다치 논리를 이용한 연산기 구현

양대영*, 김희진*, 박진우*, 송홍복*
* 동의대학교 전자공학과

Implementation of Arithmetic Processor Using Multi-Valued Logic

Dae-Young Yang*, Heui Jin Kim*, Jin-Woo Park*, Hong-Bok Song*
* Department of Electronic Engineering, Dong-eui University
E-mail : dyyang@hyomin.dongeui.ac.kr

Abstract

This paper presents CMOS full adder design method based on carry-propagation-free addition trees and a circuit technique, so called multiple - valued current-mode(MVCM) circuits. The carry - propagation-free addition method uses a redundant digit sets called redundant positive-digit number representations. The carry - propagation - free addition is by three steps, and the adder can be designed directly and efficiently from the algorithm using MVCM circuit. Also Multiplier can be designed by these adder. We demonstrate the effectiveness of the proposed method through simulation(SPICE).

Ⅰ. 서 론

최근 VLSI 기술의 급속한 발전으로 인해 많은 양의 데이터를 실시간으로 처리할 수 있게 되었다. 초고속 신호 처리 및 디지털 제어와 같은 시스템을 VLSI화하기 위해서는 다량의 정보를 신속하게 처리해야하므로 연산의 속도가 중요하게 고려되어져야한다. 그러나, 기존의 2치 논리에서는 캐리 전파에 의해서 연산의 속도가 제한되어지므로 이 문제를 해결하기 위해서는 새로운 고속 연산속도를 가지는 연산기가 요구되어진다.

이러한 요구에 대해서 다치논리를 기본으로 하는 응용분야가 주목되고 있다.[1]-[4] 다치 논리의 일반적인 특징으로는 입출력의 핀수를 감소시킴으로써 집적밀도를 증가시킬 수 있고, 배선의 복잡성을 감소시켜 회로를 정형화 함으로써 연산 속도를 높일 수 있다. 또한 직렬 접속의 단수 및 연산 반복횟수를 감소시킴으로써 연산 속도를 고속화 할 수 있다. 이러한 다치 논리의 특징들을 회로 설계에 이용한다면 종래의 2진 시스템을 간략히 실현할 수 있고 또한 효율적인 처리도 가능하다.

다치 논리 회로에서 Signed-digit (SD) 와 Redundant -Positive-digit (PD) 수 표현을 이용하여 병렬 가산을 실현한다면, 캐리 전파 없이 병렬 가산을 행할 수 있으므로 연산을 고속화 할 수 있다.[5]

본 논문에서는 캐리 전파 없이 병렬 가산이 가능하고 게이트 지연을 줄여 연산의 속도를 고속화 할 수 있는 PD 수 표현의 다 입력 병렬 가산기와 이를 이용한 승산기를 제안한다. 그리고 다 입력 병렬 가산기의 유효함은 시뮬레이션(SPICE)을 이용하여 확인할 것이다.

Ⅱ. PD수 표현을 이용한 가산기 구현

1. 2진 2입력 가산기의 구현

2진 2입력 가산에는 캐리 전파를 없애기 위해서 PD(2,3) 수 표시를 사용하였다. PD(2,3)은 {0,1,2,3}의 디지트 집합으로, PD(2,2)는{0,1,2}의 디지트 집합으로 표현된다. 먼저, PD(2,3)을 이용한 2진 2입력 가산에서 입력은 $X=(x_{n-1}\cdots x_i\cdots x_0)_{PD(2,3)}$ 와 $Y=(y_{n-1}\cdots y_i\cdots y_0)_{PD(2,3)}$ 이고, $x_i, y_i \in \{0,1,2,3\}$이다. X와 Y를 이용한 병렬가산은 다음의 과정을 통해서 이루어진다.

$$z_i = x_i + y_i \tag{1}$$

$$4c_i^{(2)} + 2c_i^{(1)} + w_i = z_i \tag{2}$$

$$s_i = w_i + c_{i-1}^{(1)} + c_{i-2}^{(2)} \tag{3}$$

여기서 선형합 $z_i \in \{0,1,\cdot\cdot,6\}$, 캐리와 중간합 $c_i^{(2)}, c_i^{(1)}, w_i \in \{0,1\}$, 최종합 $s_i \in \{0,1,2,3\}$ 이다. 위의 식을 이용해서 2진 2입력 PD(2,3) 가산기를 구성하면 다음과 같다.

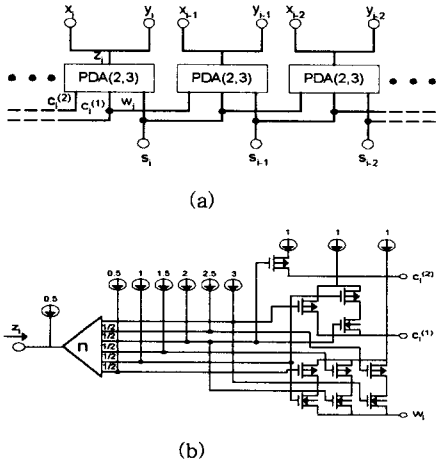


그림 3. 병렬 PD(2,3) 가산기
(a) 가산기, (b) PDA(2,3)소자
Fig 3. Parallel PD(2,3) adder
(a) adder, (b)PDA(2,3) cel

1그림 3 의 과도응답 특성은 다음과 같다.

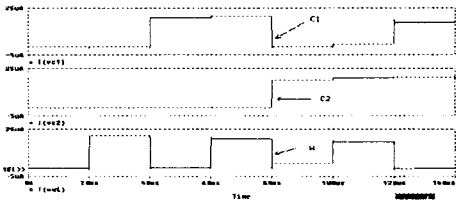


그림 4. 그림 3의 과도응답 특성
Fig 4. Transient respone characteristics of Fig 3

여기서, 입력을 논리치 {0,1,2,3,4,5,6}으로 했을 때 중간합 w_i는 {0,1,0,1,0,1,0}, 캐리 $c_i^{(2)}$은{0,0,0,0,1,1,1}, 캐리 $c_i^{(1)}$은 {0,0,1,1,0,0,1}가 나온다. 이는 식 (1)-(3)의 결과와 일치함을 알 수있다. 선형 합 z_i 와 최종 합 s_i 는 가산기 설계 시 별도의 능동 소자 없이 결선만으로도 얻어진다. 이 특징을 이용한다면 가산기의 구조를 단순화 할 수 있으므로 연산을 고속으로 수행할 수 있다.

2. 2진 3입력 가산기의 구현

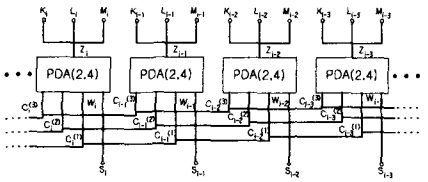
2진 3입력 가산기의 구현에서는 캐리 전파를 없애기 위해서 PD(2,4) 수 표시를 사용하였다. PD(2,4)은 {0,1,2,3,4}의 디지트 집합으로 표현된다. PD(2,4)을 이용한 2진 3입력 가산에서 입력은 $K=(k_{n-1}\cdots k_i\cdots k_0)_{PD(2,4)}$, $L=(l_{n-1}\cdots l_i\cdots l_0)_{PD(2,4)}$, $M=(m_{n-1}\cdots m_i\cdots m_0)$ 이고, $k_i, l_i, m_i \in \{0,1,2,3,4\}$ 이다. K , L , Y를 이용한 병렬가산은 다음의 과정을 통해서 이루어진다.

$$z_i = k_i + l_i + m_i \tag{4}$$

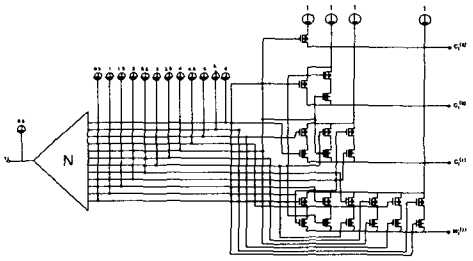
$$8c_i^{(3)} + 4c_i^{(2)} + 2c_i^{(1)} + w_i = z_i \tag{5}$$

$$s_i = w_i + c_{i-1}^{(1)} + c_{i-2}^{(2)} + c_{i-3}^{(3)} \tag{6}$$

여기서 선형합 $z_i \in \{0,1,\cdot\cdot,12\}$, 캐리와 중간합 $c_i^{(3)}, c_i^{(2)}, c_i^{(1)}, w_i \in \{0,1\}$, 최종합 $s_i \in \{0,1,2,3,4\}$ 이다. 위의 식을 이용해서 2진 4입력 PD(24) 가산기를 구성하면 다음과 같다.



(a)



(b)

그림 5. 병렬 PD(2,4) 가산기
(a) 가산기, (b) PDA(2,4)소자
Fig 3. Parallel PD(2,4) adder
(a) adder, (b)PDA(2,4) cell

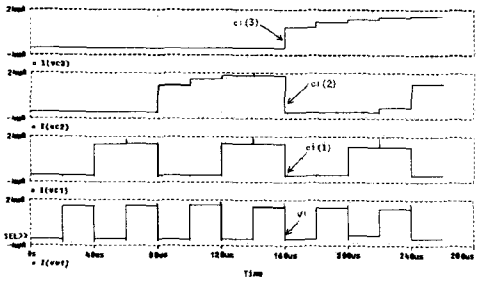


그림6. 그림 5의 과도응답 특성
Fig 6. Transient respone characteristics of Fig 5

III. PD수 표현을 이용한 승산기 구현

M개의 다 입력을 갖는 가산기를 이용하면 고속의 승산 기를 구현할 수 있다. M개의 다 입력을 갖는 가산기는 PD수 표현을 이용하여 구현되므로, 이들 가산기를 이용한 승산기도 PD수 표현을 이용하여 구현할 수 있다. 승산기의 입력은 $X=(x_{n-1}\cdots x_i\cdots x_0)_2$와 $Y=(y_{n-1}\cdots y_i\cdots y)_2$이고 여기서, $x_i\in\{0,1\}$ $(i=0,1,\cdots,n-1)$ 이고 $y_j\in\{0,1\}$ $(j=0,,\cdots n-1)$ 이다. 그 출력은 $P=(p_{2n-1}\cdots p_i\cdots p_0)_2$ 과 같은 2n-비트의 부호 없는 정수들로 표현되어진다.

여기서, $p_i\in\{0,1\}$ $(0,1,\cdots,2n-1)$ 이다. 본 논문에서는 2입력 PD(2,3) 가산기를 이용한 12×12 승산기와 3입력 PD(2,4) 가산기를 이용한 36×36 승산 기를 구현하였다.

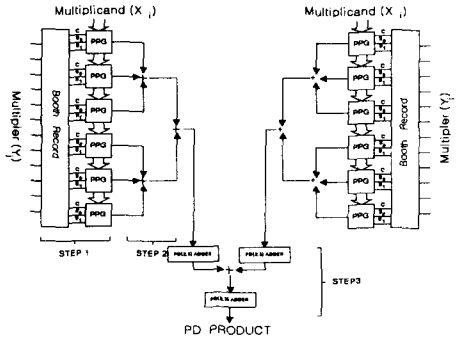


그림 7. 12×12 승산기
그림 7. 12×12 multiplier

먼저 12×12 승산기에서 입력이 $X=(x_{11}\cdots x_1\ x_0)$, $Y=(y_{11}\cdots y_1\ y_0)$일때, 승산 알고리즘은 다음과 같다.

단계 1: 여기서는 12×12의 부분 곱, p_{ij}가 만들어진다.

$$p_{i,j} = x_i \times y_j \qquad (7)$$

여기서, $p_{ij}\in\{0,1\}$ $(i=0,1,\cdots,11,\ j=0,1,\cdots,11)$
따라서, 그림6 에서 구해질 수 있는 부분 곱 P_i는 $P_i = (\ p_{i,11}\cdots p_{i,1}\ p_{i,0}\ )$ $(i=0,1,\cdots,11)$ 이다. 또한, 승수 y_j는 Booth Recorder에 의하여 보수 신호 C(j), 시프트 없음 신호 $S_0(j)$, 1-시프트 신호 $S_1(j)$로 리코딩 된다.

단계 2: 여기서는 3개의 부분 곱의 디지트를 선형적으로 디하여 다음 단의 PD(2,3) 가산기의 입력을 만든다. 즉, 3개의 부분 곱의 디지트의 선형 합에 의해서 만들어지는 부분 곱을 $p^{(0)}_{i,j'}$라고 할 때:

$$p^{(0)}_{i,j'} = p_{i,3j'} + p_{i-1,3j'+1} + p_{i-2,3j'+2} \qquad (8)$$

여기서, $p^{(0)}_{i,j'}\in\{0,1,2,3\}$, $(i=0,1,\cdots,13,\ j'=0,1,2,3)$, $j'=(j/3)-1$, $p^{(0)}_{12,j'}=p^{(0)}_{13,j'}=0$ $(i=0,1\cdots11)$. 따라서,

$p^{(0)}_{i,j'} \in \{0,1,2,3\}$이므로 이들 부분 곱 $P^{(0)}_{j'} = (p^{(0)}_{13,j'} \cdots p^{(0)}_{1,j'} p^{(0)}_{0,j'})(j'=0,1,2,3)$은 PD(2,3) 가산기에 의하여 가산이 가능하다.

단계 3: PD(2,3) 가산기를 이용하여 부분 곱을 더하고 곱 $P^{(L)}_0$을 구한다. L은 $\log_3 j'$이다. 마지막 PD(2,3)가산기에 의해서 생성된 곱 $P^{(L)}_0$에서 carry lookahead adder와 같은 2진 가산기에 의해서 최종 합(2진수)를 구할 수 있다.

위의 승산 알고리즘을 이용한 승산 기 구현에서 가산기 층의 수는 $K=[\log_2(n/3)]$이다. 여기서, n은 비트 수이고, [a]는 [a]≥a인 가장 적은 정수를 나타낸다. 또한, 위의 승산 알고리즘 중에서 단계 2의 과정을 다음과 같이 고치고,

$$p^{(0)}_{i,j'} = p_{i,4j'} + p_{i-1,4j'+1} + p_{i-2,4j'+2} + p_{i-3,4j'+3} \qquad (9)$$

여기서, $p^{(0)}_{i,j'} \in \{0,1,2,3,4\}, (i=0,1,\cdots,14,\ j'=0,1,2)$, $j'=(j/4)-1$, $p^{(0)}_{12,j'}=p^{(0)}_{13,j'}= p^{(0)}_{14,j'}= 0$, $(i,j=0,1\cdots11)$ 가산기 층의 수를 $K=[\log_3(n/4)]$로 하면, 다음과 같은 3입력 PD(2,4) 가산기를 이용한 36×36-비트 승산기를 구현할 수 있다.

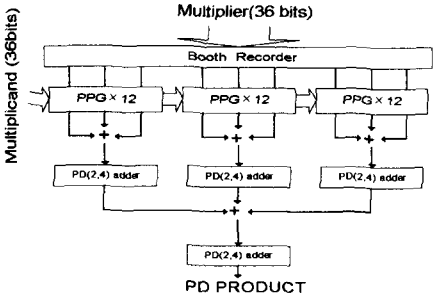


그림 8, 32×32 승산기
그림 8. 32×32 multiplier

위와 같은 PD 수 표현을 이용한 승산 알고리즘에서, 단계 2와 단계 3에서의 선형적 합은 별도의 능동소자 없이 병렬 구조를 통하여 얻어지므로, 승수와 피승수의 곱을 구하는 전체 연산 시간은 승수나 피승수의 연산자 길이 n과는 무관하고 단지 가산기 층의 수에 비례함을 알 수 있다. 따라서, 연산의 고속화가 가능하다.

Ⅳ. 결론

본 논문에서는 캐리 전파 없이 병렬 가산이 가능하고 게이트 지연을 줄여 연산의 속도를 고속화할 수 있는 PD 수 표현의 가산기와 이를 이용한 승산 기를 구성하였다. PD 수 표현의 가산기 구성의 예로써 PD(2,3)가산기와 PD(2,4)가산기를 보였다. 또한, 승산기 구성의 예로써는 PD(2,3) 가산기를 이용한 12×12 MUX 와 PD(2,4) 가산기를 이용한 32×32 MUX의 구조를 설명하였다. 각 가산기와 승산기의 유효함은 시뮬레이션(SPICE)를 이용하여 확인하였다. 이상에서, PD 수 표현을 이용하여 연산 기를 구성한다면 연산에서 구해지는 선형 합은 별도의 능동 소자 없이 결선만으로 얻어지므로 연산기의 구조를 정형화 할 수 있고, 또한 전체 연산 시간은 연산자의 길이와는 상관없이 연산 단의 수에 비례하므로 이를 이용한다면 고속의 연산자를 구성할 수 있다. 향후의 과제로는 본 논문에서 설명한 승산 알고리즘을 이용하여 전체 12×12 승산기를 실제 설계하고 제작하여 승산기의 동작을 검증할 것이다.

참고 문헌

[1] A. K. Jain, R. J. Bolton, "CMOS Multiple-Valued logic design - part I: Circuit implementation," IEEE Trans. on Circuits and Syatem-I, vol.40, pp, 503-514, Aug.1993.

[2] K. C. Smith, "Multiple-Valued logic : a tutorial and appreciation," IEEE Computer, vol. 37, no. 4, pp.17-27, 1988.

[3] M. Kameyama, T.Hanyu, and T. Higuchi, "Design and Implementation of quaternary NMOS integrated circuits for piplined image processing," IEEE J. Solid-state Circuits, vol. SC-22, no. 1, pp. 20-27, 1987.

[4] T. Hanyu, M. Arakaki and M. Kameyama, "Quanternary Universal-Literal CAM for Celluar logic Image Processing," ISMVL, vol. 26, no, 1, pp. 224-229, 1996.

[5] A. Avizienis, "Signed-digit number representation for fast parallel arithmetic," IRE Trans. Elect Computer., vol. EC-10, pp. 389-400,Sept. 1961