Former and Recent Work in Classification of Switching Functions

Radomir S. Stanković, Jaakko T. Astola, Bernd Steinbach

Dept. of Computer Science, Faculty of Electronics, Niš, Serbia

Tampere International Center for Signal Processing
Department of Signal Processing
Tampere University of Technology, Tampere, Finland

Institute of Computer Science
Freiberg University of Mining and Technology
Freiberg, Germany

Outline of the Talk

About classification of switching functions

What does it mean?

Why ít important?

Different classifications

NPN-classification

NPN in terms of Reed-Muller

Classification by Reed-Muller expressions

SD-classification

LP-classification

Classification by

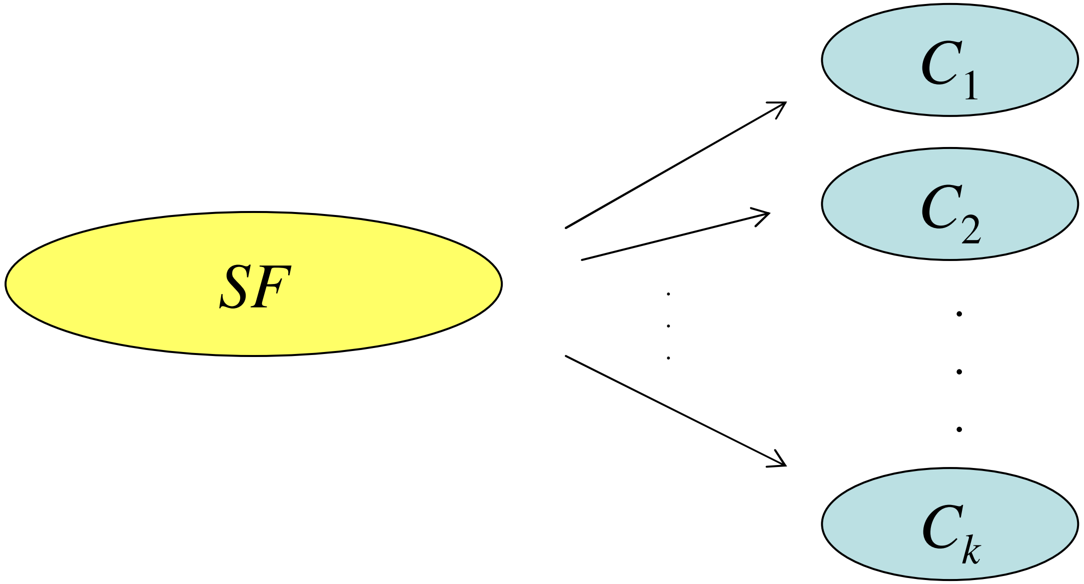
Walsh coefficients

Autocorrelation functions

Classification of Switching Functions

$$SF = \{Switching\ \ functions \mid n\}$$

$$\mid SF \mid = 2^{2^n}$$



$\{f\}$ similar with c_1

$\{f\}$ similar with c_2

$\{f\}$ similar with c_k

Classification – Why?

Identification of functionaly equivalent logic
in cell library binding

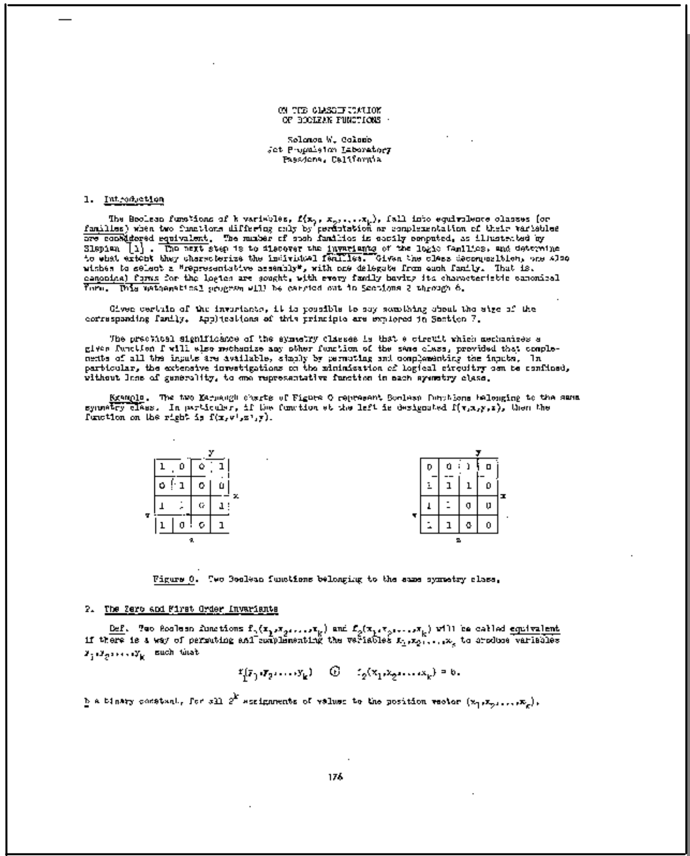
Realization by prototypes

Similar circuits for functions withing the same class

Univerzal logic modules

Standardization of methods for testing

Early Works



ON THE CLASSIFICATION
OF BOOLEAN FUNCTIONS

Solomon W. Golomb
Jet Propulsion Laboratory
Pasadena, California

1. Introduction

The Boolean functions of k variables, $f(x_1, x_2, \ldots, x_k)$, fall into equivalence classes (or families) when two functions differing only by permutation or complementation of their variables are considered equivalent. The number of such families is easily computed, as illustrated by Slepian [1]. The next step is to discover the invariants of the logic families, and determine to what extent they characterize the individual families. Given the class decomposition, one also wishes to select a "representative assembly", with one delegate from each family. That is, canonical forms for the logics are sought, with every family having its characteristic canonical form. This mathematical program will be carried out in Sections 2 through 6.

Given certain of the invariants, it is possible to say something about the size of the corresponding family. Applications of this principle are explored in Section 7.

The practical significance of the symmetry classes is that a circuit which mechanizes a given function f will also mechanize any other function of the same class, provided that complements of all the inputs are available, simply by permuting and complementing the inputs. In particular, the extensive investigations on the minimization of logical circuitry can be confined, without loss of generality, to one representative function in each symmetry class.

Example. The two Karnaugh charts of Figure 0 represent Boolean functions belonging to the same symmetry class. In particular, if the function at the left is designated $f(v,x,y,z)$, then the function on the right is $f(x,v',z',y)$.

1	0	0	1
0	1	0	0
1	[illegible]	0	1
1	0	0	1

0	0	1	0
1	1	1	0
1	1	0	0
1	1	0	0

Figure 0. Two Boolean functions belonging to the same symmetry class.

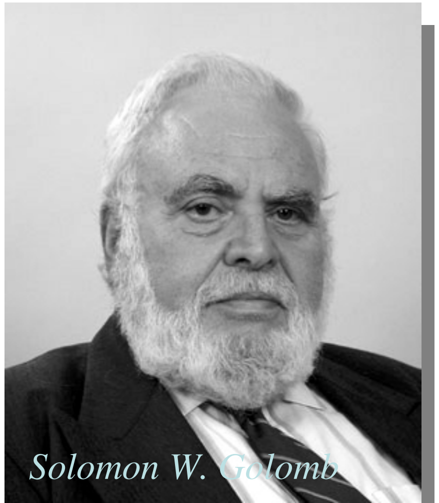
2. The Zero and First Order Invariants

Def. Two Boolean functions $f_1(x_1,x_2,\ldots,x_k)$ and $f_2(x_1,x_2,\ldots,x_k)$ will be called equivalent if there is a way of permuting and complementing the variables $x_1,x_2,\ldots,x_k$ to produce variables $y_1,y_2,\ldots,y_k$ such that

$$f_1(y_1,y_2,\ldots,y_k) \oplus f_2(x_1,x_2,\ldots,x_k) = b,$$

b a binary constant, for all 2^k assignments of values to the position vector $(x_1,x_2,\ldots,x_k)$.

176



NPN

Opeartions for classification *Class*

1. Negation of input variables (N)

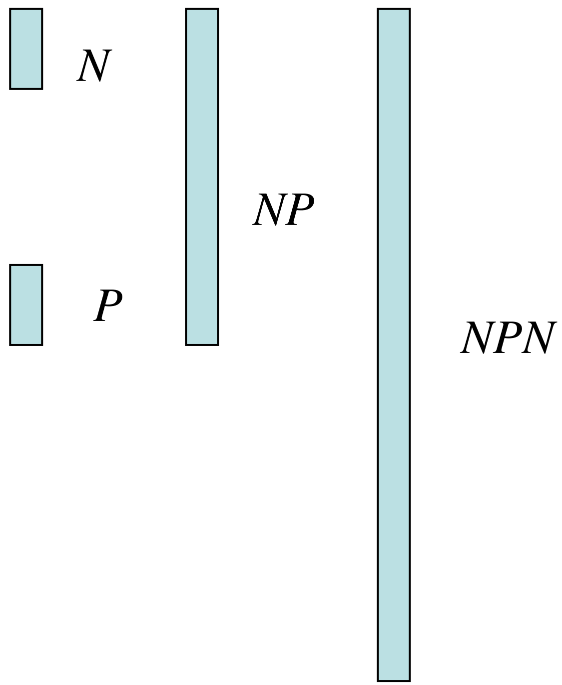
$$x_i \rightarrow \bar{x}_i$$

2. Permutation of input variables (P)

$$x_i \leftrightarrow x_j$$

3. Negation of function values (N)

$$f \rightarrow \bar{f}$$

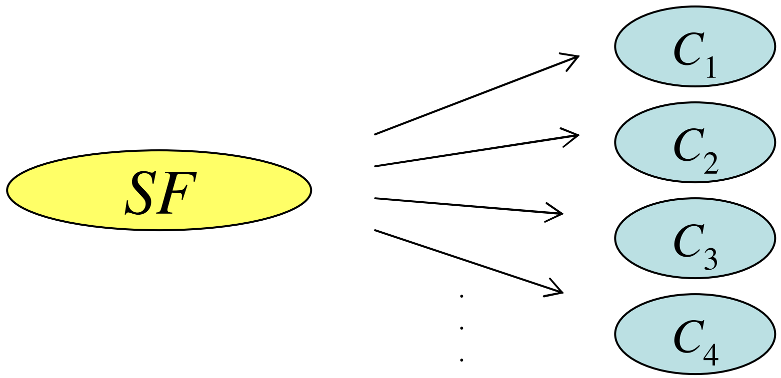


Representative Functions

Canonic order of variables $x_1, \ldots, x_n$

Positive literals, except for P-class

P		*NP*		*NPN*	
$f = 0$	(1)	$f = 0$	(1)	$f = 1$	(2)
$f = 1$	(1)	$f = 1$	(1)	$f = x_1$	(4)
$f = x_1$	(2)	$f = x_1$	(4)	$f = x_1 + x_2$	(8)
$f = \bar{x}_1$	(2)	$f = x_1 x_2$	(4)	$f = x_1 \oplus x_2$	(2)
$f = x_1 x_2$	(1)	$f = x_1 + x_2$	(4)		4
$f = x_1 \bar{x}_2$	(2)	$f = x_1 \oplus x_2$	(2)		
$f = \bar{x}_1 \bar{x}_2$	(1)		6		
$f = x_1 + x_2$	(1)				
$f = x_1 + \bar{x}_2$	(2)				
$f = \bar{x}_1 + \bar{x}_2$	(1)				
$f = x_1 \oplus x_2$	(1)				
$f = x_1 \oplus \bar{x}_2$	(1)				
	12				



Number of Equivalence Classes

	n					
	1	2	3	4	5	6
$\#f$	4	16	256	65536	$4,3\times10^{9}$	1.8×10^{19}
$\#f(n)$	2	10	218	64594	4.3×10^{9}	1.8×10^{19}
$\|C_P\|$	4	12	80	3984	3.7×10^{7}	-
$\|C_{NP}\|$	3	6	22	402	1228158	4.0×10^{14}
$\|C_{NPN}\|$	2	4	14	222	616126	2.0×10^{14}

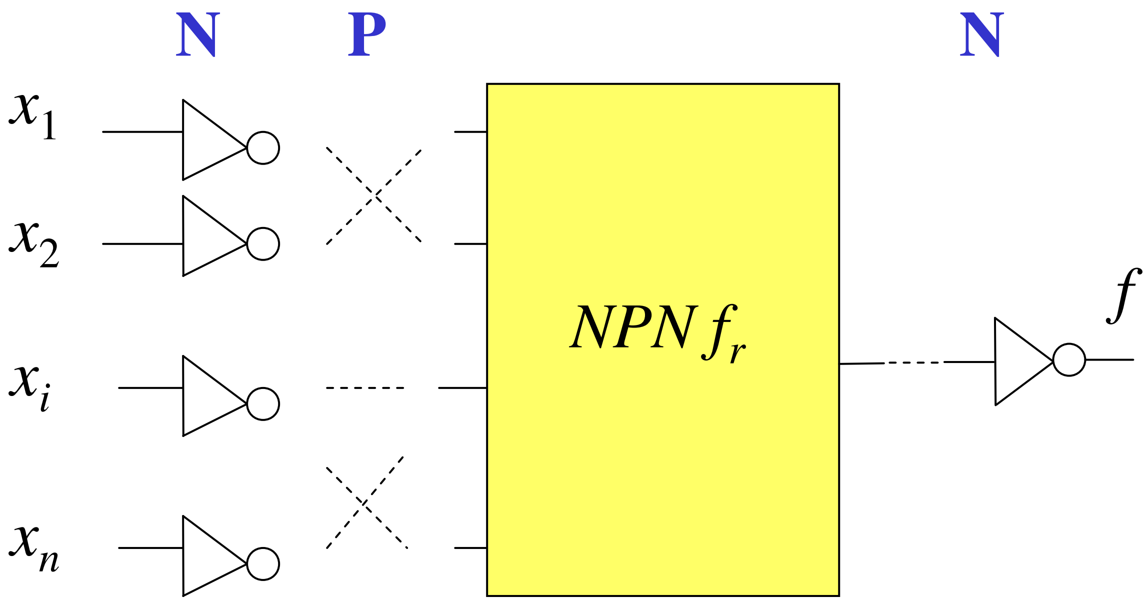
$\#f$ – number of functions of n variables

$\#f\,(n)$ – number of functions of all n variables

$|C_i|$ - number of classes

	P	N	NPN
n sufficiently large	$\frac{2^{2^n}}{n!}$	$\frac{2^{2^n}}{2^n n!}$	$\frac{2^{2^n}}{2^{n+1} n!}$

Realization through $NPN\ f_r$



f_r – *NPN representant*

1. Negation of inputs
2. Permutation of inputs
3. Negation of output

NPN

SD-Classification

$$f(x_1,\ldots,x_n,x_{n+1}) = f^d(x_1,\ldots,x_n,x_{n+1})$$

$$f^{sd}(x_1,\ldots,x_n,x_{n+1}) = x_{n+1} f(x_1,\ldots,x_n) + \bar{x}_{n+1} f^d(x_1,\ldots,x_n)$$

$$f, f^d \in |C_{NPN}|$$

f^{sd} – representant for all *NPN*-equivalent functions determined by the decomposition of f^{sd} with respect to each of its variables.

Morinochi Goto

SD-equivalent Functions (Example)

$$f_1(x_1x_2x_3) = x_1 + x_2 + x_3 \qquad NPN \qquad SD$$

$$f_2(x_1x_2x_3) = x_1(x_2 + x_3) \qquad \not\equiv \qquad \equiv$$

$$f_1^{sd}(x_1,x_2,x_3,x_4) = (x_1 + x_2 + x_3)x_4 + \overline{(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)}\bar{x}_4$$

$$= (x_1 + x_2 + x_3)x_4 + x_1x_2x_3$$

$$f_2^{sd}(x_1,x_2,x_3,x_4) = x_1(x_2 + x_3)x_4 + \overline{\bar{x}_1(\bar{x}_2 + \bar{x}_3)}\bar{x}_4$$

$$= x_1x_2x_4 + x_1x_3x_4 + (x_1 + x_2x_3)\bar{x}_4$$

$$= x_1x_2x_4 + x_1x_3x_4 + x_1\bar{x}_4 + x_2x_3\bar{x}_4$$

$$= (x_2 + x_3 + \bar{x}_4)x_1 + x_2x_3\bar{x}_4$$

$$\bar{x}_4 \leftrightarrow x_4 \qquad x_1 \leftrightarrow x_4$$

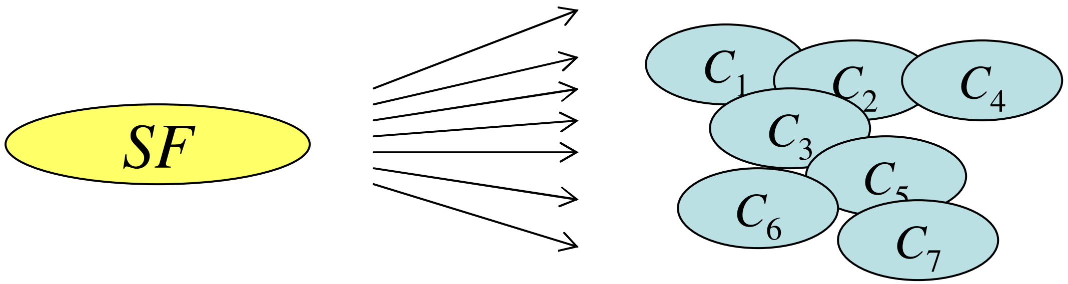
$$f_2^{sd}(x_1,x_2,x_3,x_4) = f_1^{sd}(x_1,x_2,x_3,x_4)$$

SD – classes

$n = 3$

1. x_1 (4)
2. $x_1x_2 + x_2x_3 + x_1x_3$ (32)
3. $x_1 \oplus x_2 \oplus x_3$ (8)
4. $(x_1 + x_2 + x_3)x_4 + x_1x_2x_3\bar{x}_4$ (128)
5. $(x_1x_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_3)x_4 + (x_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)\bar{x}_4$ (64)
6. $x_1(x_2x_3 + \bar{x}_2\bar{x}_3)x_4 + (x_1 + x_2\bar{x}_3 + \bar{x}_2x_3)\bar{x}_4$ (96)
7. $(\bar{x}_1x_2x_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3)x_4 + (x_1x_2 + x_2x_3 + x_1x_3 + \bar{x}_1\bar{x}_2\bar{x}_3)\bar{x}_4$ (128) 7

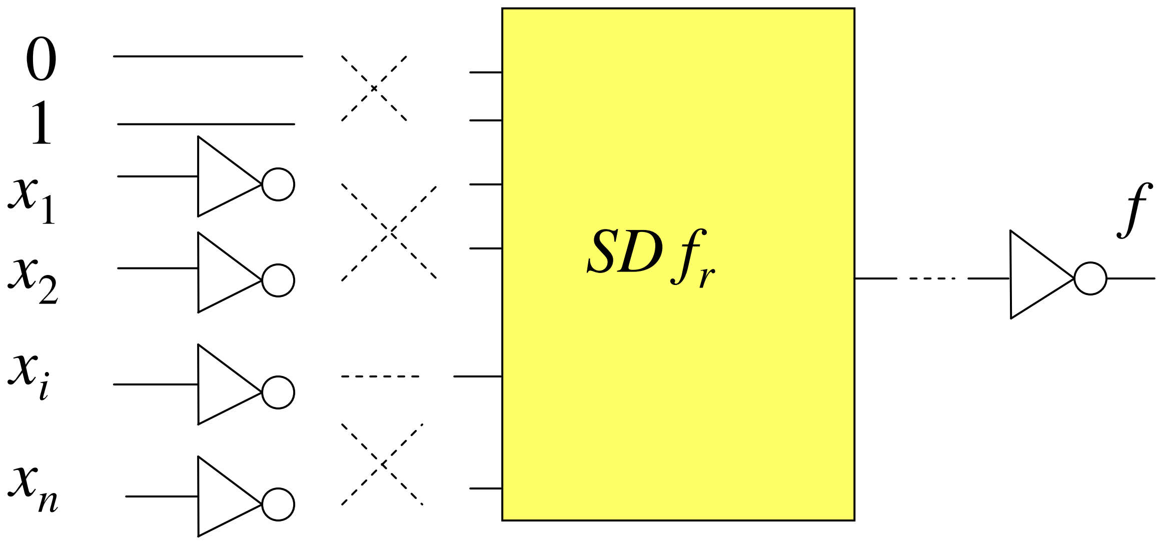
256



$\sum \neq 2^{2^n}$

Non-disjoint

Realization through $SD\ f_r$



f_r – *SD representant*

1. Selective application of 0 or 1
2. Negation of inputs
3. Permutation of inputs
4. Negation of output

LP-Classification

1.Transform over variables and logic constants

2. Permutation of variables

Addapted to AND-EXOR expressions

$n = 4$	t	AND-OR	AND-EXOR
	0	1	1
	1	81	81
	2	1804	2268
	3	13472	21744
	4	28904	37530
	5	17032	3888
	6	3704	24
	7	512	0
	8	26	0
av.		4.19	3.66

LP-transform of Variables

$$f = \bar{x}_i f_0 \oplus x_i f_1$$

$$g = \begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_0 \oplus f_1 \end{bmatrix}$$

$$g_0 = f_0$$

$$g_1 = f_0 \oplus f_1$$

$$g = \bar{x}_i g_0 \oplus x_i g_1 = \bar{x}_i f_0 \oplus x_i (f_0 \oplus f_1) = 1 \cdot f_0 \oplus x_i f_1$$

$\bar{x}_i \to 1$ $\quad f = \bar{x}_i f_0 \oplus x_i f_1 \longrightarrow g = 1 \cdot f_0 \oplus x_i f_1$

$$\begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$$

$\mathbf{M}$ – (2×2) non-singular matrix over $GF(2)$

6 different $\mathbf{M}$

LP-transforms

1. $LP_0(f) = \bar{x}f_0 \oplus xf_1$ Identical mapping
2. $LP_1(f) = \bar{x}f_0 \oplus 1 \cdot f_1$ $x_i \leftrightarrow 1$
3. $LP_2(f) = 1 \cdot f_0 \oplus xf_1$ $\bar{x}_i \leftrightarrow 1$
4. $LP_3(f) = xf_0 \oplus \bar{x}f_1$ $x_i \leftrightarrow \bar{x}_i$
5. $LP_4(f) = xf_0 \oplus 1 \cdot f_1$ $x_i \to 1 \to \bar{x}_i \to x_i$
6. $LP_5(f) = 1 \cdot f_0 \oplus \bar{x}f_1$ $\bar{x}_i \to 1 \to x_i \to \bar{x}_i$

$| f | = | g |$

number of products

$f(x, y) = x \oplus y$ $y \to \bar{y}$ $f(x, y) = x \oplus \bar{y}$

$f(x, y) = x \oplus \bar{y}$ $x \to 1$ $g = 1 \oplus xy$

LP-equivalent

$x \oplus y$ $x \cdot 1 \oplus 1 \cdot \bar{y}$ $1 \oplus x\bar{y}$ $\bar{x} \cdot 1 \oplus 1 \cdot \bar{y}$ $\bar{x} \oplus \bar{y}$

$y \to \bar{y}$ $x \to 1$ $1 \to \bar{x}$ $x \to 1$

LP-equivalence

LP-equivalence relation

1. $f \equiv f$
2. If $f_1 \equiv f(x_i, x_j)$ and $f_2 \equiv f(x_j, x_i)$ then $f_1 \equiv f_2$
3. F ESOP for $f(x_1, \ldots, x_n)$

   G ESOP derived by a LP-transform of F

   g a function represented by G Then $f \equiv g$

$\bar{x}y \equiv xy$ $\quad \bar{x} \leftrightarrow x$ $\quad x \cdot 1 \equiv 1 \cdot y$ $\quad x \leftrightarrow y$

$xy \equiv x \cdot 1$ $\quad y \leftrightarrow 1$ $\quad 1 \cdot y \equiv 1 \cdot 1$ $\quad y \leftrightarrow 1$

LP-equivalent Functions

Class	f
1	0
2	$\bar{x}\,\bar{y}$ $\bar{x}\,y$ $x\,\bar{y}$ $x\,y$ $\bar{x}$ 1 x $\bar{y}$ y
3	$x \oplus \bar{y}$ $x \oplus y$ $1 \oplus \bar{x}\,\bar{y}$ $1 \oplus \bar{x}\,y$ $1 \oplus x\,\bar{y}$ $1 \oplus x\,y$

$n = 2$

$0 \quad \bar{x}\bar{y}\bar{z} \quad \bar{x}\,y \oplus \bar{x}\,z$

$\bar{x} \oplus \bar{y}z \oplus \bar{x}yz \quad x\bar{y}\bar{z} \oplus \bar{x}yz \quad \bar{x} \oplus y\bar{z} \oplus x\bar{y}z$

$n = 3$

LP-representative Functions

n	2	3	4		
$LP(n)$	0	00	0000	016a	0678
	1	01	0001	0180	06b0
	6	06	0006	0182	06b1
		16	0016	0186	1668
		18	0018	0196	1669
		6b	0066	0660	1681
			0116	0661	1683
			0118	0662	168b
			012c	066b	18ef
			0168	0672	6bbd

Number of LP Equivalence Classes

	n					
	1	2	3	4	5	6
$\#f$	4	16	256	65536	$4,3\times10^{9}$	1.8×10^{19}
$\#f(n)$	2	10	218	64594	4.3×10^{9}	1.8×10^{19}
$\|C_P\|$	4	12	80	3984	3.7×10^{7}	-
$\|C_{NP}\|$	3	6	22	402	1228158	4.0×10^{14}
$\|C_{NPN}\|$	2	4	14	222	616126	2.0×10^{14}
$\|C_{LP}\|$	2	3	6	30	6936	$>5.5\times10^{11}$

$\#f$ – number of functions of n variables

$\#f\,(n)$ – number of functions of all n variables

$|C_i|$ - number of classes

n sufficiently large

P	N	NPN	LP
$\frac{2^{2^n}}{n!}$	$\frac{2^{2^n}}{2^n n!}$	$\frac{2^{2^n}}{2^{n+1} n!}$	$n!6^n$

Classification in terms of RM-expressions

8 classes *NEU*, *ODD*, *EVEN*, *DEG*, *SD*, *LV*, *SC*, *NSC*

1. A switching function f is called neutral if $| f | = | f |$.
   Neutral switching functions form the class *NEU*.

2. A switching function f is called odd (even) if $| f |$an odd (even) integer.
   Such functions form classes *ODD* and *EVEN*.

3. A switching function f is called degenerated if there exists a variable x_i such that $f(x_i) = f(x_i)$, i.e., x_i is not an essential variable for f. Such functions form the class *DEG*.

4. A switching function f is a self-dual function if $f(x_1,. . . ,x_n) = f(x_1,. . . , x_n)$.
   Self-dual functions form the *SD*-class.

5. A variable x_i in a switching function f is called linear if $f(x_i) = f(x_i)$. The set of functions that contain at least one linear variable form the *LV*-class.

6. A function f is self-complementary is both f and its complement f belong to the same *NP*-class. All self-complementary functions for a given number of variables n form the *SC*-class.

7. The *NSC* class is defined as the set difference of the *NEU*-class and the *SC*-class.

Motivated by Technology Mapping

In mask and field programmable technologies,
the minimization of inverters is a problem

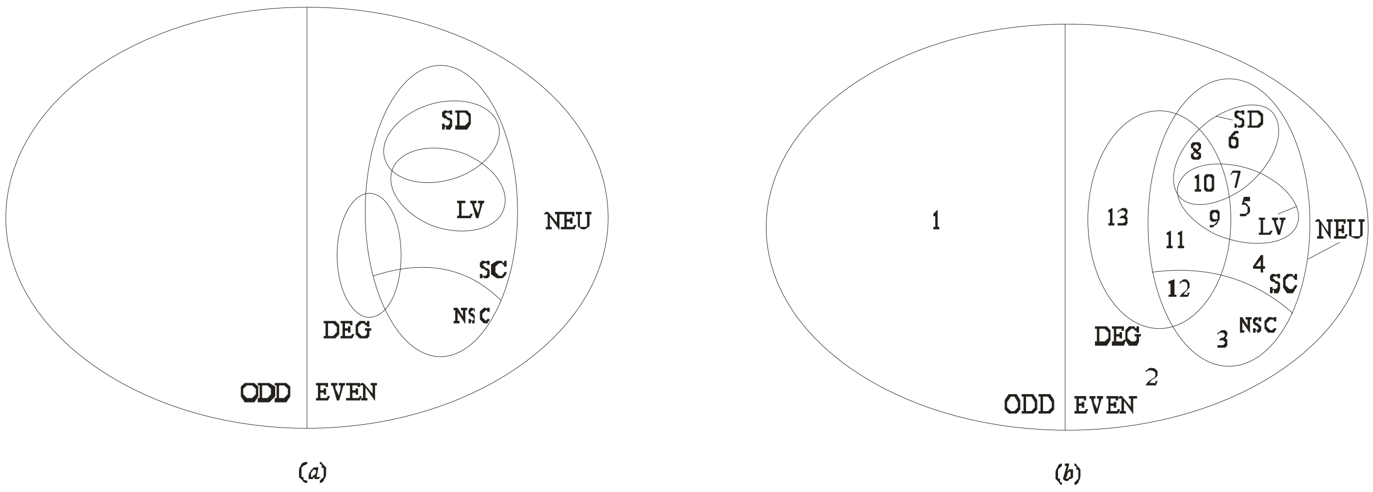
Library cells with linear variables, or cells that are capable of realizing self-complementary functions are useful in reducing the number of inverters

Some of these classes can be specified using derivative operations of the Boolean Differential Calculus

Representative of the classes specified in terms of
Fixed-polarity Reed-Muller expressions

Corrections

Classes (SDT ∩ DEG)\LV, (LV ∩ DEG)\SD and DEG∩SD∩LV are non-empty sets



Distribution of functions for $n = 4$ into 13 classes.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13
# of f	32768	19208	5856	5768	786	176	32	32	108	16	96	0	690

Class 12 empty for $n = 4$ but not for $n > 4$

Classification by Walsh Coefficients

Translation invariant spectral operations

1. Permutation of variables x_i by x_k, $i \neq k \neq 0$
2. Replacement of a variable by its complement $x_i \rightarrow x_i'$
3. Replacement of a function f by its complement f'
4. Replacement of a variable x_i by a linear combination of variables $x_i \oplus x_k$, or in more general case, replacement of x_i by $(x_a \oplus x_b \oplus \ldots \oplus x_h) \oplus x_i$, which is called *spectral translation*
5. Replacement of a function $f(x_1, \ldots, x_n)$ by the function $x_i \oplus f'(x_1, \ldots, x_n)$ where f' is uniquely defined by a particular transformation over indices of Walsh spectral coefficients for f.
   This operation is called the *disjoint spectral translation*

Classification by Autocorrelation

$$B_f(\tau) = \sum_{i=0}^{2^n-1} f(x) f(x \oplus \tau)$$

NPN-classification rules $\cup$ two new rules

1. $x_i \to x_i \oplus x_j \quad i, j \in \{1, \ldots, n\}, \quad i \neq j$
2. $f \to f \oplus x_i \quad i = 1, \ldots, n$ $f \to f \oplus x_i \quad i = 1, \ldots, n$

When encoding $\{0,1\} \to \{1,-1\}$
then $B_f(\tau)$ invariant to the classification operations

18 equivalence classes for $n \leq 4$ compared to 222 in *NPN*

Decision Diagrams in Classification

BDD are used to check is two functions belong to the same autocorrelation class

Features are not affected by the autoccorelation classification rules

1. *# of true minterms*
2. *# of essential variables*
3. *Shortest path length in the BDD*

Functions that belong to the same autoccorelation class have identical these parameters

BDD in *NPN*-classification

If $n = 3$, then 14 representative functions in *NPN*-classifictaton

10 functions essentially depend on all three varaibles

Super BDD is a combination of BDDs for these representative functions

When equipped with *selection switches*, *Super BDD* can realize all these functions

FDD in *NPN*-classification

If $n = 4$, then 222 representative functions in *NPN*-classification

208 essentially depend on all the variables

Super BDD for $n = 4$

Problems

# of bits to encode functions in Super BDD is smaller than
# of bits to encode these functions in synthesis by LUTs

The realization of the corresponding modules
may be *larger* and *slower* than that of LUTs

Alternative

FDD may produce simpler *Super Decision Diagram*

Closing Remarks

Classification – an old subject of continuous research interst

Classification methods differs by	
Domain	Boolean Spectral
Tools	Boolean difference Autocorrelation functions Reed-Muller expressions Walsh coefficients

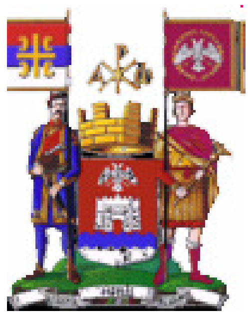
Research interests

Reduction of # of classes

Efficient classification procedures

Acknowledgments

Class
Coat of arms



Class
photos