

## INFORMACIJE

Strokovno društvo za mikroelektroniko  
elektronske sestavne dele in materiale


## MIDEM

3° 2002

Strokovna revija za mikroelektroniko, elektronske sestavne dele in materiale  
Journal of Microelectronics, Electronic Components and Materials

INFORMACIJE MIDEM, LETNIK 32, ŠT. 3(103), LJUBLJANA, september 2002

## Ceramic Capacitors





No one makes more multilayer ceramic capacitors than Murata, and no one makes them smaller! We span the field from chips you can hardly see to high power capacitors you can hardly lift. Our chip types are progressively replacing other dielectrics, especially plastic film and tantalum.

Why? Because they are smaller, more reliable, more versatile and more available.

As a materials house, Murata has the advantage of making the dielectrics for these capacitors right from the mined materials up. This gives us an unusually high level of control and expertise.

For further information, please visit us at [www.murata.com](http://www.murata.com)

 MIKROIKS d.o.o - 1521 Ljubljana - Stegne 11 - Slovenija

 **muRata**  
*Innovator in Electronics*



## INFORMACIJE

## MIDEM

3 • 2002

INFORMACIJE MIDEM	LETNIK 32, ŠT. 3(103), LJUBLJANA,	SEPTEMBER 2002
INFORMACIJE MIDEM	VOLUME 32, NO. 3(103), LJUBLJANA,	SEPTEMBER 2002

Revija izhaja trimesečno (marec, junij, september, december). Izdaja strokovno društvo za mikroelektroniko, elektronske sestavne dele in materiale - MIDEM.  
Published quarterly (march, june, september, december) by Society for Microelectronics, Electronic Components and Materials - MIDEM.

**Glavni in odgovorni urednik**  
**Editor in Chief**

Dr. Iztok Šorli, univ. dipl.ing.,  
MIKROIKS d.o.o., Ljubljana

**Tehnični urednik**  
**Executive Editor**

Dr. Iztok Šorli, univ. dipl.ing.,  
MIKROIKS d.o.o., Ljubljana

**Uredniški odbor**

Doc. dr. Rudi Babič, univ. dipl.ing., Fakulteta za elektrotehniko, računalništvo in informatiko  
Maribor

**Editorial Board**

Dr. Rudi Ročak, univ. dipl.ing., MIKROIKS d.o.o., Ljubljana  
mag. Milan Slokan, univ. dipl.ing., MIDEM, Ljubljana  
Zlatko Bele, univ. dipl.ing., MIKROIKS d.o.o., Ljubljana  
Dr. Wolfgang Pribyl, Austria Mikro Systeme International AG, Unterpremstaetten  
mag. Meta Lempel, univ. dipl.ing., MIDEM, Ljubljana  
Miloš Kogovšek, univ. dipl.ing., Ljubljana  
Prof. Dr. Marija Kosec, univ. dipl.ing., Inštitut Jožef Stefan, Ljubljana

**Časopisni svet**  
**International Advisory Board**

Prof. dr. Slavko Amon, univ. dipl.ing., Fakulteta za elektrotehniko, Ljubljana,  
PRESEDNIK - PRESIDENT  
Prof. dr. Cor Claeys, IMEC, Leuven  
Dr. Jean-Marie Haussonne, EIC-LUSAC, Octeville  
Dr. Marko Hrovat, univ. dipl.ing., Inštitut Jožef Stefan, Ljubljana  
Prof. dr. Zvonko Fazarinc, univ. dipl.ing., CIS, Stanford University, Stanford  
† Prof. dr. Drago Kolar, univ. dipl.ing., Inštitut Jožef Stefan, Ljubljana  
Dr. Giorgio Randone, ITALTEL S.I.T. spa, Milano  
Prof. dr. Stane Pejovnik, univ. dipl.ing., Fakulteta za kemijo in kemijsko tehnologijo, Ljubljana  
Dr. Giovanni Soncini, University of Trento, Trento  
Prof. dr. Janez Trontelj, univ. dipl.ing., Fakulteta za elektrotehniko, Ljubljana  
Dr. Anton Zalar, univ. dipl.ing., ITPO, Ljubljana  
Dr. Peter Weissglas, Swedish Institute of Microelectronics, Stockholm

**Naslov uredništva**  
**Headquarters**

Uredništvo Informacije MIDEM  
Elektrotehniška zveza Slovenije  
Dunajska 10, 1000 Ljubljana, Slovenija  
tel.: + 386 (0)1 50 03 489  
fax: + 386 (0)1 51 12 217  
e-mail: Iztok.Sorli@guest.arnes.si  
<http://paris.fe.uni-lj.si/midem/>

Letna naročnina znaša 12.000,00 SIT, cena posamezne številke je 3000,00 SIT. Člani in sponzorji MIDEM prejema Informacije MIDEM brezplačno.  
Annual subscription rate is EUR 100, separate issue is EUR 25. MIDEM members and Society sponsors receive Informacije MIDEM for free.

Znanstveni svet za tehnične vede I je podal pozitivno mnenje o reviji kot znanstveno strokovni reviji za mikroelektroniko, elektronske sestavne dele in materiale. Izdajo revije sofinancirajo Ministrstvo za znanost in tehnologijo in sponzorji društva.

Scientific Council for Technical Sciences of Slovene Ministry of Science and Technology has recognized Informacije MIDEM as scientific Journal for microelectronics, electronic components and materials.

Publishing of the Journal is financed by Slovene Ministry of Science and Technology and by Society sponsors.

Znanstveno strokovne prispevke objavljene v Informacijah MIDEM zajemamo v podatkovne baze COBISS in INSPEC.

Prispevke iz revije zajema ISI® v naslednje svoje produkte: Sci Search®, Research Alert® in Materials Science Citation Index™

Scientific and professional papers published in Informacije MIDEM are assessed into COBISS and INSPEC databases.

The Journal is indexed by ISI® for Sci Search®, Research Alert® and Material Science Citation Index™

Po mnenju Ministrstva za informiranje št.23/300-92 šteje glasilo Informacije MIDEM med proizvode informativnega značaja.

Grafična priprava in tisk  
Printed by

BIRO M, Ljubljana

Naklada  
Circulation

1000 izvodov  
1000 issues

Poštnina plačana pri pošti 1102 Ljubljana  
Slovenia Taxe Percue

ZNANSTVENO STROKOVNI PRISPEVKI		PROFESSIONAL SCIENTIFIC PAPERS
A.Bürmen, D.Strle, F.Bratkovič, J.Puhan, I.Fajfar, T.Tuma: Robustno načrtovanje analognih integriranih vezij z uporabo kazenskih funkcij	149	A.Bürmen, D.Strle, F.Bratkovič, J.Puhan, I.Fajfar, T.Tuma: Penalty Function Approach to Robust Analog IC Design
D.Osebik, R.Babič: Možnosti izvedbe adaptivnega FIR sita s programirnimi (FPGA) vezji	157	D.Osebik, R.Babič: The Practicability of Adaptive FIR Digital Filter Implementation with FPGA Circuits
A.Štern, J.Trontelj: Sprejemnik za RFID integrirani bralnik	167	A.Štern, J.Trontelj: An On-chip RFID Receiver Stage
A.Časar, Z.Brezočnik, T.Kapus: Uporaba simboličnega preverjanja modelov pri zaznavanju zatičnih napak v digitalnih vezjih	171	A.Časar, Z.Brezočnik, T.Kapus: Exploiting Symbolic Model Checking for Sensing Stuck-at Faults in Digital Circuits
R.Osredkar, B.Gspan: Pregled planarizacijskih metod v mikroelektronskih tehnologijah	181	R.Osredkar, B.Gspan: Planarization Methods in IC Fabrication Technologies
A.Suhadolnik, J.Petrišič: Merjenje pomika z uporabo odbojnostnih senzorjev z optičnimi vlakni	186	A.Suhadolnik, J.Petrišič: Displacement Measurement Using Optical Fiber Reflection Sensors
F.Pavlovčič, J.Nastran: Zmanjšanje EMI kolektorskih motorjev z optimiranjem prekrivanja lamel komutatorja	189	F.Pavlovčič, J.Nastran: Reducing EMI of Commutator Motors by Optimizing Brush-to-segment Width Ratio
J.Tušek, M.Uran, M.Vovk: Vpliv velikosti okna med elektrodami na proces elektrouporovnega točkovnega varjenja	194	J.Tušek, M.Uran, M.Vovk: Influence of Throat Area on the Resistance Spot Welding Process
M.Rojc, Z.Kačič, I.Kramberger: Strojna implementacija jezikovnih virov za uporabo v vdelanih sistemih	199	M.Rojc, Z.Kačič, I.Kramberger: Hardware Implementation of Language Resources for Embedded Systems
I.Ozimek: Optimalna preslikava algoritmov za hitro izvajanje v sistoličnem polju	204	I.Ozimek: Optimal Algorithm Mapping for Fast Systolic Array Implementations
J.Stergar, B.Horvat: Napovedovanje simboličnih prozodičnih mej z nevronskimi mrežami	213	J.Stergar, B.Horvat: Prediction of Symbolic Prosody Breaks with Neural Nets
S.Krivograd, B.Žalik, F.Novak: Modeliranje inženirskih podatkov s poenostavljanjem in rekonstrukcijo trikotniških mrež	219	S.Krivograd, B.Žalik, F.Novak: Triangular Mesh Decimation and Undecimation for Engineering Data Modelling
<b>APLIKACIJSKI ČLANEK</b>	<b>224</b>	<b>APPLICATION ARTICLE</b>
Izbira med ROM, FASTROM in FLASH spominom za mikrokrmilnik		Selecting Between ROM, FASTROM and FLASH for a Microcontroller
A.Paulin: Misli ob štiridesetletnici prvih magistrov	227	A.Paulin: Forty Years of Master Science Title – Reflections
MIDEM prijavnica	228	MIDEM Registration Form
Slika na naslovnici: Keramične tehnologije iz firme muRata		Front page: Total Ceramic Technology by muRata

**38<sup>th</sup> INTERNATIONAL CONFERENCE  
ON MICROELECTRONICS,  
DEVICES AND MATERIALS**

**and the WORKSHOP on  
PACKAGING AND INTERCONNECTIONS  
IN ELECTRONICS**



**CONFERENCE 2002**

**Lipica, October 09. – 11. 2002**



REPUBLIC OF SLOVENIA  
Ministry of Education,  
Science and Sport



HIPOT-HYB Production  
of Hybrid Circuits d.o.o.  
Šentjernej, Slovenia



Slovenia Chapter



Slovenia Section



# PENALTY FUNCTION APPROACH TO ROBUST ANALOG IC DESIGN

Arpad Bűrmen, Drago Strle, Franc Bratkovič, Janez Puhan, Iztok Fajfar, Tadej Tuma

University of Ljubljana, Faculty of Electrical Engineering, Ljubljana, Slovenia

**Key words:** circuit sizing, analog IC, optimization, penalty function, CAD.

**Abstract:** Automating the robust IC design process is becoming more and more important due to its complexity and decreasing time to market. In order for the circuit to be robust it must satisfy all design requirements across a range of operating conditions and manufacturing process variations. Part of the design process, which is performed by experienced analog IC designers, is automated. A transformation of the robust design problem into a constrained optimization problem by means of penalty functions is presented. The method is illustrated on a robust differential amplifier design problem. The results show that it is capable of sizing a circuit and reaching comparable or to some extent even superior performance to a humanly designed circuit. The method has great potential in parallel processing although it is efficient enough to be executed on a single computer.

## Robustno načrtovanje analognih integriranih vezij z uporabo kazenskih funkcij

**Ključne besede:** dimenzioniranje vezij, analogna integrirana vezja, optimizacija, kazenske funkcije, računalniško podprto načrtovanje.

**Izvleček:** Avtomatizacija postopka robustnega načrtovanja IV postaja vse bolj pomembna zaradi zahtevnosti samega postopka in čedalje krajšega časa od začetka načrtovanja do pojave vezja na tržišču. Da je vezje robustno, mora zadostiti vsem načrtovalskim zahtevam za dano območje pogojev delovanja in možnih variacij parametrov postopka izdelave. Predstavljen je avtomatiziran postopek načrtovanja, po zgledu postopka, ki ga izvajajo načrtovalci IV. Podana je preslikava iz problema robustnega načrtovanja v omejen optimizacijski problem. Pri tem se poslužujemo kazenskih funkcij za definicijo kriterijske funkcije. Uporaba metode je prikazana na robustnem načrtovanju diferencialnega ojačevalnika. Rezultati kažejo, da je metoda sposobna poiskati nabor parametrov vezja, ki da primerljivo ali pa do neke mere celo boljše vezje kot ga načrtuje človek. Pristop ima velik potencial v vzporednem računanju, a je kljub temu dovolj učinkovit, da lahko pridemo do sprejemljivih rezultatov z uporabo enega samega računalnika.

### 1 Introduction

A major issue in analog IC design is robustness. A robust design satisfies the design requirements in all foreseen operating conditions. Furthermore, a robust design must fulfil all design requirements regardless of the expected process variations that may occur during the fabrication of the designed IC. As the time-to-market becomes shorter automating the design process is becoming an important task [1].

By design requirements we mean circuit characteristics which are of importance to the user of the designed circuit and can be expressed by real values, such as gain, phase margin, gain-bandwidth product, common mode rejection ratio, distortion, output rise time, input impedance, current consumption, etc. A circuit fulfils the design requirements if all circuit characteristics, which are of importance to its user, lie inside some predefined intervals.

An IC must fulfil the design requirements in various operating conditions, which also include various environmental effects. Some common operating conditions whose variations can cause improper circuit operation are power supply voltage, bias currents and load characteristics. The most common environmental condition that affects the operation of a circuit is the temperature. In order to obtain a

robust design the circuit must fulfil the design requirements for a given range of operating conditions.

Process variations are another reason speaking in favour of robust design. IC manufacturers describe process variations by so called corner models. Corner models describe several extreme conditions, which can occur during IC fabrication and result in some extreme circuit behaviour. For a CMOS process usually 4 different corner models are provided to the designer: worst one (WO), worst zero (WZ), worst power (WP) and worst speed (WS). Beside corner models, IC manufacturers also supply a typical mean (TM) model.

If robustness is not foreseen at the design stage and already incorporated in the design, one can expect that only a small number of fabricated ICs will fulfil the design requirements at nominal operating conditions due to process variations. Furthermore only a fraction of these ICs will fulfil the design requirements in all foreseen operating conditions.

In the past a lot of effort was invested in finding efficient means of automated nominal design (/2/, /3/, /6/, /5/, /6/). Nominal design however does not produce robust circuits. The resulting circuits satisfy the design requirements only in nominal operating conditions and for the typ-

ical process. In order to obtain a robust circuit and additional step of design centering is required. Design centering techniques are either statistical (/7/, /8/) or deterministic (/9/, /10/, /11/).

The whole idea of robust design (as sometimes practised by IC designers) relies on the assumption, that the circuit characteristics reach their extreme values at points where the operating conditions and process variations take their so-called corner values. In order to establish, whether the design is robust, designers examine the performance of the circuit for all combinations of corner values. Every such combination represents a corner point of the design.

The number of corner points can be large. Beside 4 corner points for MOS transistors (result of the process variations), every operating condition brings along at least two extreme values - the minimal and the maximal value. For the operating temperature IC designers usually examine more than the two extreme values. The same can also be the case for other operating conditions and process variations.

The reason why one examines the circuit for more than only the extreme operating conditions is the fact that the circuit characteristics are not necessarily monotonic functions of operating conditions and process variations. When these functions are not monotonic, the probability of making a wrong conclusion increases with the distance between individual corner points. By examining the circuit at a larger number of "corner" points this distance is decreased.

In order to obtain a robust design an IC designer varies the dimensions of individual transistors and other elements of the design until the design fulfils the requirements in all relevant corner points. Whether or not a particular design is robust can be examined by simulating it at those corner points. If one examines the circuit for all combinations of 5 MOS corners, 3 temperature corners and 2 power supply voltage corners, a total of 30 corners must be examined.

IC designers practise robust design by iterating corner point simulation and circuit parameter adjustments for selected structure (topology). Obviously the only part of this process where the computer plays a role is the simulation. The parameter adjustment is still performed by the designer manually and is based on knowledge and past experience. One way of automating the process of parameter adjustment is the transformation of the robust design problem, as perceived by the IC designer, into a (constrained) optimization problem. There exist many algorithms for solving (constrained) optimization problems that can be applied to solve the IC designer's robust design problem.

The remainder of this paper is organised as follows: first the robust design method is mathematically formulated. A short introduction to optimization is given upon which the relationship between robust design and cost function used in the process of optimization is established. The cost func-

tion is divided in two parts: penalties for circuits that cause the simulator to fail at evaluating the circuit and penalties arising from design requirements. The use of the method is illustrated on a robust amplifier design problem. Finally the conclusions and ideas for future work are given.

## 2 Design Methodology

### 2.1 Circuit design and corner points

The robust design process as perceived and practised by an IC designer is based on the notion of corner points. A corner point is a combination of some process variation and  $M$  operating conditions. Suppose that we have a set of possible process variations

$$P_0 = \{p_0^1, \dots, p_0^{n_0}\} \quad (1)$$

and for every operating condition a set of values that are of particular interest to the designer

$$P_i = \{p_i^1, \dots, p_i^{n_i}\} \quad i = 1, \dots, M \quad (2)$$

$p_0^1$  stands for the characteristics of the nominal IC fabrication process and  $p_1^1, p_2^1, \dots, p_M^1$  for the nominal operating conditions. The cross product of  $M + 1$  sets from (1) and (2) is the set of corner points  $C$ . In general a subset of these points is examined during the process of robust design

$$C = P_0 \times P_1 \times \dots \times P_M \quad (3)$$

The number of corners is

$$K = \prod_{i=0}^M n_i \quad (4)$$

The performance of the circuit, (which is the result of some combination of process variations during its fabrication and operating conditions during its use), is described by a vector of  $N$  real values  $\underline{y} = [y_1, \dots, y_N] \in R^N$ .

We represent the circuit as a function that for any combination of  $n$  circuit parameters denoted by vector  $\underline{x}$  and some combination of process variations and operating conditions denoted by  $q$  produces a vector of circuit characteristics  $\underline{y}$ .

$$\begin{aligned} D: (\underline{x}, q) &\mapsto \underline{y} & \underline{x} \in R^n, q \in C, \underline{y} \in R^N \\ y(\underline{x}, q) &= [y_1(\underline{x}, q), y_2(\underline{x}, q), \dots, y_N(\underline{x}, q)] = \\ &[D_1(\underline{x}, q), D_2(\underline{x}, q), \dots, D_N(\underline{x}, q)] \end{aligned} \quad (5)$$

In the subsequent sections we also use the following notation for (5):



$$D_i : (\underline{x}, q) \mapsto y_i \quad \underline{x} \in R^n, q \in C, y_i \in R$$

Two vectors express the design requirements: a vector of lower bounds  $\underline{b} = [b_1, \dots, b_N] \in R^N$  and a vector of upper bounds  $\underline{B} = [B_1, \dots, B_N] \in R^N$ . For the sake of simplicity we allow for any lower bound to take the value  $-\infty$ , meaning that there is no lower bound on the respective circuit characteristic. Similarly any upper bound can take the value  $+\infty$ , meaning that no upper bound exists on the respective circuit characteristic. A circuit with circuit parameters  $\underline{x}$  satisfies the design requirements for a particular corner point  $q \in C$  if the following set of relations holds:

$$b_i \leq y_i \leq B_i \quad i = 1, \dots, N \quad (6)$$

Let  $g(x)$  denote some continuous monotonically increasing function defined for  $x \geq 0$ . Define a new function:

$$f(x) = \begin{cases} 0 & x < 0 \\ g(x) - g(0) & x \geq 0 \end{cases} \quad (7)$$

(7) is used to establish the relation between the robust design problem and the constrained optimization problem.

A circuit design is satisfactory if it satisfies the design requirements for all corner points from set  $C$ .

## 2.2 Constrained optimization

Problems of the form

$$\underline{x}_o = \min_{\underline{x} \in S} r(\underline{x}) \quad S \subseteq R^n$$

are  $n$ -dimensional unconstrained global optimization problems,  $\underline{x}_o$  is the global optimum and  $r(\underline{x})$  is a cost function. Most unconstrained optimization methods search merely for a local optimum, where the following relation holds:

$$\nabla r(\underline{x}) = 0$$

If the search space is constrained, i.e.  $S \subset R^n$ , the problem becomes a constrained optimization problem. The notion of global optimum remains unchanged, but the definition of local optimum changes.

The search space in constrained optimization is defined by means of constraints. In general two kinds of constraints exist. Explicit constraints have the form  $b \leq x_i \leq B$  where  $x_i$  can be any component of  $\underline{x}$ . More complex relations define implicit constraints like  $h(\underline{x}) \geq 0$  or  $h(\underline{x}) = 0$ . The former one is an inequality constraint and the latter one is an equality constraint. Note, that  $h(\underline{x})$  can be any function. Handling implicit constraints is more complicated than handling explicit constraints.

When optimizing integrated circuits, the vector of optimized parameters  $\underline{x}$  includes mostly circuit parameters like element widths and lengths, although in some cases also cur-

rent, frequency, resistance and other values can be among optimized parameters. Explicit constraints are mostly used for setting the limits imposed by the technology like minimum dimensions. Another possible use of explicit constraints is to force a parameter to remain in a particular interval, e.g. one could restrict the transistor width of a differential pair to stay above some given value. Explicit equality constraints can be used to impose a fixed dependence of a parameter on some subset of circuit parameters. Such constraints are easily enforced during optimization. More complex explicit constraints (i.e. explicit constraints on circuit characteristics, explicit inequality constraints) are also possible. Nevertheless one should keep in mind that a large number of more complex explicit constraints could in practice reduce the performance of an optimization algorithm.

Another important thing to note regarding optimization algorithms is that in practical cases they do produce a decrease in the cost function value when compared to the initial value. But in general, a large amount of computing time and resources has to be invested in order to find the global optimum of an optimization problem. Generally one is satisfied if:

- an optimization algorithm provides an improvement over the best economically justified human design,
- (at least partially) solves some problem without human intervention or
- helps the designer to speed up the design process.

In the past many efficient optimization algorithms that relied on the cost function value along with the values of its derivatives were developed. Since the sensitivity information is generally not available from circuit simulators (at least not to the extent required to calculate the partial derivatives of the cost function), one must rely to a different class of methods. Direct search methods [12] rely only on cost function value and require no derivative information from the simulator. They are the methods of choice in this work.

## 2.3 Penalty function for enforcing constraints on circuit performance

In order to exploit optimization for robust circuit design a cost function has to be defined. The cost function is supposed to rank the set of possible designs thus making it ordered. Throughout the optimization all designs have the same structure (topology). Only the nominal circuit parameter values ( $\underline{x}$ ) are varied. Consider the following penalty function:

$$F(\underline{y}) = \sum_{i=1}^N \left( f\left(\frac{y_i - B_i}{A_i}\right) + f\left(\frac{b_i - y_i}{A_i}\right) \right) \quad (8)$$

Function (8) penalises any design with one or more characteristics lying outside the intervals defined by the respective lower and upper bounds on circuit performance. The

penalty is proportionate to the distance from the boundary of the interval. For a design which characteristics lie inside the intervals defined by  $\underline{b}$  and  $\underline{B}$ , the function returns 0. Note that the penalty function applies to the circuit characteristics for a particular corner point.

Since "bad" designs are associated with higher values of the penalty function and "good" designs are associated with 0, the definition of a cost function (which will in turn be minimised by the optimization algorithm) is right at hands:

$$r_E(\underline{x}) = \sum_{i=1}^K F(D(\underline{x}, q_i)) \quad (9)$$

One can stop the optimization algorithm as soon as (9) reaches 0, since the algorithm found a point in the search space  $\underline{x}_0$  for which the corresponding design satisfies all performance constraints (6) in all corners. Furthermore, if the algorithm has a way of detecting the existence of a neighbourhood of  $\underline{x}_0$  where corresponding designs are all satisfactory, one can tell that the design requirements are too "loose". Ideally the design requirements should be so tight that every satisfactory point in the search space has no neighbourhood where all designs fulfil the design requirements. In such case one could be assured that the capabilities of the technology are fully exploited for the particular circuit structure.

## 2.4 Heuristic corner search

In previous section robust design was achieved by checking the circuit performance in all relevant corner points of the design (3). Since the total number of corner points grows exponentially with the increasing number of operating conditions (4), the analysis of circuit performance becomes intractable. Approaches for reducing the number of analysed corner points become of interest where one replaces the search through the complete set of corners  $C$  by its subset  $C_s = \{s(i) : i = 1, \dots, K_H\} \subset C$ . Consequently the number of checked corners is reduced to  $K_H = |C_s| < K$  and the corresponding term in the cost function becomes:

$$r_H(\underline{x}) = \sum_{i=1}^{K_H} F(D(\underline{x}, s(i))) \quad (10)$$

Several different heuristics can be defined for choosing the set  $C_s$ . The method of choice in this paper first examines the individual influences of operating conditions. The collected information is used for predicting the corners where circuit characteristics are expected to reach their extreme values, upon which those corners are examined. In the first part the following set of corners is examined:

$$q_{nom} = s_0^1 = s_1^1 = \dots = s_M^1 = (p_0^1, p_1^1, \dots, p_M^1)$$

$$s_i^i = (p_0^i, p_1^i, \dots, p_M^i) \quad i = 2, \dots, n_0$$

$$s_1^i = (p_0^1, p_1^i, \dots, p_M^1) \quad i = 2, \dots, n_1 \quad (11)$$

...

$$s_M^i = (p_0^1, p_1^1, \dots, p_M^i) \quad i = 2, \dots, n_M$$

Based on the results obtained for these corners, further  $2N$  corners are generated (two for every circuit characteristic; one where the lowest value and one where the highest value is expected to take place) and examined:

$$q_L^i = (p_0^{l_i^0}, p_1^{l_i^1}, \dots, p_M^{l_i^M}) \quad q_H^i = (p_0^{h_i^0}, p_1^{h_i^1}, \dots, p_M^{h_i^M})$$

$$i = 1, \dots, N$$

$$l_i^j = \arg \min_{k=1, \dots, n_j} y_i(\underline{x}, s_j^k) \quad h_i^j = \arg \max_{k=1, \dots, n_j} y_i(\underline{x}, s_j^k)$$

$$i = 1, \dots, N \quad j = 0, \dots, M \quad (12)$$

By searching through corners defined by (11) and (12) we

need to check only  $K_H = \sum_{i=0}^M n_i - M + 2N + 1$  corners. The

price to pay is the risk of obtaining a narrower range for the circuit characteristic  $y_i$  in case the function  $D_i(\underline{x}, q)$  is not monotonic with regard to the intervals enclosing operating conditions and intervals enclosing model parameters of process variations.

## 2.5 Cumulative cost function

The cumulative cost function  $r(\underline{x})$  equals (9) (or (10) if heuristic corner search is used). This causes the optimizer to search for a circuit satisfying all design requirements. The optimization can be stopped as soon as some point where  $r(\underline{x})=0$  is found. One also has to consider the case that the simulation itself fails to converge thus rendering the optimization incapable of determining the cost function value for a particular combination of circuit parameters. Besides that the simulator may succeed to simulate certain circuits, but the performance of these circuits is far from the desired performance (e.g. some of the transistors that are supposed to be in saturation, are not). To resolve the problem an additional penalty term  $r_c(\underline{x})$  is introduced into the cumulative cost function. The value of  $r_c(\underline{x})$  for such circuits should be significantly larger than the contribution of the penalty functions  $r_E(\underline{x})$  (or  $r_H(\underline{x})$ ). The additional penalty should be proportionate to the severity of the convergence problem (circuit performance problem).

Applying optimization to the cumulative cost function can solve the robust design problem. Any box-constrained optimization method can be used. The reason due to which box constraints are sufficient is the fact that we only need to constrain circuit parameters such as transistor widths and lengths to intervals of possible values. The implicit constraints arising from the design requirements are handled by the penalty functions.



### 3 Results

To illustrate the method, robust design has been applied to the circuit structure in Figure 1 /18/. The circuit is an amplifier with differential input, differential output and common mode feedback. The  $M$  and  $W/L$  values of transistors in Figure 1 (reference circuit) were designed by an IC designer.

Since the **pd** signal is kept low throughout normal operation so inverter Inv1 and transistors M1 and M2 are irrelevant to the design. An external current source pulls  $16\mu\text{A}$  from the **bias** input in order to set the operating point of the circuit. During normal operation  $V_{dda}$  is set to 5V and  $V_{ssa}$  to 0V. The **agnd** input voltage is in the middle between  $v_{dda}$  and  $v_{ssa}$  since it is the analog reference level. The differential input is at  $v(\text{inp}, \text{inn})$ , whereas  $v(\text{outp}, \text{outn})$  constitutes the differential output. Ideally the **cmf** input is kept at  $(v(\text{outp})+v(\text{outn}))/2$ .

In the circuit there are several groups of transistors whose dimensions are mutually dependent. Their ratios were kept constant throughout the search. A similar approach can be found in /13/. The lengths of transistors M4-M11 are identical. The ratios of widths for these transistors M4-M11 are also kept constant since they constitute the current mirrors that set the operating point of the circuit. The same goes for M13-M22. The widths of M3 and M12 are adjusted according to designer's experience with regard to the  $W/L$  ratios of M4 and M13. Transistors M23, M24 must have the same widths. The same goes for M25 and M26. Transistors in both differential pairs must also be of the same width (M27-M28 and M29-M30). In the automated design process the same values for  $M$  were used as in figure 1.

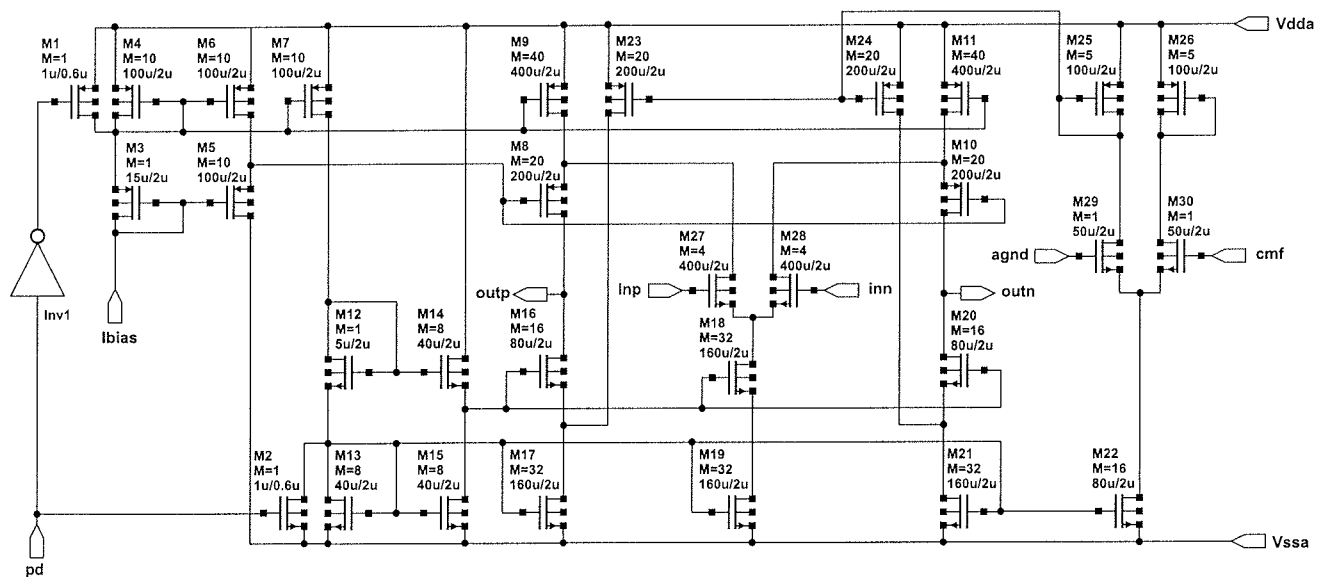


Figure 1: The differential amplifier circuit taken from a real world application.  $W/L$  and  $M$  values were designed by an IC designer.

### 3.1 Design requirements

Note that  $V_{ds}$  and  $V_{dsat}$  denote the drain-source voltage and the drain-source saturation voltage. For p-MOS they represent the absolute values of respective quantities. Refer to Figure 2 for the test circuit.

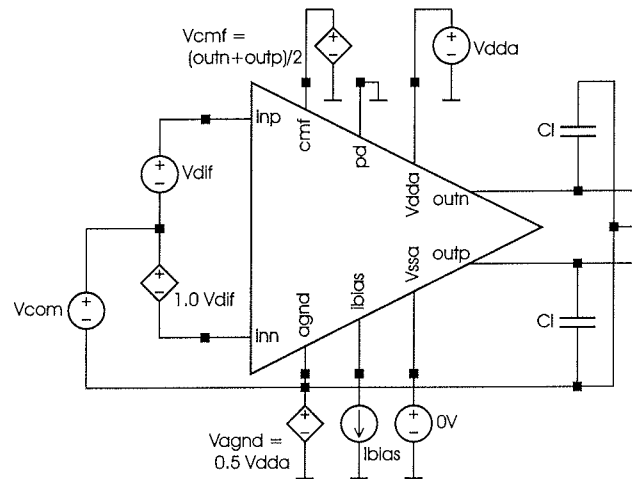


Figure 2: Test setup for the circuit in figure 1.

First of all we require that for the operating point of all transistors except M1, M2 and the transistors in Inv1  $V_{ds} > V_{dsat} + 0.005$  holds in all examined corners. Let  $M_{rel}$  denote the set of all relevant MOS transistors. The saturation measure is defined as

$$P_{sat} = \sum_{M \in M_{rel}} \text{ramp}(V_{dsat}(M) + 0.005 - V_{ds}(M))$$

Next the offset voltage (i.e. the common mode output voltage at  $V_{dif} = 0$ ,  $V_{com} = 0$ ) is measured.

The linear range is defined as the percentage of the maximal output voltage range  $[V_{ssa} - V_{dda}, V_{dda} - V_{ssa}]$  where the differential amplification is above 1/2 of its maximum value. The common mode range is measured by keeping

the input differential voltage **Vdif** at 0, sweeping the input common mode voltage **Vcom** and observing the  $V_{ds} - V_{dsat}$  difference for all transistors in  $M_{rel}$ . The lowest and the highest value of the input common mode voltage  $(V_{imp} + V_{inn})/2$  are measured where  $V_{ds} > V_{dsat}$  holds for all transistors in  $M_{rel}$ .

In the AC analysis (transfer function from (**inp**, **inn**) to (**outp**, **outn**)) the gain at 0Hz, phase margin (difference to 180° at 0dB gain) and the frequency where gain falls to 0dB are measured. Noise analysis is performed with output at (**outp**, **outn**) and input at **Vdif**. Input noise spectrum density is measured at two frequencies: 10Hz ( $n_1$ ) and 1kHz ( $n_2$ ).

The measure of the amplifier area is defined as the sum of WL products for all transistors in  $M_{rel}$ .

### 3.2 The set of corner points

A total of 5 CMOS corners arising from random process variations were examined along with the corners for temperature,  $V_{dda}$ ,  $I_{bias}$ , and  $C_1$ . See Table 1 for the complete list of examined values. A total of 405 corners for the exhaustive corner search and  $13+20=33$  corners for the heuristic corner search must be examined.

	Nominal	Extreme values
MOS corners	TM	WO, WZ, WP, WS
Temperature	25°C	-40°C, 125°C
Power supply	5V	4.5V, 5.5V
Bias current	16uA	13.6uA, 18.4uA
Load capacitance	6pF	4.2pF, 7.8pF

Table 1: Corners of the design.

Table 2 lists the design requirements (lower and upper bounds on individual circuit characteristics).

### 3.3 Results of optimization experiment

The optimizer tried to find a solution starting from a design that didn't work (all widths were 20μm, lengths 2μm and M3/M4 (M12/M13) width ratios were 0.2). 12 parameters

were optimized. The range for transistor dimensions was 0.6μm to 1000μm for widths (5 parameters), 0.6μm to 3μm for lengths (5 parameters), and 0.01 to 1.0 for M3/M4 (M12/M13) width ratios (2 parameters).

Additional penalty terms ( $r_c(x)$ ) were introduced in the following cases:

1. In case a failure in initial OP analysis occurred penalty of  $10^6$  was added. The offset was set to 10V and the remaining analyses (DC analyses, AC analysis and NOISE analysis) were skipped for the particular corner. All problems encountered in this analysis would reoccur in all other analyses since OP analysis precedes or is included in any other type of analysis.
2. In case a failure in the differential mode DC sweep analysis occurred the linear range was set to 0%.
3. In case of a failure in the common mode DC sweep analysis the lower (upper) bound for the common mode range was set to +5V (-5V).
4. In case the AC analysis failed, 0Hz gain, phase margin and 0dB frequency were set to 0.
5. In case the NOISE analysis failed  $n_1$  ( $n_2$ ) was set to  $10^{-4} V / \sqrt{Hz}$  ( $10^{-5} V / \sqrt{Hz}$ ).
6. If any of the failures from cases 1-5 occurred in the first part of the heuristic search, the second part of the search was skipped with additional penalty of  $10^9$ .
7. In case of a failure in OP analysis (case 1) when the remaining analyses were skipped for a particular corner, circuit characteristics that were supposed to result from the skipped analyses were set to the values mentioned in cases 2-5.

SPICE was used as the circuit simulator /14/. The optimization method (/15/, /16/) was a modified constrained simplex method based on /17/. The results are summarized in Tables 3-5. The optimization was stopped as soon as some circuit with cost function value less or equal 0 was found.

	Requirements		
	$b$ (min)	$B$ (max)	$A$ (1 penalty point per)
Sat. measure	$-\infty$	0	0.001m
Offset voltage	0 (or $-\infty$ )	50mV	1mV
Linear range	73%	100% (or $+\infty$ )	0.1%
CM range (low)	$-\infty$	-1.2V	1mV
CM range (high)	1.2V	$+\infty$	1mV
0Hz gain	60dB	$+\infty$	1dB
Phase margin	50°	180° (or $+\infty$ )	1°
0dB frequency	7.0 MHz	$+\infty$	0.1MHz
Noise at 10Hz	$-\infty$	620 nV / $\sqrt{Hz}$	100 nV / $\sqrt{Hz}$
Noise at 1kHz	$-\infty$	62 nV / $\sqrt{Hz}$	10 nV / $\sqrt{Hz}$
Area	0 μm <sup>2</sup> (or $-\infty$ )	8300 μm <sup>2</sup>	100 μm <sup>2</sup>

Table 2: Design requirements.



	Lowest	Nominal	Highest
Offset voltage	0.195mV	5.5mV	32.7mV
Linear range	74.0%	79.4%	81.6%
CM range (lo.)	-1.65V	-1.40V	-1.15V
CM range (hi.)	3.45V	3.95V	4.45V
OHZ gain	61.6dB	74.0dB	77.3dB
Phase margin	56.2°	62.8°	74.5°
OdB freq.	8.23MHz	13.1MHz	16.8MHz
Noise at 10Hz	332nV /√Hz	386nV /√Hz	599nV /√Hz
Noise at 1kHz	33.7nV /√Hz	39.3nV /√Hz	60.8nV /√Hz
Area	8240μm <sup>2</sup>	8240μm <sup>2</sup>	8240μm <sup>2</sup>

Table 3: Performance of the reference circuit over the set of corners examined by the heuristic search.

The results in Tables 3 and 4 represent the reference circuit's performance and the computer-designed circuit's performance. In the nominal operating conditions the circuit resulting from the optimization run has worse offset voltage. The upper boundary of common mode range, gain and noise are slightly worse. Linear range, lower boundary of common mode range, phase margin, frequency range and circuit area were better than for the reference circuit.

In the respective worst corners (as seen from the standpoint of the heuristic search) the computer-designed circuit has a slightly worse offset voltage and upper boundary of the common mode range. Linear range, lower boundary of common mode range, gain, phase margin, frequency range and noise are better than for the reference circuit.

	Lowest	Nominal	Highest
Offset voltage	24.8mV	34.4mV	38.9mV
Linear range	78.3%	83.1%	85.1%
CM range (lo.)	-1.65V	-1.45V	-1.20V
CM range (hi.)	3.35V	3.85V	4.35V
OHZ gain	72.0dB	73.2dB	74.4dB
Phase margin	62.0°	68.5°	75.0°
OdB freq.	10.2MHz	17.0MHz	23.2MHz
Noise at 10Hz	379nV /√Hz	443nV /√Hz	571nV /√Hz
Noise at 1kHz	38.3nV /√Hz	44.8nV /√Hz	57.8nV /√Hz
Area	7810μm <sup>2</sup>	7810μm <sup>2</sup>	7810μm <sup>2</sup>

Table 4: Results of the automated design process over the set of corners examined by the heuristic search.

Since the main goal of robust design is to obtain a circuit whose worst-case characteristics are as good as possible, the comparison of worst case performance is of higher relevance than the comparison of nominal performance. The key result of the experiment is not merely the proof of computer's ability to outperform a human designer, but also the fact that the computer can size a circuit with little prior knowledge of it. Only the circuit's structure, performance constraints, some penalties for circuits that don't simulate and the bias current ratios ( $M$  parameter values) were pre-

defined in the experiment. To supply such knowledge an experience of a senior designer is still required. Table 5 summarises the resulting transistor dimensions with respect to the reference design.

We expect that by replacing the device models (i.e. replacing 0.6-micron process models with 0.35-micron process models) and executing an optimization run, automated technology migration can be achieved [19]. The applicability of our method to technology migration is to be examined in our future work.

## 4 Conclusions

The robust IC design methodology applied by IC designers in their everyday work has been mathematically formulated. A general cost function approach utilising penalty functions for describing the robust IC design problem has been proposed. Penalty functions for circuits that can't be simulated were used to guide the search away from regions of search space that can't be analysed. In order to reduce the number of examined corners a heuristic search method for determining the minimal and maximal values of circuit characteristics has been used. Robust design is achieved by minimising the cumulative cost function. In order to achieve this some box constrained optimization method can be used. In our experiment the modified constrained simplex method was used due to its performance in past studies.

The automated design method was tested on an amplifier design problem. The computer attempted to design the circuit without a working initial point and with wide intervals for transistor lengths and widths.

The optimization run resulted in an overall better circuit when results were compared to the nominal circuit's performance. A bigger difference was observed when comparing the worst characteristic values of computer-designed circuits to the reference circuit. The computer-designed circuit generally outperformed the reference circuit, except in the offset voltage. The offset voltage however was more uniform for the computer-designed circuit (from 24.8mV to 38.9mV) than for the reference circuit (0.2mV to 32.7mV). This means that the dependence of the offset voltage on environmental effects is lower for the computer-designed circuit.

The experiment was run on a 450MHz Intel Pentium III computer with 128MB of RAM. Since the computer was running other tasks beside the optimization itself, the timing results may be somewhat higher than they could be. The optimization took 18 hours. 410 circuits were evaluat-

	W <sub>29</sub>	L <sub>29</sub>	W <sub>27</sub>	L <sub>27</sub>	W <sub>14</sub>	L <sub>14</sub>	W <sub>8</sub>	L <sub>8</sub>	W <sub>5</sub>	L <sub>5</sub>	r <sub>wl(3,4)</sub>	r <sub>wl(12,14)</sub>
Reference	50u	2.0u	400u	2.0u	40u	2.0u	200u	2.0u	100u	2.0u	0.150	0.125
Computer	165u	1.3u	343u	1.6u	43u	1.7u	558u	2.0u	71u	1.0u	0.100	0.138

Table 5: Results of the automated design process -transistor dimensions and ratios.

ed. An average circuit evaluation took 158s (4.8s per corner). If we take into account the fact that the state-of-the-art PC desktop computer nowadays is about 5 times faster, the optimization run would complete in 3.6 hours. Further acceleration is expected to be achieved by doing the corner analyses for several corners in parallel. The acceleration could reach

$$S = \min \left( \sum_{i=0}^M n_i - M + 1, 2N \right)$$

when the aforementioned heuristic search would be used. In the two examined cases we could expect speedups of up to 13. The achievable speedup would of course be smaller due to the synchronisation penalty. Speedups of 2-3 could be easily achieved by using a cluster of 4-5 workstations. This would bring the optimization time down to 1-2 hours for the sample circuit.

There remain several possible applications of the method to be examined in the course of future research:

- Incorporating design optimization into the method,
- Technology migration of existing designs to newer technologies (e.g. 0.6-micron to 0.35-micron migration),
- Tuning existing designs as they are reused in newer ICs in order to improve their (worst case) performance (and reduce the occupied silicon area or power consumption),

A great benefit is expected from parallel processing. Multiple corner points can be analysed in parallel. Furthermore, different types of analysis for the same corner point can also be executed in parallel. Finally a parallel optimization method (/20/, /21/) can be applied to minimise the cumulative cost function. Such multilevel parallelism could exploit the power of large clusters of workstations without utilising parallelism at the simulation level and thus take advantage of the same (thoroughly tested) simulation algorithms as those currently used in IC design.

## 5 References

- /1/ Gielen, G. G. E., Rutenbar, R. A., *Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits*. Proceedings of the IEEE, vol. 88, no. 12, pp. 1825-1854, 2000.
- /2/ del Mar Hershenson, M., Boyd, S. P., Lee, T. H., *Optimal Design of a CMOS Op-Amp via Geometric Programming*, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 20, no. 1, pp. 1-21, 2001.
- /3/ Phelps, R., Krasnicki, M., Rutenbar, R. A., Carley, L. R., Hellums, J. R., *Anaconda: Robust Synthesis of Analog Circuits via Stochastic Pattern Search*. Proceedings of the IEEE 1999 Custom Integrated Circuits Conference, pp. 567-570, 1999.
- /4/ Krasnicki, M., Phelps, R., Rutenbar, R. A., Carley, L. R., *Maelstrom: Efficient Simulation-Based Synthesis for Custom Analog Cells*. Proceedings 1999 Design Automation Conference, pp. 945-950, 1999.
- /5/ Phelps, R., Krasnicki, M., Rutenbar, R. A., Carley, L. R., Hellums, J. R., *Anaconda: Simulation-Based Synthesis of Analog Circuits via Stochastic Pattern Search*, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 19, no. 6, pp. 703-717, 2000.
- /6/ Schwencker, R., Schenkel, F., Gräb, H., Antreich, K., *The Generalized Boundary Curve - A Common Method for Automatic Nominal Design and Centering of Analog Circuits*. Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, pp. 42-47, 2000.
- /7/ Aftab, S. A., Styblinski, M. A., *IC Variability Minimization Using a New Cp and Cpk Based Variability/Performance Measure*. 1994 IEEE International Symposium on Circuits and Systems, vol. 1, pp. 149-152, 1994.
- /8/ Keramat, M., Kielbasa, R., *OPTOMEGA: an Environment for Analog Circuit Optimization*. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, vol. 6, pp. 122-125, 1998.
- /9/ Abdel-Malek, H., Hassan, A., *The Ellipsoidal Technique for Design Centering and Region Approximation*. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 10, no. 8, pp. 1006-1014, 1991.
- /10/ Dharchoudhury, A., Kang, S. M., *Worst-Case Analysis and Optimization of VLSI Circuit Performances*. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 14, no. 4, pp. 481-492, 1995.
- /11/ Krishna, K., Director, S. W., *The Linearized Performance Penalty (LPP) Method for Optimization of Parametric Yield and Its Reliability*. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 14, no. 12, pp. 1557-1568, 1995.
- /12/ R. M. Lewis, V. Torczon, M. W. Trosset, *Direct search methods: then and now*. Journal of Computational and Applied Mathematics, vol.124, no.1-2, pp.191-207, 2000.
- /13/ Gräb, H., Zizala, S., Eckmüller, J., Antreich, K., *The Sizing Rules Method for Analog Integrated Circuit Design*. IEEE/ACM International Conference on Computer-Aided Design, pp. 343-349, 2001.
- /14/ T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, *SPICE3 Version 3f4 User's Manual*, Berkeley, University of California, 1989.
- /15/ J. Puhan, T. Tuma, *Optimization of analog circuits with SPICE 3f4*. Proceedings of the ECCTD'97, vol. 1, pp. 177 - 180, 1997.
- /16/ J. Puhan, T. Tuma, I. Fajfar, *Optimisation Methods in SPICE, a Comparison*. Proceedings of the ECCTD'99, vol. 1, pp. 1279-1282, 1999.
- /17/ M. J. Box, *A New Method of Constrained Optimization and a Comparison with Other Methods*. Computer Journal, vol. 7, pp. 42-52, 1965.
- /18/ Gray, P. R., Hurst, P. J., Lewis, S. H., Meyer, R. G., *Analysis and Design of Analog Integrated Circuits*, Chapter 12. John Wiley & Sons, New York, 2001.
- /19/ Shah, A. H., *Technology Migration of a High Performance CMOS Amplifier Using an Automated Front-to-Back Analog Design Flow*. Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings of Designers' Forum, pp. 224-229, 2002.
- /20/ J. E. Dennis, Jr., V. Torczon, *Parallel Implementations of the Nelder-Mead Simplex Algorithm for Unconstrained Optimization*. Proceedings of the SPIE, Vol. 880, pp. 187-191, 1988.
- /21/ L. Coetzee, E. C. Botha, *The Parallel Downhill Simplex Algorithm for Unconstrained Optimisation*. Concurrency: Practice and Experience, vol. 10, no. 2, pp. 121-137, 1998.

Arpad Bürmen, Drago Strle, Franc Bratkovič,  
Janez Puhan, Iztok Fajfar, Tadej Tuma  
University of Ljubljana  
Faculty of Electrical Engineering  
Tržaška 25, SI-1000 Ljubljana  
E-mail: arpadb@fides.fe.uni-lj.si



# MOŽNOSTI IZVEDBE ADAPTIVNEGA FIR SITA S PROGRAMIRNIMI (FPGA) VEZJI

Davorin Osebik, Rudolf Babič

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru

**Ključne besede:** digitalna obdelava signalov, adaptivno izločanje motilnega signala, adaptivna digitalna sita, nerekurzivna digitalna sita, porazdeljena aritmetika, zaporedna aritmetična struktura

**Izveček:** V članku je prikazan matematični opis adaptivnega FIR sita s 16 koeficienti, prikaz izvedbe s programirnimi logičnimi vezji firme Xilinx in simulacija izvedbene strukture. Sito je izvedeno v porazdeljeni aritmetiki s sprotim izračunom vektorja delnih vsot koeficientov  $v(k)$ . Načrtano je v dveh delih, iz običajne enote nerekurzivnega digitalnega sita, ki mu lahko spreminjamo koeficiente, in enote za adaptivno izračunavanje koeficientov. Aritmetična enota za izračun koeficientov FIR sita izračunava koeficiente po LMS algoritmu. Izvedena je z zaporedno logiko za izvajanje aritmetično logičnih operacij. S takšnim pristopom smo dosegli linearno naraščanje aparturne kompleksnosti enote za izračun koeficientov v odvisnosti od stopnje sita, kar omogoča pomembni prihranek logičnih elementov. Za izvedbo smo uporabili dve FPGA vezji XC4013E in XC4020E. Vezje XC4013E za adaptivno enoto FIR digitalnega sita in vezje XC4020E za enoto izračuna koeficientov FIR sita. V prispevku so podani podrobni opisi notranjih spremenljivk enote za izračun koeficientov in enote FIR sita s spremenljivimi koeficienti na bitnem nivoju.

Vhodne spremenljivke adaptivnega digitalnega FIR sita smo predstavili z dolžino 16-bitov, medtem ko je dolžina registrov notranje aritmetične strukture med 16 in 24-biti. Rezultate smo zaenkrat dobili s simulacijo izvedbene strukture adaptivnega FIR sita s programskim paketom OrCAD 9.0 ob podpori razvojnega orodja firme Xilinx XACT 5.2. Rezultati so podani za primer izločanja motilnih signalov z lastnostjo hrupa prometne ulice. Pri tem je bilo doseženo izboljšanje razmerja S/N med 20 in 22 dB pri vhodnem razmerju S/N je -20. Izboljšanje razmerja S/N je odvisno od vhodnega razmerja S/N.

## The Practicability of Adaptive FIR Digital Filter Implementation with FPGA Circuits

**Key words:** digital signal processing, adaptive digital filters, FIR filters, adaptive noise cancelling, distributed arithmetic technique, serial arithmetic structure

**Abstract:** The FPGA circuits have become a good alternative for digital signal processing applications. In this article the mathematical description and computer simulation of the hardware implementation of the adaptive FIR digital filter structure in FPGA circuits is presented. The hardware implementation of digital filter structure is based on the use of distributed arithmetic filter architecture which uses no multipliers in its realisation of the filtering functions.

Adaptive digital filters are successfully used in different fields, such as communication, radar, sonar, seismic, and biomedical engineering. Almost all of these adaptive filters have one common characteristic that a reference input signal vector  $u(k)$  and primary signal  $d(k)$  are applied to compute an estimation error signal  $e(k)$ , which is used to control the values of adjustable digital filter coefficients  $h(k)$ . The presented adaptive digital FIR filter which is carried out with programmable gate array could be used for noise cancelling from the corrupted input signal.

The basic application of adaptive structure is shown in block diagram in figure 2. The first unit is the FIR filter structure, which determines the output values  $y(k)$  with distributed arithmetic principle by equations 6. The partial sums of filter coefficients signed as  $v_i$  and defined with equations 6, are calculated from vector of filter coefficients  $h(k)$  and the vector of inputs  $u(k)$ , by using of equation 12. Because of current calculations of their adapted values, they cannot be stored in the ROM memory as in the ROM-accumulator structure of classical distributed arithmetic realisation [5]. The 16 taps FIR digital filter structure, made within FPGA circuit XC4013 is shown in figure 3. The hardware complexity is accomplished with 16 bits input and output word length and with 16 to 24 bits word length of arithmetic-logic unit. The second unit in the figure 2 is the structure for adaptive filter coefficients calculation, where the least-mean-square (LMS) adaptation algorithm is used. The coefficients of nonrecursive filter vector  $h(k)$  was initially obtained from equations (2) and (3) and from (16), (17) and (19) respectively. In these equations  $\mu$  is the step of adaptation,  $e(k)$  is the estimation error signal, and  $u(k)$  is the reference signal as the input signal in the FIR filter. The product of the  $e(k)u(k)$  from (15) is also implemented with distributed arithmetic technique, where the serial logic of arithmetic operation is used. The hardware complexity of the structure for adaptive filter coefficients calculation rises linear with number of taps. In figure 4 the block diagram of unit for adaptive filter coefficients calculation is shown. For complexity of 16 taps and 19 bits of arithmetic and logic unit the structure is made within one FPGA circuit XC4020E and a adapted set of the taps is obtained every 10  $\mu$ s.

The whole hardware structure was simulated with OrCAD Express and Xilinx XACT 5.2. With 20MHz clock frequency the input signals  $u(k)$  and  $d(k)$  sampling frequency of 100 kHz was obtained. The results of presented adaptive digital filter application are shown as noise cancelling from corrupted input signal. In this application the street noise signal is taken into account. The results depends on the signal to noise (S/N) ratio of the input signal. At S/N ratio of the input signal of about -20dB the noise component is successfully eliminated, and improvement of S/N ratio of 22dB is obtained. The results are shown as comparison of input reference, primary signal and output signal in the time and frequency domain in the figures 6, 7 and 8 and as power S/N ratios of the input signal ( $P_{in}$ ), output signal ( $P_{out}$ ) and improved output signal ( $P_{res}$ ) in the figure 11 respectively.

### 1. Uvod

Adaptivna digitalna sita se uspešno uporabljajo na raznolikih področjih: v komunikacijski tehniki, radarski in sonar ni tehniki, seizmologiji in biomedicinski tehniki. Čeprav gre za različna področja, imajo vsa eno skupno lastnost: žele-

ni odziv se izračunava na osnovi ocenitvenega pogreška. Adaptivno sito je sestavljeno iz digitalnega sita in adaptivnega algoritma za izračun koeficientov. Izmed številnih aplikacij uporabe adaptivnih sit bomo predstavili aplikacijo digitalnega sita namenjeno nevtralizaciji interferenčnih signalov iz koristnega signala [1].

Pri izvedbi sistemov za digitalno procesiranje signalov (DPS) obstajata dve možnosti: izvedba aplikacije s signalnim procesorjem ali izvedba s programirnimi vezji /2/. Pri izvedbi sistemov za DSP s signalnim procesorjem proces procesiranja temelji na ustrezno zapisanem algoritmu /3/, medtem ko pri izvedbi sistemov za DPS s programirnimi vezji procesiranje temelji na aparaturi strukturi, ki omogoča implementacijo v programirna vezja. Pri tem se uporabljajo takšne strukture, ki dopuščajo enostavno povečanje aplikacije z dodajanjem paralelnih struktur. Zmanjšanje aparature kompleksnosti je možno doseči z različnimi strukturami FIR sit. Pri FIR sitih s konstantnimi koeficienti zelo poenostavi aparaturno kompleksnost uporaba pomnilnika za zapis delnih vsot koeficientov /4, 5/. Zmanjšanje pomnilnika je možno doseči tudi z izvedbo FIR sit v kaskadni obliki /6/, kjer digitalna sita z manj koeficienti povežemo v kaskadno obliko. Vse našteje strukture za zmanjšanje pomnilnika pri FIR sitih temeljijo na predhodnih izračunih delnih vsot koeficientov, ki jih vpišemo v ROM pomnilnik.

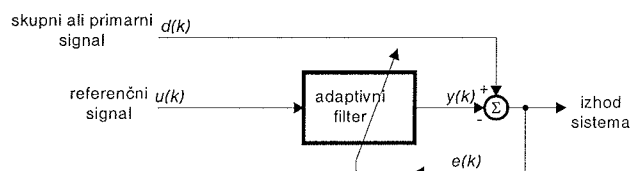
Pri uporabi FIR sit v adaptivnih sistemih se koeficienti spreminjajo za vsak otipek posebej. Uporaba struktur FIR sit s predhodno izračunanimi delnimi vsotami koeficientov v adaptivnih aplikacijah ni možna, saj operacija izračuna in vpisa delnih vsot v pomnilnik vzameta preveč časa. Za FIR sita s spremenljivimi koeficienti so zanimive strukture, ki temeljijo na množenju in akumuliranju izračunane vrednosti (Multiply and Accumulate, MAC algoritmi), in strukture, ki temeljijo na vzporedni porazdeljeni aritmetiki /7/.

Pri izvedbi algoritmov za DPS se je pokazala primerna uporaba zaporedne logike za izvajanje aritmetično logičnih operacij /8/. Z uporabo zaporedne logike izvajanja aritmetično logičnih operacij je zmanjšanje aparature kompleksnosti približno za faktor širine aritmetike, pri tem se potreben čas izračuna bistveno ne spremeni. Izračun koeficientov po LMS algoritmu pri FIR situ poteka v celoti z uporabo zaporedne logike.

V članku bomo prikazali možnost izvedbe adaptivnega nerekurzivnega digitalnega sita s programirnim poljem logičnih vezij. Posebej bomo podali matematični opis enote FIR sita v porazdeljeni aritmetiki razdeljene na  $N$  podstruktur s sprotnim izračunom delnih vsot koeficientov. Predstavili bomo izvedbo enote FIR sita s programirnim vezjem XC4013E. Pri enoti za izračun koeficientov bomo podali matematični opis koeficientov po LMS algoritmu in izvedbo v programirnem vezju XC4025E. Opisali bomo načina povezave obeh enot, ki skupaj tvorita adaptivno sito. Predstavili bomo dobljene rezultate, ki smo jih zaenkrat dobili še s simulacijo s programskim paketom Orcad 9.0 ob podpori razvojnega orodja XCAT 5.2 za načrtovanje programirnih vezij. V rezultatih podajamo uspešnost izločitve motilnega signala iz koristnega signala, podana pa je tudi primerjava med matematičnim modelom in dejansko izvedbo adaptivnega sita.

## 2. Uporaba adaptivnih sit za nevtralizacijo interferenčnih signalov

S programirnimi vezji smo realizirali adaptivno FIR sito, katerega smo uporabili za odstranjevanje prisotnih interferenc iz koristnega signala. Postopek odstranjevanja interferenc temelji na uporabi dveh senzorjev: senzorja primarnega signala  $d(k)$  in senzorja referenčnega signala  $u(k)$ . Blokovno shemo prikazuje slika 1.



Slika 1: Adaptivni sistem v aplikaciji odstranjevanja interferenčnih signalov

V splošnem je lahko uporabljen poljubni adaptivni algoritem in poljubna struktura digitalnega sita. Za aparaturno izvedbo je potrebno izbrati algoritem, ki bo računsko dovolj preprost, saj je potrebno izračun opraviti v času enega otipka vhodnih signalov  $u(k)$  in  $d(k)$ . Preprost matematični algoritem je tudi lažje izvesti v aparaturi opremi. Kriterij, ki uporablja najmanjše srednje kvadratično odstopanje - LMS kriterij, da dovolj dobre rezultate pri odstranjevanju interferenc motilnega signala iz koristnega signala. Kriterij najmanjšega srednjega kvadratičnega odstopanja temelji na minimizaciji izhodne moči prisotnega šuma v koristnem signalu. Koeficiente FIR sita za LMS kriterij je možno izračunati na optimalnem linearnem diskretnem situ, ki je poznano kot Wienerjevo sito /1/. Za aparaturno izvedbo je primernejša metoda algoritma strmega spusta, pri kateri so optimalnim koeficientom Wienerjevega sita približamo v nekaj korakih /9, 10/. S tem algoritmom dobimo dovolj natančen in robusten izračun koeficientov adaptivnega FIR sita.

Za digitalno sito smo izbrali FIR sito, pri katerem ni težav s stabilnostjo. Pri tem smo za FIR sito uporabili strukturo porazdeljene aritmetike. Koeficienti FIR sita se izračunavajo na osnovi LMS kriterija.

Izračun veličin pri adaptivnem FIR situ po LMS kriteriju z uporabo metode strmega spusta poteka po treh temeljnih enačbah:

1. izračun izhodne vrednosti FIR digitalnega sita

$$y(k) = \mathbf{h}^T(k) \mathbf{u}(k), \quad (1)$$

2. izračun ocene odstopanja

$$e(k) = d(k) - y(k), \quad (2)$$

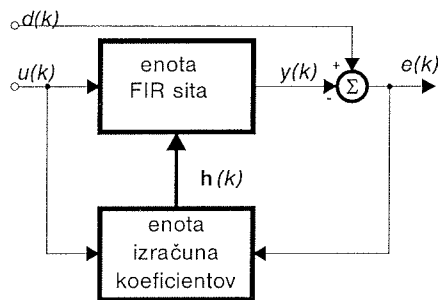
3. izračun koeficientov sita

$$\mathbf{h}(k) = \mathbf{h}(k-1) + \mu \mathbf{u}(k) e(k). \quad (3)$$

V enačbah (1), (2) in (3) imata vektor koeficientov FIR sita  $\mathbf{h}(k)$  in vektor koeficientov vhodnega signala  $\mathbf{u}(k)$  naslednje komponente:

$$\mathbf{h}(k) = \begin{bmatrix} h_0(k) \\ h_1(k) \\ \vdots \\ h_{N-1}(k) \end{bmatrix} \quad \text{in} \quad \mathbf{u}(k) = \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-N) \end{bmatrix} \quad (4)$$

Na osnovi enačb adaptivnega FIR sita smo za izračun veličin adaptivnega digitalnega sita izdelali blokovno shemo, razdeljeno na enoto FIR sita in enoto za izračun koeficientov. Blokovno shemo obeh struktur prikazuje slika 2.



Slika 2: Blokovna shema adaptivnega digitalnega sita

Adaptivno digitalno sito sestavlja enota FIR sita, ki opravlja izračun izhodne vrednosti  $y(k)$  po enačbi (1) na osnovi  $k$ -tega otipka referenčnega signala  $u(k)$  in dobljenega  $k$ -tega vektorja koeficientov  $\mathbf{h}(k)$  FIR sita. Izračun  $k$ -tega vektorja koeficientov FIR sita poteka v dveh korakih. V prvem koraku je potrebno določiti oceno odstopanja  $e(k)$ . Ocena odstopanja je določena kot razlika med  $k$ -tim otipkom primarnega signala  $d(k)$  in  $k$ -tim otipkom izhodnega signala  $y(k)$ . Izračun opisuje enačba (2). Na osnovi določene ocene odstopanja  $e(k)$  po LMS kriteriju z metodo strmega spusta je vektor koeficientov določen z enačbo (3).

Predstavili smo izračun veličin adaptivnega digitalnega sita s poljem programirnih logičnih vezij. Na osnovi podrobnega matematičnega zapisa enačb notranjih spremenljivk adaptivnega FIR sita bomo podali njihovo aparaturno kompleksnost in rešitve, ki smo jih uporabili pri izvedbi adaptivnega sita s programirnimi vezji.

## 2.1 Enota FIR sita

Pri FIR sitih izračun izhodne besede  $y(k)$  opisuje enačba (1). V enačbi (1) sta matriki koeficientov  $\mathbf{h}(k)$  in  $\mathbf{u}(k)$  za FIR sito z  $N$  koeficienti enodimenzionalni matriki dimenzije  $N$ . Elemente obeh matrik podaja zapis (4). Namesto opisa izračuna izhodne besede  $y(k)$  po enačbi (1) lahko izračun opišemo s konvolucijsko enačbo

$$y(k) = \sum_{i=0}^{N-1} h_i(k) u(k - (i+1)) \quad (5)$$

V enačbi (5) so  $h_i(k)$  komponenta vektorja koeficientov  $\mathbf{h}(k)$  digitalnega FIR sita in  $u(k-i+1)$  so komponenta vektorja koeficientov vhoda  $u(k)$ . Pri izbiri izvedbe FIR sita s spremenljivim vektorjem koeficientov  $\mathbf{h}(k)$  imamo možnost med izbiro izvedbe FIR sita v koncentrirani aritmetiki, kjer izračun izhodne besede  $y(k)$  poteka neposredno po konvolucijski enačbi (5). Pri tej metodi, ki je poznana kot izvedba FIR sita v koncentrirani aritmetiki, potrebujemo za izračun izhodne besede pri situ z  $N$  koeficienti  $N$  množilnikov. Druga možnost je izvedba FIR sita v porazdeljeni aritmetiki /5/, kjer izračun izhodne besede  $y(k)$  poteka brez uporabe množilnikov. Izvedba FIR sita v strukturi porazdeljene aritmetike temelji na uporabi pomnilnika za zapis delnih vsot koeficientov z algoritmom množenja in pomnjenja vrednosti (MAC algoritem) /7/. Izračun izhodne vrednosti izhodne besede  $y(k)$  poteka po enačbi

$$y(k) = \sum_{i=1}^{Bu-1} v_i 2^{-i} - v_{Bu} \quad (6)$$

V enačbi (6) so  $v_i$  delne vsote koeficientov, ki so odvisni od vhodnega signala  $u(k)$  in vektorja koeficientov FIR sita  $\mathbf{h}(k)$ ,  $v_i = f(u(k), \mathbf{h}(k))$ .

Dosedanji razvoj digitalnih FIR sit s konstantnim vektorjem koeficientov  $\mathbf{h}(k)$  je temeljil na izvedbi sit v različnih strukturah porazdeljene aritmetike, kjer smo z določenimi postopki zmanjševali potrebno velikost pomnilnika za zapis delnih vsot koeficientov. V primeru, da je vektor koeficientov FIR sita konstanten, je možno vse kombinacije delnih vsot izračunati vnaprej in jih vpisati v pomnilnik. Za sito z  $N$  koeficienti je vseh kombinacij delnih vsot koeficientov  $2^N$ . Pri spremenljivem vektorju koeficientov  $\mathbf{h}(k)$  je potrebno poiskati strukture FIR sit v porazdeljeni aritmetiki, kjer bo potrebno izračunati le tiste delne vsote koeficientov, ki jih potrebujemo pri danem otipku vhodnega signala  $u(k)$ . Teh delnih vsot je  $N$ . S tem se zmanjša potrebna velikost pomnilnika za zapis delnih vsot koeficientov z  $2^N$  na  $N$  pomnilniških lokacij /7/. Struktura, ki ustreza navedenim pogojem, je struktura s sprotnim izračunom delnih vsot koeficientov /7, 5/. Aparaturna izvedba FIR sita v porazdeljeni aritmetiki s sprotnim izračunom vektorja delnih vsot koeficientov je kompleksnejša od klasične ROM strukture sita FIR sita v porazdeljeni aritmetiki, je taka edina primerna za izvedbo sprotnega izračunavanja in delnih vsot koeficientov. Aparaturna kompleksnost strukture FIR sita s sprotnim izračunom delnih vsot koeficientov je ocenjena z  $N^2$ .

### 2.1.1 Opis izračuna vrednosti izhodnega signala $y(k)$ pri FIR situ s sprotnim izračunom delnih vsot koeficientov

Pri izračunu izhodne vrednosti  $y(k)$  digitalnega sita v porazdeljeni aritmetiki s sprotnim izračunom delnih vsot koeficientov izhajamo iz enačbe (1), kjer poteka množenje vektorja koeficientov  $\mathbf{h}(k)$  in vektorja koeficientov vhoda  $\mathbf{u}(k)$  po postopku porazdeljene aritmetike. Pri tem postopku vrednost vhodnega otipka  $u(k)$  zapišemo v bitni obliki z dvojiškim komplimentom. Zapis opisuje enačba



$$u(k) = -b_{u,0}(k) + \sum_{i=1}^{B_u-1} b_{u,i}(k) 2^{-i}. \quad (7)$$

V enačbi (7) najbolj utežni bit  $b_{u,0}(k)$  predstavlja predznak  $k$ -tega otipka vhodne besede  $u(k)$ . Vektor koeficientov vhoda  $u(k)$  na bitnem nivoju zapišemo z upoštevanjem dvojiškega komplimenta z

$$\mathbf{u}(k) = \begin{bmatrix} b_{u,0}(k-1) + \sum_{i=1}^{B_u-1} b_{u,i}(k-1) 2^i \\ b_{u,0}(k-2) + \sum_{i=1}^{B_u-1} b_{u,i}(k-2) 2^i \\ \vdots \\ b_{u,0}(k-N) + \sum_{i=1}^{B_u-1} b_{u,i}(k-N) 2^i \end{bmatrix}. \quad (8)$$

Pri adaptivnih sitih se vektor koeficientov  $\mathbf{h}(k)$  spremeni pri vsakem otipku vhodnega signala  $u(k)$ , zato ni potrebno opraviti izračuna celotnega nabora delnih vsot koeficientov  $v_i$ . Z uporabo sprotnega izračuna delnih vsot koeficientov, neposredno izračunavamo vektor delnih vsot koeficientov  $\mathbf{v}(k)$  iz otipka vhodne besede  $u(k)$  in vektorja koeficientov  $\mathbf{h}(k)$  po enačbi

$$\mathbf{v}(k) = \mathbf{b}_u^T(k) \mathbf{h}(k). \quad (9)$$

V enačbi (9) je vektor  $\mathbf{b}_u(k)$  bitni zapis trenutnega vhodnega otipka signala  $u(k)$  in ostalih petnajstih predhodnih vrednosti signala  $u(k)$ . Pri tem  $k$ -ti otipek vrednosti vektorja  $\mathbf{b}_u(k)$  zapišemo z

$$\mathbf{b}_u(k) = \begin{bmatrix} b_{u,0}(k-1) & b_{u,1}(k-1) & \cdots & b_{u,B-1}(k-1) \\ b_{u,0}(k-2) & b_{u,1}(k-2) & \cdots & b_{u,B-1}(k-2) \\ \vdots & \vdots & \vdots & \vdots \\ b_{u,0}(k-N) & b_{u,1}(k-N) & \cdots & b_{u,B-1}(k-N) \end{bmatrix}. \quad (10)$$

V enačbi (9) je vektor delnih vsot koeficientov  $\mathbf{v}(k)$ , funkcija vektorja  $\mathbf{b}_u(k)$  in vektorja koeficientov sita  $\mathbf{h}(k)$ . Z množenjem obeh matrik  $\mathbf{h}(k)$  in  $\mathbf{u}(k)$  dobimo matriko delnih vsot koeficientov  $\mathbf{v}(k)$ , ki se neposredno izračunava v FIR situ ob vsakem novem otipku vhodne besede  $u(k)$  po enačbi (9). Dobljeni vektor delnih vsot koeficientov  $\mathbf{v}(k)$  za  $k$ -ti otipek vhodne besede  $u(k)$  zapišemo z

$$\mathbf{v}(k) = \begin{bmatrix} h_0(k) b_{u,0}(k-1) + \cdots + h_{N-1}(k) b_{u,0}(k-N) \\ h_0(k) b_{u,1}(k-1) + \cdots + h_{N-1}(k) b_{u,1}(k-N) \\ \vdots \\ h_0(k) b_{u,B-1}(k-1) + \cdots + h_{N-1}(k) b_{u,B-1}(k-N) \end{bmatrix}. \quad (11)$$

Matrika vektorja delnih vsot koeficientov  $\mathbf{v}(k)$  ima dimenzijo  $B_u \times N$ .  $B_u$  je število bitov za zapis vhodne besede  $u(k)$ . Za izračun matrike vektorja je potrebno  $B_u$  iteracij. Aparaturna kompleksnost enote FIR sita se zaradi povečanja iteracij ne spremeni, poveča se le potreben čas izračuna izhodne besede  $y(k)$ . Enačbo vektorja delnih vsot koeficientov FIR sita  $\mathbf{v}(k)$  zaradi preglednosti zapišemo v krajši obliki

$$\mathbf{v}(k) = \begin{bmatrix} \sum_{i=0}^{B_u-1} h_i(k) b_0(k-(N+1)) \\ \sum_{i=0}^{B_u-1} h_i(k) b_1(k-(N+1)) \\ \vdots \\ \sum_{i=0}^{B_u-1} h_i(k) b_{B_u-1}(k-(N+1)) \end{bmatrix}. \quad (12)$$

V enačbi (12) so  $v_i(k)$  komponente vektorja delnih vsot koeficientov. Za sito z  $N$  koeficienti dobimo za vsak nov otipek vhodnega signala  $u(k)$   $N$  novih komponent delnih vsot koeficientov  $v_i(k)$ . Pri izvedbi adaptivnega FIR sita smo se odločili za strukturo sprotnega izračuna vektorja delnih vsot koeficientov  $\mathbf{v}(k)$ . Tako dobljeni vektor koeficientov  $\mathbf{v}(k)$  ima dimenzijo  $N$  in vsebuje naslednje komponente,

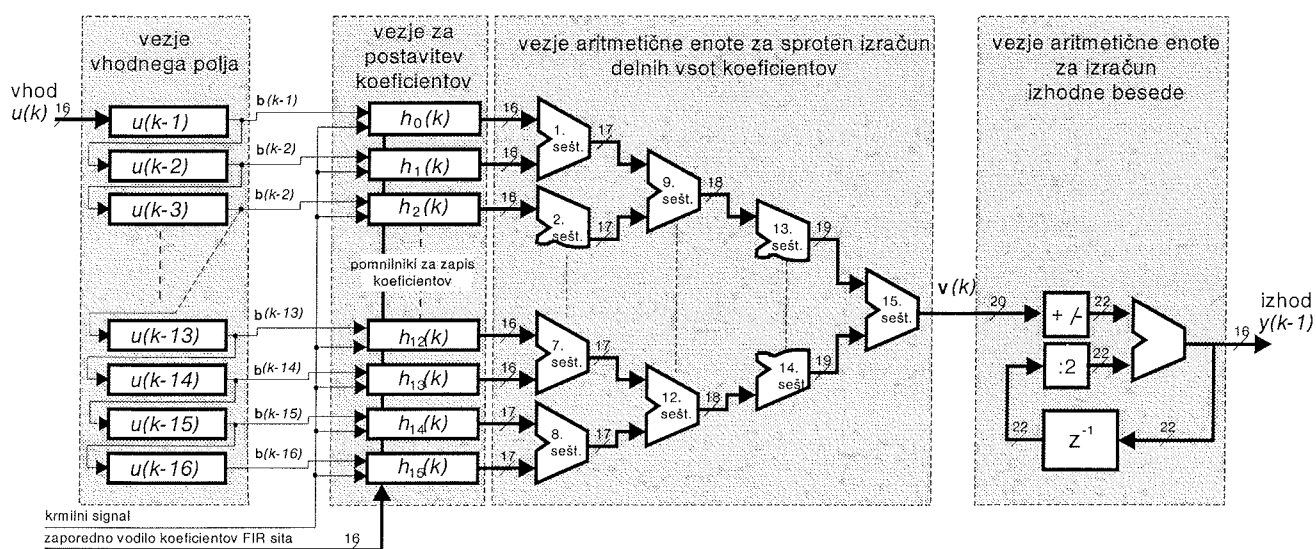
$$\mathbf{v}(k) = \begin{bmatrix} v_1(k) \\ v_2(k) \\ \vdots \\ v_{B_u}(k) \end{bmatrix}. \quad (13)$$

Izračun izhodne besede FIR sita  $y(k)$  poteka neposredno iz komponent vektorja delnih vsot koeficientov po enačbi (6). Za izračun izhodne  $y(k)$  je potrebno izvesti  $B_u$  iteracij.

### 2.1.2 Aparaturna izvedba enote FIR digitalnega sita s spremenljivimi koeficienti

Z FPGA vezjem XC4013E firme Xilinx smo realizirali enoto FIR sita z  $N=16$  koeficienti, z  $B_u=16$ -bitno dolžino vhodne besede in 16-bitno dolžino izhodne besede in dolžino notranje strukture med 16 in 22 biti. Dosegli smo frekvenco vzorčenja vhodne besede  $f_v=100\text{kHz}$  pri frekvenci osnovnega signala ure 20MHz. Izvedbo enote FIR sita v porazdeljeni aritmetiki razdeljeno na  $N=16$  podstruktur prikazuje slika 3.

Realizacijsko enoto FIR digitalnega sita smo razdelili na štiri različna vezja. V enoti FIR digitalnega sita sta vezje vhodnega polja in vezje aritmetične enote enaki kot pri klasični izvedbi adaptivnega FIR sita v porazdeljeni aritmetiki /5/. Vezje za postavitve koeficientov in vezje aritmetične enote za sprotni izračun delnih vsot koeficientov smo načrtali posebej za aplikacijo FIR sita s spremenljivimi koeficienti.



Slika 3: Blokovna shema enote digitalnega FIR sita v porazdeljeni aritmetiki razdeljenega na  $N=16$  podstruktur

**Vezje vhodnega polja** opravlja izračun vektorja  $\mathbf{b}_u(k)$  po enačbi (10), pri tem ima vektor dimenzijo  $B_u=16$  stolpcev in  $N=16$  vrstic.

**Vezje za postavitev koeficientov** skrbi samo za prenos vektorja koeficientov  $\mathbf{h}(k)$  iz vezja za izračun vektorja koeficientov  $\mathbf{h}(k)$  v enoto FIR sita in njegovo pretvorbo iz zaporedne oblike zapisa v vzporedno obliko zapisa. Vezje je sestavljeno iz šestnajstih 16-bitnih zaporedno vzporednih pretvornikov.

**Vezje aritmetične enote za sproten izračun delnih vsot koeficientov** izračunava vektor delnih vsot koeficientov  $\mathbf{v}(k)$  po enačbi (9), pri tem ima vektor  $\mathbf{v}(k)$   $N=16$  vrstic. Izračun vektorja  $\mathbf{v}(k)$  se opravi po  $B_u=16$  iteracijah. Vezje sestavlja 15 seštevalnikov, ki so dolžine od 16. do 19. bitov.

Vektor delnih vsot koeficientov se izračuna po  $B_u=16$  iteracijah. Vezje aritmetične logične enote izračunava vrednost izhodne besede  $y(k)$  po enačbi (11) oz (12). Za izračun je potrebno  $B_u=16$  iteracij.

**Vezje aritmetične enote** za izračun izhodne besede  $y(k)$  po enačbi (6) potrebuje  $B_u=16$  iteracij.

## 2.2 Enota za izračun koeficientov

Izračun vektorja koeficientov  $\mathbf{h}(k)$  adaptivnega FIR digitalnega sita poteka po algoritmu, ki skrbi za optimalno nastavljanje parametrov sita. Izračun temelji na kriteriju najmanjšega srednjega kvadratičnega odstopanja, kjer se optimalni koeficienti izračunajo z metodo strmega spusta /1/. Ta način je izbran zaradi matematične enostavnosti in robustnosti algoritme izračuna. Izbrani LMS kriterij zadostuje pogoju glede aparturne kompleksnosti enote za izračun koeficientov, časa adaptacije in izračuna koeficientov sita v realnem času. Glede na izračun koeficientov FIR sita po LMS kriteriju z algoritmom strmega spusta ločimo: izračun koeficientov s predznačeno funkcije, izračun koeficientov

z nespremenljivo adaptivno konstanto in izračun koeficientov s spremenljivo adaptivno konstanto.

### 2.2.1 Opis izračuna koeficientov FIR sita

Izračun novih koeficientov sita poteka v dveh korakih. V prvem koraku je potrebno izvesti izračun ocenitvenega odstopanja med primarnim signalom  $d(k)$  in izhodnim signalom  $y(k)$  po enačbi (2). V drugem koraku je potrebno izvesti izračun novega vektorja koeficientov FIR sita  $\mathbf{h}(k)$  po enačbi (3). Enačbo (3) razdelimo na dva dela. Del s predhodnimi vrednostmi vektorja koeficientov FIR sita  $\mathbf{h}(k-1)$  in del z vektorjem trenutnega odstopanja koeficientov FIR sita  $\mathbf{h}'(k)$ ,

$$\mathbf{h}(k) = \mathbf{h}(k-1) + \mathbf{h}'(k). \quad (14)$$

V enačbi (14) je izračun vektorja trenutnega odstopanja koeficientov  $\mathbf{h}'(k)$  za aparturno izvedbo najkompleksnejši del. Izračun tega vektorja poteka po enačbi

$$\mathbf{h}'(k) = \mu e(k) \mathbf{u}(k). \quad (15)$$

V enačbi (15) je najprej izvedeno množenje ocene odstopanja  $e(k)$  z adaptivno konstanto  $\mu$ . Pri tem je dobljen zmnožek med  $e(k)$  in  $\mu$  skalar, s katerim v naslednjem koraku izračuna pomnožimo vektor koeficientov vhoda  $\mathbf{u}(k)$  z enačbo

$$\mathbf{h}'(k) = \mu e(k) \mathbf{u}(k) = \begin{bmatrix} \mu e(k) u(k-1) \\ \mu e(k) u(k-2) \\ \vdots \\ \mu e(k) u(k-N) \end{bmatrix} \quad (16)$$

Ta izračun smo realizirali s postopkom porazdeljene aritmetike. Vektor vhodnih koeficientov  $\mathbf{u}(k)$  vhodnega refe-

renčnega signala  $u(k)$  zapišemo na bitnem nivoju z uporabo dvojiškega komplimenta. Z upoštevanjem enačbe (7) in zapisa vektorja  $\mathbf{b}_u(k)$  z enačbo (10) dobimo zapis vektorja koeficientov vhoda  $\mathbf{u}(k)$ , ki je opisan z (8). Vektor  $\mathbf{u}(k)$  je zapisan na bitnem nivoju z  $\mathbf{b}_u(k)$ , ki je bitni zapis trenutne vrednosti vhodnega referenčnega signala  $u(k)$  in preostalih  $N$ -tih predhodnih otipkov. Po produktu vektorja  $\mathbf{u}(k)$  s skalarjem  $\mu e(k)$  dobimo vektor trenutnega odstopanja  $\mathbf{h}'(k)$ , ki ga zapišemo s.

$$\mathbf{h}'(k) = \mu \begin{bmatrix} -e(k-1)b_{u,0}(k) + \sum_{i=1}^{B-1} e(k)b_{u,i}(k-1)2^{i-1} \\ -e(k)b_{u,0}(k-2) + \sum_{i=1}^{B-1} e(k)b_{u,i}(k-2)2^{i-1} \\ \vdots \\ -e(k)b_{u,0}(k-N) + \sum_{i=1}^{B-1} e(k)b_{u,i}(k-N)2^{i-1} \end{bmatrix} \quad (17)$$

V zadnjem koraku je potrebno le še izvesti seštevanje vektorja predhodnih vrednosti koeficientov  $\mathbf{h}(k-1)$  z vektorjem trenutnega odstopanja  $\mathbf{h}'(k)$  po enačbi (14). (14) Za aparturno izvedbo se izkaže, da je tudi seštevanje ugodno izvesti v iteraciji po času. Za seštevanje vektorja koeficientov FIR sita zapišemo vektor  $\mathbf{h}(k)$  na bitnem nivoju z enačbo

$$\mathbf{h}(k) = \begin{bmatrix} b_{h,0,0}(k) + \sum_{i=1}^{Bh-1} b_{h,0,i}(k)2^i \\ b_{h,1,0}(k) + \sum_{i=1}^{Bh-1} b_{h,1,i}(k)2^i \\ \vdots \\ b_{h(N-1),0}(k) + \sum_{i=1}^{Bh-1} b_{h(N-1),i}(k)2^i \end{bmatrix} \quad (18)$$

V enačbi (18) predstavlja  $b_{h,0}(k)$  najbolj utežni bit prvega koeficienta  $h_0(k)$  FIR sita. Najbolj utežni bit določa predznak,

ostali biti določajo vrednost koeficienta. Pri tem  $B_h$  predstavlja število bitov za zapis posameznega koeficienta.

Vektor koeficientov FIR sita predhodnega otipka ima po tem vrednosti podane s,

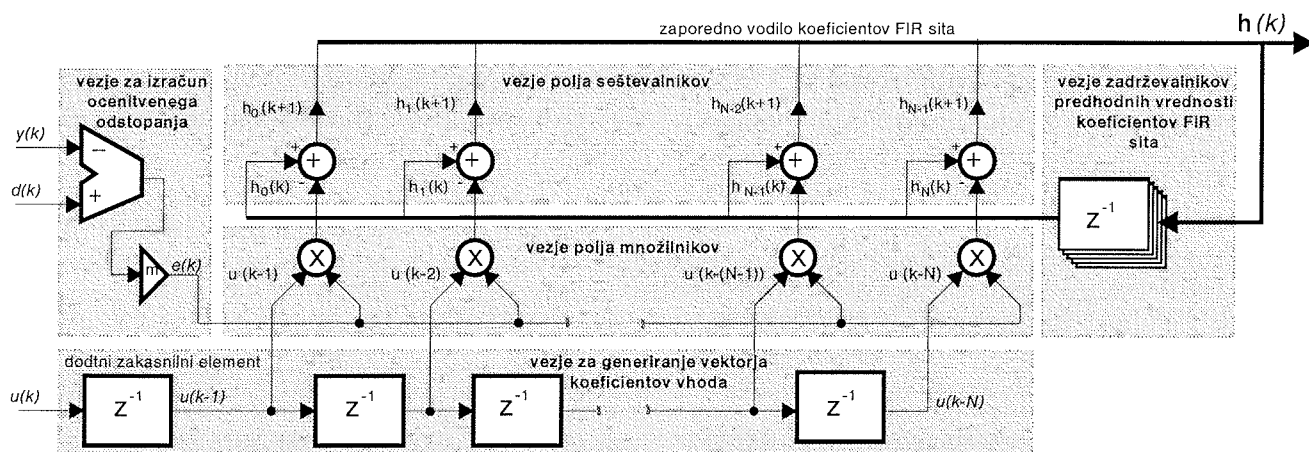
$$\mathbf{h}(k-1) = \begin{bmatrix} b_{h,0,0}(k-1) + \sum_{i=1}^{Bh-1} b_{h,0,i}(k-1)2^i \\ b_{h,1,0}(k-1) + \sum_{i=1}^{Bh-1} b_{h,1,i}(k-1)2^i \\ \vdots \\ b_{h(N-1),0}(k-1) + \sum_{i=1}^{Bh-1} b_{h(N-1),i}(k-1)2^i \end{bmatrix} \quad (19)$$

Oba vektorja koeficientov  $\mathbf{h}(k-1)$  in  $\mathbf{h}'(k)$  sta enodimenzionalna in imata  $N$  vrstic. Za njuno seštevanje potrebuje  $N$  iteracij.

## 2.2.2 Aparturna izvedba enote za izračun koeficientov FIR sita

Z FPGA vezjem smo realizirali enoto za izračun vektorja koeficientov dimenzije  $N=16$  po LMS algoritmu z metodo strmega spusta. Blokovno shemo enote prikazuje slika 4.

Enoto za izračun koeficientov FIR sita, ki izračunava koeficiente po LMS algoritmu s fiksno adaptivno konstanto  $\mu$  smo izvedli v programirnem vezju XC4020E. Enota zmore izračunavati nove koeficiente vsakih 10  $\mu s$  in jih preko vodila posredovati enoti FIR sita. Zaradi omejene možnosti izvajanja računskih operacij s FPGA vezji smo razvili takšne strukture, ki omogočajo izvajanje produkta dveh vektorjev s FPGA vezji ob majhni aparturni kompleksnosti. Posebej bi izpostavili izvedbo polja 16-tih zaporednih množilnikov s poljem zaporednih seštevalnikov in njihovo povezavo z izvedenim zadrževalnikom koeficientov ter nazadnje še izvedbo zaporednega prenosa izračunanih koeficientov v FIR sito. Pri izvedbi enote za izračun koeficientov smo uporabili zaporedno logiko, kar omogoča zmanjšanje aparturne



Slika 4: Blokovna shema enote za izračun koeficientov FIR sita

kompleksnosti. Pri izvedbi enote za izračun koeficientov z FPGA vezji bi posebej izpostavili vezje polja seštevalnikov, vezje polja množilnikov in vezje zadrževalnikov predhodnih vrednosti koeficientov. Kompleksnost enote za izračun koeficientov narašča linearno s številom koeficientov, kar je posebej ugodno za izvedbo adaptivnih sit višjih stopenj. Enoto za izračun koeficientov FIR sita smo razdelili na pet vezij.

**Vezje za izračun ocenitvenega odstopanja  $e(k)$**  opravlja izračun po enačbi (2) in je sestavljeno iz 16-bitnega seštevalnika. V tem vezju se izvede tudi množenje ocenitvenega odstopanja  $e(k)$  z adaptivno konstanto  $\mu$ .

**Vezje za generiranje vektorja koeficientov vhoda  $u(k)$** , opravlja izračun vektorja  $b_u(k)$  po enačbi (10). Vezje je enako zasnovano, kot vezje vhodnega polja v enoti FIR sita. Vektor  $b_u(k)$  nosi bitni zapis vektorja koeficientov vhoda  $u(k)$ .

**Vezja polja seštevalnikov vektorjev** opravlja seštevanje predhodnih vrednosti vektorja koeficientov FIR sita  $h(k-1)$  z vektorjem trenutnega odstopanja  $h'(k)$ . Z vezja množilnikov prihajajo zapovrstjo vrednosti vektorja trenutnega odstopanja koeficientov FIR sita  $h'(k)$  od najmanj do najbolj utežnega bita. Zaradi takšne narave podatkov se je pokazala uporaba zaporednih množilnikov zelo ugodna, vektorja koeficientov FIR sita  $h'(k)$  in  $h(k-1)$  sta že zapisana v zaporedni obliki. Zaradi uporabe zaporedne logike izvajanja aritmetičnih operacij, ki potekajo po postopku cevjenja, nam je uspelo za faktor dolžine bitnega zapisa vektorja koeficientov FIR sita  $h(k)$  zmanjša aparaturno kompleksnost izvedbe polja seštevalnikov. Izračun se izvaja po enačbi (14).

**Vezje polja množilnikov** opravlja produkt skalarja  $\mu$   $e(k)$  z vektorjem koeficientov  $u(k)$  vhodnega signala  $u(k)$  po enačbi (17). Za sito z  $N=16$  koeficienti je potrebno opraviti izračun v šestnajstih korakih. Ker se vedno izvrši produkt skalarja z enim utežnim bitom referenčnega signala  $u(k)$ , nastaja celotna vrednost vektorja trenutnega odstopanja  $h'(k)$  postopno od najmanj utežnega bita do najbolj utežnega bita za vse vrstice v matriki  $u(k)$ .

**Vezje zadrževalnikov predhodnih vrednosti koeficientov** zakasni za en otipek izračunan vektor koeficientov  $h(k)$ . Struktura vezja je podobna vezju za generiranje vektorja koeficientov vhoda.

### 2.3 Povezava enote FIR sita z enoto za izračun koeficientov

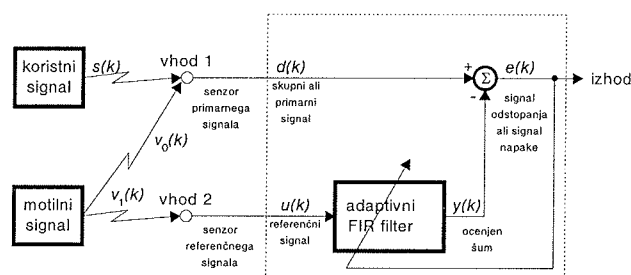
Pri aparaturni izvedbi FIR sita je vedno prisotna neka zakasnitev med izračunano izhodno vrednostjo  $y(k)$  FIR sita in otipkom vhodne besede  $u(k)$ . Zakasnitev je posledica izračuna izhodne vrednosti po enem izmed algoritmov. V našem primeru je to algoritem porazdeljene aritmetike. Zakasnitev v enačbi za zapis delnih vsot koeficientov (11) je prisotna v komponentah vektorja  $b_u(k)$ . Enaka zakasnitev je prisotna tudi v primeru, če poteka algoritem izračuna izhodne besede  $y(k)$  pri FIR situ na klasični način s kon-

volucijsko enačbo (5). V konvolucijski enačbi je zakasnitev opisana z zakasnitvijo vhodnega signala FIR sita  $u(k)$  za en otipek. Zakasnitev izračuna izhodne besede pri FIR situ je potrebno upoštevati tudi pri izračunu vektorja koeficientov  $h(k)$ . Pri izračunu vektorja koeficientov  $h(k)$  je zakasnitev že upoštevana v enačbi (3) z zapisom vektorja koeficientov vhoda  $u(k)$ , enačba (4). V aparaturni izvedbi to zakasnitev predstavlja vgrajen dodatni zakasnilni element, ki ga prikazuje slika 4. S tako dodanim blokom smo tudi pri vezju za izračun koeficientov FIR sita dobili vektor koeficientov  $u(k)$  vhodnega signala  $u(k)$ , ki je opisan s (4).

### 3. Rezultati

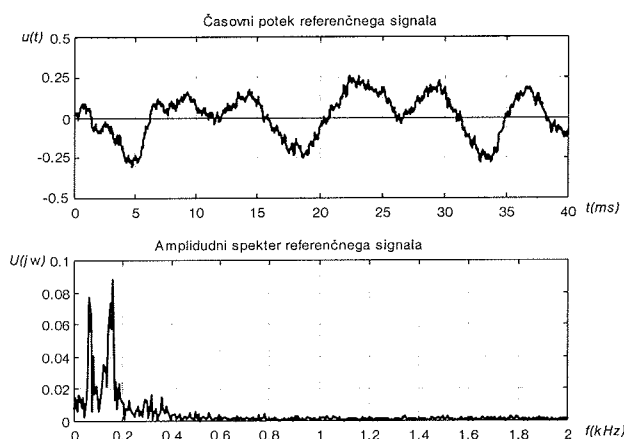
Rezultati so za enkrat še dobljeni s simulacijo enote FIR sita in enote vezja za izračun koeficientov. Pri simulaciji smo uporabili programski paket OrCad 9.0 /13/ ob podpori Xilinxovega razvojnega orodja XACT 5.2 /11/. Delovanje obeh enot smo simulirali za čas med 0 in 200ms. Za ta čas je podana primerjava med rezultati, dobljenimi s simulacijo, in rezultati dobljenimi z matematičnim modelom.

Delovanje adaptivnega sita izvedenega s programirnim vezjem smo preizkusili pri izločanju šuma /1, 12/, ki je prisoten ob prometni ulici. Sistem za izločanje šuma z adaptivnim FIR sitom prikazuje slika 5. Sitem sestavljata dva senzorja: senzor primarnega  $d(k)$  in senzor referenčnega signala  $u(k)$ .



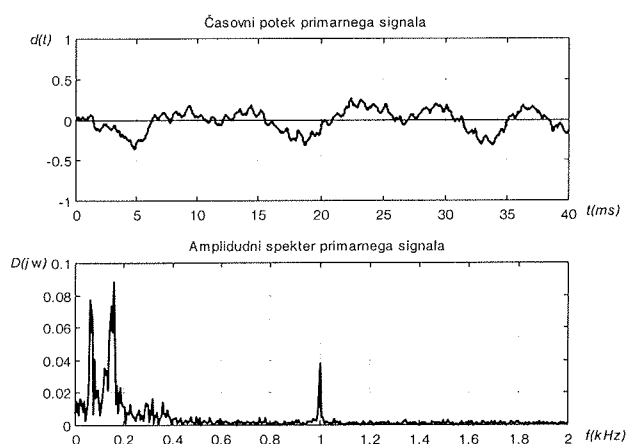
Slika 5: Adaptivno sito pri odpravi šuma superponiranega v koristnem signalu

Senzor primarnega signala zajema koristni signal  $s(k)$  in del motilnega signala  $v_0(k)$ . Senzor referenčnega signala zajema le motilni signal  $v_1(k)$ . Postopek odprave interferenc je uspešen, če sta motilna signala  $v_0(k)$  in  $v_1(k)$ , ki prihajata na senzor senzorja  $d(k)$  in  $u(k)$ , korelirana. Razmere različnega sprejema motilnih signalov smo simulirali z dodanim nizkim sitom. Za uspešno izločitev motilnega signala tudi ne sme biti koristni signal  $s(k)$  koreliran z motilnima signaloma  $v_1(k)$  in  $v_0(k)$ . Časovni potek in amplitudni spekter referenčnega signala prikazuje slika 6. Diskretna Fourierjeva transformacija je na referenčnem signalu opravljena v 1024 točkah. Frekvenca vzorčenja obeh vhodnih signalov je bila  $f_v=100\text{kHz}$ . Slika 6 prikazuje časovni potek referenčnega signala  $u(k)$  in njegov amplitudni spekter  $U(j\omega)$ .



Slika 6: Časovni potek referenčnega signala  $u(k)$  in njegov amplitudni spekter  $U(j\omega)$

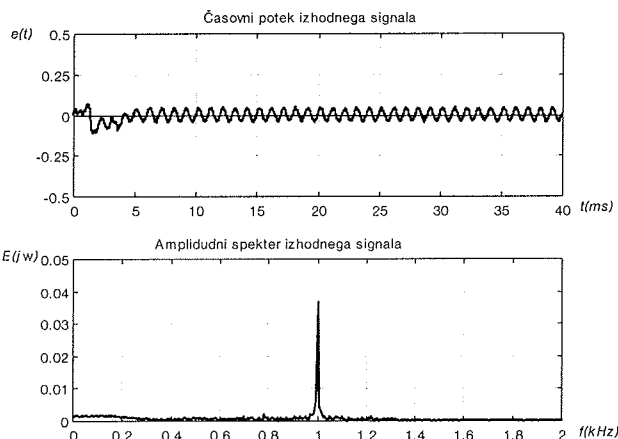
Na senzor primarnega signala  $d(k)$  prihaja vsota koristnega signala  $s(k) = 0.4\sin(2\pi 1000kT_V)$  in motilnega signala  $v_1(k)$ . Pri tem je perioda vzorčenja  $T_V = 10 \mu s$ . Slika 7 prikazuje časovni potek primarnega signala  $d(k)$  in njegov amplitudni spekter  $D(j\omega)$ .



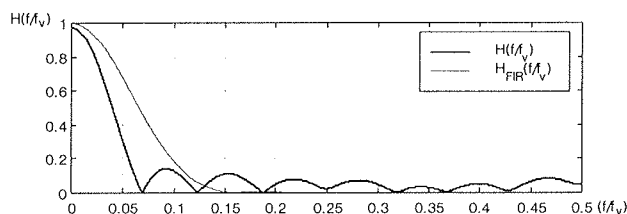
Slika 7: Časovni potek primarnega signala  $d(k)$  in njegov amplitudni spekter  $D(j\omega)$

Adaptivno sito na osnovi referenčnega signala izloči superponirano komponento motilnega signala iz primarnega signala. Časovni potek in amplitudni spekter izhodnega signala kaže slika 8.

V amplitudnem spektru  $E(j\omega)$  izhodnega signala  $e(k)$  ni več prisotnih komponent motilnega signala  $v_1(k)$ . V realnem okolju z dvema senzorjema ni možno zajeti istega motilnega signala. Zato se motilna signala  $v_1(k)$  in  $v_0(k)$ , ki ju sprejemata senzorja  $d(k)$  in  $u(k)$ , med seboj nekoliko razlikujeta. S slike 8 je razviden tudi potreben čas adaptacije adaptivnega sita, ki znaša 5 ms. Razmere različnega sprejema signalov iz dveh senzorjev v realnem okolju smo simulirali z uporabo nizko prepustnega FIR sita stopnje 16, relativne prepustne frekvence  $f_p = 0.05f_v$  in relativne zaporne frekvence  $f_z = 0.4f_v$ . Slika 9 prikazuje frekvenčni odziv FIR sita  $H_{FIR}(f/f_v)$ , s katerim smo simulirali različne pogoje sprejema motilnega signala.



Slika 8: Časovni potek primarnega signala  $e(k)$  in njegov amplitudni spekter  $E(j\omega)$



Slika 9: Frekvenčna karakteristika FIR digitalnega sita  $H_{FIR}(f/f_v)$  s katerim smo simulirali različne pogoje sprejema šuma in frekvenčna karakteristika FIR sita  $H(f/f_v)$  po opravljeni adaptaciji koeficientov

Na sliki 9 je prikazan tudi frekvenčni odziv FIR adaptivnega sita  $H(f/f_v)$  po opravljeni adaptaciji koeficientov. Frekvenčni odziv smo določili iz poprečnih vrednosti 1000 zaporednih vzorcev koeficientov po končanem prehodnem pojavu adaptacije z enačbo

$$\bar{h}(k) = \frac{1}{1000} \sum_{k=10000}^{11000} h(k). \quad (20)$$

Opravili smo tudi primerjavo med rezultati dobljenimi s simulacijo enote FIR sita in enote vezja za izračun koeficientov, z rezultati, dobljenimi z matematičnim modelom. Iz analize odstopanja med vrednostmi signalov  $e(k)$  in  $y(k)$ , dobljenimi s simulacijo obeh enot je možno sklepati o obnašanju izvedbe adaptivnega FIR sita tudi za čase, večje od 200 ms. Razliko med izhodnim signalom  $e(k)$ , dobljenim s simulacijo obeh enot, in vrednostmi signala  $e_{ref}(k)$ , dobljenimi z matematičnim modelom, podaja enačba,

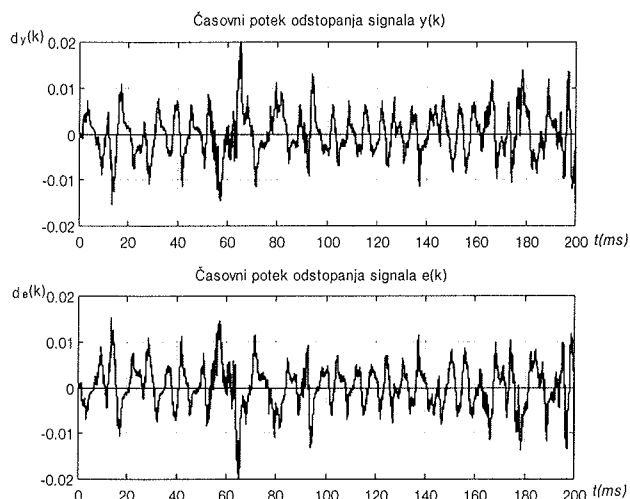
$$\delta_e(k) = e_{ref}(k) - e(k). \quad (21)$$

Razliko med izhodnim signalom enote FIR sita  $y(k)$  in vrednostmi signala  $y_{ref}(k)$ , dobljenimi z matematičnim modelom, podaja enačba

$$\delta_y(k) = y_{ref}(k) - y(k). \quad (22)$$



V enačbah (21) in (22) sta  $e_{\text{ref}}(k)$  in  $y_{\text{ref}}(k)$  izhodni vrednosti, dobljeni z matematičnim modelom FIR sita. Razlike obeh izhodnih signalov prikazuje slika 10.



Slika 10: Odstopanje izhodnega signala FIR sita  $y(k)$  in izhodnega signala aplikacije  $e(k)$  od točnih vrednosti, dobljenih z matematičnim modelom

Uspešnost izločanje motilnega signala iz koristnega signala vidimo tudi s primerjavo časovnih potekov razmerij moči med koristnim signalom in šumom (S/N). Razmerje moči S/N primarnega signala smo izračunali za vsak otipek posebej po enačbi

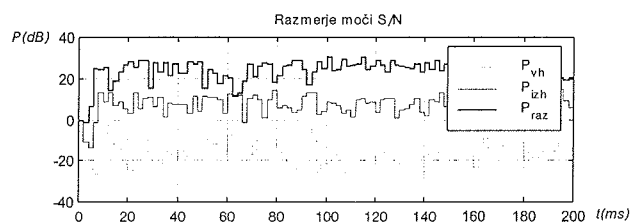
$$P_{vh}(k)[dB] = 10 \log \left( \frac{s(k)^2}{u(k)^2 + \alpha} \right) \quad (23)$$

V enačbi (23) je  $s(k)$  koristni signal,  $u(k)$  je motilni signal  $v_1(k)$ , dodana je še konstanta  $\alpha = 10^{-99}$  za vrednosti otipkov motilnega signala  $u(k)=0$ . Razmerje moči S/N na izhodu smo izračunali po enačbi

$$P_{izh}(k)[dB] = 10 \log \left( \frac{s(k)^2}{(e(k) - s(k))^2 + \alpha} \right) \quad (24)$$

V enačbi (24) je  $e(k)$  signal z izločeno komponento motilnega signala. Pri grafičnem prikazu rezultatov razmerja moči S/N smo se odločili za prikaz poprečne vrednosti razmerja moči 200-tih otipkov. Časovni potek razmerja moči S/N na vходу ( $P_{vh}$ ), na izhodu ( $P_{izh}$ ) in izboljšanje razmerja moči S/N ( $P_{raz}$ ) prikazuje slika 11.

Razmerje moči je dobljeno s simulacijo enote FIR sita in enote izračuna koeficientov s programskim paketom Orcad 9.0. Manjše, kot je razmerje moči S/N na vходу, večje je izboljšanje razmerja. Izboljšanje razmerja moči S/N se giblje med 20dB in 22dB. Čas adaptacije adaptivnega sita znaša okoli 8ms.



Slika 11: Časovni potek razmerja moči S/N primarnega signala  $d(k)$ , izhodnega signala  $e(k)$  in izboljšanje razmerja S/N

## 4. Zaključek

V članku smo prikazali matematični opis notranjih spremenljivk adaptivnega digitalnega FIR sita. Matematični opis podaja vpogled v kompleksnost izvedbe adaptivnega FIR sita s FPGA vezji in morebitne izboljšave glede izvajanja algoritmov aritmetičnih logičnih operacij.

Vezje adaptivnega sita smo razdelili na enoto FIR sita in enoto za izračun koeficientov. Enota FIR sita izračunava izhodne besede po postopku porazdeljene aritmetike iz delnih vsot koeficientov  $v_i$ . Enota FIR sita je zasnovana tako, da se vektor delnih vsot koeficientov izračunava za vsak otipek posebej. Enota za izračun koeficientov izračunava koeficiente FIR sita z LMS algoritmom po metodi strmega spusta. Ta način smo izbrali zaradi matematične preprostosti, hitrosti in robustnosti izračuna. Za izvedbo enote FIR sita smo uporabili programirno vezje XC4013E, enoto za izračun koeficientov smo izvedli s programirnim vezjem XC4020E. Izvedba s FPGA vezji zahteva drugačne algoritme za izvajanje matematično logičnih operacij. Ko so algoritmi v FPGA vezju enkrat realizirani za adaptivna FIR sita nižjih stopenj, jih ni težko razširiti na sita višjih stopenj. Vsi algoritmi izvajanja aritmetično logičnih operacij delujejo z vnaprej predvidenim številom čakalnih stanj.

Obe enoti smo med sabo povezali, tako da sta skupaj tvorili adaptivno sito z dolžino vhodno-izhodne besede 16 bitov, dolžina notranje aritmetike se zaradi potrebe po robustnosti sita giblje med 16 in 21 biti. Pri tem je perioda vzorčenja vhodne besede  $T_V = 10 \mu s$  ob frekvenci osnovne ure 20MHz.

Rezultati smo zaenkrat dobili le s simulacijo obeh enot s programskim paketom OrCAD ob podpori razvojnega orodja XACT 5.2 firme Xilinx. Pri rezultatih smo prikazali uspešno izločitev komponente motilnega signala v časovnem in frekvenčnem prostoru. Opravili smo tudi primerjavo med rezultati, dobljenimi s simulacijo obeh enot, in rezultati, dobljenimi na osnovi matematičnega opisa.

## Literatura

- /1/ Simon Haykin, Adaptive Filter Theory, Second Edition, Prentice-Hall, 1986, 1991.
- /2/ Bill Allaire, Bud Fischer, Block Adaptive Filter, Application note, Xilinx, XAPP 055, (Version 1.1), Januar 9, 1997.
- /3/ D. Čeh-Ambruš, I. Kramberger, Z. Kakčič, Razvoj razširitvene kartice s signalnim procesorjem za PCI vodilo, Informacije MIDE M, letn. 31, št. 1, str. 48-52.
- /4/ Ken Chapman, Constant coefficients Multipliers for the XC4000E, XAPP 054, (Version 1.1), December 11, 1996.
- /5/ D. Osebik, B. Jarc, M. Solar, R. Babič A 30 Tap FIR Filter realization with FPGA Circuits, Workshop on Systems, Signals and Image Processing, June 3-5, 1998, Zagreb, Croatia. - Zagreb : University of Zagreb, Faculty of Electrical Engineering and Computing, 1998. - Str. 86-89.
- /6/ Rudolf Babič, Dinamika izhodnega signala pri kaskadni obliki izvedbe nerekurzivnih digitalnih sit, Informacije MIDE M, 2001, letn. 32 št. 3, str. 153-159.
- /7/ Grogor Ray Goslin, A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance, Xilinx, Inc, San Jose, 1995.
- /8/ Hanho Lee, Gerald E. Sobelman, FPGA-based FIR filters using digit-serial arithmetic, ASIC Conference and Exhibit, 1997. Proceedings., Page(s): 225-228, Tenth Annual IEEE International, 1997.
- /9/ Sen M. Kuo, Dennis R. Morgan, Active Noise Control Systems, John Wiley & Sons, Inc., 605, 1996.
- /10/ D. Osebik, R. Babič, B. Horvat, Adaptivna digitalna sita v strukturi porazdeljene aritmetike, Informacije MIDE M, 2001, letn. 32 št. str.: 160-168.
- /11/ Xilinx, The Programmable Logic Data Book, San Jose, 1997.
- /12/ Signal Processing Toolbox User's Guide, 1988 - 1998 by The MathWorks, Inc.
- /13/ OrCAD Express User's Guide, First edition 30 November 1998, Copyright © 1998 OrCAD, 9300 SW Nimbus Ave. Beaverton, OR 97008 USA.

*izr. prof. dr. Rudolf Babič, tel, (02) 220 7230,  
E-mail, rudolf.babic@uni-mb.si  
mag. Davorin Osebik, tel, (02) 220 7238,  
E-mail, davorin.osebik@uni-mb.si*

*Univerza v Mariboru  
Fakulteta za elektrotehniko,  
računalništvo in informatiko  
Smetanova 17, 2000 Maribor  
tel.: (02) 220 7000, Fax: (02) 251 1178*

*Prispelo (Arrived): 23.05.2002 Sprejeto (Accepted): 28.06.2002*

# AN ON-CHIP RFID RECEIVER STAGE

Anton Štern, Janez Trontelj

Faculty of Electrical Engineering, University of Ljubljana, Slovenia

**Key words:** RFID, reader, receiver, CMOS, amplitude demodulation, noise, smart card

**Abstract:** The article describes the development of an on-chip receiver stage intended to be used for receiving the transponder's modulation in a single chip low power RFID reader system for the ISM 13.56MHz frequency range. The operating is based on a low noise double envelope follower for AM detection. The receiver gain and bandwidth can be optimized to be compatible with all ISO RFID communication protocols and with customer defined protocols from a low frequency direct modulation systems up to 848kHz subcarrier modulation. The receiver stage design is described and the measurement results of the implemented prototype are presented.

## Sprejemnik za RFID integrirani bralnik

**Ključne besede:** RFID, bralnik, sprejemnik, CMOS, amplitudna demodulacija, šum, pametna kartica

**Izvilleček:** V prispevku opisani sprejemnik je namenjen sprejemu modulacije kartice v celoti integriranem bralniku brezkontaktnih pametnih kartic v ISM frekvenčnem področju 13.56MHz. Amplitudni demodulator je nizko šumni vhodni sledilnik obeh ovojnic. Nastavljivo ojačenje in nastavljiva frekvenčna širina sprejemnika omogočata prilagoditev sistema vsem ISO standardiziranim in drugim uporabniško definiranim načinom komunikacije s pametnimi karticami od direktne modulacije do modulacije podnosilnega vala na 848kHz. Opisano je načrtovanje sprejemnega dela veza, podani pa so tudi rezultati, izmerjeni na izdelanih vezjih.

### 1. Introduction

In the recent years the RFID procedures have become commonly used in many areas of human activity. At some special applications, like car immobilizing, the number of the transponders is almost the same as the number of the readers (e. g. for two keys there is one reader on each side of the car and one at the ignition lock). Consequently, on the market there is a strong need for a small and cheap reader – the integrated reader. The receiver stage, which is described in this article, is an important part of such RFID system.

### 2. The receiver

The receiver described is intended to be used in a single chip 13.56MHz RFID integrated reader. In an RFID system with defined geometry and defined output power the operating range is limited by the sensitivity of the receiver on one hand and the dynamic range of the receiver on the other hand.

The reader chip which has been designed is highly versatile and can operate with all ISO standardized coding and modulation schemes /1,3/. It can also operate with known customer defined and partly standardized coding and modulation techniques. The following requirements must be covered by the receiver part: the receiver must be capable of operating with the 212kHz, 424kHz and 848kHz subcarriers. The operating with the transponders without the use of the subcarrier is also needed. Different coding protocols (OOK, BPSK, NRZ) request different receiver response at the beginning of the data stream.

Consequently, the gain and frequency response of the receiver must be optimized for each subcarrier frequency and coding used due to different noise bandwidth to achieve reasonable sensitivity. Since the receiver is intended to be used together with an on-chip low power transmitter the expected operating range for a card size transponder is between 10cm and 20cm depending on the reader antenna size and transponder type. The receiver sensitivity needed is in a range of milli-volts carried on a 5V<sub>PP</sub> RF carrier. On the other hand the receiver output signal should not be corrupted when the transponder is close to the reader antenna where modulation signal can reach up to one half of a volt on the RF carrier.

Figure 1 presents the block diagram of the receiver. It is composed of an envelope detector, DC level cancellation and gain stage, second order low-pass with gain, high-pass with gain and a digitizing circuit. The low-pass and high-pass corner frequencies and receiver gain can be set by option pins to optimize receiver performance to system demands.

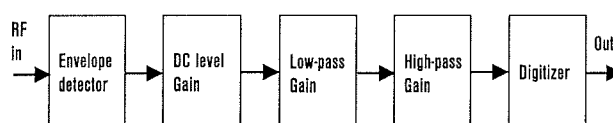


Figure 1: Block diagram of the receiver

The envelope detector is the most critical part of the receiver due to the direct impact of the detector input noise to the system noise and sensitivity performance. The schematic is presented in figure 2. It is composed of NMOS and PMOS source follower which detect upper and lower

envelope. Both signals with partly suppressed carrier are impedance decoupled with two voltage followers and then subtracted. The detector's loss is approximately 1.2dB due to the body effect of the MOS transistors. This loss is compensated in the subtraction stage. Since there is no gain in the detector the input followers, amplifiers and gain setting resistors used must ensure low noise contributions. The expected input noise is between  $25\text{nV}/\sqrt{\text{Hz}}$  and  $40\text{nV}/\sqrt{\text{Hz}}$  with process variations and in temperature range from  $-40^\circ\text{C}$  to  $120^\circ\text{C}$ .

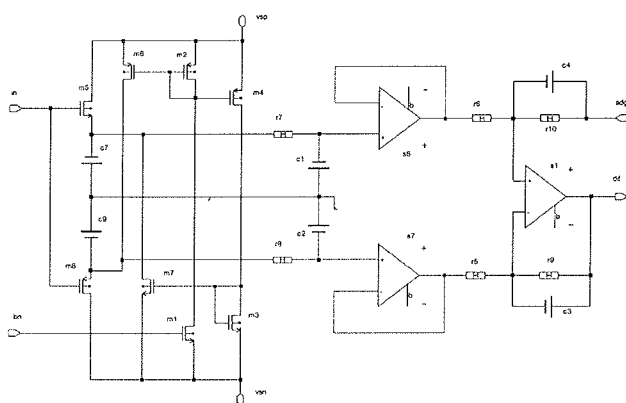


Figure 2: Envelope detector

Figures 3 and 4 present simulation results of the receiver chain. Figure 3 presents a time domain simulation where the input signal is composed of three AM bursts on an RF carrier each of a different amplitude (upper panel). In the middle one there is the signal after filtering and gain stages and in the bottom there is the digitized output signal.

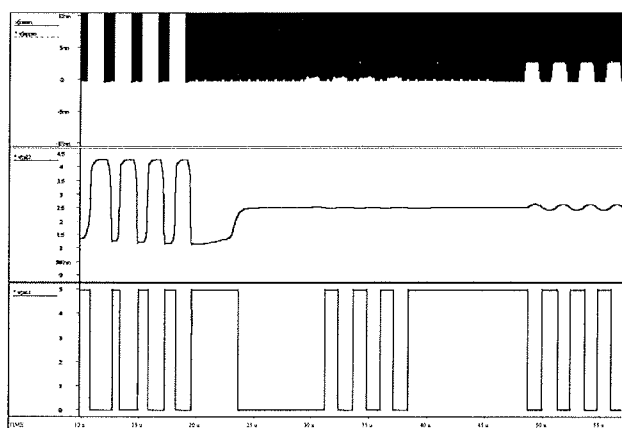


Figure 3: Time domain simulation results

The frequency domain simulation in figure 4 is used to examine the gain and filtering properties of the circuit (upper panel) and the noise levels (middle and bottom panel).

The layout of the receiver is shown in figure 5. In the bottom left hand corner there is the input envelope detector with source followers and current sources, followed by shielded filtering capacitors. In the right and on the top

there are amplifiers and in the middle there are capacitors used in high pass filter.

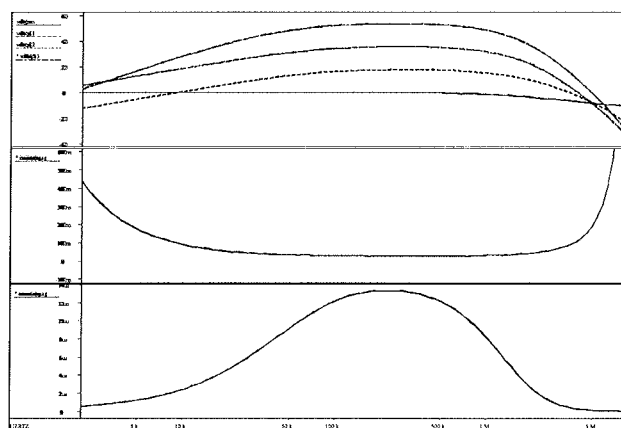


Figure 4: Frequency domain simulation results

The circuit was integrated in a double metal, double poly  $1\mu\text{m}$  C-MOS process together with the other blocks needed to complete RFID reader system. A significant care has been taken when the different blocks were put together to minimize the influence of one block to the other. Especially the capacitive cross-talks and substrate currents from the transmitter driver to the input stages of the receiver can corrupt noise and stability properties.

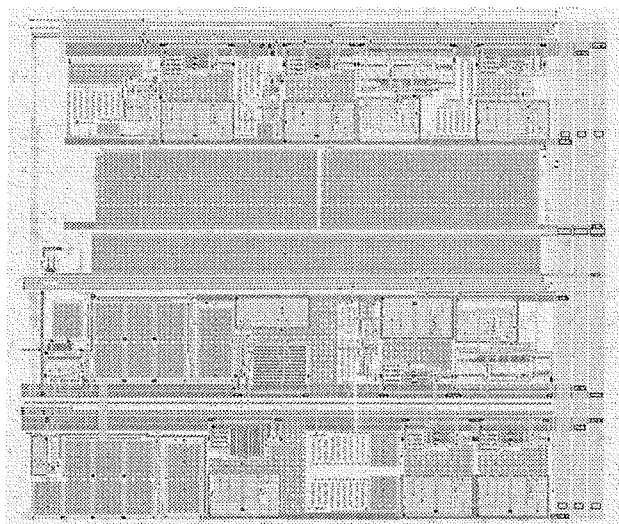


Figure 5: Layout of the receiver

The fabricated samples were tested for noise, bandwidth and sensitivity properties. In table 1 there are the results of the output spot noise level measured at different frequencies and for different bandwidth settings. In the last column there is the output noise integrated over the entire receiving bandwidth. We can calculate that typical input spot noise is approximately  $27\text{nV}/\sqrt{\text{Hz}}$  which is expected. The integrated input noise is  $35\mu\text{VRMS}$  with a high bandwidth setting. The noise spectrum is shown on figure 6.

The receiver noise is not the only noise source in the system. The transmitter's oscillator has its own phase noise which goes through the transmitter and matching network to the receiver's input. The oscillator's noise sidebands are demodulated, thus increasing the receiver's input noise. The transmitter's oscillator is designed as a quartz crystal type which exhibits lowest possible noise sidebands for a reasonable price. The oscillator's noise increases the receiver's input spot noise for approximately 30% at 100kHz. At frequencies below 100kHz the spot noise rise is higher than 30% and above 100kHz it is lower than 30%, but the RMS noise is also increased for approximately 30% in 400kHz bandwidth. This means that there is no dominant noise source in the system which could be easily decreased to efficiently improve noise and sensitivity performance of the system.

BW set to	Spot noise at 25kHz	Spot noise at 100kHz	Spot noise at 210kHz	Spot noise at 840kHz	RMS noise
kHz	$\mu\text{V}/\sqrt{\text{Hz}}$	$\mu\text{V}/\sqrt{\text{Hz}}$	$\mu\text{V}/\sqrt{\text{Hz}}$	$\mu\text{V}/\sqrt{\text{Hz}}$	$\text{mV}_{\text{RMS}}$
1000	7,2	7	6,5	5,4	8,8
400	7,2	7	6	1,6	5,4
200	7,2	6	3,4	0,35	3,7

Table 1: Noise measurement results at gain setting 48dB

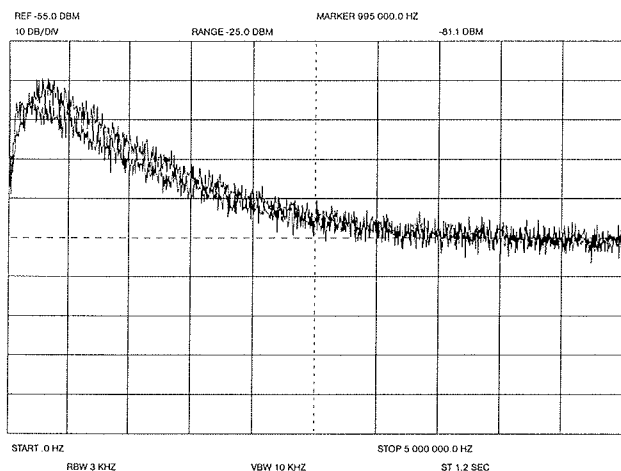


Figure 6: Noise spectrum

Low pass filter high cut-off frequency can be set to approximately 1000kHz, 400kHz and 200kHz. The belonging measured values are 1070kHz, 420kHz and 240kHz respectively at room temperature. On figure 7 the frequency response plot is presented. The gain can be set to 48dB and 42dB. The sensitivity can be set to 0.6mV<sub>PP</sub> for lower frequency range (200kHz) and to 1.5mV<sub>PP</sub> for higher frequency ranges (400kHz and 1000kHz) on a 5V<sub>PP</sub> RF carrier.

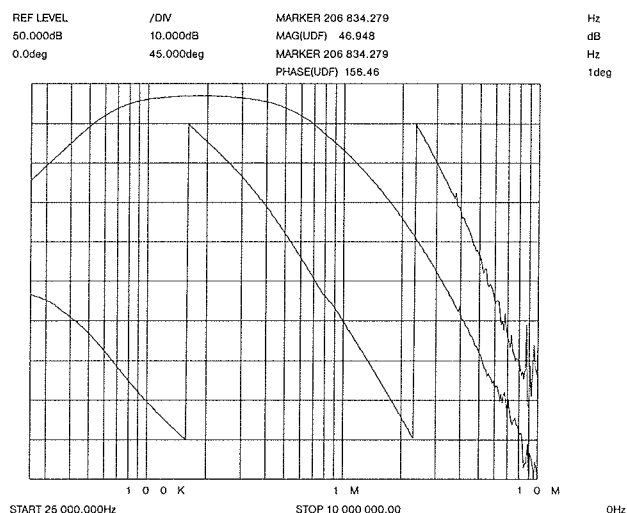


Figure 7: Frequency response of the receiver (BW set to 400kHz)

Receiver was tested in a complete RFID system together with the associated low power transmitter. The transmitter's output power was 20dBm and the reader antenna diameter was 12cm, while the Q factor was 23. The transponder was a card size transponder with a coil inductance of 1.42 $\mu\text{H}$ . The resonance at 13.56MHz is achieved by a capacitance of 97pF. The small signal quality factor  $Q=45$  was decreased to  $Q=12$  when the transponder was put in a minimum operating field strength. Both the transponder and the reader coil were tuned to 13.56MHz $\pm$ 150kHz.

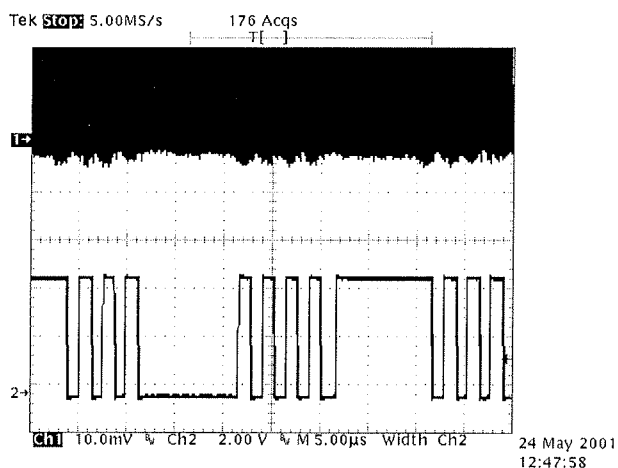


Figure 8: 424kHz subcarrier, on/off keying, distance 16cm Upper trace- RF in, lower trace - digitized output

Receiving the transponder's modulation was checked for different transponder's subcarrier frequencies at distances from the maximum operating range to the minimum operating range, which means that the transponder is at the surface of the reader antenna. On/off keying and BPSK coding was also checked. Figure 8 presents receiver's input signal (upper trace) and correct digitized receiver's



output at 16cm distance, which is close to the maximum operating range of a system when 424kHz, 53kbit/s OOK coding is used. Figure 9 presents the same signals at 848kHz, 53kbit/s OOK, but the transponder is set very close to the reader antenna. The envelope magnitude is 0.5V<sub>PP</sub> and DC level shift at modulation start and stop is approximately 0.25V. The DC level cancellation circuit is fast enough not to lose pulses at the beginning of the burst. Operating range is 16-17cm when lower frequency system is used and 15-16cm when higher frequency subcarrier system is used.

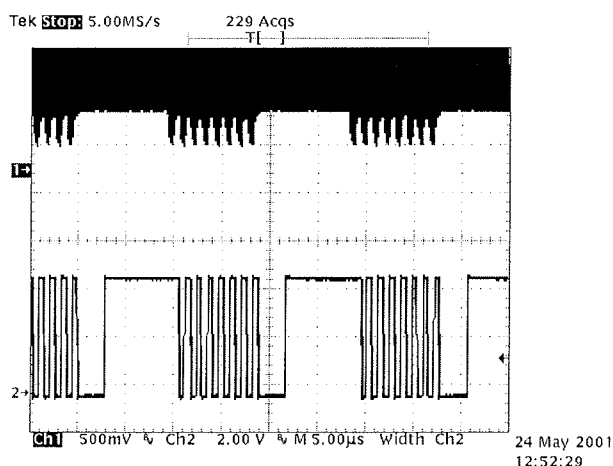


Figure 9: 848kHz subcarrier, on/off keying, distance 0.5cm Upper trace- RF in, lower trace – digitized output

### 3. Conclusion

The receiver stage for a single chip RFID integrated circuit was designed in a 1µm CMOS process. A special attention was paid in the layout phase to minimize capacitive and substrate cross-talks between the receiving and transmitting part not to corrupt receiver input noise or stability of the system. The chip was evaluated both as a stand-alone AM receiver and as a part of an RFID transmitter-reader system. The input noise level of 7µV/√Hz at 100kHz enables a system sensitivity of 1.5mV<sub>PP</sub> at 1000kHz bandwidth. When operating with an associated on-chip low power transmitter (20dBm) the single chip reader system can

communicate with a card size transponder in a 15cm-17cm operating range depending on a system used.

### 4. Acknowledgement

The authors would like to thank to the Ministry of Education, Science and Sport for the support with the research program.

### 5. References

- /1/ ISO/IEC 14443-2 *Identification cards – Contactless integrated circuit(s) cards - Proximity cards – Part 2: Radio frequency power and signal interface*, Secretariat ISO/IEC JTC1/SC17, APACS, London, 1999
- /2/ ISO/IEC CD10373-6 *Identification cards – Test Methods – Part 6: Proximity cards*, Secretariat APACS, 1999
- /3/ ISO/IEC 15693-2 *Identification cards – Contactless integrated circuit(s) cards - Vicinity cards – Part 2: Radio frequency power and signal interface*, Secretariat ISO/IEC JTC1/SC17, APACS, London, 1998
- /4/ ISO/IEC CD10373-7 *Identification cards – Test Methods – Part 7: Vicinity cards*, Secretariat DIN, 1999
- /5/ K. Finkenzeller, *RFID Handbook, Radio-Frequency Identification Fundamentals and Applications*, John Wiley and Sons Ltd, West Sussex, England 1999
- /6/ Y. Nakajima, T. Mitsuhashi, *Development of a Memory Label System for Consumer VCRs*, p. 21-30, IEEE Transactions on Consumer Electronics, Vol. 45, No 1, February 1999
- /7/ A. Štern, *Zasnova brezkontaktnega bralnika*, magistrsko delo, Fakulteta za elektrotehniko, Ljubljana, April 2001

Anton Štern, Janez Trontelj  
Faculty of Electrical Engineering  
Tržaška 25, 1000 Ljubljana, Slovenia  
E-mail: tone@kalvarija.fe.uni-lj.

Prispelo (Arrived): 27.05.2002

Sprejeto (Accepted): 28.06.2002

# EXPLOITING SYMBOLIC MODEL CHECKING FOR SENSING STUCK-AT FAULTS IN DIGITAL CIRCUITS

Aleš Časar, Zmago Brezočnik, Tatjana Kapus

University of Maribor, Faculty of Electrical Engineering and Computer Science,  
Slovenia

**Keywords:** stuck-at faults, symbolic model checking, automatic test pattern generation, testing, CTL formulas, finite state machine, binary decision diagrams

**Abstract:** This paper presents algorithms for automatic test pattern generation for discovering stuck-at faults in sequential digital circuits or proving that there are no stuck-at faults in the given circuit. A circuit is represented as a finite state machine. Properties for stuck-at faults expressed with CTL formulas which are valid in the circuit with stuck-at faults and generally not valid in the good circuit are generated. Validity of the formulas is checked by symbolic model checking, and for invalid formulas counterexamples are constructed which guide the circuit to the states which prove the absence of stuck-at faults. Test patterns guide the circuits exactly as the counterexamples. Experimental results for a set of benchmark circuits together with the time and space complexity analysis of the algorithms are also given.

## Uporaba simboličnega preverjanja modelov pri zaznavanju zatičnih napak v digitalnih vezjih

**Ključne besede:** zatične napake, simbolično preverjanje modelov, avtomatsko generiranje testnih vzorcev, testiranje, formule CTL, končni avtomat, binarni odločitveni grafi

**Povzetek:** V članku predstavljamo algoritme za avtomatsko generiranje testnih vzorcev, s pomočjo katerih pri sekvenčnih digitalnih vezjih odkrivamo zatične napake oziroma pokažemo, da zatičnih napak v danem primerku vezja ni. Vezje predstavimo kot končni avtomat. Za zatične napake generiramo lastnosti v obliki formul CTL, ki so veljavne v vezju z zatičnimi napakami in praviloma neveljavne v dobrem vezju. S simboličnim preverjanjem modelov preverimo veljavnost formul in za neveljavne formule skonstruiramo protiprimer, s katerimi vezje pripeljemo v stanja, ki dokažejo odsotnost zatičnih napak. Testni vzorci so sestavljeni tako, da izvajanje vezja vodijo po poti protiprimerov. Teoretične raziskave so podkrepljene z eksperimentalnimi rezultati. Prikazana je tudi analiza časovne in prostorske zahtevnosti.

### 1 Introduction

Testing of newly produced digital circuits is a necessity. Since circuits are becoming larger and larger, it is impossible to perform exhaustive testing of the circuits nowadays. Therefore, a suitable trade-off between exhaustive testing and speed of testing (length of test patterns) should be made. We tend to discover (or prove their absence) as many circuit faults as possible but at moderate test pattern length. Many different faults can occur in a circuit, but stuck-at faults are the most common ones. Hence, we introduce the method which will find all possible single stuck-at faults in the circuit or prove that there is no stuck-at fault present in the circuit.

Because enumeration methods [10] do not perform well with large circuits, we propose to use symbolic methods [5, 6]. A circuit is represented as a fi-

nite state machine (FSM). FSMs are represented as Boolean functions and these with binary decision diagrams (BDDs). Properties of FSMs which are to be checked by symbolic model checking are expressed with CTL formulas.

For every possible single stuck-at fault, a property which is valid in the circuit with that stuck-at fault and generally invalid in the good circuit can be generated. When the property is invalid, a counterexample can be found. If the test pattern guides the circuit exactly as the counterexample, the absence of the treated stuck-at fault is proved if testing with this test pattern ends successfully.

We neither deal with other types of possible faults nor with multiple stuck-at faults in this paper. Practice suggests that also multiple stuck-at faults can be discovered in most cases, but we did not prove that. A mentionable limitation of the proposed method is also

the necessity of the insight into the circuits (logic values stored in flip-flops).

In Section 2 we briefly show how to represent FSMs with BDDs, describe searching of reachable states, and symbolic model checking in CTL. Section 3 describes the methods of searching counterexamples and witnesses. The main part of the paper is Section 4 where we present algorithms for generation of properties for stuck-at faults. Experimental results for benchmark circuits with time and space complexity analysis are given in Section 5. We conclude with some discussion and plans for future work.

## 2 Preliminaries

Binary decision diagrams (BDDs) are compact canonical representations of Boolean functions [2]. BDDs can be used for representing and manipulating sets if we represent sets by means of their characteristic functions [5, 6].

A deterministic *finite state machine* (FSM)  $M$  is a sextuple  $M = (\Sigma, \mathcal{S}, \mathcal{O}, \delta, \lambda, s_0)$ , where  $\Sigma$  is a finite set of input symbols,  $\mathcal{S}$  a finite set of states,  $\mathcal{O}$  a finite set of output symbols,  $\delta: \mathcal{S} \times \Sigma \rightarrow \mathcal{S}$  a state transition function,  $\lambda: \mathcal{S} \times \Sigma \rightarrow \mathcal{O}$  an output function, and  $s_0 \in \mathcal{S}$  an initial state.

If we want to realize a FSM by a digital circuit, we have to encode the sets  $\mathcal{S}$ ,  $\Sigma$ , and  $\mathcal{O}$  by binary symbols (e.g. 0 and 1). States are encoded by state variables. At least  $n = \lceil \log_2 |\mathcal{S}| \rceil$  state variables,  $m = \lceil \log_2 |\Sigma| \rceil$  input variables, and  $l = \lceil \log_2 |\mathcal{O}| \rceil$  output variables are needed. Let  $\mathcal{Y}$ ,  $\mathcal{X}$ , and  $\mathcal{Z}$  represent the set of state variables, the set of input variables, and the set of output variables, respectively.

Once the states and the input symbols of the circuit are encoded, next state variables are functions of present state variables and input variables. We denote next state variables by an added prime (') and write a transition function of a state variable  $y_i$  as

$$y'_i = \delta_i(y_0, y_1, \dots, y_{n-1}, x_0, x_1, \dots, x_{m-1}) \quad (1)$$

for  $i = 0, 1, \dots, n-1$ . We rather introduce transition relations

$$T_i = y'_i \Leftrightarrow \delta_i(y_0, y_1, \dots, y_{n-1}, x_0, x_1, \dots, x_{m-1}). \quad (2)$$

Namely, relations have much greater expressive power than functions [3]. Transition relations  $T_i$  can be combined by taking their conjunction to form the *monolithic transition relation*  $T = T_0 \cdot T_1 \cdot \dots \cdot T_{n-1}$ . After the encoding, output variables are functions of present state variables and input variables. We write

$$z_i = \lambda_i(y_0, y_1, \dots, y_{n-1}, x_0, x_1, \dots, x_{m-1}) \quad (3)$$

for  $i = 0, 1, \dots, l-1$ .

Let us very briefly show how we search reachable states of a FSM [1, 5, 6]. Let  $\mathcal{S}_i$  denote a set of states

reachable in at most  $i$  steps.  $\mathcal{S}_0$  represents a set of initial states. In our case we have  $\mathcal{S}_0 = \{s_0\}$ . In general, a set of states reachable in at most  $i$  steps is represented by

$$\mathcal{S}_i = \mathcal{S}_{i-1} \cup \left\{ s' \mid \exists a \exists s [a \in \Sigma \wedge s \in \mathcal{S}_{i-1} \wedge \delta(s, a) = s'] \right\}. \quad (4)$$

We continue with this procedure until in a step  $k$  no new state is reached. In any case, this happens sooner or later, because we deal with FSMs, where the set of states  $\mathcal{S}$  is finite. Then,  $\mathcal{S}_k = \mathcal{S}_{k-1}$  is a set of all reachable states.

The logic which we use to specify properties of FSMs is a propositional temporal logic of branching time, called *Computation Tree Logic* — CTL [9]. In this logic, each of the usual future time operators of linear-time temporal logic ( $G$  — *globally or invariantly*,  $F$  — *sometimes in the future*,  $X$  — *next time*,  $U$  — *until*) must be immediately preceded by *path quantifier*  $A$  (for all computation paths) or  $E$  (for some computational path). We thus obtain eight different CTL operators:  $AG$ ,  $EG$ ,  $AF$ ,  $EF$ ,  $AX$ ,  $EX$ ,  $AU$ ,  $EU$  [1, 5, 6].

CTL formulas are constructed from *atomic propositions* using Boolean connectives and CTL operators. In the case of circuit verification, the set of atomic propositions is equal to the set  $\mathcal{Y}$  of state variables of the circuit.

## 3 Searching for Counterexamples and Witnesses

One of the most important extensions of symbolic model checking is the ability to search counterexamples for some invalid and witnesses for some valid CTL formulas [8]. Counterexamples explain why a given CTL formula is invalid in a FSM, and witnesses show why a given CTL formula is valid.

### 3.1 Counterexamples

Counterexample is a path in the computation tree which shows why a given CTL formula is invalid. Actually, this is evident from the last state on the path, but FSM must be guided to this state to demonstrate invalidity of the formula.

It is impossible to find counterexamples for all invalid formulas. According to the definition of CTL formulas [5], they are constructed from atomic propositions, Boolean operators, and CTL operators. Let us look how these three constructs affect searching of counterexamples.

Searching of a counterexample for an atomic proposition is a trivial task. Such a formula represents the characteristic function of the set of states where the formula is valid. Because counterexamples are searched only for invalid formulas, it suffices to check if the current state of the FSM is not in that set.

Although there are 16 binary Boolean operators, all of them can be expressed by negation, conjunction, and disjunction. Searching for a counterexample for negation of a function means the same as searching for a witness for that (non-negated) function. Therefore, searching of a counterexample for  $\bar{f}$  is equivalent to searching of witness for  $f$ . Searching of witnesses will be described in Section 3.2.

When dealing with conjunction, a counterexample for a formula of the form  $f \cdot g$  should be found. Because the formula is invalid (counterexamples are searched only for invalid formulas), at least one of the functions  $f$  or  $g$  is invalid. To find a counterexample for the whole function  $f \cdot g$ , it is enough to find either a counterexample for  $f$  or a counterexample for  $g$ . Of course, in the case one of the formulas  $f$  or  $g$  is valid, we should find a counterexample for the other one, which is invalid.

It is not possible to find a counterexample for disjunction in all cases. If a formula of the form  $f + g$  is invalid, then both formula  $f$  and formula  $g$  are invalid. To prove that, one should prove both simultaneously. Generally this is impossible to do with just one counterexample, but there are exceptions. If one of the formulas  $f$  or  $g$  is of such a kind that the counterexample for it is not a path but just a single state (this happens when a formula does not contain any CTL operators), then also a counterexample for disjunction can be found. We search a counterexample for the other formula and at the end also check that the current state is not in the set of states which our formula without CTL operators is the characteristic function for.

Let us now look at another interesting exception. Triple Boolean operator  $ite(f, g, h)$  can be written also as  $f \cdot g + \bar{f} \cdot h$ . If  $f$  represents a Boolean formula (without temporal operators) in such a construction, then exactly one of the disjunctives is invalid because of  $f$  in every state. There are two cases:

1. Formula  $f$  is valid in the given state. Operand  $\bar{f} \cdot h$  is invalid then and we have to show that operand  $f \cdot g$  is also invalid. It is known that factor  $f$  is valid, therefore we have to show that factor  $g$  is invalid. Actually we have to search for a counterexample for  $g$ .
2. Formula  $f$  is invalid in the given state. In that case operand  $f \cdot g$  is invalid because of  $f$ , and the only thing we have to prove is that operand  $\bar{f} \cdot h$  is also invalid. Because  $\bar{f}$  is valid here,  $h$  is invalid accordingly. Therefore, a counterexample for  $h$  should be searched for.

After explaining Boolean operators, let us devote now to CTL operators. Counterexamples can be searched only for CTL operators with universal path quantifier  $A$ . Namely, only these operators state that on *every* computation path a formula is valid, and a counterexample is *one* computation path where this formula is invalid.

$AX f$  states that formula  $f$  is valid in all successors of the present state. Problem of searching a counterexample for an invalid formula of such a kind is to find a successor of the present state where formula  $f$  is invalid. When an adequate successor is found, a counterexample for formula  $f$  should be found.

According to formula  $AF f$ , every computation path from the present state should lead to a state where formula  $f$  is valid. If this is not true, we have to find a path where we will *never* reach a state where formula  $f$  is valid. Since computation paths are infinite, how at all can we show that something never happens along a given path? Paths are infinite indeed, but they lead over finite set of states. Therefore, at least one state on the path must occur repeatedly. When an already visited state is reached, the path from this state back to itself can safely be repeated infinitely often. Actually, the path ends with a cycle. It is not necessary that the initial part of the path is part of the cycle. In that case, the path is composed of the prefix and the cycle as shown in Fig. 1. It is enough to find a prefix

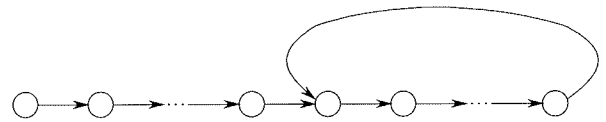


Figure 1: A path with prefix and cycle

and a cycle to find such a path. When we return to an already visited state at traversing the states, we have found a counterexample as a proof that formula  $AF f$  is invalid indeed. Formula  $f$  must be invalid in every state along that path, therefore, counterexamples for itself should be found in every state on that path. However, those "nested" counterexamples must not go off the path, since our computation path would transform to a computation tree otherwise. To avoid such situations, formula  $f$  should not contain any CTL operators.

Formula  $AG f$  says that on all the paths, formula  $f$  is valid all the time. In order to prove its invalidity, a path leading to a state where formula  $f$  is invalid has to be found. As with operator  $AX$ , from there on a counterexample for formula  $f$  has to be found in order to confirm its invalidity in the last state of the path for  $AG f$ .

The last CTL operator containing the universal path quantifier is  $A[f U g]$ , which says that on all the paths, formula  $f$  is valid until a state where formula  $g$  is valid. In order to refute validity of formula  $A[f U g]$ , either an infinite path on which formula  $g$  is never valid or a path which leads to a state where formula  $f$  is invalid and on which formula  $g$  is never valid has to be found. The case of the infinite path is similar to the counterexample construction with operator  $AF$  — only a prefix and a cycle terminating the infinite path have to be found. As in the case of  $AF$ , formula  $g$  must not contain any CTL operators. If one decides to search for a path leading to a state where formula  $f$  is invalid, one must,

of course, continue with a counterexample for  $f$ .

### 3.2 Witnesses

Witness is a path in the computation tree which indicates why a CTL formula considered is valid. In fact, the validity is evident from the last state of the path, but the FSM has to be led to that state in order for the path to demonstrate the validity.

As with counterexamples, witnesses cannot be found for every valid CTL formula. Since witnesses are in a sense dual to counterexamples, problems occur exactly with the CTL formulas dual to those CTL formulas, validity of which cannot be demonstrated by counterexamples. We now make an overview of the structural elements of CTL formulas and their influence on searching of witnesses.

Searching of a witness for an atomic proposition is in fact the same as searching of a counterexample. The difference is that witnesses are searched for valid formulas. Therefore, it must be checked if the current state of the FSM is in the set determined by the characteristic function in the form of the given atomic proposition.

Searching of a witness for negation of a function means the same as searching of a counterexample for the (non-negated) function. It follows that searching of a witness for  $\bar{f}$  is equivalent to searching of a counterexample for  $f$ . Searching of the counterexamples is described in detail in Section 3.1.

Another example is searching of a witness for a disjunction  $f + g$ . Since the formula is valid (as witnesses are searched only for valid formulas), at least one of the functions  $f$  and  $g$  is valid. In order to find a witness for function  $f + g$ , it therefore suffices to find either a witness for  $f$  or a witness for  $g$ . If one of the formulas is invalid, a witness for the valid one has to be found, of course.

With counterexamples, there were some problems with disjunction, whereas due to duality of witnesses, similar problems now occur with conjunction, which is dual to disjunction. It is not always possible to find a witness for conjunction. If a formula of the form  $f \cdot g$  is valid, then formula  $f$  and formula  $g$  must be valid. In order to demonstrate that, validity of both should be demonstrated *simultaneously*. As a witness must be a path, which must not have branches, this is generally not possible. Exceptions, however, exist also in this case. If one of the formulas  $f$  and  $g$  is such that its witness is a path containing just one state (this is the case when the formula does not contain any CTL operators), then a witness for the conjunction can be found as well. A witness for the other formula has to be found, and at the end, it has to be checked if the current state is also in the set of states whose characteristic function is the formula without CTL operators.

Finally, let us look at the CTL operators and their influence on searching of witnesses. Witnesses can only be found for the CTL operators containing exist-

tential path quantifier  $E$ . They say that there *exists* a computation path where a formula is valid, and a witness is simply *one* path which confirms the existence and validity of the formula.

Formula  $EXf$  says that there exists a successor of the current state where formula  $f$  is valid. In order to find a witness for a valid formula of this form, a successor of the current state where formula  $f$  is valid has to be found. A witness for  $f$  must then be found from the successor state on.

Formula  $EFf$  says that there exists at least one path leading to a state where formula  $f$  is valid. In order to prove its validity, a path leading to such a state has to be found and from there on, a witness for formula  $f$  has to be found.

Formula  $EGf$  says that there exists an infinite path on which formula  $f$  is valid all the time. We already know how to deal with infinite paths. A prefix and a cycle at the end of an infinite path have to be found, such that formula  $f$  is valid in every state of the path. When during passing from state to state, an already visited state is reached for the first time, the searching is finished. Since validity of formula  $f$  must be confirmed by the witness along the whole path, which must not be abandoned, the confirmation is possible only if formula  $f$  does not contain any CTL operators.

We already know that  $E[fUg]$  says there exists a path where formula  $f$  is valid until a state is reached where formula  $g$  is valid. In order to find a witness, one of such paths must, therefore, be found. Formula  $f$  must again not contain any CTL operators, and starting from the last state, a witness for the validity of formula  $g$  in that state must be found.

### 3.3 Realization

In order to implement symbolic model checking, we implemented only resolution of three CTL operators,  $EXf$ ,  $E[fUg]$ , and  $EGf$ , whereas the other five were expressed with them [5]. The same approach can be followed to realize searching of counterexamples and witnesses. For unrealized operators the searching can be realized as follows:

- searching of a counterexample for  $AXf$  is replaced by searching of a witness for  $EX\bar{f}$ ,
- searching of a witness for  $EFf$  is replaced by searching of a witness for  $E[1Uf]$ ,
- searching of a counterexample for  $AFf$  is replaced by searching of a witness for  $EG\bar{f}$ ,
- searching of a counterexample for  $AGf$  is replaced by searching of a witness for  $E[1U\bar{f}]$ ,
- searching of a counterexample for  $A[fUg]$  is replaced either by searching of a witness for  $E[\bar{g}U\bar{f} \cdot \bar{g}]$  or by searching of a witness for  $EG\bar{g}$ .



To find out if formula  $EXf$  is valid in a given state  $s$  using symbolic model checking, we start by the set  $S_f$  of states in which formula  $f$  is valid. This set will also be used to find a witness for validity of formula  $EXf$  in state  $s$ , which can be done if it is found that formula  $EXf$  is valid in state  $s$ . In order to find this witness, we have to find a successor of state  $s$  in which formula  $f$  is valid. In accordance with formula (4), the set of all successors of state  $s$  is calculated as follows:

$$S' = \{s' \mid \exists a[a \in \Sigma \wedge \delta(s, a) = s']\}. \quad (5)$$

In the set  $S'$ , we have to choose a successor of  $s$  in which formula  $f$  is valid. It follows that state  $s'$  must not only be in set  $S'$ , but also in  $S_f$ , i.e.  $s' \in S' \cap S_f$  must hold. This is illustrated in Fig. 2. The witness for

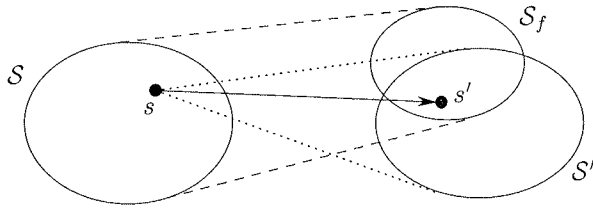


Figure 2: Witness for  $EXf$

formula  $EXf$  in state  $s$  is, therefore, composed of two states,  $(s, s')$ , and is found in one step.

When checking validity of formula  $E[f U g]$ , we start with the set  $S_g$  of all states where formula  $g$  is valid. By gradually adding all such predecessors where formula  $f$  is valid, we obtain the set of all states where formula  $E[f U g]$  is valid. Let  $S_f$  be the set of all states where formula  $f$  is valid, and let  $S^i$  denote the set of all states obtained until the  $i$ -th step of the procedure, the step included. Note that  $S^0 = S_g$ . The situation is shown in Fig. 3. The sets have the follow-

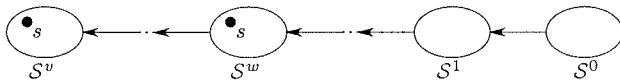


Figure 3: Set of states at checking formula  $E[f U g]$

ing characteristics:

1.  $S^i \subset S_f \cup S_g$  for  $i = 0, 1, \dots, v$ .
2.  $S^{i-1} \subset S^i$  for  $i = 1, 2, \dots, v$ .
3.  $S^i \setminus S^{i-1} \neq \emptyset$  for  $i = 1, 2, \dots, v$ . For all states in this set difference there exists a path to a state in  $S^0$  of length  $i$  and no shorter path exists to any of the states in  $S^0$ .
4.  $S^w$  is the first set on the construction path (and also the smallest set) which contains state  $s$ .
5. For  $S^v$ , there does not exist any successor in which formula  $f$  would be valid but would not already be in  $S^v$ .

In order to find a witness for formula  $E[f U g]$  in a state  $s$ , we must find a path from the state  $s$  in the set  $S^w$  to a state in the set  $S^0$ . A path of the form  $(s_w, s_{w-1}, \dots, s_0)$  will contain exactly  $w$  steps<sup>1</sup>, where for all states,  $s_i \in S^i$  and  $s_w = s$ . It is clear that states from the sets cannot be chosen arbitrarily. For any pair of states  $s_i$  and  $s_{i-1}$ , a transition from  $s_i$  to  $s_{i-1}$  must exist. We, therefore, choose in each step a state  $s_{i-1}$  which will also be in the set of all successors of state  $s_i$ . If the set is denoted by  $S^{i'}$ , then we can write  $s_{i-1} \in S^{i-1} \cap S^{i'}$ . An example for  $w = 3$  is shown in Fig. 4.

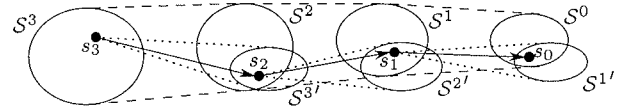


Figure 4: Witness for  $E[f U g]$

When checking a formula  $EGf$  in a state  $s$ , we calculate the set of all states where the formula is valid. Let the set be denoted by  $S$ . From every state in this set there leads an infinite path where  $f$  is valid all the time. Since any state on such a path is also a starting state of an infinite path where formula  $f$  is valid all the time, all the states on the path are in the set  $S$  and the complete path runs within  $S$ . The construction of a witness begins in the state  $s$ . In the  $i$ -th step we choose a state from the intersection of the set  $S_{i-1}'$ , which contains all successors of the current state  $s_{i-1}$ , with the set  $S$ . The general construction step is shown in Fig. 5. If the intersection contains some already visited state,

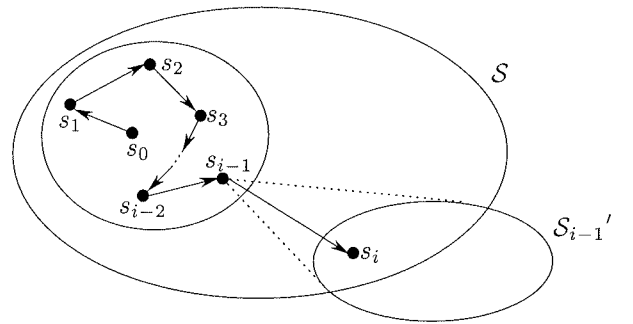


Figure 5: General construction step of the witness for  $EGf$

we choose such a state and the construction of a witness is ended; otherwise, we choose an arbitrary successor and continue the procedure. The algorithm will end anyway, as the set  $S$  contains a finite number of states. We will reach an already visited state in the  $|S|$ -th step at the latest. For a path  $(s_0, s_1, \dots, s_v, \dots, s_w)$  which is a witness, the following is true:

- $s_0 = s$ ;

<sup>1</sup>Of course, a longer path could also serve as a witness. It is, however, useful to have as short witnesses and counterexamples as possible.

- there exists a transition from state  $s_{i-1}$  to state  $s_i$  for  $i = 1, 2, \dots, w$ ;
- $s_w = s_v$  where  $v \in \{0, 1, \dots, w-1\}$ ;
- $s_i \neq s_j$  for  $i, j = 0, 1, \dots, w-1$  and  $i \neq j$ ;
- $w \leq |\mathcal{S}|$ .

The complete witness and the last construction step are shown in Fig. 6.

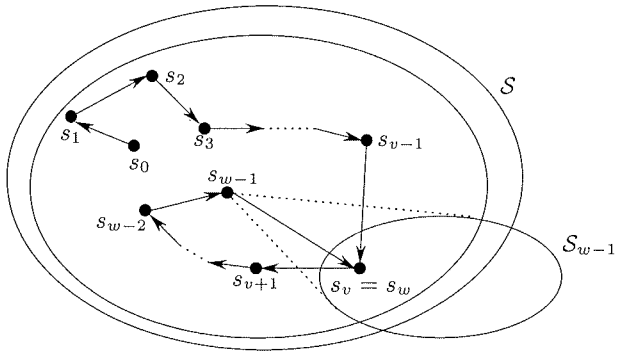


Figure 6: Witness for  $EG f$  with last construction step

## 4 Generation of Properties for Stuck-at Faults

### 4.1 Stuck-at Faults

Digital sequential circuits consist of flip-flops, logic gates, and lines between them. Generally, some lines lead to the circuit, i.e. they are inputs, whereas some are outputs, which lead out of the circuit. Such a circuit can be looked upon as a realization of a binary encoded FSM. Every flip-flop has its state variable, circuit inputs are the inputs of the FSM, and outputs are also its outputs. The state transition function is determined by logic gates and lines which connect them with the flip-flops, and similarly for the output function.

For example, let us look at an arbiter which chooses the request with the highest priority from among the three requests on its inputs. The arbiter is shown in Fig. 7. The smaller the number of a requesting device, the higher its priority. State variables of the FSM realized by the arbiter in Fig. 7 are  $IN_0, IN_1, IN_2, OUT_0, OUT_1$ , and  $OUT_2$ . Its inputs are  $REQ_0, REQ_1$ , and  $REQ_2$ , and its outputs are  $GR_0, GR_1$ , and  $GR_2$ . Here

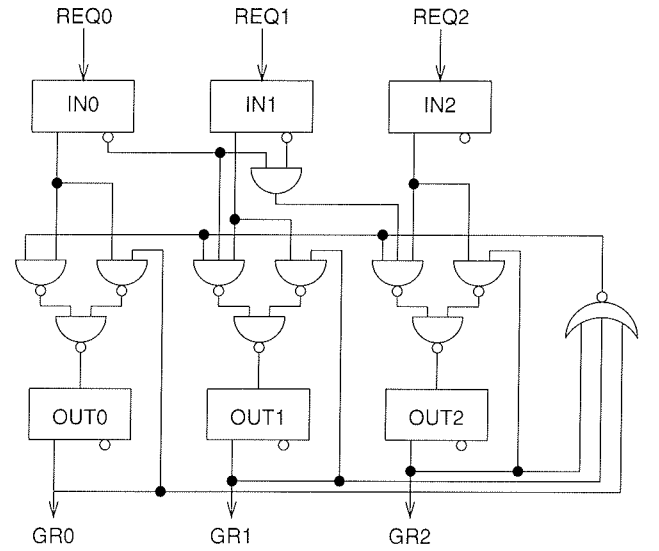


Figure 7: Digital circuit schematics of a 3-input arbiter

are its state transition functions:

$$\begin{aligned}
 IN'_0 &= REQ_0 \\
 IN'_1 &= REQ_1 \\
 IN'_2 &= REQ_2 \\
 OUT'_0 &= \overline{OUT_0} \cdot \overline{OUT_1} \cdot \overline{OUT_2} \cdot IN_0 + IN_0 \cdot OUT_0 \\
 OUT'_1 &= \overline{OUT_0} \cdot \overline{OUT_1} \cdot \overline{OUT_2} \cdot \overline{IN_0} \cdot IN_1 + \\
 &\quad IN_1 \cdot OUT_1 \\
 OUT'_2 &= \overline{OUT_0} \cdot \overline{OUT_1} \cdot \overline{OUT_2} \cdot \overline{IN_0} \cdot \overline{IN_1} \cdot IN_2 + \\
 &\quad IN_2 \cdot OUT_2
 \end{aligned} \tag{6}$$

Its output functions are as follows:

$$\begin{aligned}
 GR_0 &= OUT_0 \\
 GR_1 &= OUT_1 \\
 GR_2 &= OUT_2
 \end{aligned} \tag{7}$$

In general, a circuit gives a FSM with a set of state variables  $\mathcal{Y} = \{y_{n-1}, y_{n-2}, \dots, y_0\}$ , inputs  $\mathcal{X} = \{x_{m-1}, x_{m-2}, \dots, x_0\}$ , and outputs  $\mathcal{Z} = \{z_{l-1}, z_{l-2}, \dots, z_0\}$ . The FSM has the transition functions (1) and the output functions (3).

A stuck-at fault is caused by a short circuit between a line connecting two elements and the logical 1 or 0. If the short circuit is with the logical 0, there is an SA0 fault ("stuck-at 0"). If the short circuit is with the logical 1, there is an SA1 fault ("stuck-at 1").

The question is how a stuck-at fault in a circuit is manifested in the FSM whose realization it is. In the circuit, all the flip-flops as well as external inputs and outputs remain the same. It follows that the FSM whose realization is the circuit *with* a stuck-at fault will have the same state variables, inputs, and outputs as the FSM whose realization is the good circuit without stuck-at faults. However, some transition and output function can change due to changed connections.

Each line connects exactly one output of a logic gate or flip-flop or an input of the circuit with, in general,

many inputs of logic gates, flip-flops, or outputs of the circuit. Since the source of the line is uniquely determined, a stuck-at fault can canonically be denoted by  $G = 0$ , respectively  $G = 1$ , which means that an output of a logic gate or a flip-flop, or a circuit input  $G$  has stuck at 0, respectively to 1. With the transition functions (1) and the output functions (3), we obtain for the FSM corresponding to the circuit with a stuck-at fault  $G = b$  ( $G$  can be equal to any of the possible fault locations and  $b \in \{0, 1\}$ ) the following transition functions:

$$y'_i = f_i|_{G=b}(y_0, y_1, \dots, y_{n-1}, x_0, x_1, \dots, x_{m-1}) \quad (8)$$

for  $i = 0, 1, \dots, n-1$  and output functions:

$$z_i = g_i|_{G=b}(y_0, y_1, \dots, y_{n-1}, x_0, x_1, \dots, x_{m-1}) \quad (9)$$

for  $i = 0, 1, \dots, l-1$ . It can happen that not all transition and output functions are changed as a consequence of a stuck-at fault. It is perfectly possible that  $f_i = f_i|_{G=b}$ , respectively  $g_j = g_j|_{G=b}$ , for some values of  $i$  and  $j$ . This is in fact quite usual with real circuits. If a stuck-at fault occurs in a redundant part of the circuit, it can even happen that all the transition and output functions remain the same.

Suppose that in the circuit in Fig. 7 the output of the logic gate NOR would stick at logical 1. The circuit would become a realization of a FSM similar to the original one. Only the state transition functions for  $OUT_0$ ,  $OUT_1$ , and  $OUT_2$  would change:

$$\begin{aligned} OUT'_0 &= IN_0 \\ OUT'_1 &= \overline{IN_0} \cdot IN_1 + IN_1 \cdot OUT_1 \\ OUT'_2 &= \overline{IN_0} \cdot \overline{IN_1} \cdot IN_2 + IN_2 \cdot OUT_2 \end{aligned} \quad (10)$$

The rest of transition and output functions would remain the same.

## 4.2 Extension of FSM

Any circuit property expressed in the form of a CTL formula is valid in some states of the FSM corresponding to the circuit. The states are determined by state variable values. Input and output values do not determine the current state of the FSM.

Since stuck-at faults generally also affect the output values, they must be covered by properties as well. As outputs cannot be included in CTL formulas, a possible solution is to extend the FSM with additional state variables. For every output  $z_i$ , a state variable  $y_{n+i}$  is added. The variable's transition function is defined as  $y'_{n+i} = f_{n+i} = z_i$  for  $i = 0, 1, \dots, l-1$ . The new outputs of the extended FSM are defined as  $z_{z,i} = g_{z,i} = y_{n+i}$  for  $i = 0, 1, \dots, l-1$ .

Stuck-at faults do not affect the circuit input values. However, the input values affect the circuit (FSM) state transitions in the future. For this reason, the inputs must also be covered by properties. The solution is similar to the one for outputs, only that here, additional state variables are added at the inputs of the

original circuit. For every input  $x_i$ , a state variable  $y_{n+l+i}$  is added. Transition functions for the added state variables are defined as  $y'_{n+l+i} = f_{n+l+i} = x_{x,i}$  for  $i = 0, 1, \dots, m-1$ , where  $x_{x,i}$  is a new input of the extended circuit and  $m$  the number of inputs. In all formulas  $f_i$  and  $g_j$  where the original circuit inputs  $x_k$  occur, we take into account that  $x_k = y_{n+l+k}$  for  $k = 0, 1, \dots, m-1$ .

Graphically, a FSM extension can easily be shown in the corresponding circuit schematics. If we extend the arbiter from Fig. 7 with additional state variables (i.e. flip-flops in the circuit) in the way just described, we obtain the circuit shown in Fig. 8. The added ele-

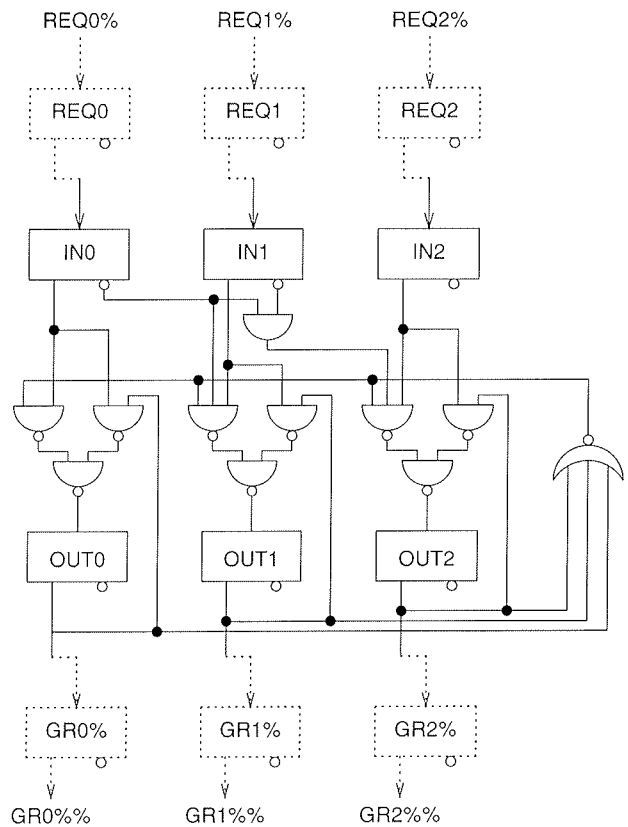


Figure 8: Extended arbiter with additional state variables

ments are drawn with dotted lines.

The newly added state variables  $y_i$  (for  $i = n, n+1, \dots, n+l+m-1$ ) can be used in CTL formulas like any other state variable. The new variables can be used to express properties about the original FSM inputs and outputs. It should be noticed that stuck-at faults can only occur on the lines in the original circuit and, consequently, in the corresponding places in the original and extended FSM.

## 4.3 Properties

In every FSM, many properties are valid. One of the most basic types of properties, which are valid in every

FSM, are properties of the form:

$$AG(f_i \cdot AX y_i + \overline{f_i} \cdot AX \overline{y_i}) \quad (11)$$

The formula says that on all possible computation paths and all the time it holds that if in some state the state transition function  $f_i$  of the state variable  $y_i$  has the value 1, then in every successor of the state, the value of  $y_i$  will also be 1; respectively, if the value of the transition function  $f_i$  of the state variable  $y_i$  in the state is 0, then the variable  $y_i$  will also be 0 in every successor state. This holds for all the state variables, including the added ones.

The properties cannot help us to find stuck-at faults. Since they are valid, counterexamples do not exist, of course. Also, for this type of formulas, even a witness cannot be constructed according to the explanation in Section 3.2.

However, following the above pattern we can write a CTL formula which will be valid in the FSM that corresponds to the circuit with a stuck-at fault. If an output of a logic gate or a flip-flop, or a circuit input  $G$  sticks at a logical value  $b$ , then the property has the form:

$$AG(f_i|_{G=b} \cdot AX y_i + \overline{f_i|_{G=b}} \cdot AX \overline{y_i}) \quad (12)$$

By rule, such a formula cannot be valid in a circuit (FSM) without a stuck-at fault. It follows that a counterexample can be constructed for such a CTL formula. The formula namely does not contain CTL operators with existential path quantifier. It contains a disjunction, but in such a way that a counterexample can be found anyway.

When a stuck-at fault  $G = b$  is inserted into a FSM, the properties (12) are valid for all the state variables, i.e. for  $i = 0, 1, \dots, n+l+m-1$ . The single properties can be applied to form the common one:

$$\bigwedge_{i=0}^{n+l+m-1} AG(f_i|_{G=b} \cdot AX y_i + \overline{f_i|_{G=b}} \cdot AX \overline{y_i}) \quad (13)$$

Due to distributivity of operator  $AG$  and conjunction, formula (13) can be rewritten as

$$AG \bigwedge_{i=0}^{n+l+m-1} (f_i|_{G=b} \cdot AX y_i + \overline{f_i|_{G=b}} \cdot AX \overline{y_i}) \quad (14)$$

Some of the conjunctives in the CTL formula (14) can be valid always and everywhere. Finding all such factors is generally a complex problem, but certainly, all the factors for which the stuck-at fault  $G = b$  does not affect the corresponding transition function and consequently,  $f_i|_{G=b} = f_i$ , are among them. If a CTL formula is valid always and everywhere, then it is equivalent to formula 1 ("true"). It follows that such factors can be left out in the conjunction. Only those have to be included, for which  $f_i|_{G=b} \neq f_i$ . The factors corresponding to the variables added at the original inputs  $y_i$  ( $i = n+l, n+l+1, \dots, n+l+m-1$ ) are also true. This is because all the lines which affect the variables

are added only in the abstract FSM, whereas they do not exist in the real circuit, in which stuck-at faults can occur. If we take into account both facts, we get the following CTL formula (property):

$$AG \bigwedge_{\substack{0 \leq i < n+l \\ f_i|_{G=b} \neq f_i}} (f_i|_{G=b} \cdot AX y_i + \overline{f_i|_{G=b}} \cdot AX \overline{y_i}) \quad (15)$$

The generation of properties continues by generating properties (15) for all possible stuck-at faults  $G = b$  in the circuit, for  $G$  equal to all possible flip-flop and logic gate outputs<sup>2</sup> and for  $b$  equal to logical values 0 in 1.

Generally, properties (15) are not valid in a FSM without stuck-at faults. If some of them is valid anyway, it means that the stuck-at fault  $G = b$  considered has no effect on the circuit behaviour and consequently, the fault need not be refuted during testing. Only the invalid properties (15) are, therefore, interesting, and we continue the work with them alone.

#### 4.4 Searching of Counterexamples

When searching counterexamples for invalid CTL formulas of the type (15), we in fact search for a witness of the following CTL formula because of the chosen way of searching and the use of DeMorgan's law:

$$E \left[ 1 U \bigvee_{\substack{0 \leq i < n+l \\ f_i|_{G=b} \neq f_i}} \overline{f_i|_{G=b} \cdot EX \overline{y_i} + \overline{f_i|_{G=b}} \cdot EX y_i} \right] \quad (16)$$

It means that we search for a path leading from the initial state to a state in a state set with the following characteristic. For each state, it is not the case that a transition function  $f_i$  is equal to 1 (respectively, 0) in the state and that at the same time, there does not exist a successor state where the corresponding state variable  $y_i$  is equal to 0 (respectively, 1).

When a witness for the operator  $EU$  in formula (16) is found, we still must find a witness from that state on for the formula

$$\bigvee_{\substack{0 \leq i < n+l \\ f_i|_{G=b} \neq f_i}} \overline{f_i|_{G=b} \cdot EX \overline{y_i} + \overline{f_i|_{G=b}} \cdot EX y_i}$$

We can choose one factor in the big disjunction and find a witness for it. Of course, we must choose a factor which is valid in the state reached when searching for a witness of the operator  $EU$ . It follows that a witness of

$$\overline{f_i|_{G=b} \cdot EX \overline{y_i} + \overline{f_i|_{G=b}} \cdot EX y_i}$$

is searched for, where  $i$  is such that the formula is valid there. This can be done by finding a counterexample for the CTL formula

$$f_i|_{G=b} \cdot EX \overline{y_i} + \overline{f_i|_{G=b}} \cdot EX y_i$$

<sup>2</sup>The extended circuit inputs need not be considered since the inputs are added artificially and stuck-at faults cannot occur on them.

There are two possibilities. The transition function  $f_i|_{G=b}$  can either be valid or invalid in the current state. We, therefore, continue along one of the following paths:

1. If  $f_i|_{G=b}$  is valid, we search for a counterexample of the formula  $\overline{EX}y_i$ , which consequently means searching for a witness of the CTL formula  $EX\overline{y_i}$ . A successor of the current state in which the value of state variable  $y_i$  is 0 has to be found.
2. If  $f_i|_{G=b}$  is invalid, we search for a counterexample of the formula  $\overline{EX}y_i$ , which consequently means searching for a witness of the CTL formula  $EXy_i$ . A successor of the current state in which the value of state variable  $y_i$  is 1 has to be found.

A counterexample found is given in the form of a sequence of state variable values of the extended FSM in the states starting with the initial one and continuing along the counterexample path. Every state variable which is equal to the one in the original FSM has the same meaning in the latter and in the extended FSM. The state variables added at the inputs of the original FSM indicate the values we have to assign to the circuit inputs in order for the circuit execution to follow the counterexample path. The state variables added at the outputs of the original FSM indicate the values the real outputs of the good circuit will have in the states on the path.

It should be noted that the circuit outputs in the form of the added variables of the extended FSM are delayed for one step. The current circuit output values occur in the added variables in the next step. This is because a state variable gets a value determined by its state function in the next state, whereas the outputs get their value in a moment.

The test pattern which belongs to the counterexample found is a sequence of values of the state variables added at the inputs. The values in the last state on the counterexample path can be left out because the circuit input values in the last step do not matter. The circuit is tested as follows. The input values from the test pattern are set on the circuit inputs one after another. When its end is reached, we check if values of the state variables (flip-flops) and circuit outputs are equal to those in the last state of the counterexample.

If all the values are equal, the absence of the stuck-at fault considered is confirmed. Afterwards, we reset the circuit and repeat the whole procedure with the test pattern for the next stuck-at fault, and so on until the last possible stuck-at fault has been considered.

## 5 Experimental results

Experiments were done on a server with AMD Athlon/800 processor, 512 MB of physical memory, and 1.4 GB of virtual memory under Linux operating system. We have used our own BDD package ([7]),

which is an efficient *ite*-based implementation of reduced ordered binary decision diagrams with complemented edges.

We generated test patterns for some ISCAS benchmark circuits. Results are shown in Table 1. From left

Table 1: Test pattern generation for ISCAS benchmark circuits

circuit	# stuck-at faults	joint length	max length	# BDD nodes	CPU time [s]
s27	38	58	3	604	0.00
s208.1	248	5474	257	50606	8.17
s298	296	1158	11	41582	1.92
s344	412	932	8	48208	7.67
s349	414	932	8	48087	6.69
s382	388	5387	102	99775	31.01
s386	372	1034	10	45819	2.29
s444	434	5990	102	182597	39.05
s526	458	7749	102	286215	79.22
s526n	460	7771	102	286183	80.24
s641	962	1787	5	1698296	362.29
s713	986	1807	5	1698421	363.47
s820	700	3090	12	257285	185.28
s832	696	3056	12	256203	179.73
s953	972	4667	11	224760	336.28
s1196	1178	2359	4	139827	69.34
s1238	1136	2254	4	134801	60.12
s1488	1410	6847	22	107989	154.61
s1494	1398	6743	22	108057	170.19

to right the columns in Table 1 refer to the circuit name, number of potential stuck-at faults, joint length of all test patterns, maximal length of test patterns, and the maximal number of BDD nodes whenever generated with the CPU time in seconds.

The number of potential stuck-at faults is approximately proportional to the number of flip-flops and gates in the circuits. The maximal length of test patterns depends on the number of steps necessary to set such values in flip-flops that stuck-at fault will demonstrate at computation of the next state. Joint length of all test patterns is a plain sum of lengths of single test patterns. It is very difficult to say anything general about the joint length.

We examined time and space complexity of the test pattern generation on a series of parametric up/down counters. Results we obtained are shown in Table 2, where the number of potential stuck-at faults, joint length of all test patterns, and maximal number of BDD nodes whenever generated with the CPU time in seconds for every counter size  $n$  are shown. Both time and space complexities are less than exponential.

We did not compare our results obtained with test pattern generation with results of other authors since we did not manage to find any contribution where authors would have generated test patterns for searching stuck-at faults with symbolic model checking. In the most similar example [4] they used symbolic state space traversal but not symbolic model checking.



Table 2: Test pattern generation for parametric counter

$n$	# stuck-at faults	joint length	# BDD nodes	CPU time [s]
10	102	156	5232	0.06
20	202	316	39312	0.42
30	302	476	78499	1.58
40	402	636	129859	4.31
50	502	796	213819	10.32
60	602	956	338379	60.19
70	702	1116	511539	250.02
80	802	1276	741299	488.44
90	902	1436	1045840	823.04
100	1002	1596	1424382	1287.52
110	1102	1756	1884720	1890.39
120	1202	1916	2434862	2709.84
130	1302	2076	3082801	3781.46
140	1402	2236	3836544	5160.15
150	1502	2396	4704084	6926.01
160	1602	2556	5693423	9169.76
170	1702	2716	6812565	11867.43
180	1802	2876	8069514	15189.49

## 6 Conclusions

We developed methods for fully automatic test pattern generation (ATPG) for discovering single stuck-at faults in synchronous sequential digital circuits. Test patterns are based on counterexamples obtained by symbolic model checking. For every possible stuck-at fault, a suitable property is generated. When the property is invalid a counterexample is found.

The described algorithms are realized in the form of a computer program for ATPG for discovering of single stuck-at faults or proving their absence. The program is based on a home-made package for manipulating FSMs which is also based on a fully home-made very efficient package for manipulating Boolean functions represented by BDDs [7].

We illustrated the usage of presented algorithms by generating test patterns for some ISCAS benchmark circuits and a parametric up/down counter. Results from the latter one also indicate time and space complexity of the algorithms.

There are a lot of possibilities for future work. The most interesting would be development of methods where stuck-at faults would manifest only at circuit outputs. Since also other types of faults can occur in a circuit, it would be interesting to discover also them. It might be useful to find out also which stuck-at fault is present in the circuit if presence of one is detected.

## References

- [1] Zmago Brezočnik, Aleš Časar, and Tatjana Kapus. Efficient Symbolic Traversal Algorithms using Partitioned Transition Relations. In Zmago Brezočnik and Tatjana Kapus, editors, *Proceedings of the COST 247 International Workshop on Applied Formal Methods in System Design*, pages 146–155, Maribor, Slovenia, June 1996.
- [2] Randal E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.
- [3] Jerry R. Burch, Edmund M. Clarke, David E. Long, Kenneth L. McMillan, and David L. Dill. Symbolic Model Checking for Sequential Circuit Verification. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, 13(4):401–424, April 1994.
- [4] Gianpiero Cabodi, Paolo Camurati, and Stefano Quer. Symbolic forward/backward traversals of large finite state machines. *Journal of Systems Architecture*, 46:1137–1158, 2000.
- [5] Aleš Časar. Verification of finite state machines with symbolic model checking. Master's thesis, University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia, June 1998. In Slovene.
- [6] Aleš Časar, Zmago Brezočnik, and Tatjana Kapus. Formal Verification of Digital Circuits using Symbolic Model Checking. *Informacije MIDEM*, 30(3(95)):153–160, September 2000.
- [7] Aleš Časar, Robert Meolic, Zmago Brezočnik, and Bogomir Horvat. Representation of Boolean Functions with ROBDDs. *Electrotechnical Review*, 59(5):299–307, December 1992. In Slovene.
- [8] E. Clarke, O. Grumberg, K. McMillan, and X. Zhao. Efficient generation of counterexamples and witnesses in symbolic model checking. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, October 1994.
- [9] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986.
- [10] Bogdan Dugonik. Metode za iskanje optimalnih vektorjev za diagnostično testiranje digitalnih vezij s pomočjo modela napak. Master's thesis, University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia, 1995. In Slovene.

mag. Aleš Časar, univ. dipl. inž. rač. in inf.  
izr. prof. dr. Zmago Brezočnik, univ. dipl. inž. el.  
izr. prof. dr. Tatjana Kapus, univ. dipl. inž. el.

Univerza v Mariboru  
Fakulteta za elektrotehniko, računalništvo in  
informatiko  
Smetanova 17

2000 Maribor

tel.: +386-2-22-07-211

fax: +386-2-25-11-178

email: {casar, brezocnik, kapus}@uni-mb.si

# PLANARIZATION METHODS IN IC FABRICATION TECHNOLOGIES

<sup>1</sup> Radko Osredkar, <sup>2</sup> Boštjan Gspan

<sup>1</sup> Faculty of Computer Sciences and Faculty of Electrical Eng.,  
University of Ljubljana, Slovenia

<sup>2</sup> Repro MS, Ljubljana, Slovenia

**Key words:** semiconductors, microelectronics, integrated circuits, topography planarization, SOG, polyimide, PECVD, sacrificial layer

**Abstract:** Planarization methods that are widely used in IC fabrication technologies employing 2 interconnect metalization layers are reviewed. Methods allowing only partial planarization are discussed, including fluidic methods (spin-on glass, polyimide), CVD dielectric layer deposition and etch-back, and some of their combinations. There is still a large interest in these relatively simple planarization methods, especially in regard to their refinement and fine tuning to the application at hand, despite the fact, that total global planarization, such as afforded by the CMP method, is widely considered to be the planarization method of the future. A representative, but by no means exhaustive, list of references is presented.

## Pregled planarizacijskih metod v mikrelektronskih tehnologijah

**Ključne besede:** polprevodniki, mikroelektronika, integrirana vezja, planarizacija topografije, SOG, centrifugalno nanašanje stekla, poliimid, PECVD, nanos iz plazemske faze, žrtvovana plast

**Izvilleček:** Predstavljene so metode planarizacije, ki se široko uporabljajo v mikrelektronskih tehnologijah za proizvodnjo integriranih vezij z dvema plastema kovinskih (aluminijevih) povezav. Te metode omogočajo le delno planarizacijo. Obravnavane so fluidična planarizacija s centrifugalnim nanosom stekla in poliimida, planarizacija s CVD dielektrično plastjo ter uporaba žrtvovane plasti v kombinaciji fluidnih in CVD plasti. Kljub temu, da je kemijsko-mehansko poliranje površine rezine planarizacijska metoda, ki bo verjetno sčasoma prevladala, vlada danes še vedno veliko zanimanje za preprostejše metode delne planarizacije, posebno v povezavi z njihovimi izboljšavami in prilagoditvijo konkretnim potrebam določenega proizvodnega procesa. Navedena je reprezentativna, nikakor pa ne popolna, literatura na opisano tematiko.

### Introduction

It is widely recognized that in IC fabrication technologies some sort of planarization, i.e. reduction of the vertical distances between topography features and reduction of the side wall slopes, becomes unavoidable as circuit features are scaled to submicron dimensions. Considerations of wafer topography become most critical during the final steps of fabrication, when several metallization and dielectric layers are deposited. The stacking of layers on top of one another in the multilevel interconnect technologies can result in poor step coverage of the metal lines as they cross over steps, and metal stringers that remain at the foot (or sides) of a sharp step after anisotropic etching [1]. If these two problems can be overcome in an IC fabrication process in which no planarization is used, eventually the limited depth of field of optical lithography tools would require some planarization to be used. In technical literature the term planarization is often used quite loosely and in connection with quite varied processing goals. In this review only planarization of dielectric layers as it applies to the multilevel interconnect technology is considered, i.e. the planarization of the dielectric layers that are formed between patterned metal layers.

There are several degrees of planarization that are achievable with different techniques, and, indeed, only different degrees of planarization are required in different cases.

1. The first degree of planarization involves only a reduction of the step slopes in the topography of the dielectric film, without significant step heights reduction.
2. Partial planarization (or semi-planarization) involves, in addition to the step slope reduction, also a reduction of the step heights.
3. In complete local planarization complete reduction of the step heights is attempted in areas of dense topography (i.e. where the spaces in the underlying metal layer topography are relatively close together), but isolated features on the wafer still exhibit some step height.
4. In the complete global planarization the surface of the dielectric layer is completely planarized over arbitrary topography.

A quantitative measure of the step-height reduction, referred to as the planarization factor  $\beta$ , is given by [1]

$$\beta = 1 - (s_1/s_0)$$

where  $s_1$  and  $s_0$  are the final and the initial step heights, respectively. In complete planarization  $\beta = 1$  and 0 if no planarization exists. In cases where the planarization process involves only a reduction of the step slopes  $\beta = 0$ , however, the effects of the planarization process can be quantified by determining the so called transition angle, i.e. the angle between the wafer plane and the tangent to the dielectric layer at the step half-heights. The transition angles can be measured directly from the electron micrographs of the planarized structures. (Fig.1)

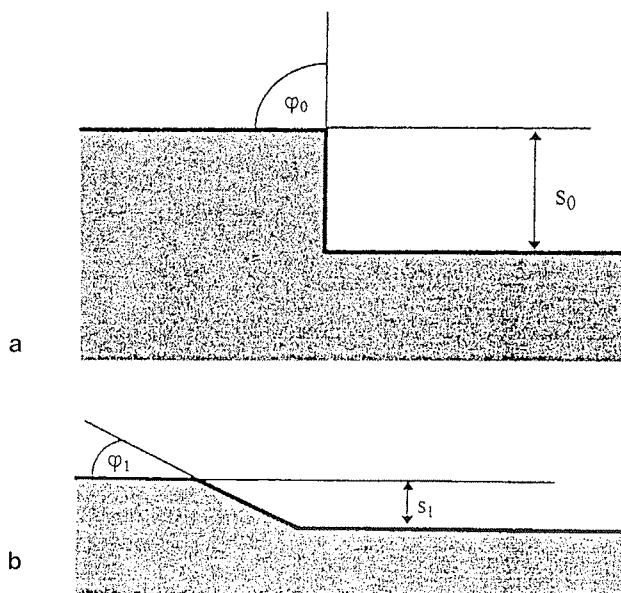


Fig. 1 (a) Definition of the planarization factor  $\beta = 1 - (s_1/s_0)$ ;  $s_1$  and  $s_0$  are the final and the initial step heights, respectively. (b) In cases where the planarization process involves a reduction of the step slope, its effects are quantified by the final transition angle  $\varphi_1$

There is a penalty involved in applying planarization techniques in IC fabrication. It is related to the fact, that after planarization the dielectric layer is by necessity not of uniform thickness all over the device area which means that the openings (vias) that allow forming electrical contacts between different metallization layers are of different depths. If the via sidewalls are sloped, the dimensions of the shallow vias will continue to increase during the time needed to completely etch the deep vias, possibly exceeding the width of the underlying metal pattern. In such cases a complete and global planarization may be in fact undesirable and a partial planarization of the device topography preferable /2/. There are at this time some technologies that require global planarization, e.g. LCD technologies where the globally planarized wafer surface is the lower electrode of a LC display /3/, in ferroelectric memory devices /4/, and devices incorporating optical components /5/. Global planarization techniques, especially chemical mechanical polishing (CMP), are at this time at the forefront of the process development efforts, however, they are not yet standard in the IC manufacturing, and will not be discussed further. On the other hand, there is still a

large interest in partial planarization techniques, especially in regard to their refinement and fine tuning to the application at hand /6,7/, and a review of these is presented in this contribution. One should bear in mind, though, that the more levels of metal are required the less effective the partial planarization techniques become and the greater the need for processing that allows vertically sided contact holes. In this sense the global and the partial planarization techniques are converging.

## Fluidic Methods – Spin-on Glass

Spin-on dielectric (SOD) planarization techniques utilize low viscosity of certain materials, e.g. photoresists, polyimides and spin-on glasses (SOG), which can fill the trenches in wafer topography /8/. These methods are simple to apply and usually require low processing temperatures (below 400 °C). However, compatibility of the fluidic materials with the standard dielectric materials can be a serious concern, and, as a rule, only limited planarization can be achieved by such methods. /9/

SOG is a frequently used planarization material /8/, and there are several different materials used for these purposes, silicate SOG and siloxane SOG being the most common. Planarization techniques utilizing these materials combine the planarizing effect of the spun-on films with the oxide like material characteristics of the SOG materials, resulting in simple and straightforward processing /10/. These materials are fairly compatible with other materials and processes in the IC fabrication technologies, and are easily integrated into the fabrication process flows. A SOG planarization film is always formed on a substrate towards which it exhibits good adhesion. Therefore, when such a film is dried and cured, shrinkage can occur only in a direction perpendicular to the substrate plain, while in the substrate plain the film is constrained. This results in build-up of the tensile stress parallel to the wafer surface. This stress is usually below  $10^9$  dyne.cm<sup>-2</sup>, however, it has been shown /11/ that due to built-in stress SOG films have a propensity for cracking.

In IC fabrication the maximum temperature at which a SOG film can be cured is usually limited to 420 °C, or lower, because of the presence of the underlying aluminum inter-

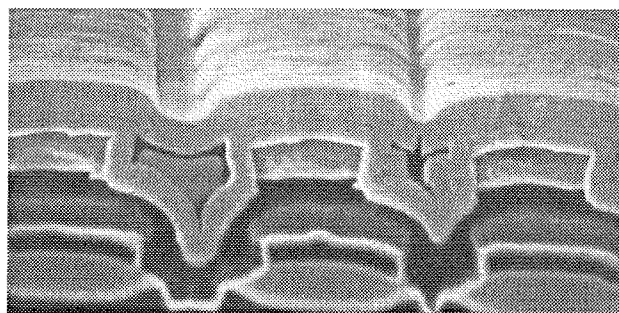


Fig. 2 Top view of a triple stack structure (for definition see text), planarized with a SOG partial etch back technique. Crack in the remaining SOG filler can clearly be noted, ( $M = 10^4$ ).

connection layer. After such a low temperature cure the SOG film is not completely densified and contains significant amount of silanols (in siloxane materials) and adsorbed water. If the film can be densified at a higher temperature, typically 800 – 900 °C, a silanol and moisture free film is obtained, as demonstrated by IR spectroscopy /12/. However, some porosity still remains, as demonstrated by re-hydration experiments in which freshly densified films are exposed to an atmosphere saturated with water and from which the films can reversibly adsorb H<sub>2</sub>O /9/. These observations indicate that moisture (and possibly silanol) content in SOG films can be, due to reactions of the moisture from the film with the aluminum, potentially a serious source of quality degradation problems /1, 13/. (Fig. 2)

Rapid evaporation of the solvent from the SOG material during the deposition process challenges detailed analysis and prediction of the degree of planarization possible with such a process. The rheological (fluidic) deposition models, which are used in analysis of the spin-on processes and are based on the Navier–Stokes equations /14, 15/, suggest that covering a patterned wafer with a fluidic film (SOG, photoresist or polyimide) results in a completely flat top surface of the film, which remains flat until a considerable amount of solvent is removed from the material. After the removal of the solvent only non-volatile components of the film material remain on the wafer surface. If the proportion of the non-volatile components in a SOG material is  $k$ , the planarization factor achieved during the evaporation of the solvents is simply  $\beta = k$ , and thus the planarization only partial. In case of Accuglass 204 siloxane SOG material, which contains 10 % of nonvolatile components (as specified by the producer), a maximal planarization factor of  $\beta = 0.1$  could be expected. However, application of such a film to the patterned wafer clearly shows that this is not the case. Due to gross transport of the planarizing material, driven by the surface tension during evaporation of the solvents, in dense topography planarization factors as high as 0.85 (depending on the details of the wafer topography) can be achieved, and less than 0.1 on isolated topography features. This indicates that the transport of the planarizing material during the evaporation of the solvents plays a crucial role in the SOG planarization process. Also, the quality of the surface underlying the planarization film appreciably effects the planarization process. These effects can not be predicted by the rheological models /9/. Typically, planarization by SOG films deposited on patterned wafers (patterned aluminum on field oxide) simultaneously exhibits its surface and topography effects. All of the above precludes total global planarization with such a process.

If two SOG films are deposited consecutively, the second film being deposited after considerable densification of the first one, an increase of the planarization factor  $\beta$  by approx. 10 % is achievable. The resulting planarization factor of a two-step planarization process can be quite reliably predicted for a predetermined site on the wafer. (Table 1, Ref. 9.).

Quite clearly a multiple-step planarization by SOG a material is a process of diminishing returns and total planarization, even locally at a predetermined site, may not be a realistic goal of such a multi-step process.

*Table 1. Planarization factors after different stages of SOG (Accuglass 204) planarization process*

Planarization stage	planarization factor
deposition (including evaporation of solvents at 100 °C): isolated lines	0.09
deposition (including evaporation of solvents at 100°C): dense topography	0.9
densification	0.9
compound (evaporation + densification): dense topography	0.81
double planarization: dense topography	0.86
triple planarization: dense topography	0.90

## Fluidic Methods – Polyimide

Polyimide films are also frequently used as an interlevel dielectrics. Their use is justified primarily by the ease of their application, which parallels the application of photoresists, and also by the good chemical and physical properties of the cured films and their general compatibility with most materials encountered in IC fabrication. As the polyimide films are formed from a solution that typically contains 15 to 30 % of solids, they have attractive planarizing properties, such as the ability of filling of narrow trenches without forming voids during curing, and relatively high (compared to SOG) planarization factors. In contrast to SOG planarization processing, application of several polyimide films consecutively presents no serious problems, thus enhancing the (local) planarization factors achievable by this material beyond 0.95. Some impressive multi metal processes using polyimide films as interlevel dielectric have been reported /16/, however, the material has not gained widespread acceptance. Among the several reasons for this is the sensitivity of its cross-linking curing process to the precise curing temperatures and schedules /17/, its hygroscopicity, questionable adhesion of the polyimide to metal under conditions of stress, and relative complexity of the integration of the polyimide etching process into a fabrication process.

## Reflow of Doped Glass

When processing temperatures above 400 °C are not a consideration (e.g. if the underlying patterned layer is polysilicone), deposition and thermal reflow of doped glasses offer an attractive step reduction and/or partial planarization possibility. It is well known that the composition of the doped glass (e.g. boron and phosphorus doped glass (BPSG)) strongly influences the reflow temperature at which the surface tension of the film drives the redistribution of the film material /1/. Optimizing glass doping for low temperature reflow makes the planarizing film susceptible to moisture, resulting in poor reliability of fabricated ICs /18/. In designing a successful planarization process involving BPSG both of these tendencies have to be considered and balanced. As an example, the transition angles after reflow planarization (30 min at 950 °C) of BPSG films deposited by a PECVD method and densified, in the range of 2.4 to 3.1 w % of B and 4.4 to 5.4 w % of P are shown in Table 2 /19/. At this compositions the transition angles

are not minimal, however, structural stability of such films is greatest. The so called single stack structure is an array of 600 nm thick and 1,2  $\mu\text{m}$  wide parallel aluminum lines formed on a flat, oxidized wafer; in the double stack structure the metal lines are formed on top of oxide lines of the same dimensions, resulting in a 1,2  $\mu\text{m}$  step in topography. Both types of structures were prepared at two different separations of the lines, 1  $\mu\text{m}$  and 2,5  $\mu\text{m}$ . The transition angle between the topography levels before the planarization has in all cases been close to 90 deg. (Fig. 3)



Fig. 3 A single stack structure (1  $\mu\text{m}$  trench width) planarized by a BPSG reflow process. The doping of the planarizing material is 3.1 w % P, 4.5 w % B, 30 min reflow at 950  $^{\circ}\text{C}$  planarization. The transition angle after reflow is 22.0 deg, and  $\beta = 0.6$ .

Table 2. Transition angles in single and double stack structures, after BPSG densification and reflow, at different film compositions.

w. % B	W. % P	transition angle (deg)			
		line separation 1 $\mu\text{m}$		line separation 1 $\mu\text{m}$	
		double s.	single s.	double s.	single s.
3.1	4.5	22.7	22.0	19.7	19.0
2.4	5.4	24.6	23.2	23.1	16.4
2.5	3.2	30.8	30.9	33.0	33.5
1.6	4.6	40.0	35.9	36.9	34.3
2.8	4.4	29.8	21.3	30.5	22.2

Thermal reflow of a BPSG film can in favorable circumstances lead to nearly complete local planarization. Recent advances in the glass-flow processing allow for simultaneous deposition and reflow and reduced processing temperature /20/.

## Etch-back and Sacrificial Layers

One of the simplest methods available for smoothing steps in wafer topography, and one that is the easiest to integrate into a fabrication technology, is the deposition of a CVD glass layer that is significantly thicker than the step it must cover and subsequent etch-back to the desired film thickness /1/. The method is based on the isotropic and thus nonconformal nature of the CVD deposition process, which results in a dielectric film with topography features

less extreme than the features of the film it covers. But the nature of the deposition process is also the cause of one of the difficulties in applying the method in topographies with high aspect ratios, i.e. the formation of voids between metal lines if the thickness of the metal layer exceeds approx. one half of the minimum spacing between the metal lines. A process combining plasma enhanced and low temperature CVD TEOS allows high aspect ratio (as high as 0.85) topographies to be filled without void formation /21/. Such a process can provide partial planarization with planarization factors exceeding 0.5, but not total planarization, either local or global.

An extension of the above method is the sacrificial layer technique. It is widely used in two-metal processes and allows a high degree of planarization between steps that are 2 to 10  $\mu\text{m}$  apart, but works less well for planarizing isolated features /7/. The process involves coating a CVD dielectric layer with a film that will later be etched off (sacrificed). Usually sacrificial layers are formed from low viscosity fluids that, after appropriate heat treatment, produce solid films with planarized, often practically flat, surfaces. Photoresists, polyimides and SOG films are used for these purposes. Therefore after the formation of the sacrificial film the planarization of the CVD film topography can not exceed the planarization that can be achieved by the fluidic methods. However, during the rapid etch back of the sacrificial layer the topmost parts of the CVD layer become exposed first and with suitable etch chemistry, that allows the CVD and sacrificial layers to be etched at equal rates, further planarization is achieved. As the sacrificial layer planarization process is well described in literature and models for predicting the degree of planarization available /22, 23/ it will not be further discussed.

When a SOG material is used as the sacrificial layer, an extension of the process is possible in which the SOG material is not etched back completely but is allowed to remain filling the gaps in the CVD dielectric layer /7/. Fig. 4 dem-

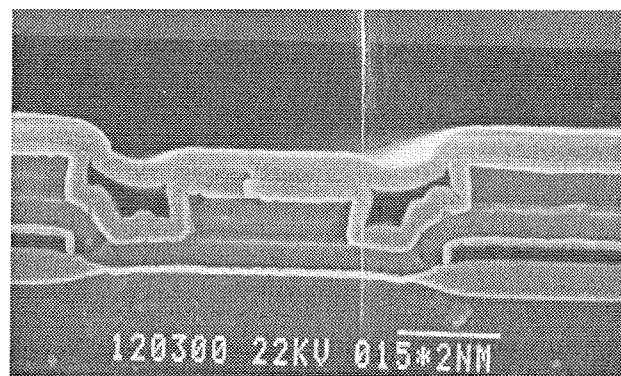
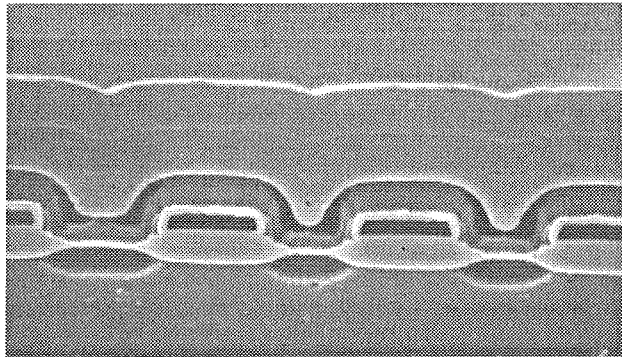


Fig. 4. Planarization of the sunken via (via itself not shown) between two double stack structures, with a SOG partial etchback process. The dark structures between the metal lines are SOG fillers that remain after the SOG etchback and Metal 2 is deposited directly on the structure.

onstrates the planarization of a sunk contact by such a method in a 1.2  $\mu\text{m}$  process, using Accuglass 204 SOG material. The SOG etchback process is sometimes extended by depositing a thin CVD dielectric film on top of the SOG fillers, thus forming a dielectric sandwich structure, (Fig. 5).



**Fig 5** A SOG – CVD sandwich structure. The double stack (patterned polysilicone on FOXT) structure has been planarized SOG partial etchback process and then covered by a double CVD layer. The second of these is a sacrificial layer to be planarized by etchback.

A major and unsolved difficulty of this process remains the partial incompatibility of the low-temperature cured SOG material with the materials it contacts and possible reliability problems this might cause. The use of polyimide instead of SOG in a similar process has been reported [24], apparently with similar reliability problems.

## Conclusion

The described planarization methods are widely used in technologies employing 2, and sometimes 3, interconnect metallization layers. A detailed understanding of the planarization process is required to design a IC fabrication process which results in the required degree of planarization on a production wafer. Global planarization with these processes is extremely difficult, if not impossible, to achieve, but local planarization with a planarization factors exceeding of 0.9 is certainly within their reach. Their use requires suitable changes in the design rules. As the trend in advanced IC design today is toward the elimination of all such restrictions regarding the dielectric layers, even in technologies with more than 2 interconnect levels, it seems that this goal, which has not yet been reached, is attainable only with development of methods of total global planarization, together with technologies that allow vertically sided contact holes and vias of varying depths to be completely filled.

## Acknowledgements

The use of IMP (San Jose, Ca., USA) facilities for some of the experimental work presented is gratefully acknowledged. The study has been supported by a grant from the

Ministry of Education, Science, and Sport of the Republic of Slovenia.

## References

- /1./ S. Wolf, Silicon Processing for the VLSI Era, Vol. 2 – Process integration, Lattice Press, Sunset Beach, Ca. USA, 1990
- /2./ P. B. Johnson and P. Sethna, Semiconductor Int., Oct. 1997, p. 80
- /3./ J. Pirš, IJS, private communication, 2000
- /4./ Lee J.W. et al, Proc. Adv. Metallization Conf. 1998, Mater Res. Soc., Warrendale, PA (USA), 1999.
- /5./ Suk-Kyoung Hong et al., Intermetal Dielectric Process Using SOG for Ferroelectric Memory Devices having  $\text{SrBi}_2\text{Ta}_2\text{O}_9$  Capacitors, J. Mater. Res., Vol. 12, No. 1, 1997
- /6./ Smith J.H. et al, Semiconductor Int., Vol 21, No. 4, April 1998
- /7./ B. Gspan, Ph.D. Thesis, Faculty of Electrical Eng., University of Ljubljana, 1995
- /8./ Kirk S. et al., Cleaning Technology in Semiconductor Device Manufacturing, Proc. 6 th. Int. Symposium, Electrochem. Soc. Proc. Vol 99-36, Electrochem. Soc, Pennington, NJ, USA, 2000
- /9./ R. Osredkar, Inf. MIDEM, Vol. 31, No. 2(89), Ljubljana, SI, June 2001
- /10./ R. Osredkar, Spin-on-Glass Material Curing and Etching, Microelectron. Reliab., Vol. 34, No. 7, 1994, p. 1265
- /11./ C.H. Ting et al., V-MIC Conf. Proc, June 1987
- /12./ S.K. Gupta, Microelectron. Mfg. Testing, April 1989
- /13./ J. D. Romero et al., Outgasing Behavior of SOG, J. Mater. Res., Vol. 11, No. 9, 1996, p.1996
- /14./ L. K. White, Approximating Spun-on, Thin Film Planarization Properties on Complex Topography, J. Electrochem Soc. Solid State Science and Technology, Vol. 132, No. 2, 1985, p. 169
- /15./ Zhvayyi S.P., et al, Technical Physics, Vol. 43, No. 11, Nov. 1998
- /16./ H. Eggers et al., Proc. 2<sup>nd</sup> Intl IEEE VMIC Conf, Santa Clara, CA, USA, 1985, p. 163
- /17./ R. Osredkar, Microelectron. Reliab., Vol. 34, No. 7, 1994, pp. 1265-1267
- /18./ Galenski N., Advanced APCVD: BPSG Film Quality and Production Reliability Report, Watkins-Johnson Report, 1992
- /19./ R. Osredkar, B. Gspan, Inf. MIDEM, Vol. 30, No. 2(94), Ljubljana, June 2000
- /20./ R. J. Kopp, Semicond. Internatl. January 1989, p 54
- /21./ M.J. Thoma et al., Proc. 4<sup>th</sup> Internatl IEEE VMIC Conf., Santa Clara, CA, USA, 1987, p. 20
- /22./ A. Schepela and B. Soller, J. Electrochem. Soc., March 1987, p. 714
- /23./ M. Popal et al., Proc. 48 th Electronic Components Technology Conf., IEEE, NY, USA, 1998
- /24./ P. Chiniwalla et al., Electrochem. Soc. Proc. Vol 99-7, Electrochem. Soc., Pennington NJ, USA, 2000, pp. 135-142

Radko Osredkar  
FRI in FE Univerze v Ljubljani  
Tržaška 25, SI 1000, Ljubljana, Slovenia  
e-mail: radko.osredkar@fri.uni-lj.si

Boštjan Gspan  
Repro MS  
Šmartinska 106, SI 1000, Ljubljana, Slovenia  
e-mail: bostjan.gspan@repro.ms.si



# DISPLACEMENT MEASUREMENT USING OPTICAL FIBER REFLECTION SENSORS

Alojz Suhadolnik, Jože Petrišič

Faculty of Mechanical Engineering, University of Ljubljana, Slovenia

**Key words:** light emitting sensors, optical fibers, optical fiber sensor, displacement measurements, intensity sensor, reflection sensor

**Abstract:** Optical fiber reflection sensors for the displacement measurement are described. We tested different optical fiber reflection sensors configurations. The displacements are measured by approaching the optical fiber tip toward reflective surface and by moving it parallel to the surface. Various sensor responses are detected by using an optical power meter and analyzed with the computer.

## Merjenje pomika z uporabo odbojnostnih senzorjev z optičnimi vlakni

**Ključne besede:** svetlobni senzorji, optična vlakna, senzorji z optičnimi vlakni, merjenje pomikov, intenzitetni senzor, odbojnostni senzor

**Izvleček:** Opisani so odbojnostni senzorji z optičnimi vlakni za merjenje pomikov. Preizkusili smo različne konfiguracije odbojnostnih senzorjev z optičnimi vlakni. Princip delovanja sloni na približevanju ali vzporednemu gibanju senzorske konice glede na površino, ki svetlobo odbija. Signale smo izmerili s pomočjo optičnega merilnika moči in analizirali z uporabo računalnika.

### 1. Introduction

Several optical fiber sensors for the displacement measurement have been designed [1]. Most of them are based on the optical intensity variation. An optical fiber micro-banding sensor was developed for measuring the displacement [2]. The optical fiber interferometers measure small displacements and vibrations with high precision [3]. The reflective type optical fiber sensors are also described [4]. They can be used for the sound detection as a microphone [5]. We measured refractive indexes of fluids using this kind of the fiber optic sensor [6]. They are simple in construction and could be used in explosive environments. In this paper we shortly describe an intensity displacement fiber optic sensor with two parallel fibers. In addition we describe the optical fiber sensor with a circular input optical fiber and a ring type output optical fiber. The displacement measurements were performed by approaching the fiber tip toward the reflective surface and by moving the sensor parallel to the surface where we used a surface with two different reflexive layers.

### 2. Principles of operation

The optical fiber reflection sensor in general consists of two multimode optical fibers. Both fibers are bound together in a sensor tip. The first fiber is the incoming one and the second is the outgoing (Fig.1). A maximum angle  $\Theta_{NA}$  within which the light leaves the output fiber at the sensor tip is equal to

$$\Theta_{NA} = \arcsin(NA), \quad (1)$$

where NA is numerical aperture of the optical fiber.

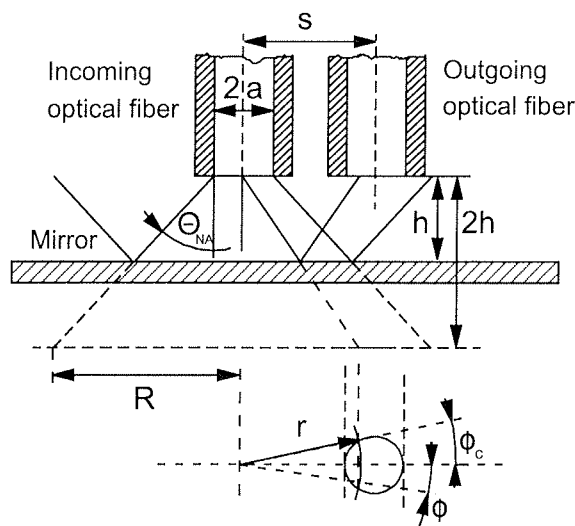


Fig.1. Schematic of the sensor

If  $P_t$  denotes the total optical power transmitted through the incoming fiber and  $P_o(2h)$  the captured power by the outgoing fiber, then the efficiency factor  $\eta(2h)$  at the distance  $h$  between the fiber tip and the reflective surface is equal to

$$\eta(2h) = P_o(2h) / P_t. \quad (2)$$

If we assume that the far field optical intensity distribution at the incoming optical fiber tip  $I(r, 2h)$  is parabolic,

$$I(r, 2h) = \frac{2P_i}{\pi R^2(h)} \left( 1 - \frac{r^2}{R^2(h)} \right), \quad (3)$$

then we can write

$$\eta(2h) = 2 \int_0^{R_2} \int_0^{\phi_c} R_s T_i T_o(r, 2h) \frac{I(r, 2h)}{P_i} r d\phi dr, \quad (4)$$

where  $R_s$  is surface reflectivity,  $T_i$  and  $T_o$  the Fresnel transmittance coefficients of the incoming and outgoing optical fibers /6/. In this equation  $R$  is the radius of the light cone at the distance  $2h$  and  $h$  is the distance between the mirror and the sensor tip

$$R = a + 2h \operatorname{tg}(\Theta_{NA}). \quad (5)$$

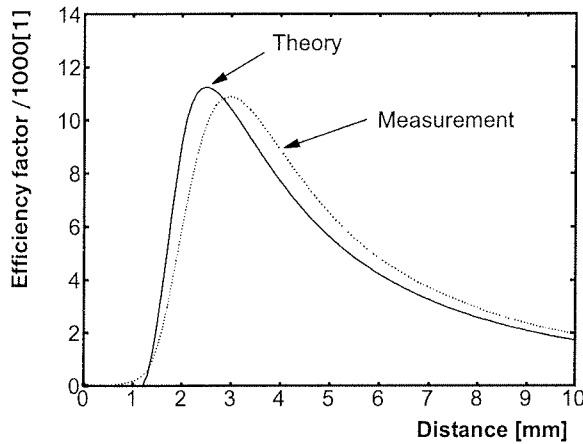


Fig.2. Theoretical and experimental sensor response in moving the sensor tip outward the mirror.

The maximum azimuth angle  $\phi_c$  is equal to

$$\phi_c = \arccos((s^2 + r^2 - a^2)/(2rs)), \quad (6)$$

where  $s$  is the distance between the fiber core axes and  $a$  the optical fiber core radius.

The first integration limit  $R_1$  in the integral (4) is  $R_1 = s - a$  if  $R > s - a$  otherwise  $R_1 = 0$ . The second limit  $R_2$  is equal to  $R_2 = s + a$  if  $R \geq s + a$  and  $R_2 = R$  if  $s - a < R < s + a$  otherwise  $R_2 = 0$ .

In Fig. 2 the theoretical efficiency and the experimental curves are shown. The sensor consists of two identical optical fibers. The optical fibers used had a core radius  $a = 0.5$  mm,  $s = 2.2$  mm and  $NA = 0.47$ .

### 3. Ring type sensor

In this section a ring type sensor is described. The schematic of this sensor is shown in fig.3.

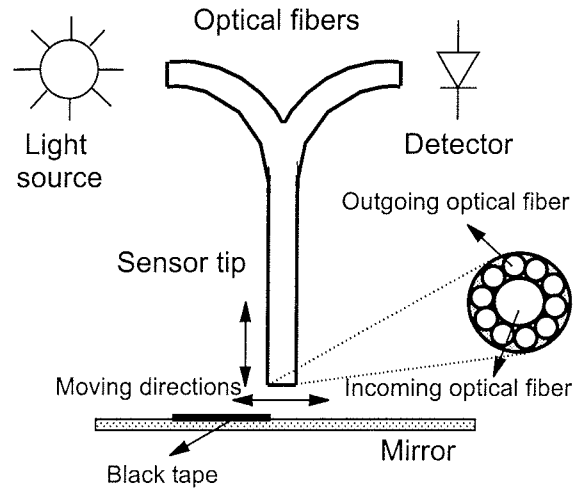


Fig.3. Ring type optical fiber reflection sensor

The ring type sensor has one incoming optical fiber and several small fiber fragments arranged in a ring around the incoming optical fiber. The ring shaped fiber fragments are gathered in the outgoing optical fiber. This type of the sensor tip is capable of collecting more light than the reflection sensor with the two circular optical fibers. We used the red LED as the light source and optical power meter as the detector. The measured results were transferred to the computer for further analysis.

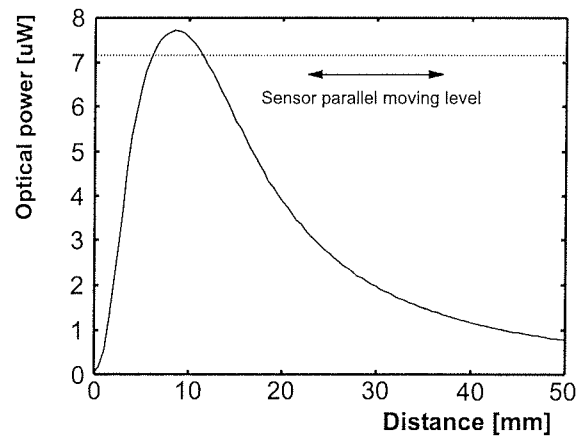


Fig.4. Sensor response for the ring type optical fiber reflection sensor

The sensor response by moving the sensor outward the reflective surface is shown in Fig.4. The ring type optical fiber sensor consists of the same type of the optical fibers as in the previous configuration. In addition we measured the sensor characteristics by moving the sensor tip parallel to the reflective surface. One part of the reflective surface was covered with the black color type. The reflectance upon the colored part of the surface is much smaller. In Fig. 5 the sensor response is shown.

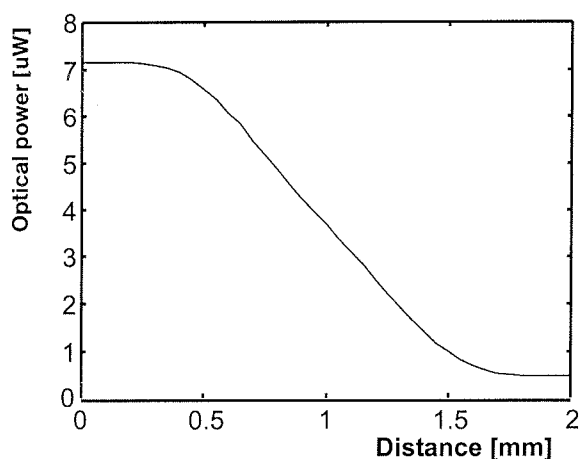


Fig.5. Sensor responses by moving the sensor tip parallel to the surface.

The sensor tip is close to the surface where the sensor response is near the maximum. The sensor characteristic decrease rapidly when the sensor tip passes over the color border. It can be seen from Fig. 5 that the sensor response is linear within 1 mm. We used this sensor for monitoring an electro welding head displacement.

#### 4. Conclusions

This paper describes two types of optical fiber reflection sensors. The first one consists of two fibers and the second one is a ring type optical fiber sensor. The sensors working principles are evaluated as well as the sensors characteristic responses. Those sensors are capable of measuring a position of a few millimeters and detecting the displacement of the objects with the different surface reflectivity.

#### References

- /1/ E. Udd, Fiber optic sensors, John Wiley & Sons, New York, 1991
- /2/ W. H. G. Horsthuis, J. H. J. Fluitman, The development of fibre optic microbend sensors, *Sensors and actuators*, 3, 1982/83, 99-110
- /3/ A. D. Drake, D. C. Leiner, Fiber-optic interferometer for remote subangstrom vibration measurement, *Rev. Sci. Instrum.*, Vol. 55, No. 2, 1984, 162-165
- /4/ R. O. Cook, C. W. Hamm, Fiber optic lever displacement transducer, *Appl. Opt.* 18, 1979, 3230-3241.
- /5/ A. Suhadolnik, A. Babnik and J. Možina, Microphone based on fibre optic reflective sensor, *Europto, Fiber Optic and Laser Sensors XIII*, SPIE Vol. 2510, Munich, FRG, Jun 20-21, 1995, 120-127
- /6/ A. Suhadolnik, A. Babnik and J. Možina, Optical fiber reflection refractometer, *Sensors and Actuators B*, B29 (1995) 428-432.

*dr.Alojz Suhadolnik and dr.Jože Petrišič*  
*Faculty of Mechanical Engineering*  
*Aškerčeva 6*  
*1000 Ljubljana*  
*Slovenia*  
*E-mail: alojz.suhadolnik@guest.arnes.si*  
*joze.petrisic@fs.uni-lj.si*

*Prispelo (Arrived): 30.05.2002      Sprejeto (Accepted): 28.06.2002*

# REDUCING EMI OF COMMUTATOR MOTORS BY OPTIMIZING BRUSH-TO-SEGMENT WIDTH RATIO

<sup>1</sup> France Pavlovčič, <sup>2</sup> Janez Nastran

<sup>1</sup> Ministry for environment, spatial planning and energy; Environmental Agency of the Republic of Slovenia, Ljubljana, Slovenia

<sup>2</sup> University of Ljubljana, Faculty of electrical engineering, Ljubljana, Slovenia

**Key words:** commutator motor, brush-to-segment width ratio, commutation, motion equation in electrical coordinates, minimization of electromagnetic interference.

**Abstract:** Commutator motors are widely used in several consumer appliances especially in household apparatuses and electrical hand tools. Therefore they are economically very important for their numerous production. Furthermore, due to their high shaft speed they achieve high power per a unit of volume, and they are produced at relatively low costs for a unit of power. But these relatively powerful and economical machines are unfavourable electromagnetic interference sources, which can compose the disturbing electromagnetic environment. Electromagnetic interference is caused primarily by a commutation of a current in an armature coil. The paper describes a theoretical approach to a minimization of electromagnetic interference of the commutator motors through concepts of analytical mechanics. These concepts involve both a magnetic system and motion systems of the motor using the magnetic and the electric energy. Further, a criterion is established for an optimization of a brush-to-segment width ratio of a commutator-brush system due to minimal amplitudes of electromagnetic interference and minimal contents of harmonics in a spectrum of electromagnetic interference.

## Zmanjšanje EMI kolektorskih motorjev z optimiranjem prekrivanja lamel komutatorja

**Ključne besede:** kolektorski motor, prekrivanje lamel, komutacija, gibalna enačba v električnih koordinatah, minimizacija elektromagnetne interference

**Izveček:** Kolektorski motorji so množično uporabljeni v številnih napravah široke potrošnje, posebno v gospodinjstskih aparatih in električnih ročnih orodjih. Zaradi njihove velikoserijske proizvodnje so ekonomsko zelo pomembni. Ker dosegajo velike hitrosti vrtenja, razvijejo veliko moč na enoto volumna in njihovi proizvodni stroški na enoto moči so relativno majhni. Toda ti zelo ugodni stroji glede na moč in ekonomiko so nezaželeni elektromagnetni interferenčni izvori in lahko tvorijo motilno elektromagnetno okolje. Elektromagnetne interferenčne motnje povzročajo predvsem komutacija toka v svetkih armature. Članek teoretično obravnava minimiziranje elektromagnetnih motenj kolektorskih motorjev z uporabo analitične mehanike. Tak pristop zahteva obravnavo magnetnega in gibalnega sistema motorja v povezavi z magnetno in električno energijo. Izdelan je bil tudi kriterij za optimiranje prekrivanja lamel glede na minimalne amplitude elektromagnetnih motenj in minimalno vsebnost višjih harmonskih komponent v njihovem spektru.

### 1. Introduction

When researching the phenomenon of the commutation two methods based on the theory of electric circuits are used /1,2,3/. When these methods are used, each specific motor require its own model, valid only for this motor. The first method applies transfer differential equations to the electric circuit of the motor, which is built up by time-depending elements. The second method is also applied to the electric circuit of the time-depending elements, but it is based on a time-varying circuit topology.

In a general case, when we are looking for general rules of the commutation, it is more convenient to use the principles of analytical mechanics in systems of the motor applying the magnetic and the electric energy of the motor. The commutation is the process of reversing the current in one coil; it begins when a brush bridges two commutator bars which are the terminals of this coil, and finishes at the latest when the brush leaves out the leading commutator bar of this coil with possible transients; geometrical ac-

tions of bridging and leaving are defined as a geometrical performance of the commutation.

Due to rotation of the commutator the brush consecutively makes and breaks electric contacts over the armature coils. Making and breaking the electric contact over the armature coils are succeeding each other: making the electric contact over one armature coil is followed by breaking the electric contact over the other armature coil. This movement is defined in this paper as an interchange of positions of the armature coils under the brush bridge. An interval between two consecutive interchanges is a pause. Because the armature coils are magnetically coupled, the commutation of the current and geometrical displacements causes changes in the magnetic energy of the whole magnetic system of the motor. Due to these changes the voltage is induced in the windings of the motor, which is the source of electromagnetic interference.

A mathematical model of the magnetic system is built up with a purpose to calculate the magnetic energy and its changes due to the commutation of the current and the

geometrical displacements of the armature coils. This mathematical model contains an algorithm of dynamic calculations of a magnetic reaction of the armature and also takes into account a kinetic behaviour of the commutator regarding the brushes. This algorithm is the only way to determine the changes in the magnetic energy as a function of a rotation angle.

The voltage of the source of electromagnetic interference is a result of a conversion of the magnetic energy into the electric energy of stray capacitance. Therefore motion systems in electrical and in geometrical coordinates are established. Because both of the motion systems are defined by conceptual the same motion equation, the experimental results of one system evaluates also the another system.

In the case without transients at the end of the commutation process, electromagnetic interference results from changes of the magnetic energy of the machine. This kind of emitted interference propagates mostly conductively to a supplying network. There are almost no radiated emissions.

Intending to make mathematical models more clear, a d.c. serial motor with two poles is chosen as a representative type of the commutator machines.

## 2. The magnetic system

To establish the mathematical model of the magnetic system /4/ of the DC motor, a system of Maxwell's equations in the integral form is to be solved. As the main purpose of the presented model is to study the commutation, it is particularly important to detail the total current in relation with the geometry and kinetic behaviour of the motor.

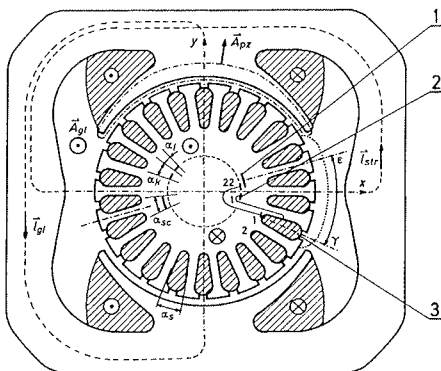


Fig.1: The schematic cross-section of the motor: (1) the brush, (2) the commutator bar, (3) the armature coil.

The total current is obtained by an integral of a conduction current density through a surface ( $A_{gi}$ ) defined by the closed contour ( $I_{gi}$ ) - Fig.1. A geometrical distribution of the coils around the armature is closely taken into account. Thus the total current depends on the number of turns of stator

coils ( $N_{st}$ ) and the number of turns of the armature coil ( $N_s$ ). Furthermore it is function of a stator current ( $i_{st}$ ), a coefficient of the coil ( $\kappa(\alpha_s) < 1$  and  $\kappa(\alpha_s) \approx 1$ ), depending on a width of the coil ( $\alpha_s$ ), and also of the number of commutator segments ( $n_l$ ), which equals the number of slots of the armature in this case. It also depends on an angle defining a position of the coils towards the commutator segments ( $\gamma$ ) and on a rotation angle ( $\alpha$ ):

$$\iint_{A_{gi}} J \cdot dA = N_{st} \cdot i_{st} + \kappa(\alpha_s) \cdot N_s \cdot \left( \cos(\alpha + \gamma) \cdot \sum_{i=1}^{i=n_l} i_s \left( \alpha - (i-1) \cdot \frac{2 \cdot \pi}{n_l} \right) \cdot \cos \left( (i-1) \cdot \frac{2 \cdot \pi}{n_l} \right) + \sin(\alpha + \gamma) \cdot \sum_{i=1}^{i=n_l} i_s \left( \alpha - (i-1) \cdot \frac{2 \cdot \pi}{n_l} \right) \cdot \sin \left( (i-1) \cdot \frac{2 \cdot \pi}{n_l} \right) \right) \quad (1)$$

An expression  $i_s(\alpha - (i-1) \dots)$  at  $i=1$  in equation (1) is a current distribution around a circumference of the armature as a function of the angle, and of intervals of the geometrical performance of the commutation. These intervals depend on widths of the commutator segment ( $\alpha_k$ ), of the commutator bar ( $\alpha_l$ ), of the brush ( $\alpha_{sc}$ ) and on an angle of the brushes towards a geometrical neutral point ( $\varepsilon$ ) - Fig.1.

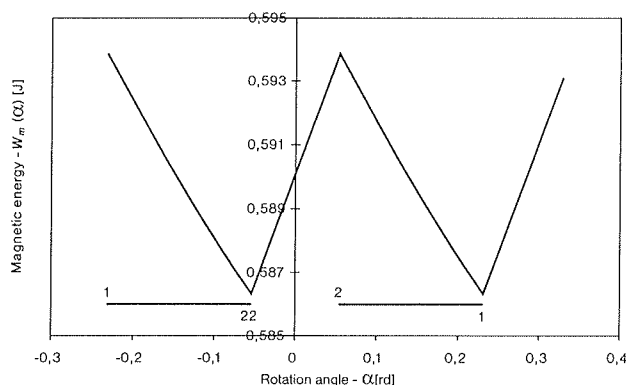


Fig.2: The magnetic energy  $W_m(\alpha)$ , depending on the rotation angle, at the linear commutation and the brush-to-segment width ratio of 1.8.

Resolving the system of Maxwell's equations, the magnetic field intensity in all parts of the magnetic system is determined as a function of the rotation angle of the armature. Due to a magnetization curve of a ferromagnetic material, the magnetic energy ( $W_m$ ) depending on the rotation angle is calculated.

Analysing the function of the magnetic energy  $W_m(\alpha)$  in Fig.2, the two interchange intervals are represented by straight lines from the point 1 to the point 22 and from the point 2 to the point 1 with the pause between them. The geometrical performance during the commutation lasted from the first point 1 (the instant of making the electric contact over the coil 1) to the last point 1 (the instant of breaking the electric contact over the coil 1). It includes both interchanges.

### 3. The motion system in electrical coordinates

The motion equation is describing the relation between the kinetic and the potential energy of any motion system in general [5]. Instead of the kinetic and the potential energy, the magnetic ( $W_m$ ) and the electric energy ( $W_e$ ) are applied respectively in a particular system where the generalized coordinates are electric charges and the generalized velocities are currents. The system is subjected to actions of the conservative forces and the nonconservative forces, which result in the active and the reactive force. The active force is defined by the derivative of the magnetic energy with respect to a supplying current of the system and further on with respect to time. But the reactive force is defined by the derivative of Rayleigh's dissipation function ( $F_i$ ) with respect to a current which causes dissipation. The dissipation function is one half of an electric power dissipation [6].

The magnetic energy is obtained from the magnetic system of the motor. From the viewpoint of the electrical coordinates, it depends only on currents which are the current of the motor ( $i_m$ ) and a current ( $i_c$ ) through stray capacitance. On the other hand the electric energy depends only on the electric charge ( $q_c$ ) found with stray capacitance of the motor. The time derivative of the electric charge is the current ( $i_c$ ) through stray capacitance. The current through stray capacitance also causes the electric power dissipation. Taking into account isomorphism of time  $\{t, +\}$  and rotation angle  $\{\alpha, +\}$ , the equation of motion is written as follows:

$$\omega_r \cdot \frac{d}{d\alpha} \frac{\partial W_m}{\partial i_c} + \frac{\partial W_e}{\partial q_c} = \omega_r \cdot \frac{d}{d\alpha} \frac{\partial W_m}{\partial i_m} - \frac{\partial F_i}{\partial i_c} \quad (2).$$

Summands of this equation (2) are generalized forces, which are, speaking in terms of the electrical coordinates, relevant voltages of the system. The first summand is the voltage across a conceptual inductance of the motor, caused by the current through stray capacitance. The second one is the voltage ( $u_c$ ) across a conceptual stray capacitance, which is the voltage on terminals of the motor. On the right side, the first summand is the induced voltage ( $u_{ind}$ ) due to the changes of the magnetic energy, which is, in fact, the voltage of the electromagnetic interference source. Further on, there is also the voltage drop across a conceptual resistance due to the electric power dissipation. Using equivalent concentrated elements instead of the conceptual ones, and introducing the voltage ( $u_c$ ), the second-order differential equation (3) is obtained:

$$\omega_r^2 \cdot L_{eq} \cdot C_{eq} \cdot \frac{d^2 u_c}{d\alpha^2} + \omega_r \cdot R_{eq} \cdot C_{eq} \cdot \frac{d u_c}{d\alpha} + u_c = \omega_r \cdot \frac{d}{d\alpha} \frac{\partial W_m}{\partial i_m} \quad (3).$$

The induced voltage ( $u_{ind}$ ) is derived from the magnetic energy, but the voltage on the terminals of the motor ( $u_c$ ) is the result of the equation (3) - Fig.3. Further on, ampli-

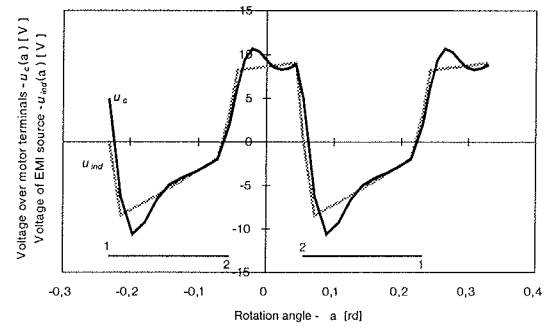


Fig.3: The voltage of EMI source  $u_{ind}(\alpha)$  and the voltage over the motor terminals  $u_c(\alpha)$ , depending on the rotation angle, at the linear commutation, the brush-to-segment width ratio of 1.8 and the shaft speed of 7.8 kcys/min.

tudes and rates of the harmonics in the spectrum of the voltage  $u_{ind}(\alpha)$  are to be determined as functions of the brush-to-segment width ratio. The minima of these functions are then established thus providing the basis for the optimization of the brush-to-segment width ratio.

### 4. The motion system in geometrical coordinates

A torque of the motor is determined using the motion system of the motor in geometrical coordinates. To simplify the motion system, it is presumed, that the system works without mechanic losses. Consequently, the motion system applies, beside conservative forces, only two nonconservative forces, which is the torque of the motor and a torque determined by the electric dissipation. Due to isomorphism of the angles  $\{\alpha\}$  and time  $\{t\}$ , the trivial value of the torque would result. Therefore displacements of the magnetic field of the armature, which directly cause the changes in the magnetic energy, are described by another angular coordinate ( $\alpha^*$ ). The rule of correspondence  $\{\alpha\} \rightarrow \{\alpha^*\}$  is obtained from the mathematical model of the magnetic system. As an inverse rule of correspondence does not exist, the torque of the motor is related to the actual angular coordinate rotation ( $\alpha$ ) and therefore to time. The equation of motion in geometrical coordinates results in the following equation:

$$M(\alpha) = \frac{\partial (W_m^*(\alpha^*) - W_e^*(\alpha^*))}{\partial \alpha^*} - \frac{\partial F_i(\omega_r)}{\partial \omega_r} \quad (4).$$

While optimizing the commutator-brush system, average values of the torque depending on the brush-to-segment width ratio are to be calculated.

### 5. The optimization of the brush-to-segment width ratio

To minimize electromagnetic interference, the brush-to-segment width ratio is chosen to be optimized. It is a quo-

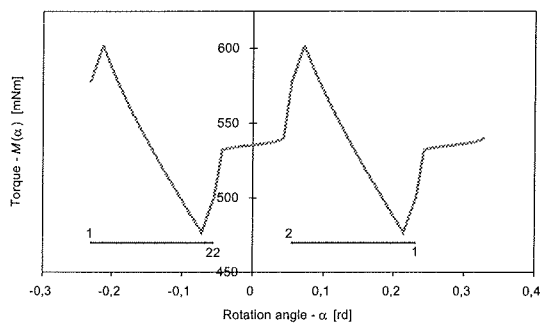


Fig.4: The torque  $M(\alpha)$ , depending on the rotation angle, at the linear commutation and the brush-to-segment width ratio of 1.8.

tient of the width of the brush and the width of the commutator segment.

According to the mathematical models of the magnetic and the motion system in the electrical coordinates - both are supplemented with the variable brush-to-segment width ratio ( $r$ ) - the functions  $W_m(\alpha, r)$  and  $u_{ind}(\alpha, r)$  are established for a chosen domain of definition of the brush-to-segment width ratio.

A function of amplitudes  $U_{ind}(r)$  of the voltage  $u_{ind}(\alpha, r)$  and a function of the rates  $h_{ind}(r)$  of the harmonics in the spectrum of this voltage, both depending on the brush-to-segment width ratio, are now obtained.

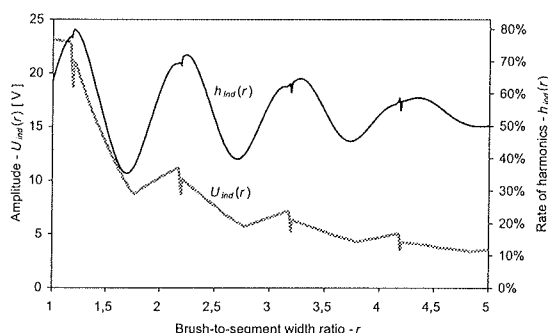


Fig.5: The amplitude of the EMI source voltage  $U_{ind}(r)$  and the rate of the harmonics  $h_{ind}(r)$ , depending on the brush-to-segment width ratio, at the linear commutation and the shaft speed of 7.8 kcs/min.

The functions  $U_{ind}(r)$  and  $h_{ind}(r)$  in Fig.5 are set up for the linear commutation current. It is seen that the minima of the amplitude function  $U_{ind}(r)$  and the minima of the harmonics rates function  $h_{ind}(r)$  do not coincide. The optima of the brush-to-segment width ratio are achieved at the minima of the amplitude functions.

The function of the torque  $M(r)$  depending on the brush-to-segment width ratio is shown on a diagram in Fig.6.

The function monotonously decreases with the increasing values of the brush-to-segment width ratio. This function has no optima of the brush-to-segment width ratio.

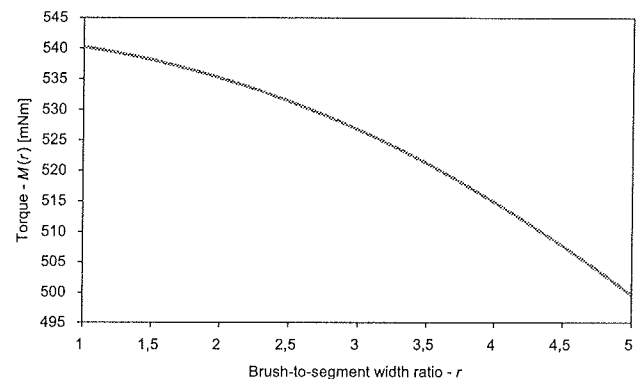


Fig.6: The torque  $M(r)$ , depending on the brush-to-segment width ratio, at the linear commutation.

## 6. The experimental evaluation of both motion systems

The numerical analyses, carried out previously by the mathematical model, took into account technical data of the commutator motor produced by Slovenian firm DOMEI. In the mathematical models, the linear commutation of the current was presumed up to this point. The motor under test, loaded by an electromagnetic brake, was especially assembled with measuring slip rings and brushes to measure currents through two successive armature coils. The measurement shows that the current of the armature coil reverses before the geometrical performance of the commutation is accomplished. There is overcommutation of the current. To compare the mathematical results and measured results, a simulation of overcommutation is done.

Diagrams in Fig.7 show the calculated induced voltage of the electromagnetic interference source and the measured voltage over the terminals of the motor.

There are differences between both functions, and these differences were also conceptually predicted by the diagrams in Fig.3. The simulated induced voltage is the source voltage, but the voltage over the terminals of the motor is the response of a spatial distributed electric circuitry of the motor on the induced voltage. The induced voltage cannot be measured. The electric circuitry of the motor has more than one resonant frequency, so the simulation of the response on the induced voltage by it is more complex than just solving the differential equation (3). Considering the optimization of the commutator-brush system to minimize electromagnetic interference, it is more appropriate to minimize the induced voltage for it is the source voltage. The comparison of both voltages, however, verifies the electric motion system of the motor.

The motion system in geometrical coordinates is established to determine the function of the torque of the motor depending on the brush-to-segment width ratio. There is also another purpose of the mathematical model of this mechanic motion system: verification of both motion systems for they are based on the same motion equation, but using the different kind of generalized coordinates.



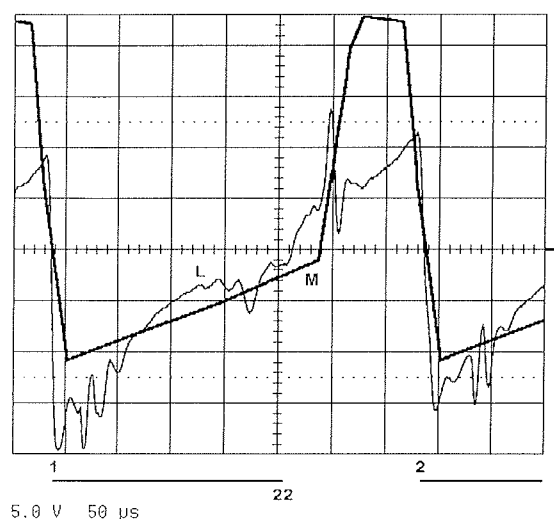


Fig.7: The voltage of EMI source  $u_{ind}(\alpha)$  simulated by the mathematical model (M) and the voltage over the motor terminals  $u_c(\alpha)$  measured in the laboratory (L), depending on time at the shaft speed of 7.8 kcys/min.

This verification is made by comparison of the calculated torque characteristic of the motor for d.c. conditions and two measured torque characteristics, one achieved by d.c. measurements in the laboratory, but the other by a.c. measurements in the factory - Fig.8. Both torque characteristics, simulated and measured in the laboratory are very close to each other. The torque characteristic, measured in the factory, shows lower values of the torque than the other two characteristics. There are some considerations about possible causes of the difference: the different kind of supply currents (d.c. and a.c.), tolerances of assembled parts and materials, some inadequacies in the simulation of the magnetic reaction of the armature and in the simulation of saturation of the ferromagnetic material.

Taking all comparisons of the simulated and the measured quantities into account, it can be considered that the magnetic and both motion systems and also the method of the optimization of the brush-to-segment width ratio are verified.

## 7. Conclusions

On the principles of analytical mechanics the mathematical model was established applying the magnetic and the electric energy of the commutator motor overall to optimize the commutator system of the motor. Knowing the current distribution around the circumference of the armature, the brush-to-segment width ratio can be optimized to minimize the amplitude of electromagnetic interference and/or the rate of the harmonics in its spectrum. In practice the arcing occurs between the brushes and the commutator bars, and in this case, the optimization method leads to minimal arc energy and so far to longer duration of the commutator.

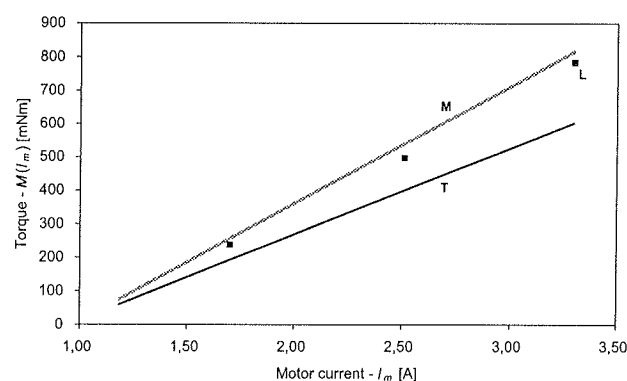


Fig.8: The torque characteristics of the motor under the test: obtained by the mathematical model (M) - d.c. conditions, measured in the laboratory (L) - d.c. supply, tested in the factory (T) - a.c. supply.

## References

- /1/ SACK, J.; SCHMEER, H.: 'Computer-aided analysis of the RFI voltage generation by small commutator motors', IEEE-EMC Society, 1985, Symposia Records 1955 to 1995, CD-ROM Database 1996.
- /2/ SURIANO, J.; ONG, C.M.: 'Modeling of electromechanical and electromagnetic disturbances in DC motors', IEEE-EMC Society, 1989, Symposia Records 1955 to 1995, CD-ROM Database 1996.
- /3/ SURIANO, C.R.; SURIANO, J.R.; THIELE, G.; HOLMES T.W.: 'Prediction of Radiated Emissions From DC Motors', IEEE-EMC Society, 1998, IEEE Inc, Symposia Records 1996 to 1999, CD-ROM Database 1999.
- /4/ WOODSON, H.H.; MELCHER, J.R.: 'Electromechanical dynamics, Part I: Discrete systems' (John Wiley & sons, inc. New York, London, Sydney 1968).
- /5/ YAVORSKY, B.; DETLAF, A.: 'Handbook of Physics' (Mir publishers, Moscow 1975, English translation).
- /6/ RAYLEIGH, J.W. STRUTT: 'The theory of sound' (Dover Publications, New York, 1945, Volume 1, 2<sup>nd</sup> edition).

dr. France Pavlovčič, univ.dipl.ing.  
Ministrstvo za okolje in prostor, Agencija RS za okolje,  
Vojkova 1b, Ljubljana  
tel.: +386 (01) 4784 098, fax: +386 (0)1 4784 054, E-mail: france.pavlovic@gov.si

prof.dr. Janez Nastran, univ.dipl.ing.  
Univerza v Ljubljani, Fakulteta za elektrotehniko,  
Tržaška 25, Ljubljana  
tel.: +386 (01) 4768 282, fax: +386 (0)1 4264 647, E-mail: janez.nastran@fe.uni-lj.si

# INFLUENCE OF THROAT AREA ON THE RESISTANCE SPOT WELDING PROCESS

Janez Tušek<sup>1</sup>, Miro Uran<sup>1</sup>, Miran Vovk<sup>2</sup>

<sup>1</sup>Institut za varilstvo, Ljubljana, Slovenia

<sup>2</sup>Faculty of Mechanical Engineering, Ljubljana, Slovenia

**Key words:** resistance welding, throat area, workpiece, welding-cable length, alternating welding current, voltage drop, contact resistance, impedance

**Abstract:** The paper describes an experimental study made in order to establish how the welding-cable length in resistance spot welding affects welding parameters and the welding process itself. A common alternating-current welding device for welding with a current frequency of 50 Hz was used to investigate optimum welding parameters with welding cables of different length and with workpieces of different materials. It was found that longer welding cables produce increases in ohmic resistance and inductive resistance. Higher ohmic resistance produces thermal losses and a lower welding current in the secondary circuit. Inductive resistance, however, produces reactance in the secondary circuit of the welding transformer, which means a loss as far as resistance welding is concerned.

## Vpliv velikosti okna med elektrodami na proces elektrouporovnega točkovnega varjenja

**Ključne besede:** elektrouporovno varjenje, okno med varilnima elektrodama, varjenec, dolžina varilnih kablov, izmenični varilni tok, padec napetosti, kontaktna upornost, impedanca

**Izvleček:** V članku z gornjim naslovom je z eksperimentalnimi raziskavami pokazano, kako dolžina varilnih kablov pri elektrouporovnem točkovnem varjenju, vpliva na varilne parametre in na sam proces varjenja. Na klasični varilni napravi za varjenje z izmeničnim električnim tokom s frekvenco 50 Hz smo raziskali optimalne varilne parametre za različno dolge varilne kable in za varjenje iz različnih materialov. Ugotovili smo, da se s podaljšanjem varilnih kablov povečata ohmska in induktivna upornost. Pri večji ohmski upornosti nastopijo toplotne izgube, v sekundarnem krogu pa teče varilni tok nižje jakosti. Zaradi induktivne upornosti se v sekundarnem krogu varilnega transformatorja pojavi jalova moč, ki za uporovno varjenje v celoti predstavlja izgubo.

### 1. Introduction

Considering the number of welds produced, resistance spot welding is undoubtedly the most frequently applied welding process in all production technologies, particularly in batch production of cars, household appliances, and electrotechnical elements. Increasing demands for product quality and traceability in their usage make manufacturers introduce new methods of monitoring and recording of welding parameters in the course of welding and their storage after welding. The most important parameters in resistance spot welding are the welding current, the weld time, and the electrode force. The welding parameters are selected with reference to material properties and thickness and a workpiece shape. It is on the workpiece size and shape that depends the throat area. A frequent difficulty encountered in resistance spot welding are welding cable lengths and different throat areas. The degree of obstruction of the throat area depends on the size and shape of the workpiece and the position of the weld spot at the workpiece, which is different for each weld spot. The size and shape of the throat area and its obstruction by the workpiece affect the welding parameters, particularly the welding current. This should be taken into account when elaborating a welding technology. The greatest influence on the welding current is exerted by the length

of cables at the secondary side, i.e., impedance of the cables. In practical applications it often happens that cables more than one metre or even several metres long have to be used because of the workpiece size. They may or may not be placed across the workpiece. This means that the throat area is fully or partly obstructed by the workpiece or it is unobstructed. This affects the welding parameters and the welding process.

### 2. Problem to be addressed

Figure 1 schematically shows the resistance spot welding process. Through both workpieces a welding current of high density is flowing. Because of ohmic resistance, particularly contact resistance, between the two workpieces, a part of the material will heat up to its melting point and the electrode force will produce a spot weld at a lap joint or a parallel joint. The figure also shows relative resistance occurring when the current is carried through the electrodes and the workpieces. It may be observed that the contact resistance between the workpieces is several times higher than the resistance in the workpieces themselves and much higher than the contact resistance between the electrodes and the workpieces. The alternating current with a frequency of 50 Hz is supplied by a welding transformer

which is the main element of a resistance welding machine. The entire circuit at the secondary side consists of the welding cables, i.e., electrode holders, the electrodes and two or sometimes more workpieces. In this circuit, physical-chemical processes are going on. In the electrode holders, i.e., in the welding cables, they will produce ohmic and inductive resistances which affect the welding current.

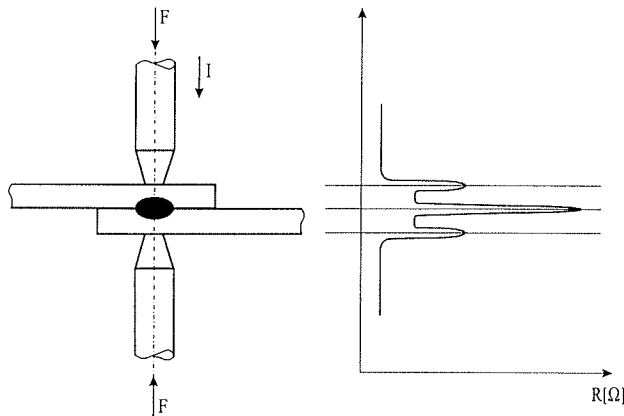


Fig. 1. Schematic representation of resistance spot welding.

In general, the welding current can be set by setting the number of windings and the conducting period of the thyristors at the primary side of the transformer. Presetting thus provides an alternating welding current in the secondary circuit which is a function of the impedance of the entire circuit. The alternating welding current generates an alternating magnetic field. Consequently, voltage is in-

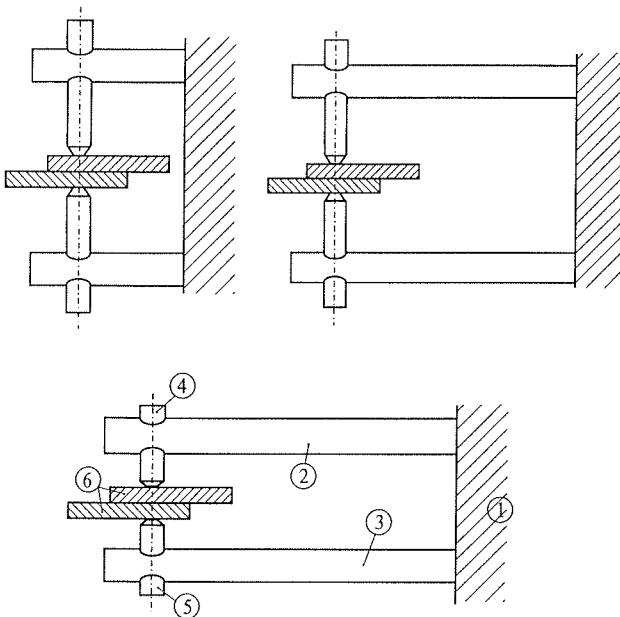


Fig. 2. Different throat areas in resistance spot welding machines. 1 - resistance spot welding machine, 2 - upper electrode holder, 3 - lower electrode holder, 4 - upper electrode, 5 - lower electrode, 6 - workpieces.

duced in the workpieces which are mainly made of ferromagnetic materials, e.g. steel, and are positioned in the throat area. The induced voltage generates eddy currents in the workpieces. The latter again generate an alternating magnetic field which, however, interferes with the alternating welding current in the secondary circuit of the welding machine.

Figure 2 schematically shows three throat areas of different shapes in resistance spot welding. In practice a number of different sizes and designs of resistance spot welding machines are used. But the throat area and shape are affected only by the size and shape of the workpieces. It is important, however, that the welding parameters are changed if the product or the workpiece shape is changed. The properties of the material to be welded have to be taken into account as well.

Figure 3 schematically shows fully obstructed and unobstructed throat areas. Two cross sections of the electrode holders shown in Fig. 4 are indicated too.

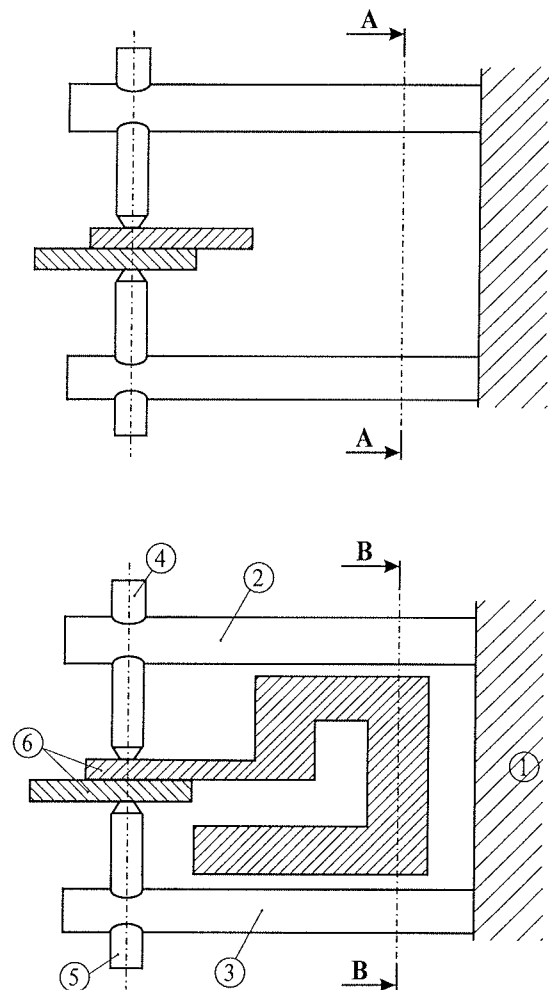
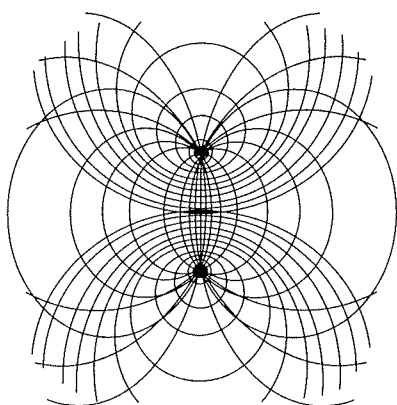


Fig. 3. Unobstructed throat area and throat area obstructed by a workpiece. 1 - resistance spot welding machine, 2 - upper electrode holder, 3 - lower electrode holder, 4 - upper electrode, 5 - lower electrode, 6 - workpieces.

Magnetic fields (Fig. 4) are shown for an unobstructed throat area and for the one obstructed by the workpiece. A change of permeability due to the presence of metal ferro-magnetic material produces a change in the magnetic field intensity in the throat area. A non-uniformly distributed magnetic field is obtained which produces very complex physical processes in the entire secondary circuit of the welding machine. The alternating magnetic flux produces eddy currents in the workpiece. The latter will heat the workpiece and produce an additional magnetic field which will hinder the flow of the welding current in the secondary circuit.

#### SECTION A-A (fig. 3)



#### SECTION B-B (fig. 3)

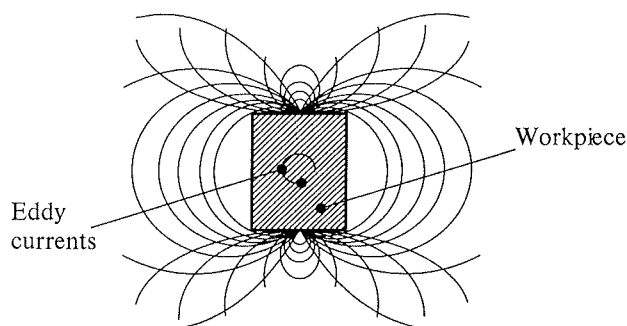


Fig. 4. Influence of workpiece in throat area on distribution of magnetic field intensity.

### 3. Experimental work

For the experiments a common industrial resistance spot welding machine was used. Welding was accomplished with a 50 Hz alternating current, different lengths of welding cables at the secondary side of the machine and different throat areas. A cross section of all the cables used was equal to 150 mm<sup>2</sup>. The aim of the investigation was to find out the influence of the cable length on the welding parameters and the welding process itself.

In the experiments, the cable length, the welding current (using different conducting periods of thyristors) and the material to be welded were varied. Figure 5 schematically shows a unit with measuring instruments to measure the

welding current and a voltage drop at the primary and secondary sides of the welding transformer. To measure the current and voltage at the primary side, common measuring instruments showing high accuracy and reliability were used. The primary current was measured with a current probe with a measuring range of 0.3 A to 700 A. As an alternating current of high intensity was flowing at the secondary side, a Rogovsky coil was used. The Rogovsky coil consisted of 3300 turns in two layers which surrounded an alternating magnetic field generated due to the flow of the alternating welding current. The welding current intensity was obtained by measurement of the voltage induced in the Rogovsky coil, the knowledge of the voltage ratio of the transformer, and mathematical integration.

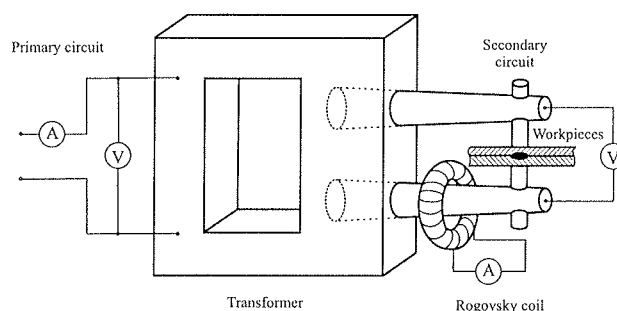


Fig. 5. Schematic of resistance spot welding machine with a measuring chain for measurement of welding parameters.

Because of the above-mentioned phenomena, ohmic and induced resistances occurred in the cables during welding, which hindered the current flow. Ohmic resistance gave rise to heating of the cables. Consequently, in the whole circuit, a lower welding current was carried. Inductive resistance produced a lag between the current and the voltage, which produced a reactive power. In resistance spot welding, this represents a pure loss.

### 4. Analysis of results

A change of the cable length entails a change of the welding parameters since in the secondary circuit ohmic and inductive resistances increase, which affects the welding current. Figure 6 shows the welding current (secondary side of the transformer) as a function of the cable length and the conducting period of the thyristors. The materials welded were steel and aluminium sheets of various thicknesses. The majority of the experiments to study the influence of the throat area, i.e., the cable length, on the welding parameters were performed using a 3 mm thick sheet. The optimum welding parameters for different cable lengths were studied using a 1 mm thick sheet. In all cases two sheets were welded together in a lap joint.

Figure 6 indicates, which is quite understandable, that an increase in the conducting period, the welding current increases too. It is, however, less understandable that long-

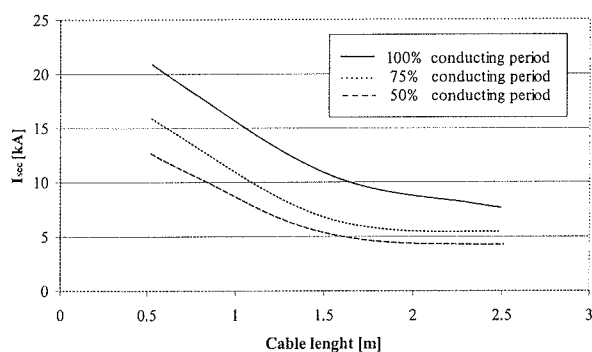


Fig. 6. Influence of cable length and conducting period of thyristors on welding current at the secondary side in resistance spot welding of steel sheet with a thickness of 2 x 3 mm.

er welding cables strongly reduce the welding current. For example, when the cable length is increased from 0.5 m to 1.5 m, the welding current drops by a half of its previous value. When the cable length is increased from 1.5 m to 2.5 m, the current drop will be, however, much smaller. This can be explained by several physical principles. It is certain that the cables and their vicinity become saturated with the magnetic field. Thus with longer cables, their influence is relatively smaller.

Figure 7 shows the influence of the cable length and the conducting period of the thyristors on the welding current at the primary side of the transformer. It is evident that with an increase of the cable length from 0.5 m to 1.5 m, the welding current at the primary side drops in an almost linear manner. When the cable length is increased from 1.5 m to 2.5 m, however, this drop is by only 30 %. This indicates that the actual losses at the secondary side could be calculated from the welding-current drops at the secondary side and primary sides.

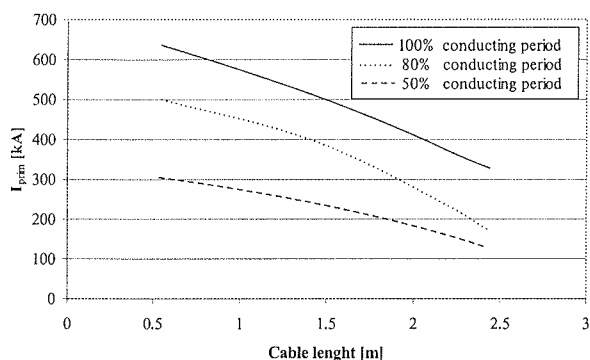


Fig. 7. Influence of cable length and conducting period of thyristors on welding current at the primary side in resistance spot welding of steel sheet with a thickness of 2 x 3 mm.

Based on the experimentally obtained results, the optimum welding current was determined with the optimum weld time for steel sheets of 1 mm in thickness (1 mm + 1 mm),

and for different cable lengths. Figure 8 shows three curves for three cables of different length. The curves plotted make it possible to determine the optimum conducting period of the thyristor, i.e., the optimum welding current, and the optimum weld time.

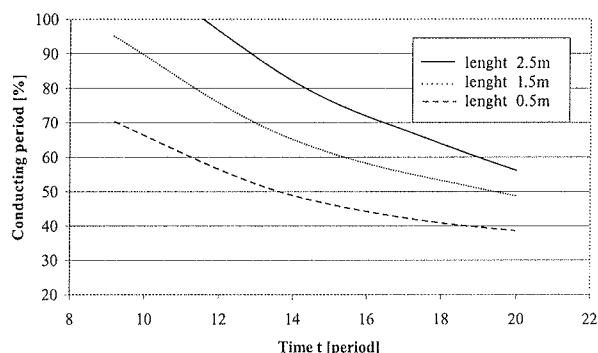


Fig. 8. Optimum welding current and optimum welding time with cable lengths of 0.5 m, 1.5 m, and 2.5 m in resistance spot welding of 1 mm thick steel sheet (1 mm + 1 mm).

Similar diagrams were plotted for other steel sheet thicknesses and an aluminium sheet. The kind of material used has a strong influence on the curve slope since different materials show different electric resistance and different contact resistance, i.e., the material properties having the strongest influence on the welding parameters. Workpiece thickness as well has a strong influence on the optimum welding current and weld time, yet no so strong as the kind of material, i.e., its ohmic and contact resistances.

## 5. Conclusions

The experimental work performed and the measurements of the welding parameters using different throat areas, i.e., different cable lengths, in resistance spot welding make it possible to draw the following conclusions:

1. the throat area, i.e., the cable length, has an important influence on the welding parameters, particularly the welding current;
2. longer welding cables in welding of large-size workpieces increase ohmic and inductive resistances in the circuit at the secondary side of the welding transformer;
3. higher ohmic resistance produces welding-current drop;
4. with the inductive resistance the reactive power occurs instead of the useful power, which is a pure loss in resistance welding;
5. with longer welding cables, it is necessary to determine the optimum welding current and the optimum weld time for each sheet thickness and for each material separately;

6. in addition to the cable length, the optimum welding parameters are affected by the kind of material welded;
7. the total or partial obstruction of the throat gap by the workpiece affects the welding parameters; this influence, however, is relatively small and should be taken into account only with very exacting structures.

*izr. prof. dr. Janez Tušek, univ. dipl. inž.*

*Miro Uran, univ. dipl. inž.*

*Institut za varilstvo*

*Ptujska 19, 1000 Ljubljana, Slovenija*

*tel: +386 01 436 77 00, fax: +386 01 436 72 22*

*e-mail: (janez.tusek, miro.uran)@guest.arnes.si*

## 6. References

- /1/ A. Wunderlin. *Elektrotechnik für die Schweißpraxis*. Expert-Verlag, Sindelfingen, 1987.
- /2/ R. Killing, R. Schäfermolte. *Elektrotechnische Grundlage der Schweißtechnik*. DVS-Verlag, Düsseldorf, 1985.
- /3/ V. Kralj, Z. Kordić, A. Köveš. *Točkovno uporovno varjenje*. Institut za varilstvo, Ljubljana, 1991.
- /4/ Z. Kordić. *Elektrotoprovno zavarivanje*. Društvo za tehniku zavarivanja Hrvatske, Zagreb, 1987.
- /5/ N.N. *Taschenbuch DVS-Merkblätter Widerstandsschweißtechnik*, 3. DVS-Verlag, Düsseldorf, 1988.

*Miran Vovk, dipl. inž.*

*Fakulteta za strojništvo*

*Aškerčeva 6, 1000 Ljubljana, Slovenia*

*tel.: +386 01 477 12 00, fax: +386 01 25 18 567*

*Prispelo (Arrived): 18.02.2002*

*Sprejeto (Accepted): 28.06.2002*

# HARDWARE IMPLEMENTATION OF LANGUAGE RESOURCES FOR EMBEDDED SYSTEMS

Matej Rojc, Zdravko Kačič, Iztok Kramberger

Institute of Electronics, Faculty of Electrical Engineering and Computer Science,  
Maribor, Slovenia

**Key words:** finite-state machines, finite-state transducers, phonetic and morphological lexicons, spoken dialogue applications, Atmel flash memory, Atmel microcontroller

**Abstract:** A lot of external natural language resources are used in spoken dialogue systems. These resources present considerable problems because of the needed space and slow lookup-time. It is, therefore, very important that the presentation of external language resources is time and space efficient. It is also very important that new language resources are easily incorporated into the system, without modifying the common algorithms developed for multiple languages. This paper presents the method and results of compiling the large Slovenian phonetic and morphology lexicons (Slflex and Slmlex) into corresponding finite-state transducers (FSTs). Representation of large lexicons using finite-state transducers is mainly motivated by considerations of space and time efficiency. In addition the approach of hardware implementation for both large (Slovenian) lexicons is described. We will demonstrate that the structure of the FST is very appropriate for storing in the Atmel AT49BV161 flash memory chip and the lookup algorithm for obtaining any desired information from the FST structure can be efficiently implemented using the Atmel AT90S8515 microcontroller. The described hardware implementation of both Slovenian lexicons can be connected directly to the PC using RS232 above all for development and test purposes and can be used especially in embedded systems which use speech technology.

## Strojna implementacija jezikovnih virov za uporabo v vdelanih sistemih

**Ključne besede:** končni stroji, končni pretvorniki, fonetični in morfološki leksikoni, Atmel flash pomnilnik, Atmel mikrokrmilnik

**Izvleček:** V govornih sistemih dialoga se uporablja mnogo jezikovnih virov. Uporaba obsežnih jezikovnih virov predstavlja velik problem tako zaradi porabljenega pomnilniškega prostora, kot tudi zaradi počasnega dostopanja do željene informacije. Zato je zelo pomembno, da je predstavitev uporabljenih virov časovno in pomnilniško optimalna. Pri večjezičnih sistemih je pomembna tudi preprosta vključitev jezikovnih virov drugih jezikov v sam sistem, ne da bi bilo pri tem potrebno spreminjati skupne algoritme razvite za več jezikov. V članku predstavljamo metodo in rezultate prevajanja obsežnega slovenskega fonetičnega in morfološkega leksikona (Slflex in Slmlex) v pripadajoča končna pretvornika (FST). Takšna predstavitev je izredno učinkovita tako glede porabe pomnilniškega prostora, kot tudi časa, potrebnega za dostop do informacije. Podrobneje bo predstavljen tudi pristop strojne implementacije obeh Slovenskih leksikonov. Struktura končnih pretvornikov je zelo primerna za njihov zapis v "flash" pomnilniško integrirano vezje (Atmel AT49BV161), algoritem pridobivanja informacije iz strukture končnega pretvornika pa lahko enostavno implementiramo z uporabo mikrokrmilnika (Atmel AT90S8515). Opisano strojno implementacijo obeh slovenskih leksikonov lahko priklopimo direktno na PC računalnik preko RS-232 serijskih vrat, predvsem za razvojne namene in testiranje. Posebej zanimiva pa je uporaba predstavitve leksikonov s končnimi stroji in njihove v članku predlagane strojne implementacije. Oba leksikona, predstavljena s končnimi stroji, lahko učinkovito uporabimo v poljubnih vdelanih sistemih, ki uporabljajo govorno tehnologijo.

### 1. Introduction

When using voice, at least speech recognition and text-to-speech synthesis technology should be integrated as significant constituent part of an embedded mobility suite in order to operate most of the common PDA (Personal digital assistance) functions such as e-mail, tasks, calendar, phone numbers and addresses. Both should allow natural way of communication using such devices. On the other hand the development of real models of human language that support research and technology development in language related fields requires a lot of linguistic data - lexicons containing thousands of words. In order to achieve the same language coverage, as in the case of e.g. the English language, such lexicons need to be up to ten times larger in the case of inflectional languages. Having a lot of Slovenian root forms can result in up-to 200 different inflectional forms. The use of such resources can, therefore, represent substantial computational load especially for embedded mobility systems, demanding low energy consumption and the smallest possible implementation. A

given spoken language system, which uses fully inflected word forms, performs much worse with highly inflected languages (e.g. Slovenian) than with non or purely inflected languages (e.g. English), where the lexicons used can be much smaller. In general, external language resources (phonetic, morphology lexicons etc.) present a problem regarding memory usage and the time spent on lookup processes.

Finite-state machines are already used in many areas of natural language processing. Their use from the computational point of view is mainly motivated by considerations of space and time efficiency. Linguistically, finite-state machines allow an easier description of most of the relevant local phenomena in the language /1/. They also provide compact representation of the specific external language resources needed for knowledge representation in the automatic text-to-speech synthesis systems. These features of finite-state machines are of major importance especially, when dealing with spoken dialogue systems.



In the following sections an approach for compiling such lexicons into finite-state transducers is first presented that represent their time and space optimal representation. The effect of using finite-state transducers for the representation of external natural language resources means a greater reduction in the memory usage required by the lexicons, and an optimal access time (required for obtaining information) which is independent of the lexicons' sizes. In the following sections the whole compilation process into finite-state transducers is presented plus the results obtained for the described Slovenian lexicons (Slflex and Slmlex). The lexicon representation appropriate for hardware implementation is then discussed in more detail. In conclusion the hardware implementation is presented of both lexicons (FST's) for use in embedded system.

## 2. Finite-state automata and finite-state transducers

### 2.1. Finite-state automata (FSA)

Finite-state automata (FSA) /2/ can be seen simply as an oriented graph with labels on each arc. Their fundamental theoretical properties make FSAs very flexible, powerful and efficient. FSAs can be seen as defining a class of graphs and also as defining languages.

#### 2.1.1. Definition

A finite-state automaton  $A$  is a 5-tuple  $(\Sigma, Q, i, F, E)$  where  $\Sigma$  is a finite set called the alphabet,  $Q$  is a finite set of states,  $i \in Q$  is the initial state,  $F \subseteq Q$  is the set of final states, and  $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$  is the set of edges.

FSAs have been shown to be closed under union, Kleen star, concatenation, intersection and complementation, thus allowing for natural and flexible descriptions. In addition to their flexibility due to their closure properties, FSAs can also be turned into canonical forms that allow for optimal time and space efficiency /3/.

### 2.2. Finite-state transducer (FST)

FSTs can be interpreted as defining a class of graphs, a class of relations on strings, or a class of transductions on strings /1/. On the first interpretation, an FST can be seen as an FSA, in which each arc is labeled by a pair of symbols rather than by a single symbol.

#### Definition

A finite-state transducer  $T$  is a 6-tuple  $(\Sigma_1, \Sigma_2, Q, i, F, E)$  such that:

- $\Sigma_1$  is a finite alphabet, namely the input alphabet
- $\Sigma_2$  is a finite alphabet, namely the output alphabet
- $Q$  is a finite set of states
- $i \in Q$  is the initial state
- $F \subseteq Q$  is the set of final states
- $E \subseteq Q \times \Sigma_1^* \times \Sigma_2^* \times Q$  is the set of edges

As with FSAs, FSTs are also powerful because of the various closure and algorithmic properties.

## 3. Use of FSMs for time and space optimal Lexicon representation

In general, when representing lexicons by automata, many entries share the same codes (strings, representing some piece of information). The number of codes is then small compared to the number of entries. Newly developed lexicons are more and more accurate and the number of codes can increase considerably. The increase in number of codes also increases the smallest possible size of such lexicons. During the construction of the automaton one needs to distinguish different codes, therefore space required for an efficient hashing of the codes can also become costly. Available lexicons that were used in this experiment suggest that the representation by automata would be less appropriate. Since morphological and phonetic lexicons can be viewed as a list of pairs of strings, their representation using finite-state transducers seems to be very appropriate. Representation of lexicons using finite-state transducers on the other hand also provides reverse look-up capability.

The methods used in the compilation of large scale lexicons into finite-state transducers (FST) assume that the lexicons are given as large list of strings and not as a set of rules as considered by Kaplan and Kay for instance /1/. In Fig. 1 some items from Slovenian phonetic (Slflex) and morphology lexicon (Slmlex) are shown. Both lexicons were compiled into corresponding finite-state transducers, using proprietary toolkit *fsmHAL*. It consists of a large set of various algorithms and tools for FSM manipulation and is written in C++ program language. During the compilation process the following algorithms were used: *union*, *determinization*, and *minimization* (Aho, 1974; Watson, 1995), (Mohri, 1995).

mod/el	model
m O - d /e: l	mod/el.N:cmsn:cmsa
mod/ela	modela
m O - d /e: - l a	mod/el.N:cmsg:cmdn:cmda
mod/elu	modelu
m O - d /e: - l u	mod/el.N:cmsd:cmsl
mod/elom	modelom
m O - d /e: - l O m	mod/el.N:cmsi:cmpd
mod/eloma	modeloma
m O - d /e: - l O - m a	mod/el.N:cmdd:cmdi
mod/elih	modelih
m O - d /e: - l i x	mod/el.N:cmdl:cmpl
mod/eli	modeli
m O - d /e: - l i	mod/el.N:cmpn:cmpi
mod/ele	modele
m O - d /e: - l E	mod/el.N:cmpa
a)	b)

Figure 1: Slovenian phonetic (a) and morphology lexicons (b). Slovenian morphology lexicon (Slmlex) is coded according to Sampa /5/ and Multext specifications /6/.



Figure 2: Part of Slovenian phonetic lexicon (Siflex) represented as FST.

The representation using finite-state transducers was performed for the Siflex and SImlex Slovenian lexicons. The starting size for Siflex was 1.8 MB (60.000 items) and 1.4 MB for SImlex (40.000 items). The final size achieved using the presented algorithms was 352 kB for Siflex and 662 kB for SImlex (Table 1) [4]. Representation of large lexicons using finite-state transducers is mainly motivated by considerations of space and time efficiency. For both lexicons a great reduction in size and optimal access time was achieved. Using such representation the look-up time is optimal, since it depends only on the length of the input word and not on the size of the lexicon.

	FST <sub>1</sub>	FST <sub>2</sub>
Number of states	69.498	90.613
Number of transitions	90.801	130.839
Size of bin file	252 kB	662 kB

Table 1: The final finite-state transducers representing Slovenian phonetic (FST<sub>1</sub>) (60.000 items) and Slovenian morphology lexicon (FST<sub>2</sub>) (40.000 items).

#### 4. Lexicons FST byte representation

Finite-state transducers representing lexicons are actually finite-state automata that have transitions labeled with two symbols. One of the symbols represents input, the other output. Therefore they translate strings. Since FST's of large-scale lexicons can be quite huge (lots of states and transitions) their implementation is not trivial. It is very important to use 'every bit' in their binary representation. The information in the final FST binary file is organized into sequences of 6 bytes. Every such byte sequence describes information of the one state transition (Fig. 3).

The information representing one transition is coded using 6 bytes. The choice depends on the maximum value of a particular data type and final FST size of the lexicons. The FST input and output alphabets for Slovenian lexicons (Siflex and SImlex) (ortographic characters, SAMPA phonetic symbols, some punctuation symbols) can be coded using eight bits (one octet) (first byte for input alphabet symbol and the second byte for output alphabet symbol). The third byte serves for flags (final bit, stop bit, next bit) and other three bytes are used for calculation of the next state

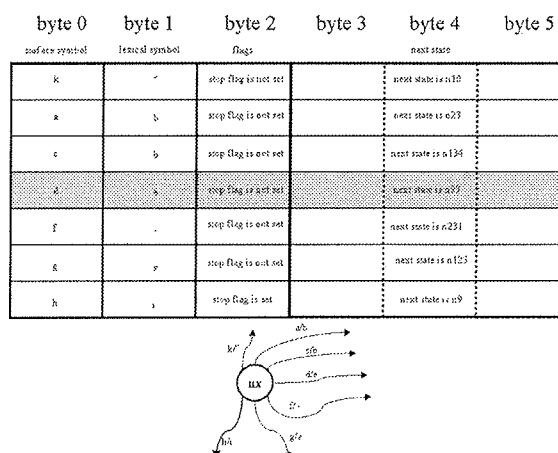


Figure 3: Lexicons FST byte representation.

(jump to the next 6-byte sequence). Using such approach, states are only indirectly marked and are actually defined with transition sequences. All the transitions not having set the stop bit belong to the same state. Next state transition start from the byte sequence, that has a stop bit set.

#### 5. Hardware implementation of the FST Lexicons

The described FST representation of the lexicons is very appropriate also for implementation using the flash memory chip (Atmel AT49BV161T). AT49BV161T is a 16-mega-bit (1Mx16/2Mx8) 3 Volt Flash Memory and is organised as 1.048.576 words of 16 bits each, or 2.097.152 bytes of 8 bits each. The x16 data appears on I/O0-I/15, the x8 data appears on I/O0-I/O7. This device can be read or reprogrammed using a single 2.65V power supply, making it ideally for in-system programming.

In the AT49BV/LV161(T) configuration, the BYTE pin controls wheather the device data I/O pins operate in the byte or word configuration. In our approach the BYTE pin is set

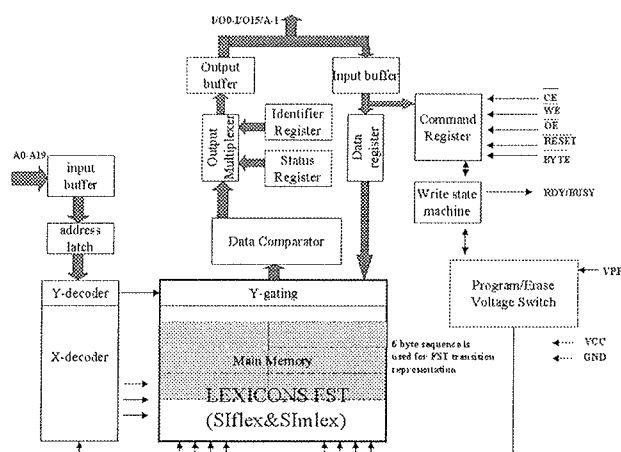


Figure 4: Lexicons FST byte representation in flash memory chip AT49BV161T.

at logic "0", and the device is in byte configuration. The data I/O pins I/O0-I/O7 are active and controlled by CE and OE. The data I/O pins I/O8-I/O14 are tri-stated, and the I/O15 pin is used as an input for the LSB (A-1) address function. All together with other address pins A0-A19, we are able to address ( $2^{21}$  bytes) 2097152 bytes, what is enough for Slovenian FST lexicons (Fig. 4).

## 6. Using FST representing both lexicons

As transducers translate a string into another string (string-to-string transducers), the lookup algorithm is straightforward - it consists of following labels from one level in a lexicon transducer.

```

procedure lookup (state, word, index, output)
  if (state  $\in$  F)
    print(output)
    for each a  $\in \Sigma \cup \{\epsilon\}$  such that  $\exists_{t \in Q} \delta(\text{state}, a) = t$ 
      lookup(t, word, index+1, output . a)
    for each a  $\in \Sigma \cup \{\epsilon\}$  such that  $\exists_{t \in Q} \delta(\text{state}, \text{word}[\text{index}], a) = t$ 
      lookup(t, word, index+1, output . a)

```

The dot operator represents concatenation. The result of concatenating string with an empty string is the same string: word .  $\epsilon$  = word. The use of the empty string in transition labels is necessary, as the lengths of the strings may not match. It is also useful for the alignment of segments of strings that represent the same features. Such alignment may reduce the size of the transducer. The presented algorithm was implemented on the Atmel Microcontroller AT90S8515. Comparison of input symbols of the FST transitions with word characters and calculation of the next state (next byte position) is performed using bit operations (Fig. 5).

## 7. Hardware implementation of the lookup algorithm

Atmel 8-bit microcontroller with 8kB downloadable flash memory was used for the implementation of the lookup algorithm. This microcontroller provides a highly flexible and cost effective solution to many embedded control applications. Raw Instruction Set architecture of used microcontroller features execution of powerful instructions in a single cycle that achieves enough throughput for high performance functionality [8]. The standard asynchronous serial interface UART of the microcontroller is mainly used for the testing purposes and it's not intended to be used for any data transmission in real-mode functionality as the data transfer rates are too low. The presented hardware architecture features in-system programming of both program Flash memory of the microcontroller and the external Flash memory used for FST Siflex and Simlex data storage. For real-mode functionality another peripheral exten-

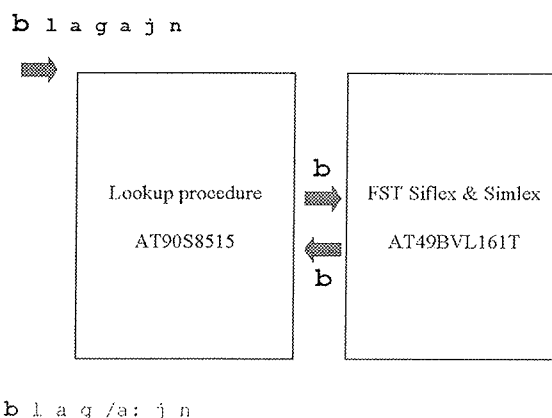


Figure 5: Lookup algorithm implementation using microcontroller AT90C8515.

sion of the microcontroller called serial peripheral interface SPI is used. SPI features high speed data transfers up to 5 Mbit/s that gives the hardware structure enough data bandwidth for efficient activity. In advance the SPI interface is already present and used in most common PDAs today as those are build around Intel's StrongARM SA-1110 microprocessor [7]. This appliance simplifies interconnections between FSA hardware and present or future embedded microcomputers.

Port A is 8-bit bidirectional I/O port. It is connected directly to flash memory over the data bus and is used as an input/output. Output represents word character (input alphabet), and the input comes from FST in the flash memory as output symbol of the FST transition.

Data addressing of the external Flash memory is done by the microcontroller lookup procedure. Due to the lack of microcontroller programmable pins and used address multiplexing it's suitable to partially compute the complete address pointer and update the separate address latches in-time between computations of the next lookup addresses. As an address output serves port C. Since we need 21 address lines to be able to address any byte in the flash memory ( $2.057.152$ ), the port C pins are connected to three latches 74HC573. Thus the microcontroller methodically computes the exact address pointer in three steps as each one of them updates appropriate address latch after its computation. In this manner there is no execution speed breakdown because of used addressing architecture.

Port D is used for driving flash memory chip and Port B for driving all three latches 74HC573. Port D is also 8 bit bidirectional I/O port and is connected with MAX3232 level shifter. The firmware of the microcontroller features two separate fully operational program modes. First one is intended for external flash programming and functionality testing (port B). In this mode the firmware is featuring low speed data transfers between personal computer and the FST. The firmware build-in Flash programming procedure re-

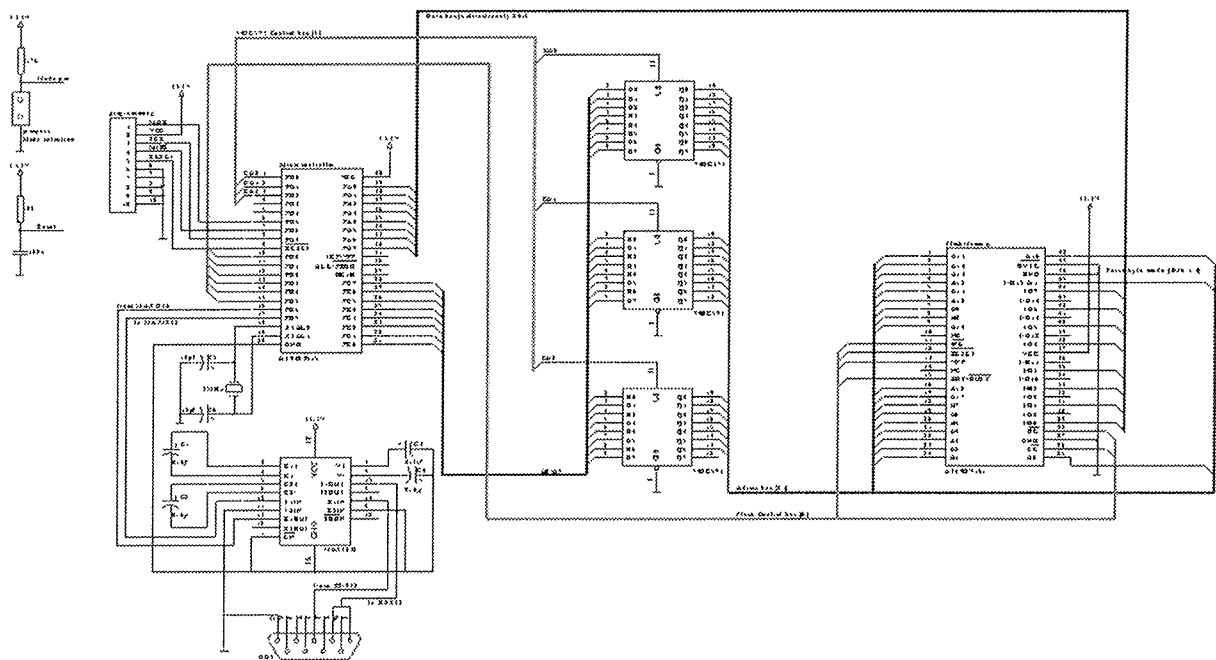


Figure 6: Hardware implementation of the system.

ceives or transmits data through asynchronous serial interface and reads or writes them in external Flash memory. In that fashion the complete FST algorithm can be tested through the personal computer. In real-mode functionality of FST the second part of the firmware is executed. In this part of the firmware the high speed serial peripheral interface is used for data transmissions. Choosing between those two modes is done simply with defining the logical state on the MODE pin of the microcontroller. As startup or as a reset condition has been applied to the microcontroller, the firmware checks the logical state on this pin and executes the appropriate functionality mode. In appliance with existing PDAs there is no need to implement the testing functionality of FST. The whole hardware implementation of the system is shown in Fig. 6.

## 8. Conclusion

Being able to operate most of the common PDA (Personal Digital Assistance) functions such as e-mail, voice, calendar, phone numbers and addresses, by using voice, the speech recognition and text-to-speech synthesis technology must be integrated into any embedded mobility suite. Development of real models of human language requires a lot of linguistic data. Finite-state machines were used for Slovenian large-scale lexicons, since they are time and space optimal solution. The effect of using finite-state transducers is great reduction of the memory usage required and the optimal access time, independent from the size of the lexicons. As showed the FST structure enable easy, flexible and efficient hardware implementation that can be used in embedded systems as significant part of the speech embedded mobility devices.

## 9. References

- /1/ Mehryar Mohri., (1995) On Some Applications of Finite-State Automata Theory to Natural Language Processing, Natural Language Engineering 1, Cambridge University Press.
- /2/ Bruce William Watson, (1995) Taxonomies and Toolkits of Regular Language Algorithms, PhD Thesis, Eindhoven University of Technology and Computing Science.
- /3/ Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman, (1974) The design and analysis of computer algorithms. Addison Wesley: Reading, MA.
- /4/ Matej Rojc, Zdravko Kačič, (2001) Representation of Large Lexica Using Finite-State Transducers for the Multilingual Text-to-Speech Synthesis System, Eurospeech 2001, Scandinavia.
- /5/ SAMPA for Slovenian, (1998) <http://www.phon.ucl.ac.uk/home/sampa/slovenian.html>
- /6/ MULTTEXT project lexical specifications, (1996) <http://www.lpl.univai.fr/projects/multtext/LEX/LEX.Specifications.html>
- /7/ Intel. Intel StrongARM SA-1110 Microprocessor: Developer's Manual. Intel, June 2000.
- /8/ Atmel. 8-Bit AVR Microcontroller with 8K bytes Downloadable Flash. Atmel, 1997.

mag. Matej Rojc  
izr. prof. dr. Zdravko Kačič  
mag. Iztok Kramberger  
Inštitut za elektroniko,  
Fakulteta za elektrotehniko, računalništvo in  
informatiko, Maribor, Slovenija

Institute of Electronics,  
Faculty of Electrical Engineering and  
Computer Science,  
Smetanova 17, 2000 Maribor, Slovenia  
Tel. +386 02 220 7000, Fax. +386 02 251 1178  
e-mail: dsplab@uni-mb.si

# OPTIMAL ALGORITHM MAPPING FOR FAST SYSTOLIC ARRAY IMPLEMENTATIONS

Igor Ozimek

Institute Jožef Stefan, Ljubljana, Slovenia

**Key words:** systolic arrays, parallel algorithm mapping, microcycled dependence graph ( $\mu$ DG), optimal scheduling, loop-extraction algorithm (LEA), VLSI

**Abstract:** There are a number of algorithms which are described by a set of recursive equations of regular dependences. Examples are certain filtering algorithms. These algorithms can be efficiently mapped to the systolic array structure, which can then be implemented in VLSI technology. This paper deals with the problem of finding optimal scheduling for a given algorithm, taking into account its exact computational requirements. First we introduce microcycled Dependence Graph (DG) and, using the notion of microcycles, define the speed of its execution that has to be maximised. Then, using Reduced Dependence Graph (RDG), we express the upper bound on the computation speed as a set of inequalities defined by the loops in RDG. To find these loops, we define a Loop Extraction Algorithm (LEA). Solving the set of inequalities obtained does not conform exactly to the linear programming problem. We describe a procedure that makes it possible to use the linear programming method to find the optimal scheduling vector.

## Optimalna preslikava algoritmov za hitro izvajanje v sistoličnem polju

**Ključne besede:** sistolična polja, vzporedne preslikave algoritmov, mikrokoračni graf odvisnosti, optimalno časovno razvrščanje, algoritem odkrivanja zank, VLSI

**Izvleček:** Mnogo algoritmov lahko zapišemo kot sistem rekurzivnih enačb z regularnimi odvisnostmi. Primer so razni filtri. Take algoritme lahko učinkovito preslikamo v sistolična polja, ki jih lahko nato izvedemo v tehnologiji VLSI. Ta prispevek se ukvarja s problemom iskanja optimalnega časovnega razvrščanja računskih operacij danega algoritma z natančnim upoštevanjem njegovih računskih zahtev. Najprej vpeljemo mikrokoračni graf odvisnosti in z uporabo pojma mikrokorača določimo hitrost izvajanja, ki naj bo čim večja. Potem z uporabo reducirane grafa odvisnosti izrazimo zgornjo mejo hitrosti računanja kot sistem neenačb, ki jih določajo zanke v reduciranem grafu odvisnosti. V ta namen določimo algoritem odkrivanja zank. Dobljeni sistem neenačb ne ustreza povsem postopku reševanja z metodo linearnega programiranja. S pomočjo dodatnega postopka omogočimo uporabo te metode za določitev optimalnega vektorja izvajanja algoritma.

### 1. Introduction

Certain real-time applications, such as signal filtering and processing in a digital communication system, require the use of a special, massively parallel computing structure called the systolic array structure to achieve acceptable performance. To implement an algorithm in this way we need a mapping procedure to map the set of equations, which describe the algorithm, to a systolic array. This mapping consists of scheduling (i.e. time mapping, mapping each DG node to a particular time instant) and space mapping (mapping each DG node to a systolic array cell). The methods described in the literature [1,2,3,4,5] are best suited for simplified systems of equations that consist of one main equation, which describes the algorithm, and a number of auxiliary equations, which are used to achieve the local communication and single assignment properties.

In this paper we consider the problem of scheduling, and develop a new approach to find the optimal scheduling of complicated algorithms described by a set of equations which have to fulfil the requirement of regularity, i.e. constant dependence vectors. Our procedure takes into account the exact computational requirements of the basic arithmetic operations used, and yields an optimal schedul-

ing vector that guarantees the fastest possible computation of the algorithm. Space mapping can then be accomplished using methods known from the literature [2].

### 2. DG and microcycled DG

DG (Dependence Graph) is one of the basic tools in the systolic array mapping process. To describe it and its modification,  $\mu$ DG (*microcycled DG*), we shall take a simple example of matrix-vector multiplication:

$$\mathbf{c} = \mathbf{A}\mathbf{b} \quad (1)$$

Eq. (1) can be written as:

$$c_i = \sum_{j=1}^N a_{ij} b_j \quad (2)$$

or, recursively, as:

$$c_i = c_i + a_{ij} b_j \quad (3)$$

To be executed by a systolic array, Eq. (1) must be transformed to its equivalent recursive form in such a way that broadcasting (variable  $b$  in Eq. (3)) and multiple assignment (variable  $c$  in Eq. (3)) are eliminated:

$$\begin{aligned} b_{i+1,j} &= b_{i,j} \\ c_{i,j+1} &= c_{i,j} + a_{i,j} b_{i,j} \end{aligned} \quad (4)$$

where the first equation is used to eliminate broadcasting of variable  $b$  (for details see [2]).

## 2.1. Dependence Graph

The corresponding DG of Eq. (4) for the case of a  $3 \times 3$  matrix  $\mathbf{A}$  and  $3 \times 1$  vectors  $\mathbf{b}$  and  $\mathbf{c}$  is shown in Fig. 1. Variables  $i$  and  $j$  are indices of the algorithm and DG nodes. Each DG node represents one iteration of (repetitive) calculations needed by the algorithm. In our case an iteration consists of a multiplication ( $ab$ ), an addition ( $c+...$ ), and a shift-through (variable  $b$ ). Each edge of DG represents a dependence between individual iterations. Since there are no multiple assignments, DG does not have any loops. Since there is no variable broadcasting, dependence vectors are local. In addition, as a prerequisite, the given algorithm is regular, i.e. its execution is independent of the indices  $i$  and  $j$ . Thus, DG is regular, localised and without loops, and is as such suitable for mapping to the systolic array structure.

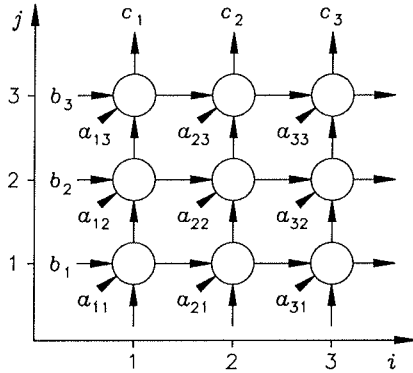


Fig. 1. DG for Eq. (4)

Scheduling for a regular DG can be represented by equitemporal lines (planes for 3-D DG or hyperplanes for multi-D DG). In Fig. 2, a simplified DG from Fig. 1 is shown together with a possible scheduling. The scheduling vector  $\mathbf{s}$  is defined as having components equal to the number of equitemporal lines between neighbouring nodes in the corresponding directions. Needless to say, these values are integers. For Fig. 2,  $\mathbf{s} = [1, 1]^T$ . By this definition, a scheduling vector is orthogonal to equitemporal lines and its size is proportional to the *slowness* of computing. Thus, the smaller  $\mathbf{s}$  becomes, the better.

The execution time index of a particular DG node can be expressed as:

$$t = \mathbf{s}^T \begin{bmatrix} i \\ j \end{bmatrix} \quad (5)$$

which is true also for the  $\mu$ DG described below.

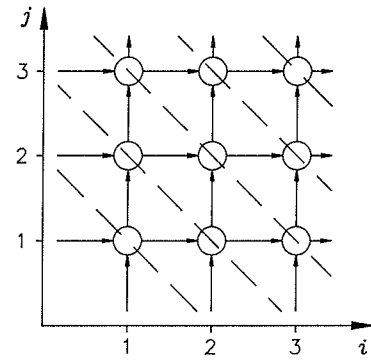


Fig. 2. Simplified DG for Eq. (4) with scheduling

## 2.2. Microcycled Dependence Graph

Scheduling in Fig. 2 does not take into account the real computation requirements within nodes. We can consider the execution within a node to be performed in microcycles and hidden from the outside world, while DG (and its corresponding systolic array) shifts data between nodes (cells) in macrocycles. These shifts can take place only after a complete (microcycled) computation within a node is finished. It can be shown that this approach is suboptimal. For our purpose it will suffice to take all three operations (multiplication, addition and shift-through) as being of equal complexity, i.e. requiring one microcycle each to execute. The main (second) equation in Eq. (4) involves a multiplication and an addition in sequence, thus requiring two microcycles. The first equation of Eq. (4) can be executed in parallel, requiring only one microcycle. There are 5 macrocycles needed for execution of DG in Fig. 2, so the total number of required microcycles is 10.

A better solution can be found using  $\mu$ DG. It eliminates the notion of macrocycles and looks at DG entirely in terms of microcycles. In Fig. 3,  $\mu$ DG is shown which corresponds to DG in Fig. 2. The black dots along the edges and within the nodes correspond to the basic arithmetic operations that can be executed within one microcycle. For the systolic array mapping, the actual execution is placed into the node to which the corresponding dependence edge is directed. Fig. 4 illustrates this by showing a node with its related operations indicated by the shaded area.

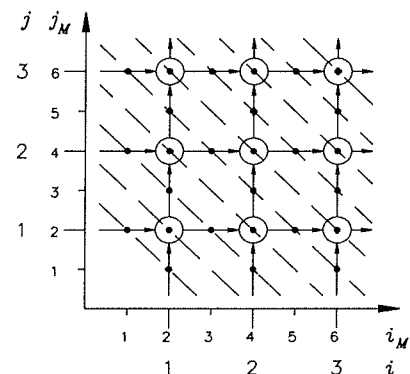


Fig. 3.  $\mu$ DG for Eq. (4)

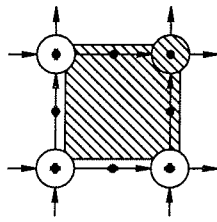


Fig. 4. A node with its related operations

There are two types of indices in Fig. 3. Indices  $i$  and  $j$  are node indices, which are the same as above, while  $i_M$  and  $j_M$  are microindices, related to our proposed microcycled scheme. The relation between them is:

$$\begin{aligned} i_M &= s_i i \\ j_M &= s_j j \end{aligned} \quad (6)$$

where  $s_i$  and  $s_j$  are the components of scheduling vector  $\mathbf{s}$ :

$$\mathbf{s} = \begin{bmatrix} s_i \\ s_j \end{bmatrix} \quad (7)$$

Using  $\mu$ DG, a better scheduling for the algorithm described by Eq. (4) can immediately be found. Since in the  $i$  direction only one cycle is needed (propagation of variable  $b$ ), scheduling can be as in Fig. 5,  $\mathbf{s} = [2, 1]^T$ , requiring 8 microcycles for complete execution.

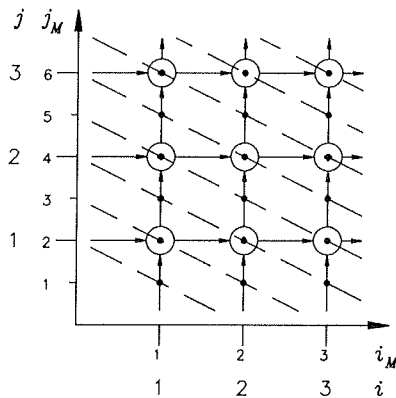


Fig. 5. A better scheduling for Eq. (4)

It can readily be shown that even the scheduling in Fig. 5 is not optimal. The optimal one is shown in Fig. 6 and is achieved by using a pipelined computation path between variables  $b$  and  $c$ . The relationship between the two variables is shown in Fig. 7.  $\mu$ DG in Fig. 6 needs a total of 5 microcycles to execute, but since the  $b$  and  $c$  variable planes are shifted relative to each other due to pipelining, 6 microcycles are actually needed, outperforming our initial solution of 10 microcycles by factor of almost 2.

In the sequel, a formal procedure will be developed for finding the optimal scheduling vector of a given algorithm.

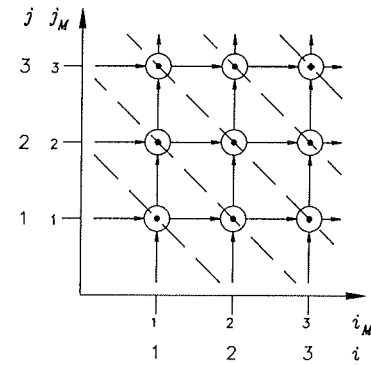
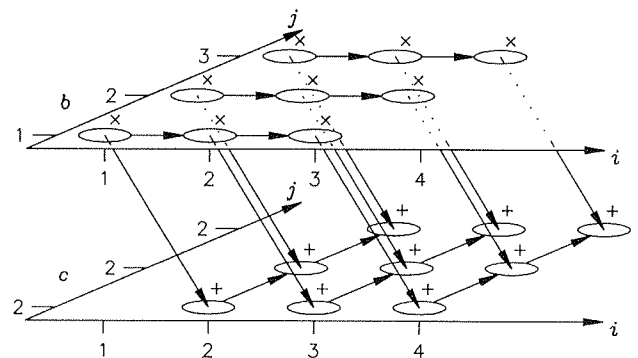


Fig. 6. The optimal scheduling for Eq. (4)


 Fig. 7. Relationship between variable  $b$  and  $c$  planes of DG

### 3. Finding the optimal scheduling

In this section we first describe RDG (*Reduced Dependence Graph*), [3], and show that the maximum speed of computation is limited by the loops in RDG. Then we propose an algorithm for automatic extraction of RDG loops. In this way we obtain a system of inequalities which define the space of possible scheduling vectors  $\mathbf{s}$ . We then use the linear programming method, with some extensions, to find the optimal scheduling vector  $\mathbf{s}$ .

#### 3.1. RDG - Reduced Dependence Graph

RDG is a graph that has a node for every variable of the given algorithm and an edge for every dependence between them. Its name (*reduced*) comes from the fact that, contrary to DG, nodes and dependences are not repeated  $n$  times but appear only once.

Two data belong to each edge  $e_k$ : the *dependence vector*  $\mathbf{d}_k$ , denoting the distance in  $\mu$ DG between the input and output variables, and *computational complexity*  $r_k$ , denoting the number of microcycles required to compute the output variable from the input variable.

RDG for our matrix-vector multiplication example is shown in Fig. 8.



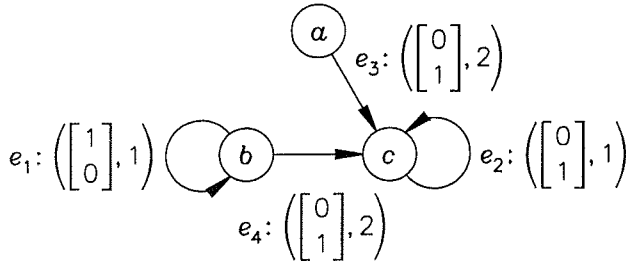


Fig. 8. RDG for Eq. (4)

The dependence vectors for RDG in Fig. 8 are:

$$\mathbf{d}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{d}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{d}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{d}_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (8)$$

and the corresponding computational complexities are:

$$r_1 = 1, \quad r_2 = 1, \quad r_3 = 2, \quad r_4 = 2 \quad (9)$$

Alternatively, we can write dependences and computational complexities in the matrix and vector forms respectively:

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad (10)$$

$$\mathbf{r} = [1 \quad 1 \quad 2 \quad 2] \quad (11)$$

### 3.2. RDG loops and the speed of computation

The fastest computation of an algorithm is defined by the loops of its RDG. A loop describes the computation of an instance of a variable on the basis of its previous instance(s). This is ultimately a sequential process, which limits the maximum speed of computation. Let us illustrate this by a simple example of the next circularly dependent algorithm:

$$\begin{aligned} a_{i,j} &= f_a(b_{i-1,j}) \\ b_{i,j} &= f_b(c_{i-1,j-1}) \\ c_{i,j} &= f_c(a_{i,j-1}) \end{aligned} \quad (12)$$

Functions  $f$  denote arbitrary arithmetic operations. The corresponding RDG is shown in Fig. 9.

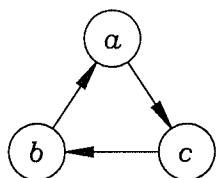


Fig. 9. RDG for Eq. (12)

Fig. 10 shows the corresponding DG, split to its three variable planes. Computation of variable  $b$  depends on one of its previous values, which is located  $[2, 2]^T$  back in DG. ("Previous" here has a somewhat special meaning, since no time is assigned to the computations at this stage of the mapping process - indices  $i$  and  $j$  are not yet related to the final space or time indices. Naturally, a result that is used further in another computation has to be computed earlier in time.) On this computation path (showed as a dashed line in plane  $b$ ) there must be enough microcycles available to perform all the necessary arithmetic operations. Since the number of available microcycles depends on the scheduling vector  $\mathbf{s}$ , this represents the lower bound on  $\mathbf{s}$  or, more exactly, on its components.

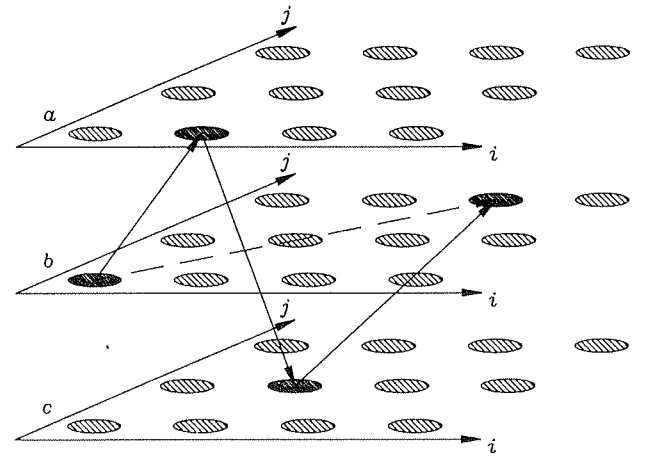


Fig. 10. DG for Eq. (12), split to variable planes

The fastest computation can be determined by finding and taking into account all the loops. For each ( $l$ -th) loop, two additional data are computed: the cumulative dependence vector,  $\mathbf{d}_{L,l}$ , and the computational complexity of the loop,  $r_{L,l}$ . They are calculated as the sum of the dependence vectors and the sum of the computational complexities, respectively, of all the edges belonging to the loop.

The number of available microcycles along the  $l$ -th loop equals  $\mathbf{s}^T \mathbf{d}_{L,l}$ . The scheduling vector  $\mathbf{s}$  must therefore satisfy the following set of inequalities:

$$\mathbf{s}^T \mathbf{d}_{L,l} \geq r_{L,l} \quad \text{for all } l \quad (13)$$

or, in the matrix form:

$$\mathbf{s}_L^T \mathbf{D}_L \geq \mathbf{r}_L \quad (14)$$

If we return to our matrix-vector multiplication example, we can easily find two loops. Their corresponding dependences and computational complexities are:

$$\mathbf{D}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (15)$$

$$\mathbf{r}_L = [1 \quad 1] \quad (16)$$

The resulting optimal scheduling vector  $\mathbf{s}$ , which satisfies (14) with equality, is:

$$\mathbf{s} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (17)$$

For our simple matrix-vector multiplication example it has been straightforward, but for a more general case neither finding the loops of a RDG nor determining the optimal scheduling vector from them is a trivial task. To illustrate its complexity it is enough to take a look at the RLSL (*Recursive Least Square Lattice*) algorithm /6/ in Table 1 and its corresponding RDG in Fig. 11. Obviously we need a more powerful approach to tackle such problems.

<b>Prediction recursions:</b>	
$\Delta_{i,j}$	$\Delta_{i,j} = \lambda \Delta_{i-1,j-1} + \frac{b_{i-1,j-1} f_{i-1,j}}{\gamma_{i-1,j-1}}$
$\Gamma_{i,j}^f$	$\Gamma_{i,j}^f = -\frac{\Delta_{i-1,j}}{\mathcal{B}_{i-1,j-1}}$
$\Gamma_{i,j}^b$	$\Gamma_{i,j}^b = -\frac{\Delta_{i-1,j}}{\mathcal{F}_{i-1,j}}$
$f_{i,j}$	$f_{i,j} = f_{i-1,j} + \Gamma_{i,j}^f b_{i-1,j-1}$
$b_{i,j}$	$b_{i,j} = b_{i-1,j-1} + \Gamma_{i,j}^b f_{i-1,j}$
$\mathcal{F}_{i,j}$	$\mathcal{F}_{i,j} = \mathcal{F}_{i-1,j} - \frac{\Delta_{i-1,j}^2}{\mathcal{B}_{i-1,j-1}}$
$\mathcal{B}_{i,j}$	$\mathcal{B}_{i,j} = \mathcal{B}_{i-1,j-1} - \frac{\Delta_{i-1,j}^2}{\mathcal{F}_{i-1,j}}$
$\gamma_{i,j-1}$	$\gamma_{i,j-1} = \gamma_{i-1,j-1} + \frac{b_{i-1,j-1}^2}{\mathcal{B}_{i-1,j-1}}$
<b>JPE recursion:</b>	
$\rho_{i,j}$	$\rho_{i,j} = \lambda \rho_{i,j-1} + \frac{b_{i,j}}{\gamma_{i,j}} e_{i,j}$
$\kappa_{i,j}$	$\kappa_{i,j} = \frac{\rho_{i,j}}{\mathcal{B}_{i,j}}$
$e_{i+1,j}$	$e_{i+1,j} = e_{i,j} - \kappa_{i,j} b_{i,j}$

Table 1. RLSL recursions

### 3.3. LEA - Loop Extraction Algorithm

To find the loops of an RDG, we propose the loop-extraction algorithm described in Table 2. With it, a number of directed trees are built. Their nodes represent the variables of the given system. Their (towards the root directed) edges represent inter-variable dependences. The following additional data belong to each edge: *the dependence vector*, denoting the distance in  $\mu$ DG between the input and output variables, and *computational complexity*, denoting the number of microcycles required to compute the

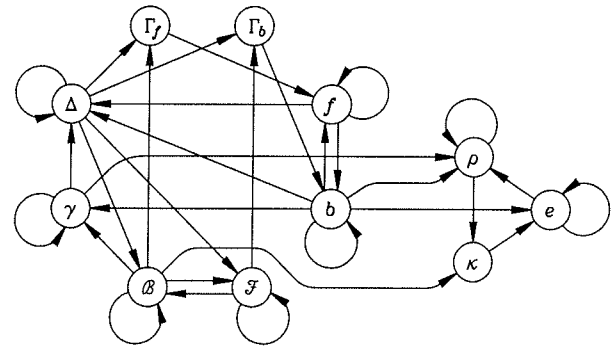


Fig. 11. RDG for RLSL (Table 1)

output variable from the input variable. Trees are built from the roots to the leaves. Each path is built until either a node is repeated or a dead end is reached. Since any repetition of a node in a path stops its growth, the maximum tree depths are equal to the number of the variables.

1. Make a list of all non-processed nodes (*all\_nodes*) and a list of all possible roots (*all\_roots*). At the beginning, both lists are equal and consist of all the variables of the given system of equations.
2. Begin with the first ( $i=1$ ) TCC (*Tightly Connected Component*, /3/).
3. For the  $i$ -th TCC, create two empty lists: a list of its roots (*roots(i)*) and a list of its loops (*loops(i)*). Move the first node from *all\_roots* to *roots(i)*.
4. From *roots(i)* take the first node (delete it from the list) and build a tree as described previously, but using only the nodes from *all\_nodes*.
5. Delete the node just taken from *roots(i)* from *all\_nodes*. This serves mainly to reduce the algorithm complexity by eliminating repetitive loop generation.
6. Each path in the tree with the ending leaf node equal to the root node represents a loop of the current ( $i$ -th) TCC. Add it to *loops(i)*.
7. Add to *roots(i)* every leaf that is not equal to the root node but is equal to some previous node on the same paths (i.e. the node is repeated) and is at the same time used in at least one loop in *loops(i)*. (These nodes have their own loops, not containing the current root node. They will be used later to build more trees for the current TCC.)
8. If *roots(i)* is not empty, go to 4.
9. The current ( $i$ -th) TCC processing has just been done. From *all\_roots* delete all the nodes that are used in any loop in *loops(i)*. If *all\_roots* is not empty, choose the next TCC (i.e.:  $i \leftarrow i+1$ ) and go to 3.
10. The processing is done. The result is a set of TCCs with corresponding sets of loops (*loops(i)*).

Table 2. LEA algorithm

The system of inequalities, Eq. (14), can be reduced by eliminating multiple dependence vectors of the same magnitude. From each set of identical dependence vectors

with different computing complexities, only the highest computational complexity is used.

An example of LEA execution for the case of the RLSL algorithm from Table 1 is shown in Fig. 12.

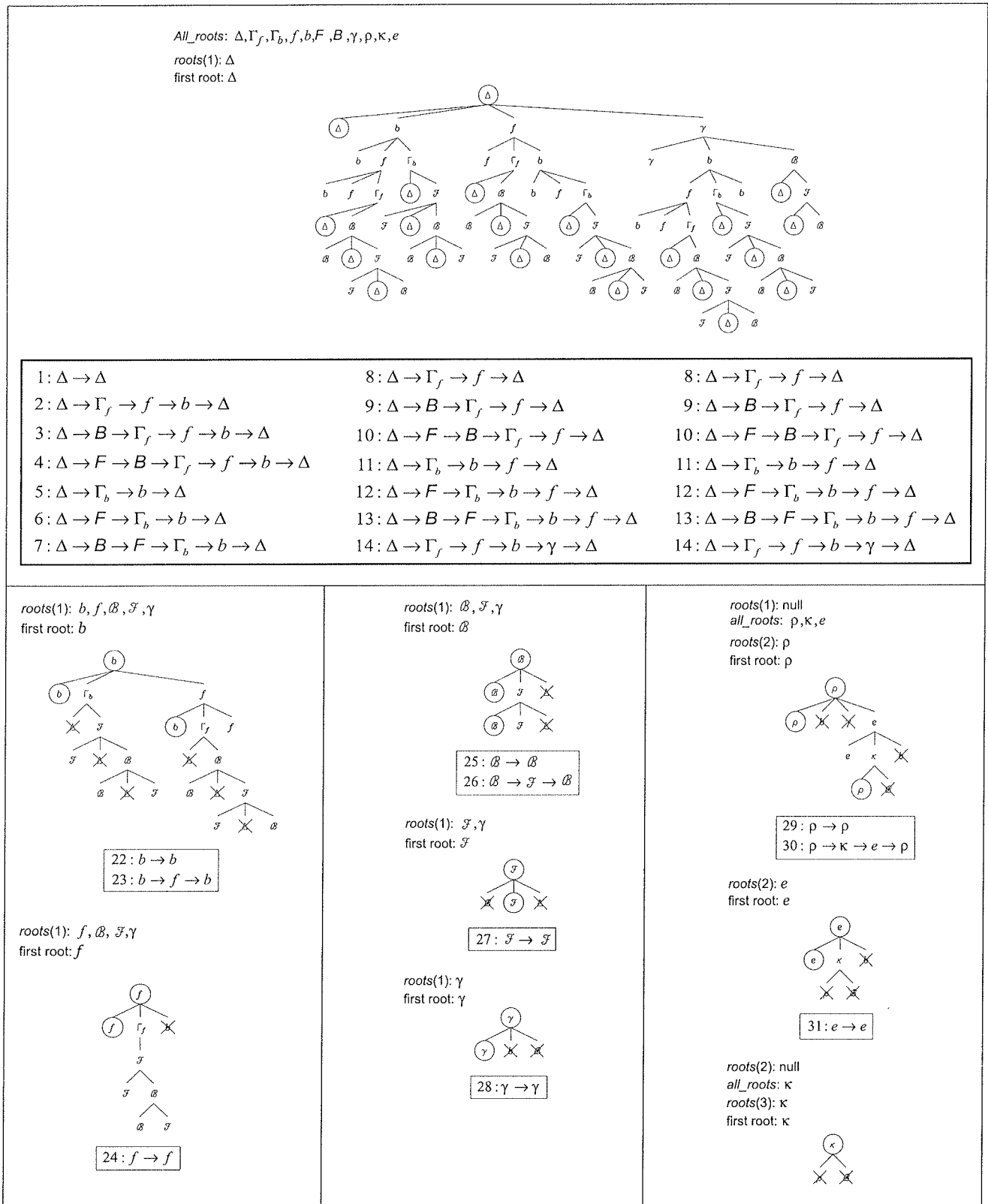


Fig. 12. LEA execution for RLSL

### 3.4. Space of possible scheduling vectors

In general, a system of inequalities like Eq. (14) need not have any solution. However, in the case of parallel algorithm mapping the corresponding DG is checked at the beginning for computability or causality, in the directions of infinite or finite dimension respectively. (For computability, no DG loops are allowed. For causality, all the edges must be oriented positively with respect to any infinite DG dimension.) Since DG is at least computable, a valid scheduling vector  $\mathbf{s}$ , and hence at least one solution of the system of inequalities, must exist. Additionally, there is no upper limit to the size of  $\mathbf{s}$  (i.e. the slowness of computing), providing that  $\mathbf{s}$  is oriented in an appropriate direction. Eq. (14) defines the space of valid scheduling vectors  $\mathcal{S}$ . This space is similar to a polygon (for 2-D DGs, polyhedron for 3-D DGs), but is unbounded in certain directions towards infinity. Fig. 13 illustrates this by showing  $\mathcal{S}$  spaces for four hypothetical algorithms with different dependences and computational complexities.

Given the space of possible scheduling vectors  $\mathcal{S}$ , the optimal scheduling vector  $\mathbf{s}$  can be found. The optimal  $\mathbf{s}$  is the one that guarantees the fastest computation of the given algorithm (the shortest or the fastest pass through its DG). For the case (a) in Fig. 13 the solution is unique,  $\mathbf{s} = [2, 3]^T$ . The solutions to the other three cases in Fig. 13 are less obvious.

### 3.5. Finding the optimal scheduling vector

With some modifications, the problem of finding the optimal scheduling vector  $\mathbf{s}$  in the space of possible scheduling vectors  $\mathcal{S}$  can be made to conform to the requirements of the linear programming method.

#### 3.5.1. Linear programming method (LPM)

The linear programming problem is defined by the following set of equations, /7/. Eq. (18) restricts the feasible region to the non-negative portion of  $R^N$ :

$$x_1 \geq 0, \dots, x_N \geq 0 \quad (18)$$

Eq. (19) is the main system of inequalities, which further bound the feasible region:

$$\begin{aligned} a_{11}x_1 + \dots + a_{1N}x_N &\geq b_1 \\ \dots \quad \dots \quad \dots \quad \dots & \\ a_{M1}x_1 + \dots + a_{MN}x_N &\geq b_M \end{aligned} \quad (19)$$

and Eq. (20) is the objective function that has to be minimised (or maximised):

$$f(x_1, \dots, x_N) = c_1x_1 + \dots + c_Nx_N \quad (20)$$

The above equations can be written in the matrix form:

$$\begin{aligned} \mathbf{x} &\geq \mathbf{0} \\ \mathbf{Ax} &\geq \mathbf{b} \\ f(\mathbf{x}) &= \mathbf{cx} \end{aligned} \quad (21)$$

#### 3.5.2. Applying LPM to the optimal scheduling problem

The optimal scheduling problem differs from the linear programming problem at two points.

The first point is that the space  $\mathcal{S}$  (defined by Eq. (13)) is not limited to the non negative portion of  $R^N$ . The problem can be solved by decomposing it to a number of subspaces, as in Fig. 14, so that none of them crosses the quadrant boundaries. LPM is then applied to each of these subspaces and the solutions are combined.

By decomposing  $R^N$  in the way described, we get  $2^N$  subspaces, not all of which are occupied by  $\mathcal{S}$ . Since the dimensionality of algorithms is usually low (typically 2 or 3) the number of subspaces is not a problem.

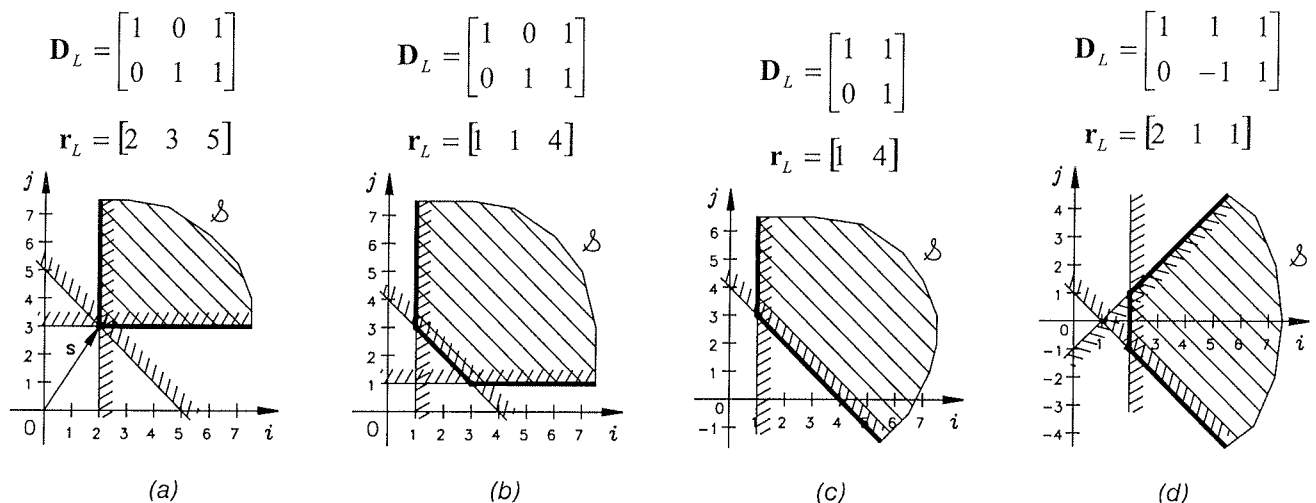


Fig. 13. Examples of 2-D  $\mathcal{S}$  spaces

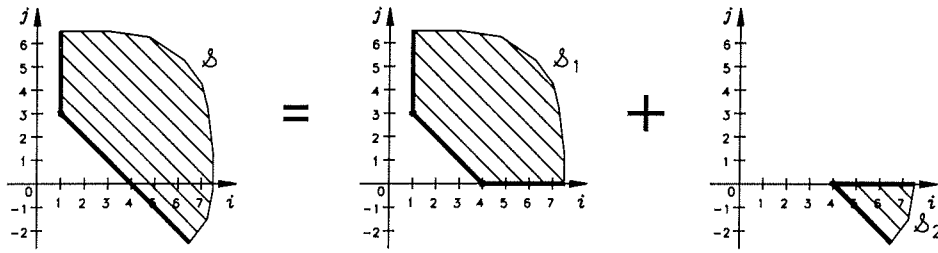


Fig. 14. Decomposition of  $\mathcal{S}$  space

As we see, the feasible region for our problem is unbounded, but the problem itself is (a minimal  $\mathbf{s}$  does exist). The reason for that is that, by definition, a possible scheduling vector exists (otherwise systolic array implementations would be impossible - which was checked for at the beginning of the mapping process), and that there certainly exists a lower bound for it (otherwise computing of the algorithm could be made arbitrarily fast with all DG nodes being computed in parallel, which is impossible due to the mutual dependences between them).

The second point of discrepancy between LPM and our scheduling problem is that the objective function for the scheduling problem, unlike Eq. (20), is not linear. For bounded DGs, the objective function can be defined as the number of cycles needed to traverse it:

$$f(s_1, \dots, s_N) = \max_{1 \leq n \leq N} (|s_n|) - 1 + \sum_{n=1}^N (l_n - 1) |s_n| \quad (22)$$

where  $s_n$  is the  $n$ -th component of the scheduling vector  $\mathbf{s}$ , and  $l_n$  is the size of DG in the  $n$ -th direction.

In the first quadrant (according to the decomposition described above)  $s_n \geq 0$ , and Eq. (22) takes the following form (for each of the quadrants the form is exactly the same, provided that the signs of variables are changed accordingly):

$$f(s_1, \dots, s_N) = \max_{1 \leq n \leq N} (s_n) - 1 + \sum_{n=1}^N (l_n - 1) s_n, \text{ for } s_n \geq 0 \quad (23)$$

where function  $\max$  is the only non-linear term.

The cycles needed for the computation to traverse a DG can be represented by equitemporal lines drawn in the DG. Fig. 15 shows this for  $3 \times 3$  DG for case (c) from Fig. 13. Equitemporal lines are shown for the scheduling vector values of (a):  $[1, 3]^T$ , (b):  $[2, 2]^T$ , (c):  $[3, 1]^T$ , (d):  $[4, 0]^T$ , and (e):  $[5, -1]^T$ . The optimal solution is (b).

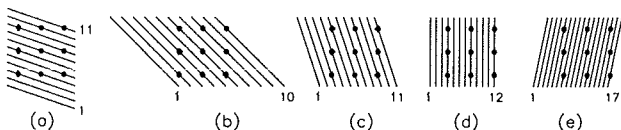


Fig. 15. Scheduling and the exact objective function Eq. (22)

By dropping the first term on the right side of Eq. (22) we obtain:

$$f(s_1, \dots, s_N) = \sum_{n=1}^N (l_n - 1) |s_n| \quad (24)$$

and Eq. (23) becomes linear, as required by LPM:

$$f(s_1, \dots, s_N) = \sum_{n=1}^N (l_n - 1) s_n, \text{ for } s_n \geq 0 \quad (25)$$

Eqs. (24) and (25) neglect those starting cycles that take place before the computation leaves the first DG node. The situation is illustrated by Fig. 16 and recapitulated in Table 3.

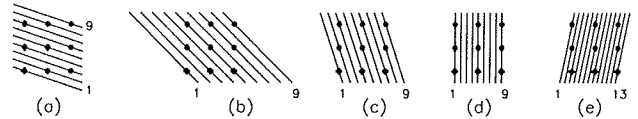


Fig. 16. Scheduling and the approximate objective function Eq. (24)

	$\mathbf{s}$	Exact cycle no.	Approx. cycle no.
(a)	$[1, 3]^T$	11	9
(b)	$[2, 2]^T$	10	9
(c)	$[3, 1]^T$	11	9
(d)	$[4, 0]^T$	12	9
(e)	$[5, 1]^T$	17	13

Table 3. Exact and approximate objective functions

The relative error of the approximate objective function can be expressed as:

$$\epsilon_r = \frac{\max_{1 \leq n \leq N} (|s_n|)}{\max_{1 \leq n \leq N} (|s_n|) + \sum_{n=1}^N (l_n - 1) |s_n|} \quad (26)$$

which decreases towards zero when DG (coefficients  $l$ ) grows towards infinity.

The objective function Eq. (22) has been defined for bounded DGs. A DG can be unbounded in a direction. This happens in the case of real time signal processing, where the input signal arrives as a continuous stream of data. The infinite dimension of DG corresponds to the time axis. For such a case the value of Eq. (22) is infinite, and the following simple objective function can be used:

$$f(s_1, \dots, s_n) = s_u \quad (27)$$

where  $s_u$  is the component of  $\mathbf{s}$  corresponding to the unbounded direction, which is presumed to be positive. This objective function is, unlike Eqs. (22) or (24), both linear and exact.

### 3.5.3. Refining the LPM solution for optimal scheduling - sub-decomposition

The solution obtained by using the approximate objective function Eq. (24) may be sufficiently good, especially for large DGs. Even for our example of the very small  $3 \times 3$  DG, the difference between the best approximate (true optimal) solution (b) and the worst approximate "optimal" solution (d) is only 20%.

If we nevertheless want to find the true optimal solution, we can do so by decomposing each subspace from section 3.5.2. in the following ways:

$$\begin{aligned} 1: & |i| \geq |j| \\ 2: & |j| \geq |i| \end{aligned}$$

for the 2-D case,

$$\begin{aligned} 1: & (|i| \geq |j|) \wedge (|i| \geq |k|) \\ 2: & (|j| \geq |i|) \wedge (|j| \geq |k|) \\ 3: & (|k| \geq |i|) \wedge (|k| \geq |j|) \end{aligned}$$

for the 3-D case, and so on.

In this way, each subspace from section 3.5.2. is further decomposed to  $N$  sub subspaces. Within each of these the exact objective function Eq. (23) becomes linear. For the first sub-subspace, Eq. (23) takes the following form:

$$f(s_1, \dots, s_N) = s_1 - 1 + \sum_{n=1}^N (l_n - 1)s_n, \text{ for } s_1 \geq s_n \geq 0 \quad (28)$$

LPM can then be applied, the solutions obtained combined together and, from them, the best one taken. Since the dimensionality of algorithms is usually small (e.g.  $N = 3$ ) the number of subspaces to be processed is not so great as to constitute a problem.

### 3.5.4. Integer programming methods

The components of scheduling vector  $\mathbf{s}$  are constrained to take only integer values. This requirement is not fulfilled by

the general LPM, so integer programming methods have to be used (e.g. branch-and-bound algorithm, cutting plane algorithm) the details of which can be found in [8] and [9]. Integer programming is an extension to LPM; if the original LPM solution does not satisfy the integer requirement, additional constraints are imposed to force an integer solution.

## 4. Conclusions

We have developed a procedure for finding the optimal scheduling for a systolic array implementation of an algorithm. The procedure has been manually tested on the RLSL algorithm from Table 1. In its RDG (Fig. 11), 31 loops have been found. The complexity of this problem is sufficiently great to make it difficult to handle without the procedures described above.

The need for sophisticated signal processing algorithms is increasing with the introduction of more and more complex communication systems. At the same time, VLSI technology is becoming capable of implementing complex computing hardware on a chip. The procedures described in this paper can be used as a tool for designing such specialised VLSI circuits.

## References

- /1/ Dan I. Moldovan, "On the Design of Algorithms for VLSI Systolic Arrays", *Proceedings of the IEEE*, Vol.71, No.1, January 1983
- /2/ S. Y. Kung, *VLSI Array Processors*, Prentice-Hall, 1988
- /3/ Sathesh K. Rao, *Regular Iterative Algorithms and Their Implementations on Processor Arrays*, Ph.D. 1986, Stanford University
- /4/ Martin D. Meyer, Dharma P. Agrawal, "Adaptive Lattice Filter Implementations on Pipelined Multiprocessor Architectures", *IEEE Transactions on Communications*, Vol. 38, No. 1, January 1990
- /5/ Michael K. Birbas, Dimitrios J. Soudris, Costas E. Goutis, "A New Method for Mapping Iterative Algorithms on Regular Arrays", *Communication, Control, and Signal Processing*, Elsevier Science Publishers B. V., 1990
- /6/ Simon Haykin, *Adaptive Filter Theory*, Prentice-Hall, 1986
- /7/ Walter J. Meyer, *Concepts of Mathematical Modelling*, McGraw-Hill Book Company, 1985
- /8/ Frank S. Budnick, *Finite Mathematics with Applications*, McGraw-Hill Book Company, 1985
- /9/ S. S. Rao, *Optimization - Theory and Applications*, Wiley Eastern Limited, 1984

Dr. Igor Ozimek  
Institut Jožef Stefan, Jamova 39, Ljubljana  
tel.: +386 1 477-3900  
fax.: +386 1 251-9385, 426-2102  
email: igor.ozimek@ijs.si

# PREDICTION OF SYMBOLIC PROSODY BREAKS WITH NEURAL NETS

Janez Stergar, Bogomir Horvat

University of Maribor, Faculty of Electrical Engineering and Computer Science,  
Maribor, Slovenia

**Key words:** symbolic prosody, data driven approach, prosodic breaks, symbolic prosodic tags, prediction, Neural Networks, Multi Layer Perceptrons, DSP

**Abstract:** In this paper the data driven prediction of word level prosody modeling for Slovenian language is presented. Automatic learning techniques depend on the construction of a large corpus labeled with appropriate labels. The labeling can be done either automatically or by hand. While automatic labeling can be less accurate than hand labeling, the latter is very time consuming and in some cases inconsistent. Therefore we will present a new semi-automatic approach for determining prosody breaks features with a graphical user interface (GUI). The GUI combines the advantage of hand labeling and automatic labeling by achieving a high consistency in labeling and reducing the time that would be needed for hand labeling. The labeled Slovenian corpus has been used to train our phrase break prediction module, using a neural network (NN) structure. We used an MLP structure suitable for Digital Signal Processor (DSP) implementation. Experiments for the data driven prediction of major/minor phrase breaks have been performed. The achieved prediction accuracy marks a good entry-level for phrase break prediction of the Slovenian language and is comparable to other approaches in phrase break prediction where more complex prediction methods were used and a much larger corpus was used for training. The achieved overall prediction accuracy is about 90 %.

## Napovedovanje simboličnih prozodičnih mej z nevronskimi mrežami

**Ključne besede:** simbolična prozodija, podatkovno vodeni pristop, prozodične meje, simbolične prozodične značnice, napovedovanje, digitalni signalni procesor, nevronske mreže, večplastni perceptroni

**Izvleček:** V članku bomo predstavili podatkovno vodeno modeliranje prozodije na nivoju besed za slovenski jezik. Samodejne tehnike učenja so odvisne od zasnove obsežne besedilne zbirke označene z ustreznimi značnicami. Označevanje lahko izvedemo samodejno ali ročno. Čeprav je samodejno označevanje ponavadi manj natančno kot ročno, predstavlja slednje časovno zelo obsežno proceduro, ki je v določenih primerih nedosledna. Predstavili bomo postopek polavtomatskega določanja prozodičnih mej z uporabo interaktivnega grafičnega vmesnika (GUI). GUI združuje prednosti ročnega s samodejnim, bolj konsistentnim označevanjem in prispeva k zmanjšanju potrebnega časa za označevanje. Označena besedilna zbirka v slovenščini je bila uporabljena pri učenju modula za napovedovanje prozodičnih mej. Modul smo zasnovali na strukturi nevronske mreže z večplastnimi perceptroni, ki je primernejša za implementacijo v digitalnih signalnih procesorjih. Izvedli smo poskuse za napovedovanje večjih/manjših prozodičnih mej. Dosežena uspešnost napovedovanja predstavlja dobro izgoščeno pri napovedovanju prozodije na nivoju besed za slovenski jezik in je primerljiva z drugimi pristopi napovedovanja prozodičnih mej, kjer so za napovedovanje uporabljene bolj kompleksne metode in strukture ter bistveno obsežnejše besedilne zbirke za učenje. Splošna uspešnost napovedovanja mej (je/ni) presega 90%.

### 1 Introduction

Automatic learning techniques offer a solution when adapting prosodic models to a new language (in a multilingual text-to-speech (TTS) system), voice or a new application. Data driven techniques allow prosodic regularities to be automatically extracted from a prosodic database of natural speech. Such techniques depend on the construction of a large corpus labelled with symbolic prosody labels.

In the first steps toward creating an inventory for the data driven approach of symbolic prosody prediction, the labelling of data can be performed by hand as well as automatically if no reference corpora is available. While automatic labelling can be less accurate than hand labelling, the latter is very time consuming.

Both goals had to be accomplished: the preparation of hand labelled corpora and in parallel the development of auto-

matic labelling techniques to somehow speed up the labelling process. Therefore a prototype of a new, interactive GUI (tool) for semi-automatic symbolic prosody labelling, which uses the segmented spoken counterpart of the text as input was developed. This tool combines the advantage of hand labelling and automatic labelling by achieving a high consistency in labelling and reduces the time needed for hand labelling.

Improvement in prosody prediction remains a challenge for producing really natural text to speech systems (TTS). As manual labelling is time and cost intensive, automatically labelled databases are preferred [6], [17].

The problem of producing good prosody models can be tackled either by using;

- linguistic expertise - adapting the models by hand or
- automatic learning techniques to adapt the models automatically by making use of large speech corpora.



The second approach offers the potential for rapid model adaptation and can to some extent be seen as language independent /2/.

Data driven approaches allow rapid adaptation to new languages and/or databases and therefore are suitable for multilingual approaches where large speech corpora are processed and models are adapted for prosody generation.

Prosodic labelling based on perceptual tests is very time consuming and usually inconsistent. People with expert phonetics and linguistic knowledge are required. In the presented approach, avoiding the necessary expert, the use of a graphic tool to minimise the required expert knowledge was proposed. Our goal was to reduce manpower, time, and expenses for prosodic labelling. The tool has a graphical interface helping the labeller (expert or novice) to consistently label symbolic phrase boundaries and, therefore, minimise the time required for labelling.

## 2 Database

To our knowledge, no prosodically labelled corpora for Slovenian language exists, that can be used for prosody research in speech synthesis. An important step during the adaptation of a TTS system to a new language is the design of a suitable database.

### 2.1 The corpus

The corpus consists of 1206 sentences in the Slovenian language (orthography) which equals app. 3 hours of speech. The selection of the text was designed to ensure good coverage of the phones in the Slovenian language, also some clauses were gathered and included from different text styles (e.g. literature and newspaper texts).

The majority of sentences in the database had between 15 and 25 words. Four different text corpora were selected and statistically analysed. The selection of sentences for the final corpus was based on a two-stage process. In the first stage an analysis based on statistical criteria was performed. In the second stage the final text was chosen based on the results of the first stage /10/.

### 2.2 Audio recordings

The audio database recordings were created in a studio environment with a male speaker reading aloud-isolated sentences in the Slovenian language. Every sentence was sampled at 44.1 kHz (16 bit).

Since the speaker was a professional radio news speaker the speech contained no disfluencies (i.e. filled pauses, repetitions and deletions) although for this particular speaker there are some indices for hesitations in the form of pauses and lengthening. Compared to the German corpus /19/ of resembling extent used in /8/, the percent-

age of hesitations differed essentially (<0,5% German, >15% Slovenian).

### 2.3 Phonetic transcription

The phonetic transcription was managed using a two step conversion module. The first step was realised with a rule-based algorithm. Subsequent to the first step the second step was designed using a data driven approach (neural networks were used).

The module was designed for the support of two approaches in grapheme-to-phoneme conversion. The first part was intended for those cases where no morphological lexica was available. The first rule based stress assignment was done, followed by a grapheme-to-phoneme conversion procedure.

The step of stress marking before grapheme-to-phoneme conversion is very important for the Slovenian language, since it very much depends on the type and place of the stress. If the phonetic lexicon is available, a data driven approach, representing the second part in the module, using neural networks can be used. Here, the phonetic lexicon was used as a data source for training the neural networks /10/.

### 2.4 Part of Speech Tags

The text corpus was hand-labelled using the following part-of-speech tags (POS) /15/:

1. SUBST for nouns,
2. VERB for verbs,
3. ADJ for adjectives,
4. ADV for adverbs,
5. NUM for ordinal and cardinal numbers,
6. PRON for pronouns (nouns, adverbs, ...),
7. PRED for predicative,
8. PREP for prepositions,
9. CONJ for conjunctions,
10. PART for particle,
11. INT for interjection,
12. IPUNC for inter punctuation and
13. EPUNC for end punctuation

All tags were combined in an environment where tracking and correcting tags were simplified for the labellers /13/.

Compared to the tag-set in the German corpus used in /8/, the tag-set for the Slovenian language is smaller. The difference in size occurs because the Slovenian corpus is hand-tagged and no reliable tagger currently exists for a large tag-set (Figure 1).

1. → Dvestodeset·STEY·centimetroy·SAM·visoki·PRID·  
Nemec·SAM·ne·PRISL·skriva·GLAG·ambicij·SAM·v·  
PREDL·ameriški·PRID·ligi·SAM·LOČ·saj·PRISL·je·  
GLAG·tik·PRID·pred·PRISL·prvenstvom·SAM·zavrnjil·  
GLAG·nekaj·ZAIM·ponudb·SAM·bogatih·PRID·  
evropskih·PRID·klubov·SAM·KLOČ

Figure 1: With POS and prosody breaks labeled clause

## 2.5 Phonetic segmentation and labelling

The spoken corpus was phonetically transcribed using HTK. Along with standard nomenclature two special markers were used for pauses between phonemes. "sil" denotes the silence before and after sentence. "sp" denotes the silence between words in the sentence. Both were determined with one state HMM and all phonemes with three state HMM respectively in the HTK environment (Figure 2).

```
#!MLF!#
"/stavek_1.lab"
0 1750000 sil
1750000 2650000 d
2650000 2950000 v
2950000 3900000 e:
3900000 5000000 s
5000000 5250000 t
5250000 5550000 O
...
92400000 92950000 k
92950000 93300000 l
93300000 94300000 u:
94300000 94550000 b
94550000 94950000 O
94950000 96500000 W
96500000 100350000 sil
```

Figure 2: An example of phonetic segmented and labeled text

## 3 Prosody breaks labelling

### 3.1 Labelling inventory

Since no inventory for symbolic prosody breaks labels is defined for the Slovenian language, we decided to use similar labels to those used in /5/ and in /7/. Thus the prosody break labels were determined through acoustic perceptual sessions and the text was labelled speaker dependent. The following inventory of prosody break labels was used for labelling the corpus /12/:

- B3 prosodic clause/phrase boundary
- B2 prosodic phrase boundary
- B9 irregular prosodic boundary, usually hesitation, lengthening and unwanted pauses and
- B0 for every other boundary.

The acoustic prosodic boundaries were determined by boundary indication, listening to audio files and visual output (pitch and energy) from our tool.

### 3.2 Semi- automatic prosody breaks labelling

A tool intended to help the labeller (novice or expert) to make decisions about prosody breaks within each sentence was designed /14/. The tool indicates possible prosody boundaries, which depend on the segmented pauses in spoken corpora and pitch accents.

Experiments on multi-lingual databases (3 languages) have shown that the strategy of segmenting the speech signal with pauses, yields a significant improvement in annotation accuracy /18/.

Syllable and word boundaries are marked by vertical lines adding overview clearness and \*B\* marks for symbolic prosody boundaries are inserted in the sentence concerned.

The tool indicates markers for prosody boundaries taking phonetic segmentation of pauses into account. The position of prosody boundaries is selected by considering the duration of silence between words. The decision of indication is made by comparison with a specific threshold. This threshold can be changed manually and can be tuned according to a specific speaker /14/.

## 4 Labelling results

Labelling experiments were performed for prosody breaks and, additionally, one experiment for accents labelling. In the first experiment prosody breaks labels were marked at positions indicated by the tool.

Additionally, a careful analysis of the f0-contours, energy contours, and perception lead to the insertion at positions that the tool did not indicate. This labelling scheme resulted in a database further referenced by DB1. In the second experiment, labels were only marked at positions indicated by the tool. This resulted in database DB2.

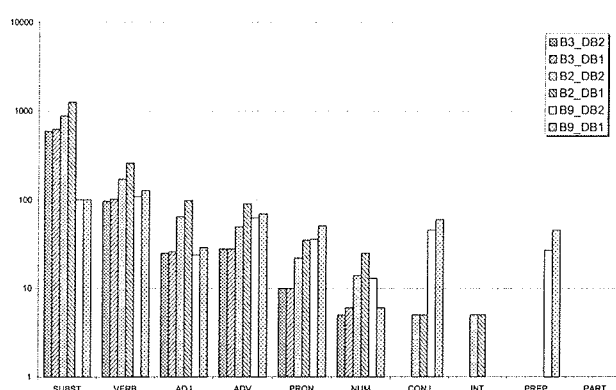


Figure 3: The occurrences of prosody break labels in DB1 and DB2

The frequencies of occurrence for each labelled break for each POS tag are presented in Figure 3. The increase of B2 tags in DB1 compared to DB2 is proportional for almost all POS tags. The increase of B9 labels is evidently minor to the increase of B2 labels and in our opinion is strongly speaker dependent.

The complete labelling of 600 clauses had now been performed. It was possible to detect 77,95% of all breaks (over 93 % for B3) and considerable shorten the time needed for labelling the database with the semiautomatic method used /13/.

## 5 Phrase break prediction module

### 5.1 Input parameters

Which parameters are relevant for symbolic prosody label prediction remains an open research question. A carefully chosen feature set can help to improve prediction accuracy, however, finding such a feature set is work intensive. In addition linguistic expert knowledge can be necessary and the feature set found can be language and task dependent. A feature set which is commonly used and which seems to be relatively independent of language and task is part-of-speech (POS) sequences /8/.

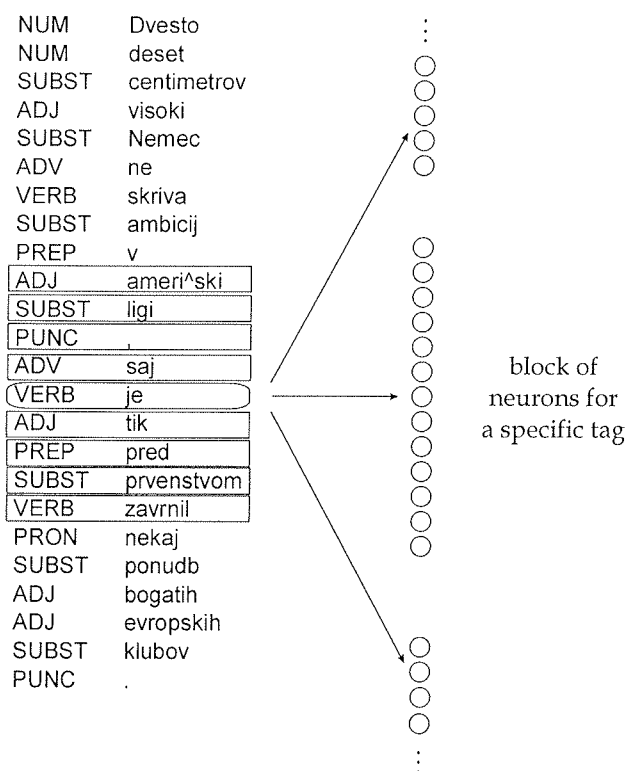


Figure 4: Input mapping into the MLP structure

POS sequences of length four to the left and right of the position in question were used. For the input of our prediction model the POS sequences were coded with a ternary logic (-1 for a non-active node, +1 for an active node, 0: not valid) /4/. Thus for each POS tag a vector was ob-

tained with a dimension of the size of the tag-set. The size of our tag set was 13. Using a POS sequence length of four to the left and right for the Slovenian language, we achieved  $m = (4+1+4) * 13 = 117$  dimensions. The dimension of our input vector as well as tag-set is similar to the English (German) language prediction tests in /8/ where a tag-set of length 14 was used (Figure 4).

### 5.2 Design of the neural network model

MLP networks are normally applied to performing supervised learning tasks, which involve iterative training methods to adjust the connection weights within the network. This is commonly formulated as a multivariate non-linear optimisation problem over a very high-dimensional space of possible weight configurations./3/. One (two) hidden layers connected to the input layer (and bias) were used and tests were performed with 30-40 neurones in each layer (Figure 5). The output layer was reduced to one (two) single outputs. The results presented (cf. Experiments and results) are for MLP structures with one hidden layer.

### 5.3 Training and pruning method

A variation of the standard back-propagation algorithm was applied to train the NN - VarioEta /11/. Patterns were selected with a quasi stochastic procedure. An approximation of the gradient was used for the determination of search direction  $\mathbf{d}_i$ , computed from the average over a subset  $M$  of all patterns:

$$\nabla E^M(w) = \frac{1}{|M|} \sum_{n \in M} \nabla E^n(w) \quad (1)$$

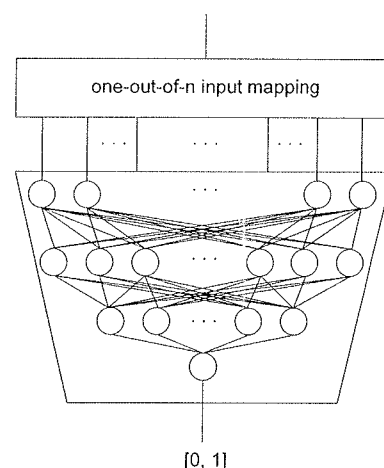


Figure 5: The MLP NN architecture

Note that  $\mathbf{d}_i$  is not normally parallel to  $-\nabla E^M(1)$  and thus VarioEta does not constitute a pure gradient method. The subset  $M$  was determined with a permutation procedure known as "drawing without replacement". During the first adaptive step,  $X$  patterns were chosen at random, during the second step  $X$  of the remaining  $|M|$  ( $|M|$  denoted the number of elements of  $M$ ), and so forth. Each pattern had

the same probability of being chosen. After  $Y$  steps, every pattern would be read in exactly once. The initial setting for the training started with a learning rate of  $\eta = 0.05$ , decreasing its value every 10 epochs. The values for  $|M|$  were varied between 50 and 250.

The network was trained so that the output error on validation data converged to a local minimum using the *late stopping strategy* /9/. A careful examination of correlation between the transformed ternary coded input patterns was performed. Nodes with high correlation rates (over 90 %) were removed from the input training inventory. Irrelevant and destructive weights were eliminated by early brain damage /16/. Node pruning based on sensitivity analysis was used to reduce the number of hidden neurones /9/.

## 6 Experiments and results

After labelling the corpus as described in the preceding section, the two databases (DB1 and DB2) were used to train the phrase break prediction module of our text-to-speech system. For both databases the B9-labels marking hesitations were removed prior to training, since hesitations generally occur at positions where a break seems unsuitable. Both databases were identically split into a training set (70% of the data), a validation set (10% of the data) used to avoid overfitting, and a generalisation set (20% of the data). All results were determined on the independent generalisation set.

Tests were made comparing the prediction results using labelled data in DB1 and DB2 as training data for our phrase break prediction module. The results were significantly better for DB2 (Table 1). This is probably due to the fact that the consistency for the labels detected by the tool is much higher than the consistency for the cases where additional breaks are labelled based on perception. It also showed that the prediction accuracy degrades only insignificantly for those cases where the minor and major breaks were grouped (minor = major) if DB2 was used to train the phrase break prediction module. This means that for the case of break vs. non-break prediction, this tool can be used for labelling without performance loss. This results in a significant reduction in time needed to label a database.

Table 1. Comparisons of phrase break prediction for DB1 and DB2

breaks	DB1	DB2
minor/major	74,05 %	79,37%
minor=major	77,74 %	80,60 %

Considering the discussed results in previous paragraphs and the fact that tests of prediction accuracy no matter what approach is used for prediction, are preferably made with hand labelled corpora /1/, /8/, for comparison reasons it was decided to conduct further experiments with the DB1.

Table 2. Results for phrase break prediction DB1

breaks	B correct	NB incorrect	overall
DB1	77,74 %	4,95 %	94,03 %

Table 3. Confusion matrix for the generalisation data

predicted/actual	breaks	non-breaks	all predicted
breaks	2008	256	2264
non-breaks	582	11176	11758
all actual	2615	11432	

In tables 2 and 3 the results are presented for the phrase break prediction module. Despite the limited labelling material available (600 labelled clauses were used) and the multi layer perceptron (MLP) NN structure used for prediction, the results accomplished are comparable to prediction accuracy for German /8/ and English /1/.

For the prediction of breaks (B correct) the results are equivalent to the achieved accuracy prediction of B correct (77,67 %) for German in /8/ or nearly equivalent to achieved accuracy prediction of B correct (79,27 %) for English in /1/ despite a much smaller inventory of clauses used. Slightly better overall and non-break (NB incorrect) prediction accuracy was achieved. In the next tables (Table 4, Table 5, Table 6, Table 7) the results for isolated major/minor phrase breaks prediction accuracy are presented.

Table 4. Results for B3 phrase break prediction DB1

breaks	B3 correct	NB3 incorrect	overall
DB1	74,05 %	0,15 %	98,36 %

Table 5. Confusion matrix for the generalisation data on B3

predicted/actual	breaks	non-breaks	all predicted
breaks	605	19	624
non-breaks	212	13211	13423
all actual	817	11432	

The results presented for achieved phrase break predictions of B3 markers are superior to the prediction for B2 markers. The reason for the difference is assumed to be the percentage of additionally hand labelled B2 markers in DB1. The first experiments using the DB2 for input data are much more promising, although the database is not covering the entire inventory of hand labelled phrase breaks in the Slovenian corpus. Therefore some additional experiments (time consuming perceptual sessions) will be necessary to evaluate the suitability of the covered prosody features.

Table 6. Results for B2 phrase break prediction

breaks	B2 correct	NB2 incorrect	Overall
DB1	66,13 %	3,71 %	92,43 %

Table 7. Confusion matrix for the generalisation data on B2

predicted/actual	breaks	non-breaks	all predicted
breaks	1189	454	1643
non-breaks	609	11795	12404
all actual	1798	12249	

## 7 Conclusion

This paper presents an approach for the labelling and classification of symbolic prosody phrase breaks for the Slovenian language. A universal tool for hand labelling the corpora with prosodic markers was designed and used for the labelling of Slovenian corpus for phrase breaks and accent prediction. This tool can be seen as a first step towards the semi-automatic labelling of prosody features for the Slovenian language. Our aim was to design a tool suitable for performing multilingual prosody labelling. Only features for prosody prediction were, therefore, used which seem to be relatively independent of language and task. Our conclusion is that the approach used - the segmented pauses in the speech corpus for phrase boundary indication - is very useful for the symbolic prosody breaks labelling of the Slovenian language. Firstly, it considerably reduces the time needed for labelling and, secondly, it provides a high level of support to a labeller for consistent labelling of prosodic events. Nevertheless the data obtained with our labelling tool seems to be much more suitable for the training of our prediction module. The NN structure used is accurate enough when compared to other, more complex, NN structures and approaches in prediction of symbolic prosody markers.

The database for the Slovenian language labelled with the proposed tool was used to train our phrase break prediction module /13/. The achieved prediction accuracy marks a good entry-level for phrase break prediction accuracy of the Slovenian language. Nevertheless a minor clause inventory was used, compared to other approaches, with equivalent or superior success in phrase break prediction accuracy.

We also conclude that the simple NN structure proposed is suitable for implementation in DSP environments for speech synthesis as an ad-on prosodic module.

## 8 References

- /1/ Black A. W., Taylor P. (1997). Assigning Phrase Breaks from Part-of-speech Sequences. Proceedings Eurospeech 97, Rhodes, Greece.
- /2/ Fackrell J. W. A., Vereecken H., Martens J.-P., Van Coile B. (1999). Multilingual Prosody Modelling using Cascades of Regression trees and Neuronal Networks. Proceedings Eurospeech 99, Budapest, Hungary.
- /3/ Gallagher M. R. Multi-Layer Perceptron Error Surfaces: Visualization, Structure and Modelling. PhD Thesis, University of Queensland, Department of Computer Science and Electrical Engineering, 2000.

- /4/ Hain H.-U. (1999). Automation of the training procedure for neural networks performing multi-lingual grapheme to phoneme conversion, Proceedings Eurospeech 99, Budapest, Hungary.
- /5/ Kompe R. (1997). Prosody in Speech Understanding Systems. Springer - Verlag Berlin Heidelberg, Lecture Notes in Artificial Intelligence.
- /6/ Malfrere F., Dutoit T. and Mertens P. (1998). Fully automatic prosody generator for text-to-speech. ICSLP 98, Sydney Australia.
- /7/ Mihelič F., Gros J., Nöth E., Dobrišek S., Žibert J. (2000). Recognition of Selected Prosodic Events in Slovenian Speech, Language Technologies, Ljubljana, Slovenia.
- /8/ Müller A. F., Zimmermann H.G., and Neuneier R. (2000). Robust Generation of Symbolic Prosody by a Neural Classifier Based on Autoassociators. Proceedings ICASSP 00, Istanbul, Turkey.
- /9/ Neuneier R., Zimmermann H.G. How to train neural networks. In Ohr G. B. and Müller K.-R., editors Neural Networks: Tricks of the Trade. Springer Verlag, Berlin, 1998.
- /10/ Rojc M., Kačič Z. (2000). Design of Optimal Slovenian Speech Corpus for use in the concatenative Speech Synthesis System. LREC 00, Athens, Greece.
- /11/ Senn Version 3.0 User Manual. SIEMENS AG. 1998
- /12/ SI1000 (1998). Prosodic Markers Version 1.0, Bavarian Archive of Speech Signals. University of Munich, Institute of Phonetics, Germany.
- /13/ Stergar J. (2000). Determining Symbolic Prosody Features with analysis of Speech Corpora. Master Thesis. University of Maribor. Faculty for EE. and Comp. Sci.
- /14/ Stergar J., Hozjan V. (2000). Steps towards preparation of text corpora for data driven symbolic prosody labelling. T. Erjavec, J. Gros, (ed.). Language technologies: proceedings of the conference. Ljubljana, Slovenia.
- /15/ Toporišič J. (1991). Slovenska slovnica. Založba obzorja Maribor.
- /16/ Tresp V., Neuneier R., Zimmermann H. G.. Early Brain Damage. In Advances in Neural Information Processing Systems, volume 9. MIT Press, 1997.
- /17/ Vereecken H., Martens J. P., Grover C., Fackrell J., Van Coile B. (1998). Automatic prosodic labeling of 6 languages. ICSLP 98, Sydney Australia.
- /18/ Vereecken H., Vorstermans A., Martens J. -P. and Van Coile B. (1997). Improving the Phonetic Annotation by means of Prosodic Phrasing. Proceedings Eurospeech 97. Rhodes, Greece.

## 9 Web References

- /19/ Institut für Phonetik und sprachliche Kommunikation: Siemens Synthese Korpus - SI1000P,  
<http://www.phonetik.uni-muenchen.de/Bas/>.

*mag. Janez Stergar, univ. dipl. inž. el.*  
*red. prof. dr. Bogomir Horvat, univ. dipl. inž. el.*  
*University of Maribor*  
*Faculty of Electrical Engineering and*  
*Computer Science*  
*Smetanova ulica 17*  
*2000 Maribor*  
*tel.: +386 2 220 7203*  
*fax.: +386 2 251 1178*

# TRIANGULAR MESH DECIMATION AND UNDECIMATION FOR ENGINEERING DATA MODELLING

Sebastian Krivograd<sup>1</sup>, Borut Žalik<sup>1</sup>, Franc Novak<sup>2</sup>

<sup>1</sup>University of Maribor, Faculty of Electrical Engineering and Computer Science,  
Maribor, Slovenia

<sup>2</sup>Jožef Stefan Institute, Ljubljana, Slovenia

**Key words:** engineering data modelling, triangular meshes, mesh decimation and undecimation, hash table.

**Abstract:** Triangular mesh decimation is the process that uses local operations on geometry and topology to reduce the number of triangles in a triangle mesh. Triangular meshes are used in many engineering applications where simple interpolation of discrete data replaces continuous and complex model of reality. Furthermore, triangular meshes are standard input to numerical analysis tools based on Finite Element Method. Manipulation with large triangular meshes is a bottleneck in engineering applications hence appropriate simplifications are needed. Apart from intuitive manual techniques, mesh decimation process is an attractive alternative providing optimal computer based solutions. The paper presents a fast algorithm for decimation of triangular meshes using vertex elimination approach. To speed-up the search for the vertex to be removed, a hash table is applied. Presented solution runs in linear time and is suitable for different applications in practice. Its usefulness is increased by an introduction of an undecimation, i.e. a reverse process restoring gradually the initial triangular mesh. An illustrative example from the analysis of a power line electric field is given.

## Modeliranje inženirskih podatkov s poenostavljanjem in rekonstrukcijo trikotniških mrež

**Ključne besede:** modeliranje inženirskih podatkov, trikotniške mreže, poenostavljanje in rekonstrukcija trikotniških mrež, sekljalna tabela.

**Izvleček:** Trikotniške mreže pogosto uporabljamo v inženirskih aplikacijah, predvsem ko želimo interpolirati diskretne odbirke kompleksnih zveznih procesov. Trikotniške mreže so prav tako standarden vhod za različne numerične analize, ki temeljijo na metodi končnih elementov. Manipulacija z zelo velikimi trikotniškimi mrežami pa predstavlja ozko grlo pri inženirskih aplikacijah, zato iščemo enostavnejše a še zmeraj dovolj verne trikotniške mreže. Procesu, ko z uporabo lokalnih operacij nad geometrijo in topologijo podane trikotniške mreže zmanjšamo število vozlišč in trikotnikov, pravimo poenostavljanje trikotniške mreže. Postopek iskanja najprimernejše trikotniške mreže je običajno prepuščen uporabniku in temelji na njegovi intuiciji, zato predstavlja računalniško podprt proces poenostavljanje zanimivo alternativo. V članku predstavimo učinkovit algoritem za poenostavljanje trikotniških mrež, ki temelji na odstranjevanju oglišč. Da bi pohitrili odločitev, katero izmed oglišč je najprimernejšo za odstranitev, uporabimo sekljalno tabelo. Predstavljena rešitev deluje v linearnem času in je primerna za različne aplikacije v praksi, predvsem tam, kjer potrebujemo hiter odziv sistema. Uporabnost pristopa povečamo z možnostjo postopnega vračanja odstranjenih oglišč - z rekonstrukcijo. Delovanje algoritma prikažemo z ilustrativnim primerom poenostavljanja trikotniške mreže električnega polja daljnovoda.

### 1 Introduction

Although natural phenomena are continuous, engineers normally measure their values only at some discrete measurement positions. The values at other positions are then calculated by interpolation. In the applications where scalar values (e.g., temperature, sea level, electric field, tension) are measured in a plane, the most suitable interpolation results in a triangular mesh /1/. Of course, the denser the mesh, the better interpolation can be constructed. Unfortunately, a huge number of measured points may cause problems in manipulation with the corresponding triangular meshes resulting in slow response time and considerable computer memory requirement. This is especially critical, when a triangulation mesh is used as an input into numerical analysis based on FEM (Finite Element Method) /2/. Furthermore, by the widespread use of the internet, large triangular meshes require long transfer time between collaborating parties. Because of that, triangulation meshes are frequently simplified to make an acceptable

compromise between the accuracy and the system limitations. The main idea stems from the fact that a triangular mesh can be simplified in regions with small or no variation of the scalar values. This task is usually performed manually using experience and intuition of a user. An alternative is the use of the so-called mesh decimation algorithms, which simplify the triangular mesh automatically.

Triangular mesh decimation approaches, developed so far, can be classified according to the elements they are taking from the mesh (i.e., vertices, edges, or triangles) /3, 4/:

- **Vertex decimation methods** are the most frequently used and are based on Schroeder simplification algorithm /5/. The vertices are evaluated, and they are incrementally removed from the mesh according to their importance. Various techniques have been proposed, and they differ on how vertices are evaluated and what type of triangulation is required (see /3/ for an overview).

- **Edge decimation methods** eliminate edges, which have been evaluated already /6/. The edge being removed is replaced by a vertex. Triangles, which degenerate to edges, are removed. One of the best edge decimation methods is based on quadric error metrics /7/.
- **Triangle decimation methods** eliminate one or more triangles. Although theoretically feasible, practical approaches using this possibility have not been reported.

In this paper we present an algorithm that combines two mesh decimation approaches. Franc and Skala used a hash table in a parallel environment for speeding-up the search of the most suitable vertex to be removed /8/. They combined the vertex and edge removal in the following way. At first the most suitable vertex is selected, and then among the edges incident to that vertex, the shortest one is contracted. Our algorithm uses the pure vertex decimation similar to the one proposed by Schroeder /5/, but using the hash table as the acceleration technique. The heuristic approach for creation of a hash table for engineering applications have been also developed. Beside that, our algorithm is equipped by an undecimation feature, i.e. by an incremental reconstruction possibility of the original mesh. The algorithm is suitable for engineering data modelling. Low time and space complexity make it even candidate for embedded system applications.

## 2 The decimation algorithm

The proposed algorithm for triangular mesh decimation is briefly sketched as follows:

1. Evaluate all vertices according to a chosen evaluation criterion and arrange them into a hash table.
2. Select the most suitable vertex using the hash table (for example, vertex  $v_i$  in Figure 1a).
3. Remove the vertex from the triangular mesh.
4. Remove all triangles incident on the removed vertex (Figure 1b).
5. Triangulate the area from where the triangles have been removed (see Figure 1c).
6. Re-evaluate vertices incident to the removed vertex (vertices  $v_j, v_k, v_l, v_m, v_n$  in Figure 1c).
7. Return to step 2 until the termination criterion is met.

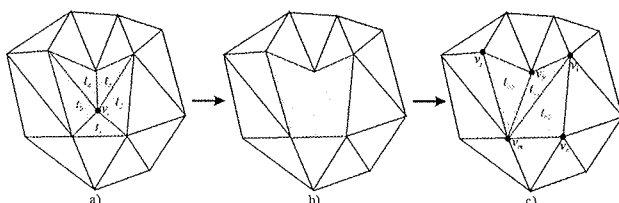


Figure 1: Vertex decimation

### 2.1 Evaluation of vertices

Before the decimation process starts, the vertices have to be evaluated. Let us take a look at Figure 2 where the

triangles with small changes of scalar values in their vertices can be reduced without much spoiling the presentation.

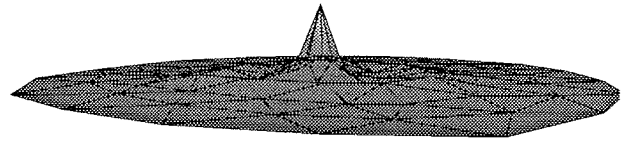


Figure 2: A triangular mesh

The evaluation of vertices is done by investigating the neighbourhood of the vertex under consideration. Different criteria can be applied. Let us mention just two of them:

- Vector  $v_{ij}$  connecting the examined vertex  $v_i$  and its neighbouring vertex  $v_j$  is formed. The angle between this vector and xy plane is calculated. The average value of all angles defined by vertex  $v_i$  is used as the evaluation value  $ev_i$ .
- The average difference of scalar values between the examined vertex  $v_i$  and the neighbouring vertices is used as the evaluation value  $ev_i$ .

Better results are obtained by the first criterion. That can be confirmed by observing Figure 3a and Figure 3b, where the scalar value against the two neighbours is the same on both figures. If the first evaluation criterion is applied, the situation in Figure 3a gives bigger evaluation value  $ev_i$  than in Figure 3b. Applying the second criterion gives the same evaluation value  $ev_i$ . In practice, however, the criterion with average difference of scalar values is normally used.

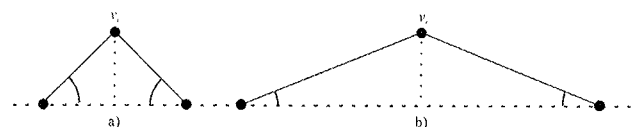


Figure 3: Evaluation of vertices

### 2.2 Selection of a vertex to be removed

The strategy is to remove vertices that cause the smallest change in data representation. Therefore, the vertex with the smallest evaluation value  $ev_i$  is selected and removed from the mesh. The evaluation values  $ev_i$  of the neighbouring vertices are changed and must be estimated again. In the next iteration, the algorithm searches for the next vertex with the smallest  $ev_i$ . It can be selected easily by walking through the set of the remaining vertices, and selecting the one with the smallest  $ev_i$ . Unfortunately, this method works in  $O(n^2)$  time and considerably slows down the algorithm. The second possibility, sorting at first all vertices according to  $ev_i$  and adjusting their position in the sorted array according to the changed  $ev_i$  works in expected  $O(n \log n)$  time. The constant expected time complexity can be achieved by introducing a hash-table /8/. However, in this case, the condition of selecting the vertex with the smallest  $ev_i$  has to be slightly relaxed. The vertices are



organised in the hash table according to their evaluation values  $ev_i$ . Figure 4 schematically shows the structure of the hash table. Vertices in each interval are stored in a FIFO queue.

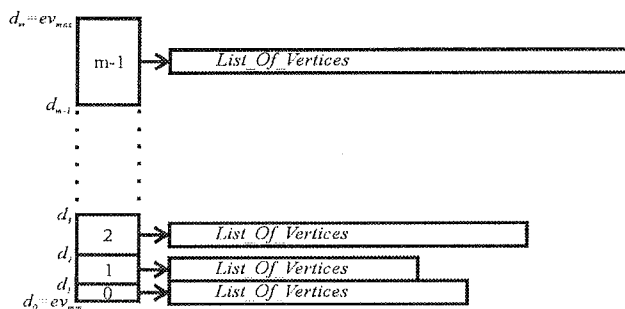


Figure 4: Arranging vertices into hash table

The hash table is usually organised according to the expected distribution of the input data. In our application, the heuristics for linear, quadratic and exponential expected data distributions are prepared. It is important to note that during the decimation process new evaluation values of  $ev_i$  can become greater than maximal evaluation value determined before the hash table has been formed. Therefore, we have to add additional entry to the hash table accepting such possible cases. Having the hash table, the next vertex to be removed from the triangular mesh is now obtained easily. The algorithm always selects the first vertex from the lowest non-empty FIFO queue and removes it. The neighbouring vertices of the removed vertex are evaluated again and inserted in the corresponding interval at the end of the FIFO queues. In that way it is prevented to perform decimation only locally. The hash table assures constant time complexity of this part of the algorithm. By storing a list of neighbouring vertices at each vertex of the mesh, the neighbouring vertices are accessible without any search (see Figure 5). As each vertex has  $l$  neighbouring vertices, in general  $l \ll n$ , the update of the estimation values is realised in  $O(l) \approx O(1)$  time.

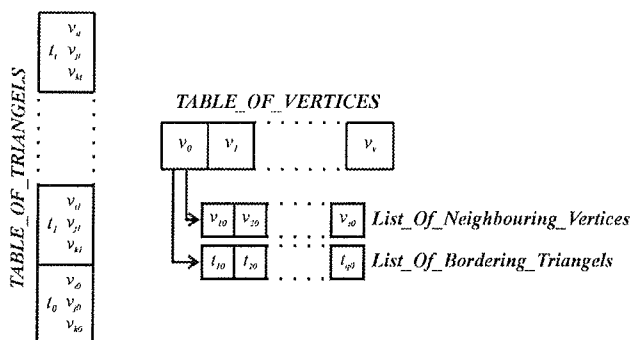


Figure 5: Direct access of the vertex neighbours

It may be desired that the border vertices of the region are eliminated from the decimation process. In this case, they are not inserted into the hash table. Similarly, we can test the shape of the newly created triangles according to the

application specific recommendation. For FEM analyses it is desired, for example, that the rate of the triangle sides does not fall below 1:1:10. If that happens, the considered vertex is not removed.

### 2.3 Triangulation of polygon area

After removing a vertex from the mesh, all the triangles incident to it are eliminated (shaded part in Figure 1). The empty region has to be filled by the new triangles. Franc and Skala applied here a clever solution by selecting the shortest edge from the removed vertex to their neighbours. The shortest edge is contracted pulling all the edges defined by the removed vertex to the opposite vertex of the shortest edge /4/. However, this elegant method works only when the obtained gap forms a convex polygon which is not always the case. Therefore we applied in our solution a classical polygon triangulation algorithm. There are many ways how a polygon can be triangulated: algorithms based on diagonal insertion, restricted Delaunay algorithms, and the algorithms using Steiner points (see /9/ for an overview). In our approach, the well-known ear-cutting algorithm proposed by ElGindy et. al is used /10/.

## 3 Undecimation

Returning the removed vertices into the mesh in the reverse order of their elimination is an extremely useful feature in practice, giving the user the opportunity to experiment with the mesh. The user may return step by step only a few vertices instead of processing the whole set of vertices again and trying different termination criteria. This feature (denoted as *undecimation*) can be realised easily and very efficiently by the proper data structure as described below.

At the beginning we have an array of vertices and an array of triangles. The position of vertices remains the same, they are just pulled-out from the mesh. The removed vertices are marked by flags. When a vertex is removed, triangles incident to it are removed, too. This is indicated in the triangle array by a flag. There are always less new triangles than original and they occupy the memory locations of the old ones. Some of the locations (at least one) are marked as empty. Figure 6a shows the state of the data structure before and Figure 6b after removing vertex  $v_0$  in the example shown in Figure 1.

The undecimation process requires the knowledge how the process of decimation was executed and what changes in the triangular mesh occurred at each step. The easiest solution would be to store a topology of each obtained mesh. This would involve file operation and would slow down the whole process. However, by the proper organisation of data, the shape of the mesh could be easily restored by a tolerable amount of additional memory. Figure 7 explains our solution. Two additional one-way linked lists are introduced at each removed vertex. The first list stores the indices of the removed triangles. It contains only two

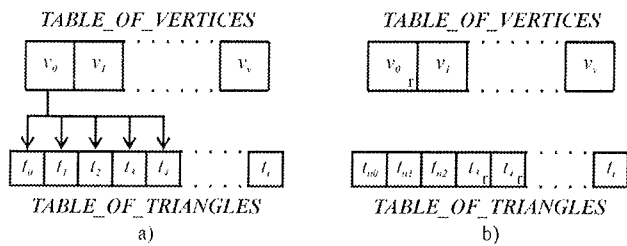


Figure 6: The state of the data structure before (a) and after (b) removing a vertex

indices, because the third one is the removed vertex itself. The second list stores the indices of the new triangles. The process of undecimation is now extremely easy. The vertex, which is going to be returned to the mesh, set-ups the flag indicating that it belongs to the mesh again. Triangles, which have been added by the polygon triangulation process, are removed and the old triangles are restored using the information from the list of the removed triangles.

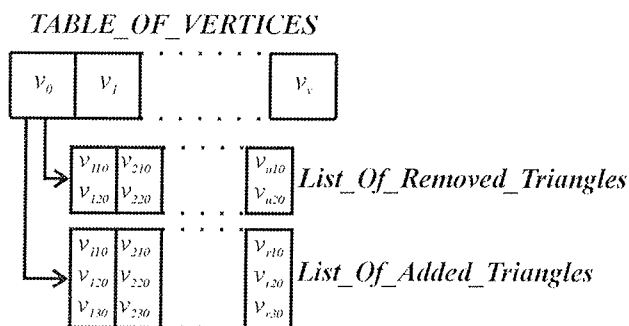


Figure 7: Additional lists by removed vertex

## 4 Results

### 4.1 Theoretical time and space complexity

The proposed algorithm for mesh decimation consists of the steps with the following complexity:

- evaluation of vertices is done in  $O(n)$ , where  $n$  is the number of the input vertices,
- removal of a vertex is realised in constant time  $O(1)$ ,
- triangulation of polygon using the ear-cutting is performed in  $O(l_i^2)$ , where  $l_i$  is the number of neighbouring vertices of the removed vertex  $v_i$ . However,  $l_i \ll n$ , and therefore this step can be considered as being done in constant time regarding  $n$ .
- re-evaluation of the neighbouring vertices of the removed vertex is done in constant time.

If  $k$  is the number of all vertices that are removed during the decimation process and  $k < n$ , the required time complexity becomes  $O(k) + O(n) = O(n)$ . The same estimation is obtained for the process of undecimation.

Let us investigate the space complexity of the algorithm. At the beginning, the space for  $n$  vertices and  $2n$  triangles is allocated (it is well-known that each triangulation consists of at most  $2n - h - 2$  triangles, where  $h$  is the number of vertices forming the convex hull of the given set of polygons /11/). At each vertex  $v_i$  being removed,  $l_i$  records the removed triangles and  $l_i - a$ ,  $0 < a \leq l_i$ ,  $l_i \ll n$ , records about the added triangles are needed. In this way, we obtain linear space complexity  $O(n)$ .

### 4.2 Practical results

Tests have been performed on various sets of engineering data and on artificially generated data. As a case study, consider the electric field of a power line borrowed from /12/. In Figure 8a the initial triangular mesh consisting of 11213 vertices and 22010 triangles is shown. The shaded triangular mesh is shown in Figure 8b.



Figure 8: Initial triangular mesh (a) shaded initial triangular mesh (b)

After that, we start the decimation process. At each step, 10% of vertices are removed from the triangular mesh. This process is shown in Figure 9. Notice that despite the smaller number of triangles, the shaded pictures do not differ noticeably until only 20% of the initial vertices remain. In this case, we obtained 2253 vertices and 4090 triangles.

To show how efficient the proposed algorithm is, we arranged the vertices in a regular grid, and triangular meshes are constructed from them. The scalar values in the vertices have been set randomly. 99% of the vertices have at first been removed, and then, all of them are returned (undecimated). The results are shown in Table 1 where CPU time for mesh decimation and undecimation is given. As seen from Figure 10, the proposed algorithm works indeed in linear time. Experiments have been performed on a PC with Celeron 600 MHz processor and 384 MB of RAM.

Table 1: Times needed for mesh decimation and mesh undecimation

INPUT	no. of vertices (x1000)	10	40	90	160	250	360	490	640	810
OUTPUT	no. of vertices	100	400	900	1600	2500	3600	4900	6400	8100
	no. of triangles	188	775	1772	3167	4959	7156	9748	12742	16121
TIME (s)	decimation	0,140	0,651	1,583	2,774	4,357	6,340	8,742	11,526	14,681
	undecimation	0,140	0,611	1,442	2,583	4,086	5,928	8,202	10,816	13,840

## 5 Conclusion

Huge surface meshes are produced in different fields like volume visualisation in architecture, GIS, industrial design, etc. In electronics, triangular meshes are used for model-

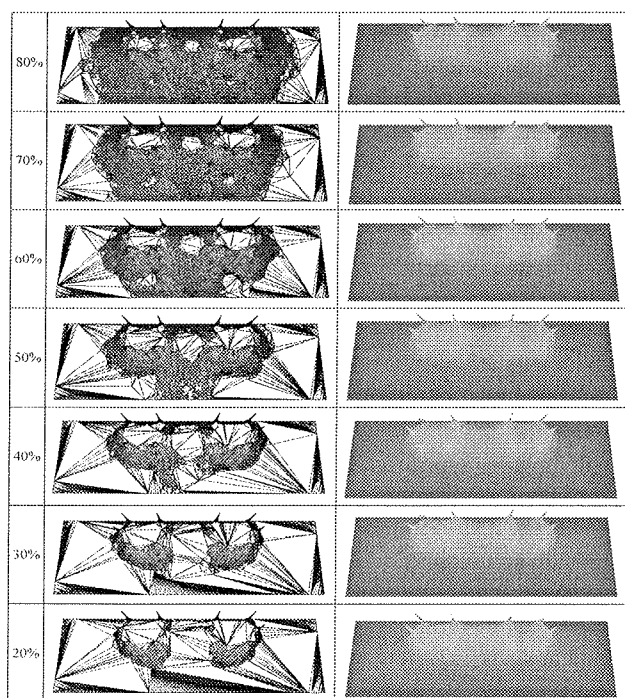


Figure 9: Decimation process

ling temperature distribution and mechanical properties of devices, as well as for visualisation of electrical parameters. Even recent workstations face problems in interactively displaying huge data sets often composed of more than a million of triangles. Reducing the complexity of surface meshes is therefore imperative for engineering applications. This hot topic motivated development of mesh decimation algorithms based on different criteria following specific objectives in practice. The algorithm proposed in this paper combines the approaches of Schroeder /5/ and Franc & Skala /8/. The resulting decimation process is performed in linear time. The usefulness of the algorithm

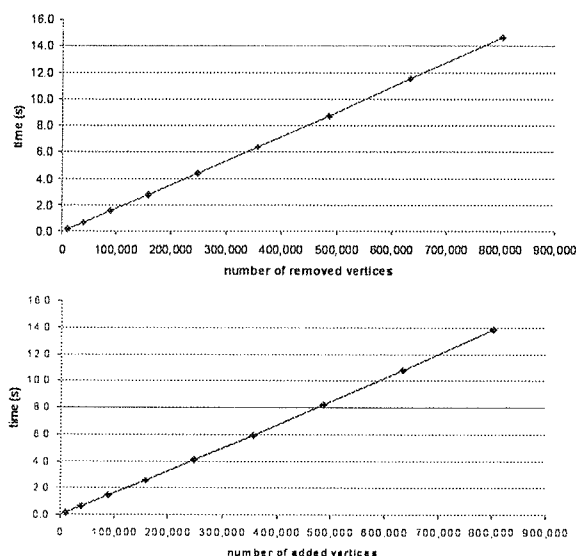


Figure 10: Graph of times needed for mesh decimation (above) and undecimation (below)

has been demonstrated in a case study of modelling of electric field of a power line. Its performance is characterised by an example of constructing artificial triangular meshes. Low time and space complexity and efficient undecimation feature illustrate the strengths of the algorithm and make it a promising solution for modelling engineering data.

## Acknowledgement

The authors would like to thank prof. dr. Mladen Trlep from Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia, for providing us with the empirical data used in the paper.

## Literature

- /1/ M. Lamot, B. Žalik, "Software Tool for the Support of On-line Thermal Monitoring of Microelectronic Systems". Midem, 2000, vol. 30, no. 3, pp. 144-147.
- /2/ M. Trlep, A. Hamler, B. Hribernik, "The analysis of complex grounding systems by FEM". IEEE transaction on magnetic, 1998, vol. 34, no. 5, pp. 2521-2524.
- /3/ M. Garland, P.S. Heckbert, "Fast Polygonal Approximation of Terrains and Height Fields", technical report, 1995 <http://www.cs.cmu.edu/~garland/scape>
- /4/ M. Franc, V. Skala, "Triangular Mesh Decimation in Parallel Environment", EUROGRAPHICS Workshop on Parallel Graphics and Visualization, Girona, Spain, 2000, pp. 39-52
- /5/ W. J. Schroeder, J. A. Zarge, W. E. Lorensen, "Decimation of Triangle Meshes", Computer Graphics, 1992, vol. 26, no. 2, pp. 65-70
- /6/ H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, "Mesh Optimization", Proceedings of the 1993 ACM SIGGRAPH conference, 1993, pp. 19-26.
- /7/ M. Garland, P. S. Heckbert, "Surface Simplification Using Quadric Error Metrics", SIGGRAPH Conference Proceedings, 1997
- /8/ M. Franc, V. Skala, "Parallel Triangular Mesh Decimation Without Sorting", SCCG Proceedings, Budmerice, 2001, pp. 69-75
- /9/ M. Lamot, B. Žalik, "A Contribution to Triangulation Algorithms for Simple Polygons". Journal of Computing and Information Technology - CIT, 2000, vol. 8, no. 4, pp. 319-331.
- /10/ H. ElGindy, H. Everett, G. Toussaint, "Slicing an Ear in Linear Time", internal memorandum, School of Computer Science, McGill University, 1989
- /11/ M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, "Computational Geometry: Algorithms and Applications", Springer 1997.
- /12/ M. Trlep, B. Hribernik, "Unified approach to solving a steady-state electromagnetic field. IEEE transaction on magnetic, 1997, vol. 33, no. 2, pp. 1974-1977.

izr. prof. dr. Borut Žalik, univ. dipl. inž.  
Sebastian Krivograd, univ. dipl. inž.

Univerza v Mariboru  
Fakulteta za elektrotehniko, računalništvo in  
informatiko, Smetanova ulica 17, 2000 Maribor  
tel: (02) 220 74 71, fax: (02) 251 11 78  
e-mail: zalik, sebastian.krivograd@uni-mb.si

izr. prof. dr. Franc Novak, univ. dipl. inž.  
Institut "Jožef Stefan"  
Jamova 39, 1000 Ljubljana  
tel: (01) 477 33 86, fax: (01) 251 93 85  
e-mail: franc.novak@ijs.si

## APPLICATION ARTICLE

## SELECTING BETWEEN ROM, FASTROM AND FLASH FOR A MICROCONTROLLER

## STM - Microcontroller Division Applications

## Introduction

A customer who develops an MCU-based application needs various levels of flexibility in order to perform code modifications at different times in the life cycle of the product (these levels are explained on the next page). To satisfy these requirements, STMicroelectronics supports several device types within two main groups of microcontroller product families:

- EPROM, OTP, FASTROM and ROM microcontroller families
- Flash, FASTROM and ROM microcontroller families

This Application Note discusses the second group of families. For information on the first group, refer to Application Note AN886.

## Definition of terms

**Flash:** Flash devices are electrically programmable and erasable. Device programming is typically performed by the customer, so changes to the program code can be made quickly and easily. Flexibility is further enhanced for

programmed by STMicroelectronics with the customer's code and selected options. The advantage of FASTROM, compared to Flash, is improved programming efficiency for large quantities (10,000+) and compared to ROM, it has the advantage of a shorter leadtime and the devices can be reprogrammed.

**ROM** (Read Only Memory): ROM devices are programmed by STMicroelectronics at the fabrication step using a special mask containing the customer code. Therefore, the code cannot be modified afterwards.

Costs are highly dependent on the **flexibility** given to the device (ability to be easily erased or programmed). ROM is the cheapest technology but provides little flexibility whereas Flash is more flexible but its manufacturing cost is higher.

## 1 Typical application development flow

When a new application is developed, different device versions will be used at each step of the development, depending on the required **programming flexibility**.

	Design Phase	Validation Phase	Pre-production Phase	Production Phase	
<b>ST Solution</b>	<i>Flash</i>	<i>Flash</i>	<i>Flash</i>	<i>Flash</i>	<i>ROM</i>
<b>Code Updates</b>	....	...	..	.	None
<b>Number of Units</b>	.	..	...	....	.....

the customer by the use of In-Circuit Programming, In Application Programming, and In-Circuit Testing.

- In-Circuit Programming (ICP) allows the customer to program or erase the device after it has been soldered on the application board.
- In Application Programming (IAP) allows the user to program or erase part of the Flash memory while the application is running.
- ICT allows the customer to program test routines in Flash memory to be executed during the board manufacturing phase and subsequently replaced by the final application code.

**FASTROM** (Factory Advanced Service Technique Read Only Memory): this type of MCU is a Flash device pre-

During the **design and validation phases**, a high flexibility is required and only a small number of parts are necessary, therefore the use of Flash is recommended.

The next step is **pre-production phase**: only a few code updates are needed at a reasonable device cost. Again, the best choice is to use Flash memory. Finally, when the **mass production phase** begins, there is no more need for corrections since the product has been fully optimized, so ROM is the most suitable if very high volumes are needed. Otherwise (low to medium volumes) the most effective solution is to continue using STMicroelectronics' competitively priced Flash.

The following table summarizes the main benefits and drawbacks of using ROM or Flash MCU devices.

	ROM	Flash
+	<b>Cheaper than Flash</b> (simpler process and testing) <b>Lower failure rate</b> (less handling, no programming)	Ability to be programmed <b>directly</b> by the <b>final user</b> Depending on the silicon technology used, 100 or 10,000 write/erase cycle endurance at 25°C ambient temperature
-	<b>Limited flexibility</b> (customer code implemented at masking stage) Higher inventory risks	Higher failure rate compared to ROM due to customer handling and programming Expensive silicon manufacturing process

## 2 Flash write/erase cycle endurance

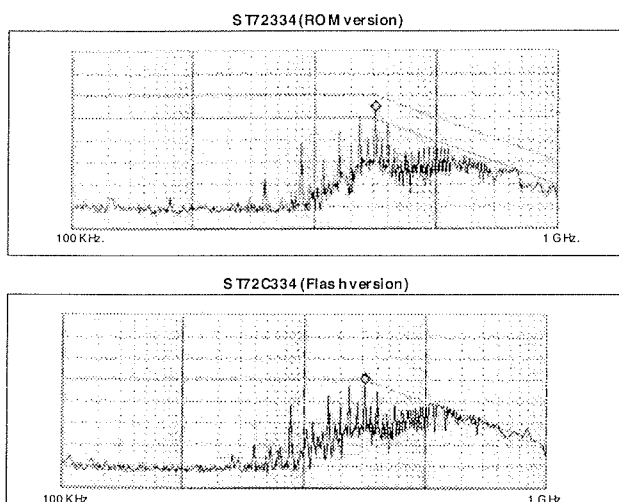
During the product and process qualification phase, STMicroelectronics performs three times the minimum number of write/erase cycles at room temperature (300 cycles to qualify a min. spec of 100 cycles at 25°C). In addition, 100 cycles are performed over the whole operating temperature range (-40 -125°C) during product qualification.

Due to process differences, some microcontrollers (all ST9 MCUs and ST7 devices) with more than 8 Kbytes of Flash memory, are specified with a min. endurance of 10,000 write/erase cycles. To reduce chip size and cost smaller ST7 devices use a process that guarantees 100-cycle endurance for Flash memory.

Data retention tests are performed at high temperature to guarantee 20 data retention.

## 3 Comparison of ROM and FLASH

ROM and Flash devices have almost the same functional and electrical behaviour in an application because STMicroelectronics designs the two products with the same methodology. ROM and Flash devices are qualified using the same procedures. They are tested using the **same test flows** (except for the Flash-specific programming tests) and with the **same parameter limits**. A good indication of device similarity is EMC (electromagnetic noise immunity) measurements performed on both versions (ROM and Flash) for the same MCU device (see two spectrums below).



## 4 Typical manufacturing lead time for ROM and FLASH

The complexity of all the operations needed to manufacture a component implies a certain time period. Understanding STMicroelectronics MCU manufacturing cycle is important in order to establish good relationships with customers. The numbers given here are typical and subject to change in the future

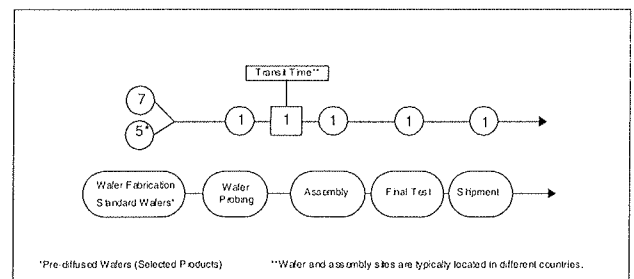


Figure 1. Typical Fabrication Leadtime.

In order to limit lead time on ROM products, STMicroelectronics has introduced pre-diffusion technology on selected products. This allows a two-week reduction in total cycle time. Also notice that MCU devices have to be ordered in specified minimum quantities for ROM versions.

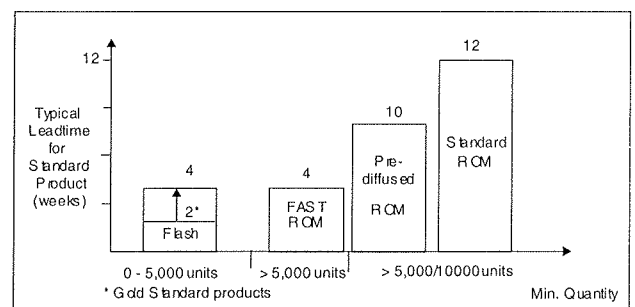


Figure 2. Typical Leadtime for Standard Products

## 5 Minimum order quantities for ROM and FASTROM

The following minimum order quantities apply to ROM and FASTROM microcontroller devices:.

	ROM			FASTROM
	<i>Minimum quantity per year</i>	<i>Minimum order quantity</i>	<i>Minimum quantity per line item</i>	<i>Minimum quantity per year and per line item</i>
ST6 Family	100000	50000	10000	5000
ST7 and ST9 Families	50000	25000	5000	5000

## 6 Flash reliability

### Why do ROM devices have very low failure rates?

For ROM parts, the customer program is included at a **specific mask level** of the wafer fabrication. Therefore, the complete product functionality is present in both the die and the assembled product. This functionality can be fully evaluated at both **wafer probing** and **final electrical test**, thus ensuring a **low reject rate** at customer manufacturing stage.

### Testing Cycling Endurance and Data Retention for Flash devices

To ensure optimal Flash quality, STMicroelectronics tests two important device characteristics: **write/erase cycling endurance** and **data retention**. The program memory of a Flash MCU should be seen as a number of cells that will be activated during programming and deactivated during erasing.

During the various test phases, the dice are electrically tested and the memory is programmed and erased to verify **cycling endurance**. They are placed in high temperature **bake** to accelerate any possible memory retention

defects. The dice are then tested again to check **data retention**.

### Description of Flash cycling endurance test

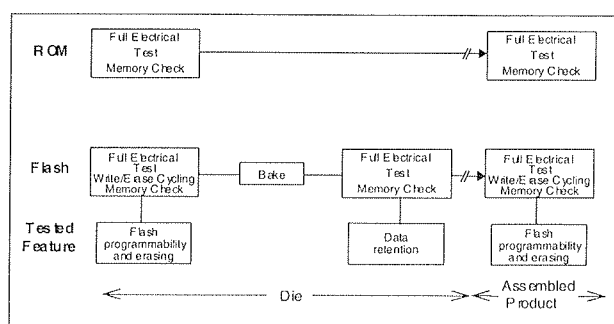


Figure 3. Comparison of the electrical tests performed on Flash and ROM

Although the write/erase cycling endurance of the Flash die is verified as fully functional at probe test, the assembly process can **affect this parameter** in the finished product (for example by damaging some cells). It is therefore necessary to make a **final test** again to check **cycling endurance**.

## Misli ob štiridesetletnici prvih magistrov

Pišem po spominu. Iz zgodovinskih vzrokov prosim cenjenega bralca za eventualno korekcijo in dopolnitve.

Zlasti na tehniki je bilo težko priti do doktorata, februarja 1963 je bil na FE, 44 let po ustanovitvi, podeljen 13. doktorat. Redni profesorji, celo Marij Osana, graditelj enkratnega oddajnika v Domžalah, prvega na Balkanu, je bil brez doktorata. Poleg doktorske teze je bilo treba položiti zahteven rigoroz. V tej zvezi je zanimiv izrek akademika prof. dr. Milana Vidmarja: »Če bi mi to znali, kar zahtevamo od novih doktorantov, bi se nivo FE zelo dvignil.«

Nekako koncem 50ih ali začetkom 60ih let se je spremenila tozadevna zakonodaja, ki je vpeljala magisterski študij. Magisterski študij je nadomestil rigoroz in omogočil direkten dostop k doktoratu. V to špranjo je vskočil po svojih rezultatih eden od najzaslužnejših znanstvenikov Slovenije, prof. dr. Aleš Strojnik in razpisal leta 1961 prvi magisterski študij iz področja **Elektronske optike in elektronsko-optičnih naprav** v Ljubljani. Študij je bil tako nov, da ni bilo jasno, ali se okrajša z mgr., ali mag., in niti se ni vedelo, kako naj izgledajo diplome. V tej neodločenosti FE si je pisec teh vrstic dal sam izdelati formular magister diplome v tiskarni državnih železnic in sicer rumene barve, ker primernejšega papirja niso imeli. Po velikosti je bila večja od diplomske in manjša od doktorske listine. Iz te Strojnikove šole je izšlo lepo število profesorjev FE v Ljubljani in Mariboru in tudi nekaj rektorjev obeh slovenskih univerz.

Po odhodu prof. Strojnika na druge kontinente (Avstralijo, Afriko in Ameriko) je ta študijska smer na FE v Ljubljani zamrla. Leta 1983 sta jo prof. dr. Evgen Kansky in pisec teh

vrstic z zelo aktivno pomočjo prof. dr. Donlagić-a in prof. dr. Kumperščak-a oživila na tehniški fakulteti, sedaj FERi v Mariboru pod imenom **Elektronska vakuumistika**. Šola je bila namenjena za celo Jugoslavijo, profesorji so bili poleg domačih tudi z Beograda in Züricha in študentov je bilo pri prvem vpisu preko 20. Žal ta smer danes le životari, doživlja slično usodo kot slovenska podjetja in banke, ki se pod tkzv. privatizacijo ponujajo tujemu kapitalu.

Po mojem videnju so bili v Strojnikovi šoli podeljeni magisteriji unikat v svetovnem merilu. Tedanji mag. sc. je bil upravičen na docentsko mesto, dr. sc. pa na položaj rednega profesorja. S tem je bil ta doktorat enakovreden nemškemu dr. habil. in ruskemu dr. nauke, mag. sc. pa nemškemu dr. in ruskemu kandidatu dr. znanosti. Naslov mag. je bil kasneje vpeljan tudi v Avstriji in je enakovreden naši diplomii. Anglosaksonski master (mojster) ni identični z našim, oz. latin-skim mag. (najvišji vodja), kot je ves njihov šolski sistem drugačen od evropskega.

Imam pa občutek, da se enako kot politično EU tudi v zahtevnosti magisterijev in doktoratov s hitrimi koraki približujemo naši severni in verjetno tudi zapadni sosedu.

Celovec, 07. 10. 2001

*Alojz Paulin*  
Ročevnica 59  
4290 Tržič



## Informacije MIDE M

Strokovna revija za mikroelektroniko, elektronske sestavine dele in materiale

### NAVODILA AVTORJEM

Informacije MIDE M je znanstveno-strokovno-društvena publikacija Strokovnega društva za mikroelektroniko, elektronske sestavne dele in materiale - MIDE M. Revija objavlja prispevke domačih in tujih avtorjev s področja mikroelektronike, elektronskih sestavnih delov in materialov, ki so lahko:

izvirni znanstveni članki, pregledni znanstveni članki, predhodne objave, strokovni članki ter predavanja in povzetki s strokovnih posvetovanj.

Strokovni prispevki bodo recenzirani.

Revija objavlja tudi aplikacijske članke, poljudne članke, noviče iz stroke, vesti iz delovnih organizacij, inštitutov in fakultet, obvestila o akcijah društva MIDE M in njegovih članov ter druge prispevke.

Strokovni prispevki morajo biti pripravljeni na naslednji način:

1. Naslov dela, imena in priimki avtorjev brez titul, imena institucij in firm.
2. Ključne besede in izveček (največ 250 besed).
3. Naslov dela v angleščini.
4. Ključne besede v angleščini (key words) in podaljšani povzetek (Extended Abstract) v angleščini.
5. Uvod, glavni del, zaključek, zahvale, dodatki in literatura v skladu z IMRAD shemo (Introduction, Methods, Results and Discussion).
6. Polna imena in priimki avtorjev s titulami, naslovi institucij in firm, v katerih so zaposleni ter Tel./Fax/Email podatki.

### Ostala splošna navodila

1. V članku je potrebno uporabljati SI sistem enot oz. v oklepaju navesti alternativne enote.
2. Risbe je potrebno izdelati ali iztiskati na belem papirju. Širina risb naj bo do 7.5 oz. 15 cm. Vsaka risba, tabela ali fotografija naj ima številko in podnapis, ki označuje njeno vsebino. Risb, tabel in fotografij ni potrebno lepiti med tekst, ampak jih je potrebno ločeno priložiti članku. V tekstu je treba označiti mesto, kjer jih je potrebno vstaviti.
3. Delo je lahko napisano in objavljeno v slovenščini ali v angleščini.
4. Uredniški odbor ne bo sprejel strokovnih prispevkov, ki ne bodo poslani v dveh izvodih.
5. Avtorji, ki pripravljajo besedilo v urejevalnikih besedil lahko pošljejo zapis datoteke na disketi (3.5"/1.44 MB) v formatih ASCII ali Word for Windows, ker bo besedilo oblikovano v programu Adobe PageMaker 6.5. Grafične datoteke so lahko v formatu TIFF, EPS, VMF, GIF ali JPEG.

Avtorji so v celoti odgovorni za vsebino objavljenega sestavka. Rokopisov ne vračamo.

Rokopise pošljite na naslov:

**Uredništvo Informacije MIDE M**

**MIDE M pri MIKROIKS**

**Stegne 11, 1521 Ljubljana**

**Slovenija**

**Email: Iztok.Sorli@guest.arnes.si**

**Tel. 01 511 22 21, fax. 01 511 22 17**

## Informacije MIDE M

Journal of Microelectronics, Electronic Components and Materials

### INFORMATION FOR CONTRIBUTORS

Informacije MIDE M is a professional-scientific-social publication of Professional Society for Microelectronics, Electronic Components and Materials - MIDE M. In the Journal contributions of domestic and foreign authors are published covering the field of microelectronics, electronic components and materials. These contributions may be:

original scientific papers, review scientific papers, preliminary communications, professional papers, conference papers and abstracts.

All professional contributions are subject to reviews.

Applications articles, scientific news, news from the companies, institutes and universities, reports on actions of MIDE M Society and its members as well as other relevant contributions are also welcome.

Each professional contribution should include the following specific components:

1. Title of paper, authors names, name of the institution/company.
2. Key Words and Abstract (not more than 250 words).
3. Introduction, maintext, conclusion, acknowledgements, appendix and references following the IMRAD scheme (Introduction, Methods, Results and Discussion).
4. Full authors' names, titles and complete company or institution address including Tel./Fax/E-mail.

COMMENT: Slovenian authors who write in English language must submit title, abstract and key words also in Slovene language.

### General informations

1. Authors should use SI units and provide alternative units in parentheses wherever necessary.
2. Illustrations should be in black on white paper. Their width should be up to 7.5 or 15 cm. Each illustration table or photograph should be numbered and with legend added. Illustrations tables and photographs are not to be placed into the text but added separately. However, their position in the text should be clearly marked.
3. Contributions may be written and will be published in Slovene language.
4. Papers will not be accepted unless two copies are received.
5. Authors may send their files on formatted diskettes (3.5"/1.44 mb/) in ASCII or Word for Windows format as text will be formatted in Adobe PageMaker 6.5. Graphic files may be in TIFF, EPS, VMF, GIF or JPEG formats.

Authors are fully responsible for the content of the paper. Manuscripts are not refunded.

Contributions are to be sent to the address:

**Uredništvo Informacije MIDE M**

**MIDE M at MIKROIKS**

**Stegne 11, 1521 Ljubljana**

**Slovenia**

**Email: Iztok.Sorli@guest.arnes.si**

**Tel. +386 1 511 22 21, fax. +386 1 511 22 17**