韓國컴퓨터情報學會 論文誌
第15卷 第2號, 2010. 2.

2010-15-2-1-2

향상된 연산시간, 회로면적, 소비전력의 절충관계를 위한 혼합가산기 기반 CORDIC

이 병 석*, 이 정 근**, 이 정 아*

CORDIC using Heterogeneous Adders for Better Delay, Area and Power Trade-offs

Byeong-Seok Lee*, Jeong-Gun Lee**, Jeong-A Lee*

요 약

모바일 임베디드 시스템에서는 성능이 우수하면서도 작은 칩 크기와 저 전력의 동작 조건이 요구된다. CORDIC 연산기는 초월 함수들을 효율적으로 계산하는 알고리즘으로, 특유의 하드웨어 간결성으로 인하여 모바일 임베디드 시스템에 매우 적합한 연산기이다. 하지만 CORDIC 알고리즘은 내부 연산의 반복 횟수에 따라 성능이 저하되는 문제점이 있다. CORDIC 연산기를 분석하면 가산기의 영향이 매우 크다는 것을 알 수 있다. 가산기의 알고리즘 종류에 따라 필요 이상의 성능 증가로 인하여 회로 면적과 소비 전력이 증가하면서 성능이 낭비되는 문제점을 해결하기 위하여 연산 시간, 회로 면적, 소비 전력에 대한 보다 심층적인 절충 관계 분석이 필요하다. 본 논문에서는 가산기에 따른 자원 낭비를 최소화하는 방법으로 혼합 가산기를 이용한 CORDIC 연산기를 제안하고, 혼합 가산기를 사용하면 요구 조건에 보다 최적화된 CORDIC 연산기를 설계할 수 있음을 실험 결과를 이용하여 보였다.

Abstract

High performance is required with small size and low power in the mobile embedded system. A CORDIC algorithm can compute transcendental functions effectively with only small adders and shifters and is suitable one for the mobile embedded system. However CORDIC unit has performance degradation according due to iterative inter-rotations. Adder design is an important design unit to be optimized for a high performance and low power CORDIC unit. It is necessary to explore the design space of a CORDIC unit considering trade-offs of an adder unit while satisfying delay, area and power constraints. In this paper, we suggest a CORDIC architecture employing a heterogeneous adder and an optimization methodology for producing better optimal tradeoff points of CORDIC designs.

▸ Keyword : CORDIC, 가산기(Adder), 혼합 가산기(Heterogeneous Adder)

• 제1저자 : 이병석 교신저자 : 이정아
• 투고일 : 2009. 11. 06, 심사일 : 2009. 11. 21, 게재확정일 : 2010. 02. 22.
* 조선대학교 컴퓨터공학과 ** 한림대학교 컴퓨터공학과
※ 이 논문은 2007년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-D00014-I00245)

Ⅰ. 서 론

무선 기술을 이용한 모바일 임베디드 시스템(Mobile Embedded System)은 유비쿼터스(Ubiquitous) 시대에서 매우 중요한 부분이다. 무선 통신의 발전과 모바일 기기의 성능 향상으로 시간과 장소의 제약 없이 실시간으로 다양한 정보와 멀티미디어 서비스를 이용할 수가 있다. 특히, 휴대 전화기의 기술은 눈부시게 발전하면서 음성 통화뿐만 아니라, 음악 청취, 영화 감상, 사진 촬영, 게임 등 생활에 밀접한 만능 통신기기로 자리를 잡고 있다. 그렇지만 무선 환경에서 실시간으로 멀티미디어 데이터를 전송하기 위해서는 기본적으로 휴대용 단말기의 성능이 향상되어야 하지만[1], 배터리로 동작하는 전력상 제약으로 인해 회로 면적과 에너지 소모를 줄이는 노력 또한 필요하다[2]. 따라서 휴대성(Portable)이 강조되는 모바일 임베디드 시스템의 경우에는 성능이 우수하면서 작은 크기(Small Area)와 저 전력(Low Power)의 조건을 만족하는 연산 모듈이 필요하다.

CORDIC(Coordinate Rotation Digital Computer) 알고리즘은 다양한 초월함수들을 효율적으로 계산해 낼 수 있는 연산 기술이다. 최초 Volder[3]에 의해 소개되었으며, 이후 Walther[4]에 의해 정규화 되었다. 이 CORDIC 알고리즘은 시프트와 덧셈(Shift-and-Add)연산의 배열만으로 수행되기 때문에 작은 면적에서 하드웨어의 수행능력을 최대한 효율적으로 사용할 수 있다. 이러한 특유의 하드웨어 간결성으로 이동전화, 무선 통신, 위성 통신, 고속 그래픽 처리, HDTV 등 통신 분야, 멀티미디어 및 그래픽 분야, 신호처리 분야 등 여러 분야에 폭넓게 응용할 수 있다[5-8].

하지만 CORDIC 알고리즘은 내부 연산의 반복 횟수에 따라 성능이 떨어지는 단점이 있다. 따라서 CORDIC 알고리즘은 크게 두 가지 방법으로 개선되고 있다. 하나는 Signed-Digital Adder, Carry Save Adder 등 Redundant Adder를 이용하여 연산 속도를 빠르게 하는 방법이고, 다른 하나는 반복 횟수를 줄이는 방법이다. 이렇게 다양한 종류의 CORDIC 알고리즘은 각각의 장·단점이 있으며, 사용하는 목적에 따라 성능의 차이가 발생한다. 따라서 다양한 환경의 모바일 임베디드 시스템 분야와 여기에 필요한 연산 모듈 역시 시스템의 목적에 적합한 CORDIC 알고리즘을 선택하여 구현하는 것이 중요하다. 본 논문에서는 CORDIC 알고리즘에서 사용하는 가산기를 혼합 가산기(Heterogeneous Adder)로 대체하여 보다 향상된 연산 시간, 회로 면적 및 전력 소비에 대한 절충관계를 갖는 혼합 가산기 기반 CORDIC 구조를 제시한다.

Ⅱ. CORDIC 알고리즘

2.1 기본 CORDIC 알고리즘

벡터 회전 연산에서 $[x(i)\,y(i)]^T$를 회전각도 θ에 대한 회전 연산 $[x(i+1)\,y(i+1)]^T$이 있을 때, 회전 행렬(Rotation Matrix)는 식 (1)처럼 정의할 수 있다.

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad \cdots\cdots (1)$$

기본 CORDIC 알고리즘에서 원소 각(Elementary Angle)인 $a(i)$는 [3], [4]에서 정의하고 있으며, 그 식은 (2)과 같다.

$$a(i) \triangleq \tan^{-1}(2^{-i}) \quad \cdots\cdots (2)$$

여기서 i번째에서 $\alpha(i)$만큼 벡터 회전을 하였을 때, i번째 마이크로 회전은 식 (3)으로 표현할 수 있다[3,4].

$$\begin{aligned} \begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} &= \cos\alpha(i) \begin{bmatrix} 1 & -\tan\alpha(i) \\ \tan\alpha(i) & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \\ &= \cos\alpha(i) \begin{bmatrix} 1 & -2^{-i} \\ 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \\ &= \sqrt{1+2^{-2i}} \begin{bmatrix} 1 & -2^{-i} \\ 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad \cdots\cdots (3) \end{aligned}$$

식 (3)을 보면 CORDIC 알고리즘은 마이크로 회전 연산을 수행할 때마다 $\cos\alpha(i)$을 곱하는 만큼의 벡터 크기가 변하게 된다. 이를 보상해주는 단계를 스케일 보상 과정(Scaling Phase)이라고 하며, 목표각인 N 번 회전 연산을 수행한 후에 식(4)와 같이 스케일 요소(Scale Factor)인 P를 곱해서 원래의 벡터 값을 보상한다.

Initialization:

$x(0)=P,\ y(0)=0,\ z(0)=\theta$

Micro-rotation:

For i = 0 to

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & -\mu(i)2^{-i} \\ \mu(i)2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}$$

Angle update:

$z(i+1)=z(i)-\mu(i)a(i)$,

여기서 $a(i)=\tan^{-1}(2^{-i})$

END

Scaling:

$$\begin{bmatrix} x_f \\ y_f \end{bmatrix} = P \begin{bmatrix} x(N) \\ y(N) \end{bmatrix} = \left(\prod_{i=0}^{N-1}\sqrt{1+2^{-2i}}\right)^{-1} \begin{bmatrix} x(N) \\ y(N) \end{bmatrix}$$

그림 1. 기본 CORDIC의 마이크로 회전과 스케일 요약
Fig. 1. Micro-rotation and Scaling of CORDIC

$$P=\left(\prod_{i=0}^{N-1}\sqrt{1+2^{-2i}}\right)^{-1} \quad \cdots\cdots (4)$$

$\mu(i)$를 i번째 회전 연산에 대한 목표각의 방향 {-1,1}이라고 할 때, 그림 1은 목표각 θ에 대한 기본적인 CORDIC 알고리즘에서 원형 모드(Circular Mode)에 대한 기본적인 연산 절차를 보이고 있다. 그림 2는CORDIC 알고리즘에 대한 하드웨어 구조로서, 매우 간단하게 하드웨어로 구현할 수 있음을 볼 수 있다. 그림 2에서 각 회전 연산에 필요한 원소 각인 $\alpha(i)$은 ROM에 저장한다.

2.2 CORDIC 연산기와 가산기와의 관계

그림 2를 보면 CORDIC 연산기는 레지스터(Register), 시프터(Shifter), 가산기(Adder) 등으로 구성할 수 있다. 입력 비트의 너비가 동일할 때, 레지스터와 시프터의 연산 속도, 회로 면적, 소비 전력은 변화가 발생하지 않는다. 하지만, 가산기에서 사용하는 알고리즘의 종류에 따라 성능의 차이가 발생한다. TSMC(Taiwan Semiconductor Manufacturing Company) 0.15um[9] 셀 라이브러리를 이용하고, 가산기 알고리즘을 가장 기본적인 RCA(Ripple Carry Adder)를 사용하였을 때, 입력 비트의 너비가 16, 24, 32, 52, 64, 128 비트에 대한 CORDIC 연산기 중에서 RCA 가산기가 차지하는 비율에 대한 결과는 표 1과 같다. 표 1에서 CORDIC 연산기의 연산 속도(Delay)에 대한 가산기 비율은 평균 94%로서 가산기에 대한 비중이 매우 높다는 것을 알 수 있다. 또한, 회로 면적(Area)과 소비 전력(Power)은 각각 43%와 55%로 가산기의 비중이 매우 높은 것을 볼 수가 있다. 이것은 CORDIC 연산기 중에서 가산기의 비중이 매우 높고 중요하며, 가산기의 성능이 CORDIC 연산기의 성능과 매우 밀접하다는 것을 의미하고 있다.

그림 3은 대표적인 가산기 알고리즘 중에서 RCA와 CLA(Carry Look-ahead Adder)를 사용하였을 경우에 CORDIC 연산기의 대한 연산 속도, 회로 면적, 소비 전력에 대한 결과이다. (실험 환경은 IV. 실험 및 결과에서 자세히 설명한다.) 그림 3(a)를 보면 CLA를 사용한 CORDIC 연산기가 RCA를 사용한 CORDIC 연산기보다 평균 1.60배(최대 2.22배) 성능이 향상되는 것을 볼 수 있다. 하지만 그림 3(b)와 그림 3(c)를 보면 회로 면적과 소비 전력은 각각 평균 1.12배(최대 1.17배)와 1.15배(최대 1.20)배 증가하는 문제가 발생한다.

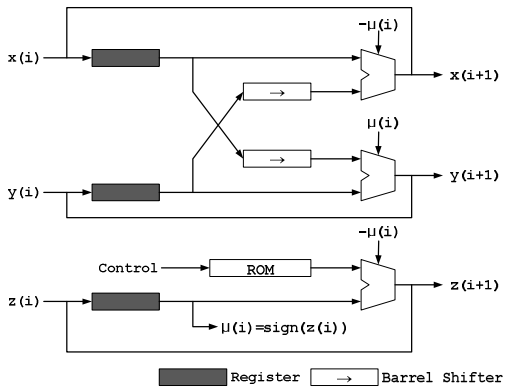
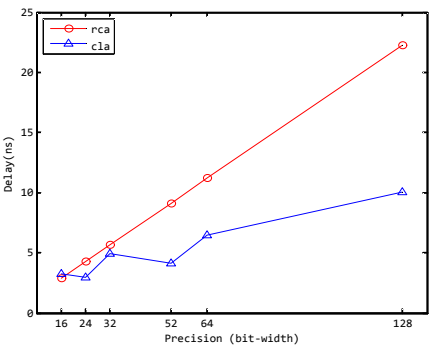


그림 2. CORDIC 알고리즘의 하드웨어 구조
Fig 2. Hardware Architecture of CORDIC

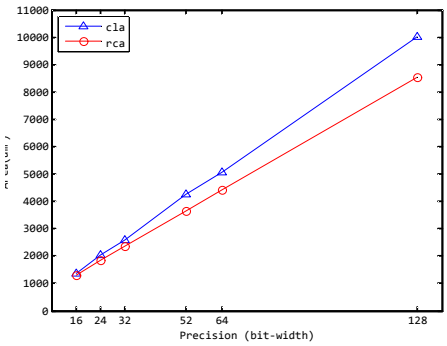
표 1. CORDIC 연산기에서 가산기(RCA) 비율
Table 1. Rate of Adder(RCA) in CORDIC unit

Bit-width	Delay	Area	Power
16	88.54%	35.33%	45.80%
24	92.48%	39.87%	51.48%
32	94.15%	42.42%	54.60%
52	96.48%	45.48%	58.80%
64	97.05%	46.61%	59.82%
128	98.51%	49.03%	62.63%
Mean	94.64%	43.14%	55.52%

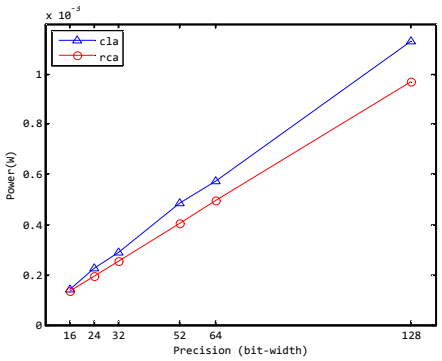
그림 3을 보면 CORDIC 연산기를 설계하면서 다음과 같은 문제가 발생할 수 있다. 성능 향상을 위해서 CLA를 가산기 알고리즘으로 사용하면 연산 속도에 대한 성능이 필요 이상 향상되지만, 요구하는 회로 면적과 소비 전력이 필요 이상 증가하는 문제가 발생할 수 있다. 이와 반대로 요구하는 회로 면적과 연산 속도를 만족하기 위해서 RCA를 가산기 알고리즘으로 사용하면 필요 이상 성능이 저하되는 문제가 발생할 수 있다. 128 비트 CORDIC 연산기를 예를 들어 설명하겠다. 먼저 설계자가 요구하는 CORDIC 연산기의 연산 속도는 15ns라고 할 때, RCA를 가산기 알고리즘으로 사용하면 CORDIC 연산기의 연산 속도는 22.28ns가 되므로 설계자의 요구 조건을 만족할 수가 없게 된다. 설계자는 CLA를 가산기 알고리즘으로 사용하면 CORDIC 연산기의 연산 속도는 10.03ns로 설계자의 요구 조건은 만족하지만 필요 이상 성능이 향상되면서 회로 면적과 소비전력은 1.17배와 1.16배 증가되는 문제가 발생한다. 이는 필요 이상의 생산 단가와 전력 소비가 증가되는 문제가 발생할 수가 있다. 다음으로 설계자가 요구하는 CORDIC 연산기가 사용할 수 있는 회로 면적이 9,000um^{2}라고 할 때, CLA를 가산기 알고리즘의 회로 면적이 10,018um^{2}이나 되므로 사용할 수가 없게 된다. 회로 면적이 8,546um^{2}인 RCA를 사용하면 설계 요구 조건을 만족하지만 0.4배 정도 성능이 저하되는 문제가 발생한다. 가산기 알고리즘의 특성상 필요 이상의 성능 증가로 인하여 자원이 낭비 되는 문제가 발생한다. 본 논문은 이러한 문제를 해결하기 위하여 CORDIC 연산기에서 혼합 가산기(Heterogeneous Adder)를 이용하여 설계자가 요구하는 조건은 만족하는 절충 방법을 제안한다.



(a)



(b)



(c)

그림 3. 가산기 종류에 대한 CORDIC 연산기의 (a) 연산속도 (b) 회로 면적 (c) 소비 전력 결과
Fig. 3. Variation of (a) delay (b) area and (c) power consumption in CORDIC unit

III. 혼합 가산기

3.1 혼합 가산기 설계 방법

모바일 임베디드 시스템에서는 작은 크기와 저 전력을 요구하기 때문에 연산 모듈 역시 작은 회로 면적과 함께 소비 전력이 작은 칩을 요구하고 있다. 하지만, 회로 면적과 소비 전력이 작으면 그 만큼 연산 모듈의 속도가 저하되는 단점이 있다. 따라서 설계가 가능한 회로 면적을 최대한 활용하는 연산기가 필요하지만, 기존에 개발된 가산기 구조는 그림 4에서 보이는 바와 같이 ①과 ②에 위치한 영역에 대한 설계는 불가능하다[10]. 이러한 문제에 대한 해결책의 하나로 혼합가산기 구조가 있다[11, 12]. 혼합 가산기는 주어진 연산 시간, 회로

면적, 소비 전력으로 최적화하는 가산기 구조를 설계할 수 있다. 가산기 구조는 모든 연산의 기본 구조이기 때문에 다양한 연산기 구조에서 혼합가산기를 활용할 수 있다.

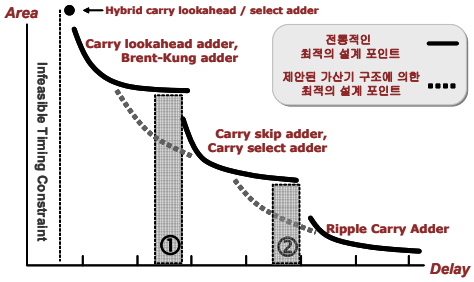


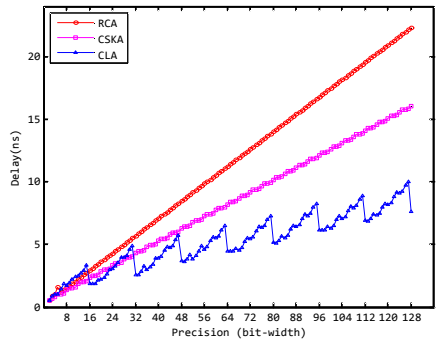
그림 4. 가산기의 설계 공간 (연산시간-회로면적 절충 관계)
Fig. 4. Design space of Adder(delay-area trade-offs)

3.2 가산기의 특징

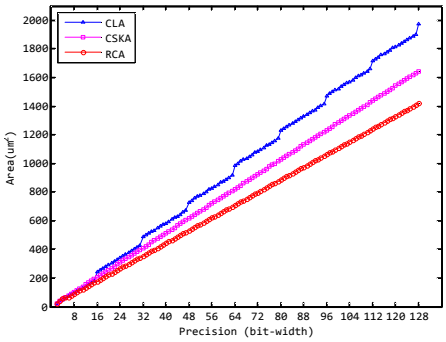
가산기는 대부분의 디지털 회로에서 사용하는 연산장치이다. 이러한 가산기는 캐리 전달 방법에 따라 성능이 달라진다. 즉, 캐리 전달 시간(Carry propagation delay)이 빠를수록 가산기의 연산 속도 역시 빨라진다. 하지만 캐리 전달 시간이 빠른 방법은 회로 면적 또한 증가하므로 시스템의 전체 크기가 커지면서 단가가 올라가는 단점이 있다. 따라서 가산기의 성능 향상을 위해 효율적인 캐리 전달 방법을 연구하고 있다[13].

대표적인 가산기의 종류는 RCA(Ripple Carry Adder), CLA(Carry Look-ahead Adder), CSKA(Carry Skip Adder) 등이 있다. RCA는 가장 간단한 구조로서 순차적으로 캐리 신호를 전달한다. CSKA는 특정 블록의 비트 패턴을 이용하여 캐리 신호를 전달한다. CLA는 입력 값에 따라 캐리 생성 식을 이용하여 계산을 한다. 가산기는 종류에 따라 연산 속도와 회로 면적에 대한 차이가 발생한다. 특히, 입력 비트의 너비가 클수록 이러한 차이는 크게 발생한다. 그림 5는 TSMC 0.15um 셀 라이브러리[9]를 이용하여 RCA, CSKA, CLA 등의 가산기에서 입력 비트 너비에 대한 연산 속도, 회로 면적과 소비 전력을 보여주고 있다. 여기서 가산기를 선택하는 중요한 문제가 발생한다.

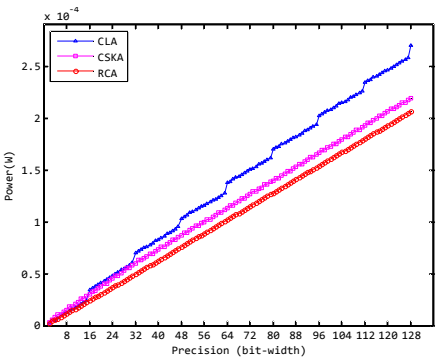
그림 5에서 가산기의 종류에 따라 입력 비트의 너비에 따른 연산 속도, 회로 면적과 소비 전력이 다르다는 것을 볼 수 있다. 따라서 모바일 임베디드 시스템이 요구하는 설계 조건이 기존의 가산기로 설계할 수 없는 부분에 위치할 경우에는 요구하는 조건보다 연산 속도가 빠르거나 늦은 가산기를 선택해야만 하는 문제가 발생한다. 예를 들어 모바일 임베디드 시스템이 요구하는 연산 속도가 설계할 수 없는 영역에 있을 경우에는 필요 이상으로 연산 속도가 빠른 가산기를 선택해야만 한다. 이렇게 되면 회로 면적이 증가할 뿐만 아니라, 제품 생산 비용 역시 증가하게 된다. 반대로 회로 면적이 가산기가 설계할 수 없는 영역에 있으면, 연산 속도가 늦은 가산기를 선택해야 하며, 이는 전체 시스템의 성능 저하를 초래한다.



(a)



(b)



(c)

그림 5. 입력비트의 너비에 따른 3종류의 가산기의 (a) 연산 속도 (b) 회로 면적 (c) 소비전력 관계
Fig. 5. Variations of (a) delay (b) area and (c) power consumption with three types of adders

3.3 혼합 가산기 구조

전통적인 단일 가산기만 사용하면 그림 5에서 보인 바와 같이 연산 속도와 회로 면적을 만족하기 위해서는 필요 이상으로 연산 속도와 회로 면적이 증가하게 된다. 하지만 두 종류 이상의 가산기를 혼합하여 사용하면, 그림 4에서 점선으로 표현한 연산 시간과 회로 면적의 절충관계를 가지는 최적화된 가산기를 설계할 수 있다. 이러한 가산기를 혼합 가산기(Heterogeneous Adder)라고 하며[11], 이에 대한 기본 예제는 그림 6과 같다. 최근에는 연산 속도와 회로 면적뿐만 아니라 소비 전력에 대한 혼합가산기 연구가 진행되었다[12].

혼합 가산기의 설계 변수는 제한 조건을 만족하는 연산 속도, 회로 면적, 소비 전력, 캐리 전달 속도, 입력 비트의 너비 등이 비선형적이다. 이러한 문제를 해결하기 위해서 정수 선형 프로그램(ILP; Integer Liner Program)을 사용한다.

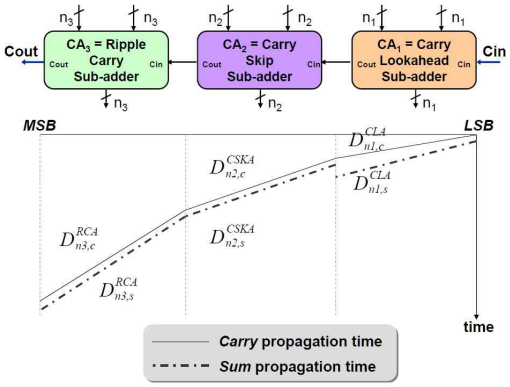


그림 6. 혼합 가산기의 연산 모델링
Fig. 6. Delay modeling of a heterogeneous adder

IV. 실험 및 결과

4.1 실험 환경

CORDIC 알고리즘에서 혼합 가산기 구조를 적용하기 위하여 먼저 2비트부터 128비트까지 RCA, CSKA, CLA 가산기를 Verilog HDL(Hardware Description Language)를 이용하여 구현하였다. 여기서 RCA와 CLA는 Synopsys사의 DesignWare[14]의 IP(Intellectual Property)를 이용하였다. TSMC의 0.15um 셀 라이브러리[9]와 Design Compiler[15]를 이용하여 합성한 후 회로 면적과 연산 속도를 구하였다. 10만개의 랜덤 값을 입력 값으로 한 테스트 벤치와 VCS-MX[16]를 이용하여 게이트 레벨 시뮬레이션을 수행한 후, PrimeTime-PX[17]에서 게이트 레벨 시뮬레이션 결과 값을 이용하여 소비 전력을 구하였다. 혼합 가산기의 구현은 연산 속도-회로 면적에 대한 ILP 모델링[11]과 연산 속도-소비 전력에 대한 ILP 모델링[12]을 이용하였다. 혼합 가산기의 설계 변수를 구하기 위해서 ILP 모델링은 lp_solve[18]를 이용하였다. CORDIC 연산기는 OpenCores의 CORDIC IP[19]를 이용하여 구현하였다.

4.2 혼합 가산기를 이용한 128비트 CORDIC 연산기

혼합 가산기를 이용하여 연산 시간, 회로 면적, 소비 전력에 대한 절충 관계를 분석하기 위해 128비트 CORDIC 연산기를 이용하여 실험을 진행하였으며, 그 결과는 그림 7과 같다. CORDIC 연산기에서 한 비트는 부호 비트이며, 1비트는 시프트 연산만이 필요하므로 실제로는 입력 비트의 너비에서 두 개 비트는 연산을 할 필요는 없다. 따라서 128비트 CORDIC 연산기는 126비트 혼합 가산기 3개가 필요하다. 혼합 가산기는 2 ~ 126비트의 RCA, CSKA, CLA 가산기 조합으로 구성한다. 그림 7에서 '||' 기호는 가산기 종류의 조합을 의미하며, 숫자는 입력 비트의 너비를 의미한다. 예를 들어 RCA046||CSKA080은 46비트 RCA와 80비트 CSKA의 조합을 의미하며, 입력 비트가 0부터 79비트는 CSKA로, 80부터 125비트는 RCA로 입력하는 것을 의미한다. θ_{delay}, θ_{area}, θ_{power}는 각각 연산 속도, 회로 면적, 소비 전력에 대한 제한 값을 의미 한다. 128비트 CORDIC 연산기에서 가산기를 제외한 나머지 유닛에 대한 연산 속도, 회로 면적, 소비 전력을 $CORDIC128_{delay}$, $CORDIC128_{area}$, $CORDIC128_{power}$라 하고, 126비트 RCA에 대한 연산 속도, 회로 면적, 소비 전력을 $RCA126_{delay}$, $RCA126_{area}$, $RCA126_{power}$라고 하며, 126비트 CLA에 대한 연산 속도, 회로 면적, 소비 전력을 $CAL126_{delay}$, $CLA126_{area}$, $CLA126_{power}$라고 할 때, ILP에서 각각 제한 값에 대한 범위는 식 (5)와 같으며, θ_{delay}는 0.1ns, θ_{area}는 4.86um^2, θ_{power}는 1uW 간격으로 증가하면서 실험을 하였다.

$$CORDIC128_{delay} + CLA126_{delay} \leq \theta_{delay} \leq CORDIC128_{delay} + RCA126_{delay}$$

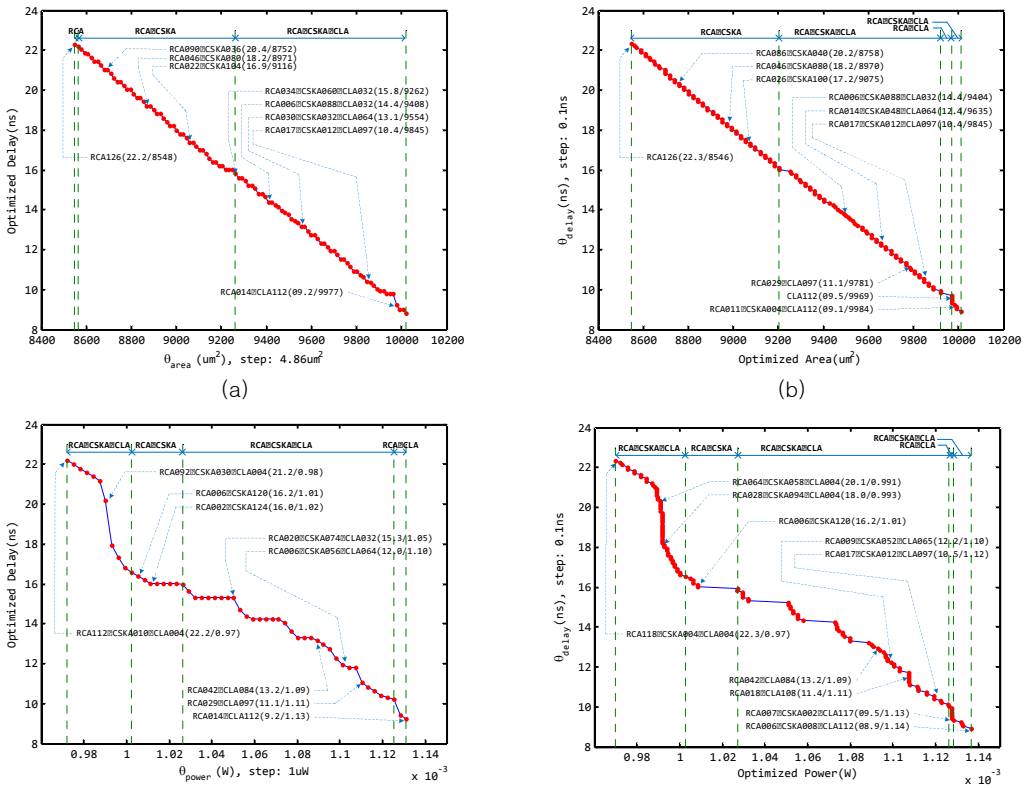


그림 7. 혼합 가산기를 이용한 CORDIC 연산기 최적화 결과: (a) 회로 면적에 대한 연산속도 최적화 (b) 연산속도에 대한 회로 면적 최적화 (c) 소비 전력에 대한 연산 속도 최적화 (d) 연산 속도에 대한 소비 전력 최적화

Fig. 7. Results of optimizations in CORDIC using heterogeneous adder: (a) area-constrained delay optimization (b) delay-constrained area optimization (c) power-constrained delay optimization (d) delay-constrained power optimization

$$CORDIC128_{area}+(RCA126_{area}\times 3) \leqq \theta_{area} \leqq CORDIC128_{area}+(CLA126_{area}\times 3)$$

$$CORDIC128_{power}+(RCA126_{power}\times 3) \leqq \theta_{power} \leqq CORDIC128_{power}+(CLA126_{power}\times 3) \quad \cdots\cdots (5)$$

그림 7(a)는 회로 면적이 제한 값인 θ_{area}에 대한 연산 속도의 최적화 그래프로 θ_{area}=8,562um^{2}까지는 RCA, θ_{area}=9,262um^{2}까지는 RCA‖CSKA, θ_{area}=10,020um^{2}까지는 RCA‖CSKA‖CLA 가산기 조합으로 혼합 가산기를 구성하는 것이 최적이라는 결과를 보여주고 있다.

그림 7(b)는 연산 속도가 제한 값인 θ_{delay}에 대한 회로 면적 최적화 그래프이고, θ_{delay}에 따라 RCA‖CSKA‖CLA, RCA‖CLA, RCA‖CSKA‖CLA, RCA‖CSKA 조합의 혼합 가산기가 최적이라는 결과를 보여주고 있다. 여기서 θ_{delay}가 9.1ns, 9.5ns, 11.ns에서만 RCA‖CSKA‖CLA, CLA, RCA‖CLA 조합이 최적인 조합이라는 결과를 보여준다. 이것은 그림 5(a)처럼 CLA는 입력 비트의 너비에 따라 연산 속도의 차이가 발생하기 때문에 중간 조합이 다른 혼합 가산기가 설계될 수도 있다.

그림 7(c)와 그림 7(d)는 소비 전력과 관련된 혼합 가산기의 최적 결과를 보여주고 있다. 그림 7(c)에서 소비 전력이 제한 값인 θ_{power}에 대하여 연산 속도가 최적화된 혼합 가산기로, RCA‖CLA, RCA‖CSKA‖CLA, RCA‖CSKA, RCA‖CSKA‖CLA 순서로 최적화된 가산기 조합을 보여주고 있다. 그림 7(d)는 연산 속도가 제한 값인 θ_{delay}에 대한 소비 전력에

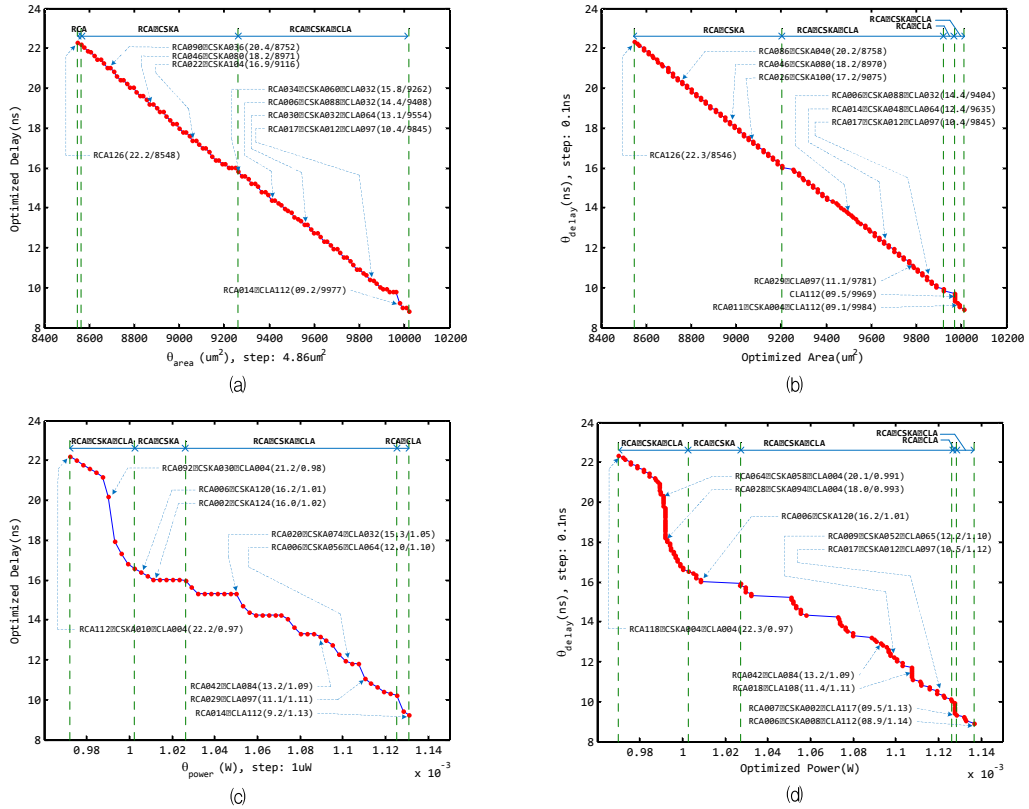


그림 7. 혼합 가산기를 이용한 CORDIC 연산기 최적화 결과: (a) 회로 면적에 대한 연산속도 최적화 (b) 연산 속도에 대한 회로 면적 최적화 (c) 소비 전력에 대한 연산 속도 최적화 (d) 연산 속도에 대한 소비 전력 최적화

Fig. 7. Results of optimizations in CORDIC using heterogeneous adder: (a) area-constrained delay optimization (b) delay-constrained area optimization (c) power-constrained delay optimization (d) delay-constrained power optimization

대한 최적화 그래프이며, RCA‖CSKA‖CLA, RCA‖CLA, RCA‖CSKA‖CLA, RCA‖CSKA, RCA‖CSKA‖CLA로 구성한 최적화된 혼합 가산기 결과를 보여준다.

일반적인 RCA와 CLA 가산기를 사용했을 경우와 혼합 가산기를 사용했을 경우에 대한 실험 결과는 표 2와 같다. RCA

또는 CLA만을 사용할 수 있을 때, 연산 속도가 18ns를 만족하는 CORDIC 연산기를 설계하기 위해서는 오직 표 2의 ⑤인 CLA만 CORDIC의 가산기로 사용할 수밖에 없다. 만약에 표 2의 ②와 같이 RCA028‖CSKA094‖CLA004 조합의 혼합 가산기를 이용하면 설계자가 요구하는 18ns를 만족시킬 수 있다. 또한 CLA를 사용했을 때 보다는 회로 면적과 소비 전력을 각각 10%와 12%정도 절약 할 수 있다. 설계자가 요구하는 회로 면적이 9270um²라고 하면, 기존의 설계 방법으로는 이 조건을 만족하기 위해서 표 2의 ①인 RCA를 CORDIC 연산기의 가산기로 사용해야만 한다. 여기서 ③인 RCA034‖CSKA060‖CLA032 조합의 혼합 가산기를 이용하면 설계자가 요구하는 회로 면적의 조건을 만족하면서도 RCA 보다 연산 속도가 1.4배 성능이 향상된 CORDIC 연산기를 설계할 수가 있다. 설계자가 저 전력 설계를 위해 1.1mW 이하의 CORDIC 연산기를 요구할 경우, 기존의 방법으로는 요구하는 조건을 만족하기 위해서 표 2의 ①과 같이 RCA를 가산기로 사용할 수밖에 없다. 하지만, ④인 RCA014‖CSKA048‖CLA064 조합의 혼합 가산기를 사용하면 저 전력의 조건을 만족하면서 RCA를 사용한 CORDIC 연산기와 비교하면 연산 성능은 1.8배 향상된 CORDIC 연산기를 설계할 수가 있다. 예를 보여준 것 같이 혼합 가산기를 사용한 CORDIC 연산기는 기존의 가산기를 사용한 것 보다 설계자의 요구조건을 성능 저하나 자원의 낭비 없이 매우 유연한 설계 환경을 제공할 수 있다.

표 2. 128비트 CORDIC 설계 예
Table 2. Example design of CORDIC 128-bit

No	Adder Type in CORDIC	Delay (ns)	Area (um^2)	Power (mW)
①	RCA	22.3	8546	0.972
②	RCA028‖CSKA094‖CLA004	18.0	9000	0.993
③	RCA034‖CSKA060‖CLA032	15.8	9262	1.028
④	RCA014‖CSKA048‖CLA064	12.4	9635	1.100
⑤	CLA	10.0	10018	1.131

V. 결론

CORDIC 알고리즘은 간단한 회로만으로 구현할 수 있기 때문에 모바일 시스템에서는 매우 적합한 알고리즘이라고 할 수 있다. CORDIC 알고리즘에서 가산기의 비중이 높으므로 가산기의 알고리즘에 따라 성능과 요구하는 자원이 결정된다. 따라서 CORDIC 연산기의 설계 환경에서 가산기는 매우 중요하다고 할 수 있다. 기존 가산기를 이용한 CORDIC 연산기는 설계자가 요구하는 조건을 만족하기 위해 CLA를 사용하면 필요 이상 성능이 올라가면서 필요한 회로 면적과 소비 전력 또한 증가하는 문제점이 있으며, RCA를 사용하면 회로 면적과 소비 전력에 대한 요구 조건을 만족할 수 있으나 성능이 저하되는 문제점이 있다. 혼합 가산기를 사용하면 요구 조건에 최적화된 CORDIC 연산기를 설계할 수 있으므로, 성능 저하 없이 필요 이상의 자원 낭비를 줄일 수 있다.

참고문헌

[1] 윤경섭, "휴대용 단말기를 위한 실시간 무선 영상 음성 전송 기술," 한국컴퓨터정보학회논문지, 제 14권, 제 4호, 111-117쪽, 2009년 4월.

[2] 김현희, 김지홍, "모바일 3D 그래픽스를 위한 저전력 텍스쳐 매핑 기법," 한국컴퓨터정보학회논문지, 제 14권, 제 2호, 45-57쪽, 2009년 2월.

[3] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Computers, Vol. C-8, pp. 330 - 334, Sep. 1959.

[4] J. S. Walther, "A Unified algorithm for elementary function," In 1971 Proc. Joint Spring Comput. Conf., pp. 379-385, 1971.

[5] T Lang, E Antelo, "High-Throughput CORDIC-based Geometry Operations for 3D Computer Graphics", IEEE Trans. on Computers, Vol.54, No.3, pp.347-361, Mar. 2005

[6] E Antelo, J Villalba and EL Zapata, "A low-latency pipelined 2D and 3D CORDIC processors," IEEE Transactions on Computers, Vol. 57, pp. 404-417, 2008.

[7] F. Angarita, M. Canet, T. Sansaloni, A. Perez-Pascual, and J. Valls, "Efficient mapping of CORDIC algorithm for OFDM-based WLAN," Journal of Signal Processing Systems, Vol. 52, pp. 181-191, 2008.

[8] Srinivasa Chaitanya. K, P. Muralidhar, C.B. Rama Rao, "Implementation of Cordic Based Architecture for WCDMA/OFDM Receiver," European Journal of Scientific Research, Vol.36 No.1, pp.65-78, 2009

[9] "*TCB015GHD TSMC 0.15um Core Library Databook*," TSMC, Release 1.0, 2003

[10] C. Nagendra, M.J. Irwin, R.M. Owens, "Area-time-power tradeoffs in parallel adders," In IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 43, pp. 689-702, Oct. 1996

[11] J.G. Lee, J.A. Lee, B.S. Lee and Milos D. Ergcegovac, "A design method for heterogeneous adders," Proc. Int. Conf. Embedded Software and Systems, Lecture Notes in Computer Science, June 2007

[12] S.H Kwak, J.G. Lee, E.G. Jung, D.S. Har, Milos D. Ercegovac and J.A Lee, "Exploration of Power-Delay Trade-Offs with Heterogeneous Adders by ILP," Journal of Circuits, Systems, and Computers, Vol. 18, No. 4, pp. 787 - 800, 2009

[13] Y. Wang, C. Pai, X. Song, "The design of hybrid carry-lookahead/carry-select adders," In IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing ,Vol. 49, Jan. 2002

[14] "*DesignWare Building Block IP User Guide*," Synopsys, July, 2009

[15] "*Design Compiler User Guide*," Synopsys, Sep. 2008

[16] "*VCS MX/VCS MXi User Guide*," Synopsys, Jun. 2009

[17] "*PrimeTime PX User Guide*," Synopsys, Dec. 2008

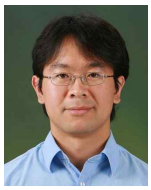
[18] lp_solve reference guide, *http://web.mit.edu/lpsolve_v55013/doc/index.htm*

[19] OpenCores CORDIC core, *http://www.opencores.org/project,cordic*

저 자 소 개

이 병 석

1997 : 조선대학교 이학사
1999 : 조선대학교 이학석사
2007 : 조선대학교 이학박사(수료)
2009 - 현재 : (주)디선테크 부설연구소 수석연구원
관심분야 : 임베디드 시스템, 디지털 연산기, 저온도 프로세서



이 정 근

1996 : 한림대학교 학사
1998 : 광주과학기술원 석사
2005 : 광주과학기술원 박사
2005 - 2007 : 캠브리지대학교 컴퓨터랩, 박사후 연구원
2008 - 현재 : 한림대학교 컴퓨터공학과 교수
관심분야 : 멀티코어 프로세서, 비동기 회로, 연산기 구조

이 정 아

1982 : 서울대학교 공학사
1985 : 인디애나 주립대학교 공학석사
1990 : 캘리포니아 주립대학교(UCLA) 공학박사
1990 - 1995 : Assistant Professor, University of Houston USA.
1995 - 현재 : 조선대학교 컴퓨터공학과 교수
관심분야 : 컴퓨터 구조, 고속 디지털 연산기, 특수 용도의 VLSI 구조