

논문 2009-46SD-12-6

Modified Booth 곱셈기를 위한 고성능 파이프라인 구조

(High-performance Pipeline Architecture for Modified Booth Multipliers)

김수진*, 조경순**

(Soojin Kim and Kyeongsoon Cho)

요약

본 논문은 modified Booth 곱셈기를 위한 고성능 파이프라인 구조를 제안하고 있다. 제안하는 곱셈기 회로는 곱셈 속도를 향상시키기 위해 가장 널리 사용되는 기술인 modified Booth 알고리즘과 파이프라인 구조에 기반을 두고 있다. 최적의 파이프라인 곱셈기를 구현하기 위해 많은 실험이 수행되었다. 파이프라인의 단 수가 증가할수록 회로 속도 향상율이 회로 크기 증가율보다 더 크며, 파이프라인 레지스터를 적절한 위치에 삽입하는 것이 중요하다는 사실이 실험 결과를 통해 확인되었다. 제안하는 modified Booth 곱셈기 회로를 Verilog HDL로 설계하였으며 0.13um 표준 셀 라이브러리를 이용하여 게이트 수준 회로로 합성하였다. 합성된 회로는 다른 곱셈기들에 비해 좋은 성능을 나타내었으며, GHz 범위에서 동작할 수 있으므로 광통신 시스템과 같은 극히 높은 성능을 필요로 하는 응용 시스템에서 사용될 수 있다.

Abstract

This paper proposes the high-performance pipeline architecture for modified Booth multipliers. The proposed multiplier circuits are based on modified Booth algorithm and pipeline architecture which are the most widely used techniques to accelerate the multiplication speed. In order to implement the optimally pipelined multipliers, many kinds of experiments have been conducted. The experimental results show that the speed improvement gain exceeds the area penalty and this trend is manifested as the number of pipeline stages increases. It is also important to insert the pipeline registers at the proper positions. We described the proposed modified Booth multiplier circuits in Verilog HDL and synthesized the gate-level circuits using 0.13um standard cell library. The resultant multiplier circuits show better performance than others. Since they operate at GHz ranges, they can be used in the application systems requiring extremely high performance such as optical communication systems.

Keywords : 곱셈기, 파이프라인 구조, 고성능, modified Booth 알고리즘

I. 서론

DSP(Digital Signal Processing)는 멀티미디어와 모바일 통신 시스템의 핵심 기술 중 하나이다. 특히 차세대 광통신 시스템과 같은 DSP에 기반한 시스템에서는 많은 양의 데이터를 고속으로 처리해야 할 필요성이 점

점 증가되고 있다. FFT(Fast Fourier Transform)와 같은 대부분의 DSP 응용은 덧셈과 곱셈 연산을 포함하며, 많은 경우에 있어서 곱셈 수행 시간이 시스템의 성능을 결정하게 되므로 곱셈기의 성능 향상을 위한 많은 연구들이 활발히 진행되어 왔다^[1~11].

곱셈기의 종류를 사용되는 알고리즘에 따라 분류하면 Booth^[12], modified Booth, Braun, Baugh-Wooley 곱셈기 등이 있으며, 회로 구조에 따라 분류하면 어레이, 병렬, 파이프라인 곱셈기 등이 있다. 이러한 곱셈기들 중에서 부분 곱의 수를 절반으로 줄이는 modified Booth 알고리즘을 이용한 곱셈기가 가장 빠르다고 알려져 있으며, 회로의 지연 시간을 줄이는 파이프라인

* 학생회원, ** 평생회원, 한국의국외대학교 전자공학과 (Hankuk University of Foreign Studies)

※ 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업의 일환으로 수행하였음. [2009-F-010-01]

접수일자: 2009년9월7일, 수정완료일: 2009년11월5일

구조가 곱셈기의 성능을 향상시키기 위해 가장 널리 사용되고 있다.

본 논문은 modified Booth 알고리즘을 이용하고 파이프라인의 단 수를 크게 한 고성능 곱셈기 회로 구조를 제안하고 있다. 여러 실험을 통해 곱셈기 회로 내 파이프라인 레지스터의 위치와 파이프라인 단 수를 조절하여 고성능 곱셈기를 설계하였으며, 설계된 곱셈기 회로는 광통신 시스템과 같은 빠른 연산이 필요한 응용 시스템에서 사용될 수 있다.

II. 일반적인 modified Booth 곱셈기

1. Modified Booth 알고리즘

곱셈 과정은 기능에 따라 크게 3개의 과정으로 나누어진다. 첫째, 부분 곱을 생성하는 과정, 둘째, 마지막 2줄이 남을 때까지 부분 곱을 더하는 과정, 셋째, 마지막 남은 2줄을 더하여 최종 결과 값을 출력하는 과정이다. Modified Booth 알고리즘은 이 중에서 첫 번째 과정에 해당하는 것으로서, 생성되는 부분 곱의 수를 절반으로 줄이는 역할을 한다. 본 논문에서는 Booth 인코더와 디코더 중 가장 효율적이라고 알려진 참고문헌 [1]의 MBE(Modified Booth Encoding) 방식을 이용하였다. Modified Booth 알고리즘을 이용하여 X*Y 연산을 하는 경우, 우선 Y를 3-비트씩 묶어서 인코딩을 하여 {-2, -1, 0, 1, 2}의 5가지 경우에 해당하는 인코딩된 신호를 생성한다. 표 1은 인코딩된 신호들을 생성하는 규칙을 나타내며 그림 1(a)는 이에 따른 블록도를 나타낸다. Booth 인코더는 이와 같이 인코딩된 신호들을 이용하여 그림 1(b)와 같은 부분 곱들을 생성한다.

그림 2는 8-비트 modified Booth 곱셈기의 부분 곱

표 1. MBE 방식의 진리표

Table 1. Truth table of MBE scheme.

Y_{i+1}	Y_i	Y_{i-1}	value	X1_b	X2_b	Neg	Z
0	0	0	0	1	0	0	1
0	0	1	1	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	2	1	0	0	0
1	0	0	-2	1	0	1	0
1	0	1	-1	0	1	1	0
1	1	0	-1	0	1	1	1
1	1	1	0	1	0	1	1

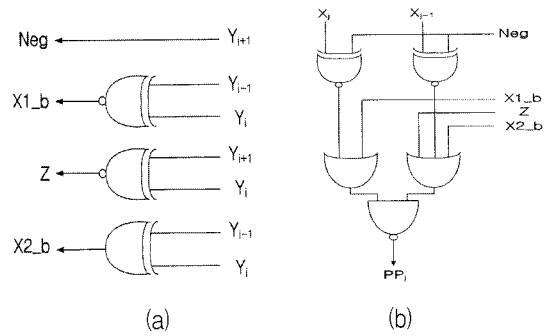


그림 1. MBE 방식의 인코더와 디코더
Fig. 1. Encoder and decoder for MBE scheme.

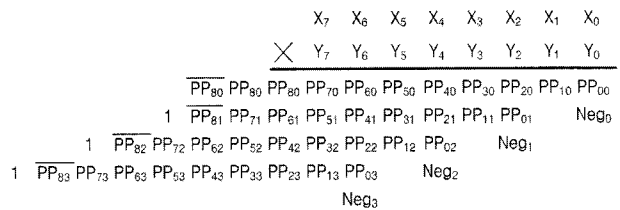


그림 2. 생성된 부분 곱과 부호 확장 방식
Fig. 2. Generated partial products and sign extension scheme.

과 부호 확장 방식^[13]을 나타낸다. Modified Booth 알고리즘을 사용하여 생성된 부분 곱들은 Wallace tree^[14]를 이용하여 마지막 2줄이 남을 때까지 병렬적으로 더해지고, 마지막 2줄은 carry propagation adder를 사용하여 더해짐으로써 곱셈기의 최종 값이 완성된다.

2. Modified Booth 곱셈기 구조

그림 3은 modified Booth 곱셈기의 구조를 나타내며, 입력 데이터인 X는 피승수(multiplicand)이고 Y는 승수(multiplier)이다. Booth 인코더는 승수 Y를 인코딩하여

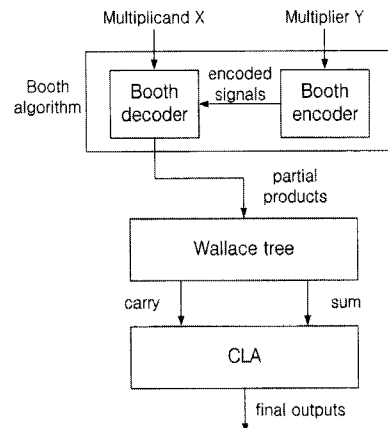


그림 3. Modified Booth 곱셈기 구조
Fig. 3. Architecture of modified Booth multiplier.

그림 1(a)와 같이 4가지의 인코딩된 신호들을 출력하고, Booth 디코더는 이 신호들을 이용하여 피승수 X의 입력에 따라 그림 1(b)와 같이 부분 곱을 생성한다. 생성된 부분 곱들은 carry save adder를 이용한 Wallace tree를 통해 마지막 carry와 sum의 2줄이 남을 때까지 더해진다. 최종적으로 남은 2줄은 carry look-ahead adder를 이용하여 더해져서 곱셈기의 최종 결과 값이 생성된다.

III. 제안하는 파이프라인 modified Booth 곱셈기

파이프라인 구조는 디지털 회로의 성능을 향상시키기 위하여 널리 사용되고 있으며, 파이프라인에 단이 많이 추가될수록 각 단의 지연 시간이 짧아지고 이에 따라 회로의 전체적인 성능이 향상된다. 본 논문에서는 최적의 파이프라인 단 수와 파이프라인 레지스터의 위치를 결정하기 위해 여러 가지 방법의 파이프라인 방식을 적용하여 찾아낸 고성능 modified Booth 곱셈기를 제안하고 있다.

1. 기본적인 3단 파이프라인 곱셈기

본 논문에서는 먼저 곱셈기의 각 기능별로 파이프라인의 단을 나누었다. 앞 절에서 언급하였듯이, 곱셈기는 크게 3가지 기능을 포함하기 때문에 modified Booth 곱셈기에 그림 4와 같이 3단 파이프라인을 적용하였다. 실험 결과를 통해 이 회로는 파이프라인을 적용하지 않은 곱셈기보다 최대 지연 시간을 절반으로 줄일 수 있

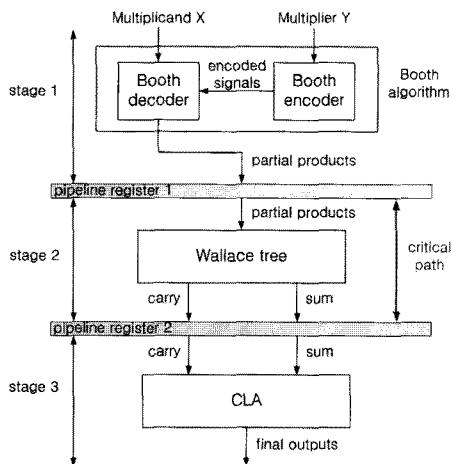


그림 4. 3단 파이프라인 modified Booth 곱셈기 구조
Fig. 4. Architecture of 3-stage pipelined modified Booth multiplier.

는 것을 확인하였다. 3단 파이프라인 곱셈기의 경우, 많은 덧셈 연산을 필요로 하는 Wallace tree 부분이 가장 지연 시간이 긴 경로가 되므로, Wallace tree 내에도 파이프라인을 추가로 적용시키면 회로의 지연 시간을 더 줄일 수 있다.

2. 파이프라인을 적용한 Wallace tree

N-비트 modified Booth 곱셈기의 경우, 생성되는 부분 곱의 수는 N/2개 이므로, 8-비트 modified Booth 곱셈기의 경우 생성되는 부분 곱의 수는 4개 이다. 본 논문에서는 2의 보수를 계산하는 회로를 따로 설계하지 않고, 그림 2와 같이 2의 보수 기능을 구현하기 위해 필요한 'Neg₀' ~ 'Neg₃' 신호를 사용하였다. 따라서 총 부분 곱의 줄 수는 마지막 'Neg₃' 신호를 포함하여 5개가 된다. Wallace tree는 3줄을 더하여 2줄을 생성하는데, 생성된 2줄은 각각 carry와 sum을 나타낸다.

생성된 부분 곱들을 더하는 Wallace tree는 carry save adder를 사용하기 때문에 일반적인 ripple carry adder보다 지연 시간이 작다는 장점이 있다. 하지만 Wallace tree는 많은 연산량으로 인해 여전히 곱셈기 전체의 연산 시간을 대부분 차지하는 부분이다. 따라서 Wallace tree 내에도 파이프라인 구조를 적용하면 곱셈기 전체의 지연 시간을 더욱 효과적으로 줄일 수 있다. 그림 5는 8-비트 modified Booth 곱셈기의 부분 곱들을 모두 더하는 Wallace tree 내의 각 단계마다 파이프라인을 적용한 예를 나타내고 있다. 8-비트 곱셈의 경우 Wallace tree 과정이 3단계로 구성되기 때문에 최대 3개의 단으로 나눌 수 있으며, 16-비트 곱셈의 경우는 7단계로 구성되기 때문에 최대 7개의 단까지 나눌 수 있다.

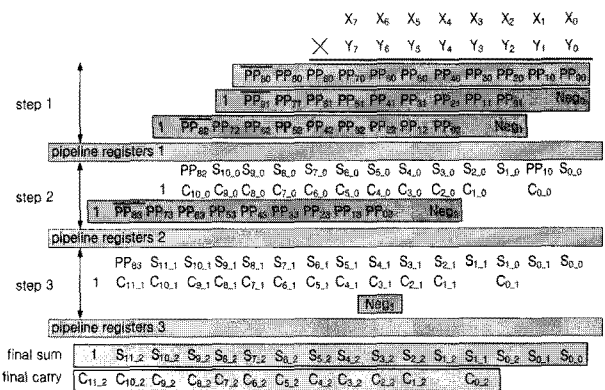


그림 5. 3단 파이프라인 Wallace tree 구조
Fig. 5. Architecture of 3-stage pipelined Wallace tree.

3. 파이프라인을 적용한 전가산기

기본적인 3단 파이프라인 곱셈기는 파이프라인을 적용하지 않은 곱셈기보다 회로의 전체 지연 시간이 절반으로 줄었다. Wallace tree의 각 단계마다 파이프라인 단을 추가로 적용하면 곱셈기 회로 전체의 지연 시간을 절반 가까이 더 줄일 수 있다. 이때 지연 시간이 가장 긴 경로는 Wallace tree의 각 단계를 구성하는 전가산기가 된다. 따라서 전가산기에도 파이프라인을 적용하면 곱셈기 전체의 지연 시간을 더 줄일 수 있다. 본 논문에서는 곱셈기 회로의 전체 지연 시간을 더 줄임으로써 한층 더 빠른 속도로 곱셈 연산을 할 수 있도록 하기 위하여 전가산기에도 파이프라인 구조를 적용하였다. 그림 6은 Wallace tree에서 사용되는 전가산기에 파이프라인을 적용한 구조를 나타낸다. 그림과 같은 2단 파이프라인 전가산기를 Wallace tree에서 사용함으로써 곱셈기의 전체적인 성능을 더욱 향상시킬 수 있었다.

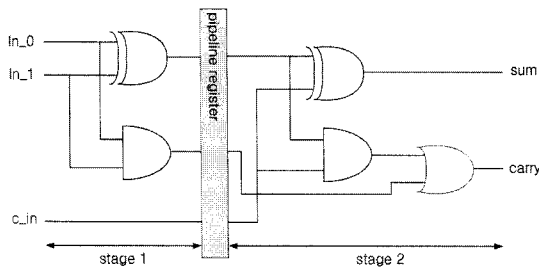


그림 6. 2단 파이프라인 전가산기 구조
Fig. 6. Architecture of 2-stage pipelined full adder.

IV. 실험 결과

본 논문에서는 modified Booth 곱셈기의 성능을 향상시키기 위하여 기본적인 8-비트와 16-비트 modified Booth 곱셈기 회로에 파이프라인 방식을 적용하였으며, 파이프라인 레지스터의 위치와 단 수를 조절하면서 여러 가지 경우에 대한 실험을 하였다. 이 절은 GHz에서 동작하는 고성능 modified Booth 곱셈기를 얻기 위한 방법을 기술하고 있다.

본 논문에서 제안하는 modified Booth 곱셈기 회로는 Verilog HDL(Hardware Description Language)을 사용하여 설계하였으며, Cadence 사의 NC-Verilog를 이용하여 동작을 검증하였다. 설계한 회로는 Synopsys 사의 Design Compiler와 0.13um 표준 셀 라이브러리 공정을 사용하여 합성하였다.

1. 8-비트 파이프라인 modified Booth 곱셈기

본 논문에서는 고성능 8-비트 modified Booth 곱셈기를 위하여 4가지 경우에 대한 실험을 진행하였다. Type A는 그림 3과 같이 파이프라인을 적용하지 않은 modified Booth 곱셈기이며, Type B는 그림 4와 같이 곱셈기의 각 주요 기능별로 파이프라인을 적용한 3단 파이프라인 modified Booth 곱셈기이다. Type C는 Type B에 그림 5와 같이 Wallace tree의 각 단계마다 파이프라인을 적용한 5단 파이프라인 곱셈기이고, Type D는 Type C에 그림 6과 같은 파이프라인 전가산기를 사용한 8단 파이프라인 곱셈기이다. 표 2는 본 논문에서 설계한 8-비트 modified Booth 곱셈기의 합성 결과를 나타내고 있다.

파이프라인의 단 수가 증가할수록 곱셈기 회로의 전체 지연 시간이 크게 감소하는 것을 표 2를 통해 확인할 수 있다. 실험 결과 8-비트 modified Booth 곱셈기의 지연 시간은 8개의 단으로 나뉜 Type D 구조가 0.34ns으로 가장 작고, 이 곱셈기는 파이프라인을 적용하지 않은 Type A보다 성능이 약 4.38배 향상되었으며 약 2.95GHz에서 동작할 수 있다.

표 2. 8-비트 modified Booth 곱셈기의 합성 결과
Table 2. Synthesis results of 8-bit modified Booth multiplier.

Type	단 수	게이트 개수	크기 증가율	지연시간 (ns)	속도 향상율
A	-	1,179	-	1.49	-
B	3	1,211	x1.03	0.82	x1.82
C	5	1,573	x1.33	0.47	x3.17
D	8	2,022	x1.71	0.34	x4.38

2. 16-비트 파이프라인 modified Booth 곱셈기

본 논문에서는 고성능 16-비트 modified Booth 곱셈기를 위하여 모두 14가지의 경우에 대한 실험을 진행하였다. Type A ~ Type D는 앞에서 설명한 8-비트 파이프라인 modified Booth 곱셈기와 같은 구조를 갖는다. Type C와 Type D에 대해서는 Wallace tree의 7개 단계에 대해 별도로 6가지 경우에 대한 추가적인 실험을 진행하였다. 표 3은 Type C와 Type D에서 나는 파이프라인 단의 위치와 수를 나타낸다. 'Step 12/34/567'은 step 2와 step 3, step 4와 step 5 사이에 파이프라인 레지스터를 삽입한 것을 의미한다. 표 4는 16-비트 파이프라인 modified Booth 곱셈기의 각각의 경우에 대해

표 3. Type C와 Type D 곱셈기에 대한 Wallace tree 내의 파이프라인 기법

Table 3. Pipeline schemes in Wallace tree for Type C and Type D multipliers.

경우	Wallace tree 내 파이프라인 위치	Wallace tree 내 파이프라인 단 수	
		Type C	Type D
1	Step 12/34/567	3	10
2	Step 12/345/67	3	10
3	Step 123/45/67	3	10
4	Step 1234/567	2	9
5	Step 123/4567	2	9
6	Step 1/2/3/4/5/6/7	7	14

표 4. 16-비트 modified Booth 곱셈기의 합성 결과

Table 4. Synthesis results of 16-bit modified Booth multiplier.

Type	경우	단 수	게이트 개수	크기 증가율	지연 시간 (ns)	속도 향상율
A	-	-	3,629	-	2.40	-
B	-	3	4,348	x1.20	1.79	x1.34
C	1	5	4,526	x1.25	0.82	x2.93
	2	5	4,528	x1.25	0.91	x2.64
	3	5	4,528	x1.25	0.91	x2.64
	4	4	4,452	x1.23	1.05	x2.29
	5	4	4,268	x1.17	1.10	x2.18
	6	9	6,069	x1.67	0.47	x5.11
D	1	12	6,834	x1.88	0.57	x4.21
	2	12	6,750	x1.86	0.61	x3.93
	3	12	6,754	x1.86	0.60	x4.00
	4	11	6,400	x1.76	0.73	x3.29
	5	11	6,419	x1.77	0.74	x3.24
	6	16	8,246	x2.27	0.38	x6.32

합성한 결과를 나타낸다.

표 4를 통해 16-비트 파이프라인 modified Booth 곱셈기도 8-비트 곱셈기와 마찬가지로 파이프라인의 단 수가 늘어날수록 회로의 전체 지연 시간이 작아지는 것을 확인할 수 있다. 회로의 속도 향상율은 회로의 크기 증가율보다 크며, 이러한 특징은 파이프라인의 단이 증가할수록 명백하게 나타난다. 어떤 경우에 있어서는 파이프라인의 단 수가 같더라도 레지스터의 위치에 따라 회로의 크기와 지연 시간이 달라진다. 실험 결과 16-비트 파이프라인 modified Booth 곱셈기의 지연 시간은 파이프라인의 단 수가 가장 많은 Type D 구조의 6번째 경우가 0.38ns으로 가장 작으며, 이 곱셈기는 파이프라인

표 5. 다른 곱셈기들과의 성능 비교

Table 5. Comparison to other multipliers.

	공정	입력 비트	최대 동작 주파수	파이프 라인	배선지연시간 고려
[5]	0.18um	8x8	1.1GHz	-	-
[6]	0.09um	12x9	2GHz	-	-
[7]	0.35um	8x8	1GHz	○	-
[8]	0.6um	8x8	500MHz	○	○
[9]	0.35um	8x8	1.2GHz	○	○
[10]	0.13um	8x8	320MHz	○	○
[11]	0.18um	8x8	2.86GHz	○	○
본 논문	0.13um	8x8	2.95GHz	○	-
		16x16	2.63GHz	○	-

인을 적용하지 않은 Type A보다 성능이 약 6.32 배 향상되었으며, 약 2.63GHz에서 동작할 수 있다.

표 5는 본 논문에서 제안하는 곱셈기와 다른 논문들에서 제안하는 곱셈기들과의 성능을 비교한 것이다. 8-비트 곱셈기들 중에서 본 논문에서 제안하는 곱셈기가 가장 좋은 성능을 갖는 것을 확인할 수 있다. 트랜지스터 수준에서 설계한 참고문헌 [11]의 곱셈기 또한 좋은 성능을 나타내지만, 본 논문에서 제안하는 곱셈기 회로 역시 트랜지스터 수준에서 최적화한다면 더 좋은 성능을 나타낼 것이다. 참고문헌 [6]의 곱셈기는 입력 비트의 수가 더 작고 더 높은 공정을 사용함에도 불구하고 본 논문에서 제안하는 16-비트 곱셈기보다 속도가 더 느리다. 또한 본 논문에서 제안하는 16-비트 곱셈기는 8-비트 곱셈기에 근접하는 성능을 갖는다는 것을 확인할 수 있다.

V. 결론

파이프라인 구조는 디지털 회로의 성능을 향상시키기 위해 가장 널리 사용되는 방법이며, 본 논문에서는 파이프라인의 단 수를 많게 한 고성능 파이프라인 modified Booth 곱셈기를 제안하고 있다. 제안하는 곱셈기는 곱셈의 주요 기능에 따라 3부분으로 나누어진다. 첫째, 부분 곱의 수를 N/2 개만 생성하는 modified Booth 인코더와 디코더, 둘째, 생성된 부분 곱을 더하는 Wallace tree, 셋째, 마지막 덧셈을 수행하는 carry look-ahead adder 이다.

본 논문에서는 파이프라인의 단 수와 파이프라인 레

지스터의 위치를 변화시키면서 여러 가지 종류의 곱셈기를 설계하였으며, 합성된 결과를 비교하여 가장 빠른 8-비트와 16-비트 modified Booth 곱셈기를 얻을 수 있었다. 8-비트 곱셈기의 경우 파이프라인을 적용하지 않은 곱셈기보다 4.38배의 성능 향상이 있었으며, 16-비트 곱셈기의 경우 6.32배의 성능 향상이 있었다. 본 논문에서는 제안하는 8-비트와 16-비트 파이프라인 modified Booth 곱셈기는 각각 2.95GHz와 2.63GHz에서 동작하므로, 빠른 곱셈 연산을 필요로 하는 광통신 시스템과 같은 응용 시스템에 적용될 수 있다.

참 고 문 헌

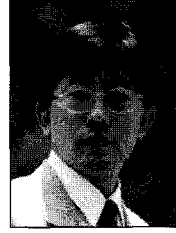
- [1] Wen-Chang Yeh and Chein-Wei Jen, "High-speed Booth Encoded Parallel Multiplier Design," IEEE Trans. on Computers, vol. 49, issue 7, pp. 692-701, July 2000.
- [2] Jung-Yup Kang and Jean-Luc Gaudiot, "A Simple High-speed Multiplier Design," IEEE Trans. on Computers, vol. 55, issue 10, pp. 1253-1258, Oct. 2006.
- [3] Shiann-Rong Kuang, Jiun-Ping Wang and Cang-Yuan Guo, "Modified Booth Multipliers with A Regular Partial Product Array," IEEE Trans. on Circuit and Systems, vol.56, Issue 5, pp. 404-408, May 2009.
- [4] Li-rong Wang, Shyh-Jye Jou and Chung-Len Lee, "A Well-structured Modified Booth Multiplier Design," IEEE International Symposium on VLSI Design, Automation and Test, pp. 85-88, April 2008.
- [5] A. A. Khatibzadeh, K. Raahemifar and M. Ahmadi, "A 1.8V 1.1GHz Novel Digital Multiplier," Canadian Conference on Electrical and Computer Engineering, pp. 686-689, May 2005.
- [6] S. Hus, V. Venkatraman, S. Mathew, H. Kaul, M. Anders, S. Dighe, W. Burleson and R. Krishnamurthy, "A 2GHZ 13.6mW 12x9b Mutiplier for Energy Efficient FFT Accelerators," ESSCIRC Proc. of Solid-state Circuits, pp. 199-202, Sept. 2005.
- [7] Hwang-Cherng Chow and I-Chyn Wey, "A 3.3V 1GHz High Speed Pipelined Booth Multiplier," IEEE ISCAS Symposium on Circuits and Systems, vol. 1, pp. 457-460, May 2002.
- [8] A. Asati and Chandrashekhar, "An Improved High Speed Fully Piplined 500MHz 8x8 Baugh Wooley Multiplier Design Using 0.6um CMOS TSPC Logic Design Style," IEEE Third International Conference on Industrial and Information Systems, pp. 1-6, Dec. 2008.
- [9] M. Aguirre-Hernandez and M. Linarse-Aranda, "Energy-efficient High-speed CMOS Pipelined Multiplier," 5th International Conference on Electrical Engineering, Computing Science and Automatic Control, pp. 460-464, Nov. 2008.
- [10] Yung-chin Liang, Ching-ji Huang and Wei-bin Yang, "A 320-MHz 8bit x 8bit Pipelined Multiplier in Ultra-low Supply Voltage," IEEE Asian Solid-state Circuits, pp. 73-76, Nov. 2008.
- [11] S. B. Tatapudi and J. G. Delgado-Frias, "Designing Pipelined Systems with a Clock Period Approaching Pipline Register Delay," 48th Midwest Symposium on Circuits and Systems, vol. 1, pp. 871-874, Aug. 2005.
- [12] A. D. Booth, "A Signed Binary Multiplication Technique," Quarterly J. Mechanical and Applied Math, vol. 4, pp.236-240, 1951.
- [13] M. D. Ercegovac and T. Lang, *Digital Arithmetic*, Morgan Kaufmann Publishersm Los Altos, CA 94022, USA, 2003.
- [14] C. S. Wallace, "A Suggestion for a Fast Multiplier," IEEE Trans. on Computers, vol. BC13, pp. 14-17, Feb. 1964.

 저 자 소 개



김 수 진(학생회원)
 2007년 2월 한국외국어대학교
 전자공학과 학사 졸업.
 2009년 2월 한국외국어대학교
 전자공학과 석사 졸업.
 2009년 2월~현재 한국외국어
 대학교 전자공학과
 박사과정.

<주관심분야 : SoC 설계>



조 경 순(평생회원)
 1982년 2월 서울대학교
 전자공학과 학사 졸업.
 1984년 2월 서울대학교
 전자공학과 석사 졸업.
 1988년 12월 미국 Carnegie
 Mellon University 전기
 및 컴퓨터 공학과 박사
 졸업.

1988년11월~1994년 8월 삼성전자(주) 반도체
 총괄 선임, 수석 연구원.

1994년 8월~현재 한국외국어대학교 전자공학과
 조교수, 부교수, 정교수.

<주관심분야 : SoC 설계>