

논문 2008-45SD-2-12

정교한 클럭 게이팅을 이용한 저전력 재구성 가능한 DSP 설계

(Design of a Low Power Reconfigurable DSP with Fine-Grained Clock Gating)

정 찬 민*, 이 영 근**, 정 기 석*

(Chan-Min Jung, Young-Geun Lee, and Ki-Seok Chung)

요 약

최근 많은 임베디드 시스템에서 통신이나 멀티미디어의 다양한 표준을 지원해야 하는 요구가 끊이지 않고 있다. 그러나 현실적으로 임베디드 시스템에서 요구하는 만큼의 표준이나 프로토콜을 위한 별개의 가속 IP들을 갖는 것은 불가능할 뿐만 아니라 상당히 힘든 작업이다. 그러므로 다양한 표준을 지원할 수 있는 가속 IP를 개발하는 것은 위와 같은 현재의 임베디드 시스템에서 요구하는 트렌드에 있어 중요하다 할 수 있다. 다양한 기능을 수행하는 하드웨어는 일반적으로 실행 환경이나 시스템 설정에 따라 다양한 기능들을 지원하기 위하여 동적으로 즉, 실행시간에 재구성 가능한 DSP를 사용하고 있다. 그러나 하나의 IP가 다양한 기능을 수행시키기 위해서는 필수불가결적으로 추가적인 면적을 차지하거나 추가적인 전력소모가 따른다. 그러므로 본 논문에서는 동적으로 재구성 가능한 하드웨어의 파워 소모를 줄이기 위해 정교한 클럭 게이팅을 사용하였고, 또한 동적으로 재구성 가능한 하드웨어의 면적을 줄이기 위해 배럴 시프터(barrel shifter)를 이용한 곱셈기를 사용하여 메모리의 계수(Coefficient) 부분을 압축을 통해 메모리의 면적을 줄였다. 실행시간에 재구성 가능한 IP는 유사하지만 다른 기능들을 효과적으로 수행하기 때문에 이런 다기능 재구성 가능한 DSP IP의 전력소모를 성능에 영향 없이 줄이는 것은 상당히 난해한 일이다. 본 논문에서 제안한 정교한 클럭 게이팅은 동적으로 재구성 가능한 시스템에 아주 효율적으로 적용되었고 효과적인 결과를 도출하였다. 실험 결과는 본 논문에서 제시한 기법을 사용했을 시 사용하지 않았을 경우보다 최대 24%정도의 파워 절감 효과를 얻을 수 있었다. 또한 면적을 줄이기 위해서 기존의 일반적인 곱셈기를 사용하는 대신에 배럴 시프터(barrel shifter)를 이용한 곱셈기를 설계해 적용하였다. 기존 곱셈기를 제안한 곱셈기로 바꾸면 설계한 재구성 가능한 DSP의 구조상 많은 면적을 줄이는 것이 가능했다. 기존 곱셈기에 비해 제안된 곱셈기는 면적은 42%가 줄었으며, 전체적인 재구성 가능한 DSP의 면적에서 14% 감소한 결과를 도출하였다. 그러므로 본 논문은 재구성 가능한 특성을 갖는 IP의 단점인 파워 소모와 추가적인 면적을 효과적으로 보완한 면에 있어 큰 의의가 있다고 할 수 있다.

Abstract

Recently, many digital signal processing (DSP) applications such as H.264, CDMA and MP3 are predominant tasks for modern high-performance portable devices. These applications are generally computation-intensive, and therefore, require quite complicated accelerator units to improve performance. Designing such specialized, yet fixed DSP accelerators takes lots of effort. Therefore, DSPs with multiple accelerators often have a very poor time-to-market and an unacceptable area overhead. To avoid such long time-to-market and high-area overhead, dynamically reconfigurable DSP architectures have attracted a lot of attention lately. Dynamically reconfigurable DSPs typically employ a multi-functional DSP accelerator which executes similar, yet different multiple kinds of computations for DSP applications. With this type of dynamically reconfigurable DSP accelerators, the time to market reduces significantly. However, integrating multiple functionalities into a single IP often results in excessive control and area overhead. Therefore, delay and power consumption often turn out to be quite excessive. In this thesis, to reduce power consumption of dynamically reconfigurable IPs, we propose a novel fine-grained clock gating scheme, and to reduce size of dynamically reconfigurable IPs, we propose a compact multiplier-less multiplication unit where shifters and adders carry out constant multiplications.

Keywords: 재구성 가능한 DSP, Variable Precision, Clock gating, Low power, Multiplier, low area

* 정회원, ** 학생회원, 한양대학교 전자컴퓨터통신공학과

(Department of Electronics, Computer & Communications, Hanyang University)

* "본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-C1090-0801-0045)

접수일자: 2007년11월23일, 수정완료일: 2008년1월28일

I. 서 론

최근 많은 H.264, CDMA, MP3 등과 같은 디지털 신호 처리 응용은 높은 성능을 갖는 휴대 가능한 장비에 필수적인 기본 요건이 되고 있다. 이들 응용들은 일반적으로 연산 집중(computation-intensive)적인 특성을 갖는다. 그러므로 이들 응용은 이러한 연산을 빠르게 수행시키기 위해 상당히 복잡한 연산 가속기를 요구하고 있는 실정이다. 이러한 응용에 특화된 DSP 가속기를 설계하는 것은 상당한 노력이 들기 때문에 다양한 가속기를 갖는 DSP는 좋지 않은 적기 출시(time-to-market)를 갖고 또한 허용할 수 없는 범위의 추가적인 면적이 소요된다.

위와 같이 다양한 응용에 적합한 DSP를 만들기 위해 많은 가속기를 사용하는 시스템에서의 적기 출시(time-to-market)와 추가적인 면적이 상당하다는 점을 극복하기 위해 최근에 동적으로 즉, 실행시간에 재구성 가능한 DSP구조가 많은 관심을 받고 있다. 이 논문에서 제시하는 동적으로 재구성 가능한 DSP는 다양한 기능을 수행할 수 있는 가속기를 사용한다.

이를 통하여 재구성 가능한 DSP가 갖는 단점인 파워 소모와 면적을 감소시키는 결과를 도출하였다. 동적으로 재구성 가능한 DSP의 파워 소모를 줄이기 위해 본 논문에서는 새로운 정교한 클럭 게이팅 방법을 제시한다. 다양한 기능을 수행하는 동적으로 재구성 가능한 하드웨어에서 파워 소모를 줄이는 것은 하나의 저전력 기법이 다양한 기능을 수행하는 하드웨어에 적용해야 하고, 파워 소모를 줄이는 기법은 기본적으로 효율적이고 추가적인 비용이 적어야 하기에 상당히 연구할 가치가 있다 하겠다.

동적으로 재구성 가능한 하드웨어가 갖는 또 다른 단점인 면적 측면에서는 본 논문에서 기존 곱셈기를 배럴 시프터(barrel shifter)를 이용한 곱셈기를 사용함으로써 개선점을 찾았다. 이 방법을 적용하였을 때 본 논문에서 제시한 재구성 가능한 DSP 구조상 가장 면적이 큰 모듈인 메모리의 면적이 감소되는 효과를 도출했다. 메모리는 기본적으로 연산에 필요한 계수(coefficient)를 지니고 있다. 만약 연산에 필요한 계수(coefficient)가 많아진다면 메모리 사이즈 역시 커질 수밖에 없다. 하지만 배럴 시프터(barrel shifter)를 사용하게 되면 계수(coefficient) 압축효과를 볼 수 있어 메모리 면적을 감소시킬 수 있다.

위에서 언급한 크게 2가지 방법, (첫째로 동적으로

재구성 가능한 DSP의 파워 소모를 줄이기 위해 정교한 클럭 게이팅을 적용시킨 방법, 둘째로 동적으로 재구성 가능한 DSP의 면적을 줄이기 위해 곱셈기를 수정한 기법)으로 재구성 가능한 DSP가 갖는 단점의 개선방향을 제시하는 논문을 서술한다.

본 논문에서 제시하는 연구 내용은 다음과 같다. 먼저 재구성 가능한 하드웨어가 갖는 문제점을 파악했다. 파워소모가 많고, 면적을 크게 갖는다는 것이 바로 그것이다. 먼저 파워소모를 줄이기 위한 방법으로 클럭 게이팅을 이용한 다양한 정확도를 지원하는 기법을 재구성 가능한 하드웨어에 적용하였고, 면적을 줄이기 위해 일반 곱셈기를 배럴 시프터(barrel shifter)를 갖는 구조로 바꾸어 연산에 필요한 계수(Coefficients)를 압축하는 기법을 적용하였다. 결과적으로 메모리 모듈의 면적을 줄임으로서 전체적인 면적을 줄이려고 노력하였다.

본 논문의 구성은 다음과 같다. II장에서는 본 논문에서 제안하는 재구성 가능한 DSP 구조를 설명한다. III장에서는 본 논문에서 제안하는 재구성 가능한 DSP가 수행 가능한 알고리즘을 설명한다. IV장에서는 재구성 가능한 하드웨어가 갖는 문제점을 해결하기 위해 본 논문에서 선택한 최적화 방안을 설명한다. 최적화 방안은 크게 2가지로 나뉜다. 면적 부분과 전력 소모 부분이 그것이다. V장에서는 실제로 제안한 재구성 가능한 DSP에 최적화 기법을 적용하여 파워적인 면과 면적적인 면을 실험하여 결과를 도출 하였다. 마지막으로 VI 장 결론으로 마무리 짓는 순서로 본 논문은 구성되어 있다.

II. 재구성 가능한 DSP 구조

재구성 가능한 DSP 는 하나의 연산 구조가 둘 이상의 알고리즘을 수행시키는 것을 특징으로 한다. 본 논문에서는 FFT, 복소수 FIR, 실수 FIR 연산을 수행시키는 재구성 가능한 DSP 구조를 구현하고 재구성이 가능하다는 특징을 갖기에 생기는 단점을 극복하는 방법을 제시한다.

아래 그림 1은 재구성 가능한 DSP를 최적화 기법 없이 구현한 구조이다.

이 기본 구조는 4개의 연산 유닛(PU)을 포함하고 있다. 여러 개의 연산 유닛은 FFT, 복소수 FIR, 실수 FIR 연산 알고리즘을 수행 하는 구조를 갖고 있다. 기본 구조는 MAC연산에 바탕을 두고 구성되었다.

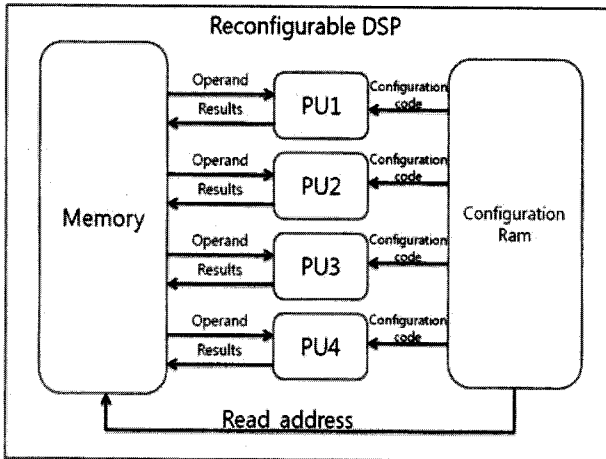


그림 1. 재구성 가능한 DSP 구조
Fig. 1. Reconfigurable DSP Architecture.

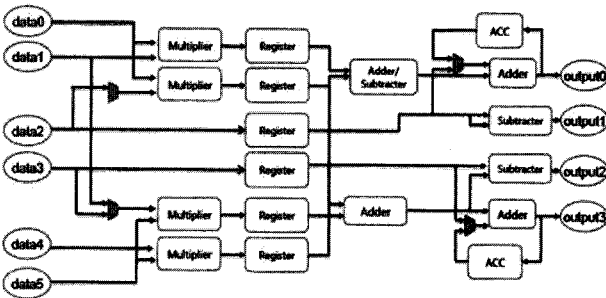


그림 2. 연산 유닛(Processing Unit)
Fig. 2. Processing Unit.

메모리는 크게 2가지 종류의 데이터를 저장하고 있다. 연산에 필요한 오퍼랜드를 저장하는 부분과 결과를 저장하는 부분으로 구성되어 있고, 오퍼랜드는 크게 3가지 종류가 있다. (1)외부에서 입력한 데이터, (2)연산에 필요한 계수 (coefficient), (3) 연산 중간 값을 저장하고 있다가 필요시에 오퍼랜드로 사용할 수 있는 쓰기 버퍼(write buffer)이다. 메모리는 연산에 필요한 오퍼랜드들을 연산 유닛에 제공하는 역할을 한다.

Configuration Ram은 각 알고리즘에 대한 마이크로 프로그램(microprogram)을 저장하고 있다. 설정 코드(Configuration code)는 연산 유닛이 특정 알고리즘을 위해 어떤 경로를 통과하며, 어떠한 연산을 수행시킬 것인지 결정해주는 마이크로코드(Microcode)이다. Read address는 메모리가 연산 유닛에게 어떠한 오퍼랜드를 제공해야 하는지를 알려주는 명령어이다.

그림 2는 연산 유닛(PU)을 보인다. 연산 유닛은 FFT와 복소수 FIR, 실수 FIR 3개의 알고리즘을 수행 할 수 있다. 6개의 register와 2개의 accumulator 역할을 하는 레지스터 2개, 1개의 adder/subtractor, 3개의 덧셈기, 2개의 뺄셈기로 구성되어 있다. 연산 유닛의 입력은 크

게 2가지이다. 첫째로 알고리즘 수행을 위한 오퍼랜드, 두 번째로 Configuration RAM으로부터 설정 코드(Configuration code)이다.

설정 코드(Configuration code)의 역할은 연산 유닛이 어떤 동작을 하는지를 결정해 준다. 즉, MUX의 선택 신호(select signal)이나 adder/subtractor을 매 클럭마다 설정 해준다. 매 클럭마다 새로운 설정 코드(Configuration code)가 Configuration RAM으로부터 적재되기 때문에 실행 시간에 연산 유닛의 동작이 변할 수 있다. 즉, 이 연산 유닛은 동적으로 재구성 가능하다.

III. 재구성 가능한 DSP의 수행 알고리즘

앞서 언급했듯이 PU는 FFT나 실수- 그리고 복소수 FIR같은 Multiply-accumulate나 Multiply-add-based 알고리즘을 처리 한다.

FFT의 기본적인 operation은 Butterfly 연산이다. Radix-2-Butterfly연산의 2개의 입력이 $x = x_r + jx_i$, $x = y_r + jy_i$ 일 때 다음과 같은 $x' = x'_r + jx'_i$, $x' = y'_r + jy'_i$ 두 개의 복소수 결과가 도출 된다. FFT 연산에 필요한 twiddle-factor가 $w = w_r + jw_i$ 라고 할 때 radix-2-Butterfly연산은 다음 4개의 식으로 나타낼 수 있다.

$$x'_r = x_r + (y_r w_r - y_i w_i) \tag{1}$$

$$x'_i = x_i + (y_r w_i - y_i w_r) \tag{2}$$

$$y'_r = x_r - (y_r w_r - y_i w_i) \tag{3}$$

$$y'_i = x_i - (y_r w_i - y_i w_r) \tag{4}$$

식에서 보듯이 butterfly연산은 Multiply-add-based 알고리즘이다. FFT 연산은 4개의 정수 곱셈 연산이 필요하다. PU에는 4개의 곱셈기가 있기 때문에 처음에는 두 cycle에 한번의 radix-2-butterfly 연산이 가능하다. 다음 radix-2-butterfly 연산은 중간에 레지스터를 이용해 파이프라인처럼 동작하여 한 사이클에 radix-2-butterfly 연산이 가능하다.

복소수 FIR 알고리즘을 위해서는 다음과 같은 연산을 수행해야 한다.

$$g(n) = \sum_{m=-\infty}^{\infty} x(m) \cdot w(n-m) \tag{5}$$

위의 식에서 보듯 FIR연산 또한 Multiply-add-based 알고리즘이다. 입력 값이나 계수가 복소수 형태로 주어진다면 복소수의 Multiply-add 연산이 되어 다음 식에서와 같이 FFT연산과 유사하게 된다.

$$Re(g) = x_r + (y_r w_r - y_i w_i) \tag{6}$$

$$Im(g) = x_i + (y_r w_i - y_i w_r) \tag{7}$$

그러므로 연산유닛는 복소수 FIR연산을 radix-2-butterfly 연산과 유사하게 수행 가능하다.

실수 FIR연산은 만약 4 tap이라면 다음과 같다.

$$g(n) = x(n-3)w_3 + x(n-2)w_2 + x(n-1)w_1 + x(n)w_0 \tag{8}$$

위의 식의 기본은 Multiply-accumulate 이다. 그러므로 연산유닛에서는 곱셈기와 누적기를 이용하여 실수 FIR연산을 수행한다.

IV. 재구성 가능한 DSP의 최적화

1. 면적 최적화 (Area optimization)

우리가 구현한 재구성 가능한 DSP의 연산 유닛 구조는 그림 3과 같다.

그림 3의 구조를 보면 하나의 연산 유닛에는 4개의 곱셈기가 들어가 있고, 6개의 입력을 받아들여 연산 후 4개의 결과를 출력으로 보내주는 역할을 한다. 특이할 점은 입력 data0과 data5는 각각 총 4개의 곱셈기 중 2개의 곱셈기의 입력 오퍼랜드로 들어가게 된다. 우리가 제안한 재구성 가능한 DSP의 연산 유닛의 Data0와 Data5의 입력은 오로지 연산에 필요한 계수(coefficient) 뿐이다.

우리의 연구 방향은 곱셈기의 오퍼랜드로 들어가는 계수 (coefficient)의 사이즈를 줄여 곱셈기의 사이즈도 줄이며, 연산에 필요한 계수(coefficient)를 저장하고 있

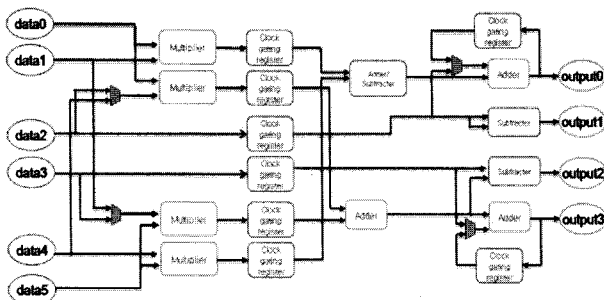


그림 3. 연산 유닛 블록 다이어그램
Fig. 3. Processing Unit Block Diagram.

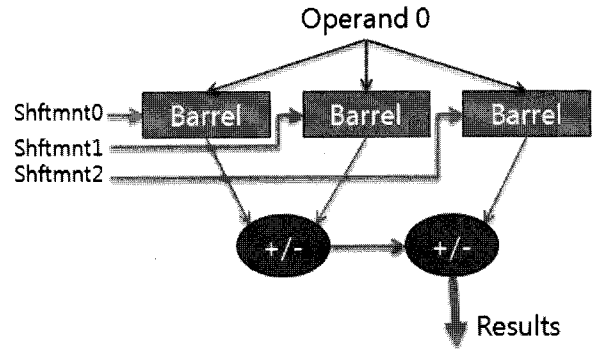


그림 4. 제안하는 곱셈기 구조
Fig. 4. Proposed Multiplier architecture.

고 가장 면적이 큰 모듈중 하나인 메모리의 면적을 줄이는 방향으로 연구를 진행하였다.

기존 곱셈기는 입력으로 오퍼랜드 2개와 곱셈 결과 출력부분 하나로 구성되어 있다. 하지만 본 논문의 기본 아이디어는 실제로 곱셈기는 다른 연산기보다 많은 면적을 차지하기에 면적이 작은 효율적인 곱셈기를 만들어보자는 데서 시작되었다. 그래서 생각한 기본 아이디어는 배럴 시프터(barrel shifter)를 이용하는 것이다. 시프터는 기본 특성상 한 비트 이동 시킬 때 마다 ‘곱하기 2’의 효과와 ‘나누기 2’의 효과를 볼 수 있다. 그러므로 곱셈기를 설계 할 때 이런 시프터의 특성을 이용하였다. 그림 4는 본 논문에서 제시하는 곱셈기의 구조를 보여주고 있다.

제안하는 곱셈기는 3개의 배럴 시프터(barrel shifter)와 2개의 덧셈/뺄셈 연산기 그리고 각 배럴 시프터(barrel shifter)의 값을 몇 비트를 시프트 시킬지를 결정하는 3개의 shift amount 입력, 그리고 하나의 오퍼랜드를 입력으로 갖고, 하나의 연산 결과 출력을 갖는 구조로 되어 있다.

이 제안하는 곱셈기의 동작 아이디어는 다음과 같다. 기존 곱셈기에서는 입력으로 오퍼랜드를 2개 받았으나, 제안한 곱셈기는 하나의 오퍼랜드를 받고 나머지 하나의 오퍼랜드는 2의 지수승의 덧셈과 뺄셈의 조합으로 표현하여 연산을 수행한다는 것이다. 식은 아래와 같다.

$$Operand1 = 2^{shftmnt0} + / - 2^{shftmnt1} + / - 2^{shftmnt2} \tag{9}$$

오퍼랜드 하나는 기존 곱셈기와 마찬가지로 입력으로 들어오고 나머지 하나의 오퍼랜드는 위의 식처럼 2의 지수 승으로 변환되어 shift amount값으로 shftmnt0, shftmnt1, shftmnt2의 입력으로 들어오게 되는 것이 본 구조의 특징이다. 예를 들어 숫자 13을 표현하기 위

해서 $2^3 + 2^2 + 2^1$ 로 표현될 수 있다. 이처럼 2의 지수 승의 덧셈과 뺄셈 조합으로 수를 표현이 가능하다. 그러므로 shiftmnt 값은 차례로 3, 2, 1을 갖게 된다. 전체적으로 곱셈 연산은 다음과 같은 식으로 이루어진다.

$$\text{Result} = \text{Operand}0 \times (2^{\text{shftmnt}0} + / - 2^{\text{shftmnt}1} + / - 2^{\text{shftmnt}2}) \tag{10}$$

시프터와 배럴 시프터(barrel shifter)는 같은 시프터라는 면에서는 같지만 차이가 존재 한다. 일반 시프터는 한 비트를 시프트하는데 한 사이클이 걸린다. 즉 n 비트를 시프트하는데 n 사이클이 걸리게 된다. 하지만 배럴 시프터 (barrel shifter)는 n 비트를 시프트하거나 한 비트 만을 시프트하던 항상 1 사이클이 걸린다. 그러므로 본 논문에서 제안한 곱셈기 구조는 1 사이클에 곱셈 결과를 얻을 수 있다.

본 논문에서 제안한 재구성 가능한 DSP의 메모리 모듈은 크게 2가지로 구성된다. 첫 번째로 연산에 필요한 오퍼랜드와 연산에 필요한 계수 (coefficient)를 지니는 레지스터 파일(register file) 부분과 연산 유닛에 오퍼랜드를 제공하는 컨트롤 로직으로 구성되어 있다. 전체 메모리 모듈의 20%만이 컨트롤 로직이고 나머지는 레지스터 파일(register file) 부분이다. 그러므로 메모리 면적을 줄이기 위해서는 컨트롤 부분을 제외한 레지스터 파일(register file)의 면적을 줄이는 것이 가장 효과적이라 할 수 있다.

일반 곱셈기를 배럴 시프터 (barrel shifter)를 이용한 곱셈기로 교체하면 결과적으로 계수(coefficient)가 압축되는 효과가 있어 전체적인 메모리 면적을 줄일 수 있다. 예를 들어 숫자 127을 표현하기 위해 일반적으로 이진수 7b1111111 이렇게 총 7비트로 표현이 가능하다. 하지만 제안한 곱셈기를 이용할 경우 127은 128-1 과 같기 때문에 27 - 20 과 같다고 할 수 있으므로 shift amount 값은 7과 0이 되므로 이진수로 111 과 0으로 표현 가능하다. 하지만 위의 구조에서 보듯이 덧셈/뺄셈기의 동작을 위해 한 비트가 더 필요하게 된다. 덧셈을 위해서는 이진수 '1'로 표현하고 뺄셈을 위해서는 이진수 '0'으로 표현해 전체적인 127의 수는 5'b111_0_0으

로 5비트로 표현이 가능하다. 그림 5는 새로 대체된 계수 (coefficient)의 포맷을 나타낸다.

shftmnt0, shftmnt1, shftmnt2 값은 각각 n 비트로 최대 2n-1까지 표현이 가능하다. 하지만 덧셈과 뺄셈의 조합으로 많은 수를 만들어 낼 수 있다. 예를 들어 -13 같은 경우는 1 +2 - 16으로 20 + 22- 24 로 구현 할 수 있으며 12'b000_1_001_0_010 으로 표현이 가능하다. 지금은 작은 수를 표현 하다 보니 압축률이 없지만 큰 수를 표현할 때는 압축률이 비교 할 수 없을 정도로 커지게 된다. 그리고 FFT나 FIR 같은 연산의 포인트 수가 증가함에 계수 (coefficient) 수가 비례하게 증가하기 때문에 포인트 수가 많아질수록 압축 효과는 더욱 필요하다고 할 수 있다. 물론 배럴 시프터 (barrel shifter)를 2개만을 사용해도 어느 정도 수의 표현이 가능하지만 배럴 시프터 (barrel shifter)의 2개와 3개의 차이는 수를 표현할 수 있는 범위에서 큰 차이가 있기 때문에 어느 정도 면적을 희생해서라도 수의 표현 범위를 높이고 오차를 줄이고자 3개의 배럴 시프터 (barrel shifter)를 사용하였다. 이런 계수 (coefficient)의 압축 효과를 이용하여 메모리의 전체 면적에서 약 14%의 절감 효과를 얻었다.

이러한 배럴 시프터 (barrel shifter)를 사용 할 경우 메모리의 절감 효과 뿐 만 아니라 곱셈기 자체의 면적 절감 효과도 존재 한다. 물론 메모리 절감 효과처럼 절감 절대 수치가 크지는 않지만 일반 곱셈기 보다 제안된 곱셈기는 기존에 비해 43%절감 효과를 보였다. 그리고 곱셈기가 있는 연산 유닛 연산 유닛에서 기존 곱셈기를 사용 했을 때 보다 17%정도 파워 절감 효과를 보였다.

2. 소비 전력 최적화 (Power Optimization)

CMOS 회로에서 동적 파워 소비 (dynamic power dissipation)은 전력 소모에 주된 원인이다. 그리고 이러한 동적으로 소모되는 파워는 데이터를 다른 값으로 변환 할 시에 발생한다. 클럭 게이팅은 클럭 신호를 끊음으로 해서 사용되지 않는 기능적인 유닛 (Functional Unit)의 값 변화를 안 시킴으로 해서 파워를 줄이는 데 사용되고 있다.

만약 미리 연산 전에 연산 정확도를 알고 있다면 클럭 게이팅을 통해 사용되지 않은 비트의 부호 비트 (sign-bit) 확장 같은 연산은 수행하지 않고 넘어 갈 수 있어 필요 없는 값을 레지스터에 저장하거나, 또 이미 메모리에 저장되어 있는 동일한 값을 반복적으로 재 저

shftmnt0	Add/Sub	shftmnt1	Add/Sub	shftmnt2
N-bit	1-bit	N-bit	1-bit	N-bit

그림 5. 새로 대체된 계수 (coefficient) 포맷
Fig. 5. Proposed Coefficient Format.

장하는 것으로 인해 소모되는 전력을 줄일 수 있다. 이 논문에서는 본 논문에서는 새로운 값을 래치(latch) 시킬 필요가 없는 레지스터의 값을 유지함으로써 정교한(Fine-grained) 클럭 게이팅 방식을 사용하여 파워 소모를 줄이는 기법을 사용하였다.

다양한 정확도를 제공한다는 것은 실행 중에 연산에 사용되고 있는 정확도를 바꿀 수 있는 것을 말한다. 결과를 올바르게 표현하기 위해 사용되는 표현 비트를 미리 알아야하기에 다양한 정확도를 제공하는 것은 쉽지 않은 일이다. 일반적으로 N 비트와 M 비트를 곱하는 연산이 있다면 결과는 M+N 비트가 될 것이다. MAC(Multiply and Accumulate) 연산은 기본적으로 곱셈 연산을 포함하고 있기 때문에 최악의 경우에는 오퍼랜드의 2배의 비트 폭을 가진 결과가 나올 수 있다.

그림 6은 가장 이상적인 클럭 게이팅 레지스터를 보여주고 있다. 레지스터들은 클럭이 1에서 0으로 변할 때나 0에서 1로 변할 때 새로운 값을 레지스터로 래치(latch)된다. 일반적으로 레지스터는 매 사이클마다 새로운 값을 읽어 들인다. 만약 많은 레지스터들이 같은 값을 오랜 시간동안 유지 하고 있다면 계속 같은 값을 반복적으로 읽어 들이게 된다. 만일 수백 클럭 사이클 동안 레지스터가 같은 값을 유지하고 있다면 이렇게 동일 값을 계속 읽어 들이는 것은 많은 에너지를 낭비하는 것이다. 특히 DSP 응용에서는 많은 변화되는 값이 점차 증가되는 경향이 있기 때문에 한꺼번에 레지스터의 값들이 크게 변하는 일들은 흔히 발생하지 않는다. 특히 레지스터의 가장 왼쪽 비트들은 결과의 부호 비트(sign bit)가 변화되지 않는 이상 변화될 일이 거의 없다. 그러므로 본 논문에서는 정교한 클럭 게이팅을 사용하여 산술연산에 있어서 다양한 정확도를 제공하는

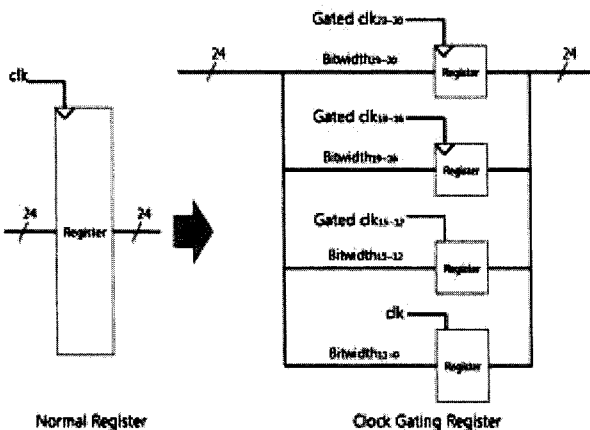


그림 6. 클럭 게이팅
Fig. 6. Clock Gating.

기능을 추가 하였다.

본 논문에서 구현된 설계에서는 각 레지스터는 24비트의 크기를 갖고 24비트 레지스터는 4개의 파트로 나누었다. 또한 본 논문에서 제시하는 DSP에서 제공하는 다양한 정확도는 24비트, 20비트, 16비트, 12비트 이렇게 4가지 연산 레벨을 제공한다. 레지스터의 하위 12비트는 항상 클럭 신호에 의해 새로운 값을 받아들이고 나머지는 컨트롤 신호에 의해 래치(latch) 될 수 있고 안 될 수 있다. 클럭 게이팅 로직은 개념적으로 아주 간단하다. 그러나 위 구조는 개념적으로 심각한 문제를 야기시킬 가능성이 있다. 만약 클럭 게이팅 로직으로부터 지연이 너무 길다면 컨트롤 신호는 클럭 신호를 정확한 시간에 막을 수 없게 된다. 특히 정교한 클럭 게이팅을 사용하는 경우에 있어서 위와 같은 컨트롤 신호가 클럭 신호를 정확한 시간에 막을 수 없을 경우에 심각한 문제를 발생시켜 시스템이 오작동을 일으킬 가능성이 있다. 이 문제를 피하기 위해 본 논문에서는 그림 7과 같은 설계를 제안한다.

그림 7은 기존의 방식을 수정한 게이팅 레지스터를

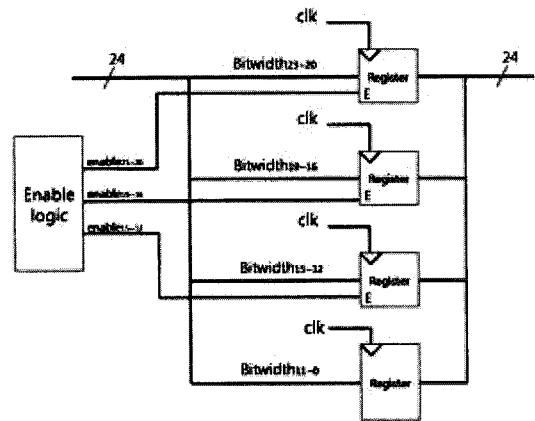


그림 7. 수정된 게이팅 레지스터
Fig. 7. Modified Clock Gating Register.

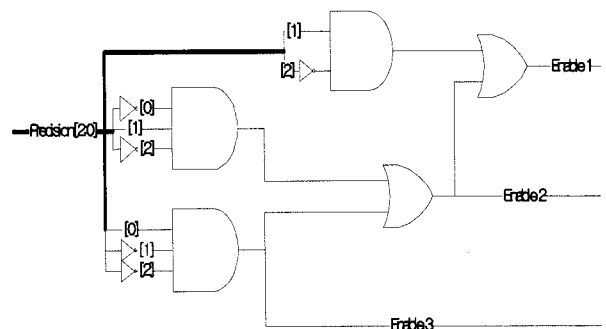


그림 8. Enable Logic.
Fig. 8. Enable Logic.

보여주고 있다. 수정된 게이팅 레지스터는 직접적으로 클럭 신호를 컨트롤 하지 않기 때문에 타이밍 문제는 발생 시키지 않는 구조이다. 이 구조는 클럭 신호를 컨트롤 하는 대신 Enable Logic을 두어 여기서 나오는 신호로 레지스터를 컨트롤 한다. 그러므로 타이밍 문제는 발생시키지 않는다.

Enable Logic의 구조는 그림 8과 같다.

4가지의 정확도를 제공하기 위해 총 3비트의 입력 신호와 3비트의 출력 컨트롤 신호가 필요하다. 이 Enable Logic은 24비트의 레지스터 비트중 상위 12비트를 게이팅 시키기 위한 신호를 출력한다. Enable1 출력 신호는 12번째 비트~15번째 비트를 컨트롤하기 위한 신호이고, Enable2는 16번째 비트~19번째 비트를 컨트롤하기 위한 신호로 사용된다. 마지막으로 Enable3는 20번째 비트~23번째 비트를 게이팅하기 위해 사용된다. 정확도 비트가 001이면 모든 24비트는 동작 한다. 반면에 정확도 비트가 100이면 오로지 가장 낮은 12비트만이 동작한다.

본 논문에서는 제안한 연산 유닛은 24비트, 20비트, 16비트, 12비트 총 4가지의 정확도를 제공한다. 4가지의 정확도는 enable logic에 의해 컨트롤 된다. enable logic에서 enable 신호를 받은 레지스터는 정해진 비트만큼의 값을 래치(latch) 할 수 있다. 그림 9는 클럭 게이팅을 적용시킨 레지스터를 사용한 재구성 가능한 DSP의 연산 유닛(PU) 을 보여준다.

그림 9의 연산 유닛은 FFT와 복소수 FIR, 실수 FIR 을 수행 할 수 있다. 6개의 레지스터와 2개의 축적기(accumulator) 역할을 하는 레지스터 2개, 1개의 덧셈/뺄셈기, 3개의 덧셈기, 2개의 뺄셈기로 구성되어 있다. 각각의 다른 연산을 위해 MUX를 통한 데이터 경로나 덧셈/뺄셈기에 관한 컨트롤은 외부 설정 RAM (Configurable RAM)에 의해 제어된다. 이 유닛은 평균

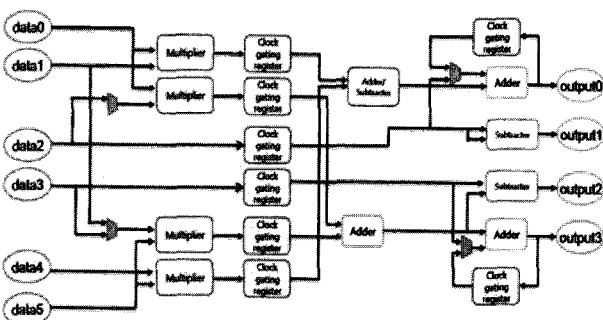


그림 9. 제안한 연산 유닛
Fig. 9. Proposed Processing Unit.

한 사이클에 2개의 버터플라이 연산을 수행시킨다. FIR연산도 한 사이클에 2개의 출력을 연산한다. 연산 유닛을 병렬 구성이 가능하기에 연산 알고리즘에 따라 연산 유닛 수를 조절할 수 있는 구조를 가진다. 연산 유닛에 있는 모든 레지스터는 모두 클럭 게이팅 구조를 두어 다양한 정확도를 지원하며, 전력 소모 또한 줄일 수 있다.

V. 실험 결과

1. 실험 환경

본 논문은 재구성 가능한 하드웨어의 단점인 파워 소모와 면적적인 문제를 해결하기 위해 크게 두 가지를 제안 하였다. 첫 번째로 재구성 가능한 DSP의 파워 소모를 줄이기 위해 정교한 클럭 게이팅을 이용해 다양한 정확도를 제공함으로써 연산의 효율과 전력을 감소 시켰고, 두 번째로 재구성 가능한 DSP의 면적을 줄이기 위해 기존 곱셈기 구조를 배럴 시프터(barrel shifter)를 이용한 곱셈기 구조로 바꿈으로써 계수 (coefficient) 압축을 통해 메모리 모듈의 면적을 줄여 본 논문에서 제안한 재구성 가능한 DSP의 면적을 줄이는 방법을 제시 하였다.

이러한 아이디어의 유효성을 판단하기 위해 본 논문에서 실험한 환경은 표 1과 같다.

실험을 위해 사용한 툴은 Synopsys의 Design Analyzer인데 이는 재구성 가능한 DSP의 RTL 코드를 합성하고 면적을 측정하고, 시뮬레이션 파일을 생성하여 파워를 측정하기 위해 사용되었다.

표 1. 실험 환경
Table 1. Experimental Environment.

Tool Name	Synopsys Design Analyzer
Library	Magnachip
Technology	0.18um
Frequency	100Mhz

2. 면적 최적화 실험 결과

본 절에서는 재구성 가능한 DSP에 본 논문에서 제시 하는 아이디어 즉, 기존 곱셈기를 배럴 시프터(barrel shifter)로 대체함으로써 생기는 계수 (coefficient) 압축 효과로 인한 메모리 모듈의 면적 감소시키는 아이디어를 앞 절에서 소개 했다. 그림 그러한 아이디어를 실제

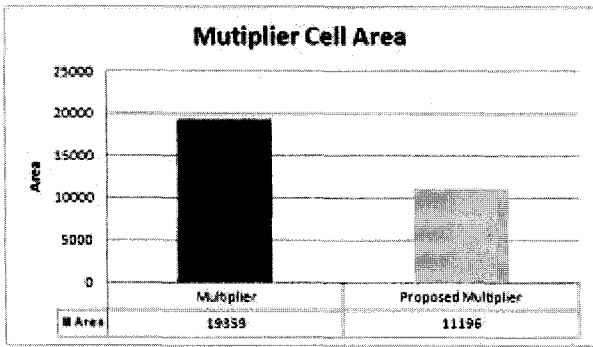


그림 10. 곱셈기와 제안한 곱셈기의 면적 비교
Fig. 10. Area Comparison with Proposed Multiplier and Normal Multiplier.

재구성 가능한 DSP에 적용시켜 결과를 본 절에서 소개 하겠다.

위의 결과는 일반 곱셈기와 본 논문에서 제안한 곱셈 기의 면적 비교이다. 위의 결과는 Design Compiler를 사용하였고 라이브러리는 메그나칩스 0.18um 공정을 사용하였다. 위의 그래프의 Y축은 모듈의 면적을 나타 내고 X축은 기존 곱셈기와 제안한 곱셈기를 나타낸다. 위의 결과에서 보듯이 기존 곱셈기보다 배럴 시프터 (barrel shifter)를 사용했을 때의 결과가 약 43%정도 절감된 것을 확인 할 수 있다.

다음 결과는 재구성 가능한 DSP에서 실제 연산을 수 행하고 곱셈기 4개를 갖고 있는 연산 유닛의 면적의 절 감 효과 결과를 보인다.

수정된 곱셈기를 갖는 연산 유닛은 기존 연산 유닛보 다 약 4%의 절감 효과를 보인다. 절감효과가 미미한 이 유에는 연산 유닛에 존재하는 모듈이 곱셈기 뿐 만 아 니라 레지스터, 덧셈기, 멀티플렉서 등 다른 모듈들이 혼재해 있기 때문에 곱셈기 자체적으로 면적 절감 효과 가 크게 영향을 주지 못하기 때문이다. 하지만 면적 절

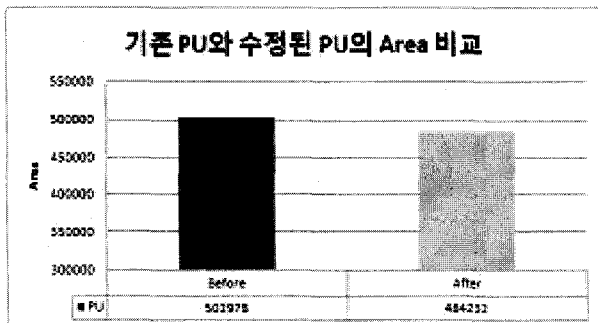


그림 11. 기존 연산 유닛과 수정된 연산 유닛의 면적 비 교
Fig. 11. Area Comparison with Proposed Processing Unit and Reference Processing Unit.

감 효과는 연산유닛(PU)가 아닌 메모리 모듈에서 더 크게 나타나고 본 논문도 연산유닛의 면적을 줄이려는 노력보다 메모리를 줄이려는 쪽으로 연구 방향을 맞추 었다.

그림 12는 메모리 모듈의 면적 절감 효과를 보여준 다. 아래의 결과를 보면 기존의 메모리에 비해 면적 절 감효과를 적용시킨 메모리는 약 16%정도 절감된 것을 확인 할 수 있다. 우선적으로 계수 (coefficient)의 압축 으로 인한 절감과 그에 따른 레지스터 비트 폭의 감소 등 부수적인 효과로 인해 16%의 절감을 보인 것이다.

그림 13은 재구성 가능한 DSP의 하위 모듈들의 전체 적인 면적의 비교를 나타내는 그래프이다. 앞서 설명한 바와 같이 재구성 가능한 DSP에서 가장 면적을 차지하 는 큰 모듈은 메모리, 연산 유닛, Configuration RAM 순이다. 본 논문에서는 메모리 면적을 줄이는 노력을 하였으며 메모리는 16% 절감 효과를 보였고 연산 유닛 은 4% 절감 효과를 보여 전체적으로 14%의 절감효과 를 얻어 낼 수 있었다.

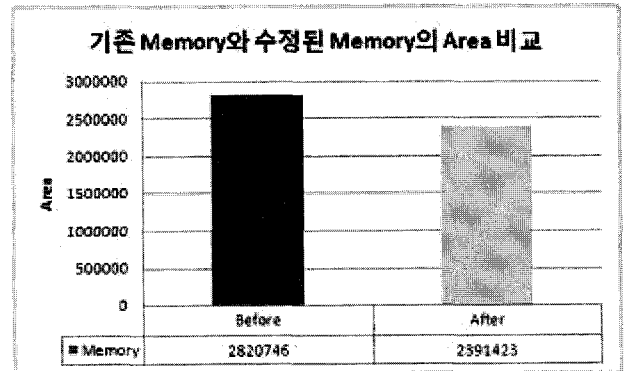


그림 12. 기존 메모리와 수정된 메모리의 면적 비교
Fig. 12. Area Comparison with Modified Memory Block and Reference Memory Block.

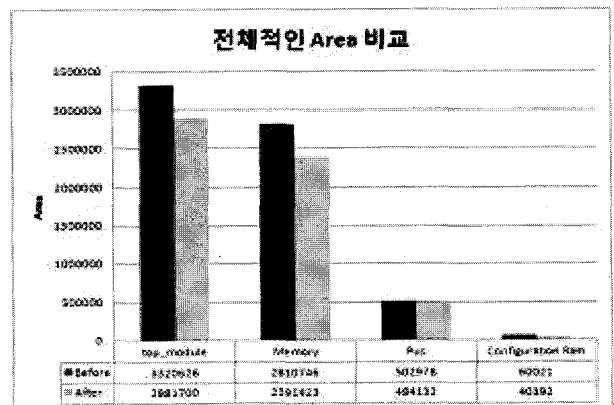


그림 13. 전체적인 면적 비교
Fig. 13. Total Area Comparison.

2. 소비 전력 최적화 실험 결과

앞 절에서는 재구성 가능한 DSP의 면적 최적화에 대한 결과를 다루었다. 본 절에서는 재구성 가능한 DSP의 소비 전력 최적화의 결과에 대해 다룬다. 재구성 가능한 하드웨어의 큰 단점 중 하나인 큰 전력 소모를 줄이기 위해 본 논문에서는 정교한 클럭 게이팅을 이용한 다양한 정확도를 제공함으로써 연산의 효율과 필요하지 않은 값을 레지스터에 래치(latch) 시키지 않음으로 해서 동적인 파워를 줄이는 기법을 적용하였다.

앞 절에서 설명한 기존 곱셈기를 배럴 시프터(barrel shifter)로 바꾸게 되면 면적의 절감적인 측면도 있지만, 부차적으로 연산 유닛에서의 파워의 절감 효과도 존재한다. 그림 14는 기존 곱셈기를 사용했을 때와 제안한 곱셈기를 사용했을 때와의 연산 유닛에서의 파워 소비와 연산 유닛안의 곱셈기에서의 파워 소비를 나타내는 그래프이다. 아래의 결과에서 볼 수 있듯이 곱셈기는 연산 유닛의 소비 파워에서 약 67%정도의 파워를 소비하는 모듈이다. 그러므로 연산 유닛에서 곱셈기의 파워 소모를 줄이는 것은 파워를 줄이기 위한 가장 효율적인 접근 방법이라 할 수 있다.

결과는 그림 14와 같다. 제안한 곱셈기를 사용했을 경우 파워는 절대 수치는 작지만 연산 유닛에서 약 17%정도 절감효과를 얻을 수 있다.

다음은 전력을 줄이기 위한 노력으로 클럭 게이팅을 이용한 다양한 정확도를 제공함으로써 생기는 파워 절감효과를 알아보도록 한다. 본 논문에서 제시한 다양한 정확도는 총 4가지이다. 24비트, 20비트, 16비트, 12비트 이렇게 4가지 정확도를 제시한다. 물론 직관적으로도

12비트의 정확도를 사용할 때 가장 적은 파워를 소모할 것으로 예상되며, 24비트를 모두 사용할 경우는 가장 많은 파워를 소모할 것으로 예상할 수 있다.

그림 15는 기존 곱셈기를 사용했을 시 연산 별 정확도 별 파워 소모를 나타낸 그래프이다. 예상했던 대로 24비트 사용 시 두 연산 모두 가장 많은 파워를 소모했다. 24비트와 12비트의 정확도를 사용했을 경우 복소수 FIR일 경우는 최대 23%정도의 절감효과를 가질 수 있었고, 정수 FIR일 경우에는 최대 20%의 절감효과를 보였다.

그림 16은 제안한 곱셈기를 사용했을 때의 정확도 별 파워 소모를 나타낸 그래프이다. 앞서 설명한 그래프와 마찬가지로 비슷한 결과를 도출 하였다. 24비트 정확도일 때 가장 많은 파워를 소모하였고, 12비트일 때

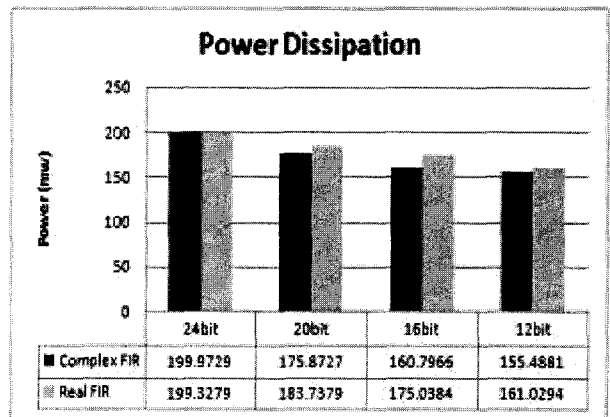


그림 15. 일반 곱셈기를 사용했을 때의 정확도 별 파워 소모
Fig. 15. Power Dissipation with Normal Multiplier.

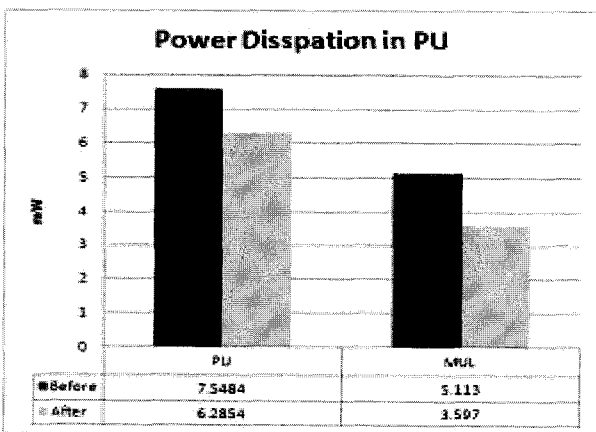


그림 14. 수정된 곱셈기와 기존 곱셈기와의 연산 유닛에서의 파워 소비
Fig. 14. Power Dissipation Comparison with Modified Multiplier and Normal Multiplier.

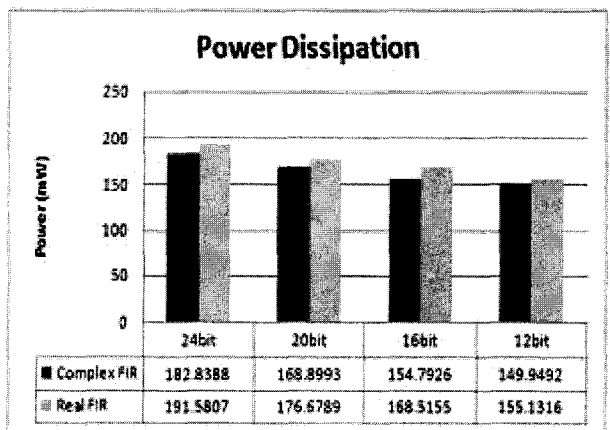


그림 16. 제안한 곱셈기를 사용했을 때의 정확도 별 파워 소모
Fig. 16. Power Dissipation with Modified Multiplier.

가장 적은 파워를 소모하였다. 복소수 FIR일 경우에는 최대 19%의 파워의 절감효과를 보였고, 실수 FIR일 경우에는 실수 FIR일 경우에도 최대 19%의 파워 절감효과를 보였다.

위의 결과로서 우리가 어떤 응용을 수행하기 전에 응용에 대한 정확도를 알고 있다면 정교한 클럭 게이팅을 이용하여 원하는 정확도만큼의 값을 래치 (latch) 시킴으로써 동적인 파워를 줄일 수 있다.

IV. 결 론

최근 많은 임베디드 시스템에서 통신이나 멀티미디어의 다양한 표준을 지원해야 하는 요구가 끊이지 않고 있다. 이러한 현실적인 요구에 따라 다양한 기능을 수행하는 하드웨어가 관심을 받고 있다. 이러한 다기능 하드웨어는 일반적으로 실행 환경이나 시스템 설정에 따라 다양한 기능들을 지원하기 위하여 동적으로 즉, 실행 시간에 재구성 가능한 DSP를 사용하고 있다. 그러나 하나의 IP가 다양한 기능을 수행시키기 위해서는 필수불가결적으로 추가적인 면적을 차지하거나 추가적인 전력소모가 따른다. 그러므로 본 논문에서는 동적으로 재구성 가능한 하드웨어의 파워 소모를 줄이기 위해 정교한 클럭 게이팅을 사용하여 다양한 정확도를 제공해 연산의 효율성과 파워 소모를 줄였다. 또한 동적으로 재구성 가능한 하드웨어의 면적을 줄이기 위해 배럴 시프터 (barrel shifter)를 이용한 Multiplier를 사용하여 메모리의 계수 (coefficient) 부분을 압축을 통해 메모리의 면적을 줄였다.

실행시간에 재구성 가능한 하드웨어는 유사하지만 다른 기능들을 효과적으로 수행하기 때문에 이런 재구성 가능한 DSP 하드웨어의 전력소모를 줄이는 것은 상당히 난해한 일이다. 본 논문에서 제안한 정교한 클럭 게이팅은 동적으로 재구성 가능한 시스템에 아주 효율적으로 적용되었고 효과적인 결과를 도출하였다. 실험 결과는 우리의 아이디어를 사용했을 시 사용하지 않았을 경우보다 최대 24%정도의 파워 절감 효과를 얻을 수 있었다.

또한 면적을 줄이기 위해서 기존의 일반적인 곱셈기를 사용하는 대신에 배럴 시프터 (barrel shifter)를 이용한 곱셈기를 설계해 적용하였다. 기존 곱셈기를 제안한 곱셈기로 바꾸면 설계한 재구성 가능한 DSP의 구조상 많은 면적 측면에서 이득을 얻을 수 있다. 기존 곱셈기에 비해 제안된 곱셈기는 면적은 42%가 줄었으며, 전

체적인 재구성 가능한 DSP의 면적적인 측면에서는 14% 감소한 결과를 도출하였다.

그러므로 본 논문은 재구성 가능한 특성을 갖는 하드웨어의 가장 큰 단점인 파워 소모와 추가적인 면적적인 면을 효과적으로 보완한 면에 있어 큰 의의가 있다고 할 수 있다.

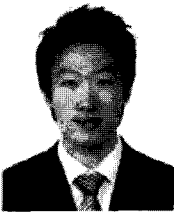
하지만 재구성 가능한 하드웨어는 여러 기능을 수행하기 위해 컨트롤 부분의 오버헤드가 존재해 성능에 영향을 주곤 한다. 그러므로 앞으로 진행될 연구는 이러한 컨트롤 부분의 최적화를 통해 면적 측면이나 추가적인 파워 소모 등을 해결하기 위한 노력이 필요하다.

참 고 문 헌

- [1] Christopher Pretty and J. Geoffrey Chase "Reconfigurable DSP's for Efficient MPEG-4 Video and Audio Decoding", The First IEEE International Workshop on Electronic Design, Test and Applications (DELTA '02) p. 63
- [2] Jung-Mok Lee, Young-Geun Lee, Chan-Min Jung, Byung-Gyu Ahn, Jong-Wha Chong and Ki-Seok Chung "Design of a low power reconfigurable DSP architecture", International Ubiquitous Convergence Conference, Dec. 2006.
- [3] Keith Reeves, Ken Sienski "Reconfigurable Hardware Accelerator for embedded DSP", SPIE Proc. SPIE Vol. 2914, p. 332-340 1996.
- [4] Ya-Lan Tsao, Ming Hsuan Tan, Jun-Xian Teng and Shy-Jye Jou "Parameterized and Low Power DSP Core for Embedded Systems", ISCAS '03. Proceedings of the 2003 International Symposium on Publication Volume:4, On page(s):V-265-V-268 vol.4
- [5] Li-Hsun Chen, Oskal T.-C. Chen and Ruey-liang Ma "A High-Efficiency Reconfigurable Digital Signal Processor for Multimedia", Circuits and Systems, 2003. ISCAS '03.
- [6] Hans Jacobson, Pradip Bose, Zhigang Hu, Alper Buyuktosunoglu, Victor Zyuban, Rick Eickemeyer, Lee Eisen, John Griswell, Doug Logan, Balaram Sinharoy and Joel Tandler "Stretching the Limits of Clock-Gating Efficiency in Server-Class Processors", High-Performance Computer Architecture, 2005.
- [7] Kimiyoshi Usami and Naoaki Ohkubo, "A Design Approach for Fine-grained Run-Time Power gating using Locally Extracted Sleep Signals", ICCD, 2006.
- [8] Rajeevan Amirtharajah and Anantha P.

Chandrakasan "A Micropower Programmable DSP Using Approximate Signal Processing Based on Distributed Arithmetic", IEEE Journal of Solid State Circuits, Vol. 39, No. 2, February 2004.

저 자 소 개



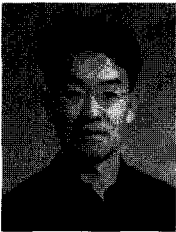
정 찬 민(정회원)
2006년 한양대학교 정보통신학부
미디어통신공학사 졸업
2008년 한양대학교 전자컴퓨터
통신공학 석사 졸업.

<주관심분야 : 재구성 가능한 DSP 설계, microprocessor 구조, 저전력 시스템, 디지털 회로 설계>



이 영 근(학생회원)
2006년 한양대학교 정보통신학부
미디어통신공학사 졸업.
2008년 한양대학교 전자컴퓨터
통신공학 석사 졸업.

<주관심분야 : 저전력 기법, 재구성 가능한 프로세서, fault tolerant system, leakage power minimization.>



정 기 석(정회원)
1989년 서울대학교 컴퓨터공학사
졸업.
1998년 Illinois university at
Urbana-Champaign
computer science
박사 졸업

1998년~2000년 Senior R&D Engineer at
Synopsis Inc

2000년~2001년 Staff Engineer at Intel Corp.

2001년~2004년 홍익대학교 컴퓨터공학과
조교수

2004년~현재 한양대학교 정보통신대학 조교수

<주관심분야: 저전력 임베디드 시스템 설계, SoC
설계 방법론, 재구성 가능한 DSP & 프로세서 설
계, SoC-platform 기반 검증, MPSoC를 위한 시
스템 소프트웨어>