СЪДЪРЖАНИЕ

CONTENTS

СЪДЪРЖАНИЕ

CONTENTS

Bulgaria Communications Chapter

ЧЕТИРИ-ФАЗОВ МИКРОКОНВЕЙЕР С ЕДНОТАКТОВИ И МНОГОТАКТОВИ МИКРОКОНВЕЙЕРНИ ЗВЕНА

Димитър С. Тянев

Резюме: Разглеждат се структурни проблеми на микроконвейер, свързани с произволното включване в състава му на еднотактови и многотактови микроконвейерни звена. С цел да бъде синтезирано общо и независимо от структурата конвейерно управление, е предложена обобщена интерпретация на конвейерната организация, поставяща в центъра микрооперация запис в конвейерните регистри. Постигането на тази организация обаче е ограничено от вида на входните регистри на многотактовите микроконвейерни звена, използващи тригери със структура Edge. Това обстоятелство предопределя използването на 4-фазовия конвейерен протокол за трансфер на данни. При тези условия е разработен асинхронният конвейерен автомат и цялостната структура на микроконвейер, който комбинира и двата вида микроконвейерни звена. Показано е функционирането на конвейерния протокол.

Ключови думи: Микроконвейери, 4-фазов асинхронен протокол с изпреварване, Синхронизация.

Four-Phase Micro-Pipeline with One-Cycle and Multi-Cycle Micro-Pipeline Sections

Dimitar S. Tyanev

Abstract: There are micro-pipeline's structural problems considered, related to the arbitrary inclusion of one-stage and multi-stage micro-pipeline sections. In order to be synthesized common and structure independent pipeline control there is generalized interpretation of pipeline organization presented, with accent on the micro-operation "store" in the pipeline's registers. The achievement of this organization is limited by the type of the used input registers in the multi-stage micro-pipeline stages, using flip-flops with Edge-structure. This circumstance predetermines the usage of 4-phase data transfer pipeline protocol. In these conditions there is an asynchronous pipeline automation developed, as well as the whole structure of micro-pipeline combining both types of micro-pipeline sections. The operation of the pipeline protocol is presented.

Keywords: Micro-pipeline, Four phase handshake protocol, Race conditions, Synchronization.

Въведение

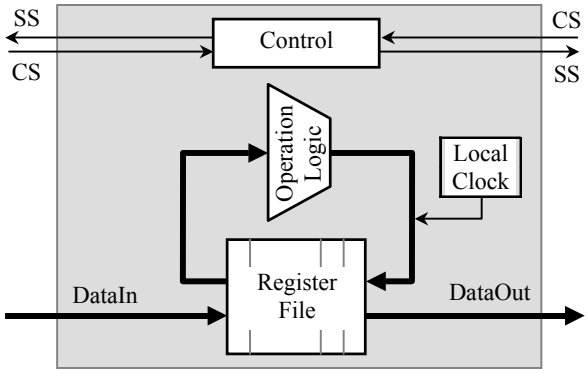
Микроконвейерите съдържат последователно включени микроконвейерни звена, чиято структура се състои от регистър-фиксатор и комбинационна логическа схема [6÷18 и др.]. Фиксаторът поддържа данните, а комбинационната схема реализира необходимите изчисления, но като такава не е задължителна. След всеки един записващ импулс, подаден към фиксатор (конвейерен регистър), в даденото звено постъпват и се преработват нови данни. Така след всеки записващ импулс резултатът от едно звено преминава в следващото звено. В този смисъл, микроконвейерните звена с такава структура могат да бъдат определени като еднотактни. Продължителността на тактовете в отделните звена се определя от продължителността на времето за превключване на комбинационната схема на звеното. Придвижването на данните в асинхронни микроконвейери от споменатия тип, от звено към звено, се реализира въз основа на принципа, наречен "ръкостискане", в 2-фазов или 4-фазов асинхронен протокол. Протоколът се реализира от управляваща схема, която съдържа някакъв вариант на известния Мюлер С-елемент [5÷18 и др.]. Същността на управлението е асинхронна, тъй като придвижването на текущите резултати от звено към звено, е възможно само ако последното е свободно.

В [1],[2],[3] и [4] са разгледани микроконвейерни звена, чиято структура се характеризира с вътрешна обратна връзка. Тези микроконвейерни звена реализират итерационен изчислителен процес и са проектирани като синхронни устройства. Те функционират като такива благодарение на локален тактов генератор. Предвид на вътрешното (локалното) тактуване, микроконвейерните звена от този вид могат да бъдат определени като многотактови. Закъсненията, които такива звена генерират, са съществено различни, ето защо като завършени устройства те могат да бъдат включвани единствено в състава на асинхронни микроконвейери.

Наличието на различни по тип микроконвейерни звена (еднотактови и многотактови) поражда нова задача при комбинирането им в един микроконвейер. Задачата се свежда до синтез на подходящ конвейерен автомат, който да е независим от типа на звената, които свързва. Тук е изложен подходът, който позволява условното уеднаквяване на звената, в резултат на което са създадени условия за синтез на независимия конвейерен автомат. Представена е неговата принципна логическа схема и организацията на микроконвейера.

Многотактово микроконвейерно звено

Многотактовите микроконвейерни звена съдържат вътрешна обратна връзка и могат да се представят със следната обобщена структура:



Фиг. 1. Обобщена структура на многотактово МКЗ

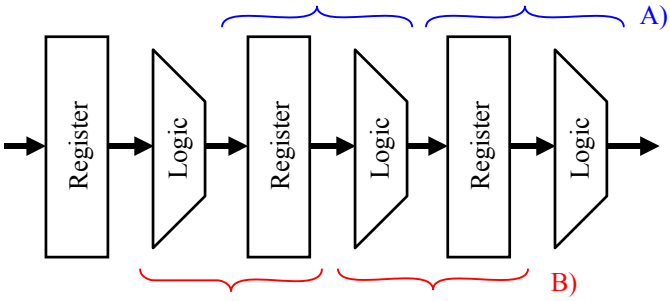
Структурата съдържа три основни елемента – регистров файл (*Register_File*), който се състои от един или повече регистри и съвкупност от комбинационни логически схеми (*Operation_Logic*), реализиращи необходимите изчисления. Най-съществената особеност на тази структура е вътрешната обратна връзка, която е естествена предпоставка за състезания. По тази причина регистрите в многотактните звена са реализирани от динамични тригери със структура *Edge*, превключващи се само по един от фронтовете на управляващия сигнал за запис.

Третият елемент (*Control*) – вътрешната управляваща схема, е неразделна част от тези звена и реализира вътрешносхемното им управление. Този елемент обаче неизбежно участва в диалога със съседните микроконвейерни звена, като обработва и генерира съответните сигнали – флагове на състоянието и управляващи (SS, CS).

Организация на микроконвейера

Микроконвейерите, като последователност от регистри-фиксатори и комбинационни схеми, имат обобщената логическа структура, показана на фигура 2. В повечето публикации тези двойки структурни елементи се разглеждат като свързани, т.е. като един общ елемент – микроконвейерно звено. Това разбиране обаче налага тук да бъдат изложени нови и

оригинални съображения относно структурата на микроконвейера, които извеждат на преден план значението на операцията запис на данни в регистъра фиксатор, а не подредбата на структурните елементи. Разбирането, което е изложено по-долу, е в основата на представеното тук по-долу решение за управление на асинхронния микроконвейер.



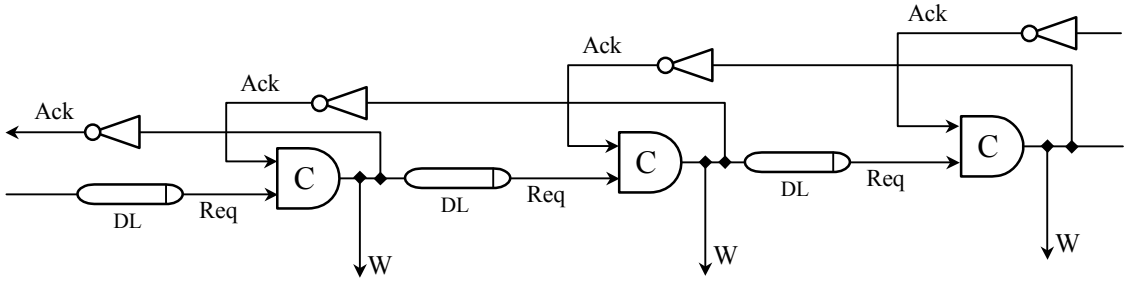
Фиг. 2. Структура на микроконвейер

В горната структура съвсем условно, под микроконвейерно звено може да се разбира двойката регистър-логика (случай А) или двойката логика-регистър (случай В).

Едно микрокроконвейерно звено може да приеме данни от предходно звено само ако е свободно, независимо дали се разглежда организация А или организация В. В първия случай това означава, че за да работи текущото звено с коректни данни, трябва да извършва запис след като логиката на предходното звено завърши своето превключване, т.е. трябва да изчаква закъснението на предходното звено, отчетено спрямо момента на неговия запис. Във втория случай, веднага след записа в регистъра, текущото звено не е в състояние да подава коректни данни към следващото звено, тъй като съответстващата му логика още не ги е изчислила. И в двата случая проблемът е един и същи и е свързан с момента на запис във всеки отделен регистър. Общият извод от тези разсъждения е, че всеки регистър трябва да поддържа данните толкова време, колкото е необходимо на следващата го логика да изчисли своите резултати. В този смисъл всеки един регистър може да бъде определен като зает във времевия интервал с начало записа в предходния регистър и с край краят на превключването на предхождащата го логика и докато данните в него са все още нужни за надеждно записване на резултата в следващия регистър. С други думи, състоянието на фиксатора се определя от това дали следващият фиксатор се е освободил и от това дали предхождащата го логика се е превключила. Както се разбира, изказаното положение не зависи от структурната интерпретация на микроконвейерното звено (случай А или случай В). От тук следва, че е правилно да се говори за свободен или зает регистър-фиксатор, а не за цяло звено.

Ако се приеме, че изчисленията са съсредоточени в логическата схема (като елемент в структурата на микроконвейера), то в общ смисъл на нейно място може да се има предвид всяко многотактово микроконвейерно звено, което представлява интерес тук. Тъй като коректността на данните в конвейера се определя единствено от моментите на запис в регистрите фиксатори, това дава основание за интерпретация като такъв на съответния входен или изходен регистър в многотактовите звена. Поради това, че в многотактовите звена тези регистри са обхванати от вътрешните обратни връзки, още веднъж следва да се отбележи, че те се превключват само по един от фронтовете на управляващите сигнали за запис, които са функция на локалния тактов генератор (*Local_Clock*) (фигура 1).

При последователното свързване на конвейерните звена за управление се използва асинхронна (събитийна) логика, основаваща се най-често на Мюлер С-елемента. При настъпване на такива събития този елемент променя всеки път своето състояние (според таблицата му на истинност) в противоположното, като с това управлява записа на данни във фиксаторите. Управлението на микроконвейера в този случай се изразява със следната последователност (фигура 3), в която не са изразени регистрите фиксатори.

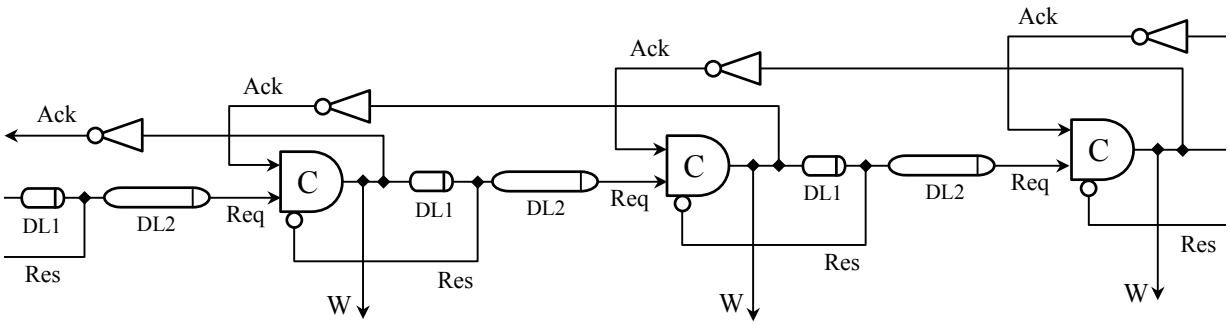


Фиг. 3. Условно представяне на 2-фазов асинхронен микроконвейер

В схемата са възприети обикновено употребяваните означения – DL (*delay*) за закъснение, *Req* (*request*) - за заявка за обмен, *Ack* (*acnowledgement*) - за потвърждение за успешен обмен, W (*write*) - за запис. Закъснителните елементи DL изразяват условно реалните закъснения на комбинационните схеми в отделните звена.

Логиката на това управление е "без връщане в нула" и реализира 2-фазов протокол на обмен. Тя изисква за реализация на фиксаторите запомнящи елементи, които записват данни и по двата фронта на сигнала W, тъй като С-елементът се превключва в кръг от две съседни звена (фигура 3). Тъй като в многотактовите микроконвейерни звена, които следва да се включат в състава на микроконвейера, се използват регистри фиксатори, които работят само по единия от фронтовете, за алгоритъм на конвейерния автомат са възможни само два протокола – 4-фазов или 4-фазов с изпреварване, т.е. логика "с връщане в нула". С цел по-висока производителност тук е приложен само протоколът с изпреварване.

Протоколът за обмен с изпреварване се постига чрез изпреварващо нулиране на С-елемента, което се предлага да се реализира, като се отчете закъснението при запис, въвеждано от тригерите на фикасаторите. Това закъснение може да се отдели условно от общото закъснение на операционната логика по начина, показан на фигура 4.



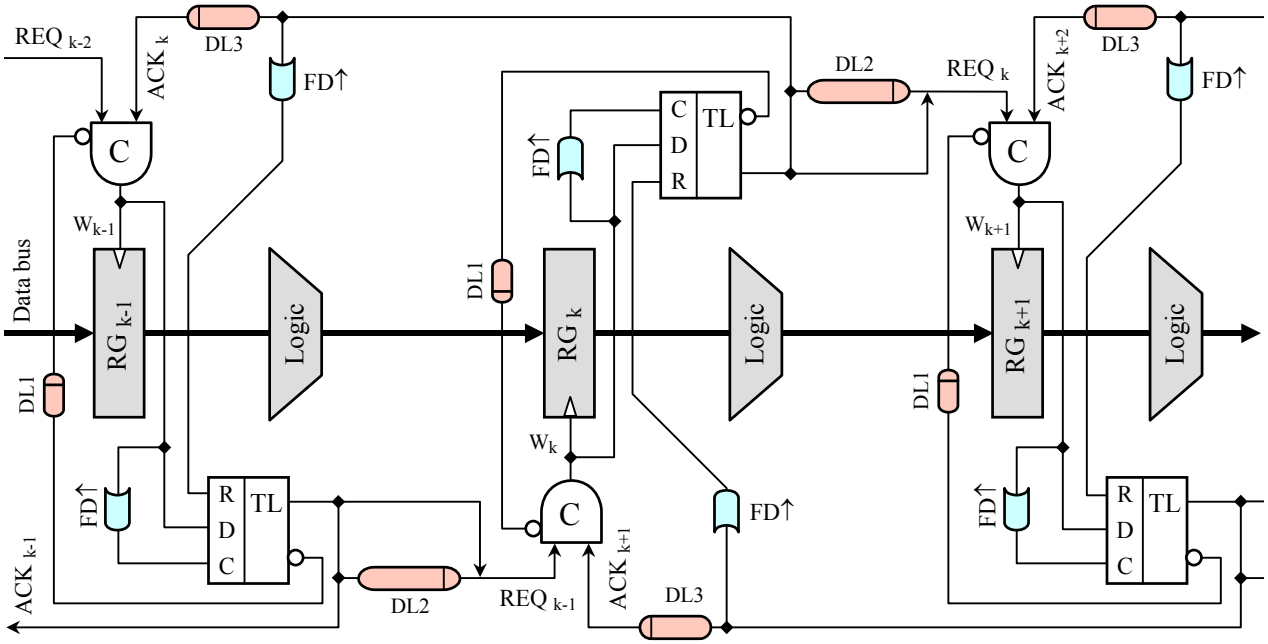
Фиг. 4. 4-фазов микроконвейер с предварително нулиране на С-елемента

От схемата се вижда, че С-елементите са снабдени с допълнителен вход за нулиране, на който е подаден сигналът от закъснението на фиксаторите *Res* (*response*). Елементът, който го реализира е означен DL1. DL2 е елементът, който отчита закъснението на операционната логика в звеното.

След отчитане на изложените съображения както относно протокола, така и относно регистрите фиксатори, е получен вариант на структурната схема на микроконвейер, в който могат да се редуват както еднотактови така и многотактови микроконвейерни звена. Схемата е представена на фигура 5.

Конвейерният автомат е изграден върху логиката, представена на фигура 4. Всеки С-елемент работи съвместно с един D-*latch* тригер, който запомня състоянието на С-елемента при запис във фиксатора, т.е. при старт на микроконвейерното звено. Състоянието на С-елемента е необходимо да се запомни, защото в противен случай не е възможно правилното генериране на сигнала *Req*. След това С-елементът изпреварващо се нулира чрез сигнала от инверсния изход на D-*latch* тригера. Тъй като С-елементът и

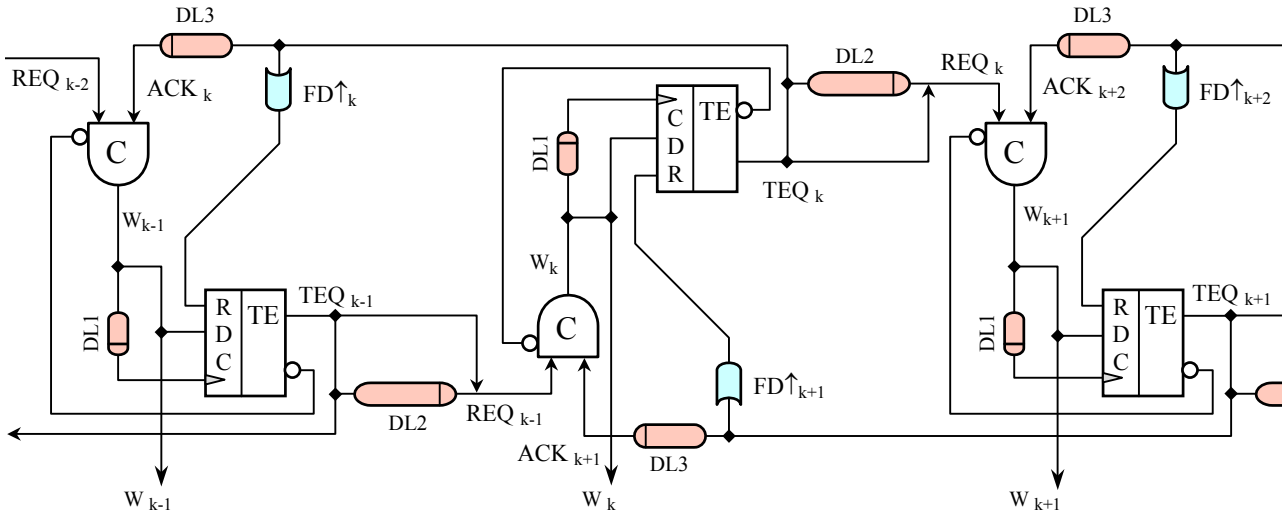
D-тригерът са обхванати от обратна връзка, при което задният фронт на сигнала W се застъпва във времето с предния фронт на неговото закъснение DL1, то надеждното записване в този тригер изисква фиксиране на състоянието по предния фронт на закъснението DL1. Именно с такава цел във веригата на вход C е включен детекторът на преден фронт DF↑, а във веригата на обратната връзка – закъснението DL1, което задържа във времето задния фронт на сигнала W, по отношение на задния фронт на записващия импулс DF↑. За използвания D-*latch* тригер продължителността на генерираният импулс t1 трябва да е по-голяма от времето за неговото превключване (t1>2.τ). С τ е означено времето за превключване на един логически елемент в схемата на тригера.



Фиг. 5. Структурна схема на микроконвейера

От своя страна D-*latch* тригерите се нулират изпреварващо от сигналите *Ack*. Това е необходимо, тъй като се очаква при създаване на съответните условия, C-елементът да се превключи отново в единично състояние, което в последствие да се фиксира от същия тригер. Имайки предвид потенциалния характер на асинхронните сигнали във веригите за управление на микроконвейера, нулирането на D-тригера трябва да се осъществи по предния фронт на сигнала *Ack*. За целта пред входа R на D-тригера е поставен детектор на фронт FD↑, аналогичен с вече описания. Така след нулиращия импулс, на входа R се установява ниско ниво, докато в същото време сигналът *Ack* продължава да има високо ниво. За да се отдалечи във времето превключването на C-елемента, сигналът *Ack* е задържан от елемента DL3. Стойността на закъснението t3 следва да бъде по-голяма от времето за превключване на C-елемента плюс времето за превключване на тригера. Така ще се осигури надеждното нулиране на последния и отпадане на нулиращия импулс, с което няма да се попречи на следващия запис в същия тригер.

Освен представения по-горе вариант за схемно решение на микроконвейерния автомат, може да се предложи още един вариант, при който се избягва въвеждането на детектор на фронт във веригата на закъснението DL1. Във втория вариант, вместо D-*latch* тригер конвейерният автомат използва динамичен D-*edge* тригер. Така по предния фронт на закъснението DL1 ще се запише единичната стойност на сигнала W, чието последващо нулиране по изходния от същия тригера сигнал и обратната връзка, ще бъде достатъчно закъсняло за да попречи на това. Схемата на това решение е показана на фигура 6.



Фиг. 6. Микроконвейер с динамичен тригер

Закъсненията DL1 и DL2 имат естествени изходни точки в принципните логически схеми на многотактовите микроконвейерни звена, които представляваха тук интерес. В този смисъл тези закъснения, въпреки че са изобразени условно, имат реално измерение, което при всяко изчисление се определя от реалните входни за звената данни и тактовата честота на локалните им генератори.

Забележка: Всички микроконвейерни звена, както и всеки конвейерен автомат в състава на даден конвейер, следва да се установяват принудително в начално състояние след включване на захранването, както и в други ситуации, изискващи това състояние (не е показано на фигура 6). В начално състояние микроконвейерните звена се поставят в състояние на готовност, при което те генерират сигналите *Ack*. За стартиране на микроконвейера е необходимо в началното звено да се подаде сигнал от тип *Req*.

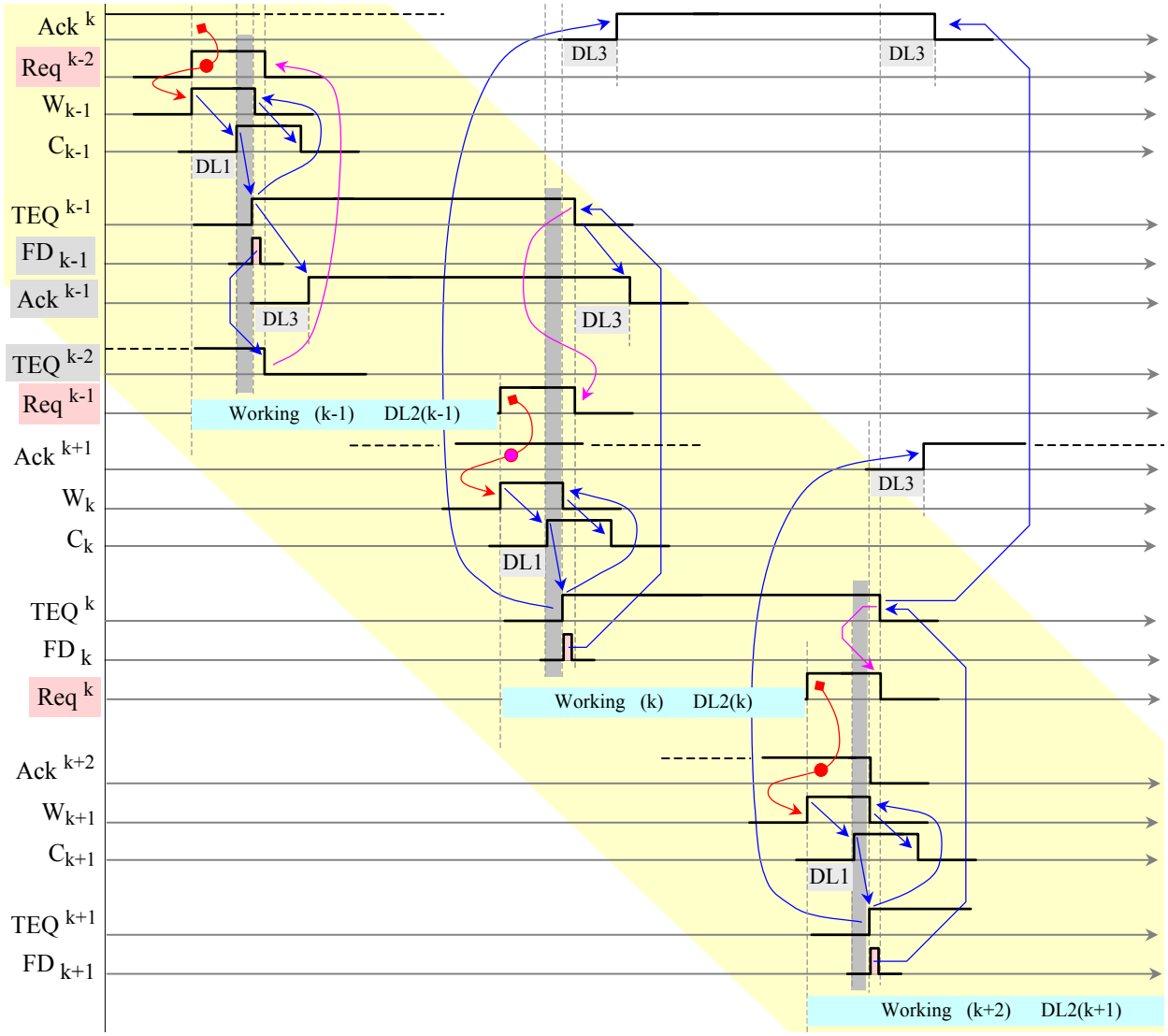
По-детайлно процесът на последователното стартиране на микроконвейерните звена №(k-1), (k) и (k+1), изобразени на фигура 6, е представен чрез времедиаграмата от фигура 7. Преди да бъде пояснена времедиаграмата обаче е полезно да се отбележи, че при стартиране, всеки С-елемент се превключва от начално в противоположно състояние (W=1), в което престоява кратко време, определено от закъснението DL1. Това състояние, преди да изчезне, както вече беше отбелязано, се запомня в D-тригера по предния фронт на сигнала, излизащ от това закъснение. Появяващата се нула от инверсния изход на този тригер превключва С-елемента обратно в начално състояние (W=0).

Генерираният от правия изход на D-тригера сигнал *TEQ*, маскира в права посока сигнала *Req*, с което изпреварващо в сигнала *Req* се въвежда заден фронт, вместо да се чака разпространението му в елемента DL2, след нулиране на този тригер.

В обратна посока разпространението на сигнала *TEQ* се задържа във времето от закъснението DL3. Това е необходимо, за да се изчака рестартирането на D-тригера по предния фронт на *TEQ*, което се постига чрез детектора на фронт във веригата на входа R. Така, след установяване на конвейерният автомат в изходно състояние и след поява на сигнала *Ack*, може да се извърши повторно стартиране на микроконвейерното звено.

Реалното свързване на конвейерния автомат с многотактово микроконвейерно звено CU_k е представено на фигура 8. В схемата е използвано многотактно микроконвейерно звено, реализиращо алгоритмична структура цикъл с предварително известен брой повторения от вида, разгледана в [2]. От схемата се вижда, че импулсът W_k записва данни във регистъра фиксатор (*Pipeline-RG*), според логиката на конвейерния автомат. За стартиране на изчисленията в многотактовото звено се използва сигналът TEQ_k, който

попада на вътрешния синхронизатор на звеното, реализиран чрез динамичен D-тригер със структура *Edge*. Логиката за управление на този тригер е построена така, че той формира необходимия синхронен с локалната тактова последователност *Clock* стартов импулс *Enable*.



Фиг. 7. Времедиаграма на конвейера

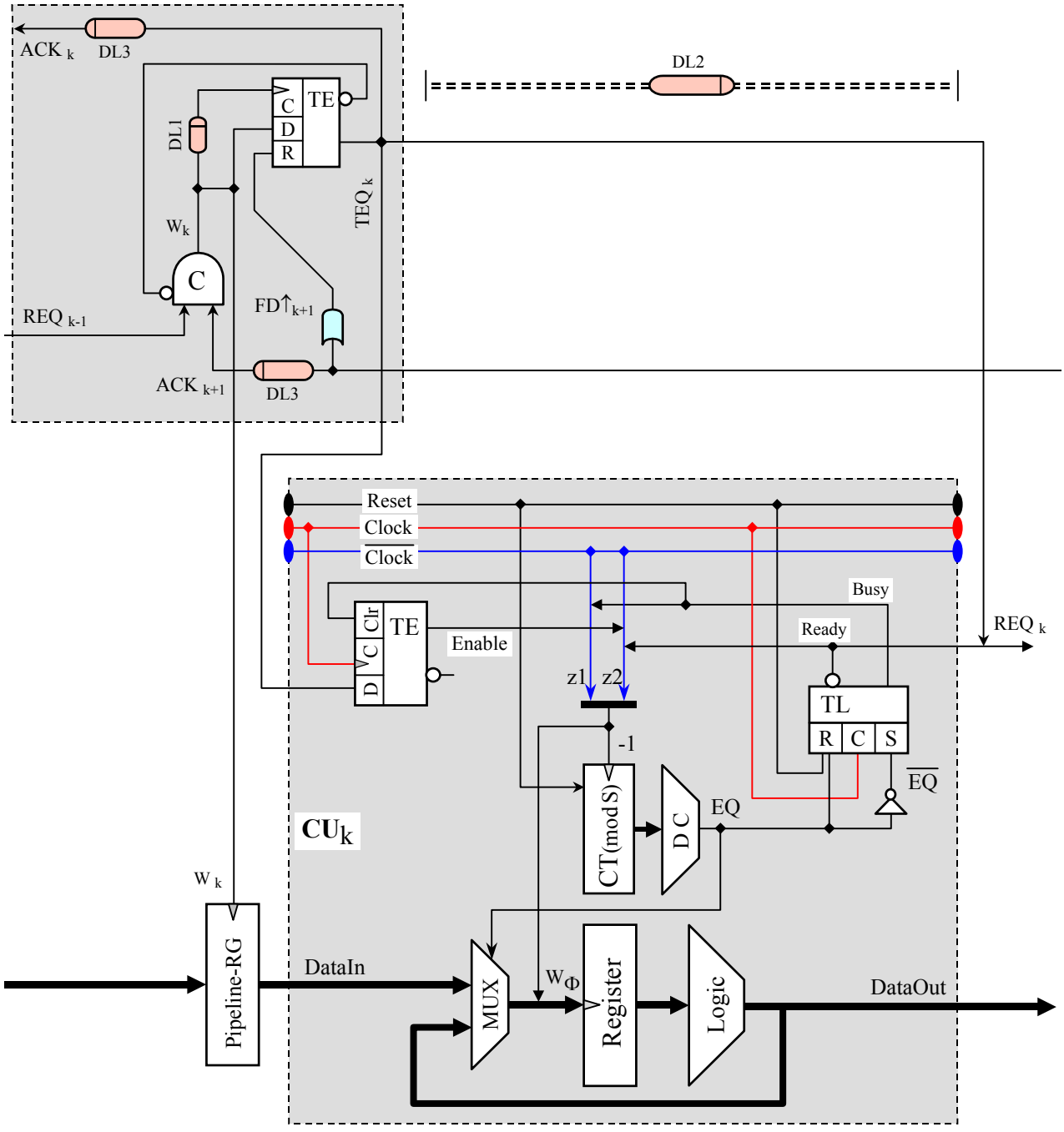
Изчисленията в многотактовото звеното завършват с появата на сигнала *Ready*, който се използва в качеството на конвейерен сигнал Req_k. Закъснението, което внася това микроконвейерно звено, беше преди това условно означавано DL2 (фигури 5 и 6). Тук реалният изход на сигнала Req_k е показаният на фигура 8.

Заключение

Общо приети правила за проектиране на микроконвейерни звена все още не са формулирани. Както тук беше споменато, микроконвейерните звена могат да имат различна вътрешна структура и начин на функциониране. От това следва, че в един реален алгоритъм е възможно тяхното последователно подреждане в разнообразни съчетания. Възможно е всяко звено да притежава различни вътрешни състояния, но то би следвало да бъде адаптивно към конвейерната организация. На това ниво всички участници са равноправни и подчинени на изложената тук организация.

Предложената обобщена интерпретация на микроконвейерните звена позволява удовлетворяване на тази организация в лицето на конвейерните протоколи и сигнали. Положителен резултат от тази интерпретация върху многотактовите микроконвейерни звена е преодоляването на заплахата от дублиране на конвейерните фиксатори с входните или с изходните регистри на звената.

Реализираният 4-фазов асинхронен конвейерен протокол на обмен между звената е във вариант с изпреварващо нулиране (връщане в нула), което води до по-висока производителност като цяло на конвейера.



Фиг. 8. Схема за свързване на конвейерен автомат
с многотактово микроконвейерно звено

Литература

[1]. Тянев, Д., С. Колев, Д. Янев, Метод за реализация на апаратни самоуправляващи се циклически структури - част II, Списание "Компютърни науки и технологии", ТУ-Варна, ISSN 1312-3335, година V, брой №2/2007, стр. 23-30.

[2]. Tyanev, D., S. Kolev, D. Yanev, Micro-pipeline Section For Condition-Controlled Loop, International Conference on Computer Systems and Technologies - CompSysTech'09, 18-19 June 2009, Ruse, Bulgaria, pp. I.4 1-5.

[3]. Tyanev, D., D. Yanev, S. Kolev, Method for realization of self-controlling loop apparatus structures, Fifth International Scientific Conference Computer Science'2009, 5-6 November 2009, Sofia, Bulgaria.

[4]. Тянев, Д. С., Колев С. И., Йосифов В., Метод за реализация на апаратни самоуправляващи се циклически структури, Годишник на ТУ-Варна, Юбилеен сборник "45 години ТУ-Варна", 2007, ISSN 1311-896X, стр. 130-135.

[5]. Миллер, Реймонд Е., Теория переключательных схем, том 2 – последовательностные схемы и машины, Москва, Издательство "Наука", 1971.

[6]. Sutherland, Ivan E., Micropipelines, http://research.sun.com/vlsi/Publications/KPDisclosed/micropipelines/cmicropipelines.pdf

[7]. Chelcea, Tiberiu Girish Venkataramani, Seth C. Goldstein, SelfResetting Latches for Asynchronous MicroPipelines, Proceedings of the 44th annual ACM/IEEE Design Automation Conference, p. 986-989, June 2007 www.cs.cmu.edu/~phoenix/papers/dac07-sr.pdf

[8]. Chammika Mannakkara, Tomohiro Yoneda, Asynchronous pipeline controller based on early acknowledgement protocol, National Institute of Informatics: Department of Informatics, Graduate University for Advanced Studies, Tokyo, Japan, NII-2009-015E, Sept. 2009. http://research.nii.ac.jp/TechReports/08-009E.pdf

[9]. GALAXY-Project, Milos Krstic, Xin Fan, Miroslav Marinkovic, Frank Gürkaynak, Christoph Heer, Sören Sonntag, Specification of optimized GALS interfaces and application scenarios, GALAXY, 12-2008. http://www.galaxy-project.org/files/D3_IHP_R002_Spec_GALS_Int_v2.pdf

[10]. Muttersbach, Jens, Globally asynchronous locally synchronous, architecture for VLSI Systems, PhD thesis [PDF], ETH Zurich, Diss. ETH №14155, 2001. http://e-collection.ethbib.ethz.ch/eserv/eth:24203/eth-24203-01.pdf

[11]. Krstic, Milos, Eckhard Grass, New GALS Technique for Datapath Architectures, in Integrated circuit and system design: power and timing modeling, by Jorge Juan Chico, Enrico Macii, p.161, books.google.com, 2003; Lecture Notes in Computer Science, ISSN 0302-9743, Volume 2799/2003.

[12]. *Eckhard* Grass, Frank Winkler, *Miloš* Krsti, Enhanced GALS Techniques for Datapath. Applications, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.72.2933&rep=rep1&type=pdf

[13]. Xin Fan, Miloš Krstić, Eckhard Grass, Analysis and Optimization of Pausible Clocking based GALS Design, www.galaxy-project.org/files/358Fan.pdf .

[14]. Yun, Kenneth Peter A. Beerely, Julio Arceo, High-Performance Asynchronous Pipeline Circuits, In Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, IEEE Computer Society Press, 1996, p. 17-28. http://citeseer.ist.psu.edu /3648.html

[15]. Ginosar, Ran, Fourteen ways to fool your synchronizer - Asynchronous Circuits, IEEE Proc. of the Ninth International Symposium on Asynchronous Circuits and Systems (ASYNC'03), http://webee.technion.ac.ilpeopleranpapersSync/Errors/Feb03.pdf .

[16]. Hirschmann, Stefan, Asynchronous processors, Seminar Embedded System Design, Institute of Computer Science, University of Innsbruck, February 25, 2008, http://informatik.uibk.ac.at/teaching/ws2007/esd/async.pdf .
[17]. Chang-Jiu Chen, Wei-Min Cheng, Hung-Yue Tsai, Jen-Chieh Wu, A quasi-delay-insensitive microprocessor core Implementation for Microcontrollers, Journal of information science and engineering 25, 543-557 2009, www.iis.sinica.edu.tw/page/jise/2009/200903_13.pdf .
[18]. Singh, MontekChapel Hill, Steven M. Nowick, US Patent, 2005/0156633, Circuits and methods for high-capacity asynchronous pipeline processing. www.patentstorm.us/patents/7053665/description.html
[19]. Tocci, Ronald J. Neal S. Widmer, Digital systems: principles and applications, 8th ed., Prentice-Hall Inc., ISBN 0-13-085634-7, 2001.
[20]. Chu, Pong P. , RTL Hardware Design Using VHDL: Coding for Eflciency, Portability and Scahbility, Wiley IEEE Press, ISBN-13: 978-0-471-72092-8, 2006.
[21].Daniel Page, Practical Introduction to Computer Architecture, Springer, ISBN 978-1-84882-255-9, 2009.
[22]. Tinder, Richard F., Asynchronous Sequential Machine Design and Analysis, Morgan and Claypool, ISBN: 9781598296907, 2009.

За контакти:
Доц. д-р инж. Димитър Тянев
Катедра “Компютърни науки и технологии”
Технически университет - Варна
e-mail: dstyanev@yahoo.com
Рецензент: доц. д-р инж. Надежда Рускова, ТУ-Варна

ЕДНА НОВА LFSR-СТРУКТУРА ЗА ГЕНЕРИРАНЕ НА ПСЕВДОСЛУЧАЙНА ПОСЛЕДОВАТЕЛНОСТ

Божидар С. Дойнов, Димитър С. Тянев

Резюме: Предложена е оригинална структура на генератор за псевдослучайни последователности с равномерен закон на разпределение, основаваща се на преместващи регистри с линейни обратни връзки. Структурата интегрира идеите на Макларън-Мерсальия и Клапер-Горецки относно генераторите на случайни числа. Представени са експерименталните статистически оценки, получени за нея чрез различни статистически тестове. Коментирани са получените положителни резултати.

Ключови думи: Псевдослучайна последователност, LFSR, Статистически тестове.

Original LFSR Structure for Generating Pseudorandom Sequences

Bozhidar S. Doynov, Dimitar S. Tyanev

Abstract: This article presents an innovative structure of a uniform distribution pseudo noise generator based on linear feedback shift registers. The structure integrates MacLaren-Marsaglia and Goresky-Klapper ideas related to random number generators. The paper presents the experimental statistic assessments of the system obtained through various statistic tests. The beneficial results are discussed.

Key words: Pseudo noise sequences, LFSR, Statistical tests.

1. Въведение

Генераторите на псевдослучайни числа са предмет на интерес в много области на научното познание, като автоматика, криптография, комуникации, радиолокация, хидроакустика и др. Генерирането на псевдослучайни последователности най-често се реализира апаратно, като законът на разпределението е равномерен.

Основните схеми на ГПСП, които се прилагат в споменатите приложения, са построени на базата на физически елементи, чиито характеристики имат псевдослучайно поведение, или чрез апаратна реализация на различни алгоритмични схеми [2,4,10,13]. Използването на физически елемент обаче е свързано с изключително прецизни схеми за поддържане на стабилен режим на работа.

Втората възможност, и в частност ГПСП, реализирани чрез преместващи регистри с линейни обратни връзки (LFSR), поради своята лесна реализация, намират широко приложение [3,5,7]. Високо качество в тези схеми обаче може да се постигне само при много големи дължини (по-голяма от 2^{15}[bits]). Както е известно обаче голямата дължина от своя страна създава изчислителни затруднения, свързани с намирането на примитивни многочлени от висока степен [11]. Освен това, генерирането на класически M-последователности с LFSR е податливо на криптографическия алгоритъм на Берлекемп-Месси [5].

В настоящата статия се предлага една нова структура за генериране на псевдослучайна последователност. Стремежът е повишаване качествата на ГПСП, реализирани чрез LFSR с по-малки дължини, като при това се цели още да бъдат подтиснати в максимална степен споменатите им недостатъци. Така, отчитайки реалните възможности на съвременната елементна база от една страна и съществуващата потребност от друга се разработват ГПСП с увеличен период и усложнена крипто зависимост.

2. Класическа схема и основни зависимости

Максимално възможният период P на изходната последователност е функция от броя на различните състояния на регистъра, както и от неговата дължина n, и се изразява както следва

$$P = 2^n - 1 \ . \tag{2.1}$$

При имплементация в двоична система, обратната задача има решението

$$n = \lfloor \log_2(P) \rfloor \ , \tag{2.2}$$

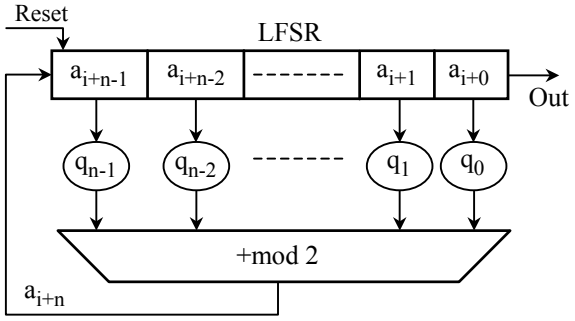
където дължината n на регистъра се измерва в битове [b].

Преследвайки псевдослучайното подреждане на изходните битове, общият вид на функцията на обратната връзка трябва да представлява неприводим полином от n-та степен [7]. Този полином се изразява чрез следната рекурентна формула

$$q(x) = q_0 + \sum_{i=1}^{n} q_i \cdot x^i \ , \tag{2.3}$$

където $q_1, q_2, ..., q_n$ са рекурсивни коефициенти, а x е формална променлива на полинома. Съвкупността от коефициентите q_i може да се разглежда като маска, която определя обратните връзки на схемата, тъй като за коефициентите са позволени само стойностите 0 и 1.

В съответствие с казаното, ГПСП на базата на LSFR може да се представи със следната схема (фигура 2.1)



Фиг. 2.1. Принципна схема на LSFR с дължина n [b]

ГПСП е изграден от един n битов регистър, който измества съдържанието си надясно и комбинационна логика, реализираща изчисленията (2.3). Правилното функциониране на схемата е осигурено за всяко начално съдържание на регистъра с изключение на нулевата комбинация. Ето защо началното съдържание в техническата реализация следва да се осигурява принудително чрез сигнала *Reset*. За начално съдържание може да бъде избрана например комбинацията “1000...00”.

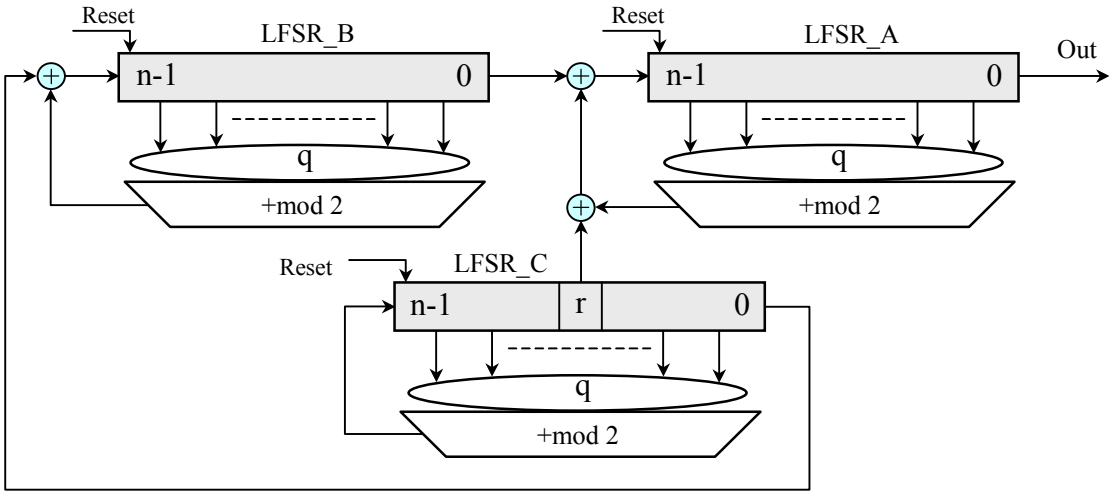
3. Предложение за схема на генератор

В настоящото изследване е представена оригинална структура на ГПСП, използваща LFSR. В нея, в подходяща комбинация, са използвани възможни за техническа реализация и с по-ниско качество ГПСП.

Предлаганата структура е постигната въз основава на идеята в метода на Макларън-Мерсалия [6,4,13], която изисква подходящо комбиниране на два генераторни елемента от един и същи ранг [6]. Освен това, имайки предвид зависимостта на класическия ГПСП от криптоатаки, предложената структура допълнително интегрира подхода на Клапер и Горецки [5]. В съответствие с този подход в структурата на ГПСП е добавена външна памет, с което

се разрушават рекурентните връзки в изходната битова последователност, в резултат на което криптоатаката е затруднена.

Позитивните резултати в предлаганата структура са търсени чрез подходящо обединяване на тези две идеи. Като допълнителна памет в структурата на ГПСП е добавен модифициран LFSR. Модификацията е реализирана въз основа идеята на Макларън-Мерсалия. Този допълнителен елемент е статистически независим. Окончателно синтезираната структура на ГПСП е представена на фигура 3.1.



Фиг. 3.1. Логическа структура на предложения генератор

В нея се съдържат три елемента от един и същи вид – A, B, C. Изходът на генериращата структура *Out* е от десния край на елемент A. Елементите B и C са включени в съответствие с идеята на Макларън-Мерсалия, в качеството им на първични генериращи елементи. От друга страна, включването на тази група към първичния генериращ елемент A, е в съответствие с идеята за допълнителна външна памет на Клапер и Горецки. Групата се подключва към елемента A чрез функциите *Mod2*. По този начин постъпващите в елемент A битове от елементите B и C са статистически независими, което разрушава рекурентните връзки (2.3) в изходната последователност. Генераторните елементи B и C се включват в обратната връзка на елемента A различно. В генераторния елемент C е създаден допълнителен изход, чрез който той се включва към обратнатна връзка на елемента A. Този изход следва да бъде взет от бит, чийто номер r представлява взаимно просто число с дължината на регистъра n. Поради тази усложнена функционалност в обратната връзка на основния генериращ елемент A главния положителен ефект се състои в значително удължаване на периода на генерираната последователност в сравнение с неговите собствени възможности.

4. Изследване на предложения генератор със статистически тестове

Предложената структура за генериране на псевдослучайна последователност (фиг. 3.1) беше оценена с различни статистически тестове, целящи да потвърдят закона на разпределение на генерираната изходна последователност, а получените за нея оценки да послужат още и за качественото й оценяване при сравнение с изходната родителска схема (фигура 2.1). Параметрите на експериментираната схема са: дължина n=16 [b] за всички регистри; комбинации от рекурсивни коефициенти за елемент A (1001101000001111), за елемент B (1001001111110011) и за елемент C (1011111000010111); изход от елемент C, r е № 3.

От множеството разнообразни статистически тестове са избрани следните 4 теста:

- Стандартен тест за проверка на закона на разпределение на случайната величина, реализиран в MatLAB ;
- Тест за честота (тест на NIST [9]);
- Графичен хистограмен тест ;
- Проверка на закона на разпределение чрез критерия на Пирсон (Хи-квадрат).

Приета е тази съвкупност от тестове за достатъчно авторитетна, имайки предвид закона на разпределение, на който се подчинява генерираната последователност.

4.1. Стандартен тест за проверка на закона на разпределение на случайната величина в среда MatLAB [8]

Тестът се провежда от вградената в MatLAB функция *runstest(x)*, чийто аргумент *x* представлява генерирана последователност с дължина *M* [b]. Командата за теста е [h,p]=runstest(x). В резултат на обработката на масива *x*, функцията *runstest* връща:

- За променливата *h* стойност нула (*h=0*), ако приема нулевата хипотеза H_0, и стойност единица (*h=1*), ако отхвърля нулевата хипотеза H_0. За нулева хипотеза е прието твърдението за равномерен закон на разпределение ;
- За променливата *p* стойност, която представлява вероятностността за приемане на нулевата хипотеза, при зададено в теста ниво на значимост от 5% .

Числените резултати за теста са сведени в таблица 4.1.1.

Таблица 4.1.1. Оценки за последователности с различни периоди

M	2^{17}	2^{18}	2^{19}	2^{20}	2^{21}
h	0	0	0	0	0
p	0.9362	0.9579	0.9791	0.9914	0.9961

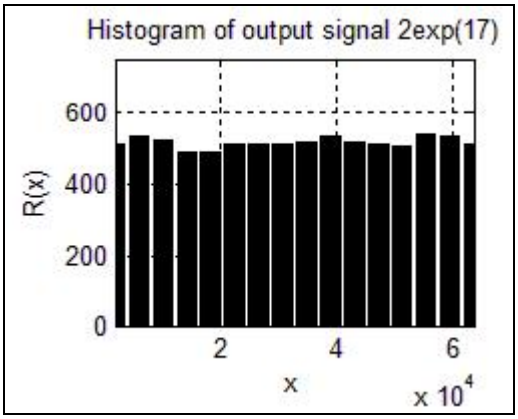
Очевидно тестът не отхвърля нулевата хипотеза, при това с нарастваща категоричност.

4.2.Тест за честота [9]

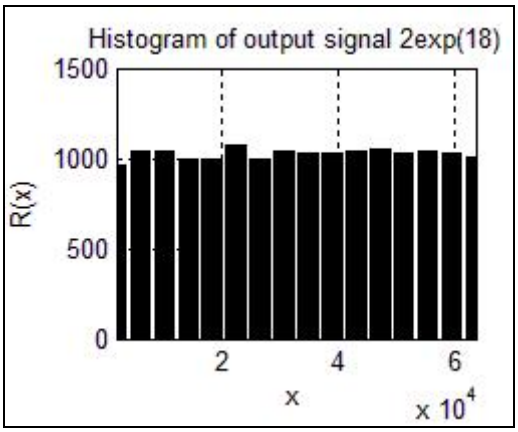
Целта на този тест е да се определи съотношението на нулите и едениците в генерираната последователност, които трябва да са почти равни на брой. Този тест е осигурен в MatLAB чрез функцията *erfc*(*x*), където аргументът $x = S_{obs}$ представлява стойност на така наречената тестова статистика, изразена с формулата $S_{obs} = |S_n| / \sqrt{2.n}$. Изчислението на стойността на тестовата статистика изисква предварителна подготовка на генерираната последователност, която се състои в следното - всички n елемента на последователността се преобразуват по правилото: всяка генерирана 1 се замества с +1, а всяка 0 с –1. След това се изчислява сумата от елементите S_n. Командата за теста е P=erfc(x). Функцията връща стойност, която позволява да се оцени последователността според изказания в началото смисъл [12]. В най-добрия случай стойността на оценъчната функция е 1. Реално изчислената след експеримента с теста оценка има стойността P=0,9860. Тя е получена при обработка на целия период на генерираната последователност и високата й близост до най-добрата стойност за теста се възприема като оценка за високо качество на генератора.

4.3. Графичен хистограмен тест [9]

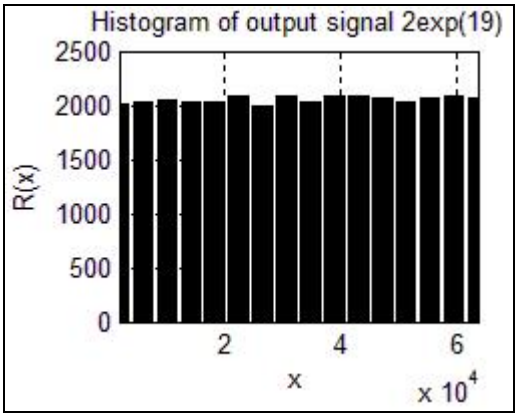
Хистограмният тест е най-често прилаган поради това, че предлага непосредствена графична визуална представа за разпределението, както и числена оценка за относителните честоти на попадение. Направено е конвертиране на 16 битови двоични числа в десетични. За получаване на оценките по този тест е използувана функция *hist* в среда на MatLAB. На фигурите 4.3.1, 4.3.2, 4.3.3 и 4.3.4 са представени хистограмите на първите 4 последователности според таблица 4.1.1.



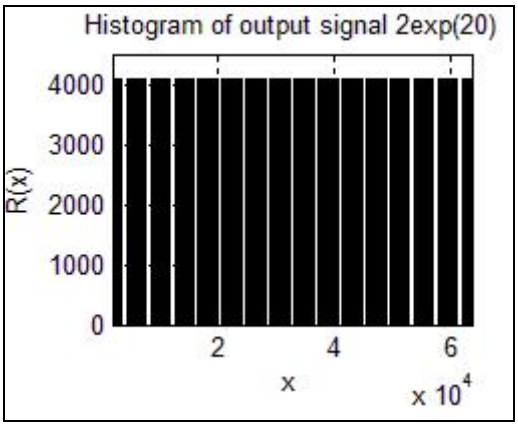
Фиг. 4.3.1. Последователност с период 2^{17}



Фиг. 4.3.2. Последователност с период 2^{18}



Фиг. 4.3.3. Последователност с период 2^{19}



Фиг. 4.3.4. Послед-т с период 2^{20}

От хистограмите, показани на фигурите 4.3.1, 4.3.2, 4.3.3 и 4.3.4, се виждат попаденията на генерираните 16 битови числа в 16 неприпокриващи се интервала. С увеличаване периода на последователността, разликата от броя на попаденията в съседните интервали стават все по-малка и за последователността с период 2^{20} те се изравняват.

Хистограма за последователност с период 2^{21} не е построена, тъй като при период 2^{21} тя има пълен брой на повторения на 16 битови числа.

Таблица 4.3.1. Параметри на последователности с различни периоди

M	2^{17}	2^{18}	2^{19}	2^{20}	2^{21}
Брой генерирани бита	131072	262144	524288	1047576	2097152
Брой 16 битови числа	8192	16384	32768	65536	131072
Брой повтарящи се числа	0	0	0	2	65538

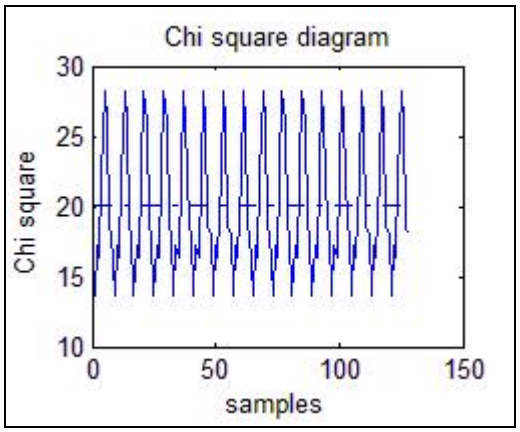
Получената при период 2^{20} последователност е с най-голяма дължина и най-малък брой на повтарящи се 16 битови числа, което се възприема като оценка за високо качеството на генерирана последователност с равномерно разпределение.

4.4. Тест с критерий на Пирсон (Хи-квадрат) [9

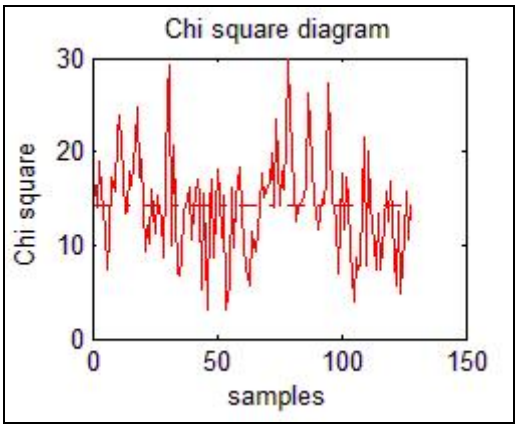
Целта на този тест е, с помощта на критерия на Пирсон, да се провери нулевата хипотеза, която издига твърдението, че генерираната псевдослучайна последователност не противоречи на равномерния закон на разпределение. Това твърдение се издига при предварително избрано ниво на достоверност. Нивото на достоверност, избрано за провеждания тест е $\alpha=0.05$. При 15 степени на свобода, стойността на критерия, взета от съответната таблица [1] е $\chi^2(obs)=25,0$.

Проведеният експеримен има следната организация – при 16 битов регистър, периодът на генерираната последователност се подчинява на зависимостта (2.1). Броят на генерираните битове е равен на броя на тактовете за изместване. Генераторите се стартират за натрупване на последователни масиви от по 509 16 битови елементи, чиито брой зависи от M. При това е спазено условието, че дължината на елементите и обема на масивите да са взаимно прости числа.

За всяка извадка от 509 16 битови числа е изчислена стойността на оценката на критерия на Пирсон. Така получените оценки са показани на фигури 4.4.1 и 4.4.2 в графичен вид съответно за родителската схема и за тук предложената.



Фиг. 4.4.1. χ^2 на родителската схема



Фиг. 4.4.2. χ^2 на предложения генератор

Изводите, които този тест позволява да се направят, са следните:

1. Предложеният генератор има по-голям период спрямо родителския ;
2. Средната стойност на оценките на критерия на Пирсон за отделните извадки (20, вижте фигура 4.4.1) е по-голяма от тази за изследвания генератор (14,5, вижте фигура 4.4.2) ;
3. За повече от 80% от извадките на предложения генератор, при еднакви условия, стойностите на оценките на критерия на Пирсон са фактически по-малки от тези на родителската схема.
4. Като цяло, тестът по този критерий също приема хипотезата, че генерираната псевдослучайна последователност не противоречи на равномерния закон на разпределение.

Заключение

Получените експериментални резултати дават основание да се твърди, че предложеният генератор е по-добър от използваните LSFR, влизащи в състава му, както по дължина на изходната последователност, така и по статистически оценки. Скоростта на генериране на изходни последователности и на двете схеми е еднаква. Максималната честота $f_{max}=(T_{min})^{-1}$ за генериране може да се оцени чрез максималното време за превключване на схемата, което е около $8.\tau=T_{min}$, където с τ е означено времето за превключване на един

градивен логически елемент. Предложената структура е независима от дължината на използваните LFSR, което я прави лесно модифицируема. Условия за това съществуват в елементната база на програмируемите логически матрици, например на фирма *Xilinx*.

Литература

[1]. Зажигаев, Л., Романиков, Ю., Методы планирования и обработки результатов физического эксперимента, Издательство "Атомиздат", Москва, 1978.
[2]. Дядюнов, Н.Г., Сенин, А.И., Ортогональные и квазиортогональные сигналы - под ред. Е. М. Тарасенко, Издательство "Связь", Москва, 1977.
[3]. Ksendzuk, A.V., Volosyuk, V.K, Some aspects of usage of the pseudo noise sequences in the radiolocation systems **,** Ultrawideband and Ultrashort Impulse Signals, 2004 Second International Workshop, Volume, Issue, 19-22 Sept. 2004.
[4]. Иванов, М.А., Чугунков, И.В., Теория, применение и оценка качества генераторов псевдослучайных последовательностей , Издательство "Кудиц-Образ", Москва, 2003.
[5]. Диксон, Р. К., Широкополосные системы, Издательство "Связь", Москва, 1979.
[6]. Кнут, Д.Е., Исскуство програмирования для ЭВМ, Том 3, Пер. с англ., Издательство "Мир", Москва, 1978.
[7]. Goresky,M. , Klapper A., Algebraic shift register sequences, 2009. http://www.cs.uky.edu/~klapper/algebraic.html
[8]. MacLaren, M.D., Marsaglia, G., Uniform Random Number Generators, Jurnal of the Association for Computing Machinery, vol.12, N1, Jan. 1965.
[9]. Skolnik, M. I., Radar Handbook, Third Edition, ISBN 978-0-07-148547-0, The McGraw-Hill Companies ,2008.
[10]. Handbook on satellite communications, 3 Edition,Wiley.
[11]. Prasad, R._Ruggieri M., Applied Satellite Navigation Using GPS, GALILEO and Augmentation Systems, Artech House, 2005.
[12]. Беджев, Б., Мутков, В., Кодиране и сигурност в телекомуникационните мрежи, РУ "Ангел Кънчев", Русе, 2009.
[13]. MatLAB, User Guide, MathWorks, 2008.
[14]. A ststistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Application, August 2008, NIST, US Department of Commerce.
[15]. Харин, Ю.С., Степанова, А.И., Практикум на ЭВМ по математической статистике, Издательство "Университетское", Минск, 1987.
[16]. Беджев, Б., "Повишаване шумозащитеността на радиолокационните комплекси на базата на съвременни алгебрични методи" - дисертация за присъждане на научна степен "доктор на науките", 2003.
[17]. Доневски, Б., Алтимирски, Е., Кратък справочник по специални функции за инженери, ТУ-София, София, 2006.
[18]. Лобоцкая, Н., Морозов, Ю., Дунаев, А., Высшая математика, Издательство "Вышэйшая школа", Минск, 1987 год.
[19]. Tyanev, D., Petkova Y., Tyaneva A., New elements in the method of Mclaren-Marsaglia, Sovidius University Annals of Mechanical Engineering, volume IV, Tom I, year 2002,

За контакти:
инж. Божидар Стефанов Дойнов
Катедра "Радиотехника"
e-mail: bdoinov@gmail.com
доц. д-р Димитър Стоянов Тянев
e-mail: dstyanev@yahoo.com

Рецензент: доц. д-р инж. Д. Генов, ТУ – Варна

РАЗРАБОТКА НА АЛГОРИТМИ И ИНСТРУМЕНТАЛНА СРЕДА ЗА ПРОЕКТИРАНЕ И ИЗСЛЕДВАНЕ НА МЕНЮТА.

Митко М.Митев

Резюме: В доклада е представена класификация на менюта за управление на програми. Дефиниран е проблема за минимизиране движението на курсора при лентови менюта. Въведена е формализация посредством теория на графите. Теоретично е обоснован априорен алгоритъм за подреждане на позициите от менюто. Предложена е технологична схема за проектиране и изследване на менюта.

Development of Algorithms and Tools Environment for Design and Research of Menus.

Mitko M. Mitev

Abstract: The paper offers a classification of menus for management of programs. The problem for minimizing the cursor movement in strip menus is defined. A formalization by theory of graph is introduced. A priori algorithm for arrangement of the menu items is theoretically proved. A technological scheme for the design and research of menus is proposed.

1. Увод

Меню-техниката е идна от основните начини за управление и настройка на програми. В общия случай, по своята структура менютата могат да се класифицират като:

- *Точкови менюта*. Активна е една единствена позиция. Същата се стартира при определена, единствена по своята същност, програмна част с относително самостоятелна функционалност.
- *Едномерни, лентови менюта*. Определена последователност от позиции, активиращи предимно части на една и съща функционалност, позволяваща известна изборност при изпълнението на програмата. Например, това са отделни технологични етапи на разработка или многовариантни решения.
- *Двумерни, таблични менюта*. Те включват в себе си като основа лентово меню и определено множество от т.нар. *падащи* менюта. Приложението им предимно е свързано с провеждането на алтернативни варианти за изпълнение, като всеки вариант е разделен на отделни етапи или други, взаимно допълващи се възможности.
- *Многомерни таблични менюта*. Практически това е двумерно меню, като на определени позиции от падащите менюта има подменюта и т.н. Използването на тези менюта се налага при сложни по своята същност програмни системи.
- *Специални менюта*. Те се характеризират с използването на *некласически* средства за управление от типа на различни бързо сменящи се анимации, движещи обекти, цветови и звукови промени и т.н.

Независимо от вида на менюто, последователността за изпълнение на проектираната функционалност е свързана с физическо движение на курсора или преместване на фокуса. От ергономична гледна точка [1] може да се приеме, че сравнително добро решение за меню-техника на управление е това решение, което позволява бързо намиране на желаната или необходима позиция при минимално движение на курсора. В повечето случай това е необходимо , но не достатъчно условия за проектиране на потребителски ориентиран интерфейс.

Целта на настоящия доклад е проектирането и изследването на едномерни, лентови менюто, доколкото те са основа и за останалите видове менюта.

2. Формализация

Нека е зададена програмна система управлявана посредством едномерно, хоризонтално(лентово) меню. Всяка позиция се характеризира с дължината на включения текст и/или фигури. При преместването на курсора от една позиция към друга, физическото разстояние е пропорционално на общата ширина на обхванатите позиции.

При тази постановка формализацията може да бъде сведена до краен ориентиран граф *G(X,U,P,W,F)* с надстройка, както следва:

$X = \{x_i\}, i \in 1,N, N = |X|$ – множество на върховете на графа, интерпретирани като номерирани позиции на лентовото меню. Мощността на множеството се определя от броя на позициите в менюто.

$U = \{u_{i,j}\}, i,j \in 1,N$ - множество от дъгите на графа. Всяка дъга се интерпретира като логическа или функционално свързаност между отделните позиции, предизвикващи изпълнението на едни или други функции. Част от дъгите могат да бъдат регистрирани и след известен период от време, като прости физически премествания на курсора между позициите в процеса на търсене или комбиниране на възможности.

$P(x_i, u_{ij} x_j)$ - триместен предикат, въвеждащ инцидентност между двойката върхове $x_i, x_j \in X$ и дъгата $u_{i,j} \in U$. Инциденторът е *истина*, ако дъгата свързва двата върха или *лъжа* в противен случай.

$W = \{w_i\}, i \in 1,N, N = |W|$ – множество от тегла на върховете, където между $X \leftrightarrow W$ съществува едно-еднозначно съответствие. Теглото на върха се определя от физическия размер на позицията, т.е. от дължината на стринга.

$F = \{f_{i,j}\}, i,j \in 1,N$ - множество от тегловни коефициенти на дъгите. Между $U \leftrightarrow F$ съществува едно-еднозначно съответствие, където $f_{i,j} \in F$ е честотата на прелитане на курсора от позиция *i* към позиция *j*. Тази честота може да се определи като априорна или апостериорна.

Всяко преномериране на върховете представлява конкретно решение от множеството на решенията *R*. Нека е зададено начално решение $r_0 \in R$. Ако курсорът се намира на *i-та* позиция и бъде преместен върху позиция *j*, то теглото на $u_{i,j} \in U$ ще бъде $v_{i,j} \in V$, където теглото на дъгата зависи от общата дължина на *прелетените* върхове в права посока и може да бъде определено по следния начин:

$$(x_i x_{i+1} \in X) P(x_i, u_{i,i+1}, x_{i+1}) \rightarrow v_{i,i+1} = f_{i,i+1} w_i$$
$$(x_i x_{i+2} \in X) P(x_i, u_{i,i+2}, x_{i+2}) \rightarrow v_{i,i+2} = f_{i,i+2}(w_i + w_{i+1})$$

В общия случай

$$(x_i x_j \in X) P(x_i, u_{i,j}, x_j)(\mathrm{i} < \mathrm{j}) \rightarrow v_{i,j} = f_{i,j} \sum_{k=i}^{j-1} w_k \quad (1)$$

в права посока и

$$(x_i x_j \in X) P(x_i, u_{i,j}, x_j)(\mathrm{j} < \mathrm{i}) \rightarrow v_{i,j} = f_{i,j} \sum_{k=i}^{j-1} w_k \quad (2)$$

в обратна посока.

Следователно, за решението $r_0 \in R$ е в сила

$$(\forall x_i x_j \subset X) P(x_i, u_{i,j}, x_j) \rightarrow V_{r_0} = \vec{V} + \overleftarrow{V} \quad (3)$$

където

$\vec{V}$ – сума от теглата на дъгите по посока на нарастване на индекса.

$\overleftarrow{V}$ – сума от теглата на дъгите в обратна посока.

V_{r_0} е функция на решението $r_0 \in R$. Същата може да се приеме в качеството на целева функция, т.к. ако нейната стойност клони към минимум (локален или глобален), то определеното решение е свързано с минимално сумарно разстояние между позициите на менюто при условие, че процесът на търсене е в двете посоки.

Алтернативно може да се дефинира решение на еднопосочно търсене като се започва винаги от началото на менюто. Това е често срещана стратегия за навигация по позициите на менюто.

За определяне на екстремална стойност на V_{r_0} е необходимо предварително да са известни стойностите на множеството *U* и съответно *F*.

Обикновено , при проектирането на дадено меню тези стойности не са известни. Следователно, това налага проектирането да се извърши при априорни условия.

3. Априорен алгоритъм

Априорните алгоритми се основават на предположението, че всички позиции от менюто са силно свързани. Тази постановка съответствува на програмни системи с активна потребителска комуникация. При това, колкото потребителският интерфейс е по-интензивен, толкова резултатите ще бъдат по-близки до идеализираната постановка.

При априорните алгоритми не се отчита честотното натоварване на преходите между позициите на менюто, т.к. се счита, че то е достатъчно високо и за отделните преходи е приблизително еднакво.

Априорните решения зависят от начина по които се избират отделните позиции. Възможни са два подхода:

- *двупосочно избиране*. Този начин е характерен за управление на добре познати програмни системи с активен потребителски интерфейс,
- *еднопосочно с връщане*. Търсенето на необходимата позиция започва винаги от началото на менюто. Това е характерно за слабо познати системи, където търсенето е отляво надясно (или отдолу нагоре) до намиране на необходимата позиция или в случаите на падащо подменю с физическо търсене отгоре надолу.

Съгласно приетата постановка за априорен алгоритъм, то

$$(\forall u_{i,j} \in U) \rightarrow f_{i,j} = const. \quad (4)$$

Компонентите от (3) изразени посредством (1) и (2) мога да се представят по следния начин:

$$\vec{V} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{k=i}^{j-1} w_k \quad (5)$$

и аналогично в обратна посока

$$\overleftarrow{V} = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \sum_{k=i}^{j-1} w_k \quad (6)$$

Изрази (4) и (5) отчитат дължината да дъгата в зависимост от “прелетените” върхове. За да се определи априорното разполагане на позициите при двупосочно търсене е необходима оценка на броя на дъгите прелитащи всеки връх.

Сумирано в двете посоки за произволно избран връх *k*

$$\Phi(k) = 2(k-1)(N-k) \quad (7)$$

където изразът притежава максимална стойност при $k = (N+1)/2$ с точност до цяло число.

Следователно, може да се докаже, че минималната стойност на V_{r_0} се получава при априорно подреждане на върховете, според техните тегла със стойности обратни на функцията $f(k)$, по правилото:

$$(\forall x_i x_{i+1} \in X)\left(i+1 \leq \frac{N+1}{2}\right) \rightarrow (w_i \geq w_{i+1}) \;\&$$

$$(\forall x_i x_{i+1} \in X)\left(i+1 \geq \frac{N+1}{2}\right) \rightarrow (w_i \leq w_{i+1}) \quad (8)$$

при условие, че сумата от теглата на върховете в лявата част на максимума на $f(k)$ е приблизително равна на дясната сума т.е. равномерно разпределение на натрупващите се суми. Разликата на двете суми не трябва да превишава теглото на следващия връх за включване в съответната сума.

На основание (8) може да се предложи априорен алгоритъм определящ *първо приближение* за разполагане позициите на едномерно лентово меню. Алгоритъмът се основава на сортиране на върховете във възходящ ред на теглата и последващо тяхно равномерно разпределение около средната точка. Ако първото приближение е решението $r_0 \subset R$ определящо обща сума от теглата на върховете, съответно V_{r_0}, то нека са зададени два произволни върха $x_i x_j \in X$, където $i \neq j$ и е определена операцията $\alpha(x_i x_j)$ разместваща местата на двата върха. Прилагането на операцията води до ново решение $r_1 \in R$ и съответно V_{r_1}.

Ако $V_{r_1} < V_{r_0}$, то новото решение приближава целевата функция до нейния глобален или локален минимум.

Следователно, основният проблем за намиране на локален или глобален екстремум се свежда до определяне на последователността от двойки върхове и критерий за достатъчност на минимизацията в случай на незатихващ процес.

В случая може да се използва добре известния алгоритъм за сортиране посредством циклично обхождане на всички двойки върхове. Въпреки, че алгоритъмът е в класа на комбинаторните, същият може да се използва за изследване и оптимизиране на менюта.

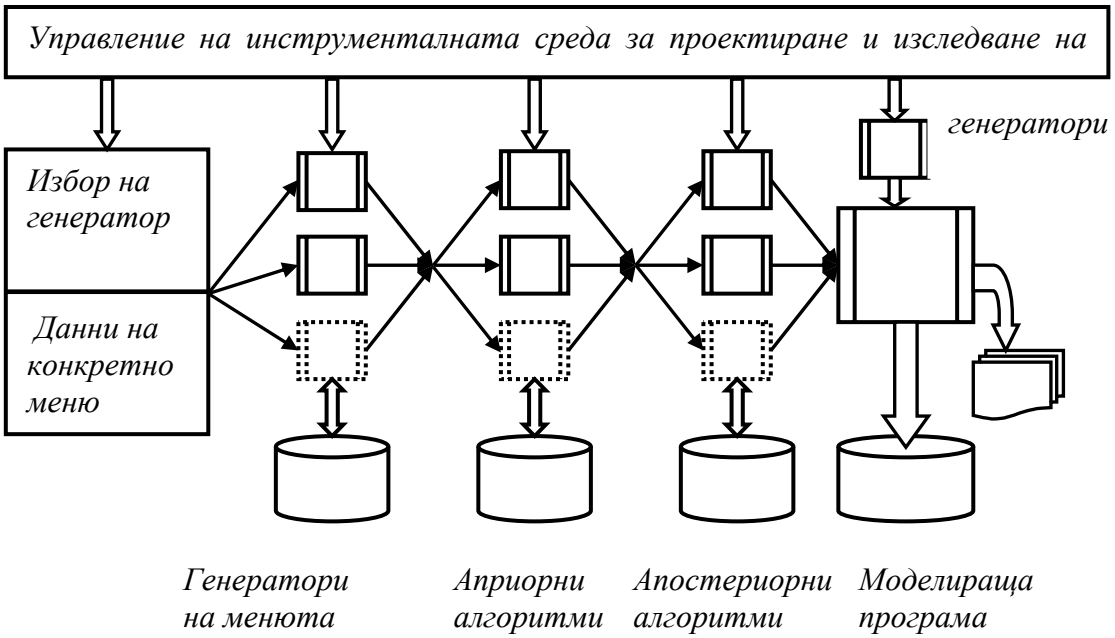
Предпоставка за това е сравнително минималното количество позиции в управляващите менюта на програмните системи. Алгоритъмът определя и сходящ процес на сортиране

4. Обобщена технологична схема.

Основните режими реализирани от предложената технологична схема (фиг.1.) са следните:

- генериране на абстрактни менюта със случайни параметри и изпълнение на произволна последователност от включените априорни и при определени условия апостериорни алгоритми,
- въвеждане на конкретни решения на действително проектирани и изпълнени менюта от реално действуващи програми и прилагането предимно на апостериорни алгоритми за структуриране и
- генериране на случайни последователности от позиции на менюто и натрупване на статистическа информация за експлоатационните характеристики на проектираното меню.

Схемата позволява съхраняване на междинните и крайни резултати, прекъсване на процеса на проектиране или изследване със следващо продължаване от мястото на прекъсване.



Фиг.1.
Обобщена технологична структура на инструментална среда за проектиране и изследване на менюта.

Инструменталната среда изпълнява следните технологични етапи:
- избор на генератор на случайни числа (различен закон на разпределение),
- генериране на хоризонтални (лентови) с определен брой позиции, ширина на всяка позиция (брой символи), генериране на преходи между позициите, генериране честота на преходите или
- въвеждане на конкретни или експертни данни за реализирано проектно решение,
- избор на алгоритъм (алгоритми) за априорно подреждане на генерираните позиции при стратегия на блуждаещо търсене в две посоки, търсене само през началото на менюто, търсене от случайна позиция и др.
- избор на алгоритъм (алгоритми) за апостериорно преподреждане позициите на менюто в зависимост от целева функция, ограничителни условия и използван метод за преподреждане.
- генериране на блуждаещо търсене и отчитане на преходите,
- всяко проектно решение се записва в база от данни и може да послужи като начално за следващия етап на преподреждане или за сравнение с предходни решения.

5. Заключение

Въведена е формализация на проблемите свързани с проектирането и изследването на предимно на лентови менюта. Представена е теоретична обосновка за разработването на априорни и/или апостериорни алгоритми за подреждане на позициите в менюта при минимизиране на сумите от дължините между тях. Разработена е технологична схема за генериране , изследване и симулиране на проектни решения за лентови менюта.

С помощта на инструменталната среда могат да бъдат генерирани различни конструктивни параметри на най-често използваните менюта на базата на генератори на случайни числа с различен закон на разпределение. Първоначалните случайни решения

могат да бъдат обработвани с различни по характер априорни алгоритми. Всяко решение може да се комбинира със следващ апостериорен алгоритъм. Първоначалните решения,както и получените априорни и апостериорни решения се съхраняват в база от данни. По отношение на тях е възможно с помощта на генератори на случайни числа да се симулират конкретни стратегии на търсене, да се отчитат дължините на преходите и съответно да се получи крайна, модел на оценка на всяко от проектните решения.

Получените алгоритми и техните програмни решения, както и предложената технологична схема могат да се използват самостоятелно или отделни техни компоненти да бъдат включвани в разработвани или съществуващи среди за проектиране на програмни системи.

Литература

[1]. Лазарова, В. Потребителска ефективност на информационните системи в интернет**:** Методологични проблеми при потребителски ориентираното проектиране. Сборник доклади "Бизнес информатика", УНСС, София, 2007, с. 191-233.
http://www.unwe.acad.bg/research/br12/6.pdf

За контакти:
доц. д-р инж. Митко М. Митев
катедра „Компютърни науки и технологии"
Технически университет – Варна
E-mail:mitevmm@abv.bg
Рецензент: доц. д-р инж. Е. Рачева, ТУ – Варна

A Goniometric Rocket Velocity Computation Algorithm

Lâtchezar I. Georgiev

Abstract: This paper proposes a simple algorithm for computation of the velocity of a rocket from a single still film frame where the oblique shock-wave cone is visible. The algorithm consists of the following steps:
1. Measurement of the visible oblique shock-wave cone angle, the rocket diameter and length.
2. Computation of the real oblique shock-wave cone angle based on these values via a simple formula.
3. Computation of the rocket cone (or cone approximation) angle from the rocket drawing dimensions.
4. Computation of the Mach number from the Taylor-Maccoll equation of the supersonic conical flow.
5. Computation of the rocket velocity from the Mach number and the rocket's atmospheric altitude.

Applying the algorithm to the *Saturn V* rocket gives a more exact real velocity at *S-IC* (stage 1) staging time.

Гониометричен алгоритъм за изчисляване на скоростта на ракета

Лъчезар Ил. Георгиев

Резюме: Предлага се прост алгоритъм за изчисляване на скоростта на ракета от единичен филмов кадър, където е видим конусът на кóсата ударна вълна. Алгоритъмът се състои от следните стъпки:
1. Измерване на ъгъла на видимия конус на кóсата ударна вълна, диаметъра и дължината на ракетата.
2. Изчисляване реалния ъгъл на конуса на кóсата ударна вълна по тези стойности с проста формула.
3. Изчисляване на ъгъла на конуса (или приближен конус) на ракетата по размерите от чертежа ѝ.
4. Изчисляване на числото на Мах от уравнението на свръхзвуков коничен поток на Тейлър-Маккол.
5. Изчисляване на скоростта на ракетата по числото на Мах и атмосферната ѝ височина.

Прилагането на алгоритъма към ракета *Saturn V* дава по-точна реална скорост при отделяне на *S-IC*.

1. Introduction

Abstract: Above a certain supersonic Mach number, a conical shock wave is attached to the aircraft cone's apex. The wave's cone angle is often used to determine the Mach number [1] (in wind tun¬nels to test airfoils, engines, etc.). Pokrovsky and Popov first did this to a rocket in open air [2], analysing NASA's Apollo 11 launch clip [3] by this and two other meth¬ods. They found the rocket's real velocity at S-IC staging time to be much less than the nomi¬nal, meaning that too little load was launched so no Moon landing occurred. But they made ap¬proximations leading to impre¬cise results. This paper aims to fully define the rocket ve¬locity computation formulas and algorithm to get a more accurate value of the rocket speed.

2. Real oblique shock-wave cone angle

If the rocket's oblique shock-wave cone is filmed, the film fixes only its projection. Measuring the projected cone angle, rocket length and diameter on the photo, and knowing the real rocket length and diameter, the real shock-wave cone angle can be derived.

Fig. 1 shows a real cone projected on the film plane with its base seen as an ellipse. The projection is symmetrical, i.e. it does not matter if the cone is visible from the apex or from the base. The unknown real cone angle θ can be expressed as a function of the projected (visible) cone angle β and the visible rocket length-shorting cos α. Such an expression (formula) shall be derived. The equation of the tangents of slope m to the ellipse is [4]:

$$y = mx \pm \sqrt{m^2a^2 + b^2} \tag{1}$$

For $x = 0$ and $m = \frac{c}{a} = \text{ctg}\ \beta$, $\quad c = y = \sqrt{a^2 ctg^2 \beta + b^2}$ (2)

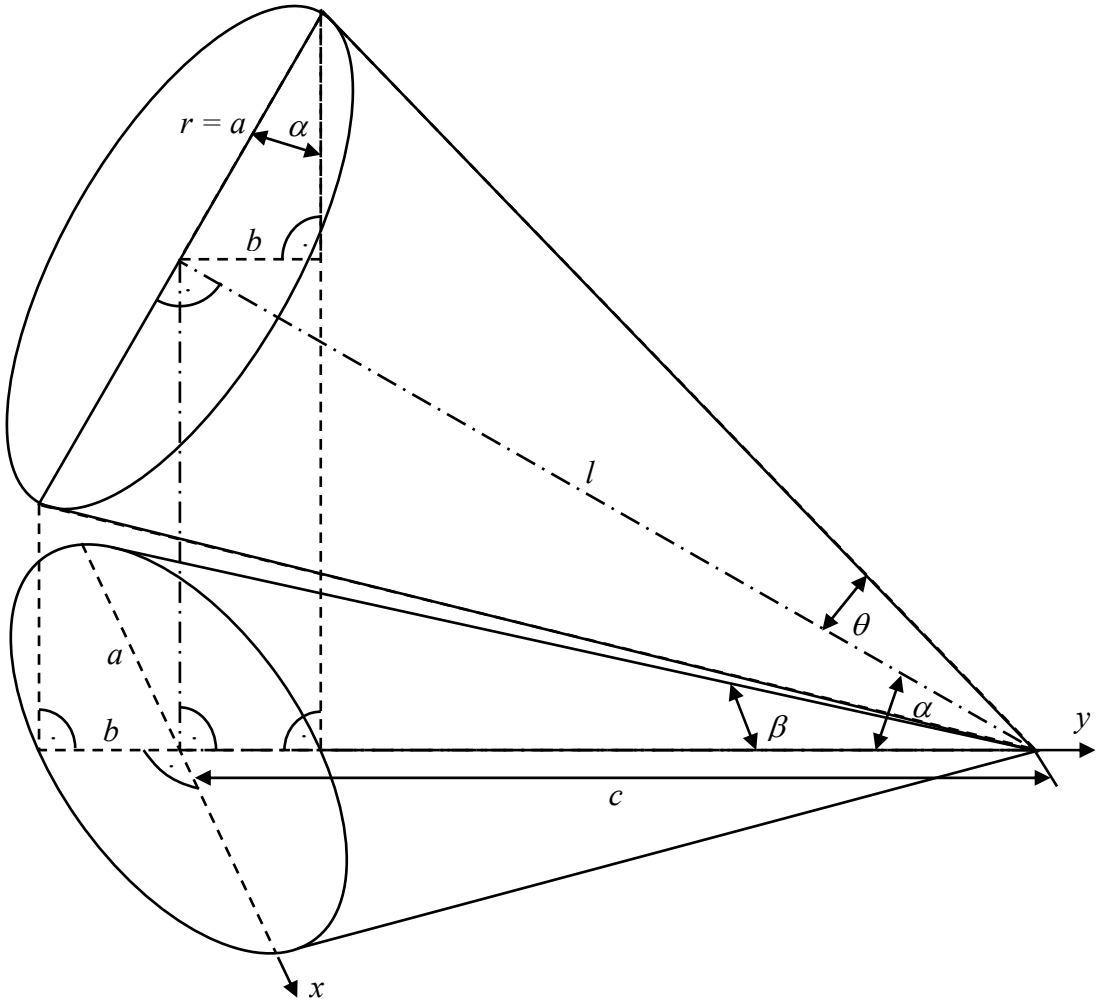


Fig. 1: A real cone (top) and its projection on the 0*xy* plane (bottom)

θ, b, and l can be derived from fig. 1 as

$$\begin{vmatrix} \theta = arctg\dfrac{a}{l} \\ b = a.\sin\alpha \\ l = \dfrac{c}{\cos\alpha} \end{vmatrix} \tag{3}$$

Substituting l in (3) gives for θ

$$\theta = arctg\frac{a.\cos\alpha}{c} \tag{4}$$

Substituting c from (2) in (4) gives

$$\theta = arctg\frac{a.\cos\alpha}{\sqrt{a^2 ctg^2\beta + a^2\sin^2\alpha}} = arctg\frac{a.\cos\alpha}{\sqrt{a^2(ctg^2\beta + \sin^2\alpha)}} = arctg\frac{\cos\alpha}{\sqrt{ctg^2\beta + \sin^2\alpha}}$$

$$\theta = arctg\frac{\cos\alpha}{\sqrt{ctg^2\beta + 1 - \cos^2\alpha}} \tag{5}$$

Equation (5) gives the desired relation but is complex. Squaring both sides helps simplify it:

$$tg^2\theta = \frac{\cos^2\alpha}{ctg^2\beta + 1 - \cos^2\alpha} \Rightarrow \cos^2\alpha = \frac{tg^2\theta}{tg^2\beta} + tg^2\theta - tg^2\theta\cdot\cos^2\alpha \Rightarrow \cos^2\alpha.(1 + tg^2\theta) = \frac{tg^2\theta}{tg^2\beta} + tg^2\theta$$

$$\cos^2\alpha\cdot tg^2\beta.(1 + tg^2\theta) = tg^2\theta.(1 + tg^2\beta) \Rightarrow \frac{\cos^2\alpha\cdot tg^2\beta}{\cos^2\theta} = \frac{tg^2\theta}{\cos^2\beta} = \frac{\sin^2\theta}{\cos^2\theta.\cos^2\beta}$$

$$\cos^2\alpha.\cos^2\beta\frac{\sin^2\beta}{\cos^2\beta}=\sin^2\theta\Rightarrow\sin\theta=\sin\beta.\cos\alpha\Rightarrow\theta=\arcsin(\sin\beta.\cos\alpha) \quad (6)$$

Now, $\cos\alpha$ can be expressed as $$\cos\alpha=\frac{\frac{l}{d}}{\frac{L}{D}}=\frac{l\cdot D}{d\cdot L} \quad (7)$$

where l, d – rocket length and diameter on the photo; L, D – real rocket length and diameter.

Since (7) is a division of two length-to-diameter ratios, as long as each pair (l and d, L and D) is in the same units, such units can be any. For example, both l and d can be in pixels, millimetres, points, etc., whereas both L and D can be in metres, feet, inches, etc. Substituting (7) in (6) gives the practical formula for obtaining the real shock-wave cone angle θ from the angle β on the photo:

$$\theta=\arcsin\left(\sin\beta\times\frac{l\times D}{d\times L}\right) \quad (8)$$

3. Rocket cone angle

The Mach number depends also on the rocket cone angle, albeit, as shall be shown, to a much lesser degree (so a cone would approximate a complex-shaped rocket well enough). This angle is easily computed from the known rocket dimensions, as follows:

$$\sigma=arctg\frac{D_2-D_1}{2\times L_{1-2}} \quad (9)$$

where D_1 – smaller diameter of a cone frustum (can be 0), D_2 – larger diameter, L_{1-2} – the length between D_1 and D_2. A frustum serves as a "benchmark" for the more complex rocket shape.

4. Mach number

If the real shock-wave cone angle and the rocket cone angle are known, the Mach number can be computed by solving the Taylor-Maccoll ordinary differential equation. It has no closed-form solutions and must be solved numerically; one such method is proposed in [5]. There exist accurate nomograms and computer programmes for such solutions.

5. Rocket velocity

With known Mach number M, rocket velocity is the product of it and the sound speed C_s [6]:

$$C_s=\left(\frac{\gamma\cdot R^*\cdot T_M}{M_0}\right)^{1/2}$$

where γ – the ratio of specific heat of air at constant pressure to that at constant volume (it is set to be 1.4 exact), R^* – the gas constant = 8,314.472 N·m / (kmol·K), T_M – the absolute temperature [K] = 273.15 + t°C, M_0 – the mean molecular weight of air = 28.96442 kg/kmol [7][9].

With a known t°C, the sound speed is [10] $C_s=20.047\sqrt{(273.15+t°C)}$ (10)

The temperature at a given altitude varies with the atmospheric model. For simplicity, the rocket can be assumed to be in the stratopause (altitude: 47–51 km) where a local maximum of the temperature (and thus sound speed) exists that is −2.5°C [8][11] in both the International and U.S. Standard Atmospheres and in GOST-4401. There, the rocket velocity is $v_{SP}=329.8\times M$ (11)

6. Velocity computation algorithm steps

1. Measure the projected oblique shock-wave cone angle β, the rocket length l, and its diameter d.
2. Compute its real shock-wave cone angle (8) and the rocket cone angle (9).
3. Find the Mach number by software solving the Taylor-Maccoll equation or a nomogram (fig. 4).
4. Compute the rocket velocity (11).

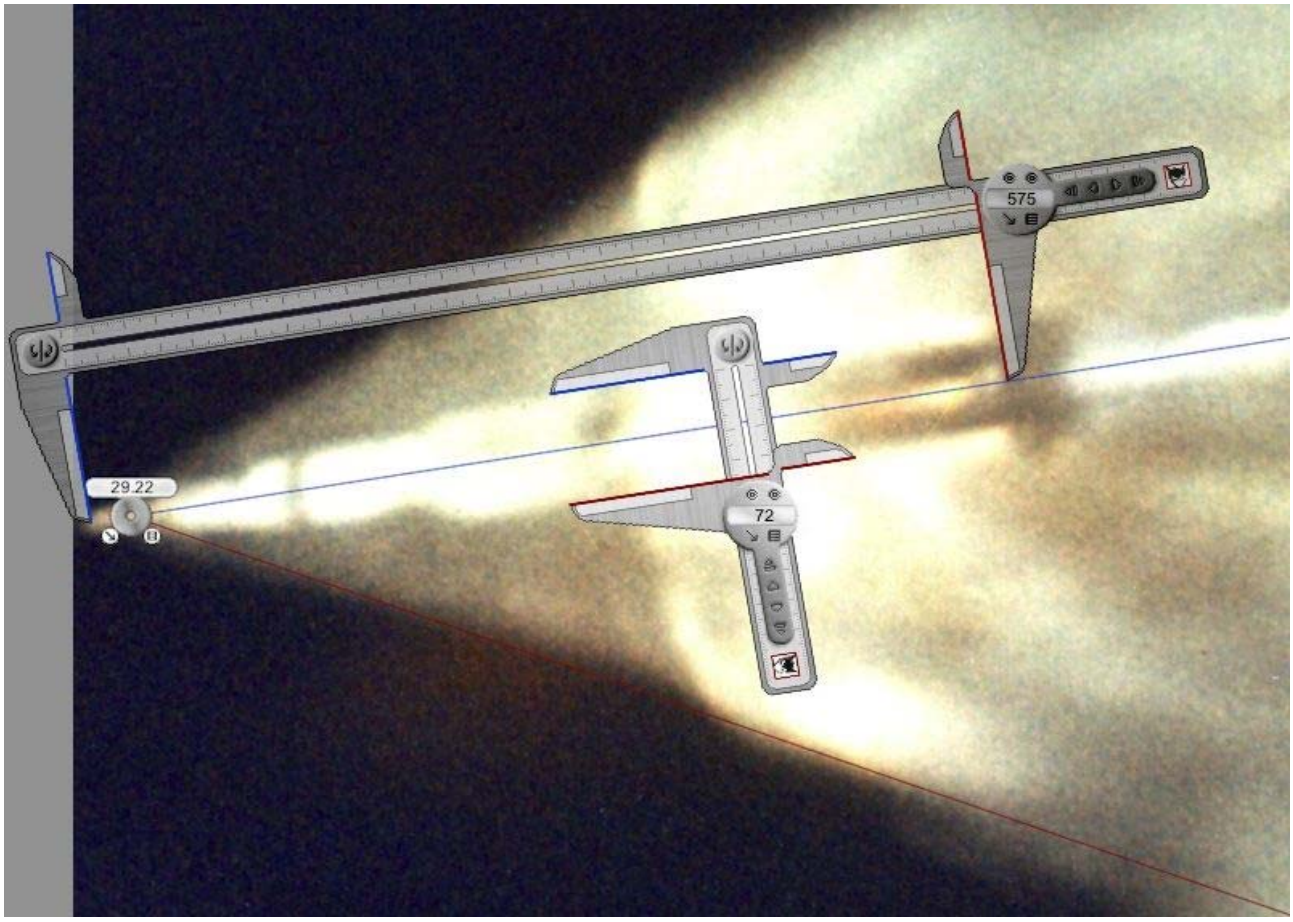


Fig. 2: NASA photo S69-39957 [12] showing the *S-IC* staging of the *Saturn V* rocket, with values of β, d, l

7. Applying the algorithm

Now that the algorithm is defined, it shall be applied to the real *Saturn V* rocket. Substituting the measured values on fig. 2 and real rocket dimensions on fig. 3 in (8), the real oblique shock-wave cone angle

$$\theta = \arcsin\left(\sin(29.22°) \times \frac{575 \times 33}{72 \times 363}\right) = 20.757°$$

As seen on fig. 3, the rocket has a complex shape, but representing its *S-IVB* stage as a cone frustum can approximate it, as its angle is small and does not affect the Mach number significantly. Substituting the dimensions on fig. 3b in (9), the rocket cone angle

$$\sigma = arctg\frac{33 - 21.6}{2 \times 59} = 5.518°$$

Entering the above values of θ as θ_S and σ as θ_C in the *Supersonic Cone* programme, it gives a Mach number of M_∞ = 2.863 (fig. 4a). The red line cross point for these θ and σ values lies on the same Mach number curve on the NACA nomogram (fig. 4b). As the latter shows, the exact rocket cone angle value only

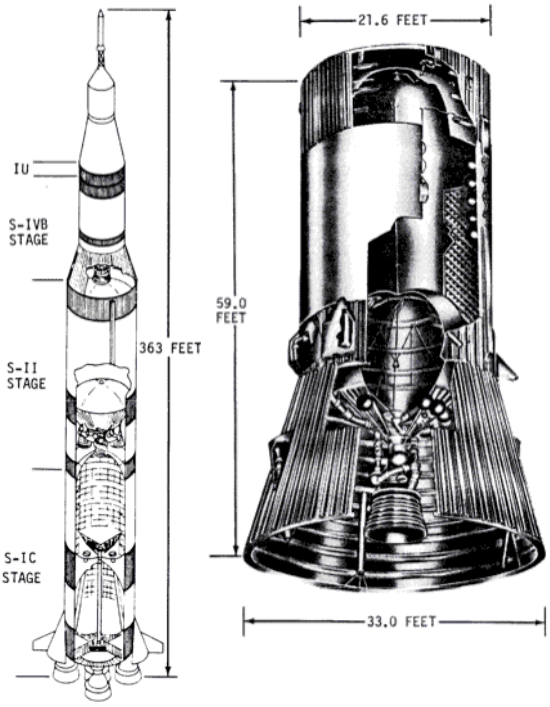
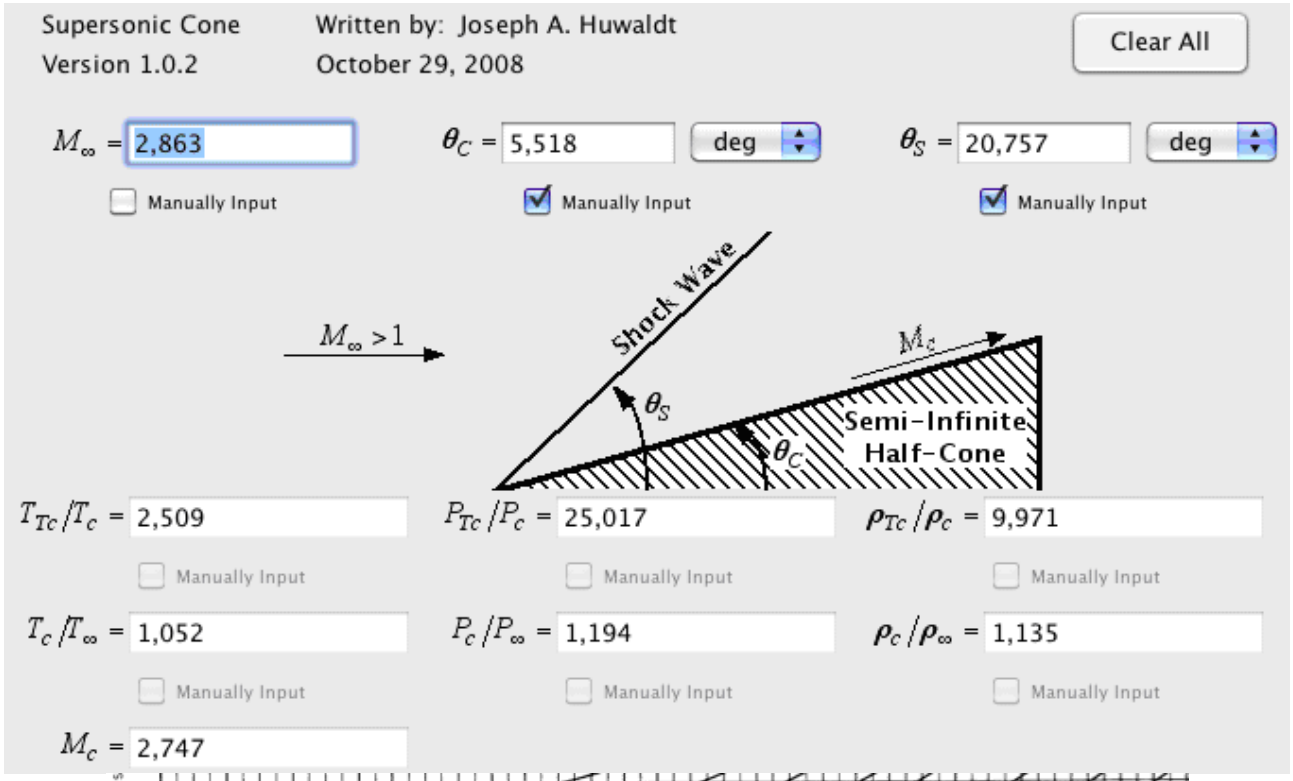


Fig. 3: Size of a) *Saturn V* [13], b) *S-IVB* [14]



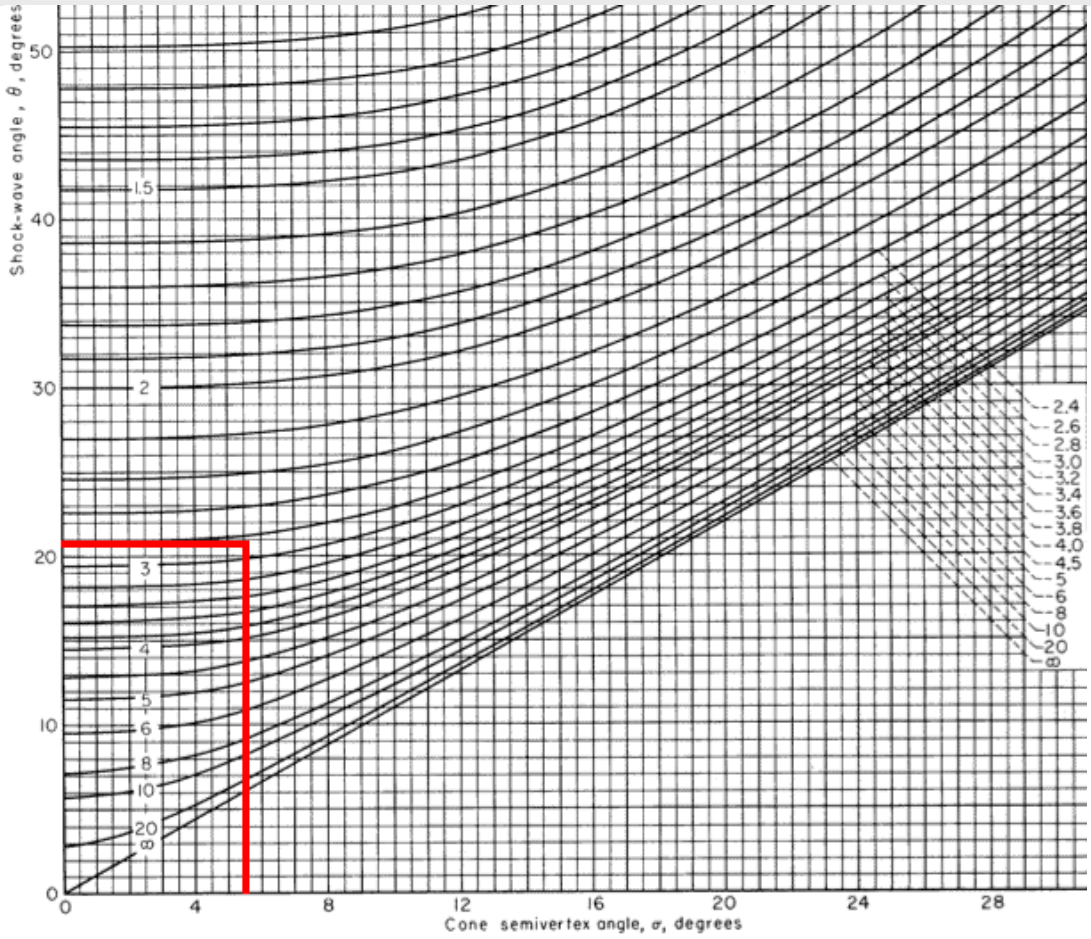


Fig. 4: The *Saturn V* Mach number in a) the *Supersonic Cone* programme [15], b) NACA nomogram [16] (the Mach number of each curve is shown on the right; the red lines connect the *Saturn V* θ and σ angles).

slightly affects the Mach number, so the approximation of this cone was justified.

The Launch-escape motor on top of the rocket is clearly visible on photo [12] (fig. 2). Therefore, the camera resolution and thus the maximum error of each measurement is less than its diameter of 26" (fig. 5). So, the total maximal error of measurement (and thus, of M) is

$$\frac{26"}{D}+\frac{2\times 26"}{L}=\frac{26"}{(12.33)"}+\frac{2\times 26"}{(12.363)"}=7.76\%$$

(Angle measurement error equals the length measurement error, so it's doubled.) Adding the stratopause temperature deviation $(290/271)^{½} = 3.4\%$ [18](10) rises it to ≈ 11%. With $M = 2.863$, the real air-fixed *Apollo 11 Saturn V* velocity at *S-IC* staging time (11)

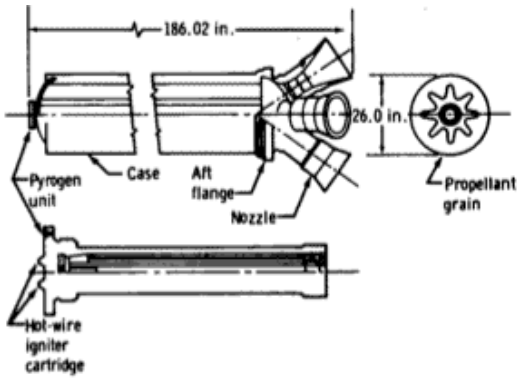


Fig. 5: *Apollo* Launch-escape motor [17]

$$v \le v_{SP} = 329.8\times 2.863 = 944 \text{ m/s} \pm 11\%$$

The 40 m/s contrary mesospheric easterlies at 28.6º in the summer hemisphere [19] reduce the real Earth-fixed velocity to about 900 m/s. Compare it to NASA's declared velocities of 9,064.5 − 1,340.67 = 7,723.83 ft/s = 2,354.2 m/s [20] and 2,402.7 m/s (2,397 nominal) [21].

8. Conclusion

A formula to get the oblique shock-wave cone angle from the projected cone angle has been derived, and a velocity computation algorithm defined. To find open-air rocket's velocity, only a photo of its oblique shock-wave cone is needed, if the well-known supersonic conical flow theory (used mostly for wind tunnels) is applied. The computed more exact real *Saturn V* velocity of about 900 m/s (2⅔ times less than the NASA's values) verifies the Pokrovsky's and Popov's findings [2].

9. Literature

[1]. Marks L., Marks' Standard book for mechanical engineers, McGraw-Hill, 2007, p. 11-74
[2]. Расследования: Луна, Сверхновый проект, 2007–2010 гг.
[3]. A view of the first stage separation of Apollo 11, NASA, 1969
[4]. Korn G., T. Korn, Mathematical handbook for scientists and engineers, Dover, 2000, p. 46
[5]. Anderson J., Modern compressible flow, McGraw-Hill, 1990, p. 301
[6]. U.S. standard atmosphere, NASA, 1976, p. 18
[7]. Ibid., p. 9, 44, 64, 178
[8]. ГОСТ-4401-81, Атмосфера стандартная, Издательство стандартов, 2004 г., стр. 174
[9]. Apollo imagery – S69-39957, NASA, 1969
[10]. Saturn V flight manual, NASA, 1968, p. 1-6
[11]. Ibid., p. 6-2
[12]. Huwaldt J., Supersonic cone program, 2008
[13]. Report 1135 – Equations, tables, and charts for compressible flow, NACA, 1953, p. 48
[14]. Launch escape subsystem, NASA, 1973, p. 5
[15]. Chandra H. et al, A Rayleigh lidar study of the atmospheric temperature structure, 2005, p. 291
[16]. Meridional cross-section of the atmosphere, Encyclopædia Bitannica, 2008
[17]. Apollo 11 press kit, NASA, 1969, p. 21
[18]. Saturn V launch vehicle flight evaluation report-AS-506, Apollo 11, NASA, 1969, p. 4-7

For contacts:
Ass. Prof. Eng. Lutchezar I. Georgiev
Computer Science and Engineering Dpt.
Technical University of Varna
E-mail: lucho@gawab.com

Рецензент: доц. д-р П. Антонов, ТУ – Варна

Impruving Security in A Particular Class of Images and Documents

George B. Varbanov

Abstract: In this article we present a modified algorithm for protection of electronic documents using watermarks. The idea is to assign to each document a unique identification number of the server, specifying users' rights for downloading. The proposed method, using watermarking allows certain corrections of errors in the watermark detection as well as enhanced opportunities for building a message with a greater length and low rates of modulation

Key words: watermarks, error correction code

Подобряване на сигурноста на определен клас електрони документи и изображения

Георги Б. Върбанов

Резюме:В тази статия представяме модифициран алгоритъм за защита на електронни документи чрез воден знак. Реализирана е идея за присвояване на уникален идентификационен номер на всеки документ от специален сървър делегиращ правата на отделните потребители при изтегляне на съответния документ. Предложения метод използва за метод за коригиране на грешките при детекция на водния знак и дава възможност за получаване на вграждане на воден знак с по голяма дължина и и достигане на по-ниски нива на модулация.

Ключови думи: воден знак, корекция на грешки

1. Introduction:

In this article we present a modified algorithm for protection of electronic documents using watermarks. The idea is to assign to each document a unique identification number of the server, specifying users' rights for downloading. The proposed method, using watermarking allows certain corrections of errors in the watermark detection as well as enhanced opportunities for building a message with a greater length and low rates of modulation.

2. Problem definition:

In any modern system for e-learning there exist a lot of problems, linked with copyrights, when we need to read or download multimedia tutorials, electronic textbooks, manuals. User Registration is done through the assignment of a username and password or ID.

When the number of users in an e-learning environment is great there comes the problem with the number of protected users who can simultaneously access documents marked with a special character in order to ensure the copyright of the drafters of these resources. Usually this is done by blocking the database for download from non registered users, but the problem with copyright still remains. One possible solution is to use a print server placed at the gate, assigning a specific marking of the electronic content of the document for downloads. In such case a watermark may be used for ID with a length of one byte limit the number of simultaneous users of the marked documents to 255. A longer protection watermark of 10, 12 or 16 bits will allow a significant increase of the number of registered users.

On the other hand the requirements for creating robust algorithms (stable) watermarks contradict with the length of the embedded watermark. In order to solve this problem we may use specific communications techniques.

One possible solution is to use BCH codes (Turbo codes)[5],[6],[7], distribution spectrum and LPDC codes. Such codes are widely used in up to date communications. We follow the consideration that LDPC codes [2] have greater opportunities to recover the errors of communications, i.e. of the embedded watermark (IDS), besides they are using comparatively weaker signals, which leads to less noise, introduced in the media as a result of the embedded watermark -This is shown in Fig.1

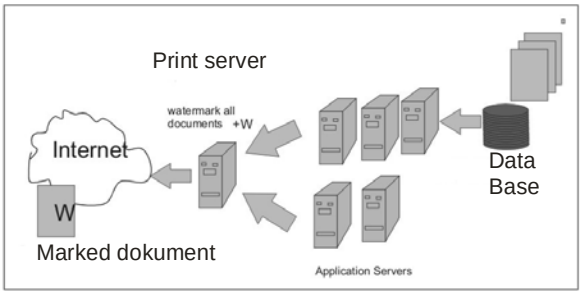


Fig.1.

3. Problem solution

Figure 2 shows the block diagram of a communication system with error correction facility. Compared to the standard system of watermark data protection we are now adding two additional blocks, one is at the encoding side and a second one for data decoding. The reduction of errors when transmitting and encoding data significantly reduces the redundancy of data and increases the speed of communication.

The "Encoder block" at the transmitter side takes the message sequence as an input and adds check bits to it.

The "Decoder block" uses the redundancy introduced by the encoder to detect and correct errors that occurred during transmission.

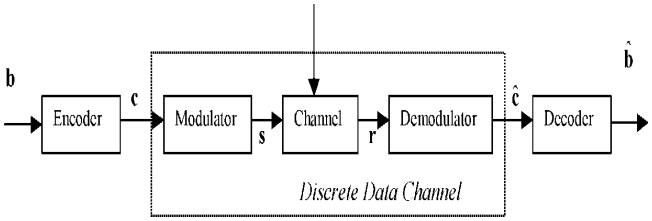


Figure2. Communication system with forward error correction facility

The binary sequence B is the watermarked message. It passes through the encoder at the transmission side, the output of which represents the coded binary sequence C. The C vector is mapped via the chosen watermarking techniques to the modulated signal S, which is transmitted through the watermarking channel.

It is referred to as an image, which contains all concomitant distortions due to the embedded watermark.

The watermarked image may suffer from various intentional and unintentional attacks. The extracted noise containing watermark analog sequence R is fed to the demodulator block, which will attempt to recover the original code sequence from the possibly distorted channel output sequence R.

The Demodulator may be constructed in 2 ways [4],[8],[12]:

-"Hard option/decision", allowing only 0 and 1 values of the output signal (0 or 1 decision)

-"Soft option/decision", allowing multiple levels of the output signal

In this article the soft option is being used, which provides a multi-value indication, which can be interpreted as a measure of how close the received signal is to the 0 or 1 decision. In other words, it is a confidence measure.

is realized by quantizing the demodulated signal range to more than two, typically 8, levels. These "soft decision" outputs are used by the decoder for specifying the values of the transmitted bit sequence.

This algorithm uses the $LDPC$ coding method, based on the soft decision decoding. The applied watermark embedding techniques is based on the algorithm given in [] Koch and Zhao

3.1. Watermarking Algorithm

3.1.1. Watermark Insertion

Step1 is the generation of binary converse message data sequence b of length T_k.

Step2 is the error correction encoding stage. Partition the message sequence b of length T_n into a number of k-length blocks and encode them with an T_k (n,k) block code $LDPC$. The coded sequence C is of length T_n.

Step3 is the repetition coding stage. Every code bit is repeated ch times. That is, a repetition-coded vector C_C of length $T_n * ch = T_C$ is generated.

Step4 is spread spectrum modulation stage. Generate a $T_k \pm 1$ pseudo-random sequence pc of length T_C using key K_i and multiply B_C with T_k Pc to obtain W_C (This represents personal IDs and data).

Step5 Apply Multilevel DWT to second level and select LL sub band to make next operation to distribution non-overlaying blocks on domain into 8x8 blocs how is shown in figure 3.2

Step6 is the generation of the watermark mask in the transform domain as follows:

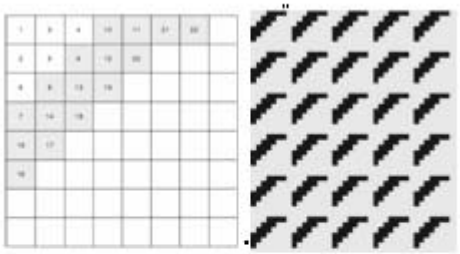


Fig. 3.a. Selection of Block DCT Coefficients, Fig. 3.b) Marked region in 8x8 DCT block and Figure3b) $N \times M$ blocks of size 8x8 in the DCT domain

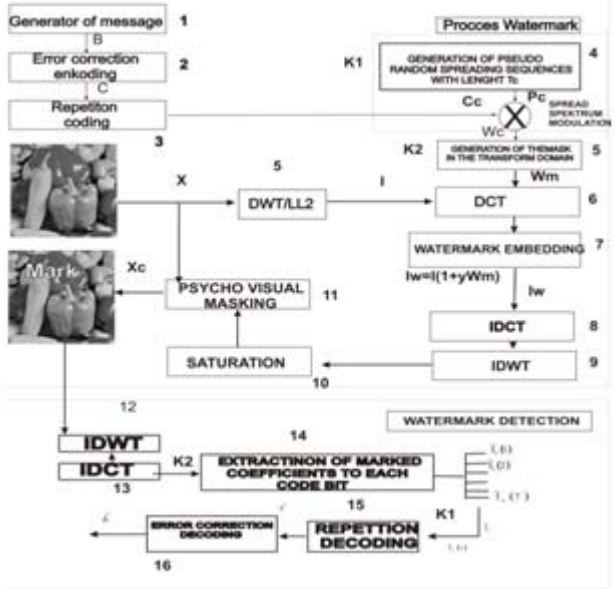


Fig. 3.2. Flow diagram of the applied watermarking algorithm

Distribute W_C randomly to the block DCT domain watermark bands, as illustrated in Figure3, using key $K2$. The regions left over, are filled with zeros, and consequently the watermark mask W_m of size $N \times N$ in DCT domain is generated.

The common practice for block DCT marking is to skip a few , say 5 or 6 coefficients (the upper left corner coefficients correspond to low frequencies - this is shown in Figure (3) in the zigzag scan pattern and to mark the subsequent coefficients band pass coefficients. The frequency sub range in each 8x8 DCT block is like a trade-off between perception of transparency and robustness In each $N \times M$ block one and the same 16 frequency band pass coefficients are selected and evenly distributed over the whole transformed domain LL of the wavelet distribution. The distribution is performed in a zigzag pattern in ascending order.

The deployment of the marked DCT coefficients over the whole image is illustrated in the $N \times M$ block image in Fig. 3.1.b.

The black regions represent the marked DCT coefficients. In each block, 16 coefficients are watermarked, which are shown with black patterns.

Any element of W_C can mark any of the 16 coefficients of any block by a random permutation operation.

Step7 is the embedding watermark process in the transformed domain.

We embed the watermark W_m to I m using an additive algorithm:

$$I_m = I\left(1+\gamma W_M\right), |\gamma| < 1 \tag{3.1}$$

The γ value is chosen to tradeoff between robustness and perceptibility. The higher γ, the more robust is the watermark, but at the same time it contradicts to visibility, i.e. distortions in the resulting image will more significant and more perceptible.

Step8- is the inverse transformation to the spatial domain. The inverse $IDCT$ is being used.

Step9 produces the $IDWT$

Step10. is saturation. Assign to value 255 the bigger coefficients and to 0 – coefficients, which are smaller than 0.

Step11 is applying a psycho-visual masking in the spatial domain and measuring the perceptual quality. Now we have the preliminary watermarked image X_w in the spatial domain, we can optionally pass X_w through visual masking using "Human Visual System" (HVS) models.

They are used in order to adaptively embed the watermark, in strongly perceptually less significant regions and weakly in perceptually significant regions. These HVS measures of interest are the contrast, frequency, luminance sensitivity, edges and textured areas [9],[10,[11].

Based on these measures, a $N \times N$ psycho-visual mask matrix M is generated, the elements of which are in the range $[0,1]$. The indices of M , where the values are higher, indicate the pixel locations where watermark can be embedded with (3.2) higher strength.

The marked image is generated from $\hat{X}_w$ in the following way:

$$X_w = M\hat{X}_w + (1-M)X \tag{3.2}$$

A straightforward first-hand perceptual quality evaluation method is to watch for watermarking artifacts. A rough measure of image fidelity is given by the peak signal to noise ratio ($PSNR$) and it is defined as:

$$PSNR = 10\log\frac{X_{peak}^2}{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[X(i,j)-X_W(i,j)\right]^2} \tag{3.3}$$

where X_{peak} is the maximum pixel luminance value in the image.

$PSNR$ Values higher than 38 dB are required for an acceptable image quality.

Another criterion for evaluating the result of the embedded mark in the document is the ratio watermark/document (WDR) and it is defined as:

$$WDR = 10\log\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left[X(i,j)-X_w(i,j)\right]^2}{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(X(i,j)\right)^2} \tag{3.4}$$

It is a measure of the ratio of watermark energy to the energy of host image. There are also more sophisticated measures such as "visual dB", a signal to noise ratio measure that takes into account the noise visibility function.

When the perceptual quality is measured, if the requirements are not satisfied the Y – force of watermark embedding should be reduced and watermark embedding stage should be repeated.

3.1.2. Watermark Detection

Step12- performing second level DWT and choosing the LL sub range

Step13 – performing DCT transform of the chosen LL sub range.

Step14 - is the extraction of the marked coefficients. Extract the marked coefficients corresponding to each bit, using key $K2$. After adding the two coefficient sets, T_n separate sets of watermarked coefficient sequences, each of size ch, are obtained.

These sets are the data independently handled at the bit decoding level. In other words, each bit is separately decoded through repetitive marking (8 times).

We denote this set of ch-length vectors as $I_w(i)$ where i is between 1 and T_n.

$$\tilde{I}_w = \left[\tilde{I}_w(1)\tilde{I}_w(2)..\tilde{I}_w(T_n)\right] \qquad \text{where } \tilde{I}_w(i) = \left[\tilde{I}_w^1(i)\tilde{I}_w^2(i)..\tilde{I}_w^{ch}(i)\right] \tag{3.5}$$

Step15 is repetition decoding. Key $K1$ is used to regenerate the same pseudo-random sequence p_C used for spectrum spreading. This p_C sequence, which is of length T_C, is divided into T_n consecutive ch length sequences $p_C(i)$ corresponding to each coded bit.

$I_W(i)$ and $p_C(i)$ are used for each bit, specified by the decoder.

Step16 is error correction decoding. The message bits are decoded from the detected code bits.

3.2. Detection Mechanisms

Low Density Parity Check ($LDPC$)[2][8] codes are linear parity check codes with parity check matrices H that have a small number of terms equal to 1, while the rest are zeros. $A(d,p,y)$ $LDPC$ code's parity check matrix H has ρ terms equal to 1 in each column and y number of ones in each row, where d is the row size. Therefore, the column length is (dp/y). The code rate is $(y-p)/y$.

3.2.1 Substitution Marking:

There are various substitution algorithms. We will shortly refer to the one developed by Coch and Zhao and [3][11].

The algorithm works on randomly selected 8x8 DCT coefficient blocks. Two coefficients from the mid-frequency range are randomly selected.

Say f_b denotes the 8x8 DCT block and $f_b(m_1,n_1)$, $f_b(m_2,n_2)$ denote the selected coefficients. The absolute difference between the selected coefficients is given by

$$A_b = \left| f_b\left(m_{1,}, n_{1,}\right)\right| - \left| f_b\left(m_{2,}, n_{2,}\right)\right| \quad (3.6)$$

To embed one bit w_i in the selected block, the coefficient pair $f_b(m_1,n_1)$, $f_b(m_2,n_2)$ is modified such that the distance becomes

$$\Delta_H = \left\{ \begin{matrix} \geq q \\ \leq -q \end{matrix} ako \begin{matrix} w=1 \\ w=0 \end{matrix} \right\} \quad (3.7)$$

And a forward DCT (3,8) where q is a parameter controlling the embeddingstrength.and 8x8 inverse DCT(IDCT) is defined as:

$$\begin{aligned} \mathrm{I}(u,v) = \frac{\zeta(u)}{2} \cdot \frac{\zeta(v)}{2} \sum_{k=0}^{7} \sum_{l=0}^{7} \mathrm{X}(k,l) \cdot \\ \cos\left(\frac{(2\kappa+1)u\pi}{16}\right) \cdot \cos\left(\frac{(2l+1)v\pi}{16}\right) \end{aligned} \quad (3.8)$$

And 8x8 inverse DCT(IDCT) is defined as:

$$\begin{aligned} \mathrm{X}(k,l) = \sum_{u=0}^{7} \sum_{v=0}^{7} \frac{\zeta(u)}{2} \cdot \frac{\zeta(v)}{2} \mathrm{I}(u,v) \cdot \\ \cos\left(\frac{(2\kappa+1)u\pi}{16}\right) \cdot \cos\left(\frac{(2l+1)v\pi}{16}\right) \end{aligned} \quad (3.9)$$

$$\begin{aligned} & where k,l,u,v \ \varepsilon = \{0,1,2,3,4,5,6,7\} \\ & and \ \zeta(u) = \begin{pmatrix} \frac{1}{\sqrt{2}} for \ u=0 \\ 1 for \ u>0 \end{pmatrix}, \zeta(v) = \begin{pmatrix} \frac{1}{\sqrt{2}} for \ v=0 \\ 1 for \ v>0 \end{pmatrix} \end{aligned} \quad (3.10)$$

This set of ch-length vector as $\tilde{I}^1{}_w(i)$ where I is between 1 and T_n

$$\begin{aligned} & \tilde{I}_w = \left[\tilde{I}_w(1), \tilde{I}_w(2) \ldots \tilde{I}_w(T_n)\right] \\ & where \quad \tilde{I}_w(i) = \left[\tilde{I}^1{}_w(i), \tilde{I}^2{}_w(i) \ldots \tilde{I}^{ch}{}_w(i)\right] \end{aligned} \quad (3.11)$$

4 . Experimental results and discussion

Experimental results are shown in fig.4.1-and fig.4,2. They are valid only for RAW images, without taking into consideration the specifics of the different algorithms for image encoding and also without taking into consideration the embedding of the whole watermark.

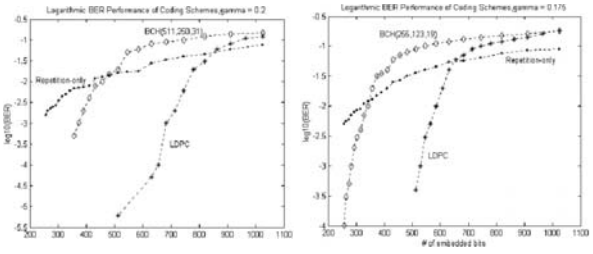


Fig.4.1 and Fig.4.2

Experiments were based on the standard images, such a Lena, Barbon, Peppers etc.

Partial simulation of the data traffic via the print server had been accounted. The research does not analyse data packages.

Simulation embedding shows that we may obtain good results for a sequence length of approximately 450 bits, together with the correction coefficients. One may see that the proposed algorithm is able to correct up to 3 % of errors. This scheme also defines the strength of embedding.

The results for the level of credibility are near to the theoretical ones, and they provide comparatively good results for a level of BER approximately equal to 0,2.

That is why we proposed the LDPC codes as they are reckoned to be the best error correction codes in the proximity of the Shannon limits[5],[12].

At the same time when choosing an error of ~ 0,175 and going in the vicinity of the Shannon threshold we observed that LDPC imposes a comparatively high level of noise, so one has to use some other algorithm instead.

5. Conclusion

Like in every communications system, watermarking systems also depend on the transmitted signal energy since BER decreases when signal energy is increased. Watermark energy increases linearly with y., BER test results of the repetition-only and $LDPC$ coded DFT techniques as a function of y are given. It is interesting to see that repetition- only logarithmic BER decreases approximately linearly whereas $LDPC$ logarithmic BER decreases at a much higher rate. As the channel condition is improved by an application of higher y, $LDPC$ coding brings in further improvement.

As we increase the watermark strength to achieve more reliable system, we decrease the perceptual quality of the marked image, which is the second performance criterion of the watermarking techniques. "Peak signal to noise ratio" ($PSNR$) and "Watermark to document ratio" (WDR) values, which were explained before, are given in Table 4.1.

The theoretical BER for this bit is equal to $Q\left(\sqrt{SNR}\right)=1/2erfc\left(\sqrt{SNR/2}\right)$. In order to determine the BER, we take the mean of the obtained values over all bits.

Table 4.1. $PSNR$ and WDR Results

Y	PSNR (dB)	WDR (dB)
0.175	42.78	-37.87
0.2	41.65	-36.74

References

[1]. Върбанов, Г., Д. Илиева, А. Недев. Робастно маркиране на изображения с воден знак, използуващо вълново преобразувание. Юбилейна научна сесия ВВУ" Н. Й. Вапцаров", 14-15 май 2001 г.

[2]. Varbanov, G. V., V. G. Naumov, D. D. Ilieva. Wavelet-Based Watermarking Algorithm for Still Images. Юбилейна научна сесия ВВУ" Н. Й. Вапцаров", 14-15 май 2001 г.

[3] Върбанов, Г. ,Д. Илиева, Н. Николов,” Защита на изображение от копиране чрез цифрово маркиране с водни знаци”. Научна конференция с международно участие ТЕЛЕКОМ – Варна, 10-12 октомври, 2001. изх.No 0183 17.10.2001
[4]. Kundur, D. , D. Hatzinakos, "Digital watermarking for telltale tamperproofing and authentication," in Proceedings of the IEEE: Special Issue on Identification and Protection of Multimedia. Information, vol. 87, pp. 1167-1180, July 1999.
[5]. Xia, X.G., C.G.Boncelet, and G.R. Arce," Wavelet transform based watermark for digital images." Optics Express 3.p. 497, December 1998.
[6]. Su, P.C., H.J. Wang, and C.C. J. Kuo, "Digital image watermarking in regions of interest" in The IS&T Image. Processing, Image Quality, Image Capture.Systems Conference. PICS '99, pp. 295-300, (Savannah, GA, USA), April 1999.
[7]. Xie L. , G. R. Arce, "Joint wavelet compression and authentication watermarking," in Proceedings of the IEEE International Conference, on Image. Processing,ICIP'93,(Chicago, IL, USA), 1998.
[8]. Wang, H. J., C. C. J.Kuo, Image protection via watermarking on perceptually significant wavelet coefficients," in Proceedings of the 1998 IEEE Workshop on Multimedia Signal Processing, (Los Angeles, CA, USA), December 1998.
[9]. Dugad, R., K. Ratakonda, and N. Ahuja, "A new wavelet-based scheme for water-marking images," in Proceedings of the.IEEE International Conference on Image. Processing, ICIP '98, (Chicago, IL, USA), October 1998.
[10]. Barni, M., F. Eartolini, V. Gappeliini, A. Lippi and A. Piva, "A DWT-based technique for spatiofrequency masking of digital signatures," in Proceedings of the ll-th SPIE Annual Symposium, Electronic. Imaging '99, Security and Watermarking of Multimedia Contents, P. W. Wong, ed., vol. 3657, (San Jose, GA, USA), January 1999.
[11]. Tsekeridou, S., I. Pitas, "Embedding self-similar watermarks in the wavelet domain," in Proceedings of the. IEEE ICASSP 2000,(Istanbul, Turkey,June 2000.
[12]. Daly, S., W. Zeng, J. Li, and S. Lei,"Visual masking in wavelet compression for JPEG2000," in Proceedings of IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II, P. W. Wong, ed., vol. 3971, (San Jose, CA, USA), January 2000.

For Contacts:
Ass. Prof. Georgi B. Varbanov
Computer Science and Engineering Dpt.
Technical University of Varna
E-mail: varbanov@ieee.bg ,gvarbanov@gmail.com
Рецензент: доц. д-р инж. Сл. Йорданова, ТУ – Варна

Bulgaria Communications Chapter

ИЗСЛЕДВАНЕ НА ВЪЗМОЖНОСТИТЕ НА КОМПЮТЪРИЗИРАНА СИСТЕМА ЗА СПИРОМЕТРИЧНА ДИАГНОСТИКА

Емилиян Б. Беков, Петър Г. Генов

Резюме: Представени са основни принципи на работа и възможности за практическо приложение на компютъризирана система за спирометрична диагностика. На основата на разнообразна клинична информация и от собствени експериментални изследвания се предлага алгоритмично оптимизиране на спирометрични изследвания чрез въвеждане на обобщен нормиран количествен показател. Същият осигурява достатъчно бърза и представителна оценка на текущото състояние на дихателната система в условията на провеждане на масови профилактични изследвания.

Analysis of the Possibilities of Computer Spirometry Diagnostic System

Emilian B. Bekov, Peter G. Genov

Abstract: The basic principles of work and practical possibilities of the computer spirometry diagnostic system are presented. On the basis of diverse clinical data and own experimental research is proposed algorithmic of optimization spirometry research by introducing normalize quantitative indicator. This indicator provides quick enough and representative assessment of the current state of the respiratory system in terms of conducting prophylactic examinations.

1. Увод

През последните години по данни от проведени профилактични прегледи се установяват значителни отклонения в състоянието на дихателната система на изследваните – средно при 25 - 30 % от общия брой изследвани. Като основни причини се посочват различни вирусни и бактериални инфекции, въздействието на алергени и вредни вещества, наличие на вредни навици (в частност тютюнопушене) и нездравословен начин на живот.

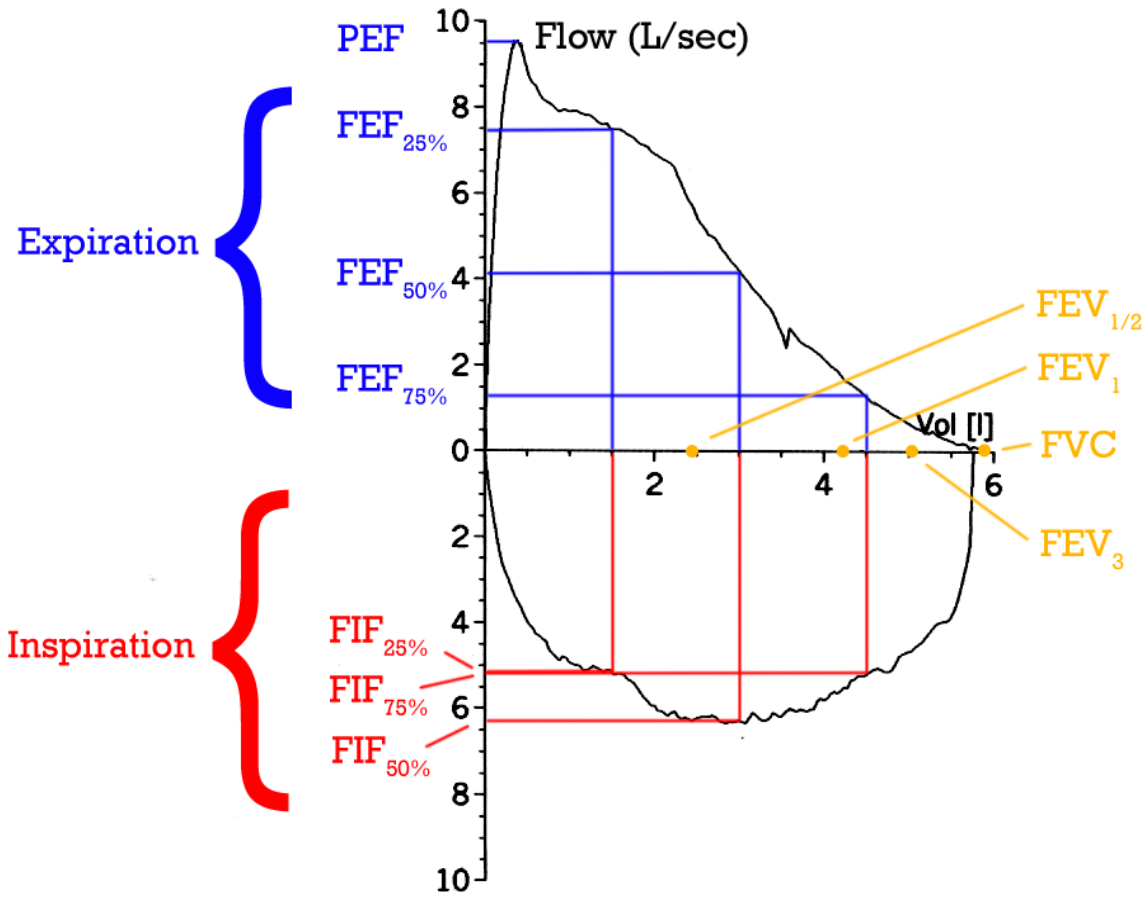
Един от най-често използваните методи за изследване и оценка на белодробната функция на човешкия организъм е спирометричният. С негова помощ се регистрират редица характеристики на дихателната система, например изменения на обема, скорост на въздушния поток и др. Най-често подобна информация се получава при форсирана експирация (издишване).

2. Оценка на белодробната функция

Обикновено при експресно изследване на белодробната функция се използват специализирани технически средства – спирометри. Типичният вид на диаграмата въздушен поток – белодробен обем е показана на фигура 1, където са представени и някои характерни показатели.

В клиничната практика се работи предимно със следните характеристики:

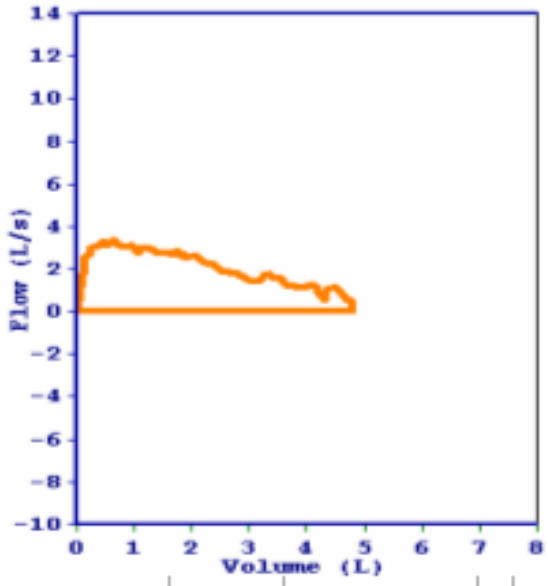
Volume - въздушен обем;
Flow - въздушен поток;
Expiration – издишване;
Inspiration – вдишване;



Фиг.1. Диаграма въздушен поток – белодробен обем

FVC - максимален жизнен въздушен обем;
PEF - максимална стойност на издишвания въздушен поток;
FEV_1 - издишван въздушен обем през първата секунда;
FEV_3 - издишван въздушен обем през първите три секунди;
$FEV_{1/2}$ - издишван въздушен обем при намаляване на стойността на издишвания въздушен поток с 50 % от максималната;
$FEF_{25\%}$ - стойността на въздушния поток при достигане на 25 % от максималния издишван жизнен въздушен поток;
$FEF_{50\%}$ - стойността на въздушния поток при достигане на 50 % от максималния издишван жизнен въздушен поток;
$FEF_{75\%}$ - стойността на въздушния поток при достигане на 75 % от максималния издишван жизнен въздушен поток;
$FIF_{25\%}$ - стойността на въздушния поток при достигане на 25 % от максималния вдишан жизнен въздушен поток;
$FIF_{50\%}$ - стойността на въздушния поток при достигане на 50 % от максималния вдишан жизнен въздушен поток;
$FIF_{75\%}$ - стойността на въздушния поток при достигане на 75 % от максималния вдишан жизнен въздушен поток;
Ординатна ос – въздушен поток [L/s];
Абсцисна ос – въздушен обем [L].

DATE		17.12.2008			
TIME		12:09:43 AM			
PHASE		PRE-BEST	PRED.		
FVC	L	4,8	FVC	4,4684	
FEV1	L	2,76	FEV1	3,6336	
FEV1/VC	%	0	FEV1/VC	81,3982	
FEV1/FVC	%	57,5	FEV1/FVC	81,3982	
FEV6	L	4,8	FEV6	4,4684	
FEV1/FEV6	%	57,5	FEV1/FEV6	81,20805369	
PEF	L/s	3,39	PEF	8,672	
FEF2575	L/s	2,19	FEF2575	3,7637	
FEV3	L	4,8	FEV3	4,19725	
FEV3/FVC	%	100	FEV3/FVC	93,93183242	
ELA	Years	81	ELA	51	
FET	s	2,85	FET	6	
FEF25	L/s	3,04	FEF25	7,997	
FEF50	L/s	2,13	FEF50	4,5625	
FEF75	L/s	1,31	FEF75	1,7693	
EVol	mL	60	EVol		
FIVC	L	0	FIVC	4,4684	
FIV1	L	0	FIV1	3,6336	
FIV1/FIVC	%	0	FIV1/FIVC	81,3982	
PIF	L/s	0	PIF	8,672	
MVV(cal)	L/min	96,6	MVV(cal)	129,13029	



Bekov
Emilian
Male
Caucasian
Age 51
175 cm
75 kg
No smoker

Фиг.2. Резултати от изследване чрез форсирана експираторна спирометрия

3. Клинични изследвания

За целите на настоящата разработка бяха проведени серия клинични изследвания с един от широко разпространените спирометрични компютъризирани апарати Minispir производство на фирмата MIR – Италия.

На фигура 2. са представени резултати от конкретно изследване, включващо графично изображение на форсирана експираторна спирометрия, компютърно получени конкретни числени стойности на съответните характеристики при различно функционално състояние на изследваното лице.

4. Заключение

Анализът на представените тук, а също така и от други източници, резултати от информационна гледна точка позволява да се направят следните обобщения:

- графичните изображения дават възможност само за качествена оценка;
- многобройните числени показатели (няколко десетки) осигуряват достатъчно подробна, но трудна за интерпретация количествена информация.

При масови профилактични спирометрични изследвания броят на пациентите е голям, а времето за интерпретация от страна на медицинските специалисти е ограничено. За успешното преодоляване на този проблем в настоящата работа се предлага синтезиране на обобщен диагностичен количествен коефициент **A.** При него се използват следните основни спирометрични характеристики: FVCi, $FEV_{50\%}$i и PEFi, нормирани с абсолютни средни стойности FVCcc, $FEV_{50\%}$cc и PEFcc, от таблицата по-горе, съответно за мъже и жени.

Тогава обобщеният нормиран коефициент **A** (с максимална стойност 100), характеризиращ текущото състояние на белодробната система се изчислява по следния начин:

$$A = 65\frac{FVCi}{FVCcc} + 25\frac{FEV_{50\%}i}{FEV_{50\%}cc} + 10\frac{PEFi}{PEFcc} \qquad (1)$$

При набирането на достатъчно статистическа информация, желателно по възрастов признак, което е задача на бъдеща работа, е възможно уточняване на доверителни интервали за оценка текущото състояние на белодробната система при масови спирометрични профилактични изследвания. В случай, че стойността на обобщения нормиран коефициент **A** се окаже извън границите на съответния доверителен интервал, е възможно със същата компютъризирана спирометрична апаратура да бъде получена подробна клинична информация.

Литература

[1] Клемент, Р.Ф., Н.А. Зильбер - Функционально-диагностические исследования в пульмонологии: Методические рекомендации, С.Петербург, 1993

[2] Савельев, Б.П. - Функциональные параметры системы дыхания у здоровых и больных детей в покое и при нагрузке: Автореф. дис. докт. мед. наук, Москва, 1997

[3] Савельев, Б.П., И.С. Ширяева - Функциональные параметры системы дыхания у детей и подростков: Руководство для врачей, Москва, 2001

[4] Brusasco, V., R. Crapo, G. Viegi, ATS/ERS task force: standardization of lung function testing, European Respiratory Journal, 2005

За контакти:
Гл.ас. инж. Емилиян Беков
катедра Електронна техника и микроелектроника
Технически университет – Варна
E-mail:emo_bekov@hotmail.com

Рецензент: доц. д-р инж. О. Железов, ТУ – Варна

ЭТАПЫ РАЗВИТИЯ РАБОЧЕГО ДИАГНОСТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

Александр В. Дрозд

Резюме: Исследуются этапы развития рабочего диагностирования вычислительных устройств. Отмечается естественный рост значимости приближенных вычислений. Оценивается достоверность методов рабочего диагностирования. Показывается низкая достоверность традиционных методов рабочего диагностирования в задачах обработки приближенных данных. Предлагаются пути повышения достоверности методов рабочего диагностирования.

Development Stages of On-Line Testing in Computing Devices

Alexander V. Drozd

Abstract: Development stages of on-line testing in computing devices are investigated. Natural growth of the importance of the approximated calculations is noted. Reliability of on-line testing is estimated. Low reliability of traditional on-line testing methods in tasks of approximate data processing is shown. Ways of increase in reliability of on-line testing methods are offered.

1. Введение

Рабочее диагностирование (РД) имеет много названий, и принято считать, что заключается в контроле правильности работы цифровой схемы вычислительного устройства (ВУ) на рабочих входных словах [1]. Такой *контроль* обнаруживает ошибку одновременно с работой ВУ (*concurrent checking* [2], *concurrent error detection* [3]), дает оперативную оценку его технического состояния, называясь также *оперативным* (*on-line testing*) [4]. Условия его выполнения подразумевают аппаратную реализацию и неразрывную связь с контролируемым ВУ, за что данный контроль называют также *аппаратным* [5] в отличие от программного и *встроенным* в противоположность выносному контролю. По характеру входных слов – рабочих или тестовых – РД получило название в сравнении с тестовым диагностированием.

В развитии РД можно выделить три основных этапа. На начальном этапе РД наследовало методы и средства из теории и практики связи для передачи сообщений на расстояния. Помехоустойчивая передача потребовала использования кодирующих и декодирующих устройств, ставших прообразом схем контроля в РД ВУ. Эти устройства считались абсолютно надежными в процессе передачи сообщений и потому проверялись только в паузах работы на тестах. Схемы контроля РД наследовали эту особенность, что и стало отличительной чертой начального этапа – РД не распространялось на собственные средства.

Следующий, второй этап, начавшийся с конца 60-х годов прошлого столетия, ознаменовался созданием и развитием теории самопроверяемых схем. Согласно определениям этой теории схема является полностью самопроверяемой в заданном множестве неисправностей, если она является защищенной и самотестируемой в этом множестве неисправностей [6].

Защищенная схема трактуется с учетом наличия в коде результата информационной избыточности, т.е. разбиения множества всех слов, принимаемых результатом, на разрешенные, которые формируются при правильной работе схемы, и запрещенные, возникающие только при ошибке. Множество запрещенных слов не является пустым.

Тогда, схема является защищенной для заданного множества неисправностей, если для каждой из них на выходе схемы на рабочих входных словах вычисляется или правильное или запрещенное слово. Схема является самотестируемой для заданного множества неисправностей, если для каждой из них на выходе схемы вычисляется запрещенное слово хотя бы для одного рабочего входного слова. Защищенная схема не позволяет перейти под действием неисправности от одного разрешенного слова к другому, что требует определенной информационной избыточности, рассчитанное на противодействие одной неисправности из заданного множества. Суть самотестируемой схемы состоит в создании условия для обнаружения первой неисправности до возникновения второй. Условие предполагает, что в интервале времени между этими неисправностями появятся все входные рабочие слова. Это возможно при редком появлении неисправностей и работе схемы на высокой частоте, т.е. при высокой надежности и производительности схемных решений.

Из определений теории самопроверяемых схем следует цель РД – контроль исправности цифровой схемы путем поиска ее неисправностей в процессе работы на фактических данных – и основное требование к методам РД – обнаружение неисправности по первой ошибке [7].

На теорию и практику самопроверяемых схем пришлось основное развитие РД – самопроверяемые схемы были разработаны для комбинационных схем [8], синхронных и асинхронных автоматов [9], а также сумматоров, умножителей, делителей, АЛУ и т.д. [10]. Но к настоящему времени количество публикаций в этом направлении резко уменьшилось, что знаменует окончание второго и начало третьего этапа в развитии РД, продиктованного современным уровнем и дальнейшим совершенствованием компьютерных систем и их компонентов, т.е. объектов РД. Наметилось определенное отставание в развитии РД по отношению к его объектам, которое проявляется с возрастанием объемов обработки приближенных данных. Даная работа посвящена анализу особенностей второго и третьего этапов развития РД, связанных с перемещением приоритетов с точных вычислений на приближенные.

2. Особенности второго этапа развития РД

Второй этап развития РД вплоть до его пика пришелся на период разработки ВУ на элементах малой и средней степени интеграции в рамках целочисленной арифметики, работающей с полной разрядностью параллельных двоичных кодов чисел.

Следует различать точные данные как целые числа по своей природе – номера элементов множеств – и приближенные данные – результаты измерений и их обработки, имеющие определенные погрешности. Человек мыслит точными данными.

Например, температуру тела 36,6° он представляет себе как 366-е по значимости числовое значение из диапазона 0 ÷ 999. В компьютере поддерживает такое же восприятие данных, поскольку представление информации двоичными или другими кодами, т.е. кодирование данных, является нумерацией. Все, что можно записать в разрядную сетку компьютера, – точные данные, поскольку их значения можно поставить в соответствие порядковым номерам, т.е. пронумеровать. Например, в 3-битовой разрядной сетке могут быть записаны двоичные коды, значения которых 0, 1, …, 7 могут быть отождествлены с их номерами: от нулевого до седьмого. В этом проявляется *модель точных данных*, которая состоит в том, что все числа, независимо от их настоящей природы, рассматриваются как точные данные.

Наш мир, напротив, является генератором приближенных данных, обладающих погрешностью, которая обусловлена не недостаточно качественным измерением, а сущностью этого мира. Действительно, почему, например, измерение длины стола не может дать точный результат? Во-первых, мы созданы такими, что не можем зафиксировать две точки, между которыми следует выполнить измерение. Во-вторых, в этом мире не найдется инструмента для

проведения измерения с нулевой инструментальной погрешностью. Но даже если это можно было бы преодолеть, то остается вопрос: «Между какими точками следует выполнять замер, между крайними молекулами стола, крайними атомами и т.д.?». Наконец, если бы удалось найти эти две точки, то время в них течет разное, а потому наш мир принципиально устроен так, что точные измерения не возможны. Все в этом мире существует в допусках и потому получаемые в нем количественные оценки являются приближенными данными. Например, температура тела живого человека находится в интервале 30° – 42°, в пределах которого реализуется право на ошибку, т.е. право на существование. Погрешность числа – это его жизненное пространство, в пределах которого это число существует, т.е. считается достоверным.

Конфликт между миром приближенных данных и моделью точных данных приводит к заблуждениям во многих областях человеческой деятельности и может быть рассмотрен на примере развития РД. Его второй этап оказался полностью в плену модели точных данных.

Действительно, поиск неисправностей в ходе вычислений противоречит здравому смыслу, также как и в процессе полета самолета с пассажирами. Если задаться такой целью, то следует делать фигуры высшего пилотажа, испытывающие самолет на прочность. Однако ставится другая цель – долететь, т.е. дорешать вычислительную задачу, что включает в себя выполнение заданного объема вычислений за ограниченное время с получением достоверных результатов.

Таким образом, настоящая цель РД состоит в *контроле достоверности вычисляемых результатов*, что позволяет решить вычислительную задачу, пересчитав выявленные недостоверные результаты за оставшееся время.

Практика подтверждает такую цель. Как известно, ошибки вызываются сбоями – кратковременными самоустраняющимися неисправностями – и отказами, т.е. устойчивыми нарушениями схемы. Сбой возникает намного чаще, чем отказ. Поэтому первая обнаруживаемая ошибка, как правило, вызывается сбоем. Обнаружение сбоя не может характеризовать исправность схемы, поскольку действует короткий период времени, после чего схема снова исправна. Поэтому обнаружение сбоя необходимо не с целью контроля исправности схемы, а для ответа на вопрос: «Можно ли использовать вычисленный результат или он недостоверен?». Более того, по первой ошибке нельзя судить о неисправности самопроверяемой схемы и в случае отказа, поскольку его невозможно идентифицировать по одной ошибке, одиночной в случае сбоя или первой из серии ошибок при отказе.

Почему же определена не настоящая цель? Обнаруженная ошибка одновременно указывает и на наличие неисправности в схеме и на недостоверный результат. Но ошибочный результат тождественен недостоверному только в случае точных данных. Тогда действительно можно контролировать достоверность результата обнаружением неисправности схемы. Вот только в РД неисправность обнаруживается по ошибке результата, а не наоборот, т.е. средства и цель в РД поменяны местами. Сама логика самопроверяемых схем – вычислять достоверные результаты на исправной схеме – следует из догмы, что исправная схема вычисляет только достоверные результаты, а недостоверные результаты вычисляются только неисправной схемой. Однако эта догма также справедлива только для случая точных данных, что будет показано далее.

Таким образом, второй этап определил развитие РД для частного случая точных данных.

3. Особенности третьего этапа развития РД

Третий этап связан с дальнейшим структурированием вычислительной техники под реалии нашего мира, которое проявляется не только в последующем распараллеливании вычислений, но также и, прежде всего, в бурном росте объемов приближенны вычислений.

Это можно проследить на примере развития персональных компьютеров в части аппаратной поддержки арифметики с плавающей точкой, используемой для приближенных

вычислений: от сопроцессоров необязательной поставки (Intel8087/287/387), до встроенных конвейеров FPU с плавающей точкой в составе Intel486, Pentium и т.д. [11].

На фоне такого развития лучше видны заблуждения, сложившиеся на втором этапе.

Основные черты третьего этапа связаны с особенностями обработки приближенных данных, которые определяют результат, состоящим из старших верных и младших неверных разрядов. В них неисправности схемы вызывают ошибки, являющиеся соответственно существенными и несущественными для достоверности результата. Особенности приближенных вычислений снижают вероятность P_C существенной ошибки (вероятность того, что возникшая ошибка является существенной).

Первой особенностью является использование операции умножения в самой записи приближенного данного в форматах с плавающей точкой: число записывается в виде произведения $m \cdot q^p$, где m – мантисса, q – основание системы счисления, p – порядок [12]. Произведение двух операндов удваивает длину результата двуместной операции. Согласно теории ошибок, количество верных разрядов результата не превышает количество верных разрядов операнда. Поэтому на неисправной схеме могут вычисляться достоверные результаты, содержащие несущественные ошибки. Данная особенность вдвое снижает P_C. Младшие неверные разряды результата, как правило, отбрасываются, что объясняет наибольшее распространение одинарных форматов с плавающей точкой, для которых результаты наследуют разрядность операндов.

Второй особенностью обработки приближенных данных является нарушение ассоциативного и дистрибутивного законов, что можно увидеть на примере сложения миллиона с миллионом единиц при использовании двуместных операций и разрядной сетки длиной $n < 20$. Сумма миллиона и единицы равна миллиону, поскольку единица теряется при выравнивании порядков. Миллион таких операций также определяет результат, равный первому числу, т.е. одному миллиону. Правильный результат – два миллиона – можно получить, изменив порядок вычислений. Сначала следует сложить пары единиц, затем полученные двойки и так продолжать до вычисления миллиона, который далее складывается с первым числом. Вычисление одного миллиона вместо двух показывает несостоятельность догмы, утверждающей, что исправная схема вычисляет достоверный результат. Для восстановления ассоциативного и дистрибутивного законов используются расширенные форматы с плавающей точкой, в которых мантиссы удлиняются за счет младших неверных разрядов, что дополнительно снижает вероятность существенной ошибки P_C.

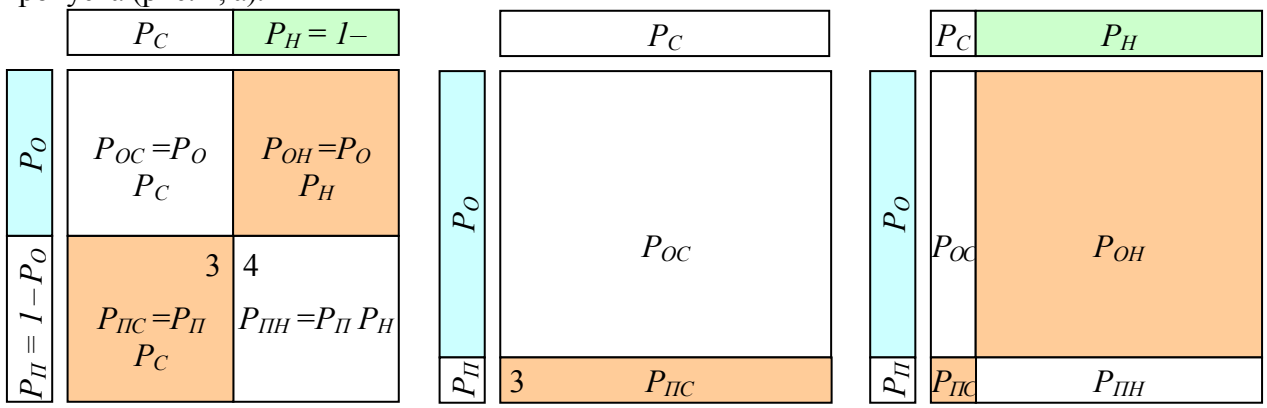
Третья особенность связана с выполнением операций нормализации и денормализации мантисс, что необходимо выполнять соответственно над результатами и операндами для согласования действий над мантиссами и порядками. Денормализация операнда выполняется арифметическим сдвигом его мантиссы вправо с потерей младших разрядов, выдвигаемых за пределы разрядной сетки. Соответственно теряются существенные ошибки, которые накапливались в результатах всех предыдущих операций, что снижает вероятность P_C. Нормализация результата при вычислении в нем незначащих цифр выполняется циклическим сдвигом его мантиссы влево с заполнением младших позиций неверными разрядами. Это ведет к уменьшению количества верных разрядов и существенных ошибок в них для результатов всех последующих операций, также снижая вероятность P_C.

Таким образом, особенности приближенных вычислений могут в значительной степени снижать вероятность существенной ошибки P_C.

Ключевым вопросом в оценке РД является достоверность его методов, которая может быть рассмотрена с использованием квадрата с единичной длиной сторон (рис. 1) [13].

На горизонтальной стороне квадрата откладываются вероятности P_C и $P_H = 1 - P_C$ того, что возникшая ошибка является существенной и несущественной, а на вертикальной стороне – вероятности P_O и $P_П = 1 - P_O$ ее обнаружения и пропуска. При этом квадрат разбивается на четыре части, определяющие вероятности $P_{OC} = P_O P_C$ и $P_{OH} = P_O P_H$ обнаружения

существенных и несущественных ошибок, а также вероятности $P_{ПС} = P_П P_С$ и $P_{ПН} = P_П P_Н$ их пропуска (рис. 1, а).



а) **б)** **в)**

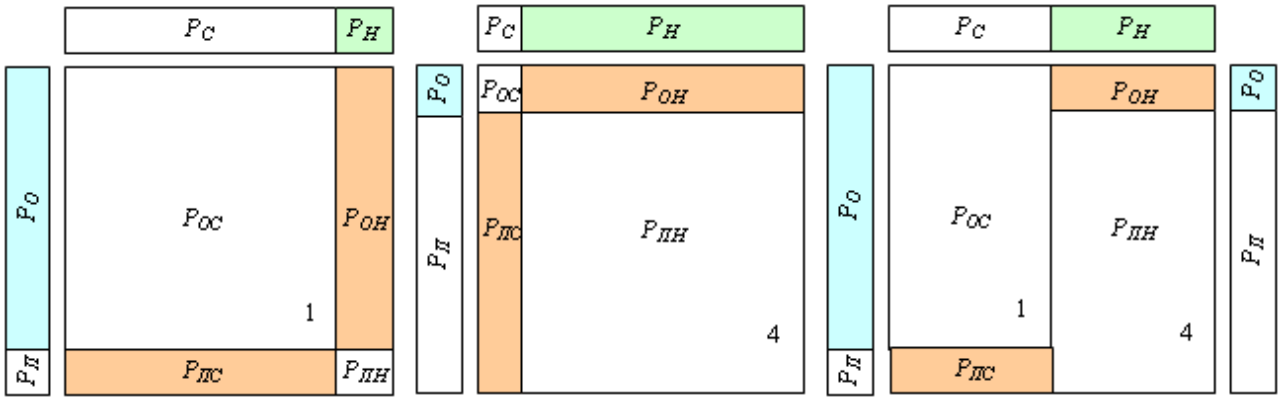
Рис. 1. Оценка достоверности методов РД

Согласно цели, следующей из теории самопроверяемых схем, метод РД является достоверным, если обнаруживается неисправность, независимо от того, является ли ошибка существенной или нет. Это определяет достоверность метода РД как сумму частей 1 и 2 квадрата по формуле $D_О = P_{ОС} + P_{ОН} = P_О$.

Согласно настоящей цели, метод РД является достоверным, если правильно причисляет вычисляемый результат к достоверным или недостоверным. Поэтому достоверность метода РД, является достоверностью контроля результатов и определяется с учетом обнаружения существенной и пропуска несущественной ошибки суммой частей 1 и 4 квадрата: $D = P_{ОС} + P_{ПН} = P_О P_С + (1 - P_О)(1 - P_С)$.

В случае точных данных все ошибки являются верными, т.е. $P_С = 1$, и методы РД с традиционно высокой вероятностью $P_О$ обнаружения ошибки имеют высокую достоверность $D = P_{ОС}$, определяемую частью 1 квадрата (рис. 1, б).

Для приближенных вычислений с вероятностью $P_С << P_Н$ традиционные методы РД характеризуются близкими к нулю частями квадрата 1 и 4, что определяет низкую достоверность контроля результатов (рис. 1, в). Методы РД демонстрируют новое качество – отбраковывать достоверные результаты по несущественным ошибкам (часть 2 квадрата).



а) **б)** **в)**

Рис. 2. Пути повышения достоверности методов РД

Анализируя квадрат, можно предложить три пути повышения достоверности методов РД:

– для $P_С > 0{,}5$ увеличивать часть 1 квадрата, повышая вероятность $P_О$ (рис. 2, а);

– для P_C < 0,5 увеличивать часть 4 квадрата, снижая вероятность P_O (рис. 2, б);

– увеличивать части 1 и 4 квадрата, делая вероятность P_O различной для обнаружения существенных (P_{O-C}) и несущественных (P_{O-H}) ошибок, $P_{O-C} > P_{O-H}$ (рис. 2, в).

4. Заключение

В развитии РД, нацеленного на контроль достоверности результатов, можно выделить три этапа. На начальном этапе достоверность результатов оценивалась без распространения РД на собственные схемы контроля. Второй этап ознаменовался созданием и развитием теории самопроверяемых схем, которая решает задачу контроля достоверности результатов только для случая точных данных. Третий этап распространяет РД на обработку приближенных данных, значимость которой с развитием вычислительной техники постоянно растет. Особенности приближенных вычислений снижают вероятность существенной ошибки, что приводит к значительному снижению достоверности традиционных методов РД в контроле результатов. Повышение достоверности контроля результатов возможно при одновременно высоких или низких значениях вероятности существенной ошибки и вероятности обнаружения ошибки, а также при большей вероятности обнаружения существенной ошибки по сравнению с несущественными.

Литература

[1]. Гуляев, В. А., С. М. Макаров, В. С. Новиков, Диагностика вычислительных машин. К. Техника 1981.

[2]. Metra C., M. Favalli and B. Ricco, Concurrent Checking of clock signal correctness. IEEE Design & Test, October, 1998.

[3]. Touba N. A. and E. J. McCluskey, Logic synthesis techniques for reduced area implementation of multilevel circuits with concurrent error detection, Proc. IEEE Inf. Conf. on Computer Aided Design, 1994.

[4]. Nicolaidis M. and Y. Zorian, On-line testing for VLSI – a compendium of approaches, Electronic Testing: Theory and Application (JETTA), 1998.

[5]. Журавлев, Ю. П., Л. А. Котелюк, Н. И. Циклинский, Надежность и контроль ЭВМ. М.: Советское радио 1978.

[6]. Anderson, D. A., G. Metze, Design of totally self-checking check circuits for m-out-of-n codes, IEEE Trans. Comput., V. C – 22, N 3, 1977.

[7]. Metra, C., L. Schiano, M. Favalli and B. Ricco, Self-checking scheme for the on-line testing of power supply noise, Proc. Design, Automation and Test in Europe Conf., Paris (France), 2002.

[8]. Smith, J. E. and G. Metze, The design of totally self-checking combinational circuits, Proc. Int. Symposium on Fault Tolerant Computing Dig, Los Angeles (USA), 1977.

[9]. Ozguner, F., Design of totally self-checking asynchronous and synchronous sequential machines, Proc. Int. Symposium on Fault Tolerant Computing Dig, Los Angeles (USA), 1977.

[10]. Nicolaidis, M., H. Bedder Efficient Implementation of Self-checking Multiply and Divide Arrays // in Proc. 1994 European Design and Test Conf., Paris (France), 1994.

[11]. Гук, М., Процессоры Intel: от 8086 до Pentium II, СПб: Питер, 1997.

[12]. ANSI/IEEE Std 754-1985. IEEE Standard for Binary Floating-Point Arithmetic. IEEE, New York, USA, 1985.
[13]. Drozd A., M. Lobachev, J. Drozd, The problem of on-line testing methods in approximate data processing, Proc. 12th IEEE International On-Line Testing Symposium, Como (Italy), 2006.

За контакти:
проф. д-р техн. наук Александр Дрозд
кафедра компьютерных интеллектуальных систем и сетей
Одесский национальный политехнический университет – Одесса
E-mail:drozd@ukr.net
Рецензент: проф. дтн. Инж. Л. Сотиров, ТУ – Варна

АВТОМАТИЗИРОВАННАЯ НАСТРОЙКА ПРОГРАММНЫХ КОМПОНЕНТ ИНФОРМАЦИОННЫХ СИСТЕМ

Александра Ю. Левченко

Резюме: Статья посвящена проблеме автоматизации процесса настройки системных программных компонент в корпоративной информационной системе. Основным способом решения является автоматизация этого процесса на основе мониторинга активности компонент. Выполнен анализ существующих подходов разработки блока настройки на основе эмпирических сравнений, механизмов обратной связи и аналитических моделей. Приведена сравнительная характеристика достоинств и недостатков подходов автоматизированной настройки в операционных системах и системах управления базами данных. Показано, что наиболее перспективной является настройка, основанная на аналитических моделях.

Automated software tuning

Alexandra Yu. Levchenko

Abstract: The article presents the overview of the problem of software tuning automation in corporate information systems. The main way to solve this problem is to automate this process using component activity monitoring. It is analyzed existing approaches to develop the tuner based on the empirical comparisons, feedback mechanisms and analytical models. The basic works using each of these approaches for automated software tuning are considered. The comparative characteristic of advantages and disadvantages of automated software tuning approaches is represented. It is shown, that the most perspective is the model-based tuning.

1. Введение

Совокупная стоимость владения информационной технологией (ССВ) все более зависит от стоимости человеческого труда. Все основные поставщики информационных технологий признают важность проблемы ССВ и пытаются решить ее путем обеспечения самоуправляемости своих систем. Автоматизация настройки программных компонент (ПК) корпоративных информационных систем (КИС) предполагает разработку информационной технологии (блока настройки), реализующей подбор значений конфигурационных параметров системы, которые обеспечивают наилучшую производительность. При этом необходимо выполнять мониторинг активности компонент операционной системы (ОС) для более точной настройки программ в заданных условиях функционирования. На сегодняшний день актуальной является задача настройки системы управления базами данных (СУБД), входящей в состав КИС. Для обеспечения наилучшей производительности различных классов КИС на базе СУБД могут потребоваться разные наборы параметров настройки. Таким образом, при настройке параметров СУБД следует учитывать класс настраиваемой КИС и поведение пользователя. Для характеристики класса КИС и поведения пользователя источником данных могут выступать отсылаемые пользователем транзакции. Большинство способов автоматизации настройки ПК КИС можно отнести к трем типам: основанные на эмпирических сравнениях, механизмах обратной связи и аналитических моделях.

Целью данной работы является анализ достоинств и недостатков указанных способов автоматизации настройки применительно к ОС и СУБД и выбор способа, эффективного для автоматизированной настройки ПК СУБД.

2. Эмпирические сравнения

Один из способов определения оптимальных значений параметров настройки для заданного класса КИС - эмпирическое сравнение производительности некоторых или всех возможных значений параметров настройки на основе сравнения производительности системы для каждого из испытанных значений. Обычно способ настройки, основанный на эмпирических сравнениях, предполагает исследование всех возможных значений параметров настройки для определения оптимальных, что может требовать слишком больших затрат времени для настройки объектов с большим количеством возможных значений параметров настройки. Возникает необходимость в методе сокращения пространства возможных значений параметров настройки.

Рассмотрим основные работы с использованием способа настройки ПК на основе эмпирических сравнений. В работе [1] предлагается методология построения ядра ОС, которое отслеживает свою собственную производительность и адаптируется к изменяющимся рабочим нагрузкам. Их подход включает использование сравнений для определения оптимальной стратегии (например, стратегия замены буферного кэша) для управления определенной рабочей нагрузкой. Сравнения выполняются путем загрузки специального модуля в ядро для разных рабочих нагрузок и последующего использования этих модулей для обработки фактических трассировок рабочей нагрузки. Данные модули не затрагивают глобальные ресурсы ОС в процессе исследования возможных альтернатив, что значительно сокращает потенциальные отрицательные эффекты влияния настройки на производительность ОС. Однако большинство ОС не обеспечивают расширяемость, необходимую для проведения данного типа моделирования.

В работе [2] также используются эмпирические сравнения для динамической регулировки значений параметров настройки в ОС. В процессе экспериментальной фазы состояние системы значительно изменялось, некоторые значения параметров настройки выбирались в случайном порядке, и замерялась производительность ОС. В течение времени система накопила достаточно много данных для определения оптимальных значений параметров настройки для каждого возможного состояния системы. Далее для каждого состояния системы проводились дополнительные эксперименты для адаптации к изменениям в окружающей среде. Авторы получили небольшое повышение производительности в экспериментах с настройкой двух параметров планирования в ОС с разделением времени. Степень ухудшения производительности ОС во время экспериментальной фазы не оценивалась.

Авторы работы [3] предлагают методологию, основанную на эмпирических сравнениях для построения самонастраивающихся ОС. Чтобы не выполнять полное сравнение всех возможных значений параметров настройки их методология использует генетические алгоритмы для поиска оптимальных значений параметров настройки. На каждом этапе сравнений проверяются несколько параметров с использованием моделирования в периоды бездействия системы. Текущая группа значений параметров настройки комбинируется и изменяется согласно их относительной производительности. Применяя наилучшие значения параметров настройки, генетические алгоритмы должны сходиться на значениях параметров, улучшающих производительность системы. Единственная ратификация этого подхода авторами представляет собой настройку параметризованного алгоритма планирования, который может быть оценен обособленно от остальной части ОС. Авторы отмечают, что могут возникнуть сложности при моделировании других аспектов работы ОС и необходимо оценивать дополнительные нагрузки в системе от моделирования в процессе настройки.

В работе [4] авторы описывают инструмент iTuned, который автоматизирует задачу идентификации хороших значений параметров настройки СУБД. Данный инструмент использует метод адаптивной выборки при выполнении плановых экспериментов для

отыскания значений параметров настройки, которые обеспечивают повышение производительности системы. Также в инструмент включен исполнительный модуль, который поддерживает Online-эксперименты в функционирующем окружении СУБД на основе концепции “с захватом цикла”, которая практически сводит к нулю дополнительную нагрузку на функционирующую систему.

3. Механизмы обратной связи

При использовании способа настройки программного обеспечения на основе обратной связи блок настройки итерационно регулирует значения параметров, основываясь на замерах одного или более критериев производительности. Текущее значение каждого показателя сравнивается с некоторым критическим значением. После применения полученного набора значений параметров, блок настройки замеряет новые значения показателей производительности и использует их для определения необходимости дополнительных корректировок. Этот повторный цикл наблюдений и корректировок часто упоминается как петля обратной связи, потому что эффекты влияния одного набора параметров на производительность подаются обратно в блок настройки и используются для определения следующего набора корректировок значений.

Рассмотрим основные работы с использованием механизмов обратной связи. В работе [5] используют подход с обратной связью для настройки мультипрограммного уровня СУБД (МПУ), параметра, который ограничивает число параллельных доступов к БД. Регулировка этого параметра основана на замерах конфликтов при блокировках в системе: когда этот показатель превышает критическое значение, МПУ уменьшается, когда он понижается ниже критического значения, МПУ увеличивается. Авторы определили критическое значение экспериментально, и они утверждают, что оно не должно быть точно настроенным. В работе представлены результаты экспериментов, демонстрирующих, что есть диапазон критических значений, которые применимы для широкого диапазона рабочих нагрузок.

Авторы работы [6] используют механизмы обратной связи для настройки параметров, связанных с управлением памятью и контролем нагрузки в СУБД. Цель настройки состоит в обеспечении целевого времени ответа для индивидуальных классов КИС в системах с несколькими классами КИС, основанных на СУБД. При этом параметры регулируются до тех пор, пока не будет достигнуто целевое значение или пока блок настройки не определит, что оно не может быть достигнуто. Для каждого класса параметры настраиваются отдельно, и для выявления взаимозависимостей между классами используется эвристика. В зависимости от характера использования памяти данным классом, регулируются один или два параметра. Экспериментальные моделирования показали, что оба блока настройки, для одного и для двух параметров, способны решить задачу для разных рабочих нагрузок. При этом авторы признают, что для достижения целевого времени ответа для определенных типов рабочих нагрузок может потребоваться достаточно много времени.

4. Аналитические модели

Блок настройки на основе аналитических моделей может прогнозировать производительность системы для любых классов КИС и значений параметров настройки. Для такого типа настройки могут использоваться статистические модели, регрессионные, графовые и вероятностные модели. Каждый тип модели имеет связанный набор параметров, значения которых обычно получаются из обучающих примеров - каждый из них состоит из статистики по характеристикам класса КИС, значений параметров настройки и производительности системы за некоторый интервал времени.

Выделим основные работы с использованием данного подхода. На платформе для дистанционных вычислений Odyssey приложения приспосабливаются к изменениям в доступных ресурсах и пользовательским целям путем изменения точности, с которой они функционируют (например, частота кадров, используемая приложениями потокового видео). Narayanan и соавторы [7] расширяют Odyssey системой, которая использует модели для прогнозирования использования ресурсов приложений как функции соответствующих входных параметров и показателей точности, и таким образом рекомендует соответствующие уровни точности для данной операции. Данные начального обучения собираются в специальном автономном режиме путем повторного выполнения заданных операций со случайно выбранной точностью и входами. Для начального опытного образца авторы используют линейную регрессию для создания моделей, которые используют линейный градиентный спуск для обновления коэффициентов моделей.

Работа [8] используют подход на основе модели для настройки измененной версии журналируемой файловой системы (ЖФС). Например, они позволяют ЖФС динамически выбрать лучший из двух методов сбора мусора в журнале регистраций, сохраняемом системой на диске. Предложенные авторами модели состоят из простых формул для оценки стоимости или отношения «затраты-выгода» возможных значений параметров настройки. Они основаны на оценке выполненных системой операций и связанными с ними затратами. Единственные параметры моделей - измерения затрат различных операций на определенном диске. Оценивается эффективность их подхода путем моделирования ЖФС и диска, на котором она хранится. Но авторы не подтверждают возможность прогнозирования производительности более сложных систем при использовании таких моделей.

Работа [9] используют настройку, основанную на моделях сетей массового обслуживания (СМО), для оптимизации качества обслуживания (КО) сайта электронной коммерции. Когда система обнаруживает, что гарантия КО была нарушена, используется метод поиска экстремума, управляемый прогнозами моделей, для поиска значений параметров настройки, которые обеспечивают достижения локального максимума КО. В результатах авторы показывают, что их блок настройки, регулирующий четыре параметра, способен обеспечить приемлемую величину КО в условиях возрастающей нагрузки. Однако, несмотря на то, что модели СМО эффективны в условиях, когда настраиваемые параметры непосредственно связаны с очередями запросов к Web-серверу и серверу приложений, неизвестно, могут ли они лечь в основу общей методологии настройки программного обеспечения.

В реляционных СУБД индексы и материализованные представления - дополнительные структуры данных, которые могут быть созданы для ускорения часто встречающихся запросов к БД. Проект AutoAdmin развил методы, основанные на модели, чтобы автоматизировать выбор при создании индексов и материализованных представлений [10, 11]. Авторы используют стоимостные оценки оптимизатора запросов СУБД как модель и развивают новые методы для того, чтобы выбрать какие индексы и материализованные представления, рассматривать и эффективно выполнять поиск во множестве возможных комбинаций индексов и материализованных представлений. Внутренние модели оптимизатора запросов в работах не обсуждаются.

В таблице 1 представлена сравнительная характеристика достоинств и недостатков предложенных способов автоматизации настройки параметров СУБД.

Таблица 1. Сравнительная характеристика способов автоматизированной настройки параметров СУБД

Способ	Достоинства	Недостатки
Эмпирические сравнения	- учитывают влияние различных значений параметров настройки на	- не позволяет делать вывод из опыта;

	производительность; - позволяет настраивать множество параметров одновременно.	- требуют дополнительного обучения, что может ухудшить производительность системы; - требуют полного перебора, что влечет большие затраты времени при настройке большого количества параметров; - требуют фактически использовать ОС или выполнять имитации.
Механизмы обратной связи	- позволяют легко приспособиться к изменениям среды; - не требуют дополнительного обучения; - эффективны при настройке в условиях резко колеблющихся характеристик рабочей нагрузки; -могут быть эффективны для определения пороговых значений, которыми управляются регуляторы блока настройки параметров.	- требует итерационных корректировок значений, может потребоваться большое количество корректировок для достижения оптимальных значений параметров настройки; - трудно применять при отсутствии очевидных критических значений; - не позволяют настраивать множество параметров одновременно; - не учитывают влияния значений параметров на производительность системы.
Аналитические модели	- позволяют настраивать множество параметров одновременно; - допускают делать вывод из опыта для прогнозирования; - позволяют определить оптимальные значения после единственного этапа вычислений; - позволяют адаптироваться к изменениям в среде путем обновления параметров модели; - сравнивают производительность различных значений параметров настройки на основе прогнозов модели.	- требуют начальных затрат времени на обучение модели; - сбор учебных примеров снижает производительность системы; - сложно разработать точную модель для сложной системы программного обеспечения.

6. Заключение

Для автоматизации задачи настройки параметров СУБД перспективным является использование подхода, основанного на аналитических моделях, который позволяет описать характеристики поведения пользователя, значения параметров настройки и производительность системы. Подход на основе моделей позволяет использовать ранее полученный опыт, и в отличие от эмпирических сравнений не требует каждый раз для настройки выполнять серию экспериментов в реальной системе для поиска наилучших значений параметров, что позволяет сократить затраты времени на настройку. Кроме того, опыт предыдущих настроек дает возможность делать прогнозы и приспосабливаться к

изменениям в среде. При использовании подхода на основе моделирования можно избежать потенциально длинного ряда повторяющихся регулировок, которых требует подход с обратной связью. С помощью моделируемого способа настройки можно настраивать множество параметров одновременно, что практически исключается при использовании подходов, основанных на эмпирических сравнениях и механизмах обратной связи. Среди проанализированных работ для настройки параметров СУБД широко используются как эмпирические сравнения [4] так и механизмы обратной связи [5,6]. Моделируемый подход для настройки параметров СУБД среди рассмотренных работ не применяется. Таким образом, учитывая выше обозначенные преимущества данного подхода, перспективным является применение его для настройки параметров СУБД.

Литература

[1]. Seltzer, M., C. Small. Self-monitoring and self-adapting operating systems. In Proceedings of the Sixth Workshop on Hot Topics on Operating Systems, Chatham, MA, 1997.

[2]. Reiner, D., T. Pinkerton. A method for adaptive performance improvement of operating systems. In Proceedings of the 1981 ACM Conference on the Measurement and Modeling of Computer Systems, Las Vegas, NV, 1981.

[3]. Feitelson, D., M. Naaman. Self-tuning systems. IEEE Software 16(2):52-60, 1999.

[4]. Duan S., V. Thummala, S. Babu. Tuning Database Configuration Parameters with iTuned. In Proceedings of the 35th Intl. Conf. on Very Large Databases (VLDB), Lyon, France, 2009.

[5]. Weikum, G., C. Hasse, A. Moenkeberg, P. Zabback. The COMFORT automatic tuning project, Information Systems 19(5):381-432, 1994.

[6]. Brown, K., M. Mehta, M. J. Carey, M. Livny. Towards automated performance tuning for complex workloads. In Proceedings of the 20th Intl. Conf. on VLDB, Santiago, Chile, 1994.

[7]. Narayanan D., J. Flinn, M. Satyanarayanan. Using history to improve mobile application adaptation. In Proceedings of the Third IEEE Workshop on Mobile Computing Systems and Applications, Monterey, CA, 2000.

[8]. Matthews, J.N., D. Roselli, A.M. Costello, R. Wang, T. Anderson. In Proceedings of the 16th ACM Symposium on Operating System Principles, Saint Malo, France, 1997.

[9]. Menascé, D. A., D. Barbará, R. Dodge. Preserving QoS of e-commerce sites through self-tuning: A performance model approach. In Proceedings of the 2001 ACM Conference on Economic Commerce, Tampa, FL, 2001.

[10]. Chaudhuri, S., V. Narasayya. An efficient cost-driven index selection tool for Microsoft SQL Server. In Proceedings of the 23rd Intl. Conf. on VLDB, Athens, Greece, 1997.

[11]. Agrawal, S., S. Chaudhuri, V. Narasayya. Automated selection of materialized views and indexes for SQL databases. In Proceedings of the 26th Intl. Conf. on VLDB, Cairo, Egypt, 2000.

Для контактов:
Александра Левченко
кафедра Системного программного обеспечения
Одесский национальный политехнический университет – Одесса
E-mail: aleksandra.levchenko@gmail.com

Рецензент: доц. д-р инж. О. Железов, ТУ – Варна

Ensuring Data Confidentiality in Relational Database Modeling.

Aleksej B. Kungurtsev, Svetlana.L. Zinovatnaya, Al Abdo Munzer

Abstract: In the report a method of data protection in assessment of information management system performance is offered. The algorithm of data encryption in relational database table columns, providing solution of the following issues: a) incapability of data decoding; b) saving variables distribution of the real database, is being considered.

Обеспечение конфиденциальности данных при моделировании реляционной базы данных

Алексей.Б. Кунгурцев, Светлана.Л Зиноватная., Аль Абдо Мунзер

Резюме: В докладе предлагается метод защиты данных при выполнении оценки производительности информационной системы. Рассматривается алгоритм шифрования данных в столбцах таблиц реляционной базы данных, который обеспечивает решение следующих задач: а) невозможность расшифровки данных; б) сохранение распределения величин из реальной БД.

1. Introduction

The use of relational databases simulations is used for the purpose of reliable estimation of application of different methods of improving informational system efficiency. The necessity of simulation at the level of relational databases separate tables emerges, for instance, in case of the index structure optimization. For tasks of efficiency improvement, based on the reconfiguration of relational databases (denormalization, application of materialized views) simulation is an effective method for assessment of applied solutions in real practice. Obviously, experiments, related to reconfiguration of active relational databases, are infeasible in most cases. Working with the copy of the database requires access rights, which is not always possible. The issue of ensuring data confidentiality along with the maintenance of reliability of carried out estimations emerges.

Encryption coding is the basic cryptographic technique for information protection, ensuring its confidentiality. Information encryption is the transformation of open data into the encrypted data and vice versa [1]. However, for the simulation purposes it is not required to solve problems of data decoding by virtue as the result of the simulation is not the changed data, but the estimation of efficiency of such model at certain settings. Therefore, it is important to provide an encryption mode to preserve basic relation between variables, stored in database table columns.

2. Data encryption technique

Data encryption technique, consisting of the following steps, is proposed.

1. Generation of group G of interconnected tables based on the analysis of a set of queries, entering into the informational system.
2. Data encryption in group tables.
3. Queries' encryption Q_G to the group.

Encryption procedure shall be completed by the person, having access rights to the data of actual database. The encrypted copy of the database shall be forwarded to the researcher, carrying out simulation of relational database.

The G group shall include tables, participating together with the present table T_i in queries to the database: $T_j \in G$, provided that $\exists Q, T_i \in T_Q, T_j \in T_Q$, where Q – query of the set of queries to the

relational database, subject to a research, T_Q – set of tables, used in the query Q. Also set of queries Q_G shall be generated for the group G: $Q \in Q_G$, provided that $T_Q \bigcap G \neq \varnothing$.

The second stage shall include encryption algorithms, providing maintenance of variables distribution in accordance with column data type. For example, compression or expansion algorithm may be used for numerical values depending on primary distribution of variables in the column.

Let us assume that the column C of numerical value of table T_i is being examined. At first the set of A values shall be defined: $A = A_C \bigcup CA_Q$, provided that A_C – set of values, available for the active database in the C column; CA_Q – set of constant values, used in Q_G queries in column C operations. The highest and the lowest values $a_{\min}$ and $a_{\max}$ respectively shall be selected and excluded from the A set.

Then, the number of repeated values m shall be determined: starting value m=0, m=m+1, provided that $\exists a_i = a_j, i \neq j, 1 = \overline{1..n-1}, j = \overline{i+1, n}$, where n – the number of A set members. The mean interval shall be calculated from the formula

$$\Delta = \frac{a_{\max} - a_{\min}}{n - m}.$$

If $\Delta \geq 16$, the interval compression encryption algorithm shall be used, otherwise - interval extension encryption algorithm.

3. Value interval compression (extension) data encryption algorithm

1. Enter the encryption key k. For the interval compression encryption algorithm $0{,}5 \leq k < 1$, for the interval extension encryption algorithm $\frac{32}{\Delta} \geq k \geq \frac{16}{\Delta}$.

2. $\delta = \Delta \cdot k$.

3. Arrange A set members ascending.

4. $a_t = a_{\min}$.

5. i=1.

6. If $a_i = a_{\min}, a_i \in A$, then $a_i^{'} = a_{\min}, a' = a_i^{'}$. Switch to paragraph 9.

7. If $a_i \neq a_t$, switch to paragraph 9.

8. $a_i^{'} = a'$.

9. i=i+1. If i<n, switch to paragraph 6.

10. $a' = a^{'} + \delta$.

11. $a_t = a_t + d$, where d is the minimum interval between A set values. If $a_t < a_{\max}$, switch to paragraph 5.

12. i=1.

13. For the interval compression encryption algorithm $a_i^{'} = a_i^{'} + \Delta \cdot k \cdot k$, for the interval extension encryption algorithm $a_i^{'} = a_i^{'} + \Delta \cdot \sqrt{k}$.

14. The encryption table entry shall be generated $T_i^K(C) =< a_i, a_i^{'} >$

15. i=i+1. If i<n, switch to paragraph 13.

16. End of the algorithm.

A special case is possible – all A set members are equal: $a_{\min} = a_{\max}, m = n-1, \Delta = 0$. Then one should introduce index (multiplying factor), all set members shall be multiplied by. The index (multiplying factor) may be generated by the random number generator.

Encryption of queries shall be performed by transformation of A_C set members in queries' listing using encryption table $T^k(C)$.

It should be mentioned that "compression" algorithm is considered as the basic encryption algorithm as far as it provides encrypted data layout in the same range as the basic data. But in case the basic data density is so high, that encryption problems may arise, and in the process of table completion simulation, the "extension" algorithm should be chosen.

The mentioned algorithm is simplified as it does not take account of the following:

1) integrity constraints may be determined for the database, which may possibly be violated as a result of data encryption;

2) in case of ties between columns the encryption of columns shall be performed consistently.

4. Sequence of columns processing in the encryption procedure

Conditions for the generation of database model under examination (refer to Figure1):

1) table groups may have crosscuts: $G_i \bigcap G_j \neq \varnothing$;

2) not all groups of the basic database shall be the subject for examination: $G^{DB} \neq G'^{DB}$;

3) data, contained in tables, may be dependent (within limits of different tables or within limits of the same table): $C^T = C^{ND} \bigcup C^D$, where C^{ND} – a set of independent columns, C^D – a set of dependent columns, may be an empty set.

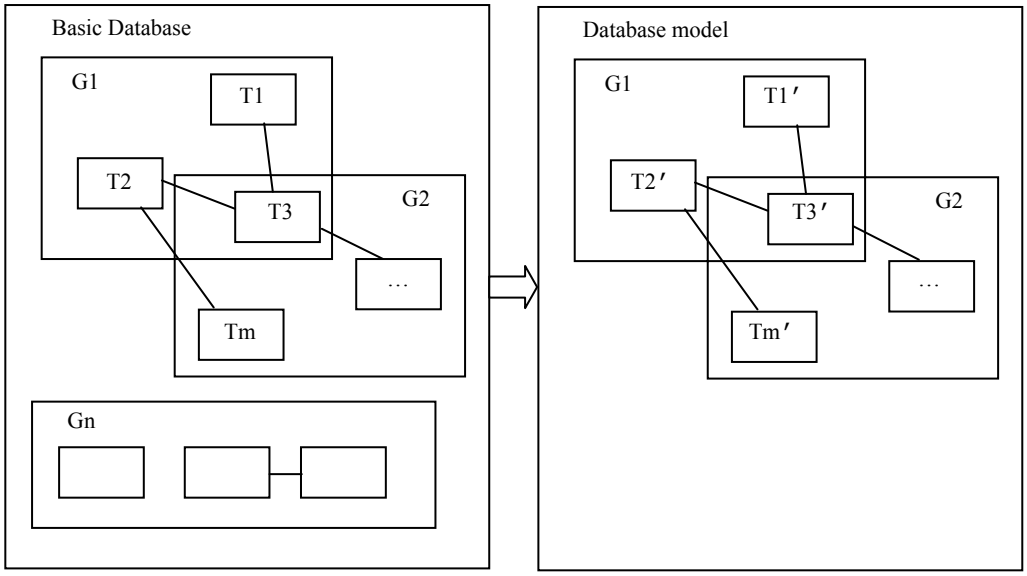


Figure 1. Conditions for the generation of database model

Dependence between columns may be of the following type:

- «parent – subordinate»;
- denormalization dependence .

The «parent – subordinate» dependence shall be determined by the "primary key - foreign key" pair.

Denormalization dependence shall be possible in case of standardized database failure. The reason for denormalization is the necessity of system efficiency improvement. The most frequently used denormalization methods are descending denormalization, ascending denormalization and input of secondary data [2, 3]. Descending denormalization provides for backup of non-key column of the parent table in the detail table; ascending denormalization provides for saving of aggregated

data, obtained on the basis of data from the detail table in the parent table; input of secondary data provides for saving of data, which may be calculated on the basis of data from other columns of the same table, in the shape of separate data column in the table.

Obviously, for detail columns' processing encryption tables of parent columns should be considered.

For this purpose for each dependence *d* the following tuple shall be determined

$$d =< T_{Ch}, C_{Ch}, T_P, C_P, f >,$$

where T_{Ch} – table, for which column dependence exists;

C_{Ch} – column T_{Ch}, for which the dependence is determined;

T_P – parent table, whose column (columns) C_{Ch} depends on;

C_P – set of table columns, data in C_{Ch} depends on;

f – dependence function, $f \in \{Eqv, Agr, Expr\}$.

$f = Eqv$ for «parent – subordinate» dependence and for descending denormalization; means that column encryption table C_P: $T^K(C_{Ch}) = T^K(C_P)$ is used for C_{Ch} processing

$f = Agr$ for ascending denormalization; means that the encryption table for C_{Ch} column shall be calculated using aggregation function $Agr: T^K(C_{Ch}) = F_{Agr}(T^K(C_P))$.

$f = Expr$ for dependence, determined by input of secondary data; similar to the previous case means that the encryption table for C_{Ch} column shall be calculated using preassigned expression $Expr: T^K(C_{Ch}) = F_{Expr}(T^K(C_P))$.

Therefore, the sequence for database table columns encryption may be determined.

1) independent columns C^{ND} for all *G* group tables shall be processed;

2) dependent columns C^D shall be processed in such sequence that $T^K(C_P)$ exists before column C_{Ch} processing.

5. Data structure for encryption procedure

All data, related to groups, database tables, encryption tables and dependences shall be stored on the CD in form of relational relationship, whose structure is shown in Figure 2.

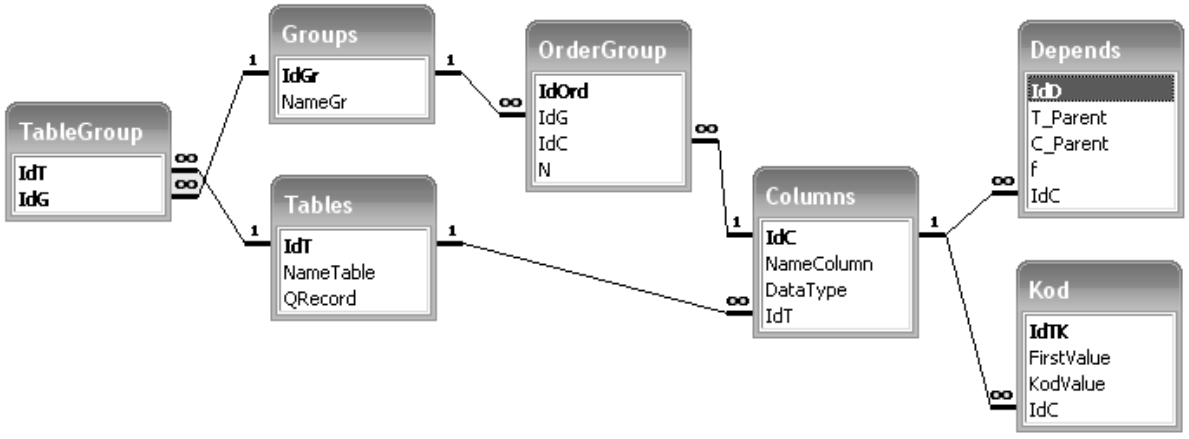


Figure 2. Structure of service database for encryption data storage

The «Tables» table determines sets of database tables. It consists of a unique table identifier, the table name and a number of lines, added to the table. The «Groups» table describes groups of interrelated tables, determined in the process of database queries analysis. The «TableGroup» table determines participation of a table in the group. The «Columns» table describes columns of each table. The «Kod» stores columns' encryption tables. The «Depends» table determines dependencies between columns. «OrderGroup» establishes the sequence for columns' processing.

The present database shall be stored on the PC of the database owner, used only for encryption purposes and deleted after the database model generation is finished.

5. Conclusion

The proposed method allows for significant enhancement of database simulation modeling due to application of model for confidential data. The method performs data encryption, which preserves statistical characteristics, close to those existing for the real database columns, more specifically data type, the relation between values ("equals", "more", "less" etc.) and data distribution characteristics are preserved. Use of this method does not allow to recover initial (base) confidential data.

Application of an elaborated encryption method provides an opportunity to carry out research of database model nearly in the same volume without reduction of results reliability, as in case of coded data.

For test purposes, the program, which made it possible to carry out experiment and to check efficiency of the elaborated method, was developed.

Literature

[1]. Panasenko, S.P. Encryption algorithms. Special quick reference guide. St.-Pb, BHV-Petersburg, 2009, 576 p.

[2] Connoly,T. , K. Begg, A. Strachan. Databases: Design, representation, maintenance. Theory and practice. Moskow, «Williams» publishing house, 2002, 1120 p.

[3] Poolet, Michelle A. Responsible Denormalization. How to break the rules and get away with it. – October 2000 edition of SQL Server Magazine. http://www.sqlmag.com/Article/ArticleID/9785/sql_server_9785.html.

Для контактов:
Проф. к.т.н. Алексей Борисович Кунгурцев
Кафедра системного программного обеспечения
Одесский национальный политехнический университет
E-mail: abkun@te.net.ua
доц. к.т.н. Светлана Леонидовна Зиноватная
Кафедра системного программного обеспечения
Одесский национальный политехнический университет
E-mail: svzino@rambler.ru
Рецензент: доц. д-р инж. П. Антонов, ТУ – Варна

Изисквания за оформяне на статиите на авторите за списание "Компютърни науки и технологии”

I. Ръкописите на статията се представят разпечатани в два екземпляра (оригинал и копие) в размер до 6 страници формат А4 и като файл на адрес: Технически университет – Варна, катедра “Компютърни науки и технологии”, ул. Студентска 1, 9010 Варна.

II. Текстът на статията трябва да включва: УВОД (поставяне на задачата) ИЗЛОЖЕНИЕ (изпълнение на задачата), ЗАКЛЮЧЕНИЕ (получени резултати) , БЛАГОДАРНОСТИ към сътрудниците, които не са съавтори на ръкописа (ако има такива), ЛИТЕРАТУРА и адрес за контакти, включващ: научно звание и степен, име, инициали, фамилия, организация, поделение(катедра), e-mail адрес.

III. Всички математически формули трябва да са написани ясно и четливо (препоръчва се използуване на Microsoft Equation). Номерацията на формулите се дава вдясно от началото на реда. Текста на формулите се позиционира в средата на реда. Експоненциалните функции се изписват чрез знака exp.

1. Текстът трябва да бъде въведен във файл във формат WinWord 2000/2003 със шрифт Times New Roman. Форматирането трябва да бъде както следва:
2. Размер на листа - А4, полета - ляво - 20мм, дясно - 20мм, горно - 15мм, долно - 35мм, Header 12.5мм, Footer 12.5mm (1.25см).
3. Заглавие - (на български език, размер на шрифта 16, bold).
4. Един празен ред, размер на шрифта 14, нормален.
5. Имена на авторите - име, инициали на презиме, фамилия, без звания и научни степени, размер на шрифта 14, нормален.
6. Два празни реда, размер на шрифта 14.
7. Резюме -(на български език, размер на шрифта 11) - до 8 реда, нормален.
8. Заглавие (на английски език, размер на шрифта 12) – bold.
9. Един празен ред, размер 11.
10. Имена на авторите (на английски език, размер на шрифта 11) – нормален.
11. Един празен ред, размер 11.
12. Резюме -(на англиийски език, размер на шрифта 11) - до 8 реда, нормален.
13. Основните раздели на статията (Увод, Изложение, Заключение, Благодарности) се форматират в едноколонен текст както следва:

n. наименование на раздел или на подраздел, размер на шрифта 12, удебелен, центриран.
o. празен ред, размер на шрифта 12;
p. текст, размер на шрифта 12, нормален, отстъп на първи ред на параграф – 10 мм; разстояние от параграф до съседните (Before и After) за целия текст – 0.
q. цитиране на литературен източник - номер на източника от списъка в квадратни скоби;
r. Номерация на формулите =- дясно подравнена, в кръгли скоби.
s. Графики : Формат Layout: In line with text
t. литература – всеки литературен източник се представя по БДС с: номер в квадратни скоби и точка, списък на авторите (първият автор започва с фамилия, останалите – с име), заглавие, издателство, град, година на издаване.
u. За контакти: научно звание и степен, име, презиме (инициали), фамилия, организация, поделение (катедра), e-mail адрес, с шрифт 11, дясно подравнено.

Образец за форматиране (компресиран Word файл) можете да изтеглите от адрес http://kst-forum.netfirms.com/Spisanie_Obrazec.zip

Този брой от списанието се отпечатва със средства от научно-изследователски проект, финансиран целево от държавния бюджет.