

# 2-BIT EX-OR LINK BASED REVERSIBLE MULTIPLIER FOR LOW POWER DSP APPLICATIONS

By

M. BHARATHI \*

NEELIMA KOPPALA \*\*

\*.\*\* Assistant Professor, Department of ECE, Sree Vidyanikethan Engineering College (Autonomous), Tirupati, Andhra Pradesh, India.

### ABSTRACT

The multiplier in any arithmetic unit dissipates significant amount of energy as large number of computations are required if the number of bits in the design increase. Thus, if efficient reversible logic is used, then the power consumption can be reduced drastically as the information bits are not lost in case of reversible computation. This paper focuses on the design of two-bit multiplier using a synthesis approach called Exorlink which reduces quantum cost compared to the technique Disjoint Sum of Products (DSOP). The design is coded in VHDL (VHSIC Hardware Description Language), simulated using ISim and synthesized using Xilinx ISE 10.1i for the device Spartan3E FPGA.

Keywords: Reversible Multiplier, Disjoint Sum of Products (DSOP), Exorlink, Quantum Cost.

### INTRODUCTION

With the high demand of low power digital systems, energy dissipation in the digital system is one of the limiting factors. Reversible logic is one of the alternates to reduce heat/energy dissipation (R. Landauer, pg.1961) in the digital circuits and has a very significant importance in bioinformatics, optical information processing, CMOS circuit design, Nanotechnology, etc.

A reversible logic circuit (M.Bharathi, K.Neelima, 2012) (V.K.Puri, 2006, A.DeVos, 2010) has important features like usage of minimum number of reversible gates, garbage outputs and constant inputs. Among these, the minimization of the garbage outputs is one of the major goals in reversible logic design and synthesis. Each Reversible gate has a cost associated with it called as Quantum Cost, which is defined by the total number of elementary quantum gates needed to realize the given function. The Quantum Cost (M. Bharathi, K. Neelima, 2012) of a circuit is proportional to the heat dissipation in the circuit. For any function realized using various types of gates, the Quantum Cost is given by the sum of the Quantum Costs of all the gates used to realize that function.

Arithmetic calculation is the one of the important part of any digital hardware. The squaring calculation has its own

advantages in different fields like digital signal processing: Convolution, deconvolution, cryptography etc. where numbers of times we have to calculate fast square of the number. In order to calculate the square of the binary numbers, there are number of faster reversible multipliers by which we can calculate the square of the numbers. The reversible multipliers proposed in the literature are generally based on the recursive method. But, to design any of the digital hardware, the circuit complexity is one of the important measures which directly correlate with the delay. So, the main motivation behind this work is to investigate the implementation of efficient and low power squaring circuit architecture for the digital hardware industries (V.K.Puri, 2006).

In this paper, we are proposing a reversible 2-bit Multiplier circuit for the fast squaring calculation instead of using multiplier for the squaring which increases the complexity and the delay in the circuit. The design of new 2-bit binary multiplier circuit is used in most of the digital signal processing hardware using standard reversible gates. The designed multiplier circuit is having less garbage outputs, constant inputs, Quantum cost and Total logical calculation i.e. less quantum cost as compared to the Disjoint Sum of Products method (DSOP). The simulation results and quantized results are also shown in the paper

which show greatest improvement in the design against the previous methodology.

In section 1, the existing method called DSOP method of computing quantum cost is discussed. Section 2 deals with the efficient method, Exorlink for evaluating the quantum cost. In section 3, the results are discussed and compared. Last Section concludes the paper.

## 1. DSOP Method

The basic differences between the various definitions are given below.

A Boolean expression is an algebraic clause representing a relationship among a set of Boolean valued literals (P. Kemtopf (2002), Ning Song and Marek A. Perkowski 1996). A Boolean function can be represented by an equation containing Boolean expressions.

$$F = x' y' z' + z \quad (1)$$

A Boolean expression containing a set of literals conjuncted together (i.e. ANDed) is called a Product/Cube.

$$F = x' y' z' \quad (2)$$

The inclusive OR function in Boolean algebra is called Disjunctive Sum.

$$F = x' + z \quad (3)$$

Two or more AND functions are ORed together to form a Sum-of-products expression. In this form, the product terms may or may not cover a common minterm.

$$F = w y' + x + y \cdot z \quad (4)$$

Two cubes are said to be disjoint if their intersection of the set of minterms is null i.e., Disjoint cubes.

$$F(w,x,y,z) = w'x \text{ and } G(w,x,y,z) = x'z \quad (5)$$

The distance of two min terms is the number of variables for which the corresponding literals have different sets of values.

i.e., the distance between 0011 and 0111 is 1

A cube cover corresponds to the familiar sum-of-products representation in Boolean Algebra with each cube corresponding to a product term and the function being the sum (logical OR) of those terms. Since a single cube represents a Boolean function on its own, the set of cubes

representing a Boolean function need not necessarily be unique (C.H.Bennett, 1998). When each pair of cubes in the cover is disjoint, i.e., when the two cubes do not share a common minterm, is the case of Interest. Such a situation corresponds to a Disjoint Sum-of-Products (DSOP) expression (M.Bharathi, Neelima Koppaia, 2014).

The example for conversion of SOP Expression to DSOP Expression (C.H.Bennett, 2014, Lokesh Shiva kumaralah and Mitchell A. Thornton) is as shown in Figure 1.

Let the Sum of Product Expression (Ning Song and Marek A. Perkowski 1996) be

$$F = abc' + bc'd + abc'd + a'bcd + ab'c + acd' \quad (6)$$

Then the Disjoint Sum of Product Expression from Figure 1 is given by eq. 7.

$$F = a'c'd' \oplus a'b'c'd \oplus abc'd \oplus a'bcd \oplus abcd' \oplus ab'c \quad (7)$$

Hence if no two product terms cover a common minterm, they are called a disjoint-sum-of-products (DSOP).

## 2. Exorlink Method

Exorlink Operation between two terms is as shown below

- Replace the bit changed with 'x' or '0' or '1' depending on the bits in the two terms (Lokesh Shiva Kumaraiah and Mitchell A. Thornton).
- The left side of the changed bit is replaced with the bits in the first term.
- The right side of the changed bit is replaced with the bits in the second term.

$$\text{Eg.: } 1011 \otimes 0111 = x111 \oplus 1x11$$

$$x0x1 \otimes 1001 = 0001 \oplus x01$$

The Reversible Circuit's (M.Bharathi, K.Neelima, 2012, Mitchell Aaron Thornton, Rolf Drechsler and D.Michael Miller) DSOP and Exorlink equations are as shown in Figure 2 and Figure 3 respectively (M.Bharathi, Neelima Koppala,

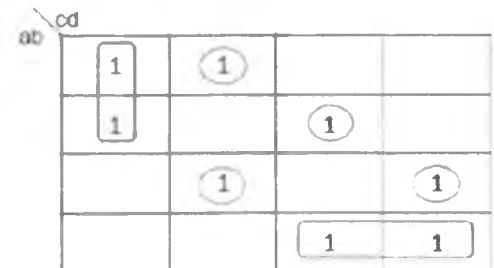


Figure 1. DSOP K-map for eq.6

2014) and their corresponding Quantum Cost is given below.

The DSOP Expression is given by

$$F = a'c'd' \oplus a'b'c'd \oplus abc'd \oplus a'bcd \oplus abcd' \oplus ab'c \quad (8)$$

The Exorlink expression is given by

$$F = a'c' \oplus bd \oplus ac \quad (9)$$

The process of converting the DSOP form to Exorlink form is shown in Figure 4 which includes 7 steps excluding the calculation of Quantum Cost.

### 3. Results and Discussion

A Full Adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs and three outputs (Ning Song and Marek A. Perkowski 1996). Two of the input variables denoted by A and B,

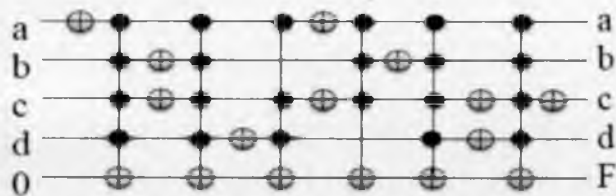


Figure 2. Reversible circuit for DSOP

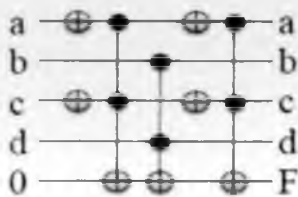


Figure 3. Reversible Logic for Exorlink

- Algorithm for conversion of DSOP to Exorlink
1. Convert the terms in DSOP equation into binary form.
  2. Find the distance between all possible pair of terms.
  3. Select a pair of terms having minimum distance (1 or 2 or 3) if exist.
  4. Choose the order of the terms to perform exorlink operation such that the resultant terms will have minimum distance with the other terms.
  5. Perform Exorlink operation and replace the xorlinked terms with resultant terms and if any pair has distance 1, perform exorlink operation on the pair.
  6. If not, check for links in k-map.
  7. a) If links are not formed between all the pairs, repeat from step 2.  
b) If links are formed between all pairs, write the corresponding exorlink equation.

Figure 4. Algorithm for Conversion of reversible circuit from DSOP to Exorlink

represent the two significant bits to be added. The third input, represents the carry from the previous lower significant position.

### 2 Bit Multiplier for Low Power DSP Application

Figure 5 shows the block diagram of 2-bit multiplier where A0, A1 --- multiplicand bits, B0, B1 --- multiplier bits and Z3, Z2, Z1, Z0 – output bits.

The truth table of 2-bit Multiplier is shown in Table 1.

From Table 1, the actual expressions relating to the output bits and input bits of multiplier and multiplicand are derived using SOP method in K-Maps. The related expressions are

$$SOP: Z3 = A1A0B1B0$$

$$Z2 = A1A0'B1 + A1B1A0'$$

$$Z1 = A1A0B0 + A1B1B0 + A1A0B1 + A0B1B0$$

$$Z0 = A0B0$$

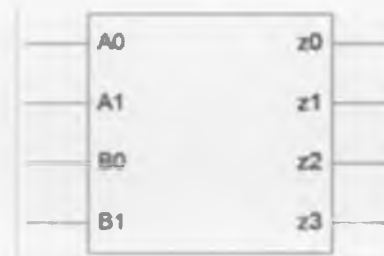


Figure 5. Block diagram of 2-bit multiplier

INPUTS				OUTPUTS			
A1	A0	B1	B0	Z3	Z2	Z1	Z0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Table 1. 2-bit Multiplier Truth Table

The corresponding DSOP and Exorlink based reversible circuit representations are shown below. Figure 6 shows the DSOP K-map for Z3 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding DSOP reversible Circuit implementation is shown in Figure 7. The quantum cost of implementation is found to be 29.

Z3 :

DSOP: A1A0B1B0

Quantum Cost = 29

Figure 8 shows the Exorlink K-map for Z3 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding Exorlink reversible Circuit implementation is shown in Figure 9. The quantum cost of implementation is found to be 29.

Exorlink: A1A0B1B0

Quantum Cost = 29

Figure 10 shows the DSOP K-map for Z2 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding DSOP reversible Circuit implementation is shown in Figure 11. The quantum cost of implementation is found to be

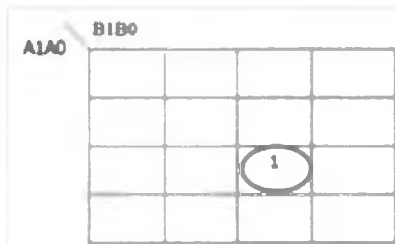


Figure 6. DSOP K-map for Z3



Figure 7. DSOP Reversible Circuit for Z3

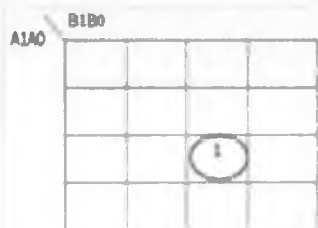


Figure 8. Exorlink K-map for Z3

46.

Z2 :

DSOP: A1A0'B1 ⊕ A1A0B1B0'

Quantum Cost = 4 + 13 + 29 = 46

Figure 12 shows the Exorlink K-map for Z2 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding Exorlink reversible Circuit implementation is shown in Figure 13. The quantum cost of implementation is found to be 34.

Exorlink: A1A0B1B0 ⊕ A1B1

Quantum Cost = 29 + 5 = 34

Figure 14 shows the DSOP K-map for Z1 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding DSOP reversible Circuit implementation is shown in Figure 15. The quantum cost of implementation is found to be 92.

Z1 :

DSOP: A1B1'B0 ⊕ A1'A0B1B0 ⊕ A1A0'B1B0 ⊕ A0B1B0'

Quantum Cost = 8 + 13 + 29 + 29 + 13 = 92

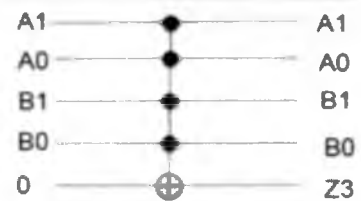


Figure 9. Exorlink Reversible Circuit for Z3

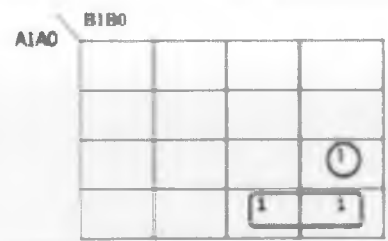


Figure 10. DSOP K-map for Z2

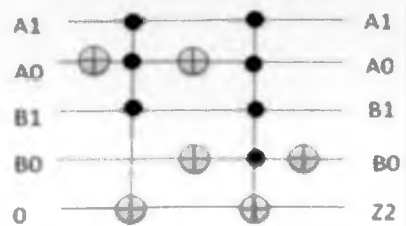


Figure 11. DSOP Reversible Circuit for Z2

Figure 16 shows the Exorlink K-map for Z1 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding Exorlink reversible Circuit implementation is shown in Figure 17. The quantum cost of implementation is found to be 10.

Exorlink:  $A1B0 \oplus A0B1$

Quantum Cost =  $5 + 5 = 10$

Figure 18 shows the DSOP K-map for Z0 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding DSOP reversible Circuit implementation is shown in Figure 19. The quantum cost of implementation is found to be 5.

Z0:

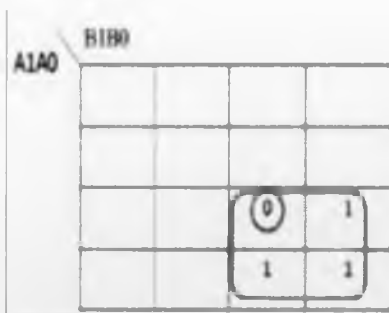


Figure 12. Exorlink K-map for Z2

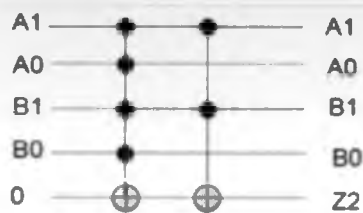


Figure 13. Exorlink Reversible Circuit for Z2

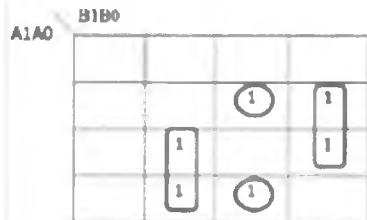


Figure 14. DSOP K-map for Z1

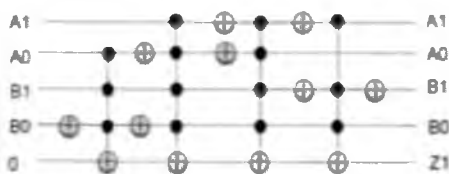


Figure 15. DSOP Reversible Circuit for Z1

DSOP: A0B0

Quantum Cost = 5

Figure 20 shows the Exorlink K-map for Z0 as a function of A1A0 and B1B0 inputs of multiplier and the corresponding Exorlink reversible Circuit implementation is shown in Figure 21. The quantum cost of implementation is found to be 5.

Exorlink: A0B0

Quantum Cost = 5

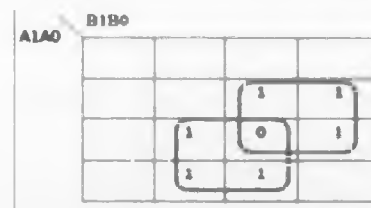


Figure 16. Exorlink K-map for Z1

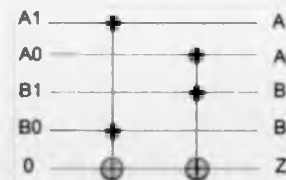


Figure 17. Exorlink Reversible Circuit for Z1

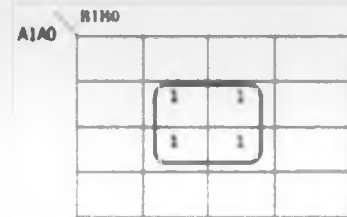


Figure 18. DSOP K-map for Z0

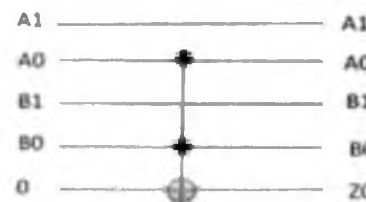


Figure 19. DSOP Reversible Circuit for Z0

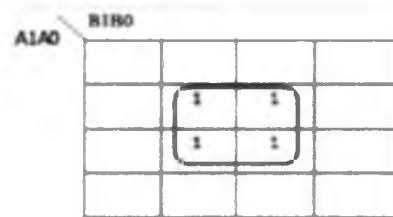


Figure 20. Exorlink K-map for Z0

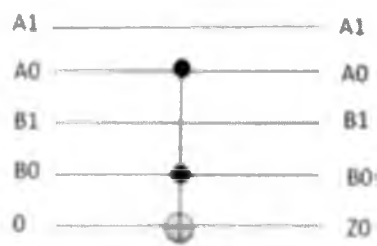


Figure 21. Exorlink Reversible Circuit for Z0

2-bit Reversible Multiplier	Total Quantum Cost
DSOP	$29 + 46 + 92 + 5 = 172$
EXORLINK	$29 + 34 + 10 + 5 = 78$

Table 2. Total Quantum Cost for 2-bit Multiplier

Thus from Table 2, it is clear that there is a drastic decrease in the Quantum Cost for Exorlink when compared to DSOP for a 2-bit reversible multiplier. Hence Exorlink based multiplier design is the one with more power efficient operation, as it can link any two cubes in an array of cubes having any arbitrary distance. Also, when compared to the DSOP logic, the output bits of multiplier realized by Exorlink can have fewer gates, fewer connections and less Quantum Cost.

### Conclusion

Exorlink logic employed in the design of multiplier proves to be the more power efficient method in Reversible logic synthesis. It can link any two cubes in an array of cubes of an arbitrary distance. Hence by using this method, a large number of variables in the expression gets minimized, thereby reducing the Quantum Cost compared to the Disjoint- Sum of Products (DSOP). Employing Reversible Logic and optimizing the Quantum Cost by using Exorlink technique reduces the heat dissipation in the circuits which leads to new approach for Integrated Circuit design and helps in continuing Moore's Law. The designed multiplier circuit is proved to have less garbage outputs, constant inputs, Quantum cost and Total logical calculation i.e. less quantum cost as compared to the Disjoint Sum of Products (DSOP) method.

### References

- [1]. M.Bharathi, Neelima Koppala, (2014). "Efficient Approach to Optimize Quantum Cost for Combinational Reversible Circuits", *International Journal of Research in computer Applications and Robotics*, Vol.2, Issue.7, pp.: 1-9, ISSN – 2320-7345.
- [2]. M. Bharathi , K. Neelima, (2012). "Scope Of Reversible Engineering At Gate-Level: Fault-Tolerant Combinational Adders", *International Journal of VLSI design & Communication Systems (VLSICS)*, Vol. 3, No. 2, Page No.: 85-98, ISSN – 0976 -1357(Online), 0976-1537(Print).
- [3]. A.DeVos, (2010). "Reversible Computing: Fundamentals, Quantum Computing, and Applications", *Weinheim: Wiley-VCH*.
- [4]. P. Kemtopf, (2002). "Synthesis of multipurpose reversible logic gates" *Euromicro Symposium on Digital System Design (DSD'02)*, pp.259-267.
- [5]. C.H.Bennett, (1998). "Notes on the History of Reversible Computation", *IBM Journal of Research and Development*, Vol. 32, pp. 16-23.
- [6]. Lokesh Shivakumaraiah and Mitchell A. Thornton (2002). "Computation of Disjoint Cube Representations Using a Maximal Binat Variable Heuristic", Department of Electrical and Computer Engineering, Mississippi State University.
- [7]. Mitchell Aaron Thornton, Rolf Drechsler and D.Michael Miller, (2001). "Spectral Techniques in VLSI CAD", *Kluwer Academic publishers*. ISBN 0792374339.
- [8]. Ning Song and Marek A. Perkowski, (1996) "Minimization of Exclusive Sum-of-Products Expressions for Multiple-Valued Input, Incompletely Specified Functions", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No.4, April.
- [9]. V.K.Puri, (2006). "Digital Electronics Circuits and Systems", *Tata McGraw-Hill*.
- [10]. R. Landauer, (1961). "Irreversibility and heat generation in the computational process", *IBM J. Res. Develop.*, Vol. 5, pp. 261-268.

## RESEARCH PAPERS

### ABOUT THE AUTHORS

*Bharathi is currently working as an Assistant Professor In the ECE Department of Sree Vidyanikethan Engineering College, Tirupati. She has completed M.Tech In VLSI Design, in Satyabhama University. Her research areas are Digital System Design, VLSI Signal Processing.*



*Neellma Koppala, is currently working as an Assistant Professor in ECE Department of Sree Vidyanikethan Engineering College, Tirupati. She has completed M.Tech in VLSI Design, in Satyabhama University. Her research areas are Digital System Design, VLSI Signal Processing.*

