Implementacija paketskog komutatora na FPGA čipu

Luka Milinković, *Member IEEE*, Zoran Čiča i Aleksandra Smiljanić, *Member IEEE*

***Sadržaj* — U radu je prikazana jedna realizacija paketskog komutatora, koji se sastoji od elastičnog bafera na ulazu i komutacione matrice *N*x*N*. Ceo projekat je testiran na brzim, gigabitskim Rocket IO interfejsima Xilinx FPGA čipa. Za testiranje su korišćeni protokoli za prenos podataka Aurora, Rapid IO i SATA, pri brzini od 2,5 Gb/s. Testovi su rađeni i softverski u Xilinx simulatoru i hardverski na Xilinx razvojnoj ploči sa Virtex-5 čipom koji poseduje Rocket IO interfejse. Na kraju, u radu je prikazana i iskorišćenost resursa testiranog čipa u zavisnosti od veličine sviča.**

***Ključne reči* — Čip, elastični bafer, FPGA, implementacija, paketski komutator, Rocket IO.**

I UVOD

A. Paketski komutator

FUNKCIONISANjE računarskih mreža danas se gotovo ne može zamisliti bez rutera visokog kapaciteta i većeg broja ulazno/izlaznih portova, koji bi mogli omogućiti povezivanje više uređaja u mreži i prenos saobraćaja visokog intenziteta. Ovakvim ruterima je neophodan i neblokirajući paketski komutator da bi zadovoljili zahteve korisnika u pogledu kapaciteta i kvaliteta servisa [1], [2]. U ovom radu je realizovan neblokirajući krosbar paketski komutator.

Prva generacija paketskog komutatora se zasniva na komutatorima sa izlaznim, odnosno zajedničkim baferom [3]. Mana ovih komutatora je ta što je njihova skalabilnost mala. Nakon njih pojavili su se paketski komutatori sa baferima na ulazu i paketskim krosbarom. Kod njih se paketi smeštaju u ulazne bafere, a zatim na algoritmom definisan način prosleđuju na izlazne portove komutatora. Krosbar je neblokirajuća komutaciona struktura, koja može istovremeno da prenosi više paketa sa ulaza na izlaze, pri čemu je u nekom trenutku moguće preneti najviše jedan paket sa jednog ulaza, i na jedan izlaz [4]. Ovi paketski komutatori su najskalabilniji jednostepeni komutatori, tj. podržavaju najviše kapacitete. U cilju postizanja veće skalabilnosti, paketi koji dolaze u komutator se dele na ćelije iste dužine čije trajanje zovemo slot. Paketski komutator se rekonfiguriše u svakom slotu a prema određenom algoritmu.

B. FPGA čipovi

Hardverski dizajn koji predlažemo u ovom radu je implementiran na Xilinx čipu. Jedan od najpoznatijih proizvođača FPGA uređaja jeste upravo Xilinx. Ova kompanija razvija dve serije čipova: Spartan i Virtex. Spartan serija obuhvata nižu i srednju klasu čipova, slabijih performansi, ali povoljnije cene, a Virtex serija je viša klasa čipova, boljih performansi, ali zato i više cene. Jedna od najnovijih Xilinx serija čipova je Virtex-5 [5]. Ovi čipovi mogu da podrže prenos podataka velikim brzinama preko svojih Rocket IO interfejsa gigabitskog protoka, koji su važni za implementaciju krosbara visokog kapaciteta. Postoje dve vrste Rocket IO interfejsa, i to GTP i GTX. Na jednom čipu se nalaze samo GTP ili GTX interfejsi. U tabeli 1 su prikazane karakteristike serije čipova Virtex-5 u pogledu broja i brzine Rocket IO interfejsa, kao i broja IO pinova.

TABELA 1: KARAKTERISTIKE XILINX ČIPOVA SA GTP I GTX INTERFEJSIMA

Rocket IO	Broj IO pinova	Broj Rocket IO portova	Najveća brzina prenosa po Rocket IO portu
GTP	172 - 960	4 - 24	3,75 Gb/s
GTX	360 - 960	8 - 48	6,5 Gb/s

C. Projekat

U ovom radu je prikazana realizacija *N*x*N* paketskog komutatora, gde *N* predstavlja broj ulaza, odnosno izlaza. Paketski komutator se sastoji iz elastičnog bafera i krosbara. Pored krosbara u radu je realizovan i elastični bafer, koji prihvata asinhroni dolazak ćelija sa *N* ulaza i sinhrono ih prosleđuje ka komutacionoj matrici. Slot se upisuje u zaglavlje ćelije da bi se na prijemu u krosbaru ćelije mogle sinhronizovati na odgovarajući način. Na *N* ulaza krosbara sinhrono dolaze ćelije koje se komutiraju u istom slotu.

Dizajn sa Rocket IO interfejsima, elastičnim baferom na ulazu i paketskim krosbarom je testiran prvo u Xilinx simulatoru, a zatim i na razvojnoj ploči ML507 sa Virtex-5 čipom XC5VFX70T. Takođe su analizirane i performanse dizajna u pogledu resursa koje zauzima za različite veličine rutera.

Rad je finansiran od strane Ministarstva za nauku i tehnološki razvoj Republike Srbije.

Luka B. Milinković, Elektrotehnički fakultet u Beogradu, Srbija; e-mail: luka.milinkovic@etf.rs

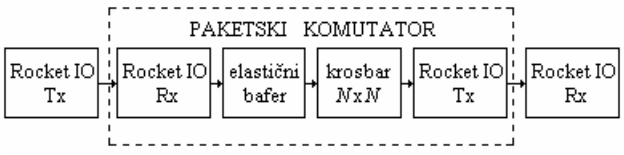
Zoran G. Čiča, Elektrotehnički fakultet u Beogradu, Srbija; e-mail: cicasyl@etf.rs

Aleksandra Smiljanić, Elektrotehnički fakultet u Beogradu, Srbija; e-mail: aleksandra@etf.rs

II OPIS DIZAJNA

A. Paketski komutator

Paketski komutator je realizovan u VHDL kodu u okviru Xilinx softvera ISE Foundation. Pošto su korišćeni i Rocket IO interfejsi, VHDL kod za njih je generisan u Xilinx programu Core Generator [6]. Slika 1 prikazuje dizajn. On se sastoji od predajne i prijemne strane sa Rocket IO interfejsima i paketskog komutatora, koji takođe ima Rocket IO interfejse. Paketski komutator prima dolazne podatke na svoje brze interfejse, a zatim ih prikuplja u elastični bafer. Odatle se podaci očitavaju i šalju u krosbar, ali tek kada stignu sve ćelije za zadati slot. Nakon konfiguracije krosbara, ćelije se kroz njega šalju na predajne Rocket IO blokove odgovarajućih izlaza.

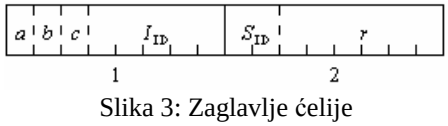


Slika 1: Blok šema realizovanog rešenja

Paketi, koji treba da se prenesu se dele na blokove fiksne dužine, na primer 64 bajta. Ovi blokovi podataka se utiskuju u ćelije dužine 68 bajtova kao na slici 2. Prvi i 68. bajt su flegovi, koji se koriste za razgraničenje ćelija, odnosno definisanje početka i kraja ćelije. Za fleg je uzet niz bita 10111100, odnosno u dekadnom zapisu broj 188, koji spada u grupu od 12 kontrolnih karaktera koji se kod 8b/10b kodovanja koriste za posebno definisane namene, a između ostalog i kao flegovi. Oznaka pomenutog karaktera je K28.5 [6]. Podrazumeva se da se svaki bajt ćelije koduje 8b/10b kodovanjem pri prenosu kroz ruter. Dva bajta posle početnog flega se koriste za zaglavlje, a naredna 64 bajta se koriste za prenos korisnog sadržaja, odnosno jednog dela sadržaja koji se nalazi u paketu.

Slika 2: Izgled ćelije

Na slici 3 je prikazan izgled i sadržaj zaglavlja. Prvi bajt zaglavlja se sastoji od bita *a* – označava prvu ćeliju paketa kada mu je vrednost 1, bita *b* – označava poslednju ćeliju paketa kada mu je vrednost 1, bita *c* – označava nepostojanje ćelije, odnosno praznu ćeliju kada mu je vrednost 1, i 5 bita, koji označavaju izvorišni port ID, I_{ID}. Prva dva bita drugog bajta označavaju Slot ID, S_{ID}, odnosno koja je ćelija po redu koja se šalje. Samim tim numerisanje ćelija se kreće od 0 do 3 i tako u krug. Preostalih 6 bita su rezervni biti, *r* biti (r_1, r_2, r_3, r_4, r_5 i r_6), koji će možda zatrebati za neku budući upotrebu.

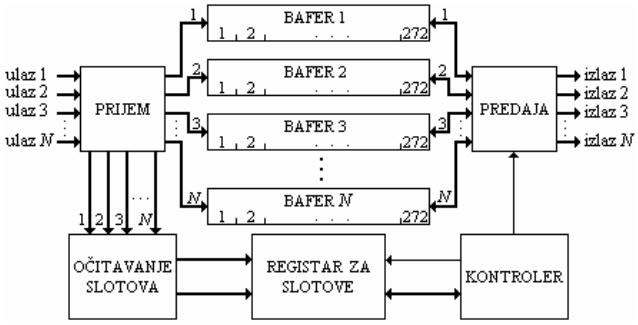


Slika 3: Zaglavlje ćelije

B. Elastični bafer

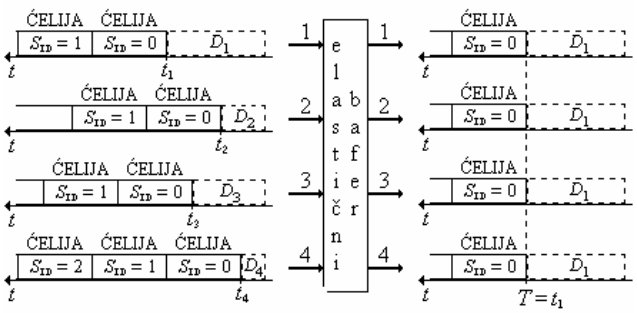
Ako na svih *N* ulaza u paketski komutator ćelije ne stižu istovremeno, na primer zbog kašnjenja izazvanog različitim dužinama putanja na ploči, može se desiti da dođe do greške pri ukrštanju paketa u krosbaru. Da se ovo ne bi desilo realizuje se elastični bafer na ulazu u krosbar. On prihvata ćelije sa ulaza, smešta ih u bafer i tek kada stignu sa svih *N* ulaza ćelije istog slota, počeće da ih prosleđuje sinhrono na svoje izlaze, odnosno ulaze krosbara. Na taj način će na sve ulaze u krosbar uvek istovremeno stizati ćelije istog slota.

Elastični bafer može da prikupi najviše po *S*=4 ćelije sa svakog ulaza, odnosno po *P*=272B, slika 4. On se sastoji od više blokova gde svaki ima svoju funkciju. Prijemni blok prima bajt po bajt ćelije sa svakog ulaza i upisuje ih redom u odgovarajuće bafere, realizovane preko FIFO memorije, a zaglavlje prosleđuje do bloka za očitavanje slotova. U ovom bloku se iz zaglavlja očitava informacija o rednom broju slota, S_{ID}, koja se zatim upisuje u registar. Registar za slotove ima matričnu strukturu, gde redovi označavaju slotove, *S*=4, a kolone ulaze, *N*. Kada se dizajn resetuje matrica *S*x*N* postaje nula-matrica. Pošto blok za očitavanje slotova uzme informaciju o rednom broju slota on upisuje jedinicu u matricu *S*x*N* na odgovarajuću poziciju posmatrajući koji je slot u pitanju i na koji je ulaz došla ćelija. Blok kontroler na osnovu vrednosti brojača, *j*, zna za naredni slot koji treba da se primi i u matrici *S*x*N* proverava da li je red *j* popunjen sa svim jedinicama, odnosno da li je na svaki od *N* ulaza počela da stiže ćelija očekivanog slota. Kada se to ispuni brojač se inkrementira, a kontroler pristupa registru za slotove i redu *j* matrice *S*x*N* dodeljuje sve nule. Istovremeno, kontroler šalje informaciju ka bloku predaja da počne sinhrono da šalje ćelije iz bafera ka krosbaru, bajt po bajt.



Slika 4: Šema elastičnog bafera

Funkcionisanje elastičnog bafera prikazano je na slici 5, gde je uzeto da je *N*=4. Vremena t_1, t_2, t_3 i t_4 pokazuju trenutke kada su ćelije odgovarajućih ulaza počele da stižu u elastični bafer, a vreme *T* je trenutak kada su ćelije počele da izlaze iz elastičnog bafera i da se sinhrono prosleđuju ka krosbaru. Vremena kada ćelije počinju da stižu u elastični bafer zadovoljavaju sledeću relaciju: $t_4<t_2<t_3<t_1$. To znači da prvo počnu da stižu ćelije na četvrtom ulazu, a poslednje ćelije na prvom ulazu. Ako se zanemari obrada koja postoji u elastičnom baferu onda bi trebalo da važi $T=t_1$, a najveće kašnjenje je ustvari razlika trenutaka dolazaka prve i poslednje ćelije istog slota, t_1-t_4.



Slika 5: Elastični bafer

III TESTIRANJE OPISNOG DIZAJNA

A. Alati korišćeni pri testiranju

Dizajn je testiran na Virtex-5 čipu sa GTX interfejsima, XC5VFX70T [5]. Ovaj čip poseduje 640 ulazno/izlaznih pinova i 16 GTX interfejsa, odnosno 16 brzih primopredajnika koji mogu da obezbede protok do 6,5 Gb/s. Brzi interfejsi su na čipu grupisani u grupe od po dva ulazno/izlazna primopredajnika i nazvani su GTX_DUAL. Na taj način ovaj čip ima 8 GTX_DUAL primopredajnika. Za generisanje koda potrebnog za rad GTX interfejsa, izbor protokola i definisanje željene brzine korišćen je program Core Generator, a za testiranje je korišćen program ISE Foundation. U programu ISE Foundation je takođe napisan VHDL kod dizajna i sprovedeno je softversko testiranje.

Kod hardverskog testiranja korišćena je razvojna ploča ML507 sa pomenutim čipom, XC5VFX70T. Za potrebe ovog testiranja u programu ISE Foundation je generisan bit strim koji je preko USB kabla spušten na ploču, a rezultati testiranja su posmatrani pomoću Xilinx programa ChipScope. U VHDL kodu se mogu definisati signali koji se prosleđuju u program ChipScope, i na taj način je moguće posmatrati samo potrebne signale.

Za bitsku brzinu je izabrana vrednost od 2,5 Gb/s, jer je to maksimalna bitska brzina štampane ploče na koju će biti postavljen dizajnirani krosbar.

B. Testirani protokoli za komunikaciju

Pre samog testa opisanog dizajna bilo je neophodno testirati protokole za prenos podataka, da bi se proverilo da li protokoli funkcionišu ili ne. Testirano je 5 protokola: Aurora, PCI-Express, Rapid IO, SATA i XAUI [6]. Svaki od ovih protokola može da podrži i veće brzine, ali je trebalo proveriti da li ispravno rade pri brzini od 2,5 Gb/s.

U tabeli 2 su prikazani protokoli koji su se koristili pri testiranju na 2,5 Gb/s. Pored pomenutih podešavanja u programu Core Generator je moguće izabrati i referentni takt. Ovaj takt određuje frekvenciju na kojoj radi dizajn, a koristi se i za formiranje željene bitske brzine za prenos podataka. Prvo je izabran takt od 100 MHz, a zatim, pri ponovljenom testu i takt od 125 MHz. Kod svakog od testiranih protokola za brzinu od 2,5 Gb/s mogla je da se izabere jedana od sledećih vrednosti takta: 100 MHz, 125 MHz, 166,67 MHz, 200 MHz, 250 MHz, 333 MHz ili 500 MHz, ali za kasniju upotrebu bili su od interesa samo taktovi od 100 MHz i od 125 MHz, kao što se vidi u drugoj koloni tabele 2. Protokoli Aurora, Rapid IO i SATA su jedini pravilno radili pri brzini od 2,5 Gb/s i kada je referentni takt bio 100 MHz i kada je on bio 125 MHz. To je naznačeno u trećoj koloni.

TABELA 2: PROTOKOLI I REFERENTNI TAKT

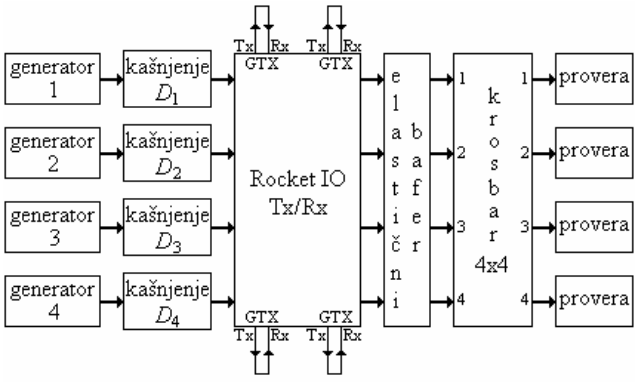
Protokoli	Referentni takt (MHz)	Rad na 2,5 Gb/s
Aurora	**100**, **125**, 166.67, 200, 250, 333, 500	DA
PCI-Express		NE
Rapid IO		DA
SATA		DA
XAUI		NE

C. Testiranje

Pošto su na definisanoj brzini pravilno radila tri protokola Aurora, Rapid IO i SATA, onda je svaki od njih zasebno testiran u opisanom dizajnu.

Pomoću programa Core Generator podešen je jedan od protokola, protok od 2,5 Gb/s, frekvencija referentnog takta od 100 MHz i izabrani su GTX_DUAL primopredajnici. Sa korišćenog čipa, XC5VFX70T, izabrana su 3 GTX_DUAL primopredajnika. Kod GTX_DUAL_X0Y3 interfejsa korišćena su oba primopredajnika, i nulti i prvi, a kod GTX_DUAL_X0Y4 i GTX_DUAL_X0Y5 korišćen je samo prvi primopredajnik jer se neki interfejsi ne mogu koristiti za ove testove. Na ovaj način su iskorišćena 4 GTX primopredajnika, pa krosbar ima strukturu 4x4.

Na slici 6 je prikazan šematski prikaz testa. Generatori generišu ćelije veličine 68 bajta, a zatim se one prosleđuju do blokova za kašnjenje, gde se ćelije ulaza *i* zakasne za određeni broj bajtova, D_i. Nakon toga se bajt po bajt ćelije prosleđuje do Rocket IO Tx/Rx bloka gde se primljeni podaci „zavrte“ preko GTX primopredajnika, odnosno pošalju se preko GTX Tx kanala, a prime preko GTX Rx kanala na Rocket IO Tx/Rx bloku. Na ovaj način se ispitalo i funkcionisanje brzih interfejsa. Sada se primljeni podaci prosleđuju do elastičnog bafera, koji ih zatim sinhrono šalje do krosbara gde se dolazne ćelije ukrštaju kako je definisano u VHDL kodu. Paketi koji izađu iz krosbara dolaze do bloka za proveru, gde se proverava da li su podaci, koji se i očekuju uspešno primljeni.



Slika 6: Šematski prikaz testa

Da bi se lakše pratili generisani podaci nakon konfigurisanja krosbara, i utvrdila ispravnost komutacije, svaki od 4 generator blokova formira uvek istu ćeliju koja se razlikuje za različite generatore. Tako postoje 4 vrste ćelija. Prva ćelija među korisnim sadržajem prenosi niz osmobitnih zapisa brojeva od 0 do 63. Druga prenosi niz osmobitnih zapisa brojeva od 63 do 0. Treća prenosi niz osmobitnih zapisa brojeva od 64 do 127, a četvrta prenosi niz osmobitnih zapisa brojeva od 127 do 64.

Ćelije iz generator blokova dolaze do blokova za kašnjenje, svaka do svog. Blokovi za kašnjenje su korišćeni da bi se veštački simuliralo kašnjenje ćelija i da bi se videlo kako će elastični bafer reagovati na to. Sprovedeno je više različitih testova u kojima su se vrednosti konstanti kašnjenja menjale. Jedan od testova je prikazan na slici 5. Kašnjenja u ovom primeru su sledeća: D_1=100B, D_2=45B, D_3=68B i D_4=25B. Ovako zakašnjene ćelije dolaze do elastičnog bafera, koji počinje prvo da prima ćelije sa 4. ulaza, a zatim i sa 2., 3. i 1., smešta ih u odgovarajuće bafere i očitava slotove pristiglih ćelija, slika 4. Kontroler elastičnog bafera proverava da li su u svaki bafer počele da se upisuju ćelije sa istim, narednim očekivanim slotom. Ako je odgovor potvrdan, kao što je u ovom primeru, šalje informaciju bloku predaja da može da počne sa ispisom ćelija. Ovaj blok čita ćelije iz 4 bafera i prosleđuje ih sinhrono na odgovarajuće izlaze. Na svaki izlaz elastičnog bafera se prosleđuje bajt po bajt odgovarajuće ćelije.

U krosbar dolaze bajtovi iz elastičnog bafera, a krosbar se konfiguriše na unapred definisan način. U ovom primeru sa krosbarom 4x4 testirane su sve konfiguracije krosbara, K=4!=24. Nakon ukrštanja bajtovi ćelija stižu do bloka za proveru gde se proverava da li su pristigli podaci isti kao i oni očekivani.

I u ovom i u svim ostalim primerima elastični bafer je obavljao svoju funkciju kao što treba. Ćelije koje mu stižu asinhrono na njegove ulaze prosleđivao je sinhrono na svoje izlaze sa minimalnim kašnjenjem. Krosbar je takođe radio kako treba. Pri testu svake konfiguracije krosbara u blok za proveru su stizali podaci koji su se i očekivali.

D.Performanse dizajna

U tabeli 3 je prikazana procentualna iskorišćenosti čipa Virtex-5 XC5VFX70T, kada se na njega spusti opisani dizajn sa krosbarom 4x4, 8x8 i 16x16. U tabeli 3 je prikazana odvojeno zauzetost elastičnog bafera i krosbara, a njihov zbir predstavlja ukupnu iskorišćenost resursa čipa.

Na čipu se nalazi 44800 registara, 44800 logičkih elemenata i 5328kB FIFO memorije (296 blokova od po 18kB) [5]. U odnosu na krosbar elastični bafer zauzima veći broj registara i logičkih elemenata. Kada se broj interfejsa udvostruči količina logičkih elemenata, koju zauzima elastični bafer se poveća dva puta, a količina koju zauzima krosbar se poveća približno četiri puta. Promena količine registara koje zauzmu ova dva bloka je linearno srazmerna sa povećanjem broja interfejsa.

Krosbar ne zauzima FIFO memoriju, dok je elastični bafer zauzima, jer nju koristi kao bafer za čuvanje ćelija koje mu asinhrono dolaze na ulaze pre neko što ih sinhrono prosledi ka krosbaru. Količina zauzete memorije je linearno srazmerna sa povećavanjem broja ulaza/izlaza.

Za referentni takt dizajna, kao što je već rečeno, izabrana je frekvencija od 100 MHz i ona je podržana kod sve tri vrste dizajna.

TABELA 3: ISKORIŠĆENOST RESURSA NA ČIPU

NxN		4x4	8x8	16x16
Registri	Elastični bafer	1,48%	2,97%	5,94%
	Krosbar	0,07%	0,14%	0,29%
Logički elementi	Elastični bafer	1,04%	2,09%	3,49%
	Krosbar	0,19%	0,84%	3,39%
FIFO memorija	Elastični bafer	1,35%	2,7%	5,41%
	Krosbar	0%	0%	0%
Podržana frekvencija od 100MHz		DA	DA	DA

IV ZAKLJUČAK

U radu je prikazan jedan deo velikog projekta Internet rutera visoke propusne moći. Taj deo koji čini elastični bafer na ulazu i komutaciona matrica je realizovan, testiran i potvrđen da radi sa gigabitskim, Rocket IO interfejsima. Testovi su i u simulatoru i na razvojnoj ploči radili kako treba za svaki od tri testirana protokola: Aurora, Rapid IO i SATA.

LITERATURA

[1] M. Petrović, M. Blagojević, A. Smiljanić, „Dizajn kontrolera neblokirajućeg IP rutera visokog kapaciteta“, *XIII Telekomunikacioni forum - TELFOR 2005*., Beograd 2005, strana 5

[2] Miloš Petrović, Aleksandra Smiljanić, Miloš Blagojević, „Design of the Switching Controller for the High-Capacity Non-Blocking Internet Router“, *IEEE Transactions on VLSI*, 2008.

[3] M. Petrović, A. Smiljanić, „Optimization of the Scheduler for the Non-Blocking High-Capacity Router“ *IEEE Communication Letters*, vol. 11, no. 6, jun 2007.

[4] M. Petrović, M. Blagojević, V. Joković, A. Smiljanić, „Design, implementation, and testing of the controller for the terabit packet router“, in *Proc. IEEE ICCCAS 2006*, Vol. 3, strane 1701-1705

[5] Xilinx Corporation, „Virtex-5 Family Overview“, septembar 2008., www.xilinx.com/support/documentation/data_sheets/ds100.pdf

[6] Xilinx Corporation, „Virtex-5 FPGA RocketIO GTX Transceiver“, novembar 2008., http://www.xilinx.com/support/documentation user_guides/ug198.pdf

ABSTRACT

In this paper one implementation of the *N*x*N* cross-bar with the input elastic buffer is presented. This implementation was tested on the Xilinx FPGA chip with Rocket IO transceivers. For tests, the communication protocols Aurora, Rapid IO and SATA were used. The implementation was tested in the Xilinx simulator and on the Xilinx development board ML507, with Virtex-5 chip and Rocket IO transceivers. The device resource utilization for different switch sizes is presented in the paper.

IMPLEMENTATION OF THE PACKET SWITCHING FABRIC ON THE FPGA CHIP

Luka Milinković, Zoran Čiča i Aleksandra Smiljanić