGA folosit la conectarea placutei la monitor.

Capitolul IV: Posibilitati de dezvoltare ulterioara

Niste idei de implementare viitoare ar putea fi: posibilitatea de a afisa pe ecran niste imagini .png sau .jpg, posibilitatea de a face sa apara si sa dispara formele pentru un interval de timp(blink mode) sau chiar de a urma un traseu hardcodat de utilizator. De asemenea, o alta idee de implementare ar fi cea de a putea modifica culoarea unei parti din figura, de exemplu liniile orizontale/verticale sa fie formate din 3-4 culori si posibilitatea de a afisa in acelasi moment mai multe figuri pe monitor sau multiplicarea figurilor.

Capitolul V: Rezultate



Capitolul VI: Bibliografie

- ❖ <https://www.youtube.com/watch?v=eJMYVLPX0no>
- ❖ <https://www.youtube.com/watch?v=wzhDRIX2Ors>
- ❖ https://www.seas.upenn.edu/~ese171/boards/XUPV2P_User_Guide.pdf
- ❖ <http://tinyvga.com/vga-timing/640x480@60Hz>