Information Theoretic Approach to Minimization of Arithmetic Expressions

D. Popel*, **S. Yanushkevich†**, **M. Perkowski‡**, **P. Dziurzanski†**, **V. Shmerko†**

† * Faculty of Computer Science & Information Systems, Technical University of Szczecin, Szczecin, POLAND
⟨ dpopel, yanushkevich, pdziurzanski, shmerko ⟩@wi.ps.pl

‡ Department of Electrical and Computer Engineering Portland State University, Portland, USA
mperkows@ee.pdx.edu

Abstract

In many applications, it is quite efficient to represent and manipulate logic functions at the word-level in contrast to bit-level representations. This paper develops information theoretic approach to minimization of arithmetic expressions using free word-level Decision Trees (DTs). We introduce new class of word-level DTs, such as free pseudo Binary Moment Trees and free pseudo Kronecker Moment Trees. Thanks to information theory paradigms, we propose an efficient algorithm to minimize multi-input/output switching functions in many circuit design styles. We have provided a comparison and experimental verification of our approach with some known approaches to minimize arithmetic expressions. In some cases, our program $Info{-}A$ produces significantly better results.

1 Introduction

A great deal of DTs to represent integer-valued functions, in particular, *Binary Moment Trees (BMTs) and Binary Moment Diagrams (BMDs)* [1], has been developed in the last few years. Unfortunately, BMDs fail for some switching functions which can be easily represented using *Decision Diagrams (DDs)* [4]. For such cases, a data structure, called *Kronecker BMD* ($KBMD$), has been proposed as a generalization of both BMD and DD [2]. The applications of BMDs, KBMDs and their extensions include verification [6], linear integer programming and functional decomposition [8]. On the other hand, arithmetic forms of switching functions have been introduced in association with such well known

*Support from State University of Informatics and Radioelectronics, Minsk (BELARUS) and the Fund of Fundamental Researches (BELARUS) is acknowledged

concepts as *pseudo-Boolean function* and *integer valued functions, arithmetic expression*, and *spectral representations* [13, 14].

All mentioned papers considered ordered word-level DTs and DDs. Firstly, the background of information theory approach to minimize switching functions was given in [7, 15]. Such an approach was based on interpretation of bit-level DTs design process from the information theory point of view. Here, we introduce another type of word-level DTs, namely, *free pseudo Kronecker Binary Moment Trees*, and extend our information theoretic approach to minimize arithmetic expressions.

The paper is organized as follows. Section 2 outlines the background of investigation and presents the necessary definitions and notations. Section 3 presents an information theoretic content of word-level DTs. The algorithm to minimize arithmetic expressions is given in Section 4. Our experimental results are reported in Section 5. Section 6 concludes the paper.

2 Background

In the following we consider a *switching function* f, as the mapping $\mathbf{B}^n \rightarrow \mathbf{B}^m$ over the variable set $X = \{x_1, \cdots, x_n\}$, where $\mathbf{B}$=$\{0,1\}$. Here, n is the number of variables, and m is the number of functions (outputs).

2.1 Word-level Decision Trees and Arithmetic Expressions

In this paper we introduce *free pseudo BMTs* and *free pseudo KBMTs* and treat them by using the following expansions:

arithmetic Shannon S_A expansion

$$f = (1-x)f_{|x=0} + xf_{|x=1},$$

arithmetic positive Davio (pD_A) expansion

$$f = f_{|x=0} + x(f_{|x=1} - f_{|x=0}),$$

arithmetic negative Davio (nD_A) expansion[1]

$$f = f_{|x=1} + (1-x)(f_{|x=0} - f_{|x=1}).$$

In Table 1, some types of DT for integer-valued representations and corresponding expansion type assignments to the DT nodes are shown. Each non-zero path from the

[1] pD_A and nD_A derive from S_A expansion, e.g. for pD_A:

$$f = f_{|x=0} - xf_{|x=0} + xf_{|x=1} = f_{|x=0} + x(f_{|x=1} - f_{|x=0})$$

Type of DT	Type of expansion
MT (Multi-terminal)	S_A
BM (Binary Moment)	pD_A, nD_A (fixed for each variable)
PSDBM (pseudo Binary Moment)	pD_A, nD_A
KBM (Kronecker Binary Moment)	S_A, pD_A, nD_A (fixed for each variable)
PSDKBM (pseudo Kronecker Binary Moment)	S_A, pD_A, nD_A

Table 1: Word-level DTs and corresponding expansion types

Logic function	Arithmetic expression	Logic function	Arithmetic expression
x	x	$x_1 \sim x_2$	$1 - x_1 - x_2 + 2x_1x_2$
$\overline{x}$	$1 - x$	$x_1 \uparrow x_2$	$1 - x_1x_2$
$x_1 \vee x_2$	$x_1 + x_2 - x_1x_2$	$x_1 \vert x_2$	$1 - x_1 - x_2 + x_1x_2$
x_1x_2	x_1x_2	$x_1 \oplus x_2$	$x_1 + x_2 - 2x_1x_2$
$x_1 \rightarrow x_2$	$1 - x_1 + x_1x_2$		

Table 2: Simple switching functions and their arithmetic equivalents

x_3x_4 \ x_1x_2	0 0	0 1	1 1	1 0
0 0	0	1	1	0
0 1	0	1	1	0
1 1	0	0	1	1
1 0	0	1	1	1

Figure 1: The Karnaugh map of a switching function f of four variables (Example 1)

root to a terminal node of the BMT (KBMT) corresponds to a term in the arithmetic expression. Multi-output functions, described by arithmetic expressions [9], can be also efficiently represented by word-level DTs.

Example 1. *Consider a function f given by the Karnaugh map in Figure 1. By applying the well-known symbolic transformations, introduced in [10] (Table 2), this function is represented by the arithmetic expression:* $f = x_1x_3 \vee x_2\overline{x}_3 \vee x_2\overline{x}_4 = (x_1x_3 + x_2\overline{x}_3 - x_1x_2x_3\overline{x}_3 + x_2\overline{x}_4 - (x_1x_3 + x_2\overline{x}_3)x_2\overline{x}_4 = (x_1x_3 + x_2(1-x_3)) + x_2(1-x_4) - (x_1x_3 + x_2(1-x_3))x_2(1-x_4) = x_1x_3 + x_2 - x_1x_2x_3 - x_2x_3x_4 + x_1x_2x_3x_4$.

2.2 Information Theoretic Concepts

In this paper we will use some concepts of information theory, namely, *entropy and conditional entropy.*

In order to quantify the content of information for a finite field of events $A = \{a_1, a_2, \cdots, a_n\}$ with the probabilities distribution $\{p(a_i)\}$, $i = 1, 2, \cdots, n$, Shannon introduced the concept of entropy. *Entropy* of the finite field A is given by

$$H(A) = -\sum_{i=1}^{n} p(a_i) \log p(a_i), \tag{1}$$

where logarithm is in base 2.

For two finite fields of events A and B with probability distribution $\{p(a_i)\}, i = 1, 2, \cdots, n$, and $\{p(b_j)\}$, $j = 1, 2, \cdots, m$, probability of the joint occurrence of a_i and b_j, $p(a_i, b_j)$, and conditional probability, $p(a_i|b_j) = p(a_i, b_j)/p(b_j)$, Shannon defined the *conditional entropy* of A given B

$$H(A|B) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(a_i, b_j) \log p(a_i|b_j). \tag{2}$$

Example 2. *Consider the function f from Example 1. The probabilities of its output values are $p_{|f=0} = {}^7/_{16}$ and $p_{|f=1} = {}^9/_{16}$. The entropy of the function is $H(f) = -{}^7/_{16} \log_2 {}^7/_{16} - {}^9/_{16} \log_2 {}^9/_{16} = 0.99$ bit/pattern, and conditional entropy of f given variable x_1 is $H(f|x_1) = -{}^5/_{16} \log_2 {}^5/_8 - {}^3/_{16} \log_2 {}^3/_8 - {}^2/_{16} \log_2 {}^2/_8 - {}^6/_{16} \log_2 {}^6/_8 = 0.88$ bit/pattern, and also $H(f|x_2) = 0.67$ bit/pattern, $H(f|x_3) = 0.98$ bit/pattern, $H(f|x_4) = 0.98$ bit/pattern .*

Entropy and conditional entropy can be exploited for minimization of switching function, which is given by AND/OR and AND/EXOR expressions [11]. Shannon entropy can be interpreted as a measure of complexity during switching function minimization and decomposition [5].

3 Information Theoretic Content of Word-level Decision Trees

We propose the following information theoretic measures of arithmetic analogues of Shannon and Davio expansions (Table 3). It implies the following lemma.

Lemma 1. *The average entropy of cofactors of S_A expansion is equal to conditional entropy $H(f|x)$ of a function f given variable x:*

$$H^{S}(f|x) = H^{S_A}(f|x) = H(f|x). \tag{3}$$

The information measures of S_A, pD_A and nD_A expansions are adequate to the information model of DT design presented in [15]. Thus, one can conclude that $H(f_{|x=0} - f_{|x=1}) = H(f_{|x=1} - f_{|x=0})$ and

$$H^{pD_A} + H^{nD_A} \geq H^{S_A}. \tag{4}$$

Type	**Expansion**	**Information measures**
S_A	$f = (1-x)f_{\|x=0} + xf_{\|x=1}$	$H^{S_A} = p_{\|x=0}H(f_{\|x=0}) + p_{\|x=1}H(f_{\|x=1})$
pD_A	$f = f_{\|x=0} + x(f_{\|x=1} - f_{\|x=0})$	$H^{pD_A} = p_{\|x=0}H(f_{\|x=0}) + p_{\|x=1}H(f_{\|x=1} - f_{\|x=0})$
nD_A	$f = f_{\|x=1} + (1-x)(f_{\|x=0} - f_{\|x=1})$	$H^{nD_A} = p_{\|x=1}H(f_{\|x=1}) + p_{\|x=0}H(f_{\|x=0} - f_{\|x=1})$

Table 3: Arithmetic analogues of Shannon (S_A) and Davio (pD_A, nD_A) expansions of switching functions and their information theoretic measures

Example 3. *For the function f from Example 1, information theoretic measures (in bit/pattern) of its S_A, pD_A and nD_A expansions are calculated as follows for all possible input variables*

X	$H^{S_A}(f\|x)$	$H^{pD_A}(f\|x)$	$H^{nD_A}(f\|x)$
x_1	*0.88*	*0.95*	*0.88*
x_2	**0.67**	*0.88*	*0.75*
x_3	*0.98*	*1.15*	*1.13*
x_4	*0.98*	*0.75*	*0.75*

Since the minimal value of the information measure has been found for the S_A expansion of function f with respect to variable x_2, we create the root of the tree with the pair (x_2, S_A), which means that we select variable x_2 and apply arithmetic Shannon expansion.

4 Algorithm to Minimize Arithmetic Expressions

A sketch of the algorithm to minimize arithmetic expressions via word-level DTs design is given in Figure 2. Our algorithm for switching functions minimization is the following.

- Attach a node to the DT, we calculate the information theoretic measures for S_A, pD_A and nD_A expansions, and the pair (x, ω) that corresponds to minimal $H^{\omega}(f|x)$, is assigned to the current DT node.
- Recursively construct sub-trees for the sub-functions obtained by expansion ω with respect to variable x.
- Algorithm terminates if the leaves are achieved for each sub-tree (i.e. DT is completed) for the given logic function f.

Example 4. *Consider the function f from Example 1. The PSDBMT by $Info{-}A^{PSDBMT}$ and PSDKBMT produced by $Info{-}A^{PSDKBMT}$ are shown in Figure 3(a) and (b), respectively. The corresponding arithmetic expression is:*

$$f = x_2 + x_1(1-x_2)x_3 - (1-x_1)x_2x_3x_4.$$

```
Input Switching function f
OutputTree - Decision Tree, arithmetic expression
Comments
For pseudo Kronecker BMT: ω ∈ {S_A, pD_A, nD_A}, for pseudo BMT: ω ∈ {pD_A, nD_A},
for Multiple-terminal DT: ω ∈ {S_A}.

Info–A(f) {
        if (f = c, where c = const) then {
           Tree ← leaf(c); return;
        }
        for (∀ pairs (x_i, ω_j))
           Calculate the information theoretic measures H^{ω_j}(f|x_i)

        Choose variable x and expansion type ω, where:
                          H^ω(f|x) = min(H^{ω_j}(f|x_i) | ∀ pairs (x_i, ω_j));
        Attach a node assigned by the pair (x, ω) to Tree
                               Tree ← node(x, ω);

        for (∀f_s of expansion ω with respect to variable x)
             Recursively construct the sub-trees Tree_s:
                          Tree_s = Info–A(f_s);
         return;
}
```

Figure 2: Sketch of the $Info$–A algorithm to minimize arithmetic expressions of switching functions

5 Experimental Results

Our $Info-A$ program in C++ implements the described above algorithm to minimize arithmetic expressions. All experiments have been carried out on Pentium 400 MHz with 128 Mb of main memory. In all Tables, the following notations are used: the number of terms T, literals L and run-time t in CPU seconds. Label **I/O** means the number of inputs and outputs.

In the first series of experiments, we studied the following criteria $H^{\omega}(f|x) = min(H^{\omega_j}(f|x_i) \mid \forall\ pairs\ (x_i, \omega_j))$, where $\omega \in \{pD_A, nD_A\}$ to design free BMT, $\omega \in \{S_A, pD_A, nD_A\}$ to design free KBMT.

The results have been compared with *Staircase* strategy to minimize arithmetic expressions via truth table manipulation, reported in [3]. Some of these results are listed for multi-output functions in Table 4.

Observation 1. *The results demonstrate the advantage of $Info-A$ in comparison to Staircase strategy [3] in the number of terms (about 2 times), literals (about 2 times) and run-time. $Info$–A^{KBMT} produces fewer terms and literals than $Info-A^{BMT}$ does.*

In the second series of experiments, we compared the results of the algorithm $Info$ [15] and the proposed $Info$–A. A fragment of our results is given in Table 5.

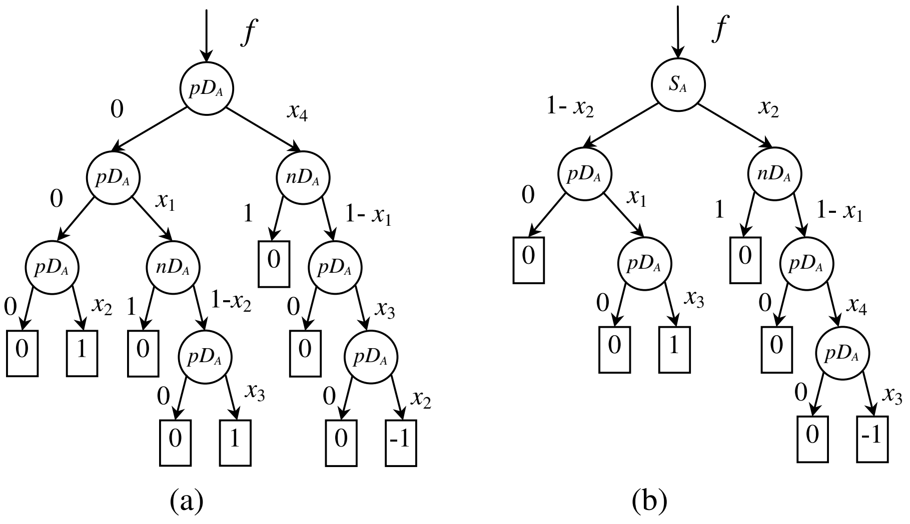


Figure 3: Pseudo BMT and pseudo KBMT for the function f from Example 1

		Staircase [3]			$Info{-}A^{BMT}$			$Info{-}A^{KBMT}$		
	I/O	T	L	t^*	T	L	t	T	L	t
xor5	5/1	31	80	0.06	31	80	0.00	31	80	0.00
bw	5/28		-		22	65	0.00	22	65	0.00
squar5	5/8	23	56	0.06	**14**	**24**	**0.00**	**14**	**24**	**0.00**
m1	6/12	30	179	0.44	**18**	**68**	**0.13**	**18**	**68**	**0.13**
rd73	7/3	127	448	0.80	**75**	**313**	**0.01**	**71**	**333**	**0.01**
rd84	8/4		-		190	857	0.06	150	836	0.07
log8mod	8/5		-		37	125	0.02	35	120	0.23
misex1	8/7	27	121	3.40	**18**	**61**	**0.01**	**16**	**60**	**0.03**
mlp4	8/8		-		16	32	0.02	16	32	0.02
clip	9/5	240	1212	62.72	**218**	**1119**	**1.50**	**135**	**773**	**2.18**
newcpla1	9/16	158	1192	51.25	**58**	**332**	**0.75**	**44**	**273**	**6.70**
newtpla2	10/4	256	1025	185.20	**10**	**55**	**0.12**	**10**	**55**	**0.12**
Total		903	4343	303.93	**442**	**2052**	**2.52**	**339**	**1666**	**9.17**
Total					707	3131	**2.62**	**562**	**2719**	9.49

Note: * *Pentium 166Mhz*

Table 4: Comparison of *Staircase* strategy [3] and $Info{-}A$

	I/O	$Info\!-\!A^{PSDBMT}$ T	L	t	$Info^{PSDKRO}$ [15] T	L	t	$Info\!-\!A^{PSDKBMT}$ T	L	t
xor5	5/1	31	80	0.00	**5**	**5**	**0.00**	31	80	0.00
squar5	5/8	**14**	**24**	**0.00**	22	54	0.01	**14**	**24**	**0.00**
bw	5/28	22	65	0.00	22	65	0.00	22	65	0.00
con1	7/2	**11**	**35**	**0.75**	13	40	0.00	**11**	**35**	**0.75**
z4	7/4	**7**	**7**	**0.00**	32	89	0.04	**7**	**7**	**0.00**
inc	7/9	41	147	0.45	32	136	0.20	34	146	0.15
5xp1	7/10	**24**	**87**	**0.06**	39	159	0.01	**22**	**82**	**0.09**
root	8/5	**80**	**381**	**0.72**	64	351	1.33	**60**	**327**	**1.12**
log8mod	8/5	**37**	**125**	**0.03**	39	153	1.77	**35**	**120**	**0.42**
f51m	8/8	44	197	2.91	**23**	**91**	**8.07**	39	183	2.73
9sym	9/1	370	1750	0.41	**133**	**971**	**0.20**	152	1137	0.18
sym10	10/1	767	4331	1.09	**193**	**1656**	**8.12**	276	2307	0.84
sao2	10/4	50	385	0.33	43	357	7.52	41	340	8.54
alu1	12/8	23	56	57.14	23	56	56.08	23	56	57.14
Total		1521	7670	63.89	683	4183	83.35	767	4909	71.96

Table 5: Comparison of $Info$ [15] and $Info\!-\!A$

Observation 2. *(i) $Info\!-\!A$ produces the better results in the number of terms and literals against $Info$ for arithmetic functions, such as squar5, con1, z4, 5xp1, root, adr4, log8mod and mlp4.*

(ii) For symmetric functions, such as xor5, f51m, 9sym and sym10, the results by $Info$ are better, in comparison with $Info\!-\!A$.

In the following series of experiments, we consider minimization of arithmetic functions, adders and multiplies by $Info$–A^{BMT}. The results are given in Table 6. In Figure 4, BMTs for add2 and mul2 are presented.

Observation 3. *We have succeeded in minimization of adders and multiplies via design of free BMTs. We have obtained that BMT for n-bit adder is of linear structure with n nodes [13]. We have shown that the arithmetic expression for n-bit adder represented by free BMT has $2n$ terms and $2n$ literals. For n-bit multiplier, the arithmetic expression represented by free BMT has n^2 terms and $2n^2$ literals.*

6 Concluding Remarks

We investigated the applications of information theoretic approach to minimize arithmetic expressions using word-level DTs. As the main contribution, we have introduced *free pseudo Binary Moment Trees* and *free pseudo Kronecker Binary Moment Trees* and

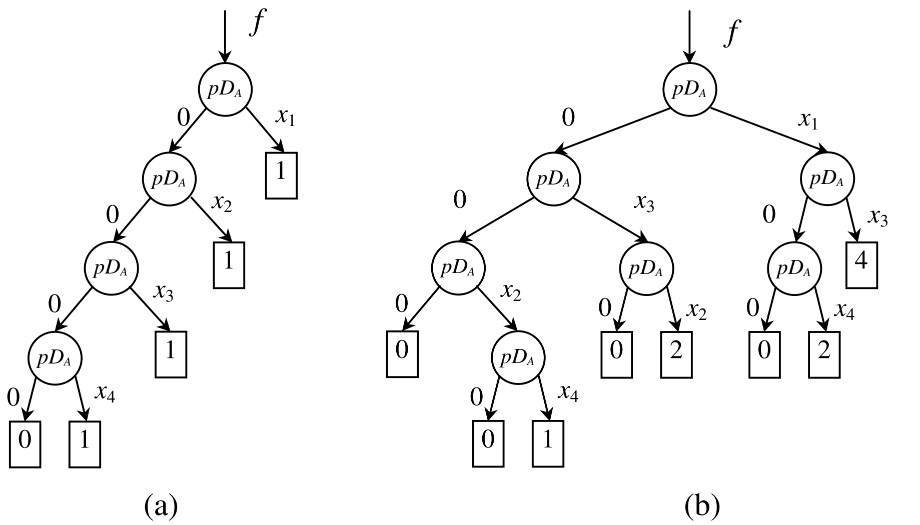


Figure 4: Free BMTs designed by $Info{-}A$ for 2-bit adder and 2-bit multiplier

	I/O	Arithmetic expression	T/L
add2	4/3	$(2x_1 + x_2) + (2x_3 + x_4)$	4/4
add3	6/4	$(4x_1 + 2x_2 + x_3) + (4x_4 + 2x_5 + x_6)$	6/6
add4	8/5	$(8x_1 + 4x_2 + 2x_3 + x_4) + (8x_5 + 4x_6 + 2x_7 + x_8)$	8/8
add5	10/6	$(16x_1 + 8x_2 + 4x_3 + 2x_4 + x_5) + (16x_6 + 8x_7 + 4x_8 + 2x_9 + x_{10})$	10/10
...			
add$n$	$2n/n+1$	$\sum_i^n (2^{n-i}x_i + 2^{n-i}x_{i+n})$	$2n/2n$
mul2	4/4	$2x_1(2x_3 + x_2) + x_2(2x_3 + x_4)$	4/8
mul3	6/6	$4x_1(4x_4+2x_5+x_4)+4x_4(2x_2+x_3)+2x_2(2x_5+x_6)+x_3(2x_5+x_6)$	9/18
mul4	8/8	$8x_1(8x_5 + 4x_6 + 2x_7 + x_8) + 8x_5(4x_2 + 2x_3 + x_4) + 4x_2(4x_6 + 2x_7 + x_8) + 4x_6(2x_3 + x_4) + 2x_3(2x_7 + x_8) + x_4(2x_7 + x_8)$	16/32
mul5	10/10	$16x_1(16x_6+8x_7+4x_8+2x_9+x_{10})+16x_6(8x_2+4x_3+2x_4+x_5)+ 8x_2(8x_7 + 4x_8 + 2x_9 + x_{10}) + 8x_7(4x_3 + 2x_4 + x_5) + 4x_3(4x_8 + 2x_9 + x_{10}) + 4x_8(2x_4 + x_5) + 2x_4(2x_9 + x_{10}) + x_5(2x_9 + x_{10})$	25/50
...			
mul$n$	$2n/2n$	$\sum_{i=1}^n x_i \sum_{j=1}^{n-i+1} 2^{2n-2j}x_{j+n} + \sum_{i=1}^n x_{i+n} \sum_{j=1}^{n-i} 2^{2n-j-i}x_{j+i}$	$n^2/2n^2$

Table 6: Arithmetic expressions for adders and multipliers obtained via free BMTs design by $Info - A$

extended our information theoretic model: the information theoretic measures for word-level counterpart of Shannon and Davio expansions have been proposed. The experiments, conducted with our programs $Info{-}A$, showed that the obtained arithmetic expressions contain about 70% fewer terms and about 60% fewer literals, in comparison with the expressions obtained using BMTs.

We have tested our program with adders and multipliers and proved, that the number of terms and literals in arithmetic expressions represented by BMT for n-bit adders, are $2n$ and $2n$ correspondingly. For n-bit multiplier, the arithmetic expressions have n^2 terms and $2n^2$ literals.

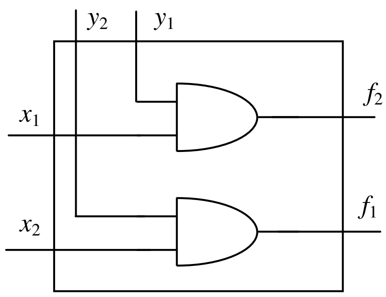
The proposed information theoretic approach can be easily extended to minimize arithmetic expressions via word-level Ternary DTs design [12].

Current and Future Work

In our current work, we are focusing on linear arithmetic representation of circuits: linear word-level DTs and arithmetic expressions. We found that a switching function f linearly depends on variable x if and only if Equation (4) turns to the equality:

$$H^{S_A} = H^{pD_A} + H^{nD_A}.$$

We apply a special operator [9], in order to indicate initial binary representation of multi-output switching function, whose arithmetic expression is linear. For instance, the linear expressions for some elementary switching functions of two variables are presented in Table 7. Also, for simple circuits such as (2+2)-input 2-output control circuit (Figure 5), their linear expression is equal to $f = 2f_2 + f_1$.



$$f = 2f_2 + f_1 = x_1 + y_1 + 2(x_2 + y_2)$$

Figure 5: (2+2)-input 2-output control circuit and their linear arithmetic expression

Function	Linear expression	Function	Linear expression
$x_1 \wedge x_2$	$x_1 + x_2$	$x_1 \rightarrow x_2$	$2 + x_1 - x_2$
$x_1 \vee x_2$	$1 + x_1 + x_2$	$x_1 \uparrow x_2$	$2 - x_1 - x_2$
$x_1 \sim x_2$	$1 + x_1 - x_2$	$x_1 \| x_2$	$3 - x_1 - x_2$

Table 7: Linear arithmetic expressions of some elementary switching functions

References

[1] R. Bryant and Y. Chen. Verification of arithmetic functions using binary moment diagrams. In *Proc. Design Automation Conf.*, pages 535–541, 1995.

[2] R. Drechsler, B. Becker, and S. Ruppertz. Manipulation algorithms for K*BMDs. In *Proc. Tools and Algorithms for the Construction and Analysys of Systems*, pages 4–18, 1997.

[3] G. Dueck, G. Holowinski, K. Malecki, V. Shmerko, and S. Yanushkevich. Development of Zakrevskij's minimization strategy towards arithmetical polynomial domain. In *Proc. Int. Workshop on Boolean Problems, Freiberg, Germany*, pages 101–108, 1998.

[4] R. Enders. Note on the complexity of binary moment diagram representation. In *Proc. IFIP WG 10.5 Workshop on Application of the Reed-Muller Expansion in Circuit Design*, pages 191–197, 1995.

[5] P. Ho and M. Perkowski. Free Kronecker decision diagrams and their application to ATMEL 6000 FPGA mapping. In *Proc. Euro-DAC'94 with Euro-VHDL'94, Grenoble, France*, pages 19–23, 1994.

[6] J. Jain. *Arithmetic transform of Boolean functions*, chapter 6, pages 133–161. In [13], 1996.

[7] A. Kabakcioglu, P. Varshney, and C. Hartman. Application of information theory to switching function minimization. *IEE Proc., Pt.E.*, 137:389–393, 1990.

[8] Y. Lai, M. Pedram, and S. Vrudhula. EVBDD-based algorithms for integer linear programming, spectral transformation, and function decomposition. *IEEE Trans. on CAD*, 13(8):959–975, 1994.

[9] V. Malyugin. Representation of Boolean functions by arithmetical polynomials. *Automation and Remote Control (USA)*, 43:496–504, 4, Pt. 1 1982.

[10] S. Papaioannou. Optimal test generation in combinational networks by pseudo-Boolean programming. *IEEE Trans. on Computers*, C-26(6):553–560, 1977.

[11] D. Popel. *Information theoretic approach to logic functions minimization.* PhD thesis, Technical University of Szczecin, Faculty of Computer Science and Information Systems, 2000.

[12] T. Sasao. Arithmetic ternary decision diagrams and their applications. In *Proc. IFIP WG 10.5 Workshop on Application of the Reed-Muller Expansion in Circuit Design*, pages 149–155, 1999.

[13] T. Sasao and M. Fujita (Eds.). *Representations of discrete functions.* Kluwer Academic Publishers, 1996.

[14] V. Shmerko, G. Holowinski, N. Song, K. Ganguly K. Dill, R. Safranek, and M. Perkowski. High quality minimization of multiple-valued input binary output EXOR sum of products expressions for strongly unspecified multi-output functions. In *Proc. Int. Conf. on Applications of Computer Systems - ACS'97, Poland*, pages 266–286, 1997.

[15] V. Shmerko, D. Popel, R. Stanković, V. Cheushev, and S. Yanushkevich. Information theoretical approach to minimization of AND/EXOR expressions of switching functions. In *Proc. IEEE Int. Conf. on Telecommunications in Modern Satellite, Cable and Broadcasting Services - TELSIKS'99*, pages 444–451 (invited paper), 1999.