DBpia

고속 4:2 컴프레서를 사용한 16×16 비트 병렬곱셈기 구조

정회원 진 용 선*, 정 정 화**

A 16x16 bit Parallel Multiplier Architecture using High Speed 4:2 Compressor

Yong-Sun Jin*, Jong-Wha Chong** *Regular Members*

요 약

본 논문에서는 고속 4-2 compressor를 사용한 계층적인 구조를 갖는 16×16 비트 병렬 곱셈기를 제안한다. 병렬곱셈기는 일반적으로 4-2 compressor를 사용한 Wallace tree로 구성된 부분곱 덧셈트리 블록 외 처리속도에 영향을 받는다. 따라서, 병렬곱셈기의 성능개선을 위해서 고속으로 동작하는 4-2 compressor의 설계가 필요하다. 그러나, 기존회로에서는 carry 신호의 전달지연시간이 길어지는 문제점이 있으므로, 본 논문에서는 carry를 입력신호의 초기단계에서부터 처리하도록 4-2 compressor 회로를 설계하여 전달 지연 시간을 감소시킬 수 있는 방법을 제안하였다. 또한, 제안하는 계층적인 곱셈기 구조는 부분곱 생성시 피승수와 승수의 비트수 비율을 2:1로 유지하면서 부분 모듈을 분할 처리함으로써 곱셈기의 입력 비트 수가 증가하여도 쉽게 곱셈기를 모듈화 하여 확장할 수 있다. 제안하는 4-2 compressor 회로를 HSPICE 시뮬레이션 하여 기존의 4-2 compressor 회로와 비교한 결과 전달 지연 시간을 14% 감소시킬 수 있었다.

ABSTRACT

In this paper, a high speed 16x16 bit parallel multiplier with a hierarchical architecture using a fast 4-2 compressor is proposed. Generally, the performance of parallel multiplier is depend on the multiplication speed of partial products summation tree that consists of wallace tree using 4-2 compressor. In order to improve the performance of parallel multiplier, the design of high speed 4-2 compressor is needed. But, the previous circuit of 4-2 compressor have the problem with the long propagation delay time of carry signal. This paper propose the new circuit of 4-2 compressor which reduce the propagation delay time. The new circuit is designed to deal with the carry signal from the beginning of input signal. The proposed hierarchical structure of multiplier can easily make possible a modular and extendable design of multiplier in spite of the large number of input bits by partionning the multiplier to submodules and keeping the bit ratio of multiplicand and multiplier 2:1 when the partial products are prepared. The propagation delay time of the proposed 4-2 compressor circuit was 14% reduced as a result of HSPICE simulation, compared with the existing 4-2 compressor circuit.

Ⅰ. 서 론

최근 정보통신의 발달로 멀티미디어 정보 전송이 요구되어짐에 따라 고품질 정보의 실시간 처리가 필요하게 되었다. 멀티미디어 정보 처리는 데이터가 디지털로 변환되며 이에 따라 실시간에 많은 양의 데이터의 처리를 위해 고속 곱셈기가 필요하게 되

* 경문대학 정보통신과(ysjin55@unitel.co.kr)
** 한양대학교 전자공학과(jchong@email.hanyang.ac.kr)
논문번호 : 00002-0206, 접수일자 : 2000년 2월 6일

www.dbpia.co.kr

었다. 고속 곱셈을 위한 곱셈기 구조는 어레이 곱셈기와 병렬 곱셈기로 나눌 수 있다.[4] 어레이 곱셈기는 단위 셀의 규칙성 및 VLSI 구현이 쉽다는 장점을 가지고 있으나 계산 속도가 곱해지는 비트 수에 비례한다는 단점을 가지고 있으며, 병렬 곱셈기의 경우 곱셈 비트 수가 증가 함에 따라 곱셈 시간이 어레이 곱셈기 보다 적다는 장점을 가지고 있다.[5] 따라서 목적 목적에 따라 다양한 형태의 하드웨어 구조가 제안되고 있으며 응용 목적에 따라 적절한 선택이 요구된다. 즉, 파이프라인 방식의 병렬 곱셈기의 경우 특정 DSP 알고리듬 구현에 적합하고 트리 구조의 병렬 곱셈기의 경우 마이크로 프로세서나 범용 DSP 프로세서에 적합하다.[10] 곱셈 과정은 일련의 부분곱 덧셈 과정을 반복적으로 수행하는 것이기 때문에 고속 승산을 위해 부분곱의 수를 줄이므로서 계산 단계를 줄이는 수정형 booth 알고리듬이 적용되고 있으며 또한 Wallace tree를 적용하여 부분곱의 빠른 덧셈을 수행하고 있다.[11][12] 그러나 [1]에서는 booth 알고리듬의 하드웨어 구현이 부분곱 수의 감소를 상쇄시킬 만큼 크기 때문에 booth 알고리듬을 적용하지 않고 부분곱을 생성하는 곱셈기를 제안 하고 있다.

본 논문에서는 [1]에서 제안한 것처럼 수정형 booth 알고리듬을 적용하지 않고 부분곱을 생성하며 빠른 부분곱 덧셈을 위하여 일반적인 4-2 compressor 회로보다 전달지연시간을 감소시킨 고속 4-2 compressor 회로를 설계하여 고속 병렬 곱셈기의 처리속도를 향상시켰다. 또한 비트 수의 증가에 따른 곱셈기의 확장설계가 용이하도록 규칙성을 갖는 계층적인 구조의 곱셈기를 제안하였다.

Ⅱ. 곱셈 처리 방식

부호화된 정수에 대한 곱셈 수행시 승수가 음수인 경우 승수의 MSB에 의해 발생되는 부분곱은 2의 보수를 취하여 계산이 되어야 한다. 또한 $n \times n$ 비트 곱셈기를 설계하고자 할 때 발생되는 부분곱의 덧셈 시, 자리수를 맞추기 위해 부호 비트가 확장되어 덧셈이 이루어 진다. 본 장에서는 이러한 문제를 별도의 하드웨어의 부가 없이 부분곱의 덧셈 과정에서 처리하는 방법에 대해 설명하고 이를 적용한다.

1. 부분곱에 대한 2의 보수

부분곱에 대한 2의 보수는 그림 1과 같이 승수가 음수인 경우에 이루어 진다. 즉, 승수의 MSB에 의해 생성된 부분곱에 대해 2의 보수를 취해야 한다. 그림 1에서 2의 보수를 수행하는 과정은 그림 2(a)와 같이 부분곱에 1의 보수를 취하고 다시 1을 더함으로서 이루어 진다. 그러나 그림 2(a)와 같이 1을 더해주는 것은 하드웨어의 추가가 필요하게 된다. 따라서 그림2 (b)와 같이 부분곱에 대한 2의 보수는 하위 비트에 1을 채우므로서 그림 2(a)의 4번째 비트 위치에 있는 1을 대체할 수 있다. 즉, i번째 비트가 1이기 위해서는 i-1번째에 1의 값을 갖는 비트가 2개 있어야 한다. 이를 반복적으로 표현하면 식 (1)과 같다.

$$
\begin{aligned}
2^{i} &= 2^{i-1} + 2^{i-1} \\
2^{i-1} &= 2^{i-2} + 2^{i-2} \\
&\vdots \\
2^{1} &= 2^{0} + 2^{0} \\
\therefore\ 2^{i} &= 2^{i-1} + 2^{i-2} + \cdots\cdots + 2^{0} + 2^{0}
\end{aligned}
\qquad (1)
$$

따라서 그림 2(b)와 같이 2개의 1이 올 수 있는 가능한 자리까지 i-1번째 비트부터 LSB까지 1을 추가함으로서 1의 보수에서 2의 보수로 변환할 수 있다. 그림 2(b)와 같이 부분곱에 대해 2의 보수를 할 경우 4-2 compressor에 의해 부분곱의 덧셈이 이루어 질 수 있으며 보다 규칙적인 구조를 갖게 된다.

```
   0111  (+7)
 × 1011  (-5)
  0111
  0111
 0000
 1001 ←——  +7에 대한 2의 보수
1011101 (-35)
```

그림 1. 곱셈의 예

```
   0111  (+7)
   1011  (-5)
  0111
  0111
 0000
 1000  ←——  +7에 대한 1의 보수
   1   ←——  2의 보수를 위해
1011101  (-35)
```

(a) 그림 1. 에 대한 보수계산

www.dbpia.co.kr

```
      0111   (+7)
      1011   (-5)
      ----
      0111
     0111
    00001
  100011
  -------
  1011101     (-35)
```

(b) 하위 비트를 이용한 보수 계산

그림 2. 2의 보수를 위한 방법

2. 부호 확장 처리

곱셈기에서 부분곱 간의 덧셈 수행시 그림 3과 같이 부호 비트를 확장하여 계산을 수행하여야 올바른 계산을 수행할 수 있다. 그러나 그림 3과 같이 부호 비트를 확장하여 덧셈을 수행하면 덧셈기가 추가로 필요하게 되어 하드웨어의 면적이 증가한다. 따라서 그림 4 (a)와 같이 부호 확장을 처리함으로서 하드웨어 면적을 줄일 수 있다. 그러나 추가되는 '1'비트에 대해 규칙적인 구조의 하드웨어를 구현하기가 어렵다. 이를 그림 2의 보수를 취하는 방법과 같은 방법을 적용하여 그림 4(b)와 같이 구현하면 보다 규칙적인 모양의 곱셈기를 구현할 수 있다.

```
                 1001   (-7)
               × 0111   (+7)
              ---------
           ┌─ 1111001
확장부호비트 ─┤  111001
           └─ 11001
              0000
              -------
              1001111   (-49)
```

그림 3. 부호 확장의 예

```
   1001  (-7)              1001  (-7)
 × 0111  (+7)            × 0111  (+7)
 -------                 -------
   1                       0001
   0001                    0001
  0001                   00011
 0001                   1000 11
1000
-------                 -------
1001111 (-49)           1001111 (-49)

   (a)                     (b)
```

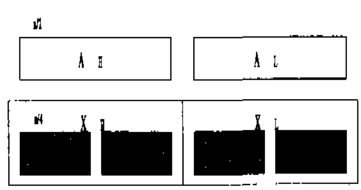
그림 4. 부호 확장 의 예

Ⅲ. 제안하는 병렬 곱셈기 구조

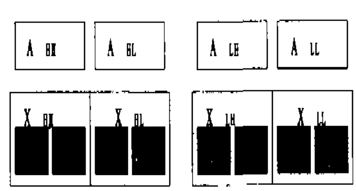
1. 계층적인 곱셈기 구조

본 논문에서는 입력이 n 비트인 곱셈기를 설계하기 위한 계층적인 곱셈기 구조를 제안한다. 제안하는 기본적인 곱셈기 형태는 그림 5와 같이 승수와 피승수의 곱셈을 n 비트 와 n/2 비트 로 나누어 곱하는 것이다. 적은 비트의 곱셈기에 의해 n×n 비트 곱셈기를 설계하고자 할 경우 승수와 피승수는 2의 배수에 의해 분할되어야 하고 곱해지는 비트의 크기는 항상 2:1을 유지한다. 이것을 그림 5에 표시하였다.

(a) 분할하지 않은 경우



(b) 1번 분할 경우



(c) 2번 분할 경우

그림 5. n×n 곱셈기의 계층적 분할

이것은 부분곱을 더하기 위하여 그림 6과 같은 덧셈 형태를 갖고 있기 때문이다. 그림 6의 각각의 부분곱 (가)(나)(다)(라)에 대한 부분곱 덧셈을 수행한후 (가)와 (나), (다) 와 (라)를 더한후 이 결과를 다시 더하여 A×X의 부분곱에 대한 합을 구한다.

www.dbpia.co.kr

이러한 덧셈 과정은 승수와 피승수를 나누어 곱셈을 수행하더라도 부분곱의 덧셈시 균형된 트리를 유지하기 위해서이다.

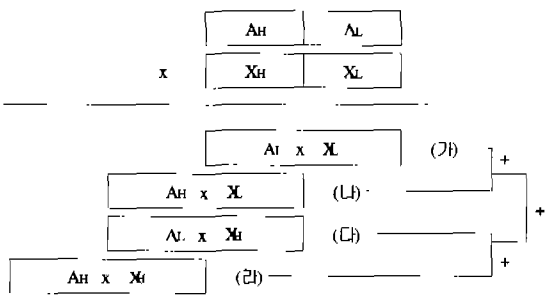


그림 6. 부분곱의 덧셈

그림 5와 그림 6에서 제시한 곱셈 과정에 대한 곱셈기 구조를 16×16 비트 곱셈기를 예를 들어 그림 7에 표시하였다.

피승수의 16 비트 입력 데이터는 2로 나누어지고 승수의 16 비트 입력 데이터는 4로 나누어져 8×4 부분곱 생성에 의한 16×16 곱셈이 처리되도록 하였다, 이때 피승수와 승수의 곱해지는 비트 수는 본 논문에서 제안하는 대로 2 : 1을 유지한다. 8×4 부분곱 생성을 선택한 이유는 그림 8 에서 제안하는 고속 4-2 compressor를 사용하고 부분곱 덧셈을 위한 규칙적인 구조의 덧셈 트리를 취할 수 있기 때문이다.

2. 고속 4-2 compressor 설계

고속 병렬 곱셈기는 3가지의 주요 부분으로 구성된다. 부분곱의 수를 줄이기 위한 수정형 booth 엔코더, 만들어진 부분곱을 더하는 Wallace tree 구조를 통한 부분곱 덧셈트리, 그리고 최종결과를 더하는 CLA (Carry Look-Ahead Adder) 덧셈기로 구성된다.

그중 부분곱 덧셈 트리가 전체 고속 병렬 곱셈기의 처리 속도를 좌우하는 커다란 비중을 차지한다. 부분곱 덧셈기는 4-2 compressor를 사용한 Wallace tree 구조를 통하여 일반적으로 구현된다. 따라서 빠른 처리속도를 갖는 4-2 compressor 의 설계는 고속 병렬 곱셈기 의 처리 속도를 개선하는데 중요한 역할을 하게된다.

본 논문에서는 기존에 제안되고 있는 일반적인 4-2 compressor에[2][3] 비교하여 빠른 처리 속도를 가진 새로운 고속 4-2 compressor를 제안한다.

그림 8에 설계된 바와 같이 최장지연경로(critical path)인 입력신호 대 carry 출력의 회로를 논문[2]에 나타난 단점을 개선하여 carry 출력을 얻기 위한 입력 신호의 처리를 최 앞단에서부터 처리하도록 논리회로를 설계함으로써 최장지연경로의 입력대 출력의 지연 시간을 감소시켰다.

이를 확인하고자 본 논문에서 제안하는 4-2 compressor와 기존 제안되고 있는 4-2 compressor[2][3]를 HSPICE를 통한 회로 시뮬레이션을 하여 입력대 최장지연경로의 전달지연(propagation delay) 시간을

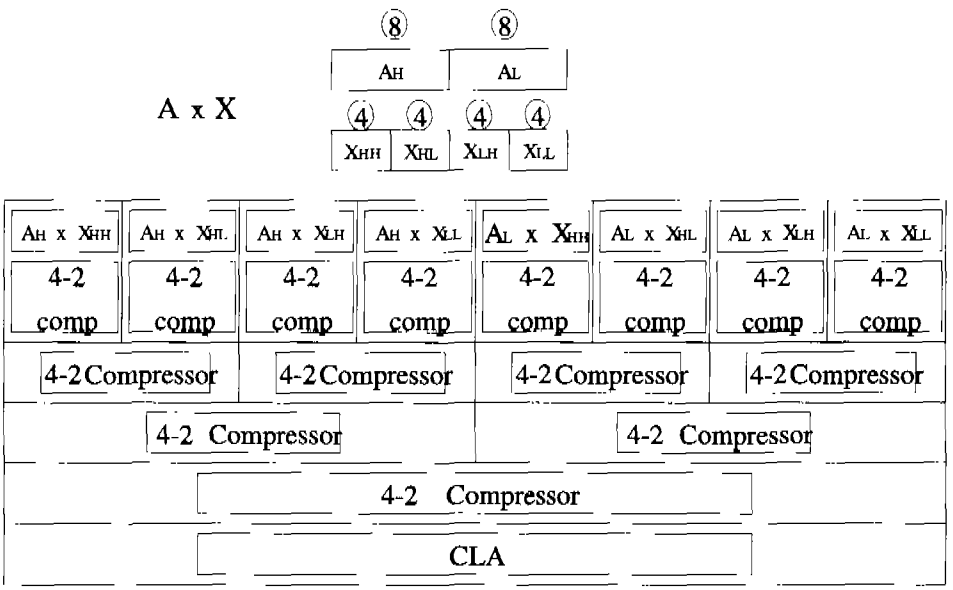
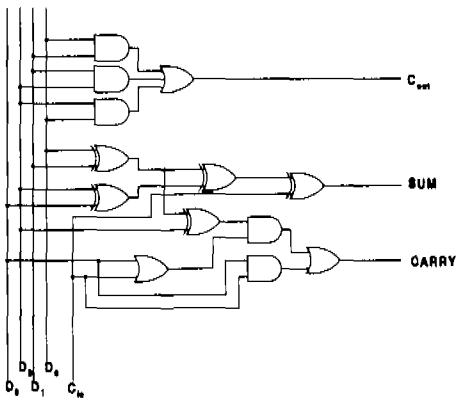


그림 7. 제안하는 계층적인 곱셈기 구조

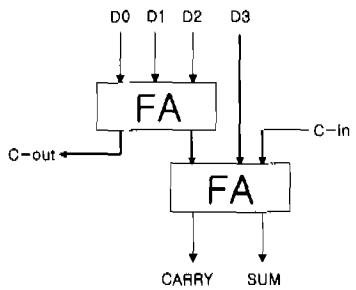
www.dbpia.co.kr

시뮬레이션하여 비교 하였다.

이때 사용된 회로는 기존에 발표된 DPL(Double Pass-Transistor Logic)[6]을 사용하여 구성하였고 HSPICE의 모델 파라미터는 0.8um CMOS 공정, 공급전압 3V에서 회로 시뮬레이션 하였다.

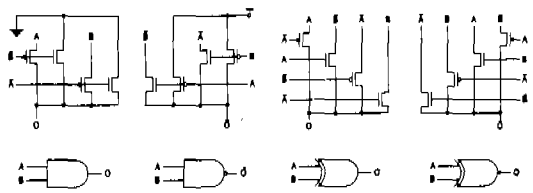


(a) 논리 회로도

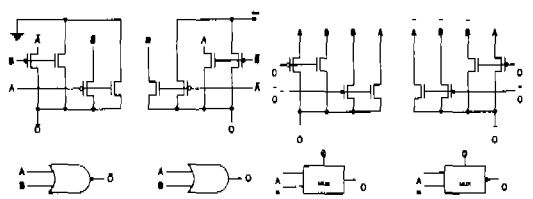


(b) 블럭도

그림 8. 제안하는 4-2 compressor



(a) AND/NAND (b) XOR/XNOR



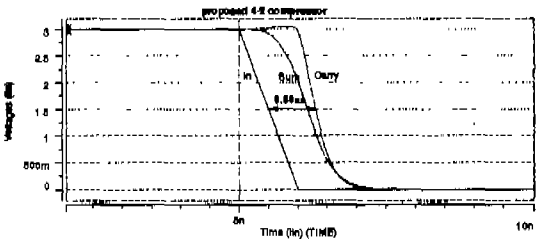
(c) OR/NOR (d) Multiplexer

그림 9. 시뮬레이션에 사용한 Double pass-transistor logic (DPL) 게이트

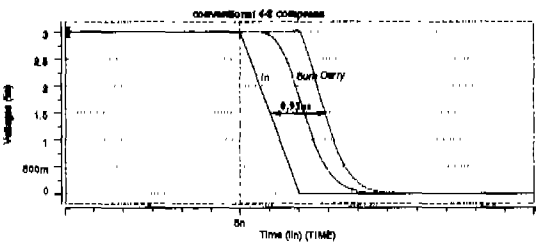
Ⅳ. 실험 및 고찰

본 논문에서 제안한 고속 4-2 compressor 의 성능 평가를 위하여 기존의 방식에 의한 4-2 compressor와 제안된 방식의 4-2 compressor의 회로를 DPL 로직을 사용하여 구성하고 HSPICE 의 모델 파라미터는 0.8um CMOS 공정, 전원전압 3V에서 회로 시뮬레이션 하였다. HSPICE를 사용한 회로 시뮬레이션 결과는 그림 10 에 나타내었다.

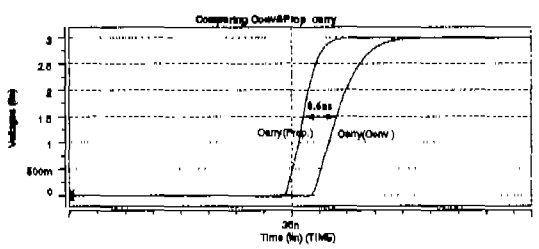
입력펄스 신호의 하강시간(falling time)이 1[ns] 일 때 본 논문에서 제안한 4-2 compressor의 전달 지연시간은 0.8[ns] 이고, 일반적인 4-2 compressor는[2][3] 각각 0.93[ns], 0.94[ns] 이었다. 결과적으로 본 논문에서 제안한 4-2 compressor의 전달지연시간이 일반적인 4-2 compressor에 비하여 14% 감소되어 동작 처리 속도가 개선되었다.



(a) 제안하는 4-2 compressor



(b) 일반적인 4-2 compressor



(c) 상승시간(rising time)에서 carry 전달지연시간 비교

그림 10. HSPICE 회로 simulation 결과

www.dbpia.co.kr

본 논문에서 제안하는 4-2 compressor와 기존 발표된 여러 가지 4-2 compressor[2][3]의 전달 지연시간의 비교를 HSPICE 시뮬레이션을 하여 그림 11에 정리 하였다. 그림 11에 나타난 바와 같이 본 논문에서 제안한대로 설계된 4-2 compressor가 가장 빠른 처리속도를 가지고 있다.

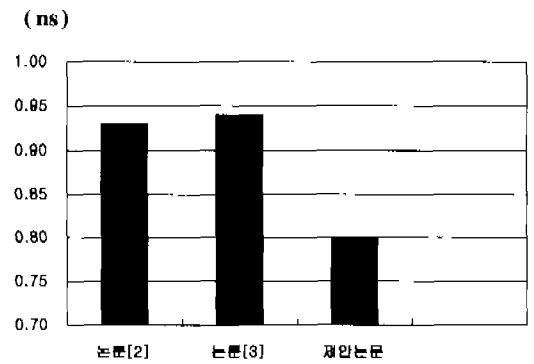


그림 11. 여러 가지 4-2 compressor 의 처리속도 비교

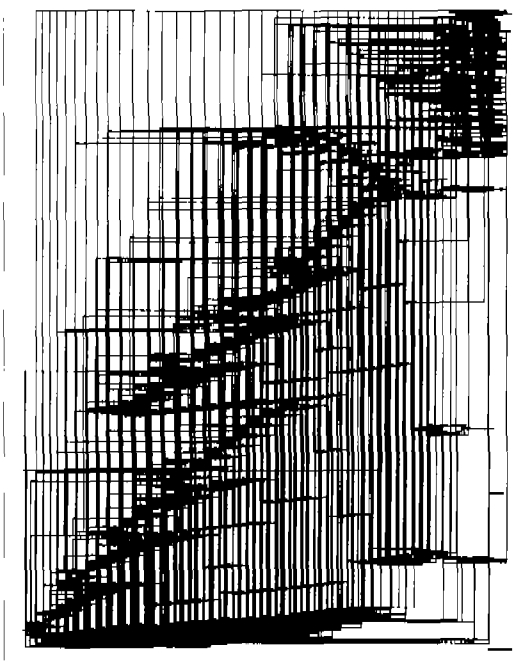


그림 12. 16×16 bit 곱셈기 회로

또한 본 논문에서 제안한 계층적 구조의 병렬 곱셈기의 회로 동작을 검증하기 위하여 본 논문에서 제안한 고속 4-2 compressor(그림 8)와 계층적인 곱셈기 구조(그림 7)를 적용하여 16×16 비트의 곱셈기를 그림 12와 같이 구현 하였다. 이때 사용한 부분곱 생성기는 booth 알고리즘을 사용하지 않고 AND 게이트의 배열로 구성하였으며 부분곱 덧셈트리는 본 논문에서 제안한 4-2 compressor를 사용하였고 최종결과 덧셈은 CLA (Carry Look- Ahead Adder) 덧셈기를 사용하였다.

그림 13 은 그림 12의 논리 시뮬레이션 결과로서 설계된 회로의 논리가 정확하게 동작하고 있음을 보여주고 있다.

V. 결 론

본 논문에서는 부분곱 덧셈 트리의 처리속도를 향상시키기 위하여 일반적인 4-2 compressor 회로보다 처리속도가 향상된 고속 4-2 compressor 회로를 제안 하였다. 제안한 고속 4-2 compressor 회로를 사용하여 16×16 비트 고속 병렬 곱셈기를 설계한 결과 일반적인 4-2 compressor 회로를 사용한 병렬곱셈기에 비하여 전달지연시간이 10% 이상 감소 되었다. 또한, 본 논문에서 제안하는 병렬곱셈기는 계층적인 구조를 가지기때문에 입력 비트수 증가에 따른 확장성이 좋아 병렬곱셈기의 확장설계에 편리하다. 제안하는 곱셈기는 고속처리속도를 가지기 때문에 멀티미디어 데이터의 실시간 신호 처리에 응용될수 있으며 확장성이 좋아 모듈화된 컴파일러 에 적용가능하다. 앞으로는 부분곱 덧셈의 처리속도를 보다 빠르게 향상시키기 위하여 부분 곱의 재배열 방법과 여러 가지 종류의 다입력 compressor 설계에 대한 연구가 뒤따라야겠다.

그림 13. 회로 그림 12. 의 논리 시뮬레이션 결과

www.dbpia.co.kr

참 고 문 헌

[1] Mark R. Santoro and Mark A. Horowitz, "SPIM : A Pipelined 64×64 bit Iterative Multiplier," IEEE J. Solid-State Circuits, vol 24, No. 2, pp.487-493, Apr 1989.

[2] Masato Nagamatsu, et al., "A 15-ns 32×32-b CMOS Multiplier with An Improved Parallel Structure," IEEE J. Solide-State Circuits, vol. 25, pp.494-497, Apr 1990.

[3] Shen-Fu Hsaio, Ming-Roun Jiang and Jia-Sien Yeh, "Design of high speed low-power 3-2 counter and 4-2 compressor for fast multiplier," ELECTRONICS LETTERS, vol. 34, No. 4, pp.341-343, Feb 1998.

[4] Israel Koren, "Computer Arithmatic Algorithms", John Wiley Inc., pp.71-123, 1993.

[5] Yoshita Harata, et al., "A High-Speed Multiplier Using a Redundant Binary Adder Tree," IEEE J. Solide-State Circuits, vol. sc-22, No. 1, pp.28-33, Feb 1987.

[6] Makoto Suzuki, et al., "A 1.5ns 32-b CMOS ALU in Double Pass-Transistor Logic" IEEE J. Solide-State Circuits, vol. 28, No. 11, pp.1145-1151, Nov 1993.

[7] Norio Ohokubo, et al., "A 4.4 ns CMOS 54×54-b Multiplier Using Pass-Transistor Multiplexer," IEEE J. Solide-State Circuits, vol. 30, No. 3, pp.251-257, Mar 1995.

[8] Kazuo Yano, et al., "A 3.8-ns CMOS 16x16-b Multiplier Using Complementary Pass-Transistor Logic," IEEE J. Solide-State Circuits, vol. 25, No. 2, pp.388-395, Apr 1990.

[9] C.F. Law and S.S. Rofail, "A Low-Power 16×16-b Parallel Multiplier Utilizing Pass-Transistor Logic," IEEE J. Solide-State Circuits, vol. 34, No. 10, pp.1395-1399, Oct 1999.

[10] 김득경, 신경욱, 이용석, 이문기, "4-비트 리코딩 기법을 이용한 64×67 병렬 승산기 설계," 대한전자공학회 추계종합학술대회 논문집 17권 2호, pp.1289-1292, 1994.

[11] 강정엽, 이원형, 권오영, 한탁돈, "4-2 콤퓨레서를 이용한 승산기 모듈생성기의 설계," 대한전자공학회 추계종합학술대회 논문집 16권 2호, pp.388-392, 1993.

[12] 김영민, 조진호, "32×32 비트 고속 병렬 곱셈기 구조," 전자공학회논문지 31권 B편 10호 pp. 1505-1510, 1994

진 용 선(Yong-Sun Jin) 정회원

1978년 9월 : 한양대학교 전자공학과 졸업(학사)
1991년 2월 : 한양대학교 대학원 전자공학과 졸업 (공학석사)
1994년 2월 : 한양대학교 대학원 전자공학과 박사과정 수료
1978년 9월~1979년 12월 : 한국전자통신연구소
1980년 1월~1988년 1월 : LG반도체 연구소
1988년 2월~1996년 6월 : TEXAS INSTRUMENT 연구소
1997년 3월~현재 : 경문대학 정보통신과 조교수
<주관심 분야> 신호처리, VLSI설계, 디지털 통신 시스템

정 정 화(Jóng-Wha Chong) 정회원

한국통신학회 논문지 제25권 제1호 참조
1980년3월~현재 : 한양대학교 전자공학과 교수

www.dbpia.co.kr