

**Proiectarea Sistemelor Numerice**

**VGA Controller**

Relizat de:

Neghină Laurențiu

Tarța Antonia

Grupa 30219

CONTROLLER VGA

Folosind plăcuțe cu FPGA să se implementeze un controller VGA. Trebuie afișate 4 imagini diferite, selecția fiind citită de la butoanele plăcii. Imaginile trebuie să demonstreze abilitatea de selecție a culorilor (minim 4 culori diferite). De asemenea, poziționarea imaginilor pe ecran trebuie să fie controlabilă pe două axe cu ajutorul butoanelor. Sugestii de imagini: pătrat, dungi verticale, dungi orizontale, triunghi, cerc etc. Proiectul va fi realizat de 2 studenți.

[**CAPITOLUL I: PROIECTARE**](#_epz3a6ri1crj) **3**

[**1.1. Schema bloc**](#_fm0c5g4le8fr) **3**

[**1.2. Unitatea de Control și Unitatea de Execuție**](#_ozv6dh2tfv0t) 4

[1.2.1. Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE](#_yl3f9d4nm71g) 5

[1.2.2. Determinarea resurselor (UE)](#_7ymco7f6soax) 5

[1.2.3. Schema bloc a primei descompuneri](#_hf5sow58tpfe) 10

[**1.3. Reprezentarea UC prin diagrama de stări(organigrama)**](#_xeatco5pn97a) **11**

[**2. CAPITOLUL II: JUSTIFICAREA SOLUȚIEI ALESE**](#_3g642j4wy880) **12**

[**2.1. Aspecte generale**](#_c0li6oai7qvc) **12**

[**2.2. Algoritmi folosiți pentru generarea imaginilor**](#_brhx7ljp910f) **12**

[2.2.1. Algoritm folosit pentru generarea pătratului](#_jfm5rbl4cx7n) 12

[2.2.2. Algoritm folosit pentru generarea cercului((x-a)2+(y-b)2=r2)](#_lcx6vxp40hy8) 12

[2.2.3. Algoritm folosit pentru generarea triunghiului](#_cr45birdl7wx) 12

[2.2.4. Modul de afișare a imaginii](#_7okgvfi6i8tb) 13

[**3. CAPITOLUL III: MANUAL DE ÎNTREȚINERE ȘI UTILIZARE**](#_8elmevmeq49k) **13**

[**3.1. Basys 3**](#_lr1cr7f8l2mi) **13**

[**3.2. Instrucțiuni pentru afișare și colorare**](#_d62acn8ow3jj) **14**

[3.2.1. Instrucțiuni de selectare a imaginii](#_2hx57cpznt6k) 14

[3.2.2. Instrucțiuni de selectare a culorii](#_m4vrjh7ecnad) 14

[**3.3. Instrucțiuni pentru mișcarea imaginii**](#_cowqna7l5au8) **14**

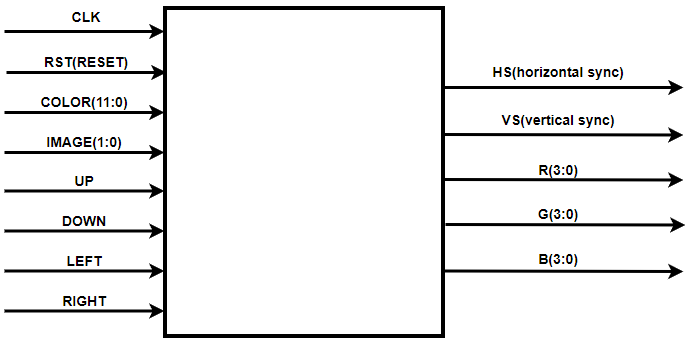
[**4. CAPITOLUL IV: POSIBILITĂȚI DE DEZVOLTĂRI ULTERIOARE**](#_2q1ne6xdjlfs) **15**

[**5. CAPITOLUL V: DISTRIBUIRE SARCINI**](#_31b8xq4iut60) **15**

[**6. CAPITOLUL VI: BIBLIOGRAFIE**](#_nsgacj4egvov) **16**

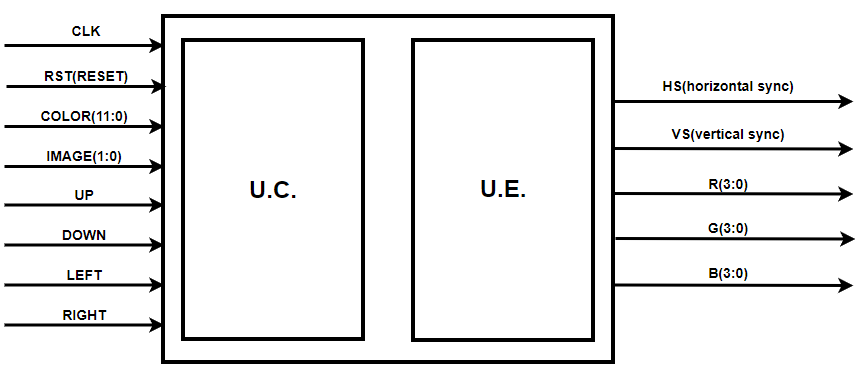
# CAPITOLUL I: PROIECTARE

## 1.1. Schema bloc

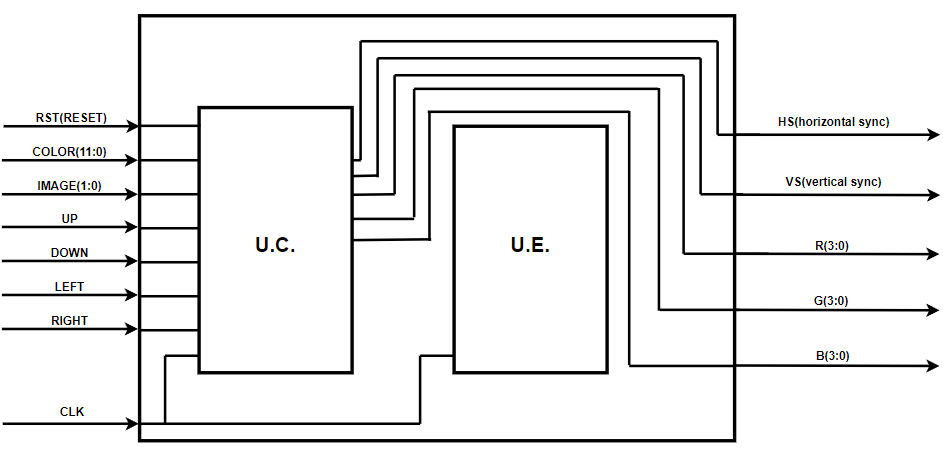


## 1.2. Unitatea de Control și Unitatea de Execuție

Prima descompunere a oricărui sistem este una în care se face diferenta între logica de control din sistem și resursele sistemului. Logica de control este reprezentată de Unitatea de Control, iar resursele sunt reprezentate de Unitatea de Execuție.



### 1.2.1. Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE



### 1.2.2. Determinarea resurselor (UE)

**RESURSE:**

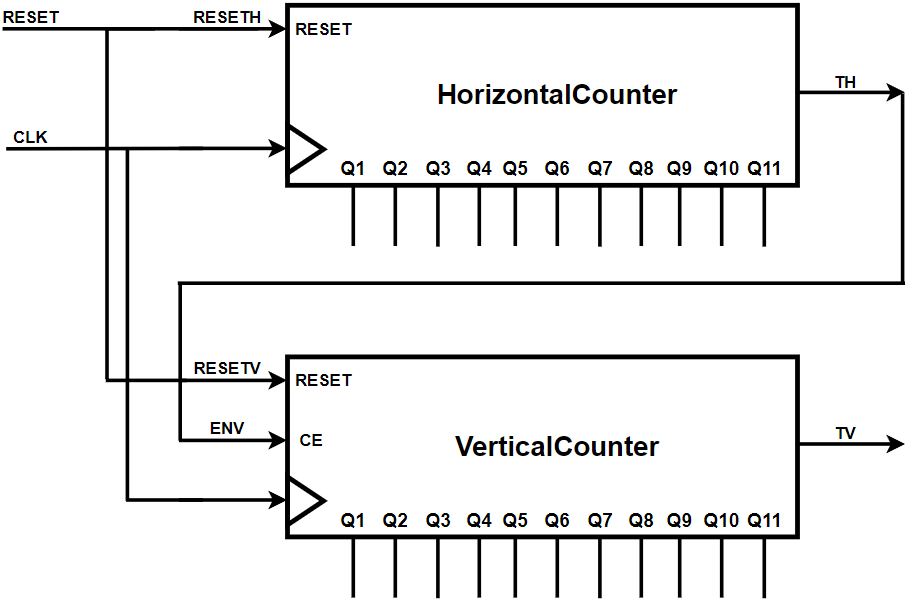
## Divizor de frecvență

Acesta este un divizor de frecvență care generează o frecvență de N MHz pentru funcționarea la o anumită rezoluție. Avem un semnal NCLK de intrare generat de UE pentru UC.

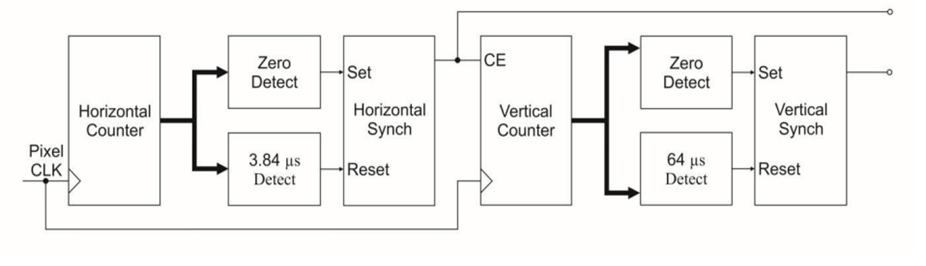


1. Horizontal/Vertical Counter

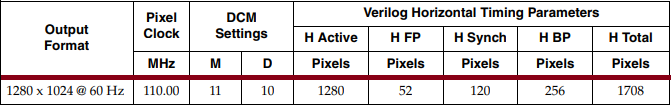
Sunt două numărătoare cascadate. În momentul în care Horizontal Counter ajunge la finalul buclei de numărare, Vertical Counter crește cu o unitate.

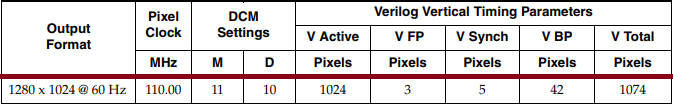


Sistemul de sincronizare pentru setarea rezoluției:



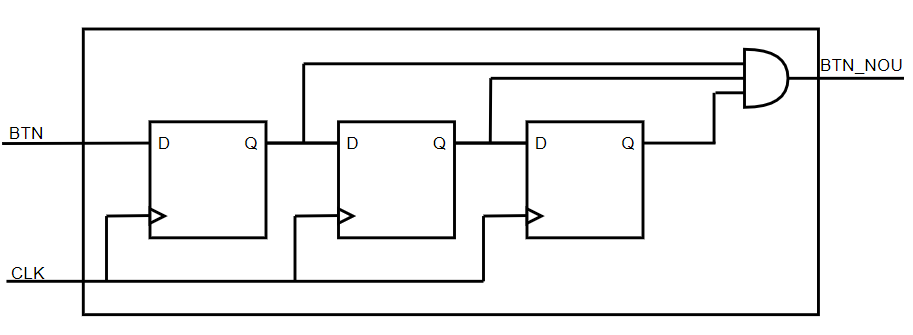
Specificații pentru rezoluția aleasă:





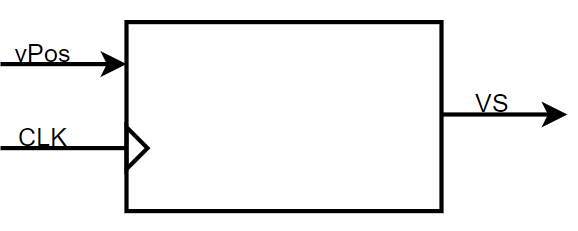
1. Debouncer

Debouncer-ul este folosit pentru a ne asigura că un singur semnal este perceput pentru o singură deschidere sau închidere a unui conctact.



### Vertical Sync

Vertical Sync este componenta responsabilă de atribuirea semnalului VS, bazându-se pe o formulă.



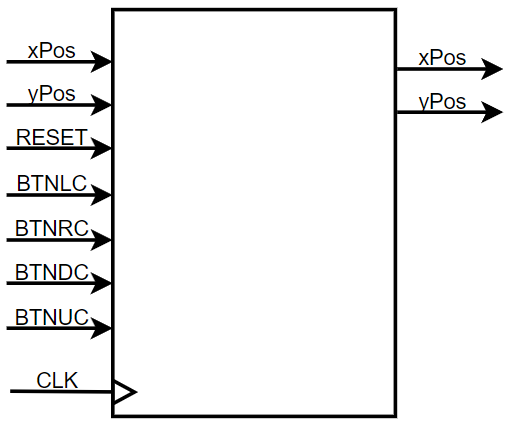
1. Horizontal Sync

Horizontal Sync este componenta responsabilă de atribuirea semnalului HS, bazându-se pe o formulă.



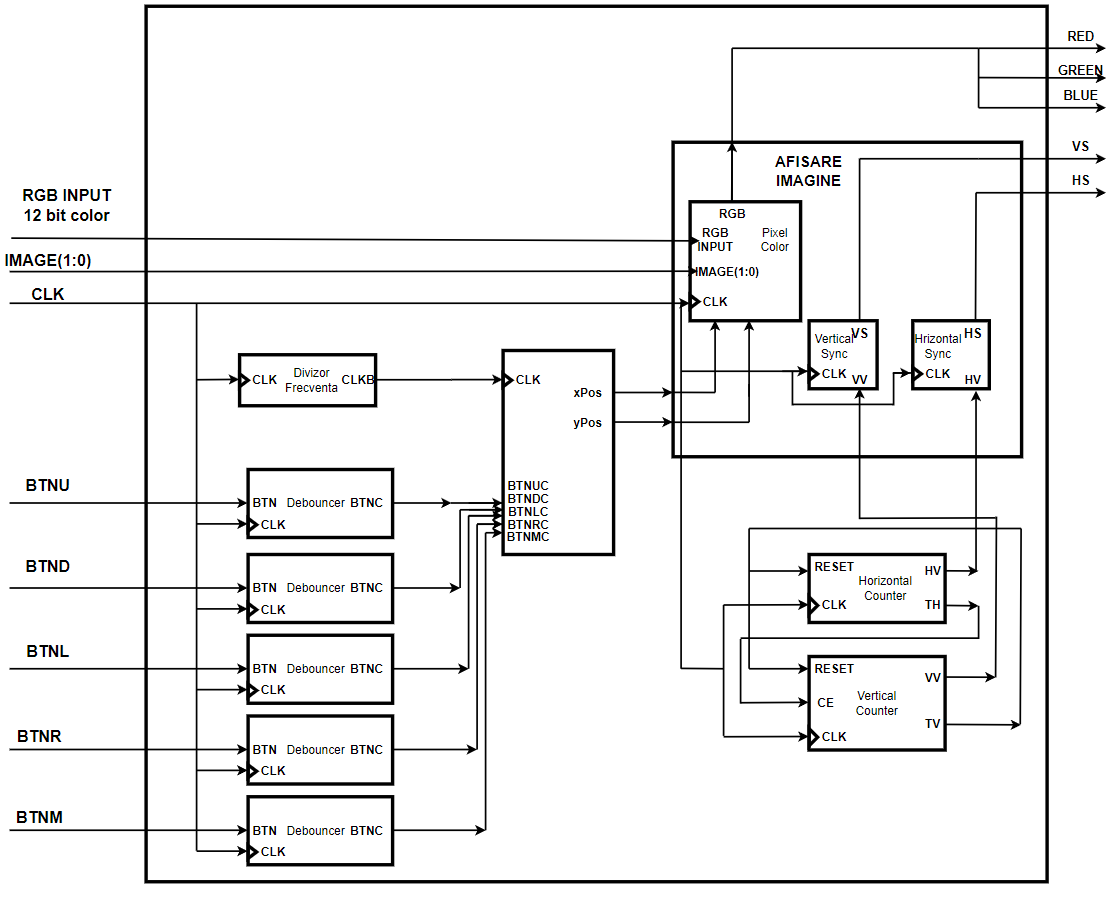
1. Button Manager

Button Manager actualizează poziția imaginii pe axele Ox și Oy pe baza apăsării butoanelor.



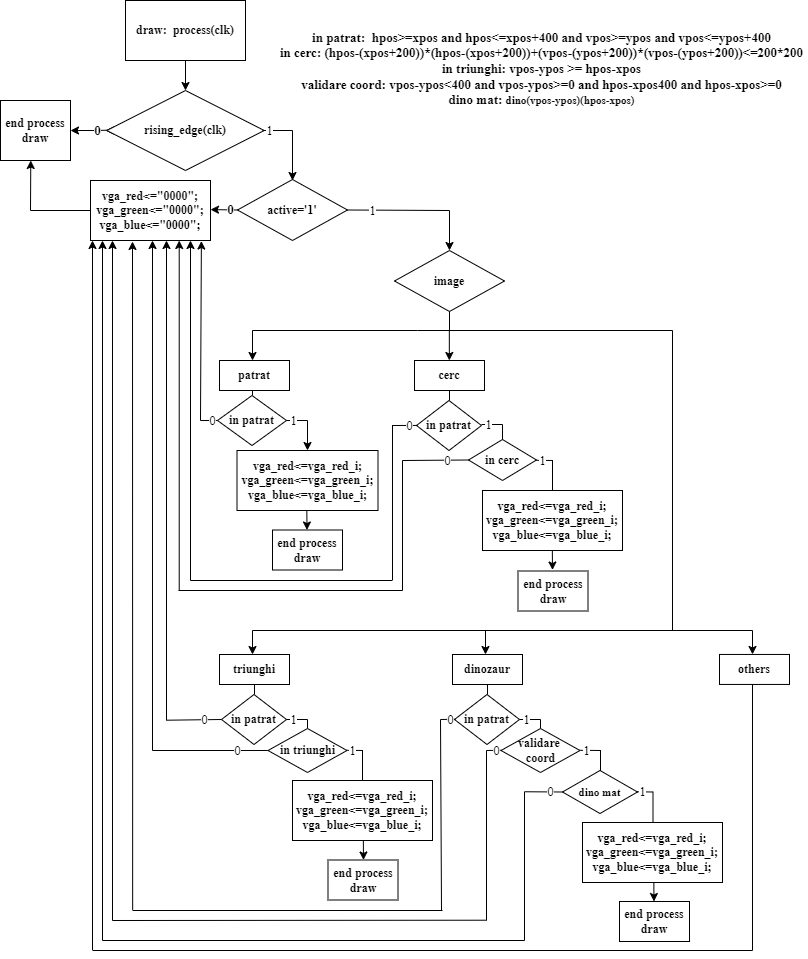
### 1.2.3. Schema bloc a primei descompuneri

### 



# 

## 1.3. Reprezentarea UC prin diagrama de stări(organigrama)



# 2. CAPITOLUL II: JUSTIFICAREA SOLUȚIEI ALESE

## 2.1. Aspecte generale

Pentru realizarea acestui Controller VGA am decis să îmbinăm utilul cu plăcutul, abordând două stiluri diferite de afișare a imaginilor. Primul mod este avantajos atât din punct de vedere al memoriei utilizate cât și al vitezei deoarece generăm imagini cu ajutorul unor formule matematice. Dorința de a genera imagini mai complexe ne-a determinat să folosim o memorie ROM, astfel descoperind al doilea mod de afișare.

Am adoptat un stil de implementare comportamental ușor de înțeles atât de programatori cât și de utilizatori. De asemenea modul de comportare al VGA controller-ului avantajează aceasta formă de programare comportamentală.

## 2.2. Algoritmi folosiți pentru generarea imaginilor

### 2.2.1. Algoritm folosit pentru generarea pătratului

Pentru generarea unui pătrat verificăm dacă coordonatele generate de numărătoare sunt încadrate între xPos, xPos+400, yPos și yPos+400, cele 4 colțuri ale pătratului. Dacă condiția este adevărată colorăm pixelul, altfel el va rămâne culoarea background-ului.

### 2.2.2. Algoritm folosit pentru generarea cercului((x-a)2+(y-b)2=r2)

Pentru generarea unui cerc folosim formula prezentată mai sus într-un pătrat de 400x400 pixeli. Aceasta verifică dacă coordonatele generate de numărătoarele noastre se află în interiorul cercului cu raza de 200 de pixeli și centrul lui de coordonate xPos+200 și yPos+200. Dacă pixelul se află în perimetrul cercului atunci el va fi colorat, altfel va rămâne culoarea background-ului.

## 

### 2.2.3. Algoritm folosit pentru generarea triunghiului

Pentru generarea unui triunghi dreptunghic ne folosim de faptul că acesta este jumătate dintr-un pătrat. Astfel, putem să verificăm dacă coordonatele actuale se află sub diagonala principală. In acest caz, colorăm pixelul actual, altfel va ramane culoarea background-ului.

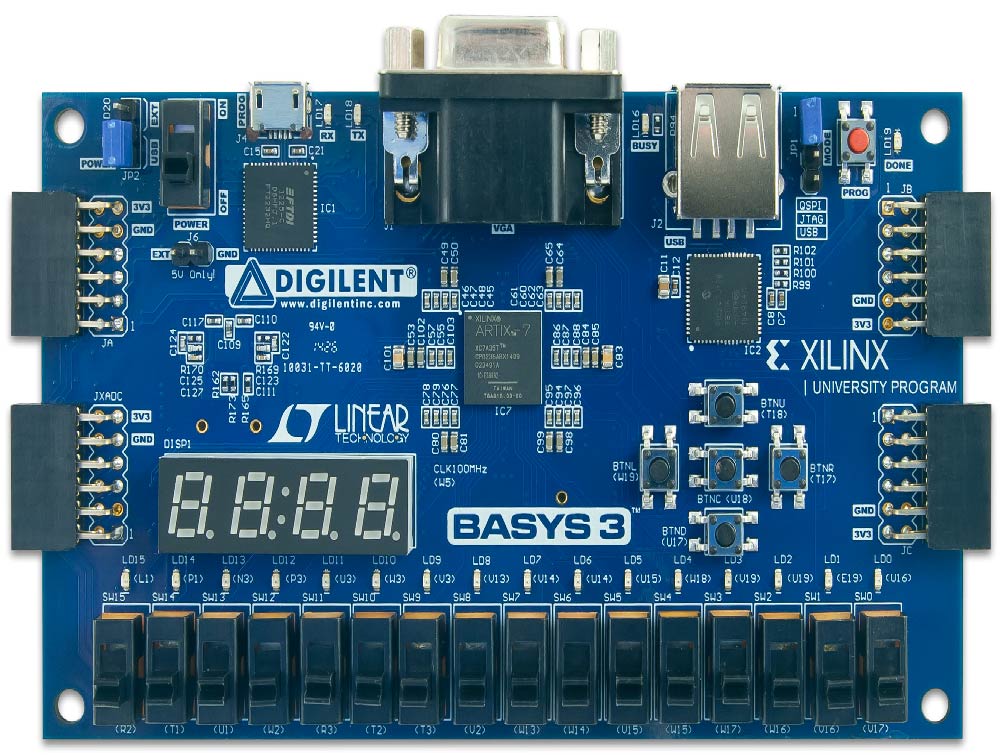
### 2.2.4. Modul de afișare a imaginii

În acest caz, parcurgem în același timp ecranul cu matricea în care se afla pixeli de 0 și 1 folosiți pentru determinarea modului de a colora pătratul. Cand valoarea este 1 pixelul este colorat, iar cand ea este 0 va ramane culoarea background-ului.

# 3. CAPITOLUL III: MANUAL DE ÎNTREȚINERE ȘI UTILIZARE

## 3.1. Basys 3

Utilizarea Controller-ului VGA presupune folosirea unei plăcuțe FPGA Basys 3.



Modul de comunicare al interfeței cu exteriorul se realizează prin:

1. *Inputuri*:

**UP - input button** pentru deplasarea imaginii pe axa Oy(în sus), este pe 1 bit;

**DOWN - input button** pentru deplasarea imaginii pe axa Oy(în jos), este pe 1 bit;

**LEFT - input button** pentru deplasarea imaginii pe axa Ox(spre stânga), este pe 1 bit;

**RIGHT - input button** pentru deplasarea imaginii pe axa Oy(spre dreapta), este pe 1 bit;

**CLK - clock**;

**RST - input button** care resetează poziția imaginii(în colțul stânga, sus), este pe 1 bit;

**COLOR - 12 switch-uri** pentru personalizarea culorii, este pe 12 biți, câte 4 pentru fiecare culoare(roșu, verde, albastru);

**IMAGE - 2 switch-uri** pentru selectarea imaginii, este pe 2 biți deoarece putem să alegem o imagine din cele 4 disponibile.

1. *Outputuri*:

**HS -** transmite coordonata pe Ox a pixelului, este pe 1 bit;

**VS -** transmite coordonata pe Oy a pixelului, este pe 1 bit;

**R -** transmite pe ecran culoarea pixelului, este pe 4 biți;

**G -** transmite pe ecran culoarea pixelului, este pe 4 biți;

**B -** transmite pe ecran culoarea pixelului, este pe 4 biți.

VGA controller-ul are o multitudine de selecții și intrări pe care utilizatorul le poate folosi.

## 3.2. Instrucțiuni pentru afișare și colorare

### 3.2.1. Instrucțiuni de selectare a imaginii

IMAGE permite selectarea unei imagini pe care dorește să o afișeze pe ecran. Numărul total de imaginii este de 4 și acestea sunt numerotate astfel:

00-pătrat

01-cerc

10-triunghi

11-dinozaur

### 3.2.2. Instrucțiuni de selectare a culorii

COLOR permite modificarea culorii pe baza căreia imaginea va fi afișată. Numărul de culori este unul mare deoarece folosim 12 biti pentru obținerea și combinarea celor 3 culori(roșu, verde și albastru).

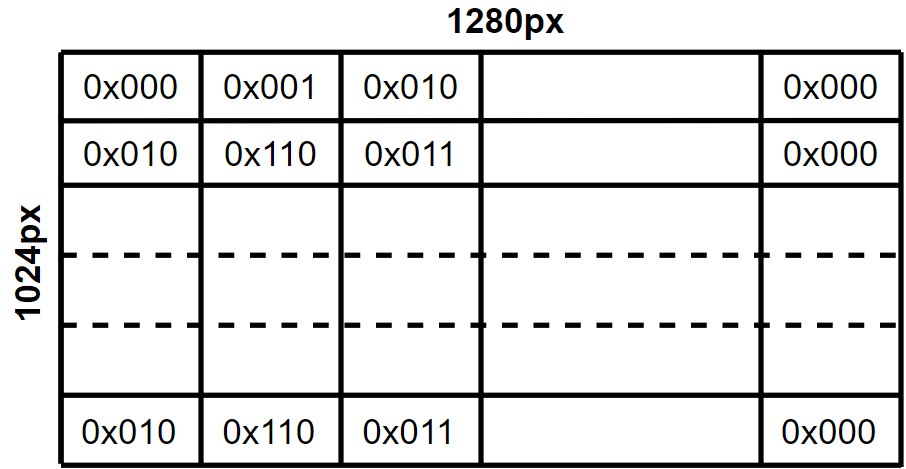
### 3.3. Instrucțiuni pentru mișcarea imaginii

Numele celor 4 butoane este sugesitv pentru direcția de mișcare a obiectului, astfel:

* UP - mișcă imaginea în sus pe axa Oy
* DOWN - mișcă imaginea în jos pe axa Oy
* LEFT - mișcă imaginea la stanga pe axa Ox
* RIGHT - mișcă imaginea în dreapta pe axa Ox
* RESET - resetează poziția imaginii

# 4. CAPITOLUL IV: POSIBILITĂȚI DE DEZVOLTĂRI ULTERIOARE

Una dintre principalele direcții în care acest vga controller se poate îndrepta este afișarea imaginilor, încărcate în prealabil în memorie, sub forma unei matrice. Astfel, matricea poate conține culoarea pixelului respectiv.



De asemenea, se poate implementa un mod în care se să se deseneze pe ecran. Imaginea se poate scrie într-o matrice de culori la fiecare pixel modificat al imaginii sau putem să resetăm doar zona în care se mișcă brush-ul astfel câștigând timp, dar imaginea pierzându-se la final.

# 

# 

# 

# 

# 5. CAPITOLUL V: DISTRIBUIRE SARCINI

### Documentație- Neghină Laurențiu & Tarța Antonia

### VHDL- Neghină Laurențiu & Tarța Antonia

### Triunghi și cerc- Tarța Antonia

### Pătrat și dinozaur- Neghină Laurențiu

# 6. CAPITOLUL VI: BIBLIOGRAFIE

* <http://tinyvga.com/vga-timing/1280x1024@60Hz>
* <https://www.seas.upenn.edu/~ese171/boards/XUPV2P_User_Guide.pdf>
* <https://www.youtube.com/watch?v=4enWoVHCykI>
* <https://www.youtube.com/watch?v=wzhDRIX2Ors>
* <https://www.youtube.com/watch?v=eJMYVLPX0no>
* <https://digilent.com/reference/learn/programmable-logic/tutorials/basys-3-programming-guide/start>
* <https://digilent.com/reference/programmable-logic/basys-3/demos/gpio>
* <https://github.com/Digilent/Basys-3/releases/tag/GPIO/2020.1-1>

### 

### 