|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Odsek za računarsku tehniku i računarske komunikacije

***Naziv predmeta:***

- Logičko projektovanje računarskih sistema 2-

**Predmetni projekat:**

8-bit MIDI (DAC)

***Mentor:***  ***Studenti:***

Prof. Dr. Savo Dragović RA117/2015

Nebojša Pjevalica Igor Ilić RA152/2015

Srđan Usorac RA060/2015

Novi Sad, (11.06.2018)

**Sadržaj**

1 Uvod.....................................................................................................................2

2 Početna tačka, korišćene periferije i generisani moduli……...................….…2

3 Logika hardverske implementacije projekta...………..……...................….…2

4 Pravljenje interrupt-a i veza hardverskog i softverskog dela…...................….…3

5 Softverska implementacija …………………..….……………...................….…3

6 Prilog…………………………………………………………………………….3

1. **Uvod**

U ovom projektu je potrebno realizovati hardver i softver za osmobitni (***8-bit***) digitalno analogni konvertor (***DAC***). Hardver je rađen u *Xilinx Platform Studiu* **(*XPS*)** kao periferija, a softver u *Xilinx* ***SDK***.

1. **Početna tačka, korišćene periferije i generisani moduli**

Postojeći projekat ***MIDI buzzer*** za ***Atmega16*** je bio početna tačka. Najvažnije je bilo saznati princip rada tog projekta, tačnije, signal koji generiše hardver ***AVR***, da bi se takav mogao projektovati. Hardver je potom urađen tako da obavlja posao specifičan za projekat. Periferija koju smo nadograđivali je ***buzzer\_per 0***. Na njoj smo dodali nova 4 porta od kojih tri pripadaju grupi izlaznih pinova, a jedan je ***axi interrupt controller***koji radi na 48kHz, što je jednako frekvenciji DAC-a. Napravili smo poseban modul u VHDL-u pod nazivom ***DAC\_8\_bit***, koji služi za slanje signala na pinove na koje priključi DAC, koji na osnovu poslatih signala vrši logiku ***Shift Registra****.*

1. **Logika hardverske implementacije projekta**

Kompletna hardverska implementacija projekta fokusira se na modul ***DAC\_8\_bit.vhdl****.* Frekvencija Bus2IP clock-a je ***100MHz***, a frekvencija našeg izlaznog clock-a je ***1MHz***. Bilo je neophodno napraviti brojač za Bus2IP koji ce brojati do 100, kako bismo redukovali frekvenciju. Implementirali smo ***FSM*** sa 18 stanja (16 stanja u kojima se dodeljuje vrednost pojedinih bita za slanje na DAC (8) i u kojima se generiše clock na 1MHz (8), jedno IDLE stanje i stanje u kojem se vrši dozvola slanja osmobitnog odbirka na izlaz DAC-a). FSM vrši prelaz stanjasvaki put kad brojač izbroji 100 taktova. Sama logika ***Shift Registra*** implementirana je u konkretnom uredjaju, naš zadaatak je bio da šaljemo u odgovarajuće vreme odgovarajuće bite iz ulaznog registra (*i\_sample*). Sama komponenta naseg modula napravljena je u ***user\_logic*-u**, a izlazni signali su sprovedeni kao izlazi iz ***buzzer\_per***.

1. **Pravljenje interrupt-a i veza hardverskog i softverskog dela**

Sam rad DAC-a je na frekvenciji od ***48kHz***, odnosno dobijanje novih *samplova* je na svaki 48000. deo sekunde. Upravo zbog toga potrebno je bilo napraviti ***axi interrupt controller***u hardverskom delu. Napravljen je kao *flag* koji se aktivira kada brojač dostigne vrednost 2083. Implementacija *interrupta* je u okviru modula ***DAC\_8\_bit.vhdl****.* Veza hardverskog i softverskog dela leži u tome što mi iz c koda dobijamo uzorke u vidu ***Bus2IP\_Data*** signala. Konkretno za implementaciju logike slanja koristimo registar broj 4.

1. **Softverska implementacija**

Na softveru je jedino ostalo popunjavanje registra hardvera odgovarajućim vrednostima (konkretno u našem slučaju koristi se registar broj 4). U softveru je odrađeno čitanje iz fajla ***zutakuca.h***, u kojem se nalazi fajl koji sadrži parove vrednosti (dužina note, frekvencija). Pod dužinom note podrazumeva se na deo sekunde koliko traje taj ton. Implementirana je metoda ***buzz\_interrupt\_handler\_1*** u kojojje odrađeno generisanje tona (sinusa preko odgovarajuce tabele sinusa iz datoteke ***ddc.c***) i upis odgovarajućih vrednosti u register uy naravno proveravanje da li je opseg prekoračen. U main() su unete neke početne vrednosti za registre, što predstavlja početni korak.

1. **Prilog**

U prilogu se nalazi link na kome se nalazi youtube snimak na kome je snimljen primer melodije puštene preko ***E2LP*** ploče uz pomoć osmobitnog DAC-a.

https://www.youtube.com/watch?v=j6Yf\_29xFwE&feature=youtu.be