

論文2003-40SD-12-10

전류모드 CMOS 다치 논리회로를 이용한 32×32-Bit Modified Booth 곱셈기 설계

(Design of a 32×32-Bit Modified Booth Multiplier Using Current-Mode CMOS Multiple-Valued Logic Circuits)

李恩實*, 金政範**

(Eun Sil Lee and Jeong Beom Kim)

요약

본 논문에서는 CMOS 다치 논리회로를 이용한 32×32 Modified Booth 곱셈기를 제시하였다. 이 곱셈기는 Radix-4 알고리즘을 이용하였으며, 전류모드 CMOS 4치 논리회로로 구현하였다. 설계한 곱셈기는 트랜지스터 수를 기존의 전압 모드 2진 논리 곱셈기에 비해 63.2%, 이전의 다치 논리 곱셈기에 비해 37.3% 감소시켰다. 이 곱셈기는 내부 구조를 규칙적으로 배열하여 확장성을 갖도록 하였다. 설계한 회로는 3.3V의 공급전압과 단위전류 10 μ A를 사용하여, 0.35 μ m CMOS 기술을 이용하여 구현하였으며 HSPICE를 사용하여 검증하였다. 시뮬레이션 결과, 설계한 곱셈기는 5.9ns의 최대 전달지연시간과 16.9mW의 평균 전력소모 특성을 갖는다.

Abstract

This paper proposes a 32×32 Modified Booth multiplier using CMOS multiple-valued logic circuits. The multiplier based on the radix-4 algorithm is designed with current mode CMOS quaternary logic circuits. Designed multiplier is reduced the transistor count by 67.1% and 37.3%, compared with that of the voltage mode binary multiplier and the previous multiple-valued logic multiplier, respectively. The multiplier is designed with a 0.35 μ m standard CMOS technology at a 3.3 V supply voltage and unit current 10 μ A, and verified by HSPICE. The multiplier has 5.9ns of propagation delay time and 16.9mW of power dissipation. The performance is comparable to that of the fastest binary multiplier reported

Keyword : 다치논리회로, 곱셈기, 연산회로, VLSI

* 正會員, 江原大學校 電子工學科

(Dept. of Electronics Eng., Kangwon National University)

** 正會員, 江原大學校 電氣電子情報通信工學部

(Dept. of Electrical and Computer Eng., Kangwon National University)

※ 이 논문은 강원대학교 두뇌한국21(BK21) 사업에 의하여 지원되었으며, 본 연구에 사용된 설계용 CAD S/W는 IDEC(반도체 설계교육센터) 지원에 의한 것임.

接受日字:2003年8月18日, 수정완료일:2003年11月27日

I. 서론

반도체 기술의 급격한 발전으로 인해 집적도의 향상은 최근 괄목할 만한 성장을 이루었다. 이러한 집적도의 향상은 SoC(System-on-Chip)를 가능하게 했으며, 이로 인해 IC를 통한 대용량 시스템의 구현이 실현되었다. 그러나 SoC를 구현하기 위해서는 보다 많은 정보량의 처리 문제와 단자수 제한 문제, 단자간의 상호

이만큼의 전류가 M12를 통해 흐르게 된다. 즉 sum의 출력전류는 $I_{in}/4$ 가 되는 것이다. M11은 다이오드로써 역 전류 방지 및 M8과 M12 사이의 상호 전기적인 영향을 없애기 위한 완충 역할을 하게 된다.

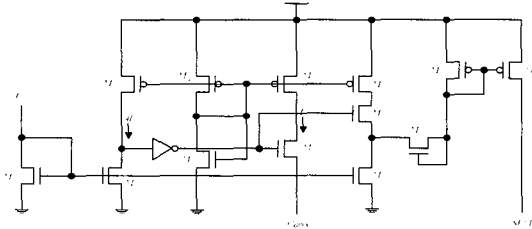


그림 3. 전류모드 CMOS 4치 논리 전가산기
Fig. 3. Current-mode CMOS quaternary logic full adder.

III. 다치 논리를 이용한 Modified Booth 곱셈기 설계

1. Booth 곱셈 알고리즘

모든 연산의 가장 기본은 가산과 곱셈이다. 따라서 최적의 가산기와 곱셈기를 사용하면 보다 효율적인 기능을 갖는 연산회로를 설계할 수 있다. 본 논문에서는 Radix 4 알고리즘을 이용한 Modified Booth 곱셈기를 설계하였다. Booth 곱셈기는 2의 보수 곱셈 연산을 효율적으로 수행할 수 있으며, 일반적인 곱셈기와 비교했을 때 부분 곱(partial product)의 수를 절반으로 감소 시킴으로써 연산 속도와 트랜지스터 수를 감소시키는 장점을 가지고 있다. Booth 부호기와 Booth 선택기의 사용으로 면적이 증가하지만 가산기 부분의 면적이 절반으로 감소하여 칩 전체 면적을 감소시킬 수 있다. 또한 n 비트 \times n 비트에서 $n/2$ (Radix-4) 또는 $n/3$ (Radix-8) 번의 가산만을 요구하고 $x \times y$ 에서 x, y 중 하나가 또는 둘 다 2의 보수일 때도 연산이 가능하다. 이 알고리즘은 승수의 인코딩 방법에 따라 Radix-2, Radix 4, Radix-8로 분리할 수 있다. 본 논문에서는 Radix-4 알고리즘을 사용하였다. 이것은 승수에서 나타나는 연속된 k 개의 '1'을 변환을 통해서 연산 횟수를 감소시킨다. 이론적으로 이것은 비트의 반을 0으로 하는 방식으로 승수를 다시 쓰는 되는데 이는 특별한 수의 체계를 사용함으로써만이 가능하다. 이것은 부호화된 표준 2's 기수를 십진수들이 집합 $\{-2, -1, 0, 1, 2\}$ 안에 존재하는 수 체계로 변환이 된다. 이러한 시스템에서는 어떤

한 수도 여러 가지 형태를 쓸 수 있다. 예를 들어 $B = b_{n-1}b_{n-2}\dots b_1b_0$ 를 2의 보수 형태로 나타내면 식 (1)과 같다.

$$B = -b_{n-1}2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i \tag{1}$$

이 식 (1)을 다시 쓰면 식 (2)와 같다.

$$\begin{aligned} B &= -b_{n-1}2 \cdot 2^{n-2} + b_{n-2}2^{n-2} + b_{n-3}2 \cdot 2^{n-3} - b_{n-3}2^{n-3} + \dots \\ &= (-2b_{n-1} + b_{n-2} + b_{n-3})2^{n-2} \\ &\quad + (-2b_{n-3} + b_{n-4} + b_{n-5})2^{n-4} + \dots \end{aligned} \tag{2}$$

이 식은 $b_i = 0$ 을 갖는 식으로 변경하면 식(3)으로 다시 쓸 수 있다.

$$B = \sum_{i=0}^{n/2-1} (-2b_{2i+1} + b_{2i} + b_{2i-1})2^{2i} \tag{3}$$

식 (3)에서 괄호 속의 항은 $\{-2, -1, 0, 1, 2\}$ 의 집합이다.

표 1. Radix-4 알고리즘 값에 따른 피승수의 동작

Table 1. Multiplicand by Radix-4 algorithm.

$(-2b_{2i+1} + b_{2i} + b_{2i-1})$	operation
1	load
-1	invert & +1
2	shift left
-2	shift left & invert & +1
0	set to 0

<표 1>에 Radix 4 알고리즘의 값에 따른 피승수의 동작을 나타내었다. 표에서 볼 수 있듯이 '1'의 값을 가질 경우에 피승수는 그 값을 그대로 읽고 '-1'인 경우에는 값을 반전시키고 1값을 더하게 된다. '2'의 값을 가질 경우에는 왼쪽으로 이동하여 2배의 값을 갖도록 하며, '-2'인 경우에는 반전시키고 1을 더하여 왼쪽으로 이동한다. '0'인 경우에는 아무런 동작도 하지 않는다. 두 개의 수 A와 B를 연산한다고 가정하자. 우선 Radix-4 알고리즘을 통해 승수 B를 인코딩하면 -2, -1, 0, 1, 2의 숫자를 갖는 B'으로 변환된다. 이러한 형태에서 B의 숫자의 반은 B'에 존재하게 된다. B'의 숫

자는 스캔되고 각 단계에서 피승수 A는 -2, -1, 0, 1, 2에 의해서 곱해진다.

2. 부호확장 제거방법

곱셈연산에서는 기본적으로 각 단계에서 부분 곱을 생성한 뒤에 다음 단의 부분 곱은 자리를 쉬프트하고 나중에 그 값들을 더하게 된다. <그림 4>에 간단한 예를 제시하였다.

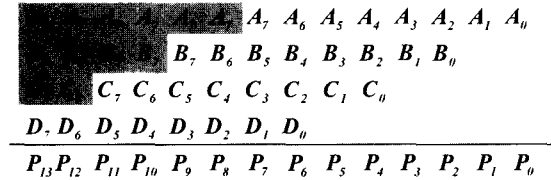


그림 4. 부호 확장 제거 방법의 예
Fig. 4. Example for the elimination of sign-bit extension.

음영으로 외각선 처리된 부분이 부호 확장된 부분이다. 이처럼 곱셈연산에서는 각 단마다 부호확장된 부분이 생기는데 간단한 연산에서는 문제가 되지 않지만 비트 수가 많아질수록 이 값을 연산하기 위해서는 부가적인 시간과 회로들이 필요하다. 그래서 부호확장된 부분을 제거하여 연산시간을 감소시키고 부가적인 회로를 사용하지 않음으로써 인한 면적을 감소시킬 수 있는 것이 부호확장 제거방법이다.

부호 비트 보수화 방법은 첫 번째 부호비트에 반전된 값을 취하고 확장 비트에 '1'을 더해준다. 이것을 식 (4)에 표현하였다. 그리고 첫 번째 단의 부분 곱에서부터 최상위 비트에 '1'값을 더해준다.

$$S S S S S X X X X X X X X \Rightarrow$$

$$1 1 1 1 1 \bar{S} X X X X X X X X$$

$$+ 1 \tag{4}$$

위의 관계를 정리하여 일반화시키면 <그림 5>와 같다.

지금까지 설명한 부호확장 제거방법을 전류모드 CMOS 4치 논리회로에 적용한 것을 <그림 6>에 나타내었다. 부호확장 제거기법을 사용한 Modified Booth 곱셈기의 첫 번째 단은 그림에서 보이는 것처럼 상단에서 내려오는 중간 결과가 없기 때문에 위와 같은 구조로 중간 결과를 만들어 낸다. 최상위 비트의 입력 값

$$1$$

$$1 \bar{A}_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$$

$$1 \bar{B}_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0$$

$$1 \bar{C}_7 C_6 C_5 C_4 C_3 C_2 C_1 C_0$$

$$\bar{D}_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$$

$$P_{13} P_{12} P_{11} P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$$

그림 5. 부호 확장 제거 방법
Fig. 5. The elimination of sign-bit extension.

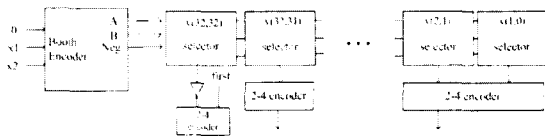


그림 6. 부호 확장 제거 방법을 이용한 곱셈기의 첫째 단 상세 회로
Fig. 6. First stage circuit of multiplier with elimination circuit of sign-bit extension.

을 보면 같은 값이 입력으로 들어가는데 이것은 부호확장을 위한 값이다. 그 값은 Booth 선택기로 들어가서 어떤 연산을 하는지 결과 값을 출력하게 되고 그 값을 인버터를 통해 반전시킨다. 이것은 앞에서 설명했듯이 부호비트를 반전시키는 역할을 한다. 그리고 첫 번째 단의 최상의 비트에 '1'값을 더해주는 역할은 First라고 표시된 회로가 하게 된다. 그러나 이 값들은 전압모드에서 동작하는 것이 아니라 전류모드에서 동작해야 하기 때문에 전압 값을 전류 값으로 변환시켜주는 회로를 사용하여야 한다.

3. 다치 논리를 이용한 Modified Booth 곱셈기 구조
전류모드 4치 논리 Modified Booth 곱셈기는 Booth 부호기와 선택기에 의해 만들어진 부분 곱이 전류모드 CMOS 2진-4치 논리변환 부호기에 입력되어 전류 값으로 변하게 된다. 그 값이 실제 연산하는 전류모드 전가산기로 입력되어 sum과 carry를 출력하게 된다. 이

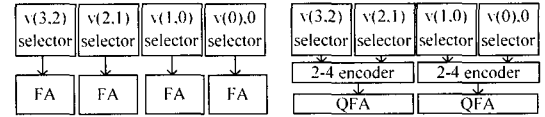


그림 7. Modified Booth 곱셈기의 배열 구조 (a)전압 모드 2진 논리 (b)전류모드 4치 논리
Fig. 7. Array architecture of Modified Booth multiplier. (a) voltage-mode binary logic (b) current-mode quaternary logic

것은 기존의 전압모드 2치 논리회로와 비교했을 때 전류 값을 전압 값으로 또는 전압 값을 전류 값으로 변환해 주기 위한 회로가 부가적으로 들어가는 단점을 가지고 있다.

그러나 곱셈기를 구현하기 위해 사용된 전가산기의 수가 절반으로 줄어들고, 연산하는 전류모드 전가산기 또한 기존의 전압모드 전가산기에 비해 훨씬 적은 트랜지스터를 사용하기 때문에 전체적으로 사용한 트랜지스터가 감소하는 장점을 가지고 있다. 또한 부분 곱을 한 비트씩 실행하는 2치 논리와는 달리 부분 곱을 두 비트씩 묶어서 연산을 실행하기 때문에 전체 시스템의 상호연결 단자수가 감소하는 장점을 가지고 있다. <그림 7>은 전압모드 2치 논리로 구현할 때의 Booth 곱셈기와 전류모드 4치 논리로 구현할 때의 Booth 곱셈기의 구조를 개략적으로 나타내었다.

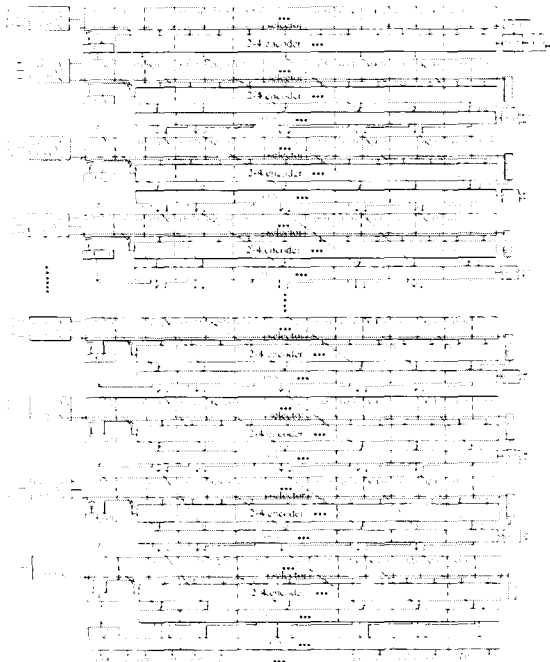


그림 8. 전류모드 CMOS 4치 논리를 이용한 32×32 bit Modified Booth 곱셈기
Fig. 8. A 32×32-bit Modified Booth multiplier using current-mode CMOS quaternary logic.

본 논문에서 구현한 전류모드 CMOS 4치 논리를 이용한 Modified Booth 곱셈기의 구조를 <그림 8>에 나타내었다. 이것은 셀 단위로 구성하여 확장성이 용이하도록 설계하였다. 곱셈기의 구조는 우선 승수는 세 비

트씩 묶어서 Booth 부호기에 입력한다. 그러나 상위 비트부터 묶으면 마지막 하위 비트는 두 비트밖에 남지 않기 때문에 그 빈자리에는 '0'값을 입력한다. 또한 세 비트씩 묶을 때는 꼭 한 비트씩 겹쳐서 묶어야 한다. 이렇게 Radix-4 알고리즘을 사용함으로써 일반적인 곱셈기와 비교했을 때 전체 열수가 $N/2$ 로 감소하기 때문에 면적 및 연산시간이 감소한다. Booth 부호기를 통해 나온 값은 피승수 두 비트와 함께 Booth 선택기의 입력으로 들어가서 부분 곱을 생성한다. Booth 선택기의 출력 값은 전류모드 CMOS 2치-4치 논리변환 부호기의 입력 값으로 들어가서 전류 값으로 변환되며 각 단의 최하위 비트에 사용된 마지막 2치-4치 논리변환 부호기는 전류모드 전가산기의 입력 값으로 들어가서 출력 값으로 나오게 된다. 이때 나오는 값은 전류 값이므로 다시 전압 값으로 변환해 주는 전류모드 CMOS 4치-2치 논리변환 부호기를 거치게 된다. 각 단의 최하위 비트에서 나온 2치-4치 논리변환 부호기의 값을 제외한 나머지 값들은 다음 단의 전가산기의 입력 값으로 들어가서 연산하게 된다. 또한 최하위 비트의 N 이라고 표시된 박스부분은 Booth 부호기에서 나온 제어 신호 중에서 음수 값을 가질 경우에는 단위전류 $10\mu A$ 를 인가하여 '1'값을 더해주는 역할을 하게 된다. 각 단의 최상위 비트는 부호확장 제거방법을 사용해서 한 비트씩 연장한 값이다. 이 비트 값으로 Booth 선택기에서 나온 값은 인버터를 거쳐 반전되고 그 값이 전류모드 CMOS 2치-4치 논리변환 부호기에 입력되어 연산하게 된다. 마지막 단에서는 전가산기의 sum값이 4치-2치 논리변환 부호기로 입력되어 전압 값으로 출력된다.

본 논문에서 설계한 Modified Booth 곱셈기는 Radix-4 알고리즘의 장점과 다치 논리의 장점을 이용하였다. 즉 Radix-4 알고리즘의 사용으로 승수를 세 비트씩 묶어서 연산하기 때문에 일반적인 곱셈기보다 $N/2$ 의 열로 단수가 줄임으로써 면적에 대한 오버헤드를 감소시켰고 다치 논리의 장점을 이용하여 연결 단자수 문제를 해결하였으며 곱셈기를 구현하기 위해 사용된 트랜지스터의 수도 감소시켰다. 그리고 셀 단위로 회로를 구성하여 확장성이 용이하도록 설계하였고 전류모드와 전압모드의 형태를 변화 시켜주는 부호기와 복호기의 사용으로 인해서 기존의 2치 논리회로와 호환성을 갖도록 하였다.

IV. 시뮬레이션 결과 및 비교

1. 시뮬레이션 결과 및 배치설계

본 논문에서는 0.35 μ m CMOS 기술을 이용하여 HSPICE로 시뮬레이션 하였다. 모델 파라미터는 Level 28을 사용하였고 공급전원은 3.3V를 사용하였다. 4치 논리신호의 단위전류는 10 μ A를 사용하여 각각의 신호 값을 0 μ A, 10 μ A, 20 μ A, 30 μ A로 정의하였다. 출력노드에 흐르는 전류를 측정하기 위해서 출력 단에는 10k Ω 의 저항을 연결하였고 전압 출력노드에는 200fF의 캐패시터를 연결하여 측정하였다. <그림 9>는 본 논문에서 설계한 곱셈기의 전체 배치설계이다.

2. 비교 및 분석

<표 2>는 본 논문에서 설계한 회로와 이전 연구에서 설계한 곱셈기를 비교한 표이다. 전압모드 2치 논리 곱셈기와 비교했을 때 트랜지스터 수는 63.2% 감소하였으며 전류모드 4치 논리로 구성된 곱셈기와 비교했을 때는 37.3%가 감소하였다. 이로 인해 전체 면적도 2진 논리회로에 비하여 39.8%감소하였다. 기존의 전류모드 4치 논리 곱셈기는 회로를 표준 CMOS 공정으로 구성된 것이 아니라 표준 CMOS 공정에 공핍모드 PMOS 공정을 추가한 공정을 사용하였으며, 연산방법에서도

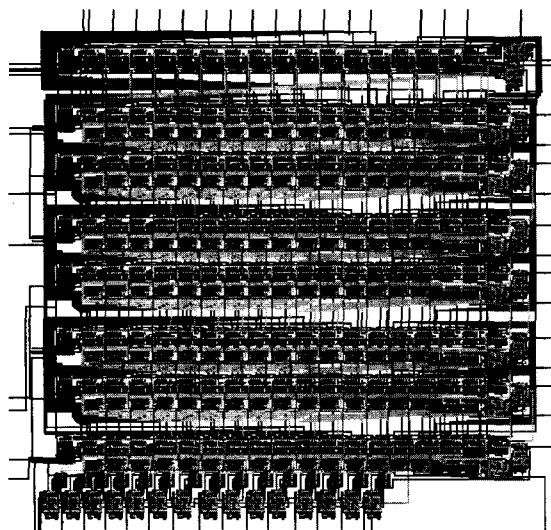


그림 9. 전류모드 4치 논리를 이용한 Modified Booth 곱셈기의 배치설계

Fig. 9. Mask layout of Modified Booth Multiplier using current-mode CMOS quaternary logic.

양방향 전류 값을 이용하였다. 그러나 본 논문에서는 표준 CMOS 공정을 사용하여 설계하였으며 전류모드 CMOS 회로의 단점의 하나인 잡음여유가 작다는 것을 감안하여 단방향으로 연산하는 방법을 사용하여 회로 동작의 오류 가능성을 최소화시켰다.

표 2. 비교표

Table 2. Comparison table.

	2치 논리 곱셈기	Shoji의 곱셈기[3]	본 논문의 곱셈기
트랜지스터 수	40,626	23,600	14,925
점유면적(mm ²)	3.8×4.8	Not Reported	2.7×4.2
평균 전력소모 (mW)	15.8	Not Reported	16.9
최대 전달지연 (ns)	5.3	Not Reported	5.9
공정기술(μ m)	0.35 μ m CMOS	2 μ m CMOS + 공핍모드 PMOS	0.35 μ m CMOS
공급전압(V)	3.3	3.3	3.3

설계한 곱셈기 회로의 최대 전달지연은 5.9ns, 평균 전력소모는 16.9mW이다. 반면에, 2치 논리 곱셈기는 최대 전달지연 5.3ns, 평균 전력소모 15.8mW으로 본 논문에서 설계한 곱셈기 보다 우수한 성능을 보이고 있다. 따라서 지금까지의 연구결과를 종합해 보면 트랜지스터 수와 점유면적은 기존 2치 논리 곱셈기에 비해 월등히 우수하지만, 최대 전달지연과 평균 전력소모는 아직 개선해야 할 특성임을 알 수 있다. 또한 4치 논리를 적용함으로써 발생하는 잡음여유의 감소 문제가 발생한다. 이러한 특성은 앞으로 전류모드 CMOS 다치 논리회로가 반드시 해결해야 할 문제이다.

V. 결론

Radix-4 알고리즘을 이용한 Modified Booth 곱셈기의 장점과, 다치 논리회로의 장점을 이용하여 새로운 전류모드 CMOS 4치 논리 Modified Booth 구조를 설계하였다. 이 곱셈기는 Radix-4 알고리즘을 이용함으로써 승수를 세 비트씩 묶어서 연산하게 됨으로 N/2열로 단수 및 연산시간을 단축시켰으며 연산과정에서 발생하는 부호비트를 제거함으로써 회로의 면적을 감소시켰다. 또한 다치 논리의 장점인 단자수 제한 문제와 단자간 상호연결 문제를 응용하여 연산하는 방법을 기존의 전압모드 2치 논리가 아닌 전류모드 4치 논리에 응

용함으로써 기존의 전압모드 회로와 비교했을 때 회로가 훨씬 단순해짐을 확인하였다. 그러나 낮은 단위전류를 사용함으로써 각각의 트랜지스터의 길이와 폭이 증가하게 되었고 그로 인해 기생 커패시턴스에 의한 전달지연이 증가하게 되었다. 이러한 문제의 해결 방법으로는 낮은 공급전원의 사용을 통한 전체 트랜지스터의 길이와 폭의 비를 감소시키거나 클럭에 의해 동기되는 동적 전류모드 CMOS 회로를 활용해 DC 전력 소모를 제거하는 것이 그 대안이 될 것이라고 생각한다.

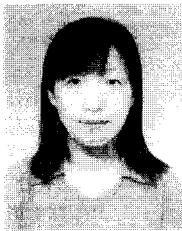
본 논문에서 설계된 곱셈기는 기존의 2차 논리 시스템과 호환성을 갖고 있으며 규칙적인 셀 구조를 가짐으로써 확장성을 갖는다. 또한 전류모드 CMOS 4차 논리를 적용하여 상호 연결 노드를 감소시켰으며 트랜지스터 수를 감소시켜 전체적인 면적 감소 효과를 얻을 수 있었다. 더욱이 0.35 μ m CMOS 기술을 통해 그 동작을 확인함으로써 전류모드 회로의 문제점 중의 하나인 미세 선폭 반도체 기술 적용에 대한 가능성을 확인함으로써 향후 VLSI 실현에 적합할 것으로 기대된다.

참 고 문 헌

[1] 이용섭, 곽철호, 김정범, "전류모드 다치논리 CMOS 회로를 이용한 전가산기 설계" 전자공학회 논문지, 제39권, SD편, 제1호, pp. 76-82, 2002년 1월
 [2] K. Wayne Current, "Current-Mode CMOS

Multiple-Valued Logic Circuits" IEEE J. Solid-State Circuits, vol.29, No.2, pp. 95-107, Feb. 1994.
 [3] Shoji Kawahito, Michitaka Kameyama, Tatsuo Higuchi, and Haruyasu Yamada, "A 32 \times 32-bit Multiplier Using Multiple-Valued MOS Current-Mode Circuits" IEEE J. Solid-State Circuits, vol.23, No.1, pp.124-132, Feb. 1988.
 [4] Gensuke Goto, Atsuki Inoue, Ryoichi Ohe, Shoichiro Kashiwakura, Shin Mitarai Tsuru, and Tetsuo Izawa, "A 4.1-ns Compact 54 \times 54-b Multiplier Utilizing Sign-Select Booth Encoders" IEEE J. Solid-State Circuits, vol.32, No.11, pp. 1676-1682, Nov. 1997.
 [5] K. Wayne Current, "Application fo quaternary logic to the design of a proposed discrete cosine transform chip" Int. J. Electronics, vol. 67, No.5, pp. 678-701, 1989.
 [6] Masayuki Mizuno, Masakazu Yamashina, Koichiro Furuta, Hiroyuki Igura, Hitoshi Abiko, Kazuhiro Okabe, Atsuki Ono, and Hachiro Yamada "A Ghz MOS Adaptive pipeline Technique Using MOS Current-Mode Logic" IEEE J. Solid-State Circuits, vol.31, No.6, pp. 784-791, June 1996.

저 자 소 개



李 恩 實(正會員)
 2000년 2월 : 강원대학교 전자공학과 졸업(학사). 2002년 2월 : 강원대학교 대학원 전자공학과 졸업(공학석사). 2002년 3월~현재 : (주)다반테크 SOC 사업본부ASIC 개발팀 연구원. <주관심분야 : SOC설계, VLSI설계, Multi-Valued Logic>



金 政 範(正會員)
 1985년 2월 : 인하대학교 전자공학과 졸업(학사). 1987년 2월 : 인하대학교 대학원 전자공학과 졸업(공학석사). 1997년 2월 : 포항공과대학교 전자전기공학과 졸업(공학박사). 1987년 1월~1992년 5월 : 금성반도체 중앙연구소 선임연구원. 1994년 8월~1997년 9월 : 현대전자 시스템 IC 연구소 책임연구원. 1997년 9월~1999년 2월 : 충북대학교 전기전자공학부. 1999년 3월~현재 : 강원대학교 전자공학과 부교수. <주관심분야 : VLSI설계, CAD, Multi-Valued Logic>