

자리올림의 전파특성을 이용한 가산회로의 설계에 관한 연구

(A Study on the Design of an Adder using Carry Propagation Characteristics)

李 庸 奭*, 鄭 己 鉉*, 金 容 得*

(Yong Seok Lee, Gi Hyun Jung and Yong Deak Kim)

要 約

본 논문에서는 새로운 가산방식을 제안하고 이에 의한 가산회로를 설계 및 성능평가를 하였다. 가산원리는 각 자리수에서 가산비트들을 논리배타합 한후 앞자리의 자리올림 상태 및 현재의 가수들의 상태로 결정되는 역전 여부로 최종합을 결정하는 것이다. 제안된 방식으로 구현된 회로는 많이 사용되고 있는 범용화된 CLA(carry lookahead adder)에 비하여 매우 단순하고 규칙적인 구조를 가지고 회로의 구현시 소요 게이트 수가 약 50%정도 적게 소요되며 동작속도가 빠르다.

Abstract

This paper suggests a new addition algorithm. A circuit to implement the algorithm and the result of its performance evaluation are presented. The basic idea of the algorithm is that to obtain the sum of two operands, two operands bits are exclusive-ORed and then the result is inverted by the carry from the previous stage. An unique carry prediction method minimizes carry propagation. The proposed circuit has a very simple and regular structure compared with CLA(carry lookahead adder). It also requires less gates for the implementation about 50% and operates faster.

1. 서론

가산기의 가장 단순한 형태는 전 가산기(full adder)를 직렬로 연결한 형태의 리플 캐리 가산기(ripple carry adder)로 이 방식은 자리올림이 매 자리수마다 전파 되므로 가산시간이 길다. 고속 가산기의 기본적인 형태는 순수 CLA(pure carry lookahead adder)로 가산속도는 매우 빠르나 가산 비트 수가 많아지는 경우 배선이 복잡해 지고 게이트

의 입출력 부하가 증가하므로 실제적인 가산기로의 응용은 어렵다. 이러한 순수 CLA의 단점을 보완한 것이 범용화된 CLA^[1,5]로 이는 순수 CLA의 구조를 수정하여 가산기의 배선과 게이트수를 줄인것으로 연산속도는 순수 CLA보다는 느리지만 효율면에서 우수하므로 범용 프로세서등의 가산회로로 많이 사용되고 있다. 자리올림 선택형 가산기는(carry select adder)^[6]는 가산비트를 일정단위 길이의 순수 CLA로 연산하고 각 단위로의 자리올림은 리플 캐리 가산기의 형태로 처리하는 방식이다. 이 방식은 구조의 규칙성은 범용화된 CLA보다 좋으나 가산 비트수가 커질수록 범용화된 CLA보다 가산속도가 느려지고 소요되는 게이트 수가 많아진다. CSA(conditional

* 正會員, 亞洲大學校 電子工學科
(Dept. of Elec. Eng., Ajou Univ.)
接受日字: 1992年 9月 8日

sum adder)^[7,8]는 각 가수들에 대하여 1과 0의 자리올림의 경우 각각에 대한 가산결과를 미리 연산하고 자리올림 상태를 참조하여 미리 가산되어 있는 가산결과를 선택하는 방식으로 가산속도는 범용화된 CLA보다 빠르나 배선이 길고 복잡하며 소요 게이트 수가 많으므로 고속의 가산이 필요한 경우에만 사용된다.

현재 사용되고 있는 각종 가산기가 추구하는 목표는 구조가 단순하고 보다 적은 게이트를 소요하면서 속도를 향상시키는 것이다. 본 논문에서는 기존 가산방식의 단점을 보완 할 수 있는 새로운 형태의 가산기를 제안한다. 본 논문에서 제안한 방식은 리플캐리 가산기와 같이 자리올림이 전파되는 방식이지만 각 자리에서의 자리올림 예측이 가능하므로 자리올림이 전파되는 경우에도 실제 연산에 소요되는 시간은 범용 CLA보다 짧고 회로가 매우 단순하며 다른 가산기에 비하여 소요되는 게이트 수가 적다. 본 논문에서는 제안된 가산기의 구조 및 회로구성에 대하여 설명하고 범용 CLA와 제안된 가산기의 게이트 수 및 동작속도에 대한 이론적 값 및 실제적 값에 대한 평가를 하였다.

II. 범용 CLA의 가산방식

가산하려는 n 비트의 두 수를 각각 A, B라고 하고 이들의 비트 요소를 A₀, A₁, ..., A_{n-1}, B₀, B₁, B₂, ..., B_{n-1}이라 하자. 이때 각 비트에서의 가산합 및 자리올림에 대한 연산식은 다음과 같다.

$$S_i = A_i \oplus B_i \oplus C_{i-1} \quad (1)$$

$$C_i = A_i B_i + B_i C_{i-1} + C_{i-1} A_i \quad (2)$$

S_i : i번째 자리에서의 합
 C_i : i번째 자리에서의 자리올림
 C_{i-1} : i-1번째 자리에서의 자리올림
 C₀ = 0

CLA의 구성을 위하여 다음의 두개의 보조함수를 정의한다.

$$G_i = A_i B_i \text{ (carry generate function)} \quad (3)$$

$$P_i = A_i \oplus B_i \text{ (carry propagate function)} \quad (4)$$

식 (3), (4)를 식 (1), (2)에 대입하면 i = n-1, ..., 1, 0 에 대하여

$$S_i = (A_i \oplus B_i) \oplus C_{i-1} = P_i \oplus C_{i-1} \quad (5)$$

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1} = G_i + P_i C_{i-1} \quad (6)$$

식 (5), (6)에 의거한 각 비트위치에서의 자리올림은 다음과 같다.

$$C_0 = G_0 + C_{-1} P_0 \quad (7)$$

$$C_1 = G_1 + C_0 P_1 = G_1 + G_0 P_1 + C_{-1} P_0 P_1 \quad (8)$$

$$C_2 = G_2 + C_1 P_2$$

$$= G_2 + G_1 P_2 + G_0 P_1 P_2 + C_{-1} P_0 P_1 P_2 \quad (9)$$

$$C_3 = G_3 + C_2 P_3 = G_3 + G_2 P_3$$

$$+ G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + C_{-1} P_0 P_1 P_2 P_3 \quad (10)$$

$$C_{n-1} = G_{n-1} + G_{n-2} P_{n-1} + C_{-1} P_0 P_1 \dots P_{n-1} \quad (11)$$

이러한 순수 CLA 방식에 의거하여 가산회로를 설계하는 경우 식 (11)에서와 같이 논리곱항의 입력수가 많으며 P₀와 같은 경우는 팬 아웃이 크고 배선이 길어지므로 실제적인 구현이 어렵다. 따라서 가수 및 피가수를 특정 비트 단위로 구분하고 이들 그룹에 대하여 블럭 자리올림 예측회로를 사용하여 해당 블럭의 낮은자리들에 대한 자리올림을 순수 CLA 방식으로 발생시키고 해당 블럭의 가장높은 자리에 대한 블럭자리올림 G*와 블럭 자리올림 전파 P*를 발생시켜 이를 블럭 자리올림 예측회로에 입력시켜 나머지 자리올림들을 발생시킨다. 16비트 범용 CLA의 경우 블럭 자리올림 예측회로는 4비트 단위로 구성하는 것이 효율적인데 이때 발생하는 G* 및 P*는 다음과 같이 표현할 수 있다.

$$P^* = P_0 P_1 P_2 P_3 \quad (12)$$

$$G^* = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 \quad (13)$$

4비트마다 발생하는 G* 및 P*의 조합은 16비트 범용 CLA의 경우 4개가 발생되며 이를 이용하여 발생하는 자리올림 C₃, C₇, C₁₁ 및 C₁₅는 각각 식 (7), (8), (9) 및 (10)에 의하여 처리된다. 연산합은 식 (5)에 의거하여 발생된다.

III. 제안된 가산방식

두 수를 가산하는 경우 가산되는 수의 대응 비트들의 가능한 조합에 대한 자리올림 특성은 표1과 같이 분류된다.

가수 및 피가수가 모두 0인 경우는 다음자리로의 자리올림이 일어나지 않는다. 그러므로 이 경우에는 앞자리에서의 자리올림이 있어도 다음자리로의 자리올림 전파는 중단된다(자리올림 출력 = 0). 가수 및 피가수가 모두 1인 경우에는 앞자리에서의 자리올림 여부에 관계없이 다음자리로의 자리올림이 발생한다

(자리올림 출력 = 1). 그러므로 이러한 두 경우는 앞 자리의 자리올림 상태를 기다릴 필요없이 다음자리로의 자리올림 여부를 결정할 수 있다.

표 1. 가수 및 피가수의 자리올림에 대한 특성
Table 1. carry propagation characteristics for an addened and augend bit pair.

대응 비트의 조합		자리 올림에 대한 특성
가수	피가수	
0	0	자리올림의 정지된다(자리올림 = 0)
0	1	앞자리에서의 자리올림을 전파시킴
1	0	앞자리에서의 자리올림을 전파시킴
1	1	자리올림이 발생된다(자리올림 = 1)

가수 및 피가수가 1,0 또는 0,1인 경우에는 다음자리의 자리올림 값은 앞자리에서의 자리올림값이 되므로 결과적으로 앞자리에서의 자리올림 값을 전파시키는 것이 된다. 그러므로 이 경우에는 앞자리에서의 자리올림 상태가 전달되어야 다음자리로의 자리올림을 결정할 수 있다.

각 자리에서의 가산함은 앞자리에서의 자리올림을 참조하여야 발생된다. 각 자리에서 가산함을 발생시키는 원리는 대응되는 가수비트 및 피가수비트의 논리 배타함을 이용하는데 앞자리에서의 자리올림이 있는 경우에는 논리 배타함의 출력을 반전시킨 값이 가산함 이 되고 자리올림 없는 경우에는 비 반전된 출력이 가산함 이 된다. 결과적으로 낮은 자리로 부터 시작하여 가수 및 피가수가 모두 0인 자리의 다음자리 부터 가수 및 피가수가 모두 1인 자리 까지의 각 자리의 가수 및 피가수의 논리 배타함은 그대로 가산함 이 되고 가수 및 피가수가 모두 1인 자리의 다음자리 부터 가수 및 피가수가 모두 0인 자리 까지의 각 자리의 가수 및 피가수의 논리 배타함을 반전한 것이 가산함 이 된다. 이때 가수의 가장 낮은 자리의 비트보다 1 자리 낮은 비트와 가수의 가장 높은 자리의 비트보다 1자리 높은 비트에는 값이 0인 가상의 가수 및 피가수가 있다고 가정한다.

가산하려는 n비트의 가수 및 피가수를 각각 A, B 라 하고 이들의 비트 요소를 $A_0, A_1, \dots, A_{n-1}, B_0, B_1, \dots, B_{n-1}$ 이라 하자. 이때 제안된 가산방식에 의한 비트 i에서 발생하는 합 및 자리올림은 다음과 같이 표현 할 수 있다.

$$S_i = C_{i-1} \oplus (A_i \oplus B_i) \tag{13}$$

$$C_i = (C_{i-1}(A_i B_i)) + ((A_i B_i)(A_i \oplus B_i)) \tag{14}$$

C_{i-1} : 비트 i-1 에서의 자리올림
 S_i : 비트 i 에서의 가산함
 C_i : 비트 1 에서의 자리올림

제안된 가산원리에 의한 크기가 10비트인 두수의 가산 예를 그림 1에 보였다. 가산하려는 두 수의 비트 자리수 9의 좌측과 비트 자리수 0의 우측에 가수 및 피가수가 모두 0인 자리가 있다고 가정하면 비트 자리수 0부터 가수 및 피가수가 모두 1인 비트 자리수 5까지인 그림 1의 영역 1의 각 자리의 가수 및 피가수의 논리 배타함은 그대로 가산함 이 되고 피가수가 모두 1인 비트 자리수 6 부터 가수 및 피가수가 모두 0인 비트 자리수 10까지인 그림 1의 영역 2의 가산함은 반전되어 가산함 이 된다.

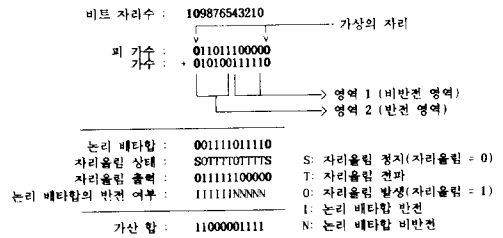


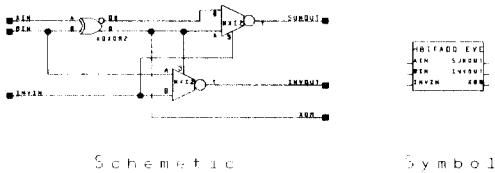
그림 1. 제안된 가산원리에 의한 가산 예
Fig 1. An addition example of the suggested algorithm.

IV. 가산회로의 설계

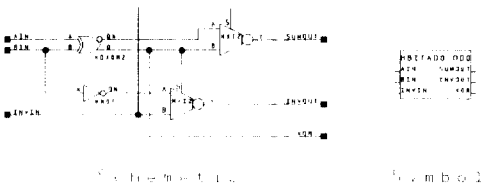
본 논문에서 제안된 가산 방식은 앞자리로 부터의 자리올림이 있는 경우에는 가수 및 피가수 비트의 논리 배타함의 결과를 반전시켜 가산함으로 출력시키고 앞자리에서의 자리올림이 없는 경우에는 논리 배타함을 가산함으로 출력한다. 또한 다음 자리로의 자리올림은 표 1과 같이 가수 및 피가수 비트의 조합에 따라 발생, 전파, 정지된다. 이러한 원리를 이용하여 설계된 각 자리에서의 전가산회로(full adder)는 그림 2와 같으며 이 회로를 가산하려는 비트 수 만큼 직렬로 연결시켜 가산회로를 구성한다. 그림 2에서는 2종류의 1비트 전가산기가 제시되어 있는데 이것은 성능 향상을 위하여 자리올림 출력단에 반전형 멀티플렉서를 사용하였기 때문이다. 자리올림 출력단에 반전형 멀티플렉서를 사용하는 경우 다음 자리에서의 자리올림 입력에는 반전된 값이 인가되게 된다. 따라서 가장 낮은 자리로 부터 짝수 번째의 1비트 전가산

기는 정상적인 자리올림 입력이 인가되고 홀수 번째의 1비트 전가산기에는 반전된 자리올림 값이 인가되게 된다. 따라서 이를 처리하기 위한 2종류의 1비트 전가산기가 필요하게 된다. 그림 2의 INV_IN 및 INV_OUT은 각 자리에서의 자리올림 입력값과 출력값이다.

각 자리에서의 가산을 수행하는 1비트 전가산회로의 자리올림 출력값은 가수 및 피가수의 조합에 의하여 결정된다. 가수 및 피가수의 조합이 0,1 또는 1,0인 경우에는 가수 및 피가수의 논리 배타합의 출력값이 1이며 이 경우에는 앞자리로 부터의 자리올림 값이 다음자리로 전파되므로 2×1 멀티플렉서(MX2)의 두 개의 입력중 앞자리로 부터의 자리올림 입력을 자리올림 출력값으로 선택하고 배타합 발생기의 출력값이 0인 경우 즉, 가수 및 피가수의 조합이 0,0 또는 1,1인 경우는 앞자리로 부터의 자리올림 값에 관계없이 가수 및 피가수의 값을 참조하여 자리올림을 발생 시킨다. 가수 및 피가수의 조합이 0,0인 경우에는 다음자리로의 자리올림이 0이고 1,1인 경우에는 다음자리로의 자리올림이 1이므로 회로의 가수 및 피가수중 어느 한 개의 비트 값을 다음자리로의 자리올림으로 선택한다.



(a)



(b)

그림 2. 제안된 원리에 의한 1비트 전가산회로 및 심벌

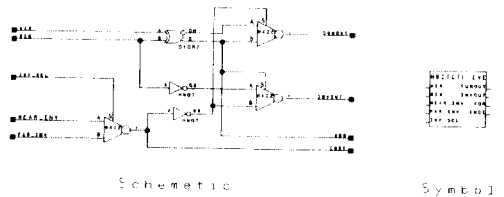
- (a) 짝수 위치형 1비트 전가산 회로 및 심벌
- (b) 홀수 위치형 1비트 전가산 회로 및 심벌

Fig. 2. 1bit full adder by suggested algorithm & its' symbol.

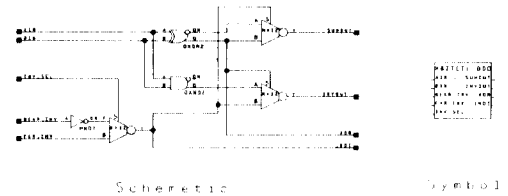
- (a) even cell & its' symbol.
- (b) odd cell & its' symbol.

가산합은 가수 및 피가수의 논리 배타합을 앞자리로 부터의 자리올림 값에 따라 자리올림이 없는 경우에는 그대로 자리올림이 있는 경우에는 반전시켜 출력하므로 회로의 논리 배타합 발생기의 출력과 그 반전된 값을 2×1 반전멀티플렉서(MX12)의 입력으로 하고 앞자리로 부터의 자리올림을 출력 선택신호로 사용한다.

직렬 가산회로가 가산에 가장 많은 시간을 소요하는 경우는 자리올림이 모두 전파되는 경우 즉, 각 자리에서의 논리 배타합이 1인 경우로서 자리올림 전파에 소요되는 시간은 n비트 가산의 경우(n+3)Δt가 되며 연산 비트수가 커질수록 범용 CLA보다 느려진다. 이러한 단점을 보완하기 위하여 자리올림 선택형 전가산기를 4비트 마다 1개씩 사용하였는데 이는 자신의 자리보다 4비트 앞자리까지의 각 자리에서의 가수 및 피가수의 조합이 0,1 또는 1,0인 경우에는 4비트 앞자리로 입력되는 자리올림이 자신에게 전파되며 이는 각 자리의 직렬 가산기의 논리 배타합의 출력이 모두 1인 경우로 그림 4의 가장 우측에 보이는 4입력 논리곱 발생기로 이를 감지하여 그 출력이 1인 경우에는 4비트 앞자리로 입력되는 자리올림 값을 자신의 자리올림 입력값으로 사용하므로써 자리올림의 전파지연을 최소화 한다.



(a)



(b)

그림 3. 자리올림 선택형 1비트 전가산기 및 심벌

- (a) 짝수 위치형 1비트 전가산 회로 및 심벌
- (b) 홀수 위치형 1비트 전가산 회로 및 심벌

Fig. 3. carry selecting 1bit full adder & its' symbol.

- (a) even cell & its' symbol.
- (b) odd cell & its' symbol.

이를 위하여는 4비트 단위로 자신보다 낮은 4자리의 가수 및 피가수의 논리 배타합이 모두 1인 경우에는 자신에게 인가되는 자리올림 입력을 5비트 앞단에서 받아들이고 그렇지 않은 경우에는 바로 앞자리에서 받아들일 수 있는 기능을 갖춘 전가산기가 필요하게 되며 이를 위한 전가산기의 구성은 그림 4와 같다. 그림 3의 자리올림 선택형 전가산기는 그림 2의 전가산기에 바로 앞자리로 부터의 자리올림과 5비트 앞자리로 부터의 자리올림 2개중 하나를 선택할 수 있는 멀티플렉서(MX2)를 추가한 회로로 4비트마다 1개의 자리올림 선택형 전가산기가 연결된다. 또한 이 경우도 반전형 멀티플렉서를 사용하므로 2종류의 자리올림 선택형 전가산기가 필요하게 된다. 자리올림 선택형 가산기에서 선택되는 자리올림 값은 다음식에 의거한다.

$$INV_SEL = (A_{i-4} \oplus B_{i-4}) (A_{i-3} \oplus B_{i-3}) (A_{i-2} \oplus B_{i-2}) (A_{i-1} \oplus B_{i-1}) \quad (15)$$

$$C_{in_i} = INV_SEL \cdot C_{i-5} + INV_SEL \cdot C_{i-1} \quad (16)$$

그림 5는 제안된 원리에 의한 직렬 전가산기(HBITADD) 및 자리올림 선택형 전가산기(HBITCTL)를 이용하여 16비트 예측형 직렬 가산기를 구현한 것이다. 자리올림 선택형 전가산기는 4비트마다 1개씩 삽입되며 4비트 단위로 자리올림의 선택을 하는 INV_SEL(자리올림 선택) 신호를 만들기 위한 4비트 논리곱 회로가 추가되어 있다.

V. 실험 및 평가

제안된 가산방식을 이용하여 설계된 예측형 직렬 가산기 회로와 범용 CLA를 설계하여 이를 모의실험하였다. 모의 실험은 VLSI설계용 시뮬레이션 프로그램을 사용하여 모든 경우의 가수 및 피가수의 조합에 대하여 수행하였다. 실험은 두가지 방법으로 수행하였는데 첫번째는 각 논리 게이트들의 지연시간을 14t로 가정하여 회로의 논리적 지연특성을 평가하였고 두번째는 각 논리 게이트에 대한 2um 게이트 어레이용 표준 셀의 지연시간을 사용하여 각 방식의 실제적인 가산지연을 평가 하였다.

첫번째 모의실험은 각 논리 게이트들의 지연시간을 14t로 가정하여 기존의 범용 CLA 및 제안된 예측형 직렬 가산기의 논리적인 지연특성을 평가 하였으며 그 결과는 표 2와 같다. 예측형 직렬 가산기는 가수가 4비트 증가할때 마다 14t 증가하며 범용 CLA는 가수가 2배 증가할때 마다 24t 증가하므로 가산수의 비트수가 길어질수록 범용 CLA의 속도가 상대적으로 빨라진다. 제안된 예측형 직렬 가산기는 범용 CLA에 비하여 가산 비트수가 커질수록 시간지연이 커지게 되는데 이에 대한 지연특성을 표 2에 보였다. 그러나 예측형 직렬 가산기에 소요되는 게이트가 모두 두개의 입력만을 가지는 반면 범용 CLA에 사용되는 게이트 들은 다입력을 가지는 부분이 많으므로 실제적인 가산시간은 예측형 직렬 가산기가 더 빠르

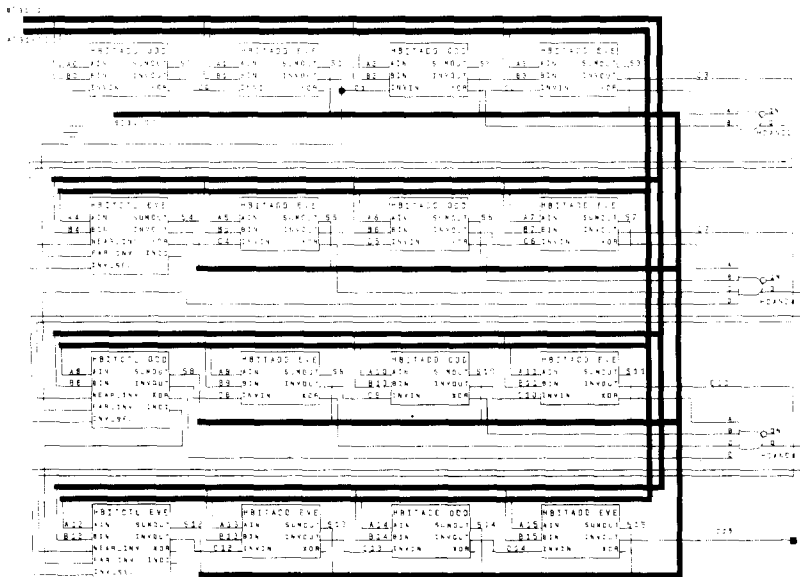


그림 4. 16비트 예측형 직렬 가산기
Fig. 4. 16 bit predictable serial adder.

다는 것이 두번째 모의 실험에 의하여 확인된다.

표 2. 예측형 직렬 가산기 및 범용 CLA의 논리적 지연

Table 2. The logical delay of predictable serial adder and generalized CLA.

가산회로 종류 \ 가산 비트수	16	32
가산회로 종류	16	32
예측형 직렬 가산기	10Δt	14Δt
범용 CLA	10Δt	12Δt

두번째 모의실험은 각 논리 게이트에 대한 2um 게이트 어레이용 표준 셀의 지연시간을 사용하여 설계된 회로에 실제적인 시간지연 특성을 적용하였으며 결과는 표 3과 같다. 이 결과는 논리적 시간지연과 많은 차이를 보이고 있는데 이는 범용 CLA에서는 다입력 게이트를 많이 사용하나 예측형 직렬 가산기에서는 자리올림 예측을 위하여 사용되는 4입력 논리곱 게이트를 제외한 모든 게이트가 2입력이기 때문이다. 각 게이트의 시간지연 특성은 부록에 표시하였다.

표 3. 예측형 직렬 가산기 및 범용 CLA의 실제적 지연

Table 3. The physical delay of predictable serial adder and generalized CLA.

시간 단위 : nsec

가산회로 종류 \ 가산 비트수	16	32
가산회로 종류	16	32
예측형 직렬 가산기	4.9	6.5
범용 CLA	8.6	11.1

앞에서는 범용 CLA 및 예측형 직렬 가산기의 최악의 논리지연에 의한 최저 가산속도를 기준으로 특성을 평가하였으며 최상의 조건에서의 가산속도는 표 4와 같다. 범용 CLA는 일정한 논리적 시간지연을 가지나 예측형 직렬 가산기는 가수 및 피가수값에 따라 논리적 시간지연이 많이 변화한다. 예측형 직렬 가산기는 가수 및 피가수의 조합이 1.1 또는 0.0인 조건에서 최소의 가산시간을 소요한다. 그러나 모든 종류의 게이트들은 실제적으로 상승 및 하강지연이 다르므로 가수 및 피가수의 조합에 따라 서로다른 가산 지연시간을 갖는다.

표 4. 예측형 직렬 가산기 및 범용 CLA의 최고 가산속도

Table 4. The best performance of predictable serial adder and generalized CLA.

가산회로 종류 \ 가산 비트수	가수 및 피가수의 조합	16	32
가산회로 종류		16	32
예측형 직렬 가산기	FFFFH+FFFFH 0000H+0000H	4Δt	4Δt
범용 CLA	모든 경우	10Δt	12Δt

표 5는 각 가산기의 설계에 소요되는 게이트를 입력수 및 종류에 관계없이 계수한 결과로 예측형 직렬 가산기가 범용 CLA의 약 반 정도의 게이트를 소요함을 알수있다. 그러나 다입력 게이트의 경우 입력이 두개인 게이트보다 소요되는 트랜지스터 수가 많으므로 회로의 구현에 소요되는 트랜지스터 수는 범용 CLA의 절반 이하로 볼 수 있다.

표 5. 예측형 직렬 가산기 및 범용 CLA의 논리 게이트 수

Table 5. The required number of gates to implement predictable serial adder and generalized CLA.

가산회로 종류 \ 가산 비트수	16	32
가산회로 종류	16	32
예측형 직렬 가산기	70	142
범용 CLA	120	277

위에서 검토한 바와 같이 예측형 직렬가산기는 범용 CLA에 비하여 절반정도의 게이트를 소요하면서 가산 비트수의 증가에 대한 게이트의 증가량이 적으며 실제적인 지연특성도 우수함을 알수 있다. 가산비트수가 증가함에 따른 가산시간의 증가량은 예측형 직렬 가산기가 매우 나쁘지만 실제적인 지연특성을 적용하면 범용 CLA보다 빠르다는 것이 실험결과로 증명되었다. 또한 각 비트 가산기간의 배선이 매우 짧고 규칙적이고 단순하므로 실제적인 회로의 구현이 매우 용이하다는 장점을 가지며 이는 실제 VLSI설계에 적용시 매우 유용하다.

VI. 결론

가산기는 가산, 승산 및 부동 소수점 연산에서의 지

수 연산기로 사용되는등 그 사용분야가 매우 넓다. 따라서 가산기의 성능향상은 컴퓨터 연산에 있어서의 전 반적인 성능향상에 큰 영향을 미치는 중요한 요소이다.

VLSI 회로의 구현에 있어서 중요한 요소들은 회로의 단순성, 반복성 및 소요되는 게이트의 수로 구분할 수 있으며 본 논문에서 제안한 가산방식은 그 구조 및 배선이 단순하고 반복되는 성질을 가지며 범용 CLA에 비하여 약 50%의 게이트만이 소요하므로 VLSI 회로 구현에 적합함을 알 수 있다. 가산을 위하여 소요되는 지연시간은 예측형 직렬 가산기는 가수가 4비트 증가할때 마다 $1\Delta t$ 증가하며 범용 CLA는 가수가 2배 증가할때 마다 $2\Delta t$ 증가하므로 가산수의 비트수가 길어질수록 범용 CLA의 가산속도가 빨라진다. 그러나 예측형 직렬 가산기에 사용된 게이트들의 입력수가 작아 실제적인 사용에 있어 범용 CLA이상의 연산속도를 가질 수 있다.

부 록 : 실험에 사용한 각 게이트의 전파지연 시간⁹⁾

Appendix : The physical gate delays used in the evaluation of the addition delay.

게이트 종류	입력수	tplh(ns)	tphl(ns)
AND	2	0.6	0.6
	3	0.6	0.7
	4	0.6	0.9
OR	2	0.8	0.5
	3	1.1	0.6
	4	0.6	1.6
XOR	2	0.9	1.1
NOT	1	0.2	0.4
MUX21 (non invert)	2	1.0	0.7
MUX21 (invert)	2	0.4	0.4

tplh : low to high logic transition time

tphl : high to low logic transition time

參 考 文 獻

[1] Kai Hwang, "Computer Arithmetic", John Willey & Sons, 1979, pp.84-91

[1] John P. Hayes, "Computer Architecture and Organization", McGraw-Hill, 1988.

[3] John L.Hennessy & David Patterson, "Computer Architecture a Quantative Approach", Morgan Kaufmann Publishers INC, 1990.

[4] Earl E.Swartzlander, Jr, "Computer Arithmetic", *IEEE Computer Society Press*, 1990

[5] Sklansky,J., "Ultimate-Speed Adders", *IRE Trans.* EC-12, no.2, April 1963, pp.142-148.

[6] Bedrij,O.J., "Carry-Select Adders", *IRE Trans.*, EC-11, no. 3, June 1962, pp.340-346.

[7] Sklansky,J., "Conditional-Sum Addition Logic", *IRE Trans.*, EC-9, no. 2, June 1960, pp.226-231.

[8] Kruy,J.F., "A Fast Conditional Sun Adder Using Carry Bypass Logic", *AFIPS Conf. Proceedings*, vol.27, FJCC 1965, pp.695-703.

[9] Data book, "HSG1K/10K channelless gate array", Hyundai Electronics industries Co.,Ltd. Korea, 1991.

著者紹介



李庸爽(正會員)

1961年 2月 12日生. 1987年 아주대학교 전자공학과 졸업. 1989年 아주대학교 대학원(공학석사). 1992年 현재 아주대학교 전자공학과 박사과정 재학중. 주관심분야 Asic 설계, FA용 통신 및 제어계측

등임.



金容得(正會員)

1946年 1月 30日生. 1971年 연세대학교 전자공학과 졸업. 1973年 연세대학교 대학원(공학석사). 1978年 연세대학교 대학원(공학박사). 1973年 ~ 1974年 블란서 ESE 연구원. 1979年 ~ 1980年 미국 Stanford대학교 연구교수 재직. 1978年 ~ 현재 아주대학교 전자공학과 교수. 주관심분야 디지털 시스템에 관련된 하드웨어 응용, 특히 FA, OA 네트워크 응용과 접속방안, 멀티미디어 시스템 구축과 화상 전송방식, ISDN망 접속 등에 흥미를 갖고 있음.

鄭己鉉(正會員) 第 29 卷 B 編 第 7號 參照

현재 ○주대학교 전자공학과 교수