

2의 보수 직병렬 승산을 위한 논리구조

An Architecture for Two's Complement Serial-Parallel Multiplication

牟相晩 · 尹龍鎬

〈要 約〉

직병렬 승산기는 피승수와 승수중 어느 하나가 병렬로 입력되고 또다른 수는 직렬로 입력되는 구조를 가지며, 디지털 신호처리, 온라인 응용, 특수 목적용 계산 시스템 등에서 많이 이용되고 있다. 본 논문에서는 2의 보수를 위한 직병렬 승산기의 논리구조를 제안한다. 제안한 2의 보수 직병렬 승산기는 효과적인 2의 보수 직병렬 승산 알고리즘에 의해서 모든 데이터 신호가 국부적 연결만으로 구성되며, 간단하고 모듈화된 하드웨어의 구성으로 쉽게 설계할 수 있다. 이 승산기는 무부호 승산과 마찬가지로 $2n+1$ 사이클만을 필요로 하고, 각 사이클 시간은 무부호 직병렬 승산에 비해서 2의 보수 승산을 위한 XOR 게이트의 지연시간이 추가된 것뿐이다. 또한, 제안한 2의 보수 직병렬 승산기는 VLSI 구현에 매우 적합한 구조를 지닌다.

I. 서 론

두 이진수를 산술적으로 곱하는 연산을 수행하는 승산기는 피승수와 승수의 입력형태에 따라 직렬 승산기(serial multiplier), 병렬 승산기(parallel multiplier), 직병렬 승산기(serial-parallel multiplier)로 분류된다. 직렬 승산기에서는 피승수와 승수 모두가 직렬로 입력되고, 병렬 승산기에서는 두 수가 모두 병렬로 입력되며, 직병렬 승산기에서는 피승수와 승수중 어느 하나가 병렬로 입력되고 또다른 수는 직렬로 입력된다. 직렬로 입력되는 수는 일반적으로 최하위 비트(LSB)에서부터 차례대로 입력된다. 이 세 종류의 승산기들은 응용분야가 서로 다르며, 본 논문에서 다루는 직병렬 승산기는 디지털 신호처리, 온라인(on-line) 응용, 특수 목적용 계산 시스템 등에서 많이 이용되고 있다.^[1, 2, 7, 8, 10]

승산과정에서 생성되는 부분곱에 포함된 이진

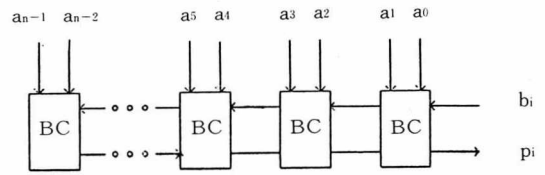
수 1의 개수를 세는 계수기(1's counter)를 이용하여 승산을 수행하는 직병렬 승산기^[3]가 제안된 후, 이 승산기에 2의 보수 연산을 수행할 수 있는 기능을 추가한 직병렬 승산기^[4]가 개발되었다. 직병렬 승산기중에서 가장 많이 사용되는 구조는 여러 문헌^[5,6,7]에서 소개되어 잘 알려져 있는 CSAS(carry-save add-shift) 직병렬 승산기이다. Danielsson^[7]은 convolver를 설계하는 데 직병렬 승산기를 사용하였고, Smith^[8]는 내적(inner product) 계산에 직병렬 승산기를 이용하는 방법을 제안하였다. Wu^[9]는 시스톨릭(systolic)구조를 갖는 직병렬 승산기를 수정형 Booth 알고리즘에 기초하여 설계하였고, 특수 목적의 프로세서를 위한 직병렬 승산기^[10]가 설계되기도 하였다.

기존의 CSAS 직병렬 승산기에서는 직렬로 입력되는 각각의 비트 데이터가 긴 연결선을 통하여 모든 연산 셀로 분배된다. 이와 같은 연결선의 전역적 분배(global distribution)구조는 VLSI 구현에 부적합하며, 비용이 증가하고 클럭 주파수를 감소시키는 부작용을 일으킨다.^[11] 따라서 CSAS 직병렬 승산기에서 나타나고 있는 연결선의 전역적 분배는 제거되는 것이 바람직하다.

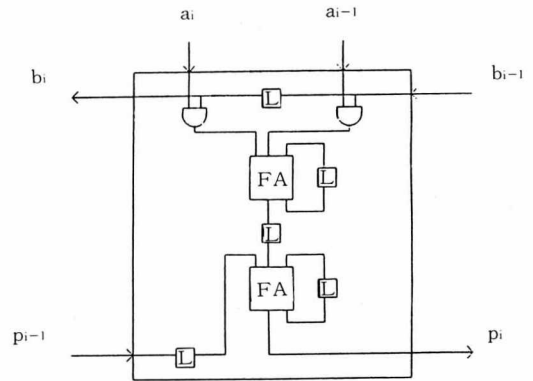
(그림 1)은 CSAS 직병렬 승산기와는 달리 각각의 비트 데이터가 국부적 연결만으로 구성된 직병렬 승산기^[12]를 보여준다. 이 승산기는 연결선의 전역적 분배구조가 제거되어 VLSI 구현이 용이하고 연산시간 또한 CSAS 직병렬 승산기에 비해 거의 차이가 없지만, 기본적으로 무보호 승산을 위한 구조이다.

본 논문에서는 모든 데이터 신호가 국부적 연결구조를 갖는 2의 보수를 위한 직병렬 승산기를 제안한다. 제안한 직병렬 승산기는 세 종류의 기본 셀들로 구성되며 구조와 연산과정이 간단하여 2의 보수 승산을 효율적으로 수행할 수 있다.

본 논문의 제 II 절에서는 2의 보수에 대한 개요와 효율적인 직병렬 승산기의 구성을 위한 2의 보수 직병렬 승산 알고리즘을 기술하고, 제 III 절에서는 II 절의 알고리즘을 기초로 하여 설



(a) 블록 다이어그램



(b) BC 셀

(그림 1) 무부호 직병렬 승산기

계한 국부적 연결구조를 갖는 2의 보수 직병렬 승산기를 기술한다. 마지막 IV 절에서는 본 논문의 결론을 기술한다.

II. 2의 보수와 직병렬 승산

2의 보수로 표현된 두 수를 A, B라 하고 각각 n비트로 구성되어 있다고 하면 A와 B는

$$A = [a_{n-1} a_{n-2} \dots a_0]^{(2)} \cdot$$

$$= \sum_{i=0}^{n-2} a_i \cdot 2^i - a_{n-1} \cdot 2^{n-1} \dots \dots \dots (1)$$

$$B = [b_{n-1} b_{n-2} \dots b_0]^{(2)}$$

$$= \sum_{j=0}^{n-2} b_j \cdot 2^j - b_{n-1} \cdot 2^{n-1} \dots \dots \dots (2)$$

로 표시할 수 있다. 상기의 식들에서 대괄호(brackets)와 아래 첨자(2)는 대괄호 안에 있는 수가 이진수로 표현된 것임을 나타낸다. 식 (1)과 (2)로 표현된 두 수의 승산결과를 P라 하면

P는 2n 비트의 데이터인

$P = [P_{2n-1} P_{2n-2} \dots P_0]_{(2)} \dots\dots\dots (3)$
 로 표시되고, 다음과 같은 연산으로 구할 수 있다.

$$\begin{aligned}
 P &= A \cdot B \\
 &= A \cdot \left(\sum_{j=0}^{n-2} b_j \cdot 2^j + b_{n-1} \cdot 2^{n-1} \right) \\
 &= A \cdot \sum_{j=0}^{n-2} b_j \cdot 2^j + A \cdot b_{n-1} \cdot 2^{n-1} \dots\dots\dots (4)
 \end{aligned}$$

상기 식(4)의 우변에서 두번째 항은 감산이므로 2의 보수를 가산해주는 방법으로 변환해야 한다. $-A \cdot b_{n-1} \cdot 2^{n-1}$ 은 $(-A) \cdot b_{n-1} \cdot 2^{n-1}$ 로 표현할 수 있고, $-A$ 는 A의 2의 보수이므로 $\bar{A}+1$ 로 나타낼 수 있다. 따라서 식(4)는 다음과 같이 다시 쓰여질 수 있다.

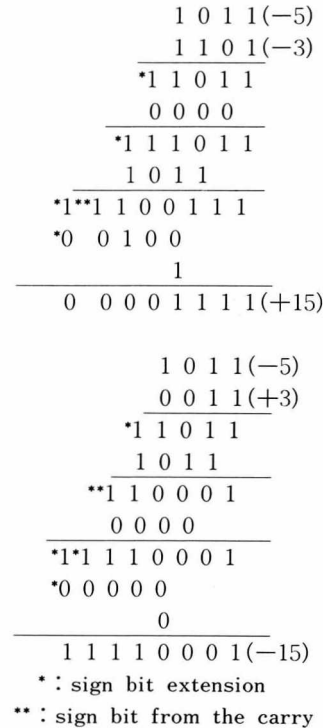
$$\begin{aligned}
 P &= A \cdot \sum_{j=0}^{n-2} b_j \cdot 2^j + (\bar{A}+1) \cdot b_{n-1} \cdot 2^{n-1} \\
 &= A \cdot \sum_{j=0}^{n-2} b_j \cdot 2^j + \bar{A} \cdot b_{n-1} \cdot 2^{n-1} + b_{n-1} \cdot 2^{n-1} \dots\dots\dots (5)
 \end{aligned}$$

식(5)에 내재되어 있는 승산방법은 승수(B)가 직렬로 입력되는 경우에 쉽게 적용될 수 있는 특성을 지니고 있다. 즉 승수를 한 순간에 한 비트씩 차례대로 입력하여 피승수(A)에 곱하고, 승수의 한 비트가 피승수와 곱하는 연산은 동시에 병렬로 수행하도록 할 수 있다. 이때 피승수의 모든 비트는 병렬로 입력되어 있어야 한다. 이와 같은 과정은 승수는 직렬로 입력되고 피승수는 병렬로 입력되는 직병렬 승산기의 연산방식에 부합된다.

식(5)의 우변에서 첫번째 항은 승수가 최하위 비트(LSB)에서부터 입력되어 n-2번째 비트까지 입력되는 과정에서 각 입력 비트가 병렬로 입력되어 있는 피승수와 승산하는 과정을 나타내고 있다. 두번째 항과 마지막 항은 승수의 최상위 비트(MSB)가 피승수에 곱해질 때의 연산을 나타낸 것으로서, 승수의 최상위 비트가 1이면 피승수의 2의 보수가 부분곱(partial product)에 더해지고 승수의 최상위 비트가 0이면 두 개의 항은 0이 되고 결과적으로 0이 더해지게 된

다.

4비트로 표현된 2의 보수의 단계적인 연산과정(그림 2)에 나타나 있다. 그림에서 관찰할 수 있는 바와 같이 부호 비트의 확장은 각 단계에서의 부분곱 부호 비트의 직접 확장이나, 또는 부분곱 가산과정에서 생성된 올림수의 결과로서 발생한다.^[6] 또한 마지막 가산단계에서는 승수의 최상위 비트(MBS)가 1이면 2의 보수가 더해지고, 그렇지 않으면 0이 더해진다. 이때 부분곱의 부호 비트 및 승수의 최상위 비트(b_{n-1})와의 연산에 의해 더해지는 수의 부호 비트가 각각 직접 확장된다. 구해진 승산결과에서 최상위 비트를 초과하여 발생한 올림수는 무시되고 2n 비트가 구하고자 하는 최종 결과값이 된다. 최종의 승산결과 역시 입력된 두 수의 표현방법과 같이 2의 보수로 표현된 값이다.

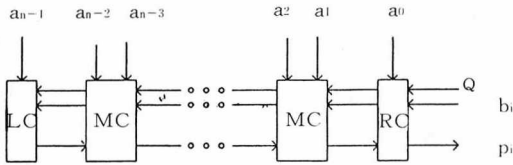


(그림 2) 2의 보수 직병렬 승산의 예

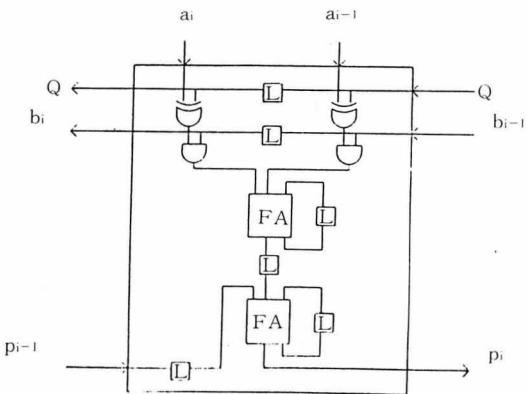
Ⅲ. 직병렬 승산기의 구조

앞절에서 설명한 2의 보수 직병렬 승산방법을

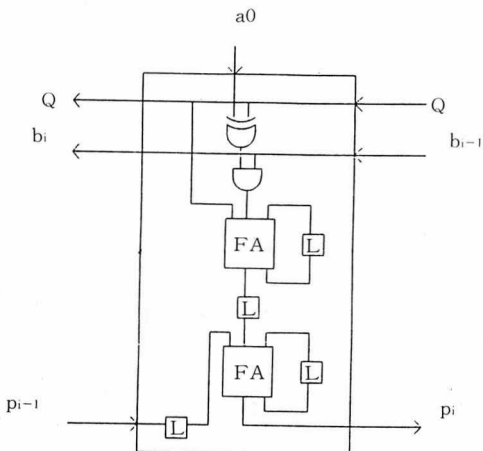
기초로 하여 설계한 직병렬 승산기가 (그림 3)에 도시되어 있다. (그림 3)에서 볼 수 있는 바와 같이 설계한 직병렬 승산기는 국부적 연결만으로 구성되는 구조를 가지며, 2의 보수에 대한 직병렬 승산을 효율적으로 수행할 수 있다. 이 승산기는 세 종류의 셀로 구성되는데, 각각은 MC(Middle Cell), RC(Rightmost Cell), LC(Leftmost Cell)이다.



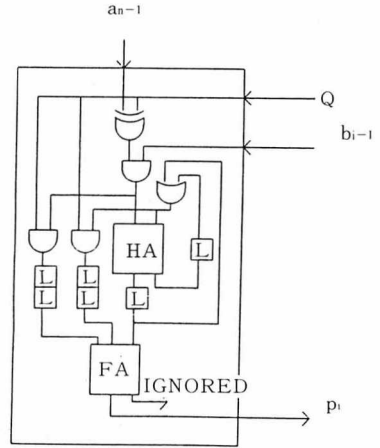
(a) 블럭 다이어그램



(b) MC 셀



(c) RC 셀



(d) LC 셀

(그림 3) 2의 보수 직병렬 승산기

승수가 직렬로 입력되는 과정에서 승수의 최상위 비트 b_{n-1} 이 입력되면 $Q=b_{n-1}$ 이 되고, 다른 비트가 입력될 때는 $Q=0$ 이 된다. 각 셀에 있는 XOR 게이트(gate)는 $Q=0$ 일 때는 피승수(A)의 모든 비트를 변화없이 그대로 출력하고, $Q=1$ 일 때는 피승수의 각 비트를 반전(inverse)시켜 출력한다. 이것은 승수의 최상위 비트 b_{n-1} 이 입력될 때 그 값이 1이면 피승수의 2의 보수를 부분곱에 더해주기 위한 구조이다. 각 셀에 있는 AND 게이트는 피승수의 각 비트와 입력되는 승수 비트를 곱하는 기능을 수행하며, FA(Full Adder)는 전가산기로서 AND연산의 결과를 부분곱에 더해주는 역할을 수행한다.

RC 셀에서는 Q가 FA 셀에 입력되는데, 이것은 (그림 2)에 나타난 바와 같이 승수의 최상위 비트 b_{n-1} 이 입력될 때 그 값이 1이면($Q=b_{n-1}=1$) 피승수의 2의 보수를 부분곱에 더해 주기 위한 것으로서, XOR 게이트에 의해 반전된 피승수의 최하위 비트에 $1(Q=1)$ 을 더해주는 역할을 수행한다. b_{n-1} 의 값이 0이면 $0(Q=b_{n-1}=0)$ 이 FA 셀에 입력된다.

LC 셀에서의 연산은 다른 셀과는 다소 다른 특별한 동작을 수행한다. HA(Half Adder) 셀에 입력되는 OR 게이트의 출력은 부호 확장을 수행하기 위한 것으로서, 부분곱 부호 비트의

직접 확장이나 부분곱 가산과정에서 생성된 올림수에 의한 확장을 수행한다. 또한, 승수의 최상위 비트 b_{n-1} 이 입력될 때는 완전한 부호를 갖는 $2n$ 비트의 승산결과를 얻기 위하여 (그림 2)에서 볼 수 있는 바와 같이 부분곱과 더해지는 수에 대하여 추가적인 부호 확장이 필요하게 된다. 이를 수행하기 위하여 두 개의 AND 게이트 출력이 FA 셀에 입력된다. 즉 부분곱의 부호 비트가 직접 확장되고, b_{n-1} 과의 연산에 의해 더해지는 수의 부호 비트가 직접 확장된다. 두 개의 AND 게이트에 의해 확장된 각각의 부호 비트는 두 개의 연속된 래치(latch)에 의해서 한 사이클 지연되어 FA 셀에 입력됨으로써 부호의 확장이 일어나게 된다. 마지막 승산결과에서 $2n$ 비트 이상으로 발생하는 올림수는 무시된다.

제안한 2의 보수 직병렬 승산기는 효과적인 2의 보수 직병렬 승산 알고리즘에 의해서 모든 데이터 신호가 국부적 연결만으로 구성되며, 간단하고 모듈화된 하드웨어의 구성으로 쉽게 설계할 수 있다. 이 승산기는 무부호 승산과 마찬가지로 $2n+1$ 사이클만을 필요로 한다. 또한 각 사이클 시간(cycle time)은 무부호 직병렬 승산^[2]에서의 사이클 시간에 비해서 2의 보수 승산을 위한 XOR 게이트의 지연시간이 추가된 것 뿐이다.

IV. 결 론

본 논문에서는 2의 보수를 위한 새로운 직병렬 승산기를 제안하였다. 제안한 2의 보수 직병렬 승산기는 효과적인 2의 보수 직병렬 승산 알고리즘에 의해서 모든 데이터 신호가 국부적 연결만으로 구성되며, 간단하고 모듈화된 하드웨어의 구성으로 쉽게 설계할 수 있다. 이 승산기는 무부호 승산에서와 같은 $2n+1$ 사이클을 필요로 하고, 각 사이클 시간은 무부호 직병렬 승산에 비해서 단지 2의 보수 승산을 위한 XOR 게이트의 지연시간이 추가된다. 본 논문에서 설계한 직병렬 승산기는 VLSI 구현에 매우 적합한 구조를 지니고 있어 디지털 신호처리, 온라인

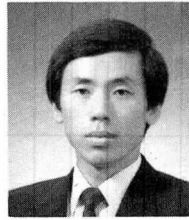
응용, 특수 목적용 계산 시스템 등에서 효과적으로 이용될 수 있다.

〈參 考 文 獻〉

1. K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*, John Wiley & Sons, 1979.
2. O. Spaniol, *Computer Arithmetic: Logic and Design*, John Wiley & Sons, 1981.
3. E. E. Swartzlander, Jr., "The Quasi-Serial Multiplier", *IEEE Trans. Computers*, Vol. C-22, No. 4, pp. 317-321 Apr. 1973.
4. T. G. McDonald and R. K. Guha, "The Two's Complement Quasi-Serial Multiplier", *IEEE Trans. Computers*, Vol. C-24, No. 12, pp. 1233-1235, Dec. 1975.
5. R. F. Lyon, *Twos Complement Pipeline Multipliers*, *IEEE Trans. Communications*, Vol. COM-12, pp. 418-425, Apr. 1976.
6. R. Gnanasekaran, "A Fast Serial-Parallel Binary Multiplier", *IEEE Trans. Computers*, Vol. C-34, No. 8, pp. 741-744, Aug. 1985.
7. P. E. Danielsson, "Serial/Parallel Convolvers", *IEEE Trans. Computers*, Vol. C-33, No. 7, pp. 652-667, Jul. 1984.
8. S. G. Smith, "Efficient Serial/Parallel Inner-Product Computation", *Electron. Lett.*, Vol. 22, No. 14, pp. 750-752, Jul. 1986.
9. I. C. Wu, "A Fast 1-D Serial-Parallel Systolic Multiplier", *IEEE Trans. Computers*, Vol. C-36, No. 10, pp. 1243-1247, Oct. 1987.
10. L. Ciminiera and A. Valenzano, "Low Cost Serial Multipliers for High-Speed Specialised Processors", *IEE Proc.*, Vol. 135, Pt. E, No. 5, pp. 259-265, Sep.

1988.

11. S. Y. Kung, "On Supercomputing with Systolic/Wavefront Array Processors", *Proc. IEEE*, Vol. 72, No. 7, pp. 867-884, Jul. 1984.
12. D. Ait-Boudaoud, M. K. Ibrahim, and B. R. Hayes-Gill, "Novel Cell Architecture for Bit Level Systolic Arrays Multiplication", *IEE Proc. -E*, Vol. 138, No. 1, pp. 21-26, Jan. 1991.



牟相晩 (Moh, Sang-Man)

1991.2 : 연세대학교 대학원
전산학과 졸업(석사)
1991.2-현재 : 한국전자통신연구소
컴퓨터구조연구실 연구원



尹龍鎬 (Yoon, Yong-Ho)

1970-1975 : 한양대학교 전자공학과
1978-1981 : 연세대학교 대학원
1985-1989 : 한양대학교 대학원
박사과정 수료
1975-1978 : 한국과학기술연구소
컴퓨터국산화연구실 연구원

1978-1991. 현재 : 한국전자통신연구소
컴퓨터구조연구실장, 책임연구원