MINISTERUL EDUCAȚIEI, CERCETĂRII, TINERETULUI ȘI SPORTULUI
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Catedra / Structura Sistemelor de Calcul

Proiect SSC

Utilizarea portului VGA al placii Basys3 pentru afisarea unor imagini si aplicarea de filtre de image processing

Studenti:

Bora Vlad

Vamvu Denisa

Grupa: 30239

Indrumator proiect:

Ing. Ratiu Vlad

An universitar: 2020-2021

Cuprins

1. Introducere

Acest document prezinta etapele de proiectare si implementare ale proiectului ce presupune afisarea imaginilor pe un monitor folosind un FPGA, ci anume un Basys3 si portul VGA al acestuia.

1.1 Context/Domeniu

Ca si domeniu, proiectul se incadreaza intr-unul hibrid, intre FPGA si procesarea imaginilor. In contextul procesarii de imagini, accelerarea hardware furnizata de un FPGA prin puterea mare de procesare este foarte utila.

1.2 Obiective

Obiectivele acestui proiect sunt:

- realizarea conexiunii dintre o placuta Basys3 si un monitor;
- afisarea unei imagini pe monitor;
- aplicarea unor filtre de Image Processing pe imagine si afisarea acesteia.

1.3 Tabel de acronime

Tabelul 1 prezinta acronimele folosite in acest document alaturi de intelesul acestora.

Tabelul 1 – Tabel de acronime

Acronim	Inteles
FPGA	Field-programmable gate array
ROM	Read Only Memory

2. Studiu bibliografic

[1] "Basys 3 Reference," [Online]. Available: https://reference.digilentinc.com/basys3/refmanual.

[2] J. C. M. M. J. M. R.-A. Carlos Alberto Ramos-Arreguína*, FPGA Open Architecture Design for a VGA Driver, Santiago de Querétaro: Elsevier Ltd, 2012.

[3] "Building a video controller: it's just a pair of counters," 29 Nov 2018. [Online]. Available: http://zipcpu.com/blog/2018/11/29/llvga.html?_ga=2.161355007.319202213.1603032862-645931092.1602168227.

[4] "Display image using VGA from block RAM," [Online]. Available: https://forum.digilentinc.com/topic/17598-display-image-using-vga-from-block-ram/.

[5] "Driving a VGA Monitor Using an FPGA," 29 July 2016. [Online]. Available: https://embeddedthoughts.com/2016/07/29/driving-a-vga-monitor-using-an-fpga/.

[6] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Kernel_(image_processing).

3. Fundamentare teoretica

Pentru inceput, se foloseste portul VGA al placii Basys3, din familia Artix7 a circuitelor FPGA de la Xilinx. S-a folosit documentatia placii [1] pentru a se face conexiunea dintre aceasta si monitor. Aici se prezinta pinii portului VGA si semnalele de control necesare.

Lucrarea [2] prezinta o propunere arhitecturala pentru ca un controller VGA sa fie folosit in sisteme embedded pe FPGA. Acest controller este realizat folosind numai limbajul VHDL. Lucrarea de fata prezinta doua tipuri de teste: afisarea a 8 culori de baza generate folosind valori RGB si afisarea unei imagini stocate intr-o memorie RAM externa, din care FPGA-ul o citeste si o afiseaza pe un monitor LCD.

Articolul [3] prezinta afisarea unei imagini pe ecran folosind portul VGA al placii Basys3 cu ajutorul unui clock generat cu o frecventa de 25MHz si atrage atentia asupra dificultatilor acestui proiect si a riscurilor ce pot surveni prin decompresia gresita a unui bit si crearea unui “garbage output”.

Pe site-ul celor de la DIGILENT, sursa [4] se prezinta afisarea unei imagini cu dimensiunea 300x300 stocate intr-un fisier cu extensia “.coe” si incarcata mai apoi intr-un bloc RAM.

In sursa [5] este explicata logica din spatele controlului unui VGA cu ajutorul FPGA: cele doua numaratoare folosite pentru reprezentarea pe orizontala si pe verticala a ecranului si divizorul de clock (tot un numarator) folosit pentru a se atinge frecventa dorita.

In fotografia digitala, colorimetrie sau in procesarea de imagini, un “grayscale” al unei imagini este o alta imagine in care valoarea fiecarui pixel este un singur esantion reprezentand o cantitate de lumina. Cu alte cuvinte, pixelul poarta doar informatii de intensitate.

Binarizarea unei imagini reprezinta procesul prin care o imagine este convertita din nuante de gri doar in negru si alb. Se efectueaza prin alegerea unei valori de “treshhold” care divizeaza imaginea in doua parti: fundalul si obiectul. Acest procedeu este folosit de obicei pentru extragerea unui obiect dintr-o imagine.

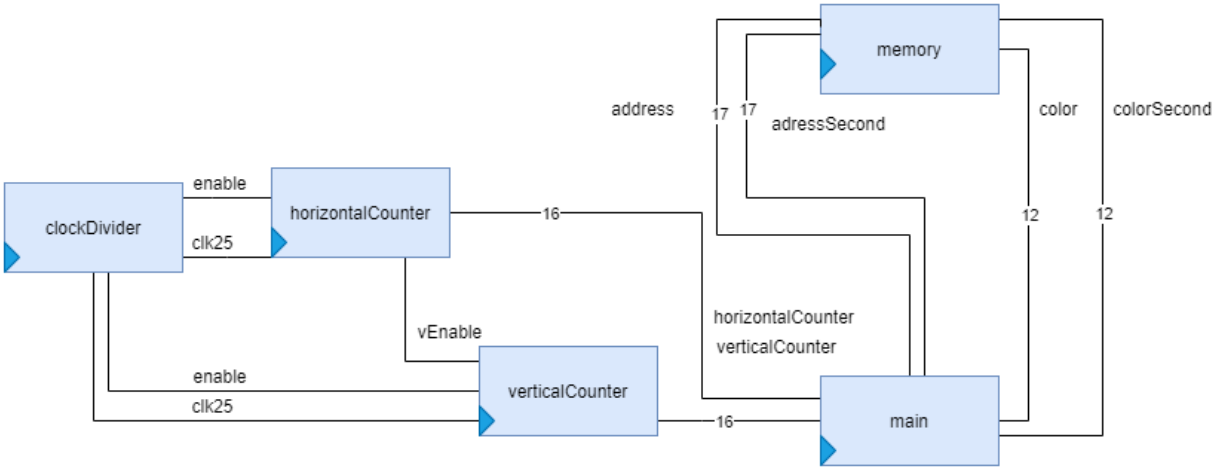
Edge detection este un filtru convolutiv (presupune aplicarea unui “kernel” [6] sau o matrice de convolutie asupra unei imagini), care ajuta la identificarea punctelor in care intensitatea luminoasa se schimba brusc sau are discontinuitati. Aceste puncte sunt practic organizate intr-un set de segmente de dreapta curbate numite “edges” (margini).

4. Proiectare

Proiectul citeste o imagine din memorie si o afiseaza pe un monitor. Asupra ei se aplica trei filtre de image processing: grayscale, binarizare si edge detection.

Pentru afisarea imaginii s-au folosit doua numaratoare care reprezinta suprafata pe care se poate afisa (din monitor) si un divizor de clock pentru a se atinge frecventa de 25 MHZ.

Pentru memorarea imaginii s-a folosit o memorie ROM adresabila pe 17 biti, cu valoarea stocata pe 12 biti la o anumita adresa. In ROM sunt stocate 90000 (pentru o imagine de 300x300) astfel de valori constituind reprezentarea RGB a fiecarui pixel.



Imaginea 1: Arhitectura

“HorizontalCounter” este un numarator care va numara intotdeauna, iar atunci cand isi atinge limita maxima va activa semnalul “enable_v_counter” pe care se bazeaza functionarea numaratorului “VerticalCounter”.

“ClockDivider” primeste un clock cu frecventa de 100MHz si il transforma intr-unul cu frecventa de 25MHz cu factor de umplere de 50. Este generat cu ajutorul tool-ului IPCatalog din mediul de dezvoltare VIvado.

Entitatea “Main” instantiaza toate componentele si imbina logicile necesare afisarii imaginii.

5. Implementare

5.1 Implementare propriu-zisa

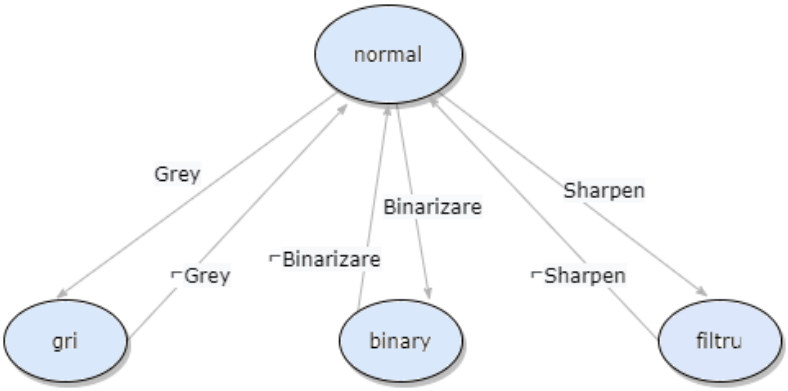
Parcurgerea suprafetei ecranului se face pe coloane, astfel ca atunci cand “HorizontalCounter” a terminat de numarat toare liniile si se reseteaza, se trece pe urmatoarea coloana cu ajutorul “VerticalCounter”. Range-ul initial de numarare este dat de limitele numaratoarelor, 640x480. Ca si implementare, acestea sunt 2 numaratoare care numara pe front crescator, iar atunci cand isi ating limita superioara, se reseteaza. In cazul primului, atingerea limitei superioare necesita activarea celui de-al doilea, lucru ce va fi implementat prin activarea unui enable pentru o perioada de ceas, suficient pentru numaratorul vertical sa faca trecerea la urmatoarea coloana.

Enitatea “Main” este cea mai complexa si detine intreaga logica a proiectului. Are ca si porturi de iesire 3 semnale pe 4 biti, reprezentant colurile RGB necesare portului VGA, si de asemenea semnalele Hsync si Vsync necesare pentru monitor. Semnalele “vgaRed”, “vgaGreen” si “vgaBlue” vor pimi cate 4 biti din vectorul de iesire care va avea valori in functie de ce stare este activa, atunci cand numaratoarele se afla pe domeniul stabilit de cateva constante reprezentand dimensiunea imaginii, altfel vor primi valoarea 0 (negru).

Se instantiaza toate componentele. De mentionat faptul ca si memoria ROM este o componenta oferita tot de IPCatalog, iar aceasta are 2 porturi de iesire (portA si portB) ce contin adresa si valoarea, care au fost folosite la implementare.

Cand “vcounter” este 0, se initializeaza vectorii folositi pentru filtrele de image processing. Cand valoarea acestuia este 1, se pregatesc “adress” si “adressSecond” pentru functionarea normala a programului. Dupa ce se parcurge coloana curenta, vectorii de prelucrare isi schimba valorile in felul urmator: trecut <- curent, curent <- viitor, viitor <- ce se va citi in continuare. Afisarea pe ecran funtioneaza cat timp “hCounter” si “vCounter” se afla in range-ul normal, cu o incrementare a adreselor. Se tine minte un index, cu ajutorul caruia se va adresa fiecare linie din cele 300.

Pentru afisarea diferitelor filtre s-a folosit unitatea de comanda cu diagrama de stare descrisa in imaginea de mai jos. Dupa cum se poate observa, o singura stare poate fi activa la un moment dat, iar pentru a trece din una in alta este necesara revenirea in starea initiala, cand imaginea este nealterata. Semnalele “Grey”, “Binarizare” si “Sharpen” sunt semnale de intrare si vor fi transmise de utilizator prin intermediul a 3 switch-uri.



Imaginea 2: Diagrama de stare

Pentru implementarea filtrului grayscale s-au folosit 5 variabile care stocheaza culoarea urmatorilor pixeli: pixelul curent, si vecinii sai, dreapta, stanga, sus, jos. Metoda de grayscale folosita este “Simple averaging” constand in media aritmetica a valorilor Red, Green, Blue. S-au implementat de asemenea cazurile speciale corespunzatoare primei si ultimei linii, cat si primei si ultimei coloane, punandu-se pe vecinul care “nu exista” efectiv culoarea pixelului curent dupa efectuarea de grayscale.

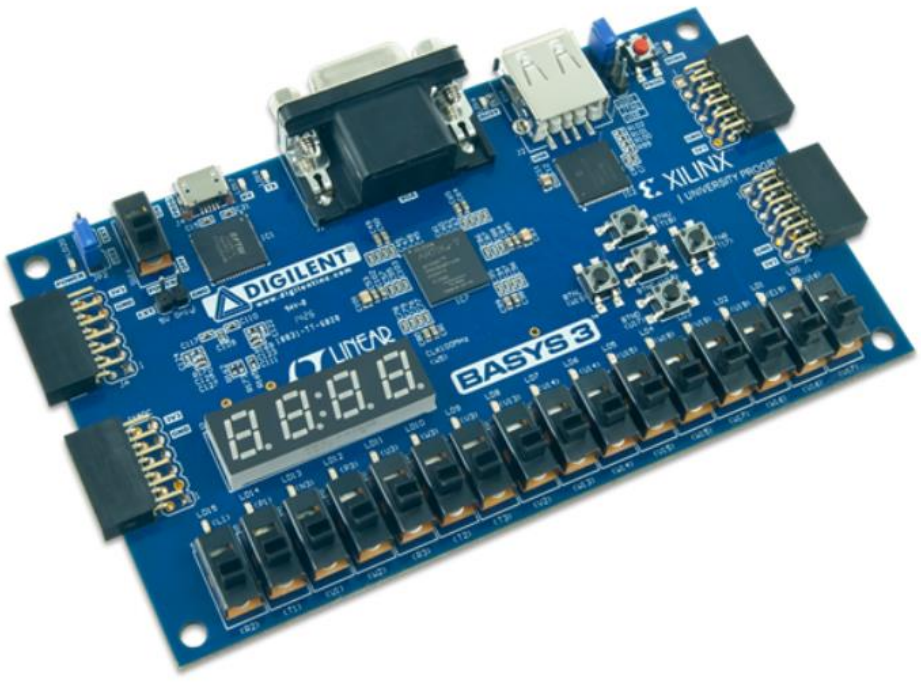
Filtrul de binarizare se executa asupra valorilor grayscalate, setandu-se un threshold intre alb si negru.

Filtrul de edge detection este aplicat asupra imaginii in format binarizat cu ajutorul urmatorului kernel: $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$.

In functie de starea in care se afla automatul, valoarea iesirii va fi selectata din cele 4 tipuri: culoare, grayscale, binarizare sau edge detection.

5.2 Echipament utilizat

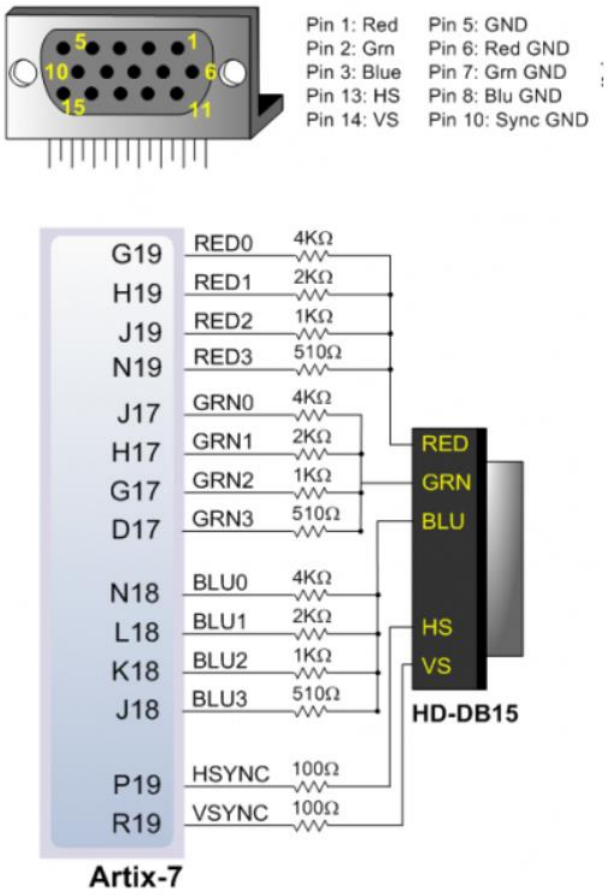
Placa Basys3 este o platforma completa de dezvoltare a circuitelor digitale bazata pe cea mai recenta versiune a familiei Artix 7 de FPGA a firmei Xilinx. Contine 16 switch-uri, 16 led-uri, 5 butoane, un afisor BCD 7 segmente cu 4 cifre, 3 porturi pentru Pmod, iar necesar pentru implementarea acestui proiect, un port VGA pentru output, de 12 biti.



Imaginea 3: Placuta Basys3

Basys3 foloseste 14 semnale FPGA pentru a crea un port VGA continand 4 biti pentru fiecare culoare si doua semnale de sincronizare (HS – Horizontal Sync, and VS – Vertical Sync). Semnalele de culoare folosesc divizoare de rezistenta pentru a crea 16 nivele de semnal pe fiecare dintre semnalele VGA destinate culorilor rosu, verde si albastru.

Circuitul prezentat in figura de mai jos permite afisarea a 4096 de culori, cate una pentru fiecare combinatie de 12 biti. Pentru a produce un sistem de afisare functional, in FPGA trebuie creat un circuit "video controller" pentru a comanda semnalele de sincronizare si pe cele de culoare.



Imaginea 4: Portul VGA al placii Basys3

La acest port va fi conectat cu ajutorul unui cablu VGA, un monitor.

Imaginea 5: Schema de montaj a proiectului

5.3 Software utilizat

Pentru sinteza si analiza design-ului s-a folosit Vivado Design Suite.

Pentru generarea fisierului cu extensia .coe ce va reprezenta imaginea incarcata in memorie, s-a folosit MatLab.

6. Manual de utilizare

Se conecteaza un monitor cu ajutorul unui cablu VGA la o placuta Basys3.

Preincarcat in memoria ROM se afla fisierul cu extensia ".coe" care reprezinta imaginea.

In Vivado, se programeaza placuta cu ajutorul fisierului ".bit" care se gaseste in folderul proiectului.

Initial, pe ecran se va afisa imaginea.

Daca se porneste switch-ul cu numarul 0, primul din stanga, pe ecran va fi afisata imaginea dupa aplicarea filtrului "grayscale".

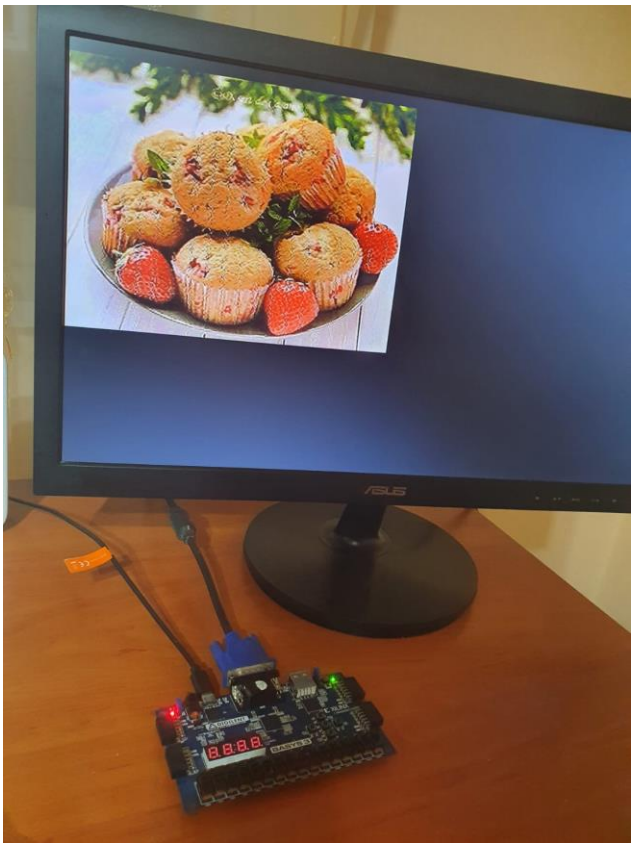
Daca se porneste switch-ul cu numarul 1, imaginea rezultata va reprezenta binarizarea imaginii initiale.

Pentru pornirea switch-ului numarul 2, rezultatul va fi aplicarea unui "edge detection".

Nota: mereu trebuie sa fie pornit un singur switch, deci, pentru afisarea unui alt filtru trebuie inchis switch-ul care este activ, si deschis cel care este dorit.

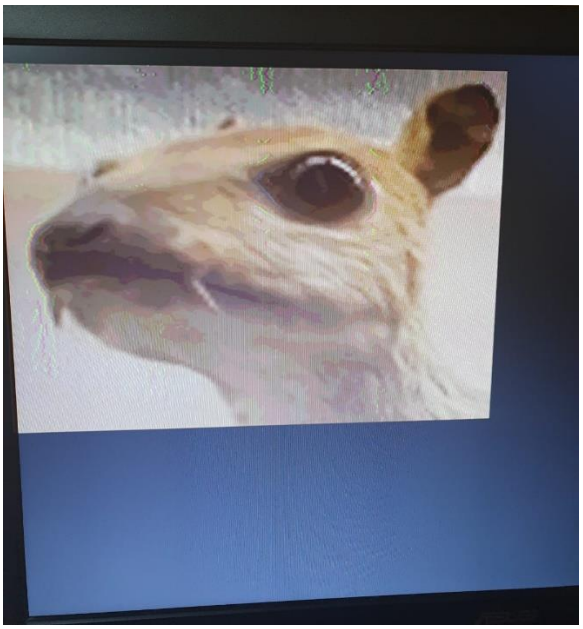
7. Rezultate

Prima testare, afisarea unei imagini, se vede in imaginea 4.

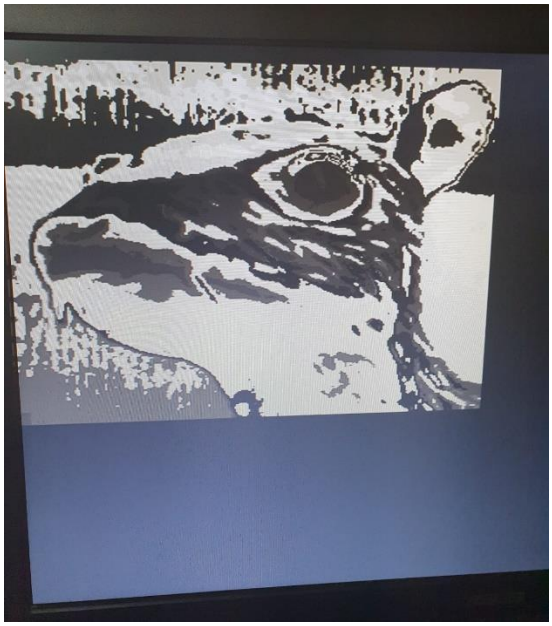


Imaginea 6: Primul test

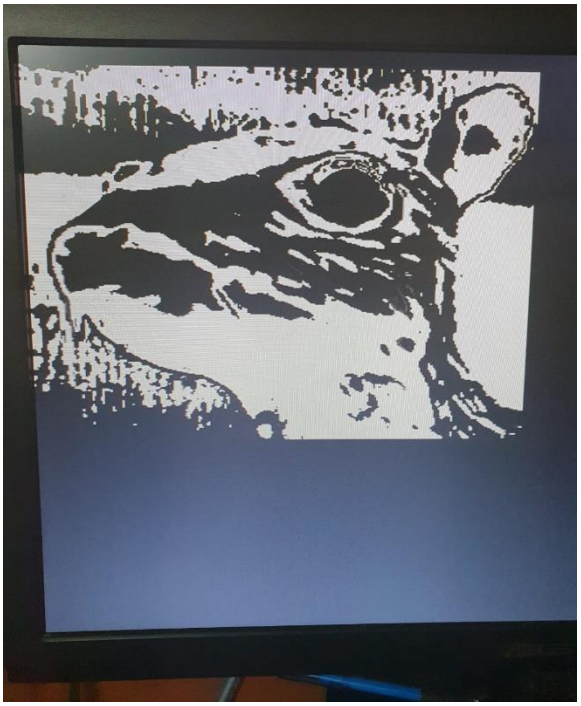
Dupa mai multe teste, pentru ca filtrele sa fie mai usor observabile, s-a decis trecerea la alta imagine. Rezultatele sunt prezentate in imaginile 5, 6, 7 si 8.



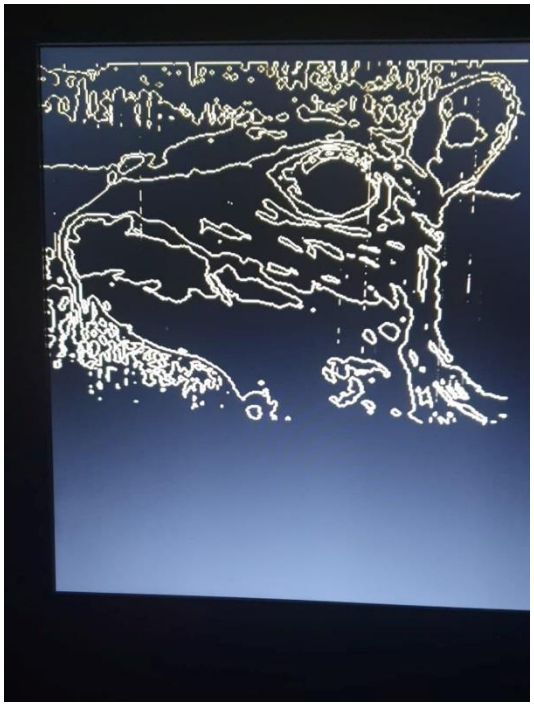
Imaginea 7: Afisarea imaginii



Imaginea 8: Aplicarea filtrului grayscale



Imaginea 9: Aplicarea filtrului de binarizare



Imaginea 10: Edge detection pe imagine

In urma testelor si a simularilor efectuate s-a constatat ca un filtru de imagine cum ar fi “sharpen” nu este tocmai potrivit situatiei din urmatorul motiv: spectrul de culori este prea mic pentru ca acesta sa fie vizibil. Din aceasta cauza s-a ales implementarea unui edge detection pe imaginea rezultata dupa un grayscale.

8. Concluzii

S-a reusit afisarea unei imagini si aplicarea filtrelor de grayscale, binarizare si edge detection. S-a observat eficienta utilizarii unui FPGA intr-un astfel de context.

8.1 Dezvoltari ulterioare

S-ar putea implementa mai multe filtre convolutive precum blur sau sharpen.

De asemenea, s-ar putea implementa prelucrarea in paralel a mai multor pixeli (sau unui grup mai mare de pixeli, in functie de problema), iar o varianta considerata este cea a prelucrarii pixelilor de pe pozitii pare intr-un ciclu de ceas si a celor de pe pozitii impare in alt ciclu de ceas.