

Leading 0/1 검출 기능을 부가한 곱셈기

김영수, 차영호, 조경연, 최혁환
부경대학교 전자.컴퓨터,정보통신공학부
전화 : (051) 620-6460 / 팩스 : (051) 620-6460

A Multiplier with Leading 0/1 Detector

Young Su Kim, Young Ho Cha, Gyeong Yeon Cho and Hyek Hwan Choi
Dept. Electronics, Computer and Telematics Engineering Pukyong National University
E-mail : hl5unmm@unitel.co.kr

Abstract

This paper describes the design of multiplier that receives two N-bit number and produces an N-bit product, with leading 0/1 detector logic for an overflow prediction. A leading 0/1 detector for two's input predict a scope of output. The part of partial products sum of N most-significant bits is exchanged for an overflow prediction. Therefore this multiplier requires less gates for the implementation about 45% than general multipliers.

I. 서론

병렬 곱셈기의 면적과 속도를 줄이기 위한 방안으로, 승수를 2 비트로 리코딩하여 부분 곱 행의 수를 감소시키는 Booth 알고리즘[1]과 이를 개선하여 승수를 3 비트 또는 그 이상으로 리코딩하는 수정형 Booth 알고리즘이 널리 이용되고 있다[2][3]. 부분 곱을 작은 면적으로 빠르게 계산하기 위해 가산기의 효율적인 배치하는 방법에 대한 연구가 최근 곱셈기에 대한 주된 연구분야이다[4].

또한 DSP와 같은 응용 분야에서는 상위 N 비트만을 취하고 하위 N 비트를 잘라내고, 이때 발생하는 에러를 보정하여 면적과 전력의 효율성을 높이는 연구가 진행되고 있다[5].

일반적인 정수형 곱셈기는 N 비트의 두 입력을 받아 2N 비트를 출력한다. 그러나 범용 마이크로 프로세서를 사용하는 고급언어는 N 비트 × N 비트의 곱셈은 N 비트로 수행된다. 정수형 곱셈기를 2N 비트 출력이 생성되도록 할 경우, 상위 N 비트를 출력하는 부분은 불필요한 많은 하드웨어 면적을 차지할 뿐 아니라, 많은 전력을 손실을 초래한다. 따라서 이 상위 N 비트의 연산부를 overflow만 검사하는 간단한 회로로 대체하면, 면적과 전력의 효율을 높일 수 있을 것으로 기대된다.

2의 보수로 표현된 정수에서 부호 비트와 같은 값이 계속 유지되는 비트까지를 검색하면, 그 수의 범위를 결정할 수 있다. 이 방법을 곱셈기에 입력되는 두 수에 대해 적용하면, 곱셈기에서 출력되는 수의 범위를 알 수 있다. 따라서 배열형 곱셈기나 수정형 Booth 곱셈기에 이 방법을 적용하여 곱셈기의 출력의 범위를 미리 확인하여 overflow 발생을 예측하는 회로를 추가하고 상위 N 비트를 버림으로서, 면적의 효율성이 높은 곱셈기의 설계가 가능하다.

II. overflow 검출

N 비트의 입력을 받아 N 비트의 출력을 발생하는 곱셈기는 overflow를 검사해야 한다. 두 N 비트 입력을 받는 곱셈기에서, A 와 B, 그리고 그들의 곱 P는 다음과 같이 표현된다.

$$A = \sum_{i=0}^{N-1} a_i 2^i \quad (1)$$

$$B = \sum_{i=0}^{N-1} b_i 2^i \quad (2)$$

$$P = \sum_{i=0}^{2N-1} p_i 2^i \quad (3)$$

이고, 여기서 $a_i, b_i, p_i \in \{0, 1\}$ 이다.

배열형 곱셈기는 N^2 비트의 부분 곱을 생성한 후, 합을 구하게 되고, Radix-4 수정형 Booth 곱셈기는 $(N+1)^2/2$, Radix-8 수정형 Booth 곱셈기는 $(N+2)^2/3$ 비트의 부분 곱의 합을 생성하고 부가적으로 Booth encoder 와 Booth selector 가 필요하다.

부분 곱의 합 P 를 상위 N 비트와 하위 N 비트로 나누면,

$$P = \sum_{i=N}^{2N-1} p_i 2^i + \sum_{i=0}^{N-1} p_i 2^i \quad (4)$$

$$= P_H + P_L$$

가 된다. 여기서 P_H 와 P_L 의 구현면적은 동일하기 때문에, N 비트로 출력되는 곱셈기에서 P_H 를 생성하는 하드웨어는 overflow를 감지하는데 이용될 뿐이지만, 실제 구현되기 위해서는 곱셈기 면적의 절반을 차지한다. 이 P_H 구현 부분을 Leading 0/1 검출기와 결합된 overflow 발생 검사 회로로 대체하였다.

overflow 발생을 미리 확인하는 회로를 구현하기 위해 Leading 0/1 검출기로 2의 보수인 두 입력으로부터 부호비트와 동일한 비트를 확인한다. 두 입력에서, 한 입력 A 가 양수이고, 부호비트와 같은 비트가 연속되는 비트수가 p 개이면, 입력 A 는 다음의 범위를 가진다.

$$2^{N-p} > A \geq 2^{N-p-1} \quad (5)$$

한편 입력 B 가 음수이고, 부호비트와 같은 비트가 연속되는 비트수가 q 개이면,

$$2^{N-q} \geq |B| > 2^{N-q-1} \quad (6)$$

의 범위를 갖는다.

그 결과 곱셈기의 출력 P 는 세 가지 경우를 가진다. 첫 번째 두 입력이 양수일 때는,

$$2^{2N-p-q} > P \geq 2^{2N-p-q-2} \quad (7)$$

의 범위를 가지고, 두 번째 두 입력의 부호가 다를 때는,

$$2^{2N-p-q} > |P| > 2^{2N-p-q-2} \quad (8)$$

의 범위를 가지며, 마지막으로 두 입력이 모두 음수일 때는,

$$2^{2N-p-q} \geq P > 2^{2N-p-q-2} \quad (9)$$

의 범위를 가진다.

위의 식(7)~(9) 를 종합하면, 출력 P 의 범위를 결정하면 식(10) 과 같다.

$$2^{2N-p-q} \geq |P| \geq 2^{2N-p-q-2} \quad (10)$$

출력의 범위를 나타내는 식(10)로부터 overflow가 발생하는 조건을 구할 수 가 있다.

두 입력이 양수일 때는 곱셈기의 출력이 양수이어야 하기 때문에 $|P| \geq 2^{N-1}$ 이면 overflow가 발생한다. 식(10)과 비교하면, $2N-p-q-2 \geq N-1$ 고, 따라서

$$p+q \leq N-1 \quad (11)$$

일 경우에 overflow가 발생한다. 또한 두 입력의 부호가 서로 다를 때에는 $|P| > 2^{N-1}$ 일 때 overflow가 발생하기 때문에 식(11)과 같은 조건일 때 overflow가 발생한다.

곱셈기는 overflow 발생조건 외의 경우에 유효하므로 Leading 0/1에서 검출한 최소 0/1의 개수는 $p+q=N$ 이어야 한다. 이 최소 값을 식(10)에 대입하여 곱셈기가 표현될 가장 큰 수의 범위를 구하면 다음과 같다.

$$2^N \geq |P| \geq 2^{N-2} \quad (12)$$

여기서 overflow가 발생하는 조건은 $|P|=2^N$ 가 되는 경우이고, 이때의 조건은 다음과 같다.

$$(p+q=N) \wedge (P_{n-1}=0) \wedge (P_{n-2}=0) \quad (13)$$

식 (13)은 두 입력이 모두 음수일 때 발생한다.

또한 $|P|=2^{N-1}$ 인 경우는 입력과 출력의 $N-1$ 번째 비트가 부호 비트이기 때문에 두 입력과 출력을 비교하여 overflow 발생을 다음과 같이 확인할 수 있다.

$$(A_{n-1}) XOR (B_{n-1}) XOR (P_{n-1}) \quad (14)$$

식 (11), (13), (14)의 3가지 overflow 발생 조건을 검색하는 회로를 부가하여, 상위 N 비트를 구현하는 하드웨어를 대체하였다.

III. 곱셈기의 구조

Leading 0/1 검출기는 부호 비트와 동일한 비트의 개수를 확인하는 회로이다. 부호 비트와의 동일성 여부를 판별하면, 2의 보수의 표현에서 부호에 관계없이 입력되는 수의 절대값을 알 수 있다. 부호 비트 자신과의 비교는 항상 '0' 이 되고, 부호 비트에서 최하위 비트까지의 비교에서 연속적으로 '0' 이면 그 수는 '0' 이 되기 때문에 그 곱은 항상 '0' 이다. 따라서 최하위 비트와의 비교 값은 어떤 값이던지 관계가 없기 때문에 항상 '1' 로 둘 수 가 있다. 최하위 비트를 항상 '1' 로 정의하면, 검출된 Leading 0/1 의 개수를 N 개 보다 작게 하여 검출기에서 판별한 '0' 의 개수를 $\log_2 N$ 비트로 표현 가능하다. 그림 1. 은 16 비트 Leading 0/1 검출기의 예이다.

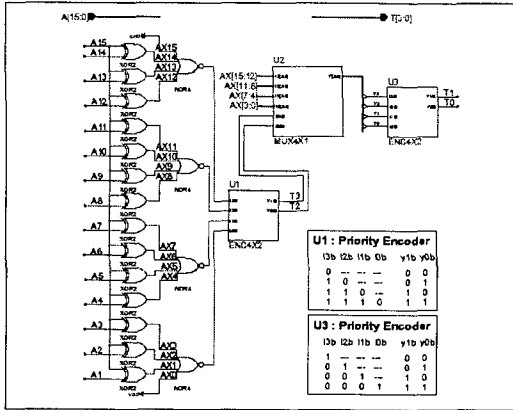


그림 1. 16 bit Leading 0/1 detector

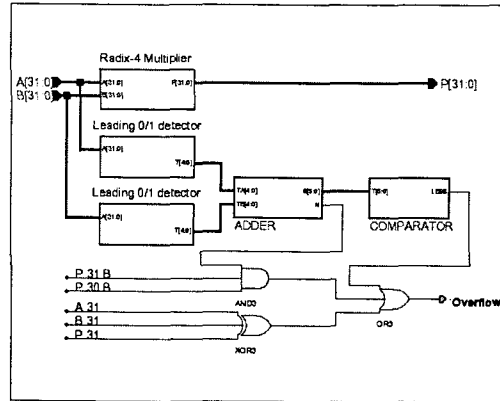


그림 2. 32 bit Radix-4 Modified Booth's Multiplier with leading 0/1 detector

2의 보수에서 부호 비트와 같은 비트를 판별하기 위해 입력의 부호 비트와 하위 비트를 XOR 취했고, 부호 비트 자신과의 비교는 XOR 대신 항상 '0' 이고, 최하위 비트는 항상 '1' 이다. 그리고 '0'의 개수를 판별하는 회로의 속도와 면적의 효율을 높이기 위해 먼저 4 비트씩 그룹으로 모아 인코딩한 후, 부호 비트와 다른 비트가 있는 그룹을 확인하고, 다시 그 그룹 내에서 최초로 부호 비트와 다른 비트를 확인하여 부호 비트와 같은 비트가 연속되는 개수를 계산한다. 32 비트에서 Leading 0/1 을 판별할 경우에는 8×3 우선순위 인코더와 4×2 우선순위 인코더를 조합하는 방법으로 확장한다. 우선 순위 인코더에 대한 진리표는 그림 1. 내에 표현되었다.

현재까지 고안된 곱셈기 중 가장 빠른 것으로 알려진 곱셈기는 Booth encoder, Booth selector 그리고 Wallace tree 구조[6] 나 (7,3) 가산기[7]를 가진 방식이다. 이 곱셈기에 overflow 검색회로를 추가하여 면적 절약형 곱셈기를 구현하였다. 곱셈기의 상위 비트 출력 P_H를 생성하는 부분을 Leading 0/1 검출기와 overflow를 판별하는 회로로 대체하고, 하위 비트 출력 P_L 은 일반적인 곱셈기와 거의 동일하다. 차이점은 P_{n-1} 비트가 곱셈의 부호를 결정하는 비트이기 때문에 P_{n-1} 을 생성하는 부분 곱을 더하는 대신, XOR을 취해 회로의 면적을 좀 더 줄일 수 있도록 하였다. 그림 2. 는 32 비트 Radix-4 수정형 Booth 곱셈기에 적용한 예이다.

이 곱셈기는 Booth encoder, Booth selector 그리고 부분 곱의 합을 구현하는 부분에 대해서 하위 N 비트를 생성하는 부분만을 구현하고, 두 입력에 대하여 부호와 연속되는 비트를 판별한 후, 그 개수를 이용하여 overflow를 생성하는 회로로 구현하였다.

Wallace tree 구조는 455 개의 전가산기, 46 개의 반가산기 그리고 57 비트의 CPA(carry propagate adder)가 필요하다. 그리고 응답속도는 7 개의 전가산기와 57 비트의 CPA를 통과하는 시간이 된다. (7,3) 가산기로 구현할 경우 452 개의 전가산기, 63개의 반가산기 그리고 60 비트의 CPA 로 구현된다. 응답속도는 9 개의 전가산기와 60 비트의 CPA를 통과하는 시간이다. 그러나 실제 응답속도는 상호배선이 작은 (7,3) 가산기로 구현된 곱셈기이다[7]. 따라서 본 논문에서는 (7,3) 가산기를 적용하였다.

IV. 구현 및 평가

Leading 0/1 검출 기능을 부가한 곱셈기는 VHDL로 구현하여 동작을 확인하였다. N 비트의 입력을 받아 N 비트 이상의 결과가 출력될 때, overflow 확인과 곱셈기의 동작을 검정하였다. 동작 검정의 정확성을 높이기 위해 C 언어의 난수 생성기를 사용하여, 임의의 32 비트 정수를 생성한 후 곱셈기의 입력으로 사용하여, overflow 검출과 곱셈 결과를 확인하였다.

VHDL로 구현한 곱셈기를 0.25μm 아남 1-poly 5-metal CMOS 공정을 이용하여 합성하였다. Leading 0/1 검출기는 16 비트 × 16 비트 곱셈기의 경우는 129 gate, 32 비트 × 32 비트 곱셈기는 252 gate 로, 곱셈기 전체 면적의 약 10 % 정도를 차지하였다. 따라서 전체 면적은 N 비트 × N 비트 연산이 2N 비트를 생성하는 대부분의 곱셈기에 비해 절반으로 줄어든다. 표 1. 에서 16 비트 곱셈기와 32 비트 곱셈기에서 Leading 0/1 검출기의 면적율을 요약하였다.

	Gate Count Leading 0/1 detector	Total Gate Count	Area rate of Leading 0/1 detector
16 bit	130	1100	11.8 %
32 bit	250	3900	6.4 %

표 1. Area rate of the Leading 0/1 detector

V. 결론

Leading 0/1 검출기를 추가하여 면적의 효율을 높인 곱셈기를 제안하였다. 그리고 부가적으로 회로가 절반으로 줄어들기 때문에 전력 소모 또한 약 45% 줄어들 것으로 기대된다. 대부분의 곱셈이 N 비트의 입력을 받아 N 비트 이하의 결과를 생성하기 때문에 상위 N 비트의 부분 곱과 합을 구현하는 회로를 Leading 0/1 검출기와 overflow를 판별하는 회로로 대체하여, 45% 정도의 면적을 절약하는 곱셈기를 구현하였다.

면적의 효율을 향상시킨 이 곱셈기는 실장제어분야와 같이 낮은 비용으로 높은 성능을 요구하는 부분에서 우수한 가격 대 성능을 줄 것으로 기대된다.

참고문헌

- [1] A. D. Booth. "A signed binary multiplication technique", Quarterly J. Mechanics, Appl. Math, vol. 4, Part 2, pp. 236-240, 1951.
- [2] L. P. Rubinfeld. "A proof of the modified Booth's algorithm for multiplication", IEEE Trans. Computers, vol. C-24, no. 10, pp.1014-1015, Oct. 1975.
- [3] H. Sam and A. Gupta. "A generalized multibit recoding two's complement binary numbers and its proof with application in multiplier implementations", IEEE Trans. on Computers, vol. 39, no. 8, pp.1006-1015, Aug. 1990.
- [4] Zhongde Wang, Graham A. Jullien, William C. Miller, "A New Design Technique for Column Compression Multipliers", IEEE Trans. on Computers, vol. 44, no. 8, pp. 962-970, Aug. 1995.
- [5] Sunder S. Kidambi, Fayez El-Guibaly, Andreas Antoniou, "Area-Efficient Multipliers for Digital Signal Processing Applications", IEEE Trans. on Circuits and Systems-II: Analog and Digital

Signal Processing, vol. 43, no. 2, pp. 90-95, Feb. 1996.

- [6] C. S. Wallace. "A suggestion for a fast multiplier", IEEE trans. on Electronic Computers, pp. 14-17, Feb. 1964.
- [7] M. Mehta, V. Parmar, E. Swartzlander, "High-speed multiplier design using multi-input counter and compressor circuits", Computer Arithmetic, Proceedings., 10th IEEE Symposium on , pp. 43-50, 1991.