

패턴 탐색 기법을 사용한 Multiplierless 리프팅 기반의 웨이블릿 변환의 설계

손창훈[†], 박성모^{**}, 김영민^{***}

요 약

본 논문은 하드웨어 곱셈 연산을 최적화하여 리프팅 기반의 9/7 웨이블릿 필터의 개선된 VLSI의 구조를 제안한다. 제안한 구조는 범용 곱셈기를 사용하는 기존의 리프팅 기반의 웨이블릿 필터와 비교하여 화질의 열화 없이 보다 적은 로직과 전력소모를 갖는다. 본 논문은 Pattern search 기반의 Lefèvre 알고리즘을 이용하여 하드웨어 구조를 개선한다. 제안한 구조는 범용의 곱셈기를 단순한 shift-add 연산으로 대체하여 하드웨어 구현을 단순하게 하고 계산 속도를 빠르게 한다. 제안한 구조와 기존의 구조를 Verilog HDL을 이용하여 구현하고 비교 실험하였다. 두 구조는 0.18um 디지털 CMOS 공정의 스탠다드 셀을 이용하여 합성된다. 제안한 구조는 200MHz의 합성 타겟 클럭 주파수에서 기존의 구조에 비해 면적, 전력소모와 최대 지연시간이 각각 약 51%, 43%와 30%로 감소하였다. 구현 결과를 통해 제안한 구조가 범용의 곱셈기 블록을 사용한 기존의 구조보다 스탠다드 셀을 이용한 ASIC 구현에 보다 적합하다는 것을 보여준다.

Design of Multiplierless Lifting-based Wavelet Transform using Pattern Search Methods

Chang Hoon Son[†], Seong Mo Park^{**}, Young Min Kim^{***}

ABSTRACT

This paper presents some improvements on VLSI implementation of lifting-based 9/7 wavelet transform by optimization hardware multiplication. The proposed solution requires less logic area and power consumption without performance loss compared to previous wavelet filter structure based on lifting scheme. This paper proposes a better approach to the hardware implementation using Lefèvre algorithm based on extensions of Pattern search methods. To compare the proposed structure to the previous solutions on full multiplier blocks, we implemented them using Verilog HDL. For a hardware implementation of the two solutions, the logical synthesis on 0.18 um standard cells technology show that area, maximum delay and power consumption of the proposed architecture can be reduced up to 51%, 43% and 30%, respectively, compared to previous solutions for a 200 MHz target clock frequency. Our evaluation show that when design VLSI chip of lifting-based 9/7 wavelet filter, our solution is better suited for standard-cell application-specific integrated circuits than prior works on complete multiplier blocks.

Key words: Multiplierless Constant multiplication(무곱셈기의 상수 곱셈), pattern research(패턴 탐색), DWT(이산 웨이블릿 변환), Lifting(리프팅)

※ 교신저자(Corresponding Author): 김영민, 주소: 광주광역시 북구 용봉로 77 전남대학교 (500-757), 전화: 062) 530-0812, FAX: 062)530-0813, E-mail: kym@chonnam.ac.kr
접수일: 2009년 12월 7일, 수정일: 2010년 2월 12일
완료일: 2010년 4월 21일

[†] 준회원, 전남대학교 전자컴퓨터공학과 박사과정
(E-mail: chson03@moiza.chonnam.ac.kr)

^{**} 정회원, 전남대학교 전자컴퓨터공학과 정교수
(E-mail: smpark@chonnam.ac.kr)

^{***} 정회원, 전남대학교 전자컴퓨터공학과 정교수

1. 서 론

영상 및 신호 처리 회로를 설계함에 있어 가능한 한 범용의 곱셈기를 사용하는 대신 그 회로에 맞는 전용의 연산기를 사용함으로써 회로의 속도, 면적, 전력 소모의 최적화가 가능하다. 만약 곱셈 연산 시 승수가 상수이면, 완전한 범용 곱셈기를 대신하여 그와 같은 연산을 수행하기 위해 shift와 addition/subtraction을 사용할 수 있다. 이러한 Constant multiplication 방법[1]을 이용하여 보다 작은, 더 빠른, 전력 소모가 덜한 회로를 만들 수 있다. 상수에 의한 곱셈은 디지털 신호 처리, 영상 처리, 데이터 통신 등의 응용 분야에 광범위하게 적용될 수 있다. 예를 들어, finite impulse response(FIR) 필터, discrete cosine transform(DCT)과 discrete wavelet transform(DWT)은 곱셈 연산이 매우 집중되면서도 계산량이 매우 많은 회로들이다. 따라서 이 회로들의 최적화는 그 회로를 사용하는 전체 시스템의 성능에 크게 영향을 준다. Constant multiplication을 이용한 최적화 설계 기법은 오랫동안 연구되어 왔다. 예를 들어, 가장 잘 알려진 Booth 알고리즘은 상수 곱셈 뿐 아니라 범용 곱셈기에 쉽게 적용될 수 있다. 이 알고리즘은 이진수로 표현된 상수안의 연속적인 1의 모임을 피하고 1의 개수를 줄인다. 이보다 나은 방법들이 Common sub-expression, Simulated annealing, Pattern search method 등을 기반으로 연구되고 있다[2]. Multiplierless constant multiplication의 연구 동향은 아래와 같이 요약될 수 있다.

1.1.1 Straightforward shift-add algorithm

상수에 대한 일반적인 이진수 표현(binary representation)의 곱셈 연산은 shift-and-add 연산으로 대체하여 하드웨어 구현 비용을 감소시킬 수 있다. 곱셈 연산시의 부분 곱들은 이진수 표현 내의 1의 수에 따라 왼쪽 shifting과 add 연산을 통해 간단하게 계산된다.

1.1.2 Canonical signed recoding

CSD(Canonical signed digit) 형식은 상수 k 의 이진수 표현 안의 redundancy를 줄이기 위해, $\{1, 0, 1\}$ 을 사용한다(1은 -1을 의미). 예를 들어, 이진수 01111은 $1000\bar{1}$ 과 같이 recoding 된다. 곱셈 연산에

서, 1은 덧셈기와 같은 하드웨어 구현 비용을 갖는 뺄셈기로 대체된다. 이 방법을 쓰면 straightforward 알고리즘보다 상수 k 내의 1의 숫자를 더 줄일 수 있다[2].

1.1.3 Cost-Function-based search method

앞서 설명한 방식들은 주어진 constant multiplication에서 shift-and-add에 사용되는 가장 짧은 코드를 만들어 낼 수 없다. Bernstein 알고리즘[3]이라고도 불리는 Cost 기반의 search 방식은 recoding 과정에서 발생한 중간 결과들을 재사용한다. 상수 $k=657$ 과 입력 x 의 경우에, $kx = (8x+x) + 8(8x+x) + 64(8x+x)$ 로 단지 세 번의 덧셈과 세 번의 시프트로 계산될 수 있다($8x+x$ 가 재사용됨). 그러나 이 알고리즘은 지수적인 복잡성을 갖기 때문에 32비트 이상의 상수들에서 실제로 적용되기 어렵다.

1.1.4 Pattern Search Methods와 Lefèvre 알고리즘

대부분의 Pattern search 방법들은 CSD 형식으로 recode된 값을 초기 값으로 하여 공통의 sub expression들을 공유하고 재사용하여 최적화한다. 논문 [4]에서는, 그래프를 이용하여 덧셈기와 뺄셈기의 수를 줄인 FIR 필터의 최적화된 결과를 만들어 낸다. 2001년에, Lefèvre[5]는 기존의 pattern search 방식보다 더 효율적인 새로운 알고리즘을 제안하였다. Lefèvre 알고리즘과 이전의 pattern search 알고리즘들과의 가장 큰 차이는 pattern의 negative 버전을 사용하여 보다 나은 최적화 결과를 만들어 낸다는 것이다. 예를 들어, 숫자 $51 = 10\bar{1}010\bar{1}_2$ 의 경우, $10\bar{1}$ 패턴은 두 개의 positive 버전($10\bar{1}$)과 negative 버전($\bar{1}01$)을 합쳐 세 번 발생한다. 숫자 51 안에는 shift 0을 통한 positive 버전, shift 2를 통한 negative 버전과 shift 4를 통한 positive 버전의 패턴들이 있다. 그러나 첫 번째와 세 번째 패턴들은 둘 다 두 번째 패턴에 포함되어 중복되게 된다. 그러나 숫자 51 안에는 또 다른 $1000\bar{1}$ 패턴이 shift 0을 통한 negative 버전($\bar{1}000\bar{1}$)과 shift 2를 통한 positive 버전이 두 번 나타난다[2]. Lefèvre 알고리즘은 이와 같이 negative 버전을 사용함으로써 보다 나은 최적화 결과를 얻을 수 있다. 본 논문에서는 Pattern search 방식의 알고리즘을 리프팅 기반의 DWT 필터에 적용하여 회로의 VLSI 구조를 최적화하였다. 영상 압축에서

DWT는 뛰어난 주파수 분석 특성으로 인해 광범위하게 사용되고 있다. Cohen-Daubechies-Feauveau (CDF) 9/7 biorthogonal 웨이블릿은 선형 위상 지연 특성을 갖기 때문에, 정지 영상 압축 표준인 JPEG2000에서 사용된다. 컨볼루션 기반의 DWT는 과도한 연산량과 메모리를 요구하기 때문에, 최근의 연구들은 보다 VLSI 구현에 적합한 리프팅(Lifting) 기반의 DWT 구조를 사용한다[6].

제안한 구조는 Verilog HDL로 구현하고 디지털 CMOS 공정을 이용하여 합성한 후, 기존의 다른 리프팅 필터의 VLSI 구조들과 비교 분석을 하였다. 기존의 구조에 비해 면적, 속도, 전력 소모 면에서 상당한 개선을 얻을 수 있었다.

1.2 리프팅 방식의 CDF 9/7 웨이블릿 변환

리프팅 방식은 spatial 접근을 통해 웨이블릿을 만들어 내는 방법이며 적은 연산과 in-place 속성, 경계 확장의 손쉬움 등 많은 장점을 갖는다. [7]에 따르면, perfect reconstruction을 만족하는 모든 DWT 필터뱅크는 리프팅 과정들의 유한한 sequence로 분해 가능하다. 이러한 분해 방식은 타겟 웨이블릿 필터의 다각 행렬을 적절한 upper와 lower 삼각 행렬들과 constant 대각선 행렬로 인수분해 하는 것이며 아래의 식 (1)과 (2)로 표현된다. 여기서 $h(z)$ 와 $g(z)$ 는 각각 low-pass와 high-pass analysis 필터, 식 (1)은 다각 분해이며, 식 (2)의 $P(z)$ 는 다각행렬이다. 리프팅 방식을 이용한 웨이블릿 정변환은 크게 분할(split), 예측(predict), 갱신(update), 조정(scaling) 단계로 이루어진다. 먼저 입력 신호 $x[n]$ 은 그림 1의 분할 과정을 거쳐 짝수 번째

$$\begin{aligned} h(z) &= h_e(z^2) + z^{-1}h_o(z^2) \\ g(z) &= g_e(z^2) + z^{-1}g_o(z^2) \end{aligned} \quad (1)$$

$$\begin{aligned} p(z) &= \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \\ &= \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \end{aligned} \quad (2)$$

신호와 홀수 번째 신호로 분리된다. 예측 단계는 짝수 번째 신호에서 예측 연산을 통해 홀수 번째 신호를 예측할 때 얻어지는 에러이며, 이것은 high-pass 필터링 결과이다. 갱신 단계는 짝수 번째 신호와 예측 에러를 더하여 구하고, 이것은 low-pass 필터링의 결과이다. 리프팅 계수 $\alpha, \beta, \gamma, \delta$ 의 곱셈을 통해

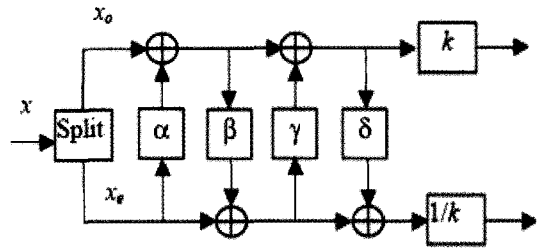


그림 1. 리프팅 방식의 CDF 9/7 DWT 구조

필터링을 수행한다. 조정 단계에서는 예측과 갱신의 결과가 일정한 상수 k 또는 $1/k$ 에 의해 크기를 조정하여 이것은 최종 결과를 얻을 수 있다. 그림 1은 본 논문에서 사용되는 리프팅 기반의 CDF 9/7 DWT의 구현 원리를 나타내는 다이어그램이다.

2. 제안 방식의 하드웨어 구조

CDF 9/7 리프팅 방식은 DWT 및 IDWT 계산 과정에서 부동 소수점 곱셈 연산이 필요하다. 그러나 이러한 곱셈기는 많은 하드웨어 자원을 사용하기 때문에 VLSI 구현에 적합하지 않아, 대부분의 리프팅 구조는 고정 소수점 방식의 범용 곱셈기를 사용한다. 9/7 웨이블릿 필터의 리프팅 계수들은 이진 표현 방법을 이용하여 나타낼 수 있다. 이러한 리프팅 계수들은 고정된 이진수의 표현이므로 복잡한 구조의 범용 곱셈기를 대신하여 Constant Multiplication 방식에 사용되는 간단한 shift-add 연산으로 대체하여 하드웨어 면적을 크게 감소시킬 수 있다. 표 1은 리프팅 필터 계수들을 계산 과정에서 충분한 정밀도를 주기 위해 15 비트를 사용하여 이진수로 표현하였다. 표 1에 나타낸 리프팅 계수들의 shifting을 통한 곱셈 연산은 이진수 표현 내 1의 개수만큼 shift와 add 연산을 필요로 한다. 예를 들어, 리프팅 계수 α 는 7 회의 왼쪽 shift 연산과 add 연산과, 계수가 음수이기 때문에, 한 번의 negation 연산을 필요로 한다. 표 1에서 보이는 바와 같이, 5에서 7회의 shift와 add-sub 연산을 사용하여 리프팅 계수들의 고정 소수점 상수 곱셈기를 구현할 수 있다. 하드웨어 구현 시 shift 연산은 단지 routing을 의미한다. 본 논문에서는, 리프팅 계수들에 Pattern search 방식을 적용하여 shift-add 연산의 횟수를 줄여, 적은 면적, 낮은 전력 소모와 빠른 동작 속도를 갖는 VLSI 구조를 제안한

표 1. 6개의 리프팅 필터 계수들의 이진수 표현

Lifting coefficient	Original CDF 9/7	Binary representation (15 bits)	Non-zero bits of the coefficient
α	-1.5861343420693648	-1.10010110000011	7
β	-0.0529801185718856	-0.00000110110010	5
γ	0.8829110755411875	0.11100010000010	6
δ	0.4435068520511142	0.01110001100001	6
κ	1.1496043988602418	1.00100110010011	7
$1/\kappa$	0.8698644516247808	0.11011110101100	9

다. 부호를 갖는 입력 x 는 첫 번째 리프팅 계수 곱셈 이후에 이어지는 덧셈 연산과 다음 단의 계수 연산들의 편이를 위해 2의 보수 형식을 사용한다. 리프팅 계수는 계수의 절대 값 크기만 이진수로 표현하여 shift-add 연산의 횟수를 구하며, 음의 계수들의 경우는 negation 연산을 추가하여 결과 값의 부호를 계산을 한다. $x \ll k$ 는 입력 x 의 k -bit left shift 연산을 의미한다. 리프팅 계수 α 에 pattern search 방식을 적용하면, 표 1의 7회의 add 연산이 아래의 식 (3), (4), (5)와 같이 세 번의 add와 식(6)의 한 번의 sub 연산으로 줄어든다. 마지막으로 음의 계수이기

$$t1 = (x \ll 1) + x \tag{3}$$

$$t2 = (t1 \ll 4) + t1 \tag{4}$$

$$t3 = (t2 \ll 9) + t1 \tag{5}$$

$$t4 = (x \ll 7) - t3 \tag{6}$$

$$t5 = t \times -1 \tag{7}$$

때문에 식 (7)의 negation 연산을 수행한다. 패턴 "11"이 반복적으로 3회 사용됨을 알 수 있다. 리프팅 계수 β 의 패턴을 찾는 과정을 자세히 살펴보면, β (-0.00000110110010)의 이진수 표현은 오른쪽에서 5, 6 번째의 1들을 CSD 형식으로 recoding하여 000000111010010 으로 나타낸다. 다시 오른쪽에서 7, 8, 9 번째 1들을 CSD 형식으로 recoding하면, 000001001010010의 이진수 표현을 구할 수 있다. 여기서 패턴 1001은 negative 버전인 1001을 포함하여

두 번 반복된다. 따라서 다섯 개의 덧셈기로 구현되는 β 의 곱셈은 2개의 뺄셈기로 구현할 수 있다. 이와 같이 Lefèvre의 알고리즘은 negative한 패턴을 사용하기 때문에, 보다 나은 최적화된 결과를 얻을 수 있다. 그림 2는 입력이 16비트일 때, 6개의 리프팅 계수에 Lefèvre의 알고리즘을 적용하여 최적화된 결과를 보여준다. 표 2는 리프팅 계수들의 곱셈 연산에 이진수 표현, CSD 형식, Lefèvre의 알고리즘을 적용할 경우의 adder cost를 나타낸다. 이진수 표현 방법에서 총 40개의 shift-add 연산이 Lefèvre의 알고리즘을 적용하여 최적화하면 총 20 개로 줄어드는 것을 알 수 있다. 따라서 리프팅 계수의 곱셈연산을 하드웨어 구현 시에 면적을 줄일 수 있을 뿐 아니라, 그림 2에서 보이는 바와 같이 adder tree의 depth가 모두 4 이하이기 때문에 연산 지연과 전력소모를 크게 줄일 수 있다.

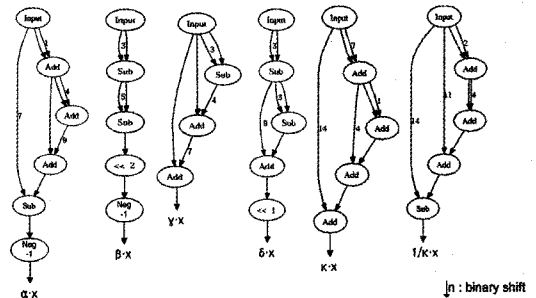


그림 2. 리프팅 계수들에 제안 방식을 적용한 결과

표 2. 리프팅 계수에 알고리즘들을 적용한 경우의 총 Adder costs

Word length	Coefficient type	Total adder costs		
		binary	CSD	Lefèvre
15	Signed magnitude	40	35	20

3. 실험 결과

본 논문은 CDF 9/7 웨이블릿 리프팅 필터에 Lefèvre의 알고리즘을 적용하여 수학적으로 최적화한 후 제안한 구조를 하드웨어로 구현하였다. 하드웨어 설계와 시뮬레이션은 Synopsis사의 Design Compiler와 Mentor사의 Modelsim을 사용하였다. 비교 실험을 위해 제안한 구조뿐 아니라, 기존의 다른 논문들에서 주된 방법으로 사용되었던 범용 곱셈기 기반의 구조도 Verilog HDL을 이용하여 구현하였다. 구현의 단순함을 위해, 필터의 입력은 정수부 14비트와 소수부 2비트를 할당하여 16비트의 2의 보수 형식을 사용하였다. 리프팅 웨이블릿 계수들은 부호-절대값 방식으로 크기 값에 1 비트와 소수부에 14비트를 할당하고 Verilog HDL을 이용하여 하드웨어 구현하였다. 이와 같은 입력과 계수들의 비트 설정을 바탕으로, 정지 영상 압축에서 널리 사용되는 512 × 512 화소 크기를 갖는 Lena, Tiffany, Peppers, Goldhill 그리고 Airplane 의 5개의 스탠다드 흑백 영상들을 이용하여 4 level의 웨이블릿 변환에 대한 실험을 하였다. 하드웨어로 구현한 출력 결과를 MATLAB에서 역 웨이블릿 변환하여 복원한 영상들로부터 표 3과 같이 약 47 dB 이상의 충분한 PSNR 결과를 얻었다. 본 논문에서는 리프팅 과정에서 overflow에 의한 정보의 손실을 없애기 위해, 필터의 중간결과들의 신호 선에 충분한 precision을 주어 주의 깊게 합성하였다. 고정 소수점 방식의 2의 보수 형식의 덧셈기를 사용하였다. 여기서 사용한 각각의 하드웨어 덧셈기는 round-off 나 overflow가 일어나지 않도록 정수부와 소수부의 비트 수를 적절히 선택

하였다. 그림 2와 같이 필터 계수들의 곱셈 연산은 부분 곱을 더하기 위해 adder tree의 구조이다. 이 회로들이 빠른 지연 시간을 갖도록 하기 위해 wallace tree 구조의 고속 adder를 이용하여 구현하였다[8]. 기존의 웨이블릿 필터 구조와 함께 제안한 하드웨어 구조는 Modelsim을 통한 기능 검증 후, 디지털 0.18um CMOS 공정을 이용하여 Design compiler를 이용하여 합성하고 비교 실험 하였다. 두 구조는 16비트 입력과 15비트의 필터 계수를 실험 조건을 설정하고 스탠다드 셀을 이용하여 합성하였다. 표 4는 100MHz와 200MHz를 타겟 클럭 주파수로 하여 합성한 결과를 나타낸다. 100MHz를 타겟 주파수로 한 경우에 제안한 구조는 기존의 구조에 비해 합성 면적이 약 64%, 전력 소모가 약 43%로 감소하였다. 또한 200MHz의 타겟 주파수 합성한 경우에는, 합성 면적과 소비 전력이 각각 약 51%와 43%로 감소하였다. 기존 구조에서는 범용 곱셈기를 사용하기 때문에 지연 시간의 개선이 크게 이루어지지 않는 반면, 제안한 구조는 최대 지연 시간도 약 30% 감소하여 보다 빠른 성능을 보였다.

4. 결론

본 논문은 하드웨어 곱셈 연산을 최적화 하여 리프팅 기반의 9/7 웨이블릿 필터의 개선된 VLSI의 구조를 제안하였다. 제안한 구조는 범용 곱셈기를 사용하는 기존의 리프팅 기반의 웨이블릿 필터와 비교하여 화질 열화 없이 보다 적은 로직과 전력소모를 갖는다. 본 논문은 Pattern search 기반의 Lefèvre 알고리즘을 이용하여 하드웨어 구조를 개선하였다. 제안

표 3. 제안한 방식을 웨이블릿 변환에 적용하여 복원한 영상들의 PSNR 실험결과

	Lena	Tiffany	Peppers	Goldhill	Airplane
PSNR(dB)	48.59	49.44	47.03	47.21	48.46

표 4. 제안 방식과 기존 방식과의 합성 결과 비교

	Target Clock Frequency					
	100MHz Clock			200MHz Clock		
	기존 구조	제안 구조	감소율	기존 구조	제안 구조	감소율
합성 면적(um ²)	217,573	77,638	64.32%	239,923	115,389	51.91%
최대 지연(ns)	9.69	9.69	-	9.38	6.55	30.17%
소비전력(mW)	20.34	11.55	43.22%	47.20	26.70	43.43%

한 구조는 기존의 방식과 같은 성능을 가지면서 범용의 곱셈기를 단순한 shift-add 연산으로 대체하여 하드웨어 구현을 단순하게 하고 계산 속도를 빠르게 하였다. 본 논문에서는 제안한 구조와 범용의 곱셈기 사용하는 기존의 구조를 Verilog HDL을 이용하여 구현하고 비교 실험하였다. 두 구조는 0.18um 디지털 CMOS 공정의 스탠다드 셀을 이용하여 합성되었다. 제안한 구조는 200MHz의 합성 타겟 주파수에서 기존의 구조에 비해 면적, 전력소모와 최대 지연시간이 각각 약 51%, 43%와 30%로 감소하였다. 본 논문의 구현 결과는 제안한 리프팅 기반의 웨이블릿 필터 구조가 범용의 곱셈기 블록을 사용한 기존의 구조보다 스탠다드 셀을 이용한 ASIC 구현에 보다 적합하다는 것을 보여준다.

참 고 문 헌

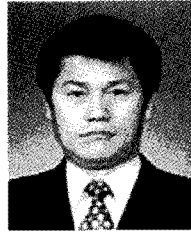
- [1] M. Potkonjak, M. B. Srivastava, and A. Chandrakasan, "Multiple constant multiplications: efficient and versatile framework and algorithms for exploring common sub-expression elimination," *IEEE Trans. Computer-Aided Design*, Vol.15, No.2, pp. 151-165, 1996.
- [2] N. Boullis and A. Tisserand, "Some optimizations of hardware multiplication by constant matrices," *IEEE Transactions on Computers*, Vol.54, No.10, pp. 1271-1282, 2005.
- [3] R. Bernstein, "Multiplication by Integer Constants," *Software-Practice and Experience*, Vol.16, No.7, pp. 641-652, 1986.
- [4] H.J. Kang and I.C. Park, "FIR Filter Synthesis Algorithms for Minimizing the Delay and the Number of Adders," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol.48, No.8, pp. 770-777, 2001.
- [5] V. Lefèvre, "Multiplication by an integer constant," INRIA, Research Report 4192, May 2001.
- [6] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, Vol.3, pp. 186-200, 1996.
- [7] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal Fourier Anal. Applicat.*, Vol. 4, 1998.
- [8] T. Kim, W. Jao, and S. Tjiang, "Arithmetic Optimization using Carry-save-Adders", DAC, pp. 433-438, 1998.



손 창 훈

2000년 2월~2005년 2월 전남대학교 정보통신공학부 공학사
 2005년 3월~2007년 2월 전남대학교 전자컴퓨터공학과 공학석사
 2007년 3월~현재 전남대학교 전자컴퓨터공학과 박사과정

관심분야: VLSI 시스템 설계, 신호처리용 ASIC 설계, 영상 압축 등



김 영 민

1976년 서울대학교 전자공학과 학사
 1978년 한국과학기술원 전기 및 전자공학과 석사
 1986년 오하이오 주립대학교 전기공학과 박사
 1978년~1979년 한국선박해양연구소 주임연구원

1979년~1982년 국방과학연구소 연구원
 1988년~1991년 한국전자통신연구원 실장
 1991년~현재 전남대학교 전자공학과 교수
 관심분야: 영상압축, VLSI 설계, RF 회로 설계, 신경회로망 등



박 성 모

1977년 서울대학교 전자공학과 학사
 1979년 한국과학기술원 전기 및 전자공학과 석사
 1997년 노스캐롤라이나 주립대학 전기 및 컴퓨터공학과 공학박사

1979년~1984년 한국전자기술연구소 설계개발부 선임연구원
 1988년~1992년 올드도미니언대학교 전기 및 컴퓨터공학과 조교수
 1992년~현재 전남대학교 컴퓨터공학과 교수
 관심분야: 멀티미디어 프로세서 구조, VLSI 시스템 설계, 신호처리용 ASIC 설계, 영상압축, 임베디드 시스템 등