

# МЕТОД ЗА РЕАЛИЗАЦИЯ НА АПАРАТНИ САМОУПРАВЛЯВАЩИ СЕ ЦИКЛИЧЕСКИ СТРУКТУРИ

## METHOD FOR REALIZATION OF SELF – CONTROLLING LOOP APPARATUS STRUCTURES

Димитър Тянев, Стамен Колев, Веселин Йосифов

**Резюме:** Изследва се нов подход за апаратна реализация на изчислителен процес, основната характеристика на който е отсъствието на микропрограмно управление в синтезираните операционни структури. В настоящата работа е изложен общ метод за апаратна реализация на самоуправляващи се циклически алгоритмични структури. Работоспособността на представените структури е доказана чрез моделиране и практическо експериментиране на реални изчислителни устройства върху елементна база на фирмата Xilinx.

**Ключови думи:** Апаратни (схемни) операционни структури, управляващи автомати.

**Abstract:** The new approach was explored for hardware implementation of calculation procedure, the cardinal characteristic to which the rule of the synthesized procedure structures in the absence of microprogram control in the synthesized operational structures. The general method for the hardware execution is exposed in the present work on autonomous loop algorithmic structures. The operating capability to the represented structures is showed by modeling and practical experiment in real calculation drives on the base of the company Xilinx.

**Keywords:** Controlling automata, computational devices.

### I. ВЪВЕДЕНИЕ

Проектирането на изчислителна апаратура в съвременни условия съществено се различава в сравнение с проектирането в недалечното минало. Това се дължи както на новите градивни елементи и техните възможности, така и на развитите програмни средства за автоматизация на проектирането [1]. Благодарение на тази еволюция на технологиите като цяло стават възможни за реализация нови подходи и методи за реализация на изчислителния процес, като разгледания в [2], чието развитие авторите предлагат по-долу. Най-същественото в предлагания подход и методи за реализация на изчислителния процес се явява възможността за апаратна реализация на безавтоматни самоуправляващи се операционни структури, съответстващи на основните алгоритмични структури, както и на формите на паралелизъм, съдържащи се в тях, елементи на които се съдържат в [4,5,6,7,8].

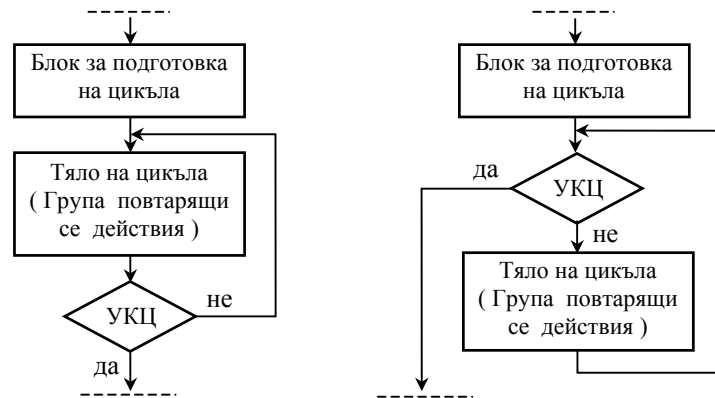
Настоящото изследване се отнася до нов метод за апаратна реализация на самоуправляващи се циклически алгоритмични структури.

Известни са два основни вида циклически алгоритмични структури:

1. С предварително известен брой повторения ;
2. С предварително неизвестен брой повторения.

За формалното организиране на структури от първия вид се използва фактът, че техният брой ( $S$ ) е предварително известен и той се контролира от отделна променлива, наричана брояч. Условието за край на цикъла (УКЦ) по същество е въпрос за отношението на съдържанието на този брояч спрямо отнапред известна константа  $S$ . Броячът може да бъде както инкрементен, така и декрементен. Най-често се избира той да бъде декрементен, тъй като проверката за достигане на неговото нулево съдържание, се реализира най-икономично, както с програмни, така и с апаратни средства. Зареждането на брояча с начална стойност се извършва в блока за подготовка на цикъла (фигура 1). Съществуват две разновидности на тази алгоритмична структура, според реда на нейните блокове. Структурата от лявата рисунка е известна като цикъл с после условие, а тази отдясно – като цикъл с пред условие. Първата структура се характеризира с това, че действията в тялото на цикъла се изпълняват винаги поне веднъж, преди да се провери условието за край на цикъла. Това предопределя началната стойност на брояча да бъде равна на стойност, която е с единица по-малка от общия брой повторения  $S$ . Втората структура се характеризира с това, че тя предполага възможността, действията от тялото на цикъла да не бъдат изпълнени нито веднъж, което означава, че началната стойност на

брояча следва да бъде равна на общия брой повторения  $S$  (и в двата случая имаме предвид декрементен брояч).



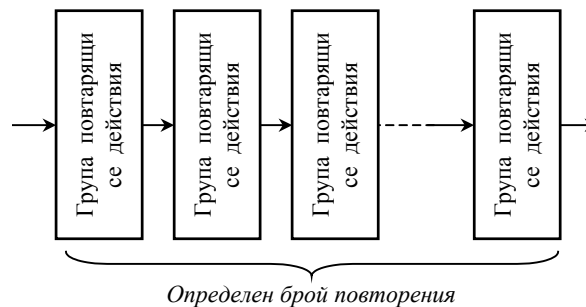
Фиг. 1 Циклически алгоритмични структури

## II. АНАЛИЗ

Известни са две разновидности за организация на регулярни повторения: буквална (фактическа) реализация на повторенията и циклическа организация на повторенията.

### *Буквална организация на регулярни повторения*

Буквалната организация на повторенията не използва условен алгоритмичен преход. Тя се постига чрез многократно ( $S$ кратно) фактическо повторение на групата действия в тялото. В резултат се получава линейната алгоритмична структура, илюстрирана на фигура 2. Апаратната реализация на такива структури, като безавтоматни и самоуправляващи се, е разгледана в [2].



Фиг. 2 Буквално повторение

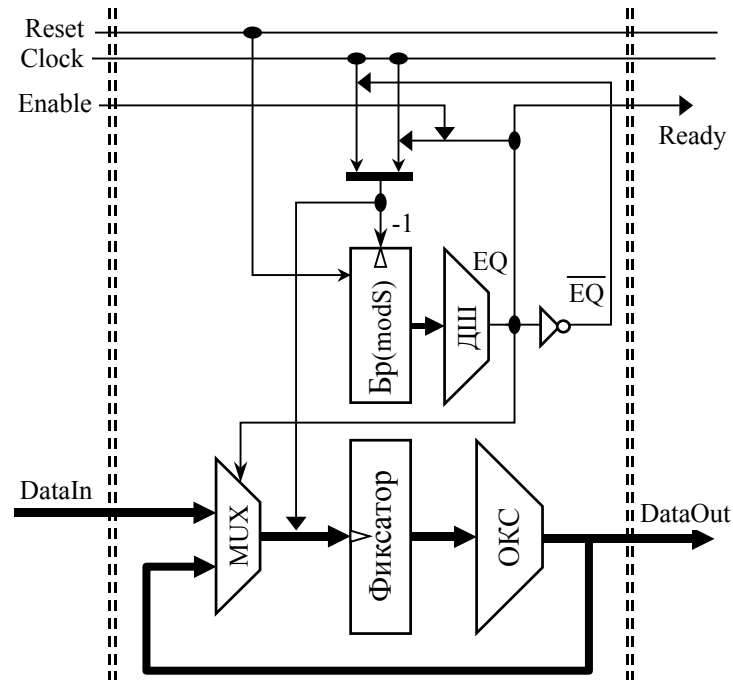
Показаната линейна апаратна реализация обаче е възможна само за цикли с конструктивно известен брой повторения. Същата не е възможна за цикли, чийто брой на повторения се определя (се изчислява) в предходни етапи на изчислителния процес, тъй като тази ситуация прави предварителната линейна апаратна реализация невъзможна. Тя е невъзможна по две причини: първо, защото броят на повторенията не е известен към момента на проектиране на реализацията и второ, по причина на възможното разнообразие за получаваната стойност. Ето защо реализацията на циклическата структура с предварително известен брой повторения, който не е конструктивно известен, а се изчислява в хода на предварителните етапи, представлява особен случай, който ще разгледаме тук отделно.

### *Циклическа организация на регулярни повторения*

Буквалната организация на регулярните повторения има съществения недостатък да води до големи апаратни разходи. Ето защо тук предлагаме реализация на алтернативната циклическа организация. На фигура 3 е показана синтезираната обща логическа структура, която е в състояние да реализира циклическата организация на цикли с конструктивно предопределен брой на повторенията  $S$ .

Апаратната реализацията на условието за край на цикъла изисква реализация на броячна променлива по модул  $S$ . Такава променлива се реализира удобно във вид на апаратен брояч  $Br_{(modS)}$ , чийто модул на броене е равен на зададената стойност  $S$ . Синтезът

на такъв брояч най-често се осъществява чрез методиката за синтез на крайни автомати. В представената реализация е прието началната стойност в брояча да бъде нула. Така, когато той отброи  $S$  на брой входни импулси, ще се окаже отново в изходно (нулево) състояние, което е удобно за следващи изпълнения. В структурата има конвейерен регистър-фиксатор  $\Phi$  за временно съхранение на данните, операционна комбинационна схема ОКС, реализираща за простота обща линейна алгоритмична структура и входен мултиплексор MUX, за реализация на входната точка в циклическата структура. Операционната част, обхваната от обратната даннова връзка, се превключва толкова пъти, колкото тактови импулса постъпят в брояча.



Фиг. 3 Циклическа операционна структура с брояч по модул  $S$

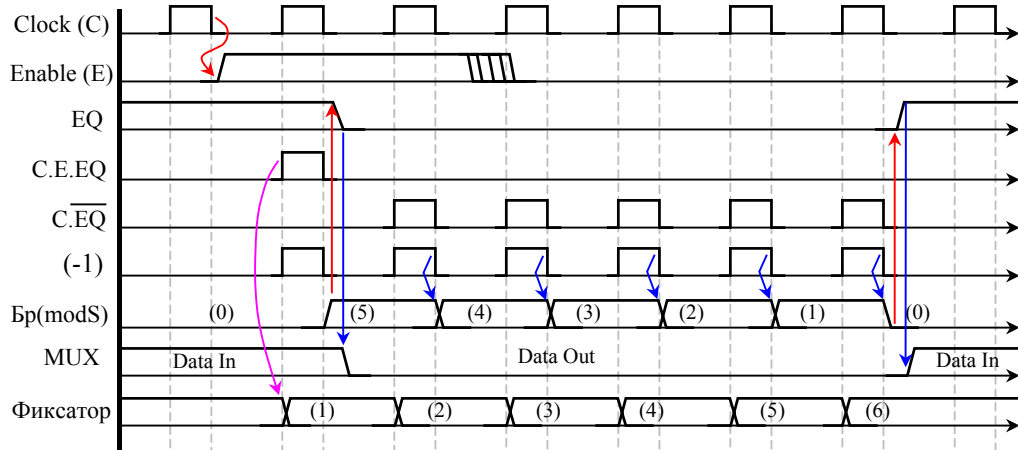
Дешифраторът ДШ разпознава нулевото съдържание в брояча и формира сигнал  $EQ$ . Чрез този сигнал и останалите входни за секцията сигнали –  $Reset$ ,  $Enable$  и  $Clock$ , се реализира управлението на структурата. Управлението се изразява в това, че постига отделяне в необходимия момент на  $S$  на брой последователни тактови импулси, изрязани от безкрайната последователност  $Clock$  на входа на брояча. Първият от тези импулси, който се създава в структурата в отговор на сигнал  $Enable$ , зарежда началните данни във фиксатора и превключва брояча в състояние  $(S-1)$ . В резултат на това изчезва сигналът  $EQ$  ( $EQ=0$ ), чиято нова стойност превключва мултиплексорът MUX към обратната връзка. Така се реализират  $S$  последователни превъртания на данните  $DataOut$  през операционната комбинационна схема ОКС. След възникване на условието за край на цикъла ( $EQ=1$ ), секцията декларира своята готовност  $Ready$  на следващите, блокира постъпването на нови тактови импулси  $Clock$  и отваря входната даннова шина  $DataIn$ , като превключва мултиплексора MUX от обратната връзка към входната шина.

Целият изчислителен процес се постига апаратно и без външно микропрограмно управление. Този процес е илюстриран с времедиagramата от фигура 4. Както е означено на фигура 3, прието е броячът да се превключва по заден фронт на импулсите  $(-1)$ , а фиксаторът – по преден фронт. Тези импулси са резултат от дизюнкцията:

$$(-1) = (Enable \cap Clock \cap EQ) \cup (Clock \cap \overline{EQ}). \quad (II.1)$$

Примерната времедиagramа по-долу на фигура 4 изразява цикъл с 6 повторения.

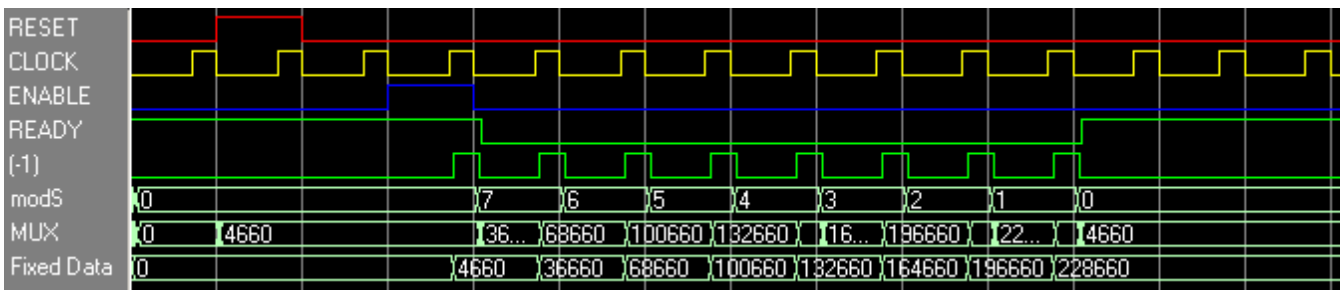
Структурата на циклическата организация функционира правилно само ако сигналът  $Enable$  от предходната секция се появява след задния фронт на тактовия сигнал, както е показано на времедиagramата. Следва да отбележим, че цикълът от вида с предусловие за тази структура е безсмислен.



Фиг. 4 Времедиаграма за изпълнение на цикъл

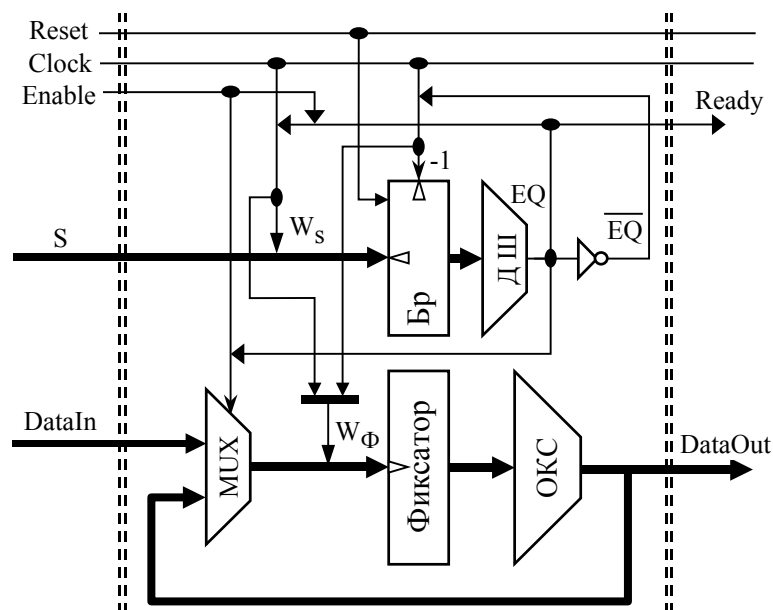
Изходният за секцията сигнал *Ready* се генерира по задния фронт на тактовия импулс. Това го прави използваем в качеството на сигнал *Enable* за следващата секция. При това условие той удовлетворява и изказаното по-горе условие.

Времедиаграмата на модела (по-долу) върху елементи *Spartan II* на фирма *Xilinx* напълно потвърждава казаното за структурата от фигура 3.



### Циклическа организация с променлив брой на повторения

Както беше изтъкнато в началото, циклическата организация с променлив брой повторения изисква използването на универсален брояч. По тази причина е синтезирана модификация на първоначалната структура, която е представена на фигура 5. Контролната стойност се записва в брояча Бр, който е декрементен. Условието за край на цикъла се формира от дешифратора ДШ, който разпознава съдържание нула и формира сигнала EQ.

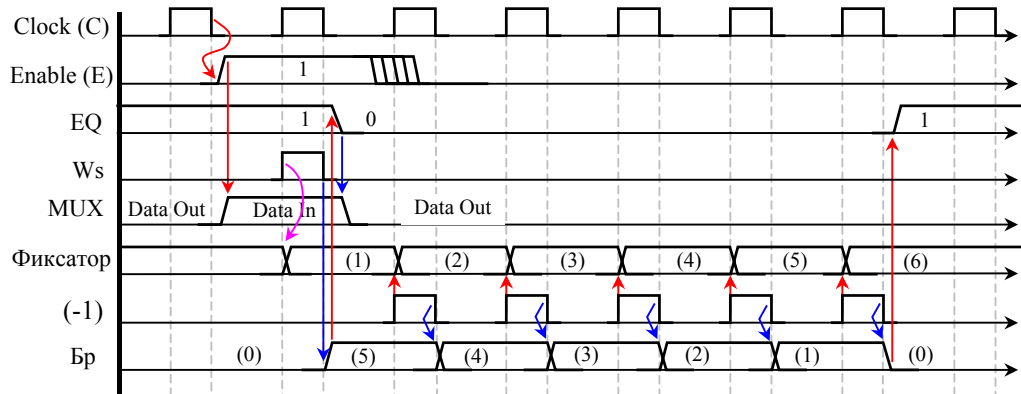


Фиг. 5 Циклическа операционна структура с променлив брой на повторенията

На фигура 6 е представена времедиаграмата за изпълнение на цикъл от 6 повторения. Както може да се види, условието за запис на нови данни и старт на изчисленията в структурата се дава от сигнала  $W_s$ , който се формира при следното условие:

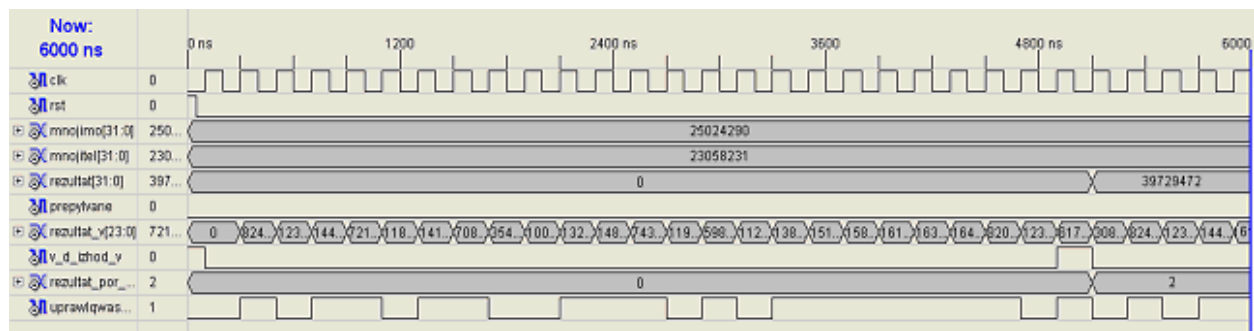
$$W_s = \text{Enable} \cap \text{Clock} \cap \text{EQ} . \quad (\text{II.2})$$

Използването на сигнала EQ в разглежданата структура е аналогично на изложеното за предидущата. За тази структура цикълът с предусловие е възможен. Възможността за неизпълнение на действията от тялото на цикъла налагат операционната схема ОКС да бъде преместена в обратната връзка (фиг. 5). Така записаните във фиксатора входни данни ще излязат без обработка на изход.

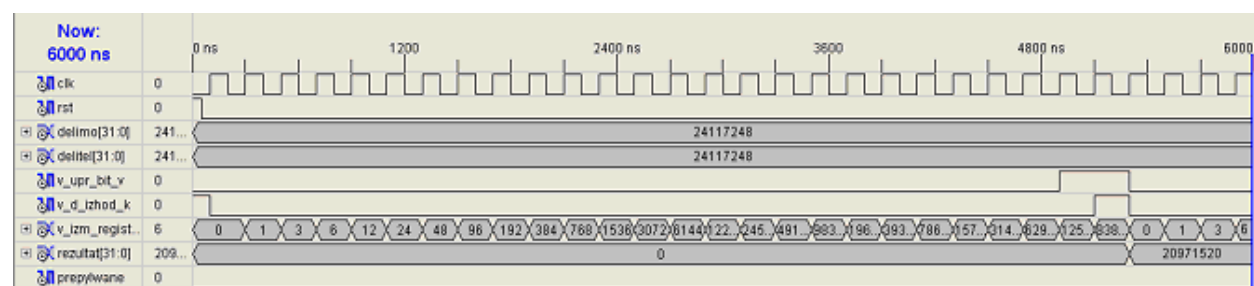


Фиг. 6 Времедиаграма за изпълнение на цикъл с 6 повторения

Представеният метод за реализация на самоуправляващи се циклически структури е успешно приложена при проектиране на реални изчислителни устройства. В [3] подробно е изложен проект за реализация на конвейерни устройства за умножение и деление на 32-битови числа  $(8+24)[b]$ , представени във форма с плаваща запетая и структура  $(P \pm M)$ . Подолу на фигури 7 и 8 са представени времедиаграми за функционирането на реално проектираните устройства. Върху тях, при умножение отпечатъците, които предоставя програмната среда, интерпретирайки съдържанието на регистрите като цяло число, съответно 250254290 и 23058231, представляват съмножителите  $X=1,98$  и  $Y=1,33$ . Произведението (отпечатък 39729472) е числото 2,6334. При деление операндите са еднакви  $(24117248 \div 1,75)$  частното е 1  $(20971520 \div 1,0)$ .



Фиг. 7 Времедиаграма на операция умножение



Фиг. 8 Времедиаграма на операция деление

### III. ИЗВОДИ

В заключение стигаме до следните изводи:

1. Изразеният чрез алгоритмична схема изчислителен процес (фиг. 1) може да бъде реализиран в изчислително устройство, чиято логическа структура не съответства на декомпозицията на Глушков [7], т.е. не съдържа управляваща част и при това се подчинява на общата ни представа за тази реализация [2, фиг. 3].

2. Така, изказаната в [2] обща теза, за възможността всички основни видове алгоритмични структури да се реализират като самоуправляващи се апаратни структури, се потвърждава и за разгледаните тук случаи.

3. Алгоритмичната структура цикъл все още не е напълно изследвана, тъй като за нея съществуват още няколко разновидности, като например цикли с предварително неизвестен брой повторения. Те обаче ще бъдат обект на следващи публикации.

4. В същност, операционна част на предлаганата тук структурата има вида на сложна комбинационна схема (еднотактна) или има вида на конвейер, ако в структурата бъдат използвани междинни фиксатори (фиг. 2).

5. Предложеният метод е успешно използван за проектиране и експериментиране на реални устройства за умножение и деление на 32-битови числа с плаваща запетая по стандарта с изместен порядък IEEE 754.

От методична гледна точка, представеният метод за синтез на операционното устройство, се базира на синтезирания за неговото функциониране алгоритъм. Този алгоритъм се декомпозира с цел да бъдат изявени описаните и изследвани първични алгоритмични структури, за които да бъдат приложени получените решения. Като цяло крайната структура на операционното устройство е своеобразна конкатенация на отделните решения.

Основното достоинство на предложения метод се състои във възможността за рязко намаляване на последователните тактове при изпълнение на алгоритъма, включително до еднотактно. Така, като цяло, апаратната самоуправляваща се реализация на изчислителния процес води до максимална производителност. Допълнително повишение на производителността може да бъде реализирано при конвейерна обработка на последователна серия от данни.

Като основен недостатък може да се посочи известното увеличаване обема на апаратните разходи в сравнение с чистите такива за операционната част на автоматно управляваното устройство. Това, разбира се, е “платената цена” за постигнатите положителни ефекти.

#### ЛИТЕРАТУРА:

- [1]. WEB-адрес на фирма XILINX – <http://www.xilinx.com/>
- [2]. Josifov V., Kolev St., Tyanev D., *Operational structures without controlling automata*, International workshop on network and grid infrastructures, Institute for parallel processing at the bulgarian academy of sciences department “distributed computing systems and networks”, Sofia, Bulgaria, 27-28 Sept. 2007, EC Project BIS21++ “Bulgarian IST centre of competence in 21 century“, <http://bis-21pp.acad.bg/events/events.htm>
- [3]. Kolev St., Josifov V., Master thesis: “*Design of asynchronous pipelines for floating point multiplication and division*”, FHTW – Berlin, Deutschland, 2007.
- [4]. Patterson D. A., Hennessy J. L., *Computer Organization And Design*, Morgan Kaufmann Publishers, ISBN 1-55860-604-1, 2005.
- [5]. Sutherland I. E., *Micropipelines*, <http://research.sun.com/vlsi/Publications/KPDisclosed/micropipelines/cmipipelines.pdf>
- [6]. John F. Wakerly, *Digital Design – principles and practices*, Fourth Edition, Prentice Hall, ISBN 0-13-186389-4, 2005.
- [7]. Булгаков С., Мещеряков В., Новоселов В., Шумилов Л., *Проектирование цифровых систем на комплектах микропрограммируемых БИС*, Москва, “Радио и связь”, 1984.
- [8]. Тянев Д. С., *Организация на компютъра – проектиране на логически структури*, ISBN 954-20-0259-9, ТУ-Варна, 2004.