

논문 2008-45SD-11-8

에러 내성을 갖는 저전력 MAC 연산기 설계

(A Design of Low Power MAC Operator with Fault Tolerance)

정한샘*, 구성관**, 정기석*

(Han-Sam Jung, Sung-Kwan Ku, and Ki-Seok Chung)

요약

오늘날 사용되는 휴대용 전자 장치들은 점점 더 강력한 DSP 능력을 요구하고 있다. 때문에 오늘날의 DSP 알고리즘들은 점점 더 그 복잡도가 높아져 가고 있는 추세이다. DSP 알고리즘의 복잡도가 높아져 감에 따라 DSP 디자인에서 결함이 발생할 확률도 높아져 가고 있다. 그렇기 때문에 디자인에서 발생한 결함을 극복할 수 있는 Fault Tolerance 설계의 필요성이 제시된다. 또한 DSP 알고리즘이 휴대용 전자 장치들에서 사용되기 위해서는 기본적으로 저전력 설계가 필요하다. 하지만 Fault Tolerance 기능을 구현하고자 한다면 추가 모듈로 인해 많은 전력소비와 증가하는 회로크기를 감수해야 한다. 이러한 이슈들을 가지고 본 논문에서는 배럴 시프터를 이용하여 구현된 결함 포용성 저전력 MAC 연산기 구조를 제안한다.

Abstract

As more DSP functionalities are integrated into an embedded mobile device, power consumption and device reliability have emerged as crucial issues. As the complexity of mobile embedded designs increases very rapidly, verifying the functionality of the mobile devices has become extremely difficult. Therefore, designs with error (fault) tolerance are often required since these capabilities will enable the design to operate properly even with some existence of errors. However, designs with fault tolerance may suffer from significant power overhead since fault tolerance is often achieved by resource replication. In this paper, we propose a low power and fault tolerant MAC (multiply-and-accumulate) design. The proposed MAC design is based on multiple barrel shifters since MAC designs with barrel-shifters and adders are known to be excellent in terms of power consumption.

Keywords : Fault Tolerance, Low Power, MAC, Barrel Shifter

I. 서론

오늘날 IT산업의 발전과 더불어 휴대용 전자 장치들의 소비는 점점 증가하고 있는 추세이다. 또한 이렇게 소비되는 휴대용 전자 장치들은 점점 더 강력한 DSP(Digital Signal Processing) 능력을 요구하고 있다. 휴대용 전자 장치에서 DSP 능력이 중요시됨에 따라

DSP 응용 분야에서는 시간지연이나 회로크기를 줄이는 문제 등에 비해서 전력 소모를 줄이는 것 또한 중요한 요소로 부각되고 있다. 그렇기 때문에 DSP 연산에 있어 정확한 동작에 영향을 주지 않는 범위에서 전력 소모를 줄이기 위한 약간의 오차는 허용되고 있는 추세이다. FIR(Finite Impulse Response) 필터와 FFT(Fast Fourier Transform)와 같은 DSP(Digital Signal Processing) 함수들은 많은 MAC(Multiply and Accumulate) 연산을 요구한다. 그렇기 때문에 저전력 DSP 시스템을 구현하기 위해 DSP의 주된 연산인 MAC 연산기의 전력 소모를 줄이기 위한 설계가 중요한 요소 중 하나가 되었다.

일반적으로 MAC 연산기의 경우 곱셈기와 덧셈기를 이용하여 구현되어진다. MAC 연산기를 구성하는 모듈 중에서도 일반적인 곱셈기는 전력소비와 회로크기가 크

* 정희원, 한양대학교 전자컴퓨터통신공학과
(Dept. of Electronics, Computer & Communication Engineering, Hanyang University)

** 정희원, 삼성테크윈 이미징사업부
(OPTICS & IMAGING DIVISION, Samsung Techwin)

※ 본 연구는 지식경제부 및 정보통신진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음.
(IITA-2008-C1090-0804-0009)

접수일자: 2008년8월5일, 수정완료일: 2008년10월30일

다. 디지털 논리에서 상수 곱셈 연산은 상수가 2의 지수승에 해당하면 시프트 연산을 이용하여 대체될 수 있다. MAC 연산기에서 곱셈기 대신에 시프터를 이용하여 곱셈연산을 수행한다면 곱셈기를 사용하는 MAC 연산기 보다 전력소비나 회로크기, 효율면에서 월등히 높은 성능을 얻을 수 있다.

한편 오늘날의 DSP 알고리즘들은 더욱 뛰어난 성능의 DSP 연산을 위하여 점점 더 그 복잡도가 높아져 가고 있다. 이렇게 알고리즘의 복잡도가 높아져 감에 따라 DSP 디자인에 결함이 발생할 확률도 높아지고 있다. 이러한 디자인 결함은 시스템 고장의 주요한 원인이 되고 있다. 원칙적으로 디자인 결함은 제거 되어질 수 있고 그렇게 되어야 하지만 실제로는 많은 시스템에서 디자인 결함의 완벽한 제거는 달성하기 어려운 일이다. 더욱이 실행 빈도가 잦고 복잡도 높은 DSP 알고리즘은 속도 및 전력 소모의 효율을 높이기 위해 특정 어플리케이션 동작을 위한 ASIC으로도 구현되어 사용되는 경우가 자주 있다. 이런 ASIC으로 구현된 알고리즘은 일단 칩이 만들어 진 후에는 수정하기가 불가능하다. 그렇기 때문에 시스템의 설계 단계에서부터 에러에 내성을 갖는 디자인을 필요로 하게 된다. 하지만 일반적인 에러에 내성을 갖는 디자인의 경우 에러를 검출하거나 복구하기 위하여 추가적인 모듈을 필요로 하게 된다. 이러한 추가 모듈의 사용은 앞서 설명한 저전력 설계나 회로크기 축소라는 측면에서 부정적인 결과를 가져오게 된다. 특히나 MAC 연산기와 같은 모듈에서 Fault Tolerance 설계를 위하여 추가 곱셈기를 사용하면 전력소비나 회로의 크기는 현저히 증가하게 될 것이다.

본 논문에서는 저전력 동작을 위하여 곱셈기를 대신하여 배럴 시프터를 이용해 곱셈연산을 수행하는 MAC 연산기를 사용하였다. 배럴 시프터를 이용하여 구현된 MAC 연산기는 전력소비와 회로크기 측면에서 월등히 높은 성능을 갖으나 곱셈기를 사용하는 MAC 연산기에 비해 정확도가 약간 떨어진다는 단점이 있다. 본 논문에서는 배럴 시프터를 사용한 저전력 MAC 연산기를 이용하여 Fault Tolerance 설계를 갖춘 저전력 MAC 연산기 구조를 제안한다.

II. 본 론

1. 저전력 MAC 연산기

그림 1은 본 논문에서 사용된 32-tap FIR 필터를 수

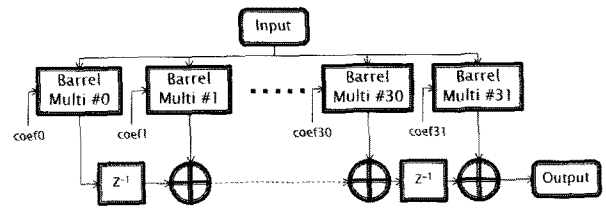


그림 1. 배럴시프터 기반 32탭 FIR 필터
Fig. 1. Barrel Shifter-based 32 tap FIR Filter.

행하는 MAC 연산기의 블록도이다. 위 MAC 연산기를 수식으로 표현하면 다음과 같다.

$$Y(n) = X(n)a(0) + X(n-1)a(1) + \dots + X(n-31)a(31) \quad (1)$$

위의 식을 구현하기 위해서는 곱셈기와 딜레이 모듈, 덧셈기가 필요하다. 일반적으로 딜레이 모듈은 플립플롭을 사용하여 구현할 수 있고, 덧셈기는 사용하는 알고리즘에 따라 그 성능이 다양하다. MAC 연산기를 구성하는 모듈 중에서 MAC 연산기의 성능에 가장 많은 영향을 주는 모듈은 곱셈기이다. 그래서 본 논문에서는 MAC 연산기의 성능 향상을 위하여 곱셈기를 대신해 시프터를 이용하여 곱셈연산을 수행하는 알고리즘을 사용하였다. 그림 1의 Barrel-Multi 블록은 시프터를 사용하여 곱셈 연산을 수행하는 블록이다. FIR 연산을 위하여 Barrel-Multi 블록은 수행하는 FIR 함수의 윈도우 크기만큼 존재하며 각각의 Barrel-Multi 블록 연산 결과는 덧셈기의 입력이 되어 딜레이 모듈과 덧셈기로 이루어진 누산기에 의해 최종 결과물을 생성한다. 곱셈기를 대신하여 곱셈연산을 수행하는 Barrel-Multi 블록의 내부 블록도는 다음 그림 2와 같다.

곱셈연산을 수행하기 위한 시프터는 배럴 시프터를 사용하였다. 배럴 시프터는 한 주기에 원하는 비트만큼

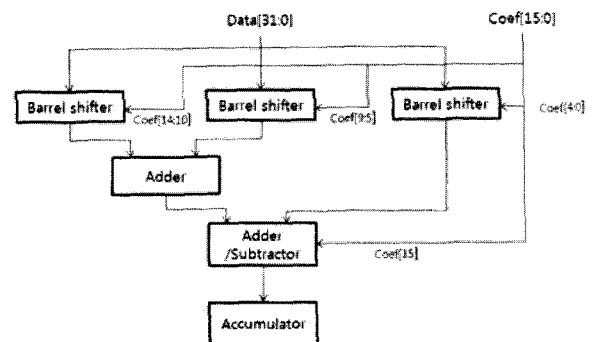


그림 2. 다중 배럴시프터와 덧셈기를 이용한 곱셈
Fig. 2. Multiplication with multiple barrel shifters and adders.

시프트가 가능한 회로이다. 따라서 배럴 시프터로 구성된 Barrel-Multi 블록은 한 클럭 사이클 안에 곱셈연산의 결과를 출력해 낼 수 있다.

그림 2를 보면 Barrel-Multi 블록이 어떻게 Barrel Shifter를 이용하여 곱셈연산을 수행하는지를 알 수 있다. 먼저 입력된 데이터는 각각 세 개의 배럴 시프터로 입력되어 같이 입력되는 Coefficient 데이터에 따라 시프트 연산을 수행한다. 배럴 시프터를 통하여 시프트된 결과는 덧셈과 뺄셈을 거쳐 최종결과물을 출력한다. 출력된 데이터는 앞서 설명한대로 누산기로 입력되어 진다. 이때 Data와 함께 입력되어 지는 Coefficient 데이터는 FIR 필터에서 수행하려는 함수에 따른 계수값을 시프터에서 사용할 수 있도록 설계 시에 미리 계산하여 둔 데이터이다. Coef 입력에는 각각의 배럴 시프터에서 수행하여야 할 시프트 횟수에 대한 데이터 뿐만 아니라 시프트 연산된 결과를 더하거나 빼기의 수행에 대한 제어 정보도 담고 있다. 시프트 연산 후에 수행되는 덧셈 및 뺄셈 연산도 한정된 시프터를 이용하여 최적의 결과를 얻어낼 수 있도록 설계 단계에서 미리 계산되어진다.

시프터를 이용하여 곱셈연산을 대신하고자 할 때 사용되는 시프터의 수가 많을수록 연산 결과의 정확도를 높일 수 있다. 하지만 본 논문에서는 시프터 세 개를 사용하였을 경우가 전력소비와 회로크기 등의 성능과 연산 결과의 오차 등을 고려하였을 때 가장 현실적으로 바람직하다는 실험결과를 토대로 구현되었다^[1].

2. Fault Tolerance를 위한 제안하는 구조

본 논문에서 제안하고자하는 Fault Tolerance 구조는 그림 3과 같다. 그림 3에서 Barrel-Multi 블록은 배

럴 시프터를 이용하여 곱셈 연산을 수행하는 블록이다. Fault Tolerance 기능을 위해 Barrel-Multi 블록이 총 세 개가 사용된다. 본래 Fault Tolerance 기능을 사용하지 않는다면 하나만 사용되면 되는 블록이지만 에러를 검출하고, 에러 발생 시에도 올바른 결과 값을 출력하기 위해 2개의 Barrel-Multi 블록이 추가되었다. 하지만 Barrel-Multi 블록이 배럴 시프터를 사용하여 저전력으로 곱셈 연산을 수행하는 블록이기 때문에 Fault Tolerance를 위해 2개의 추가 Barrel-Multi 블록이 사용되었어도, Fault Tolerance 기능이 없는 곱셈기를 사용하여 구현된 MAC 보다 전력소비나 회로크기 면에서 향상된 성능을 보이고 있다. 또한 에러 발생이 없는 때는 Barrel-Multi #1과 #2만이 사용되며, 두 Barrel-Multi 블록의 연산 결과가 서로 다르게 나타나는 에러가 검출되면 Barrel-Multi #3이 동작하여 #1과 #2로부터 얻어진 결과와 비교되어 세 개의 결과 중 두 개의 결과가 동일할 경우에만 그 결과를 출력한다. 만약 세 개의 결과가 모두 다를 경우에는 복구할 수 없는 에러로 판단하여 error 신호를 출력하게 된다. Fault Tolerance는 에러에 대해 어느 정도 유효성을 발휘할 수도 있다. 만약 검출된 에러에 대해 일정 수준의 오차를 허용한다면 그 오차 범위는 위 블록도의 Comparator #2와 #3을 통하여 조절할 수 있다.

3. Fault Tolerance를 위한 다양성

본 논문에서 제안하는 알고리즘에서 사용되는 Barrel-Multi 모듈은 모두 똑같은 블록이다. Fault Tolerance가 정상적으로 동작함을 보장해주기 위해서는 같은 결과를 얻는 연산일 지라도 다양한 방법을 통하여 그 결과를 얻고 서로 비교되어야 한다. 따라서 본 논문

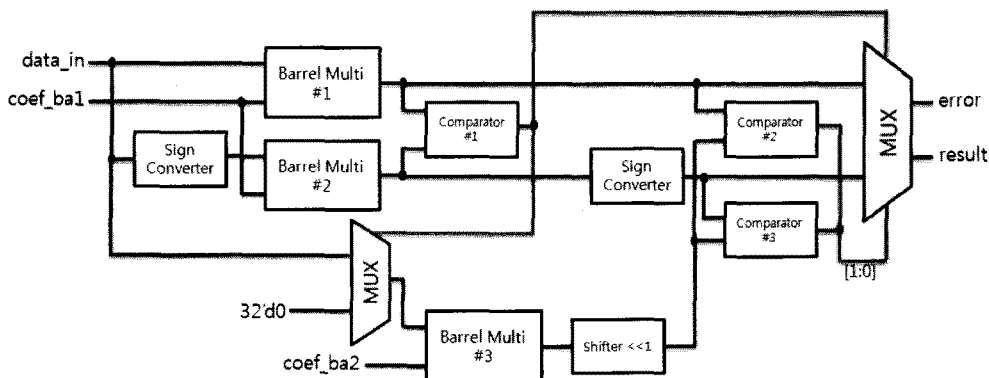


그림 3. Fault Tolerance를 위한 블록도
Fig. 3. Block Diagram for fault Tolerance.

에서는 동일한 연산 블록을 사용할 지라도 그 입력 데이터를 다양하게 만들어 동일한 결과를 얻으며 서로 다른 동작을 할 수 있도록 하고 있다. 기본적으로 Barrel-Multi 블록은 곱셈 연산을 수행하는 블록이기에 곱셈의 수학적 정의에 따라 다양한 형태로 식을 변형시킬 수 있다.

$Y_1 = X * Coef$	(2)
$Y_2 = -X * Coef$	(3)
$Y_3 = X * Coef/2$	(4)
$Y_1 = -Y_2 = 2Y_3$	(5)

그림 3의 Barrel-Multi 모듈에서 수행하는 연산들은 위와 같은 식으로 표현할 수 있다. Barrel-Multi #1 블록에서는 식(2)에 해당하는 연산을 수행한다. Barrel-Multi #2 블록에서는 식(3)에 해당하는 연산을 수행한다. 식(3)에 해당하는 연산을 수행하기 위하여 Barrel-Multi #2에 입력되는 데이터는 1의 보수법을 이용하여 입력되는 데이터의 부호를 변경시킨다. 수학적으로는 단순히 부호만 바뀐 것이지만 디지털 논리에서 보면 많은 비트들이 반전되기 때문에 부호가 바뀌기 전의 값과 비교하여 상당히 다른 데이터가 된다. 수학적으로는 부호만 다른 값을 쓰는 것이기 때문에 Barrel-Multi #1과 #2에서 출력은 서로 크기는 같고 부호가 다른 결과가 출력되게 된다. 그래서 사실 Comparator #1에서는 두 결과값의 덧셈연산을 수행하여 0이외의 결과가 나오는 것을 판단하여 에러를 검출하게 된다. 한편 에러가 검출될 경우에만 동작하는 Barrel-Multi #3은 식(4)의 연산을 사용한다. 앞서 사용된 두 식과는 달리 Coefficient 데이터에 변화를 주어 사용되어 진다. 디지털 논리에서 나누기 연산은 시프터를 이용하여 쉽게 구현될 수 있다. 본 논문에서 사용하는 Barrel-Multi 블록은 기본적으로 배럴 시프터를 사용하기 때문에 나누기 연산은 더욱 쉽게 사용할 수 있다. Coefficient 데이터는 Barrel-Multi 블록의 설계단계에서 미리 계산된 값을 사용하기 때문에 Barrel-Multi #3에서 나누기 연산의 결과를 얻기 위해서는 사전에 계산된 Coefficient 데이터를 수정하여 사용하게 된다. Barrel-Multi #3의 결과는 식(5)에 따라 Barrel-Multi #1과 #2의 결과와 비교되어 진다. 그래서 Barrel-Multi #1과 #2를 통하여 구하여진 결과와 비교하기 위해

Barrel-Multi #3의 결과에는 상수 2가 곱하여 진다. 앞서 나누기 연산을 수행한 것과 마찬가지로 이 역시 시프터를 이용하여 쉽게 구현할 수 있다. 이와 같은 방법으로 동일한 연산블록을 사용하고자도 입력값을 다르게 적용하는 방법을 통하여 Fault Tolerance를 위한 연산의 다양성을 확보할 수 있었다.

III. 실험

본 논문에서 제안된 에러 내성을 갖는 저전력 MAC 연산기의 성능평가를 위해서 그림 3의 구조를 구현하고 이에 대한 비교 대상으로 그림 3의 구조에서 Barrel-Multi 블록을 대신해 일반 곱셈기를 이용하는 모델을 구현하여 비교하여 보았다. 비교 대상에 사용된 일반 곱셈기는 시놉시스 디자인웨어 (Synopsys DesignWare) 라이브러리에서 제공하는 DW_mult를 사용하였다. 비교에 사용된 모든 MAC 구조는 32-tap FIR 필터이며, hamming 윈도우 함수에 따른 계수를 적용하여 설계된 구조에 대해 실험이 이루어졌다. 모든 구조들은 매그나칩(MagnaChip) 0.18 μ m 공정의 라이브러리를 이용하여 시놉시스 디자인 컴파일러(Synopsys Design Compiler)를 통하여 합성하였고, 시놉시스 파워 컴파일러(Synopsys Power Compiler)를 이용하여 동적 전력소비, 누설전력소비, 회로크기, 타이밍 등의 항목을 측정하였다.

그림 4는 동적전력소비 측정에 따른 결과이다.

동적전력소비는 세 가지 경우로 나누어 측정해 보았다. 먼저 배럴 시프터를 사용한 모델과 일반 곱셈기를 사용한 모델의 결과를 측정하였다. 한편으로 본 논문에서 제안한 Fault Tolerance 구조가 에러 없이 동작할 경우에는 Barrel-Multi #3 블록이 동작하지 않으므로 이에 따른 효과를 관찰하기 위해 배럴 시프터를 사용하는 모델을 대상으로 Barrel-Multi #1 블록에 고의적인 에러를 주입하여 동작시켰다. 결과로써 배럴 시프터를

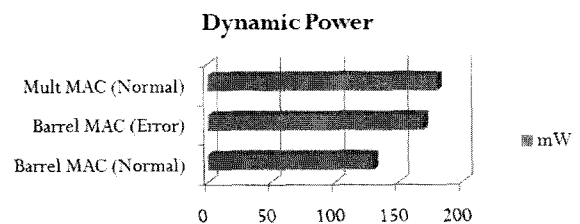


그림 4. 동적 전력 소비

Fig. 4. Dynamic Power Consumption.

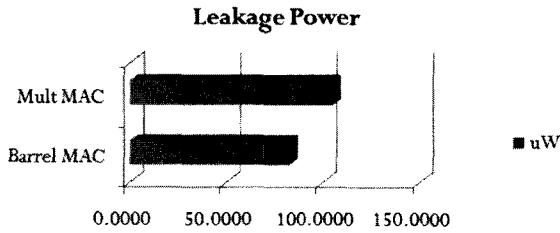


그림 5. 누설 전력 소비
Fig. 5. Leakage Power Consumption.

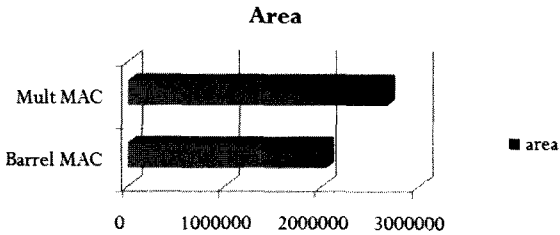


그림 6. 회로크기
Fig. 6. Area.

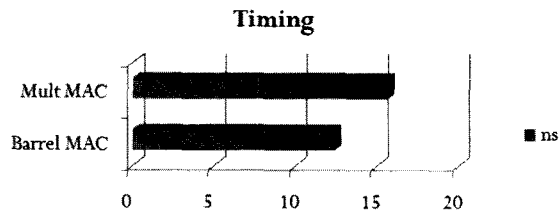


그림 7. 타이밍
Fig. 7. Timing.

사용한 모델이 일반 곱셈기를 사용한 모델에 비하여 약 28% 향상된 성능으로 동작함을 확인할 수 있었다. 또한 고의적인 에러가 주입된 모델의 경우 동적전력소비가 약 23.3% 증가된 결과를 보여주었다. 이 결과를 보면 배럴 시프터를 사용한 모델이 일반 곱셈기를 사용한 모델에 비해 동적전력소비 측면에서 더 효율적임을 확인할 수 있었다. 또한 추가된 모듈을 능동적으로 사용하도록 한 본 논문의 설계도 전력관리 측면에서 효과가 있음을 확인할 수 있었다.

누설전력 소모량의 경우도 배럴 시프터를 사용한 모델이 일반 곱셈기를 사용한 모델에 비해 약 21.3% 향상된 성능을 보이는 것을 확인할 수 있었다.

회로크기 또한 배럴 시프터를 사용한 모델이 일반 곱셈기를 사용한 모델에 비해 약 23.4% 더 적은 면적을 차지함을 확인할 수 있었다.

타이밍은 모든 구조들의 동작 주파수를 100MHz로 설정하고 최대 지연 경로를 바탕으로 측정된 결과를 비교

하였다. 그 결과 배럴 시프터를 사용한 모델은 12.42ns 일반 곱셈기를 사용한 모델은 15.7ns로 배럴 시프터를 사용한 모델이 약 20.8% 향상된 성능을 보이는 것을 확인할 수 있었다.

IV. 결 론

일반적인 경우 Fault Tolerance 구성을 하고자 할 경우 추가 모듈의 사용은 불가피하다. 이러한 추가 모듈의 사용은 회로크기를 증가시키고 그에 따른 전력소비도 증가시키는 단점을 갖게 된다. 오늘날 많은 MAC 연산기가 휴대용 전자 장비에서 사용되어지는 것을 고려할 때 MAC 연산기에는 더 작은 회로크기와 더 적은 전력소모가 요구되어지고 있다. 이에 따라 본 논문에서는 배럴 시프터를 이용하여 곱셈 연산을 수행하도록 함으로서 기존 MAC 연산기에 비해 저전력으로 동작하는 MAC 연산기를 설계하고, 이를 바탕으로 Fault Tolerance 구성을 함으로서 그에 따른 오버헤드를 최소한으로 하고자 노력하였다. 또한 추가 모듈을 제한적으로 사용하도록 설계함으로서 동적전력 관리가 능동적으로 이루어질 수 있도록 하였다. 이와 같은 노력으로 본 논문에서는 Fault Tolerance 구성에 따른 오버헤드를 최소한으로 갖는 에러 내성을 갖는 저전력 MAC 연산기를 설계할 수 있었다.

참 고 문 헌

- [1] Young-Geun Lee, Joo-Yul Park, and Ki-Seok Chung, "Design of Low Power MAC Operator with Dual Precision Mode", 13th IEEE conference on RTCSA, pp. 309-315, DAEGU, KOREA, August 2007.
- [2] D. K. Pradhan, "Fault-Tolerant Computing Theory and Techniques", Prentice Hall, pp. 417-466, 1986.
- [3] Hassan B. Diab and Albert Y. Zomaya, "Dependable Computing Systems Paradigms, Performance Issues, And Applications", WILEY-Interscience, pp. 213-241, 2005.
- [4] Barry W. Johnson, "Design and Analysis of Fault Tolerant Digital Systems", Addison Wesley, pp. 169-262, 1989.
- [5] M. Potkonajak, M. Srivastava, and A. Chandrakasan, "Multiple Constant Multiplications: Efficient and Versatile Framework and Algorithms for Exploring Common

Subexpression Elimination”, IEEE Trans. on CAD of Integrated Circuits and Systems, Vol. 15, No. 2, pp. 151-165, February 1996.

[6] Chien-Chung Chua, Bah-Hwee Gwee and Joseph S. Chang, “A low-voltage micropower asynchronous multiplier for a multiplierless FIR filter”, In Proc. of the 2003 Int'l Symposium on Circuits and Systems, pp. 381-384, May 2003.

[7] A. G. Dempster and M. D. Macleod, “Constant integer multiplication using minimum adders”,

Proc. Inst. Elec. Eng. Circuits and systems, vol. 141, no. 5, pp. 407-413, October 1994.

[8] J. C. Laprie, J. Arlat, C. Beounes, and K. Kanoun, “Definition and analysis of hardware-and-software fault-tolerant architectures”, IEEE Computer, 23: 39-51, 1990.

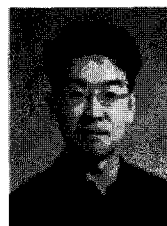
[9] R. D. Schlichting and F. B. Schneider, “Fail-stop processors: an approach to designing fault-tolerant computing systems”, ACM Transactions on Computing Systems, 1: 222-238, 1983.

저 자 소 개



정 한 샘(정회원)
 2008년 광운대학교 컴퓨터공학부
 학사 졸업.
 2008년~현재 한양대학교
 전자컴퓨터통신공학과
 석사과정.

<주관심분야 : 멀티코어, 임베디드시스템, SoC, Fault-Tolerance>



정 기 석(정회원)
 1989년 서울대학교 컴퓨터공학과
 학사 졸업.
 1997년 University of Illinois at
 Urbana-Champaign
 전산학 박사 졸업.
 1998년 University of Illinois at
 Urbana-Champaign
 강의 전담 교수.

2000년 Synopsys, Inc. Sr. R&D Engineer.
 2001년 Intel Corp. Staff Engineer.
 2001년 홍익대학교 컴퓨터공학과 조교수.
 2004년~현재 한양대학교 정보통신대학
 미디어통신공학과 조교수.

<주관심분야 : 임베디드시스템, SoC, 저전력 시스템 설계, VLSI/CAD>



구 성 관(정회원)
 2005년 Texas A&M University,
 College station BS
 (computer engineering)
 졸업.
 2008년 한양대학교 전자컴퓨터
 공학과 석사 졸업.

2008년~현재 삼성태크윈 영상정보사업팀
 이미징사업부.

<주관심분야 : VLSI, Embedded, DSP>