
인코더, 디코오더를 가지는 다치 연산기 설계

박진우*, 양대영*, 송홍복*

Design of a Multi-Valued Arithmetic Processor with Encoder and Decoder

Jin-Woo Park*, Dae-Young Yang*, Hong-Bok Song*

요약

본 논문에서는 다치 논리를 이용한 연산기를 설계하였다. 다치 논리를 구현하기 위해서 전류모드 CMOS 회로를 이용하였으며 이진 전압모드 신호를 다치 전류모드 신호로 바꾸어 주는 인코더와 연산 결과인 다치 전류모드 신호를 이진 전압모드 신호로 바꾸어 주는 디코오더를 사용하여 기존의 이진 시스템에 적용할 수 있도록 하였으며, 승산기 설계시 부분곱 수를 줄이기 위하여 기존의 Booth 알고리즘을 확장한 4진 SD수 부분곱 발생 알고리즘을 사용하였다. 제안된 회로는 SPICE 시뮬레이션 및 FPGA Chip을 이용한 하드웨어 에뮬레이션으로 그 유효함을 확인하였다.

Abstract

In this paper, an arithmetic processor using multi-valued logic is designed. For implementing of multi-valued logic circuits, we use current-mode CMOS circuits and design encoder which change binary voltage-mode signals to multi-valued current-mode signals and decoder which change results of arithmetic to binary voltage-mode signals. To reduce the number of partial product we use 4-radix SD number partial product generation algorithm that is an extension of the modified Booth's algorithm. We demonstrate the effectiveness of the proposed arithmetic circuits through SPICE simulation and Hardware emulation using FPGA chip.

* 동의대학교 전자공학과

접수일자 : 1998년 2월 26일

1. 서 론

최근 반도체 기술의 발전으로 인해 1개의 칩(chip)에 넣을 수 있는 회로의 규모와 기능이 점점 증가하여 수 천만개의 소자를 가진 집적회로가 등장하고 있다. 이러한 현재의 대부분의 디지털 회로는 신호 레벨로서 2치 표현을 기본으로 한 논리 방식이 채택되어져 있으나 서브 마이크로화가 진행됨에 따라 현재의 기술상의 2치 논리 방식의 문제점이 몇 가지 나타나기 시작하였는데, 먼저 VLSI에서 칩(Chip) 실효면적중에 내부 배선이 70%~90% 정도의 비율로 차지하고 있어 내부 배선의 복잡성 처리 및 배선 감소가 VLSI의 성능향상의 열쇠가 되고 있다.^[1] 즉 회로 기능과 병렬 구조의 고도화의 요구에 따라서 셀(Cell), 기본 블록 및 모듈(Module)간의 배선이 증가하기 때문에 배선 영역 및 배선 용량·저항의 증대 등에 의해서 성능이 저하하는 결과를 가져오게 된다. 그리고 칩(Chip) 상에 병렬 승산기 등을 내장한 마이크로 프로세스와 신호 처리 프로세스등이 계속 개발되어지고 있으며 다수의 연산기를 이용한 초병렬 처리 시스템 및 초고성능화를 목표로 많은 연구가 진행되고 있으나 연산기간의 통신시 일어나는 성능 저하의 문제가 대두되기 때문에 기존의 2치 방식은 고성능화에 장애가 되고 있다. 이와 같이 기본이 되는 고평형 연산 알고리즘을 실행하기 위해서는 2치 데이터 표현이 적합하다고 할 수 없으며 가능한 적은 게이트(Gate) 수로 연산 결과를 얻기 위해서 2치 데이터 보다는 다치 부호화 데이터 표현이 유용하다.^{[2][3][4]} 이와 같이 다치논리를 이용할 경우에는 배선의 복잡성을 감소시켜 회로의 연산 속도를 높일 수 있으며 직렬 접속의 단수 및 연산 반복횟수를 감소시킴으로써 연산 속도를 고속화할 수 있다.^{[5][6][7]}

본 논문에서는 일반 이진 시스템에 적용할 수 있도록 인코더, 디코오더를 가지는 다치 가산기 및 다치 승산기를 설계하였다. SD(Signed-Digit) 수를 사용하며 다치 논리를 구현하였으며, 기존의 Booth 알고리즘을 다치 승산기에 적용한 확장 4진 SD수 부분적 생성 알고리즘을 사용하여 부분곱 수를 줄였다. 회로에 대한 검증은 SPICE 시뮬레이션 및

FPGA를 이용한 하드웨어 에뮬레이션으로 검증을 하였다

II. 다치 전류모드 CMOS의 기본회로

SD수 연산회로는 다치 쌍방향 전류모드 CMOS 기본 블록으로 효율적으로 구성 되어질 수 있으므로 단순화·간략화 될 수 있어 적은 능동소자와 배선수로 연산회로를 구성할 수 있다. 또한 다치 전류모드 CMOS 회로에서 뛰어난 장점은 가산, 감산을 능동소자가 아닌 결선만으로 구성될 수 있으며 기본회로는 그림 1과 같다.

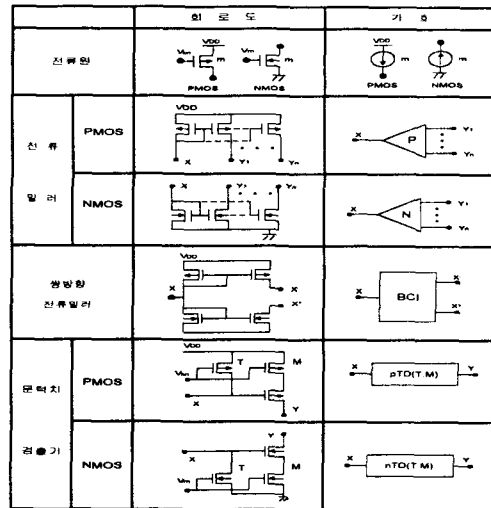


그림 1. 전류모드 CMOS회로의 기본 회로
Fig. 1. Basic circuits of current mode CMOS

- a) 전류원 : 기본 전압 V_m (또는 V_{rp})을 갖는 nMOS(또는 pMOS)로 구성되어질수 있으며 전류의 양은 MOS의 채널의 비(L/W의 비)에 의해서 제어되어진다.
- b) 전류 Miller : 입력전류를 복사하는 기능과 반전시키는 기능을 가지고 있으며 pMOS과 nMOS 형 전류 Miller가 있다.
- c) 쌍방향 전류 Miller : + · - 전류를 모두 처리할수 있도록 pMOS Miller와 nMOS Miller를 붙여 놓은 것이다. 입력의 극성을 검출해서 2개의 단방향 전류로 분해하는 기능을 가지는데,

pMOS 밀러는 -전류를 처리하고 nMOS 밀러는 +전류를 처리한다.

d) 문턱치 검출기 : 문턱치 검출기는 입력 전류값(X)과 비교 전류값(T)를 비교하여 입력 전류값(X)이 클 경우 새로운 전류원(M)이 출력으로 나오고 적을 경우에는 출력측에 전류가 흐르지 않게 된다.

본 논문에서 사용되는 기준전류 I_0 는 $20\mu A$ 로 하였다. 다치 CMOS회로에서 기준전류를 아주 짧게 잡으면 노이즈 영향이 출력에 나타나서 불안정하게 된다. 따라서 전류의 변동이 있다고 해도 그 영향이 출력에 나타나지 않고 충분히 동작 가능한 최소 기준전류가 $20\mu A$ 정도였다. 논리치와 전류치의 대응관계를 표 1에서 나타내었다.

표 1. 논리치와 전류치의 대응관계

Table 1. Relation between logical values and logical currents

논리치	-3	-2	-1	0	1	2	3
전류치	$-3I_0$	$-2I_0$	$-I_0$	0	I_0	$2I_0$	$3I_0$

Ⅲ. 인코더, 디코더의 설계

다치 연산회로를 일반 2진 시스템에 연계를 시키기 위한 블록도는 다음과 같다.

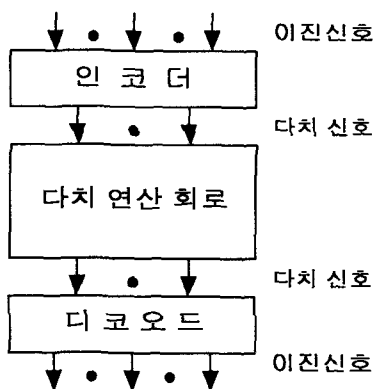


그림 2. 2진 시스템과 연계를 위한 블록도
Fig 2. Block diagram for linking Binary system

1. 인코더의 구성

2진 전압모드 수를 4진 전류모드 SD수로 변환하기 위한 인코더 블록도는 다음과 같다.

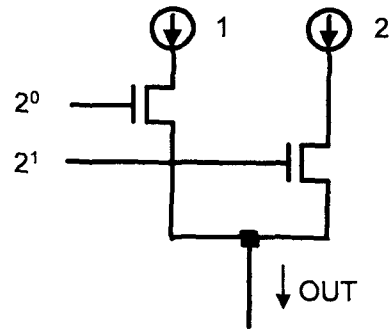


그림 3. 인코더
Fig 3. Block diagram of Encoder

예를 들어 위 인코더에 2진수 $10(2^1=1, 2^0=0)$ 이 입력될 경우의 출력 OUT의 과도응답 특성은 다음과 같다.

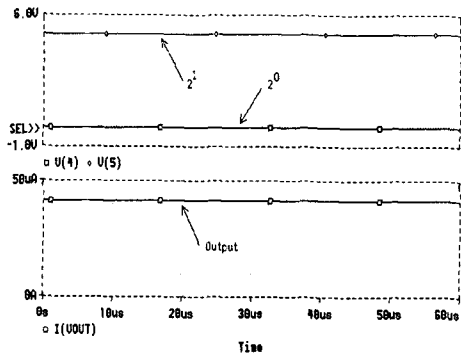


그림 4. 인코더의 과도응답 특성
Fig. 4. Transient response characteristics of Encoder

그림 4에서 출력 결과(Output: $40\mu A$)는 논리치 2이므로 입력 값 이진수 10과 일치함을 알 수 있다.

2. 디코더의 구성

연산 결과를 2진수로 바꾸기 위해 디코더 설계시 고려할 사항은 SD수 연산 결과가 양수와 음수 둘다 가지고 있으므로 둘다 처리할 수 있는

디코더가 설계되어야 하며 그 구성도는 그림 5와 같다.

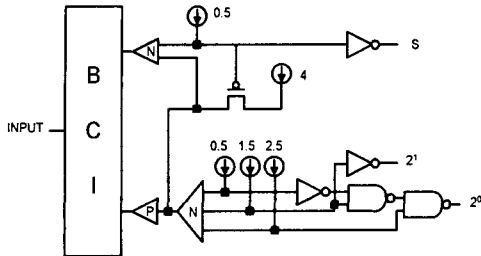


그림 5. 디코더
Fig. 5. Block diagram of Decoder

입력이 4진수이므로 입력 논리치를 {-3,-2,-1,0,1,2,3}으로 했을때의 디코더의 과도응답 특성은 다음과 같다.

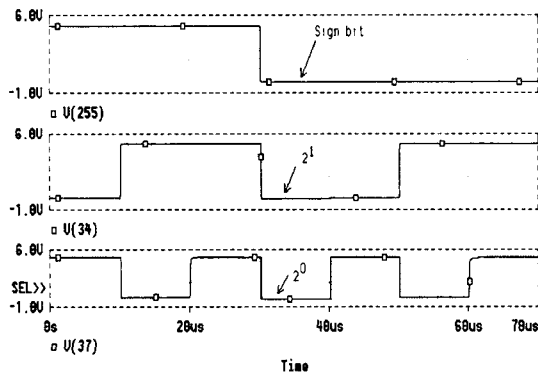


그림 6. 디코더의 과도응답 특성
Fig. 6. Transient response characteristics of Decoder

그림 6의 디코더에서 음수처리는 2의 보수를 취하고 있다.

IV. SD 가산기

1. 4진 7치 전가산기 구성

제안된 병렬 다치 가산기 및 다치 승산기는 4진 7치 전가산기를 기본으로 하여 구성된다. 4진 SD수는 아래와 같이 7개의 값을 갖는 대칭적인 디지털(Digit)로 구성된다.

$$L = \{-3, -2, -1, 0, 1, 2, 3\} \dots\dots\dots (1)$$

4진 SD수에 있어서 X, Y의 가산은 다음과 같이 3개의 식에 의해서 행하여진다.

$$Z_i = X_i + Y_i \dots\dots\dots (2)$$

$$4C_i + W_i = Z_i \dots\dots\dots (3)$$

$$S_i = W_i + C_{i-1} \dots\dots\dots (4)$$

여기서 선형합 $Z_i \in \{-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6\}$, 중간합 $W_i \in \{-2, -1, 0, 1, 2\}$, 캐리 $C_i \in \{-1, 0, 1\}$, 최종합 $S_i \in \{-3, -2, -1, 0, 1, 2, 3\}$ 이다. 제안된 가산기는 전류모드 CMOS 회로를 이용하기 때문에 식(2)와 식(4)는 선형 결선에 의해서 구성되며 식(3)는 중간합 W_i 와 캐리 C_i 를 갖는 다음 3식에 의해서 행해진다.

$$W_i = Z_i - 4, C_i = 1 \text{ (} Z_i \geq 2 \text{의 경우)} \dots\dots\dots (5)$$

$$W_i = Z_i, C_i = 0 \text{ (-2 < } Z_i < 2 \text{의 경우)} \dots\dots\dots (6)$$

$$W_i = Z_i + 4, C_i = -1 \text{ (} Z_i \leq -2 \text{의 경우)} \dots\dots\dots (7)$$

위와 같은 결과를 SD 전가산기로 구성하면 그림 2와 같다.

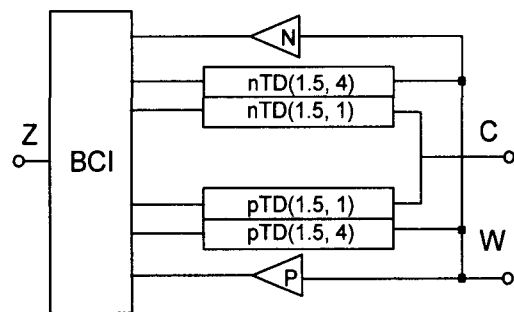
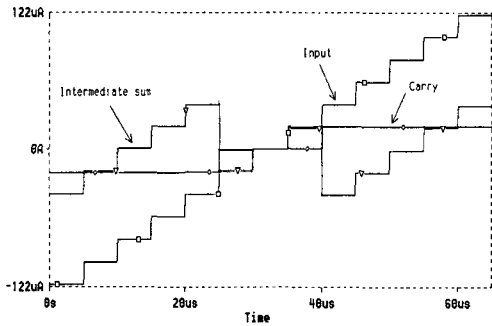


그림 7. 4진 7치 SD 전가산기(SDFA)
Fig 7. Block diagram of 4-radix 7-valued S DFA

그림 8은 SPICE 시뮬레이션을 이용해서 4진 7치 SD 전가산기(SDFA)의 과도응답 특성을 나타낸 것이다.



□ : 입력, ◇ : 캐리, ▽ : 중간합
 그림 8. 그림 2의 과도응답 특성
 Fig. 8. Transient response characteristics of Fig. 7

2. SD 전가산기(SDFA)를 이용한 병렬연산
 제안된 4진 7차 SD 전가산기(SDFA)를 이용해서 병렬 가산을 할 경우 그림 9와 같은 방법으로 구성하면 자리 올림 신호의 전송이 단수에 관계없이 1단에서 실행되어질 수가 있다.

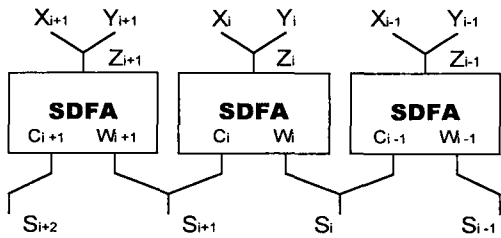


그림 9. SD 전가산기(SDFA)를 이용한 병렬 가산
 Fig. 9. Parallel add using SD full adder(SDFA)

V. SD 승산기

입출력을 이진수로 하는 승산기에 있어서도 4진 SD수를 내부에 이용함으로써 고속성 및 고집적화가 가능한 승산기를 만들 수 있는데, 그림 10에서 SD 승산기 블록도를 나타내었다. SD수 승산기는 부분적 발생기(PPG), 최종합 출력단으로 나눌 수가 있으며, 본 논문에서는 입력은 이진수, 출력은 4진 SD수로 하였으나 디코오더를 이용하면 출력도 간단히 이진수로 변환할 수가 있다.

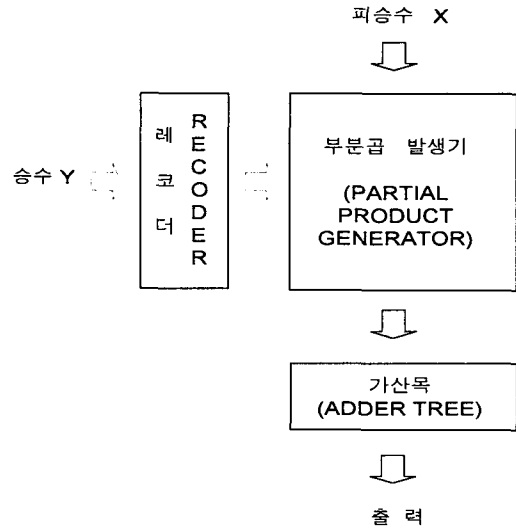


그림 10. SD수를 이용한 승산기
 Fig. 10. Multiplier using SD number

1. 부분곱 발생기

승산기에 있어서 중요한 것은 부분곱 수를 줄이는 것이다. 기존의 2진 시스템에서 부분곱 수를 줄이기 위해 Booth 알고리즘을 이용하고 있다.^[8] 본 논문에서는 기존의 Booth 알고리즘을 다치논리에 적용하여 3bit까지 시프트가 가능한 4진 SD수 부분곱 발생 알고리즘을 제안하였으며 그림 2에 부분곱 발생기를 나타내었다.

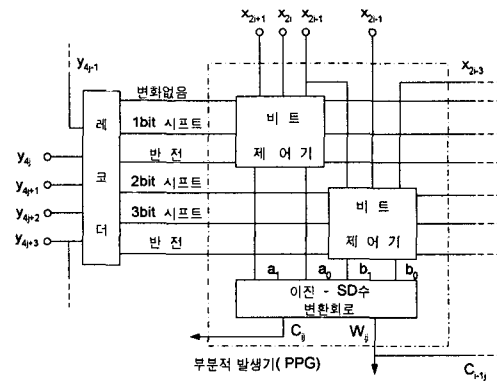


그림 11. 부분곱 발생기 (PPG)
 Fig. 11. Partial product generation (PPG)

1-1. 레코드(Recoder)

2의 보수 이진수 표현에서 승수 는 식(8)과 같이 나타낼 수 있다.

$$Y = (y_{n-1}y_{n-2} \dots y_1y_0) = -y_{n-1}2^{n-1} + \sum_{j=0}^{n-2} y_j 2^j \dots\dots\dots (8)$$

식(8)은 다음과 같이 다시 쓸 수가 있다.

$$Y = (y_{n-5} + y_{n-4} + 2y_{n-3} + 4y_{n-2} - 8y_{n-1})2^{(n-4)} \dots\dots + (y_{4j-1} + y_{4j} + 2y_{4j+1} + 4y_{4j+2} - 8y_{4j+3})2^{4j} \dots\dots + (y_3 + y_4 + 2y_5 + 4y_6 - 8y_7)2^4 + (y_{-1} + y_0 + 2y_1 + 4y_2 - 8y_3)2^0 = \sum_{j=0}^{n/4-1} (y_{4j-1} + y_{4j} + 2y_{4j+1} + 4y_{4j+2} - 8y_{4j+3})2^{4j} = \sum_{j=0}^{n/4-1} Q_j 16^j \dots\dots\dots (9)$$

여기서 $y_{-1}=0$,

$Q_j = y_{4j-1} + y_{4j} + 2y_{4j+1} + 4y_{4j+2} - 8y_{4j+3}$ 이다. Q_j 는 계산의 쉽게 하기 위해서 적절한 U_j 와 V_j 로 다음과 같이 나눌 수가 있다.

$$Q_j = U_j + V_j; \dots\dots\dots (10)$$

U_j 부분에서는 반전과 1bit 시프트 신호를 만들어 내고 V_j 부분에서는 반전, 2bit, 3bit 시프트 신호를 만들어 내게 된다. 여기서 Q_j 와 U_j, V_j 의 관계를 표 2에 나타내었다.

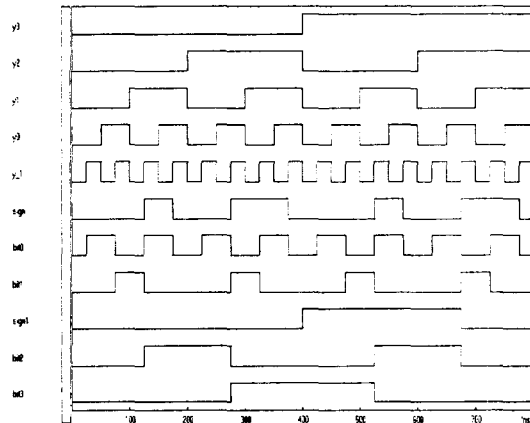
표 2. Q_j 와 U_j, V_j 의 관계
Table 2. Relation between Q_j and U_j, V_j

Q_j	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
U_j	0	1	2	-1	0	1	-2	-1	0	1	2	-1	0	1	-2	-1	0
V_j	-8	-8	-8	-4	-4	-4	0	0	0	0	0	4	4	4	8	8	8

X를 피승수라고 했을 때 식 (10)의 Q_j 와 U_j, V_j 의 곱은 다음과 같이 나타낼 수 있다.

$$P = Q_j X = \sum_{j=0}^{n/4-1} (U_j X + V_j X) 16^j \dots\dots\dots (11)$$

여기서 $U_j X \in \{-2X, -X, 0, X, 2X\}$, $V_j X \in \{-8X, -4X, 0, 4X, 8X\}$ 이다. 식(11)의 연산은 시프트와 반전회로로 구성되며, 8X, 4X 그리고 2X는 3, 2 그리고 1bit 시프트에 각각 해당되고 -X는 반전을 의미하게 된다. 그림 12는 레코드 부분을 FPGA (Xilinx사의 XC4003 이용)로 다운로드 하기 전 단계인 로직 시뮬레이션 결과이다.



(y3:y4j+3, y2:y4j+2, y1:y4j+1, y0:y4j, y-1:y4j-1, sign:반전, bit0:변화없음, bit1:1bit 시프트, sign1:반전, bit2:2bit 시프트, bit3:3bit 시프트)

그림 12. 레코드의 로직 시뮬레이션
Fig. 12. Logic simulation of RECODER

1-2. 비트 제어기

레코더에서 나온 제어 신호를 이용하여 아랫단에서의 3bit를 포함한 피승수(X) 5bit를 제어하는 부분으로써 가산목 단을 줄이기 위해 피승수 X는 다음과 같이 나타낼 수 있다.

$$X = -x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i \quad (j=\text{홀수}) \dots\dots (12)$$

$$X = x_{n-1}2^{n-1} - \sum_{i=0}^{n-2} x_i 2^i - 1 \quad (j=\text{짝수}) \dots\dots (13)$$

식(11)의 피승수 X와 승수 Y의 곱은 다음과 같이 나타낼 수 있다.

$$U_j X = -a_{n,j} 2^n + \sum_{i=0}^{n-1} a_{ij} 2^i + d, \quad (j=\text{홀수})$$

$$V_j X = -b_{n+2,j} 2^{n+2} + \sum_{i=0}^{n+1} b_{ij} 2^i + e, \quad \dots\dots\dots (14)$$

$$U_j X = a_{n,j} 2^n - \sum_{i=0}^{n-1} a_{ij} 2^i - d, \quad (j=\text{짝수})$$

$$V_j X = b_{n+2,j} 2^{n+2} - \sum_{i=0}^{n+1} b_{ij} 2^i - e, \quad \dots\dots\dots (15)$$

여기서 d_j 와 e_j 는 2의 보수 연산에 의한 1의 가산이다. 식 (14)(15)에서 a_{ij} 와 b_{ij} 는 레코더에서 나온 제어신호를 사용해서 비트 제어기 부분에서 처리될 수 있으며 그림 13에서 나타내었다.

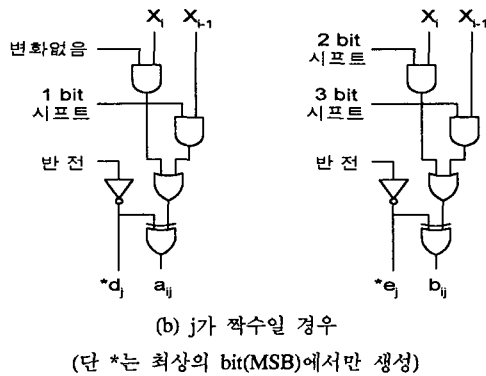
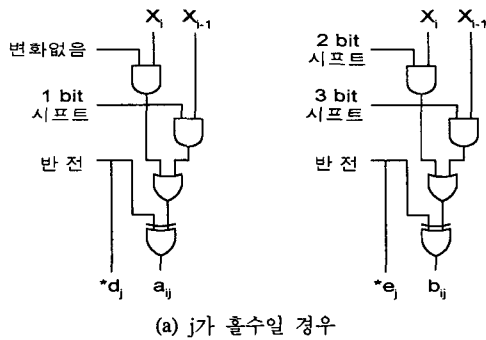


그림 13. 비트 제어기
Fig. 13. Block diagram of bit controller

1-3. 이진-SD 변환기

비트 제어기에서 나온 $U_j X$ 와 $V_j X$ 의 결과는 이진수 표현이므로 빠른 가산을 위해 4진 SD수로 변환할 필요가 있다. 2진수 표현으로 된 $U_j X$ 와

$V_j X$ 를 2bit 단위로 모아서 4진수로 변환한 다음, 앞에 나온 SD 가산기를 이용하여 가산하면 된다. 변환 알고리즘은 SD수 병렬가산과 같은 모양이고 각 단에 있어서 다음의 2단계의 방법으로 구한다.

① $U_j X$ 의 결과 2bit ($a_{2i+1,j} a_{2ij}$)와 $V_j X$ 의 결과 2bit ($b_{2i+1,j} b_{2ij}$)를 다음과 같은 방법으로 선형합을 구한다.

$$z_{n/2,j} = -4b_{n+2,j} + 2b_{n+1,j} + b_{n,j} - a_{n,j} \quad (MSB \text{ 부분}, j=\text{홀수}) \quad \dots\dots\dots (16)$$

$$z_{ij} = 2a_{2i+1,j} + a_{2i,j} + 2b_{2i+1,j} + b_{2i,j} \quad (MSB \text{ 제외부분}, j=\text{홀수}) \quad \dots\dots\dots (17)$$

$$z_{n/2,j} = 4b_{n+2,j} - 2b_{n+1,j} - b_{n,j} + a_{n,j} \quad (MSB \text{ 부분}, j=\text{짝수}) \quad \dots\dots\dots (18)$$

$$z_{ij} = -2a_{2i+1,j} - a_{2i,j} - 2b_{2i+1,j} - b_{2i,j} \quad (MSB \text{ 제외부분}, j=\text{짝수}) \quad \dots\dots\dots (19)$$

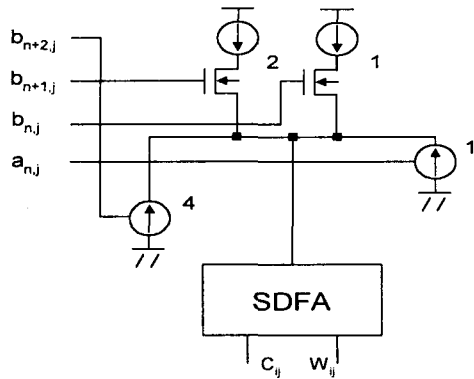
② 선형합 z_{ij} 에서 SDFA를 이용하여 중간합(w_{ij}), 캐리(c_{ij})를 구한다.

$$4c_{ij} + w_{ij} = z_{ij} \quad \dots\dots\dots (20)$$

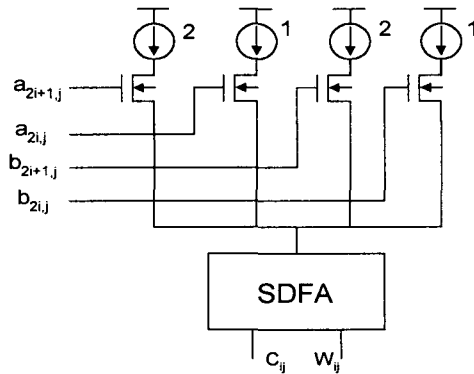
표 3은 식(17)에 대한 진리표이며, 그림 14는 식 (16)(17)에 대한 이진-SD변환기의 논리도를 나타낸 것이다.

표 3. 이진-SD 변환기의 진리표
Table 3. Truth table of Binary-SD converter

입 력				출 력	
a1	a0	b1	b0	캐리(C)	중간합(W)
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	-2
0	0	1	1	1	-1
0	1	0	0	0	1
0	1	0	1	1	-2
0	1	1	0	1	-1
0	1	1	1	1	0
1	0	0	0	1	-2
1	0	0	1	1	-1
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	1	-1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	2



(a) MSB 부분 (j=홀수)



(b) MSB를 제외한 비트 부분 (j=홀수)

그림 14. 이진-SD 변환기의 논리도
Fig. 14. Logic diagram of Binary-SD Converter

그림 15는 $a_{2i+1,j}=1, a_{2i,j}=0, b_{2i+1,j}=0, b_{2i,j}=1$ 일 때 그림 14의 (b)부분의 시뮬레이션 결과이다.

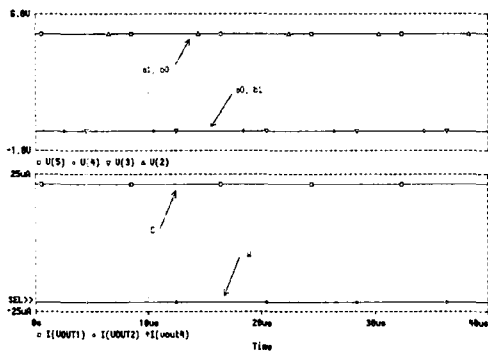


그림 15. 과도응답 특성
Fig. 15. Transient response characteristics

그림 15에서 캐리 C가 $20\mu A$ (논리치 1), 중간합 W가 $-20\mu A$ (논리치 -1)로 표 2와 일치함을 알 수 있다.

2. 부분곱 가산

승산기의 최종합 P는 앞에서 언급한 SD 가산기를 사용한 부분곱 가산에 의해서 얻어질 수 있다. 부분곱의 출력 W_{ij} 와 C_{ij} 는 식(21)과 같이 선형가산으로 구할 수 있다.

$$P_{ij} = W_{ij} + C_{i-1,j} \dots\dots\dots (21)$$

여기서 $P_{ij} \in \{-3, -2, -1, 0, 1, 2, 3\}$ 이다. 식 (21)으로부터 얻어진 P_{ij} 를 사용하여 SD수 P_j 는 식(22)와 같다.

$$P_j = (P_{n/2j} \cdot \cdot P_{ij} \cdot \cdot P_{0j}) = \sum_{i=0}^{n/2} P_{ij} 4^i \dots\dots (22)$$

최종합 P는 식 (23)과 같다.

$$P = \sum_{j=0}^{n/2-1} P_j 16^j \dots\dots\dots (23)$$

따라서 최종합은 P_j 의 다연산 가산을 행하므로서 구할 수가 있는데 이 다연산 가산은 2진 목구조 형태로 2입력 병렬 가산기를 사용하여 구성할 수 있다. 그림 16은 16×16 bit 승산의 예를 보이는데 9×4 개의 부분곱으로 구성되어 있다

그림 16에서 9×4 개의 부분곱으로 구성되어 있다. 일반적인 이진 목구조를 기본으로 하는 n-연산 가산기는 다음과 같이 L-단 SDFA로 실행된다.

$$L = \lfloor \log_2 n \rfloor \dots\dots\dots (24)$$

여기서 $\lfloor X \rfloor$ 는 $\lfloor X \rfloor > X$ 를 만족하는 가장 작은 정수이다. 첫번째 단 병렬 SD 가산기의 입력에서 선형합 Z_{ij} 는 식(25)과 같다.

$$Z_{ij} = P_{i,2j} + P_{i,2j+1} \dots\dots\dots (25)$$

여기서 $P_{i,2j} (\in \{-1, 0, 1, 2, 3\})$ 와 $P_{i,2j+1} (\in \{-3,$

-2, -1, 0, 1)는 식 (21)의 부분곱 결과의 선형합 이고 Z_{ij} 는 -4에서 4로 제한되어진다. 이 방법을 이용하면 SD 4진수의 표현 범위는 -6에서 6이므로 2의 보수 연산에 의한 증가 bit d_i 와 e_i 를 더하는 것이 가능하게 되므로 n-bit 승산에 있어서 가산기 단 L의 수는 다음과 같이 줄어들 수 있다.

$$L = \lceil \log_2(n/4) \rceil = \lceil \log_2 n - 2 \rceil \dots\dots\dots (26)$$

그러므로 32×32bit 승산의 경우는 3단의 가산목 으로 결과를 구할 수 있다.

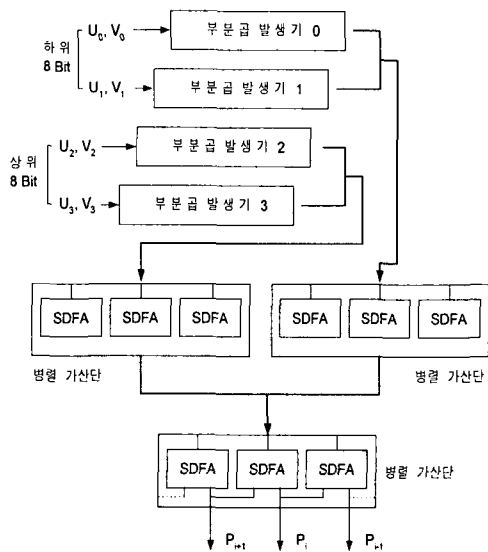


그림 16. 8×4 승산의 예
Fig. 16. Example of 8×4 multiplication

VI. 결 론

본 논문에서는 다치 가산기 및 승산기를 제안하고 전류 및 전압 모드 CMOS 회로를 이용하여 이를 구현하였으며 설계된 회로에 대한 검증은 SPICE 및 부분적으로 FPGA를 이용하여 하드웨어 에뮬레이션하였다. 일반 이진 시스템과 연계를 위하여 연산결과인 4치 SD수를 디코오더를 이용하여 이진 전압모드로 간단히 변화할 수 있게 하였다.

또한 SD수 전류모드 CMOS 회로를 사용하여 고속의 병렬가산이 가능하게 하였고, 확장 Booth 알고리즘을 다치 승산기에 적용하여 3bit까지 시프트가 가능하게 하여 부분적의 수를 줄여 고밀도를 가능케하였다.

앞으로의 과제는 최적화 및 저소비 전력화를 하여야 할 것이며, 설계된 가산기 및 승산기를 기반으로 하는 디지털처리, 화상처리 등 응용 프로세스에의 적용에 대한 연구가 진행되어야 할 것이다.

참고문헌

- [1] T.Higuchi and M. Kameyama, *Multiple-Valued Digital Processing System*, Shokodo Co. Ltd., 1989.
- [2] M. Kameyama and T. Higuchi, "Design of Radix-4 Signed-Digit Arithmetic," *IRE Trans. Elect. Comput.*, EC-10, pp. 389-400. Sept. 1961.
- [3] K. C. Smith, "Multiple-Valued logic: a tutorial and appreciation," *IEEE Computer*, vol. 37, no. 4, pp. 17-27, 1988.
- [4] T. Hanyu, M. Kameyama and T. Higuchi, "Beyond-Binary Circuits for Signal Processing," *ISSCC Dig. Tech. Papers*, pp. 134-135, 1993.
- [5] K. Shimabukuro, "Design of a Multi-Valued VLSI Processor Digital Control," *IEEE Trans. Computers*, pp. 322-328, Jan., 1992.
- [6] A. K. Jain, R. J. Bolton, "CMOS Multiple-Valued logic design - part I: Circuit implementation," *IEEE Trans. on Circuits and System-I*, vol. 40, pp. 503-514, Aug. 1993.
- [7] T. Hanyu, M. Arakaki and M. Kameyama, "Quaternary Universal-Literal CAM for Cellular Logic Image Processing," *ISMVL*, vol. 26, no.1, pp. 224-229, 1996.
- [8] L. P. Rubinfeld, "A proof of the modified Booth's algorithm for multiplication," *IEEE Trans. Comput.*, vol. C-24, pp. 1014-1015, Oct. 1975.



박진우(Jin-Woo Park)
1996년 2월 동의대학교 전자공학과 졸업(공학사)
1996년 3월~현재 동의대학교 대학원 전자공학과 석사과정
주관심 분야: 다치 논리, 자동 제어 및 마이크로프로세스 응용



양대영(Dae-Young Yang)
1997년 2월 동의대학교 전자공학과 졸업(공학사)
1997년 3월~현재 동의대학교 대학원 전자공학과 석사과정
주관심 분야: 다치 논리이론 및 VLSI 설계



송홍복(Hong-Bok Song)
1983년 2월 광운대학교 전자통신 공학과 졸업(공학사)
1985년 2월 인하대학교 대학원 전자공학과 졸업(공학석사)
1985년 3월~1990년 2월 동의공업전문대학교 전자통신과(조교수)
1989년~1990년 일본 九州공대 정보공학부 전자정보공학부 객원 연구원
1990년 8월 동아대학교 대학원 전자공학과 졸업(공학박사)
1994년~1995년 일본 宮崎대학교 공학부 전자공학과(POST-DOC)
1991년 3월~현재 동의대학교 전자공학과 부교수
주관심분야: 다치 논리 이론 및 다치 논리 시스템 설계, VLSI 설계, 마이크로 프로세스 응용 등임.