Hardware Realization of the AES Algorithm S-Block Functions in the Current-Mode Gate Technology

Oleg Maslennikow, Magdalena Rajewska, Robert Berezowski

Abstract – **In this paper, the new approach to minimization of logic functions in the current-mode gate algebra is proposed. The main purpose is reduction of the current-mode circuit hardware overhead in such a way, that chip area needed for realization of the current-mode circuits will not greater than the chip area needed for realization of similar circuits with the classical CMOS. The approach is based on the analysis of the given truth table of the target function for searching the fragments, which correspond to selected types of sub-functions, which are hardly minimized in the Boolean algebra, but are simpler minimized in the current-mode gate algebra. The correctness and efficiency of the proposed approach are proved during design of the current-mode circuits destined for realization of several functions of the S-blocks in the AES cryptographic algorithm.**

Keywords – **Logic function, Minimization method, Current-mode gate, Switching noise, AES cryptographic algorithm**

I. INTRODUCTION

Modern VLSI application specific systems usually consist of digital and analog parts, where the first part is the specialized processor, while the second part is the preprocessing and interface unit between digital part and external world. Advances of the modern VLSI technology permit to implement such mixed systems on a single die (system-on-chip, SoC). However, the problem of influence of a digital part on an analog part of such SoC must be solved during system design. Switching transients (switching noise) of the digital part can perturb the analog part of a system owing to the coupling through the substrate [1, 2]. Radical reduction of this noise is based on the implementation of the mixed system digital part with the CMOS current-mode gates [3, 4]. Due to the nearly constant value of the power supply current at the different gate states, the level of their noise is essentially lower in comparison with the classical voltage type gates. However physical and logical properties of the current-mode gates differ from corresponding properties of classical voltage-mode gates, because generally current-mode circuits operate in multiple-valued logic (MVL) [5, 6, 7]. Therefore, two special approaches to design of the current-mode circuits were proposed by the authors and were published in the several previous papers [8, 9]. Using these approaches, the current-mode prototypes of standard binary circuits (adders, decoders, multiplexers, flip-flops, registers, counters, etc.) were designed, which need less number of gates and interconnections in comparison with corresponding circuits, constructed with classical voltage CMOS gates.

However, the number of transistors needed for realization of the binary circuits with the current-mode gates is usually 2 – 2.5 times more than the number of transistors in their realization with classical voltage CMOS gates. Therefore, in this paper, more advanced approach to designing of current-mode circuits is proposed. New approach is based on the analysis of the given truth table of the target function, for searching here the fragments, which correspond to selected types of sub-functions. These sub-functions are hardly minimized in the Boolean algebra, but are very simple minimized in the current-mode gate algebra. The example of such sub-functions is the XOR function, which can have an arbitrary number of arguments. Then finding fragments are eliminated from the input function truth table and are immediately transformed into corresponding logic expressions (represented in the current-mode algebra). The resulting function expression is derived as the algebraic sum of the obtained expressions, because algebraic sum is naturally realized in the current-mode algebra.

The proposed approach has been used for design of the current-mode circuits destined for realization of several functions of the S-block in the AES cryptographic algorithm. Comparison of the hardware complexity designed current-mode circuits with their classical CMOS prototypes showed, that average number of transistors needed for realization of the both circuit versions is nearly equal. Besides, thanks to the low level of substrate noise and power supply noise, eavesdropping of current-mode gate switching is prevented, which is very important for cryptographic systems.

II. CURRENT-MODE GATES AND LOGIC OVERVIEW

The conception of the binary current-mode inverter gate with static noise margins and its possible realization are presented in fig. 1a and fig. 1b respectively. Detailed electrical parameters of this gate were presented in the previous papers [3, 4]. There are four types of the current-mode gates: inverter, anti-inverter, double-inverter and anti-double-inverter, which perform four basic current-mode operations – respectively (1), (2), (3) and (4). Moreover, all current-mode gates have only one input, while an arbitrary gate may contain several outputs, possibly, of the different types. For example, the graphical representation of the current-mode gate with the inverter Y_1, anti-inverter Y_2, double-inverter Y_3 and anti-double-inverter Y_4 outputs is shown in the fig. 2a.

The main operations in the current-mode logic (algebra) are the above-mentioned inversions and the arithmetic addition/subtraction operation. The addition operation corresponds, at the physical level, to the addition of currents, each of which represents the value of the corresponding operand. On the functional level the addition is realized by means an association of all operand lines into the one node.

Oleg Maslennikow, Magdalena Rajewska, Robert Berezowski - Department of Electronics, Technical University of Koszalin, ul. Sniadeckich 2, 75-453 Koszalin, POLAND
E-mail: oleg@ie.tu.koszalin.pl

Similarly, an arithmetic subtraction operation, in this technology, corresponds to association the line of the first operand with the output of the anti-inverter gate connected to the line of the second operand. Examples of realization of operations *(X + Y)* and *(X - Y)* are shown in fig. 2b.

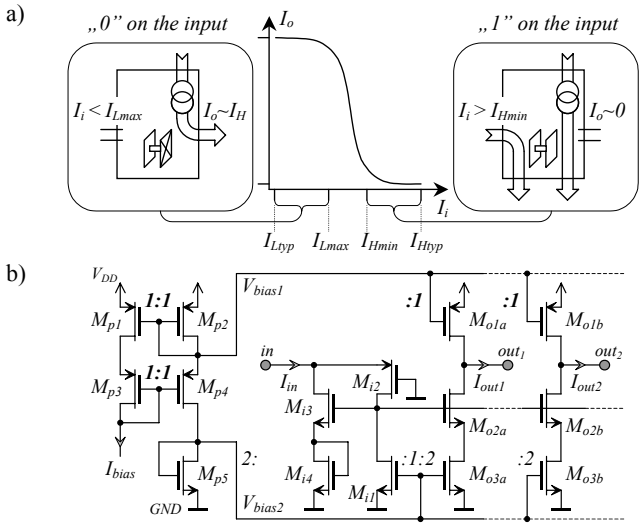

Fig. 1. Conception (a) and the example of possible realization of the current- mode inverter gate, showing the way of output signal duplication (b)

$$Y_1 = \overline{X} = \begin{cases} 1 & \text{if } X = 0, -1, -2, \ldots \\ 0 & \text{if } X = 1, 2, 3, 4, \ldots \end{cases} \quad (1)$$

$$Y_2 = \hat{X} = \begin{cases} 0 & \text{if } X = 0, -1, -2, \ldots \\ -1 & \text{if } X = 1, 2, 3, 4, \ldots \end{cases} \quad (2)$$

$$Y_3 = \overline{\overline{X}} = \begin{cases} 0 & \text{if } X = 0, -1, -2, \ldots \\ 1 & \text{if } X = 1, 2, 3, 4, \ldots \end{cases} \quad (3)$$

$$Y_4 = \hat{\overline{X}} = \begin{cases} -1 & \text{if } X = 0, -1, -2, \ldots \\ 0 & \text{if } X = 1, 2, 3, 4, \ldots \end{cases} \quad (4)$$

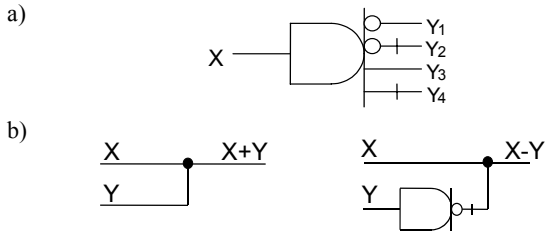

Fig. 2. Current-mode gate with four different outputs (a); realization of addition and subtraction operations in this technique (b)

It follows from the expressions (1) – (4), that arbitrary logical variable in this logic is a multi-valued one (in a general case), because the value appeared on any gate output belongs to the set {-1, 0, 1}, while the value of the variable appeared on any gate input (for example, as a result of an addition or subtraction operations) belongs to the set of integer numbers from the interval]-∞,∞[. Due to such logical properties, the Boolean algebra identities are not suitable for the current-mode algebra; however an arbitrary binary function can be realized with current-mode gates [4,5].

Really, an arbitrary Boolean expression can be transformed into the corresponding current-mode gate algebra expression and realized with current-mode gates using the following identities (5) for conversion of the main Boolean operations:

$$\begin{aligned} a \cdot b &= \overline{\overline{a} + \overline{b}}, \\ \overline{a \cdot b} &= \overline{\overline{\overline{a} + \overline{b}}}, \\ a \vee b &= \overline{\overline{a + b}}, \\ \overline{a \vee b} &= \overline{a + b}, \end{aligned} \quad (5)$$

where symbols „•", „∨" and „+" correspond to operations AND, OR and arithmetic addition respectively, and values of the variables (or functions) a and b belong to the set {0,1}. Furthermore, the current-mode algebra also has its own logical identities. Several from these identities are presented in ref. [8, 9].

III. Approaches to minimization of logic functions in the current-mode gate algebra

In a case of minimization of binary functions, the first approach, which is based on the expressions (5), can be used. It consists of two stages. In the first stage the minimization of the target function in the Boolean algebra is performed. Note, that an arbitrary from the known minimization methods can be used in this stage (for example, Veitch-Karnaugh diagrams or Quine-McCluskey method [11]). Then using the identities (5) the transformation of obtained Boolean expression in the corresponding current-mode algebra expression is carried out. This approach is very simple and suitable for immediate realization in the computer-aided design (CAD) systems [10], but its disadvantage is that there are no guaranties, that obtained current-mode algebra expression is optimal, even if corresponding criteria (which have been proposed by the authors for improving of resulting expression [9]), have been used during first minimization stage. Usually, the number of gates and interconnections in the current-mode binary circuits are 20% - 30% less, but the number of transistors needed for realization of these circuits is 2 – 2.5 times more than the number of transistors in their realization with classical voltage CMOS gates.

The second approach to minimization of logic functions in the current-mode algebra uses the characteristic features of the current-mode technology, and is based on the confirmation, that an arbitrary logical function can be represented in the current-mode gate algebra as an algebraic sum of the set of several simpler logical functions. This confirmation immediately follows from the fact, that the operations of arithmetic addition and subtraction are natural property of the current-mode gate algebra. Therefore, the second approach consists of minimization of the target logic function by means its representation as an algebraic sum of the set of more simple functions (named „radix" functions), which are selected (in general case) in a heuristic way. This approach can be used for minimization of binary functions as well as multiple-valued logic functions, and usually allows obtain the

better results in comparison with the first approach. For example, for the function $Y=f(a_1, a_2, a_3)$, given by the Veitch-Karnaugh diagram represented in fig. 3a, two blocks B_1 and B_2 (here $Y=1$) and one block B_3 (here $Y=-1$) can be selected using first proposed approach. First and second blocks are represented by the Boolean expressions $a_1 \cdot a_2$ and $\overline{a}_1 \cdot a_2 \cdot a_3$ respectively, while third block – by the expression $\overline{a_1 + a_2 + a_3}$. In the current-mode algebra the resulting expression for the function Y is equal to the algebraic sum of the blocks B_1, B_2 and ($-B_3$) after their conversion using identities (5). This expression is following:

$$Y = B_1 + B_2 - B_3 = \overline{\overline{a}_1 + \overline{a}_2} + \overline{a_1 + \overline{a}_2 + \overline{a}_3} - \overline{a_1 + a_2 + a_3}$$

and finally

$$Y = \overline{\overline{a}_1 + \overline{a}_2} + \overline{a_1 + \overline{a}_2 + \overline{a}_3} + \overline{\hat{a}_1 + a_2 + a_3} \qquad (6)$$

The current-mode circuit destined for realization of the expression (6) is represented in fig. 3b.

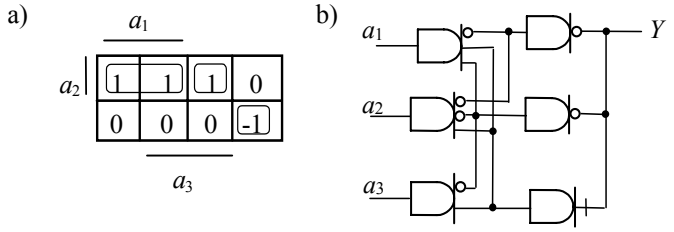

Fig. 3. Multiple-valued function $Y=f(a_1, a_2, a_3)$, given by the Veitch-Karnaugh diagram (a) and current-mode circuit for its realizaton (b)

When the second approach is used, the two another blocks B_1 (here $Y=1$) and B_2 (here $Y=-1$) can be selected in the Veitch-Karnaugh diagram of the function Y. These blocks (see fig. 4a) are represented by the Boolean expressions a_2 and $\overline{a}_1 \cdot \overline{a}_3$ respectively. The resulting expression for the function Y also is equal to the algebraic sum of the blocks B_1 and ($-B_2$) after their conversion by means identities (5). This expression is following:

$$Y = a_2 - \overline{a_1 + a_3} = a_2 + \overline{\hat{a}_1 + a_3}. \qquad (7)$$

The current-mode circuit destined for realization of the function Y in accordance with the expression (7) is represented in fig. 4b.

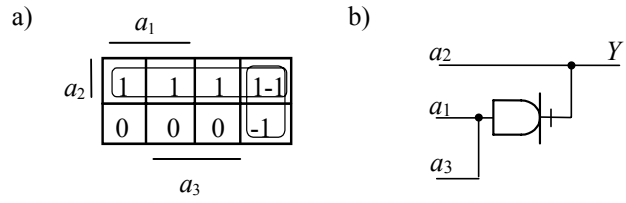

Fig. 4. Illustration of the second approach to minimization of the MVL function Y (a) and current-mode circuit for its realization (b)

Comparison of hardware complexity of the both designed circuits proves the high effectiveness of the second approach to minimization of logic functions in the current-mode algebra. Unfortunally, this approach has the important disadvantage. It is non systematic and therefore it doesn't suitable for computer realization.

Therefore, in this paper, the new approach to minimization of binary function in the current-mode algebra is proposed, which is based on the both above described approaches and combines of their advantages. The main purpose is the further reduction of the current-mode circuit hardware overhead in such a way, that chip area needed for realization of the current-mode circuits is not greater than the chip area needed for realization of similar circuits with the classical CMOS. This approach has been designed after selection of several often used binary functions, which are hardly minimized in the Boolean algebra, and are simpler minimized in the current-mode gate algebra. Such functions are for example, the *XOR* function (*XOR3*) and the carry out (*C3*) function. Note that when both these functions have three arguments, they determine the arithmetic addition operation, which is performed by each arithmetic-logic unit. Representations of these functions by means Veitch-Karnaugh diagrams represent fig. 5a and fig. 5b respectively. The minimized expressions of these functions in the Boolean and the current-mode algebra represent the expressions (8) and (9) respectively. The possible realizations of the expressions (9) are represented in the fig. 6a i fig. 6b respectively.

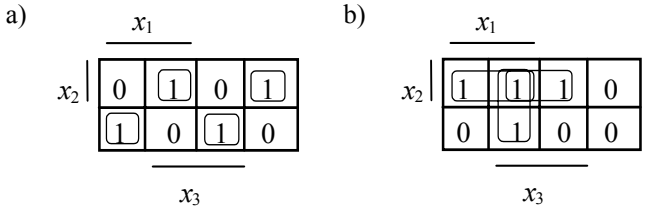

Fig. 5. Examples of the radix functions *XOR3* (a) and *C3* (b) given by the Veitch-Karnaugh diagrams

$$xor3 = x_1 \oplus x_2 \oplus x_3, \quad C3 = x_2 \cdot x_3 \vee x_1 \cdot x_3 \vee x_1 \cdot x_2 \qquad (8)$$

$$xor3 = \overline{x_1 + x_2 + x3 - 2(x_1 + x_2 + \hat{\bar{x}}_3)} \qquad (9)$$
$$C3 = \overline{\bar{x}_1 + \bar{x}_2 + \hat{x}_3}$$

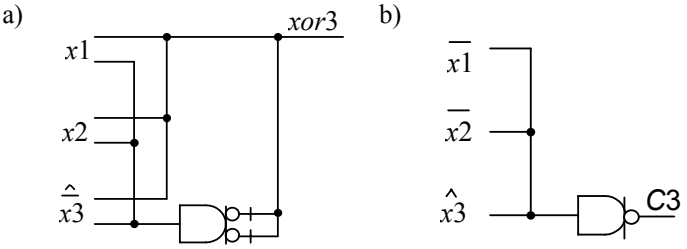

Fig. 6. Current-mode realizations of the radix functions *XOR3* (a) and *C3* (b)

Based on these circuits, the one-bit current-mode adder can be constructed, which is represented in the fig. 7. This adder consists of minimum 1 and maximum 4 gates (in a case when all input variables should be duplicated by means three extra gates), therefore its hardware overhead hesitates from 14 to 52 transistors. Because the number of interconnections in the current-mode adder is nearly 2 times less in comparison with the similar classical CMOS adder, the chip area needed for

realization of this adder is not greater than the chip area needed for realization of the classical CMOS adder.

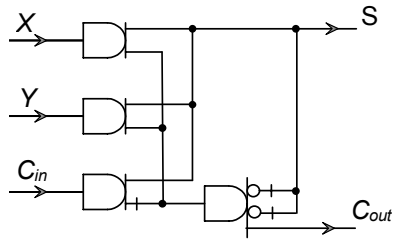

Fig. 7. Current-mode versions of the one-bit adder

The main idea of the proposed method is based on using these functions as the radix functions and is following. In the truth table of target function the 8 row fragments are selected, which correspond to the *XOR3* or *CO* functions. Note that these fragments forms the corresponding blocks in the Veitch-Karnaugh diagram of the target function, each of them consists of 8 cells. Then these fragments are eliminated from the truth table (or these 8-cell blocks are eliminated from the Veitch-Karnaugh diagram of the target function), and the corresponding expressions (9) are added to the resulting expression representing target function after minimization. Remaining fragments of the target function truth table or fragments of the Veitch-Karnaugh diagram are minimized in accordance to the above-described first approach to minimization logic function in the current-mode gate algebra. The correctness and efficiency of the new minimization approach is proved in the next section on the example of minimization several functions, which are performed in the AES cryptographic algorithm S-blocks. Note, that in this example, extra radix functions $R1$ and $R2$ have been used during minimization of the target S-block functions. The Veitch-Karnaugh diagrams and corresponding representations of these functions are shown in fig. 8.

a)

b \ a				
	0	1	1	0
	1	1	1	0

c

$$R1 = \overline{\overline{a - b} - c}$$

b)

b \ a				
	1	1	1	1
	0	1	1	1

c

$$R2 = \overline{a - b - c}$$

Fig. 8. Examples of the radix functions $R1$ (a) and $R2$ (b) given by the Veitch-Karnaugh diagrams and corresponding expressions

IV. Minimization and Hardware Realization of the AES Algorithm S-Block Functions

AES (Advanced Encryption Standard) is the modern and widely used cryptographic algorithm standardized by NIST (National Institute of Standards and Technology) in 2002 [12]. It usually uses 128-bit symmetrical key and operates with 128-bit input data blocks. The main functional part of this algorithm is S-block, which can be represented as the block of eight binary functions Z0,…,Z7 of eight different arguments $x0,\ldots, x3$ and $y0,\ldots, y3$. The example of one of such blocks represents the table 1, in which each cell contains the values of all functions Z7,…,Z0, i.e. 8 bits represented by two hexagonal digits. In order to hardware realization of these function by the corresponding combinatorial circuit, the Quine-McCluskey method [11] has been used for function minimization. After minimization of the function Z0, $Z0 = f(x3, x2, x1\ x0, y3, y2, y1, y0)$ the following expression (10) has been obtained:

$$\begin{aligned}
Z0 = {} & \overline{x3}\,\overline{x2}x1\overline{x0}y3y2y1y0 + \overline{x3}x1x0y3y2y1y0 \\
& + \overline{x3}\,\overline{x2}x1\overline{x0}y3y2y1 + \overline{x3}x2x1x0y3y1y0 \\
& + x3\overline{x2}x1\overline{x0}y3\overline{y1}y0 + \overline{x3}x2x1y3y2\overline{y1}y0 \\
& + x3\overline{x2}x0y3y2\overline{y1}y0 + \overline{x3}x2x0y3y2y1y0 \\
& + \overline{x3}x\overline{x0}y3y2\overline{y1}y0 + \overline{x3}x2\overline{x1}y3y2y1 \\
& + \overline{x3}\,\overline{x2}x1\overline{x0}y3\overline{y2}y0 + \overline{x3}x2x1x0y3\overline{y2}y1 \\
& + \overline{x3}x2x1\overline{x0}y3y2\overline{y1}y0 + x3\overline{x2}x1x0y3\overline{y2}y1 \\
& + \overline{x3}x2x1\overline{x0}y3y2y1 + x3x2x1x0y3y2y1 \\
& + \overline{x2}x1\overline{x0}y3\overline{y2}y0 + \overline{x3}x2x1x0y2y1y0 \\
& + \overline{x2}x1x0y3\overline{y2}y1y0 + \overline{x3}x2x1\overline{x0}y2\overline{y1}y0 \\
& + \overline{x2}x0y3\overline{y2}y1y0 + \overline{x3}x2x0y3\overline{y2}y1 \\
& + \overline{x3}x2x0\overline{y3}\,\overline{y2}y1 + \overline{x3}x1y3\overline{y2}y1y0 \\
& + \overline{x3}x2\overline{x1}\,\overline{y2}\,\overline{y1}y0 + \overline{x3}x2x0y3\overline{y2}y1 \\
& + x3\overline{x1}\,\overline{y3}\,\overline{y2}y1\overline{y0} + \overline{x3}x0y3y2y1\overline{y0} \\
& + \overline{x2}x1y3y2\overline{y1}\,\overline{y0} + x2\overline{x1}x0y3y2\overline{y1} \\
& + \overline{x3}x2x1\overline{x0}y3y2 + x3x0y3\overline{y2}y1\overline{y0} \\
& + x3x1\overline{y3}y2\overline{y1}\,\overline{y0} + \overline{x3}x2\overline{x1}x0y1y0 \\
& + \overline{x3}x2x0\overline{y3}y1y0 + x2x1x0y3\overline{y2}y0 \\
& + x3\overline{x2}\,\overline{x1}y3y2y1 + x3\overline{x1}y3y2y1\overline{y0} \\
& + x3x2x1y3\overline{y2}\,\overline{y0} + \overline{x3}\overline{x2}y3y1\overline{y0} \\
& + \overline{x1}\,\overline{x0}y3\overline{y2}y0 + x2x1\overline{y2}y1y0 \\
& + x3x2x0y3\overline{y2}
\end{aligned}$$

This expression determines the main parameters of the combinatorial circuit, which is destined for realization of the function $Z0$ and is based on the classical CMOS NAND and NOR gates: type of gate and number of gate inputs and number of transistors needed for construction of each type of

gate. Approximated values of these parameters are represented in the table 2.

Table 1. Truth table of the output functions *Z*7,*Z*6,…,*Z*0 of the S-block in the AES algorithm (input and output data are given in the hexagonal format)

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Table 2. The main parameters of the combinatorial circuits destined for realization of the function *Z*0

Type of gate	Number of gates in the circuit	Number of transistors per one gate	Number of inputs	Whole transistor account
NAND 8	1	18	8	18
NAND 7	22	16	154	352
NAND 6	22	14	132	308
NAND 5	4	12	20	48
NOR 49	1	100	49	100
Total	**51**		**237**	**826**

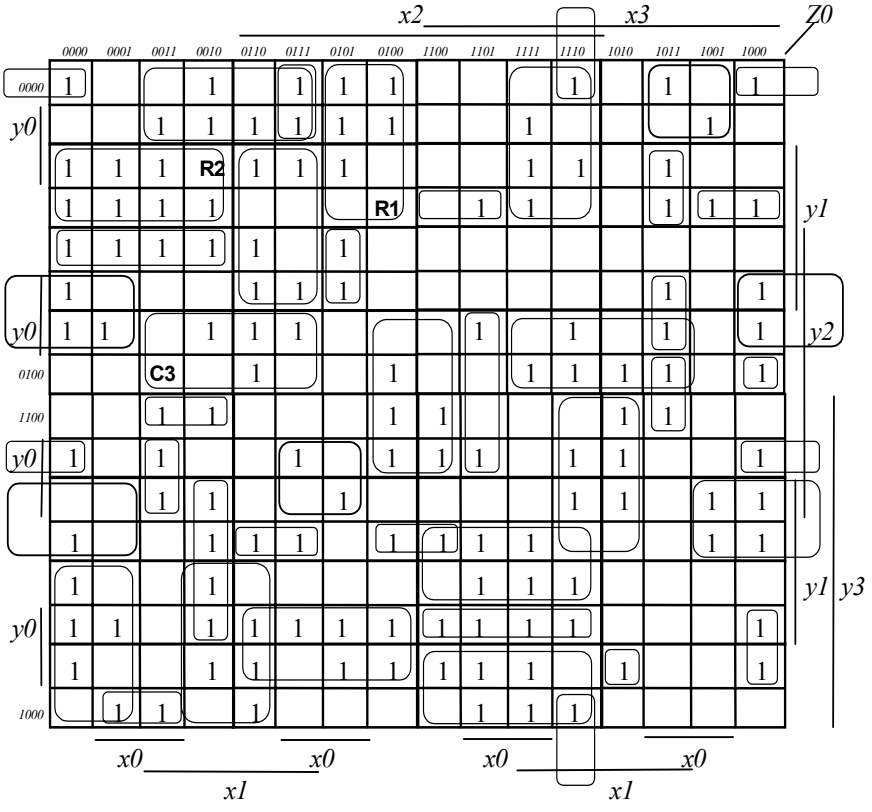

Fig. 9. Truth table of the function Z0 = *f*(*x*3, *x*2, *x*1 *x*0, *y*3, *y*2, *y*1, *y*0) – one of the functions of the S-block

In a order to minimization of the function *Z*0 in the current-mode gate algebra, the new minimization approach has been used. Note that the target function *Z*0 has been represented initially by the Veitch-Karnaugh diagram. This diagram with marked 8-cell blocks, which are described the radix functions *XOR*2, *C*3, *R*1 and *R*2, is shown in fig. 9. The total number of selected blocks is equal 40, and fig. 10 represents the examples of the hardware realization of the several from them, in particularly, the *C*3, *R*1 and *R*2 blocks.

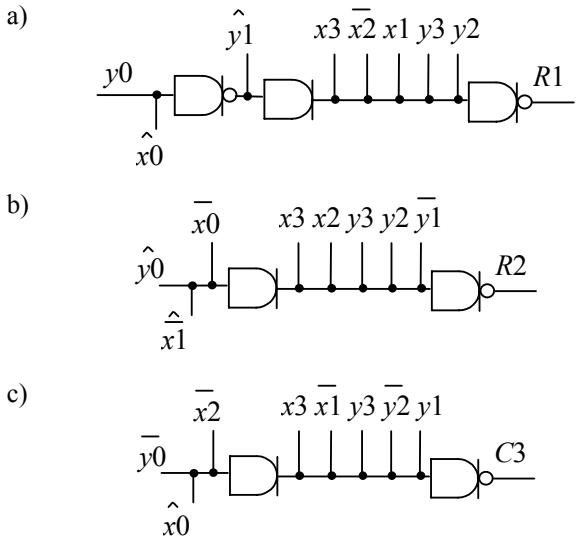

Fig. 10. Current-mode realizations of the 8-cell blocks in the Veitch-Karnaugh diagram of the *Z*0 function, which correspond to the functions *R*1 (a), *R*2 (b) and *C*3 (c).

Based on these blocks, the following resulting expression described the function *Z*0 in the current-mode gate algebra are obtained:

$$
\begin{aligned}
Z0 = {} & \overline{x3+\overline{x2}+\overline{x1}+y3+\overline{y1}+\overline{y2-x0}-y0} \; + \\
& \overline{x2+x1+y3+\overline{y2}+\overline{y0}+\overline{\overline{x3}-y1-x0}} \; + \overline{x3+\overline{x2}+\overline{y3}+y2+\overline{y0}+\overline{x0-\overline{y1}-x1}} \\
& + \overline{x2+x1+x0+y3+y2+y1+y0} \; + \overline{\overline{x3}+\overline{x2}+\overline{x1}+x0+y2+y1+y0} + \\
& + \overline{\overline{x3}+\overline{x2}+x1+y3+y2+y1+y0} + \overline{\overline{x3}+x2+\overline{x1}+\overline{x0}+y3+y2+\overline{y1}} + \\
& \overline{\overline{x3}+x2+x1+y3+y2+\overline{y1}+y0} + \overline{\overline{x3}+x2+x1+x0+y3+\overline{y2}+y1+y0} + \\
& + \overline{x3+\overline{x2}+x1+\overline{x0}+y3+\overline{y2}+\overline{y1}} + \overline{\overline{x3}+\overline{x2}+x1+\overline{x0}+\overline{y2}+y1+\overline{y0}} + \\
& \overline{x3+x2+\overline{x1}+\overline{x0}+\overline{y3}+\overline{y2}+\overline{y0}} + \overline{x3+x2+\overline{x1}+\overline{y3}+\overline{y2}+y1+y0} + \\
& + \overline{x2+x1+x0+\overline{y3}+\overline{y2}+y1+\overline{y0}} + \overline{x3+\overline{x2}+\overline{x1}+\overline{y3}+\overline{y2}+\overline{y1}+y0} + \\
& + \overline{x3+x2+\overline{x1}+x0+\overline{y3}+\overline{y1}} + \overline{x3+x2+\overline{x0}+\overline{y3}+y2+y1+y0} + \\
& + \overline{\overline{x3}+x2+\overline{x1}+x0+\overline{y3}+y2+y1+\overline{y0}} + \overline{x3+x2+y3+\overline{y2}+\overline{y1}+y0} + \\
& + \overline{\overline{x3}+x2+\overline{x1}+\overline{x0}+y3+\overline{y2}+\overline{y0}} + \overline{x3+x2+\overline{x1}+\overline{x0}+\overline{y2}+y1+y0} + \\
& + \overline{\overline{x3}+\overline{x2}+\overline{y3}+y2+\overline{y1}+\overline{y0}} + \overline{\overline{x2}+x1+x0+\overline{y3}+\overline{y2}+\overline{y1}+y0} + \\
& + \overline{\overline{x3}+x2+x1+x0+\overline{y3}+y2+\overline{y0}} + \overline{x3+\overline{x2}+\overline{x1}+\overline{x0}+y3+y2+y1} + \\
& + \overline{x3+\overline{x1}+y3+\overline{y2}+y1+\overline{x2}+\overline{y0-x0}} \; + \\
& + \overline{x3+\overline{x1}+y3+y2+y1+x2+\overline{y0}-\overline{x0}} \; +
\end{aligned}
$$

$$
\begin{aligned}
& + \overline{x3+x2+x1+\overline{y3}+y2+\overline{y1}+x0-y0} \; + \\
& + \overline{\overline{x3}+\overline{x1}+y3+\overline{y2}+y1+\overline{x2}+y0-\overline{x0}} \; + \\
& + \overline{\overline{x3}+\overline{x2}+\overline{x1}+y3+y2+\overline{y1}+\overline{x0}-y0} \; + \\
& + \overline{x3+\overline{x1}+x0+\overline{y3}+y2+y1+\overline{x2}-y0} \; + \\
& + \overline{x3+\overline{x2}+x1+y3+y2+\overline{y0-\overline{x0}-\overline{y1}}} \; + \\
& + \overline{\overline{x2}+x1+x0+\overline{y2}+y1+\overline{y0}-x3-y3} + \\
& + \overline{\overline{x3}+\overline{x2}+\overline{y3}+\overline{y1}+y0+\overline{x1-y2}-x0} + \\
& + \overline{\overline{x3}+\overline{x2}+\overline{y3}+y2+y1+\overline{x1}-\overline{y0}-x0} \; + \\
& + \overline{\overline{x3}+\overline{x1}+x0+\overline{y3}+\overline{y2}+\overline{y1-x2}-y0} \, .
\end{aligned}
$$

This expression determines the current-mode combinatorial circuit for realization of the function *Z*0. The resulting parameters of the designed circuit are represented in the table 3, where "Max" notes the case when all input variables isn't received from the previous circuits, and should be duplicated by means extra gates in this circuit.

Comparison of hardware complexity of the current-mode and classical CMOS circuits destined for realization of the remained S-block functions Z1,…,Z7 showed, that the average numbers of gates, interconnections and transistors in the current-mode circuits (which are obtained based on the proposed approach to minimization of logic functions) are approximately equal to the numbers of gates, interconnections and transistors needed for realization of these circuits with classical voltage CMOS gates respectively.

REFERENCES

[1] Makie-Fukuda K., Kikuchi T., Matsuura T., M.Hotta. "Measurement of digital Noise in Mixed-Signal integrated circuits", *IEEE J. Of Solid-State Circuits*, No.2*,* 1995, s. 87–92.

[2] M. Ingels, M.S.J. Steyaert, "Design Strategies and Decoupling Techniques for Reducing the Effects of Electrical Interference in Mixed-Mode IC's", *IEEE J. of Solid-State Circuits*, V. 32. No. 7, 1997, pp. 1136-1141.

[3] P. Pawlowski, A. Guzinski, "Low-voltage current-mode gates for MAD systems", *Proc. of the European Conference on Circuit Theory and Design, ECCTD*,1999, Stresa, Italy, 1999, pp. 507-510

[4] Maslennikow O., Pawłowski P., Sołtan P., Berezowski R. Current-Mode Digital Gates and Circuits:Conception, Design and Verification. *Proc. of the IEEE Int. Conf. on Electronic Circuits and Systems, ICECS'2002, Horwacja, Vol.2,* pp. 623-626.

[5] Jain A.K., Bolton R.J., Abd-El-Barr M.H.: CMOS Multiple-Valued Logic Design – Part I: Circuit Implementation, *IEEE Trans.on Circuits & Systems-I*, V. 40, No.8, 1993, s. 503–514.

[6] Jain A.K., Bolton R.J., Abd-El-Barr M.H.: CMOS Multiple-Valued Logic Design – Part II: Function Realization, *IEEE Trans.on Circuits & Systems-I*, V.40, No.8, 1993, s. 515–522.

[7] Maslennikow O., Gretkowski D., Maslennikowa N., Pawłowski P. Current Mode Multipliers and Constant Coefficient Multipliers for Radix *N* and Modulo *N*

Arithmetic. *Proc. of the 11-th Int. Conf. on Mixed Design of Integrated Circuits and Systems, MIXDES'2004,* Szczecin, pp. 162-167.

Table 3. The main parameters of the combinatorial circuits destined for realization of the function *Z*0

Type of gate	Min/Max number of gates in the circuit	Number of transistors per one gate	Min/Max number of outputs	Min/Max whole transistor account
1 output inverter gate	47/47	7	47/47	329/329
2 output anti-inverter gate	2/2	8	4/4	32/32
37 output inverter or double inverter gate	0/8	0/115	0/296	0/920
Total	**49/57**		**51/347**	**361/1281**
Average	**53**		**199**	**821**

[8] O. Maslennikow, "Approaches to Designing and Examples of Digital Circuits Based on the Current-Mode Gates", *Data Recording, Storage & Processing*, Vol. 3, No. 2, 2001, pp. 84-98

[9] Maslennikow O. Podstawy teorii zautomatyzowanego projektowania reprogramowalnych równoległych jednostek przetwarzających dla jednoukładowych systemów czasu rzeczywistego. (Monografia habilitacyjna). *Wydawnictwo Uczelniane Politechniki Koszalińskiej*, Koszalin, 2004, 273 s.

[10] Maslennikow O., Gretkowski D., Sołtan P., Pawłowski P. Computer-aided design, verification visualization of digital circuits based on current-mode gates. *Proc. 8-th Int. Conf. Experience of Designing and Application of CAD Systems in Microelecronics, CADSM'2005,* Lwów, 2005, pp. 193-197.

[11] DeMicheli G. Synthesis and Optimization of Digital Circuits. *McGraw-Hill*, 1994.

[12] Federal Information Processing Standards Publication 197, November 26, 2001, "Announcing the ADVANCED ENCRYPTION STANDARD AES"

V. Conclusion

In this paper, the advanced approach to design of current-mode circuits has been proposed, which is based on the analysis of the given truth table of the target function, for searching here the fragments, which correspond to selected types of sub-functions. This approach has been designed after selection of several often used binary functions, which are hardly minimized in the Boolean algebra, and are simpler minimized in the current-mode gate algebra. Such functions are for example, the *XOR* and the carry out functions, which can have two or more arguments, and determine the arithmetic addition operation. Then finding fragments are eliminated from the input function truth table and are immediately transformed into corresponding logic expressions (represented in the current-mode algebra). The resulting function expression is derived as the algebraic sum of the obtained expressions, because algebraic sum is naturally realized in the current-mode algebra. The main purpose is reduction of the current-mode circuit hardware overhead in such a way, that chip area needed for realization of the current-mode circuits will not greater than the chip area needed for realization of similar circuits with the classical CMOS.

The proposed approach has been used for design of the current-mode circuits destined for realization of several functions of the S-block in the AES cryptographic algorithm. Comparison of the hardware complexity designed current-mode circuits with their classical CMOS prototypes showed, that average number of transistors needed for realization of the current-mode circuits is nearly equal to the number transistors in their CMOS prototypes.