Kalganova T., N. Lipnitskaya, G. Holowinski (1997) Modular Decomposition of the NOR-TSUM multiple-valued PLA. Proc. of the 19th International Scientific Symposium of Students and Young Research Workers. Zielona Gora, Poland. pp.138-144.

MODULAR DECOMPOSITION OF THE NOR-TSUM MULTIPLE-VALUED PLA

T. KALGANOVA†, N. LIPNITSKAYA†, G. HOLOWINSKI‡

† Belarusian State University of Informatics and Radioelectronics, Laboratory of Image Processing and Pattern Recognition. Brovky St., 6, 220600, Minsk, Republic of Belarus, Phone: (*0375172*) 49-19-81, Fax: (*0375172*) 495-106, E-mail: jack@expert.belpak.minsk.by or pottosina@risq.belcaf.minsk.by

‡ Institute of Computer Science & Information Systems, Technical University, ul. Zolnierska 49, Szczecin, Poland, Fax. (+4891) 48-76439, E-mail: holowins@dedal.man.szczecin.pl

ABSTRACT. A method for designing PLA-based combinational circuits by modular decomposition is presented. Main subjects are 1) Specific properties of TSUM operator, 2) MIN-TSUM and NOR-TSUM expansions with respect to the bound set, X_1 of variables, 3) Realization of functions by multiple-valued PLA-based combinational circuits, 4) Comparison with other methods. Experimental investigations show that the size of suggested combinational circuit is the same as the size of multiple-valued PLA implementing a multiple-valued logic function with large number of variables.

KEY WORDS. Multiple-valued logic, modular decompositions, logic design, multiple-valued programmable logic array (MV PLA)

1. Introduction

Multiple-valued logic is one of the active area of research. It is known that multiple-valued logic allows interconnections to be reduced into a chip, as well as a reduction in the numbers of inputs and outputs. Different MV PLA structures to implement the multiple-valued logic (MVL) functions have been proposed [1, 2, 3, 4]. These MV PLA's realize the logic functions using the MIN (r-valued minimum), MAX (r-valued maximum), MODSUM (modular sum), linear SUM and TSUM (truncated sum) arrays. The number of inputs, outputs and product terms of suggested MV PLA's have physical restrictions. These structures do not permits the MVL functions with the large number of variables to be implemented because of the number of inputs is smaller than the number of input variables. The solution of this problem is to use the functional decomposition technique. The decomposition of a logic function can lead to a systematic design which incorporates readily designed simpler functions. Ideally, a systematic design would posses such desirable properties as regularity and testability.

One approach to decomposition of multiple-valued logic functions (MVL functions) is to extend disjunctive decomposition for the binary case [5]. The main drawback of this approach is follows. Since only a very small number of functions will have a specific decomposition property [5], if the function does not have the desired property, the analysis does not yield a design. In other hand, the specific property of MVL functions is that an arbitrary logic function can be given by different set of multiple-valued operators. These determines the basic multiple-valued gates from which a complex digital circuit implemented given MVL function is synthesized.

In this paper a NOR-TSUM MV PLA, proposed by Pelayo and etc., is assumed to be the component used to implement the function. A decomposition approach for the modular design of combinational multiple-valued circuits based on the NOR-TSUM PLA is described. The primary objective of this approach is to develop procedures which are applicable to the design of PLA-based combinational circuits and to extent the class of decomposable functions.

The rest of the paper is structured as follows. Section 2 provides definitions and notations, the distinctive features of TSUM operator are investigated in Section 3, Section 4 is devoted to MIN-TSUM and NOR-TSUM decompositions of the MVL function; the MV PLA - based technique is explained in Section 5, experimental results are disscussed in Section 6, the Section 7 draws a conclusion.

2. Definition and notations

Let us examine some r-valued operators by which we can represent any r-valued logic function. Let x and y be two r-valued variables. Then,

(a) $x \wedge y = x \cdot y = MIN(x, y)$;

(b) $x \oplus y = TSUM(x+y, r-1)$;

$$\begin{cases} r\text{-1}, & \text{if } x = s, \\ & \end{cases}$$

(c) $x^s =$ 0, otherwise, $s \in \{0,1,...,r\text{-}1\}$.

A product of the literals $x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \ldots \cdot x_n^{\sigma_n}$, is reffered to as a *product term* (also called *term* or *product* for short). In a product term, all the operations are AND operations. A product term that includes literals for all variables $x_1, x_2, ..., x_n$ is called a full term. An *r*-valued minimum of term and *r*-valued constant, $f(\boldsymbol{\sigma}) \cdot x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \ldots \cdot x_n^{\sigma_n}$ is denoted as an *r*-valued minimum term (also called *r-valued term*). An *r*-valued term that contains literals for all variables $x_1, x_2, ..., x_n$ is called a *full r-valued term*. A truncated sum of *r*-valued terms is called a truncated sum-of-minimums (TSOM) expression:

$$f(X) = \sum_{\boldsymbol{\sigma}=0}^{r^n-1}{}^{\mathrm{T}} f(\boldsymbol{\sigma}) \cdot x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \ldots \cdot x_n^{\sigma_n} \tag{1}$$

where the symbol $\sum^{\mathrm{T}}$ stands for TSUM operator, $f(\boldsymbol{\sigma})$ is the value of function evaluated with $X=\boldsymbol{\sigma}$, $\boldsymbol{\sigma}=(\sigma_1\sigma_2\ldots\sigma_n)_r$, $0 \neq f(\sigma) \in E^r=\{0, 1, ..., r\text{-}1\}$, $x_i \in E^r$, $\sigma_i \subseteq E^r$.

In other hand, this expression may be written as follows [1]:

$$f(X) = \sum_{\boldsymbol{\sigma}=0}^{r^n-1}{}^{\mathrm{T}} f(\boldsymbol{\sigma}) \Lambda \Big(\overline{\bigcup_{i=1}^{n}} x_i^{\overline{\sigma_i}} \Big). \tag{2}$$

where the symbol $\overline{\cup}$ stands for binary NOR operator.

Expression (2) states that any *r*-valued function may be expanded as a truncated sum of binary NOR terms weighted by the *r*-valued constants $f(\boldsymbol{\sigma})$.

Definition **1.** Let X is the set of *r*-valued variables. $\{X_1, X_2\}$ is a partition of X when $X_1 \cap X_2 = \varnothing$ and $X_1 \cup X_2 = X$.

To represent *r*-valued functions with minimal expressions, the following notations are adopted:

E^r : the set of constants is an *r*-valued Rosser algebra, $E^r=\{0, 1, ..., r\text{-}1\}$

X : the set of n *r*-valued variables $\{x_1,x_2,...,x_n\}$, where n is the number of elements of X.

X_1 : the bound set of variables;

X_2 : the free set of variables;

$d(X)$: the number of elements of the set X.

$\boldsymbol{\sigma}$: $(\sigma_1\sigma_2\ldots\sigma_n)$, where $\sigma_i \in E^r$, $i=1,2, ..., n$.

$\boldsymbol{\tau}$: $(\tau_1\tau_2\ldots\tau_{n_1}) \in \boldsymbol{\sigma}$, where $\tau_i \in E^r$, $i=1,2,...,n_1$, $d(X_1)=n_1$.

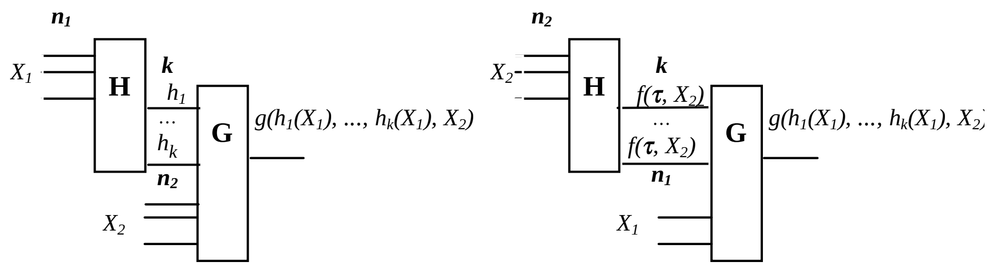
$\boldsymbol{\eta}$: $(\eta_1\eta_2\ldots\eta_{n_2}) \in \boldsymbol{\sigma}$, where $\eta_i \in E^r$, $i=1,2,...,n_2$, $d(X_2)=n_2$.

Definition **2.** Let $f(X)$ be an *r*-valued function and $\{X_1, X_2\}$ is a partition of X. Then the projection of $f(X)$ over $X_1=\boldsymbol{\tau}$, $f(\boldsymbol{\tau}, X_2)$ is the value of $f(X)$ evaluated with $X_1=\boldsymbol{\tau}$.

Definition **3.** An *r*-valued function $f(X)$ is said to have a *generalized decomposition* with a bound set X_1 and a free set X_2 if there exist *r*-valued functions $h_1,h_2,...,h_k$ and g such that

$$f(X)=g(h_1(X_1), h_2(X_1),..., h_k(X_1), X_2) \tag{3}$$

where $\{X_1, X_2\}$ is a partition of X.



(a) (b)

Fig. 1 Circuit diagram of $f(X)=g(h_1(X_1), h_2(X_1),..., h_k(X_1), X_2)$. (a) h_i is coded as *r*-valued variables, (b) $h_i=f(\boldsymbol{\tau}, X_2)$ is examined as the projections of $f(X)$ over $X_1=\boldsymbol{\tau}$.

There are two ways to realize generalized decomposition of MVL functions. The first way is applied in classical decomposition methods. Here, *r*-valued functions h_i are coded as *r*-valued variables of g (Fig. 1(a)). By second way, *r*-valued functions $h_1(X_1), h_2(X_1),..., h_k(X_1)$ can be examined as the projection of $f(X)$ over $X_1=\boldsymbol{\tau}$, $f(\boldsymbol{\tau},X_2)$ (Fig. 1(b)). In this paper the circuit diagram of $f(X)=g(h_1(X_1), h_2(X_1),..., h_k(X_1), X_2)$, shown in Fig. 1 (a), will be consider more detail.

3. The specific properties of TSUM operator

In this section we derive the specific properties of TSUM operator. Based on these features the MIN-TSUM and NOR-TSUM expansions with respect to a group of variables (Theorem 1 and Theorem 2) are obtained which will be used in the next section for designing the PLA-based combinational circuits.

Let x, y and z be r-valued logic variables. The following operations hold for multiple-valued algebra:

1. Associative laws: $(x\oplus y)\oplus x=x\oplus(y\oplus z)$
2. Commutative laws: $x \oplus y = y \oplus x$
3. Identities:
   - 3A. $(r\text{-}1) \oplus x=r\text{-}1$
   - 3B. $0 \oplus x=x$
   - 3C. $x^0\oplus x^1\oplus...\oplus x^{r-1}=r\text{-}1$
   - 3D. $1 \wedge x^1 \oplus 2 \wedge x^2 \oplus...\oplus (r\text{-}1) \wedge x^{r-1}=x$
4. Distributive laws:
   - 4A. $(x\wedge y)\oplus z=(x\oplus z)(y\oplus z)$
   - 4B. $(x\oplus y)\wedge z\neq(x\wedge z) \oplus (y\wedge z)$
   - 4C. $(x\oplus y)\wedge z^i=(x\wedge z^i) \oplus (y\wedge z^i)$.

All specific properties of TSUM operator are proved in Appendix 1.

4. Decomposition of NOR-TSUM and MIN-TSUM expansions

In this section we will describe the new representations of NOR-TSUM and MIN-TSUM expansions over the bound set, X_1 of variables.

The $Q_\sigma(X)$ term is a binary function defined by the Boolean product of literals and taking values in the $\{0, r\text{-}1\}$ set:

$$Q_\sigma(X) = x_1^{\sigma_1}\cdot x_2^{\sigma_2}\cdot.....\cdot x_n^{\sigma_n}$$

Theorem A. Let $f(X)$ be an r-valued logic function and $\{X_1, X_2\}$ be a partition of X. Let $n_1=d(X_1)$ and $n_2=d(X_2)$. Then the MIN-TSUM expansion of $f(X)$ over X_1 is given by

$$f(X) = \sum_{\tau=0}^{r^{n_1}-1}{}^{\mathrm{T}} Q_\tau(X_1) \sum_{\eta=0}^{r^{n_2}-1}{}^{\mathrm{T}} f(\tau,\eta)\wedge Q_\eta(X_2) \qquad (4)$$

where $f(\sigma) =f(\tau,\eta)\cdot$ is the value of $f(X)$ evaluated with $X_1=\tau$ and $X_2=\eta$, i.e. simple the value of $f(X)$ for σ, $\sigma=(\sigma_1\sigma_2......\sigma_n)_r$ be the input combination of X, $\sigma=\{\tau, \eta\}$, $Q_\sigma(X) = Q_\tau(X_1)\wedge Q_\eta(X_2)$.

(Proof) Transform expression (1) using conjunction and truncated sum properties. By definition of $Q_\sigma(X)$ term, expression (1) can be represented as:

$$f(X) = \sum_{\sigma=0}^{r^n-1}{}^{\mathrm{T}} f(\sigma)\wedge Q_\sigma(X)$$

Because $\{X_1, X_2\}$ is a partition of X, the function $Q_\sigma(X)$ can be expressed as $Q_\sigma(X)=Q_\tau(X_1)\wedge Q_\eta(X_2)$, where $\sigma=\{\tau, \eta\}$ is the input combination of X and τ and η are the input combinations of X_1 and X_2, respectively. Therefore,

$$f(X) = \sum_{\sigma=0}^{r^n-1}{}^{\mathrm{T}} f(\sigma)\cdot\wedge Q_\tau(X_1) \wedge Q_\eta(X_2)$$

Both conjunction and truncated sum are commutative (9), then

$$f(X) = \sum_{\sigma=0}^{r^n-1}{}^{\mathrm{T}} Q_\tau(X_1)\wedge f(\sigma)\cdot\wedge Q_\eta(X_2)$$

Note, that $Q_\tau(X_1)$ and $Q_\eta(X_2)$ are the binary functions and $f(\sigma)\cdot$takes the value of 0, 1, ..., $(r\text{-}1)$. Because $\sigma=\{\tau, \eta\}$ and both conjunction and truncated sum are submitted to all distributive laws (14, (16), so

$$f(X) = \sum_{\tau=0}^{r^{n_1}-1}{}^{\mathrm{T}} Q_\tau(X_1) \sum_{\eta=0}^{r^{n_2}-1}{}^{\mathrm{T}} f(\tau,\eta)\wedge Q_\eta(X_2)$$

Hence, the theorem is proved. ***Q.E.D.***

Note that the projection of $f(X)$ over X_1 for MIN-TSUM expansion is determined as follows:

$$f(\tau, X_2) = \sum_{\eta=0}^{r^{n_2}-1}{}^{T} f(\tau,\eta)\Lambda Q_{\eta}(X_2)$$

Example A. Consider a 3-valued 3 variable logic function $f(x_1,x_2,x_3)$ (r=3, n=3), shown in Table 1. Than the MIN-TSUM expansion with respect to x_1, x_2 variables for this function is given as

Таблица 1

Decomposition table for $f(x_1, x_2, x_3)$

x_1	0	0	0	1	1	1	2	2	2
x_3 x_2	0	1	2	0	1	2	0	1	2
0	2	2	1	1	0	1	0	1	0
1	2	1	0	1	2	0	0	0	2
2	1	0	1	1	0	2	2	1	0

$f(X)= \boldsymbol{x_1^0 x_2^0}(2\cdot x_3^0 \oplus 2\cdot x_3^1 \oplus 1\cdot x_3^2) \oplus \boldsymbol{x_1^1 x_2^0}(1\cdot x_3^0 \oplus 1\cdot x_3^1 \oplus 1\cdot x_3^2) \oplus \boldsymbol{x_1^2 x_2^0}(0\cdot x_3^0 \oplus 0\cdot x_3^1 \oplus 2\cdot x_3^2) \oplus \boldsymbol{x_1^0 x_2^1}(2\cdot x_3^0 \oplus 1\cdot x_3^1 \oplus 0\cdot x_3^2) \oplus \boldsymbol{x_1^1 x_2^1}(0\cdot x_3^0 \oplus 2\cdot x_3^1 \oplus 0\cdot x_3^2) \oplus \boldsymbol{x_1^2 x_2^1}(1\cdot x_3^0 \oplus 0\cdot x_3^1 \oplus 1\cdot x_3^2) \oplus \boldsymbol{x_1^0 x_2^2}(1\cdot x_3^0 \oplus 0\cdot x_3^1 \oplus 1\cdot x_3^2) \oplus \boldsymbol{x_1^1 x_2^2}(1\cdot x_3^0 \oplus 0\cdot x_3^1 \oplus 2\cdot x_3^2) \oplus \boldsymbol{x_1^2 x_2^2}(0\cdot x_3^0 \oplus 2\cdot x_3^1 \oplus 0\cdot x_3^2)$. (*End of example*)

Theorem B. Let $f(X)$ be an r-valued logic function and $\{X_1, X_2\}$ be a partition of X. Let $n_1=d(X_1)$ and $n_2=d(X_2)$. Then the expansion of $f(X)$ over X_1 for NOR-TSUM expression is given by

$$f(X) = \sum_{\tau=0}^{r^{n_1}-1}{}^{T} \Big(\overline{\bigcup_{j=1}^{n_1}} x_j^{\bar{\tau}_j}\Big) \sum_{\eta=0}^{r^{n_2}-1}{}^{T} f(\tau,\eta)\Lambda\Big(\overline{\bigcup_{i=1}^{n_2}} x_i^{\bar{\eta}_i}\Big) \tag{5}$$

where the symbol $\overline{\cup}$ stands for binary NOR operator.

(Proof) By *theorem a* any r-valued function can be represented as follows:

$$f(X) = \sum_{\tau=0}^{r^{n_1}-1}{}^{T} Q_{\tau}(X_1) \sum_{\eta=0}^{r^{n_2}-1}{}^{T} f(\tau,\eta)\Lambda Q_{\eta}(X_2) \tag{6}$$

Each term in previous expression may be converted as follows [1]

$$Q_{\tau}(X_1) = x_1^{\tau_1}\cdot\Lambda x_2^{\tau_2}\cdot\Lambda\ldots\ldots\cdot\Lambda x_{n_1}^{\tau_{n_1}} = x_1^{\bar{\tau}_1}\cdot \overline{\cup}\, x_2^{\bar{\tau}_2}\cdot \overline{\cup}\ldots\ldots\cdot \overline{\cup}\, x_{n_1}^{\bar{\tau}_{n_1}} = \overline{\bigcup_{i=1}^{n_1}} x_i^{\bar{\tau}_i} \tag{7}$$

Taking into account the expression (7) we can make the substitution in (6). Hence we have a theorem. ***Q.E.D.***

Note that the projection of $f(X)$ over X_1 for NOR-TSUM expansion is determined as follows:

$$f(\tau,X_2) = \sum_{\eta=0}^{r^{n_2}-1}{}^{T} f(\tau,\eta)\Lambda\Big(\overline{\bigcup_{i=1}^{n_2}} x_i^{\bar{\eta}_i}\Big)$$

Example B. The NOR-TSUM expansion with respect to x_1, x_2 variables for function described in *example a* is offered as

$f(X)= \boldsymbol{x_1^0}\ \overline{\cup}\boldsymbol{x_2^0}\,\Lambda(0\cdot\Lambda x_3^0 \oplus 2\cdot\Lambda x_3^1 \oplus 0\cdot\Lambda x_3^2) \oplus \boldsymbol{x_1^1}\ \overline{\cup}\ \boldsymbol{x_2^0}\,\Lambda(1\cdot\Lambda x_3^0 \oplus 0\cdot\Lambda x_3^1 \oplus 2\Lambda\cdot x_3^2) \oplus \boldsymbol{x_1^2}\ \overline{\cup}\ \boldsymbol{x_2^0}\ \Lambda(1\Lambda\cdot x_3^0 \oplus 0\Lambda\cdot x_3^1 \oplus 1\Lambda\cdot x_3^2) \oplus \boldsymbol{x_1^0}\ \overline{\cup}\boldsymbol{x_2^1}\,\Lambda(1\cdot\Lambda x_3^0 \oplus 0\Lambda\cdot x_3^1 \oplus 1\cdot\Lambda x_3^2) \oplus \boldsymbol{x_1^1}\ \overline{\cup}\boldsymbol{x_2^1}\,\Lambda(0\cdot\Lambda x_3^0 \oplus 2\Lambda\cdot x_3^1 \oplus 0\Lambda\cdot x_3^2) \oplus \boldsymbol{x_1^2}\ \overline{\cup}\boldsymbol{x_2^1}\,\Lambda(2\Lambda\cdot x_3^0 \oplus 1\Lambda\cdot x_3^1 \oplus 0\Lambda\cdot x_3^2) \oplus \boldsymbol{x_1^0}\ \overline{\cup}\boldsymbol{x_2^2}\,\Lambda(0\cdot\Lambda x_3^0 \oplus 0\cdot\Lambda x_3^1 \oplus 2\Lambda\cdot x_3^2) \oplus \boldsymbol{x_1^1}\ \overline{\cup}\boldsymbol{x_2^2}\,\Lambda(1\Lambda\cdot x_3^0 \oplus 1\Lambda\cdot x_3^1 \oplus 1\cdot\Lambda x_3^2) \oplus \boldsymbol{x_1^2}\ \overline{\cup}\boldsymbol{x_2^2}\,\Lambda(2\cdot\Lambda x_3^0 \oplus 2\cdot\Lambda x_3^1 \oplus 1\cdot\Lambda x_3^2)$. (*End of example*)

5. *Design method for MV PLA-based combinational circuits*

In this section we present a design method for MV PLA based combinational circuits using decomposition of NOR-TSUM and MIN-TSUM expansions.

This novel method incorporates the concept of systematically reexpressing the given MVL function. In our approach, we do not attempt to rewrite the function in terms of functions which are then used as new variables in the logic equation. Rather, we identify the minimum number of projections of $f(X)$ over X_1 and reexpressed the given function by obtained partition of X. Each of fucntion projection is then utilized as r-valued constant in the logic equation. The function is reespressed recursively until it is represented entirely by the function projections. In this case the partition of X composes p subsets of variables $\{X_1, X_2, ..., X_p\}$, where p is the number of function reexpressions. In this paper we consider only partition of X containing two subsets. Therefore, function $f(X)$ is reexpressed only once (Theorem 1 and Theorem 2).

The r-valued combinational circuit composes of two level. On the first level the projections of $f(X)$ over X_1 are implemented by MV PLA. Let us consider the MV PLA proposed in [1], and called NOR-TSUM PLA. In this case the number of inputs corresponds to the number of variables in the free set, X_2. The number of outputs is determined by the number of different projections in the NOR-TSUM expansion. On the second level of combinational circuit the function projections over X_1 are identified as r-valued constants. The number of inputs in the second NOR-TSUM PLA is determined by the number of variables in bound set X_1. Fig. 2(a) illustrates the structure of two-level PLA-based combinational circuit.

Note that instead of MV PLA we can use any kind of universal logic module such as T-gate (multiplexer), programmable logic array (PAL), look-up table type FPGA (field programmable gate array) and etc..

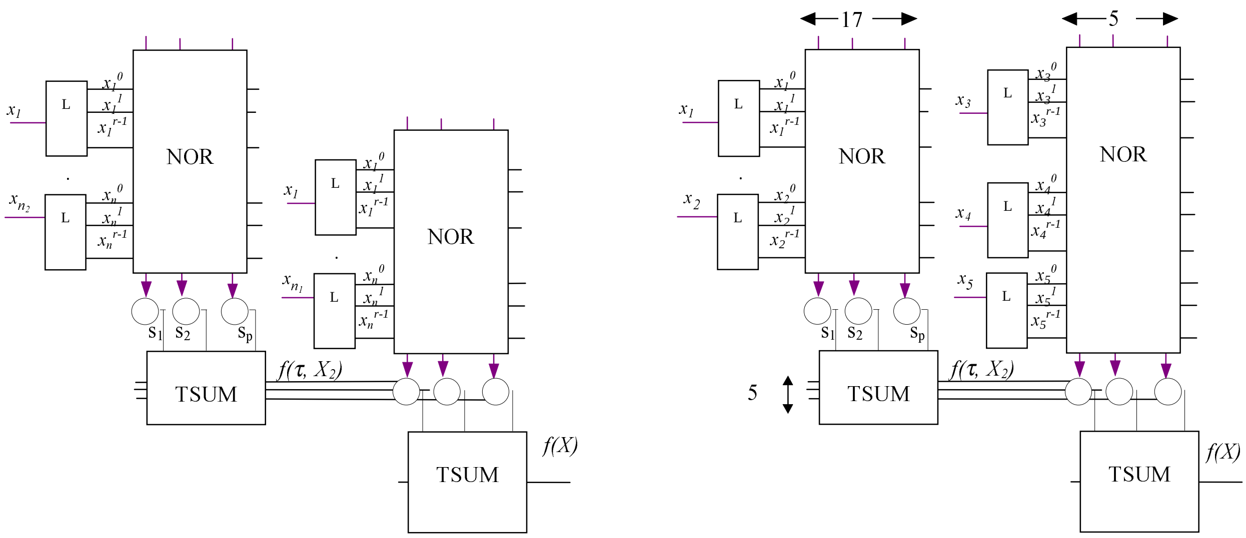
Example 3. Таблица 2 shows a 3-valued 5 variable function, $f(x_1, x_2, x_3, x_4, x_5)$. A 3 variable MV PLA is assumed to be the component used to implement the function. Since the MV PLA uses not more than 3 variables, only 3 variables are used to partition function $f(X)$. No attempt is made to try 4 variables in an attempt to decompose the function.

Selecting x_1 and x_2 as the variables of the free set, X_2, the function $f(X)$ is rewritten as a 3 variable function and its functional values become the projections of $f(X)$ over X_1. Then, there are 5 different projections of $f(X)$ over $X_1=\tau$, $f(\tau, X_2)$, that is the number of product terms in the second MV PLA. The r-valued combinational circuit implementing $f(X)$ function is shown in Fig. 2(b).

Таблица 2

Decomposition table of $f(x_1, x_2, x_3, x_4, x_5)$

x_5	0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1	2 2 2 2 2 2 2 2 2
x_4	0 0 0 1 1 1 2 2 2	0 0 0 1 1 1 2 2 2	0 0 0 1 1 1 2 2 2
x_3	0 1 2 0 1 2 0 1 2	0 1 2 0 1 2 0 1 2	0 1 2 0 1 2 0 1 2
x_2 x_1			
00	2 0 0 0 0 2 2 2 0	0 2 2 2 2 0 1 0 2	2 0 0 0 0 2 2 2 0
01	2 1 1 1 1 2 0 2 1	0 0 0 0 0 0 1 0 0	2 1 1 1 1 2 0 2 1
02	1 0 0 0 0 1 2 1 0	0 1 1 1 1 0 0 0 1	1 0 0 0 0 1 2 1 0
10	1 0 0 0 0 1 2 1 0	0 1 1 1 1 0 0 0 1	1 0 0 0 0 1 2 1 0
11	2 0 0 0 0 2 2 2 0	0 2 2 2 2 0 1 0 2	2 0 0 0 0 2 2 2 0
12	2 1 1 1 1 2 0 2 1	0 0 0 0 0 0 1 0 0	2 1 1 1 1 2 0 2 1
20	2 1 1 1 1 2 0 2 1	0 0 0 0 0 0 1 0 0	2 1 1 1 1 2 0 2 1
21	2 0 0 0 0 2 2 2 0	0 2 2 2 2 0 1 0 2	2 0 0 0 0 2 2 2 0
22	1 0 0 0 0 1 2 1 0	0 1 1 1 1 0 0 0 1	1 0 0 0 0 1 2 1 0
	a b b b b a c a b	*0 d d d d 0 e 0 d*	*a b b b b a c a b*



(b) (b)

Fig. 2 The structure of PLA-based combinational circuit (a) common case, (b) for function $f(x_1, x_2, x_3, x_4, x_5)$.

Таблица 3

Function	PLA by method [1]				Combinational circuit by the proposed method								
					The first PLA				The second PLA				
	In	Out	W^*	S^*	In	Out	W_1	S_1	In	Out	W_2	S_2	S_1+S_2
$X+Y$													

S_0	4	1	13	**153**	2	1	8	136	2	1	1	17	**153**
S_1	4	1	35	**459**	2	4	15	300	2	1	16	272	**572**
S_2	4	1	11	**187**	2	3	4	76	2	1	10	170	**246**
$X*Y$													
M_0	4	1	7	**213**	2	4	3	60	2	1	9	153	**213**
M_1	4	1	156	**2320**	2	16	64	2048	2	1	16	272	**2320**
M_2	4	1	138	**2549**	2	15	74	2294	2	1	15	255	**2549**
M_3	4	1	57	**992**	2	10	29	754	2	1	14	238	**992**
3 digit shift register													
R_0	4	1	5	**128**	2	3	3	60	2	1	4	68	**128**
R_1	4	1	5	**51**	2	3	2	34	2	1	1	17	**51**
R_2	4	1	5	**51**	2	3	2	34	2	1	1	17	**51**
R_3	4	1	5	**128**	2	3	3	60	2	1	4	68	**51**

6. Experimental results

Таблица 3 compares the size of synthesized circuits with the size of MV PLA implemented the initial function. This table shows that the present method produces the competitive results with other methods. Because the proposed decomposition approach permits only one *r*-valued function to be decomposed the functions from the standardized benchmarks are separately implemented. Here such benchmarks as 2-figure quaternary adder, 2-figure quaternary multiplier, 3 digit shift register are used for experimental investigations.

7. Conclusion

A new technique for realizing PLA-based combinational circuits has been proposed. This technique yields multiple-valued circuit of reduced size. This has been illustrated by considering the realization of functions from standardized benchmarks. Within suggested approach the novel MIN-TSUM and NOR-TSUM expansions with respect to X_1 are derived. The specific properties of TSUM operator are investigated too. It is shown that truncated sum of *r*-valued variables does not submit one of the distributive laws. It needs to take into account the consideration at examining the MIN-TSUM and NOR-TSUM expansions.

Future work is carried out in frame of generalization the proposed approach for system of MVL functions. It leads to investigate the performance of new approach on other standardized benchmark problem.

8. Acknowledgment

The authors sincerely thank Dr. V. Shmerko fron Technical University (Szczecin, Poland) for his helpful comments and suggestions on this paper.

Appendix 1

1. Associative laws: (8) $(x \oplus y) \oplus x = x \oplus (y \oplus z)$

(Proof) $(x_1 \oplus x_2) \oplus x_3 = MIN\ (x_1+x_2,\ r-1) \oplus x_3 = MIN\ (x_1+x_2+x_3,\ r-1+x_3,\ r-1) = MIN(x_1+x_2+x_3,\ r-1)$, (*a*) because $r-1+z \geq r-1$ for all values of *z*.

In other hand, $x_1 \oplus (x_2 \oplus x_3) = x_1 \oplus MIN(x_2+x_3,\ r-1) = MIN\ (x_1+x_2+x_3,\ r-1+x_1,\ r-1) = MIN(x_1+x_2+x_3,\ r-1)$, (*b*) because $r-1+x \geq r-1$ for all values of *x*.

Since expressions (*a*) and (*b*) are equivalent, then *TSUM* operator is associative. ***Q.E.D.***

2. Commutative laws: (9) $x \oplus y = y \oplus x$

(Proof) This expression is proved using the definition of TSUM operator.

3. Identities: (10) $(r-1) \oplus x = r-1$

(Proof) $(r-1) \oplus x = MIN(r-1+x,\ r-1) = r-1$, because $r-1+x \geq r-1$ for all values of *x*. ***Q.E.D.***

(11) $0 \oplus x = x$

(Proof) $0 \oplus x = MIN(0+x,\ r-1) = MIN(x,\ r-1) = x$. ***Q.E.D.***

(12) $x^0 \oplus x^1 \oplus ... \oplus x^{r-1} = r-1$

(Proof) This expression describes all input combination of *x*, therefore by definition of literal function Eq. (12) takes value $(r-1)$. ***Q.E.D.***

(13) $1 \Lambda x^1 \oplus 2 \Lambda x^2 \oplus ... \oplus (r-1) \Lambda x^{r-1} = x$, where the symbol Λ stands for *r*-valued minimum.

(Proof) $x=0 \Lambda x^0 \oplus 1 \Lambda x^1 \oplus 2 \Lambda x^2 \oplus ... \oplus (r-1) \Lambda x^{r-1}=1 \Lambda x^1 \oplus 2 \Lambda x^2 \oplus ... \oplus (r-1) \Lambda x^{r-1}$, because $0 \Lambda x^0=0$. ***Q.E.D.***

*4. D*istributive laws.

(14) $(x \Lambda y) \oplus z=(x \oplus z)(y \oplus z)$, where $x, y, z \in \{0, 1, ..., r-1\}$

(Proof)

a) $x \Lambda y \oplus z = MIN (MIN (x, y) + z, r-1) = MIN (x+z, y+z, r-1)$

b) $(x \oplus z)(y \oplus z)=MIN (x+z,r-1) \Lambda MIN (y+z,r-1)=MIN(MIN(x+z,r-1),MIN (y+z,r-1))=MIN (x+z, y+z, r-1)$.

Since expressions (*a*) and (*b*) are equivalent, then *TSUM* operator submit to the distributive law. ***Q.E.D.***

(15) $(x \oplus y) \Lambda z \neq (x \Lambda z) \oplus (y \Lambda z)$.

(Proof)

a) $(x \oplus y) \Lambda z = MIN (x+y, r-1) \Lambda z = MIN(MIN(x+y, r-1), z))= MIN (x+y, z, r-1)$.

b) $(x \Lambda z) \oplus (y \Lambda z)=MIN (x, z) \oplus MIN(y, z) =MIN(MIN(x, z)+MIN(y, z), r-1) = MIN(x+y, z+x, 2z, y+z, r-1)$

Because, the expressions (*a*) and (*b*) are not equivalent, *TSUM* operator does not submit to the distributive laws. ***Q.E.D.***

(16) $(x \oplus y) \Lambda z^i=(x \Lambda z^i) \oplus (y \Lambda z^i)$.

(Proof)

a) $(x \oplus y) \Lambda z^i = MIN (x+y, r-1) \Lambda z^i = MIN(MIN(x+y, r-1), z^i)) = MIN (x+y, z^i, r-1)$.

b) $(x \Lambda z^i) \oplus (y \Lambda z^i)= MIN(x, z^i) \oplus MIN(y, z^i) =MIN(MIN(x, z^i)+MIN(y, z^i), r-1) = MIN(x+y, z^i+x, 2z^i, y+z^i, r-1)$

Because $z^i \in \{0,r-1\}$, then

$$x+z^i=MAX(x, y) = \begin{cases} x, & \text{если } z=0, \\ x+r-1, & \text{otherwise.} \end{cases}$$

But $x+r-1 \geq r-1$, Eq. (*b*) takes into account variable z^i, therefore $x+z^i$ and z^i do not examine.

The expression $y+z^i$ is analyzed by analogy with previous expression.

(c) $MIN (x+y, z^i+x, 2z^i, y+z^i, r-1)=MIN(x+y, z, r-1)$.

Since expressions (*a*) and (*c*) are equivalent, then *TSUM* operator submits to the distributive law. ***Q.E.D.***

References

1 Pelayo F.J. and others, CMOS Current - Mode Multiple-Valued PLA’s. *IEEE Trans. Circut and Syst.*, 1991, Vol. 38, pp. 434-441.

2. Asari K., C. Eswaran, An Optimisation Technique for the Design of Multiple-Valued PLA’s. *IEEE Trans. Comput.*, vol.43, No. 1, 1994, pp.118-122.

3. Abd-El-Barr M. and H. Choy, Use of Simulated Annealing to Reduce 2-bit Decoder PLAs. *Int. J. Electronics*, 1993, Vol. 74, NO. 3, pp.441-450.

4. Deng X., T.Hanyu and M.Kameyama, Design And Evaluation Of A Current-Mode Multiple-Valued PLA Based On A Resonant Tunnelling transistor Model. *IEE Proc.- Circuits Devices Syst.*, Vol. 141, No.6, 1994, pp.445-450.

5. Abugharieh S.B., S.C. Lee, A Fast Algorithm For The Disjunctive Decomposition Of *M*-Valued Functions. Part 1. The Decomposition Algorithm. *IEEE Trans. on Computers*. 1995, pp. 118-125.