**UNIVERZITET U BEOGRADU**

**ELEKTROTEHNIČKI FAKULTET**

***Katedra za elektroniku***

**Uvod u projektovanje VLSI sitstema**

****

**Projektni zadatak:** Osciloskop

**Izveštaj**

Grupa 11

Nikola Jugović 0408/2016

Uroš Cvjetinović 0093/ 2016

# Sadržaj:

[**UVOD..............................................................................................................................................2**](#_UVOD)

[**INICIJALIZACIJA PROJEKTA....................................................................................................3**](#_Inicijalizacija_projekta)

**[IMPLEMENTACIJA U VHDL-u..................................................................................................4](#_Implementacija_u_VHDL-u)**

[**BLOK DIJAGRAM.........................................................................................................................6**](#_Blok_dijagram)

[**TABELA NAJVAŽNIJIH SIGNALA............................................................................................6**](#_Tabela_najvažnijih_signala)

[**TESTIRANJE I SIMULACIJA PROJEKTA.................................................................................8**](#_Testiranje_i_simulacija)

[**ZAKLJUČAK...................................................................................................................................10**](#_Zaključak)

[**LITERATURA..............................................................................................................................10**](#_Literatura)

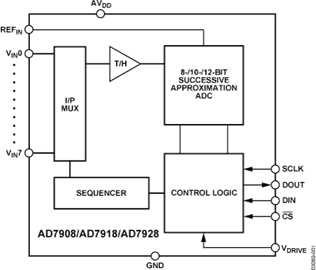
# UVOD

Tema projektnog zadatka je implementacija osciloskopa sa jednim kanalom koji kao jedinu mogućnost ima prikaz ulaznog signala.Realizacija projekta vrši se na Altera DE1-SoC pločici sa FPGA čipom u VHDL-u (VHSIC Hardware Description Language). Za dizajn projekta koristi se Altera Quartus II softver.

Ulazni signal u vidu napona se dovodi na osmokanalni AD konvertor AD7928 koji Altera DE1-SoC pločica ima ugrađen na sebi. AD7928 konvertuje ulazni napon na 12 bita koji se obrađuju i putem VGA kabla prenose na ekran za iscrtavanje signala.

# Inicijalizacija projekta

Ulazni signal se odabira frekvencijom od 80,64KHz na 1344 odbirka. AD konvertor (AD7928) korišćen u ovom projektu podatke prima i šalje serijski, zato radi uspešne realizacije AD konvertor mora da radi na frekvenciji 31 puta većoj od frekvencije odabiranja. Ovim se postiže da vreme mirovanja, pauza između odabiraka, je dovoljno duga da AD konvertor izvrši konverziju uspešno. Podaci koje prima AD konvertor definišu njegov način rada , za ovaj projekat AD konvertor radi u normalnom režimu rada. Signal *Din*, AD konvertora, podešen je tako da AD konvertor radi u normalnom režim (*Din=1000001100010000*), nakon prva tri odabiranja po deaktiviranju reseta. Ovim se izbegavaju prva tri odabirka koja imaju neispravnu vrednost. Ulazni podatak *Din* šalje se jedan po jedan bit na frekvenciji od 31\*80,64Khz , pri čemu za ovakav upis pravi komponenta koja će od paralelnog upisa napraviti serijski na frekvenciji na kojoj je potrebno. Signal *CS* aktivan je u logičkoj nuli i aktivira se posle dve periode signala takta po gašenju reseta, a deaktivira se posle 16 perioda signala takta nakon čega ostaje deaktiviran sve do 31. periode. Za vreme aktivnog *CS* signala vrši se upis *Din* signala i paralelno tome se vrši ispis *Dout* (tokom prve aktivacije signala *Cs,* signal *Dout* je neaktivan). Izlazni signal *Dout* ispisuje se serijski i to u formatu ‘0A2A1A0D11D10…D0’ pri čemu A2A1A0 adresa porta ulaznog signala , a D11...D0 izlazni konvertovani napon potreban za dalju obradu. Pošto se *Dout* ispisuje serijski, potrebno je odmah po izlazu izvršiti njegovo prebacivanje u paralelni ispis sa usklađivanjem signala takta (takt paralelnog ispisa će sada biti 80,64Khz). Nakon paralelnog ispisa *Dout* signala, vrši se duplo baferisanje podataka. Duplo baferisanje se postiže tako što oba bafera rade na isti takt, takt odabiranja, i signal rd\_wr koji je na jednom komplemtiran ulaz ovog signala. Bafer koji sa logičkom jedinicom na ulazu upisuje podatke sa ulaza i komunicira sa AD konvertorom, dok drugi ispisuje vrednosti koje je uskladištio u prethodnom ciklusu i obavlja komunikaciju sa iscrtavanjem signala preko vga kabla.



## Implementacija u VHDL-u

* PS - (Parallel-in to Serial-out (PISO) Shift Register)

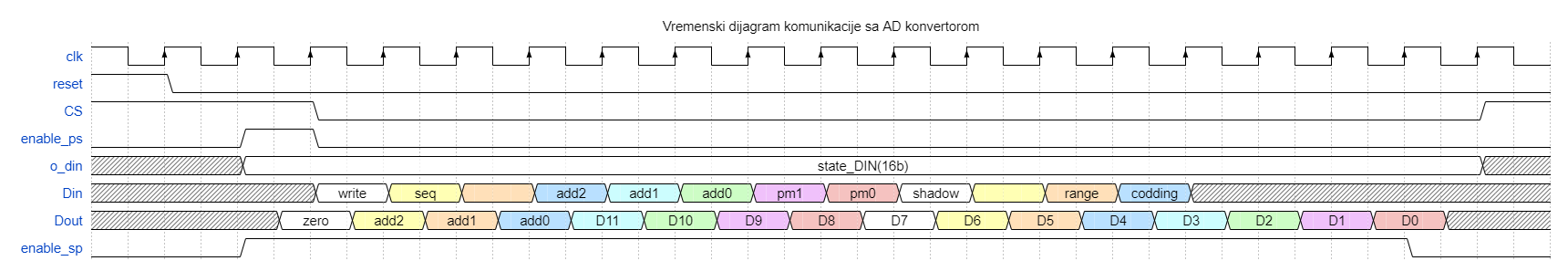
Komponenta *ps.vhd* konvertuje 16-bitnu reč koju generiše stanje komponente *adccontroler.vhd*  u signal usklađen sa taktnim signalom i traje 16 taktova, gde u svakom taktu se ispisuje najviši neispisan bit te 16-bitne reči. Ova komponenta je usklađena sa signalom *CS* pomoću signala *enable\_ps*  i služi za prosleđivanje reči Din u kontrolni registar AD konvertora.

* SP - (Serial-in to Parallel-out (SIPO) Shift Register)

Komponenta *sp.vhd* pamti bite signala *Dout* koji generiše AD konvertor u 16-bitnu reč, koja se prosleđuje na oba *sample\_buffer.vhd* . Ovim postupkom se memoriše jedan odabirak u bafer koji je zadužen za upisivanje, nakon 1344 odabirka baferi menjeju uloge. Ova komponenta je usklađena sa signalom *CS* preko signala *enable\_sp*  i služi za prosleđivanje reči Din u kontrolni registar AD konvertora.

* ADCCONTROLER

Komponenta adccontroler.vhd generiše sve pomoćne signale koji služe za sinhronizaciju upisa i čitanja iz AD konvertora. Ona takođe ima dva stanja, startup\_mode i normal\_mode, koja služe da se preskoče prva tri odabiranja nakon uspostavljanja napona (što se smatra nakon deaktiviranja reseta). Prva tri odabiranja se preskaču tako što se u tom periodu ne generiše takt za učitavanje u bafer i signal dozvole bafera.



* SAMPLE\_BUFFER

Dvostruko baferisanje je postignuto preko dve komponente *sample\_buffer.vhd*  koje preko taktnog signala *r\_sample\_clk* i signala *rd\_wr* koje generiše komponenta *adccontroler.vhd* rade naizmenično upisivanje u jedan bafer i ispisivanje odabiraka iz drugog bafera. Ovim se postiže da slika sa vga kanala bude bez prekida, i da se u isto vreme čita iz AD konvertora i crta prethodnih 1344 odabiraka na ekran.

* PLL

Komponenta pll.vhd služi za podizanje taktnog signala od 50 MHz na 65 MHz potrebnih za rad sa ekranom preko vga kanala.

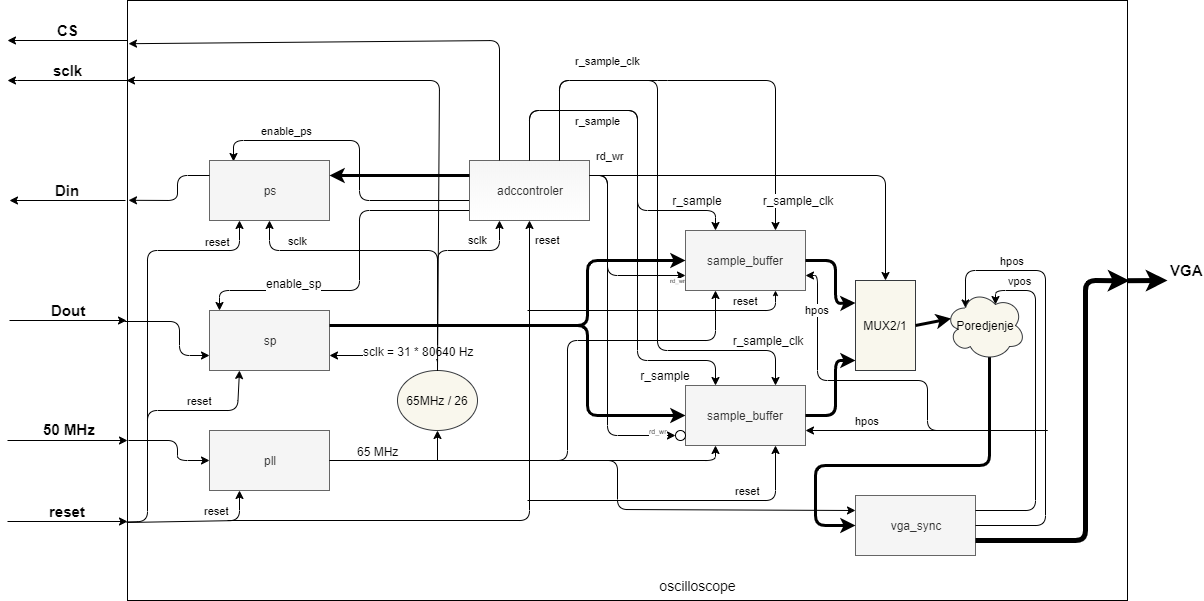
* VGA\_SYNC

Komponenta *vga\_sync.vhd* služi za sinhronizaciju signla koji generišu sliku na ekranu sa komponentom *oscilloscope.vhd*

* VGA logika

Iscrtavanje na ekran monitora je realizovano poređenjem vrednosti odabirka na poziciji *hpos* u baferu sa vrednošću *vpos*  uz određen offset kojim pomeramo nulu na sredinu ekrana. Ovde se može konfigurisati boja pozadine i signala, kao i potencijalne modifikacije u ovom projektu.

## Blok dijagram



# Tabela najvažnijih signala

|  |  |  |
| --- | --- | --- |
| **entity** | **SIGNAL** | **ZNAČENJE DATOG SIGNALA** |
| **oscilloscope.vhd** | clk\_50mhz | Ulazni signal takta koji daje sam FPGA i u odnosu na koji se prave svi ostali signali takta koji su potrebni tokom izrade |
| i\_dout | Izlaz AD konvertora koji se dovodi na ulaz oscilloscope.vhd entitija |
| reset | Taster za reset na FPGA pločici , služi za resetovanje svih signala |
| sclk | Izlazni signal takta od 31\*80,64KHz koji je potreban za serijski upis/ispis |
| cs | Signal koji služi kao enable kod AD konvertora, aktivan je u nuli |
| o\_din | Ulazni signal AD konvertora koji definiše njegov način rada |
| VGA\_CLK | Izlazni signal takta od 65MHz koji je potreban za sinhronizaciju sa VGA kablom |
| VGA\_HS | Signal horizontalne sinhronizacije kod iscrtavanja |
| VGA\_VS | Signal vertikalne sinhronizacije kod iscrtavanja |
| VGA\_BLANK\_N | Signal koji određuje da li je korišćen direct blanking |
| VGA\_SYNC\_N | Signal koji određuje da li je korišćen sync on green. |
| VGA\_R | Osmobitni niz koji definiše intenzitet crvene boje |
| VGA\_G | Osmobitni niz koji definiše intenzitet zelene boje |
| VGA\_B | Osmobitni niz koji definiše intenzitet plave boje |
| **ps.vhd i sp.vhd** | i\_clk | Signal takta od 31\*80,64KHz za konverziju paralelnog prenosa u serijski |
| i\_rstb | Resetovanje svih signala |
| i\_data\_ena | Signal koji kaže da je podatak za ulaz spreman |
| i\_data | Ulazni podatak |
| o\_data\_valid | Signal koji kaže da je izlazni podatak spreman |
| o\_data | Izlazni podatak |
| **adccontroler.vhd** | reset |  |
| sclk | Signal takta od 31\*80,64KHz za sinhronizaciju sa *ps.vhd* komponentom |
| o\_data\_ena\_ps | Signali sinhronizacije između *ps/sp I adccontroler.* Signal koji govori da je podatak za sp*/ps* spreman |
| o\_data\_ena\_sp |
| o\_cs | Signal koji služi kao enable kod AD konvertora |
| o\_din | Ulazni signal AD konvertora koji definiše njegov način rada |
| sample\_clk | Signal takta odabiranja 80,64KHz |
| sample | Signal koji dozvoljava rada sa baferima |
| **sample\_buffer.vhd** | reset | Resetovanje svih signala |
| enable | Signal koji dozvoljava rad bafera |
| clk\_65mhz | Signal takta od 65MHz koji predstavlja frekvenciju slanja niza 12obitnih podataka ka VGA izlazu |
| sample\_c | Signal takta odabiranja 80,64KHz |
| rd\_wr | Signal koji kontroliše koji bafer će biti aktivan |
| i\_data | Ulaznih 12bita koji predstavljaju jedan odbirak. Prosleđivanje ovih podataka u bafer se vrši na učestanosti od 80,64Khz, odnosno na frekvenciji od 60Hz se prosleđuje niz od 1344 12obitnih podataka. |
| wr\_element | Informacija o poziciji elementa u nizu od 1344 odbirka |
| o\_data | Izlazni podatak od 12 bita koji se prosleđuje na VGA izlaz na frekvenciji od 65MHz. |

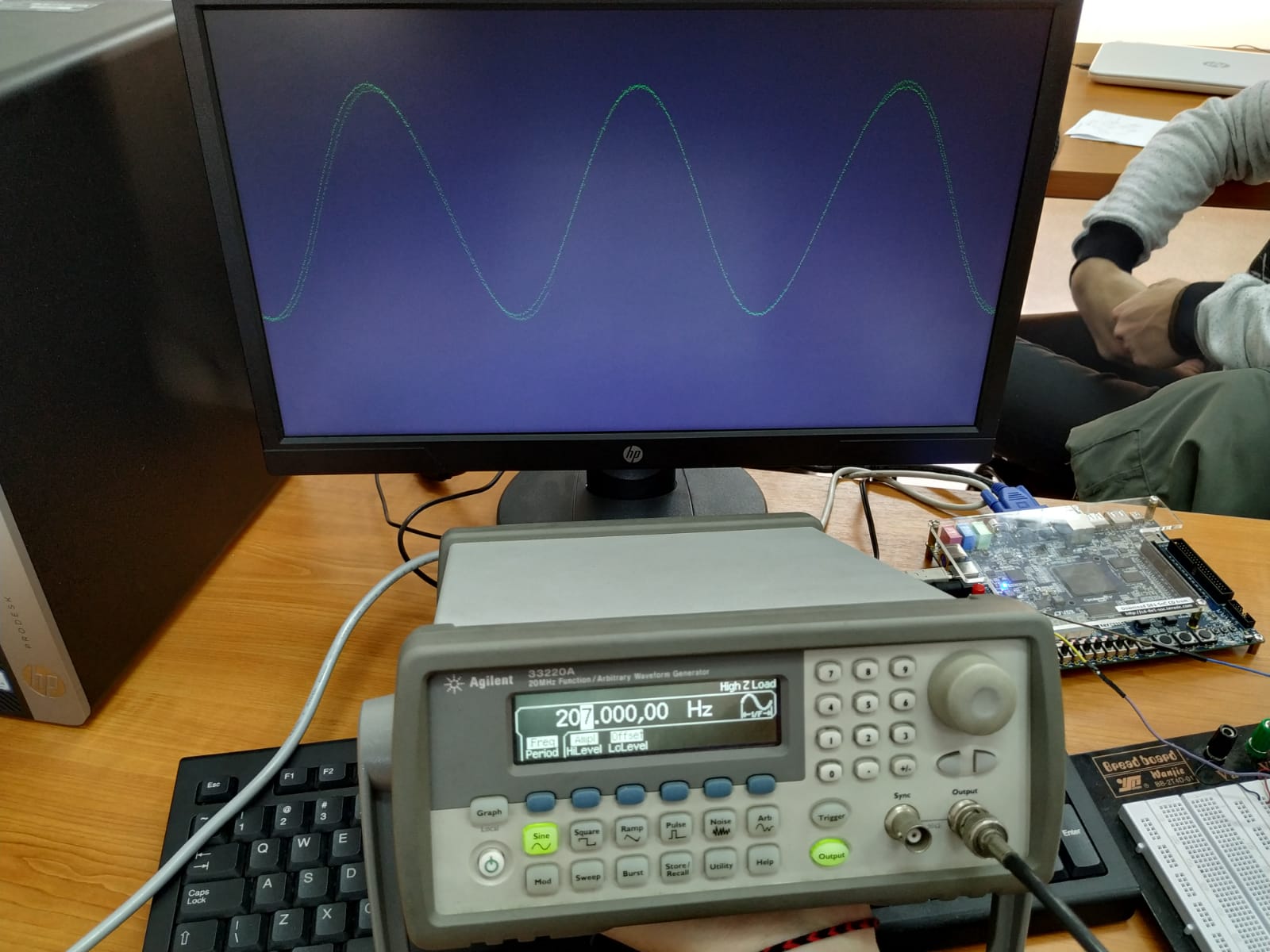
|  |  |  |
| --- | --- | --- |
| **vga\_sync.vhd** | clk | Signal takta od 65Mhz koji se koristi za iscrtavanje |
| reset | Resetovanje svih signala |
| hsync | Horizontalna sinhronizacija |
| vsync | Vertikalna sinhronizacija |
| hpos | Horizontalna pozicija trenutnog piksela |
| vpos | Vertikalna pozicija trenutnog piksela |
| sync\_n | Sync on green |
| blank\_n | Direct blanking |
| Rin | 8bitni podatak vrednosti crvene boje |
| Gin | 8bitni podatak vrednosti zelene boje |
| Bin | 8bitni podatak vrednosti plave boje |
| Rout | 8bitni podatak vrednosti crvene boje kome se prosleđuje *Rin* u zavisnosti od toga da li se trenutni piksel nalazi u vidljivom delu |
| Gout | 8bitni podatak vrednosti zelene boje kome se prosleđuje *Gin* u zavisnosti od toga da li se trenutni piksel nalazi u vidljivom delu |
| Bout | 8bitni podatak vrednosti plave boje kome se prosleđuje *Bin* u zavisnosti od toga da li se trenutni piksel nalazi u vidljivom delu |

\***Signali koji su obojeni istom bojom predstavljaju iste signale samo pozivane u okviru različitih komponenti**

## Testiranje i simulacija projekta

Simulacija projektra vršena je korišćenjem programa ModelSim-IntelFPGA. Pisanjem testbench fajla u okviru programa , moguće je posmatrati promenu signala od interesa i na taj način odrediti funkcionalnost projektra.

Testiranje projekta, odnosno hardversko „spuštanje“ na ploču je prošlo uz više sitnih problema, tako da sam projekat nije pokrenut pri prvom testiranju. Nakon popravke određenih stvari oko bafera i iscrtavanja slike projekat je uspešno spušten na FPGA pločicu.



Slika 1-Sinusoida

Na *Slici 1* se jasno vidi način testiranja projekta, kao i njegova funkcionalnost. Korišćenjem signal generatora generisana je sinusoida apmplitude od 1V sa ofsetom od 1V. Generisani signal se preko protoborda dovodi direktno na ulaz 0 (ADC\_IN0) AD konvertora koji je ugrađen na FPGA pločici. FPGA pločica preko VGA kabla koji je povezan na monitor prikazuje signal na već pomenuti način.



Slika 2 - “Testera”

# Zaključak

Projekat je prilično zahtevan i obiman, uspešna implementacija projekta zahteva dobro poznavanje VHDL programskog jezika i ispravnu logiku prilikom izrade. Najveći problem koji se javlja pri izradi projekta je usklađivanje različitih taktova koji se javljaju prilikom povezvanja komponenti. Implementaciju projekta bi olakšao i AD konvertor koji ima mogućnost paralelnog upisa i ispisa umesto serijskog.

Prostor za unapređenje projekta najvećoj meri postoji kod grafičkog dela, odnosno kod iscrtavanja signala. Projekat može biti realizovan tako da ima veliki broj funkcija i ispisa koje poseduje i pravi osciloskop. Takođe i broj kanala može biti povećan na veliki broj. Što se tiče optimizacije samog koda ne postoji mnogo prostora za optimizaciju.

# Literatura

[1] Upustvo za AD konvertor

<http://tnt.etf.bg.ac.rs/~oe4upv/materijali/projekti/ADC_AD7928.pdf>

[2] Uputstvo za konverziju serijskog prenosa u paralelni

<https://surf-vhdl.com/how-to-implement-a-serial-to-parallel-converter/>

[3] Uputstvo za konverziju paralelnog prenosa u serijski

<https://surf-vhdl.com/how-to-implement-a-parallel-to-serial-converter/>