

論文2001-38SD-9-8

2D Image Filter에 적합한 저전력 FIR Filter의 구현 (Low Power Architecture of FIR Filter for 2D Image Filter)

韓昌永*, 朴炯俊*, 金利燮*

(Chang Young Han, Hyoung Jun Park, and Lee Sup Kim)

요약

본 논문은 여러 이산 신호 처리(Digital Signal Processing)에서 많이 사용되는 FIR Filter의 전력 소모를 줄이는 새로운 방법을 제안한다. FIR Filter에서 소모되는 전력 중 곱셈기가 차지하는 비중이 매우 높다는 사실과 2D 영상에서 이웃한 픽셀 값의 공간 상관성이 높다는 성질을 이용하였다. 곱셈기의 입력인 영상 데이터를 상대적으로 상관성이 높은 상위 비트(MSBs)와 상관성이 낮은 하위 비트(LSBs)로 구분하고, 각각에 대해서 필터링을 수행하도록 하였다. 또한, 입력의 상위 비트와 필터 계수와의 곱셈 결과는 캐쉬(cache)에 저장하여 재사용함으로써 불필요한 상위 비트의 연산을 줄이도록 하였다. 이러한 방법을 SMT(Separated Multiplication Technique)라 부르기로 한다. FIR Filter를 사용함에 있어 제안된 SMT를 이용하였을 경우에 15%정도의 전력 이득 효과를 얻을 수 있었다.

Abstract

This paper proposes a new power reduction method for 2D FIR (Finite Impulse Response) filters. We exploited the spatial redundancy of image data in order to reduce power dissipation in multiplication of FIR filters. Since the higher bits of input pixels are hardly changed, the redundant multiplication of higher bits is avoided by separating multiplication into higher and lower parts. The calculated values of higher bits are stored in memory cells, cache such that they can be reused when a cache hit occurs. Therefore, we can reduce power in 2D FIR Filter modules about 15% by using the proposed separated multiplication Technique (SMT).

1. Introduction

오늘날 무선 이동 통신 시장의 급격한 성장으로 말미암아 작고 가벼우며 오래 쓸 수 있는 단말기의 요구가 무엇보다 높아졌다. 차세대 이동 통신은 지금의 음성 신호 위주에서 영상 신호 위주의 산업으로 재편될

것이 확실시 되므로 영상 데이터를 처리하기 위해 보다 많은 전력을 소모할 수 밖에 없다. 그러나, 배터리의 발전 속도는 매우 더디므로 단말기를 오래 사용하기 위해서는 저전력으로 칩을 설계하는 것이 필수적이다. 음성 및 영상 신호 처리에서 FIR Filter는 주요한 요소 중의 하나로서 최근에 주목받고 있는 DWT와 같은 영상 압축 알고리즘에서도 널리 사용된다. FIR Filter에서 소모되는 전력의 대부분은 곱셈기에 의한 것이므로 곱셈 연산에서의 전력을 줄이는 것이 무엇보다 중요하다.

지금까지도 저전력을 위한 FIR Filter 및 곱셈기의 구조에 대한 연구는 활발히 이루어지고 있다. 이러한 노력의 하나로 입력의 높은 상관성을 이용한 방법이 제시되었다^[1-3]. 필터 계수 또는 입력의 차이만을 곱하고 이전 계수들의 결과를 순차적으로 더하는 DCM/

* 正會員, 韓國科學技術院 電子電算學科
(Division of Electrical Engineering, Dept. of
Electrical Engineering and Computer Science,
KAIST)

接受日字:2000年6月13日, 수정완료일:2001年7月24日

DCIM(Differential Coefficient/Input Method)이 그것이다. 또한, 곱셈을 보다 작은 곱셈의 형태로 나누고 이의 부분 결과(partial products)를 ROM에 저장한 뒤에 최종 결과는 ROM의 값을 참조하여 조합하는 방법도 있다^[4]. 본 논문에서 제안하는 SMT는 이미지 영상의 이웃한 픽셀 값들은 그 차이가 그리 크지 않으므로 매번 비슷한 값을 연산할 필요가 없다는 점을 이용하였다. 즉, 입력을 상관성이 높은 부분과 낮은 부분으로 나누고, 상관성이 낮은 부분은 매번 연산을 수행하지만 상관성이 높은 부분은 가능한 적은 횟수의 연산을 하도록 하였다. 이의 결과를 버퍼에 저장하고 재사용함으로써 상위 비트의 연산을 억제하여 전력의 낭비를 줄였다.

본 논문의 제 2장에서는 데이터의 상관성을 이용한 다른 연구로서 DCM 알고리즘에 대해서 살펴보고, 제안하는 방법의 플랫폼으로서 DWT에 대해서 소개한다. 또한, SMT를 제안하는 배경에 대해서도 언급한다. 그리고, 제 3장에서는 제안하는 알고리즘과 그 구조의 장단점을 소개한다. 나머지 장에서는 모의 실험 결과를 정리하여 결론을 맺도록 한다.

II. Previous Works and Motivations

(1) Review of DCM/DCIM Algorithm

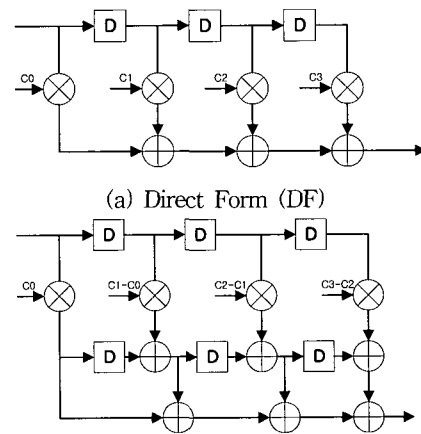
필터 계수가 C_k , 입력 시퀀스가 X_j , 출력 시퀀스가 Y_j 인 N -tap FIR Filter는 수식 (1)과 같이 표현된다.

$$Y_1 = \sum_{k=0}^{N-1} C_k X_{j-k} \quad (1)$$

$$\begin{aligned} Y_{j+1} &= C_0 X_{j+1} + C_1 X_j + \dots + C_{N-1} X_{j-N+2} \\ &= C_0 X_{j+1} + ((C_1 X_j - C_0 X_j) + C_0 X_j) + \dots \\ &\quad + (C_{N-1} X_{j-N+2} - C_{N-2} X_{j-N+2}) \\ &\quad + C_{N-2} X_{j-N+2} \\ &= C_0 X_{j+1} + (dC_1^1 X_j + C_0 X_j + \dots \\ &\quad + (dC_{N-1}^1 X_{j-N+2} + C_{N-2} X_{j-N+2}) \dots \quad (2) \\ &\quad \text{where } dC_k^1 = C_k - C_{k-1}, \quad k=1 \text{ to } (N-1) \end{aligned}$$

그림 1(a)은 4-tap FIR Filter의 기본적인 구조로써, 필터는 입력과 계수의 곱들의 합으로 표현됨을 보여주고 있다. 이와 달리, DCM은 필터의 계수와 입력을 직

접 곱하는 형태가 아니라, 입력과 필터 계수의 차이만을 곱하고, 이전 계수들의 결과를 순차적으로 더함으로써 직접 곱하는 경우와 동일한 값을 얻는 방법이다. 수식 (2)는 수식 (1)을 필터 계수의 차이로 재구성하여 DCM 알고리즘으로 정의한 것이며 이를 구현한 형태가 그림 1(b)이다. 이 알고리즘의 확장된 형태로 필터 계수의 차이뿐만 아니라 입력의 차이도 동시에 고려하는 방법이 DCIM이다^[2].



(b) Differential Coefficients Method (DCM)

그림 1. 4-탭 FIR 필터 구조

Fig. 1. 4-tap FIR filter structure.



그림 2. DWT를 이용한 변환 결과

Fig. 2. Image analysis with 3 octaves.

(2) DWT(Discrete Wavelet Transform)

DWT는 2D 영상을 압축하는 방법의 하나로서 저 비트율로 통신해야 하는 제한된 Bandwidth 환경 하에서 DCT(Discrete Cosine Transform)보다 압축률이 높은 것으로 알려져 있다. Mallat과 Daubechies 등에 의해 수식 (3)과 같이 MRA(Multi-Resolution Analysis)가 가능하며, FIR 필터를 이용한 고속 알고리즘이 존재한다는 사실이 밝혀진 후 지난 10여년간 이에 대해 많은 VLSI 구조 연구가 진행되어 왔다^[5-10].

$$c_j(n) = \sum_{k=0}^{N-1} h(k-2n)c_{j-1}(k) \quad (3)$$

현재 출력 C_j 를 얻기 위해 이전에 필터링 된 결과(C_{j-1})를 다시 입력으로 받아 여러 번 반복하는 구조이며, $h(-n)$ 과 같은 필터 계수에 따라 저대역 또는 고대역 필터링이 된다. 그림 2)는 이미지 영상을 각각 가로 및 세로 방향으로 3회 반복 필터링하여 변환한 결과이다. DWT를 적은 수의 곱셈기로 구현한 방법으로 준순환적 피라미드(Semi-recursive pyramid) 구조가 있다^[10]. 이 구조의 1차원 FIR 필터 모듈은 그림 3)과 같으며, 본 논문에서는 이를 기반으로 분석하였다. 이는 Daubechies 4-탭 필터 계수를 사용하여 저대역 필터링과 고대역 필터링을 매 클럭 번갈아가면서 수행하는 방식을 택하고 있다. 이러한 1D 필터 모듈 4개를 사용하고, 8×8 서브 블록에 기반하여 영상을 압축할 수 있도록 되어 있다. DCT는 서브 블록 내에서만 변환이 이루어지지만 DWT는 현재 블록을 필터링하기 위해 이전 블록의 일부 픽셀들을 필요로 한다. 그러므로, blocking artifact가 발생하지 않는 장점이 있다.

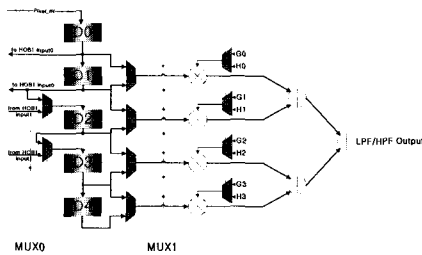


그림 3. 준-순환적 DWT의 1D 필터 구조
Fig. 3. 1D FIR filter of semi-recursive pyramid architecture.

(3) Motivations

ROM을 이용한 방법은 입력의 크기가 작은 경우엔 상대적으로 ROM의 면적 비율이 매우 크므로 적용하기 어렵다. 예를 들어, 8×8 곱셈을 4×4 곱셈으로 표현할 경우 ROM의 크기는 256×8 이 된다. 그러므로, 8비트로 표현되는 이미지 영상 처리에 활용하기에는 부담스럽다.

DCIM은 필터 계수의 차이 또는 입력의 차이를 이용하므로 그 차이가 적다면 적은 수의 비트로 구현 가능하고, 곱셈기의 크기도 줄일 수 있다. 그러나, 이전에 계산된 곱셈 결과를 계속해서 더해야 하므로 추가적인 리소스가 필요하며 필터 계수의 차이가 클 경우 표현해야 하는 비트 수가 그다지 줄지 않으므로 만족할만

한 효과를 얻을 수 없다. 필터 계수의 차이를 이용하려면 계수들이 높은 상관성을 갖도록 설계해야 하므로 경우에 따라서는 새로운 필터를 설계해야 하는 부담이 있다. 본 논문처럼 2D 영상 데이터를 입력으로 사용하는 경우를 생각해 보자. 전체적으로 연속하는 픽셀 값의 차이가 적지만 경우에 따라선 타일 무늬처럼 0에서 255로 급격히 변화하는 경우도 있다. 즉, 최대 차이가 255이므로 곱셈기의 크기를 줄일 수가 없으므로 2D 영상 처리에 적합하지 않다.

또한, 일반적으로 2D 필터링은 1D 필터링을 가로 방향과 세로 방향으로 분리하여 수행하며, 변환 메모리(Transpose Memory)를 줄이기 위해 전체 2D 영상을 8×8 픽셀로 이루어진 조그마한 서브 블록으로 나누어서 필터링하는 방법을 많이 사용한다. DCM을 DWT에서 사용되는 FIR 필터에 적용한다면, 다음에 처리될 바로 이웃한 블록의 변환을 위해 현재 블록의 부분 곱셈 결과를 따로 저장해야 한다. 그리고, 한 row의 연산이 끝나고 다음 row를 처리하기 위해서는 이전에 계산된 값들은 유효하지 않으므로 리셋시켜주어야 하며 Latency가 증가하는 단점이 있다. 따라서, DCM 알고리즘을 2D 영상 필터링에 적용하기는 어렵다.

이러한 문제점의 해결을 위해 DCM처럼 입력 데이터의 높은 상관성을 이용하면서 서브 블록에 기반한 2D 영상 처리에 효율적인 방법이 필요하다.

III. SMT(Separated Multiplication Technique)

(1) Array multiplier의 단점

2D 영상에서 이웃한 픽셀 값의 차이는 매우 적는데, 일반적으로 이를 일컬어 공간 상관성이 높다고 한다. 8비트 흑백 영상에 대해서 상위 3비트가 전체 영상에서 이웃한 픽셀과 일치할 확률이 대략 80%정도로 상관성이 높음을 모의 실험을 통해 확인하였다. 상위 4비트는 60%이었다. 이제 이러한 특성을 지닌 2D 영상을 어레이 곱셈기를 이용하여 필터링하는 구조를 생각해 보자. 실제 8비트 또는 경우에 따라서는 16비트에 대해서도 어레이 곱셈기를 많이 사용한다. 곱셈기의 입력으로 상위 비트의 변화가 적은 데이터가 연속적으로 곱셈기의 입력으로 유입된다고 가정하자. 그러나, 하위 비트의 천이는 빈번히 일어나며 어레이 곱셈기의 구조상 하위

비트의 변화가 상위비트를 연산하는 덧셈기(Adders)에 까지 영향을 미치게 된다. 즉, 입력의 변화가 적다고 하여도 어레이 곱셈기는 내부적으로 수많은 천이가 일어날 수 밖에 없다. 이는 곧바로 곱셈기에서의 전력 소모를 증가시킨다. FIR 필터에서 곱셈기가 차지하는 전력 소모는 80%~90%에 이르기 때문에 곱셈기에서 발생하는 천이의 수를 줄인다면 전력 감소에 큰 이득을 볼 수 있다. 따라서, 그림 5)처럼 곱셈 연산을 둘로 나누는 방법을 제안하며 이를 SMT라 부르기로 한다.

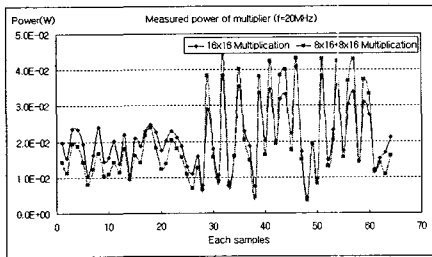


그림 4. Hspice로 측정된 곱셈기의 소모 전력
Fig. 4. Measured power of multiplier.

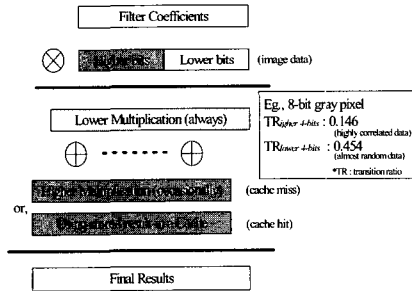


그림 5. SMT(곱셈 연산 분할 기법)
Fig. 5. Separated Multiplication Technique.

(2) SMT with no buffer

그림 4)는 16×16 곱셈기를 합성한 뒤 Hspice로 실제 소모되는 전력을 측정된 결과이다. 하나는 기존의 경우처럼 16×16 곱셈기 하나만을 사용한 경우이며, 다른 하나는 8×16 곱셈기 2개로 나누고 각 결과를 더하여 얻은 결과이다. 테스트 입력은 실제 영상의 8×8 한 블록, 64개의 샘플을 취하였다. 대체로 곱셈기를 두 개로 나누어 계산하여도 소모되는 전력이 5% 정도 감소됨을 알 수 있었다. 샘플 후반의 그래프가 급격히 변화하는 이유는 추출한 샘플 데이터는 필터 모듈에서 내부적으로 사용되는 데이터로서 양과음(+/-)이 번갈아가면서 나타났기 때문이다. (비트의 변화가 매우 크다). 번갈아

가면서 나타나는 입력도 상위 비트가 거의 같으므로 이를 버퍼링하여 재사용할 수 있다면 더 큰 전력 이득 효과를 볼 수 있음을 짐작할 수 있다. 따라서, 제안하는 SMT처럼 캐쉬를 사용하는 것이 효율적이다.

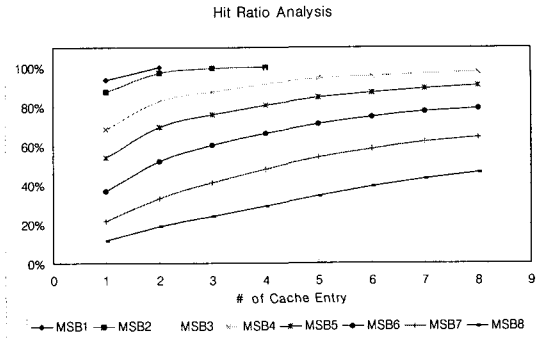


그림 6. 비트 크기와 캐쉬 엔트리에 따른 적중률
Fig. 6. Hit-ratio according to two variables. One is the compared bit-width and the other is storage size, the number of cache entry.

(3) SMT with buffer/cache

8비트 영상 데이터를 상위 4비트와 하위 4비트로 구분되었을 때, 입력 스트림에서의 비트 변화를 실험한 결과, 천이하는 비율이 각각 0.146과 0.454로 나타났다(그림 5). 상위 4비트의 경우 매 클럭마다 평균적으로 4비트 중에서 0.584(4×0.146)비트만이 변화하고 하위 4비트는 그 확률이 0.454로 이미 상관성(correlation)이 거의 없는 random data임을 확인할 수 있었다. 따라서, 천이가 거의 없는 상위 비트와 자주 천이가 발생하는 하위 비트의 필터링을 나누어서 수행하는 구조를 생각할 수 있다. 이에 대한 수학적 표현은 수식 (4)와 같이

$$\begin{aligned}
 & \text{If (unsigned multiplication)} \{ \\
 & \quad A(N\text{-bits}) \times B(M\text{-bits}) \\
 & = \left(\sum_{k=0}^{N-1} a_k 2^k \right) \times \left(\sum_{k=0}^{M-1} b_k 2^k \right) \quad (a_k, b_k \in \{0,1\}) \\
 & = \left\{ \left(\sum_{k=0}^{N-1} a_k 2^k \right) \times \left(\sum_{k=0}^{M-1} b_k 2^{k+l} \right) \right\} + \left\{ \left(\sum_{k=0}^{N-1} a_k 2^k \right) \times \left(\sum_{k=l}^{M-1} b_k 2^k \right) \right\} \\
 & \quad (l = l\text{-bits from MSB}) \\
 & = \text{higher-bits computation} + \text{lower-bits computation} \\
 & \} \tag{4}
 \end{aligned}$$

정리된다. 이러한 구조를 택함으로써 하위 비트의 천이가 상위 비트를 연산하는 덧셈기에 영향을 끼치지 못

하도록 막을 수 있으므로 하나의 곱셈기를 사용할 때와 비교하여 내부적인 천이의 수가 급격히 줄어든다. 상위 비트의 변화는 거의 없으므로 이에 대한 연산의 결과는 버퍼(캐쉬)에 저장하여 재사용한다. 하위 비트의 연산은 매번 일어나고 상위 비트의 연산은 캐쉬를 참조하여 hit가 발생하면 이미 저장되어진 결과를, miss가 발생하면 새롭게 계산한 결과를 하위 비트의 연산 결과와 더하여 최종 결과를 얻을 수 있다. 이 때 사용되는 버퍼는 매우 작은 크기의 단순화된 캐쉬로 구현 가능하다. Cache miss에 의해 계산된 결과는 캐쉬에 저장된다. 따라서, 상위 비트를 연산하는 곱셈기는 miss가 발생했을 때에만 동작하므로 캐쉬 적중률이 높을수록 전력 감소 효과는 커진다. 캐쉬의 hit/miss 신호는 곱셈 연산 이전에 체크가 되어 상위 비트를 연산하는 곱셈기를 동작시키는 데에는 문제가 없으며, 곱셈 결과를 캐쉬에 업데이트함에 있어서도 성능 저하를 야기하지 않는다. 이제 남은 것은 어떻게 입력을 상위 비트와 하위 비트로 구분해야 하는지와 최적화된 캐쉬의 크기는 얼마인지를 정하는 것이다.

IV. 제안하는 구조

(1) 최적화된 스펙(specification) 정하기

SMT를 구현하기 위해선 다음의 두 가지를 최적화해야 한다. 하나는 입력의 상위 비트와 하위 비트를 구분하는 위치를 정하는 것이고, 다른 하나는 캐쉬의 크기를 정하는 것이다. 모의 실험에 사용한 입력은 256x256 크기의 8비트 흑백 영상이며 C로 DWT를 모델링하여 스펙을 결정하였다. 그림 6)에는 상위 1비트에서부터 8비트까지 실험한 8개의 곡선이 도시되어 있다. x축은 캐쉬 엔트리의 수를 나타낸다. 예를 들어, MSB3은 8비트 입력을 상위 3비트와 하위 5비트로 구분하였다는 것을 의미한다. 일반적으로 캐쉬의 적중률은 80%~100% 사이에서 결정을 하므로 그림 6)의 결과를 토대로 곱셈기는 상위 4비트와 하위 4비트를 기준(MSB4)으로 입력을 나누고, 캐쉬 엔트리는 4개만 사용하는 것이 최적화된 결과이다. 이 때의 적중률은 91.60%이다. 캐쉬 엔트리의 수가 4보다 작다면 캐쉬의 적중률이 80% 이하로 감소되어 큰 전력 이득 효과를 기대할 수 없으며, 4보다 크게 선택된다면 면적이 급격히 증가한다.

(2) 제안하는 1D FIR 필터

그림 7)은 모의 실험 결과를 바탕으로 SMT를 적용한 구조이며 상위 비트 연산 부분을 기준으로 도시하였다. 입력의 기본 크기는 4비트이며 곱셈기 이후의 크기는 20비트이다. 하위 비트를 연산하는 부분은 캐쉬와 연결만이 없는 단순한 구조이므로 그림 7)에서 곱셈 결과와 캐쉬 데이터를 선택하는 MUX 이전은 동일하며, 이후의 리소스는 공유하는 구조이다. 단, 곱셈기의 입력을 제어하는 MUX의 제어 신호만이 다를 뿐이다. 하위 비트의 경우엔 매 클럭마다 연산이 이루어지므로 MUX1의 제어 신호가 그대로 사용된다. MUX1은 저대역 필터링과 고대역 필터링을 번갈아가면서 수행하도록 제어하는 신호이다. 그러나, 상위 비트를 연산할 경우에는 cache miss가 발생할 때에만 상위 비트용 곱셈기가 동작을 한다. 따라서, MUX1의 제어 신호는 cache miss 신호와 AND 게이트를 거친 신호로 제어되어야 한다. 이는 gated clock의 원리와 같다. D0~D4는 입력으로 들어온 픽셀이 순차적으로 옮겨가는 레지스터이고 앞 부분의 MUX는 서브 블록 단위로 필터링을 하므로 중첩되는 픽셀을 처리하기 위한 부분이다. 먼저, 서브 블록간에 중첩되는 픽셀을 처리함에 있어서 기존의 구조에서 잘못 구현된 부분을 수정하였고, 필터 계수를 저대역 필터 계수만 사용함으로써 캐쉬의 크기를 줄이도록 변경하였다. Daubeches 저대역 필터 계수와 고대역 필터 계수는 상보적(complementary)인 관계에 있기 때문이다. SMT를 적용하여 각각의 곱셈기가 2개의 조그마한 곱셈기로 나뉘어졌고, 다시 그 결과를 더하기 위한 덧셈기가 추가되었다. 그리고, 그림 7)에 표시된 수치는 필터 모듈에서 소모하는 전력을 보다 정확하게 계산하기 위해 Verilog HDL 시뮬레이션을 통해 얻은 각 노드에서의 천이 비율이다. 소괄호 안의 수치는 하위 비트를 연산하는 노드에서의 값이다.

일반적인 캐쉬는 프로세서와 메모리 모두와 통신하는 양방향성이지만, 본 구조에서는 곱셈기에만 종속된 단방향성이다. 캐쉬의 내용을 참조하기 위한 태그(tag)는 입력의 상위 비트가 되며, 그 내용은 필터 계수와 곱해진 결과이다. 필터 계수가 N개이면 상위 비트를 연산하는 곱셈기도 N개이다. 그리고, 캐쉬도 N개의 곱셈기에 독립적으로 응답해야 하므로 N개의 I/O port가 있어야 한다. 즉, N개의 bank가 있고 캐쉬의 태그를 저장하는 메모리는 모든 bank가 공유하고 있는 형태를 지닌다. 이러한 점들을 고려하여 본 구조에 알맞게 캐쉬를 단순화하였다. 캐쉬의 크기는 곱셈 결과의 비트

수와 캐쉬 엔트리 수에 의해 결정된다. 필터 계수는 16

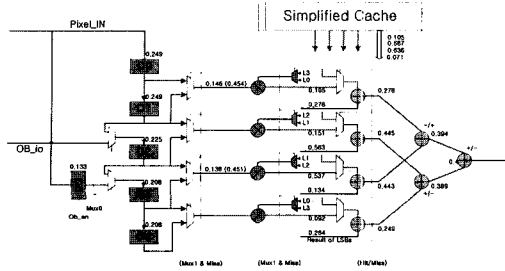


그림 7. 제안하는 1D FIR 필터 구조

Fig. 7. Proposed 1D FIR filter structure.

비트이고 입력의 상위 부분은 4비트로 표현되므로 곱셈 결과는 20비트가 된다. 또한, 엔트리 수가 4이고 4 탭 필터이므로 캐쉬의 크기는 40바이트(20×4×4)이면 충분하다. 마이크로프로세서에서는 캐쉬의 기본 단위로 SRAM cell 사용한다. 그러나, 본 논문에서는 D-Latch로 구현하였기에 면적면에서 큰 차이를 보인다. 따라서, 실제 SMT를 적용할 경우에 SRAM cell로 구현한다면 면적을 크게 감소시킬 수 있다.

V. 결과 분석

전력 소모와 면적 비교는 (구)LG반도체의 0.6um - 5V Datapath Library를 바탕으로 추정된 것이다. 라이브러리에서 제공하는 전력 및 면적 예측 모델을 사용하여 다음의 결과를 얻을 수 있었다. 또한 면적은 Verilog HDL로 서술하고 IDEC Library (구)LG반도체의 0.6um 공정을 기반으로 만들어졌음 -를 사용하여 Synopsis로 합성 및 검증 과정을 거쳤다.

(1) 전력 소모 비교

표 1)은 하나의 곱셈기에 SMT를 적용한 경우 각각의 구성 요소에서 소모되는 전력과 면적을 예측한 예이다. %ofActiveBits는 입력의 천이 비율을 나타내며, extAF는 각 셀이 동작하는 비율로써 상위 비트 연산과 관련된 부분이 매우 낮음을 알 수 있다. 이는 캐쉬의 적중율이 적용되었기 때문이다. Cell Power는 라이브러리에서 제공하는 각 셀에 bit-width를 고려한 평균 소모 전력이다. 입력의 bit-width가 N, M비트인 곱셈기는 다음의 수식으로 계산된다. $(-263.90 \times (17.80 \times N \times M)) \times \text{intAF}$. Cext는 출력단에 연결되는 load capacitance이다. 위에 열거된 변수들의 보다 정확한

의미와 계산 방법은 앞에서 언급한 라이브러리를 참조하기 바란다. 위의 변수들을 이용하여 각 셀에서 소모하는 전력은 다음의 수식에 의해 구해진다.

$$\text{PowerDissipation} = [P + ((V_{dd}^2 \times C_{ext}) \times AF)] \times F$$

표의 하단에는 SMT를 이용했을 때의 이득과 캐쉬의 overhead를 정리하였다. 곱셈기에 SMT를 적용하면 15%의 전력 이득 효과를 얻을 수 있다.

표 2)에는 준순환적 피라미드 구조에서 사용되는 HDWT1, VDWT1, HDWT2, VDWT2라는 4개의 모듈에 대한 전력 소모량이 비교되어 있다. HDWT1은 다른 모듈과 달리 8×16 곱셈기를 사용하였고, 나머지는 16x16 곱셈기를 사용하였다. 따라서, HDWT1에서 소모하는 절대적인 전력 소모량이 다른 모듈에 비해 약 절반 가량 된다. 이는 FIR Filter의 소모 전력 중에서 곱셈기가 차지하는 비율이 매우 높음을 나타내며, 표 1)의 방식대로 계산한 결과 80%~90%를 차지한다. 4개의 필터 모듈을 통틀어 기존의 구조에 비해 10%~15% 정도의 전력 이득 효과가 있다.

(2) 면적 증가 및 성능 향상 비교

그림 3)과 그림 7)을 비교하여 볼 때, 몇몇 구성 요소가 더 사용되었으며 캐쉬도 추가되었다. 따라서, 표 2)에서와 같이 면적이 증가하였다. HDWT1의 면적 증가가 큰 이유는 HDWT1 모듈에서 사용된 곱셈기는 다른 모듈에서의 곱셈기보다 작은 반면 캐쉬의 크기는 별 차이가 없기 때문에 상대적으로 면적이 더 증가해 보이는 것이다. 변환 메모리(Transpose Memory)는 핑퐁(Ping-Pong)구조로 8×8 서브블록의 데이터를 저장할 수 있을 만큼 큰 블록이며 D-Latch로 모델링되었다. 변환 메모리의 게이트 수가 매우 크므로 전체 DWT구조에서 큰 비중을 차지한다. 따라서, 표 3)과 같이 각각의 필터 모듈과 변환 메모리 및 control unit의 크기를 고려할 때 전체적으로 20%의 면적 증가가 있었다. 각 필터 모듈에서 사용된 캐쉬는 D-Latch로 구현되었으나, 마이크로프로세서에서 널리 사용되는 것처럼 메모리 셀로 구현된다면 면적 증가는 크게 문제가 되지 않는다.

입력을 상위 비트와 하위 비트로 구분하여 필터링을 하기 때문에 더 작은 크기의 곱셈기가 사용되었다. 입력의 bit-width 감소는 어레이 곱셈기에서 critical path의 감소로 직결된다. 따라서, 그 만큼의 속도 향상을 가

표 1. 각각의 셀에서의 전력 계산 예

Table 1. Example of power estimation per cell.

Unit	pF	uW/MHz	pF	uW/MHz	Kum ²	Eqa Gates
Type	%ActiveBits	extAF	Cell Power	Cost	Total Power	Area(A)
Conventional Architecture						
D F/F (pixel)	0.249	0.125	65.760	0.416	1146.660	8.362
MUX	0.249	0.037	2014.500	0.024	40303.968	216.640
D F/F (Out)	0.192	0.059	111.520	0.832	2259.535	16.784
			Sum		43710.223	241.816
Proposed Architecture						
Datapath						
D F/F (High)	0.146	0.044	2.342	0.208	52.377	4.196
D F/F (Low)	0.454	0.136	27.980	0.208	574.714	4.196
MUX (High)	0.146	0.022	73.525	1.055	1484.555	105.560
MUX (Low)	0.454	0.068	875.300	0.864	17541.545	105.560
D F/F (High_O)	0.105	0.032	11.710	0.960	252.786	20.930
D F/F (Low_O)	0.278	0.033	111.520	1.184	2289.856	16.784
MUX	0.374	0.034	83.350	1.340	1843.206	11.530
AUD	0.239	0.060	305.877	0.624	9140.191	28.840
D F/F (F_Out)	0.233	0.072	111.520	0.832	2266.642	16.784
Cache						
TagRAM	0.286	0.085	36.032	0.328	757.705	16.784
DataRAM	0.359	0.109	68.960	1.020	1445.947	48.800
TreeBuffer	0.038	0.025	98.000	0.854	1774.429	34.160
Control & Others	0.324	0.081	54.881	0.384	1115.658	20.054
			Cache Sum		5133.640	119.808
			Total Sum		37519.511	436.288
			Cache Ratio		0.137	0.275
			Ratio		0.860	1.804

표 2. 각 필터 모듈에서의 소모 전력과 면적 비교

Table 2. Power and area gain in each filter module.

(Power: uW/MHz Area: Equi. Gates)

	HDWT1			VDWT1		
	Conventional	Proposed	Gain	Conventional	Proposed	Gain
Power	1899.27	1632.33	-13.6%	3362.32	2921.61	-13.1%
Area	5536	9786	77.0%	9872	13186	33.8%

	HDWT2			VDWT2		
	Conventional	Proposed	Gain	Conventional	Proposed	Gain
Power	3456.43	3025.94	-12.4%	3502.14	3088.67	-11.8%
Area	10012	13364	33.5%	10012	13384	34.5%

표 3. 2D DWT 칩에서의 면적 비교

Table 3. Area overhead in 2D DWT chip.

Functional Block	Equivalent Gate Count		Percentage
	Conven.	Proposed	
First Octave Filter	39,877	47,451	118.99%
Second Octave Filter	29,678	35,778	120.55%
Output Control	5,001	5,001	100.00%
Total Control	204	223	109.31%
Total Gates	74,760	88,453	118.32%

저온다. HDWT1 모듈의 경우 8x16 곱셈기가 4x16 곱셈기 2개로 나뉘어졌고, 나머지 모듈에서는 16x16 곱셈기가 6x16과 10x16 곱셈기로 나뉘어졌으므로 전체 시스템에서의 critical path는 10x16 곱셈기에 의해 결정된다. HDWT1의 경우 8비트 입력의 radix point가 비트^[0]이고, 나머지 모듈의 경우 비트^[10]이므로 모든 실험 결과의 최적화된 결과를 만족하기 위해선 16비트의 경우엔 상위 6비트와 하위 10비트로 나누어야 한다. 이를 검증하기 위해 IDEC의 standard cell library를 사용하여 Synopsis로 합성한 결과 10x16과 16x16 곱셈기, 캐쉬의 delay는 각각 39.84ns, 47.59ns, 16.36ns로 캐쉬의 delay는 문제가 되지 않았다. 또한, 위의 곱셈기에 대한 실험 수치를 비교해 볼 때 20%정도의 속도 향상을 기대할 수 있다. 준순환적 피라미드 구조는 실시간

압축이 가능하기 때문에, 제안하는 구조에서 얻은 속도 향상을 공급 전원을 낮추는데 사용한다면 더 많은 전력을 감소시킬 수 있다.

VI. 결 론

DWT와 같은 FIR Filter를 사용하는 구조에서는 많은 수의 곱셈기가 쓰이며 이들이 차지하는 전력은 전체의 80%~90%에 이른다. 따라서, 저전력의 FIR Filter를 구현하기 위해 영상의 높은 공간 상관성을 이용한 방법을 제시하였고, 이를 SMT(Separated Multiplication Technique)로 명명하였다. 8비트 영상 입력의 경우 상위 4비트와 하위 4비트로 구분지어 각각에 대해서 곱셈 연산을 수행한다. 그리고, 상위 비트의 결과는 엔트리 수가 4인 캐쉬에 저장하여 재사용하도록 하였다. 이렇게 함으로서 DWT의 FIR 필터에서 약 15% 정도의 전력 이득 효과를 볼 수 있었다. 또한, 작은 크기의 곱셈기를 사용하므로 기존의 구조보다 빠른 연산을 수행할 수 있다. 속도 향상에 의한 이점을 공급 전원을 낮추는데 이용한다면 더 높은 전력 이득 효과를 볼 수 있을 것으로 기대된다.

참 고 문 헌

- [1] N. Sankarayya, K. Roy, and D. Bhattacharya, "Algorithms for low power and high-speed FIR filter realization using differential coefficients", IEEE Trans. Circuits and Sys. 2, Vol 44, pp. 488~497, June 1997.
- [2] T. S. Chang and C. W. Jen, Low power FIR filter realizations with differential coefficients and inputs, Proc. ICASSP, pp. 3009~3012, 1998.
- [3] S. Ramprasad, N.R. Shanbhag, and I. N. Hajj, Decorrelating(DECOR) transforms for low-power adaptive filters, Proc. ISLPED, Aug. 1998.
- [4] Shlomo Waser, Michael J. Flynn, Introduction to arithmetic for digital systems designers, New York: Holt, Rinehart and Winston: CBS Colleg Pub., pp.137~139, 1982.
- [5] S. Mallat, "A theory for multi-resolution signal

- decomposition : The wavelet representation", IEEE Trans. Pattern Anal. And Machine Intell., Vol. 11, No. 7, pp. 674~693, July 1989.
- [6] I. Daubeches, "Orthonormal bases if compactly supported wavelets", Commun. Pure Appl. Math., Vol. 41, pp. 909~996, Nov. 1988.
- [7] K. Parhi, T. Nishitani, "VLSI architecture for the discrete wavelet transforms", IEEE Trans. on VLSI Systems, Vol. 1, No.2, pp. 191~202, July 1993.
- [8] M. Vishwanath, A recursive pyramid algorithm for the discrete wavelet transform, IEEE Trans. on Signal Process., Vol.42, No.3, pp. 673~677, March 1994.
- [9] T. Denk, K. Parhi, "Calculation of minimum number of registers in 2-D discrete wavelet transforms using lapped block processing", Int. Symp. on Circuit and Systems, Vol. 26, No. 15, pp. 1184~1185, July 1990.
- [10] S. K. Paek, H. K. Jeon and L. S. Kim, "Semi-recursive VLSI Architecture for Two Dimensional Discrete Wavelet Transform", ISCAS 1998.

저 자 소 개



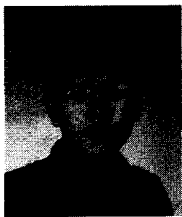
韓 昌 永(學生會員)

1998년 2월 : 한국과학기술원 전기 및 전자공학과 졸업(공학사). 2000년 2월 : 한국과학기술원 전자전산학과 전기및전자공학 전공 졸업(공학석사). 2000년 3월~현재 : 한국과학기술원 전자전산학과 전기및전자공학 전공 박사과정. <주관심 분야> 3D Graphics, Graphics Processor, VLSI 회로 설계



金 利 燮(正會員)

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



朴 炯 俊(學生會員)

1995년 2월 : 한국과학기술원 전기 및 전자공학과 졸업(공학사). 1997년 2월 : 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 1997년 3월~현재 : 한국과학기술원 전자전산학과 전기및전자공학 전공 박사과정. <주관심 분야> 3D Graphics, Graphics Processor, VLSI 회로 설계