ELECTRONICS AND TELECOMMUNICATIONS QUARTERLY

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI

IMPORTANT MESSAGE FOR THE AUTHORS

The Editorial Board during their meeting on the 18th of January 2006 authorized the Editorial Office to introduce the following changes:

1. PUBLISHING THE ARTICLES IN ENGLISH LANGUAGE ONLY

Starting from No 1'2007 of E&T Quarterly, all the articles will be published in English only.

Each article prepared in English must be supplemented with a thorough summary in Polish (e.g. 2 pages), including the essential formulas, tables, diagrams etc. The Polish summary must be written on a separate page. The articles will be reviewed and their English correctness will be verified.

2. COVERING THE PUBLISHING EXPENSES BY AUTHORS

Starting from No'2007 of E&T Quarterly, a principle of publishing articles against payment is introduced, assuming non-profit making editorial office. According to the principle the authors or institutions employing them, will have to cover the expenses in amount of 760 PLN for each publishing sheet. The above amount will be used to supplement the limited financial means received from PAS for publishing; particularly to increase the capacity of next E&T Quaterly volumes and verify the English correctness of articles. It is neccessary to increase the capacity of E&T Quarterly volumes due to growing number of received articles, which delays their publishing.

In case of authors written request to accelerate the publishing of an article, the fee will amount to 1500 PLN for each publishing sheet.

In justifiable cases presented in writing, the editorial staff may decide to relieve authors from basic payment, either partially or fully. The payment must be made by bank transfer into account of Warsaw Science Publishers The account number: Bank Zachodni WBK S.A. Warszawa Nr 94 1090 1883 0000 0001 0588 2816 with additional note: "For Electronics and Telecommunications Quarterly".

Editors

El
Elektr

Th
of Pol
Quarte
theore
recogn
tronics

Th
young

Th
critica
branch
mathe
and IS

Al
specia
The pu
Scienc

Th
telecon
Moreo

Ea
distrib
author
access

Th
and th
public
editori

Th
office

Dear Authors,

Electronics and Telecommunications Quarterly continues tradition of the "Rozprawy Elektrotechniczne" quarterly established 53 years ago.

The E&T Quarterly is a periodical of Electronics and Telecommunications Committee of Polish Academy of Science. It is published by Warsaw Science Publishers of PAS. The Quarterly is a scientific periodical where articles presenting the results of original, theoretical, experimental and reviewed works are published. They consider widely recognised aspects of modern electronics, telecommunications, microelectronics, optoelectronics, radioelectronics and medical electronics.

The authors are outstanding scientists, well-known experienced specialists as well as young researchers – mainly candidates for a doctor's degree.

The articles present original approaches to problems, interesting research results, critical estimation of theories and methods, discuss current state or progress in a given branch of technology and describe development prospects. The manner of writing mathematical parts of articles complies with IEC (International Electronics Commision) and ISO (International Organization of Standardization) standards.

All the articles published in E&T Quarterly are reviewed by known, domestic specialists which ensures that the publications are recognized as author's scientific output. The publishing of research work results completed within the framework of *Ministry of Science and Higher Education* GRANTs meets one of the requirements for those works.

The periodical is distributed among all those who deal with electronics and telecommunications in national scientific centres, as well as in numeral foreign institutions. Moreover it is subscribed by many specialists and libraries.

Each author is entitled to free of charge 20 copies of article, which allows for easier distribution to persons and institutions domestic and abroad, individually chosen by the author. The fact that the articles are published in English makes the quarterly even more accessible.

The articles received are published within half a year if the cooperation between author and the editorial staff is efficient. Instructions for authors concerning the form of publications are included in every volume of the quarterly; they may also be obtained in editorial office.

The articles may be submitted to the editorial office personally or by post; the editorial office address is shown on editorial page in each volume.

<div align="right">Editors</div>

M. Raws
A. Baran
K. Bronk
    (MI
J. Stancli
P. Bedna
B. Śwista
Informati

CONTENTS

tal
is i
On
the
deg
con
fur
des

Ke

Fun
much re
(FPGAs
is based
of impl
for LUT
nodes w
by a si
structure
breaking
co-oper

Decomposition of Boolean Function Sets

MARIUSZ RAWSKI

Warsaw University of Technology
Institute of Telecommunications
Nowowiejska 15/19 Warszawa, Poland
e-mail: rawski@tele.pw.edu.pl

The functional decomposition is recognized as very efficient synthesis method of digital circuits. Logic circuits have usually many outputs. Independent synthesis of each output is inefficient. There are several methods that allow decomposing multiple-output function. One of them is decomposition based on blanket calculus. However this method requires the multiple-output function to be represented as single truth table. This limits in the great degree the synthesis strategies that can be used. In this paper new method o functional decomposition based on blanket calculus is presented. This method allows the multiple-output function to be represented as a set of separate truth tables for each output. This allows the designer to apply a much wider range of synthesis strategies.

Keywords: Boolean function sets, logic synthesis, functional decomposition

1. INTRODUCTION

Functional decomposition is a logic synthesis method that has recently gained much recognition. The main reason is the evolution of field programmable gate-arrays (FPGAs) as a new technology for digital system implementation. Architecture of FPGA is based on the lookup table (LUT) as basic building block. An n-input LUT is capable of implementing any Boolean function of up to n variables. Thus, logic synthesis for LUT-based FPGAs must transform a logic network into network that consists of nodes with up to n inputs only. Each node of such network can be then implemented by a single LUT. For this reason, for the case of implementation targeting FPGA structure, decomposition is a very efficient method. Functional decomposition relies on breaking down a complex system into a network of smaller and relatively independent co-operating sub-systems, in such a way that the original system's behavior is preserved.

A system is decomposed into a set of smaller subsystems, such that each of them is easier to analyze, understand and synthesize. Decomposition allows synthesizing the Boolean function into multilevel structure that is built of components, each of which is in the form of LUT logic block specified by truth tables.

Since the Ashenhurst-Curtis decomposition have been proposed, the research has been focused in forming new decomposition techniques [8]. The researchers have developed many types of decompositions, but they are still based on Ashenhurst's ideas. Thanks to the fact that the functional decomposition gives very good results in the logic synthesis of combinational circuits, it is viewed for the most part, as a synthesis method for implementing combinational functions into FPGA-based architectures [2], [4], [9], [16]. However, the decomposition-based method can be used beyond this field. Decomposition-like synthesis methods are not limited only to logic synthesis of digital circuits. The strong motivation for developing decomposition techniques comes recently from modern research areas such as pattern recognition, knowledge discovery and machine learning in artificial intelligence [7], [10], [14], [15].

Efficiency of functional decomposition has been proved in many theoretical papers [1], [2], [9], [13].

Logic circuits usually have many outputs. The functional decomposition in such case has additional advantage, since it allows extracting shared logic. However the synthesis of multi-output circuit requires such representation of multiple-output function that allows efficient creation of sub-circuits common to all of the outputs. There have been proposed methods to represent multiple-output functions by using compact binary decision diagrams (BDDs). One of the first methods is a multi-terminal binary decision diagram (MTBDD) [12]. Unfortunately, MTBDDs tend to be too large to construct. The second method is a binary decision diagram (BDD) for the characteristic function (CF) of the multiple-output function. The advantage of the CF is its small evaluation time. CFs are used in logic simulation and multi-level logic optimization [3]. The third method is a shared binary decision diagram (SBDD) [12]. In many cases, SBDDs are smaller than corresponding MTBDDs and BDDs for CFs. Recently an encoded characteristic function for non zero outputs has been proposed for compact representation of multi-output Boolean function [11].

A very promising decomposition alternative based on the blanket calculus approach to circuit synthesis has been proposed in [5]. This synthesis process does not use the technology independent optimization phase, because this would destroy the design-freedom represented by "don't cares" of the original function specification and construct the network without any relation to the actual synthesis target. The functional decomposition is applied directly to the original function specified by truth table in order to use its whole design-freedom to directly construct a feasible network of LUTs optimized for a given FPGA. Moreover, the concept of parallel decomposition was introduced and effectively applied in the so called balanced decomposition method [6]. Based on the input variable analysis of each single output of a multi-output function F, parallel decomposition separates F into two or more sub-functions, each of which

has as its inputs and outputs a subset of the original inputs and outputs of F. Although the decomposition method uses a mix of both the classic functional decomposition and parallel decomposition, the crucial process in the whole mapping is the functional decomposition, which – in contrast to parallel – is called serial decomposition.

Functional decomposition methods based on BDD, as well as on blanket calculus require the decomposed multi-output function to be represented as single BDD or truth table. This makes decomposition of function sets difficult task, when single functions depend on different sets of input variables or are specified by different input patterns.

In the paper a method based on blanket calculus for functional decomposition of function sets is presented. Separate functions can be specified by truth tables with different input patterns and can depend on different input variables. Method allows computing decomposition for any subset of functions, if only a set of input variables common for all decomposed function can be found. The method allows much greater control over the multilevel decomposition process in comparison to balanced decomposition.

After an introduction to cube representation of function, blanket calculus is presented. Following that functional serial decomposition is described. Next the balanced decomposition is discussed. Subsequently new method is described and compared to balanced decomposition.

2. BASIC THEORY

Here only some information that is necessary for an understanding of this paper is reviewed. More detailed description of functional decomposition based on partition calculus can be found in [1].

2.1. CUBE REPRESENTATION OF BOOLEAN FUNCTIONS

A Boolean function can be specified using the concept of **cubes** (input terms, patterns) representing some specific sub-sets of minterms. In a minterm, each input variable position has a well-specified value. In a cube, positions of some input variables can remain unspecified and they represent "any value" or "don't care" (–). A cube may be interpreted as a p-dimensional subspace of the n-dimensional Boolean space or as a product of $n - p$ variables in Boolean algebra (p denotes the number of components that are '–'). Boolean functions are typically represented by truth tables. Truth table description of function using minterms requires 2^n rows for function of n variables. For function from Table 1a truth table with $2^6 = 64$ rows would be required. Since cube represents a set of minterms, application of cubes allows for much more compact description in comparison with minterm representation. For example cube 0101-0 from row 1 of truth table from Table 1a represents set of two minterms {010100, 010110}.

For pairs of cubes and for a certain input subset B, we define the **compatibility relation** COM as follows: each two cubes S and T are compatible (i.e. $S, T \in \text{COM}(B)$) if and only if $x(S) \sim x(T)$ for every $x \subseteq B$. The compatibility relation $\sim$ on $\{0, -, 1\}$ is defined as follows [1]: $0 \sim 0$, $- \sim -$, $1 \sim 1$, $0 \sim -$, $1 \sim -$, $- \sim 0$, $- \sim 1$, but the pairs $(1, 0)$ and $(0, 1)$ are not related by $\sim$. The compatibility relation on cubes is reflexive and symmetric, but not necessarily transitive. In general, it generates a "partition" with non-disjoint blocks on the set of cubes representing a certain Boolean function F. The cubes contained in a block of the "partition" are all compatible with each other.

"Partitions" with non-disjoint blocks are referred to as blankets [1]. The concept of blanket is a simple extension of ordinary partition and typical operations on blankets are strictly analogous to those used in the ordinary partition algebra.

Table 1

Set of two Boolean function: a) $y_1 = f(x_1, x_2, x_3, x_4, x_5, x_7)$, b) $y_2 = f(x_1, x_3, x_6, x_7, x_8)$

a)

	x_1	x_2	x_3	x_4	x_5	x_7	y_1
1	0	1	0	1	–	0	0
2	0	1	0	–	0	0	0
3	–	1	0	0	0	–	0
4	0	1	0	1	1	–	0
5	0	0	1	–	–	1	0
6	–	–	1	1	–	1	0
7	1	–	1	1	0	–	0
8	0	0	–	–	–	0	1
9	–	1	0	0	1	–	1
10	1	–	1	0	–	–	1

b)

	x_1	x_3	x_6	x_7	x_8	y_2
1	0	1	0	1	–	0
2	1	1	0	–	0	0
3	0	–	1	–	–	1
4	0	–	–	0	–	1
5	–	0	–	1	0	1
6	1	1	0	–	1	1
7	–	1	1	1	–	1

2.2. REPRESENTATION AND ANALYSIS OF BOOLEAN FUNCTIONS WITH BLANKETS

A **cover** on a set S is such a collection of (not necessary disjoint) subsets B_i of S, called blocks, that

$$\bigcup_i B_i = S$$

The product of two covers σ_1 and σ_2 is defined as follows:
$\sigma_1 \bullet \sigma_2 = \{B_i \cap B_j \mid B_i \in \sigma_1 \text{ and } B_j \in \sigma_2\}$.

A **blanket** on a set S is a cover $\beta = \{B_1, \ldots, B_k\}$ of nonempty and distinct subsets of S, called blocks.

Define "nonempty" operator *ne* as follows. For any set $\{S_i\}$ of subsets of set S, $ne\{S_i\}$ is $\{S_i\}$ with empty subset removed, if was originally present and only one instance of block if more similar block were originally present.

The product of two blankets β_1 and β_2 is defined as follows:

$$\beta_1 \bullet \beta_2 = ne\{B_i \cap B_j \mid B_i \in \beta_1 \text{ and } B_j \in \beta_2\}.$$

For two blankets we write $\beta_1 \leq \beta_2$ if and only if for each B_i in β_1 there exists a B_j in β_2 such that $B_i \subseteq B_j$. The relation $\leq$ is reflexive and transitive.

Each block B_i of cover (blanket) has its cube representative $r(B_i)$ that indicates the value of variables inducing this cover (blanket) corresponding to this block.

Example 1 (Blanket-based representation of Boolean functions).

For function F from Table 1a, the blankets induced by particular input and output variables on the set of function F's input patterns (cubes) are as follows:

$$\beta_{x1} = \{B_1; B_2\} = \{\overline{1,2,3,4,5,6,8,9}; \overline{3,6,7,9,10}\},$$

$$\beta_{x2} = \{\overline{5,6,7,8,10}; \overline{1,2,3,4,6,7,9,10}\},$$

$$\beta_{x3} = \{\overline{1,2,3,4,8,9}; \overline{5,6,7,8,10}\},$$

$$\beta_{x4} = \{\overline{2,3,5,8,9,10}; \overline{1,2,4,5,6,7,8}\},$$

$$\beta_{x5} = \{\overline{1,2,3,5,6,7,8,10}; \overline{1,4,5,6,8,9,10}\},$$

$$\beta_{x7} = \{\overline{1,2,3,4,7,8,9,10}; \overline{3,4,5,6,7,9,10}\},$$

$$\beta_{y1} = \{\overline{1,2,3,4,5,6,7}; \overline{8,9,10}\}.$$

The representative of block B_1 of blanket β_{x1} is $r(B_1) = 0$, since variable x_1 has value 0 for input patterns 1, 2, 3, 4, 5, 6, 8, 9. Similarly for block B_2 of blanket β_{x1} representative is $r(B_2) = 1$.

Product of blankets $\beta_{x2}, \beta_{x4}, \beta_{x5}$ presented as cover σ_{x2x4x4} may have empty and repetitive blocks.

$$\sigma_{x2x4x5} = \beta_{x2} \bullet \beta_{x4} \bullet \beta_{x5} = \{B_1; B_2; B_3; B_4; B_5; B_6; B_7; B_8\} =$$

$$= \{\overline{5,8,10}; \overline{5,8,10}; \overline{5,6,7,8}; \overline{5,6,8}; \overline{2,3,9,10}; \overline{9,10}; \overline{1,2,4,6,7}; \overline{1,4,6}\}.$$

The cube representative of a block B_3 of cover σ_{x2x4x4} is $r(B_3) = 010$, since this block was obtained from blocks B_1 of β_{x2}, B_2 of β_{x4} and B_1 of β_{x5}. The representatives of these blocks are respectively 0, 1 and 0. Representatives of cover's blocks are always minterms.

Product of blankets $\beta_{x2}, \beta_{x4}, \beta_{x5}$ presented as blanket β_{x2x4x5} has empty and repetitive blocks removed.

$$\beta_{x2x4x5} = \beta_{x2} \bullet \beta_{x4} \bullet \beta_{x5} = \{B_1; B_2; B_3; B_4; B_5; B_6; B_7\} =$$
$$= \{\overline{5, 8, 10}; \overline{5, 6, 7, 8}; \overline{5, 6, 8}; \overline{2, 3, 9, 10}; \overline{9, 10}; \overline{1, 2, 4, 6, 7}; \overline{1, 4, 6}\}.$$

The relationship between blocks of β_{x2x4x5} and their cube representatives $r(B_i)$, relies on containment of block B_i in blocks of blankets used in product. Denoting blocks of β_{x2x4x5} as B_1 through B_7, we have $r(B_1) = 00-$. This is because $B_1 = \{5, 8, 10\}$ is included in the first blocks of β_{x2}, β_{x4} and in both blocks of β_{x5}. For $B_2 = \{5, 6, 7, 8\}$, we have: B_2 is included in the first block of β_{x2}, in the second block of β_{x4} and in first block of β_{x5}. Hence, $r(B_2) = 010$.

Information on the input patterns of a certain function F is delivered by the function's inputs and used by its outputs with precision to the blocks of the input and output blankets. Knowing the block of a certain blanket, one is able to distinguish the elements of this block from all other elements, but is unable to distinguish between elements of the given block. In this way, information in various points and streams of discrete information systems can be modeled using blankets.

2.3. FUNCTIONAL SERIAL DECOMPOSITION

The set X of function's input variable is partitioned into two subsets: *free variables* U and *bound variables* V, such that $U \cup V = X$. Assume that the input variables $x_1, ..., x_n$ have been relabeled in such way that:

$U = \{x_1, ..., x_r\}$ and
$V = \{x_{n-s+1}, ..., x_n\}$.

Consequently, for an n-tuple x, the first r components are denoted by x^U, and the last s components, by x^V.

Let F be a Boolean function, with $n > 0$ inputs and $m > 0$ outputs, and let (U, V) be as above. Assume that F is specified by a set F of the function's cubes. Let G be a function with s inputs and p outputs, and let H be a function with $r + p$ inputs and m outputs. The pair (G, H) represents a serial decomposition of F with respect to (U, V), if for every minterm b relevant to $F, G(b^V)$ is defined, $G(b^V) \in \{0, 1\}p.$, and $F(b) = H(b^U, G(b^V))$. G and H are called blocks of the decomposition (Fig. 1).

Theorem 1. Existence of the serial decomposition [1].
Let β_V, β_U, and β_F be blankets induced on the function's F input cubes by the input sub-sets V and U, and outputs of F, respectively.
If there exists a blanket β_G on the set of function F's input cubes such that $\beta_V \leq \beta_G$, and $\beta_U \bullet \beta_G \leq \beta_F$, then F has a serial decomposition with respect to (U, V).

Proof of Theorem 1 can be found in [1].

As follows from Theorem 1 the main task in constructing a serial decomposition of a function F with given sets U and V is to find a blanket β_G which satisfies the
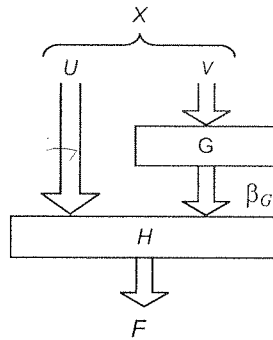
Fig. 1. Schematic representation of the serial decomposition

condition of the theorem. Since β_G must be $\geq \beta_V$, it is constructed by merging blocks of β_V as much as possible.

Two blocks B_i and B_j of blanket β_V are compatible (mergeable), if blanket γ_{ij} obtained from blanket β_V by merging B_i and B_j into a single block satisfies the second condition of Theorem 1, that is, if $\beta_U \bullet \gamma_{ij} \leq \beta_F$. Otherwise blocks B_i and B_j are incompatible (unmergeable). A subset δ of blocks of the blanket β_V is a compatible class of blocks if the blocks in δ are pairwise compatible. A compatible class is maximal if it is not contained in any other compatible class.

From the computational point of view, finding maximal compatible classes is equivalent to finding maximal cliques in a graph $\Gamma = (N, E)$, where the set N of vartices is the set of blocks of β_V and set E of edges is formed by set of compatible pairs.

The next step in the calculation of β_G is the selection of a set of maximal classes, with minimal cardinality, that covers all the blocks of β_V. The minimal cardinality ensures that the number of blocks of β_G, and hence the number of outputs of the function G, is as small as possible.

In certain heuristic strategies, both procedures (finding maximal compatible classes and then finding the minimal cover) can be reduced to the graph coloring problem. Calculating β_G corresponds to finding the minimal number k of colors for graph $\Gamma = (N, E)$.

Example 2. For the function from Table 1a specified by a set F of cubes numbered 1 through 10, consider a serial decomposition with $U = \{x_2, x_4, x_5\}$ and $V = \{x_1, x_3, x_7\}$.
We find

$$\beta_U = \beta_{x2x4x5} = \beta_{x2} \bullet \beta_{x4} \bullet \beta_{x5} = \{\overline{5, 8, 10}; \ \overline{5, 6, 7, 8}; \ \overline{2, 3, 10}; \ \overline{1, 2, 6, 7}; \ \overline{9, 10}; \ \overline{1, 4, 6}\},$$

$$\beta_V = \beta_{x1x3x7} = \beta_{x1} \bullet \beta_{x3} \bullet \beta_{x7} = \{\overline{1, 2, 3, 4, 8, 9}; \ \overline{3, 4, 9}; \overline{8}; \ \overline{5, 6}; \ \overline{3, 9}; \ \overline{7, 10}; \ \overline{6, 7, 10}\},$$

$$\beta_F = \beta_{y1} = \{\overline{1, 2, 3, 4, 5, 6, 7}; \ \overline{8, 9, 10}\}.$$

For $\beta_V = \left\{ \dfrac{B_1}{1, 2, 3, 4, 8, 9;} \dfrac{B_2}{3, 4, 9;} \dfrac{B_3}{8;} \dfrac{B_4}{5, 6;} \dfrac{B_5}{3, 9;} \dfrac{B_6}{7, 10;} \dfrac{B_7}{6, 7, 10} \right\}$ the following are the unmergeable pairs: (B_1, B_4), (B_1, B_6), (B_1, B_7), (B_2, B_6), (B_2, B_7), (B_3, B_4), (B_3, B_6), (B_3, B_7), (B_4, B_6), (B_4, B_7), (B_5, B_6) and (B_5, B_7). Using graph coloring procedure we find that three colors are needed here (Fig. 2).
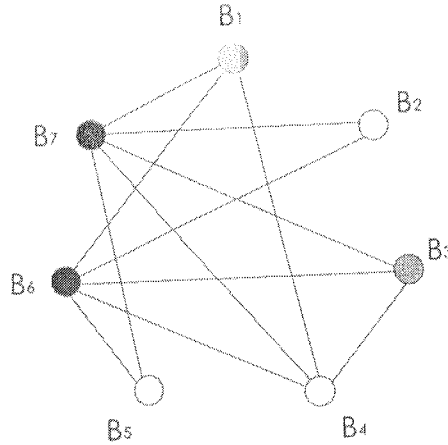

Fig. 2. Incompatibility graph of β_V's blocks for function y_1

Vertices B_1, B_3 are assigned one color, nodes B_2, B_4, B_5 are assigned second color and third color is assigned for nodes B_6, B_7. The sets of vertices assigned to different colors form the blocks of β_G.

$$\beta_G = \{\overline{1, 2, 3, 4, 8, 9} \; ; \; \overline{3, 4, 5, 6, 9} \; ; \; \overline{6, 7, 10}\}$$

It is easily verified that β_G satisfies the condition of Theorem 1. Thus function F has a serial decomposition with respect to (U, V).

Since β_G has 3 blocks, to encode blocks of this blanket two encoding bits g_1 and g_2 have to be used. Let us assume that we use the encoding

$$\beta_G = \left\{ \dfrac{00}{1, 2, 3, 4, 8, 9;} \dfrac{01}{3, 4, 5, 6, 9;} \dfrac{10}{6, 7, 10} \right\}.$$

To define a function G by a set of cubes we calculate all the cube representatives $r(B_i)$, assigned to each block B_i of β_V. The relationship between blocks of β_V and their cube representatives $r(B_i)$, relies on containment of block B_i in blocks of β_{xj} from $x_j \in V$.

Denoting blocks of β_V from Example 2 as B_1 through B_7, we have $r(B_1) = 000$. This is because $B_1 = \{1, 2, 3, 4, 8, 9\}$ is included in the first blocks of β_{x1}, β_{x3} and β_{x7}. For $B_2 = 3, 4, 9$, we have: B_2 is included in the first block of β_{x1}, in the first block β_{x3}

and in both blocks of β_{x7}. Hence, $r(B_2) = 00$–. Similarly $r(B_3) = 0$–0, $r(B_4) = 011$, $r(B_5) = $–$0$–, $r(B_6) = 11$–, $r(B_7) = 111$.

Finally, the value of function G is obtained on the basis of containment of blocks B_i in blocks of β_G. Block $B_1 = \{1, 2, 3, 4, 8, 9\}$ of blanket β_V is contained in block β_G that has been encoded with 00. Since $r(B_1) = 000$ we have $G(r(B_1)) = G(x_1 = 0, x_3 = 0, x_6 = 0) = 00$. Similarly, $G(r(B_3)) = 00$, $G(r(B_4)) = 01$, $G(r(B_6)) = 10$ and $G(r(B_7)) = 10$. However block $B_2 = \{3, 4, 9\}$ is contained in two blocks of β_G (one encoded "00" and second "01"). The representative "00–" of this block has nonempty product with representative "000" of B_1 and representative "0–0" of B_3, which was assigned output "00". To avoid conflicts we must subtract cubes "000" and "0–0" from cube "00–". The result is cube "001" that may be assigned output "01". The same applies to block B_5. The representative "–0–" of this block has nonempty product with representative of B_1 and B_3, which was assigned output "00". We must subtract cubes "000" and "0–0" from cube "0–0" and the result in form of cube "10–" may be assigned output "01".

Truth table of function G is presented in Table 2a.

To compute the cubes for function H we consider each block of the product $\beta_U \bullet \beta_G$. Their representatives are calculated in the same fashion. Finally, the outputs of H are calculated with respect to β_F (Table 2b).

Table 2

2a. Function G of the serial decomposition

	x_1	x_3	x_7	g_1	g_2
1	0	0	0	0	0
2	0	0	1	0	1
3	0	–	0	0	0
4	0	1	1	0	1
5	1	0	–	0	1
6	1	1	–	1	0
7	1	1	1	1	0

2b. Function H of the serial decomposition

	x_2	x_4	x_5	g_1	g_2	y_1
1	1	1	–	0	0	0
2	1	–	0	0	0	0
3	1	0	0	0	0	0
4	1	0	0	0	1	0
5	1	1	1	0	0	0
6	1	1	1	0	1	0
7	0	1	1	0	1	0
8	–	1	–	0	1	0
9	–	1	–	1	0	0
10	–	1	0	1	0	0
11	0	–	–	0	0	1
12	1	0	1	0	0	1
13	1	0	1	0	1	1
14	–	0	–	1	0	1

The process of functional decomposition consists of the following steps:

- the selection of an appropriate input support V for block G (input variable partitioning),
- the calculation of the blankets β_U, β_V and β_F,
- the construction of an appropriate multi-block blanket β_G (this corresponds to the construction of the multi-valued function of block G),
- the creation of the binary functions H and G by representing the multi-block blanket β_G as the product of a number of certain two-block blankets (this is equivalent to encoding the multi-valued function of block G defined by blanket β_G with a number of binary output variables).

3. BALANCED FUNCTIONAL DECOMPOSITION

Balanced decomposition relies on partitioning of a switching function with either parallel decomposition or serial decomposition applied at each phase of the synthesis process. Serial decomposition brakes the initial, possibly multi-output function into two cooperating blocks (Fig. 1). In the parallel decomposition, the set of output variables Y of a multi-output function F is partitioned into subsets, Y_g and Y_h, and the corresponding functions, G and H, are derived so that, for either of these two functions, the input support contains fewer variables than the set of input variables X of the original function F (Fig. 3). An objective of the parallel decomposition is to minimize the input support of G and H. Parallel decomposition brakes initial multi-output function into two independent block.
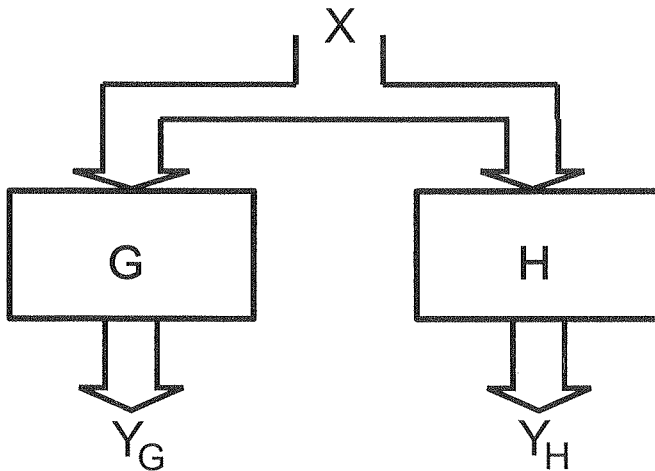

Fig. 3. Schematic representation of the parallel decomposition

The serial and parallel decompositions are intertwined in a top-down synthesis process to obtain the required topology. At each step, either parallel or serial decom-

position is performed, both characterized by operation input parameters. Intertwining of serial and parallel decomposition strategies opens up several interesting possibilities in multilevel decomposition. Experimental results show that the right balance between the two strategies and the choice of operation parameters severely influence the area and depth of the resultant network.

Table 3

Truth table of function F

type fr	0001001110 01
.i 10	0110000110 01
.o 2	1110110010 10
.p 25	0111100000 00
0101000000 00	0100011011 00
1110100100 00	0010111010 01
0010110000 10	0110001110 00
0101001000 10	0110110111 11
1110101101 01	0001001011 11
0100010101 01	1110001110 10
1100010001 00	0011001011 10
0011101110 01	0010011010 01
	e.

Example 3. The influence of the parallel decomposition on the final result of the FPGA-based mapping process will be explained with the function F given in Table 3, for which cells with 4 inputs and 1 output are assumed (this is the size of Altera's FLEX FPGAs).

As F is a ten-input, two-output function, in the first step of the decomposition either parallel or serial decomposition can be applied. If we first apply serial decomposition (Fig. 4), then the algorithm extracts function g with inputs x_1, x_3, x_4, and x_6, thus the next step deals with seven-input function h, for which again serial decomposition is assumed, now resulting in block g, with 4 inputs and 2 outputs (implemented by 2 Logic Cells – LC). It is worth noting that the obtained block g takes as its input variables x_0, x_2, x_5, and x_7, which, fortunately, belong to primary variables, and therefore the number of levels is not increased in this step. In the next step we apply parallel decomposition. Parallel decomposition generates two components, both with one output but 4 and 5 inputs, respectively. The first one forms a logic cell. The second component is subject to two-stage serial decomposition shown in Fig. 4. The obtained

network can be built of 7 (4 to 1) cells, where the number of levels in the critical path is 3.
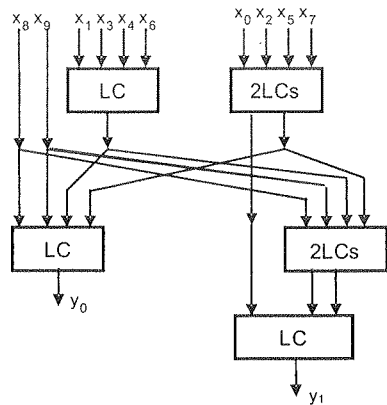

Fig. 4. Decomposition of function F where serial decomposition is performed first

Decomposition of the same function such that the parallel decomposition is applied in the first step leads to completely different structure (Fig. 5). Parallel decomposition applied directly to function F, generates two components both with 6 inputs and one output. Each of them is subject to two-stage serial decomposition. For the first component, a disjoint serial decomposition with four inputs and one output can be applied. The second component can be decomposed serially as well, however with the number of outputs of the extracted block G equals to two. Therefore, to minimize the total number of components, a non-disjoint decomposition strategy can be applied.
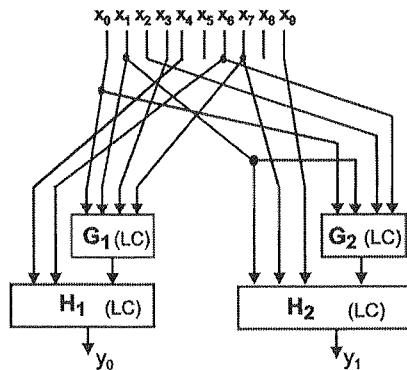

Fig. 5. Decomposition of function F where parallel decomposition is performed at first

The main consequence of application of parallel decomposition in balanced decomposition process is inability to construct common sub-functions for outputs belonging to different subsets Y_g and Y_h anymore. Since parallel decomposition brakes initial

multi-output function into two independent block, after this kind decompositions is applied blocks G and H are further synthesized in separate processes. This limits in grate degree possible synthesis strategies influencing quality of obtained result.

4. DECOMPOSITION OF FUNCTION SETS

There are many methods for decomposition of multi-output Boolean functions. These methods are mostly developed for cube or BDD-based representation of logic circuit. However, the common characteristics of these method is requirement, that decomposed multi-output Boolean function is represented by single truth table in case of cube representation or single BDD. In case of BDD, this causes the necessity of development of special techniques that allow efficiently representing many functions with single BDD. Since Boolean functions can be not fully specified (can have *don't cares*) and each function can depend on different input variables it is difficult to find efficient solution to this problem [11].

In case of decomposition methods based on cube representation it is easier, since to represent multi-output Boolean function espresso format can be used. However cube representation can be very inefficient when many functions which depend on different input variable have to be decomposed. Here a method for decomposition of Boolean function sets based on cube representation is presented.

The logic multi-output circuit of n input variables X can be represented as a set of Boolean functions $f_1(X_1), \ldots, f_m(X_m)$, where X_i is sub-set of X. In general case each single function may depend on different subset of input variables and can be described by different set of cubes. Application of serial functional decomposition based on blanket calculus is impossible in this case. We propose the modified serial functional decomposition algorithm that allows decomposing function sets described by separate truth tables, even if each single function depends on different subset of input variables and truth tables consist of different sets of cubes.

When decomposing a set of Boolean functions $f_1, \ldots, f_m$ it is necessary to find such sub-function G that satisfies decomposition condition for each function:

$$f_1 = h_1(U_1, G(V)),$$
$$f_2 = h_2(U_2, G(V)),$$
$$\ldots$$
$$f_m = h_m(U_m, G(V)).$$

Decomposition process consists of few steps. First an appropriate input support V has to be selected for block G (input variable partitioning). In case function are specified by different truth tables such set V has to be selected that is subset of input variables of each decomposed function (V must consist variables common to all decomposed function). Next for each function f_i blankets β_{Ui}, β_{Vi} and β_{fi} have to be computed. Since truth tables of each function may consist different cubes, blankets β_{Ui}, β_{Vi} and β_{fi} for each function may be different. Having this done an appropriate multi-block blankets β_{Gi} have to be constructed. In functional decomposition algorithm construction

of this blanket corresponds to the construction of the multi-valued function of block G. Since we want to find the decomposition with the same block G for all functions f_i, such blankets β_{Gi} should be created that satisfy decomposition condition for all these functions and correspond to the same function G at the same time.

Blanket β_G is constructed by merging blocks of β_V as much as possible. In this process graph is used, where each vertex corresponds to one of blocks of blanket β_V. Two vertices are connected with an edge if corresponding blocks are unmergeable. Next graph is colored and blocks β_G are created by merging blocks of blanket β_V that have been assigned this same color and binary code is assigned to each block. Representatives of blocks of blanket β_V and codes assigned to blocks of blanket β_G are used to create function G (Example 2).

To create block G satisfying decomposition condition for all decomposed functions $f_1, \ldots, f_m$ such graph should be created that represents mergeability of blocks of all blankets β_{Vi}. However, in general case each function can be described by different set of cubes. This implies that blankets β_{Vi} for each function would be different (number of blocks may be different and representatives of blocks may be different) and common graph cannot be created.

Example 4. For the function from Table 1b consider a serial decomposition with $U = \{x_6, x_8\}$ and $V = \{x_1, x_3, x_7\}$.

We find

$$\beta_{x1} = \{\overline{1, 3, 4, 5, 7}; \overline{2, 5, 6, 7}\},$$

$$\beta_{x3} = \{\overline{3, 4, 5}; \overline{1, 2, 3, 4, 6, 7}\},$$

$$\beta_{x6} = \{\overline{1, 2, 4, 5, 6}; \overline{3, 4, 5, 7}\},$$

$$\beta_{x7} = \{\overline{2, 3, 4, 6}; \overline{1, 2, 3, 5, 6, 7}\},$$

$$\beta_{x8} = \{\overline{1, 2, 3, 4, 5, 7}; \overline{1, 3, 4, 6, 7}\},$$

$$\beta_{y2} = \{\overline{1, 2}; \overline{3, 4, 5, 6, 7}\}.$$

Then

$$\beta_U = \{\overline{1, 2, 4, 5}; \overline{1, 4, 6}; \overline{3, 4, 5, 7}; \overline{3, 4, 7}\},$$

$$\beta_V = \{\overline{3, 5}; \overline{3, 4}; \overline{1, 3, 7}; \overline{5}; \overline{2, 6}; \overline{2, 6, 7}\}.$$

Using graph coloring procedure we find that three colors are needed thus the blanket β_G will have 3 blocks (Fig. 6). Let us assume that we use the encoding

$$\beta_G = \left\{ \frac{00}{3, 4, 5}; \frac{01}{1, 3, 7}; \frac{10}{2, 6, 7} \right\}.$$

The function G will have 3 input variables and 2 output variables (Tab. 4).

Although decompositions of function from Table 1a and b allow constructing block G with the same input variables (Example 2 and 4) using decomposition method based
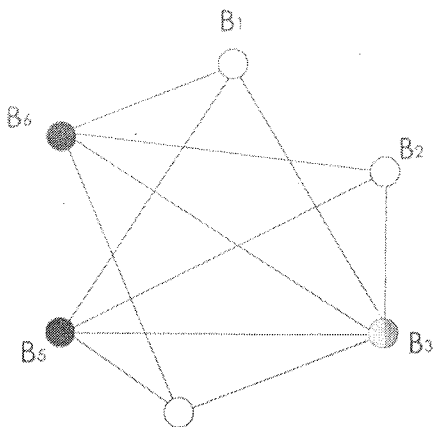
Fig. 6. Incompatibility graph of β_V's blocks for function y_2

Table 4

Function G of the serial decomposition of function from Table 1b

	x_1	x_3	x_6	g_1	g_2
1	0	–	0	0	0
2	0	1	1	0	1
3	–	0	1	0	0
4	1	1	–	1	0

on blanket calculus it is impossible to construct common block G. This is because graphs from Fig. 2 and Fig. 6 represent completely different information since have different number of vertices and their vertices represent blocks with different representatives, i.e vertex B_1 in Fig. 2 corresponds to block for which the representative is $r(B_1) = 000$, while vertex B_1 in Fig. 6 corresponds to block for which the representative is $r(B_1) = 001$.

To solve the problem the decomposition methods is proposed that instead of computing blanket β_V computes the cover σ_V. Representatives of blanket's blocks are cubes because blankets have empty or repetitive blocks removed. This is not a case when covers are used. In covers all blocks are preserved, therefore representatives of blocks are minterms. This means that covers are not sensitive to different cube representations of function if block representatives are taken into account. Covers induced by k variables always have 2^k blocks and representative of each block is one of 2^k minterms of size k. If only two covers σ_{V1} and σ_{V2} are induced by the same input variables representatives of corresponding blocks of these covers are similar, although they may be induced on different sets of cubes.

This allows easy computation of decomposition for sets of Boolean function. Algorithm is presented in Figure 7.

```
Input:     truth tables of functions f₁,…,fₘ

Output:    function G being common sub-function of all
           functions f₁,…,fₘ

    (1) select set V of input variables common for all
        functions f₁,…,fₘ

    (2) for each function fᵢ compute blankets βUᵢ, βfᵢ and
        cover σVᵢ

    (3) for each function fᵢ create incompatibility graph
        Γᵢ = (N, E)

    (4) for each graph Γᵢ = (N, E) find such colouring that
        assigns the same colour to corresponding vertices
        of each graph

    (5) encode each colour with binary variables

    (6) construct function G by using representatives of
        blocks corresponding to vertices as input patterns
        and codes assigned to vertices' colours as output
        patterns
```

Fig. 7. Functional decomposition algorithm for set of Boolean functions

In proposed algorithm first set V of input variables is chosen. Since in decomposition process sub-function G common to all functions has to be computed, variables from set V have to be common to all functions. Next blankets β_{Ui}, β_{fi} and cover σ_{Vi} is computed for all functions. Covers σ_{Vi} are induced by the same input variables, thus each of them has the same number of blocks and representatives of corresponding blocks of these covers are similar. This allows creating a graph $\Gamma_i = (N, E)$ for each function f_i with the same number of vertices. The most important step in this algorithm is introduction of such coloring that assigns the same color to corresponding nodes of all graphs Γ_i. Ones, such coloring is found function G is constructed in similar way like in case of functional decomposition algorithm based on blanket calculus (Example 2). Each color is assigned a binary code. Next truth table of function G is created by using representatives of cover σ_{Vi}'s blocks as input patterns and codes assigned to vertices' colors as output patterns. The presented algorithm is demonstrated in Example 4.

Example 5. For the functions from Table 1a and b consider a serial decomposition with bound set consisting variables x_1, x_3, x_7. For function y_1 we have:

$$U_1 = \{x_2, x_4, x_5\},$$
$$V_1 = \{x_1, x_3, x_7\},$$
$$\beta_{U1} = \beta_{x2x4x5} = \{\overline{5, 8, 10}; \overline{5, 6, 7, 8}; \overline{2, 3, 10}; \overline{1, 2, 6, 7}; \overline{9, 10}; \overline{1, 4, 6}\},$$
$$\sigma_{V1} = \sigma_{x1x3x7} = \{\overline{1, 2, 3, 4, 8, 9}; \overline{3, 4, 9}; \overline{8}; \overline{5, 6}; \overline{3, 9}; \overline{3, 9}; \overline{7, 10}; \overline{6, 7, 10}\}.$$

For function y_2 we have:

$U_2 = \{x_6,\ x_8\}$,

$V_2 = \{x_1,\ x_3,\ x_7\}$,

$\beta_{U2} = \beta_{x6x8} = \{\overline{1,\ 2,\ 4,\ 5};\ \overline{1,\ 4,\ 6};\ \overline{3,\ 4,\ 5,\ 7};\ \overline{3,\ 4,\ 7}\}$,

$\sigma_{V2} = \sigma_{x1x3x7} = \{\overline{3,\ 4};\ \overline{3,\ 5};\ \overline{3,\ 4};\ \overline{1,\ 3,\ 7};\ \overline{};\ \overline{5};\ \overline{2,\ 6};\ \overline{2,\ 6,\ 7}\}$.

Covers σ_{V1} and σ_{V2} are induced by the same input variables. Although they are induced on different sets of cubes, representatives of corresponding blocks of these covers are similar. Representative of first block of cover σ_{V1} as well as σ_{V2} is $r(B_1) = 000$. Similarly representative of last block of both covers is $r(B_7) = 111$.
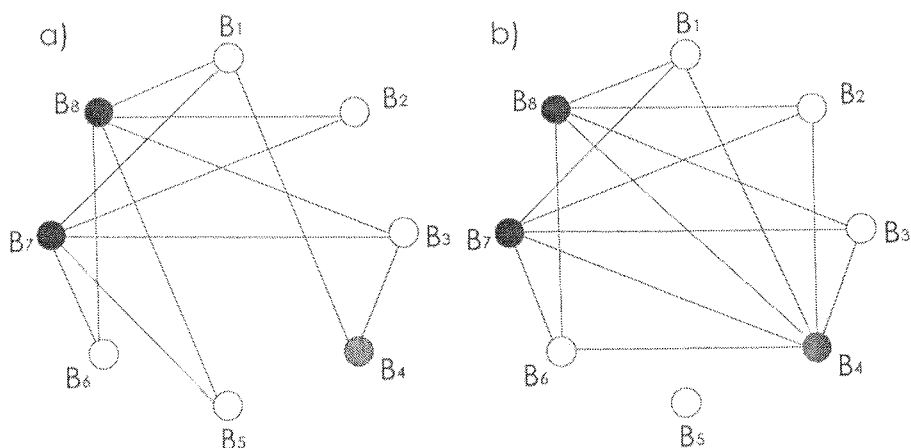
Fig. 8. Incompatibility graph of blocks σ_V for: a) function y_1, b) function y_2

Figures 8a and b present incompatibility graphs of σ_V's blocks for function y_1 and y_2 respectively. Such coloring of these graphs is introduced that corresponding vertices have the same color assigned. Three colors are needed, thus two encoding binary variables are necessary. Appropriate β_G are as follows:

$$\beta_{G1} = \left\{ \frac{00}{1,2,3,4,8,9};\ \frac{01}{3,4,5,6,9};\ \frac{10}{6,7,10} \right\}.$$

$$\beta_{G2} = \left\{ \frac{00}{3,4,5};\ \frac{01}{1,3,7};\ \frac{10}{2,6,7} \right\}.$$

Ones such coloring is found functions G can be constructed. Vertices B_1 of both graphs correspond to blocks of covers σ_{V1} and σ_{V2} for which representatives are $r(B_1)$ = 000. These blocks are contained in blocks of blankets β_{G1} and β_{G2} that have been encoded with 00. Thus the first row in function G for both decomposed function will have input pattern 000 and output pattern 00. It is easy to verify that all rows will be the same, thus the same function G has been found that satisfies decomposition condition for both decomposed function:

$$y_1 = h_1(x_2, x_4, x_5, G(x_1, x_3, x_7)),$$

$$y_2 = h_2(x_6, x_8, G(x_1, x_3, x_7)).$$

Truth table of function G is presented in Table 5.

Table 5

Function G of the serial decomposition of function from Table 1a and b

	x_1	x_3	x_6	g_1	g_2
1	0	0	0	0	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	1	1	0	1
5	1	0	0	0	0
6	1	0	1	0	0
7	1	1	0	1	0
8	1	1	1	1	0

5. CONCLUSIONS

Logic circuits usually have many outputs. There are many methods for decomposition of multi-output Boolean functions. These methods are mostly developed for cube or BDD-based representation of logic circuit. However, the common characteristics of these method is requirement, that decomposed multi-output Boolean function is represented by single truth table in case of cube representation or single BDD.

The proposed decomposition method allows decomposing set of Boolean functions described by separate truth tables. Decomposed functions may depend on different input variables and can be described by different set of cubes. Application of covers in decomposition algorithm allows efficiently create sub-functions common for all decomposed functions. This allows the designer to apply a much wider range of synthesis strategies.

1. J. A. of Mt
2. S. C. Basea 1996)
3. G. D
4. Y. L menta
5. T. Ł cation 3(3-4)
6. M. N Algori Szcze
7. M. P posiur
8. M. P Facult
9. M. R port S Archit
10. P. S a in Lo Curre
11. T. S a Multij
12. T. S a
13. C. S Publis
14. H. S Machi Applic
15. M. V Decor Intelli
16. B. W compc Desigi

6. ACKNOWLEDGEMENTS

7. REFERENCES

1. J. A. B r z o z o w s k i, T. Ł u b a: *Decomposition of Boolean Functions Specified by Cubes.* Journal of Multi-Valued Logic and Soft Computing 9 (2003): 377-417.
2. S. C. C h a n g, M. M a r e k - S a d o w s k a, T. T. H w a n g: *Technology Mapping for TLU FPGAs Based on Decomposition of Binary Decision Diagrams.* IEEE Trans. on CAD 15, no. 10 (October 1996): 1226-1236.
3. G. D e M i c h e l i: *Synthesis and Optimization of Digital Circuits.* McGraw-Hill, 1994.
4. Y. L a i, K. R. P a n, M. P e d r a m: *OBDD-based function decomposition: algorithms and implementation.* IEEE Trans. on CAD of Integrated Circuits and Systems 15, no. 8 (1996): 977-990.
5. T. Ł u b a, H. S e l v a r a j: *General Approach to Boolean Function Decomposition and its Applications in FPGA-based Synthesis.* VLSI Design, Special Issue on Decompositions in VLSI Design 3(3-4) (1995): 289-300.
6. M. N o w i c k a, T. Ł u b a, M. R a w s k i: *FPGA-Based Decomposition of Boolean Functions. Algorithms and Implementation.* Sixth International Conference on Advanced Computer Systems. Szczecin, 1999. 502-509.
7. M. P e r k o w s k i, et al.: *Decomposition of Multiple-Valued Relations.* Proc. of International Symposium on Multiple-Valued Logic. Antigonish, 1997. 13-18.
8. M. P e r k o w s k i: *A Survey of Literature on Function Decomposition.* Final Report for Summer Faculty Research Program, Air Force Office of Scientific Research, Wright Laboratory, 1994.
9. M. R a w s k i, L. J ó ź w i a k, T. Ł u b a: *Functional Decomposition with an Efficient Input Support Selection for Sub-functions Based on Information Relationship Measures.* Journal of Systems Architecture 47 (2001): 137-155.
10. P. S a p i e c h a, H. S e l v a r a j, M. P l e b a n: *Decomposition of Boolean Relations and Functions in Logic Synthesis and Data Analysis.* Edited by Wojciech Ziarko and Y Y Yao. Rough Sets and Current Trends in Computing. Springer, 2000. 487-494.
11. T. S a s a o, M. M a t s u u r a, Y. I g u c h i, S. N a g a y a m a: *Compact BDD Representations for Multiple-Output Functions and Their Application.* ISMVL. 2001. 207-212.
12. T. S a s a o, M. F u j i t a: *Representations of Discrete Functions.* Kluwer Academic Publishers, 1996.
13. C. S c h o l l: *Functional Decomposition with Application to FPGA Synthesis.* Kluwer Academic Publisher, 2001.
14. H. S e l v a r a j, P. S a p i e c h a, T. Ł u b a: *Functional Decomposition and Its Applications in Machine Learning and Neural Networks.* International Journal of Computational Intelligence and Applications 1, no. 3 (2001): 259-271.
15. M. V e n k a t e s a n, H. S e l v a r a j, R. B i g n a l l: *Character Recognition using Functional Decomposition.* Edited by H. Selvaraj and B. Verma. International Conference on Computational Intelligence and Multimedia Applications. Singapore, 1998. 721-726.
16. B. W u r t h, U. S c h l i c h t m a n n, K. E c k l, K. A n t r e i c h: *Functional multiple-output decomposition with application to technology mapping for lookup table-based FPGAs.* ACM Trans. Design Autom. Electr. Syst. 4, no. 3 (1999): 313-350.

Sche

virt
for
Ma
Sele
and
Per
and
and

Key

Virt
ted at an
Each arr
to its det
[1]. How
output p
thm is ne
must reso
(QoS) re
Queuing

Scheduling algorithms for Virtual Output Queuing switches

ANNA BARANOWSKA, WOJCIECH KABACIŃSKI

Katedra Sieci Telekomunikacyjnych i Komputerowych
Politechnika Poznańska
ul. Polanka 3, 60-965 Poznań
anna.baranowska, wojciech.kabacinski@et.put.poznan.pl

In this article we have given an overview of different scheduling algorithms for virtual output queuing switches. We have also proposed three new scheduling algorithms for VOQ switches called Hierarchical Round-Robin Matching ($HRRM$), Matching Matching with Random Selection ($MMRS$) and Maximal Matching with Round-Robin Selection ($MMRRS$). Evaluation of the proposed algorithms was performed under Bernoulli and bursty packet arrivals with uniform and non-uniform distributions to output ports. Performance evaluation of the proposed algorithm under different traffic models is given and compared with other well-known algorithms. The simulation results show that MMRRS and HRRM achieves 100% throughput and low MTD for high loads.

Keywords: packet scheduling algorithms, Virtual Output Queuing switches, packet switching

1. INTRODUCTION

Virtual Output Queuing (VOQ) switches, in which one queue per output is allocated at an input port, were proposed to remove head-of-line (HOL) blocking problem. Each arriving packet is classified and then queued in an appropriate VOQ according to its determined destination output port (Figure 1). VOQ was first proposed in Tamir [1]. However, since each input port can transmit only one packet in a time slot and one output port can receive also only one packet in a time slot, a proper scheduling algorithm is needed for choosing queued cells for transmission. Such a scheduling algorithm must resolve output contention swiftly, provide high throughput, meet quality-of-service (QoS) requirements, and should be easily implemented in hardware. Virtual Output Queuing is widely used in fixed-length high-speed switches.
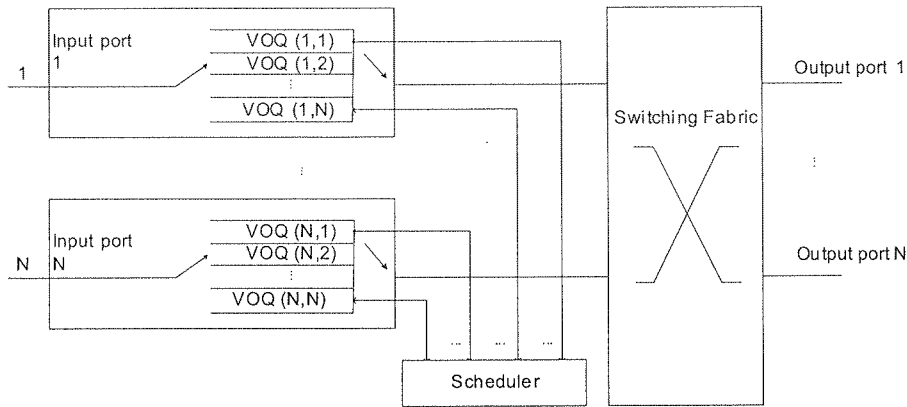
Fig. 1. General architecture of the switch

Many algorithms have been proposed so far to schedule packets for transmission in VOQ switches. This article introduces the new algorithms for VOQ switches called Hierarchical Round-Robin Matching (*HRRM*) and Maximal Matching with Round--Robin Selection (*MMRRS*), proposed recently by us [15], [3]. We also analyze the throughput and the Mean Time Delay (*MTD*) of well known scheduling algorithms such as: Parallel Iterative Matching (*PIM*) [4], Iterative Round-Robin Matching (*iRRM*) [5], iSLIP [6]. In addition, this paper shows that MMRRS and HRRM achieve 100% throughput. Simulations show, that for each load the HRRM algorithm is better than other known algorithms.

The paper is organized as follows. In section 2 and 3 the scheduling algorithms are described. In the next section performance evaluation of the proposed algorithms is given and compared with other algorithms. At the end some conclusions are given.

2. THE SWITCH ARCHITECTURE

The general architecture of the switch considered in the paper is shown in Figure 1. At each input, cells are stored in the respective VOQ, one separate queue is maintained at each input for each output, and denoted by VOQ (i, j), where i is the input port number and j is the output port number, $1 \leq i \leq N, 1 \leq j \leq N$. The total number of input queues in the switch is N^2. Packets transmitted through the switching fabric are fixed-length packets, and the switching fabric is nonblocking. Packets at an input port of the switch are divided into fixed length packets and next transmitted through the switch and put back together before they leave the switch. One packet occupies one time slot. In one input port only one packet can arrive in one time slot. In each time slot the scheduler selects packets for transmission in the next time slot, i.e. it finds one-to-one matching of non-empty VOQs. The average cell arrival rate at input i and directed to output j (with $i; j = 0; 1; \ldots; N - 1$) is denoted by λ_{ij}; the average traffic pattern is $\Delta = [\lambda_{ij}]$, then a schedulable flow is [7]

port 1

port N

$$\sum_{i=1}^{N} \lambda_{i,j} \leq 1 \tag{1}$$

for all j; and

$$\sum_{j=1}^{N} \lambda_{i,j} \leq 1 \tag{2}$$

for all i.

The centralized scheduler is used to resolve output contention. Main factor in achieving high performance of VOQ switches is the scheduling algorithm. The architecture of the scheduler and scheduling algorithms will be considered in the next section.

nsmission
hes called
th Round-
nalyze the
algorithms
ng ($iRRM$)
eve 100%
better than

algorithms
algorithms
are given.

3. SCHEDULING ALGORITHMS

Scheduling algorithms for VOQ switches presented in this article can be divided into two groups: dynamic and static. The first group of algorithms is called dynamic because pointer upgrading depends on the number of chosen ports. The following algorithms belong to this category: Parallel Iterative Matching (PIM) [4], Iterative Round-Robin Matching ($iRRM$) [5], iSLIP [6], FCFS In Round-Robin Matching ($FIRM$) [8]. Static algorithms belonging to the other group are based on desynchronization of particular arbiter pointers, which influences the improvement of particular algorithm's efficiency. In each time slot pointer value is increased by one (modulo N) regardless off the number of the chosen ports. In this section we introduce static algorithms: Single Static Round-Robin ($SSRR$), Double Static Round-Robin ($DSRR$), Rotating Double Static Round-Robin ($RDSRR$). Static algorithms were developed by Jiang and Hamdi [9]. The basic idea is to keep full pointer-desynchronization in the output and/or input arbiters.

n Figure 1.
maintained
input port
number of
fabric are
input port
arough the
cupies one
each time
.e. it finds
input i and
rage traffic

3.1. DYNAMIC ALGORITHMS

The PIM algorithm

Parallel Iterative Matching (PIM) is the basic matching algorithm. PIM was developed by DEC Systems Research Center for 16-port, 1 Gbps AN2 switch [4]. PIM uses random selection to solve the contention in inputs and outputs. [10] Furthermore, it is an iterative algorithm. Only those inputs and outputs that are not matched at the end of previous iterations will be eligible to participate in the next matching iteration. All inputs and outputs are initially unmatched. Each iteration consists of three steps and operates in parallel at inputs and outputs (Figure 2a).

The steps are:

1. *Request* – Each unmatched input sends a request to every output for which it has queued cells.
2. *Grant* – If an unmatched output receives any request, it grants to one by randomly selecting a request uniformly over all requests.
3. *Accept* – If an input receives the grant, it accepts one by selecting an output randomly among those that granted to this output.

In each iteration of random matching, a minimum average 75% of the remaining possible connections are matched or eliminated [4]. No input-output connection is starved of service, it means that no memory or state is used to keep track of how recently a connection was made in the past. The disadvantage of PIM algorithm is that it is expensive and difficult to implement in hardware and second. Each scheduler must make a random selection among the members of a varying set, it can lead to unfairness between connections. The propability that an input will remain ungranted is $\left(\dfrac{N-1}{N}\right)^N$, hence as N increases, the throughput tends to $1 - \dfrac{1}{e} \approx 63\%$. [11]
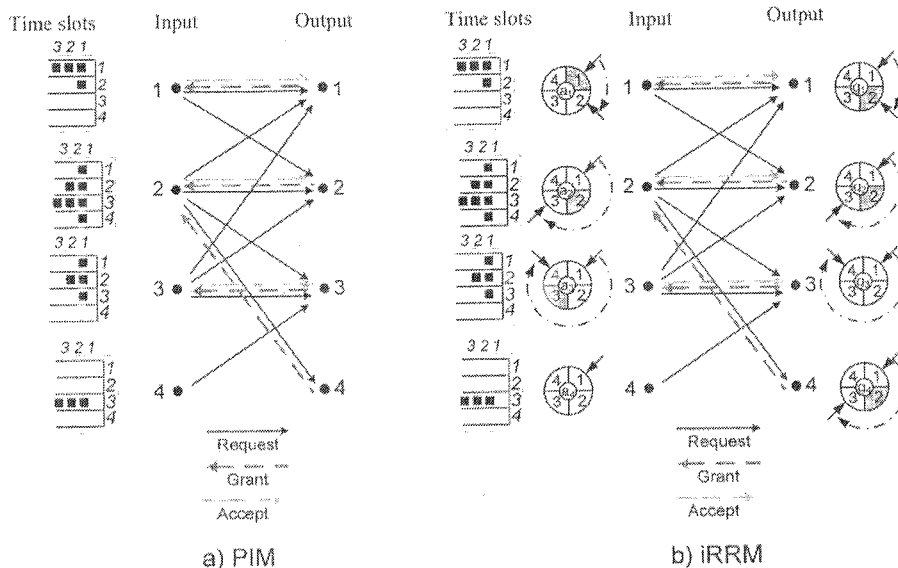

Fig. 2. Example of scheduling algorithms in a 4×4 switch a) PIM, b) iRRM

The iRRM algorithm

The round robin algorithm is destined to overcome the complexity and unfairness in PIM algorithm. Iterative Round Robin Matching (*iRRM*) overcomes the unfairness of random matching by granting requests and accepting grants according to a round robin priority scheme [5, 11]. Each arbiter maintains a pointer pointing at the port that

has the highest priority. Such a pointer is called the accept pointer a_i at input i or the grant pointer g_j at output j. The pointers actualization in each iteration is the same as in the first step. Here are the three steps that each iteration must follow:

1. *Request* – Each unmatched input sends a request to every output for which it has queued cells.
2. *Grant* – If an output receives any request, it chooses the one that appears next in a fixed, round-robin schedule starting from the highest priority element. The output notifies each input whether its request was granted. The pointer g_i is incremented (modulo N) to one location beyond the granted input. If no request is received, the pointer stays unchanged.
3. *Accept* – If an input receives a grant, it accepts the one that appears next in a fixed, round-robin schedule starting from the highest priority element. The pointer a_i is incremented (modulo N) to one location beyond the accepted output. If no request is received, the pointer stays unchanged.

Figure 2b shows an example of one iteration of the three phases iRRM algorithm. States of HOL cells in time slots 1, 2 and 3 are given, where black square, denotes that the given HOL cell has a packet for transmission. a_1, a_2, a_3, a_4 are arbiter pointers for input 1, 2 ,3 and 4, and g_1, g_2, g_3, g_4 are arbiter pointers for outputs 1, 2, 3 and 4. We assume that initially all arbiter are set to 1. In this example the inputs request transmission to all outputs that they have a cell destined for. Input 1 has HOL cells to outputs 1 and 2, input 2 has cells to all outputs, input 3 has HOL cells for outputs 1, 2, 3, and input 4 has one cell for output 4. In step 2, among all received request, each grant arbiter selects the requesting input that is nearest to the one currently pointed to. Output 1 chooses input 1, output 2 chooses input 2, output 3 chooses input 3, and output 4 chooses input 2. Each grant pointer move one position beyond the selected one. In this case, $g_1 = 2, g_2 = 3, g_3 = 4, g_4 = 3$. In step 3, each accept pointer decides which grant is accepted. Input 1 accepts output 1. For input 2, a_2 is set to 1, which means the packet should be sent to output 1. However, since there is no any packet to output 1, a_2 chooses the packet for output 2. Input 3 accepts output 3. The pointer a_i is incremented (modulo N) to one location beyond the accepted output, then $a_1 = 2, a_2 = 3, a_3 = 4, a_4 = 1$.

The reason for the poor performance of iRRM lies for updating the pointers at the output arbiters. The main disadvantage of iRRM algorithm is blocking the cell in some traffic patterns, when they are waiting for transmission from the input port to the output port. The reason is to grant the output in the way, that discriminates the particular input. [11]

The iSLIP algorithm

The iSLIP, like PIM, is an iterative algorithm and was first described in [11]. The iSLIP is a starvation-free algorithm, and it works in a similar way to iRRM, with a small but important difference: in step 2 mentioned above, the pointer to the highest

priority element is updated only if the grant is accepted by the input. The iSLIP changes the time of updating the priority pointer for single iteration and multiple iterations.

Figure 3a shows an example of iSLIP arbitration algorithm. Input 1 has one or more cells for outputs 1, 2, input 2 has one or more cells for all outputs, input 3 has one or more cells for outputs 1, 2, 3, input 4 has one cell for output 3. In the request step, each input sends the request to each output. Since round-robin pointer g1 is pointing to 1, output arbiter 1 sends a grant to input 1 and updates its pointer to 2 after the grant is granted by input 1. To consider output 2 in the grant step, since its round-robin pointer g_2 is pointing to 4, output arbiter doesn't receive request from input 4 that's why input 2 sends the grant to input 1. Since round-robin pointer g_3 is pointing to 2, output arbiter 3 sends a grant to input 2 and updates its pointer to 3 after the grant is granted by input 2. For output 4, g_4 is set to 3, which means the packet should be sent to output 3. However since there is no request from inputs 3, 4, 1, output 4 chooses input 2, and sends grant to input 2. At the end of the time slot a_1 is incremented to 2, and a_2 is incremented to 4.
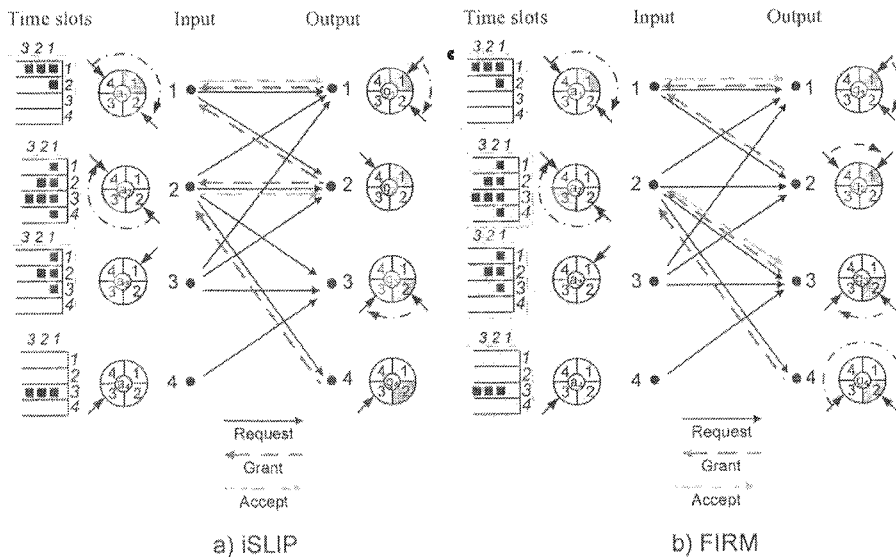

Fig. 3. Example of scheduling algorithms in a 4×4 switch a) iSLIP, b) FIRM

The FIRM algorithm

The First Come, First Server algorithm (*FIRM*) was first proposed in [8]. The specification of FIRM algorithm is as follows:

1. *Request* – All inputs send their requests to the outputs;
2. *Grant* – Every output sends a grant to the request that appears next in a fixed, round-robin schedule. If the grant is accepted in Step 3, the round-robin pointer g_k

is incremented (modulo N) to one location beyond the granted input. If the grant is not accepted, pointer g_k is placed to the granted input.

3. *Accept* – Every input accepts the grant that appears next in a fixed, round-robin schedule. The pointer a_i is incremented (modulo N) to one location beyond the accepted output.

This algorithm is the modification of iRRM and iSLIP algorithms. We can see the difference in updating their pointers [8, 9]. The pointers are updated in all iterations in contrast to the iSLIP algorithm. FIRM has better desynchronization effect than iSLIP. The difference in the update of the g_k pointers. If the grant is not accepted, pointer g_k is placed to the Grant'ed input. In the FIRM algorithm the same cells are sent as in the iSLIP algorithm, except that g_2 and g_4 pointers are placed to this request for which the Grant has not been accepted (Figure 3b).

The DRRM algorithm

The Dual Round-Robin Matching algorithm is just like iSLIP and FIRM the next modification of iRRM algorithm. DRRM adopts the unicast style to issue the requests, and it replaces iSLIP's three steps with only two: Request and Grant. The input ports send requests to the outputs, so the situation, when the one output is receiving more then one Grant is not possible. Then, there is no need to decide which output the acceptance signal should be send to. Therefore this step is then removed [12, 13]. In one matching cycle, DRRM has one less operational step than iSLIP, and less data exchange is needed between inputs and outputs. This scheme follows the request – grant steps and these are:

1. *Request – Each* input sends an output request corresponding to the first nonempty VOQ in a fixed round-robin order, starting from the current position of the pointer. The pointer remains at that nonempty VOQ if the selected output is not granted in step 2. The pointer of the input arbiter is incremented by one location beyond the selected output if, and only if, the request is granted in *Step 2*.

2. *Grant* – If an output receives one or more requests, it chooses the one that appears next in a fixed round-robin schedule starting from the current position of the pointer. The output notifies each requesting input whether or not its request was granted. The pointer of the output arbiter is incremented to one location beyond the granted input. If there are no requests, the pointer remains where it is.

Figure 4a shows an example of the DRRM arbitration algorithm. In a request phase, each input chooses a VOQ and sends a request to an output arbiter. Assume input 1 has HOL cells destined for both outputs 1 and 2. For input 1 a_1 is set to 4, which means the packet should be sent to output 4. However since there isn't any packet to output 4, a_1 chooses the packet for output 1 and updates its pointer to 2 after the request is granted by output 1. Input 2 has cells destined for all outputs. Since round-robin pointer a_2 is pointing to 2, input arbiter 2 sends a request to output 2

and updates its pointer to 3 after the request is granted by output 2. Input 3 has cells destined for 1, 2 and 3 outputs. Input 3 sends request to output 1. Input 4 has cells only destined for output 3 and updates its pointer to 4 after the request is granted by output 3. To consider output 1 in the grant phase, since its round-robin pointer g_1 is pointing to 1, output arbiter 1 grants input 1 and updates its pointer to 2. Output 2 receives request only from input 2, and sends grant to input 2 and the pointer is set to 3. The same situation takes place in output 3. The pointer g_3 is set to 1.
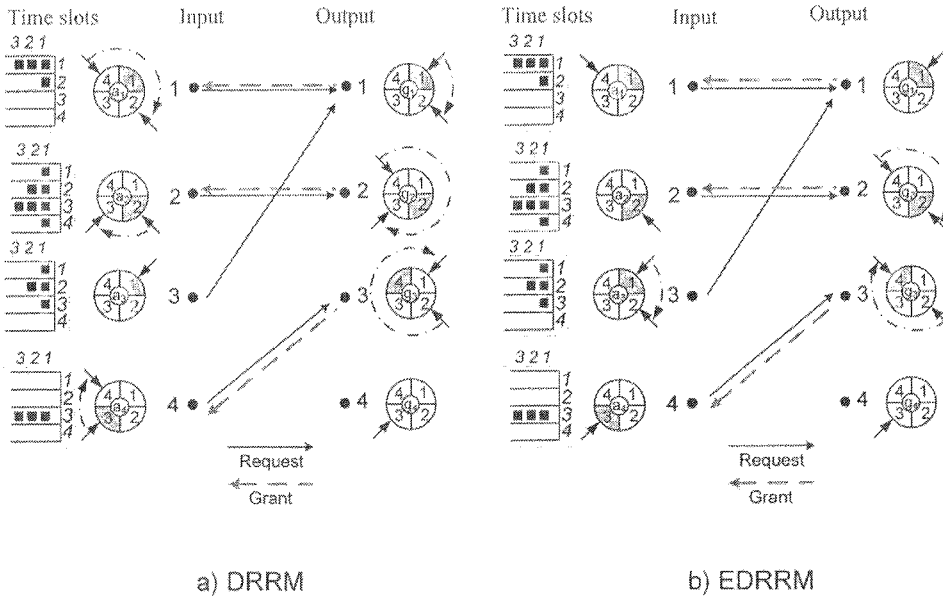

a) DRRM   b) EDRRM

Fig. 4. Example of scheduling algorithms in a 4×4 switch a) DRRM, b) EDRRM

The EDRRM algorithm

In the Exhaustive Service DRRM algorithm the pointers of inputs and outputs are updated in a different way from DRRM. EDRRM was proposed in [13] to improve the performance under bursty and nonuniform traffic. In EDRRM, a VOQ will keep being served until it is exhausted. The algorithm has two steps:

1. *Request* – Each input moves its pointer to the first nonempty VOQ in a fixed round-robin order, starting from the current position of the pointer, and sends a request to the output corresponding to the VOQ. The pointer of the input arbiter is incremented by one location beyond the selected output if the request is not granted in Step 2, or if the request is granted after one cell is served this VOQ becomes empty. Otherwise, the pointer remains at that nonempty VOQ.

2. *Grant* – If an output, receives one or more requests, it chooses the one that appears next in a fixed round-robin schedule starting from the current position of the pointer. The output notifies each requesting input whether or not its request as granted.

Figure 4b shows an example of EDRRM algorithm. Arbiters are setting initially the same as DRRM. At the beginning of time slot 1 a_1 points to output 4, while g_1 points to input 1. Input 1 has a packet to outputs 1and 2. a_1 chooses the packet to output 1. a_2 is pointing to 4 and sends a request to output 4. a_3 is pointing to 2, which means the packet should be sent to output 2. Similarly, a_4 requests output 3.

3.2. STATIC ALGORITHMS

The SSRR algorithm

In the SSRR (*Stable Round-Robin Scheduling Algorithms*) algorithm the input pointers are set to 1's. The output pointers are set to some initial pattern so that there is no duplication among the pointers [14]

1. *Request* – Each input sends a request to every output for which it has a queued cell.
2. *Grant* – If an output receives any requests, it chooses the one that appears next in fixed, round-robin schedule starting from the highest priority element. The output notifies each input whether or not its request was granted. The pointer to the highest priority element of the round-robin schedule is always incremented by one (modulo N) whether there is a grant or not.
3. *Accept* – If an input receives a grant, it accepts the one that appears next in a fixed round-robin schedule starting from the highest priority element. The pointer to the highest priority element of the round-robin schedule is incremented (modulo N) to one location beyond the accepted one.

Figure 5a shows an example of SSRR algorithm. In SSRR algorithm initially all arbiters are set to 1. g_1, g_2, g_3, and g_4 are arbiter pointers for outputs 1, 2, 3 and 4, respectively. In this example the inputs request transmission to all outputs that they have cells destined for. In step 2, among all received requests, each grant arbiter selects the requesting input that is nearest to the one currently pointed to. Output 1 chooses input 1, output 2 chooses input 2, output 3 chooses input 3. Each grant pointer is incremented by one (modulo N), than $g_1 = 2, g_2 = 3, g_3 = 4, g_4 = 1$. In step 3, each accept pointer decides which grant is accepted. Input 1 accepts grant from output 1. Since round-robin pointer a_2 is pointing to 1 input 2 accepts output 2. Input 3 accepts output 3. The pointer a_i is incremented to one location beyond the accepted output, than $a_1 = 2, a_2 = 3, a_3 = 4, a_4$ is not incremented.

Fig. 5. Example of scheduling algorithms in a 4×4 switch a) SSRR, b) DSRR

The DSRR algorithm

For the DSRR (*Double Static Round-Robin*), in the initialization part, the input pointers will be set to the same pattern as the outputs. Step 1 and 2 is the same as in the SSRR algorithm. Step 3 – *Accept* is changed to:

Accept – If an input receives a grant, it accepts the one that appears next in the fixed round-robin schedule starting from the highest priority element. The pointer to the highest priority element of the round-robin schedule is always incremented by one (modulo N) whether the grant is an accept or not [9].

Figure 5b shows an example of DSRR algorithm. Step 1 and 2 is the same as in the SSRR algorithm. In step 3 all accept pointers are incremented to next position (modulo N), than $a_1 = 4, a_2 = 1, a_3 = 2, a_4 = 3$.

The RDSRR algorithm

For the RDSRR (*Rotating Double Static Round-Robin*) the output pointers are set to the initial pattern such that there is no duplication among the pointers. The same is done to the input pointers. Here are the three steps that each iteration must follow:

1. *Request* – Each input sends a request to every output for which it has a queued cell.
2. *Grant* – If an output receives any requests, it chooses the one that appears next in the fixed round-robin schedule starting from the highest priority element.

To achieve fairness, clockwise and counter-clockwise rotation of the arbiter pointers are done alternatively, each for one time slot. The output notifies each input whether or not its request was granted. The pointer to the highest priority element of the round-robin schedule is always incremented by one (modulo N) whether there is a grant or not.
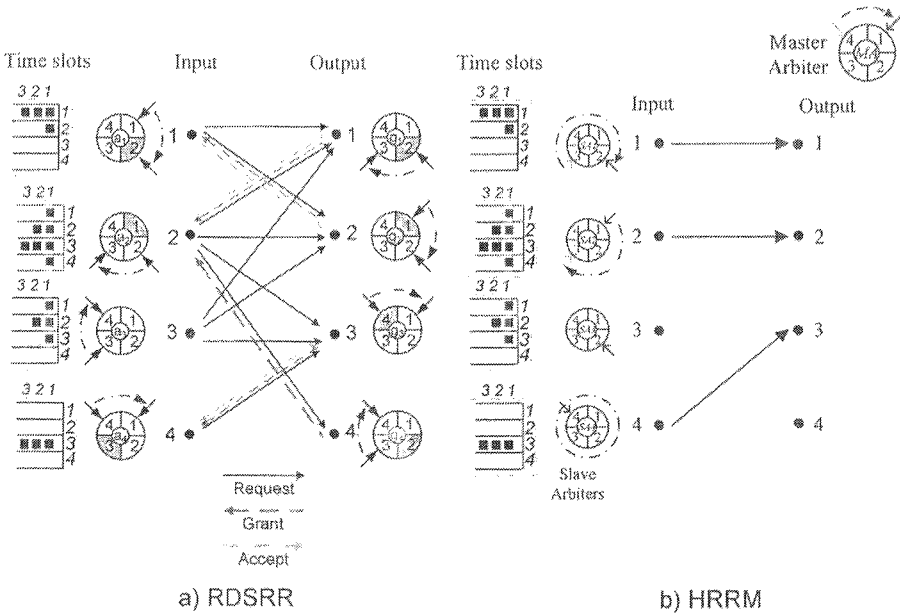
3. *Accept* – If an input receives a grant, it accepts the one that appears next in the fixed round-robin schedule starting from the highest priority element. The pointer to the highest priority element of the round-robin schedule is always incremented by one (modulo N) whether there is a grant or not.

Fig. 6. Example of scheduling algorithms in a 4×4 switch a) RDSRR b) HRRM

Figure 6 shows an example of RDSRR algorithm. In the Request step, each input sends Request to each output. Output 1 chooses input 2, output 2 chooses input 1, output 3 chooses input 4, and output 4 chooses input 2. In step 3, each accept pointer decides which grant is accepted. Input 1 accepts output 2, input 2 accepts output 1, input 4 accepts output 3. Each accept and grant pointer move one position (modulo N).

3.3. MMRS AND MMRRS ALGORITHMS

The general architecture of the scheduler for the Maximal Matching with Random Selection (MMRS) and the Maximal Matching with Round-Robin Selection (MMRRS) algorithms is shown in Figure 7a. It consists of N logical circuits called counters and one circuit called the selector. Each VOQ is connected to the scheduler by means of

two control lines. One line is used for sending the request signal from the VOQ to the scheduler, and the other line is used for sending the grant signal from the scheduler to the VOQ. When all VOQs have HOL packets, these N counters check N perfect matchings between inputs and outputs. The example of such perfect matchings for a 4×4 switch is shown in Figure 7b. Counters are then connected to the selector. The selector chooses the set of requests which will be realized in the successive time slot. These algorithms are performed in two steps: request and grant. [3, 15, 16]
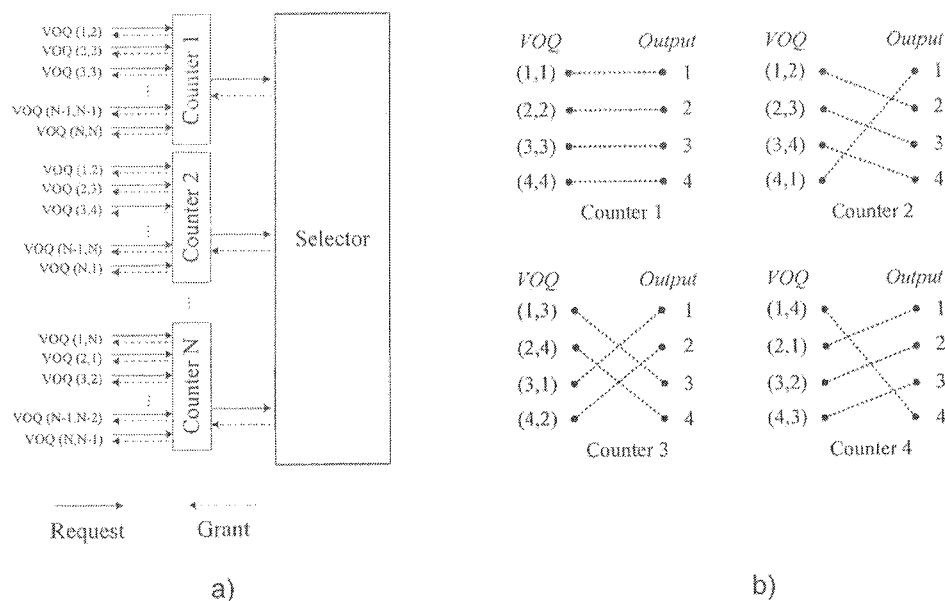
Fig. 7. a) General architecture of the scheduler, b) Matching being checked in counters 1 to 4 in 4×4

1. *Request* – Each non-empty VOQ sends request signal to the scheduler. Counters determine the number of request signals received and pass the result to the selector.
2. *Grant* – The selector selects the counter with the biggest number of requests and sends a grant signal to the selected counter. This signal is then passed by the counter to the respective VOQs.

In the Request step each non-empty VOQ sends the request signal to the arbiter. In the arbiter, these signals are grouped into N groups of N signals in such a way that each group constitutes a one-to-one matching. Signals from group i are then counted in the counter $i, 1 \leq i \leq N$. Each counter determines the number of request signals received, and sends it to the selector. The selector selects the counter with the biggest number of requests. In the Grant step the selector sends the grant signal to the selected counter and this counter sends grant signals to respective VOQs which sent request signals. When there are more than one counter with the same number of request signals, the selector chooses one of these counters either at random or using

the round-robin algorithm. In the former case the algorithm will be called the *Maximal Matching with Random Selection* (MMRS). In the latter case, the selector selects the first counter with the biggest number of request signals starting from the next counter to those selected in the previous time slot. We will call this algorithm the *Maximal Matching with Round-Robin Selection* (MMRRS). In both cases, the selection is done only between counters with the same and the biggest number of request signals.
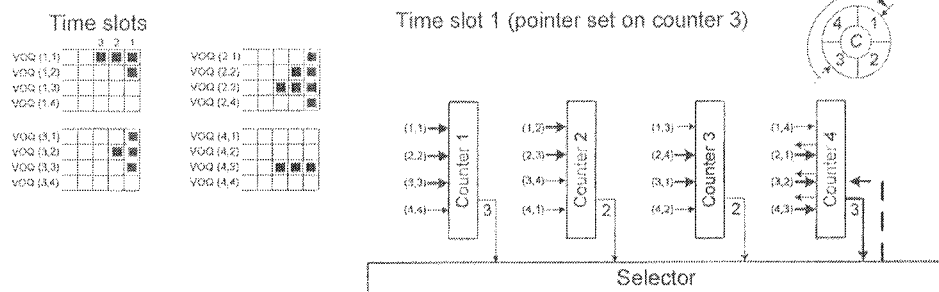
Fig. 8. Example of the MMRRS execution in 4×4 switch

The example of the MMRRS algorithm execution is shown in Figure 8. States of HOL cells in time slots 1, 2 and 3 are given in Figure 8a, where the black square denotes that the given HOL cell has the packet for transmission. At the beginning of time slot 1 we assume that in the previous time slot counter 2 was selected and the pointer in the selector is set to 3. In the first step each VOQ which has the HOL packet sends the request signal to the arbiter. These signals are shown at the inputs of the counters of Figure 8b in bold arrows. Each counter counts the number of the request signals received and pass it to the selector. In time slot 1 counters 1, 2, 3, and 4 receive 3, 2, 2, and 3 request signals, respectively. Two counters receive three requests. Counter 4 is selected by the selector by sending the grant signal (bold and dashed arrow). The pointer is set to 2.

3.4. THE HRRM ALGORITHM

The scheduling algorithm we propose is called Hierarchical Round-Robin Matching. The algorithm uses two kinds of arbiters, one master arbiter (MA), which points to the highest priority input port and N slave arbiters (SA), which point to output ports of the highest priority. Matching begins from the highest priority input port assigned by MA. A packet is sent from the input assigned by MA to the first free output port, starting from the highest priority port pointed by SA. If there is not any packet in a given queue or the output is already engaged the arbiter goes on to the next queue. After setting up the matching the given port is treated as engaged and is not taken into account during the following matchings. The other inputs will follow the same pattern during the matching process. MA upgrading takes place in each time slot to

one location beyond the highest priority port. Each SA in each time slot is incremented to one location beyond the granted output when the packet has been sent. Otherwise there is no upgrading. [17]

Figure 6b shows an example of the HRRM algorithm execution. States of HOL cells in time slots 1 and 2 are given, where the black square denotes that the given HOL cell has a packet for transmission. Grey arrows denote the states of arbiters at the beginning of each time slot, whereas black arrows denote the states of arbiters at the end of the time slot. At the beginning of time slot 1 we assume that MA is set to 4, SA1 and SA3 are set to 3, SA2 is set to 2, and SA4 is set to 4. In the first time slot the matching begins from the 4th input port (MA=4), for which SA4 is set to 4. Input 4 has HOL cells to outputs 1 and 4. The packet from input 4 is sent to output 4, which is the result of SA4 being set to 4. Inputs 1 and 2 have HOL cells for outputs 1 and 2. For input 1 SA1 is set to 3, which means the packet should be sent to output 3. However, since there is no any packet to outputs 3 and 4, SA1 chooses the packet for output 1. For input 2, SA2 is set to 2, the packet from this input is sent to output 2. For input 3, SA3 is set to 3, which means the packet is sent to output 3. At the end of the first time slot MA is upgraded to 1. At the end of the time slot SA1 is incremented to 2, SA2 is incremented to 3, SA3 is incremented to 4 and SA4 is incremented to 1.

4. SIMULATION RESULTS

The scheduling algorithms were evaluated and compared to each other by using simulation. The simulation results were gathered for 4×4, 8×8, and 16×16 switches. In this part we will consider the algorithms with one iteration. The results introduced in this section are presented for a 16×16 switch with VOQs of infinite size. The performance of algorithms was simulated under Bernoulli arrivals and bursty packet arrivals with average burst lengths of 16, 32, 64 cells for switches of different sizes. Packets directed to particular outputs were distributed uniformly and non-uniformly. For non-uniformly distributed traffic the distribution method is presented in Table 1, where rows and columns correspond to inputs and outputs, entry (i, j) denotes the probability that a packet which arrives at input i will be directed to output j. For Bernoulli arrivals it was assumed that one packet size is equivalent to 1 time slot and a packet may arrive at the input in the given time slot with probability p. Next, we studied the performance under bursty packet arrivals. The traffic model we use is an on-off Markov-modulated process. An active period in this traffic is usually called a *burst*. Each source generates a burst of full cells with the same destination in an alternative way. Than there are idle periods of empty cells, the number of cells in each burst or idle period is geometrically distributed.

Table 1

The traffic matrix for a 4×4 switch under nonuniformly distributed traffic

	Output 1	Output 2	Output 3	Output 4
Input 1	1/2	1/6	1/6	1/6
Input 2	1/6	1/2	1/6	1/6
Input 3	1/6	1/6	1/2	1/6
Input 4	1/6	1/6	1/6	1/2

Throughput ρ was calculated according to Equation (3).

$$\rho = \frac{\sum_z a_z}{\sum_z b_z} \tag{3}$$

where: a_z denotes the number of cells passed in z time slot through the switch fabric, and b_z is the number of cells which arrive to the switch in time slot z, where z is the time slot number.

The Mean Time Delay (MTD) of a cell is measured as quotient, in which numerator is the sum of the difference between the time a cell gets into the VOQ and the time when the same cell is transferred to its designated output, and denominator is the number of sent cells. The MTD was calculated according to equation:

$$MTD = \frac{\sum_i (z_i^d - z_i^a)}{\sum_z a_z} \tag{4}$$

where: z_i^d denotes the number of time slot cell i departed, and z_i^a is the number of time slot cell i arrives to the switch. Simulations were carried out in 15 series for each traffic load. For each point the 95% confidence interval was calculated. The results obtained for MMRS and MMRRS are very similar, so only one curve for MMRRS is shown in the Figures. The results obtained for the algorithms proposed by us are shown in figures by means of bold lines. The simulation results are shown as mean values derived from ten independent simulation processes with the 95% confidence internal (in an independent way for every algorithm, traffic type, and load).

4.1. WITH BERNOULLI ARRIVALS

4.1.1. Bernoulli arrivals with uniformly distributed traffic

In a uniformly distributed traffic,

$$p_{ij} = p/N \ \forall i,j. \tag{5}$$

In case of the uniformly distributed traffic the possibility that a cell will appear in the given time slot is always the same and independent of the current state of other time slots. This is the most common traffic pattern used in literature. [18, 19].
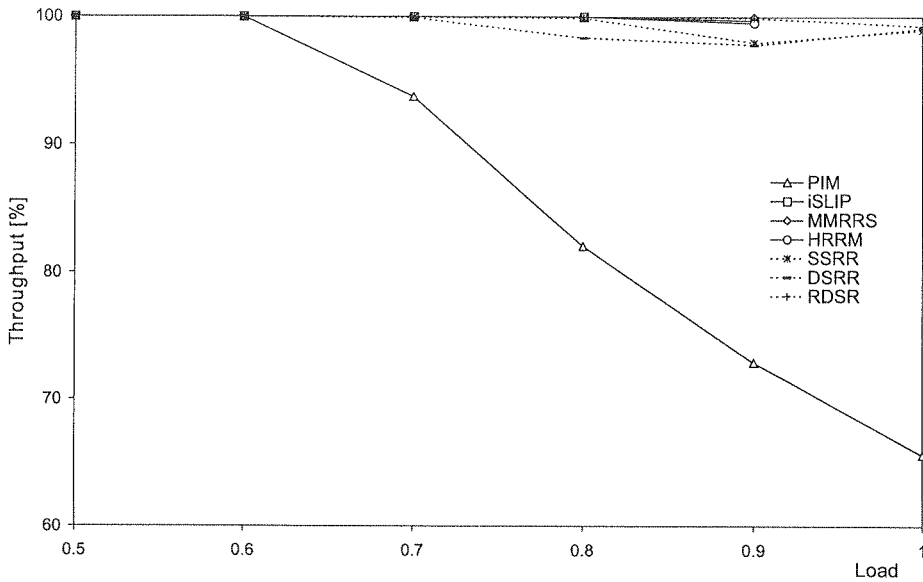

Fig. 9. The throughput in a 16×16 switch for Bernoulli arrivals with destinations uniformly distributed over all outputs

In Figure 9, we compare the throughput of PIM, iSLIP, MMRRS, HRRM, SSRR, DSRR and RDSR for Bernoulli arrivals with destinations uniformly distributed traffic. Simulation shows that iSLIP, MMRRS, HRRM, SSRR, DSRR and RDSR can achieve 100% throughput. However, throughput for PIM decreases as the load increases for $p > 0.6$.

Figure 10 shows the Mean Time Delay in a 16×16 switch with different scheduling algorithms. Analyzing the results for iRRM algorithm we noticed tendencies towards arbiters' synchronization. As the load increases the MTD also becomes higher and the outputs are blocked. The results of the simulations prove that MTD in PIM algorithm increases rapidly for the load greater than 0.6. In case of iSLIP, MMRRS, HRRM and static algorithms no output blocking has been observed. We can see that iSLIP is better than SSRR and DSRR under high load (over 0.65). There is not much difference between iSLIP and RDSRR. For the load greater than 0.75 MMRRS has a lower MTD in comparison with other algorithms considered in the literature. Special attention should be paid to HRRM algorithm, which allows to obtain lower MTD both for a high and for a low traffic loads. For the load lower than 0.9 the MTD is lower than 10 cells.
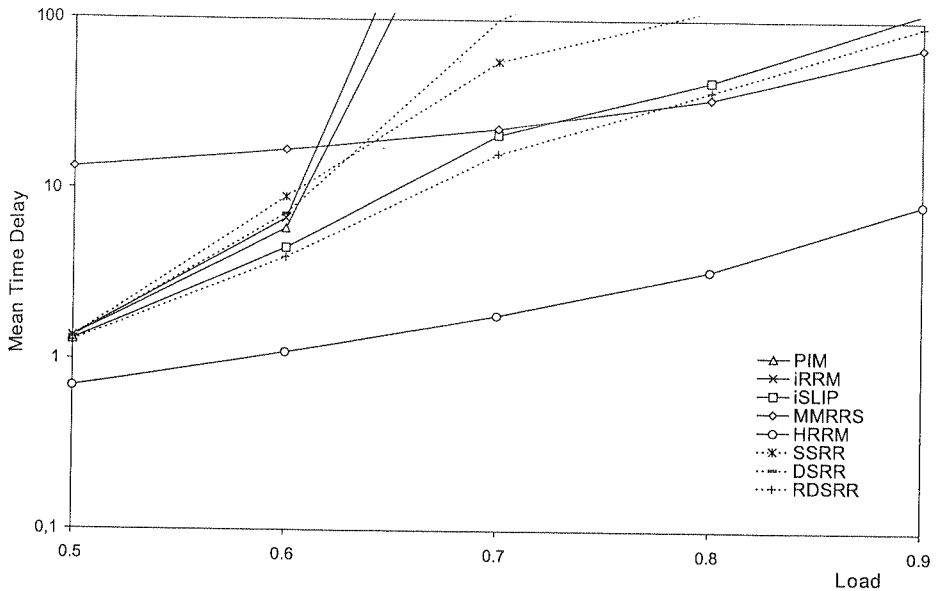
Fig. 10. The Mean Time Delay in a 16×16 switch for Bernoulli arrivals with destinations uniformly distributed over all outputs

4.1.2. Bernoulli arrivals with nonuniformly distributed traffic

The next considered traffic is Bernoulli arrivals with destinations non-uniformly distributed over all outputs, in which some outputs have a higher probability of being selected (Table 1), and respective probability p_{ij} is

$$p_{ij} = \frac{p}{2} \tag{6}$$

for $i = j$ and

$$p_{ij} = \frac{p}{2(N-1)} \tag{7}$$

for $i \neq j$.

The throughput was considered in a 16×16 switch with PIM, iSLIP, MMRRS, HRRM, SSRR, DSRR and RDSR scheduling algorithms for Bernoulli arrivals with destinations non-uniformly distributed over all outputs. Increase of the load results in decreasing of the throughput when PIMs, iRRM, iSLIP, and static algorithms are used. For MMRRS and HRRM nearly 100% throughput was observed. HRRM achieves lower throughput than MMRRS for load of around 0.7. (Figure 11). For load 0.9 throughput of other algorithms decreases to about 70%. Figure 12 shows the MTD in a 16×16 switch with different scheduling algorithms under nonuniformly distributed traffic. For the load above 0.7 MMRRS and HRRM obtains far lower MTD in comparison with other

algorithms. Static algorithms and PIM, iRRM and iSLIP show similar performance. We can see that SSRR, DSRR and RDSRR don't perform well under this traffic model.
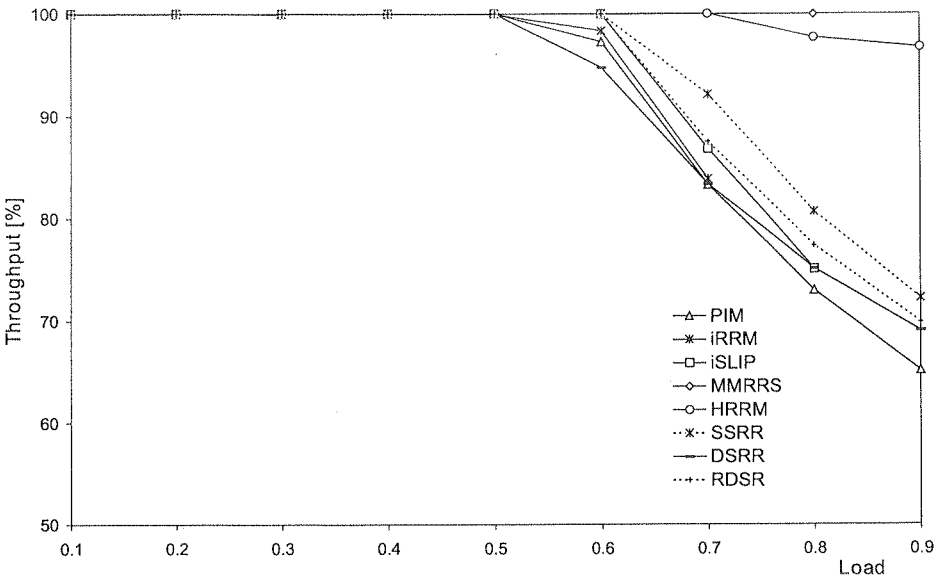

Fig. 11. The throughput in a 16×16 switch for Bernoulli arrivals with destinations nonuniformly distributed over all outputs
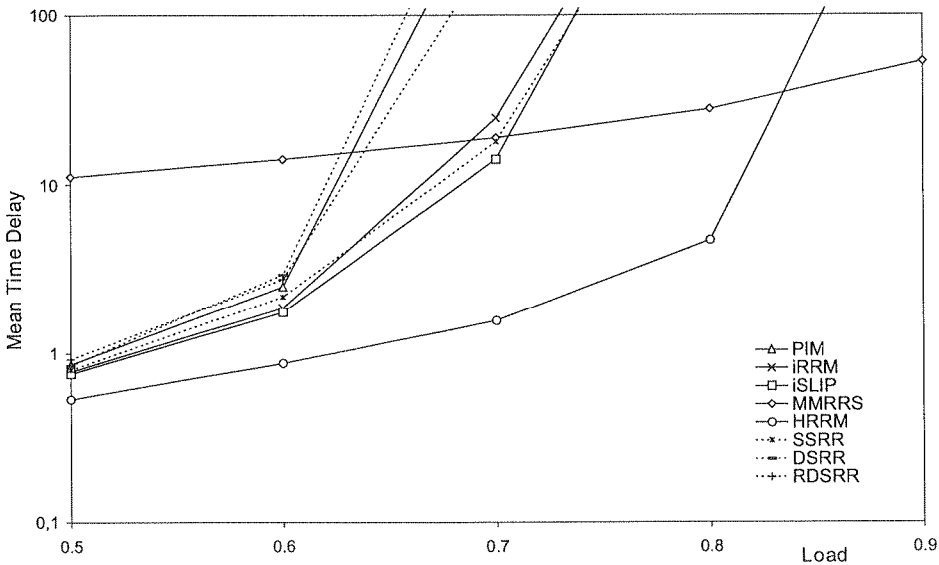

Fig. 12. The Mean Time Delay in a 16×16 switch for Bernoulli arrivals with destinations nonuniformly distributed over all outputs

4.2. WITH BURSTY ARRIVALS

In the bursty traffic (BT) model, each input alternates between active and idle periods of geometrically distributed duration. During an active period, cells destined for the same output arrive continuously in consecutive time slots. As the load increases the mean time delay also becomes higher. The probability that an active or an idle period will end at a time slot is fixed. Denote: $s = \text{Pr}[\text{BT is in OFF state at } t + 1$ — BT is in ON state at $t]$, $q = \text{Pr}[\text{BT is in ON state at } t + 1$ — BT is in OFF state at $t]$, $1 - s = \text{Pr}[\text{BT is in ON state at } t + 1$ — BT is in ON state at $t]$, $1 - q = \text{Pr}[\text{BT is in OFF state at } t + 1$ — BT is in OFF state at $t]$. As it has been said, an active period is called a burst. The mean burst length is $b = \dfrac{1}{s}$, and the load ρ is [10].

$$\rho = \frac{q}{q + s - sq} \tag{8}$$

In our simulation experiments the mean burst length has been $b = 16$ with all slots in an "ON" state always fully filled with cells.

4.2.1. Bursty arrivals with uniformly distributed traffic

Figure 13 compares throughput of well-known algorithms for bursty arrivals with destinations uniformly distributed over all outputs. For iSLIP, MMRRS, HRRM RDSR is almost 100% and for DSRR this throughput decreases to about 95% for the load greater than 0.7. Simulation shows that throughput for SSRR decreases as the load increases for $p > 0.5$.
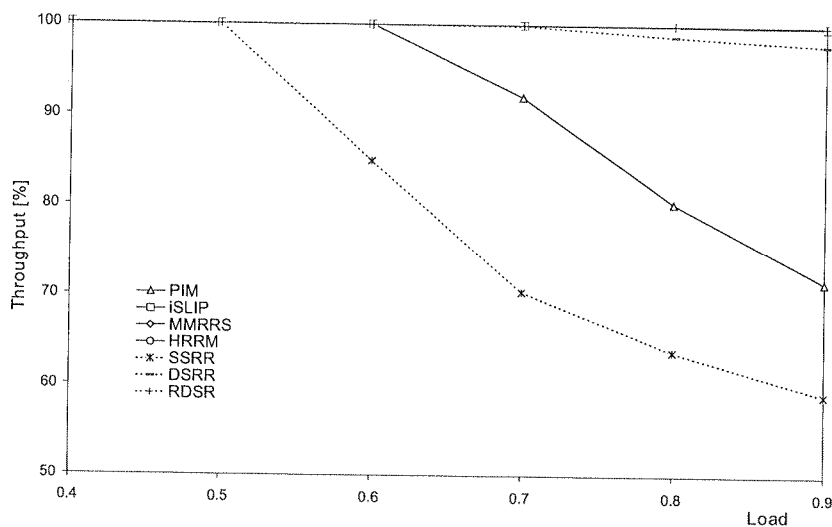

Fig. 13. The throughput in a 16×16 switch for bursty arrivals with destinations uniformly distributed over all outputs
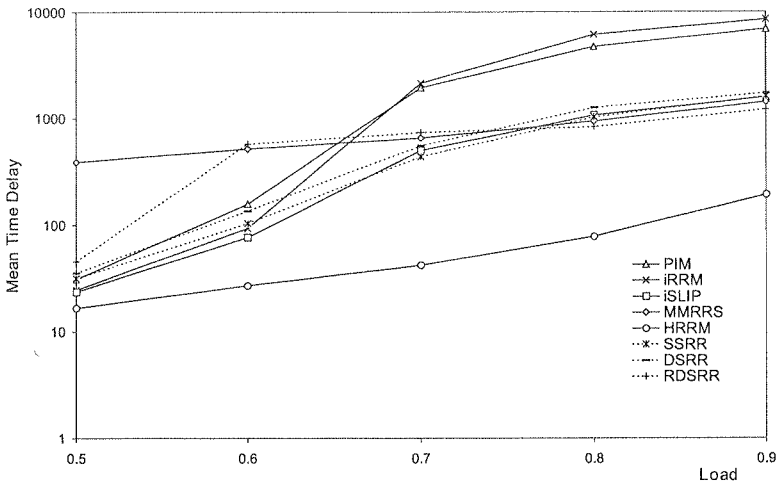
Fig. 14. The Mean Time Delay in a 16×16 switch for bursty arrivals with destinations uniformly distributed over all outputs

Figure 14 shows MTD in a 16×16 switch with different scheduling algorithm for bursty arrivals with destinations uniformly distributed over all outputs. PIM and iRRM show similar performance. Simulation results show that MTD for HRRM is better than for other algorithms.

4.2.2. Bursty arrivals with nonuniformly distributed traffic

Figure 15 shows throughput in a 16×16 switch for bursty arrivals with destinations nonuniformly distributed traffic. It shows that the throughput of PIM, iSLIP and static
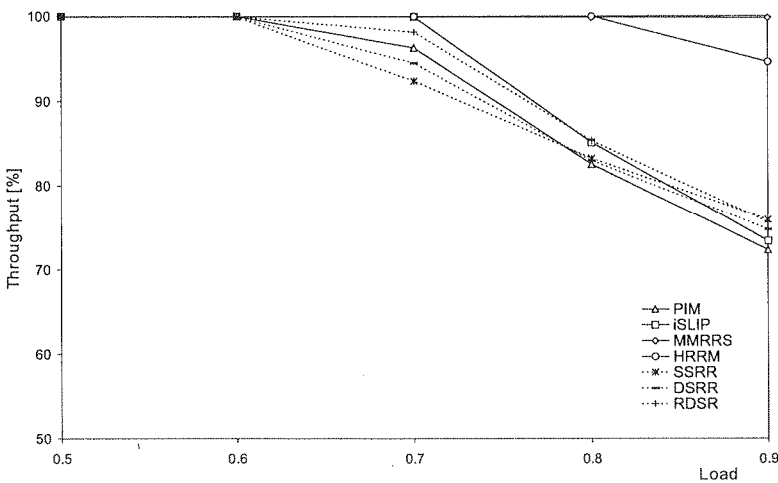

Fig. 15. The throughput in a 16×16 switch for bursty arrivals with destinations nonuniformly distributed over all outputs
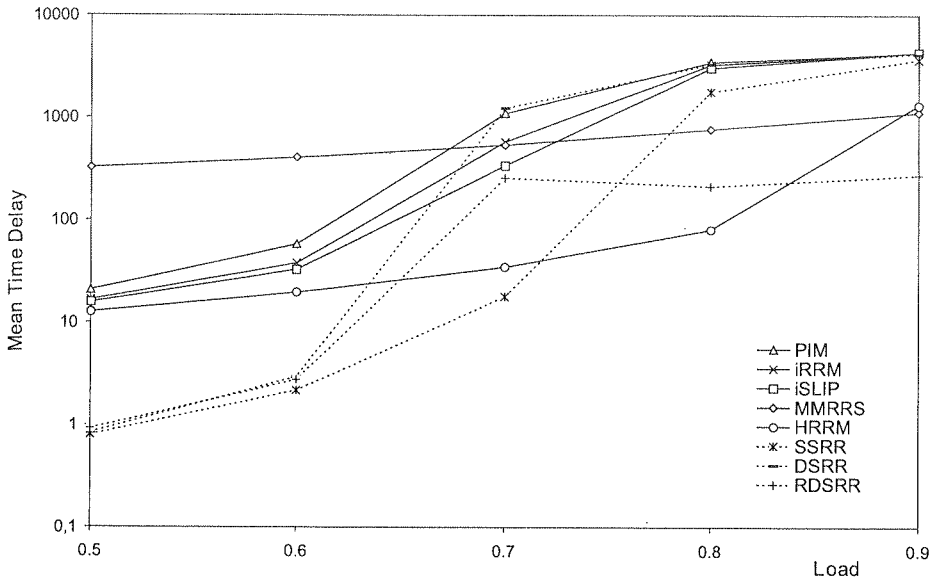
Fig. 16. The Mean Time Delay in a 16×16 switch for bursty arrivals with destinations nonuniformly distributed over all outputs

algorithms are similar to each other and much smaller than the throughput of MMRRS and HRRM. For MMRRS and HRRM algorithms the throughput for bursty packet arrivals with destinations nonuniformly distributed over all outputs are near 100%. Figure 16 shows the MTD in a 16×16 switch for bursty packet arrivals with destinations nonuniformly distributed over all outputs. As the load increases the MTD also becomes higher. For the load greater than about 70% the results obtained for MMRRS are better than for PIM, iSLIP and iRRM algorithms.

5. CONCLUSION

In this paper, we presented scheduling algorithm for switches with Virtual Output Queuing. We introduced a novel scheduling algorithms, MMRS, MMRRS and HRRM. The performance of the proposed algorithms was evaluated using simulations. Simulation results were executed for Bernoulli and bursty packet arrivals. We have considered uniformly and nonuniformly distributed traffic. The results obtained show that proposed algorithms not only allow to achieve low MTD but also ensure high fairness in choosing packets for transmission. For MMRRS and HRRM algorithms the throughput are almost 100% for each distributed traffic. The results obtained show that the proposed algorithms not only allow to achieve low awaiting times but also ensure high fairness in choosing packets for transmission.

6. ACKNOWLEDGMENTS

7. REFERENCES

1. Y. Tamir, G. Frazier: *High performance multiqueue buffers for VLSI communication switches.* Proc. 15th Annu. Symp. Comput. Arch., June 1988, pp. 343-354.
2. A. Baranowska, W. Kabaciński: *MMRS and MMRRS Packet Scheduling Algorithms for VOQ Switches.* Third Polish-German Teletraffic Symposium, September 2004, pp. 359-368.
3. A. Baranowska, W. Kabaciński: *The New Packet Scheduling Algorithms for VOQ Switches,* Lecture Notes in Computer Science, 2004, 3124, pp.711-716.
4. T. Anderson, S. Owicki, J. Saxe: *High Speed Switch Scheduling for Local Area Networks,* ACM Trans. Comput. Syst., 1993, vol. 11, no 4, pp. 319-352.
5. N. McKeown, P. Varaiya, J. Warland: *Scheduling cells in an Input-Queued Switch,* IEEE Electron. Lett., 1993, pp. 2174-2175.
6. N. McKeown: *The iSLIP scheduling algorithm for input-queued switches,* IEEE Transactions on Networking, April 1999, vol. 7, no 2, pp. 188-201.
7. K. Yoshigoe, K. Christensen: *An evolution to crossbar switches with virtual output queuing and buffered cross points,* IEEE Network, Vol.17, no 5, 2003, pp. 48-56.
8. D. Serpanos, P. Antoniadis: *FIRM: A Class of Distributed Scheduling Algorithms for Highspeed ATM Switches with Multiple Input Queues,* IEEE INFOCOM 2000 – The Conference on Computer Communications, 2000, pp. 548-555.
9. Y. Jiang, M. Hamdi: *A Fully Desynchronized Round-Robin Matching Scheduler for a VOQ Packet Switch Architecture,* IEEE HPSR'2001, 2001, pp. 407-411.
10. H. J. Chao, C. H. Lam, E. Oki: *Broadband Packet Switching Technologies: A Practical Guide to ATM Switches in IP Routers,* New York: Willey, 2001.
11. N. McKeown: *Scheduling Algorithms for Input-Queued Switches,* Ph.D. thesis, University of California at Berkeley, 1995.
12. J. Chao: *Saturn: A Terabit Packet Switch Using Dual Round-Robin,* Global Telecommunications Conference, GLOBECOM '00, 2000, vol. 1, pp. 487-495.
13. Y. Li, S. S. Panwar, H. J. Chao: *On the Performance of a Dual Round-Robin Switch,* Proceedings of IEEE INFOCOM 2001, April 2001, pp. 22-26.
14. J. Liu, H. C. Kit, M. Hamdi, Chi Ying Tsui: *Stable Round-Robin Scheduling Algorithms for High-Performance Input Queued Switches,* in Proceedings of the 10th Symposium on HotI'02, 2002, pp. 43, IEEE Computer Society.
15. A. Baranowska, W. Kabaciński: *MMRS and MMRRS Packet Scheduling Algorithms for VOQ Switches,* Proc. MMB PGTS 2004, 2004, pp. 359-368.
16. A. Baranowska, W. Kabaciński: *Evaluation of MMRS and MMRRS Packet Scheduling Algorithms for VOQ Switches under Bursty Packet Arrivals,* 2005, Hong Kong.
17. A. Baranowska, W. Kabaciński: *Hierarchical Round-Robin Matching for Virtual Output Queuing Switches,* A-ICT 2005, Portugalia, 2005, pp. 196-201.
18. P. Giaccone, B. Prabhakar, and D. Shah. Randomized Scheduling Algorithms for High-Aggregate Bandwidth Switches. IEEE J. Select. Areas Commun., May 2003, 21(4), pp. 546-559.
19. D. Shah, P. Giacconeand, B. Prabhakar: *Efficent Randomized Algorithms for Input-Queued Switch Scheduling,* Hot-Interconnects IX, 2002, 22(1), pp.10-18.

An overview of Multiple-Input Multiple-Output (MIMO) systems

KRZYSZTOF BRONK[1], ADAM LIPKA[1], RYSZARD J. KATULSKI[2], JACEK STEFAŃSKI[2]

[1] *National Institute of Telecommunications*
Marine Telecommunications Department
Jaskowa Dolina 15, 80-252 Gdańsk
e-mail: <K.Bronk, A.Lipka>@itl.waw.pl
[2] *Gdansk University of Technology*
Department of Radiocommunications Systems and Networks
Narutowicza 11/12, 80-952 Gdańsk
e-mail: <rjkat, jstef>@eti.pg.gda.pl

In this article, a novel multiple-input multiple-output (MIMO) technology is presented. MIMO is a technique that gives the opportunity to achieve very high data rates in radio networks, thanks to utilizing numerous spatial channels, resulting from the use of many antennas on one or both sides of the radio link. In the first part of this article, we describe the basic facts referring to MIMO, including the major benefits from using this technology (out of which diversity and multiplexing gains are arguably the most significant). After that, the mathematical background of MIMO is briefly outlined. The subsequent part refers to one of the most significant issues connected with multiple-antenna systems, that is the channel capacity. Employing MIMO allows to increase this capacity and consequently to boost transmission rate. Basic definitions and formulas for capacity in different cases are provided. The next part explains two most common methods of transmission, i.e.: spatial multiplexing and space-time coding. Finally, in the last part we summarize and conclude the article.

Keywords: multiple-input multiple-output, MIMO, channel capacity, spatial multiplexing, space-time coding

1. INTRODUCTION

In this article, the theoretical basics of multiple-antenna systems will be provided. In the most intuitive approach, these systems utilize transmission, reception, or both

transmission and reception using many antennas. In the first case, we say about transmit diversity and MISO systems (Multiple Input Single Output); in the second case – about receive diversity and SIMO systems (Single Input Multiple Output), and finally in the third case – we say about MIMO systems (Multiple Input Multiple Output). Thanks to using many antennas in the radio link, several benefits referring to quality of a system and achievable transmission rates are gained without the need for the increase of frequency band, delays or number of carrier frequencies.

In general, there are two basic ways of transmission in multiple-antenna systems, i.e.: spatial multiplexing which assumes that different information are broadcast by different antennas, and space time coding in which the same information (appropriately encoded) are sent by multiple antennas according to a strictly defined pattern.

In spatial multiplexing, transmission from each of the transmit antennas is realized in the same frequency band and at the same time (simultaneously), so incomparably more information can be broadcast at any given moment of time than in the case of only one transmit antenna (to be precise: up to M times more information, where M denotes the number of transmit antennas). In other words: the utilization of the band is M-times more efficient, and the whole stream spread over M antennas is transmitted up to M-times faster. In case of space-time coding, the main benefit of this technique is the increase of the overall quality of transmission by decreasing bit error rate (BER). When the appropriately encoded (in time and space) information is sent from many antennas, the receiver will use numerous replicas to make a decision, and even if some of those replicas are distorted due to transmission in a radio channel, it will not degrade the quality to such an extent as it would in the case of a single-antenna system (SISO). Both spatial multiplexing and space-time coding will be analyzed in greater details in the further part of this article.

The MIMO systems are based on spatial separation of multiple streams containing original data. Information are sent over spatial subchannels, whose number depends on the number of antennas at both sides of the link. It is noteworthy that multiple-antenna technique exploits multipath propagation in a positive way. This phenomenon has been considered to be definitely a negative one for a very long time, and many research have been carried out in order to minimize its destructive influence on transmission quality. This time, however, environment with rich multipath propagation is the most desired one for multiple antenna systems since it provides the possibility to distinguish and separate multiple substreams and to increase channel capacity which is one the most significant features of MIMO systems.

2. THEORY AND TECHNIQUE OF THE MIMO SYSTEMS

2.1. THE MAJOR BENEFITS OF MIMO SYSTEMS

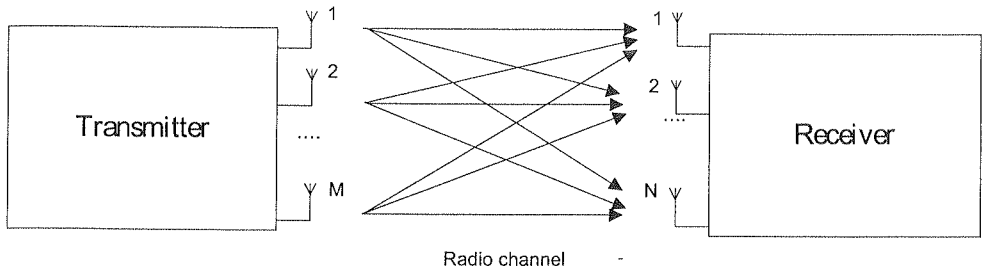The conceptual scheme of a MIMO system using M transmit and N receive antennas is depicted in Fig. 1.

Fig. 1. An intuitive definition of a multiple-input multiple-output system

There are 4 phenomena which result in performance enhancement in MIMO systems, i.e.: array gain, diversity gain, spatial multiplexing gain and co-channel interference. In the following part they will be briefly described [1].

2.1.1. Array gain

Array gain is a result of signal processing in transmitter (beamforming) and/or receiver. For instance in transmitter it comes down to forming the signal sent from antennas with so-called weight coefficients. This gain is expressed by the increase of the average SNR (signal to noise ratio) as a result of combining signals (replicas) from multiple antennas (these signals are added up in the receiver). Transmit/receive array gain requires channel knowledge in transmitter or receiver, respectively, and depends on the number of antennas. Channel knowledge is necessary since it enables to implement such signal processing algorithms which would reflect a current channel state and thus maximize benefits resulting from this kind of processing. Obviously, obtaining channel knowledge in the receiver is relatively easy to perform (e.g. with training sequences), whereas in the transmitter – much more difficult and requires so-called closed-loop systems.

2.1.2. Diversity gain

Diversity gain is a result of using multiple spatial paths for transmission of the signal (in ideal case, paths should be independent from one another). In that case, even if the quality of signal carried by one of the paths is bad, the other paths can

still be exploited to decode the information correctly, and the bit error rate will not be influenced as severely as it would happen in the SISO system. In general, diversity is used to reduce the negative influence of fading in the radio link. If we consider the system depicted in Fig. 1, we can easily notice that there are MxN spatial channels (paths) altogether which can be used to transmit information. Assuming that:

- fadings in each of these channel are independent (uncorrelated),
- transmitted signal has appropriate features,

the amplitude fluctuations of the resulting signal (containing replicas from each paths) will be reduced comparing to the SISO system.

To achieve diversity, one might exploit paths that are independent in either time, frequency or space. Out of these three options, the third one seems to be optimal as it does not require longer time of transmission or wider bandwidth.

In order to maximize spatial diversity without channel knowledge in transmitter (the most common case) we use the special signals (codewords), created as a result of space-time coding.

2.1.3. Multiplexing gain

MIMO technique can be employed to increase channel capacity. In general, this increase is linear and proportional to min(M,N) and is obtained due to spatial multiplexing. This technique (unlike space-time coding) assumes sending different (independent) signals by different transmit antennas. It is realized in systems like BLAST, where data stream is divided into M layers (substreams) broadcast simultaneously by M antennas.

Each of the mentioned layers due to radio channel influence can be unambiguously distinguished by the so-called spatial signatures (representing distortion of each of the individual radio subchannels). These signatures are used in the receiver to separate the layers, prior to their decoding. For this reason, it is crucial to ensure the proper environment with rich scattering, where multipath propagation is strong, because in such conditions paths reaching the receiver will be most distinguishable, and consequently, multiplexing gain will be maximized.

There are many different techniques of spatial multiplexing that vary in the way of separating and processing substreams, but most of them are highly sophisticated and complicated in terms of the relevant algorithms and required processing resources, which in some cases might make their implementation much harder or even impossible. It should also be noted that systems using spatial multiplexing require at least the same number of antennas in the receiver as in the transmitter (in contrast to systems with space-time coding which provide diversity gain even with a single receive antenna). This requirement results in considerable costs increase due to the necessity of implementing redundant elements in both parts of the transceiver.

2.1.4. Co-channel interferences reduction

Co-channel interferences are caused by frequency reusing (frequencies are used many times in cells according to a particular pattern, and cells using the same frequencies interfere with one another). Employing MIMO technique can yield reduction of this effect thanks to the already mentioned spatial signatures which reflect radio channel influence on a given signal path. In order to distinguish the desired signal, one need to have its channel knowledge. Co-channel interference reduction is achievable in the transmitter and is realized by an appropriate signal processing, such that the signal that might be received by co-channel users is maximally attenuated and, on the other hand, maximally amplified in the direction of desired users. Obviously, in that case it is possible to use the same frequencies more often, and consequently – to increase system capacity.

2.2. THE MATHEMATICAL DESCRIPTION OF THE MIMO SYSTEMS

We shall now take a closer look at the functioning of multiple antenna systems. To begin with, we will analyze a system with a two antennas at both the transmitter and receiver ($M = 2$ and $N = 2^{1}$) (Fig. 2).
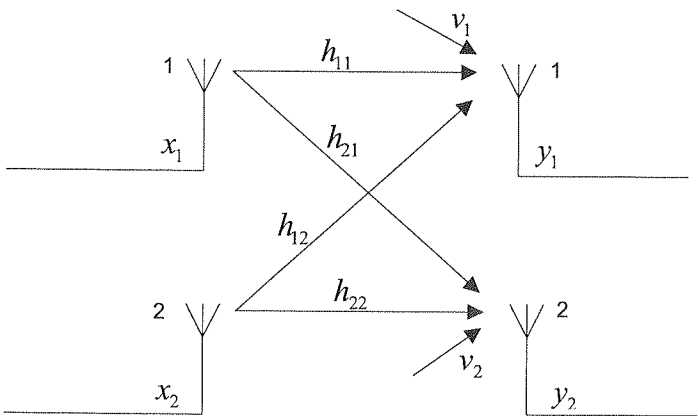

Fig. 2. MIMO system (2 transmit, 2 receive antennas)

A modulated radio signal x_j ($j = 1, 2$) propagating between transmit and receive antenna is attenuated and phase shifted. The radio channel's influence on this signal is described by the channel impulse response h_{ij}, where the element h_{ij} represents the channel between $j - th$ transmit antenna and $i - th$ receive antenna. Let $x_j(t)$ denote the signal transmitted in time t, and $y_i(t)$ ($i = 1, 2$) represents the received signal. $n_i(t)$ denotes white additive noise influencing the signal.

---

[1] In this article, M-means the number of transmit antennas, N-means the number of receive antennas.

In this case, channel (i.e. spatial propagation paths) characteristic is represented by the following matrix.

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \tag{1}$$

Consequently, we can define the set of equations valid for the analyzed system:

$$\begin{aligned} y_1(t) &= h_{11} \cdot x_1(t) + h_{12} \cdot x_2(t) + n_1(t) \\ y_2(t) &= h_{21} \cdot x_1(t) + h_{22} \cdot x_2(t) + n_2(t)' \end{aligned} \tag{2}$$

which can be expressed in the matrix form

$$\mathbf{Y} = \mathbf{H} \cdot \mathbf{X} + \mathbf{N}, \tag{3}$$

where $\mathbf{Y} = [y_1, y_2]^T$ – denotes the vector of received symbols, $\mathbf{X} = [x_1, x_2]^T$ – denotes the vector of transmitted signals, $\mathbf{N} = [n_1, n_2]^T$ – denotes the noise vector.

As we shall later explain, the transmission in the systems like in Fig. 2 is equivalent to a transmission over two independent radio channels which share the same frequencies and results in two-time increase of spectral efficiency. Thanks to a couple of antennas at both radio link sides we can transmit the information twice as fast (ideally) as in classic SISO system. The actual increase of the spectral efficiency (and thus – data rate) depends upon mutual independence (or lack of correlation) of each radio link.

As mentioned before, in practical implementations to achieve large spectral efficiency values, the spatial multiplexing is employed (most common, in the form of BLAST systems: V-BLAST, D-BLAST).

Let us analyze the system in Fig. 2 one more time and expand it to M transmit and N receive antennas. In that case the channel matrix will be as follows:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{NM} \end{bmatrix} \tag{4}$$

On the other hand, the signal received by the $i-th$ antenna, under the assumption that the $j-th$ transmit antenna sent $x_j(t)$, can be presented in such a way:

$$y_i(t) = \sum_{j=1}^{M} h_{ij} x_j(t) + n_i, \ \mathrm{j} = 1, 2, ..., N \tag{5}$$

To sum up the above introductory description we should underline two main goals of the MIMO technology. Firstly, it is quality of transmission improvement, i.e. bit error rate reduction achieved by space-time coding; and secondly, a noticeable spectral efficiency increase, which yields noticeable transmission rate improvement due to spatial multiplexing.

2.3. CHANNEL CAPACITY OF THE MIMO SYSTEMS

In this paragraph, we describe one of the most important issues connected with multiple antennas technique. A possibility of achieving much higher capacities, considerably exceeding values gained in classic systems (SISO systems), is one of the major merits of this technology. After a theoretical introduction, methods of channel capacity assessment will be presented:

• with an assumption that channel is deterministic (only one realization of the channel is considered, which is true only in the given time),

• for a flat fading channel,

We will define an analytical expression describing channel capacity of MIMO systems. But as a reference we will begin with a capacity of a classic system SISO, for which a matrix $\mathbf{H}$ is a scalar ($\mathbf{H} = h$) known in a receiver. It can be calculated as follows [5, 6]:

$$C_{SISO} = \log_2\left(1 + \rho \, |h|^2\right),\qquad(6)$$

where ρ denotes a ratio between the signal power and the noise power expressed in a linear scale.

The above relation results in a conclusion that the capacity of the SISO system grows logarithmically with a growth of a ratio of signal power to noise power. Consequently, if we want to increase the capacity by for example 10 bit/s/Hz we would have to increase SNR approximately 1000 times [5].

For a SIMO channel the matrix $\mathbf{H}$ is expressed as $\mathbf{H} = [h_1, h_2, ..., h_N]^T$. In this case the capacity can be obtained from [5, 6]:

$$C_{SIMO} = \log_2\left(1 + \rho \, |\mathbf{H}|^2\right) = \log_2\left(1 + \rho \sum_{i=1}^{N} |h_i|^2\right).\qquad(7)$$

In case of the multiple antennas in the receiver, and only the single antenna in the transmitter, the growth of the capacity is logarithmical and depends on a number of the antennas in the receiver.

In case of a MISO channel, the matrix $\mathbf{H}$ can be expressed as: $\mathbf{H} = [h_1, h_2, ..., h_M]$ and the channel capacity is defined as follows [5, 6]:

$$C_{MISO} = \log_2\left(1 + \frac{\rho \, |H|^2}{M}\right) = \log_2\left(1 + \rho \frac{\sum_{i=1}^{M} |h_i|^2}{M}\right).\qquad(8)$$

For a MISO channel, the growth of the number of the antennas in the transmitter does not result in any growth of its capacity.

The last possibility is the MIMO channel. We assume here that the receiver possesses a complete information about a channel state (matrix $\mathbf{H}$) and we treat this

information as a deterministic one. We can now calculate the channel capacity using an expression [1, 6, 7]:

$$C_{MIMO} = \log_2 \left[\det \left(\mathbf{I_N} + \frac{\rho}{M} \mathbf{HH^H} \right) \right], \tag{9}$$

where: det( ) – determinant of a matrix, $\mathbf{I_N}$ – NxN identity matrix (N – a number of receive antennas), $(\ )^H$ – conjunction and transposition of a matrix.

We can often encounter a simplified assumption that the matrix $\mathbf{H}$, which describes a channel, for large values of M and a fixed value of N can be presented as [8]:

$$\frac{1}{M} \mathbf{HH}^H = \mathbf{I}_N. \tag{10}$$

Let us assume the same number of the antennas at both sides of a radio link (M = N). We can now exploit the equation (9) to transform the expression (10). Now it is easy to notice that:

$$C_{MIMO} = \log_2 \left[\det \left(\mathbf{I}_N + \rho \cdot \mathbf{I}_N \right) \right] = \log_2 \left\{ \det \left[\mathbf{I}_N \left(1 + \rho \right) \right] \right\}. \tag{11a}$$

On the basis of a matrix algebra it is well known that: $\det(\mathbf{I}_n \cdot q) = q^n$, where q is the scalar and $\mathbf{I}_n$ is $n \times n$ identity matrix. Then after a further transformation of the expression (9) we obtain:

$$C_{MIMO} = \log_2 \left(1 + \rho \right)^N = N \cdot \log_2 \left(1 + \rho \right) = N \cdot C_{SISO}. \tag{11b}$$

It shows that for MIMO system a relation between the capacity and the number of antennas is linear [7]. We showed analytically that for the multiple antennas system the capacity of the channel grows theoretically N-times in comparison with the capacity of the SISO system. In general, this growth is proportional to *min(M, N)*, which was mentioned before in this article.

The above analysis can be further enhanced by making an eigenvalues decomposition of the matrix product $\mathbf{HH^H}$, which can be expressed as:

$$\mathbf{HH}^H = \mathbf{E} > \mathbf{E}^H, \tag{12}$$

where: $\mathbf{E}$ – matrix of eigenvectors, $\mathbf{\Lambda}$ – diagonal matrix containing the eigenvalues on a main diagonal.

Using the equation (12), and keeping in mind that $\det(\mathbf{I}) = 1$, we can now calculate the capacity of the MIMO system as follows:

$$C_{MIMO} = \sum_{i=1}^{Rank} \log_2 \left(1 + \frac{\rho}{M} \lambda_i \right), \tag{13}$$

where: Rank – matrix rank $(Rank \leq \min(M, N))$, λ_i – positive eigenvalues of the product $\mathbf{HH^H}$ $(i = 1, 2, ..., M)$.

The expression (13) results in an important conclusion. The capacity of the MIMO channel can be interpreted as a sum of parallel SISO channels' capacities whose gains are expressed by the parameters λ_i. To maximise the channel capacity it is desired to ensure as many of the eigenvalues to be nonzero and big as possible.

Similar conclusion can be derived from SVD (Singular Value Decomposition) decomposition of the matrix $\mathbf{H}$, which can be exploited instead of the eigenvalue decomposition [8]. As a result the simplest MIMO system (MIMO(2,2)) can be represented in a way shown in Fig. 3.
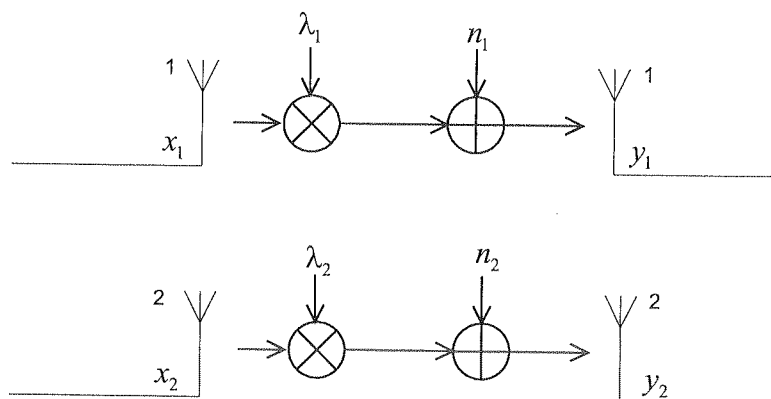

Fig. 3. The MIMO(2,2) system interpreted as two independent and parallel SISO channels

The all above analysis where done with an assumption that the channel is deterministic. Reality is different, however, because we have to consider the fading channel which causes fluctuations of the h_{ij} values. In such a situation, we need to use a bit different approach to estimate the channel capacity.

Two classes of the capacity can be defined i.e.: an ergodic capacity and an outage capacity. In the first case, through calculating an expected value of the expression (9), an average MIMO channel capacity can be obtained as [1]:

$$C_{ergodic} = E\left\{\log_2\left[\det\left(I_N + \frac{\rho}{M}\mathbf{H}\mathbf{H}^{\mathbf{H}}\right)\right]\right\}, \qquad (14)$$

where E( ) denotes an average (expected) value. On the other hand, the outage capacity $C_{out,q}$ describes the throughput which is ensured with the probability $(100 - q)\%$ (for $(100 - q)\%$ realizations of the radio channel) i.e. [1]:

$$P(C \leqslant C_{out,q}) = q\%, \qquad (15)$$

where $P(C \leq C_{out,q})$ denotes the probability (P) of the event that the capacity (C) will not exceed the outage capacity ($C_{out,q}$). In this case the capacity can be interpreted as a random variable which value depends on a temporary channel state (its impulse response). The probability in the expression (15) denotes that for the q% of the chan-

nel's realizations (the different channel's responses) the value of the capacity will drop below a threshold indicated by the outage capacity $C_{out,q}$.

3. METHODS OF TRANSMISSION IN MIMO SYSTEMS

3.1. SPATIAL MULTIPLEXING

Spatial multiplexing is one of the most common techniques of transmission being utilized in MIMO systems. In general, it comes down to transmitting M independent data symbols in a single symbol period. As a result, M-times more information can be broadcast per second than in a single-antenna system operating at the same data rate. To this end, there is no need to increase either the number of carriers or transmitting power, which is an additional benefit. In practical solutions, spatial multiplexing is performed by dividing a single input stream into M substreams (often referred to as "layers"), which are simultaneously broadcast by transmit antennas.

As of now, several algorithms of spatial multiplexing have been proposed; the main difference between them is a way of dividing a stream into substreams and methods of subsequent signal processing for each of substreams. Chronologically, the first spatial multiplexing algorithm was so-called Layered-Space-Time architecture (LST) or D-BLAST (Diagonal – Bell Labs Layered Space Time Architecture) proposed in 1996 by Gerard Foschini [9, 10]. It offers very high spectral efficiency values, and thus impressive data rates, but at the same time it is very computationally demanding and inefficient, which limits the possibility of its practical implementations. For this reason, it was necessary to find novel solutions which would constitute a convenient trade-off between capabilities and efficiency of the given method.

In the receiver, in most cases, decoding is performed separately for each of the layers, so it is necessary to first separate these layers from the received signal. At the moment, there are several ways to conduct this algorithmically convoluted processes. It should be noted that the choice of a given decoding method influences the overall quality of the system.

In the following part we will briefly outline the most common algorithms of spatial multiplexing.

3.1.1. H-LST Algorithm

Horizontal LST (H-LST) [1, 10, 11] is the simplest algorithm of spatial multiplexing. The main idea can be described in the following way: the input data stream is divided into M substreams (layers) in a serial to parallel converter (S/P). After that, bits belonging to the given substream are encoded (block or convolutional encoding), modulated and interleaved. After such a processing, the signals from each layer are transmitted from a selected antenna; it should be noted, however, that antenna-substream

association remains the same in time. Spatial rate (defined as the number of independent symbols in a transmitted word divided by the frame length) equals M for H-LST, and diversity gain is only N (any information bit can be transmitted by only one, selected antenna but can be received by all N receive antennas). For this reason, H-LST is not an optimal architecture but it strongly simplifies the structure of the receiver.

The transmitter of H-LST system is depicted in Fig. 4, and the antenna-substream association – in Fig. 5, where a number represents the number of layer from which symbols are transmitted in a given moment of time and through a given antenna.
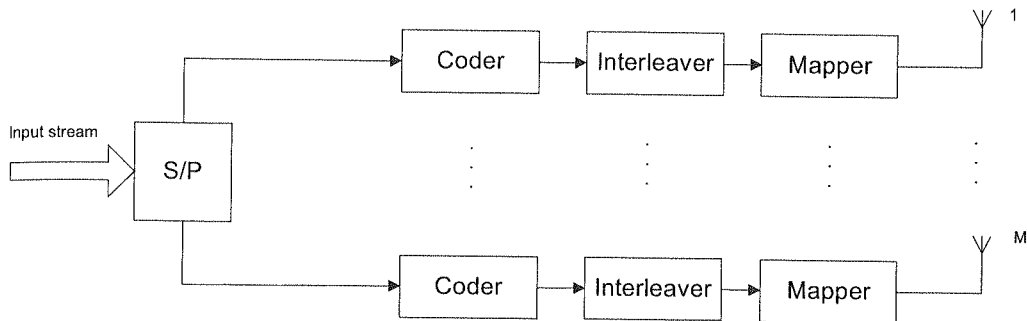

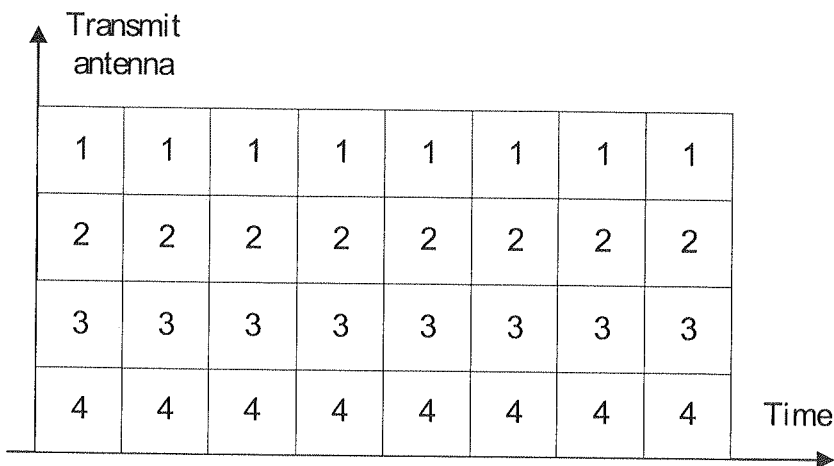Fig. 4. H-LST transmitter. (S/P – serial to parallel converter, M – number of transmit antennas)

Fig. 5. Stream-antenna association for H-LST systems

3.1.2. V-LST Algorithm

In the vertical LST algorithm [1], the input stream is encoded, modulated and interleaved prior to the eventual demultiplexing into M substreams, which is the major difference between this technique and the H-LST.

The stream-antenna association is in general the same as for H-LST (Fig. 5), however due to the reversed order of encoding and demultiplexing, the information of any input bit can be carried by streams associated to various antennas, so, unlike in H-LST, a single bit may be "spread" across every transmit antenna, which yield maximum diversity gain of M × N.

This algorithm offers additional coding gain (its value depends on the exploited code), and the spatial ratio of V-LST is M (like in H-LST).

Another technique which stems from the V-LST is an algorithm called V-BLAST [12]. In this case, the branches of transmitter are the QAM mappers, and the stream processing is simply bit demultiplexing, but in contrast to V-LST, mapping is performed for each layer independently. The V-BLAST technique does not require any coding, either of the substreams or any other, however if necessary – may be implemented. The scheme of the V-BLAST transmitter is shown in Fig. 6.
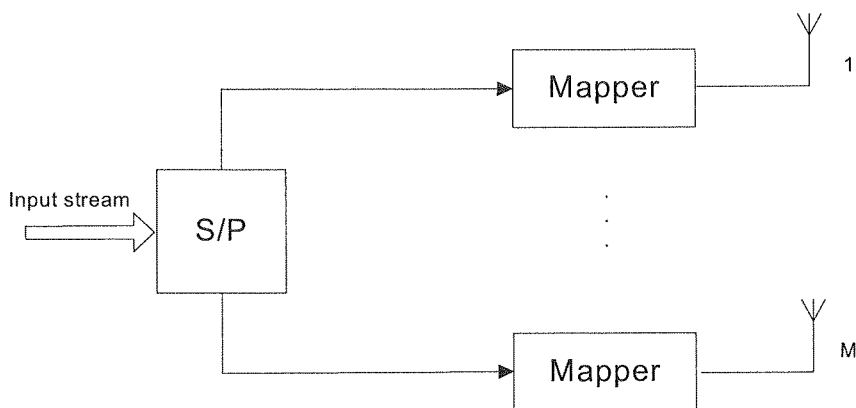

Fig. 6. V-LST Transmitter (S/P – serial to parallel converter, M – number of transmit antennas)

V-BLAST is known for ensuring tremendous values of spectral efficiency. In ref. [12, 13] it is claimed that during laboratory indoor research, the spectral efficiency values of 20 – 40bit/s/Hz for SNR of 24-34dB were achieved.

3.1.3. D-BLAST Algorithm

The diagonal BLAST is an algorithm where the two above techniques (H-LST and V-LST) converge. The transmit system can be described in the following way. The input stream is demultiplexed into M layers and each layer is encoded and mapped onto symbols. Those symbols are afterwards associated to the antennas in a diagonal manner, which means that for D-BLAST, the stream-antenna association is not fixed and varies in time. For M=4, such an association can be described like in Fig. 7.

The main drawback of the algorithm is a wastage of some time and spatial resources due to initial lack of transmission (indicated by the shaded area in the Fig. 7).

Such an
wasted s

To a
choose a
streams
Likelihoc
additiona
of these
levant ref
detector c
requireme
solutions
use the m

Space
a techniqu
method. It
designed c
quality is
same data
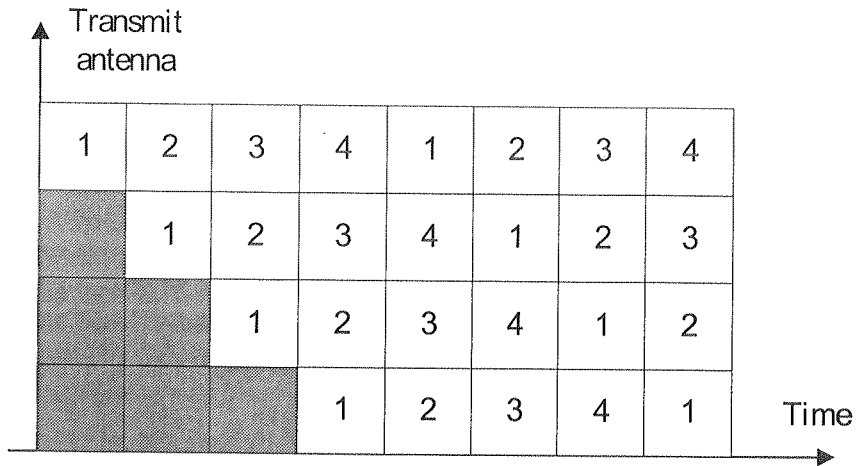copies in
through ex
assignmen
assignmen

Fig. 7. Stream-antenna association for D-BLAST systems

Such an approach results in simplifying the receiver structure but the losses of these wasted slots are critical and non-negligible especially for short blocks of information.

To achieve all the benefits offered by spatial multiplexing systems, it is vital to choose a proper receive algorithm, which will enable to separate and decode the sub-streams most reliably and efficiently. Some of the known algorithms are Maximum Likelihood and Minimum Mean Square Error decoding, Zero-Forcing receiver, and additionally Successive Cancellation and Ordered Successive Cancellation. The theory of these algorithms is well beyond the scope of this article and can be found in re-levant references (eg. [1]). It should be noted however that the Maximum Likelihood detector offers the best quality results but at a price of very high computational power requirements; on the other hand MMSE algorithm proves to be the worst. In practical solutions these methods are often joined, for example in V-BLAST it is proposed to use the method which benefits from ZF and OSUC algorithms.

3.2. SPACE-TIME CODING

Space-time coding is another way of transmission in the MIMO systems. It is not a technique which increases the channel capacity as it was for the spatial multiplexing method. Its main purpose is to achieve a maximum diversity gain through a properly designed codes. As a result, we get decrease of the bit error ratio (BER) – a transmission quality is improved. A diversity maximization can be achieved thanks to sending the same data (properly encoded) by each transmit antenna and correct assembling signal copies in the receiver. Information bits are mapped onto complex symbols obtained through exploiting a proper constellation. Meanwhile, the MIMO coder specifies an assignment of these symbols between transmit antennas in successive time slots. The assignment is often presented as a matrix whose successive rows describe the successive

time moments and columns represent the respective transmit antennas. It is noteworthy that if fading in all spatial subchannels will be uncorrelated then for the MIMO(M,N) system we achieve maximal diversity, that is M·N. It can be interpreted as a number of propagation paths of the signal and also as a number of the signal copies obtained in the receiver.

In practice, we can distinguish two types of space-time coding:
- Space-Time Trellis Coding (rarely used in practice),
- Space-Time Block Coding.

The space-time block coding (STBC) is the most common way of a realization of the space-time coding for the multiple antennas systems. An important task here is to design the optimal codes, which will determine the quality of STBC. Major criteria are [15]:

- An assurance of possibly big diversity (characterized by a number of an independent channels in which respective symbols are transmitted);
- A spatial code rate (a number of the transmitted symbols with respect to a length of a space-time block);
- A delay (the length of the block of the space-time code).

Most often, it is desired to achieve the maximum spatial code rate and the minimal delay with the assurance of full diversity. The codes, which are utilized for the purpose of the STBC technique, are created on the basis of an orthogonal design [16]. Generally, for the real constellations, the matrix describing an orthogonal code is a n × n square matrix. An orthogonality can be achieved only for specific values of n (n = 2, 4 or 8). The full spatial code rate can only be obtained when the orthogonality is ensured.

3.2.1. Alamouti Code

Alamouti code [17] is a quite simple code related to the case with two transmit antennas. It offers the full spatial code rate and its matrix is:

$$\begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}. \tag{16}$$

This coding can be described as follows: there is a complex constellation consisted of 2^b signals. During the first time slot, the coder gets 2b bits which decide about choosing two complex symbols s_1 and s_2. These symbols are simultaneously transmitted through both antennas. During the second time slot symbols $-s_2^*$ and s_1^* are transmitted through both antennas (Fig. 8).

Example:

Let us consider a system with two transmit antennas and with QPSK modulation whose constellation is depicted in Fig. 9. We can now assume a following sequence of the symbols: {0 1 1 3} or in binary form: {00 01 01 11}. In the first time slot the Alamouti coder gets two first symbols: 0 $(00)_2$ and 1 $(01)_2$. According to the assumed
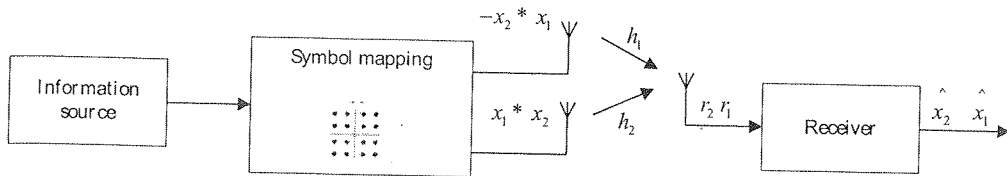
Fig. 8. A simplified scheme of the MIMO(2,1) system with the Alamouti coding

constellation, the symbol '0' is mapped onto $s_1 = 1$ and '1' onto $s_2 = j$. During the transmission in the first time slot the sequence {s_1 (first antenna) and s_2 (second antenna)}, that is {1, j}, will be sent, then in the second time slot the sequence {-s_2* (first antenna) and s_1* (second antenna)}, that is {j, 1}, etc.
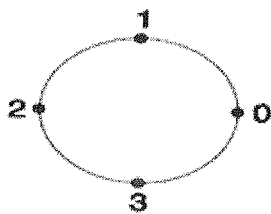

Fig. 9. The assumed QPSK constellation

The Alamouti coding can be then described as follows:

$$
\begin{array}{lcccc}
\text{Input sequence :} & 0 & 1 & 1 & 3 \\
\text{First antenna :} & 1 & j & j & -j \\
\text{Second antenna :} & j & 1 & -j & -j \\
\end{array}
$$

Let us consider a MIMO system with two transmit and two receive antennas. In that case, the signals received by each antenna can be defined, i.e.: the first receive antenna obtains, in the successive time slots, the signals r_1 and r_2 and the second one receives r_3 and r_4 (Fig. 10):
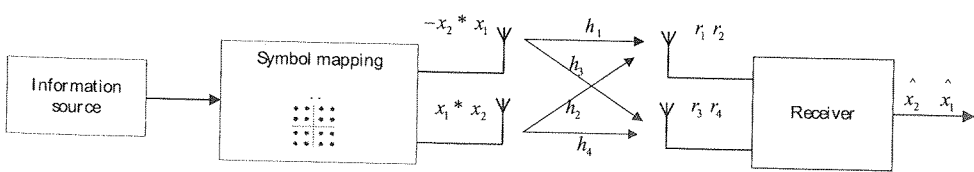

Fig. 10. A simplified scheme of the MIMO(2,2) system with the Alamouti coding

$$r_1 = h_1 s_1 + h_2 s_2 + n_1$$
$$r_2 = -h_1 s_2^* + h_2 s_1^* + n_2$$
$$r_3 = h_3 s_1 + h_4 s_2 + n_3 \tag{17}$$
$$r_4 = -h_3 s_2^* + h_4 s_1^* + n_4$$

In the receiver the two following signals are created:

$$\widetilde{s_1} = h_1^* r_1 + h_2 r_2 + h_3^* r_3 + h_4 r_4^*$$
$$\widetilde{s_2} = h_2^* r_1 - h_1 r_2^* + h_4^* r_3 - h_3 r_4^* \tag{18}$$

We can now substitute (17) into (18) and obtain the input signals passed then to the ML decoder:

$$\widetilde{s_1} = \left(|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2\right) s_1 + h_1^* n_1 + h_2 n_2^* + h_3^* n_3 + h_4 n_4^*$$
$$\widetilde{s_2} = \left(|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2\right) s_2 - h_1 n_2^* + h_2^* n_1 - h_3 n_4^* + h_4^* n_3 \tag{19}$$

In last years, many research have been carried out which prove that multiple antenna systems with the Alamouti coding enable to achieve a given BER with much lower SNR that in case of a single antenna system (1,1) with no coding implemented. This reduction of the required SNR is the undoubted value of the technique. For example in [2] it was shown for a simple (2,1) system, a BER $= 10^{-3}$ is achieved for SNR 5dB lower that for SISO system. When two antennas were used at both ends of the radio link (the (2,2) system), the gain was even greater – we could reduce SNR by 18dB to achieve BER $= 10^{-3}$ in comparison to the single antenna systems. This proves how promising the space-time coding method is.

4. CONCLUSION

Radio link is arguably the most inconvenient and toughest medium of transmission, and for this reason achieving high data rates in radio networks was considered as a difficult and sometimes even impossible task. Emerging of the MIMO technology can dramatically change this situation, and give the possibility to realize data rates that used to be achievable only in cable or optical networks. The key to success was to provide techniques of transmission that would ensure high level of spectral efficiency.

In this article, the basics of this novel, highly promising technology, were presented. We discussed the major benefits of utilizing MIMO, capacity formulas for different cases and finally two methods of transmission in MIMO systems, i.e. spatial multiplexing and space-time coding.

We can expect that in the future, MIMO as a system offering enormous data rates with more than sufficient quality will become common or even dominant in some branches of contemporary radiocommunications. At the moment, MIMO is included

(17)

(18)

then to

(19)

multiple
th much
emented.
que. For
eved for
ends of
SNR by
is proves

smission,
ered as a
logy can
rates that
s was to
fficiency.
vere pre-
mulas for
e. spatial

data rates
in some
included

into IEEE 802.16 (WMAN networks) standard as an option, and is proposed to be used in HSDPA (a subsystem of UMTS). Moreover, it is foreseen that in 2008, the IEEE 802.11n standard will be published, which will provide details of implementing multiple-antenna systems in WLAN networks. As we can see, the interest in MIMO from the major standardization organizations (IEEE, 3GPP), gives a "green light" for a further development of this technology. We can also expect that many theoretical aspects of MIMO, a few of which have been presented in this article, can evolve or change, so to stay up-to-date with MIMO, it is necessary to follow appropriate literature and references.

5. REFERENCES

1. A.J. P a u l r a j et al., *An Overwiev of MIMO Communications – A Key to Gigabit Wireless*, Proceedings of the IEEE, no. 2, vol. 92, 2004, pp. 198-216.
2. H. B o l c s k e i, A.J. P a u l r a j, *Multiple-Input Multiple-Output (MIMO) Wireless Systems, in book The Communications Handbook* edited by J. D. Gibson, CRC Press, 2002.
3. W. D z i u n i k o w s k i, W. L u d w i n, *Układy antenowe MIMO w sieciach bezprzewodowych*, Przegląd Telekomunikacyjny, nr 5, 2003.
4. R.J. K a t u l s k i, M. M i k o ł a j s k i, *Technika zwielokrotnionego nadawania i zbiorczego odbioru w telekomunikacji bezprzewodowej*, Przegląd Telekomunikacyjny, nr 2-3, 2005.
5. H. L i u, L. G u o q i n g, *OFDM-Based Broadband Wireless Networks*, John Wiley & Sons, 2005.
6. D. G e s b e r t, M. S h a f i, et. al., *From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems*, IEEE Journal On Selected Areas In Communications, vol. 21, no. 3, pp. 281-302, 2003.
7. K. W e s o ł o w s k i, *Systemy wielowejściowe/wielowyjściowe (MIMO) w radiokomunikacji ruchomej*, Przegląd Telekomunikacyjny, nr 6, 2004.
8. B. H o l t e r, *On the Capacity of the MIMO Channel – A Tutorial Introduction*, http://www.iet.ntnu.no /projects/beats/Documents/MIMO_introduction.pdf.
9. G.J. F o s c h i n i, D. C h i z h i k, et. al., *Analysis and Performance of Some Basic Space-Time Architectures*, IEEE Journal on Selected Areas in Communications, vol. 21, no. 3, pp. 303-320, 2003.
10. G.J. F o s c h i n i, *Layered Space-Time Architecture for Wireless Communication in Fading Environment when Using Multi-Element Antennas*, Bell Labs Technical Journal, vol. 2, 1996.
11. D. S h i u, J.M. K a h n, *Layered Space Time Codes for Wireless Communications using Multiple Antennas*, IEEE International Conference on Communications, vol. 1, pp. 436-440, 1999.
12. P.W. W o l n i a n s k y, et al., *V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel*, Proc. ISSSE-98, Pisa, Italy, 1998.
13. G.D. G o l d e n, et al., *Detection algorithm and Initial Laboratory Results Using V-BLAST Space-Time Communication Architecture*, Electronics Letter, vol. 35, no. 1, pp. 14-15, 1999.
14. V. T a r o k h, et al., *Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction*, IEEE Transactions on Information Theory, vol. 44, no. 2, pp. 744-765, 1998.
15. A. H o t t i n e n, *Multi-antenna Transceiver Techniques for 3G and Beyond*, John Wiley & Sons, 2003.
16. V. T a r o k h, et al., *Space-Time Block Codes from Orthogonal Designs*, IEEE Transactions on Information Theory, vol. 45, no. 5, pp. 1456-1467, 1999.
17. S.M. A l a m o u t i, *A Simple Transmit Diversity Technique for Wireless Communications*, IEEE Journal on Selected Areas in Communications, vol. 16, no. 8, pp.1451-1458, 1998.

Mo

r
r
p
s
p
r
c
r

K

Ma
electro
of inte
operati
The ma
does n
These e

Th
output

Modelling the changes of saturation voltages in high-power operational amplifiers

JÓZEF STANCLIK

*Wroclaw University of Technology,
Institute of Telecommunications, Teleinformatics and Acoustics
50-370 Wrocław, ul. Wybrzeże Wyspiańskiego 27
jozef.stanclik@pwr.wroc.pl*

A modification of the macromodel of operational amplifiers is proposed to icorporate a relationship between saturation voltages and output current. Such modification improves the macromodel accuracy over a wide range of load, power supply voltage and input signal amplitude variations. The modification consists in changing the EMF values of two autonomous sources in the initial macromodel and adding a resistor or a non-linear controlled source, its parameters having been derived from the relevant data sheets by applying the least squares method. A sample modification is shown for a macromodel of the OPA512 high-power operational amplifier, applicable in the PSpice program. Accuracy and applicability of the modified macromodel have been evaluated.

Keywords: macromodel, high-power operational amplifier, PSpice program

1. INTRODUCTION

Macromodels of operational amplifiers commonly used in computer simulation of electronic circuits take into consideration only the basic characteristics and features of integrated circuits at the settled operating conditions. The saturation voltage of operational amplifiers depends on the output current and may change by several volts. The macromodel of operational amplifiers, which is derived from Boyle's model [1], does not rely on this relationship. This approach leads to erroneous simulation results. These errors may even reach 40% while calculating the maximum output power [4].

The modelling of the maximum output voltage of operational amplifiers versus the output current is discussed in the literature [4, 5]. In [4], series resistances were introdu-

ced into the models of the voltage limiter diodes. Voltage drops across these resistances move the output voltage limiting thresholds in relation to the output current values and bring the saturation voltages of the macromodel closer to the saturation voltages of the integrated circuit. Similar effect was achieved when two output current-controlled voltage sources located in the output voltage limiters were included in the macromodel [5]. This modification allows modeling the non-linear relationship of the saturation voltage and the maximum output voltage versus the output current.

This paper outlines the modification to the macromodel of operational amplifiers, which is simpler than the modification proposed in [5], and which improves macromodel accuracy and extends variation ranges for the load, supply voltage and input voltage while the improved accuracy is maintained.

2. MODIFICATION OF THE MACROMODEL

In the original macromodel, the output voltage is limited by limiters composed of diodes, DC and DE, sources, VC and VE, and supply sources, Vs^+ and Vs^- (Fig. 1). The positive saturation voltage is equal to the difference between the positive supply voltage and the maximum output voltage. In practice, the positive saturation voltage is constant as the change of voltage drop on the limiter's conducting diode is not higher than several dozen of millivolts. Similarly the negative saturation voltage is nearly constant.
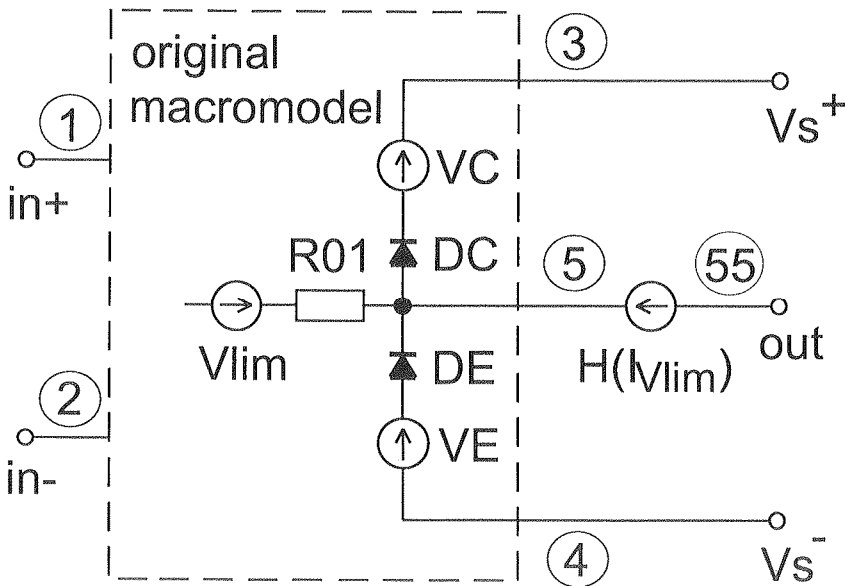

Fig. 1. The structure of the modified macromodel of an operational amplifier

The saturation voltages of the macromodel can be made dependent on the current by inserting either a voltage source, H, controlled by the output current or a resistor, R, into the output circuit of the macromodel. The source, $Vlim$, was used to sample the output current. In the currently used macromodels, the current of this source, I_{Vlim}, is nearly equal to the output current, I_{out} (the limiter's diode current is smaller by several orders). Within the linear operational range, the effect of the source, H, or the resistor, R, on the macromodel's parameters is, in essence, eliminated by negative feedback, which is a strong one in usual applications of amplifiers.

The positive and negative saturation voltages are given by the formulae (1) and (2).

$$V_{sat}^+ = Vs^+ - V_{out}^+ = VC - V_{DC} + H\left(I_{Vlim}\right) \tag{1}$$

$$V_{sat}^- = -Vs^- + V_{out}^- = VE - V_{DE} + H\left(I_{Vlim}\right) \tag{2}$$

where V_{DC} and V_{DE} are voltage drops across limiter's diodes. In order to make saturation voltages of the operational amplifier, V_{satIC}, equal to the saturation voltages of the macromodel, V_{satmm}, at the current I_{out} equal to zero, the EMFs of the sources, VC and VE, were modified as follows:

$$VC' = VC - V_{sat\,mm}^+\left(I_{out} = 0\right) + V_{sat\,IC}^+\left(I_{out} = 0\right) \tag{3}$$

$$VE' = VE - V_{sat\,mm}^-\left(I_{out} = 0\right) + V_{sat\,IC}^-\left(I_{out} = 0\right) \tag{4}$$

Application of the current-controlled voltage source, given by polynomial (5), allows modeling the non-linear relationship of saturation voltages versus output current.

$$H\left(I_{out}\right) = a_0 + a_1 I_{out} + a_2 \left(I_{out}\right)^2 + ... + a_k \left(I_{out}\right)^k \tag{5}$$

This polynomial approximates the reduced saturation voltage, ΔV_{sat}, expressed as follows:

$$
\Delta V_{sat}\left(I_{out\,i}\right) = \\
= \begin{cases} V_{sat\,IC}^+\left(I_{out\,i}\right) - V_{sat\,mm}^+\left(I_{out\,i}\right) + VC - VC', & 1 \le i \le n, I_{out\,i} \ge 0 \\ V_{sat\,mm}^-\left(I_{out\,i}\right) - V_{sat\,IC}^-\left(I_{out\,i}\right) - VE + VE', & 1 \le i \le m, I_{out\,i} < 0 \end{cases} \tag{6}
$$

where $n + m > k$ (n, m are the numbers of the values of the saturation voltages $V_{sat\,IC}^+$ and $V_{sat\,IC}^-$, respectively, taken from the specification sheets, and $V_{sat\,mm}^+$ and $V_{sat\,mm}^-$ calculated for the original macromodel). The approximation error is given by the formula (7).

$$
\varepsilon_i(a_0, a_1, ..., a_k) = \Delta V_{sat}\left(I_{out\,i}\right) + \\
- \left\{ a_0 + a_1 I_{out\,i} + a_2 \left(I_{out\,i}\right)^2 + ... + a_k \left(I_{out\,i}\right)^k \right\}, 1 \le i \le n + m \tag{7}
$$

The coefficients of the polynomial (5) are derived by using the least squares method. In most cases, the third-order polynomial ensures satisfactory approximation accuracy. The function (6) intersects the origin of coordinate system and is nearly symmetrical with respect to the origin. Therefore, the coefficients a_0 and a_2 are close to zero. Assuming these coefficients are equal to zero, the error resulting from approximating the voltage $\Delta V_{sat}(I_{out})$ given by equation (6), by the third order polynomial is as follows:

$$\varepsilon_i(a_1, a_3) = \Delta V_{sat}(I_{out\ i}) - a_1 I_{out\ i} - a_3(I_{out\ i})^3, \qquad 1 \le i \le n + m - 1 \qquad (8)$$

The mean square error of the approximation is given by the relationship (9).

$$S(a_1, a_3) = \sum_{i=1}^{n+m-1} \varepsilon_i^2 = \sum_{i=1}^{n+m-1} \left\{ \Delta V_{sat}(I_{out\ i}) - \left[a_1 I_{out\ i} + a_3(I_{out\ i})^3 \right] \right\}^2 \qquad (9)$$

The partial derivatives of the function (9) at its minimum are equal to zero. Therefore, the following set of equations appears:

$$a_1 \sum_{i=1}^{n+m-1} (I_{out\ i})^2 + a_3 \sum_{i=1}^{n+m-1} (I_{out\ i})^4 = \sum_{i=1}^{n+m-1} \left\{ \Delta V_{sat}(I_{out\ i}) \cdot I_{out\ i} \right\}$$

$$a_1 \sum_{i=1}^{n+m-1} (I_{out\ i})^4 + a_3 \sum_{i=1}^{n+m-1} (I_{out\ i})^6 = \sum_{i=1}^{n+m-1} \left\{ \Delta V_{sat}(I_{out\ i}) \cdot (I_{out\ i})^3 \right\} \qquad (10)$$

Upon resolving this set of equations, the values of coefficients of the polynomial (5) are as follows:

$$a_1 = \frac{\sum\limits_{i=1}^{n+m-1} (I_{out\ i})^6 \sum\limits_{i=1}^{n+m-1} \left\{ \Delta V_{sat}(I_{out\ i}) \cdot I_{out\ i} \right\} - \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^4 \sum\limits_{i=1}^{n+m-1} \left\{ \Delta V_{sat}(I_{out\ i}) \cdot (I_{out\ i})^3 \right\}}{\sum\limits_{i=1}^{n+m-1} (I_{out\ i})^2 \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^6 - \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^4 \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^4}$$

$$\qquad (11)$$

$$a_3 = \frac{\sum\limits_{i=1}^{n+m-1} (I_{out\ i})^2 \sum\limits_{i=1}^{n+m-1} \left\{ \Delta V_{sat}(I_{out\ i}) \cdot (I_{out\ i})^3 \right\} - \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^4 \sum\limits_{i=1}^{n+m-1} \left\{ \Delta V_{sat}(I_{out\ i}) \cdot I_{out\ i} \right\}}{\sum\limits_{i=1}^{n+m-1} (I_{out\ i})^2 \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^6 - \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^4 \sum\limits_{i=1}^{n+m-1} (I_{out\ i})^4}$$

When the function (6) is nearly linear, the polynomial (5) can be reduced to the single factor $H(I_{out}) = a_1 I_{out}$. The error of such approximation of the voltage $\Delta V_{sat}(I_{out})$ is specified by the folowing equation:

$$\varepsilon_i(a_1) = \Delta V_{sat}(I_{out\ i}) - a_1 I_{out\ i}, \qquad 1 \le i \le n + m - 1 \tag{12}$$

The mean square error of approximation is as follows:

$$S(a_1) = \sum_{i=1}^{n+m-1} \varepsilon_i^2 = \sum_{i=1}^{n+m-1} \{\Delta V_{sat}(I_{out\ i}) - a_1 I_{out\ i}\}^2 \tag{13}$$

At the minimum of this function, its derivative equals to zero, hence the coefficient a_1 is given by the formula (14).

$$a_1 = \frac{\sum\limits_{i=1}^{n+m} \{\Delta V_{sat}(I_{out\ i}) \cdot I_{out\ i}\}}{\sum\limits_{i=1}^{n+m} (I_{out\ i})^2} \tag{14}$$

The source $H = a_1 I_{out}$ converts the current flowing through it into a voltage drop. The same is achieved by a linear resistor; hence the coefficient a_1 can be interpreted as a resistance of the resistor R which replaces the controlled source H.

3. SAMPLE MODIFICATION OF THE MACROMODEL OF THE OPA512 INTEGRATED CIRCUIT

The relationship of saturation voltages versus output current given by the data sheet of the high-power operational amplifier, type OPA512, is shown in Fig. 2 [2]. The curves of saturation voltages for the original model, as found by PSpice computer simulation, are also drawn with dotted lines in this figure. They differ from the data sheet values and the maximum error is 2.4 V.
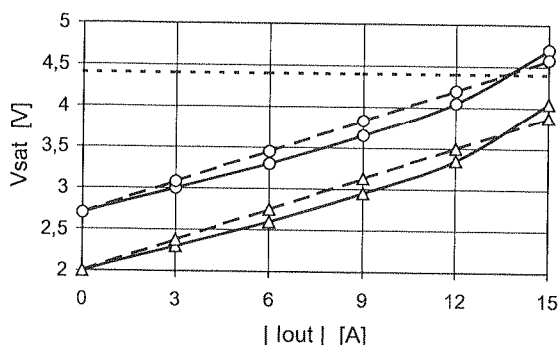

Fig. 2. Relationships between the saturation voltages and the output current for the high-power operational amplifier, type OPA512, and its macromodels: __ the data sheet values, ⋯ the original macromodel, - - - the modified macromodel, Δ – the positive saturation voltage, ○ – the negative saturation voltage

The proposed modification was introduced into this macromodel. Electromotive forces of the VC and VE sources were changed according to the formulae (3) and (4): $VC' = 5.0 - 4.4 + 2.0 = 2.6$ V, $VE' = 5.0 - 4.4 + 2.7 = 3.3$ V (the values of VC and VE were read out of the original macromodel's netlist partially shown in Table 1 [3]). Then, the reduced saturation voltage $\Delta V_{sat}(I_{out})$ was calculated using (6) and the third-order polynomial coefficients, $H = 0.085 I_{out} + 0.22 \cdot 10^{-3} (I_{out})^3$, were found using the formula (11). The modifications of the macromodel of the OPA512 chip were incorporated into its netlist. Using this macromodel, the reduced saturation voltage was calculated and shown in Fig. 3 using dotted line (macromodel 1). This voltage differs from IC's data sheet values over the full range of the output current variation by no more than several dozen of millivolts.

Table 1

Modification of the OPA512 macromodel subcircuit (shown in bold type)

```
.SUBCKT OPA512/MOD ;BB

+ 1 ; non-inverting input

+ 2 ; inverting input

+ 3 ; positive power supply

+ 4 ; negative power supply

+ 55 ;5 ; output

.

.

.

VC 3 53 DC 2.6 ;5

R 5 55 .12

VE 54 4 DC 3.3 ;5

.

.

.

.ENDS
```

An attempt was also made to model the changes in saturation voltages in the OPA512 op-amp using resistor R. The resistor's value calculated using the equation (14) was $R = 0.12\Omega$. Modification of the macromodel (changes in voltages of the VC and VE sources and the resistor R) was incorporated into its netlist; a sample of changes is presented in Table 1. The modified macromodel was used to calculate the saturation voltages. The results are shown with dashed line in Figs. 2 and 3 (macromodel 2). As
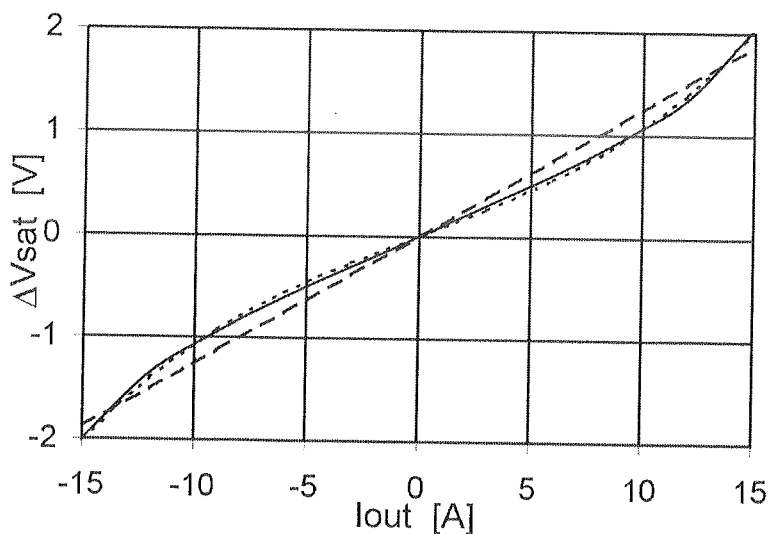
Fig. 3. The dependences between the reduced saturation voltages and the output current for the high-power operational amplifier, type OPA512, and its macromodels: — the data sheet values, ··· the modified macromodel 1, - - - the modified macromodel 2

shown, the saturation voltages differ from the data sheet figures over the full range of the output current variation by less than ± 0.2 V.

4. CONCLUSIONS

The proposed modification of the macromodels of operational amplifiers enables the relations between IC's saturation voltages and output current to be modelled. Commonly used macromodels (provided by IC' manufacturers) take into account no variations of these voltages. Therefore, the modification ensures higher accuracy of modelling the relation between the saturation voltages and the output current. As the result of the modification, the macromodel's accuracy has been improved over wide variations of the load, power supply voltage and input signal. This fact is particularly important for high-power IC amplifiers where errors of simulation calculations for the maximum output power, supply power, power dissipation, etc. may reach several dozens of per-cent in case of the original macromodels.

There is no significant rise in the macromodel complexity as the modification consists in supplementing the macromodel with one controlled source or resistor placed in its output circuit. Data sheets of ICs are sufficient to determine new parameters of the macromodel. To introduce the modification to a macromodel netlist, we need to renumber the output node, to insert one additional line to describe a non-linear

controlled source, H, or a resistor, R, and to change records in two lines of original netlist (change to EMF values of VC and VE sources).

The above example of the modification of high-power operational amplifier macro-model provided by its manufacturer, Burr-Brown Company, proves the practicality of modified macromodel to such applications where IC can operate over a wide range of supply voltages, load resistance and control signals, and where high modelling accuracy for saturation voltages (and also maximum and minimum output voltages) is required. Even in the case of a simple modification of the company's macromodel consisting in an addition of just a single resistor, the modelling error for the relationship between the saturation voltages and the output current is not greater than a fraction of a volt and is by one order of magnitude lower than the error attained for the original macromodel.

5. REFERENCES

1. G. R. B o y l e, B. M. C o h n, D. O. P e d e r s o n, J. E. S o l o m o n: Macromodeling of Integrated Circuit Operational Amplifiers, IEEE J. Solid-State Circuits, 1974, SC-9, pp. 353-364.
2. BURR-BROWN IC Data Book – Linear Products 1996. Burr-Brown Corporation, 1996/1997, pp. 3.2.51-3.2.56.
3. MicroSim PSpice A/D. Ver. 6.3. Reference Manual. MicroSim Corporation, April 1996.
4. J. S t a n c l i k: Accuracy increasing of macromodel for integrated circuit audio power amplifiers, Electronics and Telecommunications Quarterly, 2001, Vol. 47, pp. 463-476.
5. J. S t a n c l i k: Modelling the relation between maximum output voltage and output current of operational amplifiers, Electronics and Telecommunications Quarterly, 2004, Vol. 50, pp. 411-427.

Review of image compression methods of electronic systems

PIOTR BEDNARCZUK

Institute of Computer Science
Lublin University of Technology, Poland
pinarczuk@poczta.onet.pl

The article presents the comparison of digital images algorithms together with the author's realization of an algorithm based on the singular value decomposition method. Images from the field of electronics, from four different categories were examined. Qualitative studies were conducted according to the level of compression, we also conducted comparative studies with other methods of compression. The results of our measures show a little usefulness of such simple-form compression algorithm based on SVD, but they also point to the possibility of using SVD method as one of the stages of more complex image compression algorithm.

Keywords: singular value decomposition, electronic image compression, comparing compression method

1. INTRODUCTION

Together with the increase of hard disks capacity, we have not abandoned complex methods of data compression. Archiving the results in graphic form, widely used in many types of measurements, brings about the need of searching for more effective compression algorithms, including image algorithms. An important argument for this is also widespread sending and presenting graphic images in the Internet. In this article, we evaluated the usefulness of the singular value decomposition in compression of digital images, comparing it to other known compression algorithms.

2. SVD DECOMPOSITION AND RECONSTRUCTION

Singular value decomposition consists in saving a set of data in the form of a smaller set of data with the most effective features of the former, preserving most of the information about that set.

For every real matrix A of the dimensions m by n, where m is the number of lines and n – number of columns, there are such real orthogonal matrices U of the dimensions m by m and V of the dimensions n by n, for which the following dependence is true:

$$U^T AV = \Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_l) \tag{1}$$

where l=min(m,n). If the size of A matrix equals r. then:

$$\sigma_1 \geqslant \sigma_2 \geqslant, ..., \geqslant \sigma_r > 0, \sigma_{r+1} = \sigma_{r+2} =, ..., = \sigma_l = 0 \tag{2}$$

The columns of matrices U and V are called left and right singular vectors, respectively.

Matrix $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_n)$ is a diagonal matrix of singular values σ of dimensions m x n arranged in descending order.

There are many algorithms of numeric matrix decomposition according to singular values. One of them, a numerically stable algorithm proposed by G.H. Golub and C. Reinsch is based on orthogonal transformation of the input matrix using Householder transform to the form of bidiagonal matrix, which is subjected to diagonalization, in the process of which we obtain singular values [1].

To reconstruct the original matrix, we only need to transform the l values to the following form:

$$A = U\Sigma V^T \tag{3}$$

In the process of reconstruction it is not necessary to use entire U, V, Σ matrices, but only their parts, decreasing the order of singular value matrices Σ. We obtain then more or less exact image of the original matrix A dependent on the number of singular value elements σ taken into account.

3. USING SINGULAR VALUE DECOMPOSITION METHOD FOR LOSSY IMAGE COMPRESSION

Using one of the SVD properties, stating that in order to reconstruct the original image with high similarity, it is not necessary to use all the elements of UV Σ matrices obtained during decomposition, but only parts of them, SVD method was used for lossy image compression.

The original image, a bitmap of dimension m-by-n saved as one matrix, has the size represented by the formula:

$$M = b \cdot m \cdot n \tag{4}$$

where: M – memory size in bits, b – number of bytes needed for pixel, m – image width in pixels, n – image height in pixels.

An image subjected to SVD decomposition using r singular elements can be saved as U matrix of dimension m-by-r, V matrix of dimension n-by-r and singular value vector of diagonal matrix Σ of dimension r. So this image can be saved to the memory of size expressed by the formula:

$$M_{SVD} = b_{SVD} \cdot r(1 + m + n) \tag{5}$$

where: M_{SVD} – memory space taken up by an SVD compressed image, r – singular value matrix row, $b_S VD$ – number of bits needed for save one element with SVD compression.

From the above data we can calculate that the difference in memory space needed to save the original and the compressed image is represented in the following way:

$$M_{dif} = M - M_{SVD} = b \cdot m \cdot n - b_{SVD} \cdot r(1 + m + n) \tag{6}$$

We can easily see that this difference in memory size will be positive $M_{dif} > 0$, i.e. compression will take place if the following inequality is fulfilled:

$$b_{SVD} \cdot r(1 + m + n) < b \cdot m \cdot n \tag{7}$$

which conditions that in order to obtain economical use of memory needed to save an image, the singular value matrix row cannot exceed the value expressed by the formula:

$$r < \frac{b \cdot m \cdot n}{b_{SVD}(1 + m + n)} \tag{8}$$

Only if the inequality 7 is fulfilled, we can apply SVD data compression in order to save memory space.

Assuming that:

$$b = b_{SVD} \tag{9}$$

We will obtain a simplified formula:

$$r < \frac{m \cdot n}{1 + m + n} \tag{10}$$

Actually, we cannot make such an assumption. It results from the fact that the original image, a bitmap, is saved using numbers from 0 to 255, i.e. 8-bit numbers. Using singular value decomposition method we obtain three smaller matrices, but matrices of real numbers, at least 8-bytes. To save detailed real number matrices, we need more memory space. That is why memory size needed to store an SVD

compressed image is usually larger than it arises from the size of a matrix itself obtained in the process of decomposition. The percentage compression level can be represented by the following formula:

$$k = (1 - \frac{M_{SVD}}{M}) * 100 = (1 - \frac{b_{SVD} \cdot r(1 + m + n)}{b \cdot m \cdot n}) \cdot 100 \qquad (11)$$

It is therefore necessary to apply additional operations reducing memory space needed to reconstruct information image, e.g. quantization[7][10]. However, quantization introduces additional distortion of the image in the form of quantization noise visible in the image as colour hues of the original image. We can apply more detailed quantization, but it forces the use of more memory space, reducing compression level.

4. ANALYSING SIMILARITY LEVEL OF COMPRESSED IMAGE WITH ORIGINAL IMAGE

To evaluate compression quality using SVD method, as well as in comparing with other compression methods, the following image similarity criteria were used:
- subjective evaluation
- comparing histograms
- determining PSNR ratio

Subjective evaluation consists in visual comparison of images, and their total similarity was determined on the basis of visual identity.

Comparing histograms consists in making diagram of differences in color hues.

Detemining PSNR (*Peak Signal to Noise Ratio*)[7], in order to determine similarity between the original and the compressed image expressed in decibels consisted in calculating the value from the formula:

$$PSNR = 10 \log_{10} \frac{c^2}{MSE} \qquad (12)$$

where: c – image color number coefficient, MSE – mean squared error of difference between two images, represented by the formula:

$$MSE = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} ([I_o(i, j) - I_c(i, j)]^2) \qquad (13)$$

where: m – image width, n – image height, I_o – original image of i, j coordinates, I_c – compressed image of i, j coordinates, i – image width coordinate, j – image height coordinate.

The above methods were not used to determine different values between images, but to specify their similarity. In other words, it was important to determine whether the compared images are similar to such an extent to compare their different compression algorithms.

5. REALIZATION OF COMPRESSION ALGORITHM USING SINGULAR VALUE DECOMPOSITION METHOD

The algorithm using SVD method for lossy image compression was made in Matlab environment. The original bitmap image was loaded to memory and saved in the form of a matrix. SVD decomposition of the original image was made using the existing SVD function. As a result of the decomposition, two matrices U and V and one real type singular value vector S were obtained.
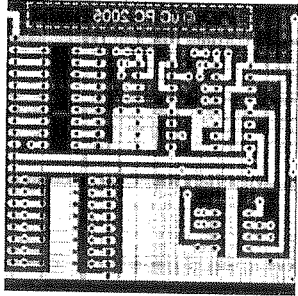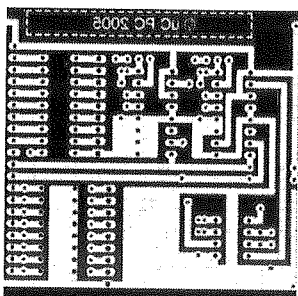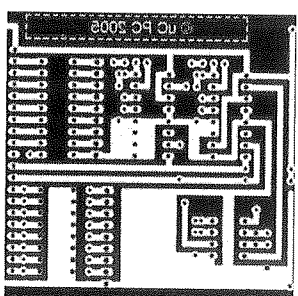
Table 1

Main stages of SVD image compression

	Lp	Algorithm stage	Matlab function
Compression	1	Loading the original image to matrix A	A=imread(bmp)
	2	SVD decomposition, resulting in obtaining matrices U, S, V	[U,S,V] = svd(A)
	3	Uniform quantization of matrices U and V	quantiz
	4	Saving to a file with U, V matrix and S vector	fwrite
Decompr.	1	Loading U, V matrices and S vector from the file	fread
	2	Reconstruction of the original matrix $A = U\,SV^{T}$	$A = U^{*}S^{*}V'$
	3	Saving A matrix to file in the form of a bitmap	imwrite(A, bmp)

Table 2

Influence of the number of quantization levels on the level and quality of compression

Quantization = 4bits	Quantization = 8bits	Quantization = 12bits
Compression level = 82.85%	Compression level = 66.19%	Compression level = 49.53%
Quantization noise : 0.0131	Quantization noise : 0.000773	Quantization noise : 0.0000495
PSNR = 17.13 dB	PSNR = 27.60 dB	PSNR = 33.58 dB

Then, the matrices were subjected to quantization in order to change data type of U and V matrices. The quantization process enables us to reflect a real value matrix in the form of an integer matrix, reducing the use of memory needed to save it. It is possible by assigning every real value, according to their range, to the number of a given range. It results in the irreversible loss of information, because rounding leads to irreproducible loss of precision when reconstructing the original image. So, quantization introduces additional distortions of the image in the form of quantization noise visible in the image as color hues of the original image. We can make quantization more precise, but this forces us to use more memory, which eventually reduces the level of compression. Table 2 presents the relationship between the level and quality of compression and the number of quantization ranges. To reconstruct the original image, it is enough to subject the matrices obtained during decomposition to the reconstruction process described by dependence 3.

6. RELATIONSHIP BETWEEN COMPRESSION QUALITY AND COMPRESSION LEVEL IN SINGULAR VALUE DECOMPOSITION COMPRESSION

Singular value decomposition has already been examined in relation to its use in many different images compression. Here, we examined its usefulness in electronic image compression. Almost all types of electronic images have the following common features:

- small number of colours
- huge unicoloured areas
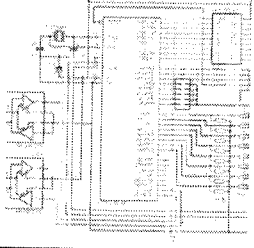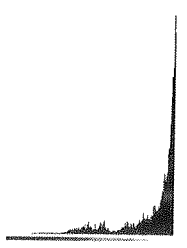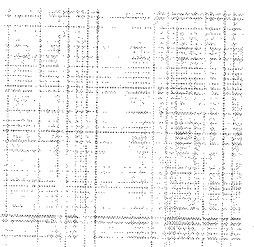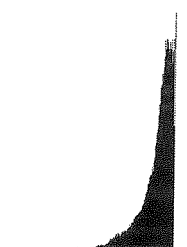- thin lines
- include text
- include details.
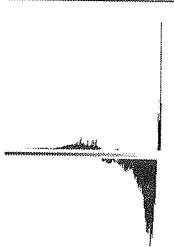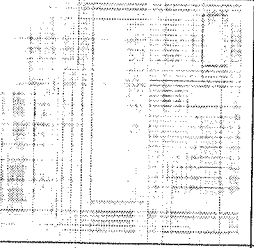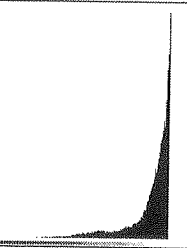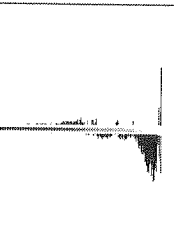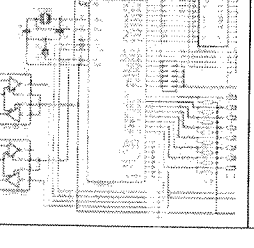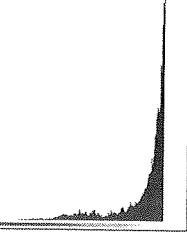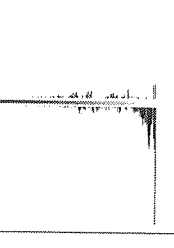
These features impose high requirements on compression algorithms and because of those requirements, lossless compression methods are usually applied to image compression.

According to formula 6 an 11, the highest level of compression occurs when the singular value matrix row is the smallest. However, for low values of r, i.e. for small singular value matrix row decomposition is so lossy that the compressed image is significantly different from the original one. It is visible especially in histograms and results from a small PSNR similarity ratio. Therefore it is needed to find balance between compression level and image quality. In table 1 a comparison of the original image with compressed images using different number of singular values is presented. As we can easily see, the bigger is the singular value matrix row, the smaller are histogram differences, higher PSNR similarity level and lower quality level. The differences in histogram will always occur, because it is a lossy compression, it is only about that the compressed image contained necessary pieces of information from the original

image. The compressed image is a bitmap of 270 kB size, RGB colour mode, 24 bits of colours for pixel, 300×300 pixels.

Table 3

Comparing images compressed using different singular values, histograms and PSNR ratio

Signatures used in the table:	Original image:	Original image histogram	Original image data
r - rank of singular value; decomposition matrix S; k – level compression; PSNR – images similarity ratio			Image size: 300 by 300 pixel; Colours: 24bits by pixel; Colour mode: RGB; Original image file size: 370kB
Compression ratios:	Compressed image	Compressed image histogram.	Differences in histograms
r=1 k=98.95% PSNR=18.13 dB			
r=10 k=89.87% PSNR=23.35 dB			
r=100 k=15.75% PSNR=31.52 dB			

7. COMPARING SINGULAR VALUE DECOMPOSITION COMPRESSION WITH OTHER LOSSY AND LOSSLESS DATA COMPRESSION METHODS

Currently, electronic images are compressed using methods widely applied to other image categories. However, because of specific features of those images listed in chapter 6, we look for more effective methods. Six compression algorithms were compared, three algorithms of lossless compression:

Table 4

Comparing different compression methods with singular value decomposition method

Comparable electronic image categories and their data after SVD compression.	Image example
Image category: electronic schematic diagrams Quantization: 12bits Quantization noise: 0.0000673 Number of singular values: 60 Percentage level compression: 39.4395 % PSNR ratio: 33.27 dB	
Image category: printed circuit boards (PCB) Quantization: 9bits Quantization noise: 0.000391 Number of singular values: 60 Percentage level compression: 54.96% PSNR ratio: 35.26 dB	
Image category: digital oscilloscope oscillogram. Quantization: 10bits Quantization noise: 0.000501 Number of singular values: 60 Percentage level compression: 49.44% PSNR ratio: 34.08 dB	
Image category: electronic block diagrams Quantization: 10bits Quantization noise: 0.000227 Number of singular values: 70 Percentage level compression: 41.02% PSNR ratio: 33.09 dB	

- LZW – dictionary coding scheme, used in GIF formats
- LZ77 – improved LZW, used in PNG format
- JPEG lossless – used in JPEG format

and three algorithms of lossy compression:

- DCT – discrete cosine transform, used in JPEG format
- DWT – discrete wavelet transform, used in JPEG 2000 format
- SVD – singular value decomposition algorithm, named ISVD format.

To compare, four electronic image categories were chosen: schematic diagram, PCBs, oscillograms, block diagrams. For each of the categories, a minimum of three images were examined. The mean results of our measures are presented in table 4 and bar graph 1.
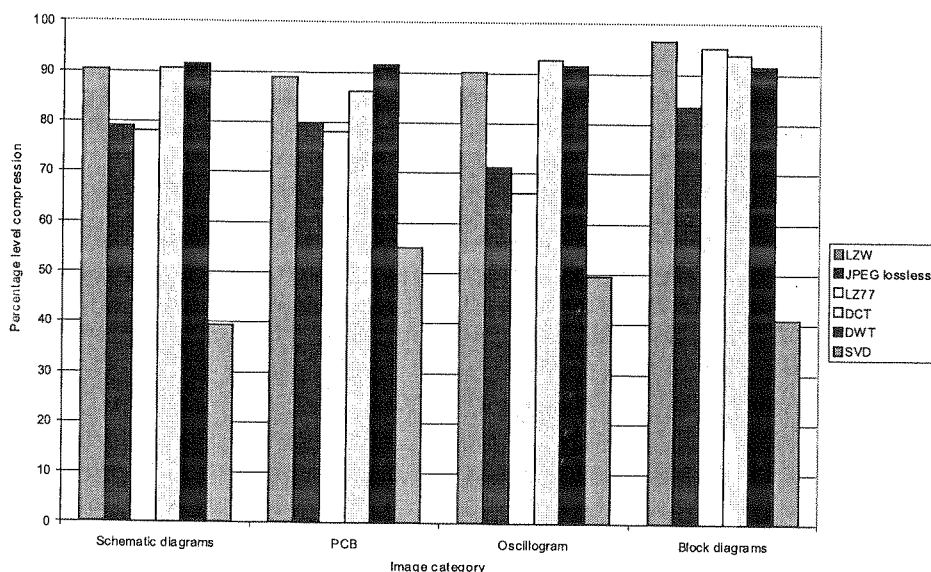

Fig. 1. Comparable electronical image categories and their data after SVD compression

8. CONCLUSION

In the work, a method of electronic images compression using singular value decomposition was presented. The qualitative (chapter 6) and comparative (chapter 7) studies conducted show that SVD compression in its simple form does not produce better results than the existing lossy image compression methods. What is more, such a compression produces even worse results than those of lossless compressions, which are ideal for image compression. However, the best level of compression for digital images ensures LZW dictionary method used in GIF formats. It results from the specific digital image features. e.g. small number of colours. It results mainly from the

differences between the data type of the original image matrix and data type of the SVD decomposition matrix. However, usually we do not use only one method while compressing, but a whole set of algorithms decreasing image size in several stages. So, the SVD matrix decomposition method can be used as one of the image compression stages, which can be supported by other methods decreasing memory size, e.g. quantization, coding or transform [3][4]. In this case, there is a chance that compression method at least comparable to the results of other existing image compression methods can be obtained.

9. REFERENCES

1. H. G. G o l u b, C. F. V a n L o a n: *Matrix Computations THIRD EDITION* The Johns Hopkins University Press, Baltimore and London 1996.
2. J. C h e n: *Image Compression with SVD*, Scientific Computation ESC 289K, 13.XII.2000.
3. A. D e p e n a, S. A h a l t: *A hybrid DCT-SVD image coding algorithm*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY 12(2), pp. 114-121 FEB 2002.
4. A. D r o z d e m: *Wprowadzenie do kompresji danych*, WNT, Warszawa 1999.
5. J. K i t l i ń s k i: *Kompresja sygnałów*, z Materiały II Seminarium Doktorantów WE PL, ss. 70-76 , Lublin 2002.
6. L. K n o c k a e r t, B. D e B a c k e r, D. D e Z u t t e r: *SVD Compression, unitarny transforms and computation complexity*. IEEE TRANSACTIONS ON SIGNAL PROCESSING 47(10), pp. 2724-2729 OCT 1999.
7. E. K o r n a t o w s k i: *Uniwersalna miara oceny jakości obrazów cyfrowych*, Kwartalnik Elektroniki i telekomunikacji, 2006, 52, z. 4, ss. 511-520.
8. J. L e v i n e: *Programowanie plików graficznych w C/C++*, Translator, Warszawa 1994.
9. D. W a t k i n s, A. S a d u n, S. M a r e n k a: *Nowoczesne metody przetwarzania obrazów* WNT, Warszawa 1995.
10. A. Z a l e w s k i, R. C e g i e ł a: *Matlab*, NAKOM Poznań 2002.

C
is to
electr
butior
the co
which
the ef
T
and a

A bipolar space charge problem for solids including a secondary electron emission

BRONISŁAW ŚWISTACZ

The Wrocław University of Technology,
Institute of Electrical Engineering Fundamentals,
Wybrzeże St. Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: bronislaw.swistacz@pwr.wroc.pl

In this paper a new mathematical model for a space charge transport through a solid placed between the two electrodes is presented. Using new equations for allowed electron-hole transitions, the effect of the light on the electric field distribution and on the shape of current - voltage characteristic is determined. For a space charge distribution some new singular solutions are obtained. Also, some new shapes of current - voltage characteristic with negative resistance are determined. In this paper it is found that the system can act as an n-p-n or p-n or n-n blocking diode.

Keywords: double injection, current flow problem, trapping levels

1. INTRODUCTION

One of the fundamental problems of a macroscopic theory of electric conduction is to find the total concentration of charge carriers in a solid placed between the two electrodes. In this paper, we will assume that the divergence of the electric field distribution will be defined by the total concentration of carriers. Also, we will suppose that the contact processes have an influence on the shape of the electric field distribution, which corresponds to a current – voltage characteristic. Additionally, we will determine the effect of the light on the space charge density distributions.

The purpose of this work is to find a relation between the space charge distributions and a current-voltage characteristic of the metal – solid – metal system.

2. THE NEW MATHEMATICAL MODEL

In this paper, we will consider a solid in which the system of atoms is very chaotic and the different structural defects (pollutants and impurities, the Frenkl defects) and dislocations occur. This property will be characterised by the Zeemann internal effect (the splitting of the total energy of an orbital electron). For an orbital electron in the given atom, we will assume that the total energy (the sum of the positive kinetic energy and the negative potential energy of the electric field of the positive nucleus) is negative and that the zero reference level is at a finite distance from the nucleus. These properties are shown in Fig. 1 and in Fig. 2. In the case of an isolated atom (here in Fig. 1a it is symbolised by the A atom) the total energy of an orbital electron is negative for any distance from the nucleus. When an orbital electron absorbs a portion of the kinetic energy of a photon or a phonon, the total energy of the electron increases. The inverse case is when an orbital electron can lose a portion of the total energy. This is caused by the Coulomb force interaction between the orbital electron and the positive nucleus. With such the physical assumptions we will determine some internal and boundary processes (which are shown in Fig. 1-3). First let us return to Fig. 1a. Here, let us take into account the two valence electrons of the isolated atom A.
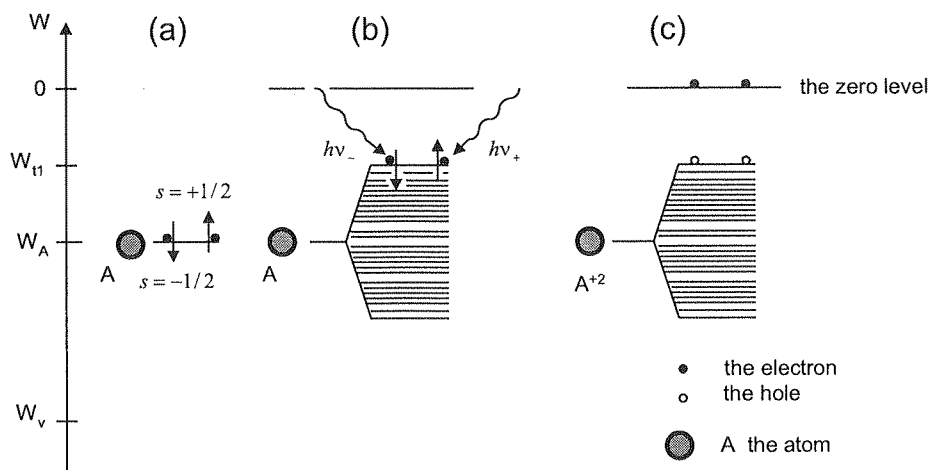

Fig. 1. The splitting of the energy state and an electron-hole generation: (a) two valence electrons of the A isolated atom; (b) the A atom is packed closely together with many other adjacent atoms and two valence electrons fill the W_{t1} higher energy level as well as these electrons absorb the energy $h\nu_+$ and $h\nu_-$ of incident photons; (c) the same situation for the A atom in a solid and two empty states are left by two valence electrons. W is the total energy of an electron, h is the Planck constant and $\nu_+; \nu_-$ denote the frequencies of photons

These electrons occupy the W_A energy state with the spin number $s = \pm 1/2$. When this atom is packed into a solid (that is, the A atom presents a pollutant or an impurity in Fig. 1b), these electrons can occupy the higher energy state. Such the property can be
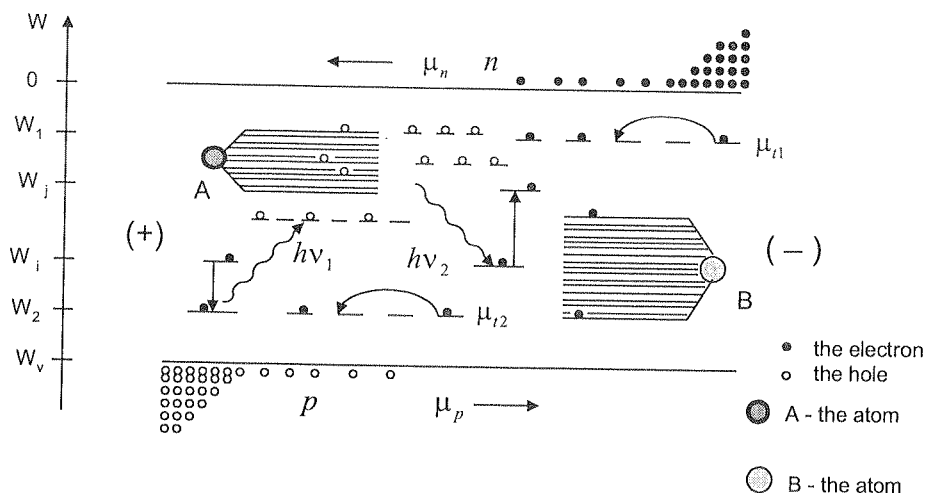
Fig. 2. The energy diagram for the internal and boundary processes in a solid placed between the (+) anode and the (−) cathode. W denotes the total energy of an electron

explained by the splitting of the W_A energy state. In other words, this is caused by the different internal interactions (for example, the Zeemann internal effect) between the A atom and many adjacent atoms. Next, when a portion of the kinetic energy is given to the two valence electrons, these electrons can become free. The two empty energy states (which are left by the electrons) represent the two holes (this effect is shown in Fig. 1c). In Fig. 2 some internal and boundary processes are presented when a solid is placed between the anode and the cathode. Here the pollutants or the impurities are symbolised by the A or B atoms. Between the zero and valence levels many energy states (the trapping states) are available for the holes and for the electrons. When an orbital electron absorbs a photon, this electron can pass from the valence level to the zero level via trapping levels. In Fig. 2 this is symbolised by the photon energy $h\nu_2$. The inverse case is when an orbital electron can pass from the higher trapping level to the lower trapping level and an energy portion $h\nu_1$ is emitted. Here, h is the Planck constant and $\nu_1 ; \nu_2$ denote the photon frequencies. Analogously, we are known as the allowed transitions for the trapped holes. Such the allowed electron – hole transitions are called carrier generation - recombination processes. When an external electric field supplies the different kinetic energy to the two adjacent atoms (in Fig. 2 this property is characterised by the mobilities μ_{t1} and μ_{t2}), the trapped electron can pass from trap to trap in the given trapping level. When the external electric field is applied the electron can pass from the cathode into a solid. Moreover, this electron can become free and have the mobility μ_n. Similarly, the hole injection occurs when the valence electron can pass from the bulk into the anode and the valence state is empty. When the external electric field gives a portion of kinetic energy to the valence electron of an adjacent atom, this electron can pass from the atom to the given atom and can fill the empty

state on the valence level. Such the transport is characterised by the mobility μ_p in Fig. 2.
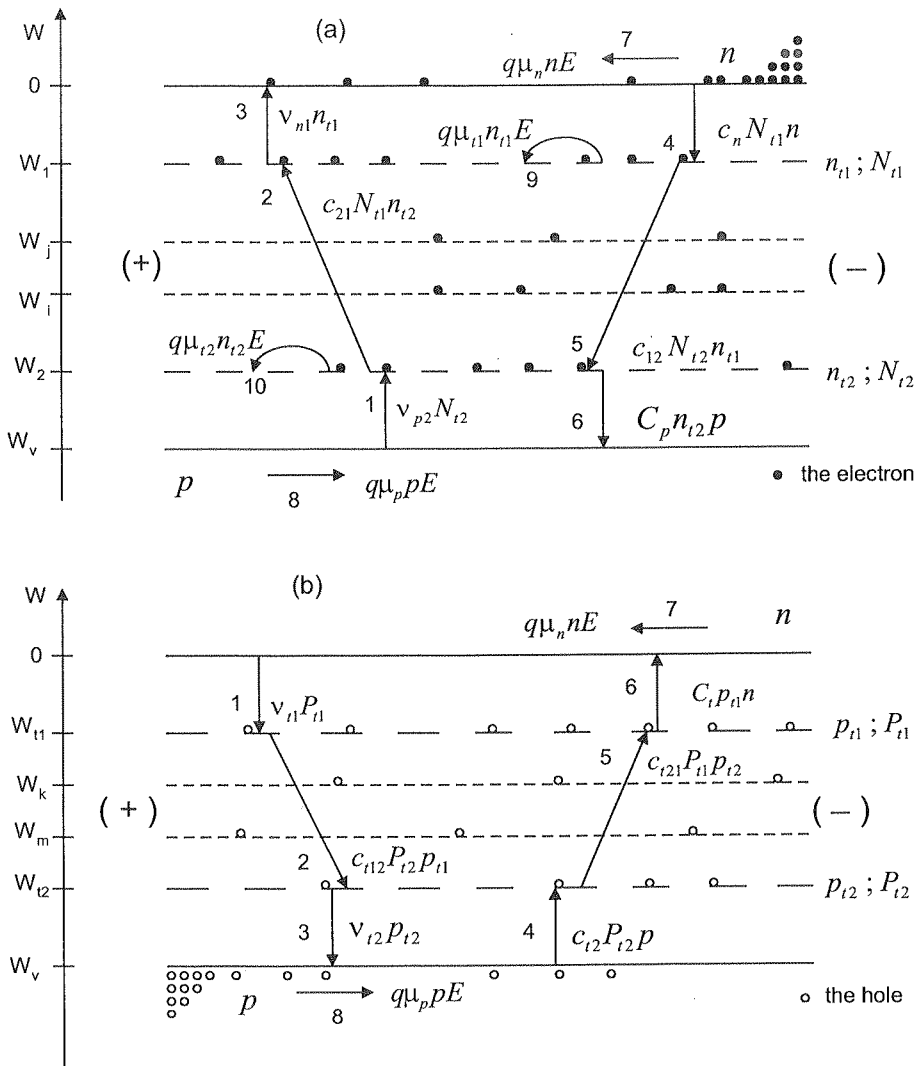
Fig. 3. The energy diagram illustrating a model (1)-(7): (a) allowed electron transitions; (b) allowed hole transitions. W is the total energy of an electron

For our mathematical considerations, the trapping levels will be grouped into the four permissible energy levels (this concept is shown in Fig. 3). With this assumption, the so-called effective parameters such as the frequency parameters c_{21} and c_{t12} as well as the recombination parameters c_{12} and c_{t21} will be used. For the trapped electrons, the concentrations of traps in the first and second trapping level will be represented

Fig. 4. A planar capacitor is connected with an external resistance and a voltage source. Here,(+);(−) denote the terminals of a voltage source
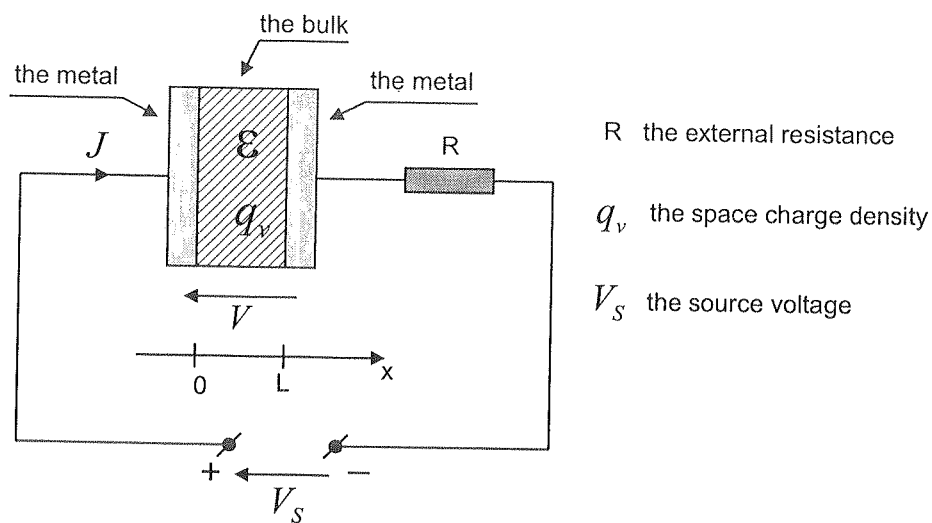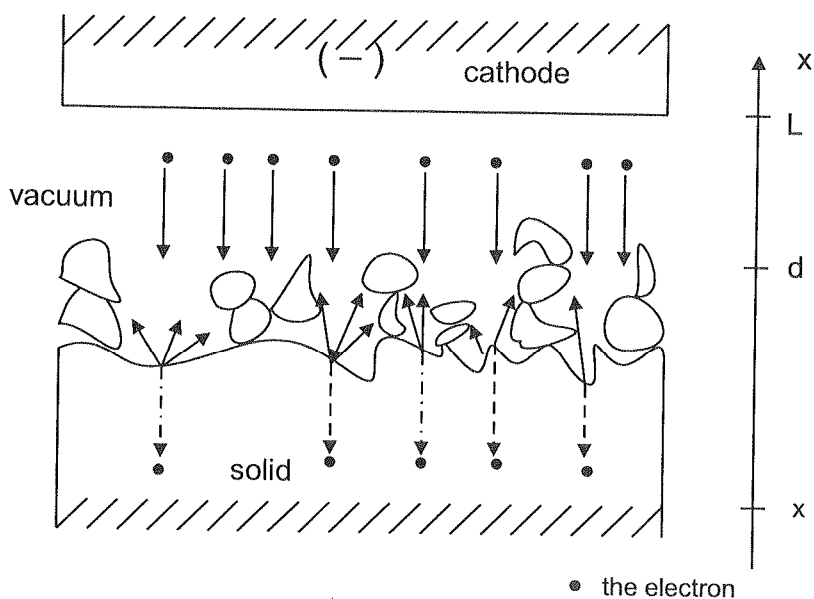

Fig. 5. An electron transport between the cathode and a solid plane makes a model (9)-(12)

by N_{t1} and N_{t2}, respectively. Analogously, for the trapped holes, the concentrations of traps in the first and second trapping level will be equal to P_{t1} and P_{t2}, respectively. The system of atoms will be treated as an unlimited reservoir of traps, that is $P_{t1} >> p_{t1}$; $P_{t2} >> p_{t2}$; $N_{t1} >> n_{t1}$ and $N_{t2} >> n_{t2}$. The metal-solid-metal system will be represented by a planar capacitor system with the anode $x = 0$ and the cathode $x = L$ (the system is shown in Fig. 4). Also, L denotes the distance between the electrodes. Moreover, we will assume that the diffusion current is negligible [1-6]. For a solid, we will assume that the polarisation effect is characterised by the dielectric constant ε. Additionally, the carrier mobilities are independent of the electric field intensity E [7-9]. For the $x = L$ contact, we will consider a special case of electron emission from the cathode into a solid (this property is shown in Fig. 5). Here, for a solid, we assume that the contact structure is strongly nonhomegenous. On these conditions a secondary electron emission occurs [10-14]. This property will be represented by a vacuum capacitor with the contact voltage V_2 and a distance parameter denoted by $L-d$ [15]. In our considerations, for the planar capacitor system, the basic equations such as the Gauss equation, the continuity equation, the generation-recombination equations and the field integral will be used. On this basis, the space charge transport through the bulk is described by

$$\frac{\varepsilon}{q}\frac{\partial E(x,t)}{\partial x} = p(x,t) + p_{t1}(x,t) + p_{t2}(x,t) - (n(x,t) + n_{t1}(x,t) + n_{t2}(x,t)) \qquad (1)$$

$$\frac{\partial}{\partial x}\left\{\left[\mu_n n(x,t) + \mu_p p(x,t) + \mu_{t1} n_{t1}(x,t) + \mu_{t2} n_{t2}(x,t)\right]E(x,t)\right\} + \frac{\partial p(x,t)}{\partial t}$$

$$+\frac{\partial p_{t1}(x,t)}{\partial t} + \frac{\partial p_{t2}(x,t)}{\partial t} - \frac{\partial n(x,t)}{\partial t} - \frac{\partial n_{t1}(x,t)}{\partial t} - \frac{\partial n_{t2}(x,t)}{\partial t} = 0 \qquad (2)$$

$$\frac{\partial n(x,t)}{\partial t} = v_{t1}P_{t1} + v_{n1}n_{t1}(x,t) - c_n N_{t1}n(x,t) - C_t n(x,t)p_{t1}(x,t) + \frac{\partial}{\partial x}\left[\mu_n n(x,t)E(x,t)\right] \quad (3)$$

$$\frac{\partial p_{t1}(x,t)}{\partial t} = v_{t1}P_{t1} + c_{t21}P_{t1}p_{t2}(x,t) - c_{t12}P_{t2}p_{t1}(x,t) - C_t n(x,t)p_{t1}(x,t) \qquad (4)$$

$$\frac{\partial p_{t2}(x,t)}{\partial t} = c_{t12}P_{t2}p_{t1}(x,t) - c_{t21}P_{t1}p_{t2}(x,t) - v_{t2}p_{t2}(x,t) + c_{t2}P_{t2}p(x,t) \qquad (5)$$

$$\frac{\partial n_{t1}(x,t)}{\partial t} = c_{21}N_{t1}n_{t2}(x,t) - v_{n1}n_{t1}(x,t) - c_{12}N_{t2}n_{t1}(x,t) + c_n N_{t1}n(x,t)$$

$$+\frac{\partial}{\partial x}\left[\mu_{t1}n_{t1}(x,t)E(x,t)\right] \qquad (6)$$

$$\frac{\partial n_{t2}(x,t)}{\partial t} = v_{p2}N_{t2} + c_{12}N_{t2}n_{t1}(x,t) - c_{21}N_{t1}n_{t2}(x,t) - C_p p(x,t)n_{t2}(x,t)$$
$$+\frac{\partial}{\partial x}\left[\mu_{t2}n_{t2}(x,t)E(x,t)\right] \tag{7}$$

with the voltage condition for a solid

$$\int_0^d E(x,t)dx = V_1 \tag{8}$$

And also, the electron transport between the electrode and a solid is as follows

$$\varepsilon_0 \frac{\partial E_0(x,t)}{\partial x} = -qn_0(x,t) \tag{9}$$

$$\frac{\partial}{\partial x}\left[n_0(x,t)\vartheta(x,t)\right] = \frac{\partial n_0(x,t)}{\partial t} \tag{10}$$

$$-2q\int_L^x E_0(x,t)dx = m\vartheta^2(x,t) - m\vartheta^2(L,t) \tag{11}$$

$$\int_d^L E_0(x,t)dx = V_2 \tag{12}$$

Here, $q = 1.602 \cdot 10^{-19}C$, $\varepsilon_0 = 8.85 \cdot 10^{-12}F/m$; E_0 is the electric field intensity in a vacuum, m is the effective mass of an electron, x is the distance from the electrode, t is the time, ϑ is the velocity of an electron; n, n_0 and p are the free hole and electron concentrations, respectively; $n_{t1}, n_{t2}, p_{t1}, p_{t2}$ are the concentrations of the trapped holes and electrons, respectively; μ_{t1} and μ_{t2} are the mobilities of trapped electrons; $v_{p2}, v_{n1}, v_{t1}, v_{t2}$ denote the frequency parameters; c_n, C_p, C_t, c_{t2} denote the recombination parameters. The applied voltage V between the electrodes of a planar capacitor is

$$V = V_1 + V_2 \quad \text{and} \quad V = \text{co}\,nst. > 0 \tag{13}$$

With the above equations (1)-(13) we shall define the stationary state and we shall find different current-voltage characteristics.

3. THE STATIONARY STATE

Letting $(\partial/\partial t) = 0$ in (1)-(12), the stationary state of the space charge transport is determined by the following equations

$$\frac{\varepsilon}{q}\frac{dE(x)}{dx} = p(x) + p_{t1}(x) + p_{t2}(x) - (n(x) + n_{t1}(x) + n_{t2}(x)) \tag{1a}$$

$$J = qE(x)\left[\mu_n n(x) + \mu_p p(x) + \mu_{t1} n_{t1}(x) + \mu_{t2} n_{t2}(x)\right]; J = const. \tag{2a}$$

$$v_{t1}P_{t1} + v_{n1}n_{t1}(x) - c_n N_{t1}n(x) - C_t n(x)p_{t1}(x) + \frac{d}{dx}\left[\mu_n n(x)E(x)\right] = 0 \tag{3a}$$

$$v_{t1}P_{t1} + c_{t21}P_{t1}p_{t2}(x) - c_{t12}P_{t2}p_{t1}(x) - C_t n(x)p_{t1}(x) = 0 \tag{4a}$$

$$c_{t12}P_{t2}p_{t1}(x) - c_{t21}P_{t1}p_{t2}(x) - v_{t2}p_{t2}(x) + c_{t2}P_{t2}p(x) = 0 \tag{5a}$$

$$c_{21}N_{t1}n_{t2}(x) - v_{n1}n_{t1}(x) - c_{12}N_{t2}n_{t1}(x) + c_n N_{t1}n(x) + \frac{d}{dx}\left[\mu_{t1}n_{t1}(x)E(x)\right] = 0 \tag{6a}$$

$$v_{p2}N_{t2} + c_{12}N_{t2}n_{t1}(x) - c_{21}N_{t1}n_{t2}(x) - C_p p(x)n_{t2}(x) + \frac{d}{dx}\left[\mu_{t2}n_{t2}(x)E(x)\right] = 0 \tag{7a}$$

$$\int_0^d E(x)dx = V_1 \tag{8a}$$

$$\varepsilon_0 \frac{d\,E_0(x)}{dx} = -qn_0(x) \tag{9a}$$

$$J = qn_0(x)\vartheta(x) \tag{10a}$$

$$-2q\int_L^x E_0(x)dx = m\vartheta^2(x) - m\vartheta^2(L) \tag{11a}$$

$$\int_d^L E_0(x)dx = V_2 \tag{12a}$$

Here, J is the current density. The space charge transport through the system will be characterised by a current density-voltage function in the form $J = J(V)$ or voltage – current density $V = V(J)$. In order to find these functions, we have to define the boundary conditions describing the mechanisms of carrier injection from the anode $x = 0$ and the cathode $x = L$ into the bulk.

First let us combine (9a)-(12a) in order to find a function $E_0(x)$. From (9a) and (11a) it follows that the derivative $dE_0/d\vartheta$ is

$$\frac{dE_0}{d\vartheta} = \frac{dE_0/dx}{d\vartheta/dx} = \frac{mJ}{q\varepsilon_0 E_0} \tag{14}$$

Hence, with the assumption $E_0(\vartheta = 0) = 0$ we obtain the electron velocity in a vacuum

$$\vartheta = \frac{q\varepsilon_0}{2mJ} \cdot E_0^2 \tag{14a}$$

Next, on this basis we can find a space charge density distribution $\varepsilon_0 \dfrac{dE_0}{dx}$. Using (9a) and (10a), we have

$$\varepsilon_0 \frac{dE_0}{dx} = -\frac{J}{\vartheta} = -\frac{2mJ^2}{q\varepsilon_0 E_0^2} \tag{14b}$$

Thus, a function $E_0(x)$ is in the form of

$$E_0(x) = \left(\frac{6mJ^2}{q\varepsilon_0^2}(L - x) + E_0^3(L) \right)^{\frac{1}{3}} \tag{15}$$

Substituting (15) into the voltage integral (12a), the contact voltage V_2 can be expressed by the boundary value $E_0(L)$ in the form

$$V_2 = \frac{q\varepsilon_0^2}{8mJ^2} \left\{ \left[\frac{6mJ^2}{q\varepsilon_0^2}(L - d) + E_0^3(L) \right]^{\frac{4}{3}} - E_0^4(L) \right\} \tag{16}$$

Now, for a solid region $x \in < 0, d >$, we will assume that generation processes are dominant. According to (1a)-(8a), this property will be expressed by the following condition

$$c_n = c_{12} = C_p = C_t = 0 \tag{17}$$

Under conditions of (17), as a function of x, the hole current density distribution is linear

$$q\mu_p p(x)E(x) = q\left(v_{p2}N_{t2} + v_{t1}P_{t1}\right) \cdot x + C_1 \tag{17a}$$

where C_1 is the constant of integration. For the trapped holes we will introduce into our analysis the following time parameters

$$c_{t21}P_{t1} = c_{t12}P_{t2} = \tau_{t12}^{-1}; \quad c_{t2}P_{t2} = \tau_{t2}^{-1} \tag{18}$$

Hence, on the basis of (4a) and (5a) we ascertain that the trapped hole concentration depends on the free hole concentration

$$p_{t1}(x) + p_{t2}(x) = \frac{2\tau_{t2}^{-1}}{v_{t2}} \cdot p(x) + v_{t1}P_{t1}\left(\frac{2}{v_{t2}} + \tau_{t12}\right) \tag{18a}$$

For carrier generation processes, we additionally assume that the trapped electrons in the second trapping level are immobile, that is $\mu_{t2} = 0$. With this assumption, from (7a) and from (17) it follows that the concentration $n_{t2}(x)$ is uniform

$$n_{t2}(x) = v_{p2}N_{t2}\tau_{21}; \tau_{21}^{-1} = c_{21}N_{t1} \tag{19}$$

For our carrier generation problem, we will consider a case in which the time parameters satisfy the following equation

$$\tau_{21} = \frac{v_{t1}P_{t1}}{v_{p2}N_{t2}}\left(\frac{2}{v_{t2}} + \tau_{t12}\right) \tag{20}$$

With (20) we will determine the right hand side of (1a). First, let us notice that there is

$$p(x) + p_{t1}(x) + p_{t2}(x) - n_{t2}(x) = \left(1 + \frac{2\tau_{t2}^{-1}}{v_{t2}}\right) \cdot p(x) \tag{20a}$$

Now, instead of (1a) we have

$$\frac{\varepsilon}{q}\frac{dE(x)}{dx} = \left(1 + \frac{2\tau_{t2}^{-1}}{v_{t2}}\right) \cdot p(x) - n(x) - n_{t1}(x) \tag{20b}$$

In order to determine a function $E(x)$, we introduce into (20b) the new variables

$$\lambda_1 = \frac{q\mu_p p(x)E(x)}{J}; \ \lambda_2 = \frac{q\mu_p n(x)E(x)}{J}; \ \lambda_3 = \frac{q\mu_p n_{t1}(x)E(x)}{J} \tag{21}$$

Thus, (1a) and (2a) are respectively written as

$$\frac{\varepsilon\mu_p}{J}E\frac{dE}{dx} = \left(1 + \frac{2\tau_{t2}^{-1}}{v_{t2}}\right) \cdot \lambda_1 - \lambda_2 - \lambda_3 \tag{22}$$

$$1 = \lambda_1 + r_1\lambda_2 + r_2\lambda_3; \ r_1 = \frac{\mu_n}{\mu_p}; \ r_2 = \frac{\mu_{t1}}{\mu_p} \tag{23}$$

For our analysis we assume that $v_{n1} \approx v_{t1}$ and $P_{t1} \approx N_{t1}$. With such internal parameters we have a function $\lambda_2(x)$

$$\lambda_2 = -\frac{q v_{t1} P_{t1}}{r_1 J} \cdot x + C_2 \tag{24}$$

Here C_2 is a constant of integration. Combining (22)-(24), we get

$$\frac{\varepsilon \mu_p}{J} E \frac{dE}{dx} = \left(1 + \frac{1}{r_2} + \frac{2\tau_{t2}^{-1}}{v_{t2}}\right) \cdot \lambda_1 - \frac{r_2 - r_1}{r_2}\lambda_2 - \frac{1}{r_2} \tag{25}$$

Next, taking into account (24) and (21) as well as (17a), (25) is written as

$$E\frac{dE}{dx} = \alpha_0 x + C_3 \tag{26}$$

where

$$\alpha_0 = \frac{q}{\varepsilon \mu_p} \cdot \left(1 + \frac{1}{r_2} + \frac{2\tau_{t2}^{-1}}{v_{t2}}\right)\left(v_{p2}N_{t2} + v_{t1}P_{t1}\right) + \frac{q(r_2 - r_1)}{\varepsilon \mu_p r_1 r_2}v_{t1}P_{t1} \tag{27}$$

and C_3 denotes a constant of integration. We can notice that there is $\alpha_0 > 0$. In the particular case, when $\mu_p = \mu_n = \mu_{t1}$, (27) results in

$$\alpha_0 = \frac{2q\theta}{\varepsilon \mu_p}\left(v_{p2}N_{t2} + v_{t1}P_{t1}\right); \theta = 1 + \frac{\tau_{t2}^{-1}}{v_{t2}} \tag{27a}$$

Let us notice that the equivalent form of (26) is

$$\frac{dE}{dx} = \frac{\alpha_0(x - x_0)}{E}; C_3 = -\alpha_0 \cdot x_0 \tag{28}$$

Here x_0 denotes a new constant of integration. From (28) we obtain two singular solutions, namely

$$E_1(x) = -\sqrt{\alpha_0} \cdot (x - x_0) \quad \text{and} \quad E_2(x) = \sqrt{\alpha_0} \cdot (x - x_0) \tag{29}$$

Another solution follows from (26), that is

$$E(x) = \sqrt{\alpha_0 \cdot x^2 + b \cdot x + E^2(0)} \tag{30}$$

where b is a constant of integration. Using (29)-(30) and (16) and the applied voltage $V = V_1 + V_2$, we can define the current-voltage characteristic in the following parametric form

$$V = V_1[J, E(0)] + V_2[J, E_0(L)] \quad \text{and} \quad J = f_0[E(0)] \quad \text{and} \quad J = f_L[E_0(L)] \tag{31}$$

The boundary function f_0 and f_L describe the mechanisms of carrier injection from the anode and the cathode into the interior of the system. First, let us consider a singular solution $E_1(x)$. The integral (8a) is

$$V_1 = \int\limits_0^d E_1(x)dx = -\frac{\sqrt{\alpha_0}}{2}\left(d^2 - 2x_0d\right) \quad \text{for} \quad x_0 \geq d \qquad (32)$$

that is

$$V_1 = -V_{10} + E_1(0) \cdot d \;\; ; \;\; E_1(0) = \sqrt{\alpha_0} \cdot x_0 \;\; ; \;\; J = f_0[E_1(0)] \qquad (32a)$$

where

$$V_{10} = \frac{\sqrt{\alpha_0}}{2}d^2 \qquad (32b)$$

A formula (32a) is equivalent to

$$J = f_0\left[(V_1 + V_{10})/d\right] \;\; ; \;\; \frac{E_1(0)}{\sqrt{\alpha_0}} \geq d \qquad (33)$$

Now, we will define a current density – voltage characteristic (31) when a boundary function $J = f_0[E_1(0)]$ is strongly monotonic. Using (16) for (31), we have (a case of an n-n junction)

$$V = -V_{10} + f_0^{-1}(J) \cdot d + \frac{q\varepsilon_0^2}{8mJ^2}\left\{\left[\frac{6mJ^2}{q\varepsilon_0^2}(L - d) + E_0^3(L)\right]^{\frac{4}{3}} - E_0^4(L)\right\} \qquad (34)$$

$$\text{and} \;\; J = f_L[E_0(L)]$$

Here $f_0^{-1}(J)$ denotes the inverse function. A boundary function $J = f_L[E_0(L)]$ describes the mechanism of electron injection from the cathode into a vacuum. Analogously, proceeding with a singular solution $E_2(x)$, (33) and (34) are replaced by (a case of a p-n junction)

$$J = f_0\left[(V_1 - V_{10})/d\right] \;\; ; \;\; x_0 \leq 0 \qquad (35)$$

as well as

$$V = V_{10} + f_0^{-1}(J) \cdot d + \frac{q\varepsilon_0^2}{8mJ^2}\left\{\left[\frac{6mJ^2}{q\varepsilon_0^2}(L - d) + E_0^3(L)\right]^{\frac{4}{3}} - E_0^4(L)\right\} \qquad (36)$$

$$\text{and} \;\; J = f_L[E_0(L)]$$

For singular solutions (29), another case is when there is $0 \leq x_0 \leq d$. Thus, in a solid the electric field distribution can be defined as follows (a case of an n-p-n junction)

$$E(x) = \begin{cases} E_1(x) \; ; \; 0 \le x \le x_0 \\ E_2(x) \; ; \; x_0 \le x \le d \end{cases} \qquad (37)$$

Consequently, the integral (8a) is written as

$$V_1 = \int\limits_0^{x_0} E_1(x)dx + \int\limits_{x_0}^d E_2(x)dx$$

Hence, on the basis of (37) and (29), we have

$$V_1 = \frac{E^2(0)}{\sqrt{\alpha_0}} - d \cdot E(0) + V_{10} \; ; \; \frac{1}{2}V_{10} \le V_1 \le V_{10} \qquad (38)$$

A voltage parameter V_{10} is expressed by (32b). Taking into account (38) and (16), a function (31) results in

$$V = V_{10} - E(0) \cdot d + \frac{E^2(0)}{\sqrt{\alpha_0}} + \frac{q\varepsilon_0^2}{8mJ^2} \left\{ \left[\frac{6mJ^2}{q\varepsilon_0^2}(L-d) + E_0^3(L) \right]^{\frac{4}{3}} - E_0^4(L) \right\} \qquad (39)$$

$$\text{and} \quad J = f_0[E(0)] \quad \text{and} \quad J = f_L[E_0(L)]$$

Here $J = f_0[E(0)]$ describes the mechanism oh hole injection from the anode into the bulk and $J = f_L[E_0(L)]$ describes the mechanism of electron injection from the cathode into a vacuum. Now, let us return to (30). Another case of an n-p-n junction occurs when there is $E(0) = E(d)$, namely

$$E(x) = \sqrt{\alpha_0 \left(x - \frac{d}{2} \right)^2 + K} \; ; \; K = E^2(0) - \frac{1}{4}\alpha_0 d^2 \ge 0 \qquad (40)$$

Here K is a constant of integration. Substituting (40) into the integral (8a), we obtain

$$V_1 = \frac{1}{2\sqrt{\alpha_0}} \left\{ \sqrt{\alpha_0}d \cdot E(0) + K \cdot \ln \left| \frac{E(0) + \frac{d}{2}\sqrt{\alpha_0}}{E(0) - \frac{d}{2}\sqrt{\alpha_0}} \right| \right\} \; ; \; K \ge 0 \; ; \; K < 0 \qquad (41)$$

We can show that a function (41) is defined for $K < 0$. Some mathematical properties of (41) are as follows

$$V_1 = \begin{cases} \dfrac{4E^3(0)}{3\alpha_0 d} \; ; \; E(0) << \dfrac{d}{2}\sqrt{\alpha_0} \\[4mm] E(0) \cdot d \; ; \; E0) >> \dfrac{d}{2}\sqrt{\alpha_0} \end{cases} \qquad (41a)$$

$$V_1 \to \frac{1}{2}V_{10} \quad as \quad E(0) \to \frac{d}{2}\sqrt{\alpha_0} \tag{41b}$$

$$\frac{dV_1}{dE(0)} > 0 \; ; \; E(0) \neq \frac{d}{2}\sqrt{\alpha_0} \tag{41c}$$

$$\frac{dV_1}{dE(0)} \to +\infty \quad as \quad E(0) \to \frac{d}{2}\sqrt{\alpha_0} \tag{41d}$$

Thus, substituting a function (41) and a function (16) into (31), we ascertain that a function $V = V(J)$ is defined. As an example, let us consider a case in which the cathode injects an infinite amount of electrons. For example, this is acceptable when there is $E_0(L) = 0$. With this boundary condition, a function (31) becomes

$$V = \beta J^{\frac{2}{3}} + \frac{1}{2\sqrt{\alpha_0}}\left\{ \sqrt{\alpha_0}d \cdot E(0) + K \cdot \ln\left|\frac{E(0) + \frac{d}{2}\sqrt{\alpha_0}}{E(0) - \frac{d}{2}\sqrt{\alpha_0}}\right| \right\} \; ; \; \beta = \frac{3}{4}\left(\frac{6m}{q\varepsilon_0^2}\right)^{\frac{1}{3}}(L-d)^{\frac{4}{3}} \tag{42}$$

$$and \quad J = f_0[E(0)]$$

And a boundary function f_0 describes the mechanism of hole injection from the anode into the bulk. A special case occurs when $d \to L$ and $\varepsilon E(d) = \varepsilon_0 E_0(d)$. Now, instead of (42), a function (31) takes the following limiting form

$$V = \frac{1}{2\sqrt{\alpha_0}}\left\{ \sqrt{\alpha_0}d \cdot E(0) + K \cdot \ln\left|\frac{E(0) + \frac{d}{2}\sqrt{\alpha_0}}{E(0) - \frac{d}{2}\sqrt{\alpha_0}}\right| \right\} \; ; \; J = f_0[E(0)] \tag{42a}$$

Here, there must be $f_0[E(0)] \equiv f_L[\varepsilon_r E(L)]$ and $\varepsilon_r = \varepsilon/\varepsilon_0$.

4. DISCUSSION and CONCLUSIONS

In the above, we have determined the divergence of the electric field distribution. For the bulk, when carrier generation processes are dominant and the condition (20) is valid, the electric field intensity satisfies a homogenous equation (26) or (28). Upon these conditions there exist two singular particular solutions $E_1(x)$ and $E_2(x)$ (which (29) expresses) as well as the general integral (30). First let us discuss the shape of a current density – voltage characteristic $J = J(V)$, which is determined by (34) and by (36). Here, let us assume that a boundary function $f_0[E(0)]$ is strongly increasing and $f_0(0) = 0$. Similarly, the same assumption is for a boundary function $f_L[E_0(L)]$. According to (33)-(36), we see that the $J(V)$ curve is displaced and strongly increasing. For (34), we ascertain that the $J(V)$ curve can exist only for $V > V_{10}$ and for $f_0^{-1}(J) \geq d \cdot \sqrt{\alpha_0}$. In other words, there exists a set of values of applied voltage V in which

the current density J is not defined. This property denotes that the whole system acts as a solar cell. In the case of (36), we see that the $J(V)$ curve is strongly increasing for $V > V_{10}$ and $J(V) \equiv 0$ for $0 \le V \le V_{10}$. Therefore, the system acts as a perfect blocking diode and a voltage stabiliser. Also, this mathematical property corresponds to a solar cell. The third case of singular solutions is expressed by (37)-(39). Here, in what follows, we take into consideration the same as above boundary functions $f_0[E(0)]$ and $f_L[E_0(L)]$. On the basis of (38), we ascertain that a function (39) can exist only in a set of $0 \le J \le f_0(2E_{10})$, where there is $E_{10} = \dfrac{d}{2} \sqrt{\alpha_0}$. For (34) and (36) as well as (39), some example of the shape of the $J(V)$ curve is illustrated in Fig. 6. Now, let us return to the particular integral (40). In this case, at the anode region $0 \le x \le \dfrac{d}{2}$, the space charge density $q_v = \varepsilon \dfrac{dE}{dx} < 0$ is negative. Analogously, for the bulk, at the $x = d$ contact region, that is $\dfrac{d}{2} \le x \le d$, the space charge density is positive
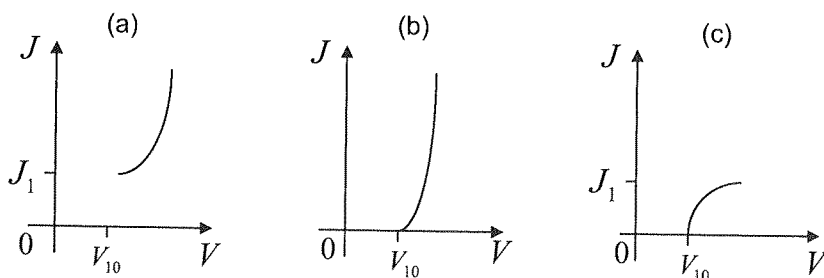

Fig. 6. The $J(V)$ curves shaped by the singular solutions: (a) the curve corresponds to $E_1(x)$ or to (34); (b) the curve corresponds to $E_2(x)$ or to (36); (c) the curve corresponds to (37) or to (39)
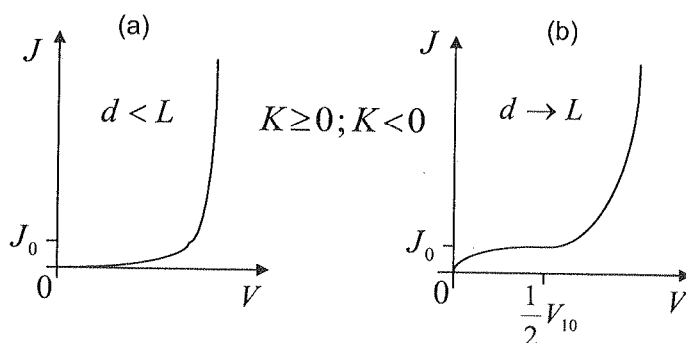

Fig. 7. The $J(V)$ curves correspond to (40) for $K \ge 0$ or $K < 0$: (a) the effect of a secondary electron emission is important; (b) a secondary electron emission is weak
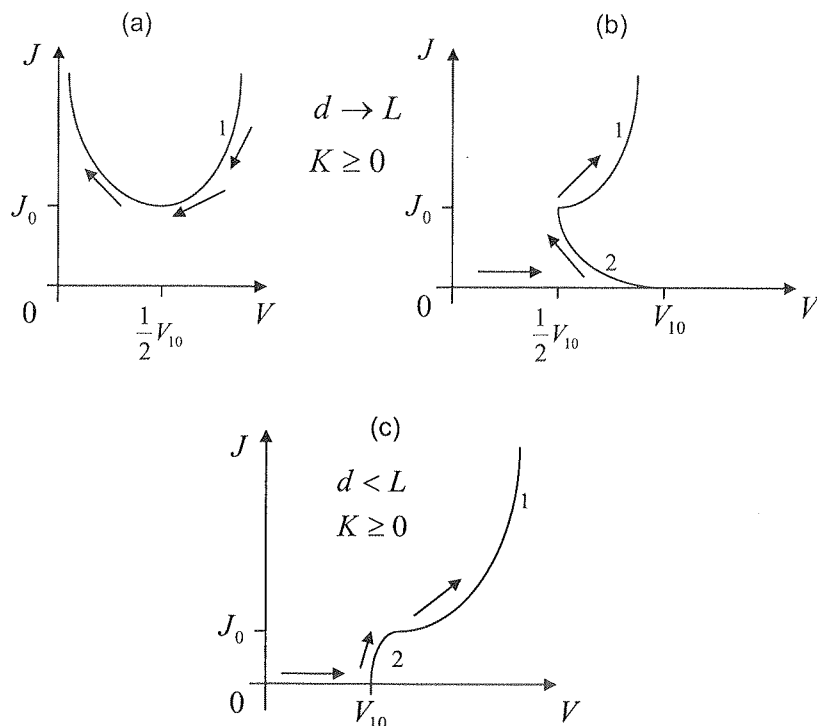
Fig. 8. The $J(V)$ curves correspond to (40) or to (37) for $K \geq 0$: (a) a boundary function f_0 is decreasing for $2V < V_{10}$ or f_0 is increasing for $2V \geq V_{10}$ as well as a secondary electron emission is weak; (b) f_0 is increasing and $f_0(0) = 0$ as well as a secondary electron emission is weak; (c) a secondary electron emission is important and f_0 is increasing and $f_0(0) = 0$. Here 1 – denotes (42)-(42a) and 2 – is (39)

$q_v = \varepsilon \dfrac{dE}{dx} > 0$. Thus, the whole system acts as an n-p-n junction and the $J(V)$ curve can be of the form (42). Moreover, we see that the bulk absorbs a portion of external solar energy in order to form the internal electric field (this physical property is expressed by $K < 0$). Some example of the shape of (42) is illustrated in Fig. 7a, when a boundary function $f_0[E(0)]$ is strongly increasing and $f_0(0) = 0$. If there is $\beta \to 0$ as $d \to L$, then (42) can be replaced by (42a) (an example is illustrated in Fig. 7b). A function from Fig. 7 is typical for the *SiC* structure [1, 12, 13]. In Fig. 6-8 there is $J_0 = f_0(E_{10})$ and $J_1 = f_0(2E_{10})$.

5. SUMMARY

In the above we have supposed that the divergence of the electric field can be described in terms of the total energy of an orbital electron.

The main idea of a space charge theory is to identify the internal and boundary processes that occur in the metal – solid – metal system. In general, this problem can be solved by the following electric field conditions [1-3, 5, 6, 13, 14]:

(a) the transient state of the discharging capacitor, characterised by

$$\int_0^d E(x,t)dx + \int_d^L E_0(x,t)dx = 0 \quad \text{for} \quad t \geq 0 \tag{43}$$

(aa) the transient state or the stationary state of the charging capacitor, described by

$$\int_0^d E(x,t)dx + \int_d^L E_0(x,t)dx = V(t) \neq 0 \quad \text{for} \quad t \geq 0 \tag{44}$$

where the voltage function is usually of the form $V(t) = const.$ or $V(t) = V_m \sin \omega t$, and the parameters V_m and ω are given; as well as

(aaa) the open system, in which the total current density J_t is

$$J_t(t) = \varepsilon \partial E(x,t)/\partial t + J_c(x,t) \equiv 0 \, or \, J_t(t) = \varepsilon_0 \partial E_0(x,t)/\partial t + J_{c0}(x,t) \equiv 0 \tag{45}$$

and

$$-\frac{\partial q_s}{\partial t} = J_{c0}(d,t) - J_c(d,t) \tag{45a}$$

where J_c and J_{c0} are respectively the convection current densities in a solid and in a vacuum, and q_s is the surface charge density on the plane $x = d$.

In this work, by making use of the current density – voltage characteristic $J(V)$, we have identified the interior and the boundaries together.

6. REFERENCES

1. M. A. L a m p e r t, P. M a r k: *Current injection in solids.* Academic Press, New York, 1970.
2. K. C. K a o: *New theory of electrical discharge and breakdown in low-mobility condensed insulators.* J. Appl. Phys., vol. 55, 1984, pp. 752-755.
3. K. C. K a o: *Double injection in solids with non-ohmic contacts: II. Solids with defects.* J. Phys. D.: Appl. Phys., vol. 17, 1984, pp. 1449-1467.
4. R. B. S c h i l l i n g, H. S c h a c h t e r: *Neglecting Diffusion in Space-Charge-Limited Currents.* J. Appl. Phys., vol. 38, 1967, pp. 841-844.
5. A. M a n y, G. R a k a v y: *Theory of transient space – chagre limited currents in solids in the presence of trapping.* Phys., Rev., vol. 126, 1962, pp. 1980-1988.
6. M. Z a h n, G. F. T s a n g, S. C. P a o: *Transient electric field and space charge behaviour for unipolar in conduction.* J. Appl. Phys., vol. 45, 1974, pp. 2432-2440.
7. A. G. M i l n e s: *Deep Impurities in Semiconductors.* John Wiley and Sons, New York – London – Sydney – Toronto, 1973.

8. R. P a t s c h: *Space-Charge Phenomena in Polyethylene at High Electric Fields.* J. Phys. D.: Appl. Phys., vol. 23, 1990, pp. 1497-1505.

9. J. S i m o n, J.-J. A n d r e: *Molecular Semiconductors. Photoelectrical Properties and Solar Cells.* Berlin, Springer, 1985.

10. O. H a c h e n b e r g, W. B r a u e r: *Secondary Electron Emission from Solids, Advances in Electronics and Electron Physics*, New York, Academic Press, vol. 11, 1959, pp. 413-499.

11. B. M a z u r e k, J. D. C r o s s: *Fast cathode processes in vacuum discharge development.* J. Appl. Phys., vol. 63, 1988, pp. 4899-4904.

12. P. A. B u d d, B. J a v i d i, J. W. R o b i n s o n: *Secondary Electron Emission from a Charged Dielectric.* IEEE Trans. Electr. Insul. vol. EI-20, 1985, pp. 485-491.

13. A. Z i e l: *Solid state physical electronis.* New Jersey, USA, 1976.

14. H. M a t s u u r a, H. O k u s h i: *Schottky barrier junctions of hydrogenated amorphous silicon-germanium alloys.* J. Appl. Phys., vol. 62, 1987, pp. 2871-2879.

15. B. Ś w i s t a c z: *Some further description for a current flow trough an amorphous solid. A case of an imperfect contact.* Electron. Tele. Quart., vol. 53, No. 2, Warsaw, 2007, pp. 143-154.

INFORMATION FOR AUTHORS OF E.T.Q.

An article published in other magazines can not be submitted for publishing in E.T.Q. TThe size of an article can not exceed 30 pages, 1800 character each, including figures and tables.

Basic requirements

The article should be submitted to the editorial stuff as a one side, clear, black and white computer printout in two copies. The article should be prepared in English. Floppy disc with an electronic version of the article should be enclosed. Preferred wordprocessors: WORD 6 or 8.
Layout of the article.
− Title.
− Author (first name and surname of author/authors).
− Workplace (institution, adress and e-mail).
− Concise summary in a language article is prepared in (with keywords).
− Main text with following layout:
  • Introduction
  • Theory (if applicable)
  • Numerical results (if applicable)
  • Paragraph 1
  • Paragraph 2
  • . . . .
  • . . . .
  • Conclusions
  • Acknowledgements (if applicable)
  • References
− Summary in additional language:
  • Author (firs name initials and surname)
  • Title (in Polish, if article was prepared in English)
  • Extensive summary, hawever not exceeching 3600 characters (along with keywords) in Polish, if artide was prepared in English). The summary should be prepared in a way allowing a reader to obtaoin essential information contained in the artide. For that reason in the summary author can place numbers of essential formulas, figures and tables from the article.
Pages should have continues numbering.

Main text

Main text cannot contain formatting such as spacing, underlining, words written in capital letters (except words that are commonly written in capital letters). Author can mark suggested formatting with pencil on the margin of the article using commonly accepted adjusting marks.
Text should be written with double line spacing with 35 mro left and right margin. Titles and subtitles should be written with small letters. Titles and subtitles should be numberd using no mor than 3 levels (i.e. 4.1.1.).

Tables

Tables with their titles should be places on separate page at the end of the article. Titles of rows and columns should be written in small letters with double line spacing. Annotations concerning tables should be placed directly below the table. Tables should be numbered with Arabic numbers on the top of each table. Table can contain algorithm and program listings. In such cases original layout of the table will be preserved. Table should be cited in the text.

Mathematical formulas

Characters, numbers, letters and spacing of the formula should be adequate to layout of main text. Indexes should be properly lowered or raised above the basic line and clearly written. Special characters such as lines, arrows, dots

should be placed exactly over symbols which they ar attributed to. Formulas should be numbered with Arabic numbers place in brackets on the right side of the page. Units of measure, letter and graphic symbols should be printed according to requirements of IEE (International Electronical Commission) and ISO (International Organisation of Standardisation).

References
References should be placed at the and of the main text with th subtitle „References". Refrences should be numbered (without brackets) adequately to references placed in the text. Examples of periodical [@], non-periodical [2] and book [3] references:
1. F. Valdoni: A new milimetre wave satelite. E.T.T. 1990, vol. 2, no 5, pp. 141–148
2. K. Anderson: A resource allocation framework. XVI International Symposium (Sweden). May 1991. paper A 2.4
3. Y.P. Tvidis: Operation and modeling of the MOS transistors. Nw York. McGraw-Hill. 1987. p. 553

Figures
Figures should be clearly drawn on plain or milimetre paper in the format not smaller than 9×12 cm. Figures can be also printed (preferred editor – CorelDRAW). Photos or diapositives will be accepted in black and whit format not gratd than 10×15 cm. On th margin of each drawing and on th back side of each ohoto authorp name and abbreviation of the title of article should be placed. Figure's captures should be given in two languages (first in the language the article is writtes in and then in additional language). Figure's captures should be also listed on separate page. Figures should be cited in the text.

Additional information
On the separate page following information should be placed:
– mailing address (home or office),
– phone (home or/and office),
– e-mail.
Author is entiled to free of charge 20 copies of article. Additional copies or the whole magazine can be ordered at publisher at th one's expense.
Author is obliged to perform the author's correction, which should be accomplished within 3 days starting from the date of receiving the text from the editorial stuff. Corrected text should be returned to the editorial stuff personally or by mail. Correction marks should be placed on th margin of copies received from the editorial stuff or if needed on separate pages. In the case when the correction is not returned within said time limit, correction will be performed by technical editorial stuff of the publisher.
In case of changing of workplace or home address Authors are asked to inform the editorial stuff.

FORMULARZ ZAMÓWIENIA

Imię, nazwisko/ nazwa Firmy...

Adres...

NIP...

Zamawiam prenumeratę następujących tytułów, których dystrybucję prowadzi

WDN PAN w Warszawie, ul. Śniadeckich 8, 00–656 Warszawa

Lp.	Tytuł	Cena 1 egz. w PLN	Cena za 4 numery w 2007 roku w PLN	Zamawiam
1	Electronics and Telecommunications Quarterly/Kwartalnik Elektroniki i Telekomunikacji	40,00	160,00	

	Jestem zainteresowany zakupem następujących numerów .... z lat poprzednich, po obniżonych cenach.

	Tak, jestem zainteresowany otrzymaniem wiadomości o kolejnych tytułach oraz o szczegółowej zawartości poszczególnych tytułów. Faxem, numer.. e-mailem, adres..

Do faktury doliczamy koszt wysyłki

Zamówienia przyjmujemy drogą e-mail, faxem lub pocztą na poniższy adres:
Warszawska Drukarnia Naukowa Polskiej Akademii Nauk
00–656 Warszawa ul. Śniadeckich 8,
tel/fax 022 628-76-14, 022 628-87-77
e-mail: wdnpan@home.pl

Płatność:
- na konto Bank Zachodni WBK S.A. 94 1090 1883 0000 0001 0588 2816
  za pobraniem   TAK / NIE (odpowiednie podkreśl)