Department of Computer Science
Technical University of Cluj-Napoca

Activitate de practica

Nume:Trif Gheorghe Andrei
Grupa:30235

Perioada:28.07.2020—01.09.2020
Profesor indurmator:Lisman Dragos-Florin

Sinteza

In aceasta perioada de practica am incercat sa implementez un proiect,iar la indrumarea domnului profesor am ales un proiect pe care pot sa il pot dezvolta foarte mult ca sa il pot folosi in viitor ca si proiect de licenta.Fiind un proiect de inceput pentru licenta am ales sa combin diferite domenii astfel incat sa rezulte un proiect pe care sa il pot extinde suficient de mult.Astfel ca idee de baza am ales sa folosesc un fpga si diferitele module periferice pe care acesta le poate utiliza pentru a putea aplica diferitii algoritmi de procesare al imaginilor.

Idea de baza de la care am pornit a fost ca avand un monitor,o placa de dezvoltare FPGA (Nexys 4 DDR) si o camera VGA(Camera OV7670) , sa incerc sa le conectez astfel incat sa imi apara imaginea pe monitor in timp real.Am inceput prin a cauta modul posibil de a conecta camera la placuta si prin a citi datasheet-ul camerei.Din datasheet am aflat ce semnale foloseste si cum trebuie sa le configurez.Astfel odata ce am inteles rolul semnalelor si cum pot sa le conectez cu placa Nexys 4 DDR am inceput prin a crea diagrama bloc generala a sistemului si a stabili modulele de care o sa aibe nevoie acest sistem.Practic acesta diagrama se imparte in 3 mari module lasand la o parte micile module (Divizorul de frecventa ,Circutul de debounce) care ajuta ca sitemul sa functioneze corect.Cele trei mari module sunt descrise un paginile ce urmeaza : blocul de configurare al camerei,blocul de capturare al imaginii si blocul video .

In pasul urmator ,dupa ce logica sistemului si componentele au fost stabilite am inceput scrierea si implementarea lor in mediul de dezvoltare Vivado Xilinx cu ajutorul limbajului VHDL.Dupa implementarea celor trei mari parti din acest sistem am continuat cu implementarea si integrarea micilor module impreuna cu cele trei mari blocuri.

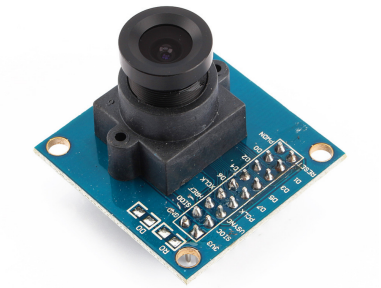
In ultimul pas am incercat sa testez intreaga implementare cu ajutorul testbench urilor deoarece data find situatia nu am avut placuta Nexys 4 DDR ca sa pot face testul fizic.Astfel dupa cateva teste reusit sa verific sistemul si am increderea ca functioneaza si la un viitor test fizic.

In paginile urmatoare sunt exemplificate fiecare componenta a sistemului, precum si functiile pe care le indeplineste.

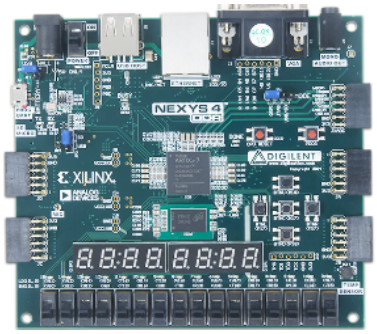
Descrierea activitatilor desfasurate

Am inceput activitatea de practica prin a cauta informatii despre dispositivele pe care o sa le folosec in proiect si despre cum functioneaza .

Componente

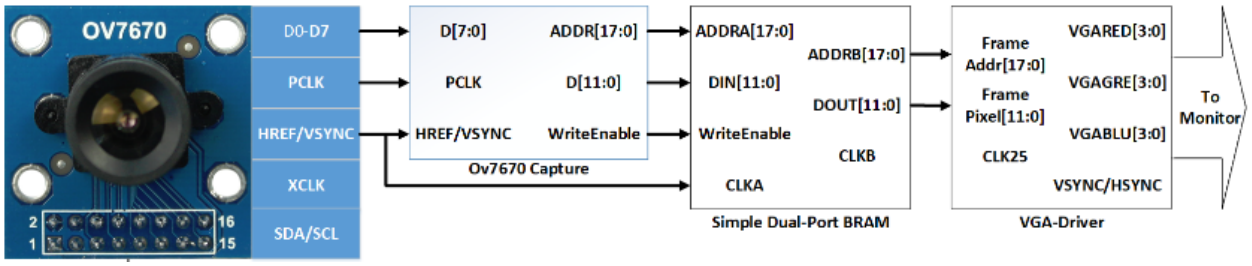


Camera OV7670 este un senzor de imagine CMOS cu un singur procesor care furnizeaza o multime de functii .Aceasta este capabila sa furnizeze pana la 30 de frame-uri pe secunda.



Placa Nexys 4 DDR este o platformă de dezvoltare digitală completă, gata de utilizare, bazată pe cea mai recentă Artray-7 FPGA de la Xilinx. Acest FPGA de mare capacitate prezinta memorii externe generoase și colecție de porturi USB, Ethernet și alte porturi .Nexys4 DDR poate găzdui proiecte care variază de la circuite combinative introductive la procesoare încorporate puternice. Câteva periferice încorporate, inclusiv un accelerometru, senzor de temperatură, microfon digital MEMs, amplificator de difuzoare și mai multe dispozitive de I / O .

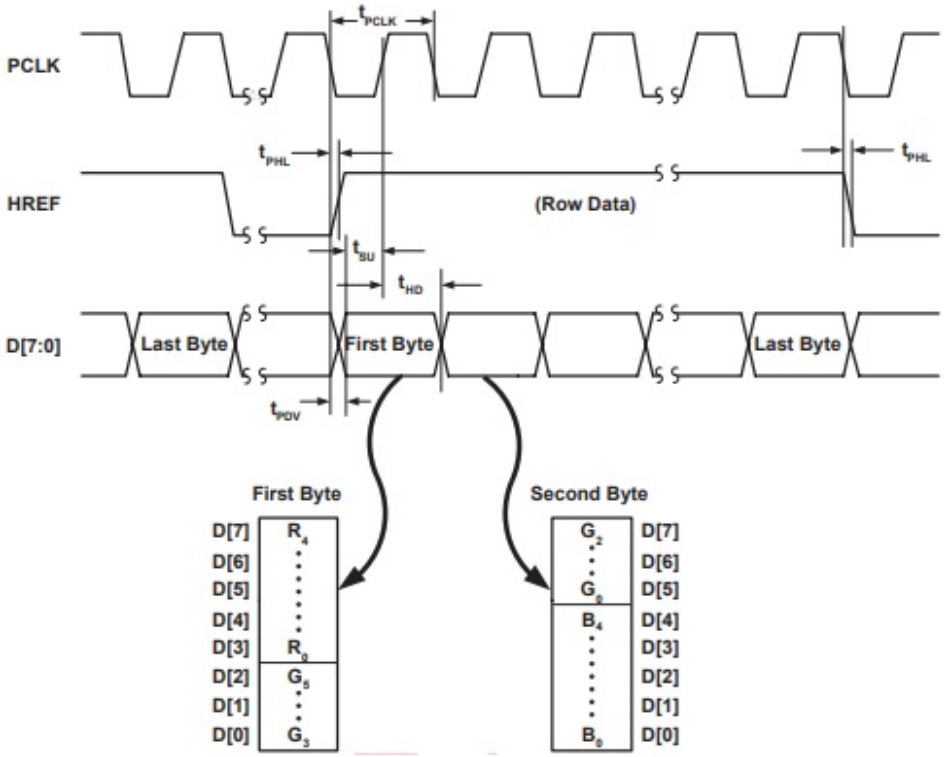
Schema block



Pinii D[7:0] reprezinta output-ul RGB , pinul PCLK reprezinta perioada de ceas pe care o sa se scrie datele pe iesire,iar HREF/VSYNC sunt semnale de sincronizare.

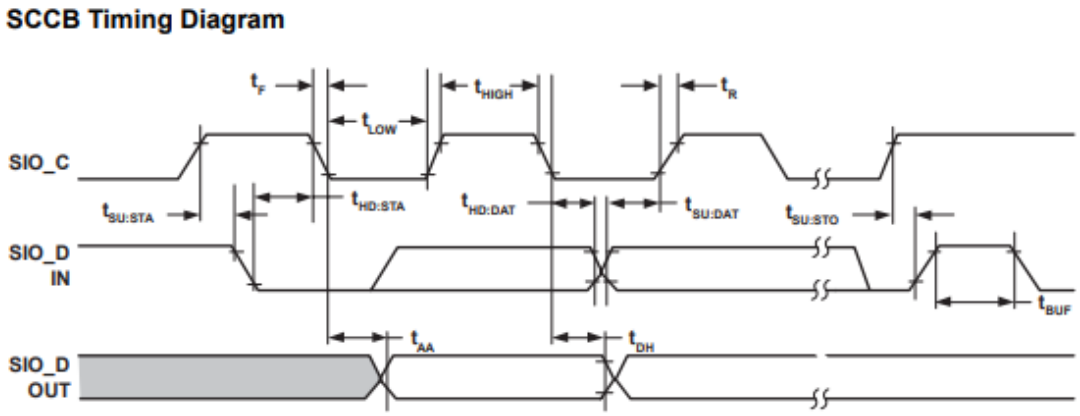
Capturarea imagini

Camera Ov7679 poate transmite datele (pixeli) prin mai multe formate precum GRB(4:2:2),RGB(5:6:5),RGB(5:5:5),YUV(4:2:2).Astfel eu am folosit formatul RGB(5:6:5).Dar camera transmite doar jumate din cei 16 biti intr-un ciclu de ceas PCLK.Astfel pentru a avea datele de iesire am folosit un bloc de bistabile D care salveaza prima jumatate a datelor primite si astepta cea dea doua jumatate astfel in final avand un format RGB(4:4:4) si datele care vor fi scrise in BRAM.



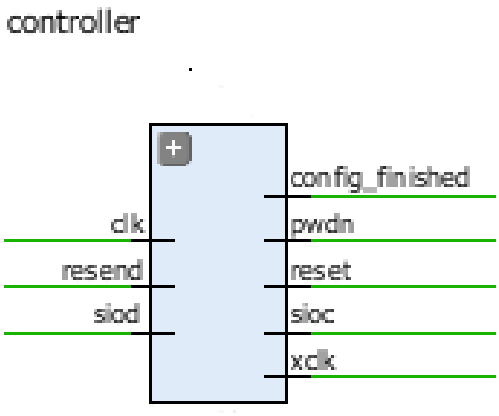
Configurarea camerei

Pentru a putea configura camera primul pas este acela de a determina ce protocol de cumunicatie foloseste.Din cele citite in datasheet am inteles ca aceasta camera foloseste un protocol SCCB care este compatibila cu I2C.Astfel blocul de control este compus din bus-ul masterului I2C si instructiunile de setare ale registrilor camerei, FPGA-ul devineind master ,iar camera slave.



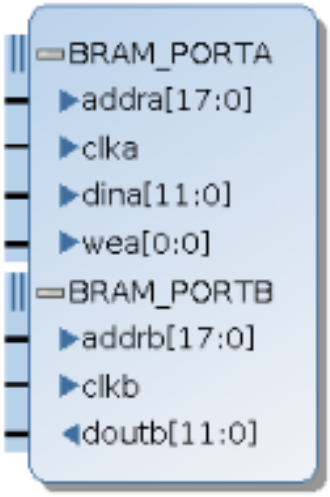
Registri camerei sunt scrisi o singura data la pornirea sistemului.In fisierul registers pentru a putea fi posibil acest lucru am folosit un numarator si o constructie case.Cand numaratorul numara peste ultima valoare din case se ajunge in starea de terminat semnificand ca registri sau configurat .Astfel configurarea se face o singura data.

In cea de-a doua componenta a unitati de control se trimit comenzile spre camera OV7670 printr-un bus asemanator I2C.Aceasta componenta furnizeaza semnalele SI0C(SCCB serial interface input clock) si SIOD(SCCB serial interface data I/O). Astfel controller-ul furnizeaza toate semnalele necesare configurari camerei :SI0C,SIOD,RESET (reinitializeaza toti registri la default),PWDN (Power Down Mode Selection),XCLK(System clock input),Config-finished.

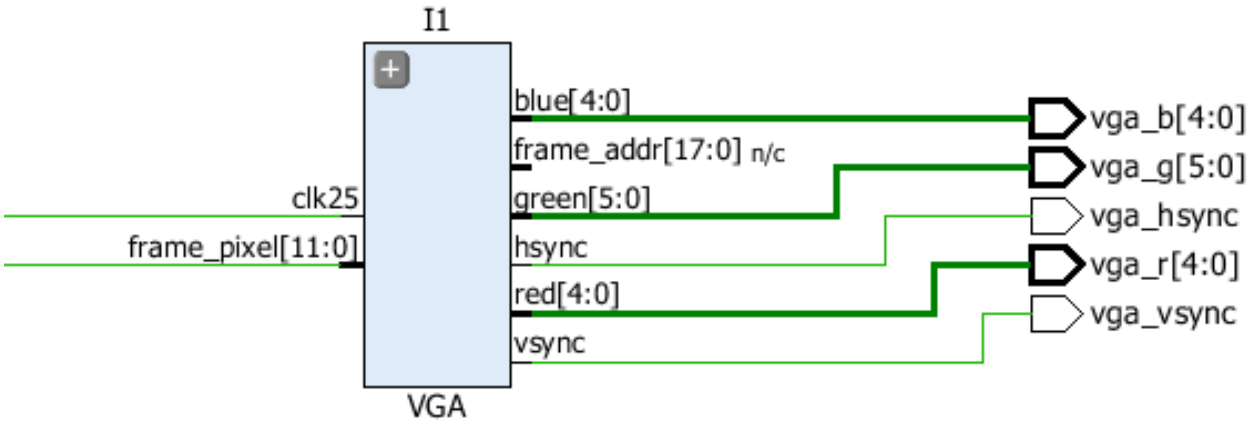


Blocul Video

In blocul video se genereaza semnalele HSYNC si VSYNC.Tot in acest bloc se determina si pixeli imagini sub formatul RGB(5:6:5).Astfel dimensiunea imagini ajunge ajunge la 640x480. Pentru a stoca fiecare frame am folosit o memorie BRAM Dual Port . Din aceasta memorie se poate citi/scrie in acelasi timp un nr de 12 biti,adica aceasta este dimensiunea unui slot din memorie. Aceasta prezinta si 153600 de sloturi.



In imaginea urmatoare este prezentata schema block a unitati video.



Video Block

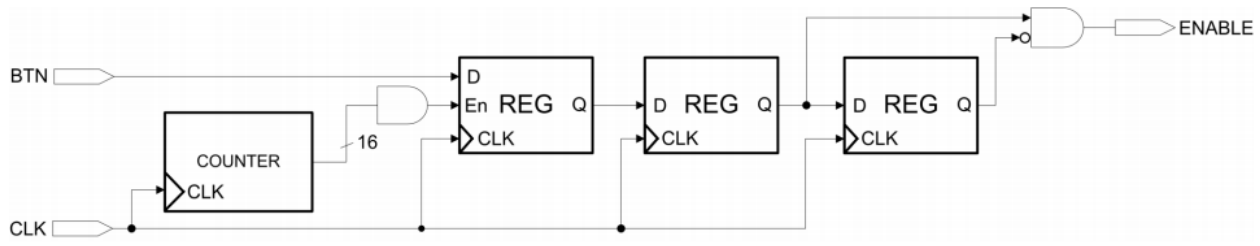
Componente secundare

Pentru ca circuitul sa fie complet am mai adaugat un circuit de debounce si un divizor de frecventa pentru am putea diviza ceasul de 100MHz al placi in 50 MHZ si 25 MHz.

Circuitul de debounce

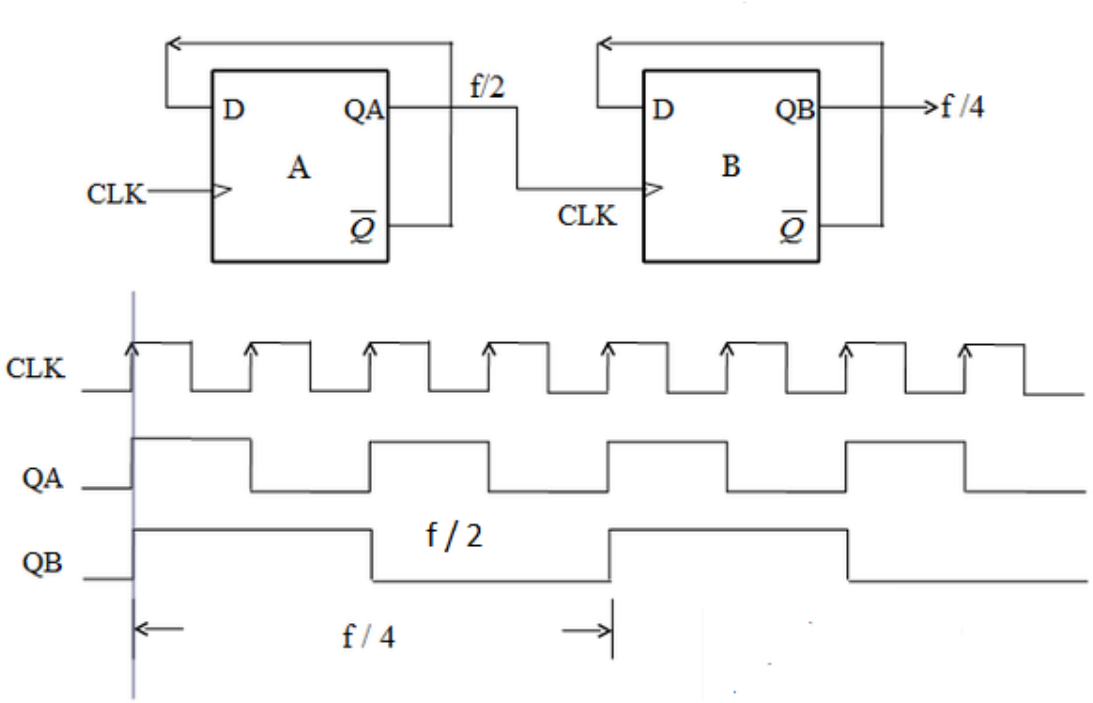
În circuitele secvențiale, orice variabilă de intrare care nu este dependentă de ceasul sistemului, este o intrare asincronă. Majoritatea acestor intrări asincrone provin de la dispozitive mecanice, cum ar fi cele patru butoane de pe placa de dezvoltare. Pe lângă faptul ca sunt asincrone, aceste semnale nu sunt perfect dreptunghiulare, elementele mecanice din interiorul butoanelor avand nevoie de un mic timp pentru a se stabiliza. In acest timp semnalul generat este format din multe spike-uri, care conduc la funcționarea gresită a circuitului . Circuitul de Debounce, care primește la intrare un semnal, asemanator lui in din figura de mai jos și generează la ieșire un semnal perfect dreptunghiular, de durata unei singure perioade de ceas.

Am folosit acest circuit pe pentru ca stabiliza butonul care este conectat la intrarea "resend" a cintrolarului .



Divizorul de frecventa

Acesta este un circuit foarte usor de implementat are o singura intrare si doua iesiri.Intrare este frecventa de ceas de 100 MHZ iar cele doua iesiri sunt frecventele de 50 MHz si 25 MHz .Memoria si controlerul folosesc clock-ul de 50 Mhz iar blocul VGA pe cel de 25 MHz.Acest divizor este construit din doua bistabile D care in care intrarea D a fiecarui bistabil primeste iesire Qn proprie.Astfel frecventa se divide cu 2 la fiecare bistabil.Dupa cum se poate vedea pe schema iesire QA va avea frecventa de 50 MHz , iar iesirea QB va acea frecventa de 25 MHz.



Testarea sistemului

O data ce am implementat toate blocrile enumerate mai sus folosind mediul Vivado am rulat functiile de Synthesis si Implementation pentru a verifica daca codul si comonentele sunt lipsite de erori .Dupa ce am eliminat toate erorile am inceput verificarea semnalelor si iesirilor circuitului.

Pentru a testa sistemul am creat un test-beanch deoarece din cauza conditiilor oferite de aceasta perioada nu am avut placuta Nexys 4 DDR.Astfel am creat un test bench is l-am rulat cu ajutorul mediului Vivado.

Concluzii si Dezvoltarii ulterioare

Conectarea camerei a fost interesanta , a fost un pic greu pana am inteles cum sa iau informatiile din datasheet si cum sa le folosesc in circuitul meu.Find un proiect de inceput al licientei o sa continui dezvoltarea .Ca si faze de dezvoltare ulterioare o sa incerc sa adaug un modul care proceseaza cod c/c++ deoarece a vrea sa aplic pe imaginea in real time diferite filtre/algoritmi de procesare al imaginilor.

Modulul de procesare al imaginilor o sa fie patrea ce mai interesanta deoarece dupa implementarea acestuia ma gandesc sa introduc algoritmi precum: gray-scale image, Sobel operator, Roberts Cross operator, Canny edge si multi alti.Inca nu stiu cum o sa ruleze acesti algoritmi pe o imagine de 640x480 dar o sa incerc sa ii implementez astfel incat sa aibe o acuratete cat mai mare.

Ca o ultima concuzie as vrea sa prezint o mica intrebuintarea a unui astfel de sistem.Vazand situati din acest an si ne putand face materia de procesarea a imaginilor la facultate un astfel de sistem poate fi util prin a arata studentilor cum ar trebui sa se comporte un algoritm de procesare al imaginilor precum Canny edge in mod normal , deoarece in unele cazuri mi-a fost dificil sa imi dau seama daca un algoritm de procesare al imaginilor are rezultatul dorit.