

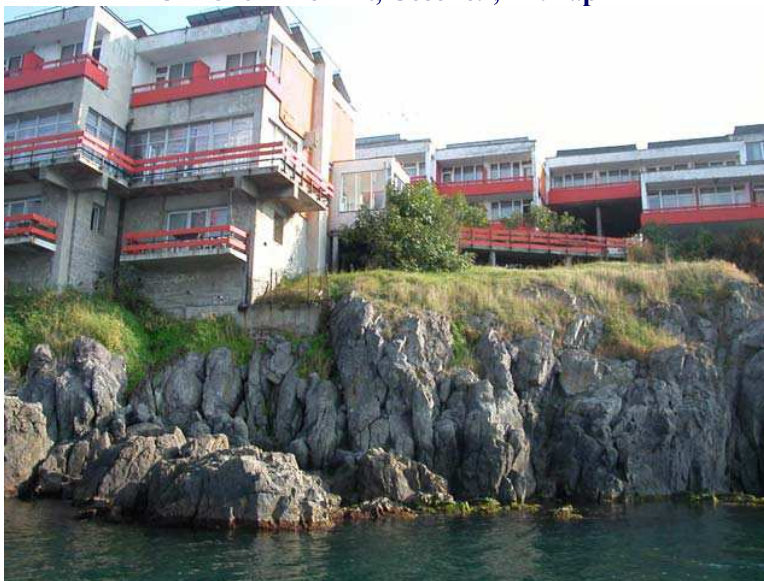


ISSN 1311-0829

ГОДИШНИК НА ТЕХНИЧЕСКИ УНИВЕРСИТЕТ-СОФИЯ

Том 64, книга 1, 2014

МЕЖДУНАРОДНА КОНФЕРЕНЦИЯ АВТОМАТИКА'2014, ФА
юбилей "40 ГОДИНИ ФАКУЛТЕТ АВТОМАТИКА"
13 - 15 юни 2014 г., Созопол, България



PROCEEDINGS OF TECHNICAL UNIVERSITY OF SOFIA

Volume 64, Issue 1, 2014

INTERNATIONAL CONFERENCE AUTOMATICS'2014, FA
Anniversary "40 YEARS FACULTY OF AUTOMATICS "
June 13 - 15, 2014, Sozopol, Bulgaria

СЪДЪРЖАНИЕ том 64, книга 1

АВТОМАТИКА

1. Иван Уливеров, Евтим Йончев, Тодор Йонков	15
<i>Управление на трифазни импулсни AC/DC преобразуватели с двустранен обмен на енергия със захранващата мрежа - част I</i>	
2. Евтим Йончев, Иван Уливеров, Тодор Йонков	25
<i>Управление на трифазни импулсни AC/DC преобразуватели с двустранен обмен на енергия със захранващата мрежа - част II</i>	
3. Марин Жилевски, Михо Михов.	33
<i>Методика за избор на подавателни задвижвания за фрезови машини</i>	
4. Васил Балавесов	43
<i>Адаптивно управление за мехатронни системи с директен синтез на управлението</i>	
5. Атанас Димитров, Владимир Заманов.	51
<i>Телеуправление на изследователски мобилен робот</i>	
6. Дочо Цанков.	61
<i>Оценка на енергийната икономичност при управление на локални конвектори</i>	
7. Тодор Йонков, Христо Стоянов	71
<i>Управление на система с превключваща структура за поддържане на надналягане в чисти помещения в системите за сградна автоматизация</i>	
8. Христо Стоянов, Теодора Петрова	77
<i>Приложение на програмируеми логически контролери за управление на оптични слънцезащитни щори в системи за сградна автоматизация</i>	
9. Георги Иванов, Иван Костов	83
<i>Изследване на MRAS схема за оценка на скоростта на асинхронен двигател, ориентирана по намагнитващия ток</i>	
10. Ганчо Божилов.	93
<i>Изследване на пусковия процес на асинхронен двигател в еднофазен режим</i>	
11. Емил Николов.	101
<i>Пространствен преход на проекцията при управление на параболични обекти с разпределени параметри</i>	
12. Василка Стоилова, Емил Николов.	111
<i>Моделиране на транспортния трафик в класа на обектите с разпределени параметри</i>	
13. Снежана Йорданова	121
<i>Електронни форми за учене и развитие на компетенции по интелигентни системи за управление</i>	
14. Снежана Йорданова, Владимир Янков.	131
<i>Размит супервайзор за нелинейна самонастройка на регулатори при работа в реално време</i>	
15. Нина Николова	141
<i>Двустепенно репетитивно управление, част I филтри</i>	

16. Нина Николова	149
<i>Двустепенно репетитивно управление, част II системи</i>	
17. Борис Киров, Васил Гълъбов	157
<i>Нискотарифен набор от оборудване за производството на микрофлуидни чипове с приложение в биоинженерните научни изследвания и изработката на биоинженерни устройства</i>	
18. Станислав Енев	163
<i>нелинейно управление на трифазни асинхронни двигатели с компенсиране на кръстосаните връзки между електромагнитния момент и роторното потокосцепление</i>	
19. Александър Ефремов.	171
<i>Оптимизация на пазарни стратегии и числена реализация на симплекс метода</i>	
20. Методи Георгиев	179
<i>Приложение на системи с размита логика при управление на балансиращи товари в енергийните системи на индустриални обекти</i>	
21. Весела Карлова-Сергиева, Соня Николова	187
<i>Методически аспекти при подготовката на материали за електронно обучение</i>	
22. Александър Маринчев, Весела Карлова-Сергиева	195
<i>Експериментално изследване на лабораторен стенд "регулиране на ниво"</i>	
23. Борислав Георгиев.	201
<i>Предизвикателства пред висшето техническо образование в контекста на развитието на образователни технологии и глобализирания пазар на компетенции</i>	
24. Десислава Стоицева-Деличева	211
<i>Метод за повишаване на енергийната ефективност при конвективно сушене чрез реализиране на точност на сушене без директно измерване на влагосъдържанието на материала</i>	
25. Александър Маринчев	219
<i>Математичен модел на затворен контур за управление, в който участва човека като функционално звено</i>	
26. Чавдар Дойчинов.	227
<i>Оптимизиран подход за обработка на информацията при управление на стандартна балансираща група</i>	
27. Николинка Христова.	235
<i>Интегриране на оперативния мениджмънт и техническата поддръжка на съоръженията в производството на пелети</i>	
28. Елица Петрова, Пламена Златева	245
<i>Анализ на динамиката на клетъчния растеж на <i>Acidithioacillus Ferrooxidans</i> при адаптация към високи концентрации на субстрат</i>	
29. Николай Гуров, Веселка Иванчева, Пламен Цветков, Георги Милушев, Георги Дюкенджиев, Христо Радев.	253
<i>Обучение по „Метрология и Измервателна Техника“ в съответствие с изискванията на бизнеса</i>	

30. Веселка Иванчева, Николай Гуров, Силвия Качулкова, Божидар Джуджев	261
<i>Електронно обучение по дисциплината „Измерване на Неелектрични Величини”</i>	
31. Радослав Делийски	269
<i>Алгоритъм за класификация на обекти при визуален контрол и инспекция на качеството</i>	
32. Владислав Славов	279
<i>Виртуално лабораторно упражнение за измерване на напрежение с аналогов волтметър</i>	
33. Емил Гарипов, Величка Руйкова, Тодор Пелтеков	287
<i>Обобщено предсказващо управление при различна степен на неопределеност в модела на обекта</i>	
34. Емануил Марков, Иван Евгениев	297
<i>Симулация на енергийната консумация за целите на управление и икономия на енергия в обществена сграда</i>	
35. Александър Ищев	303
<i>Моделно базирана диагностика на неизправности. Приложение при следяща система - част 1</i>	
36. Александър Ищев	313
<i>Моделно базирана диагностика на неизправности. Приложение при следяща система - част 2</i>	
37. Камен Перев	323
<i>Апроксимация с полиноми на Чебишев за редуциране на реда на модела</i>	
38. Йордан Кралев, Филип Филипов, Красимира Филипова	333
<i>Аритметично устройство за числа с плаваща запетая за FPGA прибори</i>	
39. Цоньо Славов, Йордан Кралев, Петко Петков	343
<i>Идентификация на двуколесен робот като обект с неопределеност</i>	
40. Цоньо Славов, Теофана Пулева	353
<i>Хардуерна симулация на система за адаптивно управление с програмирано усилване на скоростта на хидроагрегат</i>	
41. Теофана Пулева, Георги Ружеков	363
<i>Компенсация на смущения при управление на активната мощност в енергийна система с ограничена мощност</i>	
42. Андрей Йончев	373
<i>Моделно прогнозиращо управление на вятърно енергийна система</i>	
43. Андрей Йончев	383
<i>Проектиране на робастно и управляващо устройство за модел на система хидрогенератор</i>	
44. Аспарух Марковски	393
<i>Робастно управление на лабораторен модел - сервосистема</i>	
45. Емануил Марков	399
<i>Подход на програмната генерация при създаване на полу-натурни симулатори</i>	

46. Здравко Каракехайов	407
<i>ZETA алгебра: оптимизиране на вградените компютри за работа в реално време и ниска консумация</i>	
47. Веселин Георгиев	415
<i>Относно един подход за повишаване на надеждността на софтуера за системи за управление</i>	
48. Галина Чернева	425
<i>Оптимизация на формата на сигнала за отстраняване на преходния процес при предаването му</i>	
49. Атанас Атанасов, Ваня Кузманова	433
<i>Ролята на инженеринга на потребителските изискванията за създаването на успешен софтуерен продукт</i>	
50. Стоян Кирилов, Георги Ценов, Валери Младенов	443
<i>Синтез и анализ на мемристорен честотен умножител</i>	
51. Анна Апостолова-Йорданова, Снежана Терзиева	451
<i>Проблеми и задачи на прехода от средната към висшата техническа образователна степен (въз основа на опита при обучението по “Електротехника“ на ученици в ТУ-ЕС към ТУ и студенти във ФЕТТ на ТУ)</i>	
52. Симона Петракиева	459
<i>Приложение на дистанционната форма на обучение в курсовете по “Теоретична Електротехника” при анализа на линейни електрически вериги</i>	
53. Атанас Червенков, Тодорка Червенкова, Ивелина Христова	467
<i>Изследване на капацитивното влияние на въздушна линия за високо напрежение върху електротехническите съоръжения</i>	
54. Георги Ценов	477
<i>Обучение в работната среда на Matlab с модул за графичен потребителски интерфейс</i>	
55. Радослава Авджиева	483
<i>Детектор на лъжата с използване на ЕЕГ сигнали от човешкия мозък и класификация с невронни мрежи</i>	

АРИТМЕТИЧНО УСТРОЙСТВО ЗА ЧИСЛА С ПЛАВАЩА ЗАПЕТАЯ ЗА FPGA ПРИБОРИ

Йордан Кралев, Филип Ф. Филипов, Красимира Филипова

Резюме: Методите на автоматичното управление изискват изчисления с реални числа. В цифровата техника, аритметиката с плаваща запетая е най-доброто решение за целта. В доклада са представени Simulink модели, съобразени със стандарта IEEE 754, за операциите събиране, изваждане и умножение на числа с плаваща запетая с единична точност. От моделите е генериран VHDL код, който е вграден в Spartan 3E FPGA прибор. Проведени са симулационни и експериментални изследвания за валидация на разработените модули. Направено е сравнение относно заеманото място на чипа с други предлагани решения.

Ключови думи: аритметика с плаваща запетая, Simulink HDL Coder, вградени системи, IEEE 754, VHDL 2008.

FLOATING-POINT OPERATIONS FOR FPGA

Jordan Krlev, Filip F. Filipov, Krasimira Filipova

Abstract: Automatic control requires real numbers in its calculations. In digital electronics, the best approximation to the ring of real numbers is floating point arithmetic. In the paper we present Simulink models for summation, subtraction and multiplication operations for single floating-point numbers, according to IEEE 754. From them we generate a VHDL code, which is used to build simple embedded system for Spartan 3E FPGA. We validate our floating-point units by simulation and experimental procedures. A comparison is made for the required chip area, with other solutions.

Keywords: floating-point arithmetic, Simulink HDL Coder, embedded systems, IEEE 754, VHDL 2008

1. Въведение

Множеството на реалните числа R съдържа безброй елементи и не може да бъде кодирано с крайни множества от електрически състояния, върху каквито е базирана съвременната цифрова техника. Възможно решение е избор на подходящо крайно подмножество $A \subset R, |A| = 2^N$. Тогава, за да се запази алгебричната структура на пръстена на реалните числа, т.е. действията събиране и умножение, се налага въвеждане на операция закръгление, което обаче изменя смисъла

тези действия. Затова избора на елементи на A се оказва сложен. Към момента може да се счита, че за практически задачи, най-добрата извадка от множеството на реалните числа са числата във формат с плаваща запетая (ЧФПЗ), дефинирани от стандарта на IEEE 754 [1]. За алгебричните операции се използват алгоритми, съставени от реализуеми логически инструкции.

За разлика от микропроцесорите, FPGA приборите дават свобода на проектанта да програмира хардуерното ниво на системата, а не само софтуерното [2]. Процеса на разработка е ефективен, когато проектанта разполага с голям набор от спецификации на хардуерни компоненти. Аритметичните устройства за операции с ЧФПЗ са ключов компонент при всяка компютърна архитектура, ориентирана към числени алгоритми за обработка на информация. Съществуват някои готови решения за такива устройства, които най-общо се разделят на такива с отворен код [3], и комерсиални. От решенията с отворен код, трудно може да се попадне на висококачествени такива, с оглед на изискванията за надеждност и консумирано пространство от FPGA прибора.

За провеждане на изчисления с ЧФПЗ, във FPGA, е необходимо вграждане на съответна система в прибора. Архитектурите на FPGA включват логически блокове за общо предназначение (CLB) и специализирани логически блокове (SLB). Проектантът въвежда конфигурационен файл в прибора, и така дефинира различни системи от базовите блокове. Съвременният подход за проектиране на вградени системи за FPGA, предполага спецификация на език от високо ниво (например Simulink, VHDL, MATLAB, ...). Следва автоматично генериране на последователност от нарастващи по детайлност описания, завършваща с окончателния конфигурационен файл. В доклада са разгледани Simulink модели за операциите събиране и умножение, както и тяхната симулационна и експериментална валидация.

2. Представяне на числата с плаваща запетая

Най-общо, представянето на числото X , като ЧФПЗ е във вида $\pm d_0.d_1d_2 \dots d_{p-1} \times \beta^e$, където $d_0.d_1d_2 \dots d_{p-1}$ се нарича мантиса, а e – порядък, т.е.

$$X = \pm d_0.d_1d_2 \dots d_{p-1} \times \beta^e = (\pm \sum_{k=0}^{p-1} d_k \beta^{-k}) \beta^e, \quad 0 < d_i < \beta$$

Ако цялата част на мантисата е съставена само от една цифра, напр. $d_0 = 1$, тогава числото е *нормализирано* [4]. От изискването за нормализация следва единствеността на представянето. За измерване на грешката от апроксимация на реално число Z с ЧФПЗ X съществуват следните начини [5]:

- $e_{ulps} = \left| d_0.d_1d_2 \dots d_{p-1} - \frac{Z}{\beta^e} \right| \beta^{p-1}$ – брой грешни значещи цифри;
- $e_{rel} = \frac{|Z-X|}{Z}$ – относителна грешка.

Освен това, съществува свързващата зависимост: $0.5\beta^{-p} \leq 0.5e_{rel} \leq 0.5\beta^{-p}$.

При стандарта IEEE 754 основата е $\beta = 2$, и съществуват 4 различни нива на точност, чиито параметри са дадени в табл.1.

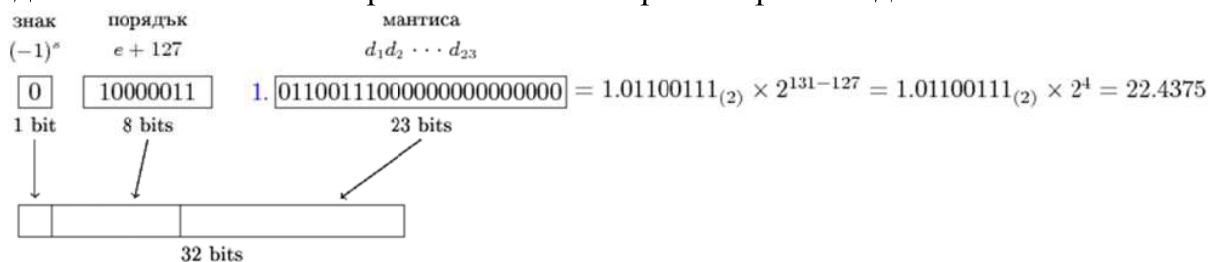
Табл.1. Нива на точност според IEEE 754 [1]

Параметър [битове]	Ниво на точност			
	Единична	Единична разширена	Двойна	Двойна разширена
p	24	≥ 32	53	≥ 64
e	8	≥ 11	11	≥ 15
X	32	≥ 43	64	≥ 79

В настоящата работа се използва формата с единична точност. При него числата се съхраняват в 32 бита, съобразно следната структура:

бит за знак _ 8 бита порядък _ 23 бита мантиса

Порядъкът може да бъде в интервала $-2^7 + 1 \leq e \leq 2^7$. Той се съхранява като 8 битово цяло число, което се получава с прибавяне на отместване 127 към порядъка e . Мантисата съдържа само дробната част на числото, защото при нормализираните числа се подразбира, че цялата част е 1. Например, едно ЧФПЗ с единична точност се съхранява в 32 битов регистър по следния начин:



Стандартът IEEE 754 дефинира и как трябва да се извършват аритметичните операции с ЧФПЗ. За събирането и умножението се изисква задължително получаване на точния резултат от операцията, след което се прилага закръгляване.

$$\begin{aligned}
 A + B &= a_0.a_1 \dots a_{p-1} \times 2^{e_a} + b_0.b_1 \dots b_{p-1} \times 2^{e_b} \\
 &= (a_0.a_1 \dots a_{p-1} + 0.0 \dots 0b_0.b_1 \dots b_{p-1}) \times 2^{e_a} \quad (1) \\
 &= c_0c_1.c_2 \dots c_m \times 2^{e_a} \\
 &\approx c_0.c_1 \dots c_{p-1} \times 2^{e_c} = C, \quad e_a - e_b = e_\Delta > 0
 \end{aligned}$$

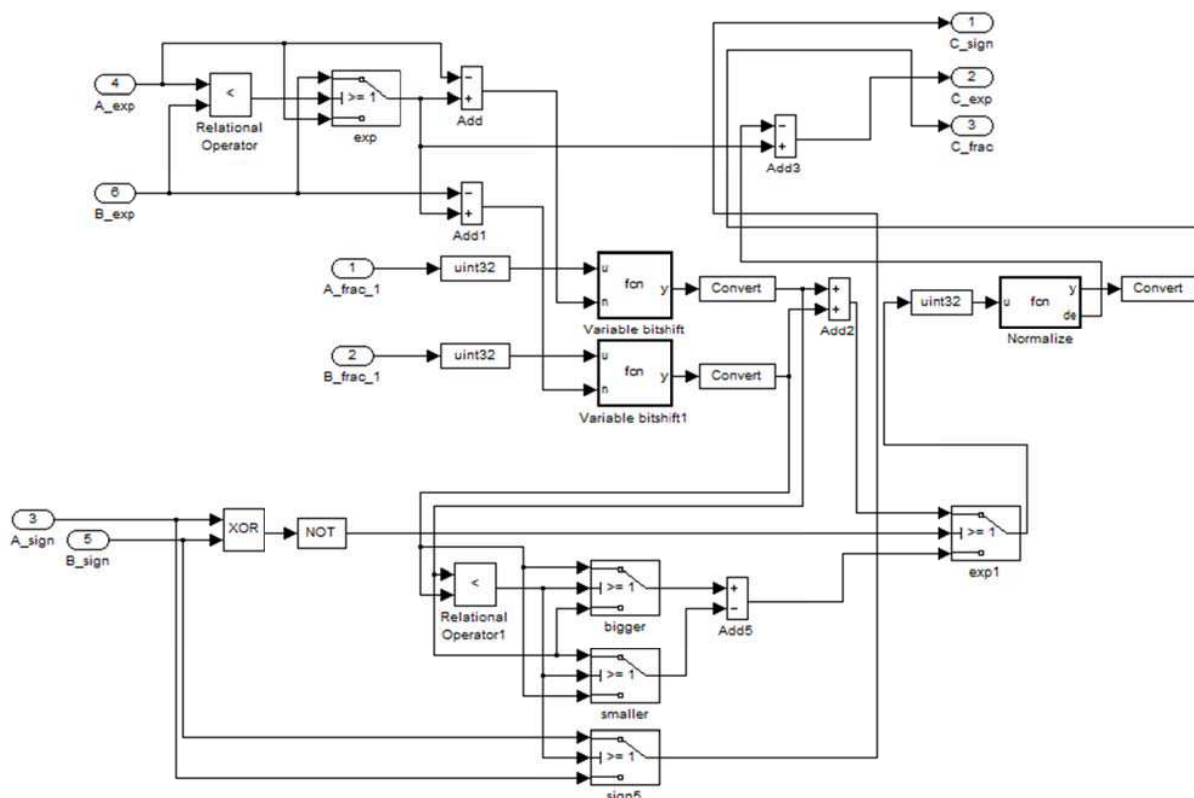
$$\begin{aligned}
 A * B &= (a_0.a_1 \dots a_{p-1} \times 2^{e_a}) * (b_0.b_1 \dots b_{p-1} \times 2^{e_b}) \quad (2) \\
 &= c_0c_1.c_2 \dots c_{2p} \times 2^{e_a+e_b} \\
 &\approx c_0.c_1 \dots c_{p-1} \times 2^{e_c} = C
 \end{aligned}$$

3. Simulink модели

Модел за събиране и изваждане

Simulink моделът за събиране на ЧФПЗ, с единична точност, е показан на фиг.1. Той е изграден от следните блокове: Relational operator, Switch, Data Type Conversion, Logical Operator, Sum, Embedded MATLAB function. Входовете на

модела са двата операнда A и B, декомпозирани по своите компоненти - мантиса, порядък, знак.



Фиг.1. Simulink модел за операциите събиране и изваждане на ЧФПЗ

Блокът Relational Operator сравнява порядъка на операнда A с този на B. По-големият порядък се записва в сигнала E_{ref} , като за получаването му в Simulink се използва блока exp (от тип Switch).

$$P_1: A_{exp} < B_{exp}$$

$$E_{ref} = \max(A_{exp}, B_{exp}) = \begin{cases} A_{exp}, & \text{not } P_1 \\ B_{exp}, & P_1 \end{cases} \quad (3)$$

Събирането изисква двата операнда да са с еднакъв порядък, което се постига с побитово преместване надясно (**shr**) на ефективните мантиси на двата операнда, съответно с N_A и N_B бита. Броят на битовете, с които трябва да бъде преместено всяко число се изчислява посредством сигнала E_{ref} , чрез 8 битовите суматори (Add и Add1).

$$\begin{aligned} \bar{A}_{frac,1} &= A_{frac,1} \text{ shr } N_A, & N_A &= E_{ref} - A_{exp} \\ \bar{B}_{frac,1} &= B_{frac,1} \text{ shr } N_B, & N_B &= E_{ref} - B_{exp} \end{aligned} \quad (4)$$

Ефективните мантиси $A_{frac,1}$ и $B_{frac,1}$ се получават чрез конкатенация на единица към мантисата на операндите, т.е.

$$A_{frac,1} = 1 * A_{frac}, \quad B_{frac,1} = 1 * B_{frac}$$

Побитовото преместване надясно се извършва чрез командата $y = \text{bitshift}(u, -n)$, която е записана в блока Variable bitshift (от тип Embedded Function). Блокът Relational Operator 1 сравнява мантисата на операнда A с тази на B. За опреде-

лянето на по-голямата и по-малката мантиса се използват, съответно, блоковете bigger и smaller (от тип Switch).

$$P_2: A_{frac,1} < B_{frac,1}$$

$$L_B = \max(\bar{A}_{frac,1}, \bar{B}_{frac,1}) = \begin{cases} \bar{B}_{frac,1}, P_2 \\ \bar{A}_{frac,1}, \text{not } P_2 \end{cases} \quad (5)$$

$$L_S = \min(\bar{A}_{frac,1}, \bar{B}_{frac,1}) = \begin{cases} \bar{A}_{frac,1}, P_2 \\ \bar{B}_{frac,1}, \text{not } P_2 \end{cases}$$

Блокът Add2 извършва сумирането на мантисите, а Add5 изваждането им. Това са 25 битови суматори за 24 битови числа, т.е. включват и един защитен бит.

$$R_{sub} = L_B - L_S, \quad d(R_{sub}) = 25 \text{ [бита]} \quad (6)$$

$$R_{sum} = \bar{A}_{frac,1} + \bar{B}_{frac,1}, \quad d(R_{sum}) = 25 \text{ [бита]}$$

Вида на операцията се определя от сигнал T, който управлява ключа exp1.

$$T = \text{not} (A_{sign} \text{ xor } B_{sign})$$

$$C_{frac,un} = \begin{cases} R_{sum}, T \\ R_{sub}, \text{not } T \end{cases} \quad (7)$$

Резултатът от операцията е ненормализирано число $C_{frac,un}$. Алгоритмът за нормализацията му е въведен в блока Normalize (от тип Embedded Function).

<pre>function [y,de] = fcn(u) %#eml buf=u; p=int8(-1); c=bitand(buf,2^24); for ind=1:25 if (~c) p=p+1; buf=bitshift(buf,1); c=bitand(buf,2^24); end end bufy = bitand(bitshift(u,p),2^23-1); y=bufy; de=p;</pre>	<ol style="list-style-type: none"> 1. $p = FF_{(16)} = -1_{(10)}$, $buf = C_{frac,un}$, $c = buf \text{ bitand } 1000000_{(16)}$, $ind = 1$ 2. Ако $c = 0$, тогава $p = p + 1$, $buf = buf \text{ shl } 1$, $c = buf \text{ bitand } 1000000_{(16)}$ 3. $ind = ind + 1$, Ако $ind < 26$, тогава <i>стъпка 2</i> 4. $C_{frac} = (C_{frac,un} \text{ shr } p) \text{ bitand } 7FFFFFF_{(16)}$, $\delta e = p$
--	--

Тази функция ($f_{normalize}$) прави корекция на мантисата и изчислява отместване (δe), с което трябва да се коригира порядъка

$$\begin{pmatrix} C_{frac} \\ \delta e \end{pmatrix} = f_{normalize}(C_{frac,un})$$

$$C_{exp} = E_{ref} - \delta e, \quad C_{sign} = \begin{cases} A_{sign}, P_2 \\ B_{sign}, \text{not } P_2 \end{cases} \quad (8)$$

Модел за умножение

Simulink моделът за умножение на ЧФПЗ, с единична точност, е показан на фиг.2. Блоковете Gain и Gain1, преобразуват ефективната мантиса на операндите към формат с фиксирана запетая, при който има 1 бит за цялата част и 23 бита - за дробната.

Този формат се записва като $FIX(s, L, F) = FIX(0, 24, 23)$, или съответно,

$$X_{FIX(s,L,F)} = X 2^{-F},$$

където X е цяло число, представено в допълнителен код.

$$A_{mul} = \left(\frac{A_{frac,1}}{2^{23}} \right)_{FIX(0,24,23)} \quad B_{mul} = \left(\frac{B_{frac,1}}{2^{23}} \right)_{FIX(0,24,23)} \quad (9)$$

Блокът Product е умножител на числа с фиксирана запетая. Резултатът е вход на блока за нормализация. След нормализацията се прави закръгление, при което мантисата се преобразува от 48 до 23 бита.

$$\left(\begin{matrix} C_{frac} \\ \delta e \end{matrix} \right) = f_{normalize}(C_{mul}) \quad (10)$$

Порядъкът на произведението е сума от порядъците на операндите, при сумирането трябва да се отчете наличието на отместване 127 в представянето на порядъка на ЧФПЗ. Суматорите са 9 бита.

$$C_{exp,sum} = A_{exp} + B_{exp} - 127 \quad C_{exp} = C_{exp,sum} - \delta e \quad (11)$$

Знакът на резултата се определя чрез операцията изключващо или.

$$C_{sign} = A_{sign} \mathbf{xor} B_{sign} \quad (12)$$

Алгоритмът за нормализация е въведен в блока Normalize_mul (от тип Embedded MATLAB Function) и е аналогичен на този, използван при събирането.

4. Верификация и валидация

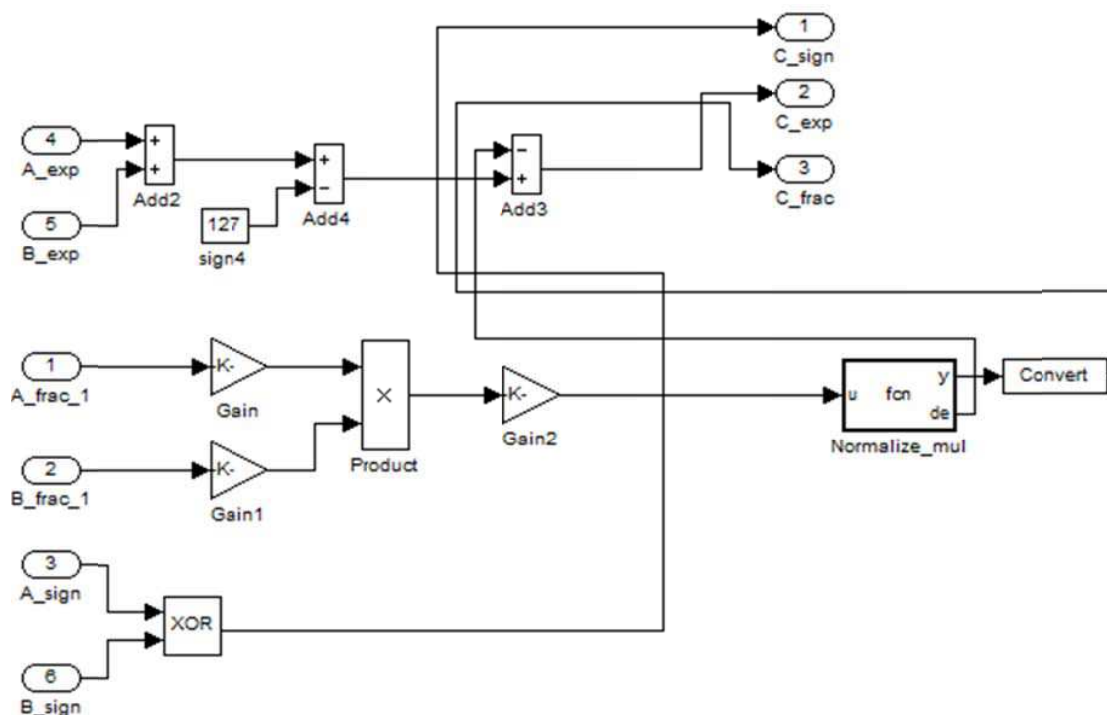
За верификацията на Simulink моделите се използва направеното им формално описание в т. 3. Съответствието на моделите, с изискванията дадени в стандарта IEEE 754, е обобщено в табл.2.

Проведени са симулационни и експериментални изследвания за валидация на моделите. За осъществяване на симулация в Simulink са формирани допълнителни блокове, които преобразуват въведено ЧФПЗ до неговите компоненти и обратно.

На фиг.3 е показан резултата от симулация на операцията умножение.

За експериментална валидация е използван FPGA прибор Spartan 3E и развойна платка Spartan 3E Starter-Kit [6]. Чрез инструмента Simulink HDL Coder [9] е генерирано поведенческо описание на езика VHDL, за двата аритметични модула.

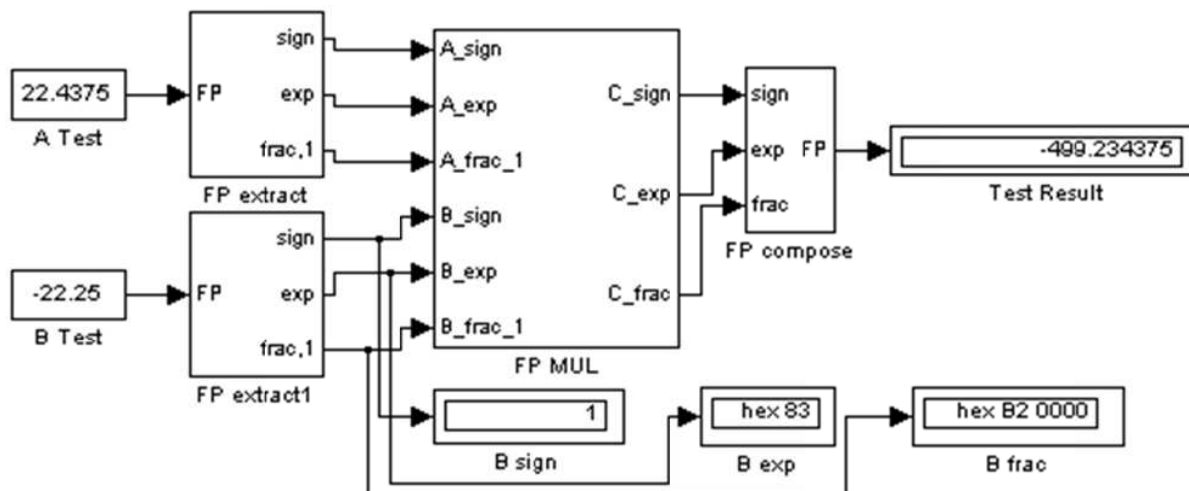
След това, те са интегрирани в развойната среда Xilinx ISE, чрез два модула: Ethernet/IP/UDP комуникационен и управляващ (фиг.4). Всяко устройство, в локалната Ethernet мрежа, да зададе операндите и вида на операцията. Вградената система извършва аритметичната операция и връща резултата на заявителя ѝ.



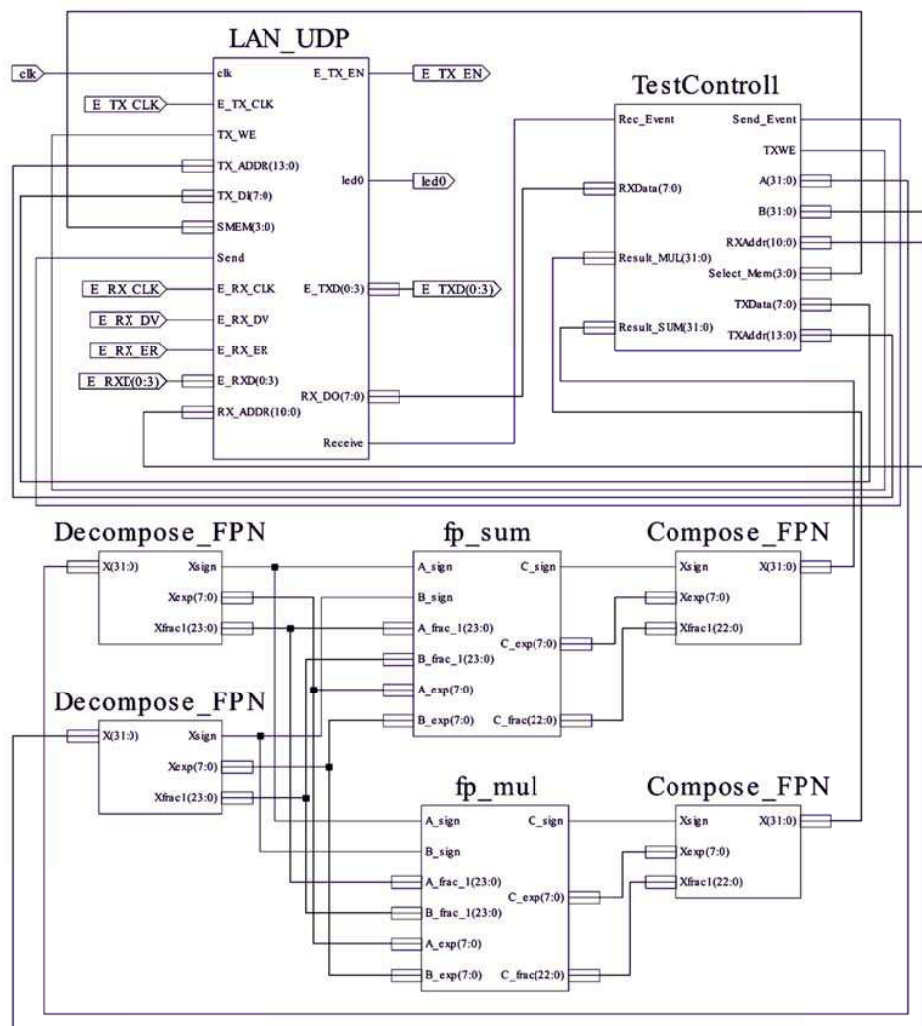
Фиг.2. Simulink модел за операцията умножение на ЧФПЗ

Табл.2. Изпълнение на изискванията на IEEE 754

<i>Изискване по стандарт</i>	<i>Изпълнение в Simulink</i>
бит за знак _ 8 бита порядък _ 23 бита мантиса	фиг. 1, фиг. 2
Получаване на точен резултат и закръгляване - (1),(2)	(6), (10)
Нормализация на числата	(8)
Изключения (Inf, NaN,...)	не се поддържа
Защита от препълване	Настройка на параметрите на блокове тип Add, Product, Convert



Фиг.3. Симулация за валидация на операцията умножение



Фиг.4. Архитектура на вградена система в FPGA за операции с плаваща запетая

Отчет за използваните ресурси от FPGA прибора, за реализация на аритметичните операции, е даден в фиг.6 и фиг.8. Направено е сравнение с библиотеката за плаваща запетая в VHDL 2008[10] – нов стандарт за езика, който, за съжаление, и до днес не се поддържа от средите на всички производители на FPGA. На фиг.5 е показана статистика след синтезиране и имплементиране на написан VHDL код за събиране на числа, представени с плаваща запетая. На фиг. 6 е показана същата статистика, с използване на същите настройки, при имплементиране и синтезиране, но чрез MATLAB блока за събиране.

Видно е, че при използване на VHDL библиотеката са необходими повече ресурси, но това е обяснимо с факта, че там се правят проверки за Inf, NaN и др. , които за момента не са реализирани в MATLAB блока. При операцията умножение (фиг.7 и фиг.8) заеманите ресурси от MATLAB блока са два пъти повече, което показва, че при VHDL библиотеката е използван по-добре оптимизиран алгоритъм. За да се избягне автоматичното използване на повече битове от MATLAB блока, при преминаване от един тип данни в друг, планираме бъдещи промени в него.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	1,297	9,312	13%
Number of occupied Slices	658	4,656	14%
Number of Slices containing only related logic	658	658	100%
Number of Slices containing unrelated logic	0	658	0%
Total Number of 4 input LUTs	1,299	9,312	13%
Number used as logic	1,297		
Number used as a route-thru	2		
Number of bonded IOBs	98	232	42%
IOB Flip Flops	32		
Number of BUFMUXs	1	24	4%
Average Fanout of Non-Clock Nets	3.84		

Фиг.5. Използвани ресурси при използване на VHDL библиотеката за операцията събиране от FPGA прибора, след синтез и имплементиране

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	995	9,312	10%
Number of occupied Slices	502	4,656	10%
Number of Slices containing only related logic	502	502	100%
Number of Slices containing unrelated logic	0	502	0%
Total Number of 4 input LUTs	995	9,312	10%
Number of bonded IOBs	98	232	42%
Average Fanout of Non-Clock Nets	4.22		

Фиг.6. Използвани ресурси от MATLAB блока за събиране за операцията събиране от FPGA прибора, след синтез и имплементиране

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	292	9,312	3%
Number of occupied Slices	161	4,656	3%
Number of Slices containing only related logic	161	161	100%
Number of Slices containing unrelated logic	0	161	0%
Total Number of 4 input LUTs	314	9,312	3%
Number used as logic	292		
Number used as a route-thru	22		
Number of bonded IOBs	98	232	42%
IOB Flip Flops	32		
Number of BUFMUXs	1	24	4%

Фиг.7. Използвани ресурси при използване на VHDL библиотеката за операцията умножение от FPGA прибора, след синтез и имплементиране

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	562	9,312	6%
Number of occupied Slices	301	4,656	6%
Number of Slices containing only related logic	301	301	100%
Number of Slices containing unrelated logic	0	301	0%
Total Number of 4 input LUTs	596	9,312	6%
Number used as logic	562		
Number used as a route-thru	34		
Number of bonded IOBs	98	232	42%
Number of MULT18X18SIOs	4	20	20%
Average Fanout of Non-Clock Nets	2.89		

Фиг.8. Използвани ресурси от блока за умножение за операцията умножение от FPGA прибора, след синтез и имплементиране

5. Заключение

В статията са разгледани Simulink модели за реализация на операциите събиране, изваждане и умножение на ЧФПЗ, с единична точност. Генериран е VHDL код и е вградена система за аритметика с плаваща запетая във FPGA прибор. Показани са резултати, доказващи работоспособността ѝ. Направени са сравнения, по отношение на използваните ресурси, с решения за аритметика с плаваща запетая във FPGA прибор.

По отношение на стандарта IEEE 754, не е изпълнено единствено изискването за обработка изключения, което ще бъде предмет на бъдеща работа. Трябва да се има в предвид, че това несъответствие не влошава сигурността при извършване на операциите, тъй като са взети мерки за защита от препълване.

БЛАГОДАРНОСТИ

Изследванията представени в доклада са изпълнени и финансирани по договор към ТУ-София, No 142ПД0008-08.

ЛИТЕРАТУРА

- [1] IEEE 1987, *IEEE Standard 754-1985 for Binary Floating-Point Arithmetic*, IEEE. Reprinted in SIGPLAN 22, 2, 9-25
- [2] Brown St., Vranesic Z., *Fundamentals of Digital Logic with VHDL Design*, 2008, ISBN-10: 0077221435 , ISBN-13: 978-0077221430.
- [3] OpenCores. *Arithmetic Core Projects*. www.opencores.org
- [4] Kahan W., *IEEE Standard 754 for Binary Floating-Point Arithmetic*, Lecture notes on the status of IEEE 754, University of California, 14.01.1996
- [5] Goldberg D., *What Every Computer Scientist Should Know About Floating-Point Arithmetic*, ACM Computing Surveys, Vol. 23, No. 1, 03.1991
- [6] Кралев Й., Филипов Ф.Ф., Филипова Ф., *Разработка на FPGA системи за управление чрез Simulink*, САИ, София 2013.
- [7] Rafiquzzaman M., *Fundamentals of Digital Logic and Microcomputer Design*, 2005, ISBN-13: 978-0471727842.
- [8] Roth, Charles H., *“Digital systems design using VHDL”*, CL-Engineering, 2007, ISBN 978-0534384623.
- [9] Mathworks Inc. *Simulink HDL Coder User’s Guide*. www.mathworks.com
- [10] EDA Industry Working Groups. *VHDL-2008 Support Library*. <http://www.eda.org/fphdl/>

Автори: Йордан Кралев, маг. инж. докторант, катедра „Системи и управление“, Факултет Автоматика, Технически Университет - София, E-mail address: jkralev@yahoo.com ; Филип Ф. Филипов, маг. инж. докторант, докторантско училище ФаГИОПМ, Технически Университет - София; Красимира Филипова, доц. д-р, катедра „Системи и управление“, Факултет Автоматика, Технически Университет - София, E-mail address: krfil@tu-sofia.bg

Постъпила на 29.04.2014

Рецензент проф. д-р П. Петков