

論文2002-39SD-9-7

# 보수 이론을 이용한 32비트 파이프라인 캐리 선택 가산기

## (A 32-bit Pipelined Carry-select Adder Using the Complementary Scheme)

金榮俊\*, 金利燮\*\*

(Youngjoon Kim and Lee-Sup Kim)

## 요약

캐리 선택 가산기에 파이프라인을 적용하면 적은 수의 파이프라인 스테이지를 가지면서 많은 수의 파이프라인 스테이지를 갖는 가산기처럼 높은 주파수 상에서 구동한다. 이 논문에서는 캐리 선택 가산기 구조를 적용한 4 블록 5스테이지 파이프라인 32비트 가산기를 제안하였다. 이 제안된 가산기는 기존의 16스테이지 파이프라인 32비트 가산기와 같이 높은 주파수에서 동작한다. 그럼에도 불구하고 이 제안된 가산기는 기존 16스테이지 파이프라인 가산기 보다 3배 적은 트랜지스터로 구현 가능하다. 이 가산기는 0.25um CMOS 공정으로 구현할 때 2.5V 전압에서 1.67GHz으로 동작한다.

## Abstract

Using the carry-select adder scheme, an adder with small number of stages can be operated as fast as an adder with large number of stages. In this paper, a 4-block 5-stage 32-bit pipelined carry-select adder is designed and implemented. The proposed adder operates as fast as a conventional 16-stage 32-bit pipelined adder while the number of registers required is nearly same as a conventional 4-stage pipelined adder. This adder is operated at 1.67GHz clock frequency in a standard 0.25um CMOS technology with 2.5 V supply voltage.

**Key Words** : adder, pipeline, carry-select

## I. 서론

신호처리나 영상처리 어플리케이션에 쓰이는 가산기는 일반적으로 사용되는 가산기와는 다르게 구현된다. 그러한 가산기는 많은 양의 데이터를 단시간에 처리

해야 하기 때문에 많은 덧셈이 단기간에 수행될 수 있도록 구현되어야 한다. 따라서 가산기의 동작 주파수를 높이고 빠른 시간 내에 많은 결과를 얻기(Throughput) 위하여 파이프라인이 사용된다.

기존 파이프라인 가산기가 만약  $n$ 개의 스테이지를 가진다고 하면 이상적으로는 동작주파수가  $n$ 에 비례하여 향상이 된다. 따라서 32비트 가산기에서는 32스테이지 또는 더 많은 스테이지로 구현할 때 가산기를 가장 빠르게 구동시킬 수 있다. 하지만 하나의 스테이지를 늘릴 때마다 전체 비트 수보다 더 많은 레지스터가 필요하게 된다. 즉 32비트 가산기를 32스테이지로 구현하면 천 개 이상의 레지스터가 필요하다. 이처럼 많은 수의 레지스터는 많은 양의 전력 소모와 큰 면적을 필요로 할 뿐만 아니라 입력이 들어가서 출력이 나오는 시

\* 學生會員, \*\* 正會員, 韓國科學技術院 電子電算學科 電氣 및 電子工學

(Division of Electrical Engineering, Dept. of Electrical Engineering & Computer Science, KAIST)

※ 본 연구는 한국과학재단과 미세정보 시스템 연구센터의 지원을 받아 수행되었습니다.

接受日字:2001年10月9日, 수정완료일:2002年8月19日

간(latency)도 길어지게 된다. 따라서 어플리케이션에 맞게 스테이지를 선택하여 가산기를 구현하는 것이 중요하다.

이 논문에서는 레지스터의 수를 최소화 하면서 높은 주파수를 가질 수 있도록 파이프라인 가산기에 캐리 선택 가산기를 적용한 구조를 제안하였다. 제안된 4 블록 5 스테이지 32비트 파이프라인 캐리 선택 가산기는 기존의 16스테이지 32비트 파이프라인 가산기처럼 높은 주파수에서 동작한다. 하지만 기존 16스테이지 파이프라인 가산기를 구현할 때 500개 이상의 레지스터가 필요한 것에 반하여 제안된 가산기는 276개의 레지스터만으로 구현된다.

본 논문의 구성은 다음과 같다. II장에서는 기존에 사용되고 있는 파이프라인 가산기의 구조에 대하여 간략히 소개하였다. III장에서는 새롭게 제안된 캐리 선택 파이프라인 가산기의 구조를 설명하였다. IV장에서는 제안된 구조를 분석하여 다른 가산기들과 비교하고 구현된 가산기의 모의 실험 결과를 보여주었다. 마지막 장에는 논문의 전체적 요약과 결론을 제시하면서 끝맺음을 하였다.

II. 기존의 파이프라인 가산기 구조

전 장에서 언급하였듯이 하나의 파이프라인 스테이지를 늘리려면 큰 면적과 많은 전력이 요구된다. 따라서 기존의 32비트 가산기에서는 주로 8스테이지나 그 이하 스테이지로 구현되었다. 그림 1에서는 기존의 4스태이지 32비트 파이프라인을 보여준다. 8비트의 리플캐리 가산기(RCA)부터의 캐리 출력이 다음 클럭에 다음 스테이지로 전파될 수 있도록 구성되어 있다. 이 가산기의 최상 동작속도는 스테이지 사이에 들어가는 8비트 RCA의 속도와 레지스터의 속도에 좌우된다. 그림 1의 32비트 파이프라인 가산기를 구현하기 위해서는 148개의 레지스터와 32개의 전가산기가 필요하다.

32비트 가산기를 그림 2와 같이 구현함으로써 레지스터의 수를 줄일 수 있다. 이 가산기의 구조는 다음과 같다. 4개의 독립된 8비트 RCA가 첫 번째 스테이지에서 평행으로 수행된다. 다음 스테이지에서는 9비트 RCA 또는 1 증가기 회로가 첫 번째 RCA에서 나온 캐리 출력과 두 번째 가산기에서 나온 출력을 더한다. 이 회로에서는 9스테이지의 리플캐리 가산기 대신에 1 증

가기 회로를 사용함으로써 가장 느린 스테이지를 첫째 스테이지로 만들면 그림1에서 보여주었던 구조와 같은 속도로 구동하게 된다. 그림1의 구조와 비교해서 3개의 9비트 1 증가기 회로가 늘어나는 반면에 레지스터의 수는 102로 줄어든다. 따라서 그림 1의 구조보다 약 15%적은 트랜지스터로 가산기를 구현할 수 있다.

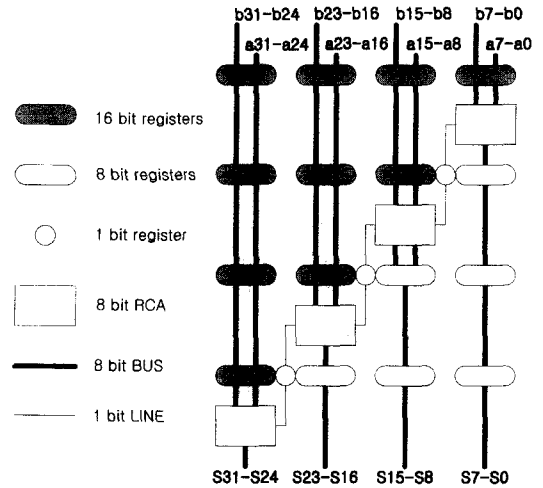


그림 1. 기존의 4단 32비트 파이프라인 가산기  
Fig. 1. Conventional 4-stage 32-bit pipelined adder.

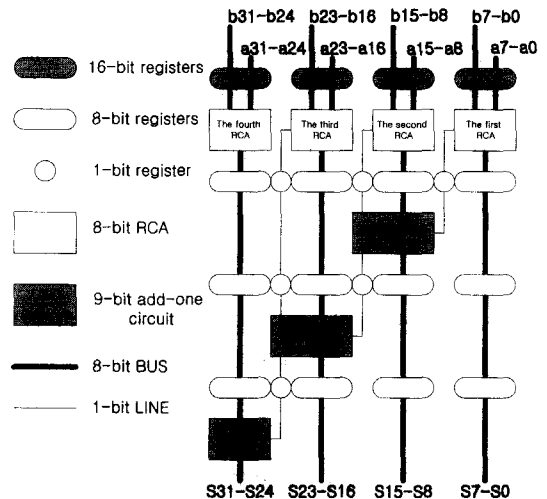


그림 2. 레지스터의 수를 줄인 기존의 4단 32비트 파이프라인 가산기  
Fig. 2. Conventional 4-stage 32-bit pipeline adder with reduction on number of registers.

이 두 개의 가산기는 파이프라인을 32 스테이지로 구현한 가산기, 즉 동작주파수가 전가산기 하나와 레지

스터 하나로 이루어진 가산기보다 약 8배정도 낮다.

### III. 제안된 CSA 구조

많은 수의 레지스터를 사용하지 않고 주파수를 높이기 위하여 보수 이론을 이용한 캐리 선택 가산기가 사용되었다. 제안된 가산기의 구조를 설명하기 이전에 보수 이론부터 보도록 하자.

#### 1. 하나의 리플캐리 가산기로 이루어진 캐리 선택 가산기

제안된 파이프라인 가산기에는 보수이론을 기반한 캐리 선택 블록이 이용되었다. Chang이 처음 제안한 보수이론은 캐리인이 1인 합은 캐리인이 0인 합에서부터 1을 더함으로써 구할 수 있다는 것으로부터 시작이 된다.<sup>[1]</sup> 만약  $(S_{n-1}^0, S_{n-2}^0, S_{n-3}^0, \dots, S_0^0)$ 이 캐리가 0일 때의 합이고  $(S_{n-1}^1, S_{n-2}^1, S_{n-3}^1, \dots, S_0^1)$ 이 캐리가 1일 때의 결과라고 가정한다면 보수 이론에서는  $S_n^0$ 를 반전 또는  $S_n^0$  자체로써  $S_n^1$ 을 구할 수 있다.

그림 3에서는 보수 이론의 두 가지 예를 보여주고 있다.

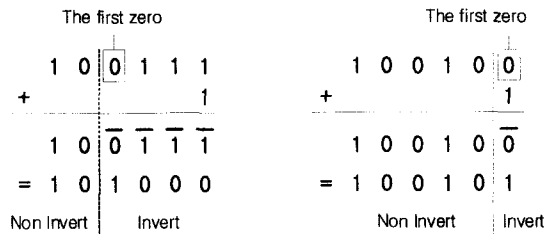


그림 3. 보수 이론의 두 가지의 예  
Fig. 3. Two examples for the complement scheme.

이 보수 이론을 이용한 캐리 선택 가산기는 멀티플렉서를 이용하여 하여 구현할 수 있다. 멀티플렉서를 기반으로 한 캐리 선택 블록의 부분이 그림 4에 나타나 있다. 그림 4의 회로는 블록 안에서 캐리가 0이라 가정 한 가산기 즉 리플캐리 가산기에서 나온 결과를 가지고 캐리가 1일 때의 결과를 구하는 회로이다.

인버터(inverter)가 각각의 합을 반전하고 멀티플렉서가 제어신호에 따라서 합 또는 반전된 합을 취할 수 있게 하여준다. 멀티플렉서에 들어가는 컨트롤 시그널은 첫 번째 0을 찾는 회로로부터 생성이 된다. 첫 번째 0을 찾는 회로는 그림 4에서 회색으로 칠해진 부분이

며 다음과 같이 구동한다. GND가 처음 0이 나오는 노드까지 전파 되고 처음 0이 나온 후 그 후의 노드부터 마지막 노드까지 대해서는 그 다음 값에 관계없이 VDD가 전파된다. 따라서 그림 4에서 보여주는 회로는 캐리가 0이라고 가정했던 결과(sumk)로부터 캐리가 1이라고 가정했던 결과(S+1<sub>k</sub>)를 구할 수 있다.<sup>[2]</sup>

Check Sums의 신호는 그림 4의 회로에서 캐리가 0이라고 가정했을 때의 모든 결과(sums)가 1 일 때만 1이 되고 그 이외의 경우에는 0이 된다. 그림 5에서 볼 수 있듯이 이 신호는 다음 단의 캐리를 결정하는 멀티플렉서의 제어 신호로 사용된다. 다음 단의 캐리를 결정하는 멀티플렉서는 만약 캐리가 0이라고 가정했을 때의 모든 결과(sum)가 1이면 기존 단의 캐리를 다음 단으로 전달되는 캐리로 선택하고 sum의 값이 하나라도 0이면 캐리가 0이라고 가정한 회로(리플캐리 가산기)에서 구해진 캐리를 다음 단의 캐리로 선택한다. 즉 sum이 하나라도 0이라면 그 전달에서 온 캐리는 다음 단으로 전달 되는 캐리에 아무런 영향을 미치지 않는다.

첫 번째 0을 찾는 회로에서 NMOS는 항상 0만이 지나 갈수 있는 곳에 쓰였고 PMOS는 항상 1만이 지나 갈수 있는 곳에 쓰였다. 0과 1일 모두 지나가는 패스에서는 트랜스미션 게이트를 써서 문턱전압에 의하여 생기는 전압 손실 및 이에 의한 전력손실을 막았다.

가장 느린 동작은 NMOS, n-1 트랜스미션 게이트 인버터, 그리고 버퍼들을 지나는 패스이다. 이 회로에서는 전체 지연시간이 비트 수 n과 비례하게 된다.

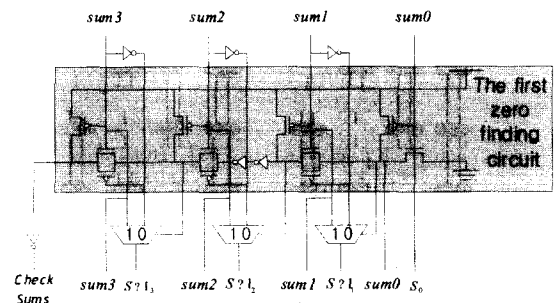


그림 4. 캐리를 1로 가정한 결과를 캐리를 0으로 가정한 결과로부터 구하는 회로  
Fig. 4. Circuit for generating sums for carry = 1 from sums for carry = 0.

밀의 수식은 기존의 캐리 선택 가산기와 보수이론을 적용한 캐리 선택 가산기의 트랜지스터 수를 나타낸다.

그림 5에서 볼 수 있듯이 제안된 가산기에는 8비트로 구현이 된 그림 4의 회로가 세 번 쓰임으로 블록 수를 3으로 곱해주었다. 그림 4와 같이 캐리 선택 가산기의 블록을 구현하면 기존의 듀얼 리플캐리 가산기를 사용한 구조 보다 약 25% 적은 수의 트랜지스터로 캐리 선택 가산기의 블록을 구현할 수 있다.

두 개의 리플캐리 가산기를 이용한 캐리 선택 가산기에 사용되는 트랜지스터의 수

$$3(\text{블록 수}) \times (24(\text{전가산기}) + 6(\text{리플캐리 가산기}) + 6(\text{MUX})) \times (8(\text{블록크기})) = 1296 \text{ tr.}$$

그림 4의 회로를 이용한 캐리 선택 가산기 사용되는 트랜지스터의 수

$$3(\text{블록 수}) \times ((24(\text{전가산기}) + 6(\text{MUX}) + 3(\text{pass tr.}) + 6(\text{MUX})) \times (8(\text{블록크기})) + 2(\text{인버터})) \times 7(\text{버퍼수}) = 950 \text{ tr.}$$

그림 4로 회로를 구현하였을 때의 동작시간은 듀얼 리플캐리로 사용하였을 때보다 조금 느려진다. 하지만 그림 4의 회로는 전체 동작 주파수에 영향을 주지 않는다. 동작 주파수에 대한 자세한 사항은 다음 장에서 설명하기로 한다.

2. 제안된 가산기의 구조

그림 5에서는 제안된 가산기를 전체 구조를 보여준다. 32비트의 가산기를 4 개의 블록 캐리 선택 가산기로 구현하고 파이프라인을 적용하였다. 그러면 각각의

블록들은 서로 평행하게 동작을 하게 된다. 이러한 각각의 블록들은 독립적으로 작동하고 5스테이지로 나누어져 있다. 마지막 스테이지에서는 멀티플렉서가 블록으로 들어오는 캐리가 1이나 0이냐에 따라서 합 또는 반전된 합을 선택 할 수 있게 하여준다. 그림 4의 캐리가 1일 때의 결과를 구하는 회로가 모든 블록에 들어간다. 그런데 이 회로의 동작속도는 2비트 리플캐리 가산기보다 크기 때문에 전체 동작 주파수에 영향을 주지 않게 구조를 짜기 위해선 이 회로를 그림 6과 같이 여러 개의 작은 블록으로 나누어야 한다. 이렇게 나눈 회로는 2비트 RCA와 동시(parallel)에 동작하고 속도가 2비트 RCA보다 빠르기 때문에 이 전체 동작주파수는 이 회로에 상관없이 2비트 RCA와 레지스터에 의해 결정된다.

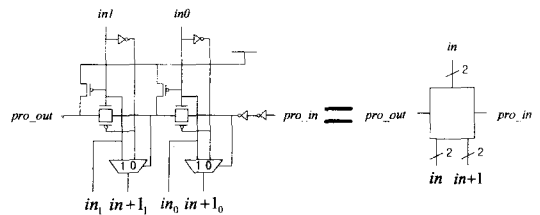


그림 6. 캐리가 1을 가정한 두 비트 합 생성블록  
Fig. 6. The 2-bit sum generation block assuming carry equals one.

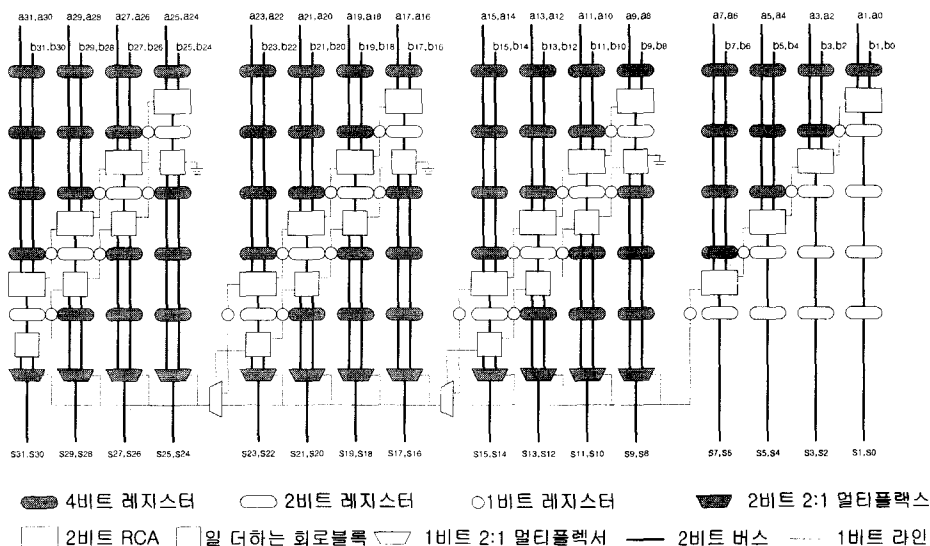


그림 5. 새로 제안된 4스테이지 32비트 파이프라인 가산기  
Fig. 5. The proposed 4-stage 32-bit pipeline architecture.

또한 마지막 스테이지의 동작시간도 전체 주파수에 영향이 미치지 않는다. 마지막 스테이지는 3개의 멀티플렉서를 거치는데 이것의 동작시간도 2비트RCA보다 작기 때문이다. 따라서 가장 느린 회로는 2 비트 RCA가 되고 이것이 전체 동작주파수를 결정한다. 기존의 4 스테이지 파이프라인 가산기에서는 8비트 RCA가 전체 동작주파수를 결정하는데 제안된 구조는 기존의 4스테이지 파이프라인 가산기보다 대략 4배정도 빠르다. 즉, 제안된 5스테이지 파이프라인 가산기는 대략 기존의 16 스테이지 파이프라인 가산기와 동작 주파수가 비슷하게 된다. 그럼에도 불구하고 제안된 구조는 276개의 레지스터만으로 구현이 가능하다.

그림 7에는 이 제안된 구조에 쓰인 레지스터를 보여준다. 이 레지스터는 power PC 603 내부의 주요 레지스터로 쓰였고 이 레지스터는 낮은 전력과 빠른 구동 능력을 가진다.<sup>[3]</sup> 이 레지스터는 하강 모서리 천이(falling edge)로 작동을 한다.

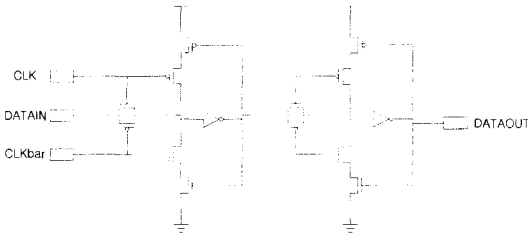


그림 7. 제안된 가산기에 사용된 D플립플롭  
Fig. 7. D-flip-flop used in the proposed adder.

전가산기로는, 트랜스미션 게이트로 구성된 18개의 트랜지스터로 이루어진 회로가 사용된다. 이 전가산기는 안정적이고 다른 CMOS로 이루어진 전가산기 보다 빠르다. 또한 합과 캐리가 출력되는 시간이 비슷하기 때문에 이 두 가지 모두 전체 동작 주파수에 영향을 주는 제안된 구조에 적합하다.

IV. 시뮬레이션 결과 및 비교

파이프라인 스테이지 수 또는 블록 수에 따라서 가산기에 들어가는 트랜지스터 수가 어떻게 달라지는지 예측 분석하였다. 이 예측 분석에 들어간 플립플롭은 16개의 트랜지스터로 구현되었고 전가산기의 수는 18개의 트랜지스터로 구현되었다. 그림 8에서 볼 수 있듯이 같은 수의 스테이지를 가진 가산기들은 블록수가

늘어남에 따라서는 많은 변화가 일어나지 않는다. 즉 가산기의 트랜지스터의 수는 블록 수의 변화에 크게 영향을 받지 않는다. 하지만 트랜지스터의 수는 파이프라인의 스테이지수가 늘어남에 비례하여 늘어난다.

그림 9에서는 여러 가지 수의 스테이지와 블록의 가산기들의 예측 동작 시간을 분석한 그래프를 보여준다. 그림에서 알 수 있듯이 블록을 많이 나눌수록 동작 주파수는 높아진다. 16스테이지로 32비트 파이프라인 가산기를 구현했을 때 8스테이지 2블록 또는 4스테이지 4블록으로 구현한 가산기와 주파수가 거의 같다는 것을 알 수 있다. 이 예측 분석에서는 표 1에서 보여주는 것과 같이 SPICE로 모의 실험 한 기초 게이트를 기반으로 하였다. 표 1은 기초 게이트를 0.25um CMOS공정으로 구현하고 2.5V의 전압을 주었을 때 SPICE로 모의 실험 한 지연전달시간을 보여준다.

표 1. 기본 게이트의 동작시간 모의 실험  
Table 1. Simulated actual delay time of basic gates.

Basic gates	Delay
Inverter	0.08ns
NAND	0.13ns
Multiplexer(sel)	0.11ns
Multiplexer(thru)	0.05ns
XOR	0.11ns
D-flipflop	0.20ns
Sum(Full Adder)	0.21ns
Cout(Full Adder)	0.21ns

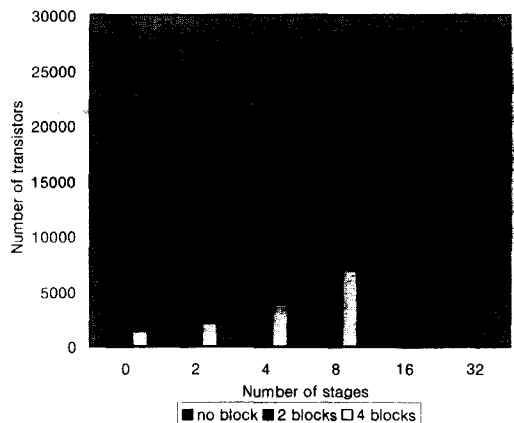


그림 8. 블록 수와 파이프라인 스테이지 수에 따른 트랜지스터의 수

Fig. 8. Number of transistors according to the number of blocks and stages.

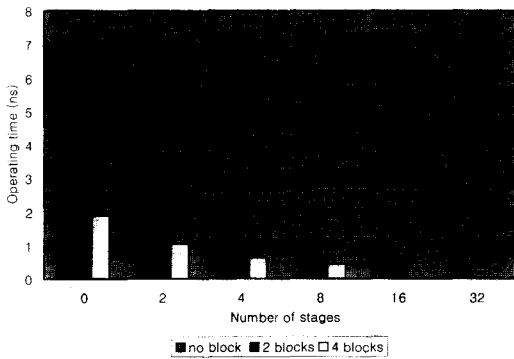


그림 9. 블록 수와 스테이지 수에 따른 동작시간  
Fig. 9. Operation times for different number of blocks and different number of stages.

그림 8과 그림 9에서 볼 수 있듯이 트랜지스터의 수는 블록수가 두 배로 늘어남에 따라서 크게 변하지 않지만 동작 주파수는 약 두 배로 늘어난다. 따라서 가산기를 만들 때 여러 개의 블록을 합쳐서 구현함으로써 적은 트랜지스터를 가지고 높은 주파수에서 가산기를 구현할 수 있다.

이 논문은 4블록 5스테이지 32 비트 덧셈기를 0.25um CMOS공정으로 구현하였다. 그림 10은 HSPICE로 제안된 덧셈기를 모의 실험한 결과이다. 이 모의 실험에서는 모든 b의 입력을 1로 세트하고 다른 모든 비트를 0으로 세트한 후 a24 입력에 두개의 사각형 신호를 보내고 HSPICE로 출력을 관찰한 결과이다. 그림 10에서 볼 수 있듯이 이 덧셈기는 1.67GHz에서 안정적으로 작동한다.

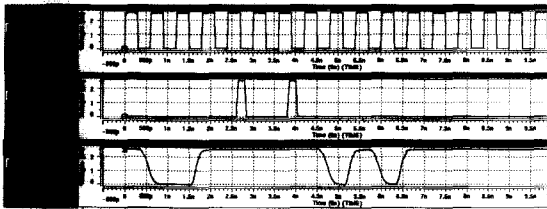


그림 10. HSPICE로 모의 실험한 결과  
Fig. 10. HSPICE simulated result of the proposed adder.

## VI. 결 론

캐리 선택 가산기에 파이프라인을 적용하면 5스테이지 4블록 32비트 파이프라인 가산기가 16스테이지 32비트 파이프라인 가산기처럼 높은 주파수에서 동작할

수 있다. 기존의 16스테이지에 요구되는 약 500개의 레지스터와 비교하면 새로 제안된 가산기는 276 레지스터로만으로 구현된다. 새로 제안된 가산기는 작은 수의 레지스터를 쓰기 때문에 면적이 적고 전력소모가 적다. 이 제안된 가산기는 2.5V의 전압을 가할 때 1.67GHz에서 동작한다.

## 참 고 문 헌

- [1] T. Y. Chang and M. J. Hsiao, "Carry-select adder using single ripple-carry adder". *Electronics Letters*, Vol. 34, No. 22, Oct 1998, pp 2101~2103.
- [2] Y. Kim and L. S. Kim, "A 64-bit carry-select adder with reduced area". *Electronics Letters*, Vol. 37, No. 10, May 2001, pp. 614~615.
- [3] V. Stojanovic and V. G. Oklobdzija, "Comparative Analysis of Master-Slave Latches and Flip-Flops for High-Performance and Low-Power Systems". *IEEE J. Solid-State Circuits*, Vol. 34, No. 4, April 1999, pp. 536~548.
- [4] N. H. E. Weste, and K. Esharghian, *Principle of CMOS VLSI Design*, Addison-Wesley, 1992, pp 526.
- [5] B. Lim and J. Kang, "A Self-Time Wave Pipelined Adder Using Data Align Method". *The Second IEEE Asia Pacific Conference on ASICs*, Aug. 2000, pp. 77~80.
- [6] L. Dadda and P. Vincenzo, "Pipelined Adders". *IEEE Transactions on Computers*, Vol. 45, No. 3, Mar. 1996, pp. 348~356.
- [7] R. Rogenmoser and Q. Huaung, "An 800-MHz 1-um CMOS Pipelined 8-b Adder Using Single-Phase Clocked Logic-Flip-Flops". *IEEE J. Solid-State Circuits*, Vol. 31, No 3, Mar. 1996, pp. 401~409.
- [8] D. A. Patterson and J.L. Hennessy, *Computer Organization & Design*, 2nd Ed., Morgan Kaufmann Publishers, 1997, pp. 439~449.

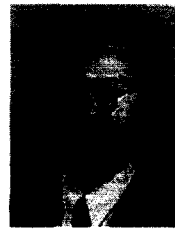
저 자 소 개

金 榮 俊(學生會員)

1999년 12월 University of Michigan 전자공학과 학사 졸업 (공학사). 2000년 3월~한국과학기술원 전자전산학과 전기및전자공학 전공 졸업 (공학석사). <주관심분야 : Analog & Digital VLSI 회로 설계>



金 利 燮(正會員)



1982년 서울대학교 전자공학과 학사. 1986년 Stanford University 전자공학과 석사. 1990년 Stanford University 전자공학과 박사. 1990년~1993년 Toshiba Corporation 연구원. 1993년~현재 한국과학기술원 전자전산학과 전기및전자공학 전공 부교수. <주관심분야: 멀티미디어 VLSI 설계, 저전력 설계, EML 설계, 3D Graphics>