

## SFP MODULI I NJIHOVA ULOGA U MERNO-INFORMACIONIM SISTEMIMA

### SFP MODULES AND ITS APPLICATION IN MEASUREMENT AND INFORMATION SYSTEMS

Tomislav Pap, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** – *Ovaj rad sadrži uvid u fizičku realizaciju SFP komunikacionog protokola i njegove primene u merno-informacionim sistemima. Opisane su osnovne strukture protokola. Prikazano je hardversko rešenje na FPGA uređaju. Sagledana su moguća unapređenja u dizajnu.*

**Ključne reči:** SFP, FPGA, merenje, sistem na čipu, IP

**Abstract** – *This paper contains implementation of SFP communication protocol and its application in measurement and information systems. Basic structures of the protocol are described. Hardware solution on FPGA device is given. Possible improvements are perceived.*

**Keywords:** SFP, FPGA, measurements, System on Chip, IP

#### 1. UVOD

Danas, u 21. veku uz veliko napredovanje tehnologije, možemo reći da ne postoji oblast privrede i nauke koja može da funkcioniše bez neke vrste merenja. Merenje predstavlja određivanje (detektovanje) neke veličine ili objekta. Ta veličina se iskazuje numerički (brojnom vrednošću) i jedinicom određene merne veličine po SI međunarodnom sistemu jedinica. U *embedded* sistemima gde njihova funkcija predstavlja i jedan deo merenja ili transporta mernih veličina, neophodno je velika količina memorije, a nekada i dodanih paralelnih jedinica za obradu mernih podataka. Jedan dobar primer koji spaja dve takve jedinice za obradu mernih podataka jeste i SFPmodul koji je realizovan kao IP jezgro na FPGA ploči.

SFP modul se koristi kao industrijski standard širom sveta od strane velikih proizvođača već gotovo 20 godina u mnogim mrežnim rešenjima [1] kao što su: SONET, Gigabitni ethernet, optički kablovi, PON itd. SFP modul, kao mali modularni predajnik, zamenio je svuda GBIC modul, jer njegova veličina omogućava da se implementira u jako maloj fizičkoj zapremini i uspostavi komunikacija između svičeva i bitnih serverskih konekcija. Kako je SFP modul *hot-pluggable* to mu omogućava da se lako prilagodi postojećim mrežnim protokolima bez nužnog redizajniranja cele kablovske infrastrukture. Tema ovog rada su SFP moduli i njihova uloga u merno-informacionim sistemima gde se realizuje

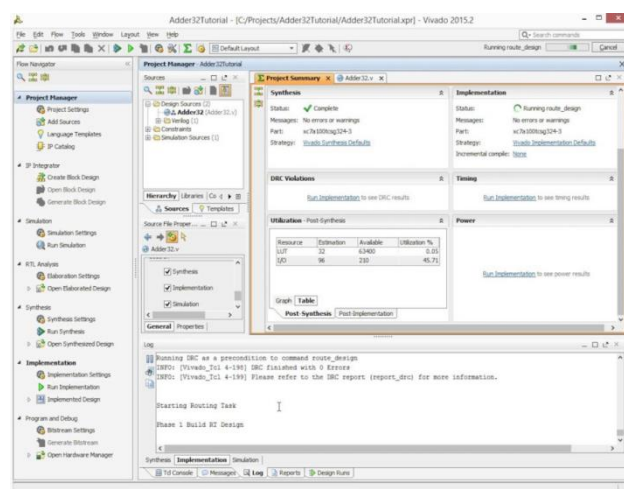
#### NAPOMENA:

**Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Platon Sovilj, red. prof.**

gigabitna komunikacija između dve FPGA jedinice. Realizovan je otvoreni SFP protokol sa namerom za moguće dalje nadogradnje u budućnosti.

#### 2. RAZVOJNI ALATI

Po prirodi zadatka, hardverska realizacija je komplikovanija od softverske realizacije pa se kao odgovarajuće rešenje nameće da se prvo željena funkcionalnost istestira u softveru, pa nakon toga da se implementira u hardveru. Kako je ovu funkcionalnost dosta teže opisati u softveru, jer se sama funkcionalnost i performanse protokola zasnivaju na hardveru, odmah je započeta hardverska realizacija, gde je podrška u vidu simulacije sistema bila podržana alatom za simulaciju kompanije Xilinx. Za hardversku realizaciju dizajna neophodni su alati kao što su Xilinx ISE, Xilinx SDK i Xilinx Vivado 2015.2 (Slika 1).



Slika 1 - Xilinx Vivado 2015.2

Xilinx Vivado 2015.2 je alat koji se koristi za razvoj i simulaciju hardverskog dizajna. Pomoću njega, moguće je zapakovati dizajn hardvera u IP i povezati ga sa drugim IP-evima, testirati dizajn gledajući signale koji se generišu u simulaciji, kao i generisati *bit stream* pomoću koga se programira FPGA. Može se koristiti grafičko okruženje ili napisati tcl skripta.

Vivado Xilinx može da se koristi sa grafičkim interfejsom, ali mnogo češće se upotrebljava kao terminal/konzolni interfejs, gde se kao krajnji proizvod dobija grafički interfejs, na kome su prikazani signali svakog modula u dizajnu i povezanost celog sistema. Preglednost krajnjeg proizvoda je lakša u grafičkom interfejsu, ali njegov razvoj, ponovljivost, modularnost i

automatizacija je naprednija, brža i sa dosta više opcija u konzoli. Potrebno je vreme savladati konzolni interfejs, odnosno alat za pisanje skripti koja se izvršava u konzoli pod xilinxovom podrškom (Xilinx Batch mod), kao alat zvani TCL (eng. terminal console interface).

Svaka komanda u Xilinx Vivadu, ima svoju odgovarajuću komandu u tcl jeziku. U suštini, ceo Vivado je napisan u tcl jeziku, pa zbog toga i postoji takva paralela. Pomoću Vivada, već postojeći projekat je moguće sačuvati u tcl skript obliku.

TCL je jezik za pisanje skripti koji je inkorporiran u spektar alata Xilinx familije. TCL jezik dozvoljava interaktivne upite ka alatima za dizajn sa ciljem kako bi automatizovao neke procese. Pomoću njega moguće je vršiti upite prema "bazi podataka" samog dizajna, trenutnog stanja projekta i njegovih podešavanja. Na primer, moguće je vršiti upit rezultat specifične vremenske analize, na osnovu kojih se dalje kreiraju dodatna ograničenja/limiti pomoću kojih se dizajn lakše implementira na FPGA ploču [2].

VHDL je jezik za opis hardvera koji se koristi u automatizaciji elektronskog dizajna za opisivanje digitalnih sistema i mešovitih-sistema kao što su FPGA. VHDL je definisan IEEE standardima i najpopularnije standardizacija je IEEE VHDL- 93. Svi programski jezici se izvršavaju sekvencijalno, dok u VHDL slučaju svaka instanca koda se izvršava paralelno jer zaista tako i funkcioniše hardver u realnosti.

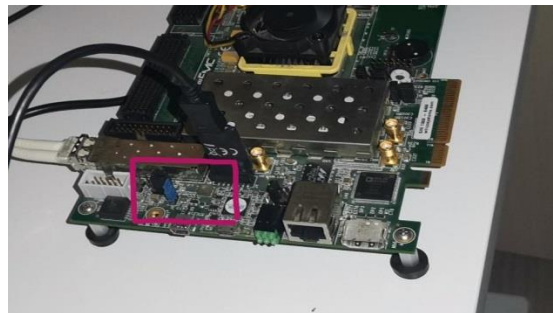
Iako jezik za opis hardvera, kao takav potrebno je imati alat za kontrolu verzije izvornog koda. Alat za kontrolu verzije koda (eng. SCM – Source code management) je način za praćenje promena u kodu, tako da ako nešto pođe po zlu, imamo mogućnost da uporedimo različite verzije koda, i da se vratimo na bilo koju prethodno sačuvanu verziju koda koju želimo. Prilikom razvoja protokola, bio je jako koristan za strateški razvoj deo po deo celog protokola. Takođe kada neki delovi VHDL koda nisu bili u stanju da zadovolje kriterijume implementacije, lako je bilo se vratiti na prethodnu verziju gde je sve radilo kako treba. Prilikom napredovanja protokola i praćenje verzija koda, stekao se efikasan uvid u to koliko svaka dodatna opcija protokola i deo koda zauzimaju dodatne FPGA logike, što je kasnije bilo korisno za analizu rezultata implementacije i detekciju mogućih unapređenja.

## 2. FIZIČKA REALIZACIJA SFP MODULA

FPGA uređaji imaju svestrane mogućnosti i funkcionalnosti visokih performansi. Fizička realizacija koja može da se mapira na njihove logičke gradivne jedinice je ograničena i treba pažljivo projektovati željenu funkcionalnost. U slučaju složenih embeded sistema, često se javlja potreba za pribavljanjem određenih promenljivih iz spoljašnje memorije ili nekog drugog podsistema. Razlog može biti obrada podataka neke spoljašnje jedinice ili komunikacija sa drugim uređajem.

Prilikom realizacije komunikacije sa spoljašnjim jedinica-ma ili memorijom, često se koriste standardizovani protokoli kao što su AXI, I2C ili SPI. Svim ovim protokolima je limitirajući faktor brzina, te je za brzu komunikaciju, reda velične Gb/s, mnogo praktičnije koristiti SFP protokol.

SFP je hardverski modularni slot koji služi za komunikaciju dveju FPGA ploča ili nekim drugim uređajima. SFP slot se direktno mapira na spoljašnje pinove FPGA ploče, a sama funkcionalnost se projektuje na hardveru, jer softverska realizacija zbog prirode SFP protokola, odnosno zbog njegove brzine nije moguća u softveru. Dobar primer SFP protokola je kod kompanije Tajfun HIL, gde je SFP protokol na uređaju HIL404, bidirekcion, otvoren protokol, koji omogućava veliki protok (do 5 Gbps), malu latenciju prilikom razmene podataka između uređaja. Ovaka protokol je pogodan jer ne zavisi direktno od proizvođača, može da razmeni do milion poruka u nekoj tipičnoj aplikaciji i pouzdan je jer ima ugrađenu sinhronizaciju SFP protokol je razvijen na



Slika 2 - VC707 ploča sa položajem džampera za SFP komunikaciju

Xilinx Virtex-7 FPGA VC707 ploči koja ima na sebi SFP port modul i nakon mapiranja pinova, potrebno je još uraditi i nameštanje džampera u odgovarajući položaj, jer uobičajeno nije na toj poziciji zbog uštede potrošnje energije i dodatnih drajvera u softveru potrebnih da bi SFP radio. Položaj džampera treba namestiti kao na slici 2.

Kako je SFP široka, i kako se konfiguracija SFP protokola razlikuje od slučaja do slučaja, moguće ga je pripremiti za buduću komunikaciju putem softverske inicijalizacije pre samog uspostavljanja linka protokola. Da bi se realizovao SFP protokol, neophodne su SFP sonde (koji su prikazane na slici 3 (a)), kao i optički kabl (slika 3 (b)).

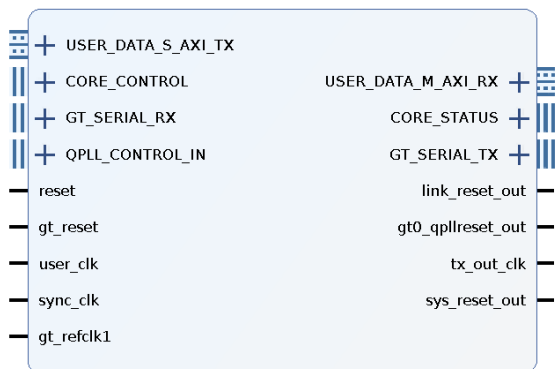


Slika 3 - SFP sonda (a) i optički kabl (b)

## 3. GLAVNI ASPEKTI FIZIČKE REALIZACIJE

Projektovanje kompletnog embeded sistema je težak zadatak i postoje aspekti koje bi trebalo sagledati prilikom kreiranja i projektovanja dizajna na platformi kao što je FPGA, što znatno olakšava razvoj sistema u kasnijim fazama. Postojeći programabilni hardverski deo FPGA uređaja je strukturiran i njegove karakteristike su dobro poznate alatu za projektovanje digitalnih sistema. Uzimajući to u obzir, postoji velika prednost u

projektovanju sistema, ako neki delovi sistema mogu da se iskoriste iz prethodnih projekata ili ako se trenutni delovi projektuju na takav način, da bi funkcionalno mogli da se iskoriste na nekom drugom projektu. Time se stiče ušteda vremena za projektovanje i verifikaciju. Kao takav kriterijum, Aurora IP jezgro je generisano na taj način da se ima na umu ponovno korišćenje istog u nekim drugim projektima. Interakcija sa Aurora IP-om (Aurora interfejs je prikazan na slici 4) zahteva dosta poznavanje modula [3].



Slika 4 - Aurora 8B/10B interfejs

Unutrašnja struktura jezgra je komplikovana i treba obratiti pažnju na parametre koji se konfigurišu. Kao krajnji ishod, kreiran je top modul, koji obuhvata unutar sebe ceo sadržaj Aurora IP-a pod nazivom hssl\_ctrl\_11. Njegov interfejs je jednostavniji za krajnjeg korisnika i sadrži parametre sa logičnijim nazivima za konfiguraciju Aurora jezgra. Kako je sve konfigurisano za jednu liniju protokola podataka, ovaj modul je moguće više puta instancirati, čime se dobija mogućnost proširenja u budućnosti.

Svaki protokol, pa tako i SFP protokol je podložan promenama i unapređenjima te je stoga veoma bitno imati na umu parametrizaciju interfejsa protokola. U zavisnosti od standarda neke kompanije, ili karakteristike uređaja potrebno je imati mogućnost da se izmene neki parametri kako bi mogda da se ostvari željena komunikacija između uređaja. SFP protokol ima mogućnost komunikacije, odnosno sprezanje više proča istovremeno. Ulazni i\_unit\_id signal služi kako bi se postavio ID (identifikacioni broj) uređaja, da bi se na osnovu toga moglo detektovati u pristigloj poruci da li je poruka namenja baš za taj uređaj ili neki drugi koji je u spregnutoj komunikaciji. Isti ID broj se lepi na heder poruke koja se šalje kako bi drugi uređaji mogli da identifikuju poruku.

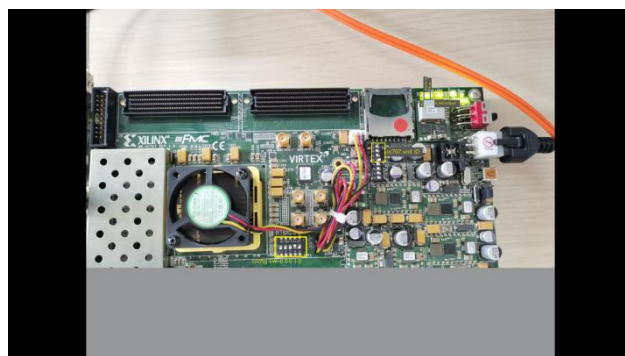
Konfiguracija ID broja je direktno mapirana na prekidače FPGA uređaja. Kako različiti FPGA uređaji imaju različiti broj dostupnih prekidača, ostavljena je mogućnost i\_unit\_id širine da bude konfigurabilna. Slična logika se može primeniti i na izlazni signal o\_led. O\_led signal služi za signalizaciju inženjeru da li se komunikacija odigrava dobro i na njega postoji mogućnost mapiranja različitih unutrašnjih signala za dalje debugovanje. U zavisnosti od dostupnosti LED dioda na FPGA uređaju, moguće je veći ili manji broj debugovanja unutrašnjih signala.

Zbog zahteva fizičke realizacije SFP protokola, softverski deo FPGA uređaja nije bio detaljno razmatran kao deo razvoja protokola, ali uzeta je činjenica u obzira za buduće proširenje i konfigurisanje protokola preko softverskog dela.

U zavisnosti od nivoa apstrakcije embeded sistema, zavisi i njegova dinamika i teškoća razvoja, kao i krajnje performanse. Razmotrivši sve prednost i mane, kao finalna odluka uzeta je da se SFP protokol razvija na RTL (eng. register transfer level) nivou apstrakcije. Zbog RTL nivoa razvoja, konfigurisanje zaglavlja poruke SFP protokola je intuitivna za dizajnera. Postoji mogućnost menjanje širine bita kao i preuređivanje redosleda podataka u zaglavlju, a to su: unit\_id, destination\_id, podatak i rezervisani biti koju su napravljeni da se celo zaglavlje smesti u tridesetdvobitnu vrednost, a mogu reprezentovati i neku dodatnu informaciju poruke ukoliko ima potrebe za tim.

Da bi SFP protokol ima ispravnu komunikaciju, moraju se ispoštovati zahtevi Aurora IP jezgra na čemu se i bazira ceo protokol. To znači, da je neophodna stalna sinhronizovana komunikacija između uređaja i da ne sme da se desi da ne dođe do odgovora na poslatu/primljenu poruku unutar komunikacije. Kako bi se to realizovalo u samom dizajnu digitalnog sistema se koriste FSM (eng. finit-state machine) mašine.

Jedna FSM mašina služi za pakovanje podataka za slanje. Ona garantuje ispravnost poruke i da će poruka biti poslata tek kada postoji aktivan link između uređaja. Druga FSM mašina pakuje podatke i šalje ispravan broj paketa u zavisnosti od zaglavlja poruke. Kako je ceo protokol zamišljen kao otvorena instanca komunikacije putem SFP linka sa tajfun HIL uređajima, ceo protokol je projektovan tako da dizajner ima celu kontrolu nad suštinom stvari koje se šalju, a to su zaglavlje i podaci poruke. Kao takav, lak je izbor za dodatne nadogradnje koje specifično odgovaraju određenim uređajima i izmene protokola koje bi karakterisali uzajamnu komunikaciju između ploča. Ceo digitalni sistem je razvijen na Xilinx Virtex-7 FPGA VC707 ploči (slika 5).



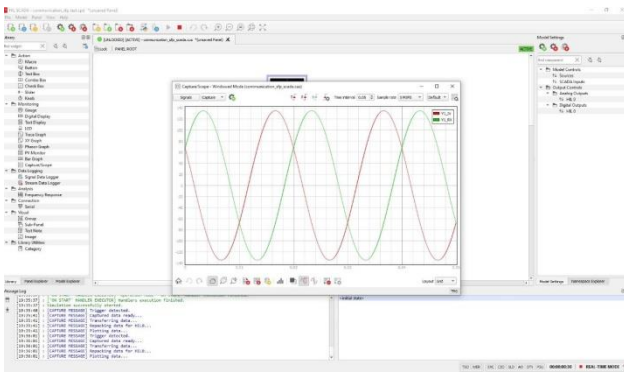
Slika 5 - Xilinx Virtex-7 FPGA VC707 ploča

#### 4. ZAKLJUČAK

Ovaj protokol je trenutno limitiran Aurora IP jezgrom 8B/10B mogućnostima, što bi značilo komunikaciju do 5 GBs protoka podataka. Sa druge strane, Aurora IP je nezavisna od proizvođača, te se ovaj protokol lako implementira na bilo koji FPGA uređaj koji podržava SFP

komunikaciju. U jednoj tipičnoj aplikaciji, ovaj protokol je sjajan jer pruža širok protok podataka i zadovoljava velik opseg korisnika.

Brzina prenosa podataka je do jednog miliona primljenih ili poslanih poruka u jednoj sekundi. Uzevši sve to u obzir, u uređajima najnovije generacije, latencija je manja od 1 mikrosekunde (slika 6). Cela komunikacija je pouzdana jer je potkrepljena sa sinhronizacijom i mehanizmom zaštite od prekida toka komunikacije što je veoma bitno kod nekih mernih sistema gde je zaštita od greške usled prekida u realnom vremenu od velikog značaja.



Slika 6 – Analiza slanja i primanja promenljivih

Svi zahtevi koji su na početku zadati i predviđeni je ovaj digitalni dizajn ispunio, ali ostaje prostora za unapređenje jer su ostavljeni rezervisani biti u zaglavlju poruke, koji nisu iskorišćeni. Takođe, kako je ovo otvoren protokol, on nije dovoljno fleksibilan, što se utvrdilo tek pri samom kraju izrade istog. Tehnički je limitiran na Aurora IP 8B/10B jezgro, i nije pružena veća generalizacija problema u smislu da je ovo Aurora jezgro lako zameniti sa nekim drugim te ostaje kao tegoba za ubuduće ako se bude želelo prelaziti na još bržu podržanu komunikaciju kao što je 10 ili 20 Gbs. Ostaje prostora pre svega za unapređenje u protočnoj obradi podataka, kako bi bilo lakše za alate, koji služe za implementaciju dizajna, kao i mogućnost boljeg iskorišćenja dodatnog kanala za komunikaciju. Kako je ova realizacija isključivo u hardverskom delu, prepoznata je potreba da se deo prebaci i u sofver, pre svega zbog lakše konfiguracije protokola, a i dobijanju u nekim delovima na performansama.

## 5. LITERATURA

- [1] <https://community.fs.com/blog/sfp-module-what-is-it-and-how-to-choose-it.html>
- [2] <https://docs.xilinx.com/v/u/2019.2-English/ug835-vivado-tcl-commands>
- [3] <https://docs.xilinx.com/v/u/2019.2-English/ug939-vivado-designing-with-ip-tutorial>