

# Dizajn i implementacija množača realnih brojeva u formatu IEEE-754

Miloš N. Jorgovanović, Obrad M. Đorđević

Mentor: dr Jelena S. Popović

**Sadržaj** — U ovom radu je opisan hardverski dizajn množača realnih brojeva u formatu opisanim standardom IEEE-754. Množać je realizovan kao *pipeline* struktura nivoa 7. Kao takav on se može koristiti u aplikacijama u kojima je potrebno izvršiti sekvencu množenja, pošto je efektivno trajanje množenja u tom slučaju jedan takti interval. Pored opisa dizajna koji je realizovan u VHDL-u, dati su i osnovni podaci o implementaciji množača na Xilinx FPGA čipu Virtex4 familije.

**Gljučne reči** — ASIC, CLA, FPGA, IEEE-754, množać, *pipeline*.

## I. UVOD

**MNOŽAČ** realnih brojeva se može javiti kao komponenta u realizaciji različitih DSP algoritama, kao i pri projektovanju različitih matematičkih i namenskih aritmetičkih koprocesora, koji se često koriste kao delovi složenijih sistema za obradu signala u digitalnim telekomunikacijama. Pored toga, potreba za ovakvim množačem se može ispoljiti i pri brznoj obradi i kontroli podataka.

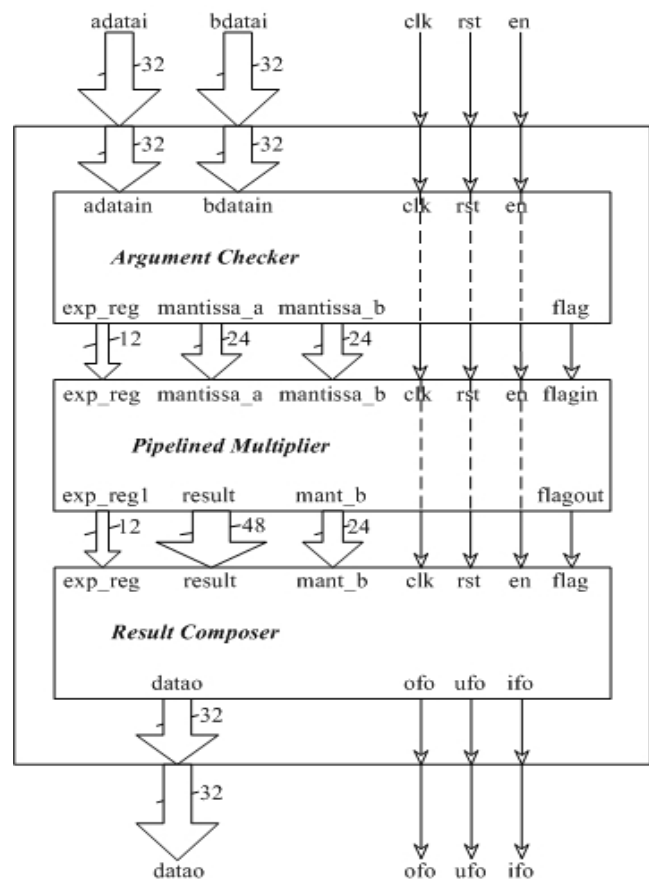
Struktura množača je projektovana tako da je izvršeno potpuno prilagođenje standardu IEEE-754 za 32b brojeve [1]. Ovaj standard je ukratko opisan u drugom poglavlju, pri čemu su date i formule za konvertovanje brojeva iz formata IEEE-754 u decimalni zapis. Pored toga, vodilo se računa da interfejs strukture bude što jednostavniji, da ne bude potrebno dodatno programiranje, i da se postigne puna tačnost rezultata. Kompletan hardverski dizajn množača realizovan je u VHDL-u, sa ciljem maksimizacije brzine i minimizacije resursa.

Osnovna prednost realizacije opisane u ovom radu se ispoljava pri množenju dugačkih sekvenci brojeva, jer se više množenja vrši paralelno (novi činiooci se prihvataju pre nego što je izračunat rezultat prethodnih množenja), čime se ostvaruje veća brzina rada. Ovo se postiže *Pipeline* strukturom množača koja je sposobna da prihvata nove podatke (činioce) na svaki signal takta, i da daje rezultat (proizvod) na svaki signal takta (konceptualno, *Pipeline* je opisan u [2]). Na taj način se dobija efektivno trajanje množenja od jednog takt intervala, a na svaki rezultat se čeka 7 takti intervala od prihvatanja činilaca, što odgovara nivou ulančavanja u strukturi. Opis interfejsa i

svakog od strukturnih blokova dizajna, pri čemu je posebna pažnja posvećena centralnom *Pipeline* bloku koji vrši množenje mantisa, dat je u trećem poglavlju.

U četvrtom poglavlju su dati osnovni podaci o implementaciji dizajna na FPGA čipu. Pored toga, jasno je naglašena karakterističnost prikazane realizacije, kao ASIC čipa. Takođe priložen je i dijagram koji demonstrira ispravan rad implementiranog množača na maksimalnoj radnoj frekvenciji.

Projekat na kome se zasniva ovaj rad je realizovan u okviru predmeta Projektovanje Integriranih Kola 2, na 5. godini smeru za Elektroniku na Elektrotehničkom fakultetu u Beogradu. Specifikacije projekta su date u [3], odakle je preuzeta i predložena ideja podele strukture množača na blokove (sl. 1).



Sl. 1. Blokvska struktura množača realnih brojeva.

## II. IEEE-754 STANDARD

IEEE-754 standard je jedan od najčešće korišćenih formata za prikazivanje realnih brojeva. U ovom radu je

Miloš N. Jorgovanović, Elektrotehnički fakultet u Beogradu, Srbija (e-mail: mjorgovanovic@gmail.com).

Obrad M. Đorđević, Elektrotehnički fakultet u Beogradu, Srbija (e-mail: lobrad@gmail.com).

korišćen deo standarda koji opisuje 32-bitni zapis brojeva [1], [4]. Prema navedenom standardu zapis se vrši na sledeći način:

- najstariji bit (MSB) predstavlja znak željenog broja (*sign bit*),
- narednih 8 bita predstavljaju eksponent broja 2 (*exponent*), i
- preostala 23 bita predstavljaju razlomljeni deo mantise broja (*mantissa*).

Ovim standardom se mogu prikazati normalizovani, denormalizovani brojevi, nula, beskonačnosti i *Not-a-Number* (NaN). Denormalizovani brojevi se koriste za prikazivanje ekstremno malih brojeva koji ne ulaze u opseg za normalizovane brojeve, i njihov nerazlomljeni deo je 0. Normalizovani brojevi imaju nerazlomljeni deo 1.

Pretvaranje iz IEEE 754 zapisa u standardni decimalni zapis broja vrši se na sledeći način:

$$N_{norm} = (-1)^S \cdot 2^{E-127} \cdot (1 + M(22) \cdot 2^{-1} + \dots + M(0) \cdot 2^{-23}) \quad (1)$$

za normalizovane brojeve, dok je formula za denormalizovane:

$$N_{denorm} = (-1)^S \cdot 2^{-126} \cdot (0 + M(22) \cdot 2^{-1} + \dots + M(0) \cdot 2^{-23}) \quad (2)$$

pri čemu je sa  $N$  označena decimalna vrednost broja, a sa  $S, E$  i  $M$  bit znaka, vrednost eksponenta uvećana za 127 prema standardu IEEE-754 (*biasing*) i razlomljeni deo mantise, respektivno [5].

### III. DIZAJN MNOŽAČA

Množać je realizovan strukturnim dizajnom, dakle podelom na funkcionalne blokove, kao što je to prikazano na sl. 1. Ulazni portovi su signal takta (*clk*), reset (*rst*) i signal dozvole (*en*), kao i portovi za činioce (*adatai* i *bdatai*). Izlazni portovi su port za proizvod (*datao*), i signali prekoračenja (*ofo*, *ufo*, *ifo*), koji redom označavaju da je rezultat prekoračio opseg predviđen standardom sa gornje strane, sa donje strane, odnosno da je rezultat NaN.

U nastavku je opisan dizajn svake od osnovnih komponenti množaća.

#### A. Argument Checker

U okviru ovog bloka se najpre vrši prijem ulaznih podataka u prihvatne registre, a zatim sledi prvi nivo obrade. To podrazumeva sabiranje eksponenata, formiranje bita znaka, i pripremanje mantisa za njihovo množenje, koje će se obaviti u narednom bloku. Bit znaka se dobija prostom primenom xor operacije na bitove znaka ulaznih brojeva, dok se formiranje mantisa svodi na dodavanje bita „1“ kao MSB-a ukoliko je broj normalizovan, odnosno bita „0“ ukoliko je broj denormalizovan (preostalih 23 bita koji se očitavaju direktno iz ulaznih 32 bita predstavljaju, prema standardu IEEE-754, razlomljeni deo – deo iza tačke).

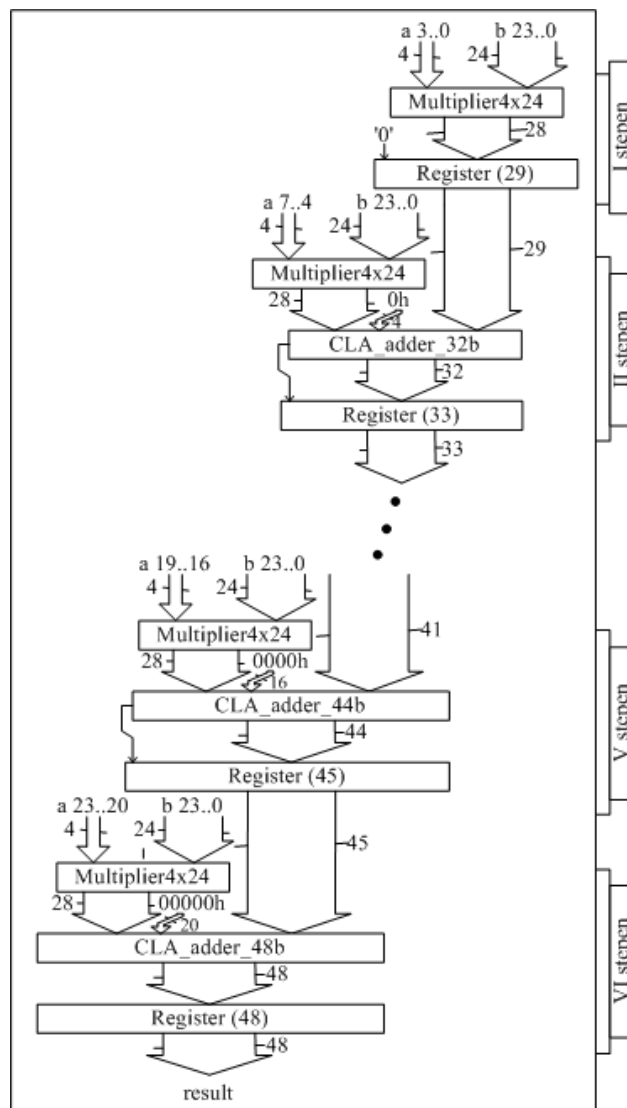
Preliminarno računanje eksponenta je malo složenije, i svodi se na osim prostog sabiranja eksponenata ulaznih brojeva, i na *biasing* rezultata [1]. Naime, za ulazne eksponente  $x_1$  i  $x_2$ , koji nakon *biasing*-a imaju format  $x_1 + 127$  i  $x_2 + 127$  respektivno, treba dobiti rezultat u formatu  $x_1 + x_2 + 127$ . Dakle, pored sabiranja  $x_1 + 127 + x_2 + 127$ , mora se izvršiti oduzimanje za jednu *bias* vrednost, tj. 127. Binarno – treba oduzeti „01111111“,

što je za ovaj format ekvivalentno sabiranju sa „10000001“. To se veoma lako realizuje bez korišćenja dodatnog sabirača tako što se u sabirač eksponenata kao  $C_{in}$  uvede „1“, a na kraju se invertuje vrednost MSB bita rezultata. Pri ovom sabiranju se detektuje prekoračenje, što se signalizira aktiviranjem internih *ofo*, *ufo* i *ifo* signala.

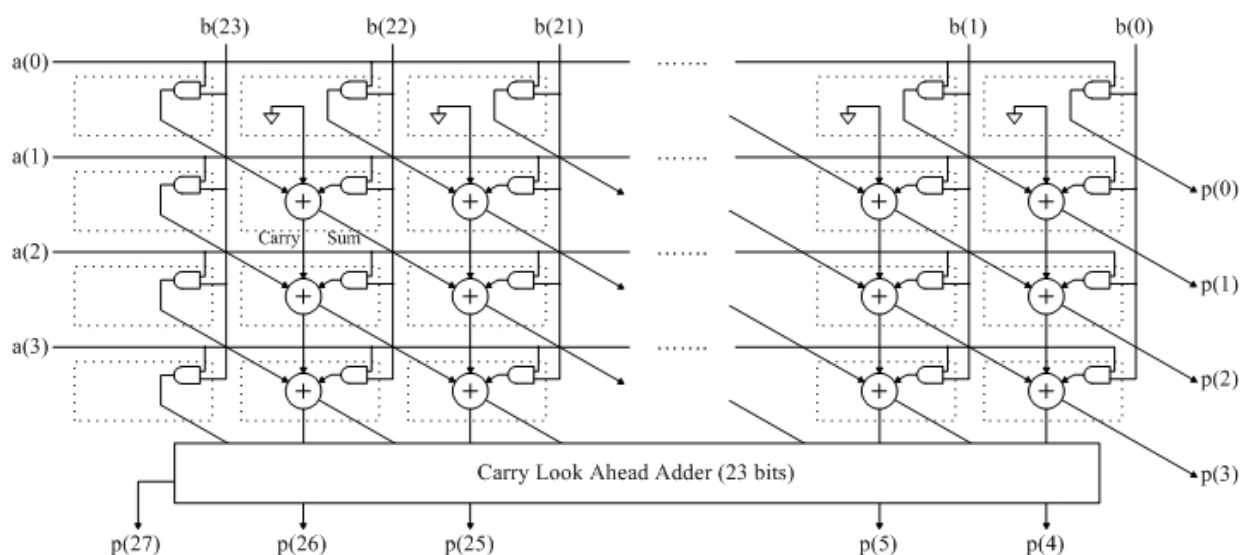
Ukoliko se još pri računanju eksponenta rezultata ustanovi da će rezultat biti beskonačnost ili NaN, izlaznom bloku će se proslediti ova informacija i on će zanemariti rezultat množenja centralnog bloka.

#### B. Pipelined Multiplier

Ovaj blok predstavlja srž cele strukture, i pri njegovom projektovanju se maksimalno vodilo računa o maksimizaciji brzine i minimizaciji resursa. Stoga je i odabrana *pipeline* struktura kao odličan način da se višestruko smanji efektivno vreme trajanja množenja dva 24b-na broja, što presudno utiče i na brzinu cele strukture. Množenje se izvodi u 6 nivoa ulančavanja, u svakom nivou pomoću množaća 4b\*24b (sl. 2). Blok za množenje 4b\*24b (prikazan na sl. 3) je realizovan kao verzija paralelnog (odnosno *carry-save*) množaća, pri čemu je sabirač u poslednjem stepenu realizovan kao *Carry Look-Ahead* (CLA), da bi se dobilo što brže kolo za množenje.



Sl. 2. Strukturni prikaz *Pipelined Multiplier-a*.



Sl. 3. Šematski prikaz realizacije množača  $4b \cdot 24b$ .

Naime, maksimalno kašnjenje množača prikazanog na sl. 3 se izračunava prema najdužoj logičkoj putanji, a to je očigledno putanja koja vodi ka izračunavanju MSB bita proizvoda ( $p(27)$ ). Pored kašnjenja kroz tri potpuna sabirača po vertikali, presudno je kašnjenje po horizontali, kroz prikazani 23b-ni sabirač. Klasična realizacija paralelnog množača ([6]) predviđa upotrebu *Carry-Ripple* sabirača koji bi se u ovom slučaju sastojao od 23 potpuna sabirača. Pošto je kroz takav sabirač vreme propagacije veoma dugo, nađena je optimalnija struktura koja će ubrzati ovo sabiranje, a time i množenje – korišćenjem CLA sabirača. Naime, 23b-ni *Carry Look-Ahead* sabirač je realizovan kao lanac šest CLA sabirača (pet 4b-nih i jednog 3b-nog, čija struktura je opisana u [6]), pri čemu je množenje brže za oko 70%, u odnosu na varijantu sa *Carry-Ripple* sabiračem, a količina potrebnih logičkih kola približno ista. Sa druge strane, ukoliko bi se koristio 23b-ni CLA sabirač umesto predloženog lanca, množenje bi trajalo duplo kraće, ali bi bilo potrebno i 15 puta više logike, tako da predloženo rešenje predstavlja kompromis između ova dva oprečna zahteva.

Nakon šest uzastopnih  $4b \cdot 24b$  množenja i odgovarajućih šiftovanja i sabiranja (za sva sabiranja su takođe korišćeni lančani CLA sabirači) u svakom nivou *pipeline*-a (sl. 2), rezultat je dostupan nakon šest ciklusa takta.

### C. Result Composer

U ovom bloku se vrši konačno prilagođenje rezultata množenja standardu IEEE-754 [3], [4], [5]. Pri tome se javljaju i neki specifični slučajevi koji se moraju posebno razmotriti:

- Ukoliko je dobijeni rezultat suviše mali, i mora se predstaviti kao denormalizovani broj, vrši se korekcija eksponenta na „00000000“, prema standardu, i vrši se korekcija mantise.
- Ukoliko je dobijeni rezultat normalizovan broj, on se mora svesti na format koji odgovara standardu, da mantisa predstavlja frakcioni deo iza tačke, a da ispred tačke bude 1. Dakle, mora se pronaći pozicija prve jedinice u rezultatu, uzeti narednih

23b iza nje za mantisu, a eksponent korigovati u zavisnosti od te pozicije.

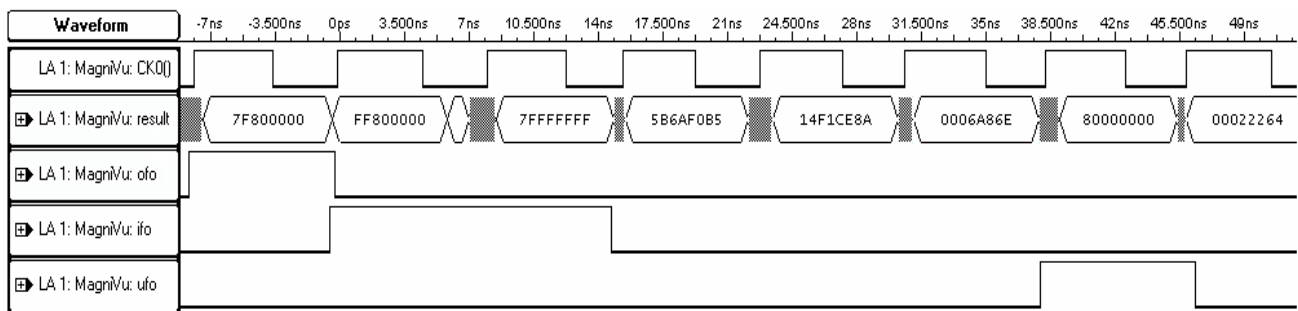
- Ukoliko postoji mogućnost za šiftovanjem mantise u levo, denormalizovan broj se može dobiti i tako što se broj koji je nakon *Argument Checker*-a označen kao *ufo*, predstavi kao denormalizovan.

Pored eksponenta i mantise, znak, *ofo*, *ufo* i *ifo* signali se takođe formiraju u ovom bloku, i celokupan rezultat se pamti u izlaznom registru, koji ujedno predstavlja i sedmi, poslednji nivo ulančavanja. Nakon upisa u registar, podaci se mogu očitavati na silaznu ivicu takta.

## IV. IMPLEMENTACIJA DIZAJNA

Množak je implementiran koristeći Xilinx-ov Virtex4 (XC4VLX25) čip i ploču ML401 [7], [8], [9]. Osnovna logička jedinica Virtex4 čipa je *Slice* – koji sadrži određenu količinu logike, RAM memorije, 2 *Look-up* tabele i 2 Flip-flop-a. Korišćeni čip sadrži i neke karakteristične blokove koji omogućavaju bolju FPGA implementaciju određenog dizajna. Konkretno, od interesa je posmatrati programabilne DSP48 blokove (ukupno 48 na ovom čipu) koji imaju mogućnost da optimalno izvršavaju operacije sabiranja, oduzimanja, množenja, i MAC (*Multiply + Accumulate*) funkciju, i samim tim značajno ubrzaju dizajn i drastično smanje iskorišćenje resursa.

Dijagrami rezultata implementacije dobijeni upotrebom *Logic Analyser*-a prikazani su na sl. 4, pri čemu su, pored signala takta, prikazani signali na izlaznim portovima (proizvod – *result*, i signali prekoračenja - *ofo*, *ufo*, *ifo*). Test primeri za činioce su smešteni u ROM memoriju odvojenu od samog dizajna, i odabrani su tako da prezentuju sve slučajeve obuhvaćene standardom (a naročito karakteristične varijante opisane u prethodnom poglavlju). Konkretno za prvi signal takta na sl. 4 ulazni signali su 755556A5 i 55D55655 (odabrani tako da rezultat pređe dozvoljeni opseg), za drugi 2AAD2AFF i FF800000 (odabrani tako da rezultat bude NaN), i td. Sam hardverski dizajn je realizovan u VHDL-u ([6], [10]), sa namerom da bude optimizovan za korišćenje kao



Sl. 4. Rezultati množenja dobijeni implementacijom na Virtex4 FPGA čipu ( $f_{clk} = 130\text{MHz}$ ).

komponenta nekog složenijeg ASIC čipa, a ne za implementaciju na određenom FPGA čipu. Stoga i ne čudi podatak da predloženo rešenje zauzima 18% raspoloživih *Slice*-eva na pomenutom Virtex4 čipu, ali nije iskorišćen ni jedan DSP48 programabilni blok, koji bi za FPGA realizaciju ovakve strukture bio logičan izbor za operaciju množenja. Maksimalna frekvencija rada koja obezbeđuje pouzdan rad množača za sve slučajeve obuhvaćene standardom je 130MHz, odnosno struktura je u mogućnosti da obavi 130 miliona množenja realnih brojeva u sekundi. Prema [3], maksimalna moguća brzina množača realizovanog na istom čipu je 30% veća, ali je svakako posledica dizajna usmerenog ka FPGA, a ne ASIC implementaciji, što se vidi po korišćenju DSP48 programabilnih blokova (ukupno 4), a samim tim i manjem utrošku logike.

#### V. ZAKLJUČAK

Potreba za brzim množenjem je danas sve izraženija u raznim strukturama u digitalnim telekomunikacijama i drugim sistemima u kojima se vrši obrada signala. Zbog toga je i korišćena *pipeline* struktura množača, kao optimalna za postizanje najvećih brzina. Pored toga, dodatno ubrzanje množenja je postignuto daljom optimizacijom množača  $4b \cdot 24b$ , korišćenjem lančanog CLA sabirača umesto klasičnog *Carry-Ripple*. Dizajn je projektovan direktno u VHDL-u, i to tako da bude pogodan za korišćenje kao komponenta nekog složenijeg ASIC čipa. U zavisnosti od potreba, dalja istraživanja bi mogla da idu u pravcu optimizacije predložene realizacije

za opisani Virtex4 ili neki drugi FPGA čip, čime bi se dobile optimalne vrednosti brzine i iskorišćenih resursa.

#### LITERATURA

- [1] <http://ieeexplore.ieee.org/ISOL/allstandards.jsp>
- [2] Dr Miodrag V. Popovic, Digitalna obrada signala, Akademski misao, Beograd, 2003
- [3] <http://www.dcd.pl/acore.php?idcore=24>
- [4] <http://http.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF>
- [5] <http://babbage.cs.qc.edu/courses/cs341/IEEE-754references.html>
- [6] Volnei A. Pedroni, Circuit Design with VHDL, MIT Press 2004
- [7] <http://www.xilinx.com/bvdocs/userguides/ug083.pdf>
- [8] <http://www.xilinx.com/bvdocs/userguides/ug080.pdf>
- [9] <http://direct.xilinx.com/bvdocs/userguides/ug070.pdf>
- [10] [http://download.xilinx.com/direct/ise8\\_tutorials/ise8tut.pdf](http://download.xilinx.com/direct/ise8_tutorials/ise8tut.pdf)

#### ABSTRACT

This Paper describes the hardware design of multiplier of real numbers in format according to standard IEEE-754. The multiplier is designed as a pipelined structure of level 7. This implies its use in applications in which it is necessary to perform a sequence of multiplications, since the effect duration of one multiplication is one clock interval. Whole design is described in VHDL, and then implemented on Xilinx Virtex4 FPGA chip, with basic statistics and results of implementation also provided.

#### DESIGN AND IMPLEMENTATION OF REAL NUMBERS MULTIPLIER ACCORDING TO IEEE-754 STANDARD

Miloš N. Jorgovanović, Obrad M. Đorđević