

치환리터럴에 의한 Quaternary Galois Field Sum-Of-Product(QGFSOP)형 1-변수 함수의 합성과 실현

Syntheses and realization of Quaternary Galois Field Sum-Of-Product(QGFSOP) expressed 1-variable functions by Permutational Literals

박동영*, 김백기*, 성현경**

Dong-Young Park*, Baek-Ki Kim* and Hyeun-Kyeong Seong**

요 약

Quaternary 논리에서 생성 가능한 1-qudit(1-variable quantum digit) 함수는 총 256개가 존재하지만 이들 중에서 가장 유용한 것은 "0,1,2,3"의 치환에 의해 $Ax^C + D(GF4)$ 형의 QGFSOP 표현이 가능한 24개이다. 본 논문에서는 24개 1-qudit 함수들의 $Ax^C + D(GF4)$ 연산에서 피연산자인 피승수 A와 피가수 D를 다단 종속된 치환리터럴의 제어인자로 사용하는 치환리터럴(Permutational Literals, PL) 표현과 QPL(Quaternary PL) gate를 제안하였다. 그리고 상호치환 'ab', 가산 '+D', 그리고 승산 'xA'와 같은 세 개의 PL 연산자를 사용하여 QGFSOP 표현된 24개 (1-qudit) 함수를 합성하기 위한 PL 합성법을 제안하였다. 끝으로 PL 합성법을 실현하기 위한 $Ax^C + D(GF4)$ 구조와 연산회로 및 CMOS 실현 방법을 제시하였다.

Abstract

Even though there are 256 possible 1-qudit(1-variable quantum digit) functions in quaternary logic, the most useful functions are $4!=24$ ones capable of representing in QGFSOP expressions by possible permuting of 0,1,2, and 3. In this paper, we propose a permutational literal(PL) representation and a QPL(Quaternary PL) gate which use the operands of a multiplicand A and an augend D in $Ax^C + D(GF4)$ operation as a control variable of multi-cascaded PLs. And we also present new PL synthesis algorithms to synthesize QGFSOP expressed 24 (1-qudit) functions by applying three PL operators as ab (mutual permutation), $+D$ (addition), and $\times A$ (multiplication). Finally architectures, circuits, and a CMOS implementation to realize proposed PL synthesis algorithms for $Ax^C + D(GF4)$ functions are presented.

Key words : quaternary logic, 1-qudit, QGFSOP, Permutational Literal(PL), QPL gate.

I. 서 론

Binary를 두 비트 씩 묶어서 quaternary 논리를

* 강릉원주대학교 정보통신공학과(Dept. of Information & Telecommunication Eng., Gangnung-Wonju National University)

** 상지대학교 컴퓨터정보공학부(School of Computer, Information and Communication, Sang-Ji University)

· 제1저자 (First Author) : 박동영

· 투고일자 : 2010년 8월 16일

· 심사(수정)일자 : 2010년 8월 17일 (수정일자 : 2010년 10월 18일)

· 게재일자 : 2010년 10월 30일

표현하는 binary 인코딩 방법은 binary 경우에 필요한 본래 크기의 약 $1/\log_2 4 = 1/2$ 배 만큼 실제의 크기를 줄일 수 있게 해준다. 따라서 같은 수의 비트로 훨씬 더 많은 정보를 표현할 수 있는 정보표현 능력이거나 소자 면적의 절감 측면뿐만 아니라 동작 속도와 전력 소비 측면 등에서도 비약적 발전을 보이고 있는 ternary나 quaternary 논리 같은 MV(multiple-valued) 논리가 현존 binary 논리의 약속받는 미래 대안이다. MV 논리는 초기에 signed digit의 특성을 그대로 반영할 수 있는 ternary 논리가 각광을 받았으나, 최근에는 quaternary Feynman 게이트와 Toffoli 게이트를 이용한 QGFSOP 표현된 quaternary 논리 함수의 합성과 실현[1,2], CMOS 논리 설계를 이용한 quaternary MUX 설계[3]와 PLA 설계[4], bipolar 기술을 이용한 quaternary 반산기 설계[5], DPC(differential pair circuit)를 이용한 저전력 다치 VLSI 설계[6] 및 MV를 이용한 GF 연산[7] 등에서 볼 수 있는 것처럼 binary 논리의 확장 측면에서 ternary보다 유리한 quaternary 논리에 대한 연구가 활발하다.

본 논문에서는 quaternary quantum 논리에서 사용하는 주요 함수들의 실현에 liquid ion-trap 기술을 사용하는 quaternary Feynman 게이트나 Toffoli 게이트 대신에 현존하는 CMOS 기술로 실현이 가능한 출력 치환 게이트들을 이용해 QGFSOP 표현된 1-qudit 함수들의 합성과 실현 방법을 연구하였다. 본 논문의 구성은 다음과 같다. 먼저 II절에서는 유용한 1-qudit 함수들로 구성 가능한 QGFSOP 표현 함수들을 살펴 보았다. III 절의 3-1장에서는 QGFSOP 표현된 24개 1-qudit 함수의 4-tupled 벡터들에 의한 $Ax^C + D(GF4)$ 가산 연산의 출력벡터들 간에 일어나는 실질적 치환 작용을 분석하고, 세 가지 PL 연산자(상호치환 'ab', 가산 '+D', 그리고 승산 'xA')를 정의하였다. 3-2장에서는 PL 표현법과 디코더를 정의한 후 세 개의 정리를 통해 QGFSOP 표현된 24개 1-qudit 함수들이 본 논문의 PL 합성법으로 재합성될 수 있음을 보였다. IV절에서는 전 장의 PL 합성법을 실현하기 위해 먼저 QPL gate 구조와 $Ax^C + D(GF4)$ architecture 및 이들의 실현 회로들을 제시하였다. 다음에 S=0123인 경우에 $S + D(GF4)$ architecture를 0.18um CMOS 기술로 설계한 HSPICE simulation 결과를 제시하고, 본 논문의 PL 합성법과 이의 실현 방법을 Perkowski[1,2] 방법과 비교하였다. 결론인 V 절에서는 본 논문의 제안사

항과 향후 연구 과제를 서술하였다.

II. 유용한 quaternary 함수

어떤 진수(radix)가 r이고 변수 수가 n인 MVL 함수의 가능한 함수 수 N은 식 (1)과 같다.

$$N = r^{(r^n)} \quad (1)$$

예로써 1-변수 binary 논리에서 r=2 및 n=1인 경우는 입력 x=01에 대하여 i번째 1-qudit 함수를 Fi라 하면 F1=00, F2=01, F3=10, F4=11과 같은 4개의 binary 함수들이 존재한다. 변수가 한개 증가한 r=2 및 n=2인 2변수 binary 논리에서는 16개의 가능한 함수들이 존재한다. 따라서 r=4 및 n=1인 quaternary 논리에서는 표 1과 같이 총 256개의 1-qudit 함수들이 존재한다.

표 1. 1-qudit 함수
Table 1. 1-qudit functions.

X	F ₁	F ₂	F ₃	F ₄	F ₅	...	F ₂₈	...	F ₇₉	...	F ₁₇₈	...	F ₂₂₉	...	F ₂₅₆
0	0	0	0	0	0	...	0	...	1	...	2	...	3	...	3
1	0	0	0	0	0	...	1	...	0	...	3	...	2	...	3
2	0	0	0	0	1	...	2	...	3	...	0	...	1	...	3
3	0	1	2	3	0	...	3	...	2	...	1	...	0	...	3

표 1은 산업적으로 매우 유용한 일부 함수들은 물론 아직까지 활용되지 못하고 있는 모든 불용 함수들도 포함하고 있다. 표 1의 일부 함수들을 이용하면 표 2와 같이 산업적으로 매우 유용한 QGF 가산과 승산 진리표를 구성할 수 있다.

표 2. QGF 가산과 승산 진리표
Table 2. The truth table on QGF addition and multiplication.

+	F ₂₈	F ₇₉	F ₁₇₈	F ₂₂₉
	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

•	F ₁	F ₂₈	F ₄₆	F ₅₅
	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

표 3은 mod3 연산을 실행하는 quaternary 반가산 진리표이다.

표 3. Quaternary 반가산 진리표

Table 3. Truth table of quaternary half addition.

⊕	F ₂₈	F ₁₀₇	F ₁₇₈	F ₁₉₉
	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

표 4는 “0,1,2,3”의 모든 치환으로 생성 가능한 $Ax^C + D(GF4)$ 형의 QGFSOP 표현된 24개의 고유한 1-qudit 함수들이다.

표 4. QGFSOP 표현

Table 4. QGFSOP expressions.

x	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
x	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
x	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
x	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
x	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
x	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

III. 치환리터럴을 이용한 quaternary 함수 합성

3-1 QGFSOP 표현 함수의 성질

표 4에서 $x=0123$ 인 경우의 각 출력벡터에 대응하는 24의 1-qudit 함수를 구성하는 출력벡터 원소들은 $a \neq b \neq c \neq d$ 와 같은 고유한 quaternary 수에 의해 $f=abcd$ 로 표현될 수 있다. 만일 $i_i \in \{a,b,c,d\}$ 인 경우에 $i_1 = 0$ 을 포함하는 2-tupled 출력벡터 $i_1i_2 \in \{01, 02, 03\}$ 이므로 i_1i_2 를 뺀 나머지 2-tupled 출력벡터는 $i_3i_4 \in \{23, 13, 12\}$ 이다.

이제 표 4의 $Ax^C + D(GF4)$ 연산을 분석하기 위해 $x = abcd$ 로 표현된 4-tupled 출력벡터 상의 임의의 두 출력벡터 간의 거리(distance)를 식 (2)에서와 같이 두 출력벡터쌍의 차 절대치 중의 최대치로 정의한다.

$$D \equiv \text{Max}(|i_1 - i_2|, |i_3 - i_4|) \quad (2)$$

정리 1 : $x = 0123$ 일 때 $a \neq b \neq c \neq d$ 로 고유한 $Ax^C = abcd$ 에 대한 모든 $Ax^C + D(GF4)$ 연산으로 생성된 24개의 1-qudit 함수들은 $abcd$ 에 대하여 1) 무치환 2) ‘0과 1’ 및 ‘2와 3’간의 상호치환, 3) ‘0과 2’ 및 ‘1과 3’ 간의 상호치환, 그리고 4) ‘0과 3’ 및 ‘1과 2’ 간의 상호치환 등과 같은 네 종류의 치환에 의해 구할 수 있다.

증명 : $Ax^C + D(GF4)$ 연산에서 1) 무치환은 $D = 0$ 인 경우에 $abcd + 0(GF4) = abcd$ 이며, 2) ‘0과 1’ 및 ‘2와 3’간의 상호치환은 $D = 1$ 인 ‘01과 23’의 두 출력벡터쌍의 각 벡터간의 상호치환으로서 이 치환결과는 $abcd + 1(GF4)$ 연산과 같다. 3) ‘0과 2’ 및 ‘1과 3’ 간의 상호치환은 $D = 2$ 인 ‘02와 13’의 두 출력벡터쌍의 각 벡터간의 상호치환으로서 이 치환결과는 $abcd + 2(GF4)$ 연산과 같으며, 그리고 4) ‘0과 3’ 및 ‘1과 2’ 간의 상호치환은 $D = 3$ 인 ‘03과 12’의 두 출력벡터쌍의 각 벡터간의 상호치환으로서 이 치환결과는 $abcd + 3(GF4)$ 연산과 같다. 그러므로 표 4에서 $Ax^C + D(GF4)$ 연산으로

생성된 24개의 1-qudit 함수들은 $D = 0, 1, 2, 3$ 에 따른 네 가지 치환으로 생성할 수 있다.

Q.E.D.

정의 1 : $x = 0123$ 에 대해 $Ax^C = abcd$ 가 $a \neq b \neq c \neq d$ 인 고유한 quaternary 1-qudit 함수일 때 $Ax^C + D(GF4)$ 연산을 위한 치환, 가산 및 승산 연산자들을 각각 ab , $+D$ 및 $\times A$ 로 표기하며 이들은 각각 다음과 같은 성질을 갖는다.

1) ab 는 a 에서 b 및 b 에서 a 로의 상호치환 연산자로서 두 개 이상의 병렬 상호치환은 AND 연산자를 사용하여 $ab \cdot cd$ 로 표기한다.

2) $+D = abcd + D(GF4) \equiv ab \cdot cd$.

3) $\times A = abcd \times A(GF4)$.

3-2 QGFSOP 표현된 함수의 치환리터럴 합성 알고리즘

본 논문에서 QGFSOP 표현된 함수의 합성에 사용한 치환리터럴의 성질은 정의 2와 같다.

정의 2 : 제어변수 m 을 적용한 치환 $P = \{ab, \times A\}$ 의 승산치환 P^m 과 $P = +D$ 의 가산치환 mP 가 소스함수 $S = abcd$ 을 대상으로 실행됨을 치환리터럴 S^{P^m} 과 S^{mP} 로 표기하면

$$S^{P^m} = S^P, \text{ if } m = 1 \quad (3)$$

$$= S, \text{ if } m = 0.$$

$$S^{mP} = S^{+mD} = S^{+D}, \text{ if } m = 1 \quad (4)$$

$$= S^{+0}, \text{ if } m = 0.$$

식 (3)에서 S^{P^m} 은 $m=1$ 일 때 소스함수 $S = abcd$ 에 대하여 치환 P 를 실행하며, $m=0$ 이면 $P^0 = 1$ 이 되어 소스함수가 치환작용 없이 통과(pass)된다. 또한 식 (4)에서 $m=1$ 일 때 $P = +D$ 및 $S^{mP} = S^{+D}$ 이므로 소스함수 $S = abcd$ 에 대하여 $abcd + D(GF4)$ 를 실행하며, $m=0$ 이면 $S^{+0} = S$ 가 되어 치환작용 없이 소스함수가 통과된다.

이제 $Ax^C + D(GF4)$ 연산에서 $H \in \{A, D\}$ 인 quaternary를 binary로 바꾸기 위한 quaternary-to-binary decoder를 식 (5)로 정의한다.

$$H = \sum_{i=1}^2 m_i \cdot 2^{i-1} \equiv m_1 m_2. \quad (5)$$

표 2의 QGF 가산 진리표에 대한 본 논문의 함수 합성 알고리즘은 $P = +D$ 의 치환연산을 실행하는 치환리터럴의 cascade 연산을 피가수의 디코딩으로 생성한 치환제어인자에 의해 효과적으로 제어하는 정리 2이다.

정리 2 : $S + D(GF4)$ 연산을 위한 소스함수가 $x = 0123$ 에 대해 $S = Ax^C = abcd$ 인 경우에 피가수가 $D \equiv m_1 m_2$ 이면

$$S + D(GF4) = S^{+(+m_1)2m_2}. \quad (6)$$

증명 : $S = x = 0123$ 인 경우에

1) $D \equiv 00$; $S^{+(+0)0} = 0123^{+0} = 0123$,
 $\therefore S^{+(+0)0} = S + 0(GF4)$.

2) $D \equiv 10$; $S^{+(+1)0} = 0123^{+1} = 1032$,
 $\therefore S^{+(+1)0} = S + 1(GF4)$.

3) $D \equiv 01$; $S^{+(+0)2} = 0123^{+2} = 2301$,
 $\therefore S^{+(+0)2} = S + 2(GF4)$.

4) $D \equiv 11$; $S^{+(+1)2} = 1032^{+2} = 3210$
 $\therefore S^{+(+1)2} = S + 3(GF4)$.

($S = Ax^C$ 인 나머지 모든 경우도 만족) Q.E.D.

표 2의 $Ax(GF4)$ 승산 연산을 위한 본 논문의 함수 합성 알고리즘은 $P = \times A$ 의 치환연산을 실행하는 치환리터럴의 cascade에 $A = m_1 m_2$ 를 치환제어 지수인자와 zero 검출항($\overline{m_1 \cdot m_2}$)에 적용한 정리 3이다.

정리 3 : $S = x = 0123$ 및 $A = m_1 m_2$ 일 때

$$Ax(GF4) = \overline{\overline{m_1 \cdot m_2}} S^{\times(\times 3)^{2^{m_1}} 2^{2^{m_2}}} \quad (7).$$

증명 : 식 (7)에서 1단의 $S^{\times 3} = 0123^{\times 3} = 0312$ 치환결과는 2단의 공통입력이다.

- 1) $A \equiv 00$; $\overline{\overline{m_1 \cdot m_2}} = \overline{\overline{0 \cdot 0}} = 0$.
 $\therefore 0x(GF4) = 0 \cdot S^{\times(\times 3)^{2^{m_1}} 2^{2^{m_2}}}$.
- 2) $A \equiv 10$; $\overline{\overline{1 \cdot 0}} = 1$.
 2단 치환 $S^{\times 2} = 0312^{\times 2} = 0123$,
 3단 치환 $S^{\times 1} = 0123^{\times 1} = 0123$ (pass).
 $\therefore 1x(GF4) = 1 \cdot S^{\times(\times 3)^{2^1}}$.
- 3) $A \equiv 01$; $\overline{\overline{0 \cdot 1}} = 1$.
 2단 치환 $S^{\times 1} = 0312^{\times 1} = 0312$ (pass),
 3단 치환 $S^{\times 3} = 0312^{\times 3} = 0231$.
 $\therefore 2x(GF4) = 1 \cdot S^{\times(\times 3)^{1 \cdot 3}}$.
- 4) $A \equiv 11$; $\overline{\overline{1 \cdot 1}} = 1$.
 2단 치환 $S^{\times 2} = 0312^{\times 2} = 0123$,
 3단 치환 $S^{\times 3} = 0123^{\times 3} = 0312$.
 $\therefore 3x(GF4) = 1 \cdot S^{\times(\times 3)^{2 \cdot 3}}$.

따라서 식 (7)은 모든 경우에 대하여 $Ax(GF4)$ 연산을 만족한다.

Q.E.D.

표 4에서 $S = x = 0123$ 인 경우에 $Ax^2(GF4)$ 의 승산 진리표는 표 5와 같으며, 표 5에 대한 본 논문의 함수 합성법은 $P = ab$ 의 치환연산을 실행하는 치환리터럴의 cascade에 $A = m_1 m_2$ 를 치환제어 지수인자와 zero 검출항($\overline{\overline{m_1 \cdot m_2}}$)에 적용한 정리 4이다.

표 5. $Ax^2(GF4)$ 진리표

Table 5. The truth table on $Ax^2(GF4)$.

A x	$Ax^2(GF4)$			
	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	3	1	2
3	0	2	3	1

정리 4 : $S = x = 0123$ 및 $A \equiv m_1 m_2$ 일 때

$$Ax^2(GF4) = \overline{\overline{m_1 \cdot m_2}} S^{((23^{m_1})12^{m_2})23^{m_1 m_2}} \quad (8).$$

증명 :

- 1) $A \equiv 00$; $\overline{\overline{m_1 \cdot m_2}} = \overline{\overline{0 \cdot 0}} = 0$.
 $\therefore 0x^2(GF4) = 0 \cdot S = 0$.
- 2) $A \equiv 10$; $\overline{\overline{1 \cdot 0}} = 1$ 이므로 $S^{((23^1)12^0)23^0}$ 에 대해
 1단 치환 $S^{23} = 0123^{23} = 0132$,
 2단, 3단 치환 $S = 0132$ (pass).
 한편, $1x^2(GF4) = 0132$.
 $\therefore 1x^2(GF4) = S^{((23^1)12^0)23^0}$.
- 3) $A \equiv 01$; $\overline{\overline{0 \cdot 1}} = 1$ 이므로 $S^{((23^0)12^1)23^0}$ 에 대해
 1단 치환 $S = 0123$ (pass),
 2단 치환 $S^{12} = 0123^{12} = 0213$,
 3단 치환 $S = 0213$ (pass).
 한편, $2x^2(GF4) = 0213$.
 $\therefore 2x^2(GF4) = S^{((23^0)12^1)23^0}$.
- 4) $A \equiv 11$; $\overline{\overline{1 \cdot 1}} = 1$ 이므로 $S^{((23^1)12^1)23^1}$ 에 대해
 1단 치환 $S^{23} = 0123^{23} = 0132$,
 2단 치환 $S^{12} = 0132^{12} = 0231$,
 3단 치환 $S^{23} = 0231^{23} = 0321$.
 한편, $3x^2(GF4) = 0321$.
 $\therefore 3x^2(GF4) = S^{((23^1)12^1)23^1}$.

따라서 식 (8)은 표 5의 모든 경우에 대하여 $Ax^2(GF4)$ 연산을 만족한다.

Q.E.D.

정리 2, 3, 4의 함수 합성알고리즘을 “PL 합성법”이라 명명한다.

IV. Architecture

본 장에서는 3장에서 제안한 (QGFSOP 표현 함수의) PL 합성법들을 모듈 구조 게이트로 설계하는 방법을 제시하였다. 그림 1은 본 논문의 식 (3)과 식 (4)에서 제안한 치환리터럴의 구조적 실현으로써 이

회로를 "QPL(Quaternary Permutational Literal) gate"라 명명한다.

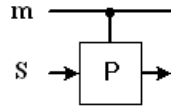


그림 1. QPL 게이트
Fig. 1. An QPL gate.

다음의 연산구조들은 QGFSOP 표현된 함수들을 본 논문의 PL 합성법에 의해 QPL gate의 다단 종속 연결로 설계한 구조들로서 그림 2는 식 (6)을 실현한 $S + D(GF4)$ 구조이다.

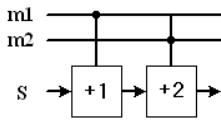


그림 2. $S + D(GF4)$ 구조
Fig. 2. $S + D(GF4)$ architecture.

그림 3은 식 (7)에 대한 $Ax(GF4)$ 구조이다.

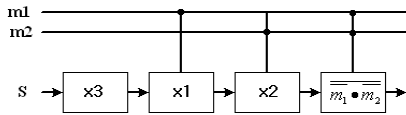


그림 3. $Ax(GF4)$ 구조
Fig. 3. $Ax(GF4)$ architecture.

그림 4는 식 (8)에 대한 $Ax^2(GF4)$ 구조이다.

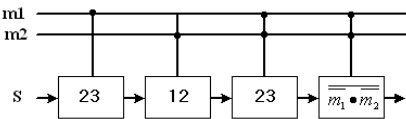


그림 4. $Ax^2(GF4)$ 구조
Fig. 4. $Ax^2(GF4)$ architecture.

소스함수부는 전류원 회로로 구성하며 임의 $S = Ax^C = abcd$ 에 대응하는 전류모드의 1-qudit 함수를 공급한다. 전단에 $S = Ax^C$ 의 승산회로를 cascade할 경우의 소스함수는 $x = S = 0123$ 이다.

치환게이트부는 $P = \{ab, +D, \times A\}$ 의 치환연산자들을 하드웨어적으로 실현하기 위해 $m=0$ 인 통과용 경로와 $m=1$ 인 치환경로를 갖는 이중 전류 경로를 갖는다. $P = +D$ 게이트 실현은 D의 크기에 따라 $+1 \equiv 01 \cdot 23$, $+2 \equiv 02 \cdot 13$, 그리고 $+3 \equiv 03 \cdot 12$ 의 가산 치환을 실행하도록 pass transistor를 이용하여 실현한다. $P = \times A$ 게이트 실현은 $m=1$ 인 경우에 $\times 2$ 와 $\times 3$ 의 치환연산자에 대해서 각각 '0'이 아닌 출력백

터열 전체를 한 자리씩 up-shift와 down-shift시켜 실현하며, 이때 '0'은 다음 단으로 직결한다. $P = ab$ 실현은 $m=1$ 일 때 해당 출력벡터만 상호치환하고 나머지 잔여 벡터들은 다음 단으로 직결한다. 한편 HDL 등에 의한 치환게이트부의 소프트웨어적 실현은 식 (6)~(8)의 직접적 실현이다.

시뮬레이션을 위해 본 논문에서 사용한 quaternary-to-binary decoder는 $I_0 = 10[\mu A]$ 일 때 0, 1I₀, 2I₀, 3I₀의 전류모드 문턱 검출기를 사용하여 0.5I₀, 1.5I₀ 및 2.5I₀의 전압모드 문턱값을 생성한 후 이들의 윈도우 리터럴 조합으로 binary 출력 $m_1 m_2$ 을 생성하였다.

식 (6)~(8) 연산에 대한 단일 출력은 출력 단에 4-to-1 MUX 1개를 추가하면 된다. 실제로 본 논문에서 가산기의 단일출력 실현을 위한 4-to-1 MUX는 당초 3um CMOS 설계규칙으로 설계된 Pelayo[4]의 literal generator를 0.18um CMOS 설계규칙을 적용하여 실현하였다.

그림 (5), (6) 및 (7)은 $S = Ax^C$ 일 때 각각 $S + D(GF4)$, $Ax(GF4)$ 및 $Ax^2(GF4)$ 연산구조들을 실현한 회로들이다.

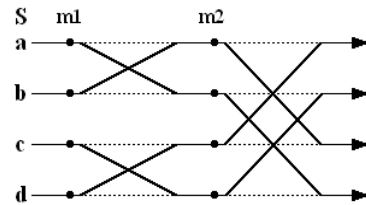


그림 5. $S + D(GF4)$ 실현
Fig. 5. Realization of $S + D(GF4)$.

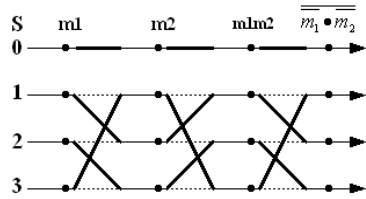


그림 6. $Ax(GF4)$ 실현
Fig. 6. Realization of $Ax(GF4)$.

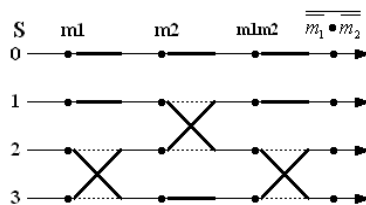


그림 7. $Ax^2(GF4)$ 실현
Fig. 7. Realization of $Ax^2(GF4)$.

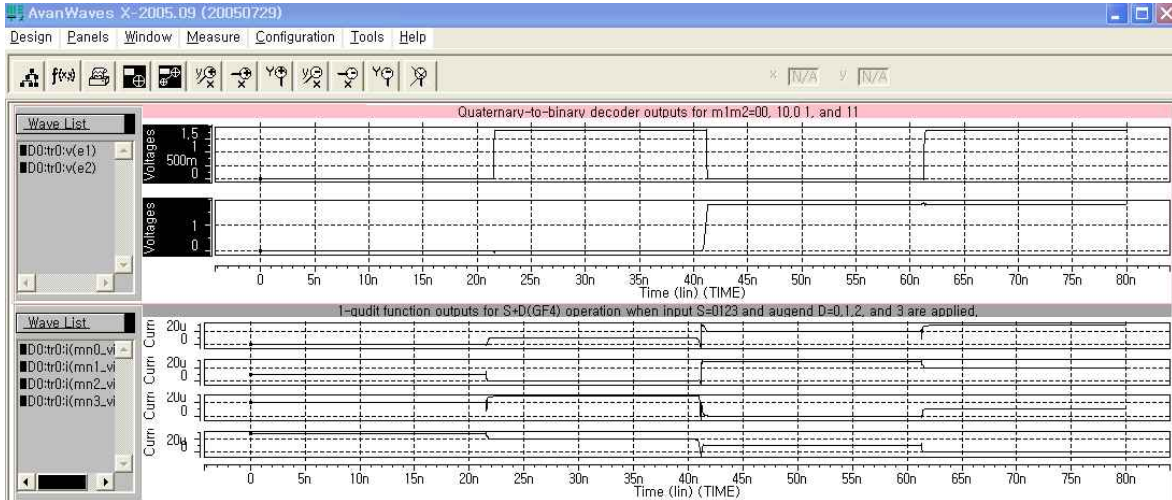


그림 8. 입력 S=0123에 대한 그림 5의 HSPICE simulation 결과
 Fig. 8. HSPICE simulation results of the circuit in Fig.5 for the input S=0123.

그림 8은 그림 5를 0.18um CMOS 설계기술로 실현한 HSPICE simulation 결과이다. 총 여섯 개의 그래프 중에서 첫 번째와 두 번째 그래프는 피가수 $D = 0, 1, 2, 3$ 에 대한 quaternary-to-binary decoder 출력으로써 좌측부터 세로 두 열은 각각 $D \equiv m_1 m_2 = 00, 10, 01$ 및 11일 때의 binary 출력들이다. 다음에 세 번째에서 여섯 번째 그래프들은 위의 decoder 출력으로 제어된 $S + D(GF4)$ 함수의 출력들으로써 좌측부터 네 개씩의 세로 열 그래프들은 소스함수가 $S = 0123$ 일 때 각각 0123, 1032, 2013, 그리고 0321에 대응하는 1-qudit 함수들의 전류값들을 보여 주고 있다. 이 결과들은 피가수와 피승수를 QPL gate의 제어변수로 사용하여 QGFSOP 표현 함수를 다단 종속으로 합성한 본 논문의 PL 합성법이 현존 CMOS 기술로 실현될 수 있음을 보여 준다.

표 6은 본 논문의 함수 합성법을 Perkowski[1,2] 방법과 비교한 결과이다.

표 6. 함수 합성 방법 비교

Table 6. The comparison of functional syntheses.

비교	Perkowski	본 논문
실현 기술	theoretical liquid ion-trap	CMOS
사용 게이트	Feynman gate, Toffoli gate	QPL gate
사용 연산자	5	3
주요 특징	다변수 확장	모듈설계용이

V. 결 론

본 논문에서는 $Ax^C + D(GF4)$ 연산의 피연산자인 피승수 A와 피가수 D를 다단 종속된 PL의 제어인자로 사용하는 새로운 PL 표현법과 QPL gate를 제안하였으며, 리터럴 연산 시에 치환, 가산 및 승산의 세 가지 연산자만을 사용하여 $Ax^C + D(GF4)$ 형의 QGFSOP 표현된 24개 1-qudit 함수들이 공통 모듈구조의 QPL gate들에 의해 효율적으로 실현될 수 있음을 보였다. 제안한 PL 합성법의 타당성을 검증하기 위해 소스 함수가 S=0123인 경우의 $S + D(GF4)$ 구조를 1.5V 0.18 CMOS 기술로 설계한 후 HSPICE simulation으로 함수 합성과 실현에 대한 전반적 설계 이론 및 실현 회로들의 정상 동작을 입증하였다. 향후의 연구과제는 미개척 불용함수들의 활용 방안연구와 합성알고리즘의 확장, 함수 간략화 및 양자화 기술과의 접목이라 할 수 있다.

감사의 글

본 논문은 2009년도 강릉원주대학교 교수연구년 연구 지원에 의하여 수행되었음.

참 고 문 헌

[1] Mozammel H. A. Khan and Marek A. Perkowski, "GF(4) Based Synthesis of Quaternary Reversible/Quantum Logic Circuits," *Proc. of 37th IEEE Int. Symp. on Multiple-Valued Logic(ISMVL 2007), May 13 - May 16, Norway, Oslo, pp.11-16, 2007.*

[2] Mozammel H. A. Khan, Nafisa K. Siddika, and Marek A. Perkowski, "Minimization of Quaternary Galois Field Sum of Products Expression for Multi-Output Quaternary Logic Function using Quaternary Galois Field Decision Diagram," *Proc. of 38th IEEE Int. Symp. on Multiple-Valued Logic (ISMVL 2008), pp.125-130, 2008.*

[3] Ricardo Cunha, Henri Boudinov, and Luigi Carro, "Quaternary Look-up Tables Using Voltage-Mode CMOS Logic Design," *Proc. of 37th IEEE Int. Symp. on Multiple-Valued Logic(ISMVL 2007), May 13 - May 16, Norway, Oslo, pp.56-61, 2007.*

[4] F.J. Pelayo, A. Prieto, A. Lloris, and J. Ortega, "CMOS Current-Mode Multivalued PLA's," *IEEE Tran. on Circuits and Systems, vol.38, no.4, pp. 434-441, April 1991.*

[5] Carlos Roberto Mingoto Junior, "A Quaternary Half-Adder Using Current-Mode Operation with Bipolar Transistor," *Proc. of 36th IEEE Int. Symp. on Multiple-Valued Logic(ISMVL 2006), pp.15-20, 2006.*

[6] Nobuaki Okada and Michitaka Kameyama, "Low-Power Multiple-Valued Reconfigurable VLSI Using Series-Gating Differential-Pair Circuits," *Proc. of 37th IEEE Int. Symp. on Multiple-Valued Logic(ISMVL 2007), May 13 - May 16, Norway, Oslo, pp.25-30, 2007.*

[7] Nabil Abu - Khader and Pepe Siy, "Inverion/ Division in Galois Field Using Multiple- Valued Logic," *Proc. of 37th IEEE Int. Symp. on Multiple-Valued Logic(ISMVL 2007), May 13 - May 16, Norway, Oslo, pp.17-22, 2007.*

박 동 영 (朴東泳)



1980년 2월 : 인하대학교 전자공학과 (공학사)
 1985년 8월 : 인하대학교 대학원 전자공학과(공학석사)
 1995년 8월 : 인하대학교 대학원 전자공학과(공학박사)
 1991년 9월 ~현재 강릉원주대학교 정보통신공학과 교수

관심분야: MVL 및 VLSI 회로설계, 데이터 변환 등.

김 백 기 (金栢起)



1985년 2월 : 경희대학교 전자공학과 (공학사)
 1987년 8월 : 경희대학교 대학원 전자공학과(공학석사)
 1997년 2월 : 경희대학교 대학원 전자공학과(공학박사)
 1990년 9월 ~현재 강릉원주대학교 정보통신공학과 교수

관심분야: 디지털신호처리 등

성 현 경 (成賢慶)



1982년 2월 : 인하대학교 전자공학과 (공학사)
 1984년 2월 : 인하대학교 대학원 전자공학과(공학석사)
 1991년 2월 : 인하대학교 대학원 전자공학과(공학박사)
 2005~2006년 미국 Naval Postgraduate School 방문교수

1991년~현재 상지대학교 컴퓨터정보공학부 교수

관심분야: Multiple-Valued Logic Design, Computer Architecture Design, Information & Coding Theory, Cryptography Theory & Security, RFID/WSN 설계 및 응용 등.