# FIRST THOUGHTS ON DETERMINING A METHOD FOR FAST AUTOCORRELATION CLASSIFICATION

*J. E. Rice*

Dept. of Math & Computer Science
University of Lethbridge, Alberta, Canada
j.rice@uleth.ca

**ABSTRACT**

Classification of Boolean functions is a useful tool; it greatly reduces the $2^{2^n}$ Boolean functions to a much more feasible number. Recent work has suggested the use of the autocorrelation transform to generate coefficients that may be used as a classification tool for Boolean functions. An outstanding question, however, is how to quickly identify whether or not two given functions are in the same class, given that generating the $2^n$-sized spectrum may not always be feasible. This work addresses this question, and presents some preliminary heuristics that are analyzed for their suitability in solving this problem.

## 1. INTRODUCTION

It is a well-known problem that for even relatively small values of $n$ there are far too many Boolean logic functions to enumerate or analyze in any useful way. Thus classification techniques and identification of useful properties within classes of functions are useful methods for reducing unmanageable numbers of functions to reasonable sizes [1, 2].

A new technique for classifying Boolean functions was introduced in [3]. This technique is based on the autocorrelation spectra. A major problem, however, is that of quickly identifying which class a function belongs to. As $n$ grows larger it becomes infeasible to quickly compute the autocorrelation spectra, as there are $2^n$ coefficients for each function with $n$ inputs. Thus we must identify some fast process for matching a function to another without necessarily computing the entire spectra. This is an area where much previous work has been done, as in [4] and other publications. We re-address the issue for our specific application of quickly classifying a function; that is, for fast determination of whether a given function belongs to the same autocorrelation class as another.

## 2. BACKGROUND

The following provides some background and notation that is pertinent to the remainder of the paper.

### 2.1. The Autocorrelation Transform

The autocorrelation function is defined as follows [5]:

$$B^{\,ff}(\tau) = \sum_{v=0}^{2^n-1} f(v) \cdot f(v \oplus \tau) \tag{1}$$

The superscripts $ff$ are generally omitted. Values for $\tau$ range from $0$ to $2^n - 1$ where $n$ is the number of inputs to the Boolean function $f(X)$. The autocorrelation function or transform, when applied to the outputs of $f(X)$, transforms the outputs from a two-valued domain to the domain of the real numbers. The resulting coefficients may be referred to as the autocorrelation spectra of the function.

The outputs of the function may be encoded as $0$ for false and $1$ for true, or $+1$ for false and $-1$ for true. The first is referred to as $\{0, 1\}$ encoding, and if used results in the autocorrelation coefficients being referred to as $B(\tau)$. The $\{+1, -1\}$ coefficients are referred to as $C(\tau)$.

### 2.2. The Autocorrelation Classes

Work in [6] demonstrated that there are four invariant operations for the $\{+1, -1\}$ autocorrelation coefficients. These invariant operations are as follows:

(i) permutation of any input variables $x_i$ and $x_j$, $i, j \in 1..n$, $i \neq j$,
(ii) negation of any input variable $x_i$, $i \in 1..n$,
(iii) negation of the output of the switching function, and
(iv) replacement of any input variable $x_i$ with $x_i \oplus x_j$, $i, j \in 1..n$, $i \neq j$.

Application of these invariant operations leads to a classification scheme in which certain properties become apparent [3, 7, 8]. Table 1 lists the autocorrelation classes for $n \leq 4$.

### 2.3. Decision Diagrams

Due to their efficiency at representing Boolean functions and the speed in which various operations can be carried

| $k/s/p$ | **spectra** | **sop** |
|---|---|---|
| 0/0/0 | 16x16 | $f_0(X) = 0$ |
| 8/1/1 | 16x8 -16x8 | $f_{255}(X) = x_4$ |
| 4/2/1 | 16x4 0x12 | $f_{15}(X) = x_3x_4$ |
| 8/3/2 | 16x2 0x12 -16x2 | $f_{831}(X) = x_1x_2 + x_1x_3 + x_2x_3$ |
| 2/3/1 | 16x2 8x14 | $f_3(X) = x_2x_3x_4$ |
| 6/3/2 | 16x2 8x6 -8x8 | $f_{63}(X) = x_3x_4 + x_2x_4$ |
| 7/4/1 | 16x1 12x7 -12x8 | $f_{127}(X) = x_1x_2 + x_1x_3 + x_1x_4$ |
| 1/4/1 | 16x1 12x15 | $f_1(X) = x_1x_2x_3x_4$ |
| 3/4/2 | 16x1 12x3 4x12 | $f_7(X) = x_2x_3x_4 + x_1x_3x_4$ |
| 5/4/2 | 16x1 12x3 4x4 -4x8 | $f_{31}(X) = x_1x_2x_4 + x_3x_4$ |
| 7/4/2 | 16x1 12x1 -12x2 4x6 -4x6 | $f_{319}(X) = x_1x_2x_3 + x_3x_4 + x_2x_4$ |
| 4/4/2 | 16x1 8x6 0x9 | $f_{23}(X) = x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_4$ |
| 8/4/3 | 16x1 -16x1 8x7 -8x7 | $f_{383}(X) = x_1x_2x_3 + x_3x_4 + x_2x_4 + x_1x_4$ |
| 6/4/2 | 16x1 8x3 -8x3 0x9 | $f_{287}(X) = x_3x_4 + x_1x_2x_4 + x_1x_2x_3$ |
| 8/4/2 | 16x1 8x2 -8x4 0x9 | $f_{863}(X) = x_2x_3 + x_1x_4 + x_3x_4$ |
| 5/4/2 | 16x1 4x10 -4x5 | $f_{279}(X) = x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_4 + x_1x_2x_3$ |
| 7/4/2 | 16x1 4x6 -4x9 | $f_{855}(X) = x_2x_3 + x_1x_4$ |
| 6/4/2 | 16x1 0x15 | $f_{854}(X) = \overline{x}_1x_2x_3 + x_1\overline{x}_2x_4 + x_1\overline{x}_3x_4 + x_2x_3\overline{x}_4$ |

**Table 1**. The $n \leq 4$ autocorrelation classes and their underlying functions.

out using this representation, we decided to use decision diagrams as our primary representation for the functions in question. Decision diagrams were first introduced as binary decision diagrams (BDDs) by Lee [9] and later by Akers [10], and were popularized more recently as reduced, ordered BDDs (ROBDDs) by Bryant [11]. Many other types of decision diagrams have since been proposed [12]. The reader is directed to these references for details and implementation for this data structure.

## 3. COMPARISON TECHNIQUE

It seems logical, when attempting to determine whether or not two functions lie in the same class, to eliminate certain factors as quickly as possible. Thus we suggest a step-by-step process, based first on information that is readily apparent and moving from there to information that may require additional computation. The goal is to eliminate the possibility of a match as quickly as we can, and hence with a minimal amount of work.

Some factors that are quickly identifiable when using BDDs are

1. the number of true minterms,
2. the support size, and
3. the shortest path length.

These criteria are listed in Table 1 for representatives of the $n \leq 4$ autocorrelation classes. In this table $k$ refers to the number of true minterms, $s$ refers to the support size and $p$ refers to the shortest path length after minimization of the BDD.

We begin with the computation of $k$, or, the number of true minterms in the function. For the remainder of this work we will refer to the two functions being compared as $f_i$ and $f_j$, and any data related to these functions with appropriate subscripts. If we determine that $k_i = k_j$ or $k_i = 2^n - k_j$ then the functions may be in the same class, and so we must continue on with the process. However, if neither of these equalities hold then we can immediately state that $f_i$ and $f_j$ do not belong to the same autocorrelation class. We can see that in the 18 functions in Table 1 there are at most 4 for which the functions will have the same value of $k = 8$ and yet potentially be in different classes. For $k = 0, 1, 2, 3$ and 5 this first step is sufficient to identify, for $n \leq 4$, which class a function belongs to, and thus indicate whether a function $f_i$ belongs to the same class as any function $f_j$ from one of these classes.

If, as in the case of $k = 8$, the values of $k$ for each function cannot rule out a class match then we continue to the next criteria, that of variable support. For the cases of $k = 4$ and $k = 6$ this is enough to separate the two classes. Indeed, for $k = 8$ although there are still two classes that we have not differentiated between, there are two classes which have been ruled out. However, for the case of $k = 7$ there are still 3 remaining classes between which we have not distinguished.

Finally, we compare the shortest path length in the DD for the functions. For our $n \leq 4$ classes this has the effect of eliminating overlaps between the two classes for which $k = 8$ and $s = 4$. For $k = 7$ and $s = 4$, however, there are still 2 classes with the same length shortest path. Short of generating the autocorrelation spectra, at this point we have not yet identified the distinguishing factors between these two classes.

## 4. DISCUSSION

The selection of criteria for determining whether $f_1$ and $f_2$ belong to the same autocorrelation class is fairly commonsense: we require criteria easy to compute, but that will rule out non-matches as quickly as possible. We also desire criteria that will not be affected by many of the invariant operations, such as, at the very least, variable permutation and negation.

### 4.1. Comparison of $k$

We first prove the following theorem.

**Theorem 1** *If $k_1 \neq k_2$ and $k_1 \neq 2^n - k_2$ then $f_1$ and $f_2$ must belong to different autocorrelation classes.*

**Proof:** The invariant operations for the autocorrelation classes consist of the following:

(i) permutation of any input variables $x_i$ and $x_j$, $i, j \in 1..n$, $i \neq j$,
(ii) negation of any input variable $x_i$, $i \in 1..n$,
(iii) negation of the output of the switching function, and

(iv) replacement of any input variable $x_i$ with $x_i \oplus x_j$, $i, j \in 1..n$, $i \neq j$.

$k$, the total number of true minterms in the function is not changed by any of these operations except for operation (iii). In this case the modified function will have $2^n - k$ true minterms. Thus two functions $f_1$ and $f_2$ with true minterm counts respectively of $k_1$ and $k_2$, $k_1 \neq k_2$ and $k_1 \neq 2^n - k_2$ must belong to different autocorrelation classes, as there is no way to transform functions within a class that affects the counts of the minterms. ■

Given a BDD representation using a product such as CUDD [13] this information is readily available and has a large chance of identifying immediately that the functions cannot belong to the same class.

### 4.2. Comparison of Support Size

The determination of the support, or alternatively, of which variables the function is independent can distinguish between functions for which $n_i \neq n_j$. Differing values of $n$ need not lead to differing values of $k$, and yet may not be known before building the BDD. Hence we generate this information only if it is needed.

If $k_i = k_j$ and $s_i \neq s_j$ then we have determined that the two functions have the same number of true minterms, yet the functions depend on differing numbers of variables. For instance, $f_1 = \overline{x}_1\overline{x}_2x_3 + \overline{x}_0x_1x_2$ has $k_1 = 4$ and $s_1 = 4$. $f_2 = \overline{x}_2\overline{x}_3$ has $k_2 = 4$ and $s_2 = 2$. These two functions are in different autocorrelation classes. However, if we apply invariant operation (iv), replacement of any input variable $x_i$ with $x_i \oplus x_j$, to $f_2$ we can generate $f_3 = \overline{x}_0\overline{x}_2\overline{x}_3 + x_0x_2\overline{x}_3$ which has $k_3 = 4$ and $s_3 = 3$. Although the support size differs between $f_2$ and $f_3$ they are in the same autocorrelation class.

Despite this, we felt this test still to be useful; if we assume that invariant operation (iv) is NOT applied then any two functions for which $s_i \neq s_j$ can be assumed to be in differing classes, as none of the other three invariant operations can affect the support size of the function. It should be noted that we are, in effect, identifying the NPN classes [14]. At the end of the elimination process it will be necessary to apply invariant operation (iv) to combine some of the classes into the smaller set of autocorrelation classes.

The identification of support size is a trivial matter with the use of DD representations, at least for single-output functions to which we are currently restricting our work. Any DD not containing any node with a variable labeling $i$ does not depend on variable $i$, and so $i$ is not in the support of the function. Thus it is a matter of counting the unique node labelings in the DD, which can be done in time linear to the length of the longest path in the DD.

### 4.3. Comparison of Shortest Path Length

The determination of the shortest path length, in this work, was done after sifting was performed. That there was a difference in this criteria, even in these relatively small functions, is indicative that this does indeed reflect the difference in the structures of the functions in question.

It is a much more difficult matter to argue the efficacy of this test. It is dependent on the results of the sifting heuristics, and is reflective of the structure of the function rather than a fixed property such as the number of true minterms or the support size. Indeed, changing the variable ordering may cause the structure of the function to appear to change in a drastic manner. However, it has been our experience that in the autocorrelation classes functions with visibly different structures result in a BDD that is more (or less) compact. For example, we can examine the function $f_{383}$, which has 8 true minterms, support size of 4, and a shortest path length of 3. If we compare this to function $f_{863}$ which also has 8 true minterms, a support size of 4, but a shortest path of 2 it may be possible to see a difference in their structures. The Karnaugh maps for these functions are shown in Figure 1.

a)

| $x_1x_2 \backslash x_3x_4$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 0 | 1 | 1 | 1 |
| 10 | 0 | 1 | 1 | 0 |

b)

| $x_1x_2 \backslash x_3x_4$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 0 | 1 | 1 | 1 |
| 10 | 0 | 1 | 1 | 0 |

**Fig. 1**. a) $f_{383} = x_1x_2x_3 + x_3x_4 + x_2x_4 + x_1x_4$ and b) $f_{863} = x_2x_3 + x_1x_4 + x_3x_4$.

However, a different in structure is not always identified by the shortest path measure. For example, let us examine the functions $f_{279}$ and $f_{31}$ from Table 1. The Karnaugh maps for these functions are shown in Figure 2. Both of

a)

| $x_1x_2 \backslash x_3x_4$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | 0 | 1 | 1 | 1 |
| 10 | 0 | 0 | 1 | 0 |

b)

| $x_1x_2 \backslash x_3x_4$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | 0 | 1 | 1 | 0 |
| 10 | 0 | 0 | 1 | 0 |

**Fig. 2**. a) $f_{279} = x_1x_2x_3 + x_1x_3x_4 + x_1x_2x_4 + x_2x_3x_4$ and b) $f_{31} = x_1x_2x_4 + x_3x_4$.

these functions have 5 true minterms, support size of 4 and a shortest path in the BDD (after sifting) of 2. Yet they are in different autocorrelation classes, and a visual inspection of the Karnaugh maps can identify a significant structural difference in the grouping of the true minterms. We hypothesize that this is related to the functional complexity of the underlying circuit, and so known estimation techniques such as those in [15] may play a role. Work is continuing in this area.

At this point in the research we have examined a very small set of experimental result that seem to indicate that our measure of shortest path length may help distinguish between autocorrelation classes. However, before any proof of the validity of this hypothesis can be determined we must formally describe this structural difference that seems to be present in functions such as $f_{383}$ and $f_{863}$, and relate this difference to the autocorrelation invariant operations.

## 5. CONCLUSIONS AND FUTURE WORK

As we continue to pursue research on the uses and applications of the autocorrelation classification, we have found that it is absolutely essential to have some fast method for determining if a function lies in the same class as another. Computation of the entire spectrum of autocorrelation coefficients is not feasible, when performed on every function, although it may be possible to use this as a last resort, or to use fast hardware methods [16] when absolutely necessary.

This work presents our initial investigations into determining a set of heuristics for fulfilling these requirements; a series of steps that, with the use of BDDs, can be performed quickly and will in most cases distinguish between Boolean functions belonging to distinct autocorrelation classes. Analysis of our first choice of heuristics has highlighted some problems with them; for instance, the support size of two functions in the same class may be different due to the application of invariant operator iv, and the shortest path measure does not at this time have any formally defined reason for determining whether or not two functions belong to the same class. Clearly these are areas in which future work will be concentrated.

Future work in this area will also focus on refining and building upon these heuristics. We intend to perform analysis to determine exactly how many cases may cause these heuristics to fail, as well as generating experimental results.

## 6. REFERENCES


[1] C. Edwards, "The Application of the Rademacher-Walsh Transform to Boolean Function Classification and Threshold Logic Synthesis," *IEEE Trans. on Comp.*, pp. 48–62, Jan. 1975.

[2] C.-C. Tsai and M. Marek-Sadowska, "Boolean Functions Classification via Fixed Polarity Reed-Muller Forms," *IEEE Trans. on Comp.*, pp. 173–186, Feb. 1997.

[3] J. E. Rice and J. C. Muzio, "Use of the Autocorrelation Function in the Classification of Switching Functions," in *Euromicro Symposium on Digital System Design: Architectures, Methods and Tools (DSD)*, 2002, pp. 244–251.

[4] J. Mohnke, P. Molitor, and S. Malik, "Limits of Using Signatures for Permutation Independent Boolean Comparison," in *Proceedings of the Asian and South Pacific Design Automation Conference (ASP-DAC)*, 1995, pp. 459 –464.

[5] M. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices*, John Wiley & Sons, 1976.

[6] J. E. Rice, *Autocorrelation Coefficients in the Representation and Classification of Switching Functions*, Ph.D. thesis, University of Victoria, 2003.

[7] J. E. Rice and R. Jansen, "Symmetrical, Dual and Linear Functions and Their Autocorrelation Coefficients," in *to appear in the Proceedings of the International Workshop on Logic Synthesis (IWLS)*, 2005.

[8] J. E. Rice, "Making a Choice Between FDDs and BDDs," in *to appear in the International Workshop on Logic Synthesis (IWLS)*, 2005.

[9] C. Y. Lee, "Representation of Switching Circuits by Binary Decision Diagrams," *Bell System Technical Journal*, pp. 958–999, 1959.

[10] S. Akers, "Binary Decision Diagrams," *IEEE Trans. on Comp.*, vol. C-27, no. 6, pp. 509–516, June 1978.

[11] R. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. on Comp.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[12] R. Drechsler and D. Sieling, "Binary Decision Diagrams in Theory and Practice," *Int. Journal on Software Tools for Technology Transfer (STTT)*, pp. 112–136, May 2001.

[13] F. Somenzi, "CUDD: Colorado University Decision Diagram Package," version 2.3.0, Department of Electrical and Computer Engineering, University of Colorado at Boulder, *Fabio@Colorado.EDU*.

[14] S. L. Hurst, D. M. Miller, and J. C. Muzio, *Spectral Techniques in Digital Logic*, Academic Press, Inc., Orlando, Florida, 1985.

[15] S. Aborhey, "Functional Complexity Estimation for Large Combinational Circuits," *Computers and Digital Techniques, IEE Proceedings-*, vol. 149, no. 2, pp. 39–45, March 2002.

[16] J. E. Rice, K. Kent, T. Ronda, and Z. Yong, "Configurable Hardware Solutions for Computing Autocorrelation Coefficients: a Case Study," in *Proceedings of the 15th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2005.