

Portland State University

PDXScholar

Systems Science Faculty Publications and
Presentations

Systems Science

2004

Reversible Modified Reconstructability Analysis of Boolean Circuits and Its Quantum Computation

Anas Al-Rabadi

Portland State University

Martin Zwick

Portland State University, zwick@pdx.edu

Follow this and additional works at: https://pdxscholar.library.pdx.edu/sysc_fac



Part of the [Logic and Foundations Commons](#), and the [Systems Architecture Commons](#)

Let us know how access to this document benefits you.

Citation Details

Al-Rabadi, A., and Zwick, M. (2004). "Reversible Modified Reconstructability Analysis of Boolean Circuits and Its Quantum Computation." *Kybernetes*, vol. 33, No. 5/6, pp. 921-932

This Post-Print is brought to you for free and open access. It has been accepted for inclusion in Systems Science Faculty Publications and Presentations by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

REVERSIBLE MODIFIED RECONSTRUCTABILITY ANALYSIS OF BOOLEAN CIRCUITS AND ITS QUANTUM COMPUTATION

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

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

KEYWORDS: Reconstructability Analysis, Boolean Circuits, Reversible Logic, Quantum Computing.

ABSTRACT

Modified Reconstructability Analysis (MRA) can be realized reversibly by utilizing Boolean reversible (3,3) logic gates that are universal in two arguments. The quantum computation of the reversible MRA circuits is also introduced. The reversible MRA transformations are given a quantum form by using the normal matrix representation of such gates. The MRA-based quantum decomposition may play an important role in the synthesis of logic structures using future technologies that consume less power and occupy less space.

1 INTRODUCTION

Decomposition is one methodology to analyze data and identify “hidden” relationships between variables. One major decomposition technique for discrete static or dynamic systems is Reconstructability Analysis (RA), which is developed in the systems community to analyze qualitative data. (Klir 1985, Krippendorff 1986). A recent short review of RA is given in (Zwick 2001). Logic circuits that realize RA have been also shown (Zwick 1995). This paper develops a methodology for reversible and quantum implementation of RA. Due to the anticipated failure of Moore’s law around the year 2020, quantum computing may play an important role in building more compact and less power consuming computers (Nielsen and Chuang 2000). Because all quantum computer gates must be reversible (Bennett 1973, Fredkin 1982, Landauer 1961, Nielsen and Chuang 2000), reversible computing will also be increasingly important in the future design of *regular*, *minimal-size*, and *universal* systems.

The remainder of this paper is organized as follows: A review of our new approach to RA decomposition of logic functions is presented in section 2. Background on reversible logic and the reversible realization of RA-based Boolean

circuits is presented in section 3. The implementation of reversible Boolean RA-based circuits using quantum logic is introduced in section 4. A more expanded complete discussion of quantum computing is given in section 5. Conclusions and future work are included in section 6.

2 RECONSTRUCTABILITY ANALYSIS: CONVENTIONAL (CRA) VERSUS MODIFIED (MRA)

We are concerned here with “set-theoretic” RA, i.e. the analysis of crisp possibilistic systems (Klir and Wierman 1998). Enhancement of lossless set-theoretic conventional RA (CRA) has been presented in (Al-Rabadi 2001, Al-Rabadi et al 2002). This new enhanced RA is called “Modified Reconstructability Analysis” (MRA). The procedure for the lossless MRA decomposition is as follows: For every structure in the lattice of structures, decompose the Boolean function for *one* function value only (e.g., for value of “1”) into the simplest error-free decomposed structure. One thus obtains the 1-MRA decomposition. This model consists of a set of projections which when intersected yield the original Boolean function.

It has been shown in (Al-Rabadi et al 2002) that lossless MRA yields much simpler logic circuits than the corresponding lossless conventional RA (CRA), while retaining all information about the decomposed logic function. Figure 1 from (Al-Rabadi et al 2002) illustrates the decomposition of all non-degenerate NPN-classes (Hurst 1978) of 3-variable Boolean functions.

3 REVERSIBLE MRA

A (k,k) reversible circuit is a circuit that has the same number of inputs (k) and outputs (k), and is a one-to-one mapping between a vector of inputs and a vector of outputs. Thus, the vector of input states can always be uniquely reconstructed from the vector of output states (Bennett 1973, Fredkin 1982, Kerntopf 2000, Landauer 1961). As it was proven

NPN-Representative Function	Simplest CRA model	Simplest 1-MRA model	1-MRA Circuit																																																																																										
<i>Class 1</i> $F = x_1x_2 + x_2x_3 + x_1x_3$	<table border="1"> <tr><td>x_1</td><td>x_2</td><td>f_1</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>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <table border="1"> <tr><td>x_2</td><td>x_3</td><td>f_2</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>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <table border="1"> <tr><td>x_1</td><td>x_3</td><td>f_3</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>0</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	0	1	0	0	1	1	1	x_2	x_3	f_2	0	0	0	0	1	0	1	0	0	1	1	1	x_1	x_3	f_3	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr><td>x_1</td><td>x_2</td><td>f_1</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 border="1"> <tr><td>x_2</td><td>x_3</td><td>f_2</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 border="1"> <tr><td>x_1</td><td>x_3</td><td>f_3</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	
x_1	x_2	f_1																																																																																											
0	0	0																																																																																											
0	1	0																																																																																											
1	0	0																																																																																											
1	1	1																																																																																											
x_2	x_3	f_2																																																																																											
0	0	0																																																																																											
0	1	0																																																																																											
1	0	0																																																																																											
1	1	1																																																																																											
x_1	x_3	f_3																																																																																											
0	0	0																																																																																											
0	1	0																																																																																											
1	0	0																																																																																											
1	1	1																																																																																											
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																																																																																											
<i>Class 2</i> $F = x_1 \oplus x_2 \oplus x_3$	non-decomposable	non-decomposable	-																																																																																										
<i>Class 3</i> $F = x_1 + x_2 + x_3$	non-decomposable	non-decomposable	-																																																																																										
<i>Class 4</i> $F = x_1(x_2 + x_3)$	<table border="1"> <tr><td>x_1</td><td>x_2</td><td>f_1</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>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <table border="1"> <tr><td>x_2</td><td>x_3</td><td>f_2</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	0	1	0	0	1	1	1	x_2	x_3	f_2	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><td>x_1</td><td>f_1</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table> <table border="1"> <tr><td>x_2</td><td>x_3</td><td>f_2</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																																								
x_1	x_2	f_1																																																																																											
0	0	0																																																																																											
0	1	0																																																																																											
1	0	0																																																																																											
1	1	1																																																																																											
x_2	x_3	f_2																																																																																											
0	0	0																																																																																											
0	1	1																																																																																											
1	0	1																																																																																											
1	1	1																																																																																											
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																																																																																											
<i>Class 5</i> $F = x_1x_2x_3 + x_1x_2x_3'$	non-decomposable	<table border="1"> <tr><td>x_1</td><td>x_2</td><td>f_1</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 border="1"> <tr><td>x_1</td><td>x_3</td><td>f_2</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																																																													
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> $F = x_1x_2x_3 + x_1x_2x_3' + x_1x_3$	non-decomposable	non-decomposable	-																																																																																										
<i>Class 7</i> $F = x_1(x_2x_3 + x_2x_3')$	non-decomposable	<table border="1"> <tr><td>x_2</td><td>x_3</td><td>f_1</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 border="1"> <tr><td>x_1</td><td>f_2</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																																																																						
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> $F = x_1x_2 + x_2x_3 + x_1x_3$	<table border="1"> <tr><td>x_1</td><td>x_2</td><td>f_1</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>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <table border="1"> <tr><td>x_1</td><td>x_3</td><td>f_2</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	0	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 border="1"> <tr><td>x_1</td><td>x_2</td><td>f_1</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 border="1"> <tr><td>x_1</td><td>x_3</td><td>f_2</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																															
x_1	x_2	f_1																																																																																											
0	0	0																																																																																											
0	1	0																																																																																											
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	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																																																																																											
<i>Class 9</i> $F = x_1x_2x_3 + x_1x_2x_3' + x_1x_2x_3$	non-decomposable	non-decomposable	-																																																																																										
<i>Class 10</i> $F = x_1x_2x_3' + x_2x_3$	non-decomposable	<table border="1"> <tr><td>x_1</td><td>x_3</td><td>f_1</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>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <table border="1"> <tr><td>x_2</td><td>x_3</td><td>f_2</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	0	1	1	1	x_2	x_3	f_2	0	0	1	0	1	0	1	0	0	1	1	1																																																													
x_1	x_3	f_1																																																																																											
0	0	0																																																																																											
0	1	1																																																																																											
1	0	0																																																																																											
1	1	1																																																																																											
x_2	x_3	f_2																																																																																											
0	0	1																																																																																											
0	1	0																																																																																											
1	0	0																																																																																											
1	1	1																																																																																											

Figure 1. Conventional RA (CRA) versus Modified RA (MRA) for the decomposition of all non-degenerate NPN-classes of 3-variable Boolean Functions.

(Landauer 1961) it is a necessary but not sufficient condition for not dissipating power in a physical circuit that all sub-circuits must be built using reversible logical components. Many reversible gates have been proposed as building blocks for reversible computing (Kerntopf 2000, Nielsen and Chuang 2000). Figure 2 shows some of the gates that are commonly used in the synthesis of reversible Boolean logic circuits. It

has been shown in (Fredkin 1982) that for a (k,k) reversible gate to be universal the gate should have at least three inputs (i.e., (3,3) gate). (A gate is universal if it can implement all functions for a given number of arguments.) One should note that not all (3,3) reversible gates are universal, but each universal reversible gate has at least to be a (3,3) gate. Boolean reversible

(3,3) gates which are universal in two arguments have been shown in (Kerntopf 2000).

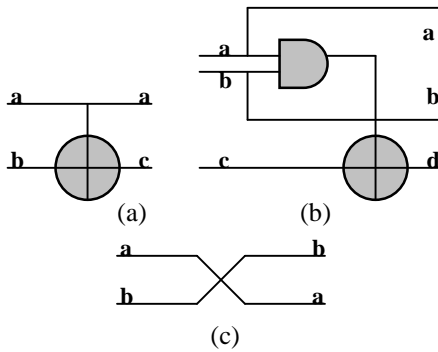
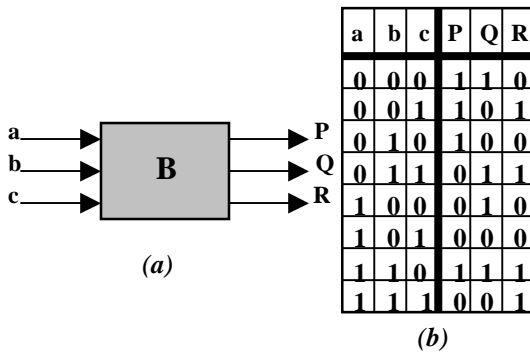


Figure 2. Binary reversible gates: (a) (2,2) Feynman gate which uses XOR, (b) (3,3) Toffoli gate which uses AND and XOR, and (c) (2,2) swap gate which is two permuted wires.

Reversible (3,3) gates, that are universal in two arguments, can be used for the construction of reversible MRA circuits. Figure 3 illustrates one example of a binary (3,3) reversible gate which is universal in two arguments.



Inputs	Outputs
a=0	$R=c, Q=(b\oplus c)', P=b'+c'$
a=1	$R=b, P=bc'$
b=0	$P=a'$
b=1	$R=a+c, Q=a\oplus c, P=c'$
c=0	$R=ab, Q=a+b', P=a'+b$
c=1	$Q=a'b, P=a'b'$
a=1,b=0	$P=0$
a=1,b=1	$R=1$

Figure 3. (a) Diagram of the reversible (3,3) Boolean logic circuit, (b) truth table of this gate, and (c) proof of universality of the gate in two arguments.

The following example illustrates the use of the reversible gate in Figure 3 for the synthesis of 1-MRA circuit for class 5 from Figure 1. The

1-MRA decomposed Boolean circuit of class 5 in Figure 1 can be realized using the binary (3,3) reversible circuit in Figure 3b. This is done with the reversible circuit in Figure 4, where blocks B1 and B2 are the reversible (3,3) gate from Figure 3b, and block B3 is the reversible (3,3) gate from Figure 2b. For B3, $c = 0$ and thus B3 is a reversible logic AND gate.

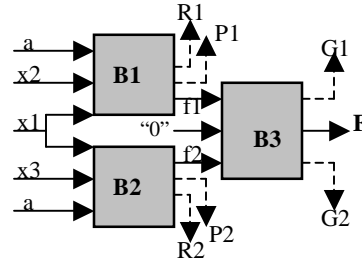


Figure 4. Reversible (7,7) Boolean circuit that implements the 1-MRA circuit from class 5 in Figure 1. Input {a} in B1 and B2 and the set of outputs {R1, P1, R2, P2, G1, G2} are needed for reversibility. Input {a} also selects the appropriate function value of which the universal B gate (Figure 3b) is to implement.

Utilizing Figure 3c, the Boolean reversible circuit in Figure 4 implements the 1-MRA circuit of class 5 (in Figure 1) using the following input settings:

$$a = 0 \Rightarrow Q1 = f1 = (x1 \oplus x2)'$$

$$a = 0 \Rightarrow Q2 = f2 = (x1 \oplus x3)'$$

$$F = Q1 \wedge Q2 = f1 \wedge f2 = (x1 \oplus x2)' \wedge (x1 \oplus x3)'$$

$$= x1x2x3 + x1'x2'x3'$$

For block B3, in Figure 4, one could alternatively use the gate described in Figure 3b: for $c = 0$ output R is the logical AND; in this case the reversible circuit is fully regular (i.e., made up of only one kind of gate). However, using the Toffoli gate (Figure 2b) for B3 is less complex; in this case the circuit is semi-regular (i.e., all the gates in the first level are the same, but the AND of the second level is done by a different gate).

Using similar substitutions with appropriate input values according to Figure 3b, the reversible circuit in Figure 4 can realize all 1-MRA circuits from classes 8 and 10 in Figure 1, respectively. The remaining classes from Figure 1 can be realized using analogous techniques, by adding one more block from Figure 3b to the first level of Figure 4 in the case of class 1, and removing one block from the first level of Figure 4 in the case of classes 4 and 7, respectively.

4 QUANTUM MRA

Quantum computing is a recent trend in logic computation that utilizes the atomic structures to perform the logic computation processes (Nielsen and Chuang 2000). Although the underlying principles for quantum computing are the theorems and principles of quantum mechanics (Dirac 1930), it has been shown (Nielsen and Chuang 2000) that the physical quantum evolution processes can be reduced to algebraic matrix equations. Such matrix representation is a pure mathematical representation that can be realized physically using the corresponding quantum devices.

Figure 5 illustrates this matrix formalism, where each evolution matrix is unitary (Nielsen and Chuang 2000). Each matrix representation in Figure 5 is obtained through the solution of a set of linearly independent equations that correspond to the mapping of input vector to an output vector.

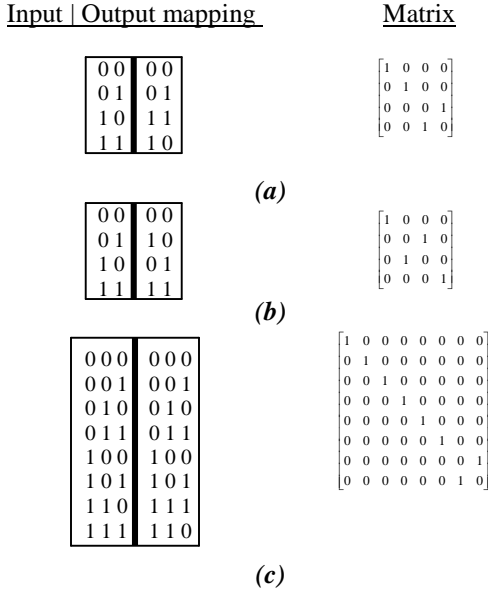


Figure 5. Input-Output (I/O) mapping and matrix representations of quantum gates: (a) (2,2) Feynman gate, (b) (2,2) Swap gate, and (c) (3,3) Toffoli gate.

In Figure 5, the matrix representation is equivalent to the input-output (I/O) mapping representation of quantum gates, as follows. If one considers each row in the input side of the I/O map in Figure 5 as an input vector represented by the natural binary code of 2^{index} with row index starting from 0, and similarly for the output row of the I/O map, then the matrix transforms the input vector to the corresponding

output vector by transforming the code for the input to the code for the output. For example, the following matrix equation is the I/O mapping using the Feynman matrix from Figure 5a:

$$[\text{Feynman matrix}][\text{input code}] = [\text{output code}]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

One notes from this example, that the Feynman gate, and similarly all quantum gates in Figure 5, are merely permuters, i.e. they produce output vectors which are permutations of the input vectors.

Figure 6 shows the quantum evolution matrices for blocks B1 (also B2) and B3 in Figure 4, respectively.

$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
(a)	(b)

Figure 6. Quantum transformations for the reversible (7,7) circuit in Figure 4: (a) Input mapping block B1 (also B2), and (b) output mapping block B3.

5 QUANTUM COMPUTING

Although the gates in Figure 5 are merely permuters, not all quantum gates do simple permutations (Nielsen and Chuang 2000). The mapping of a set of inputs into any set of outputs in Figure 4 can be obtained in general using quantum computing. The following discussion explains the general principles of quantum computing, and we follow the standard notation that is used in quantum mechanics from (Dirac 1930).

Definition 1. A binary quantum bit, or qubit, is a binary quantum system, defined over the Hilbert space H_2 with a fixed basis $\{|0\rangle, |1\rangle\}$.

Definition 2. In binary quantum logic system, qubit-0 and qubit-1 are defined as follows:

$$\text{qubit-0} = |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{qubit-1} = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Figure 7 illustrates the process of evolving the input binary qubits using the corresponding quantum circuits. Let us evolve the input binary

$$\text{qubit } |11\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \text{ where the tensor}$$

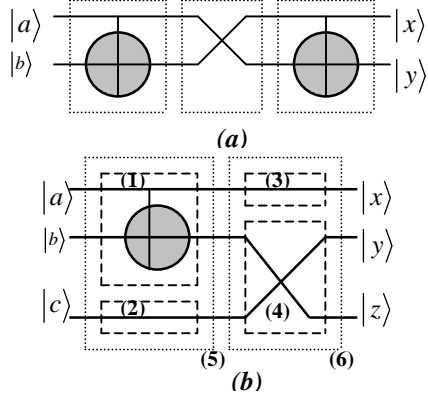


Figure 7. Quantum logic circuits.

product \otimes gives the corresponding binary natural code, using the serially interconnected quantum circuit in Figure 7a, which is composed of a serial interconnection of two Feynman gates (Figure 2a) connected by a swap gate (Figure 2c). The evolution of the input qubit can be viewed in two equivalent perspectives. One perspective is to evolve the input qubit step by step using the serially interconnected gates. The second perspective, is to evolve the input qubit using the total quantum circuit at once, since the total evolution transformation $[M_{net}]$ is equal to the multiplication of the individual evolution matrices $[M_q]$ that correspond to the individual quantum primitives: $\therefore [M_{net}]_{serial} = \prod_q [M_q]$.

Perspective #1:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective #2:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Thus, the quantum circuit shown in Figure 7a evolves the qubit $|11\rangle$ into the qubit $|01\rangle$.

The quantum circuit in Figure 7b is composed of a serial interconnect of two parallel circuits as follows: dashed boxes ((1),(2)) and ((3),(4)) are parallel interconnected, and dotted boxes (5) and (6) are serially interconnected. The total evolution transformation $[M_{net}]$ of the total parallel-interconnected quantum circuit is equal to the tensor (Kronecker) product of the individual evolution matrices $[M_q]$ that correspond to the individual quantum primitives: $\therefore [M_{net}]_{parallel} = \otimes [M_q]$. Thus, analogously to

the operations of the circuit in Figure 7a, the evolution of the input qubit, in Figure 7b, can be viewed in two equivalent perspectives, respectively. One perspective is to evolve the input qubit stage by stage. The second perspective is to evolve the input qubit using the total quantum circuit at once. Let us evolve the input binary qubit $|111\rangle$ using the quantum circuit in Figure 7b. The evolution matrices of the parallel-interconnected dashed boxes in (5) and (6), are as follows (where the symbol \parallel means parallel connection):

$$\text{input} = |1\rangle \otimes |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = [00000001]^T$$

The evolution matrix for (5) = (1) \parallel (2) is:

$$\text{Feynman} \otimes \text{Wire} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & & & & & & \\ & 1 & 0 & & & & & \\ & & 1 & 0 & & & & \\ & & & 1 & 0 & & & \\ & & & & 1 & 0 & & \\ & & & & & 1 & 0 & \\ & & & & & & 1 & 0 \\ & & & & & & & 1 \end{bmatrix}$$

The evolution matrix for (6) = (3) \parallel (4) is:

$$\text{Wire} \otimes \text{Swap} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Perspective #1: $\text{input} \Rightarrow (5) \Rightarrow \text{output}_1, \text{input}_2 (= \text{output}_1) \Rightarrow (6) \Rightarrow \text{output}_2$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |110\rangle$$

Perspective #2: $\text{input} \Rightarrow ((6)(5)) \Rightarrow \text{output}_2$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |110\rangle$$

Thus, the quantum circuit shown in Figure 7b evolves the qubit $|111\rangle$ into the qubit $|110\rangle$.

By applying this formalism to the quantum matrices from Figure 6, the reversible MRA circuit of Figure 4 is represented compactly by the following transformations:

$$[B_1] |ax_2x_1\rangle = |R_1P_1f_1\rangle \quad (1)$$

$$[B_2] |x_1x_3a\rangle = |f_2P_2R_2\rangle \quad (2)$$

$$[B_3] |f_10f_2\rangle = |G_1FG_2\rangle \quad (3)$$

where, in equation (3), the qubit $|0\rangle$ is used to generate the AND operation in block B3 (Toffoli gate from Figure 2b) in Figure 4, and $|\alpha\beta\gamma\rangle = |\alpha\rangle \otimes |\beta\rangle \otimes |\gamma\rangle$, where α , β , and γ are single binary qubits.

6 CONCLUSIONS AND FUTURE WORK

Reversible realization of Modified Reconstructability Analysis (MRA) decomposition and its quantum computation are presented. A comprehensive treatment of reversible MRA and its quantum computing with supplementary materials is provided in (Al-Rabadi 2002).

Future work will involve the investigation of other possible reversible realizations of binary and multiple-valued MRA decompositions of logic circuits and their corresponding quantum computations. The use of the natural parallelism of quantum entanglement for the realization of MRA-based circuits will also be investigated.

7 REFERENCES

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., M. Zwick, and M. Perkowski 2002. "A Comparison of Enhanced Reconstructability Analysis and Ashenurst-Curtis Decomposition of Boolean Functions". Accepted to WOSC/IGSS'2002 Conference.

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.

Bennett, C. 1973. "Logical Reversibility of computation". *IBM Journal of Research and Development*, 17, pp.525-532.

Dirac, P. 1930. *The Principles of Quantum Mechanics*, first edition, Oxford University Press.

Fredkin, E. and T. Toffoli 1982. "Conservative Logic". *Int. Journal of Theoretical Physics*, 21, pp.219-253.

Hurst, S. L. 1978. *Logical Processing of Digital Signals*. Crane Russak and Edward Arnold, London and Basel.

Kerntopf, P. 2000. "A Comparison of Logical Efficiency of Reversible and Conventional Gates". *Proc. On 3rd Symposium on Logic, Design and Learning*, Portland, Oregon.

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

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.

Landauer, R. 1961. "Irreversibility and heat generation in the computational process". *IBM Journal of Research and Development*, 5, pp.183-191.

Nielsen, M. and I. Chuang, 2000. *Quantum Computation and Quantum Information*. Cambridge University Press.

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

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, 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.