GF(4) Based Synthesis of Quaternary Reversible/Quantum Logic Circuits

MOZAMMEL H. A. KHAN[1] AND MAREK A. PERKOWSKI[2]

[1]*Department of Computer Science and Engineering, East West University*
43 Mohakhali, Dhaka 1212, BANGLADESH,
Email: mhakhan@ewubd.edu, mhakhan59@yahoo.com
[2]*Department of Electrical and Computer Engineering, Portland State University*
1900 SW 4th Avenue, Portland, OR 97201, USA,
Email: mperkows@ee.pdx.edu

Galois field sum of products (GFSOP) has been found to be very promising for reversible/quantum implementation of multiple-valued logic. In this paper, we show nine quaternary Galois field expansions, using which quaternary Galois field decision diagrams (QGFDD) can be constructed. Flattening of the QGFDD generates quaternary GFSOP (QGFSOP). These QGFSOP can be implemented as cascade of quaternary 1-qudit gates and multi-qudit Feynman and Toffoli gates. We also show the realization of quaternary Feynman and Toffoli gates using liquid ion-trap realizable 1-qudit gates and 2-qudit Muthukrishnan-Stroud gates. Besides the quaternary functions, this approach can also be used for synthesis of encoded binary functions by grouping 2-bits together into quaternary value. For this purpose, we show binary-to-quaternary encoder and quaternary-to-binary decoder circuits using quaternary 1-qudit gates and 2-qudit Muthukrishnan-Stroud gates.

Keywords: encoded binary logic, multiple-valued logic, quaternary logic, quaternary Galois field sum of products, quaternary Galois field decision diagram, quantum logic, reversible logic

1 INTRODUCTION

Multiple-valued quantum logic synthesis, specially, ternary logic synthesis has become popular in the recent years [1-7]. Among them [3, 5, 6] used cascades of ternary reversible gates like Feynman and Toffoli gates to realize ternary logic functions. The advantage of this approach is that ternary logic functions having many input variables can be easily expressed as ternary Galois field sum of products (TGFSOP) expression and can be realized using cascade of ternary Feynman and Toffoli gates [3, 5, 6].

Though a considerable number of good works have been done on reversible/quantum ternary logic synthesis, it has the limitation that conventional binary logic functions cannot be very easily represented using the ternary base and the developed methods are applicable only for logic functions expressed in ternary base. A very promising alternative may be quaternary logic, using which, besides quaternary logic functions, binary logic functions can be expressed by grouping 2-bits together into quaternary values. For a Hilbert space of N dimension, a binary quantum system requires $n_2 = \log_2 N$ qubits (quantum bits), whereas a quaternary quantum system requires $n_4 = \log_4 N = \frac{\log_2 N}{\log_2 4} = \frac{\log_2 N}{2}$ qudits (quantum digits). Therefore, we have

$$\frac{n_4}{n_2} = \frac{\frac{\log_2 N}{2}}{\log_2 N} = \frac{1}{2} \qquad (1)$$

From (1), we find that 2-bit encoded quaternary realization of binary logic functions requires 1/2 times qudits than the qubits needed for the binary realization. Assuming qubit implementation and qudit implementation are of the same technological complexity, 2-bit encoded quaternary realization of binary function is efficient than the binary realization. However, for this purpose, circuits for input encoding and output decoding will be needed and we show such encoder and decoder circuits in Section 11.

From the experience of ternary logic, it can be expected that quaternary logic functions having many input variables can also be easily expressed as quaternary Galois field sum of products (QGFSOP) expression and can be realized using cascade of quaternary Feynman and Toffoli gates. But the question arises whether quaternary Feynman and Toffoli gates are physically realizable or not. Muthukrishnan and Stroud [8] proposed a family of 2-qudit multiple-valued ($d \geq 2$) quantum gates, which is theoretically realizable in liquid ion-trap technology. The macro-level quaternary Feynman and Toffoli gates can be realized on the top of Muthukrishnan-Stroud gates as discussed in Sections 8 and 9.

The advantages of quaternary logic and physical realizability of quaternary Feynman and Toffoli gates require that methods should be developed for synthesizing quaternary logic functions as a cascade of quaternary Feynman and Toffoli gates. However, as far as we know, no work has yet been done on expressing quaternary logic function as QGFSOP expression and realizing the QGFSOP expression as a cascade of quaternary Feynman and Toffoli gates. Realization of QGFSOP expression as a cascade of quaternary Feynman and Toffoli gates is discussed in Section 10. But the main difficulty arises with expressing quaternary logic function as QGFSOP expression. The standard technique used in binary and ternary cases is that a set of expansions is used to construct optimum decision diagram (DD) and the DD is flattened to determine the minimized GFSOP expression. In the QGF case the same technique will be useful. But, as far as we know, no such quaternary Galois field expansion (QGFE) and algorithm for constructing optimum quaternary Galois field decision diagram (QGFDD) have yet been reported in the literature.

In this paper, we have developed nine QGFEs using which we can construct optimum QGFDD. However, constructing optimum QGFDD requires efficient algorithm for selecting QGFE for each variable and algorithm for variable ordering. In this paper we concentrated only on developing QGFEs and showing their usefulness. Development of the above mentioned algorithms will be the focus of future research. By flattening the optimum QGFDD we can generate minimized QGFSOP expression. This QGFSOP expression can be realized as cascade of quaternary 1-qudit gates and multi-qudit Feynman and Toffoli gates.

2 GALOIS FIELD

A field is a set F with two binary operations - addition (denoted by +) and multiplication (denoted by · or absence of any operator) are defined, which satisfies the following axioms:

(A1) $a + (b + c) = (a + b) + c$ (associative law for addition)
(A2) $a + b = b + a$ (commutative law for addition)
(A3) There is an element 0 (zero) such that $a + 0 = a$ for all a
(A4) For any a, there is an element $(-a)$ such that $a + (-a) = 0$
(M1) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ (associative law for multiplication)
(M2) $a \cdot b = b \cdot a$ (commutative law for multiplication)
(M3) There is an element 1 (not equal to 0) such that $a \cdot 1 = a$ for all a
(M4) For any $a \neq 0$, there is an element a^{-1} such that $a \cdot a^{-1} = 1$
(D) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ (distributive law)

If p is a prime number, then the *integers mod p* form a Galois field (also known as finite field): its elements are the congruence classes of integers mod p, with addition and multiplication induced from integer mod operations.

We can construct a Galois filed with $q = p^r$ elements [abbreviated as GF(*q*) or GF(p^r)], where *p* is a prime. The construction of the field is as follows. First, let F_p be the field of integers mod *p*. Now choose an irreducible polynomial *f*(*X*) of degree *r* over F_p as below:

$$f(X) = X^r + C_{r-1}X^{r-1} + \cdots + C_1X + C_0 \tag{2}$$

Now, the elements of F_q are all the expressions of the form

$$x_0 + x_1a + x_2a^2 + \cdots + x_{r-1}a^{r-1} \tag{3}$$

where *a* is required to satisfy *f*(*a*) = 0, and $x_0, \cdots, x_{r-1} \in F_p$. The number of expressions of the form of (3) is p^r, since there are *p* choices for each of the *r* coefficients $x_0, \cdots, x_{r-1}$. Adding these expressions is straightforward. To multiply, we have to consider that

$$a^r + C_{r-1}a^{r-1} + \cdots + C_1a + C_0 = 0 \tag{4}$$

For GF(4), $q = 2^2$, where *p* = 2 and *r* = 2. Then (2) reduces to

$$f(X) = X^2 + C_1X + C_0 \tag{5}$$

Equation (5) remains irreducible if we take $C_1 = C_0 = 1$ and the equation reduces to

$$f(X) = X^2 + X + 1 \tag{6}$$

Also equation (3) reduces to

$$x_0 + x_1a \tag{7}$$

where, $x_0, x_1 \in \{0,1\}$. Now, putting different values of x_0 and x_1 in (7), we can find the four elements of F_4 as follows:

$$\begin{gathered} 0 + 0 \cdot a = 0 \\ 0 + 1 \cdot a = a \\ 1 + 0 \cdot a = 1 \\ 1 + 1 \cdot a = 1 + a \end{gathered}$$

That means, $F_4 = \{0,1, a, a+1\}$.

The additions of these elements are mod 2 additions as follows:

$$
\begin{gathered}
0+0=0 \\
0+1=1 \\
0+a=a \\
0+(1+a)=1+a \\
1+0=1 \\
1+1=0 \\
1+a=1+a \\
1+(1+a)=a \\
a+0=a \\
a+1=1+a \\
a+a=0 \\
a+(1+a)=1 \\
(1+a)+0=1+a \\
(1+a)+1=a \\
(1+a)+a=1 \\
(1+a)+(1+a)=0
\end{gathered}
$$

For GF(4), equation (4) reduces to

$$a^2 + C_1 a + C_0 = 0 \tag{8}$$

Taking $C_1 = C_0 = 1$, equation (8) reduces to

$$a^2 + a + 1 = 0 \tag{9}$$

From (9), we have using mod 2 addition

$$a^2 = 1 + a \tag{10}$$

The multiplications of the elements are as follows:

$$
\begin{gathered}
0 \cdot 0 = 0 \\
0 \cdot 1 = 0 \\
0 \cdot a = 0 \\
0 \cdot (1+a) = 0 \\
1 \cdot 0 = 0 \\
1 \cdot 1 = 1 \\
1 \cdot a = a
\end{gathered}
$$

TABLE 1
GF(4) operations.

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

$$1\cdot(1+a)=1+a$$
$$a\cdot 0=0$$
$$a\cdot 1=a$$
$$a\cdot a=a^2=1+a$$
$$a\cdot(1+a)=a+a^2=a+1+a=1$$
$$(1+a)\cdot 0=0$$
$$(1+a)\cdot 1=1+a$$
$$(1+a)\cdot a=a+a^2=1$$
$$(1+a)\cdot(1+a)=1+2a+a^2=1+2a+1+a=a$$

Now, taking a = 2, $F_4=\{0,1,2,3\}$ and the addition and multiplication over GF(4) are as shown in Table 1.

GF(4) is also known as quaternary Galois field (QGF).

3 QUATERNARY GALOIS FIELD SUM OF PRODUCTS EXPRESSION

In quaternary quantum logic system the unit of memory (information) is a qudit (quantum digit). Logic values of 0, 1, 2, and 3 are represented by a set of distinguishable different states of an object that represent the qudit. Quantum gates carry around and manipulate the quantum information. Any transformation of the qudit state represented by a $4^n\times 4^n$ unitary matrix specifies a valid *n*-qudit quantum gate. There are many such non-trivial *n*-qudit gates. In this paper, we consider only the permutation quantum gates, where the characteristic unitary matrix of the gate has only one 1 in each row and column and the other elements of the matrix are 0. The advantages of permutation quantum gates are that they are reversible gates and their outputs can be described using both truth table and Galois field expression.

There are 4! = 24 possible permutations of 0, 1, 2, and 3. Therefore, there are 24 possible reversible truth tables and corresponding permutation unitary matrices for 1-qudit transformations resulting into 24 possible 1-qudit permutation/reversible gates. We represent these 1-qudit

TABLE 2
Basic quaternary reversible-literals.

Input	x	x+1	x+2	x+3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0
Input	2x	2x+1	2x+2	2x+3
0	0	1	2	3
1	2	3	0	1
2	3	2	1	0
3	1	0	3	2
Input	3x	3x+1	3x+2	3x+3
0	0	1	2	3
1	3	2	1	0
2	1	0	3	2
3	2	3	0	1
Input	x^2	x^2+1	x^2+2	x^2+3
0	0	1	2	3
1	1	0	3	2
2	3	2	1	0
3	2	3	0	1
Input	2x^2	2x^2+1	2x^2+2	2x^2+3
0	0	1	2	3
1	2	3	0	1
2	1	0	3	2
3	3	2	1	0
Input	3x^2	3x^2+1	3x^2+2	3x^2+3
0	0	1	2	3
1	3	2	1	0
2	2	3	0	1
3	1	0	3	2

transformations by 24 basic quaternary reversible-literals as shown in Table 2. As it is very difficult to adopt 24 different symbols for the literals, we represent a literal by the QGF expression representing the literal. A basic quaternary reversible-literal multiplied by the constant 2 or 3 produces another basic quaternary reversible-literal as shown in Tables 3 and 4.

Product of two or more basic quaternary reversible-literals is called a QGF product (QGFP). For example, $(2x+2)(3x^2+2)(2x^2)$ is a QGFP. Sum of two or more QGFP is called a QGFSOP expression. For example, $(2x+2)(3x^2+2) + (3x+1)(2x) + x$ is a QGFSOP expression.

TABLE 3
Product of basic quaternary reversible-literal and the constant 2.

literal	x	x+1	x+2	x+3
2(literal)	2x	2x+2	2x+3	2x+1
literal	2x	2x+1	2x+2	2x+3
2(literal)	3x	3x+2	3x+3	3x+1
literal	3x	3x+1	3x+2	3x+3
2(literal)	x	x+2	x+3	x+1
literal	x^2	x^2+1	x^2+2	x^2+3
2(literal)	2x^2	2x^2+2	2x^2+3	2x^2+1
literal	2x^2	2x^2+1	2x^2+2	2x^2+3
2(literal)	3x^2	3x^2+2	3x^2+3	3x^2+1
literal	3x^2	3x^2+1	3x^2+2	3x^2+3
2(literal)	x^2	x^2+2	x^2+3	x^2+1

TABLE 4
Product of basic quaternary reversible-literal and the constant 3.

literal	x	x+1	x+2	x+3
3(literal)	3x	3x+3	3x+1	3x+2
literal	2x	2x+1	2x+2	2x+3
3(literal)	x	x+3	x+1	x+2
literal	3x	3x+1	3x+2	3x+3
3(literal)	2x	2x+3	2x+1	2x+2
literal	x^2	x^2+1	x^2+2	x^2+3
3(literal)	3x^2	3x^2+3	3x^2+1	3x^2+2
literal	2x^2	2x^2+1	2x^2+2	2x^2+3
3(literal)	x^2	x^2+3	x^2+1	x^2+2
literal	3x^2	3x^2+1	3x^2+2	3x^2+3
3(literal)	2x^2	2x^2+3	2x^2+1	2x^2+2

4 QUATERNARY GALOIS FIELD EXPANSIONS

In binary logic, a sum of product expression can be expanded using Shannon expansion for constructing binary decision diagram. An EXOR-sum of products expression can be expanded using Shannon, positive Davio, and negative Davio expansions for constructing Kronecker functional decision diagram. In a similar line of thinking, we develop quaternary Galois field expansions (QGFE), so that we can expand a QGFSOP expression and construct the corresponding quaternary Galois field decision diagram (QGFDD).

1-reduced Post literal (1-RPL) is defined as follows:

$$^{i}x = \begin{cases} 1 & \text{if } x = i \\ 0 & \text{otherwise} \end{cases}$$

Quaternary 1-RPLs are shown below:

$$^{0}x = x^3 + 1 \tag{11}$$

$$^{1}x = x^3 + x^2 + x \tag{12}$$

$$^{2}x = x^3 + 2x^2 + 3x \tag{13}$$

$$^{3}x = x^3 + 3x^2 + 2x \tag{14}$$

The cofactors of a quaternary function $f(x_1, x_2, \cdots, x_i, \cdots, x_n)$ with respect to the variable x_i are defined as follows:

$$\begin{aligned} f_0 &= f(x_1, x_2, \cdots, 0, \cdots, x_n) \\ f_1 &= f(x_1, x_2, \cdots, 1, \cdots, x_n) \\ f_2 &= f(x_1, x_2, \cdots, 2, \cdots, x_n) \\ f_3 &= f(x_1, x_2, \cdots, 3, \cdots, x_n) \end{aligned}$$

We will use sum of two or more cofactors (with or without multiplying by quaternary constant) and they will be called *composite cofactors* and designated as follows:

$$\begin{aligned} f_{01} &= f_0 + f_1 \\ f_{02} &= f_0 + f_2 \\ f_{03} &= f_0 + f_3 \\ f_{12} &= f_1 + f_2 \\ f_{13} &= f_1 + f_3 \\ f_{23} &= f_2 + f_3 \\ f_{012} &= f_0 + f_1 + f_2 \\ f_{013} &= f_0 + f_1 + f_3 \\ f_{023} &= f_0 + f_2 + f_3 \\ f_{123} &= f_1 + f_2 + f_3 \\ f_{0123} &= f_0 + f_1 + f_2 + f_3 \\ f_{1(2\cdot 2)(3\cdot 3)} &= f_1 + 2f_2 + 3f_3 \end{aligned}$$

$$f_{1(3\cdot 2)(2\cdot 3)} = f_1 + 3f_2 + 2f_3$$

Theorem 1 *Any quaternary function* $f(x_1, x_2, \cdots, x_i, \cdots, x_n)$ *can be expanded with respect to the variable* x_i *using the following expansion:*

$$f(x_1, x_2, \cdots, x_i, \cdots, x_n) = {}^0x_i f_0 + {}^1x_i f_1 + {}^2x_i f_2 + {}^3x_i f_3 \qquad (15)$$

Proof: If $x_i = 0$, then (15) reduces to

$$f(x_1, \cdots, 0, \cdots, x_n) = 1\cdot f_0 + 0\cdot f_1 + 0\cdot f_2 + 0\cdot f_3 = f_0 .$$

If $x_i = 1$, then (15) reduces to

$$f(x_1, \cdots, 1, \cdots, x_n) = 0\cdot f_0 + 1\cdot f_1 + 0\cdot f_2 + 0\cdot f_3 = f_1 .$$

If $x_i = 2$, then (15) reduces to

$$f(x_1, \cdots, 2, \cdots, x_n) = 0\cdot f_0 + 0\cdot f_1 + 1\cdot f_2 + 0\cdot f_3 = f_2 .$$

If $x_i = 3$, then (15) reduces to

$$f(x_1, \cdots, 3, \cdots, x_n) = 0\cdot f_0 + 0\cdot f_1 + 0\cdot f_2 + 1\cdot f_3 = f_3 .$$

Therefore, we have theorem 1. □

Theorem 2 *Any quaternary function* f *can be expanded with respect to the variable* x *using any of the following nine quaternary Galois field expansions (QGFE):*

QGFE 1:

$$f = f_0 + x(x+2)(x+3)f_{01} + x(x+1)(x+3)f_{02} + x(x+1)(x+2)f_{03}$$

QGFE 2:

$$f = f_1 + (x+1)(x+2)(x+3)f_{01} + x(x+1)(x+3)f_{12} + x(x+1)(x+2)f_{13}$$

QGFE 3:

$$f = f_2 + (x+1)(x+2)(x+3)f_{02} + x(x+2)(x+3)f_{12} + x(x+1)(x+2)f_{23}$$

QGFE 4:

$$f = f_3 + (x+1)(x+2)(x+3)f_{03} + x(x+2)(x+3)f_{13} + x(x+1)(x+3)f_{23}$$

QGFE 5:

$$\begin{aligned} f &= f_{012} + x^2 x f_{03} + (x^2+1)(x+1)\, f_{13} + (x^2+3)(x+2)\, f_{23} \\ &= f_{012} + 2x^2 3x f_{03} + (2x^2+2)(3x+3)\, f_{13} + (2x^2+1)(3x+1)\, f_{23} \\ &= f_{012} + 3x^2 2x f_{03} + (3x^2+3)(2x+2)\, f_{13} + (3x^2+2)(2x+3)\, f_{23} \end{aligned}$$

QGFE 6:

$$\begin{aligned} f &= f_{013} + x^2 x f_{02} + (x^2+1)(x+1)\, f_{12} + (x^2+2)(x+3)\, f_{23} \\ &= f_{013} + 2x^2 3x f_{02} + (2x^2+2)(3x+3)\, f_{12} + (2x^2+3)(3x+2)\, f_{23} \\ &= f_{013} + 3x^2 2x f_{02} + (3x^2+3)(2x+2)\, f_{12} + (3x^2+1)(2x+1)\, f_{23} \end{aligned}$$

QGFE 7:

$$\begin{aligned} f &= f_{023} + x^2 x f_{01} + (x^2+3)(x+2)\, f_{12} + (x^2+2)(x+3)\, f_{13} \\ &= f_{023} + 2x^2 3x f_{01} + (2x^2+1)(3x+1)\, f_{12} + (2x^2+3)(3x+2)\, f_{13} \\ &= f_{023} + 3x^2 2x f_{01} + (3x^2+2)(2x+3)\, f_{12} + (3x^2+1)(2x+1)\, f_{13} \end{aligned}$$

QGFE 8:

$$\begin{aligned} f &= f_{123} + (x^2+1)(x+1)\, f_{01} + (x^2+3)(x+2)\, f_{02} + (x^2+2)(x+3)\, f_{03} \\ &= f_{123} + (2x^2+2)(3x+3)\, f_{01} + (2x^2+1)(3x+1)\, f_{02} + (2x^2+3)(3x+2)\, f_{03} \\ &= f_{123} + (3x^2+3)(2x+2)\, f_{01} + (3x^2+2)(2x+3)\, f_{02} + (3x^2+1)(2x+1)\, f_{03} \end{aligned}$$

QGFE 9:

$$f = f_0 + x^2 x f_{0123} + x^2\, f_{1(2\cdot 2)(3\cdot 3)} + x f_{1(3\cdot 2)(2\cdot 3)}$$

Proof: From the expansion of (15) and the definition of quaternary 1-RPL of (11) to (14), we have

$$\begin{aligned} f &= (x^3+1)\, f_0 + (x^3+x^2+x)\, f_1 + (x^3+2x^2+3x)\, f_2 + (x^3+3x^2+2x)\, f_3 \\ &= f_0 + x^3 f_0 + (x^3+x^2+x)\, f_1 + (x^3+2x^2+3x)\, f_2 + (x^3+3x^2+2x)\, f_3 \\ &= f_0 + [(x^3+x^2+x) + (x^3+2x^2+3x) + (x^3+3x^2+2x)]\, f_0 + \\ &\quad (x^3+x^2+x)\, f_1 + (x^3+2x^2+3x)\, f_2 + (x^3+3x^2+2x)\, f_3 \\ &= f_0 + (x^3+x^2+x)(f_0 + f_1) + (x^3+2x^2+3x)(f_0 + f_2) + \\ &\quad (x^3+3x^2+2x)(f_0 + f_3) \\ &= f_0 + (x^3+x^2+x)\, f_{01} + (x^3+2x^2+3x)\, f_{02} + (x^3+3x^2+2x)\, f_{03} \\ &= f_0 + x(x+2)(x+3)\, f_{01} + x(x+1)(x+3)\, f_{02} + x(x+1)(x+2)\, f_{03} \end{aligned}$$

TABLE 5
An example quaternary function.

xy	*f*
00	0
01	1
02	2
03	3
10	1
11	0
12	3
13	2
20	2
21	3
22	0
23	1
30	3
31	2
32	1
33	0

Thus we prove QGFE 1. Similarly, we can prove QGFE 2 to 9. Thus we have theorem 2. □

5 QUATERNARY GALOIS FIELD DECISION DIAGRAM

Using the QGFE 1 - 9, we can construct quaternary Galois field decision diagram (QGFDD) of any quaternary logic function. Using different choices of expansions for a variable and using different variable ordering, we can construct many Kronecker like QGFDD. By flattening these decision diagrams, we can generate several QGFSOP expressions. A decision diagram with minimum number of paths to non-zero leaves will produce a minimum QGFSOP expression. A minimum QGFSOP expression will result into a minimum quantum circuit realizing the function.

A 2-input 1-output quaternary function [GF(4) sum of *x* and *y*] is shown in Table 5. Applying QGFE 1 to variable *x* and QGFE 2 to variable *y*, we get the QGFDD for the function of Table 5 as shown in Figure 1. Observation of the QGFE 1 - 9 reveals that a QGFE is a QGFSOP of four products and each of the products is a product of one composite cofactor and one or more quaternary reversible literals. Therefore, each node of the QGFDD has four children and they are arranged from left to right in the same order as in the corresponding QGFE. In the QGFDD, we write only the subscripts of the corresponding composite cofactors along the edges and

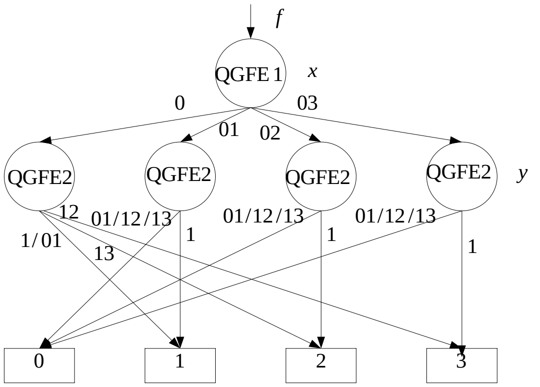

FIGURE 1
QGFDD for the function of Table 5 using QGFE1 and 2.

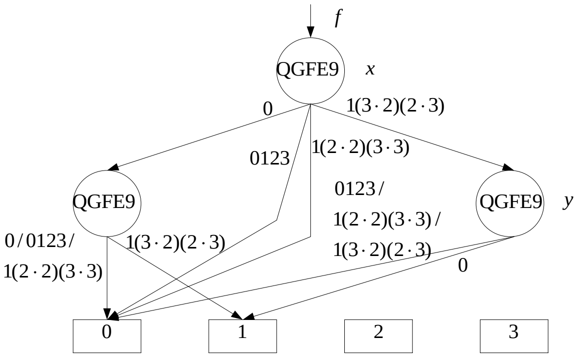

FIGURE 2
QGFDD for the function of Table 5 using QGFE9.

the corresponding product of reversible literals are implied. For example, we applied QGFE 1 to variable *x*. QGFE 1 has four parts having composite cofactors f_0, f_{01}, f_{02}, and f_{03} and they are written in the QGFDD of Figure 1 along the edges of the four children of the top node. If more than one child goes to the same node, then the corresponding composite cofactors are written along the single edge separated by slash (/). If we apply QGFE 9 on both the variables *x* and *y*, then we get the QGFDD as shown in Figure 2. Construction of QGFDD is not within the scope of this paper and, therefore, is not discussed.

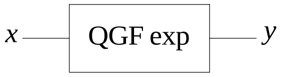

FIGURE 3
Representation of quaternary reversible 1-qudit gates.

The QGFDD is flattened to write the QGFSOP expression corresponding to the QGFDD. For flattening the QGFDD, we write the products of the reversible literals of the edges and the leaf constant along all paths. For example, flattening of the left most path of the QGFDD of Figure 1 yields $1\cdot 1\cdot 1+1\cdot(y+1)(y+2)(y+3)\cdot 1=1+(y+1)(y+2)(y+3)$. As product of zero and anything is zero, we need not to flatten the paths terminated at zero-leaf. By flattening the QGFDD of Figure 1, we have the following QGFSOP expression:

$$\begin{aligned} f &= 1+(y+1)(y+2)(y+3)+3y(y+1)(y+3)+ \\ &\quad 2y(y+1)(y+2)+x(x+2)(x+3)+ \\ &\quad 2x(x+1)(x+3)+3x(x+1)(x+2) \end{aligned} \tag{16}$$

Similarly, by flattening the QGFDD of Figure 2, we have the following QGFSOP expression:

$$f = x+y \tag{17}$$

These examples show that efficient algorithms are needed to select expansion for each variable and to construct the QGFDD so that the QGFDD and the resulting QGFSOP expression are minimum. In this work, we concentrated only on developing the expansions, not on the development of the said algorithms, which will obviously be our next attention.

6 QUATERNARY 1-QUDIT REVERSIBLE/QUANTUM GATES

Each of the 24 quaternary reversible-literals can be implemented as 1-qudit gates using quantum technology [8] and other reversible technologies. We will graphically represent these 1-qudit gates as shown in Figure 3.

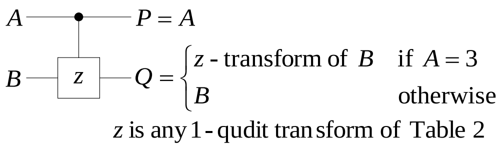
A
P = A
B
z
Q = { z - transform of B if A = 3; B otherwise
z is any 1-qudit transform of Table 2

FIGURE 4
Quaternary Muthukrishnan-Stroud gate family.

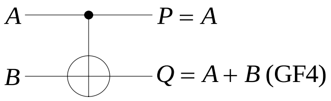

FIGURE 5
Quaternary Feynman gate.

7 QUATERNARY 2-QUDIT MUTHUKRISHNAN-STROUD GATE FAMILY

Muthukrishnan and Stroud [8] proposed a family of 2-qudit multiple-valued gates that are realizable (theoretically shown but not tested in the lab) in liquid ion-trap quantum technology. The quaternary form of the gate family is shown in Figure 4. We will refer this family of gates as quaternary Muthukrishnan-Stroud (M-S) gates. M-S gate is basically a controlled 2-qudit gate that applies a 1-qudit transform on the controlled input *B* when the controlling input *A* is 3.

8 QUATERNARY FEYNMAN GATE

Quaternary Feynman gate is shown in Figure 5, where *A*, *B* are inputs, *P* = *A* is the pass through output and $Q = A + B\,(\text{GF4})$ is the controlled output. Quaternary Feynman gate is a macro-level gate and can be realized using M-S primitive gate as shown in Figure 6. In Figure 6, $P = ((A+2)+3)+1 = A$. If $A = 0$, then $a_1 = 2$, $a_2 = 1$, $a_3 = 0$ and none of the transform will be applied to *B* and the output will be $Q = B = B + 0 = A + B$. If $A = 1$, then $a_1 = 3$, $a_2 = 0$, $a_3 = 1$ and only the left transform (*B*+1) will be applied to *B* and the output will be $Q = B + 1 = A + B$. If $A = 2$, then $a_1 = 0$, $a_2 = 3$, $a_3 = 2$ and the middle transform (*B*+2) will be applied on *B* and the output will be $Q = B + 2 = A + B$. If $A = 3$, then $a_1 = 1$, $a_2 = 2$, $a_3 = 3$ and the right

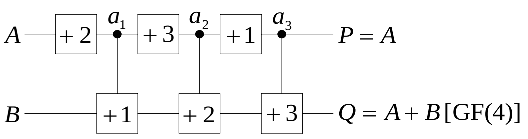

FIGURE 6
Realization of quaternary Feynman gate. [+1 → *x* + 1, +2 → *x* + 2, +3 → *x* + 3, where *x* is the corresponding input]

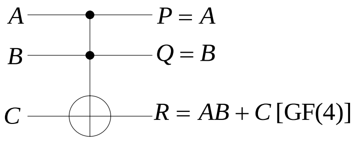

FIGURE 7
Quaternary Toffoli gate.

transform (*B*+3) will be applied on *B* and the output will be $Q = B + 3 = A + B$. Therefore, for all values of *A*, *P* = *A* and *Q* = *A* + *B* [GF(4)].

9 QUATERNARY TOFFOLI GATE

Quaternary Toffoli gate is shown in Figure 7, where *A* and *B* are controlling inputs and *C* is the controlled input. *P* = *A* and *Q* = *B* are the pass through outputs and $R = AB + C$ [GF(4)] is the controlled output. Quaternary Toffoli gate is a macro-level gate and can be realized using M-S primitive gates as shown in Figure 8.

In Figure 8, the resultant transformation along the input line *B* is $((B + 2) + 3) + 1 = B$, i.e., $Q = B$. Now, we will check the correctness of *P* = *A*. If $B = 0$, then $b_1 = 2$, $b_2 = 1$, $b_3 = 0$ and no controlled-transformation along the input line *A* will be applied. In this case, only the uncontrolled-transformations will be applied along the input line *A* and $a_1 = A$, $a_2 = 2a_1 = 2A$, $a_3 = 2a_2 = 2 \cdot 2A = 3A$, and $P = 2a_3 = 2 \cdot 3A = A$. If *B* = 1, then $b_1 = 3$, $b_2 = 0$, $b_3 = 1$ and only the controlled-transformations of segment-1 along the input line *A* will be applied. In this case, $a_1 = ((A + 2) + 3) + 1 = A$, $a_2 = 2a_1 = 2A$, $a_3 = 2a_2 = 2 \cdot 2A = 3A$, and $P = 2a_3 = 2 \cdot 3A = A$. If *B* = 2, then $b_1 = 0$, $b_2 = 3$, $b_3 = 2$ and only the

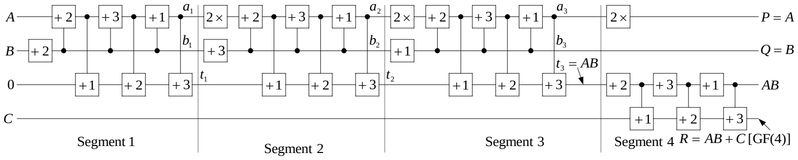

FIGURE 8
Realization of quaternary Toffoli gate. [+1 → x + 1, +2 → x + 2, +3 → x + 3, 2× → 2x, where x is the corresponding input]

controlled-transformations of segment-2 will be applied along the input line A. In this case, $a_1 = A$, $a_2 = ((2a_1 + 2) + 3) + 1 = 2a_1 = 2A$, $a_3 = 2a_2 = 2 \cdot 2A = 3A$, and $P = 2a_3 = 2 \cdot 3A = A$. If B = 3, then $b_1 = 1$, $b_2 = 2$, $b_3 = 3$ and only the controlled-transformations of segment-3 will be applied along the input line A. In this case, $a_1 = A$, $a_2 = 2a_1 = 2A$, $a_3 = ((2a_2 + 2) + 3) + 1 = 2a_2 = 2 \cdot 2A = 3A$, and $P = 2a_3 = 2 \cdot 3A = A$. Therefore, for all values of the controlling input B, P = A. Now we will verify the correctness of $t_3 = AB$. If $B = 0$, then as discussed earlier, $a_1 = A$, $a_2 = 2A$, $a_3 = 3A$. If A = 0, then $a_1 = a_2 = a_3 = 0$ and no controlled-transformation will be applied along the constant input line 0. So, $t_1 = t_2 = t_3 = 0 = 0 \cdot 0 = AB$. If A = 1, then $a_1 = 1$, $a_2 = 2$, $a_3 = 3$ and only the controlled-transformations of segment-3 will be applied along the constant input line 0. So, $t_1 = t_2 = 0$, $t_3 = ((0 + 1) + 2) + 3 = 0 = 1 \cdot 0 = AB$. If $A = 2$, then $a_1 = 2$, $a_2 = 3$, $a_3 = 1$ and only the controlled-transformations of segment-2 will be applied along the constant input line 0. So, $t_1 = 0$, $t_2 = ((0 + 1) + 2) + 3 = 0$, $t_3 = t_2 = 0 = 2 \cdot 0 = AB$. If $A = 3$, then $a_1 = 3$, $a_2 = 1$, $a_3 = 2$ and only the controlled-transformations of segment-1 will be applied along the constant input line 0. So, $t_1 = ((0 + 1) + 2) + 3 = 0$, $t_2 = t_3 = t_1 = 0 = 3 \cdot 0 = AB$. Therefore, for B = 0 and all possible values of A, $t_3 = AB$. In a similar fashion, it can be shown that for all possible values of A and B, $t_3 = AB$. Segment-4 of Figure 8 is a quaternary Feynman gate and the output R = AB + C [GF(4)]. A mirror circuit can be used at the right of segment-4 to restore the constant 0 for further use in the cascaded application of the gate.

Toffoli gates with more than three inputs are often used. A four-input Toffoli gate and its realization using three-input Toffoli gates are shown in Figure 9. The first three-input Toffoli gate with input 0 is used to generate

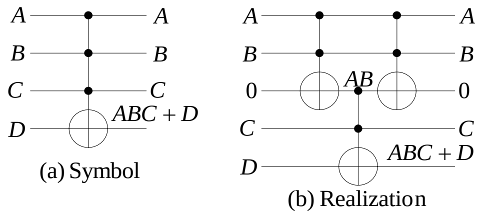

FIGURE 9
Four-input quaternary Toffoli gate.

AB. The second three-input Toffoli gate is used to generate $(AB)\cdot C + D\,(\mathrm{GF4})$. The third Toffoli gate is used as a mirror gate to restore the constant 0 for reusing in the next stage of the cascade. Using similar technique, Toffoli gate with any number of inputs can be realized.

10 SYNTHESIS OF QGFSOP EXPRESSION

The synthesis of a QGFSOP expression can be done as follows:

1. Realize a literal of a variable by using a quaternary 1-qudit gate.
2. If multiple literals of a variable is simultaneously needed, generate copies of the variable using quaternary Feynman gates by connecting the variable to the controlling input and putting a 0 to the controlled input [0 + *x* (GF4) = *x*]. Then realize the literals along the copies.
3. Realize a QGF product using a quaternary Toffoli gate by connecting the literals of the product to the controlling inputs and a 0 to the controlled input [0 + *xyz* (GF4) = *xyz*].
4. Realize a sum of two products by connecting the controlled output of the first Toffoli gate (implementing the first QGF product) to the controlled input of the second Toffoli gate (implementing the second QGF product).

Realization of the QGFSOP expression of (16) is shown in Figure 10. The QGFSOP expression (17) can be implemented with a single Feynman gate.

11 BINARY-TO-QUATERNARY ENCODER AND QUATERNARY-TO-BINARY DECODER CIRCUITS

The developed QGFSOP method can be effectively used for synthesis of binary function by grouping 2-bit together into quaternary value. For this

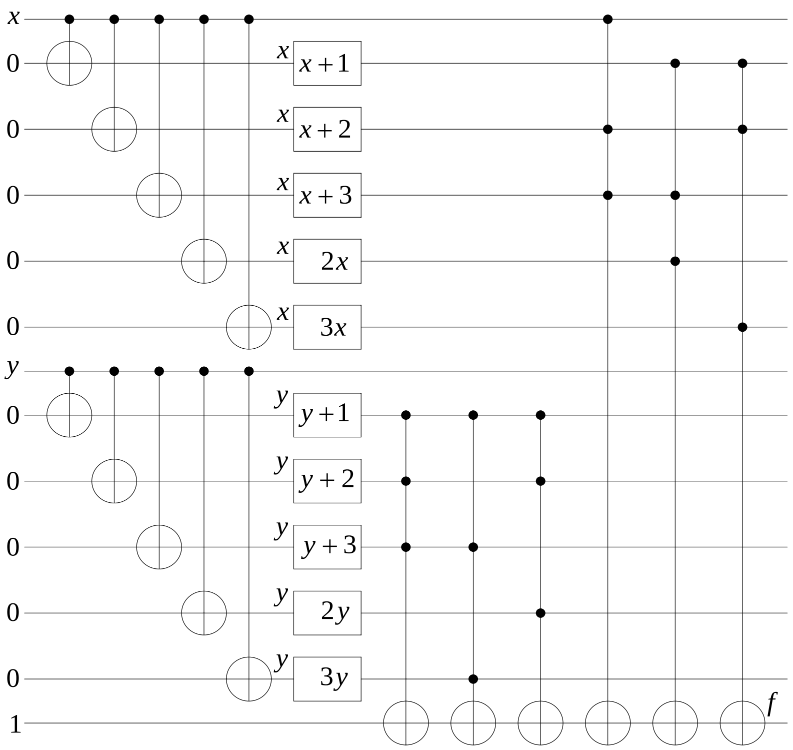

FIGURE 10
Realization of QGFSOP expression.

purpose we need binary-to-quaternary encoder and quaternary-to-binary decoder circuits.

A binary-to-quaternary encoder circuit using quaternary 1-qudit gates and 2-qudit M-S gates is shown in Figure 11. It is assumed that among the four quaternary qudit states only 0 and 1 will used for inputs A_1A_0. If $A_1A_0 = 00$, then $A_1^*A_0^* = 22$ and no transformation will be applied on *B*, therefore, $B = 0$. If $A_1A_0 = 01$, then $A_1^*A_0^* = 23$ and only +1 transformation will be applied on *B*, therefore, $B = 1$. If $A_1A_0 = 10$, then $A_1^*A_0^* = 32$ and only +2 transformation will be applied on *B*, therefore, $B = 2$. If $A_1A_0 = 11$, then $A_1^*A_0^* = 33$ and both +1 and +2 transformations will be applied on *B*, therefore, $B = 3$.

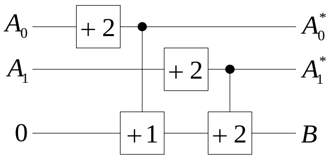

FIGURE 11
Binary-to-quaternary encoder circuit using quaternary 1-qudit gates and 2-qudit M-S gates.

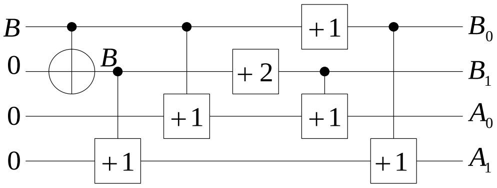

FIGURE 12
Quaternary-to-binary decoder circuit using quaternary Feynman gate, 1-qudit gates and 2-qudit M-S gates.

A quaternary-to-binary decoder circuit using quaternary Feynman gate, 1-qudit gates, and 2-qudit M-S gates is shown in Figure 12. If $B = 0$, then $B_1B_0 = 21$ and no transformation will be applied along A_1 and A_0, therefore, $A_1A_0 = 00$. If $B = 1$, then $B_1B_0 = 32$ and only +1 transformation will be applied along A_0 and no transformation will be applied along A_1, therefore, $A_1A_0 = 01$. If $B = 2$, then $B_1B_0 = 03$ and only +1 transformation will be applied along A_1 and no transformation will be applied along A_0, therefore, $A_1A_0 = 10$. If $B = 3$, then $B_1B_0 = 12$ and +1 transformation will be applied along both A_1 and A_0, therefore, $A_1A_0 = 11$.

12 CONCLUSION

Multiple-valued logic functions having many input variables can be easily expressed as Galois field sum of products (GFSOP) expression and can be realized using cascade of multiple-valued 1-qudit gates and multi-qudit Feynman and Toffoli gates [3, 5, 6]. Though a considerable number of useful works have been done on ternary logic synthesis, as far as we know, no work has yet been done on expressing quaternary function as quaternary Galois field sum of products (QGFSOP) expression and realizing the QGFSOP expression as a cascade of quaternary gates. In this paper, we have developed nine quaternary Galois field expansions (QGFE). These expansions can be used for constructing quaternary Galois field decision diagrams (QGFDD) of any quaternary function. By flattening the QGFDD we can generate quaternary Galois field sum of products (QGFSOP) expression for the function. However, in this work, we did not explore the algorithms for selecting expansion for a variable and to construct the QGFDD so that the resulting QGFDD and the QGFSOP expression are minimum. We have also shown example of implementation of QGFSOP expression as a cascade of quaternary 1-qudit gate, Feynman gate, and Toffoli gate.

For QGFSOP based quantum realization of functions with many input variables, we need to use quantum gates with many inputs. However, quantum gate with more than two inputs is very difficult to realize as a primitive gate, since interaction of more than two particles is nearly impossible to manage. In this paper, we have shown the quantum realization of macro-level quaternary 2-qudit Feynman and 3-qudit Toffoli gates on the top of theoretically liquid ion-trap realizable 1-qudit gates and 2-qudit Muthukrishnan-Stroud primitive gates [8]. We also show the realization of *m*-qudit ($m > 3$) Toffoli gates using 3-qudit Toffoli gates.

The quaternary base is very useful for encoded realization of conventional binary function by grouping two bits together into quaternary values. We show quantum circuit for binary-to-quaternary encoder and quaternary-to-binary decoder for this purpose.

The presented method is especially applicable to quantum oracles where only one function output is of importance and input qudits are copied to output. Such circuits are of particular importance in Grover algorithm or similar quantum algorithms. Our method can be adapted to multi-output reversible functions with paying the price of having one ancilla qudit for every output function. Observe that this method, in contrast to most methods from literature, also performs a conversion of a non-reversible function to a reversible one as a byproduct of the synthesis process. Comparing to the methods from literature, our method can be used for large functions. As it is using Galois logic, the circuits are highly testable [9]. In contrast, other papers on multiple-valued quantum logic cascade synthesis [1, 7] use the directly controlled multi-input gates based on Muthukrishnan-Stroud gates rather than the Galois-based Toffoli gates proposed here. How to build an optimal multi-input Toffoli gate using minimum number of quantum multiplexers or technology-realizable M-S gates remains an open problem. Some studies were done in [10]. It will be interesting to compare the methods from [1, 7] adapted to quaternary logic and the method proposed in this paper in terms of costs of truly quantum-realizable pulses or other low-level primitives as those discussed in [10].

Our future research includes (1) developing more QGFEs, if such expansions exist and (2) developing algorithms for (i) selecting expansion for each variable, (ii) variable ordering, and (iii) constructing QGFDD (Kronecker and pseudo-Kronecker types) for both single-output and multi-output functions so that the resulting QGFDD and the corresponding QGFSOP expression are minimized.

REFERENCES

[1] Curtis, E., Perkowski, M. (2004). A transformation based algorithm for ternary reversible logic synthesis using universally controlled ternary gates. *Proc. IWLS 2004*, Tamecula, California, USA, 2-4 June 2004.

[2] Denler, N., Yen, B., Perkowski, M., Kerntopf, P. (2004). Synthesis of reversible circuits from a subset of Muthukrishnan-Stroud quantum multi-valued gates. *Proc. IWLS 2004*, Tamecula, California, USA, 2-4 June 2004.

[3] Khan, M. H. A., Perkowski, M. A., Khan, M. R., Kerntopf, P. (2005). Ternary GFSOP minimization using Kronecker decision diagrams and their synthesis with quantum cascades. *Journal of Multiple-Valued Logic and Soft Computing, 11*, 2005, pp. 567-602.

[4] Khan, M. H. A., Perkowski, M. A. (2004). Genetic algorithm based synthesis of multi-output ternary functions using quantum cascade of generalized ternary gates. *Proc. of 2004 IEEE Congress on Evolutionary Computation (CEC 2004)*, Portland, Oregon, USA, 19-23 June 2004, pp. 2194-2201.

[5] Khan, M. H. A., Perkowski, M. A., Khan, M. R. (2004). Ternary Galois field expansions for reversible logic and Kronecker decision diagrams for ternary GFSOP minimization. *Proc. of 34th IEEE Int. Symp. on Multiple-Valued Logic (ISMVL 2004)*, Toronto, Canada, 19-22 May 2004, pp. 58-67.

[6] Khan, M. H. A., Perkowski, M. A., Kerntopf, P. (2003). Multi-output Galois field sum of products synthesis with new quantum cascades. *Proc. 33rd IEEE Int. Symp. On Multiple-Valued Logic (ISMVL 2003)*, Tokyo, Japan, 16-19 May 2003, pp. 146-153.

[7] Miller, D. M., Dueck, G., Maslov, D. (2004). A synthesis method for MVL reversible logic. *Proc. 34th IEEE Int. Symp. On Multiple-Valued Logic (ISMVL 2004)*, Toronto, Canada, 19-22 May 2004, pp. 74-80.

[8] Muthukrishnan, A., Stroud Jr., C. R. (2000). Multivalued logic gates for quantum computation. *Physical Review A, 62*, 052309/1-8.

[9] Kalay, U., Hall, D., Perkowski, M. (1998). A minimal and universal test set for multiple-valued Galois field sum-of-products circuits. *Proc. 7th Workshop on Post-Binary ULSI Systems*, Fukuoka, Japan, May 1998, pp. 50-51.

[10] Giesecke, N., Kim, D. H., Hossain, S., Perkowski, M. (2007). Search for universal ternary quantum gate sets with exact minimum costs. *37th IEEE Int. Symp. On Multiple-Valued Logic (ISMVL 2007)*, Oslo, Norway, 14-15 May 2007.