

論文96-33A-3-20

새로운 동적 CMOS 논리 설계방식을 이용한 고성능 32 비트 가산기 설계

(Design of a High-Speed 32-Bit Adder Using
a New Dynamic CMOS Logic)

金 强 哲 *, 韓 哲 鵬 **

(Kang Chul Kim and Seok Bung Han)

요 약

본 논문은 동적 CMOS 논리회로에서 기존의 MODL(multiple output domino logic) 방식의 장점을 유지하면서, MODL 논리회로 보다 면적과 신호의 전파속도를 줄일 수 있는 ZMODL(zipper-MODL)과 EZMODL(enhanced-ZMODL) 회로 설계방식을 제안하였다. 이들을 이용하여 고성능 32 비트 CLA(carry look-ahead adder)를 설계하였고, 2μ 이중금속 CMOS 공정변수를 사용하여 SPICE 3 시뮬레이션에 의해 동작과 성능을 확인하였다. EZMODL 회로로 설계된 CLA는 MODL 회로로 설계된 CLA 보다 약 8%의 트랜지스터를 줄일 수 있었고, 4.8NS 이내에 32 비트 캐리전파가 가능하였다. 본 논문에서 제안한 동적 CMOS 설계 방식은 반복성이 많고, 고속의 연산을 요구하는 회로에서 뛰어난 성능을 제공할 것으로 기대된다.

Abstract

This paper proposes two new dynamic CMOS logic styles, called ZMODL(Zipper-MODL) and EZMODL(enhanced-ZMODL), which can reduce more area and propagation delay than conventional MODL(multiple output domino logic). The 32-bit CLAs(carry look-ahead adder) are designed by ZMODL, EZMODL circuits, and their operations are verified by SPICE 3 with 2μ double metal CMOS parameters. The results show that the CLA designed by EZMODL circuit has achieved 32-bit addition time of less than 4.8NS with VDD=5.0V and 8% of transistors can be reduced, compared to the CLA designed by MODL. The EZMODL logic style can improve the performance in the high-speed computing circuits depending on the degree of recurrence.

I. 서 론

CMOS VLSI 설계기술과 반도체 공정기술의 발전에

* 正會員, 晉州産業大學校 電子計算學科
(Chinju National University)

** 正會員, 慶尙大學校 電子工學科, 自動化 및 컴퓨터
應用技術研究所 研究員
(Gyeongsang National University)

※ 본 연구는 ISRC 91-E-DE-C004의 지원에 의해
수행되었음.

接受日字: 1995年4月4日, 수정완료일: 1996年2月13日

따라 최근에 하나의 IC에 집적도가 높고 고속 연산을 수행할 수 있는 칩들이 다양하게 발표되고 있다. 산술 연산을 수행하는 가산기는 ALU의 성능에 매우 중요한 역할을 하며, 어떠한 캐리 생성 방식을 사용하느냐에 따라 그 성능이 결정될 수 있다. 가산기는 캐리 지연속도와 면적의 트레이드 오프(trade-off) 관계를 고려하여 캐리선택 가산기(carry selection adder)가 많이 이용되고 있으나 적은 면적에 가산기를 구현하고자 할 경우에는 리플캐리 가산기(ripple carry adder)가 사용되며, 면적이 크지만 고속연산이 필요할 경우에는 캐리 룩어헤드 가산기(carry look-ahead adder

:CLA)가 많이 이용되고 있다.^{[11][12][13]} 이러한 가산기들은 서로의 장점 및 단점을 함께 갖고 있으며, 캐리 전파시간이 짧은 CLA는 리플캐리 가산기에 비해 게이트 수가 약 4 배로 증가하여 비트 수가 클수록 회로가 매우 커지고 복잡하게 된다. 따라서 마이크로프로세서나 DSP 칩 등에 사용되는 고성능 연산기를 설계하기 위해서는 집적도가 높고, 소비전력이 적은 논리소자를 사용하여야 하며, 빠른 동작속도를 갖는 연산회로의 설계가 필수적이다.

CMOS 소자는 낮은 전력소비와 고집적도 그리고 높은 잡음여유 등의 특징을 가지므로 대부분의 VLSI 칩에서 사용되고 있다. 특히 domino CMOS 와 NORA CMOS 그리고 Zipper CMOS 등과 같은 동적(dynamic) CMOS는 기존의 정적(static) CMOS에 비하여 동작 속도가 1.5 ~ 2 배 정도 빠른 장점을 갖는 동시에 소비전력은 거의 동일하므로 연산기의 구성 소자로 많은 연구가 수행되고 있다.^{[4][5][6][7]} 그러나 동적 CMOS를 사용하여 회로를 설계할 경우 내부 신호 지연에 의한 시차 문제(race problem), 누설 전류(leakage current), 내부 함수블럭 내에서 발생하는 전하 결합(charge coupling), 전하재분배(charge redistribution) 현상 등으로 회로의 안정도가 떨어지며, 외부의 잡음 등에 쉽게 영향을 받는 문제점들이 있다.^{[8][9]}

동적 CMOS의 이러한 문제점을 해결하기 위하여 MODL(multiple output domino logic) CMOS 설계 방식이 제안되었으며^{[10][11]}, 이 방식은 기존의 domino CMOS 논리 구조에 프리차지(precharge) 소자와 출력 인버터(inverter)를 사용하여 다중 함수를 생성할 수 있도록 변형한 것이다. MODL 회로는 반복되는 게이트를 제거시킬 수 있으므로 적은 면적에 빠른 연산 속도를 갖는 회로의 설계가 가능하고, 동적 CMOS의 누설전류나 전하 재분배 현상을 해결할 수 있다.^[12]

본 논문에서는 기존의 MODL 방식의 장점을 유지하면서, MODL 회로보다 면적과 신호 전파속도를 줄일 수 있는 EMODL(enhanced-MODL), ZMODL(zipper-MODL)^[13]과 EZMODL(enhanced-ZMODL) 회로 설계방식을 제안하였다. ZMODL은 2 단으로 구성된 MODL 회로에서 제 2 단의 N 블럭을 P 블럭으로 대체하여 MODL 게이트의 출력단에 존재하는 인버터를 제거하고, 단일 논리 게이트에서 다중함

수를 생성할 수 있도록 설계하여 트랜지스터의 수가 줄어들도록 설계한 방식이고, EZMODL은 ZMODL의 P 블럭에서 프리차지 소자를 제거하여 면적과 지연속도를 향상시킨 것이다. 이러한 설계방식들은 반복성이 많은 캐리 룩어헤드 가산기와 같은 회로에서 많은 면적을 줄일 수 있다. 그리고 이러한 동적 CMOS 설계방식들을 이용하여 고속연산이 가능한 32 비트 CLA를 설계하고, SPICE 3^[14] 시뮬레이션에 의해 회로의 성능을 확인하였다.

본 논문의 II에서는 EMODL, ZMODL과 EZMODL의 구조 및 동작에 대하여 설명하였고, III.에서는 이들을 이용하여 32 비트 가산기를 설계하였다. 그리고 IV에서는 MODL, EMODL, ZMODL과 EZMODL로 설계된 가산기를 SPICE 시뮬레이션 결과를 이용하여 비교설명하고, V에서 결론을 기술한다.

II. ZMODL과 EZMODL의 구조

Domino CMOS 회로는 주어진 논리 게이트로부터 하나의 출력함수를 얻을 수 있으나, MODL 회로는 단일 논리 게이트에서 다중 함수를 생성할 수 있도록 설계된 회로이다. 어떤 함수가 다른 함수의 부함수(subfunction)가 되는 경우, 즉 아래의 식 (1)과 (2)와 같은 부울 함수로 표현되는 두 함수 F1 과 F2 를 얻고자 할 때에 기존의 domino CMOS 논리 설계방식은 그림 1(a)와 같이 2 개의 domino CMOS 논리 게이트로 구성되어야 한다.

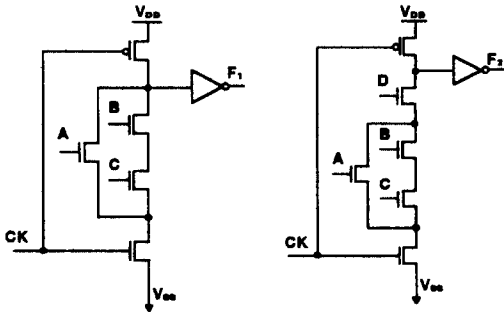
$$F1 = A + B \cdot C \quad (1)$$

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

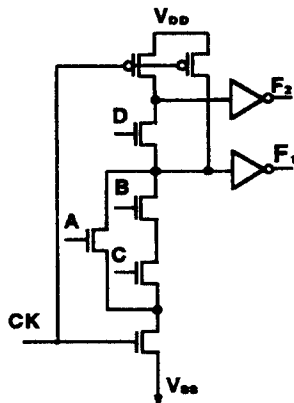
그러나 MODL 회로에서는 그림 1.(b)와 같이 1 개의 논리 게이트로 2 개의 함수를 동시에 설계할 수 있으므로 부함수를 별도로 설계할 필요없이 면적을 감소시킬 수 있다. 이 경우에 회로가 동적 CMOS로 구성되므로 트랜지스터의 크기가 커져 트랜지스터 수의 감소에 의한 면적을 줄일 수 없으나, 게이트 수의 감소로 인한 금속도선 등의 감소로 인하여 전체 면적이 줄어들 수 있고, 지연시간을 줄일 수 있다.^[12] 이 회로의 설계방식은 캐리생성과 같은 반복적인 구조를 갖는 논리회로에 대해 적합한데, CLA와 같은 회로는 반복성

이 매우 크므로 MODL 회로를 이용하여 회로를 설계 하면 domino 논리 회로를 사용하여 설계한 것보다 트랜지스터 수를 많이 감소 시킬수 있다.

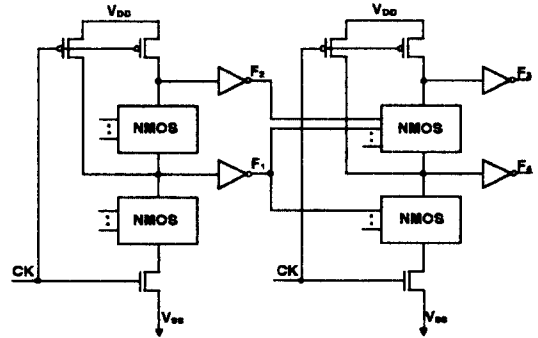
MODL은 그림 1(c)에서와 같이 2 단으로 구성되는 회로에서도 같은 논리 구조가 반복된다는 것을 알 수 있다. 그러나 Zipper CMOS에서와 같이 MODL 회로의 첫단 출력의 인버터를 제거하고, 다음 단의 N 블럭을 P블럭으로 바꾸면 다음 단의 인버터가 제거되어 면적을 줄일 수 있고, 2 게이트의 지연시간을 줄일 수 있는데 이러한 구조를 ZMODL이라 한다. 그림 1.(d)는 그림.1(c)의 회로를 ZMODL로 바꾼 것으로 4 개의 게이트가 줄어들고, 2 게이트의 전파지연을 줄일 수 있다는 것을 알 수 있다. 그러나 ZMODL 회로의 P 블럭에는 NMOS 보다 캐리어의 이동도가 떨어지는 PMOS가 직렬로 연결되어 신호지연이 증가하고, 출력의 구동능력이 약화될 수 있고, CK'가 사용되므로 클럭회로가 복잡하게 되는 단점들이 있다.



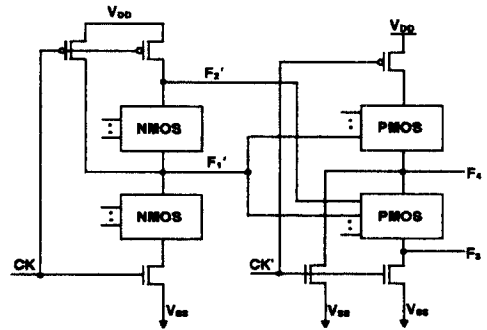
(a) Domino CMOS 회로
(a) Domino CMOS circuit



(b) MODL 회로
(b) MODL circuit



(c) 2단으로 구성된 MODL 회로
(c) MODL circuit with two stages

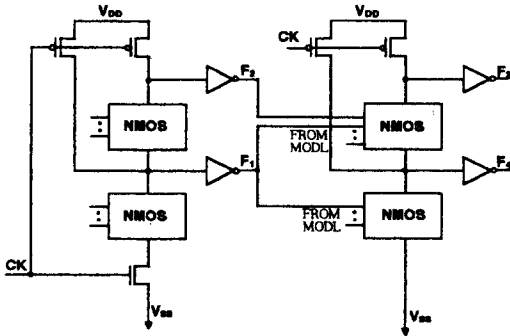


(d) ZMODL 회로
(d) ZMODL circuit

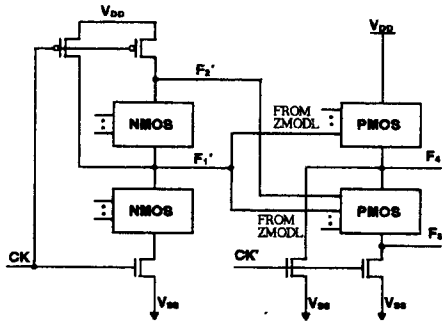
그림 1. Domino CMOS, MODL, ZMODL 회로
Fig. 1. Circuits of domino CMOS, MODL and ZMODL.

기본적인 동적 CMOS 게이트는 CK가 0 일 때 (precharge phase) CK에 연결된 PMOS를 통하여 프리차지(precharge)시키고, CK가 1 일 때 (evaluation phase) CK에 연결된 NMOS를 통하여 디스차지(discharge) 시킨다. 그러나 2 단으로 구성되는 MODL 회로에서는 두 번째 단계에 사용되는 MODL 게이트의 모든 입력이 MODL 게이트의 출력으로부터만 들어오는 경우에 프리차지 시간 동안에 입력이 모두 LOW가 되므로 디스차지 소자 NMOS는 회로의 동작에 아무런 영향을 미치지 못한다. 따라서 그림 2(a)에서와 같이 CK에 연결되어 있는 디스차지 소자 NMOS를 제거할 수 있는데 이러한 구조를 EMODL이라 한다. 같은 방법으로 그림2(b)와 같이 ZMODL에서도 완정소자 PMOS를 제거할 수 있고 이를 EZMODL이라 한다. EZMODL의 경우에는 직렬

로 연결된 PMOS가 1 개 줄어들어 전파지연 시간이 크게 감소할 수 있을 것이다.



(a) EMODL의 구조
(a) Structure of EMODL



(b) EZMODL의 구조
(b) Structure of EZMODL

그림 2. EMODL과 EZMODL의 구조
Fig. 2. Structures of EMODL and EZMODL.

동적 CMOS가 다단으로 구성되어 있을 때는 전단의 내부 신호지연에 의한 시차문제가 발생하고, 내부 함수 블럭내에서 발생하는 누설 전류, 전하 재분배 그리고 전하결합 현상으로 인하여 출력 전압이 변동하고 잡음 여유가 감소한다. ZMODL과 EZMODL 회로는 함수블럭 내부 노드에 다출력 형태의 프리차지 소자를 사용하여 내부 회로간에 발생하는 누설전류나 전하재분배 현상 및 전하결합 등의 문제를 해결하여 출력 전압의 변동이 제거되고 잡음여유가 증가되어 안정된 회로로 동작할 수 있다. 시차 문제는 동적 CMOS가 다단으로 구성되어 있을 때 전단의 디스차지 시간지연으로 인하여 판정시간(evaluation time)에 다음 단의 출력이 예측할 수 없는 상태로 나타나는 현상이다. 이러한 문제점들은 ZMODL 회로에서는 각 단의 함수

블럭들이 NMOS와 PMOS를 번갈아 사용되므로 회로의 신호지연에 의한 시차문제가 해결된다. 즉 프리차지 시간 동안에 첫단의 프리차지 소자 PMOS가 ON 되어 첫 단의 출력을 1로 프리차지시키고, 두 번째 단의 프리디스차지 소자 NMOS가 ON 되어 P 블럭의 출력은 0으로 프리디스차지되어 신호 지연에 의한 시차 문제를 해결할 수 있다. 그리고 EMODL(EZMODL) 회로에서는 두 번째 단의 모든 입력이 첫 단에서만 들어오므로 프리차지 시간 내에 HIGH(LOW)가 되어 시차문제가 해결된다.

III. 32 비트 가산기 설계

가산기에서 가산이 수행되는 시간은 모든 전가산기를 통하여 캐리의 전파에 요구되는 시간에 의해 결정된다. CLA는 캐리의 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.

$$G_i = A_i \cdot B_i \tag{3}$$

$$P_i = A_i \cdot B_i' + A_i' \cdot B_i \tag{4}$$

$$C_{i+1} = G_i + P_i \cdot C_i \tag{5}$$

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

$$EGP_{i+7} = P_{i+7} \cdot P_{i+6} \cdot P_{i+5} \cdot P_{i+4} \cdot P_{i+3} \cdot P_{i+2} \cdot P_{i+1} \cdot P_i \tag{6}$$

$$EGG_{i+7} = G_{i+7} + G_{i+6} \cdot P_{i+7} + G_{i+5} \cdot P_{i+7} \cdot P_{i+6} + G_{i+4} \cdot P_{i+7} \cdot P_{i+6} \cdot P_{i+5} + G_{i+3} \cdot P_{i+7} \cdot P_{i+6} \cdot P_{i+5} \cdot P_{i+4} + G_{i+2} \cdot P_{i+7} \cdot P_{i+6} \cdot P_{i+5} \cdot P_{i+4} \cdot P_{i+3} + G_{i+1} \cdot P_{i+7} \cdot P_{i+6} \cdot P_{i+5} \cdot P_{i+4} \cdot P_{i+3} \cdot P_{i+2} + G_i \cdot P_{i+7} \cdot P_{i+6} \cdot P_{i+5} \cdot P_{i+4} \cdot P_{i+3} \cdot P_{i+2} \cdot P_{i+1} \tag{7}$$

위 식들은 같은 부울함수가 반복되므로 MODL 회로를 사용하기 위하여 식 (6)과 (7)을 다시 쓰면 식 (8)과 (9)가 된다.

$$EGP_{i+7} = EP_{i+7} \cdot EP_{i+5} \cdot GP_{i+3} \tag{8}$$

$$EGG_{i+7} = EG_{i+7} + EP_{i+7} \cdot (EG_{i+5} + EP_{i+5} \cdot GG_{i+3}) \quad (9)$$

식 (8)과 (9)에서 사용된 부울함수는 아래와 같다.

$$\begin{aligned} EG_{i+1} &= G_{i+1} + G_i \cdot P_{i+1}, \\ EP_{i+1} &= P_{i+1} \cdot P_i \\ GP_{i+3} &= EP_{i+3} \cdot EP_{i+1}, \\ EP_{i+3} &= P_{i+3} \cdot P_{i+2} \\ EG_{i+3} &= G_{i+3} + G_{i+2} \cdot P_{i+3}, \\ GG_{i+3} &= EG_{i+3} + EP_{i+3} \cdot EG_{i+1} \\ EG_{i+5} &= G_{i+5} + G_{i+4} \cdot P_{i+5}, \\ EP_{i+5} &= P_{i+5} \cdot P_{i+4} \\ EG_{i+7} &= G_{i+7} + G_{i+6} \cdot P_{i+7}, \\ EP_{i+7} &= P_{i+7} \cdot P_{i+6} \end{aligned}$$

위에서 GP_{i+3} 와 GG_{i+3} 는 4 비트 그룹캐리 생성항과 전파항이다. 32 비트의 경우에 8 비트로 분할하여 $i=8, 16, 24, 32$ 에 대하여 캐리를 부울함수로 나타내면 식 (10), (11)과 같이 표현된다.

$$C_8 = EGG_7 + EGP_7 \cdot C_0 \quad (10)$$

$$C_{16} = EGG_{15} + EGP_{15} \cdot C_8 \quad (11)$$

$$C_{24} = EGG_{23} + EGP_{23} \cdot C_{16} \quad (12)$$

$$C_{32} = EGG_{31} + EGP_{31} \cdot C_{24} \quad (13)$$

그리고 식 (8)과 (9)를 ZMODL과 EZMODL에서 사용하기 위하여 PMOS에 대한 부울함수를 구하면 아래의 식으로 표현된다.

$$GP_{i+3} = EP_{i+3}' + EP_{i+1}' \quad (14)$$

$$GG_{i+3} = EG_{i+3}' \cdot (EP_{i+3}' + EG_{i+1}') \quad (15)$$

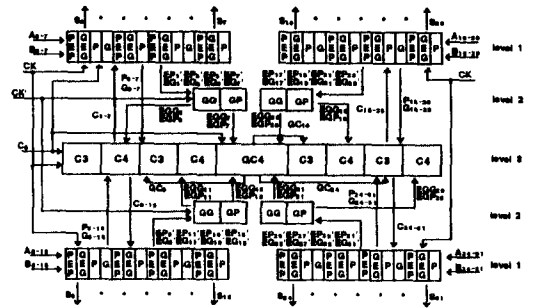
$$EGP_{i+7} = EP_{i+7}' + EP_{i+5}' + EP_{i+3}' + EP_{i+1}' \quad (16)$$

$$EGG_{i+7} = EG_{i+7}' \cdot (EP_{i+7}' + EG_{i+5}' \cdot (EP_{i+5}' + EG_{i+3}' \cdot (EP_{i+3}' + EG_{i+1}')))) \quad (17)$$

본 논문에서 설계한 CLA는 캐리 생성 블럭과 합(SUM)을 구하는 블럭으로 구성된다. 합을 구하는 블럭은 기존의 동적 CMOS를 사용하여 XOR 게이트로 설계하였으며, 캐리 생성 블럭은 다중 함수를 출력하는 EMODL, ZMODL과 EZMODL 회로를 각각 사용하여 설계하였다.

그림 3은 32 비트 CLA 가산기의 전체 블럭도를 나타낸 것으로 4 개의 8 비트 가산기로 구성된다. 계층(level) 1에는 입력 A, B를 받아 캐리 생성항과 캐리 전파항을 발생시키고, 계층 3으로부터 받아들이는 캐리

를 이용하여 합을 발생시킨다. 계층 2에서는 계층 1로부터 얻어진 캐리 생성항과 전파항을 이용하여 8 비트 그룹캐리 생성항과 전파항을 만들어낸다. 이 계층은 P 블럭으로 이루어지므로 CK'가 사용된다. 계층 3은 캐리를 발생하는 부분으로 계층 1로부터 들어오는 입력을 이용하여 캐리(C_0-C_{31})들을 발생하는 부분과 계층 2로부터 얻은 그룹캐리 생성항과 전파항을 이용하여 그룹캐리 $C_8, C_{16}, C_{24}, C_{32}$ 를 만드는 부분으로 이루어진다. C_{32} 는 64 비트 가산기를 만들 때 이용될 수 있다.



- PEP : P_i, EP_{i+1} circuit,
- GEG : G_i, EG_{i+1} circuit
- P : carry propagate circuit,
- G : carry generate circuit
- C3 : 3 bit carry circuit
- C4 : 4-bit carry circuit
- GC4 : 4-bit group carry circuit
- GG : 8-bit Group carry generate circuit
- GP : 8-bit group carry propagate circuit

그림 3. 32 비트 CLA 블럭 다이어그램
Fig. 3. Block diagram of 32-bit CLA.

그림 4는 ZMODL로 설계된 CLA에서 사용되는 게이트들을 나타낸 것이다. 그림 4(a), (b)는 1,2 비트 캐리 생성항(G_i)과 캐리 전파항(P_i)을 구하는 회로를 나타낸 것으로, 합을 구하는 곳에 사용되는 G_i 와 P_i 는 인버터를 거쳐서 출력이 나온다. 그러나 다음 단에서 사용될 EG_{i+1}' 와 EP_{i+1}' 는 출력 인버터를 거치지 않고 직접 P 블럭의 입력으로 사용되므로 MODL 회로보다 2 개의 출력 게이트를 제거할 수 있다. 그림 4(c), (d)는 그룹캐리 생성항(EGG_{i+7})과 그룹캐리 전파항(EGP_{i+7})을 발생하는 회로이다. 식 (16)과 (17)을 P 블럭으로 구성하므로 입력에 $EG_{i+1}', \dots, EG_{i+7}'$ 과 $EP_{i+1}', \dots, EP_{i+7}'$ 가 사용되고, 출력에는 출력 인버터를 통과하지 않고 EGG_{i+7} 와 EGP_{i+7} 를 생성하게 되므로

MODL에서 사용되는 출력단의 인버터를 제거할 수 있어 면적과 신호전파 지연시간을 줄일 수 있다. 그림 4 (e)는 그룹캐리 발생회로로 다음 단의 ZMODL로 연결될 출력이 없으므로 MODL에서와 같은 회로가 사용되며, 3-비트, 4-비트캐리 발생회로에서도 사용될 수 있다. MODL로 설계된 CLA는 6 게이트 캐리전과 지연시간을 가지고 있으며, ZMODL과 EZMODL로 설계된 CLA는 4 게이트의 캐리전과 지연시간이 소요됨을 알 수 있다.

ZMODL과 EZMODL에 사용된 회로에는 다중 함수를 구현하기 위하여 여러 개의 프리차지와 프리디스차지 소자가 사용되기 때문에 역전류 통로(reverse current path)가 형성될 수 있다. 이러한 문제를 해결하기 위하여 첫 단계에서는 그림 4(a), (b)에서와 같이 OR 게이트 대신에 XOR 게이트를 사용하였고, 두 번째 단계에서는 역전류 통로가 형성되는 입력신호가 발생하지 않도록 하였다.

그림 4의 (c), (d) 회로에서 프리차지 시간 동안에 입력신호 EG_{i+1}' , ..., EG_{i-7} 와 EP_{i+1}' , ..., EP_{i-7} 은 모두가 앞단의 ZMODL 회로의 출력으로부터 나오므로 HIGH가 되어 프리차지 소자는 회로의 동작에 아무런 영향을 미치지 못하므로 그림 5와 같이 EZMODL 회로로 바꿀 수 있다. 그러나 입력신호가 HIGH에 도달하는 시간이 길어지면 PMOS가 완전히 OFF되지 않고 NMOS와 PMOS가 동시에 ON되어 EGG_{i+3} 과 EGG_{i+7} 이 LOW로 되지 않을 수도 있으나, 계층 1의 PMOS의 크기를 조절하여 프리디스차지 시간 내에 충분히 출력을 HIGH로 만들 수 있다. 따라서 직렬로 연결된 5개의 PMOS가 4 개로 구성될 수 있으며, ZMODL에서 문제가 되었던 신호지연시간과 구동능력이 개선되어 현격한 속도의 향상을 가져올 수 있을 뿐만 아니라 면적도 줄일 수 있다.

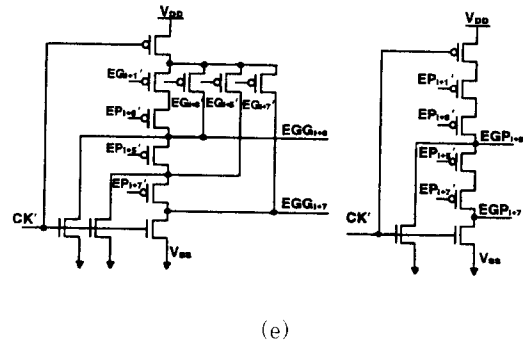
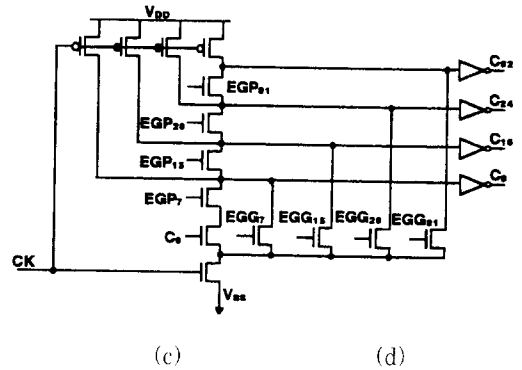
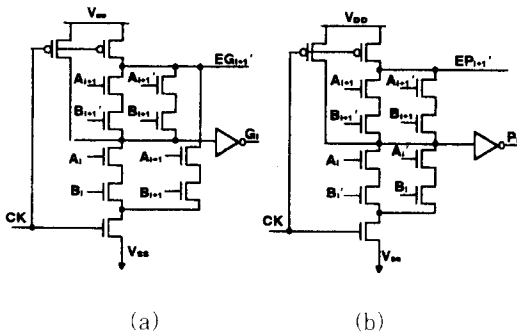


그림 4. CLA에 사용된 ZMODL 게이트
 (a) 1-,2-비트 생성회로(G_i, EG_{i+1}') (b) 1-,2-비트 전파회로(P_i, EP_{i+1}') (c) 4-,8-비트 그룹캐리 전파회로(EGP_{i+3}, EGP_{i+7}) (d) 4-,8-비트 그룹캐리 생성회로(EGG_{i+3}, EGG_{i+7}) (e) 그룹캐리 회로

Fig. 4. ZMODL gates used in CLA.
 (a) 1-, 2-bit generate circuit (b) 1-, 2-bit propagate circuit (c) 4-, 8-bit group propagate circuit (d) 4-, 8-bit group generate circuit (e) group carry circuit

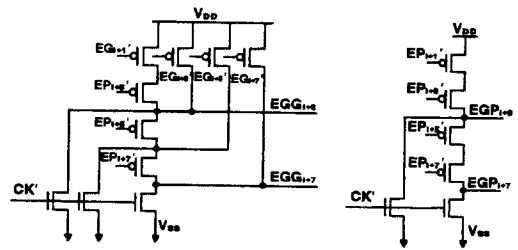


그림 5. EZMODL의 8-비트 그룹 캐리생성과 전파회로
 Fig. 5. 8-bit group carry generate and propagate circuit of EZMODL.

IV. 시뮬레이션 결과 및 고찰

본 논문에서는 서울대학교 반도체공동 연구소의 2um double metal polysilicon 게이트 제조 공정에서 추출한 공정변수의 값을 SPICE 파라메타로 사용하였다. 그리고 본 논문에서 제안한 EMODL, ZMODL과 EZMODL 회로로 설계된 32 비트 CLA를 SPICE 3로 시뮬레이션하여 기존의 MODL 회로와 비교하였다. 시뮬레이션에는 20NS(50MHZ)의 주기를 가진 CK와 CK'가 사용되었다.

그림 6는 ZMODL, EZMODL 회로를 이용하여 설계한 CLA의 시뮬레이션한 결과를 MODL로 설계한 CLA와 비교한 것으로, 입력에 $A=FFFFFFF_{(16)}$, $B=00000000_{(16)}$, $C_0=1$ 을 인가하였다. ZMODL 회로에서는 캐리가 전파되는 임계지연 통로(critical path)에서 2 게이트가 줄어 MODL 회로의 CLA 보다 캐리전파 속도가 2/3로 감소해야 되지만, C_{31} 과 GC_{32} 의 지연시간이 각각 5.8NS, 4.6NS로서 MODL의 5NS, 3.6NS 보다 0.8NS 이상 길어짐을 알 수 있다. 이러한 현상은 5 개의 PMOS가 직렬로 연결되어서 저항이 커져 구동능력이 약해지고, PMOS 소자의 캐리 이동속도가 느린 것에 기인한 것임을 알 수 있다. EZMODL 회로로 설계된 CLA에서는 C_{31} 과 GC_{32} 의 캐리 지연시간이 각각 4.8NS, 3.4NS이며, ZMODL 회로로 설계된 CLA보다 1NS 이상, MODL 회로로 설계된 CLA 보다 0.2NS 빨라짐을 알 수 있다. 그러나 EMODL 회로로 설계된 CLA는 디스차지 NMOS가 제거되었지만, 캐리지연 통로에 영향을 미치지 못하므로 캐리 전파속도는 거의 변화가 없었다.

EZMODL 회로의 경우에 그림 5에서 입력신호 EG_{i+1}' , ..., EG_{i+7}' 와 EP_{i-1}' , ..., EP_{i+7}' 의 프리차지 시간이 늦어지면 8 비트 그룹캐리 생성항과 전파항 회로의 NMOS와 PMOS가 동시에 ON이 되어 8 비트 그룹캐리 생성항과 전파항 신호가 오동작을 일으킬 수 있다.

그림 7은 EZMODL로 설계된 CLA에서 (a)는 입력신호가 $A=FFFFFFF_{(16)}$, $B=00000000_{(16)}$, $C_0=1$ 일 경우에, (b)는 입력신호가 $A=B=FFFFFFF_{(16)}$, $C_0=1$ 일 경우에 EP_7' , EG_7' , EGP_{31} 와 EGG_{31} 의 출력파형을 보여준다. 그림 7(a), (b)에서와 같이 EGP_{31} 와 EGG_{31} 는 캐리의 전파신호 보다 훨씬 짧은 1.8NS 이내에 완전히 LOW 상태가 되고, 입력으로

사용된 EP_7' 과 EG_7' 은 2.8NS 이내에 HIGH로 프리차지되므로 프리차지 소자 PMOS를 제거한 EZMODL 회로가 정상적으로 동작하고 있다는 것을 알 수 있다.

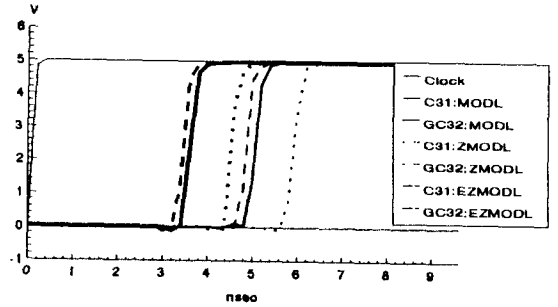
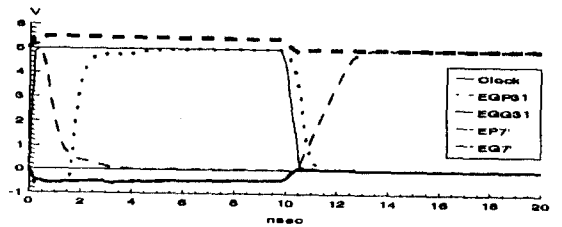
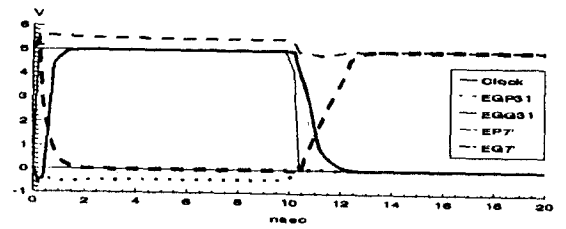


그림 6. ZMODL, EZMODL과 MODL CLA의 출력 Fig. 6. Output waveforms of ZMODL, EZMODL and MODL CLA.



(a) $A=FFFFFFF_{(16)}$, $B=00000000_{(16)}$, $C_0=1$



(b) $A=B=FFFFFFF_{(16)}$, $C_0=1$

그림 7. EZMODL CLA의 그룹캐리 생성항과 전파항의 출력 파형

Fig. 7. Output waveforms of group carry generate and propagate in EZMODL CLA.

표 1은 본 논문에서 제안하는 EMODL, ZMODL, EZMODL 회로와 기존의 MODL 회로로 설계된 32 비트 CLA의 캐리블럭의 트랜지스터 수와 캐리전파 지연시간을 비교한 것이다. MODL 회로의 경우에는 946 개의 트랜지스터가 사용되었고, ZMODL 회로에

서는 878 개의 트랜지스터가 사용되어 면적이 줄어들었으나, P 블럭의 사용으로 캐리 전파속도가 오히려 증가하였다. EMODL 회로의 경우에는 4 개의 트랜지스터가 줄어들었으나 모두 디스차지 소자였기 때문에 캐리 전파속도에는 큰 영향을 미치지 못하였다. EZMODL 회로로 설계된 CLA에서는 약 8%(72 개)가 줄어든 874 개의 트랜지스터가 사용되었고, 캐리 전파시간도 0.2NS가 줄어들었다. 따라서 본 논문에서 제안한 EZMODL 회로로 설계된 32 비트 CLA가 면적과 캐리지연속도에서 MODL 회로로 설계된 CLA 보다 우수함을 알 수 있다.

표 1. 32 비트 CLA의 지연시간과 캐리블럭의 트랜지스터 수

Table 1. # of TRs and carry delay time of CLAs.

	# of TRs	GC32(NS)	C31(NS)
MODL	946(1138)	3.6	5.0
ZMODL	878(1070)	4.6	5.8
EMODL	942(1134)	3.6	5.0
EZMODL	874(1066)	3.4	4.8

() 안은 SUM 회로를 포함한 CLA 전체 트랜지스터 수

V. 결 론

본 논문에서는 칩면적이 작고 소비 전력이 적으며 동작 속도가 빠른 동시에 회로 안정도가 뛰어난 새로운 동적 CMOS 회로인 EMODL, ZMODL과 EZMODL 설계방식을 제안하였다. 그리고 이들의 회로를 이용하여 고속연산이 가능한 32 비트 CLA를 설계하였으며, SPICE 시뮬레이션을 수행하여 회로 성능 평가와 논리 정확성을 확인하였다. ZMODL 회로로 설계된 CLA는 면적은 줄어들었으나 P 블럭의 사용으로 인하여 캐리전파속도가 증가하여 고속 연산기의 설계에는 어려움이 있다는 것을 확인하였다. 그러나 EZMODL 회로로 설계된 CLA는 MODL 회로를 이용한 CLA보다 임계지연속도(C_{31})가 4.8NS로 0.2NS 이상 줄어들었고, 약 8%의 트랜지스터를 줄일 수 있어 가산기의 설계에 가장 중요한 요소인 면적과 캐리 지연속도에서 성능이 우수하다는 것을 확인하였다.

앞으로 EZMODL 회로를 기본으로 32 비트 CLA를 제작할 예정이며, 본 연구에서 제안하는 EZMODL 회로는 고성능 연산기능을 요구하는 고성능 컴퓨터의

연산장치, 디지털 신호 처리 시스템 및 특수 목적용 칩 등에서 뛰어난 성능을 제공할 수 있을 것으로 기대된다.

참 고 문 헌

- [1] K.Hwang, Computer Arithmetic : Principles, Architecture, and Design, John Wiley & Sons, 1979.
- [2] J.J.F Cavanagh, Digital Computer Arithmetic : Design and Implementation, McGraw-Hill, 1985.
- [3] N.Weste and K.Eshraghian, Principles of CMOS VLSI Design, Adderison-Wesley, 1988.
- [4] J.A. Pretorius, A.S. Shubat and C.A.T. Salama, "Analysis and design optimization of Domino CMOS logic with application to standard cells," IEEE Journal of Solid-State Circuits, vol. SC-20, pp. 523-530, April 1985.
- [5] N.P Goncalves and H.J. de Man, "NORA: a racefree dynamic CMOS technique for pipelined logic structures," IEEE Journal of Solid-State Circuits, vol. SC-18, pp.261-268, June 1983
- [6] C.M Lee and E.W.Szeto, "Zipper CMO S," IEEE Circuits and Devices magazine, pp. 10-17, May 1986.
- [7] R.H.Krambeck, C.M.Lee and H.F.S. Law, "High-speed compact circuits with CMOS," IEEE J. Solid-State Circuits, vol. SC-17, pp. 614-619, June 1982.
- [8] J.A. Pretorius, A. S. Shubat and C.A.T. Salama, "Charge redistribution and noise margins in Domino CMOS logic," IEEE Transactions on Circuits and Systems, vol. CAS-33, no.8, pp.768-793, Aug. 1986.
- [9] David A. Hodges and Horace G. Jaskson, Analysis and Design of Digital Integrated Circuits, pp. 68-124, New York, 1983.
- [10] I.S.Hwang and A.L.Fisher, "A 3.1 ns 32b CMOS Adder in multiple output

- domino logic." in ISSCC Dig. Tech. Pupers. 1988. pp.140-141.
- [11] I.S.Hwang and P.S.Magarshack. "A high-speed dynamically reconfigurable 32 bit CMOS adder." in Proc. CICC. 1988. pp. 17.5.1-17.5.6.
- [12] I.S.Hwang and A.L.Fisher. "Ultrafast Compact 32-bit CMOS Adders in Multiple-Output Domino Logic." IEEE Journal of Solid-State Circuits, vol. 24, No. 2, pp.358-368, April 1989.
- [13] 서울대학교 반도체공동연구소, 고성능 Micro-processor를 위한 새로운 다이내믹 CMOS 논리설계 방식 및 Testable Design. ISRC 91-E-DE-C004 최종보고서
- [14] Walter Banzhaf. Computer-Aided Circuit analysis using SPICE, 1989.

— 저 자 소 개 —



金 强 哲(正會員)

1958년 12월 13일생. 1981년 2월 서강대학교 전자공학과 졸업(공학사). 1983년 2월 서강대학교 대학원 전자공학과 졸업(공학석사). 1992년 3월 ~ 경상대학교 대학원 전자공학과 박사과정. 1983년 3월 한국전자통신 연구소 반도체 연구단. 1989년 7월 삼성종합기술원 정보시스템 연구소. 1990년 3월 진주농림전문대학 전자계산학과. 1993년 7월 ~ 현재 진주산업대학교 전자계산학과 조교수.

韓 哲 鵬(正會員) 第 32卷 B編 第 11號 參照
현재 경상대학교 전자공학과 부교수