

고속 저전력 곱셈기에 적합한 ENMODL CLA 설계

Design of ENMODL CLA for Low Power High Speed Multiplier

백한석, 한석봉

H. S. Baek, S. B. Han,

요약

본 논문에서는 고속 저전력 곱셈기에 적합한 CPA(Carry Propagation Adder)부분의 ENMODL (Enhanced NORA MODL) 설계방식을 제안한다.

ENMODL 설계방식은 반복성이 많은 CLA(Carry Look-ahead Adder) 가산기와 같은 회로에서 면적을 줄일 수 있고 동작 속도를 빠르게 할 수 있다. 기존의 NMODL CLA와 비교하여 6.27% 동작속도가 빠르다.

따라서 본 논문에서는 저전력 고속 곱셈기에 적합한 CPA 부분을 ENMODL CLA 가산기로 설계했고 현대 0.6 μm 2-poly 3-metal 공정파라미터를 이용하여 HSPICE로 시뮬레이션 하여 회로의 성능을 확인하였다.

또한, CADENCE tool을 이용하여 16비트 곱셈기에 적합한 ENMODL CLA를 레이아웃 하였다.

Abstract

In this paper, we propose a new ENMODL(Enhanced-NORA-MODL) CLA(Carry Look-ahead Adder) for high speed and low power multiplier. To reduce transistor counts, area, and power dissipation we developed new-approaches. The method makes use of a dynamic CMOS logic ENMODL CLA. The advantage of ENMODL is small area and high speed. The speed of ENMODL CLA is increased by 6.27% as compared with conventional NMODL CLA.

The proposed method was verified by HSPICE simulation and layout through 0.6 μm CMOS process.

Key words : ENMODL, multiplier, CLA, dynamic CMOS logic

I 서론

현재 컴퓨터 아키텍처 및 CMOS VLSI 설계기술의 발전으로 집적도가 높고 고속 연산 기능을 수행할 수 있는 고성능 마이크로 프로세서의 개발이 가속화되고 있다.

특히 마이크로 프로세서나 영상신호처리용에 이용되는 곱셈기는 중요한 연산회로로서 어떤 알고리즘 및 캐리 생성 방식을 사용하느냐에 따라 그 성능이 결정된다. 현재 널리 사용되고 있는 곱셈기의 구조는 Booth 인코더(encoder)와 부분곱 생성부분인 Booth 디코더(decoder), 부분곱 압축블록, 최종합을 계산하는 CPA(Carry Propagation Adder) 부분으로 구성되어 있고 많이 사용되는 Booth 알고리즘 방식은 개선된 Booth 알고리즘

(Modified Booth Algorithm)으로써 Booth 인코더와 디코더 부분으로 구성되어 있다[1,2,3].

본 논문에서 사용한 Booth 알고리즘은 기존 방식들 중 성능을 비교하여 가장 우수한 알고리즘을 사용하였다.

Booth 알고리즘에서 생성된 부분곱을 압축하는 부분곱 압축 블록과 CPA는 곱셈기의 연산에서 큰 비중을 차지하고 있다. 본 논문에서 사용한 부분곱 압축 블록은 기존 방식 중 속도 및 면적에서 우수한 4-2 압축블록을 사용하였다[1].

곱셈기의 최종합을 생성하는 CPA부분은 CLA가 많이 사용되고 있으며 현재 빠른 동작 속도와 적은 면적이 요구되고 있다.

본 논문에서는 곱셈기의 최종합을 계산하는 CPA부분

에 다이내믹 CMOS회로인 ENMODL CLA를 사용하여 기존의 정적 CMOS에 비해 칩 면적이 작고, 고속으로 논리를 구현하였다.[5]

그리고 설계한 ENMODL CLA를 현대 0.6 μ m 2-poly 3-metal 공정파라미터를 이용하여 HSPICE 시뮬레이션하여 회로의 성능을 확인하였고, 또한, CADENCE tool을 이용하여 ENMODL CLA를 가지는 16비트 곱셈기를 레이어아웃 하였다.

본 논문의 구성은 서론에 이어 II장에서는 기본 곱셈기의 구조인 Booth 알고리즘, 부분곱 압축블록, CPA에 대하여 설명하고, III장에서는 본 논문에서 제안한 ENMODL의 구조와 동작을 설명하였으며, IV에서는 ENMODL을 이용하여 CLA 설계에 대하여 설명하고, V에서는 기존의 방식과 제안한 방식의 모의실험 결과를 평가하였고 마지막으로 VI에서 결론을 맺는다.

II. 곱셈기의 구조

그림 1은 곱셈기의 기본 블록 다이어그램으로 크게 승수와 피승수를 이용하여 부분곱을 만드는 Booth 알고리즘 부분과, 부분곱을 압축하는 부분, 마지막으로 최종합을 출력하는 CPA부분으로 이루어져 있다[1,2,3].

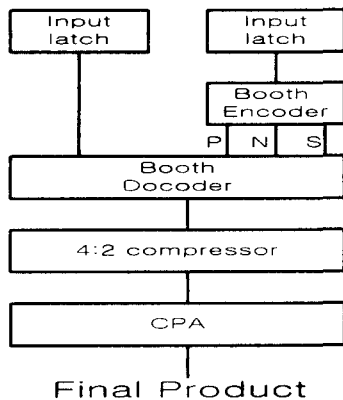


그림 1. 곱셈기의 구조
Fig. 1. Structure of multiplier

먼저 Booth는 2의 곱셈을 효율적으로 수행하기 위한 알고리즘을 제안하였다. 제안된 알고리즘은 승수에서 나타나는 연속된 k개의 '1'을 변환을 통하여 2k+1-1로 바꾸어 필요한 연산을 줄이는 것이다.

Booth 알고리즘은 일반적으로 덧셈 또는 뺄셈 연산을 결정하는 Booth 인코더와 여기서 결정된 신호로부터 계산에 사용할 값을 만들어 내는 Booth 디코더로 구성된다 [2]. 현재 많이 사용되는 Booth 알고리즘은 개선된 Booth 알고리즘이며[1,2], 표 1은 기존의 Booth 알고리즘을 비교 분석한 것으로 참고문헌 3의 Booth 알고리즘의 성능이 가장 우수함을 알 수 있다. 따라서 본 논문의 곱셈기에는

비교 논문 중 성능이 가장 우수한 참고 문헌 3의 Booth 알고리즘을 사용하였다.

표 1. Booth 알고리즘 비교
Table 1. Comparison of Booth algorithm

	Number of TRs	소비전력 (pw)	동작속도 (nS)
참고문헌 1	58	1.26897	5.82e-10
참고문헌 2	50	1.0728	4.33e-10
참고문헌 3	46	1.32287	3.23e-10

큰 수의 곱셈에 있어서 부분곱의 개수는 곱셈기에서 큰 부분을 차지하게 된다. 따라서 부분곱 연산 속도가 곱셈기의 성능에 영향을 주게 되므로, 현재 많이 사용하는 방법이 부분곱 압축블록이다[2].

대표적인 부분곱 압축블록은 4-2 압축블록, 7-2 압축블록, 9-3 압축블록이 있다[4].

본 논문에서는 이들 중 면적과 속도면에서 우수한 4-2 압축블록을 사용하였다.

CPA는 곱셈기의 최종합을 출력하는 부분으로써 기본 회로는 가산기로 구성되어 있다. CPA의 종류는 CSA(Carry Select Adder), Ripple Carry Adder, CLA 등이 있다. Ripple Carry Adder는 간단한 전가산기로 이루어져 있으나 캐리전파 시간이 늦은 단점이 있다. 또한 CLA를 사용하면 캐리 전파시간이 빠르나 회로가 커지는 단점을 가지고 있다. 따라서 회로의 면적이 작으면서 속도가 빠른 연산회로를 만드는 방법이 요구된다. 회로가 작으면서 빠른 연산이 가능한 회로는 다이내믹 로직이 있다.

Static CMOS는 부하 회로(load circuit) pMOS로 구성된 p 블록과 구동회로(driver circuit) nMOS로 구성된 n 블록이 서로 상보적(complementary)으로 연결되어 이루어진다. p 블록은 Vdd에 n 블록은 Vss에 각각 연결되어 주어진 입력신호의 상태에 따라 p 블록이나 n 블록 중 어느 한 블록에만 경로가 활성화되어 출력 값을 결정하게 된다. static CMOS는 p 블록과 n 블록으로 구성되어 면적을 많이 차지하며, 입력용량(capacitance) 증가 등의 이유로 속도가 느리다. 이에 비해 다이내믹 CMOS 회로는 기존의 CMOS(static CMOS) 회로로부터 소비전력을 줄이고 회로면적을 줄이면서도 빠른 동작속도를 갖기 위하여 변형한 형태이다. 즉 static CMOS 회로에서 p 블록 부분을 제거시키고, 각각 한 개씩의 pMOS와 nMOS를 부가한 클럭킹 게이트와 회로의 출력 값을 결정하는 합수블록으로 구성한 것이다.

일반적으로 다이내믹 CMOS회로는 Static CMOS 회로와 비교하여 칩면적의 감소, 동작 속도의 개선 등의 장점을 가지고 있다. 그러나 다이내믹 CMOS회로를 다단으

로 연결 할 때 시차 문제. 전하 재분배등의 큰 단점을 가지고 있다. 위의 단점을 보완하기 위하여 Domino, NORA, Zipper CMOS 등의 설계방식이 등장하였다. 다음 장에서 본 논문에서 제안한 ENMODL 구조에 대하여 기술하였다.

본 논문에서는 16×16곱셈기에 적합한 다이내믹 로직 ENMODL CLA를 설계하였다.

III. 다이내믹 로직 및 ENMODL 구조

식 (1)과 (2)와 같은 함수를 설계할 때, MODL 회로를 이용하면 그림2와 같이 1 개의 논리 게이트로 2 개의 함수를 동시에 설계할 수 있으므로 부함수를 별도로 설계할 필요가 없어 면적을 감소시킬 수 있다.

$$F1 = A + BC \quad (1)$$

$$F2 = D \cdot F1 = D \cdot (A + BC) \quad (2)$$

이 경우에 회로가 동적 CMOS로 구성되므로 트랜지스터의 크기가 커져 트랜지스터 수의 감소에 의한 면적을 줄일 수 없으나, 게이트 수의 감소로 인한 금속 도선 등의 감소로 인하여 전체 면적이 줄어들 수 있고, 지연시간을 줄일 수 있다.

이 회로의 설계방식은 캐리 생성과 같은 반복적인 구조를 갖는 논리회로에 대해 적합한데, CLA 가산기와 같은 회로는 반복성이 매우 크므로 MODL 회로를 이용하여 회로를 설계하면 Domino 논리 회로를 사용하여 설계한 것보다 트랜지스터 수를 많이 감소시킬 수 있다

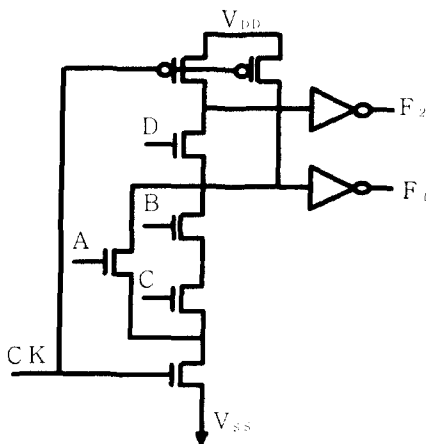


그림 2. MODL 회로
Fig. 2. Circuit of MODL

MODL은 그림 3(a)에서와 같이 2 단으로 구성되는 회로에서도 같은 논리 구조가 반복된다는 것을 알 수 있다. 그러나 NORA CMOS에서와 같이 MODL 회로의 첫 단 출력의 인버터를 제거하고, 다음 단의 N 블록을 P 블록

으로 바꾸면 다음 단의 인버터가 제거되어 면적을 줄일 수 있고, 2 개의 게이트 지연시간을 줄일 수 있는데 이러한 구조를 NMODL(NORA-MODL)이라 한다.

기본적인 동적 CMOS 게이트는 CK가 0일 때 (precharge phase) CK에 연결된 PMOS를 통하여 프리차지시키고, CK가 1일 때 (evaluation phase) CK에 연결된 NMOS를 통하여 디스차지시킨다. 그러나 NMODL 회로에서는 두 번째 단에 사용되는 P 블록의 모든 입력이 앞 단 NMODL 게이트의 출력으로부터만 들어오는 경우 프리차지 시간 동안에 입력이 모두 HIGH가 되므로 클럭킹 소자의 PMOS는 회로의 동작에 아무런 영향을 미치지 못한다. 따라서 CK에 연결되어 있는 클럭킹 소자의 PMOS를 그림 3(b)와 같이 제거할 수 있는데 이러한 구조를 ENMODL이라 한다.

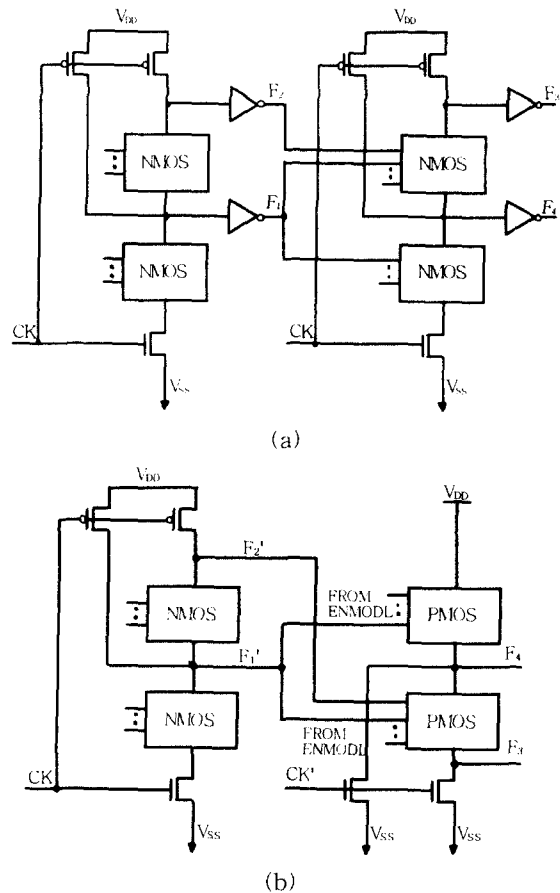


그림 3. MODL, ENMODL 회로
(a) 다단 MODL 회로 (b) ENMODL 회로
Fig. 3. Circuit of MODL and ENMODL
(a) Circuit of multistage MODL
(b) Circuit of ENMODL

ENMODL 회로는 함수블록 내부 노드에 다출력을 위한 클럭킹 프리차지 소자를 사용하여 내부 회로간에 발생하는 누설전류나 전하재분배 현상 및 전하결합 등의

문제를 해결함으로써 출력 전압의 변동이 제거되고 잡음 여유가 증가되어 안정된 회로로 동작할 수 있다. 여러 단으로 구성된 ENMODL 회로에서는 각 단의 함수 블록들이 홀짝에 따라 NMOS와 PMOS 블록이 번갈아 사용되므로 프리차지 시간 동안에 홀수 단의 클럭킹 소자의 PMOS가 ON 되어 출력을 1로 프리차지시키고, 짝수 단의 클럭킹 소자의 NMOS가 ON 되어 P 블록의 출력은 0으로 프리차지되어 신호 지연에 의한 시차 문제를 해결할 수 있다. 또한 PMOS 블록에서 디스차지 소자인 PMOS를 제거하여도 그 블록의 입력이 인버터를 제거한 다이내믹 CMOS의 NMOS 블록으로부터만 들어온다면 프리차지 기간에는 각 트랜지스터가 OFF 상태를 유지하므로 입력 신호의 시차문제에 영향을 받지 않고 안정적인 동작을 할 수 있다.

IV. 16×16 곱셈기용 ENMODL CLA 설계

가산기에서 가산이 수행되는 시간은 모든 전가산기를 통하여 캐리의 전파에 요구되는 시간에 의해 결정된다. CLA 가산기는 캐리를 미리 생성하여 전파지연을 제거함으로써 속도를 증가시키는 가산기이다. 캐리생성항을 G_i , 캐리전파항을 P_i , i 단의 입력 캐리를 C_i , 출력 캐리를 C_{i+1} 라 하면 다음 식으로 표시된다.

$$G_i = A_i \cdot B_i \quad (1)$$

$$P_i = A_i \cdot B_i' + A_i' \cdot B_i \quad (2)$$

$$C_{i+1} = G_i + P_i \cdot C_i \quad (3)$$

i 가 커지게 되면 캐리발생회로가 커지게 되므로 분할하여 그룹캐리 생성항과 그룹캐리 전파항을 설계하여야 한다. 따라서 8비트로 분할하여 이에 대한 그룹캐리 생성항($EGG_{i,7}$)과 그룹캐리 전파항($EGP_{i,7}$)을 부울 함수로 표현하면 식 (5), (6)과 같다.

$$GP_{i+3} = EP_{i+3}' + EP_{i+1}' \quad (4)$$

$$GG_{i+3} = EG_{i+3}' \cdot (EP_{i+3}' + EG_{i+1}') \quad (5)$$

$$EGP_{i+7} = EP_{i+7}' + EP_{i+5}' + GP_{i+3} \quad (6)$$

$$EGG_{i+7} = EG_{i+7}' \cdot (EP_{i+7}' + EG_{i+5}' \cdot (EP_{i+5}' + GG_{i+3})) \quad (7)$$

본 논문에서 설계한 CLA 가산기는 캐리 생성 블록과 합(SUM)을 구하는 블록으로 구성된다. 합을 구하는 블록은 기존의 CMOS를 사용하여 XOR 게이트로 설계하였으며, 캐리 생성 블록은 다중 함수를 출력하는 ENMODL 회로를 사용하여 설계하였다. XOR 게이트는 그림 4에 나타내었다.

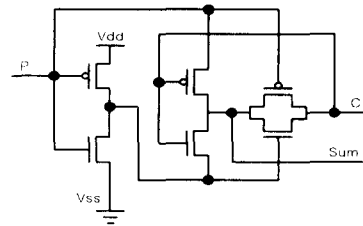


그림 4. SUM 회로
Fig. 4. Circuit of SUM

그림 5는 26 비트 CLA 가산기의 기본 블록도를 나타낸 것으로 본 논문에서 제안하는 26비트 CLA는 3개의 8비트 가산기와 나머지 2비트 처리부분으로 구성된다. 계층(level) 1에는 입력 A, B를 받아 캐리 생성항과 캐리 전파항을 발생시키고, 계층 3으로부터 받아들이는 캐리를 이용하여 합을 발생시킨다. 계층 2에서는 계층 1로부터 얻어진 캐리 생성항과 전파항을 이용하여 8 비트 그룹캐리 생성항과 전파항을 만들어 낸다. 이 계층은 P 블록으로 이루어지므로 clk'가 사용된다. 계층 3은 캐리를 발생하는 부분으로 계층 1로부터 들어오는 입력을 이용하여 캐리(C_0-C_{26})들을 발생하는 부분과 계층 2로부터 얻은 그룹캐리 생성항과 전파항을 이용하여 그룹캐리 C_8, C_{16}, C_{24} 를 만드는 부분으로 이루어진다.

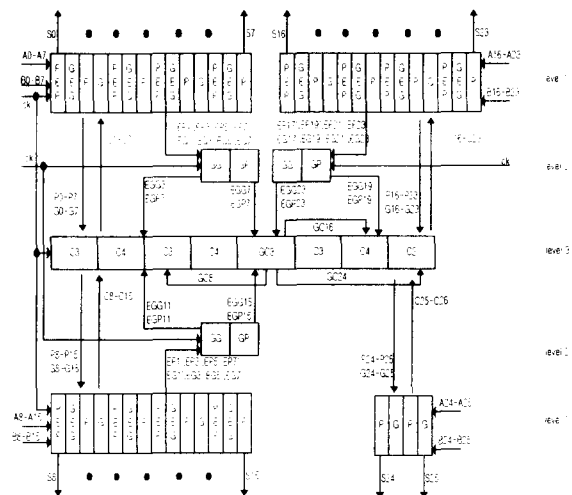


그림 5. 26비트 ENMODL CLA 블록도
Fig. 5. Block diagram of 26bit ENMODL CLA

계층2의 그룹캐리 생성항과 전파항은 본 논문에서 제안하는 ENMODL 구조를 이용하여 기존의 NMODL 보다 빠른 동작 및 면적을 줄이는 효과를 가져왔다. 아래 그림 6은 본 논문에서 제안한 ENMODL구조로 설계된 그룹캐리 생성항과 전파항으로써 ENMODL의 특징인 프리차지 소자 pMOS가 제거된 것을 확인할 수 있다.

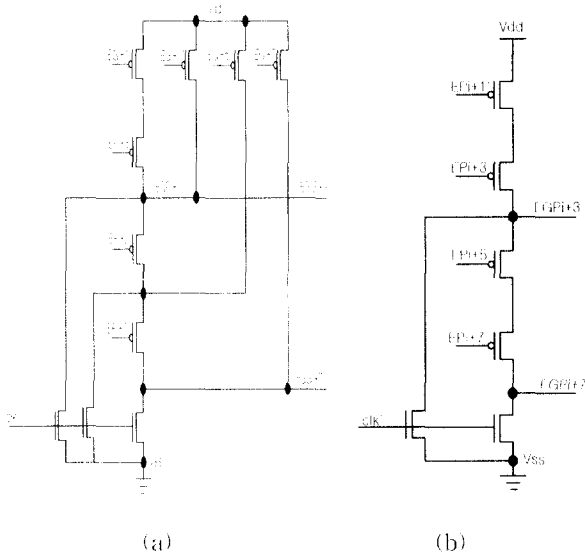


그림 6. ENMODL 그룹캐리 생성, 발생 회로
(a) 그룹캐리 생성회로 (b) 그룹캐리 전파회로

Fig. 6. Group carry generate and propagate circuit of ENMODL

(a) Group carry generate
(b) Group carry propagation

V. 모의실험 및 레이아웃

1. 모의실험

본 논문에서 제안한 방식을 확인하기 위하여 현대 0.6 μm 2 poly, 3 metal 공정 파라미터를 이용하여 CADENCE 로 레이아웃한 후 netlist를 추출하여 HSPICE로 모의실험 하였다.

그림 7은 NMODL 방식과 본 논문에서 제안한 ENMODL 방식의 CLA 동작 속도를 비교한 것이다.

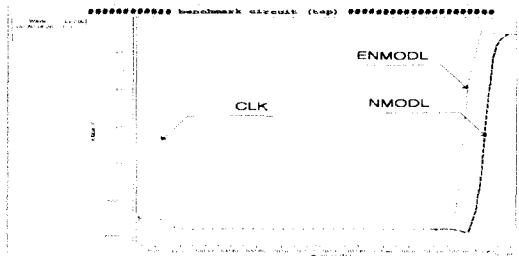


그림 7. 최종합 모의실험

Fig. 7. Simulation of final adder

표2는 본 논문에서 제안하는 방식과 기존방식의 트랜지스터 개수와 동작속도를 비교한 것이다.

표 2. 최종합 생성 블록 모의실험

Table 2. Simulation of final adder

	Number of TRs	C ₃₀ (nS)	Power consumption
NMODL	996	3.19	26.25nW
ENMODL	982	2.39	25nW

표2에서 ENMODL 방식은 NMODL 보다 적은 개수의 트랜지스터로 구성 되어있다. ENMODL 회로의 특징인 프리차지 소자 pMOS의 감소로 인하여 면적 및 속도에 유리한 영향을 준다. 따라서 연산속도는 NMODL 보다 25% 성능 향상을 보여준다.

2. 레이아웃

본 논문에서 제안한 방식의 검증을 위하여 현대 0.6μm 2 poly, 3 metal 공정을 이용하여 CADENCE로 레이아웃 하였다.

그림 7은 본 논문에서 제안한 ENMODL 구조를 가지는 그룹 캐리 생성회로와 그룹캐리 전파회로의 레이아웃이다. ENMODL CMOS 구조의 속도를 최적화 하기 위하여 MOS의 크기를 각각 다르게 하여 레이아웃 하였다.

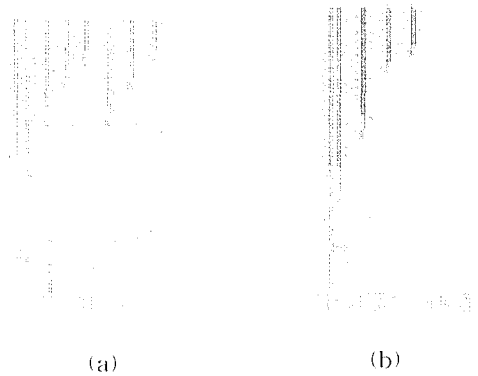


그림 8. ENMODL 그룹캐리 생성, 발생 회로 layout

(a) 그룹캐리 생성회로 (b) 그룹캐리 전파회로

Fig. 8. Group carry generate and propagate layout of ENMODL

(a) Group carry generate
(b) Group carry propagation

그림 9는 ENMODL CLA의 전체 레이아웃으로 3개의 8비트 블록과 2개의 sum 블록 그리고 1개의 2비트 CLA와 sum 블록으로 구성되어 있다.

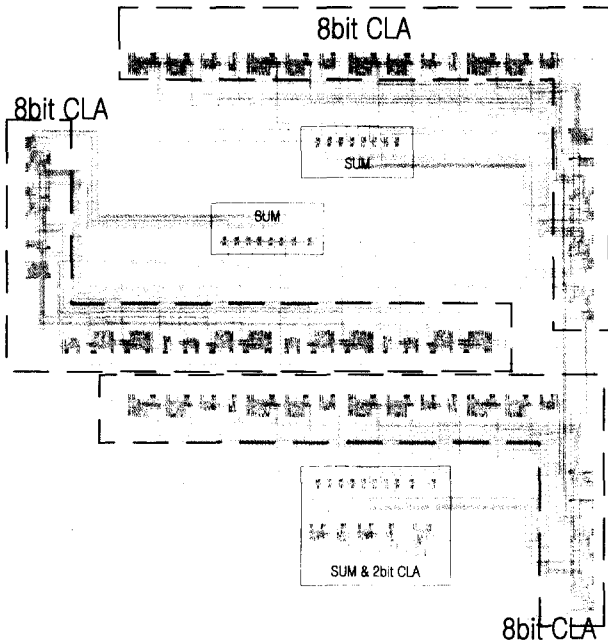


그림 9. 26비트 ENMODL CLA 레이아웃
Fig. 9. Layout of 26 bit ENMODL CLA

VI. 결론

본 논문에서는 고속 저전력 곱셈기에 적합한 CPA부분을 다이내믹 로직인 ENMODL CLA로 설계하였다.

제안한 방식을 검증하기 위하여 현대 0.6 μ m 2-poly, 3-metal 공정 파라미터를 사용하여 모의실험 하였다.

모의실험 결과 제안한 ENMODL CLA 성능이 기존의 NMODL 방식보다 연산속도가 약 25% 우수함을 알았다.

모의실험 결과의 검증은 위하여 같은 공정을 사용하여 16 \times 16비트 곱셈기에 적합한 ENMODL CLA를 레이아웃 하였다.

본 논문에서 제안하는 ENMODL CLA를 가지는 곱셈기 회로는 고성능 연산기능을 요구하는 컴퓨터의 연산장치, 디지털 신호처리 시스템 및 특수 목적용 칩 등에서 뛰어난 성능을 제공할 수 있을 것으로 사료된다.

접수일자 : 2001. 8. 10 수정완료 : 2001. 10. 24

본 연구는 반도체설계교육센터(IDECE)의 지원을 받아 수행되었습니다.

참고문헌

[1] Norio Ohkubo, Makoto Suzuki, "A 4.4 ns CMOS 54 \times 54 b Multiplier Using Pass Transistor Multiplexer" IEEE Journal of Solid State Circuits,

vol. 30, No. 3, pp.251-257, March 1995

[2] Wen-Chang Yeh, Chein-Wei Jen, "High-Speed Booth Encoded Parallel Multiplier Design" IEEE Transactions on Computer, VOL.49, NO.7, JULY 2000

[3] Gensuke Goto, Atsuki Inoue, Ryoichi Ohe, Shoichiro Kashiwakura, Shi mitarai, Takayuki Tsuru, and Tetsuo Izawa "A 4.1 ns Compact 54 \times 54-b Multiplier Utilizing Sign-Select Booth Encoders." IEEE Journal of Solid-State Circuits, Vol. 32, No. 11, pp.1676-1681, November 1992

[4] K. Hwang, Computer Arithmetic/Principles, Architecture, and Design, New York : Wiley, 1979.

[5] 이효상, 송근호, 류진수, 김강철, 한석봉, " 개선된 MODL을 이용한 고성능 32비트 CLA가산기 설계" 대한전자공학회 하계종합학술대회 논문집.18(1) 1995 :357-360



백한석(H. S. Baek)

準會員

2000년 2월 경상대학교

전자공학과 졸업(공학사)

2002년 8월 경상대학교 대학원

전자공학과 수료.

관심분야: VLSI Design, 고성능 연산 회로 설계, VLSI Testing 등임



한석봉(S. B. Han)

正會員

1982년 2월 한양대학교

전자공학과 졸업(공학사)

1988년 2월 한양대학교 대학원

전자공학과 졸업(공학박사)

1988년 3월 경상대학교 전자공학과

전임강사

1992년 1월~1993년 1월 sanford University Department of electrical Engineering and Computer Science POST DOC.

1993년 4월 경상대학교 전자공학과 부교수

1999년 4월~ 경상대학교 전자공학과 정교수

관심분야: VLSI Design(CMOS 및 BiCMOS), VLSI Testing, Analog Testing, CMOS 칩의 Reliability Testing, Design for testability, Fault Tolerance Computing System, VLSI/CAD, ASIC Design