

論文94-31B-5-10

RSD 수 표현 체계를 이용한 셀프-타임드 제산기의 구조

(A Self-Timed Divider Structure using RSD Number System)

崔起榮*, 姜俊宇**

(Kiyoung Choi and JunWoo Kang)

要約

본 논문에서는 캐리 전파가 필요 없는 RSD 수 표현 체계와 셀프-타임드 링 구조를 조합한 제산기의 구조를 제안한다. 셀프-타임드 링 구조를 이용함으로써 작은 실리콘 면적을 차지하면서도 조합 회로 배열 구조의 제산기와 견줄 만한 속도를 내는 제산기를 구현할 수 있다. 또한, 캐리 전파가 필요 없는 수 표현 체계를 사용함으로써 실리콘 면적과 계산 시간을 더욱 줄일 수 있다. 제안된 제산기의 알고리즘과 구조는 VHDL 모델에 의한 모의 실험을 통하여 성공적으로 검증되었으며 예비적인 실험을 통하여 효과적임을 확인하였다.

Abstract

This paper proposes a divider structure that combines a carry-propagation-free division algorithm using RSD number system and a self-timed ring structure. The self-timed ring structure enables the divider to compute at a speed comparable to that of combinational array dividers with less silicon area. By exploiting the carry-propagation-free division algorithm, we can achieve further reduction of silicon area and computation time. The algorithm and structure of the proposed divider have been successfully verified through VHDL modeling and simulation. Preliminary experimental results show the effectiveness of the algorithm and structure.

1. 서론

제산기를 구현하는 방법은 크게 두가지로 나눌 수

있다. 그 중 하나는 시간적으로 반복 계산하는 방법이며 다른 하나는 조합 회로의 배열을 이용하는 방법이다.^[1] 일반적으로 반복 계산 방법은 작은 실리콘 면적을 차지하는 대신에 한번 계산하는 데 여러 클락 사이클이 소요되므로 동작이 느리다는 단점이 있다. 조합 회로의 배열로 되어 있는 경우는 계산이 빠른 반면에 실리콘 면적을 많이 차지한다는 단점이 있다. 본 논문에서 제안하는 구조는 반복 계산 방법을 채택하여 실리콘 면적을 작게 차지하면서도 셀프-타임드 (self-timed) 논리와 RSD(Redundant Signed Digit) 수 표현 체계^[2]를 동시에 이용하여 나눗셈이

* 正會員, 서울대학교 半導體 共同研究所
(Inter-university Semiconductor Research Center, Seoul Nat'l Univ.)

** 正會員, 韓國電子通信研究所
(Electronics and Telecommunications Research Institute)

接受日字 : 1993年 10月 21日

고속으로 이루어질 수 있도록 한다.

셀프-타임드 논리를 이용한 계산기는 이미 Williams와 Horowitz에 의하여 발표된 바 있다.^[4] 그들은 5 단의 셀프-타임드 링(ring) 구조를 이용하여 조합 회로 배열 구조의 계산기와 견줄 만한 속도를 내면서 면적은 훨씬 작은 계산기를 구현하였다.

RSD 수 표현 체계도 이미 나눗셈 알고리즘에 적용된 바 있다.^{[4] [5]} 이 알고리즘은 RSD 수 표현 체계를 이용하면 캐리의 전파가 없는 계산을 할 수 있다는 사실을 이용한다. 캐리의 전파가 없으면 이에 의한 시간 지연이 없으므로 덧셈이나 뺄셈을 고속으로 할 수 있으며 이를 이용하여 곱셈이나 나눗셈도 고속으로 할 수 있게 된다.^[4] 특히 [5] 에 발표된 알고리즘은 하나의 bit에 대하여 한 개씩의 전가산기(full adder)로 처리가 가능하므로 최소한의 하드웨어를 이용한 구현이 가능하다.

본 논문에서는 위의 두 방법을 조합한 구조를 제안한다. 이 구조는 링 구조를 가지므로 조합 회로의 배열 구조와 견줄만 한 속도를 가지면서 하드웨어의 크기를 많이 줄일 수 있다.

동시에 RSD 수 표현 체계를 이용하므로 나눗셈의 각 반복 계산 과정에서 캐리가 전파되지 않는 덧셈과 뺄셈을 할 수 있어서 빠른 계산이 가능해진다.^[6]

[3] 에서도 부분 나머지(partial remainder)를 구하는 데에는 CSA(carry-save adder)를 사용하여 캐리의 전파가 필요없는 계산을 하지만 몫의 선택을 위해서는 3-bit CPA(carry-propagate adder)를 사용해야 하므로 이를 위한 시간과 면적이 필요하게 된다. 또한, 본 논문에서 제안하는 계산기는 구조적으로 링을 구성하는 단의 수를 많이 줄일 수 있다. 따라서, 본 논문에서 제안하는 계산기의 구조를 이용하면 더 작은 면적으로 더 빠른 나눗셈을 할 수 있게 된다.

본 논문의 II 절에서는 제안하는 계산기의 알고리즘, 전체 구조, 링을 구성하는 각 단의 내부구조, 설계된 일부 셀프-타임드 논리 회로 등을 설명한다. III 절에서는 이 계산기에 대한 타이밍 분석을 한다. 마지막으로 IV 절에서는 예비적인 실험 결과와 앞으로의 연구 등에 대하여 기술하고 결론을 맺는다.

II. 알고리즘

본 논문에서는 [5] 에 발표된 알고리즘과 [3] 에 발표된 구조를 조합하여 사용함으로써 훨씬 작은 면적으로 좋은 성능을 갖는 계산기를 제안한다.

이 계산기의 알고리즘은 덧셈/뺄셈과 왼쪽으로의 자리 이동(shift-left)을 반복하여 2진법의 nonr-

estoring division을 수행하는 SRT division 방법^[7] 을 사용한다. 이는 다음과 같은 식으로 표현된다.

$$R^{j+1} = 2(R^j - q_j \cdot D) \quad (1)$$

$$-2D < R^j < 2D \quad (2)$$

여기에서 R^j ($j = 1, 2, \dots$)는 j 번째 단계에서의 부분 나머지고 R_0 는 피제수, D 는 제수이며, q_j 는 몫의 j 번째 자리이다.

식 (1)에서 R^j 의 값을 계산하기 위한 덧셈 또는 뺄셈은 캐리의 전파를 필요로 하는 가산기를 사용하면 많은 시간을 필요로 하므로 시간을 줄이기 위하여 여기에서는 RSD 수 표현 체계를 사용하는 1-bit 전가산기들로 구성된 가산기를 이용한다. 이때, 피제수와 부분 나머지는 RSD 표현을 사용하고 제수는 보통의(nonredundant) 2진 표현을 사용한다. 일반적으로 사용되는 1-bit 전가산기는 모든 입력과 출력에 값이 1인 가중치가 적용되는 것으로 생각할 수 있다. 그러나, 입력과 출력의 가중치를 다르게 해 줌으로써 4가지 형태의 1-bit 전가산기를 만들 수 있다.^[8] 여기에서 사용하는 RSD 가산기는 type-1 전가산기만 구성된다. 그림 1은 RSD 표현의 수와 2진 표현의 수의 덧셈과 뺄셈을 위한 전가산기의 동작과 함께 RSD 표현을 2진 표현으로 변환하는 표를 보여 준다. 여기에서 $\bar{1}$ 는 -1의 값을 가지며 $\bar{0}$ 는 -0으로 0과 같은 값을 갖는다. 이와 같은 전가산기들로 구성된 가산기를 이용하여 q_j 의 값에 따라 덧셈 또는 뺄셈을 하면 캐리의 전파 없이 R^j 의 값에 대한 RSD 표현을 계산할 수 있다.^[5]

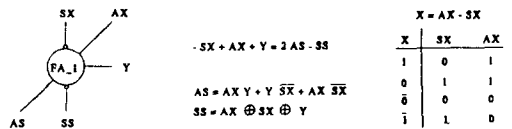


그림 1. RSD 수 표현 체계를 이용한 Type-1 전가산기의 동작

Fig. 1. Type-1 full adder operation with RSD number system.

표 1. R_j 의 세 MSD에 의한 q_j 값의 결정
Table 1. The value of q_j determined by three MSDs of R_j .

q_j	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	
R_j	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0
R_j	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0
q_j	1	1	1	1	1	0	1	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1

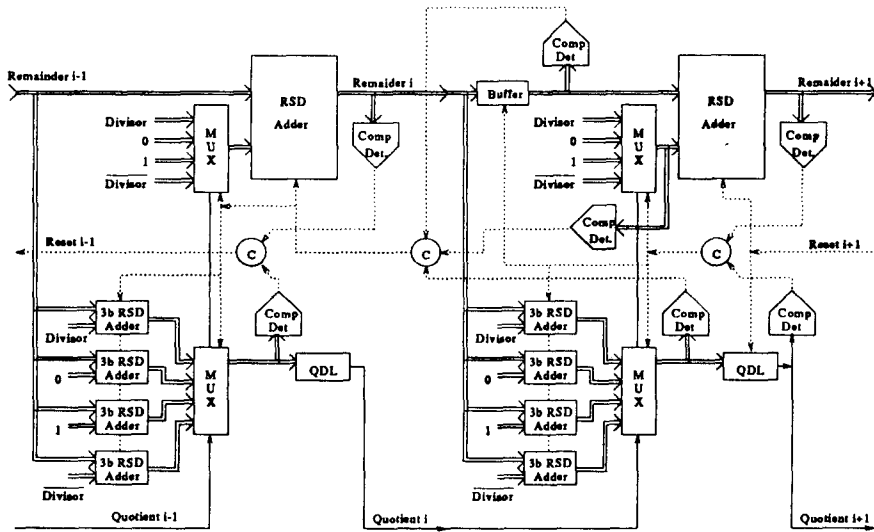


그림 4. Overlapped execution의 경우 링을 구성하는 각 단의 내부 구조
 Fig. 4. Internal structure of each stage in the ring for overlapped execution.

각 큰 단마다 3-bit RSD 가산기가 네 개 추가될 뿐만 아니라 링을 구성하는 단의 수가 두 배로 증가하므로(III 절 참고) 전체 하드웨어가 많이 커지게 된다. 그러나 4 개의 큰 단이면 충분하므로 [3]에 발표된 것보다는 작은 면적으로 구현이 가능하다.

Overlapped execution이 이루어지는 과정을 설명하면 다음과 같다. 나머지 $i-1$ 과 몫 $i-1$ 이 동시에 들어 온다고 가정하면 RSD 가산기가 나머지 i 를 계산하는 동안 3-bit RSD 가산기는 QDL의 입력을 계산한다(멀티플렉서에 의한 시간 지연으로 인하여 RSD 가산기가 조금 늦게 계산을 시작함). 그러면 QDL이 계산을 하는 동안 나머지 i 는 이미 계산되어 있으므로 다음 단(큰 단)의 3-bit RSD 가산기가 동시에 계산할 수 있게 된다. 이 두 계산이 끝나면 다음 단의 RSD 가산기와 QDL은 동시에 나머지 $i+1$ 과 몫 $i+1$ 을 각각 계산할 수 있게 된다.

일반적으로 셀프-타임드 논리에서 계산이 완료되었음을 감지하기 위하여 dual-rail encoding을 이용한다. 본 논문에서 제안하는 계산기에 사용되는 type-1 전가산기는 하드웨어가 크게 증가함 없이 dual-rail로 만들 수 있다. 그림 5는 dual-rail type-1 전가산기의 구조를 보여 준다. 그림에서 보듯이 이 전가산기는 dual-rail encoding이 된 세 입력을 받아서 dual-rail encoding이 된 캐리를 계산하는 캐리 생성기와 합을 계산하는 합 생성기로 구성된다. 이 두 회로는 DCVSL(Differential Cascode Voltage Switch Logic) [9]로 설계하였으며 그 결과는 그림 6과 같다.

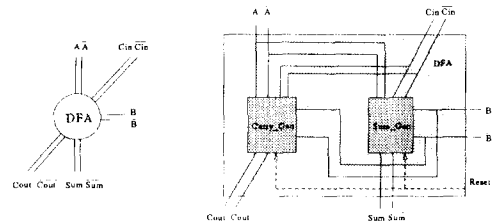
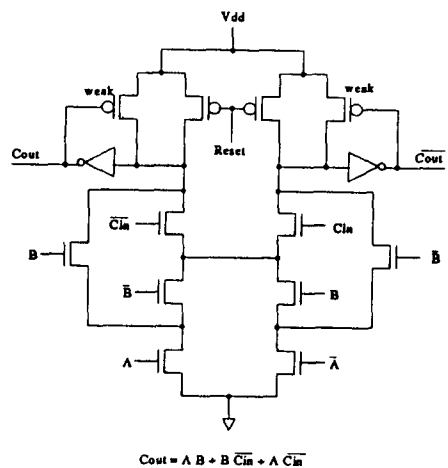


그림 5. Dual-rail 전가산기와 그 내부 구조
 Fig. 5. Dual-rail full adder and its internal structure.



(a)

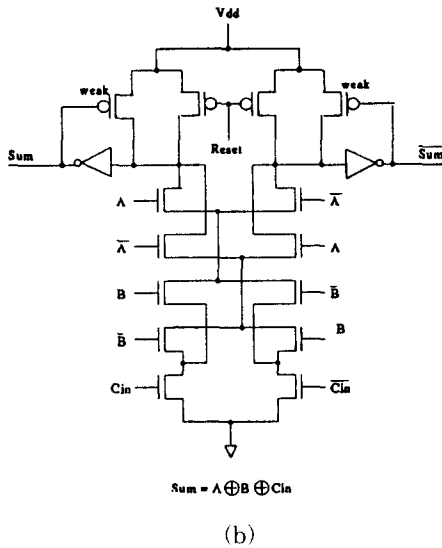


그림 6. Dual-rail 전가산기의 DCVSL 회로 (a) carry generator (b) sum generator
 Fig. 6. DCVSL circuits for dual-rail full adder (a) carry generator, (b) sum generator.

Ⅲ. 타이밍 분석

본 논문에서 제안하는 제산기의 구조에서는 부분 나머지가 계산되어 흘러가는 순방향과 완료 신호가 발생되어 흘러가는 역방향에 대한 타이밍 분석이 모두 필요하다. 각 컴포넌트의 시간 지연을 추정하여 분석해 보면 다음과 같다.

먼저, non-overlapped execution의 경우 각 단에서 순방향의 임계 경로(critical path)는 멀티플렉서, RSD 가산기, QDL의 세 부분 회로로 구성되며 세 개의 작은 단에 걸쳐진다. Backward 방향의 임계 경로는 우선 순방향으로 QDL을 거치고 역방향으로 completion detector, C-element를 거쳐 reset 신호를 만들어 RSD 가산기를 precharge시키고 다시 completion detector 거쳐 reset 신호를 풀어 줌으로써 멀티플렉서를 enable 상태로 바꾸어 주는 경로가 된다. Backward 방향의 임계 경로는 두 개의 작은 단에 걸쳐서 존재하게 된다 성능을 저하시키지 않는 범위에서 링을 구성하는 최소한의 단의 수 N은 다음 식과 같이 신호가 순방향으로 흘러갈 때의 지연 시간(forward latency) L_f 와 반대 방향으로 흘러갈 때의 지연 시간(reverse latency) L_r 의 비에

의하여 결정된다. [3]

$$N = \lceil 2(1 + L_r / L_f) \rceil \quad (3)$$

[3] 에 발표된 제산기의 경우 L_r/L_f 가 1.1이므로 다섯 개의 단으로 링을 구성하였다. 본 논문에서 제안하는 구조는 위에서 설명한 임계 경로를 생각하면 L_r/L_f 가 2.0을 넘지 않을 것으로 예상 되어 여섯 개의 작은 단이면 충분하다. 그러므로 두 개의 큰 단으로도 최대한의 성능을 낼 수 있다.

Overlapped execution의 경우 순방향의 임계 경로는 3-bit RSD 가산기, 멀티플렉서, QDL, 다음 단에서의 멀티플렉서, QDL의 다섯 부분 회로로 구성되며 세 개의 작은 단에 걸쳐서 존재하게 된다. Backward 방향의 임계 경로는 우선 순방향으로 QDL을 거치고, 역방향으로 completion detector, C-element를 거쳐 reset 신호를 만들어 바로 전의 작은 단에서 3-bit RSD 가산기, QDL, latch를 precharge시키고 다시 completion detector와 C-element를 거쳐 reset 신호를 풀어 줌으로써 그 전의 작은 단에서 RSD 가산기, 3-bit RSD 가산기, 두개의 멀티플렉서를 enable 상태로 바꾸어 주는 경로가 된다. Backward 방향의 임계 경로는 두 개의 작은 단에 걸쳐서 존재하게 된다.

그림 7은 overlapped execution의 경우 각각의 작은 단에서 이루어지는 계산에 대한 타이밍을 보여 준다. 이 경우는 네개의 큰 단으로 이루어지므로 작은 단은 여섯 개가 된다. 이 제산기는 클럭 신호에 의하여 동기되어 있지 않고 셀프-타임드 논리에 의하여 동작할 뿐만 아니라 링 구조로 되어 있기 때문에 각 단의 연산이 시작되거나 끝나는 시점은 단에 따라서 다르고, 또한, 같은 단이라도 몇 번째 반복 연산인가에 따라서도 다르므로 링을 풀어서 계산하기 전에는 정확한 타이밍을 보이기 어렵다. 그림 7은 각 시점들의 정확한 타이밍을 보여 주는 것은 아니며 다만 전체적인 동작이 시간적으로 어떻게 이루어지는가를 보여 주는 것이다. 그림에서 보듯이 하나의 작은 단에서 evaluation이 끝나면 그 단은 hold 상태가 되고 바로 다음 작은 단의 evaluation이 시작된다. 동시에 바로 전의 작은 단으로 reset 신호를 보내 precharge 시킨다. 바로 전 작은 단이 precharge 되면 전의 전 작은 단으로의 reset 신호를 풀어 enable 시킨다.

Overlapped execution의 경우에도 L_r/L_f 가 2.0 보다는 작을 것으로 예상되어 여섯 개의 작은 단이면 충분하다. 그러나 두 개의 큰 단이 세개의 작은 단을

가지므로 결과적으로 네 개의 큰 단이 필요하게 된다. 이는 본 논문에서 제안하는 non-overlapped 제산기와 비교하면 훨씬 큰 하드웨어를 필요로 하지만 [3]에 발표된 제산기와 비교하면 단의 수가 다섯 개에서 네 개로 줄어 더 작은 실리콘 면적으로 구현할 수 있다.

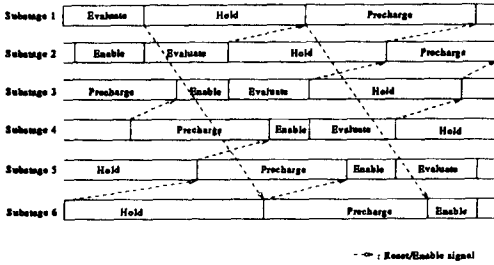


그림 7. 4-단 셀프-타임드 제산기의 타이밍 분석
Fig. 7. Timing analysis of the 4-stage self-timed divider.

IV. 결론

본 논문에서는 기존의 링 구조를 갖는 제산기에 비하여 작은 면적을 차지하면서 성능은 더 좋은 제산기의 구조를 제안하였다. 이는 셀프-타임드 링 구조와 캐리의 전파가 필요 없는 RSD 수 표현 체계를 동시에 사용함으로써 가능하였다.

제안된 제산기는 아직 칩으로 구현되지는 않았으나 그 알고리즘 및 구조는 VHDL^[10]로 표현하고 모의 실험을 함으로써 검증하였다. 또한 non-overlapped execution의 경우 실험적인 레이아웃도 MOSIS 1.2 μm CMOS 설계 규칙을 이용하여 완성하였는데 링을 구성하는 중심부의 면적은 4.2 mm^2 가 되었으며 SPICE를 이용한 모의 실험 결과 한번 나눗셈을 하는데 193 ns가 걸렸다. SPICE를 이용한 모의 실험에서는 나눗셈의 마지막 단계에서 RSD 형태의 몫을 받을림하고 이진 형태로 바꾸어 주기 위한 CPA (carry-lookahead adder 사용)는 제외되었다.

[3]에서는 면적이 7 mm^2 , 속도가 225 ns로 되어 있는데 면적의 경우는 어떠한 부분이 포함된 것인지 명확하지 않아 공정한 비교가 어려우나 속도는 본 논문에서 제안하는 방식이 다소 빠를 것으로 추정된다. 이 실험은 아직 예비적인 것으로 레이아웃에 대한 면적과 성능의 최적화를 통하여 더욱 좋은 결과를 얻을 수도 있을 것으로 생각한다.

제안된 제산기는 mantissa 부분의 나눗셈만 수행하며 IEEE-754에 의한 배정도 부동 소숫점 연산을

하려면 exponent 사이의 뺄셈, 부호의 결정, 등에 대한 고려가 따로 이루어져야 할 것이다.

감사의 글

본 논문에서 제안하는 알고리즘 및 구조에 대한 검증 및 레이아웃을 도와 준 이기종씨께 감사를 드립니다.

參考文獻

[1] T. E. Williams and M. A. Horowitz, "A self-timed chip for division," in *Proc. Conf. Advanced Research in VLSI* (P. Losleben, ed.), pp. 75-95. MIT Press, Mar. 1987

[2] A. Avizienis, "Signed digit number representation for fast parallel arithmetic," *IRE Trans. Electronics Computers*, vol. EC-10, pp. 389-400, Sept. 1961.

[3] T. E. Williams and M. A. Horowitz, "A zero-overhead self-timed 160-ns 54-b CMOS divider," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1651-1661, Nov. 1991

[4] S. Kuninobu, T. Nishiyama, H. Edamatsu, T. Taniguchi, and N. Takagi, "Design of high speed mos multiplier and divider using redundant binary representation," in *Proc. 8th IEEE Symp. Computer Arithmetic*, pp. 80-86, 1987.

[5] A. Vandemeulebroecke, E. Vanzielegem, T. Denayer, and P. G. A. Jespers, "A new carry-free division algorithm and its application to a single-chip 1024-b RSA processor," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 748-756, June 1990.

[6] J. -W. Kang and K. Choi, "An efficient self-timed divider," in *Technical Digest 3rd Int. Conf. on VLSI and CAD*, Taejon, Nov. 1993.

[7] J. Robertson, "A new class of digital division methods," *IRE Trans.*

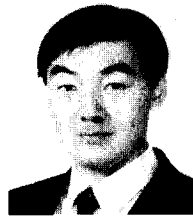
- Electronics Computers, vol. EC-7, pp. 218-222, 1958.
- [8] K. Hwang, Computer Arithmetic: Principles, Architecture, and Design. Wiley, 1979
- [9] L.G. Heller and W. R. Griffin, "Cascode voltage switch logic: A differential CMOS logic family," in ISSCC Digest of Technical Papers, New York, Feb. 1984.
- [10] The Institute of Electrical and Electronics Engineers, Inc., IEEE Standard VHDL Language Reference Manual, IEEE Std 1076-1987, 1988.

— 著 者 紹 介 —



崔 起 榮(正會員)

1955年 8月 30日生. 1978년 서울대학교 전자공학과 졸업. 1980年 한국과학기술원 전기 및 전자공학과 석사. 1989年 미국 Stanford 대학 전기공학과 박사. 1978년 ~ 1983年 (주)금성사 중앙연구소 근무. 1989年 ~ 1991年 미국 Cadence Design System, Inc. 근무. 1991年 ~ 현재 서울대학교 반도체공동연구소 및 전자공학과 조교수. 주관심 분야는 CAD, VLSI 설계 등임



姜 俊 宇(正會員)

1958年 3月 13日生. 1980年 서울대학교 전자공학과 졸업. 1989年 미시간 주립대학교 전기공학과 석사. 1993年 미시간 주립대학교 전기공학과 박사. 1982年 ~ 1987年 금성통신(주) 근무. 1987年 ~ 1988年 금성산전(주) 근무. 현재 전자통신연구소 컴퓨터 H/W 연구실 선임연구원. 주관심 분야는 CAD, 병렬처리 컴퓨터 등임.