

A COMPARISON OF MODIFIED RECONSTRUCTABILITY ANALYSIS AND ASHENHURST-CURTIS DECOMPOSITION OF BOOLEAN FUNCTIONS

Anas N. Al-Rabadi⁽¹⁾, Martin Zwick⁽²⁾, and Marek Perkowski⁽¹⁾

(1) ECE Department (2) Systems Science Department @Portland State University
[alrabadi, mperkows@ece.pdx.edu]⁽¹⁾, [zwick@sysc.pdx.edu]⁽²⁾

KEYWORDS: Reconstructability Analysis, Ashenhurst-Curtis Decomposition, Boolean Functions, NPN-Classification, Log-Functionality Complexity Measure.

ABSTRACT

Modified Reconstructability Analysis (MRA), a novel decomposition technique within the framework of set-theoretic (crisp possibilistic) Reconstructability Analysis, is applied to 3-variable NPN-classified Boolean functions. MRA is superior to conventional Reconstructability Analysis (CRA), i.e. it decomposes more NPN functions. MRA is compared to Ashenhurst-Curtis (AC) decomposition using two different complexity measures: log-functionality, a measure suitable for machine learning, and the count of the total number of two-input gates, a measure suitable for circuit design. MRA is superior to AC using the first of these measures, and is comparable to, but different from AC, using the second.

1 INTRODUCTION

One general methodology for understanding a complex system is to decompose it into less complex sub-systems. Decomposition is used in many situations; for example, in *logic synthesis* (Ashenhurst 1953, Ashenhurst 1956, Ashenhurst 1959, Curtis 1963, Curtis 1962, Files 2000, Grygiel 2000, Jozwiak 1995, Muroga 1979) where the number of inputs to the gates is high and cannot be mapped to a standard library and in *machine learning* where data is noisy or incomplete (Files 2000, Grygiel 2000). The primary criteria for evaluating the quality of the decomposition process are the amount of *information (or loss of information, i.e., error)* existing in the decomposed system and the *complexity* of this decomposed system. The *objective* is obvious: *decompose the complex system (data) into the least-complex most-informative (least-error) model*. Simplicity is desired since, according to the Occam Razor principle, *the simpler the model is, the more powerful it is for generalization*. Least error is desired since one wants to retain as much

information as possible in the decomposed system, when compared to the original data. The decomposition processes can be generally dichotomized into *lossless (no error) versus lossy decomposition*. In this paper, a comparison of three types of lossless decomposition are considered: the disjoint Ashenhurst-Curtis (AC) decomposition and set-theoretic conventional and Modified Reconstructability Analysis (CRA and MRA, respectively).

The remainder of this paper is organized as follows: section 2 presents background and related work on this subject. CRA, MRA, and AC complexity results are presented in section 3. Conclusions and future work are discussed in section 4.

2 LOGIC FUNCTIONS CLASSIFICATION, COMPLEXITY MEASURES, AND DECOMPOSITIONS

This section introduces the basic background of the NPN-classification of three-variable 2-valued logic functions, Ashenhurst-Curtis (AC) and Reconstructability Analysis (RA) decomposition methods that are used in this work, and complexity measures that are utilized to compare the efficiency of such decompositions.

2.1 NPN-Classification of Logic Functions

There exist many classification methods to cluster logic functions into families of functions (Muroga 1979). Two important operations that produce equivalence classes of logic functions are *negation* and *permutation* (Muroga 1979). Accordingly, the following classification types result:

1. *P-Equivalence class: a family of identical functions obtained by the operation of permutation of variables.*
2. *NP-Equivalence class: a family of identical functions obtained by the operations of negation or permutation of one or more variables.*

3. *NPN-Equivalence class: a family of identical functions obtained by the operations of negation or permutation of one or more variables, and also negation of function.*

The NPN-Equivalence classification will be used in this work. Table 1 lists 3-variable Boolean functions, for the non-degenerate classes (i.e., the classes depending on all three variables).

Class	Representative Function	Number of Functions
1	$F = x_1x_2 + x_2x_3 + x_1x_3$	8
2	$F = x_1 \oplus x_2 \oplus x_3$	2
3	$F = x_1 + x_2 + x_3$	16
4	$F = x_1(x_2 + x_3)$	48
5	$F = x_1x_2x_3 + x_1'x_2'x_3'$	8
6	$F = x_1'x_2x_3 + x_1x_2'x_3' + x_1x_2x_3'$	24
7	$F = x_1(x_2x_3 + x_2'x_3')$	24
8	$F = x_1x_2 + x_2x_3 + x_1'x_3$	24
9	$F = x_1'x_2x_3 + x_1x_2'x_3' + x_1x_2x_3'$	16
10	$F = x_1x_2'x_3' + x_2x_3$	48

Table 1. NPN-Equivalence classes for non-degenerate Boolean functions of three binary variables (Muroga 1979). These classes contain 218 out of the possible 256 functions.

2.2 Complexity Measures

Decomposability means complexity reduction. Many complexity measures exist for the purpose of evaluating the efficiency of the decomposition of complex systems into simpler sub-systems. Such complexity measures include: the *Cardinality* complexity measure (DFC) (Abu-Mostafa 1988), the *Log-Functionality* (LF) complexity measure (Grygiel 2000), and the *Sigma* complexity measure (Zwick 1995). In the first two measures, complexity is a *count* of the total number of possible functions realizable by all of the sub-blocks; the third just indicates the level of decomposition in the lattice of possible structures. *The complexity of the decomposed structure is always less or equal to the complexity of the original look-up-table (LUT) that represents the mapping of the non-decomposed structure.* That is, if a “decomposed” structure has higher complexity than the original structure, then the original structure is said to be non-decomposable. Although the DFC measure is easier

and more familiar, LF is a better measure because it more properly deals with non-disjoint systems (Grygiel 2000). Also, DFC does not correct for function repetition (redundancy). Consequently, the LF measure will be used in this paper. The DFC and LF complexity measures are illustrated using Figure 1, which exemplifies AC decomposition, as follows:

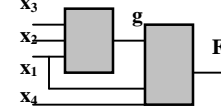


Figure 1. Generic non-disjoint decomposition.

In Figure 1, for the first block, the total number of possible functions for three 2-valued input variables is $2^3 = 256$. Also, for the second block, the total number of possible functions is similarly 256. The total possible number of functions for the whole structure is equal to $256 \cdot 256 = 65,536$. The DFC measure is defined as:

$$DFC = O \cdot 2^I \quad (1)$$

$$C_{DFC} = \sum_n DFC_n \quad (2)$$

where O is the number of outputs to a block, I is the number of inputs to the same block, equation (1) is the complexity for every block, and equation (2) is the complexity for the total decomposed structure. For instance, the DFC for Figure 1 is: $C_{DFC} = 1 \cdot 2^3 + 1 \cdot 2^3 = \log_2(65,536) = 16$, which is the same as the cardinality of the LUT.

It was shown in (Grygiel 2000) that, for Figure 1, the *Log-Functionality complexity measure* (C_{LF}) for Boolean functions can be expressed as follows:

$$C_{LF} = \log_2(C_F) \quad (3)$$

$$\text{where: } C_F = (C'_F)^{p_{X_3}}$$

$$C'_F = \sum_{i=0}^{p_{Y_1}-1} P(p_{Y_2}^{\frac{p_{X_2}}{p_{X_3}}}, p_{Y_1}-i) S(\frac{p_{X_1}}{p_{X_3}}, p_{Y_1}-i)$$

$$P(n, k) = \frac{n!}{(n-k)!},$$

$$S(n, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^n, \quad \binom{k}{i} = \frac{k!}{i!(k-i)!},$$

$$X_1 = \{x_1, x_2, x_3\}, X_2 = \{x_1, x_4\}, X_3 = X_1 \cap X_2 = \{x_1\}$$

$$p_{X_1} = \prod_{x_i \in X_1} |x_i|, p_{X_2} = \prod_{x_i \in X_2} |x_i|, p_{X_3} = \prod_{x_i \in X_3} |x_i|,$$

$$p_{Y_1} = \prod_{y_i \in Y_1} |y_i|, p_{Y_2} = \prod_{y_i \in Y_2} |y_i|$$

where X_1 is the set of input variables to the first block, X_2 is the set of input variables to the second block, X_3 is the set of overlapping variables between sets X_1 and X_2 , P_{X_i} is the product of cardinalities of the input variables in set X_i , and P_{Y_i} is the product of cardinalities of output variables in set Y_i . For example, the LF for Figure 1 is:

$$X_1 = \{x_1, x_2, x_3\}, X_2 = \{x_1, x_4\}, X_3 = X_1 \cap X_2 = \{x_1\}$$

$$\therefore p_{X_1} = 2 \cdot 2 \cdot 2 = 8, p_{X_2} = 2 \cdot 2 = 4, p_{X_3} = 2,$$

$$p_{Y_1} = 2, p_{Y_2} = 2,$$

$$C'_F = \sum_{i=0}^1 P(2^2, 2-i) S(4, 2-i) = 88$$

$$\therefore C_F = 7,744 \Rightarrow C_{LF} = \log_2(7,744) = 12.92.$$

Note that using the DFC measure (16) we would not consider Figure 1 to achieve any complexity reduction (i.e. successful decomposition), but using the LF (12.92), Figure 1 does achieve complexity reduction.

Figure 1 shows a four input function, where the variable sets for the first and second blocks are not disjoint. In this paper we are concerned with 3-input functions, and in this case an AC decomposition, which is successful using the LF measure, results in a structure shown in Figure 2. Note that the variable sets for the two blocks with outputs g and F are necessarily disjoint, because if the two blocks shared one input variable, F would have three inputs and the decomposed structure would be more complex than the original non-decomposed 3-input function.

Example 1.

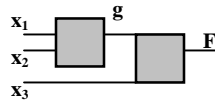


Figure 2. A decomposed structure.

The Log-Functionality complexity measure of the structure in Figure 2 is obtained as follows: Each sub-block in Figure 2 has a total of $2^{2^2} = 16$ possible Boolean functions. Figure 3 illustrates all of the possible 16 two-variable Boolean functions per sub-block in Figure 2.

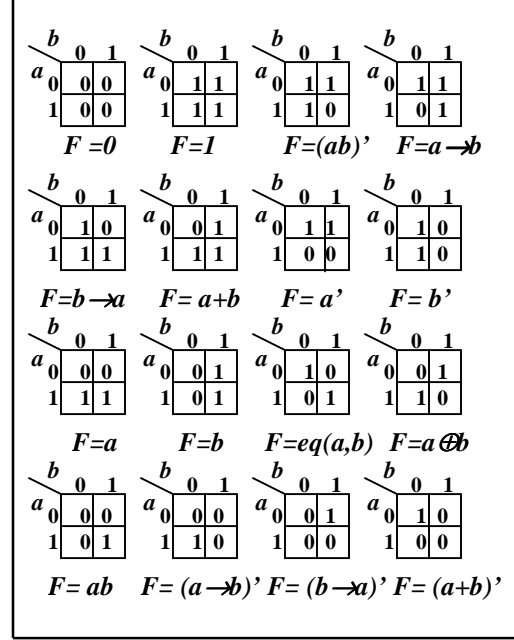


Figure 3. Maps of all 16 possible Boolean functions of two variables. (The single quote means negation.)

By allowing g and F in Figure 2 to take on all possible maps from Figure 3, one obtains the following *count of total non-repeated (irredundant) 3-variable functions*, as follows: $C_F = 88 \Rightarrow C_{LF} = 6.5$. This answer agrees with the result of equation (3) (Grygiel 2000).

Example 2. For 3-variable functions, RA produces four different types of decomposition structures, two of which are shown in Figure 4. (See also Table 3 under “Simplest Modified RA Circuit”.)

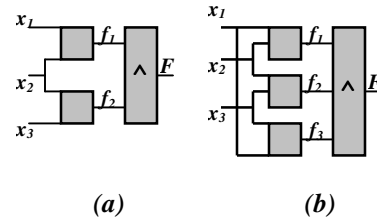


Figure 4. Some RA decomposed Structures.

The Log-Functionality complexity measure for the structures in Figure 4, is obtained as follows. Figure 5 represents a tree that generates all possible functions for the structures 4a and 4b, respectively. (Superscripts of functions denote the specific edge between two nodes in the tree).

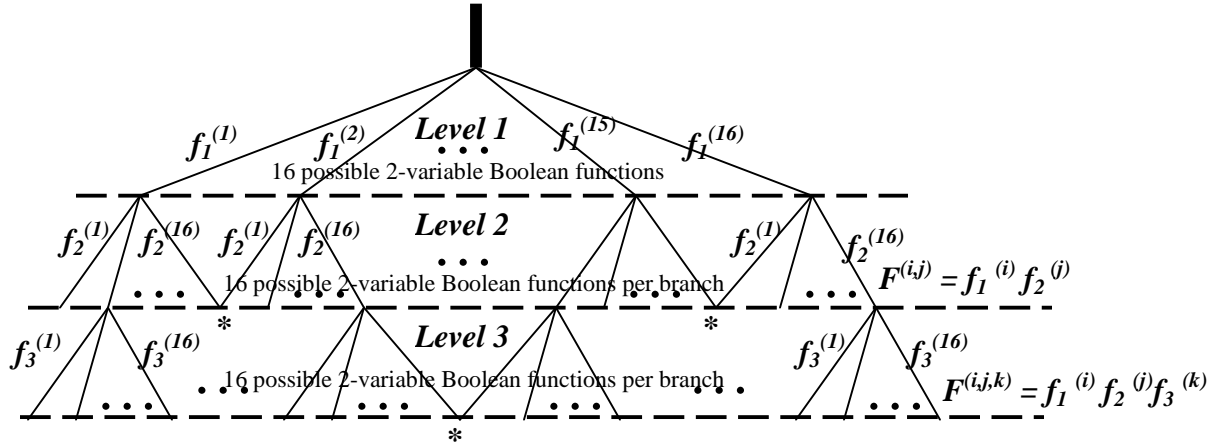


Figure 5. All possible combinations of sub-functions $f_1^{(i)}$, $f_2^{(j)}$, and $f_3^{(k)}$ in Figures 4a and 4b, respectively. The *log-functionality complexity measure* is the count of all the *irredundant* functions, that is all *different* sub-functions and $F^{(i,j)}$ within Figure 4a, and all *different* sub-functions and $F^{(i,j,k)}$ within Figure 4b. Where two nodes of the tree are superposed (*), they are counted only once. At level 2, 100 of the $(16)^2$ possible nodes are irredundant, and at level 3, 152 out of $(16)^3$ are irredundant.

Utilizing this methodology of removing redundant functions, one obtains the following results for Log-Functionality: for Figure 4a, the total number of irredundant sub-functions at level 2 is $C_F = 100 \Rightarrow \therefore C_{LF} = \log_2(100) = 6.6$, and for Figure 4b, the total number of irredundant sub-functions at level 3 is $C_F = 152 \Rightarrow \therefore C_{LF} = \log_2(152) = 7.2$.

2.3 Ashenhurst-Curtis Decomposition

Ashenhurst-Curtis (AC) decomposition (Ashenhurst 1953, Ashenhurst 1956, Ashenhurst 1959, Curtis 1962, Curtis 1963, Files 2000, Grygiel 2000) is one of the major techniques for the decomposition of functions commonly used in the field of logic synthesis. The main idea of AC decomposition is to decompose logic functions into simpler logic blocks using the compression of the number of cofactors in the corresponding representation. This compression is achieved through exploiting the logical compatibility (i.e., redundancy) of cofactors (i.e., column multiplicity). As a result of AC decomposition, intermediate constructs (latent variables) are created. A general algorithm of the AC decomposition utilizing Karnaugh map (K-map) representation (Muroga 1979), for instance, is as follows:

(1) Partition the input set of variables into free set and bound set, and label all the different columns.

(2) Decompose the bound set and create a new K-map for the decomposed bound set (utilizing minimum graph coloring, maximum clique, or

some other algorithm to combine similar columns into a single column). Each cell in the new K-map represents a labeled column in the original K-map.

(3) Encode the labels in the cells of the new K-map using minimum number of intermediate binary variables. These intermediate variables are shown as g and h in Example 3 (Figure 6). Express the intermediate variables as functions of the bound set variables.

(4) Produce the decomposed structure, i.e., a K-map specifying the function (F) in terms of the intermediate variables and the free set variables.

In general, steps (1) and (3) determine the optimality of the AC decomposition (i.e., whether the resulting decomposed blocks are of minimal complexity or not).

Example 3. For the following logic function $F = x_2x_3 + x_1x_3 + x_1x_2$, let the sub-set of variables $\{x_2, x_3\}$ be the *Bound Set*, and the sub-set of variables $\{x_1\}$ be the *Free Set*. The following is the disjoint AC decomposition of F (where $\{-\}$ means don't care):

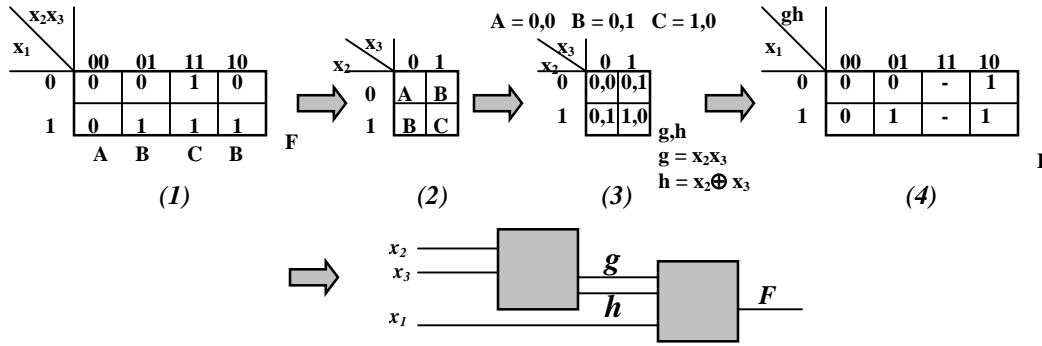


Figure 6. AC decomposition. Steps (1)-(4) are discussed in the text.

In Example 3, the first block of the decomposed structure has two outputs (intermediate variables g and h). The DFC measure of the decomposed structure is $= 2 \cdot 2^2 + 1 \cdot 2^3 = 16$, while the DFC of the original LUT is $= 1 \cdot 2^3 = 8$. (This again shows the inadequacy of DFC as a measure of complexity because the decomposition produces a more complex structure than the non-decomposed LUT.)

LF for the decomposed structure in Figure 6 is 8, which does not exceed the complexity of the LUT. However, since the decomposition does not reduce the complexity, for the purposes of this paper, the decomposition is not successful and thus rejected. This will be true whenever the first block of the decomposed function has two outputs. For other NPN functions AC decomposition produces only one output in the first block. These decompositions are not rejected, and are listed in Table 3.

2.4 Reconstructability Analysis: Conventional RA Versus Modified RA

Reconstructability Analysis (RA) is a decomposition technique for qualitative data (Conant 1981, Klir 1985, 1996; Krippendorff 1986). A review with additional references is provided in (Zwicky 2001). RA data is typically either a set theoretic relation or mapping or it is a probability or frequency distribution. The former case is the domain of “set-theoretic” - or more precisely crisp possibilistic - RA. The latter is the domain of “information-theoretic” - or more precisely probabilistic - RA. The RA framework can apply to other types of data (e.g., fuzzy data) via generalized information theory (Klir and Wierman 1998).

RA decomposition can also be lossless or lossy. In this paper, we are concerned only with lossless decomposition, i.e., with decomposition which produces no error. This paper introduces an innovation in set-theoretic RA, which we call

“modified” RA (or MRA) (Al-Rabadi 2001) as opposed to the conventional set-theoretic RA (or CRA). While CRA decomposes for all values of Boolean functions, MRA decomposes for an arbitrarily chosen value of the Boolean functions (e.g., for value “1”). *The completely specified Boolean function can be retrieved if one knows the MRA decomposition for the Boolean function being equal either to “1” or to “0”*. MRA and CRA are illustrated and compared in Example 4.

Example 4. For the logic function: $F = x_1x_2 + x_1x_3$ Figure 7 illustrates the simplest model using both CRA and MRA decompositions.

CRA decomposition (Conant 1981, Zwicky 1995, Zwicky and Shu 1995) is illustrated in the upper half of the figure, while MRA decomposition (Al-Rabadi 2001) is illustrated in the lower half of the figure. MRA decomposition yields a much simpler logic circuit than the corresponding CRA decomposition, while retaining complete information about the decomposed function.

For CRA as shown in the top middle part of the figure, the calculated function for the model $x_1x_2f_1:x_1x_3f_2:x_2x_3f_3$ (i.e., $\alpha:\beta:\gamma$) is defined as follows: $x_1x_2x_3F_{x_1x_2f_1:x_1x_3f_2:x_2x_3f_3} \equiv (x_1x_2f_1 \otimes x_3) \cap (x_1x_3f_2 \otimes x_2) \cap (x_2x_3f_3 \otimes x_1)$. (For lossless CRA decomposition, this equals the original function $x_1x_2x_3F$ that is shown at the top left of the figure; for lossy CRA $x_1x_2x_3F_{x_1x_2f_1:x_1x_3f_2:x_2x_3f_3}$ would not be equivalent to $x_1x_2x_3F$). The CRA model can be interpreted by the circuit shown at the top right of the figure.

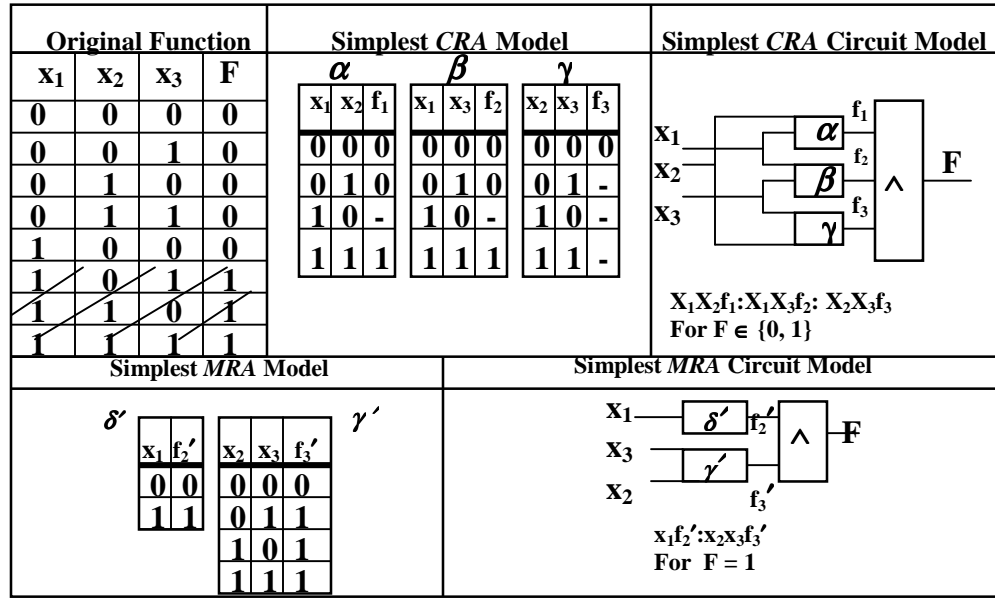


Figure 7. Conventional versus Modified RA decompositions for the Boolean function: $F = x_1x_2 + x_1x_3$.

MRA simplifies the decomposition problem by focusing, in the original function F , on the tuples for which $F=1$. (One could alternatively have selected the tuples for which $F = 0$.) The procedure used to obtain the MRA in Figure 7 is as follows (Al-Rabadi 2001):

- (1) Select the relation defined by (x_1, x_2, x_3) tuples with value “1” (shaded in top left of Figure 7).
- (2) Obtain the simplest lossless CRA decomposition.
- (3) Assign value “1” to tuples in the resulting projections. Add all tuples that are missing in the projections and give them function value “0”.
- (4) Perform the intersection in the output block to obtain the total functionality.

Steps (2)-(4) are as follows:

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

Step (2) gives model: $(x_1: x_2x_3)$

x_1	x_2	x_3
1	0	1
	1	0
	1	1

Step (3)

x_1	f_2'	x_2	x_3	f_3'
0	0	0	0	0
1	1	0	1	1
		1	0	1
		1	1	1

$(x_1 \otimes 1 \cup x_1' \otimes 0) \quad (x_2x_3 \otimes 1 \cup (x_2x_3)' \otimes 0)$
where (\cdot) means set complement.

↓ Step (4)

$$F = f_2'(x_1) \wedge f_3'(x_2, x_3)$$

Table 2 gives the complexities of the decomposition of all NPN-classes of 3-variable Boolean functions (Table 1) using CRA decomposition and MRA decomposition, respectively.

NPN-Representative Function	Simplest Modified RA model	Simplest Conventional RA model	C (LUT)	C _{LF} (CRA)	C _{LF} (MRA)																																																																																										
<i>Class 1</i> (8) F = x ₁ x ₂ + x ₂ x ₃ + x ₁ x ₃	<table><tr><td>x₁</td><td>x₂</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₂</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₁</td><td>x₃</td><td>f₃</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	x ₂	f ₁	0	0	0	0	1	1	1	0	1	1	1	1	x ₂	x ₃	f ₂	0	0	0	0	1	1	1	0	1	1	1	1	x ₁	x ₃	f ₃	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><td>x₁</td><td>x₂</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>-</td></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₂</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>-</td></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₁</td><td>x₃</td><td>f₃</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>-</td></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	x ₂	f ₁	0	0	0	0	1	-	1	0	-	1	1	1	x ₂	x ₃	f ₂	0	0	0	0	1	-	1	0	-	1	1	1	x ₁	x ₃	f ₃	0	0	0	0	1	-	1	0	-	1	1	1	8	7.2	7.2
x ₁	x ₂	f ₁																																																																																													
0	0	0																																																																																													
0	1	1																																																																																													
1	0	1																																																																																													
1	1	1																																																																																													
x ₂	x ₃	f ₂																																																																																													
0	0	0																																																																																													
0	1	1																																																																																													
1	0	1																																																																																													
1	1	1																																																																																													
x ₁	x ₃	f ₃																																																																																													
0	0	0																																																																																													
0	1	1																																																																																													
1	0	1																																																																																													
1	1	1																																																																																													
x ₁	x ₂	f ₁																																																																																													
0	0	0																																																																																													
0	1	-																																																																																													
1	0	-																																																																																													
1	1	1																																																																																													
x ₂	x ₃	f ₂																																																																																													
0	0	0																																																																																													
0	1	-																																																																																													
1	0	-																																																																																													
1	1	1																																																																																													
x ₁	x ₃	f ₃																																																																																													
0	0	0																																																																																													
0	1	-																																																																																													
1	0	-																																																																																													
1	1	1																																																																																													
<i>Class 2</i> (2) F = x ₁ ⊕ x ₂ ⊕ x ₃	non-decomposable	non-decomposable	8	8	8																																																																																										
<i>Class 3</i> (16) F = x ₁ + x ₂ + x ₃	non-decomposable	non-decomposable	8	8	8																																																																																										
<i>Class 4</i> (48) F = x ₁ (x ₂ + x ₃)	<table><tr><td>x₁</td><td>f₁</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table> <table><tr><td>x₂</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	f ₁	0	0	1	1	x ₂	x ₃	f ₂	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><td>x₁</td><td>x₂</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₂</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₁</td><td>x₃</td><td>f₃</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>-</td></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>-</td></tr></table>	x ₁	x ₂	f ₁	0	0	0	0	1	0	1	0	-	1	1	1	x ₂	x ₃	f ₂	0	0	0	0	1	0	1	0	-	1	1	1	x ₁	x ₃	f ₃	0	0	0	0	1	-	1	0	-	1	1	-	8	7.2	6.5																								
x ₁	f ₁																																																																																														
0	0																																																																																														
1	1																																																																																														
x ₂	x ₃	f ₂																																																																																													
0	0	0																																																																																													
0	1	1																																																																																													
1	0	1																																																																																													
1	1	1																																																																																													
x ₁	x ₂	f ₁																																																																																													
0	0	0																																																																																													
0	1	0																																																																																													
1	0	-																																																																																													
1	1	1																																																																																													
x ₂	x ₃	f ₂																																																																																													
0	0	0																																																																																													
0	1	0																																																																																													
1	0	-																																																																																													
1	1	1																																																																																													
x ₁	x ₃	f ₃																																																																																													
0	0	0																																																																																													
0	1	-																																																																																													
1	0	-																																																																																													
1	1	-																																																																																													
<i>Class 5</i> (8) F = x ₁ x ₂ x ₃ + x ₁ 'x ₂ 'x ₃ '	<table><tr><td>x₁</td><td>x₂</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₁</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	x ₂	f ₁	0	0	1	0	1	0	1	0	0	1	1	1	x ₁	x ₃	f ₂	0	0	1	0	1	0	1	0	0	1	1	1	non-decomposable	8	8	6.6																																																												
x ₁	x ₂	f ₁																																																																																													
0	0	1																																																																																													
0	1	0																																																																																													
1	0	0																																																																																													
1	1	1																																																																																													
x ₁	x ₃	f ₂																																																																																													
0	0	1																																																																																													
0	1	0																																																																																													
1	0	0																																																																																													
1	1	1																																																																																													
<i>Class 6</i> (24) F = x ₁ 'x ₂ x ₃ + x ₁ x ₂ ' + x ₁ x ₃ '	non-decomposable	non-decomposable	8	8	8																																																																																										
<i>Class 7</i> (24) F = x ₁ (x ₂ x ₃ + x ₂ 'x ₃ ')	<table><tr><td>x₂</td><td>x₃</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₁</td><td>f₂</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x ₂	x ₃	f ₁	0	0	1	0	1	0	1	0	0	1	1	1	x ₁	f ₂	0	0	1	1	non-decomposable	8	8	6.5																																																																					
x ₂	x ₃	f ₁																																																																																													
0	0	1																																																																																													
0	1	0																																																																																													
1	0	0																																																																																													
1	1	1																																																																																													
x ₁	f ₂																																																																																														
0	0																																																																																														
1	1																																																																																														
<i>Class 8</i> (24) F = x ₁ x ₂ + x ₂ x ₃ + x ₁ 'x ₃	<table><tr><td>x₁</td><td>x₂</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₁</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	x ₂	f ₁	0	0	1	0	1	1	1	0	0	1	1	1	x ₁	x ₃	f ₂	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><td>x₁</td><td>x₂</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>-</td></tr><tr><td>0</td><td>1</td><td>-</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₁</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>1</td><td>1</td><td>-</td></tr></table>	x ₁	x ₂	f ₁	0	0	-	0	1	-	1	0	0	1	1	1	x ₁	x ₃	f ₂	0	0	0	0	1	1	1	0	-	1	1	-	8	6.6	6.6																														
x ₁	x ₂	f ₁																																																																																													
0	0	1																																																																																													
0	1	1																																																																																													
1	0	0																																																																																													
1	1	1																																																																																													
x ₁	x ₃	f ₂																																																																																													
0	0	0																																																																																													
0	1	1																																																																																													
1	0	1																																																																																													
1	1	1																																																																																													
x ₁	x ₂	f ₁																																																																																													
0	0	-																																																																																													
0	1	-																																																																																													
1	0	0																																																																																													
1	1	1																																																																																													
x ₁	x ₃	f ₂																																																																																													
0	0	0																																																																																													
0	1	1																																																																																													
1	0	-																																																																																													
1	1	-																																																																																													
<i>Class 9</i> (16) F = x ₁ 'x ₂ x ₃ + x ₁ x ₂ 'x ₃ ' + x ₁ x ₂ x ₃ '	non-decomposable	non-decomposable	8	8	8																																																																																										
<i>Class 10</i> (48) F = x ₁ x ₂ 'x ₃ ' + x ₂ x ₃	<table><tr><td>x₁</td><td>x₃</td><td>f₁</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><td>x₂</td><td>x₃</td><td>f₂</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	x ₃	f ₁	0	0	0	0	1	1	1	0	1	1	1	1	x ₂	x ₃	f ₂	0	0	1	0	1	0	1	0	0	1	1	1	non-decomposable	8	8	6.6																																																												
x ₁	x ₃	f ₁																																																																																													
0	0	0																																																																																													
0	1	1																																																																																													
1	0	1																																																																																													
1	1	1																																																																																													
x ₂	x ₃	f ₂																																																																																													
0	0	1																																																																																													
0	1	0																																																																																													
1	0	0																																																																																													
1	1	1																																																																																													

Table 2. Conventional RA (CRA) versus Modified RA (MRA) for the decomposition of all NPN-classes of 3-variable Boolean functions. C_{LUT} is the cardinality of the lookup table; for non-decomposable functions C_{LF} = C_{LUT}. Compare the right-most two columns.

The table shows that in 6 NPN classes (classes 1, 2, 3, 6, 8, 9) MRA and CRA give equivalent complexity decompositions, but in the remaining

four classes (classes 4, 5, 7, 10) MRA is superior in complexity reduction.

3 COMPLEXITY OF MRA VERSUS AC DECOMPOSITION

Utilizing the methods described above, one obtains the following results in Table 3 for the

decomposition of 3-variable NPN-classified Boolean functions (Table 1) using MRA and AC decomposition.

NPN-Representative Function	Simplest Modified RA model	Simplest AC circuit	Simplest Modified RA circuit	DFC (SOP)	C _{data} (LUT)	C _{LF} (MRA)	C _{LF} (AC)
Class 1 (8) $F = x_1x_2 + x_2x_3 + x_1x_3$		 $g = x_2x_3, h = x_2 \oplus x_3, F = g + x_1h$		20	8	7.2	8
Class 2 (2) $F = x_1 \oplus x_2 \oplus x_3$	non-decomposable	 $g = x_2 \oplus x_3, F = x_1 \oplus g$	-	8	8	8	6.5
Class 3 (16) $F = x_1 + x_2 + x_3$	non-decomposable	 $g = x_2 + x_3, F = x_1 + g$	-	8	8	8	6.5
Class 4 (48) $F = x_1(x_2 + x_3)$		 $g = x_2 + x_3, F = x_1g$		12	8	6.5	6.5
Class 5 (8) $F = x_1x_2x_3 + x_1'x_2'x_3'$		 $g = x_2x_3, h = x_2 \oplus x_3, F = x_1g'h' + x_1g$		20	8	6.6	8
Class 6 (24) $F = x_1'x_2x_3 + x_1x_2' + x_1x_3'$	non-decomposable	 $g = x_2x_3, F = x_1 \oplus g$	-	24	8	8	6.5
Class 7 (24) $F = x_1(x_2x_3 + x_2'x_3')$		 $g = x_2 \oplus x_3, F = x_1g'$		20	8	6.5	6.5
Class 8 (24) $F = x_1x_2 + x_2x_3 + x_1'x_3$		 $g = x_2, h = x_2 \oplus x_3, F = x_1g + gh' + x_1'g'h$		20	8	6.6	8
Class 9 (16) $F = x_1'x_2x_3 + x_1x_2'x_3 + x_1x_2x_3'$	non-decomposable	 $g = x_2x_3, h = x_2 \oplus x_3, F = x_1h + x_1'g$	-	32	8	8	8
Class 10 (48) $F = x_1x_2'x_3' + x_2x_3$		 $g = x_2x_3, h = x_2 \oplus x_3, F = g + x_1g'h'$		16	8	6.6	8

Table 3. AC decomposition versus MRA decomposition for the decomposition of all NPN-classes of 3-variable Boolean functions. C_{LUT} is the cardinality of the lookup table; for non-decomposable functions C_{LF} = C_{LUT}. Compare the right-most two columns. (DFC_(SOP) is the cardinality of the Sum-Of-Product form for the NPN class.)

		MRA				AC		MRA	
		<i>ND</i>	<i>D</i>			<i>ND</i>	<i>D</i>	<i>ND</i>	<i>D</i>
<i>AC</i>	<i>ND</i>	1 (9: 16)	4 (1,5,8,10: 88)	<i>CRA</i>	<i>ND</i>	3 (5,9,10: 72)	4 (2,3,6,7: 66)	<i>CRA</i>	<i>ND</i>
	<i>D</i>	3 (2,3,6: 42)	2 (4,7: 72)		<i>D</i>	2 (1,8: 32)	1 (4: 48)		<i>D</i>
		(a)				(b)			
								(c)	

Figure 8. Comparison of Decomposability (*D*) versus Non-Decomposability (*ND*) for *AC versus MRA* (a), *CRA versus AC* (b), and *CRA versus MRA* (c), respectively. The number of classes and the number of functions are listed in each cell.

Table 3 shows that in three NPN classes (4, 7, 9) MRA and AC give equivalent complexity decompositions. In three other classes (2, 3, 6), which encompass 42 functions, AC is superior, but in four classes (1, 5, 8, 10), which encompass 88 functions, MRA is superior. We can summarize these results by comparing the decomposability versus non-decomposability for the various approaches. Figure 8 shows the number of classes and functions decomposable by one method but not by another (upper right and lower left cells). One concludes that for NPN-classified 3-variable Boolean functions, MRA decomposition is superior to AC decomposition (88 versus 42), AC decomposition is superior to CRA decomposition (66 versus 32), and MRA decomposition is superior to CRA decomposition (80 versus 0).

While the log-functionality complexity measure that is used in Table 3 is a good cost measure for machine learning, it is not a good measure for circuit design. An alternative cost measure for circuit design is the count of the total number of two-input gates (from Figure 3) in the final circuit ($C_{\#}$). Utilizing the resulting decompositions from Table 3, Table 4 presents a comparison between MRA and AC for 3-variable NPN classes of Boolean functions using the $C_{\#}$ complexity measure.

Table 4 shows that, using the $C_{\#}$ cost measure, in five NPN classes (1,2,3,6,9) which encompass 66 logic functions AC is superior to MRA for both including and not including the cost of the inverters. For two NPN classes (4,8), which encompass 72 logic functions, AC is equivalent to MRA for both including and not including the cost of the inverters. For two NPN classes (5,10), which encompass 56 logic functions, MRA is superior to AC for both including and not including the cost of the inverters. For one NPN class (7), which encompasses 24 logic functions, MRA is

superior to AC when including the cost of the inverters, but the same as AC when inverters are not included. Thus, counting inverters, MRA is superior to AC (80 versus 66), while not counting inverters AC is superior to MRA (66 versus 56).

Class	$C_{\#}$ W/ Inverters (AC)	$C_{\#}$ W/O Inverters (AC)	$C_{\#}$ W/ Inverters (MRA)	$C_{\#}$ W/O Inverters (MRA)
1 (8)	4	4	5	5
2 (2)	2	2	-	-
3 (16)	2	2	-	-
4 (48)	2	2	2	2
5 (8)	9	6	3	3
6 (24)	2	2	-	-
7 (24)	3	2	2	2
8 (24)	4	3	4	3
9 (16)	6	5	-	-
10 (48)	7	5	3	3
(66)		(66)	(80)	(56)

Table 4. Comparison of AC versus MRA using the $C_{\#}$ cost measure. The number in parenthesis at the bottom of each table column is the sum of the numbers of functions in the shaded cells of that column.

The results of Table 4 are technology independent, that is, every logic function in Figure 3 is given the same cost. However, from a technology dependent point of view, the costs of

the different logic functions of Figure 3 may not be the same, and the comparisons of Table 4 would have to be modified accordingly.

4 CONCLUSION

A novel RA-based decomposition is introduced; *Modified Reconstructability Analysis (MRA)*. MRA is compared to conventional Reconstructability Analysis (CRA) and disjoint Ashenhurst-Curtis (AC) decomposition using the log-functionality complexity measure which is a suitable measure for machine learning. It is shown that in 3 out of 7 NPN classes while 3-variable NPN-classified Boolean functions are not decomposable using CRA, they are decomposable using MRA. Also, it is shown that whenever a decomposition of 3-variable NPN-classified Boolean functions exists in both MRA and CRA, MRA yields a simpler or equal complexity decomposition. While both the disjoint AC decomposition and MRA decompose some but not all NPN-classes, MRA decomposes more classes and consequently more Boolean functions than AC.

For the purpose of circuit design, complexity can be defined by counting the total number of two-input gates. Using this measure, MRA is superior to AC when including the cost of the inverters and AC is superior to MRA when not including the cost of the inverters. Extensions of this MRA approach to reversible logic and quantum computing is presented in (Al-Rabadi and Zwick 2002); extensions to many-valued logic is presented in (Al-Rabadi and Zwick 2002). A comprehensive treatment of MRA with supplementary material is provided in (Al-Rabadi 2002).

Future work will include the investigation of the MRA decomposition of logic relations as opposed to functions, and multi-valued and fuzzy functions. The use of gates other than the logical AND gate (e.g., OR, XOR, NAND) at the final stage of RA-based decompositions to reduce the complexities of the decomposed structures will also be investigated.

5 REFERENCES

Abu-Mostafa, Y. 1988. *Complexity in Information Theory*, Springer-Verlag, New York.

Al-Rabadi, A. N. 2001. "A Novel Reconstructability Analysis For the Decomposition of Boolean Functions," Technical Report #2001/005, Electrical and Computer Engineering Department, Portland State University, Portland, Oregon, 1st July 2001.

Al-Rabadi, A. N. and M. Zwick, 2002. "Reversible Modified Reconstructability Analysis of Boolean Circuits and its Quantum Computation," *Book of Abstracts of the WOSC-IIGSS 2002*, Pittsburgh, Pennsylvania, p. 90.

Al-Rabadi, A. N. and M. Zwick, 2002. "Modified Reconstructability Analysis for Many-Valued Logic Functions," *Book of Abstracts of the WOSC-IIGSS 2002*, Pittsburgh, Pennsylvania, p. 90.

Al-Rabadi, A. N. 2002. *Novel Methods for Reversible Logic Synthesis and their Application to Quantum Computing*, Ph.D. dissertation, Portland State University, Portland, Oregon.

Ashenhurst, R. L. 1953. "The Decomposition of Switching Functions," *Bell Laboratories' Report*, Vol. 1, pp. II-1-II-37.

Ashenhurst, R. L. 1956. "The Decomposition of Switching Functions," *Bell Laboratories' Report*, Vol. 16, pp. III-1-III-72.

Ashenhurst, R. L. 1959. "The Decomposition of Switching Functions," In: *International Symposium on the Theory of Switching Functions*, pp. 74-116.

Conant, R. 1981. "Set-Theoretic Structural Modeling," *Int. J. General Systems*, Vol. 7, pp. 93-107.

Curtis, H. 1963. "Generalized Tree Circuit," *ACM*, pp. 484-496.

Curtis, H. 1963. "Generalized Tree Circuit-The Basic Building Block of an Extended Decomposition Theory," *ACM*, Vol. 10, pp. 562-581.

Curtis, H. A. 1962. *A New Approach to the Design of Switching Circuits*, Princeton, Van Nostrand, NJ.

Files, C. M. 2000. *A New Functional Decomposition Method as Applied to Machine Learning and VLSI Layout*, Ph.D. dissertation, Portland State University, Portland, Oregon.

Grygiel, S. 2000. *Decomposition of Relations as a new Approach to Constructive Induction in Machine Learning and Data Mining*. Ph.D. dissertation, Portland State University, Portland, Oregon.

Klir, G. 1985. *Architecture of Systems Problem Solving*. Plenum Press, New York.

Klir, G., editor, 1996. "Reconstructability Analysis Bibliography," *Int. Journal of General Systems*, 24: 225-229.

Klir, G. and M. J. Wierman, 1998. *Uncertainty-Based Information: Variables of Generalized Information Theory*, Physica-Verlag, New York.

Krippendorff, K. 1986. *Information Theory: Structural Models for Qualitative Data*. Sage Publications, Inc.

Muroga, S. 1979. *Logic Design and Switching Theory*, Wiley, New York.

Shannon, C. E. and W. Weaver 1949. *A Mathematical Theory of Communication*, University of Illinois Press.

Zwick, M. 1995. "Control Uniqueness in Reconstructability Analysis," *International Journal of General Systems*, 23(2).

Zwick, M. and H. Shu, 1995. "Set-Theoretic Reconstructability of Elementary Cellular Automata," *Advances in System Science and Application, Special Issue I*, pp. 31-36.

Zwick, M. 2001. Wholes and Parts in General Systems Methodology. In: *The Character Concept in Evolutionary Biology*, edited by G. Wagner, Academic Press.

BIOGRAPHY

Anas N. Al-Rabadi is currently a Ph.D. candidate in the Electrical and Computer Engineering Department at Portland State University, Portland, Oregon. He received his M.S. in Electrical and Computer Engineering from Portland State University in 1998 in the specialty of Power Electronics and Control Systems Design. His current research includes reconstructability analysis, reversible logic, quantum logic, optical computing, systems philosophy, and logic synthesis.

Martin Zwick is a Professor of Systems Science at Portland State University. Prior to taking his current position at PSU, he was a faculty member in the Department of Biophysics and Theoretical Biology at the University of Chicago, where he worked in macromolecular structure and mathematical crystallography. In the 1970's his interests shifted to systems theory and methodology. Since 1976 he has been on the

faculty of the PSU Systems Science Ph.D. Program and during the years 1984-1989 he was the program coordinator and then director. His research interests are in discrete multivariate modeling (reconstructability analysis), "artificial life" and theoretical/computational biology, and systems philosophy.

Marek Perkowski is a Professor in the Department of Electrical and Computer Engineering at Portland State University. He received Ph.D. in automatic control from Warsaw University of Technology, Poland. His research interests include logic synthesis, robotics, machine vision, reversible logic, and quantum logic. He is a member of the IEEE Computer Society. He is an editorial board member of the soft computing journal and an academic advisor of the PSU student chapter of the IEEE Robotics and Automation Society.