論文98-35C-6-1

하이브리드 로직 스타일을 이용한 저전력 ELM 덧셈기 설계

(A Design of Low Power ELM Adder with Hybrid Logic Style)

金 文 洙 *, 柳 凡 善 *, 姜 成 現 *, 李 重 錫 *, 趙 泰 元 *

(Moon-Soo Kim, Beom-Seon Ryu, Sung-Hyun Gang, Jung-Sok Yi, and Tae-Won Cho)

요 약

본 논문에서는 동일 칩 내부에 static CMOS와 하이브리드 로직 스타일(hybrid logic style)을 이용하여 저전력 8비트 ELM 덧셈기를 설계하였다. 두 개의 로직 스타일로 설계된 8비트 ELM 덧셈기는 0.8㎛ 단일 폴리 이중 금속, LG CMOS 공정으로 설계되어 측정되었다. 하이브리드 로직 스타일은 CCPL(Combinative Complementary Pass-transistor Logic), Wang's XOR 게이트와 ELM 덧셈기의 속도를 결정하는 임계 경로(critical path)를 위한 static CMOS 등으로 구성된다. 칩 측정 결과, 전원 전압 5.0V에서 하이브리드 로직으로 구현한 ELM 덧셈기가 static CMOS로 구현한 덧셈기에 비해 각각 전력소모 면에서 9.29%, 지연 시간 면에서 14.9%, PDP(Power Delay Product)면에서 22.8%의 향상을 얻었다.

Abstract

In this paper, we designed a low power 8bit ELM adder with static CMOS and hybrid logic styles on a chip. The designed 8bit ELM adder with both logic styles was fabricated in a 0.8㎛ single-poly double-metal, LG CMOS process and tested. Hybrid logic style consists of CCPL(Combinative Complementary Pass-transistor Logic), Wang's XOR gate and static CMOS for critical path which determines the speed of ELM adder. As a result of chip test, the ELM adder with hybrid logic style is superior to the one with static CMOS by 9.29% in power consumption, 14.9% in delay time and 22.8% in PDP(Power Delay Product) at 5.0V supply voltage, respectively.

I. 서 론

최근 전자기기들의 소형화 추세에 따라 저전력으로 동작할 수 있는 구조들이 연구되고 있다. 그중 덧셈기는 모든 기기들에서 사용되는 가장 보편적인 장치라 할 수 있다. ELM 덧셈 알고리즘은 1992년 펜실버니아 주립대학에서 개발한 VLSI CAD Tool의 하나인 FACTOR에 의해서 개발된 것으로써 이진 트리 구조에서 각각의 합이 병렬적으로 직접 계산되어 셀 간의 연결선의 수가 적고, 면적이 작고 속도가 빠르며, 셀이 규칙적으로 배열되어 레이 아웃이 용이한 장점이 있다[1, 2]. ELM이란 용어는 어떤 용어의 약자는 아니며 덧셈 알고리즘 및 덧셈기 명칭을 ELM이라 한다[1]. 이 덧셈기는 여러 종류의 병렬 덧셈기 중에서 면적, 속도, 전력소모 등을 고려했을 때 모든 비트 수에 걸쳐서 가장 성능이 우수한 것으로 알려졌다[2].

초기 디지탈 회로를 구성하는 기본 로직은 CMOS가 주류를 이루어왔으나, 최근에는 그보다 개선된 새로운 로직들이 연구되어 왔다[3-5]. 본 ELM 덧셈기

* 正會員, 忠北大學校 電氣電子工學部

(Dept. of Electronics and Electrical Eng., Chung-pok Univ.)

接受日字:1998年4月23日, 수정완료일:1998年5月26日

에 사용된 주된 로직은 단락 회로 전류(short-circuit current)가 적고, single-rail 로직인 CCPL을 주로 사용하되[6], 덧셈기에서 높은 비중을 차지하는 XOR 연산은 CCPL보다 PDP가 우수한 Wang이 제안한 회로를 사용하였고[7] 전체 ELM 덧셈기의 속도를 결정짓는 임계 경로에는 속도가 빠른 static CMOS 등, 다양한 로직 스타일을 혼합한 하이브리드 로직을 사용하였다. CCPL은 단락 회로 전류가 적어 전력소모가 적으나 로직 체인(logic chain)이 길어질 수록 신호의 지연 시간이 증가되어 적절한 버퍼링이 필요하다. 따라서 덧셈기를 구성한 각 셀의 기본 로직에서 임계 경로는 속도가 빠른 static CMOS를 사용하였다.

하이브리드 로직으로 구현된 ELM 덧셈기의 성능을 측정하기 위해서 하나의 칩에 8비트의 ELM 덧셈기를 각각 static CMOS 및 하이브리드 로직으로 구현하여 제작한 다음 지연 시간과 전력소모를 측정, 비교하였다.

본 논문의 2절은 ELM 덧셈기의 알고리즘 및 구조에 대해서 설명한다. 3절에서는 ELM 덧셈기를 구성하는 기본 셀들의 구현 방법 및 기본 셀을 구현하는 다양한 로직 스타일의 성능을 비교한다. 4절에서는 8비트 ELM 덧셈기를 3가지 로직 스타일로 구현 및 비교한다. 5절에서는 레이 아웃 및 칩 제작을 논하고 6절에서는 측정 결과를 보이고 7절에서는 연구에 대한 결론을 맺는다.

II. ELM 덧셈기 구조

ELM 알고리즘은 1992년 미국 펜실버니아 대학에서 개발된 VLSI CAD tool의 하나인 FACTOR라는 multi-synthesis tool에 의해 개발된 것으로 셀 간의 연결횟수를 최소화한 알고리즘이다[1]. 이 알고리즘은 1982년 Brent와 Kung에 의해 개발된 B & K 알고리즘과 약간의 차이가 있다[8]. B & K 알고리즘은 기존의 덧셈기의 설계 기준이 되었던 게이트 수(gate count) 보다는 설계 규칙성(design regularity)과 칩 면적(chip area)을 고려하여 제시한 알고리즘으로써 각 비트의 캐리를 parallel prefix computation방식으로 계산하였다.

이 알고리즘은 임의의 노드에 대해서 모든 비트의 블록 전파 신호(block propagation signal) 및 블록 발생 신호(block generation signal)를 계산하여 얻어진 블록 발생 신호가 i번째 비트의 캐리가 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 신호를 발생시키면서 나머지 비트의 블록 발생 신호들은 이전 레벨의 부분 합에 반영된다. 따라서 이 알고리즘의 장점은 B & K에 비해 셀 간의 연결선이 적다는 점이다. 표 1에서 B & K와 ELM 알고리즘을 요약하였다.

표 1. B & K 과 ELM 덧셈 알고리즘의 비교
Table 1. Comparison of B & K and ELM addition algorithm.

항목	B & K	ELM
공통점	-. CLA(Carry Lookahead Adder)의 변형 -. 규칙적인 레이 아웃 -. Parallel prefix computation방식으로 블록 발생 신호를 계산함	
차이점	-. 임의의 노드에 대해 모든 비트에 해당하는 블록 발생 신호 및 블록 전파 신호를 계산함	-. 임의의 노드에 대해 모든 비트에 해당하는 블록 전파 신호 및 부분합과 한 비트의 블록 발생 신호를 계산함
상대적인 장점		-. 셀 간의 연결선의 최소화로 속도 및 전력 소모 개선

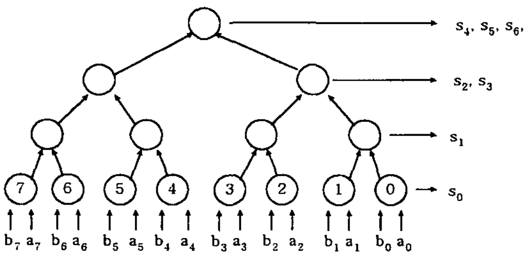


그림 1. 8비트 ELM 블록도
Fig. 1. ELM block diagram for 8bit.

이진 트리 구조의 알고리즘은 $O(\log n)$의 계산 시간이 필요하고, 전체 덧셈기를 구현하기 위해서 레이 아웃될 때에는 $O(n \cdot \log n)$의 면적이 요구된다.

ELM 덧셈기의 계산 알고리즘을 간단히 설명하면 다음과 같다. 올림수 예견 덧셈기(Carry Lookahead

Adder)에도 사용된 generate signal(g_i)과 propagate signal(p_i)의 식은 (1)과 같이 정의된다.

$$g_i = a_i b_i, \quad p_i = a_i \oplus b_i \quad \text{(단 i는 비트 위치를 나타냄)} \quad (1)$$

그리고, $0 \le i < j \le n-1$에 대해서, 식 (2), (3)을 정의할 수 있다[1].

$$P(j, i) = p_j P(j-1, i), \quad P(i, i) = p_i \quad (2)$$

$$G(j, i) = g_j + p_j G(j-1, i), \quad G(i, i) = g_i \quad (3)$$

즉, $P(j, i)$는 i번째 이전에 발생된 캐리(carry)가 j번째 비트까지 전파 될 수 있는 지의 여부를 판단하는 블록 전파 신호이고, $G(j, i)$는 i번째 비트에서 j번째 까지의 블록 발생 신호를 나타낸다.

그림 2는 이진 트리 구조를 취하는 ELM 덧셈 알고리즘에서 임의의 노드에서 계산되어지는 값들을 정리한 것이다.

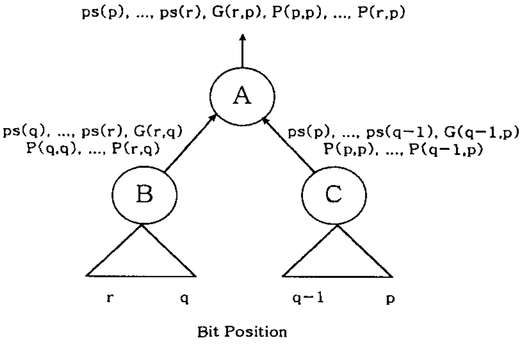


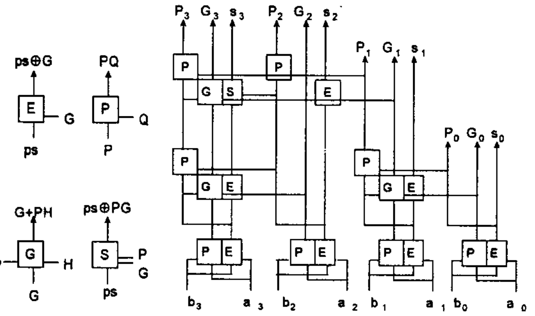
그림 2. 각 노드에서 계산되는 신호
Fig. 2. Computed signals at each node.

여기서 A노드의 최종적인 부분 합은 식 (4)로 나타낼 수 있다.

$$ps^A_{q+j} = ps^B_{q+j} \oplus [P^B(q+j-1, q) G^C(q-1, p)] \quad (4)$$

위첨자 A, B, C는 노드를 나타낸다. 이때 A노드에서의 $q+j$번째 비트의 부분 합(ps^A_{q+j})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^B_{q+j})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 발생되는 블록 발생 신호와 B노드에서 발생되는 블록 전파 신호의 AND 신호에 의해 결정된다.

이때 식 (2) (3) (4)의 연산을 위해서 AND, OR, XOR의 기본 연산으로 구성되는 4개의 기본 셀이 필요하다. 각 셀의 이름은 E셀, G셀, P셀 및 S셀이고, 각 셀이 수행하는 연산은 각각 E = ps⊕G, G = G+PH, P = PQ, S = ps⊕PG이다. E 및 S셀은 부분 합을 계산하고 G셀은 블록발생신호를, P셀은 블록 전파신호를 계산한다. 그림3의 (a)는 4개의 기본 셀을 나타내고 (b)는 4비트 ELM 덧셈기의 구조이다.



(a) E, P, G 및 S 셀 (b) 덧셈기의 구조

그림 3. 4비트 ELM 덧셈기의 구조
Fig. 3. 4bit ELM adder.

Ⅲ. 로직 스타일

저전력 회로 구현을 위해 회로레벨에서는 적절한 로직 스타일의 선택이 필요하다. static CMOS는 안정성 및 논리함수의 구현이 용이한 반면에 출력 노드가 스위칭 할 때 단락 회로 전류가 흐르므로 전력소모가 큰 단점이 있다. 한편 CPL(Complementary Pass-transistor Logic)로 대표되는 패스 트랜지스터 로직들은 단락 회로 전류가 적고 XOR 게이트나 멀티플렉서 등을 효율적으로 구현할 수 있으나 dual rail logic이므로 연결선이 복잡하고 이에 따른 전력소모가 큰 단점이 있다[3, 4, 9]. CCPL은 패스 트랜지스터 로직이면서 single rail logic이므로 전력소모가 적은 장점이 있다. 반면에 CCPL은 로직 체인이 길어짐에 따라서 적절한 버퍼링이 필요하여 지연 시간이 큰 단점이 있다. 한편 XOR 게이트는 1994년 Wang이 CCPL을 개선한 회로를 개발하였다[7]. 이 회로는 입력 단에 상보 입력이 필요하지 않고 6개의 적은 트랜지스터로 구성되면서도 출력 단에 구동력을 가지는 장점이 있다. Wang의 XOR게이트는 static CMOS의 XOR 게이

트에 비해 동일한 구동력을 가지면서 트랜지스터 수가 작아 전력 소모가 작고 반전 레벨(inversion level)이 작아 지연 시간이 작은 장점이 있다.

따라서 전력소모 및 속도를 모두 고려한 로직 스타일은 주된 로직은 CCPL을 사용하고 XOR게이트는 Wang 게이트를, 그리고 임계 경로에 대해서는 static CMOS 등 다양한 로직을 혼합한 하이브리드 로직 스타일이 최적이라 할 수 있다.

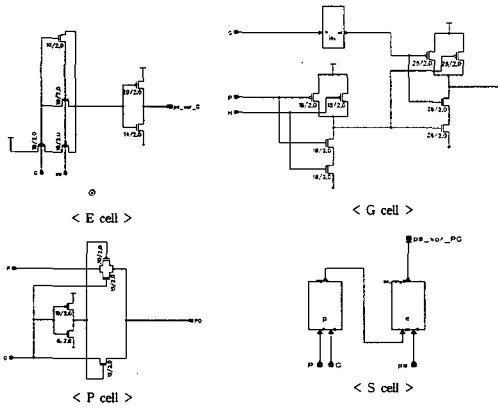


그림 4. 하이브리드 로직으로 구현한 기본 셀들
Fig. 4. Basic cells with hybrid logic.

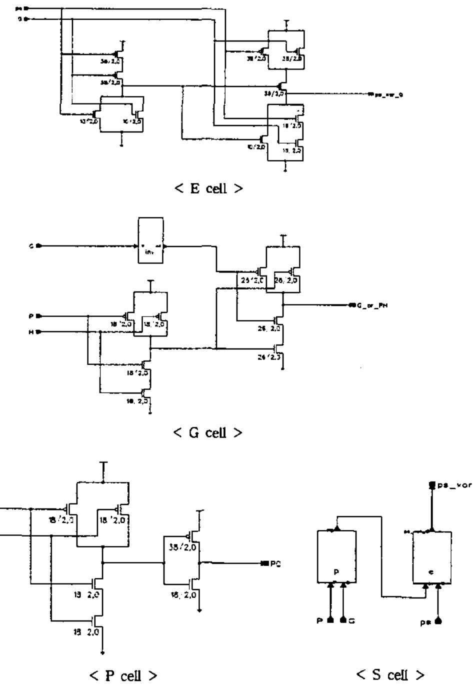


그림 5. static CMOS으로 구현한 기본 셀들
Fig. 5. Basic cells with static CMOS.

그림 4는 4개의 기본 셀을 하이브리드 로직으로 구현한 그림이다. E셀은 Wang 게이트, P 및 S셀은 CCPL, 그리고 전체 덧셈기의 지연시간을 좌우하는 G셀은 static CMOS로 구현하였다. 그림에 표시된 크기는 람다(λ) 단위로써 2λ=0.8㎛이다.

성능 비교를 위해 그림 5는 static CMOS로 4개의 기본 셀을 구현한 회로도이다.

다음 표 2는 4개의 기본 셀에 대해 static CMOS와 CCPL 및 Wang 게이트의 지연 시간 및 전력소모를 비교한 표이다. 모의실험 조건은 동작전압 5.0V, 온도 25℃, 부하 캐패시턴스는 500fF으로 놓고, LG 0.8㎛ 전형적인 공정 변수를 이용하였다.

표 2. 로직 형태에 따른 셀의 성능 비교
Table 2. Comparison of cells with different logic styles.

Cell	Logic Styles	Delay (ns)	Rising Time (ns)	Falling Time (ns)	Power (㎽)	PDP (pJ)
E	Wang	1.08	1.99	1.68	1.14	1.23
	static CMOS	1.13	2.12	2.00	1.27	1.44
G	CCPL	0.76	1.35	1.92	1.65	1.25
	static CMOS	0.74	1.19	1.18	1.75	1.30
P	CCPL	0.66	1.91	1.60	0.88	0.59
	static CMOS	0.74	1.07	1.08	1.08	0.80
S	CCPL	1.11	1.40	1.38	1.58	1.75
	static CMOS	1.19	1.44	1.99	1.86	2.21

표 2에서 보면 E, P 및 S셀의 경우에는 Wang과 CCPL이 static CMOS에 비해 지연 시간과 전력소모가 적었고, G셀의 경우에는 static CMOS가 전력소모가 많은 반면에 지연 시간이 작았다. 그런데 ELM 덧셈기 구조에서 속도를 좌우하는 셀은 블록 발생 신호를 계산하는 G셀이다. 저전력 회로의 목표는 기존의 동작속도를 유지하면서 전력소모를 최소화하는 것이 가장 기본적인 목표이다. 따라서 G셀은 전체 덧셈기의 속도를 고려하여 전력소모가 적은 CCPL보다는 지연 시간이 적은 static CMOS가 효과적이다. 즉 하이브리드 로직의 ELM 덧셈기는 E셀은 Wang 게이트, P 및 S셀은 CCPL, G셀은 static CMOS로 구성할 수 있다.

Ⅳ. 8비트 ELM 덧셈기 설계

앞 절에서 결정한 기본 셀을 이용하여 8비트 ELM

덧셈기를 설계하고 이들의 성능을 비교하였다. 그림 6은 8비트 ELM 덧셈기의 셀 위치를 중심으로 나타낸 구조이다.

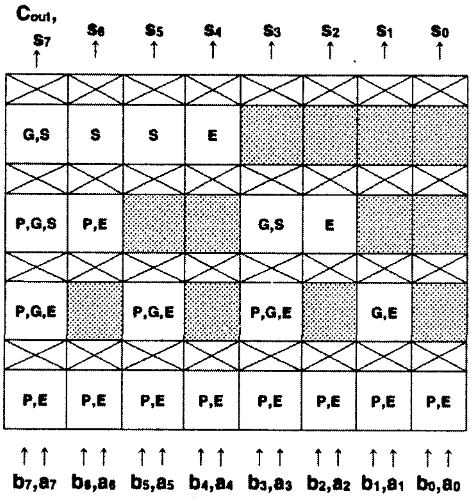
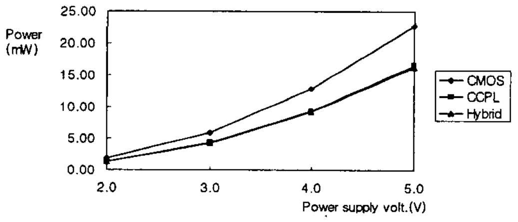
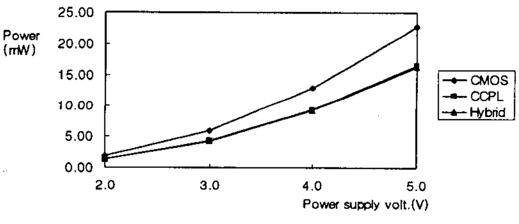


그림 6. 8비트 ELM 덧셈기의 구조
Fig. 6. 8bit ELM adder.

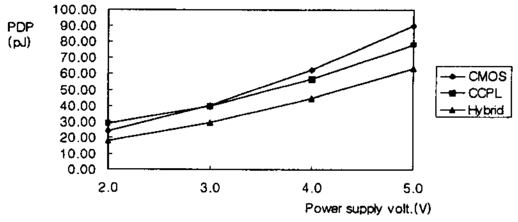
ELM 덧셈기는 각 비트 위치에서 반복적으로 구성된 셀 들의 조합으로 구성된다. ELM 덧셈기의 장점 중의 하나는 레이 아웃 형태를 회로도와 동일하게 그릴수 있다는 점이다. 따라서 전단계 설계시 보다 정확하게 레이 아웃 이후의 영향을 고려한 설계가 가능하다. 지연 시간 계산을 위한 입력 패턴은 '7FH'과 '01H'으로 하였다.



(a) Comparison of delay time



(b) Comparison of power consumption



(c) Comparison of PDP

그림 7. 다양한 로직 스타일의 성능 비교
Fig. 7. Performance comparison of various logic styles.

ELM 덧셈기는 G와 S셀 중 S셀의 지연 시간이 크기 때문에 s_7의 지연 시간이 가장 크다. 그림 7은 8비트 덧셈기에 대한 static CMOS, CCPL 및 하이브리드 로직 등 3가지 로직 스타일의 지연 시간, 전력 소모 및 PDP를 비교한 것이다.

모의 실험 결과는 pre-layout simulation을 수행한 결과이다. 모의 실험 결과, static CMOS는 CCPL과 비교하여 지연시간이 적은 반면에 전력소모가 많았다. 한편 하이브리드 로직이 다른 두 로직에 비해서 지연 시간 및 전력소모가 모두 우수하였다.

V. 레이 아웃

하이브리드 로직으로 구현된 ELM 덧셈기의 성능을 평가하기 위해서 ELM 덧셈기를 하이브리드 로직과 CMOS로 각각 구분하여 설계하였다.

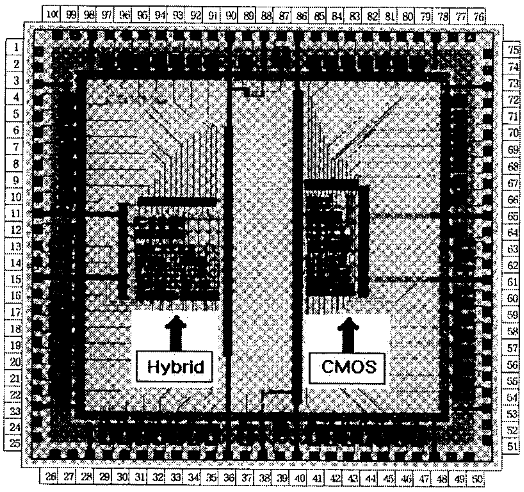


그림 8. 8비트 ELM 덧셈기의 레이아웃
Fig. 8. Layout of 8bit ELM adder.

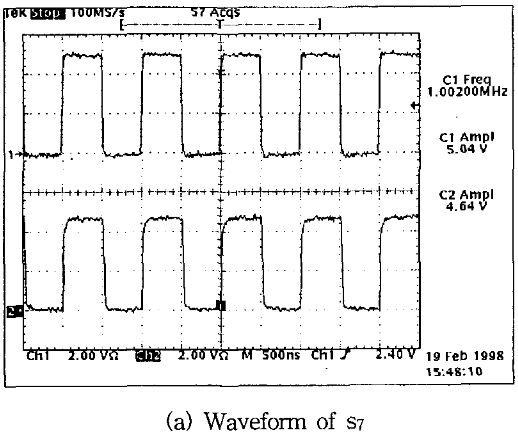
두 로직의 전력소모를 측정하기 위해 코어에 공급하는 Vdd와 Vss를 분리하여 공급하였다. 그림 8은 설계한 8비트 ELM 덧셈기의 전체 레이 아웃 도면이다.

전체 칩의 크기는 5×5㎟이고, 도면중 코어의 왼쪽에 위치한 것이 하이브리드 로직으로 구현된 8비트 ELM 덧셈기이고, 오른쪽의 것이 static CMOS로 구현된 것이다. 트랜지스터 수는 static CMOS가 492개, 하이브리드가 387개 였고 면적은 하이브리드 로직으로 구현한 것은 0.96㎟로, static CMOS로 구현한 것의 0.75㎟에 비해 21%의 면적이 증가되었다. 하이브리드가 트랜지스터 수가 적으면서도 면적이 증가한 것은 최적의 레이 아웃이 이루어지지 않았기 때문이다.

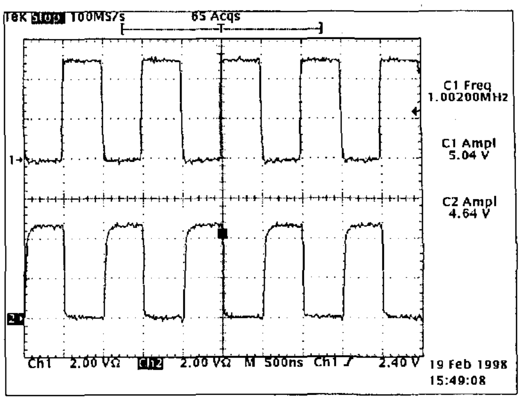
VI. 측정 결과

전력소모와 지연 시간을 측정하기 위해서 다음과 같이 입력 패턴을 설정했다. 하나의 입력은 '7FH'로 하고 다른 하나의 15비트 입력은 '0'로 한 다음, LSB를 1㎒의 구형파 발생기에 연결한다. 이 경우 입력은 '7FH', '00H'와 '7FH', '01H'를 반복하게 되고, 출력은 각각의 입력에 따라서 '7FH'와 '10H'를 반복하게 된다. 따라서 최대 지연 시간을 갖는 s_7의 출력을 관찰할 수 있다. 그림 9는 s_7과 s_6의 파형을 오실스코프로 본 것이다. s_7이 상승할 때 s_6는 하강하는 것을 통해서 정상적으로 동작함을 확인할 수 있다.

그림 10은 동작전압 5.0V에서 하이브리드 로직으로 구현한 ELM 덧셈기의 지연 시간을 측정하기 위해서 오실로스코프로 입력 파형과 s_7 파형을 관찰한 것이다. 15.42ns의 지연 시간이 생김을 관찰할 수 있다.



(a) Waveform of s_7



(b) Waveform of s_6

그림 9. 하이브리드 로직으로 구현한 8비트 ELM 덧셈기의 출력 파형

Fig. 9. Output waveform of 8bit ELM adder with hybrid logic.

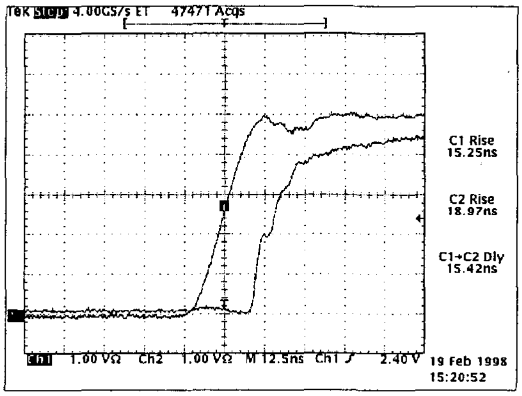


그림 10. 지연 시간 측정

Fig. 10. Measure of delay time.

전력소모 측정은 입력신호를 계속적으로 인가하면서 그때의 Vdd와 Vss사이에 흐르는 전류를 측정한 다음 동작전압을 곱하여 계산하였다. 표 3은 칩 시험결과를 나타낸 표이다. 동작전압 5.0V에서 그림7의 모의실험 결과와 제작 칩과의 지연 시간의 차이가 큰 이유는 인가되는 입력신호의 상승/하강시간의 차이가 크기 때문이다. 모의 실험에서는 입력신호의 상승/하강시간을 0.1ns로 하였으나 실제 시험에서는 그림 10에서 보는 바와 같이 입력신호의 상승/하강시간이 15.25ns였다. 최대 동작 주파수는 12㎒까지 동작하였으나 PAD 설계상의 문제로 그 이상은 동작하지 않았다.

그림 11은 시험 결과를 토대로 PDP를 비교한 것이다. 모든 전압 범위에서 하이브리드 로직으로 구현한

표 3. 8비트 ELM 덧셈기의 시험 결과
Table 3. Test result of 8bit ELM adders.

Vdd(V)			5.0			4.0			3.0			2.0		
logic styles	# of TRs	Area (㎟)	delay (ns)	pwr (㎽)	PDP (pJ)	delay (ns)	pwr (㎽)	PDP (pJ)	delay (ns)	pwr (㎽)	PDP (pJ)	delay (ns)	pwr (㎽)	PDP (pJ)
static CMOS	492	0.75	18.12	7.97	144.42	24.57	4.26	104.67	39.70	1.95	77.42	106.8	0.75	80.10
Hybrid	387	0.96	15.42	7.23	111.49	18.12	3.83	69.40	27.90	1.69	47.15	63.00	0.62	39.06

ELM 덧셈기가 우수한 PDP를 나타냈다. 동작 전압 5.0V에서 하이브리드 로직이 static CMOS에 비해 전력소모는 9.29%, 지연 시간은 14.9%, PDP는 22.8%가 향상되었다. 5.0V에서 하이브리드 로직의 지연 시간이 static CMOS보다 작은 이유는 임계 경로에 연결되어 있는 S 및 E셀의 입력 캐패시턴스 값이 하이브리드 로직이 static CMOS보다 작기 때문이다.

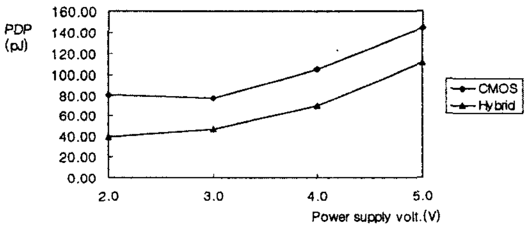


그림 11. 8비트 ELM 덧셈기의 시험 결과
Fig. 11. Test result of 8bit ELM adder.

Ⅶ. 결 론

저전력 소모를 위한 다양한 방법 중에서 회로 레벨에서 접근할 수 있는 방법중의 하나는 최적의 로직 스타일을 선택하는 것이다. 패스 트랜지스터 계열의 로직 스타일은 static CMOS에 비교하여 단락 회로 전류가 적기 때문에 저전력 회로에 적합하다. 여러 종류의 패스 트랜지스터 논리 중에서 CCPL은 CPL계의 회로에 비교하여 single-rail 로직이므로 연결 복잡도 및 전력 소모가 적다. 반면 CCPL은 static CMOS에 비해서 지연 시간이 큰 단점이 있다. 따라서, 임계 경로에 대해서는 static CMOS를, 다른 부분은 CCPL을 혼합한 하이브리드 로직이 속도 및 전력 소모를 모두 고려하여 가장 좋은 성능을 나타낸다. 8비트 ELM 덧셈기를 0.8㎛ LG CMOS 공정을 이용하여 static CMOS 및 하이브리드 로직으로 설계, 제작 및 시험한 결과, 동작 전압 5.0V에서 하이브리드 로직이 static CMOS에 비해 면적의 증가는 있었으나 속도 면에서 14.9%, 전력소모 면에서 9.29%, PDP면에서 22.8%의 개선을 얻었다. 하이브리드 로직으로 구현한 ELM 덧셈기는 앞으로 저전력 마이크로 프로세서나 DSP의 ALU블록에 적용이 가능하리라 생각된다.

감사의 글

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

참 고 문 헌

[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, Sept. 1992.

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

[3] A. Bellaouar and M. I. Elmasry, LOW-POWER DIGITAL VLSI DESIGN CIRCUITS AND SYSTEMS, Kluwer Academic Publishers, 1995.

[4] A. P. Chandrakasan and R. W. Brodersen, LOW POWER DIGITAL CMOS DESIGN, Kluwer Academic Publishers, 1995.

[5] J. M. Rabaey and M. Pedram, LOW POWER DESIGN METHO- DOLOGIES, Kluwer Academic Publishers, 1995.

[6] 최준림, 허원준, 장락현, "Combinative Complementary Pass Transistor Logic을 이용한

Datapath 산술 연산기의 설계," 1995년도 ASIC DESIGN WORKSHOP, pp. I.27-I.40, 1995.

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

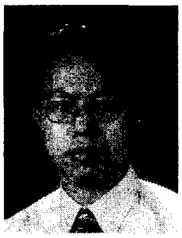
[8] R. P. Brent and H. T. Kung, "A regular layout for parallel adder," IEEE Trans. Comput., vol. C-31, pp. 260-264, Mar. 1982.

[9] reto zimmermann and wolfgang fichtner "Low power logic style : CMOS verse pass transistor Logic", IEEE J. of solid-state circuit, vol. 32, no. 7, July, 1997.

저 자 소 개

趙 泰 元(正會員)

1950년 2월 17일생. 1973년 2월 서울대학교 전자공학과 졸업(공학사). 1986년 5월 미국 루이빌대 전자공학과 졸업(공학석사). 1992년 5월 미국 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

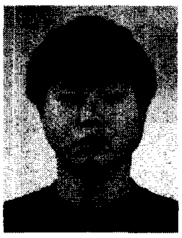


柳 凡 善(正會員)

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(합). 주관심분야는 저전력 회로 설계, DSP core 설계

姜 成 現(正會員)

1971년 2월 15일생. 1997년 2월 충북대학교 전자공학과 졸업(공학사). 1997년 3월 ~ 현재 충북대학교 대학원 전자공학과 석사과정. 주관심분야는 저전력 회로 설계



李 重 錫(正會員)

1972년 1월 1일생. 1997년 2월 충북대학교 전자공학과 졸업(공학사). 1997년 3월 ~ 현재 충북대학교 대학원 전자공학과 석사과정. 주관심분야는 저전력 회로 설계, DSP core 설계

金 文 洙(正會員)

1973년 5월 25일생. 1996년 2월 충북대학교 전자공학과 졸업(공학사). 1998년 2월 충북대학교 대학원 전자공학과 졸업(공학석사). 1998년 4월 ~ 현재 국제전자공업(주) 연구원. 주관심분야는 저전력 회로 설계