

論文98-35C-8-4

가변 크기 셀을 이용한 저전력 고속 16비트 ELM 가산기 설계

(A Design of High Speed and Low Power 16bit-ELM Adder Using Variable-Sized Cell)

柳 凡 善 * , 趙 泰 元 *

(Beom Seon Ryu and Tae Won Cho)

요 약

일반적인 ELM 가산기 구조에서 하위 비트의 논리 레벨(logic depth)이 상위 비트보다 작은 사실을 이용하여 가변 크기 셀(variable-sized cell)을 이용한 16비트 저전력 고속 ELM 가산기를 설계하였다. 0.8 μ m 단일 폴리 이중 금속 LG CMOS 공정 변수를 이용하여 16비트 ELM 가산기를 모의 실험한 결과, 가변 크기 셀로 구현한 16비트 ELM 가산기가 기준 크기 셀(reference-sized cell)로 구현한 ELM 가산기에 비해 power-delay-product가 19.3% 감소됨을 보였다. static CMOS, 패스 트랜지스터 로직(pass-transistor logic), Wang의 XOR/XNOR게이트 등 다양한 로직 스타일(logic style)을 혼합하여 최적 설계를 하였다. ELM 가산기의 최대 지연 경로는 S셀의 구현 방법 및 로직 스타일에 따라서 달라진다.

Abstract

We have designed a high speed and low power 16bit-ELM adder with variable-sized cells utilizing the fact that the logic depth of lower bit position is less than that of the higher bit position in the conventional ELM architecture. As a result of HSPICE simulation with 0.8 μ m single-poly double-metal LG CMOS process parameter, our 16bit-ELM adder with variable-sized cells shows the reduction of power-delay-product, which is less than that of the conventional 16bit-ELM adder with reference-sized cells by 19.3%. We optimized the design with various logic styles including static CMOS, pass-transistor logic and Wang's XOR/XNOR gate. Maximum delay path of an ELM adder depends on the implementation method of S cells and their logic style.

I. 서 론

최근 전자기기들의 소형화 경량화 추세에 따라서 저전력으로 동작할 수 있는 구조들이 연구되고 있다. 그 중 가산기는 모든 기기 들에서 사용되는 가장 보편적인 장치라 할 수 있다. 본 연구에서는 16비트 가산기를 ELM 구조로 구현하였으며 ELM 가산기의 지연 시간이 레벨이 증가할수록 커지는 것을 이용하여 레벨

에 따라서 트랜지스터 사이징(transistor sizing)을 다르게 하여 전력 소모를 줄였다. ELM 가산기는 1992년 펜실버니아 주립대학에서 개발한 VLSI CAD Tool의 하나인 FACTOR에 의해서 개발된 구조로서 이진 트리 구조에서 각각의 합이 병렬적으로 직접 계산되어 셀 사이의 연결선의 수가 작고, 면적이 작고 속도가 빠르며, 셀이 규칙적으로 배열되어 레이아웃이 용이한 장점이 있다.^[1-2] FACTOR는 임의의 회로에서 통신 복잡도(communication complexity)를 최소로 하는데 기반을 둔 multi-level 논리 합성 틀이다. ELM이란 용어는 약지는 아니며 알고리즘 및 가산기 명칭을 ELM이라 한다.^[1] 이 가산기는 여러 종

* 正會員, 忠北大學校 電氣電子工學部
(School of Electronic and Electrical Engineering,
ChungBuk National Univ.)
接受日字:1997年11月17日, 수정완료일:1998年7月23日

류의 병렬 가산기중에서 면적, 속도, 전력 소모 등을 고려했을 때, 모든 비트 수에 걸쳐서 가장 속도가 빠르며 가장 적은 power-delay-product를 나타낸다.^[1-2] ELM 가산기는 합이 병렬적으로 직접 계산되므로 상위 레벨에 위치한 합들의 지연 시간이 크다. 따라서 본 연구에서는 하위 레벨에 대해서 트랜지스터 다운 사이징(transistor down sizing)을 통해 전력 소모를 줄이는 방법에 대해 논한다. 로직 스타일은 static CMOS, 팩스 트랜지스터 로직, Wang의 XOR/XNOR 게이트 등 여러 로직 스타일을 혼합하여 지연 시간 및 전력 소모를 최소화하였다.^[3-5] 본 논문의 구성은 다음과 같다. II장에서는 ELM 가산기의 알고리즘 및 구조를 BCL(Binary Carry Look-ahead) 가산기와 비교한 후 기본 셀의 구현에 대해 알아본다. III장에서는 16비트 ELM 가산기를 기준 크기 셀을 이용하여 구현하고 IV장에서는 가변 크기 셀을 이용하여 구현한다. V장에서는 모의 실험 결과를 분석한다.

II. ELM 가산기

1. 알고리즘

ELM 알고리즘은 1992년 미국 펜실버니아 대학에서 개발된 VLSI CAD tool의 하나인 FACTOR에 의해 개발된 알고리즘이다.^[11] 이 알고리즘은 1982년 Brent와 Kung에 의해 개발된 B & K 알고리즘과 약간의 차이가 있다.^[6] B & K 알고리즘에 의해 구현된 가산기가 B & K 가산기 또는 BCL(Binary Carry Lookahead) 가산기이다.^[1, 6-7] 두 알고리즘의 공통점은 기존의 설계 기준이 되었던 게이트 수보다는 설계 규칙성과 면적을 고려한 알고리즘으로서 각 비트의 캐리를 parallel prefix computation 방식으로 계산한 점이다. 반면에 차이점은 임의의 노드에서 계산하는 신호가 다르다. B & K는 임의의 노드에 대해서 모든 비트의 블록 전파 신호(block propagation signal) 및 블록 발생 신호(block generation signal)를 계산한다. 이때 얻어진 블록 발생 신호가 i번째 비트의 캐리 입력이 된다. 이에 비하여 ELM 알고리즘은 임의의 노드에 대하여 각 비트의 블록 전파 신호, 각 비트의 부분합(partial sum), 그리고 한 비트의 블록 발생 신호만 발생시킨다. B & K 알고리즘과의 차이점은 임의의 노드에 대해 한 비트의 블록 발생 신호

를 발생시키면서 나머지 비트의 블록 발생 신호들은 이전 레벨의 부분 합에 반영된다. 이렇게 함으로써 내부 셀 간의 연결선 수가 B & K보다 줄어들어 면적이 작아지고 속도가 빨라진다.^[1-2, 7]

ELM 알고리즘을 간단히 설명하면 다음과 같다. 그림1은 이진 트리 구조를 갖는 8비트 ELM 가산기의 블록도를 나타낸 것이다.

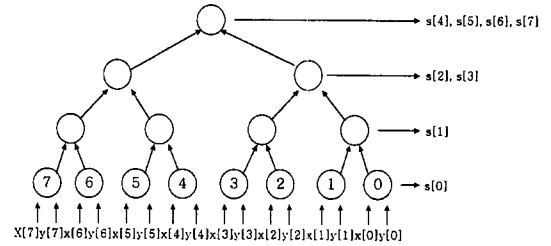


그림 1. 8비트 ELM 블록도
Fig. 1. ELM block diagram for n=8.

이진 트리 구조의 알고리즘은 $O(\log n)$ 의 계산 시간이 필요하고 $O(n \cdot \log n)$ 의 면적이 요구된다. 올림수 예견 가산기에서 정의된 발생 신호(generate signal)와 전파 신호(propagate signal)는 다음식 (1) (2)와 같다.^[8]

$$g_i = x_i y_i, \text{ (단 } i \text{ 는 비트 위치를 나타냄)} \quad (1)$$

$$p_i = x_i \oplus y_i, \quad (2)$$

그리고 $0 \leq i < j \leq n-1$ 에 대해서, 다음식 (3)~(6)과 같이 정의할 수 있다.^[11]

$$P(j, i) = p_j P(j-1, i) \quad (3)$$

$$P(i, i) = p_i, \quad (4)$$

$$G(j, i) = g_j + p_j G(j-1, i) \quad (5)$$

$$G(i, i) = g_i, \quad (6)$$

즉, $P(j, i)$ 는 i번째 이전에 발생된 블록 발생 신호가 j번째 비트까지 전파 될 수 있는지의 여부를 판단하는 블록 전파 신호이고, $G(j, i)$ 는 i번째와 j번째 사이의 블록 발생 신호를 나타낸다.

그림 2는 이진 트리 구조를 취하는 ELM 알고리즘에서 임의의 노드에서 계산되어지는 값들을 정리한 것이다.^[11]

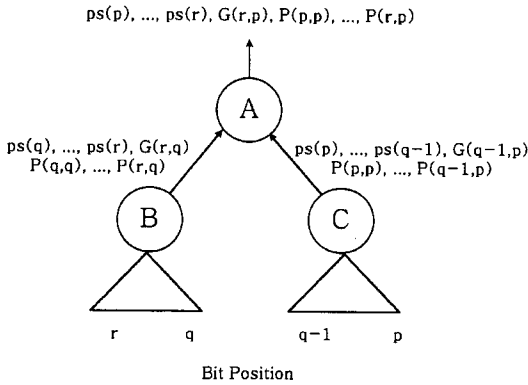


그림 2. ELM 가산기에서 계산되는 신호들
Fig. 2. Computed signals in ELM adder

여기서 A노드의 최종적인 부분 합은 다음 식(7)과 같이 나타낼 수 있다. A, B, C는 노드를 나타낸다. 이때 A노드에서의 $q+j$ 번째 비트의 부분 합(ps_{q+j}^A)은 B노드의 블록 전파 신호($P^B(q+j-1, q)$)와 C노드에서 발생된 블록 발생 신호($G^C(q-1, p)$)를 AND 연산을 한 다음, 이 신호와 B노드의 부분 합(ps_{q+j}^B)과 XOR 연산을 하여 얻어진다. 즉, C노드의 부분 합은 곧바로 A노드의 부분 합이 되며 B노드의 부분 합이 A노드의 부분 합 또는 그의 보수가 될지는 C노드에서 발생하는 블록 발생 신호와 B노드에서 발생하는 블록 전파 신호의 AND 신호에 의해 결정된다.

$$ps_{q+j}^A = ps_{q+j}^B \oplus P^B(q+j-1, q)G^C(q-1, p) \quad (7)$$

2. 구조

ELM 가산기는 식 (3)~(6)을 수행하는 E, P, G, S 등 4개의 기본 셀이 존재하는데 각 셀이 수행하는 연산은 각각 $E = ps \oplus G$, $P = PQ$, $G = G + PH$, $S = ps \oplus PG$ 이다. E 및 S셀은 부분 합을 계산하고 G셀은 블록 발생 신호를, P셀은 블록 전파 신호를 계산한다. 그림 3의 (a)는 ELM 가산기의 4개의 기본 셀을 나타내고 (b)는 4비트 ELM 가산기의 구조이다. ELM과 BCL의 비교를 위해서 그림4의 (a)와 (b)는 BCL의 기본 셀 및 4비트 BCL을 나타낸다. BP는 Black Processor를 나타내며 각 비트의 블록 전파 신호 및 블록 발생 신호를 계산한다.^[6-7]

그림 3에서 보면 출력인 S_3, S_2, S_1, S_0 와 0비트에서 3비트까지의 블록 전파를 나타내는 블록 전파 신호 P_i , 그리고 4비트 블록 발생 신호인 G_3 가 있다.

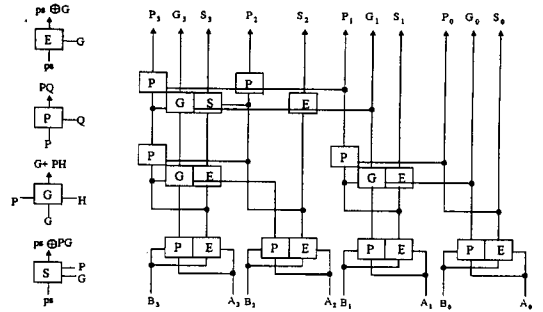


그림 3. 4비트 ELM 가산기의 구조
Fig. 3. A 4bit-ELM adder

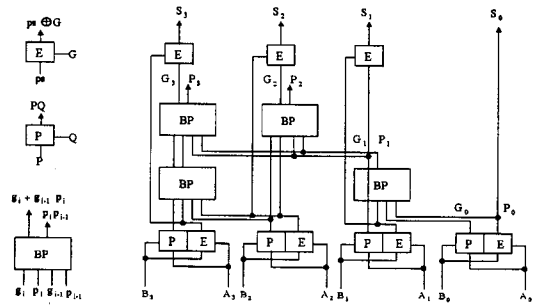


그림 4. 4비트 BCL 가산기의 구조
Fig. 4. A 4bit-BCL adder

P_i 는 0비트에서 3비트까지의 각 비트를 XOR 연산 후에 AND 연산을 한 것으로 다음 식(8)로 쓸 수 있으며 이는 우측 노드의 블록 발생 신호인 G_i 를 현재 비트의 합에 전달할지를 결정한다. 즉, P_i 가 '1'이면 현재 비트에 우측 노드의 G_i 가 전달되고, P_i 가 '0'이면 G_i 가 전달되지 않는다.

$$P_i = (A_i \oplus B_i) (A_{i-1} \oplus B_{i-1}) (A_{i-2} \oplus B_{i-2}) \dots (A_{i-3} \oplus B_{i-3}) \quad (i = 0, 3) \quad (8)$$

이처럼 4비트 ELM 가산기는 두 개의 4비트 입력을 받아들이며, 4비트의 부분 합, 4비트의 P_i , 그리고 1비트의 G_3 로 구성된다. 그림3과 그림4를 비교해 보면 공통적으로 이진 트리로 구성되어 있고 각 비트의 캐리를 parallel prefix computation 방식으로 계산한다.^[9] 반면에 차이점은 BCL은 BP셀로 모든 비트의 블록 발생 신호를 계산하는 반면에, ELM은 G셀로 각 노드에서 한 비트의 블록 발생 신호를 계산한다. 즉, BCL은 모든 비트의 블록 발생 신호인 G_0, G_1, G_2, G_3 를 모두 계산하지만, ELM은 각 노드의 블록 발생 신호인 G_0, G_1, G_3 를 계산한다.

따라서 ELM은 BCL에 비해 연결선수가 작고 면적이 작으며 속도가 빠르다.^[1-2] 실제로 그림에서 보면 셀 간의 연결선수가 BCL은 29이고 ELM은 24이다. 그림 5는 16비트 ELM 가산기의 구조를 나타낸 그림이다.

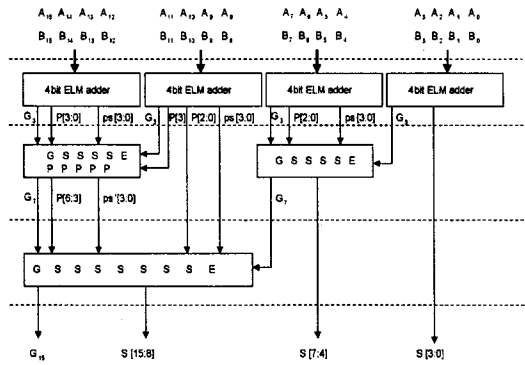


그림 5. 16비트 ELM 가산기의 구조
Fig. 5. 16bit-ELM adder.

3. 기본 셀의 설계

ELM 가산기를 구성하는 기본 셀에는 그림 3에서와 같이 P, E, G, S 등 4개의 셀이 존재한다. 회로적인 면에서 고속 및 저전력을 실현하기 위해서는 최적의 로직 스타일을 선택하여야 한다.^[3-4, 7, 10-13] 각 셀의 로직 스타일을 선택하는 방법에는 하나의 로직 스타일을 사용하는 방법이 있고, 여러 가지 로직 스타일을 혼합하여 사용하는 방법이 있다. 로직 스타일에는 크게 static CMOS, dynamic CMOS, 패스 트랜지스터 로직 등 3가지로 나눌 수 있다. Dynamic logic은 입력 캐패시턴스 값이 작고, 빠른 동작 속도를 갖는 반면에, 클럭에 의한 부하 및 높은 천이수 때문에 저전력 회로에는 적합하지 않다. 패스 트랜지스터 로직은 트랜지스터 수가 작고, 입력 쪽의 부하가 작은 장점을 갖는 반면에 레벨 복원을 위한 PMOS 및 출력단의 구동을 위한 버퍼링이 필요하다. 또한 dual rail logic으로 인한 연결도가 복잡해지고 이에 따른 캐패시턴스의 증가로 칩 면적 및 전력 소모의 증가가 따른다. 지연 시간 및 전력 소모 모두를 고려한 설계는 하나의 로직 스타일로 설계하는 것보다는 다양한 로직 스타일을 혼합한 형태가 바람직하다.^[13] 예를 들면, AND/OR 게이트는 static CMOS와 패스 트랜지스터 로직을 혼합한 형태가, NAND/NOR 게이트는 static CMOS가, XOR/XNOR 게이트는 Wang

게이트가 최적의 성능을 나타낸다.^[13]

그림 6~10은 기준 크기 셀로 구현한 4개의 기본 셀을 나타낸 그림이다. 트랜지스터 사이징은 람다규칙 (lambda based rule)으로 하였으며 $2\lambda = 0.8\mu\text{m}$ 이다.

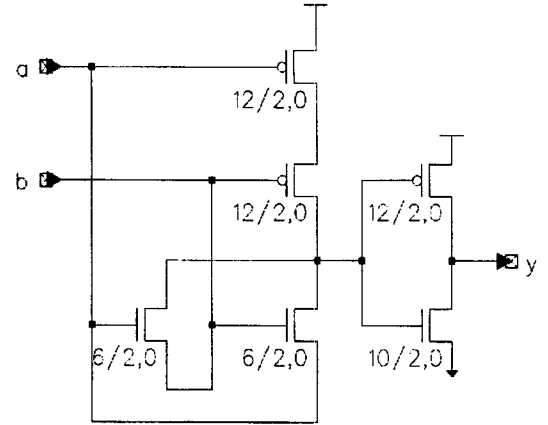


그림 6. 기준 크기 E 셀.
Fig. 6. Reference-sized E cell.

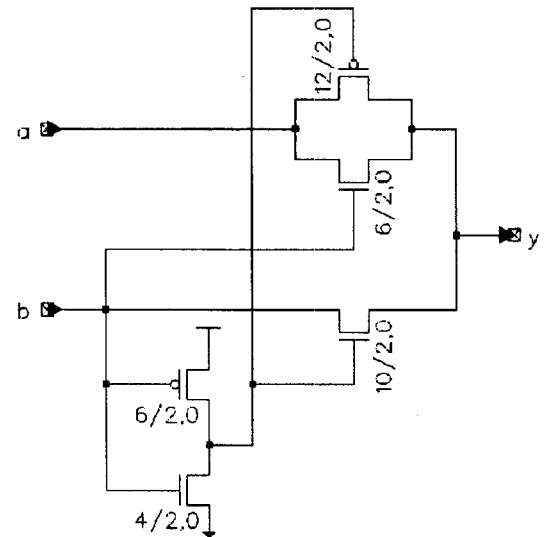


그림 7. 기준 크기 P 셀.
Fig. 7. Reference-sized P cell.

그림 6의 E셀은 1994년 Wang이 개발한 XOR 게이트로서 6개의 트랜지스터로 구동 능력이 있으면서 지연 시간이 작은 장점을 가진다.^[14] 반면에 입력 값이 '11'이 들어온 경우, 신호가 NMOS를 통과하므로 인버터 앞단에서 NMOS 문턱 전압만큼의 전압 강하가 일어나므로 이 입력이 들어오는 경우에 지연 시간이 길어지는 단점이 있다.

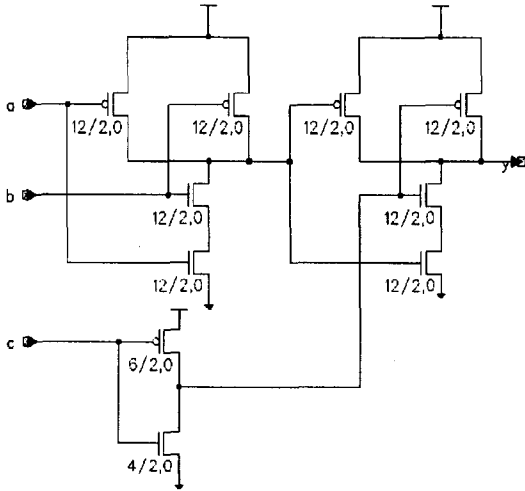


그림 8. 기준 크기 G 셀.
Fig. 8. Reference-sized G cell.

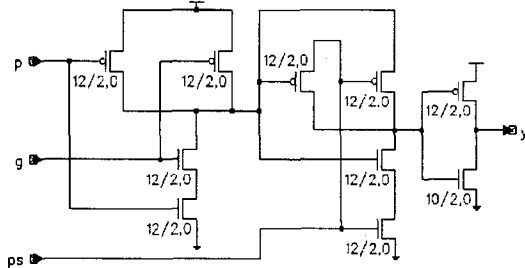


그림 9. 기준 크기 S1 셀.
Fig. 9. Reference-sized S1 cell.

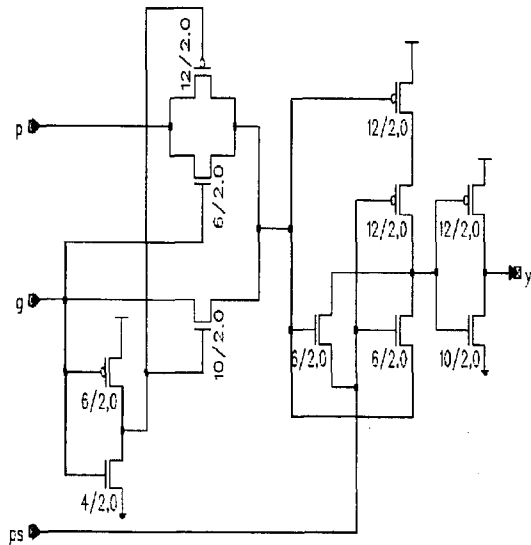


그림 10. 기준 크기 S2 셀.
Fig. 10. Reference-sized S2 cell.

반대로 Wang의 XNOR 게이트는 입력 값이 '00'가 들어온 경우에 지연 시간이 길어지는 단점이 있다. P 셀은 AND 게이트로서 그림 7과 같이 static CMOS와 패스 트랜지스터 로직을 혼합하여 구현하였다.^[4] 이 셀은 static CMOS로 구현한 것보다 전력 소모가 작고 지연 시간도 작다.^[13] 이러한 셀을 사용하는 경우 출력 단의 부하 캐패시턴스가 클 경우는 버퍼링이 필요하다.

G셀은 그림 8과 같이 AND-OR 함수로서 이것을 NAND-NAND 구조로 바꾸어 속도를 최소화하였다. G셀을 static CMOS로 구현한 이유는 G셀이 ELM 가산기의 속도를 결정하기 때문이다. S셀은 AND-XOR 함수로서 static CMOS와 Wang 게이트를 혼합하여 구성하였으며, 그림9, 10에서와 같이 두 개의 셀을 준비하여 마지막 레벨의 S셀은 그림10의 AND-XOR로, 각 레벨의 내부에 사용된 S셀은 그림 9의 NAND-XNOR를 사용하였다. 그 이유는 마지막 레벨의 각 비트의 합 발생을 위한 최악 조건의 입력을 주었을 경우, 각 내부 S셀의 입력 'P', 'G', 및 'ps'에는 '000'가 도달하고, 각 레벨의 마지막 S셀에는 '110'가 도달하게 되는데 마지막 레벨의 S셀을 NAND-XNOR를 사용하면, Wang의 XNOR게이트 입력 값이 '00'가 되어 지연 시간이 가장 크게 된다. 반면에 마지막 레벨의 S셀에 AND-XOR를 사용하면 동일한 입력 조건 하에서 Wang의 XOR게이트에 입력 값이 '10'가 되는데 이것은 최악조건이 아니므로 상대적으로 지연 시간이 작다. 내부의 S셀을 NAND-XNOR를 사용한 것은 전력 소모 면에서 NAND-XNOR가 AND-XOR보다 작기 때문이다.

III. 기준 크기 셀로 구현한 ELM 가산기

1. 모의실험 결과

앞에서 언급한 그림 6~그림 10의 E,P,G 및 S1, S2를 이용하여 16비트 ELM 가산기를 구현한 다음 모의실험 하였다. 기준 크기 셀이란 모든 노드에 동일한 크기의 셀로 구현한 것을 말한다. 공정 변수는 0.8 μ m LG 공정 변수를 이용하여 COMPASS에서 스키매틱을 그리고 HSPICE(level=13)를 이용하여 모의 실험을 하였다. 이 경우 모든 노드에 동일한 크기의 셀이 사용되었기 때문에 각 노드에서 우측 상단에 위치한 셀들의 논리 레벨(logic depth)이 작으므로 다른 경로

에 비해 지연 시간이 작다. 동작 전압은 5.0V, 부하는 50fF, 온도는 70℃, 표준 공정 변수를 이용하였다. 모든 비트의 합 및 캐리 출력 발생을 위해 서로 다른 16개의 입력 조건을 주고 모의 실험을 수행하였다. 실험 결과는 표 1과 같다.

표 1. 기준 크기 셀로 구현한 16비트 ELM 가산기의 모의 실험 결과
Table. 1. Simulation results of 16bit-ELM adder with reference-sized cells

level	sum	delay (ns)	Power dissipation (mW)	Input stimulus
0	S ₀	0.4611	0.5468	a: 0001H b: 0000H
1	S ₁	0.6908	0.6549	a: 0001H b: 0001H
2	S ₂	1.1955	1.3600	a: 0001H b: 0003H
	S ₃	1.4145	2.2440	a: 0001H b: 0007H
3	S ₄	1.8925	3.3630	a: 0001H b: 000FH
	S ₅	1.9940	4.6190	a: 0001H b: 001FH
	S ₆	2.1420	5.9170	a: 0001H b: 003FH
	S ₇	2.1390	7.2760	a: 0001H b: 007FH
4	S ₈	2.9915	9.0870	a: 0001H b: 00FFH
	S ₉	2.3610	10.17	a: 0001H b: 01FFH
	S ₁₀	2.5125	11.37	a: 0001H b: 03FFH
	S ₁₁	2.4955	12.71	a: 0001H b: 07FFH
	S ₁₂	2.6745	14.20	a: 0001H b: 0FFFH
	S ₁₃	2.5265	15.57	a: 0001H b: 1FFFH
	S ₁₄	2.6725	17.09	a: 0001H b: 3FFFFH
	S ₁₅	2.6415	18.65	a: 0001H b: 7FFFFH
	Cout	2.7760	20.33	a: 0001H b: FFFFFH

2. 최대 지연 경로 고찰

대부분의 가산기의 경우, 최대 지연 시간을 갖는 입력 조건은 캐리 입력이 모든 단을 거치면서 최종 출력

으로 전파되는 경우이다. ripple carry adder나 carry select adder, carry skip adder, conditional sum adder, carry lookahead adder등은 각 단을 단위로 또는 블록 단위로 캐리가 전파되는 구조이므로 캐리 출력의 지연시간이 가장 크다.^[12] 그런데 ELM 가산기의 경우는 다르다. 표1에서 보면 레벨4에서 S₈의 지연시간이 가장 큰 이유는 다음과 같다. 16비트의 경우, 최악조건 입력시 레벨3에서 G신호가 '1'이 발생되고, 마지막 단인 S셀 입력인 P,G신호에 '1'이 되며 ps는 '0'가 된다. 동시에 S₈을 출력시키는 E셀의 입력신호인 G신호는 '1', ps는 '0'가 된다. 그런데 여기서 주목할 것은 S셀 입력의 G신호와 P신호의 타이밍 관계이다. 두 신호가 AND 게이트를 통과하면서 원래의 G신호에 비해 하강시 지연 시간이 작아짐을 그림 11을 통해서 확인할 수 있다. 동일한 레벨에서 P신호의 지연 시간은 비트 수가 높을수록 증가한다. 그림 11은 S₁₅를 발생하는 S셀 입력의 G신호와 P신호의 파형을 나타낸다.

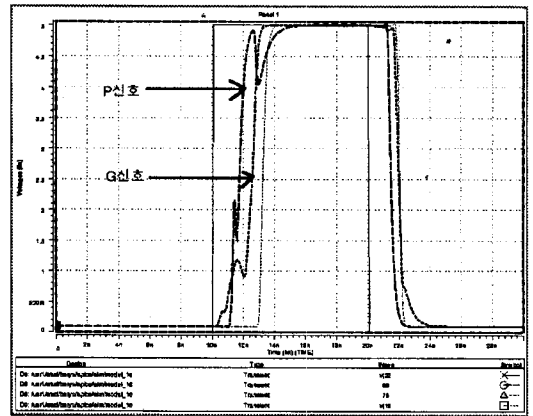


그림 11. S₁₅의 P, G 입력 파형
Fig. 11. P, G input waveform of S₁₅

그림 11에서 보면 P신호가 먼저 도착하고 G신호가 나중에 도착한다. 그 이유는 두 신호가 거치는 셀의 종류 및 수가 다르기 때문이다. 실제로 S₁₅의 경우 P 및 G신호의 경로를 살펴보면, G신호는 G셀 3개, 버퍼 2개를 통과하며 P신호는 P셀 4개, 버퍼 1개를 통과한다. G셀과 버퍼는 반전 레벨(inversion level)이 2이고 P셀은 반전 레벨이 1이므로 G신호는 10개의 반전 레벨을 갖는데 비하여, P신호는 6개의 반전 레벨을 갖는다. 따라서 이 두신호를 ANDing하면 원래의 G신호에 비해 하강시 지연 시간이 작다. 이 경우, S₈과

비교해 보면, S₈경우는 원래의 G신호가 그대로 XOR 게이트를 거쳐 출력에 나타나지만 S₉~S₁₅의 경우는 G신호와 각각의 P신호가 ANDing함으로써 원래의 G 신호에 비해 하강시 지연 시간이 작아지게 된다.

한편, 레벨2 및 레벨3에서 E셀 출력보다 S셀의 출력의 지연 시간이 큰 이유는 S셀을 NAND-NXOR를 사용했기 때문이다. 이 S셀을 사용하면 출력 비트를 '1'이 되게 하는 입력조건을 주었을 때, Wang의 XNOR게이트의 입력이 '00'가 되어 이때의 지연 시간이 E셀에서의 Wang의 XOR게이트 입력조건인 '01' 일때의 지연 시간보다 크다.

한편 S셀의 AND게이트 또는 NAND게이트를 static CMOS로 구현하면 늦게 도착하는 신호인 G신호를 출력에 가까운 위치에 연결하게 되면, 최대 지연 시간은 마지막 레벨의 MSB가 된다. 결론적으로 ELM 가산기의 최대 지연 시간은 S셀의 구현 방법 및 로직 스타일에 따라서 달라지게 된다. 본 연구에서는 마지막 레벨의 S셀을 AND-XOR구조로 구현하고 로직 스타일은 AND게이트는 static CMOS와 패스트 트랜지스터 로직을 혼합한 로직을, XOR 게이트는 Wang 게이트를 사용하였으므로 S₈의 지연 시간이 가장 크다.

IV. 가변 크기 셀로 구현한 ELM 가산기

기준 크기 셀을 이용한 실험 결과를 보면, 이 가산기의 속도는 임계경로(critical path)의 지연 시간에 의존하므로, 비임계경로(non-critical path)에 대해서는 지연 시간이 빠를 필요는 없다. 따라서 비임계경로에 대해서는 로직 게이트의 캐패시턴스를 줄이기 위한 방법으로 트랜지스터 다운 사이징이 하나의 효과적인 방법이다. ELM 가산기는 G신호가 임계경로이므로 이 신호에 영향을 주는 모든 셀들은 기준 크기 셀로 구현하고, 이 신호와 무관한 경로에 있는 하위 비트에 위치한 셀들은 기준 크기 셀보다 작은 트랜지스터로 구현하였다. 본 연구에서 이처럼 모든 노드에 동일한 크기의 셀을 사용하지 않고 레벨에 따라 트랜지스터 크기를 다르게 한 셀을 가변 크기 셀이라 부른다. 가변 크기 셀은 그림 12와 그림 13에서 보듯이 E 및 S셀이다. 트랜지스터 크기는 공정에서 허용하는 최소 크기로 하였다.

가변 크기 셀은 주로 각 노드의 우측 상단에 위치한 S 및 E셀이며 16비트의 경우 총 13개의 셀이다. 가변 크기 셀로 구현한 16비트 ELM 가산기의 실험 결과는 표 2와 같다.

표 2. 가변 크기 셀로 구현한 16비트 ELM 가산기의 모의 실험 결과
Table. 2. Simulation results of 16bit-ELM adder with variable-sized cells

level	sum	delay (ns)	Power dissipation (mW)	Input stimulus
0	S ₀	0.7085	0.2879	a: 0001H b: 0000H
1	S ₁	0.9194	0.4703	a: 0001H b: 0001H
2	S ₂	1.379	1.232	a: 0001H b: 0003H
	S ₃	1.9145	1.934	a: 0001H b: 0007H
3	S ₄	2.025	2.921	a: 0001H b: 000FH
	S ₅	2.489	3.884	a: 0001H b: 001FH
	S ₆	2.6075	4.902	a: 0001H b: 003FH
	S ₇	2.5945	5.917	a: 0001H b: 007FH
4	S ₈	2.910	7.534	a: 0001H b: 00FFH
	S ₉	2.33	8.605	a: 0001H b: 01FFH
	S ₁₀	2.4755	9.794	a: 0001H b: 03FFH
	S ₁₁	2.4495	11.10	a: 0001H b: 07FFH
	S ₁₂	2.6325	12.55	a: 0001H b: 0FFFH
	S ₁₃	2.4780	13.91	a: 0001H b: 1FFFH
	S ₁₄	2.6290	15.40	a: 0001H b: 3FFFH
	S ₁₅	2.5940	16.94	a: 0001H b: 7FFFH
	Cout	2.5155	18.60	a: 0001H b: FFFFH

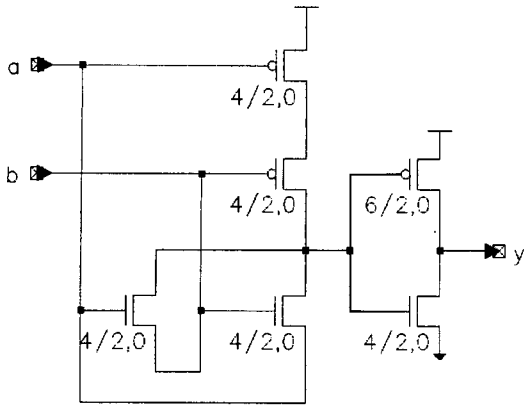


그림 12. 가변 크기 E 셀.
Fig. 12. Variable-sized E cell.

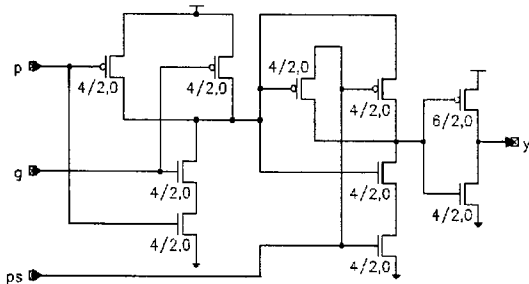


그림 13. 가변 크기 S 셀.
Fig. 13. Variable-sized S cell.

V. 결과 분석

표 3은 기준 크기 셀로 구현한 것과 가변 크기 셀로 구현한 16비트 ELM 가산기의 실험 결과를 비교한 것이다.

표 3. 두 가산기의 성능 비교
Table. 3. Performance comparison between different ELM adders

출력	구분	delay (ns)	power (mW)	PDP (norm.)	PDP reduction (%)
s[8]	Variable-sized cell	2.910	7.534	1.00	19.3
	Reference-sized cell	2.9915	9.087	1.24	
cout	Variable-sized cell	2.5155	18.60	1.00	17.1
	Reference-sized cell	2.776	20.33	1.21	

최대 지연 시간을 갖는 S_8 은 power-delay-product 면에서 19.3%의 향상을 가져왔고, 최대 전력 소모를 갖는 캐리 출력은 17.1%의 향상을 가져왔다. 가변 크기 셀을 이용한 회로의 지연 시간이 감소된 이유는 트랜지스터의 크기가 작아지면서 게이트 및 확산 캐패시턴스 값이 작아져 임계경로의 부하 캐패시턴스 값이 작아졌기 때문이다.

VI. 결론

현재까지 가장 속도가 빠르고 power-delay-product가 적은 ELM 가산기를 다양한 로직 스타일로 기본 셀을 구성하고, 각각 기준 크기 셀 및 가변 크기 셀을 이용하여 16비트 가산기를 구현한 다음, 모의실험을 하였다. 로직 스타일은 AND게이트는 static CMOS와 패스 트랜지스터 로직을 혼합한 회로를, XOR/XNOR게이트는 Wang 게이트, 그 밖의 게이트는 static CMOS 등 다양한 로직 스타일을 혼합하여 구현하였다. ELM 가산기는 각 비트의 합의 지연 시간이 상위 레벨로 올라갈수록 증가하므로 비임계경로인 하위 레벨의 트랜지스터를 다운 사이징하여 동작 속도를 개선시키고 전력 소모를 줄였다.

모의 실험 결과, 16비트의 경우 S_8 의 발생 입력 조건에 대해서 가변 크기 셀을 이용한 것이 기준 크기 셀을 이용한 것 보다 power-delay-product면에서 19.3%의 향상을 가져왔다. 비트 수가 증가할수록 다운 사이징 트랜지스터 수가 증가하므로 ELM 가산기의 속도저하 없이 더 많은 전력 소모의 절감을 기대할 수 있다. 또한 ELM 가산기에서의 최대 지연 경로는 고정되어 있는 것이 아니고 S셀의 구현 방법 및 로직 스타일에 따라 달라질 수 있다. 설계된 가산기는 마이크로 프로세서나 DSP의 ALU 블록에 응용이 가능하리라 생각된다.

감사의 글

※ 이 논문은 반도체 설계 교육 센터로부터의 부분적인 지원을 받아 이루어졌습니다.

참고 문헌

[1] T. P. Kelliher, R. M. Owens, M. J. Irwin,

and T. T. Hwang, "ELM-A Fast Addition Algorithm Discovered By a Program" IEEE Transactions on Computers, vol. 41, no. 9, Sep. 1992.

[2] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-Time-Power Tradeoffs in Parallel Adders" IEEE Transactions on Circuits and Systems, vol. 43, no. 10, Oct. 1996.

[3] Anantha P. Chandrakasam, Robert W. Brodersen, "Low Power Digital CMOS Design", Kluwer academic publishers, pp 55-102, 1995.

[4] 최준립, 허원준, 장락현, "Combinative Complementary Pass Transistor Logic을 이용한 Datapath 산술 연산기의 설계," 1995년도 ASIC DESIGN WORKSHOP, pp. I.27-I.40, 1995

[5] Uming Ko, Poras T. Balsara, and Wai Lee, "Low-Power Design Techniques for High-Performance CMOS Adders" IEEE Transactions on VLSI systems., vol. 3, no. 2, Jun. 1995.

[6] R. P. Brent and H. T. Kung, "A regular layout for parallel adder," IEEE Trans. Comput., vol. C-31, pp. 260-264, Mar. 1982

[7] Gustavo A. Ruiz, "Evaluation of Three 32-Bit CMOS Adders in DCVS Logic for Self-Timed Circuits" IEEE J. of Solid-state Circuits, vol. 33, no. 4, April, 1998.

[8] S. Waser, M. J. Flynn "Introduction to Arithmetic for Digital systems Designers". New York: Holt, Rinehart and Winston, 1982.

[9] R. E. Ladner and M. J. Fischer, "Parallel prefix computation," J. ACM, vol. 27, no. 4, pp. 831-838, Oct. 1980.

[10] Abdellatif Bellaouar, Mohamed I. Elmasry, "Low Power Digital VLSI Design", Kluwer academic publishers, pp 176-187, 1995.

[11] Anantha P. Chandrakasam, Robert W. Brodersen, "Low Power Digital CMOS Design", Kluwer academic publishers, pp 55-102, 1995.

[12] Jan M. Rabaey, Massoud Pedraw, "Low Power Design Methodologies", Kluwer academic publishers, pp. 37-62, 1996.

[13] 조태원, 김문수, "Hybrid 로직을 이용한 저전력 8bit ELM 덧셈기 설계", 제2회 IDEC MPW 논문집, pp. 13-17, Jan. 1998

[14] J. M. Wang, S. C. Fang, and W. S. Feng, "New Efficient Designs for XOR and XNOR Functions on the Transistor level" IEEE J. of Solid-state Circuits, vol. 29, no. 7, July, 1994.

저 자 소 개



柳 凡 善(正會員)

1968년 8월 24일생. 1991년 2월 충북대학교 전자공학과 졸업(공학사). 1997년 2월 충북대학교 대학원 전자공학과 졸업(공학석사). 1997년 3월 ~ 현재 충북대학교 대학원 전자공학과 박사과정. 1991년 1월 ~ 1992년 12월 아남반도체기술(주) 설계부. 1993년 1월 ~ 1994년 12월 대양상사. 주관심분야는 저전력 회로 설계, DSP core 설계



趙 泰 元(正會員)

1950년 2월 17일생. 1973년 2월 서울대학교 전자공학과 졸업(공학사). 1986년 5월 미 루이빌대 전자공학과 졸업(공학석사). 1992년 5월 미 켄터키 주립대 전자공학과 졸업(공학박사). 1973년 8월 ~ 1983년 10월 금성전선(주). 1977년 1월 ~ 1977년 3월 영국 및 프랑스의 ITT 계열사 연수. 1992년 9월 ~ 현재 충북대학교 전기전자공학부 부교수. 주관심분야는 집적회로 설계, 컴퓨터 구조, 저전력 회로 설계, DSP core 설계