

UNIVERSITATEA TEHNICA CLUJ-NAPOCA

FACULTATEA DE AUTOMATICA SI CALCULATOARE

---

# Osciloscop Digital

---

*Autori:*

Alexandru PANA

Adrian SOUCUP

grupa 30232

*Indrumator:*

Cristi MOCAN

10 iunie 2013

# Cuprins

<b>1</b>	<b>Rezumat</b>	<b>2</b>
<b>2</b>	<b>Introducere</b>	<b>3</b>
2.1	Osciloscopul Digital . . . . .	3
2.2	Tema Proiectului . . . . .	4
2.3	Continutul si Structura Documentului . . . . .	5
<b>3</b>	<b>Fundamentare teoretica</b>	<b>6</b>
3.1	Semnalul video VGA . . . . .	6
<b>4</b>	<b>Proiectare si implementare</b>	<b>8</b>
4.1	Blocul de esantionare . . . . .	9
4.2	Logica de iesire . . . . .	13
<b>5</b>	<b>Rezultate experimentale</b>	<b>17</b>
<b>6</b>	<b>Concluzii</b>	<b>18</b>

# 1 Rezumat

Proiectul de fata isi propune implementarea unui osciloscop digital, folosind o placa de dezvoltare Basys2. Implementarea se va face in limbajul VHDL.

In acest document exploram notiunea de osciloscop digital, plecand de la definitie, analizand modul de functionare dar si modul de folosire. Vom analiza componentele necesare implementarii unui osciloscop digital: logica de esantionare, o memorie de esantioane si o logica de iesire. Fiecare componenta este explicata din punct de vedere al functionarii, dupa care se prezinta modul de implementare al acesteia specific pentru placa de dezvoltare Basys2. Pentru logica de esantionare vom explora utilizarea unui convertor analog digital si logica ce controleaza modul de esantionare. Pentru logica de iesire vom explora portul VGA, ce este acesta, cum functioneaza, si cum este implementat pe placa Basys2.

In descrierea implementarii proiectului avem in considerare tehnicitatile specifice placutei de dezvoltare cum ar fi constrangerile, limitarile dar si avantajele acesteia. Vom urmari schema osciloscopului si modul de functionare al acestuia.

In sectiunea de rezultate experimentale se analizeaza functionearea proiectului atat in mediu simulat dar si in mediu real. Se pune in functiune placa Basys2 si se descriu rezultatele obtinute. Scopul aceste sectiuni este de a demonstra corectitudinea implementarii si functionarii proiectului.

Documentul se incheie cu concluziile autorilor, in care acestia prezinta experienta lor pe parcursul dezvoltarii proiectului, notiunile si lectiile invatate, dar si ganduri despre ce ar putea fi imbunatatit sau adaugat.

## 2 Introducere

### 2.1 Osciloscopul Digital

**Osciloscopul** este un aparat electronic de masura ce permite vizualizarea formei unui semnal de voltaj in timp. Osciloscopul este folosit pentru a analiza forma de unda descrisa de un semnal repetitiv. Aceasta vizualizare ofera informatii despre perioada semnalului, amplitudinea acestuia, dar si despre modificari in amplitudine sau perioada. O alta intrebuintare a osciloscoapelor este analiza zgomotului unui semnal. Fiind aparate de masura ce ofera o reprezentare vizuala detaliata a unui semnal, acestea joaca un rol important in analiza si depanarea circuitelor electrice.



Figura 1: Osciloscop Digital

Un osciloscop digital este format din patru componente: un ecran pentru vizualizarea semnalului impreuna cu 3 panouri de configuratie ce controleaza dimensiunea amplitudinii semnalului pe monitor, frecventa de esantionare a semnalului, si modul de declansare a esantionarii. Unele osciloscoape digitale ofera posibilitatea masurarii a doua semnale concurrent. Aceasta functionalitate este utila daca se urmareste comparatia a doua semnale. De asemenea se pot analiza si semnale digitale daca semnalul analogic de intrare este de asa natura.

**Ecranul de vizualizare** este, de cele mai multe ori, un mic monitor CRT integrat in osciloscop. Pe acest monitor se afiseaza semnalul analizat, impreuna cu alte informatii suplimentare cum ar fi o grila de referinta, scala la care este reprezentat semnalul, axele X si Y, sau alti indicatori.

**Controlul amplitudinii** este factorul ce determina conversia din valoarea semnalului analogic ( in volti ) in spatiul de afisare al ecranului ( in pixeli ). Folosind acest control putem modifica scala la care este afisat semnalul pe axa Y.

**Controlul frecvenței de esantionare** ne permite să modificăm frecvența la care osciloscopul citește semnalul analog. Cu cât frecvența este mai mare, cu atât reconstruirea semnalului pe ecranul de vizualizare este mai fidelă. Unele osciloscoape digitale dispun și de un frecventmetru ce le oferă abilitatea de recunoaștere a perioadei unui semnal. Pentru ca perioada calculată să coincidă cu cea reală, adică să nu apară problema de aliasing, frecvența de esantionare trebuie să fie suficient de mare.

**Modul de declansare** a procesului de esantionare determină dacă semnalul este esantionat în mod continuu, o singură dată sau ca răspuns la un semnal extern. Dacă esantionarea se face în mod continuu, imaginea osciloscopului se va schimba constant pentru a reflecta schimbările semnalului de intrare. Dacă esantionarea se face o singură dată, monitorul osciloscopului va afișa o imagine statică a semnalului pe o perioadă de timp. Dacă se alege folosirea unui declansator extern, semnalul va fi esantionat doar la evenimente generate de semnalul declansator.

## 2.2 Tema Proiectului

Proiectul de față își propune implementarea unui osciloscop digital într-un limbaj de descriere hardware, pentru a fi sintetizat și rulat pe o placă FPGA.

Analizând scopul și modul de funcționare a unui osciloscop, se pot observa două responsabilități importante ale acestuia:

- a. **Esantionarea semnalului** presupune citirea valorii semnalului analogic la intervale regulate, conversia acestora în valori digitale și salvarea acestor esantioane într-o memorie internă. Vom implementa această responsabilitate într-un bloc independent. Acest bloc va avea ca intrări semnalul analog de esantionat, împreună cu valorile de control pentru *frecvența de esantionare* și *modul de declansare*. La ieșirea acestui bloc se vor trimite pe rând esantioanele citite.

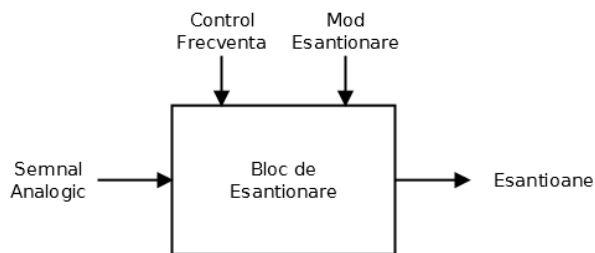


Figura 2: Blocul de esantionare

- a. **Afisarea semnalului** presupune reprezentarea esantioanelor prezente în memoria internă pe un ecran. Afisarea poate fi implementată independent de modulul de esantionare, comunicând cu acesta printr-o memorie de esantioane. Blocul de afisare va avea ca intrare memoria de esantionari, iar ca ieșire un semnal VGA ce poate fi trimis către orice monitor.

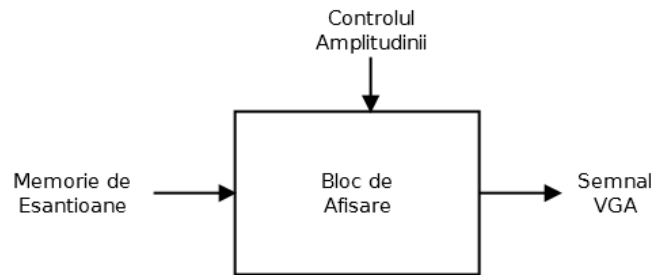


Figura 3: Blocul de afisare

Cele doua blocuri vor fi implementate separat si vor comunica sincron printr-o memorie de esantioane. Aceasta memorie va contine suficiente esantioane ca sa se poata construi forma de unda pe ecran. Detaliile implementari pot fi gasite in sectiunea urmatoare a documentatiei.

### 2.3 Continutul si Structura Documentului

Acest document este impartit in sase sectiuni, fiecare completand si continuand sectiunea anterioara. Aceste sectiuni sunt:

1. **Rezumat** Aceasta sectiune contine o sinteza scurta a continutului intregului document. Aici sunt prezentate ideile principale din fiecare sectiune care urmeaza.
2. **Introducere** Introducerea are rolul de a plasa problema proiectului in context, de a prezenta scopul proiectului si pozitia acestuia in domeniul studiat. Se analizeaza utilitatea proiectului luand in considerare constrangerile platformei pe care este implementat si timpul de dezvoltare al acestuia.
3. **Fundamente Teoretice** In sectiunea de fundamente teoretice sunt explicate notiunile de baza ale domeniului studiat (analiza semnalelor analogice periodice), implementari similare si o descriere a solutiei propuse. Sunt amintite referintele bibliografice din care s-au documentat autorii.
4. **Proiectare si Implementare** Aceasta sectiune descrie pe larg structura si implementarea proiectului pe baza notiunilor teoretice prezentate anterior. Este prezentata platforma hardware folosita, impreuna cu deciziile pe care autorii le-au luat la implementarea proiectului.
5. **Rezultate Experimentale** Aceasta sectiune are rolul de a demonstra corectitudinea implementarii prin exemplificarea functionalitatii. Sunt analizate rezultatele proiectului prin simulare asistata de calculator dar si prin implementarea si rularea sa pe suportul hardware.
6. **Concluzii** Sectiunea de concluzii cuprinde propriile concluzii ale autorilor, experienta dobandita in urma implementarii proiectului dar si o analiza asupra procesului de implementare si produsul realizat.

### 3 Fundamentare teoretica

Osciloscopul este un dispozitiv electronic de testare ce permite observarea unor semnale ce variaza constant sub forma unui grafic bidimensional. Graficul foloseste axa Y pentru a reprezenta valoarea semnalului in functie de timp.

### 3.1 Semnalul video VGA

Acesta este semnalul de la iesirea osciloscopului. Folosind un semnal VGA, osciloscopul construit in acest proiect poate fi conectat la orice monitor, si nu necesita unul incorporat. Semnalul VGA este usor de generat, iar majoritatea monitoarelor existente acum au conectori compatibili. In plus, construirea unui monitor integrat care sa afiseze semnalul este in afara scopului acestui proiect. Folosind un monitor extern osciloscopul nostru beneficiaza de o flexibilitate ridicata.

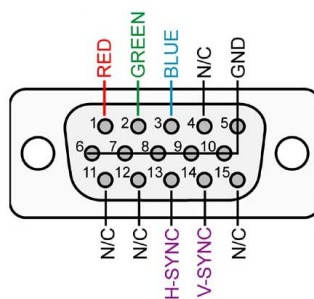


Figura 4: Conectorul VGA

Conectorul VGA utilizeaza 5 pini, dintre care 3 pentru culoare ( Rosu, Verde si Albastru ), impreuna cu 2 de sincronizare care reseteaza pozitia fascicolului de electroni pe verticala, respectiv orizontala. Valoarea tensiunii dintre fiecare pin pentru culoare si valoarea tensiunii pinului GND va determina intensitatea canalului respectiv. Pentru a intelege modul de utilizare al pinilor *H-SYNC* si *V-SYNC* trebuie sa intelegem mai intai modul de functionare a fascicolului de electroni in interiorul monitorului.

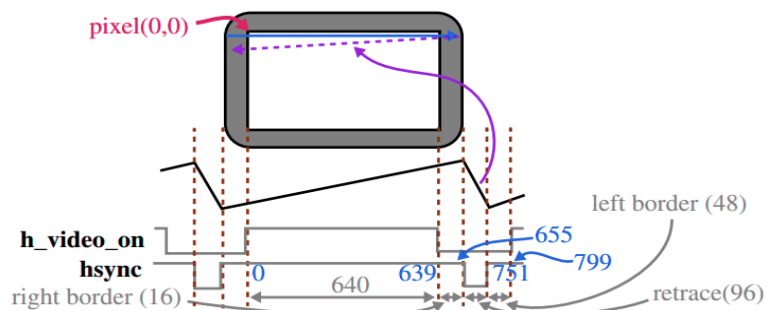


Figura 5: Timprii de sincronizare VGA

Fascicolul de electroni se misca cu viteza constanta de la stanga monitorului spre dreapta. Se observa totusi ca pozitia de inceput a acestuia se afla inaintea zonei vizuale. Cand fascicolul de electroni se afla in stanga zonei vizuale, spunem ca se afla la marginea stanga. Definim in mod similar zona marginii drepte. Semnalul **h\_video\_on** din Figura 5 are valoarea *1* logic cand fascicolul de electroni se afla in zona vizuala, respectiv *0* logic cand acesta se afla fie la marginea stanga, fie la marginea dreapta. Cand fascicolul a ajuns la capatul marginii drepte, se activeaza semnalul *H-SYNC* pentru a incepe resetarea acestuia la marginea dreapta. In concluzie, fascicolul de electroni se poate afla in una din starile urmatoare:

1. **marginea stanga** atunci cand se afla in stanga zonei vizuale
2. **zona vizuala** cand fascicolul pargurge suprafata monitorului
3. **marginea dreapta** cand fascicolul a depasit suprafata monitorului
4. **curs de intoarcere** cand fascicolul este resetat la inceput

Semnalul *H-SYNC* trebuie sa fie tinut la valoarea *0* in perioada a patra, cand fascicolul este in curs de intoarcere. In functie de rezolutia dorita, si frecventa monitorului, fiecare din aceste stari are o perioada fixa de timp. Semnalul *V-SYNC* are acelasi rol, dar reseteaza fascicolul pe verticala. Deoarece ne vom referi la zonele din afara monitorului atat pe orizontala cat si pe verticala, le vom numi de acum inainte *margine de inceput* si *margine de sfarsit*. Ne vom referi la tabelul urmator pentru informatii despre timpi, considerand o rezolutie de 640x480, si o frecventa de 60Hz:

Stare	Timp Orizontala	Timp Verticala
margine inceput	16	10
zona vizuala	640	480
margine sfarsit	48	33
intoarcere	96	2
total	800	525

Timpii sunt calculati in impulsuri de clock, iar frecventa clock-ului pentru un monitor de frecventa 60Hz este 25.175MHz.

Valorile semnalelor *Red*, *Green* si *Blue* se schimba la fiecare impuls de clock cand fascicolul de electroni se afla in zona vizuala, pentru fiecare pixel in parte. Cand fascicolul se afla intr-una din marginile exterioare zonei vizuale sau se afla in starea de intoarcere, aceste semnale sunt setate la nivelul *GND*.



## 4 Proiectare si implementare

Osciloscopul va fi format din 3 parti: Logica de intrare, Memoria esantioanelor si logica de iesire.

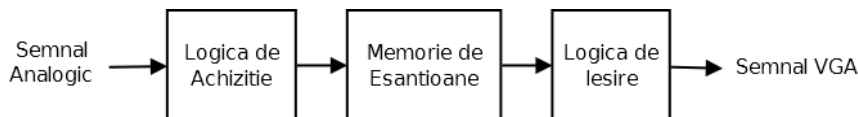


Figura 6: Schema Block

Propunem o implementare in limbaj de descriere hardware a unui osciloscop digital ce poate fi sintetizat pe o placa **FPGA Basys 2 (Spartan 3E)**. Pentru esantionarea semnalului de intrare vom folosi un convertor analog digital. Vom realiza un protocol de comunicare intre convertor si placuta FPGA astfel incat sa fie usor de schimbat cu alt convertor de exemplu. Frecventa de esantionare a semnalului o vom controla de la switch-uri sau butoane.

Disponem de o placuta **CEREBOT II**, placuta care are integrat un **convertor AD/10 biti** pe care il putem folosi. Pentru afisarea datelor vom folosi un monitor cu intrare **VGA**. Placuta Basys 2 dispune de o iesire VGA integrata. Controlul osciloscopului se va face cu ajutorul butoanelor si switch-urilor placutei Basys 2.

Ideea de baza a dispozitivului e sa esantioneze un semnal la un interval de timp modificabil si sa salveze pentru fiecare perioada de timp o valoare care reprezinta amplitudinea semnalului de intrare. Dimensiunea intervalului de timp monitorizat este prestabilit sau se poate controla de la switch-uri/butoane. De fiecare data cand intervalul este baleiat vom trimite datele spre controller-ul VGA spre a fi afisate.

Pentru afisarea graficului putem utiliza interpolare spline liniara intre valorile succesive ale amplitudinii semnalului. Cel mai probabil vom avea nevoie de o matrice bidimensionala de pixeli (framebuffer) in care vom desena forma de unda a semnalului de intrare.

## 4.1 Blocul de esantionare

Blocul de esantionare este o componenta logica care citeste valorile tensiunii semnalului de intrare la un anumit interval de timp si salveaza in memoria video aceste valori.

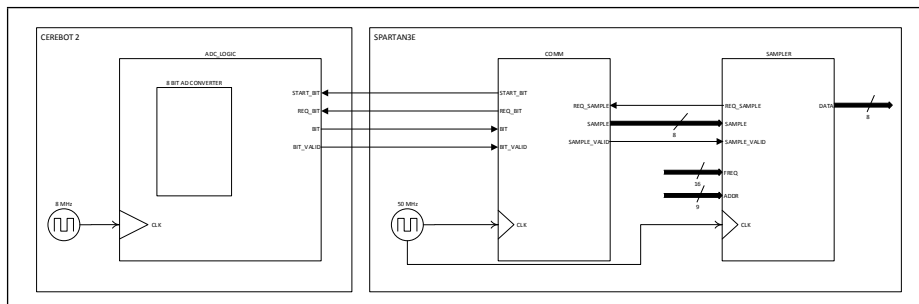


Figura 7: Schema blocului de esantionare cu convertor AD real

Acesta este compus din trei subcomponente majore. Componenta ADC, componenta COMM si componenta SAMPLER. Fiecare din aceste componente este un automat de stare.

Componenta COMM este responsabila cu transmiterea de informatii dintre convertorul analog digital si componenta de esantionare. Protocolul de comunicare este simplu de inteles si robust. Transmisia bitilor se face in mod serial in pachete de cate 8. In modul IDLE liniile START\_BIT si REQ\_BIT sunt tinute pe zero logic. Cand se doreste transmiterea unui pachet de 8 biti, COMM va trimite un impuls de o lungime fixa (in cazul nostru de 50MHz div 64) pe linia START\_BIT. Dupa o anumita perioada COMM va incepe sa trimita impulsuri pe linia REQ\_BIT de aceiasi lungime cu impulsul de pe START\_BIT. Placuta CEREBOT va detecta aceste impulsuri ca intreruperi externe. Dupa ce s-a trimis un impuls pe REQ\_BIT, COMM va intra intr-o stare de asteptare pana cand linia BIT\_VALID va deveni 1 logic. Este obligatoriu ca aceasta linie sa fie tinuta pe 1 logic pana cand se va transmite un alt impuls pe REQ\_BIT. COMM va citi bitul de pe linia BIT intr-un registru de deplasare, iar mai apoi va trimite din nou o cerere REQ\_BIT pana cand registrul de deplasare este plin.

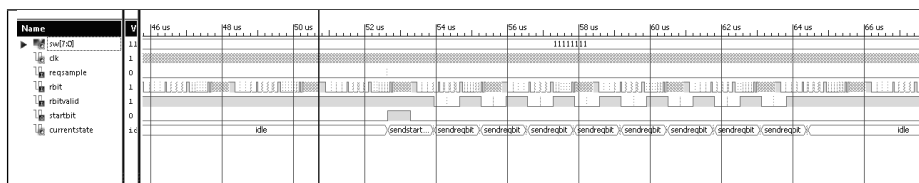


Figura 8: Formele de unda pentru COMM

COMM este un automat de stare. In continuare este prezentata o schema cu tranzitiile intre stari. Starile auxiliare BEGIN\* exista deoarece in ele se va initializa un numarator care va avea rolul de a genera impulsuri de o lungime mai mare decat durata tactului de pe placuta FPGA. Avem nevoie de impulsuri mai lungi deoarece placuta CEREBOT2 nu poate detecta intreruperi care sunt mai scurte decat durata clock-ului propriu (8 MHz).

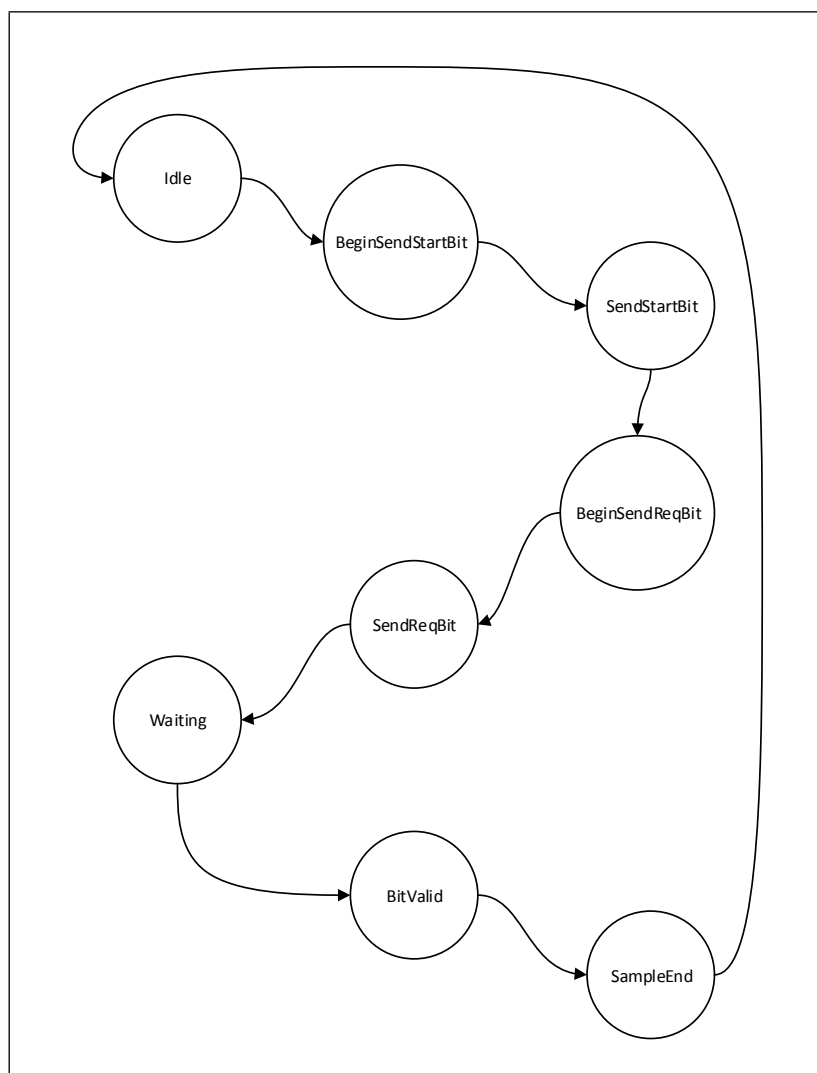


Figura 9: Diagrama cu tranzitii de stare

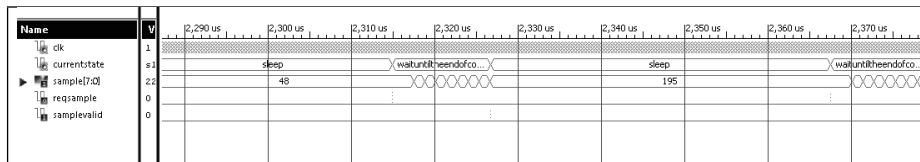
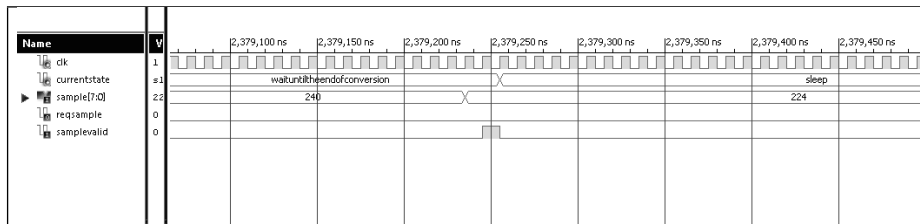
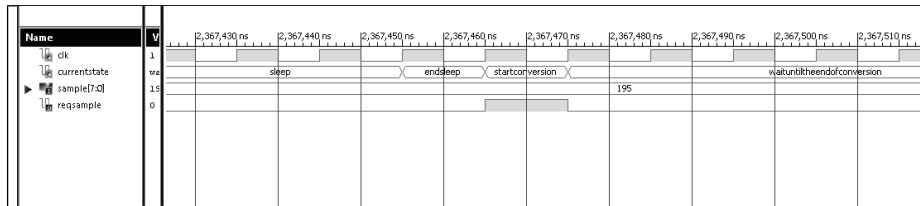
Componenta SAMPLER are rolul de declansa conversia analog digitala cu ajutorul bitului REQ\_SAMPLE si a salva valoarea din magistrala SAMPLE cand bitul SAMPLE\_VALID devine 1 intr-o locatie din memoria interna. Memoria video are dimensiunea de 512 x 8 si in ea se vor regasi toate esantioanele capturate.

Frecventa de esantionare este controlata cu variabila  $FREQ$ . Aceasta este conectata la SWITCH-urile placutei FPGA si reprezinta numarul de CLK-uri de durata 20ns pe care unitatea sa le astepte intre doua esantioane consecutive.

$$\nu = \frac{1}{ConversionTime + 20ns * FREQ}$$

In interiorul componentei se foloseste un registru in care se afla valoarea curenta a adresei de scriere a memoriei. Dupa fiecare esantion capturat valoarea din registru se incrementeaza cu 1, astfel urmatorul sample va fi salvat in continuare in memorie. Exista doua moduri de functionare a osciloscopului: Timp Real si Captura. In modul Timp Real memoria este baleiata si actualizata in mod continuu, in timp ce in modul Captura memoria este actualizata doar atunci cand se detecteaza o schimbare pe portul  $FREQ$ .

### Exemplu de functionare



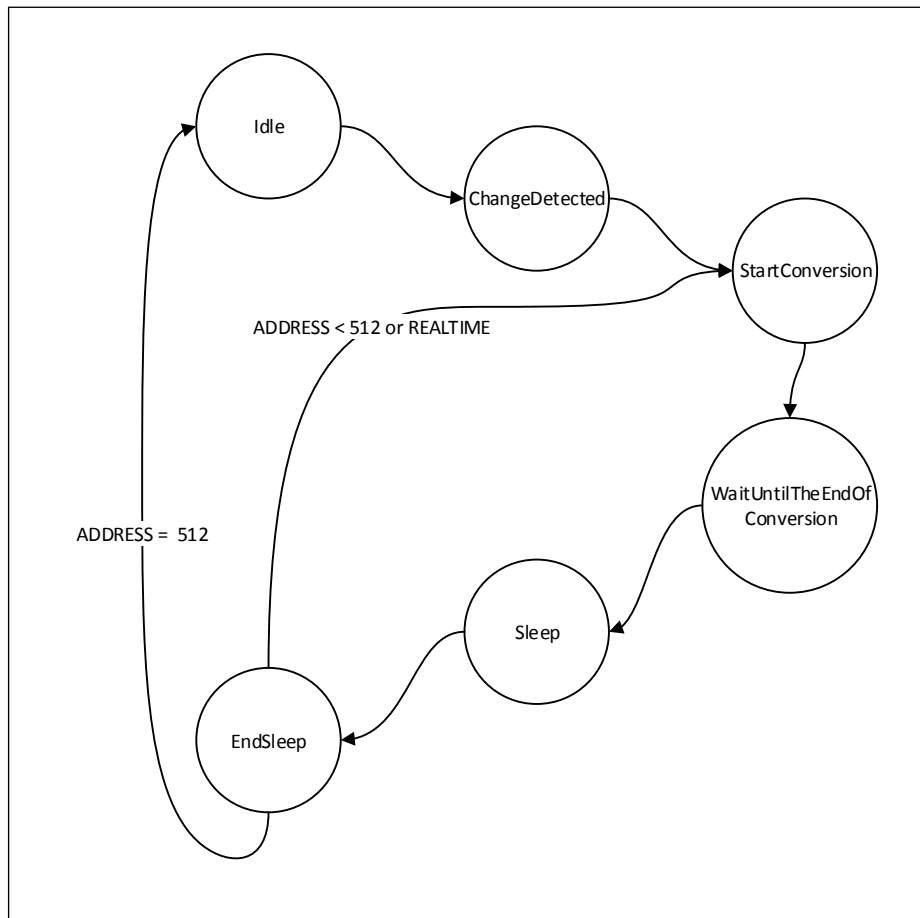


Figura 10: Diagrama de stare a automatului SAMPLER

Starea **ChangeDetected** reprezintă starea de initializare a registrului de baleiaj cu adresa 0. Se intră în această stare din **Idle** când se detectează o schimbare a datelor de pe magistrala **FREQ**. În starea **StartConversion** bitul **REQ\_SAMPLE** devine 1 logic, astfel se semnalează o cerere de conversie spre modulul de comunicare (COMM). Se așteaptă în starea **WaitUntilTheEndOfConversion** până când bitul **SAMPLE\_VALID** devine 1 logic iar apoi se intră în starea **Sleep** în care se incrementează un registru la fiecare tact până la valoarea semnalului **FREQ**. Când se atinge această valoare se intră în starea **EndSleep** de unde se poate continua baleierea adreselor sau înghețarea automatului în starea **Idle**. Modul de funcționare "Timp Real" presupune tranziția în starea **StartConversion** iar modul "Captură" tranziția în starea **Idle**.

Protocolul de comunicare este implementat pe placuta CEREBOT2 folosind intreruperile externe. Cand se detecteaza o tranzitie de tip front crescator al bitului `START_BIT` se va genera o intrerupere in care se cere o conversie AD. Valoarea unui registru (de stare) va fi setata pe `Waiting`. Cand conversia AD este terminata se va genera o alta intrerupere unde se salveaza valoarea convertita intr-un registru. Dupa fiecare intrerupere `REQ_BIT` se va trimite spre iesire valoarea convertita, se va face 1 logic `BIT_VALID` iar apoi registrul se deplasa cu un bit. Daca conversia nu este gata cand vine prima cerere `REQ_BIT` vom astepta pana cand va fi gata intr-o stare de `Wait (while(1))`. `BIT_VALID` va deveni 0 de fiecare data cand se executa bucla principala a programului.

Pentru testarea in simulator noi am folosit un bloc care genereaza esantioane pseudo- aleatoare cu bistabile si porti xor. Aceasta componenta respecta interfata de comunicare `COMM` si este o masina de stare.

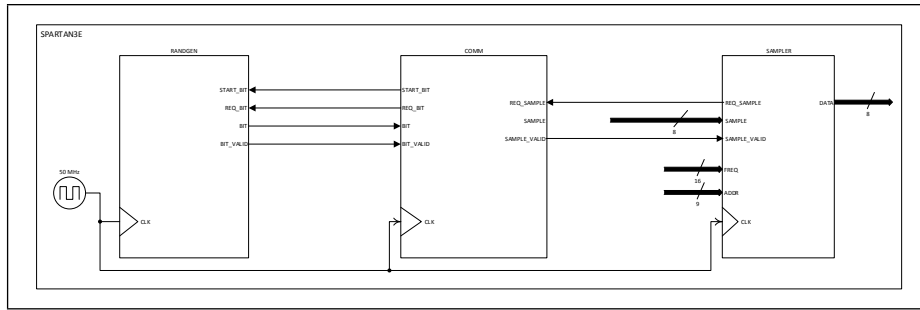


Figura 11: Schema blocului de esantionare cu generator de valori aleatoare pentru testarea in simulator

## 4.2 Logica de iesire

Aceasta sectiune a documentului este dedicata implementarii blocului de afisare din Figura 3.

Dupa cum s-a vazut la capitolul de fundamente teoretice, un controller VGA trebuie sa furnizeze cinci semnale: trei pentru culoarea curenta, si 2 pentru sincronizarea fascicolului de electroni. Din fericire, placuta *Basys2* pune la dispozitie 10 semnale pe care le converteste la un port VGA dupa cum urmeaza:

Observam ca *Basys2* ne pune la dispozitie trei semnale digitale pentru canalul culorii *Rosu*, trei pentru *Verde* si doar 2 pentru *Albastru*. Impreuna formeaza un bus de 8 biti. Culoarea albastra primeste o rezolutie mai mica deoarece ochiul uman este mai sensibil la verde si rosu. *Basys2* converteste aceste valori digitale la un semnal analogic pentru fiecare canal. Semnalele pentru sincronizarea orizontala si verticala sunt legate fara logica suplimentara.

Deoarece clock-ul placutei *Basys2* este de 50Mhz, putem aproxima o frecventa de 25.157Mhz impartind la doi frecventa placutei. O frecventa de 25Mhz este suficienta pentru ca majoritatea monitoarelor sa se recunoasca rezolutia.

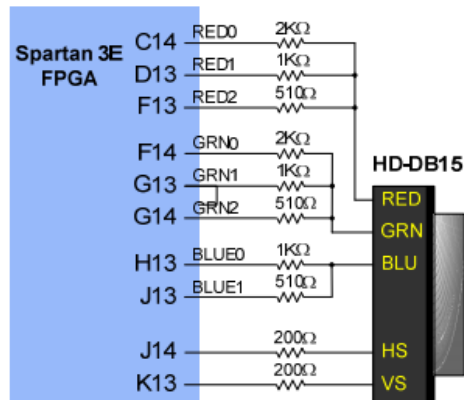


Figura 12: Basys2 VGA Port

```

if rising_edge(clk) then
    if clk25MHz = '1' then
        — logica la 25Mhz
    end if;
    clk25MHz <= not clk25MHz;
end if;

```

Din pacate, oscilatorul placutei Basys2 nu este suficient de stabil pentru a sincroniza monitoarele moderne. Se recomanda folosirea unui oscilator separat, conectat la slotul dedicat al placutei. Autorii proiectului nu au avut acces la un oscilator stabil. Imaginea rezultata folosind oscilatorul nativ sufera de artefacte verticale. Cu toate acestea, forma de unda este vizibila si poate fi recunoscuta de catre utilizator.

Componenta de control a logicii de iesire are ca semnal de intrare impulsul de clock, iar ca semnale de iesire pozitia pe X si Y a fascicolului de electroni pe suprafata monitorului, impreuna cu un semnal de control care indica daca fascicolul se afla pe suprafata vizibila sau nu.

Folosind doua comparatoare, se determina daca pozitia fascicolului pe verticala ( axa Y ) se afla intre valorile esantioanelor de la adresele X, respectiv X-1. Aceasta metoda presupune ca dimensiunea memoriei de esantionare este egala cu rezolutia verticala a monitorului in pixeli, iar valorile esantioanelor sunt scalate in prealabil pentru a fi mapate direct in spatiul monitorului.

Pentru a face comparatia intre pozitia curenta si valoarea a doua esantioane intr-un singur impuls de clock ( respectiv doua, luand in considerare faptul ca fascicolul de electroni petrece doua semnale de clock deasupra unui pixel, frecventa clock-ului intern al controllerului VGA fiind jumătate din frecventa clock-ului placutei ), se memoreaza valoarea esantionului de pe pozitia anterioara intr-un bistabil D. Daca pozitia curenta a fascicolului se afla intre valorile esantioanelor, culoarea trimisa la portul VGA este cea alba, adica "11111111". In caz contrar, sau daca fascicolul se afla in afara zonei vizibile, culoarea trimisa este cea neagra, respectiv "00000000". Acest lucru se face folosind un multiplexor legat la o logica combinationala ce foloseste valorile de iesire ale comparatoarelor, si semnalul ce indica daca fascicolul se afla in zona vizuala sau nu.

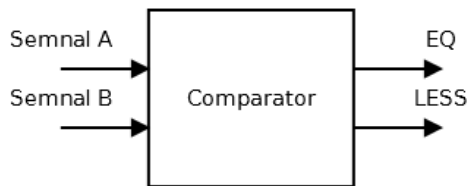
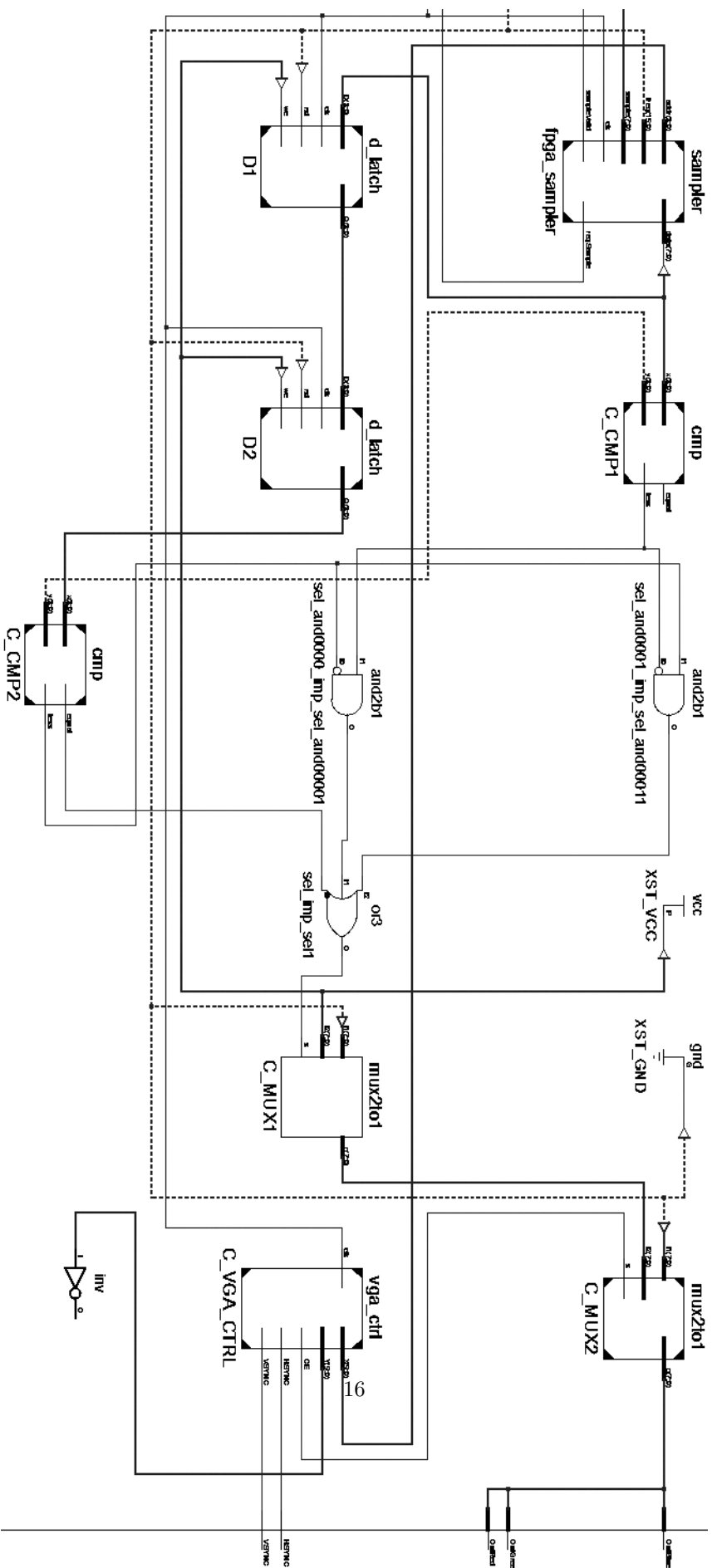


Figura 13: Comparator

Comparatoarele sunt circuite combinationalale ce primesc ca semnale de intrare doua valori de comparat, si prezinta la iesire doua semnale digitale numite *EQ* si *LESS*, active pe '1' logic, ce au ca semnificatie:

EQ	LESS	Interpretare
1	*	Semnalele sunt egale
0	0	Semnalul A este mai mic decat B
0	1	Semnalul A este mai mare decat B





## 5 Rezultate experimentale

## 6 Concluzii

În primul rând acest proiect este mai mult didactic și nu poate înlocui soluții care sunt mai eficiente din punct de vedere calitate/preț. Contribuția noastră a fost determinarea componentelor unui osciloscop digital și implementarea acestuia pe o placă FPGA în limbajul VHDL. Automatele de stare au fost folosite excesiv pentru controlul componentelor, astfel demonstrându-se puterea lor în a rezolva probleme de timp real.

Design-ul oferă un anumit grad de flexibilitate, osciloscopul având posibilitatea de a fi conectat la orice monitor VGA și la orice sistem de achiziție atât timp cât se respecta protocolul de comunicare (COMM).

Pentru entuziaști, sistemul poate fi folosit pentru verificarea unor semnale analogice cu condiția ca acestea să aibă tensiunea suportată de ADC-ul plăcii CEREBOT 2. Sistemul poate fi extins adăugând module care să afișeze date statistice sau auxiliare pe monitorul VGA, cum ar fi o legendă de timp, sau frecvența și amplitudinea unei detectate.

## Bibliografie

- [1] Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison Wesley, Massachusetts, 2nd Edition, 1994.
- [2] Analog and Digital Oscilloscope  
<http://en.wikipedia.org/wiki/Oscilloscope>
- [3] Practical questions and answers about oscilloscope use  
<http://forum.allaboutcircuits.com/showthread.php?t=2645>
- [4] Basys 2 Reference Manual  
[http://www.digilentinc.com/Data/Products/BASYS2/Basys2\\_rm.pdf](http://www.digilentinc.com/Data/Products/BASYS2/Basys2_rm.pdf)
- [5] VGA controller and specification  
<http://www.eng.utah.edu/~cs3710/labs/VGA.pdf>
- [6] Hardware Design with VHDL : VGA Example  
[http://www.ece.unm.edu/~jimp/vhdl\\_fpgas/slides/VGA.pdf](http://www.ece.unm.edu/~jimp/vhdl_fpgas/slides/VGA.pdf)
- [7] VGA Timings  
[http://hamsterworks.co.nz/mediawiki/index.php/VGA\\_timings](http://hamsterworks.co.nz/mediawiki/index.php/VGA_timings)
- [8] Cerebot II Reference Manual  
[http://www.digilentinc.com/Data/Products/CEREBOT-II/Cerebot\\_II\\_rm\\_RevB.pdf](http://www.digilentinc.com/Data/Products/CEREBOT-II/Cerebot_II_rm_RevB.pdf)