

# 스위칭 액티비티를 최소화한 저전력 DCT 아키텍처 구현

정희원 김 산\*, 박종수\*\*, 이용주\*\*, 이용석\*\*

## Design and Implementation of Low-Power DCT Architecture by Minimizing Switching Activity

San Kim\*, Jongsu Park\*\*, Yong-joo Lee\*\*, Yong-Surk Lee\*\* *Regular Members*

### 요약

저전력 설계는 시스템의 소모전력을 줄임으로써 에너지 절약과 함께 휴대용 장치의 배터리 수명을 극대화시킴에 있어 직면한 가장 중요한 문제이다. 본 논문에서는 개량형 CSHM을 이용하여 저전력 DCT 구조를 제안하였다. 제안된 구조는 Computation Sharing Multiplication 연산 과정 중 불필요한 비트에 대한 연산을 수행하지 않는다. 실험 결과, 기존의 DCT 알고리즘과 동일한 연산 결과를 보이면서도 최대 약 9%의 소모전력이 감소하였다. 따라서 제안된 저전력 DCT 구조는 저전력 및 고성능으로 DCT 알고리즘을 처리해야하는 휴대용 멀티미디어 시스템에 적용이 가능하다.

**Key Words** : DCT, CSHM, Low-Power, Multiplier

### ABSTRACT

Low-power design is one of the most important challenges encountered in maximizing battery life in portable devices as well as saving energy during system operation. In this paper we propose a low-power DCT (Discrete Cosine Transform) architecture using a modified Computation Sharing Multiplication (CSHM). The overall rate of power consumption is reduced during DCT: the proposed architecture does not perform arithmetic operations on unnecessary bits during the Computation Sharing Multiplication calculations. Experimental results show that it is possible to reduce power dissipation up to about 7~8% without compromising the final DCT results. The proposed low-power DCT architecture can be applied to consumer electronics as well as portable multimedia systems requiring high throughput and low-power.

### I. 서론

오늘날 멀티미디어 이동통신 장비의 핵심 부품들은 저가격화, 소형화, 성능 최적화등과 함께 저전력화의 필요성이 중요시되고 있다. 특히 멀티미디어 이동통신 장비의 장시간 휴대 동작을 위한 소모전력의 최소화로 배터리의 사용시간을 최대화할 필요가 있다. 이에 따라 저전력 DCT 응용들이 휴대용 멀티미디어 장치로까지 확장되고 있고, 저전력 설계

에 칩들이 개발되고 있는 추세이다.

얼마 전까지 SoC(System on a Chip)를 위한 기능의 구현과 고속으로 동작 가능한 IC를 구현하는 것이 관심의 초점이었기 때문에 칩의 소모전력은 심각한 문제가 아니었다. 그러나 휴대용 전자기기에 대한 수요가 늘어나고 한정된 배터리 사용시간 동안 오래 동작할 수 있는 IC의 수요가 증가함에 따라 저전력 설계기술을 사용한 DCT 아키텍처의 개발이 시급하게 되었다.

\*본 연구는 한국과학기술원 특정기초연구(R01-2006-000-10156-0)지원으로 수행되었음.

\* 연세대학교 전기전자공학과 프로세서연구실 (jspark@dubiki.yonsei.ac.kr)

논문번호: KICS2005-06-255, 접수일자: 2005년 6월 27일, 최종논문접수일자: 2006년 5월 19일

DCT에 관하여 부연하자면, 이것은 연산측면의 강도 높은 트랜스폼으로써 많은 곱셈과 덧셈을 필요로 하기 때문에 저전력 DCT를 개발하기 위하여 다수의 DCT 알고리즘들이 제안 되어오고 있다. 곱셈기를 이용한 구조는 그것의 복잡한 연결 관계 때문에 일반적이지 못한 VLSI 구조를 가지며, 많은 하드웨어 면적을 필요로 한다. 반면 분산연산 DA(Distributed Arithmetic) 알고리즘[2]에 기초한 DCT 구조는 VLSI 구현에 적당한 규칙성과 모듈로 만들 수 있는 성질로 이어진다. DA 알고리즘은 곱셈연산을 위해 look-up 테이블을 사용하기 때문에 곱셈 연산이 필요하지 않다는 장점이 있다. 그러나 입력의 숫자가 증가하고 내부 precision이 증가함에 따라 DA는 커다란 크기의 ROM을 필요로 한다. 따라서 하드웨어의 복잡도를 크게 증가시킨다. 더욱이 DA가 비트 순차형 연산이기 때문에 우수한 성능을 위해서 그것은 unfolding[3], pipelining[4]을 필요로 한다. 따라서 전력은 면적 및 성능과 동시에 고려해야 하므로 더욱 복잡한 최적화의 문제가 발생되며, 그것을 해결하기 위해서 새로운 알고리즘 및 아키텍처의 개발이 절실히 요구되는 것이다.

저전력 설계를 위한 알고리즘/아키텍처는 결과값의 손상없이 설계의 전 과정을 통해 소모전력 감소를 위한 노력이 필요하다. 특히 알고리즘 및 아키텍처 변환을 통한 전력 감축 효과는 레이아웃 최적화 및 device scaling을 통한 전력 감축 효과와 마찬가지로 영향이 가장 클 것으로 기대된다. 따라서 저전력 설계[5] 알고리즘 및 아키텍처는 3년에 1/3의 전력감축, 5년에 1/5의 전력 감축의 목표를 달성하기 위한 필수적인 요건이 되고 있다.

본 논문에서는 스위칭 액티비티를 줄인 저전력화를 위한 연구이므로 스위칭 액티비티를 간략히 살펴보면 다음과 같다.

CMOS 회로에서의 소모전력은 식 (1)과 같이 보다 적은 수의 스위칭 액티비티가 발생하도록 통해 절약될 수 있다.

$$P_{switching} = a \cdot C \cdot V_{dd}^2 \cdot f_{clk} \quad (1)$$

$a$ 는 스위칭 액티비티 변수,  $C$ 는 loading capacitor,  $V_{dd}$ 는 공급전압 그리고  $F$ 는 동작 주파수를 나타낸다. 이 중에서 알고리즘 수준에서 감소시킬 수 있는 유일한 변수는 스위칭 액티비티이다. 따라서 알고리즘 수준에서 연산 중 스위칭 액티비티를 최소화 시키는 과정이 알고리즘을 하드웨어로 구현하

는 과정보다 먼저 고려되어야한다<sup>[6]</sup>.

본 논문은 CSHM(Computational SHaring Multiplication)에 기초한 저전력 DCT 아키텍처를 제안한다. 제안된 아키텍처는 DCT 연산결과와 왜곡없이 CSHM을 사용한 DCT 구조에서의 중복되는 불필요한 연산을 skip함으로써 소모전력을 줄였다. 소모전력을 측정하기 위하여 Verilog-HDL을 사용하여 하향식 설계 방식으로 제안된 구조를 설계하였고, 하이닉스 0.25um CMOS standard cell 라이브러리를 이용하여 합성을 하였으며, net-list를 추출하여 gate-level 시뮬레이션을 수행하였다. 또한, 파워 컴파일러를 통하여 제안되어진 DCT의 전력 소모를 측정하였다. 실험결과, 제안된 구조의 소모전력이 기존의 구조보다 최대 약 9%의 소모전력이 감소하였다. 이 결과는 비교대상인 기존의 CSHM을 적용한 DCT 구조가 월레스트리(WTM) 알고리즘을 적용한 DCT 구조보다 약 20%의 소모전력을 감소시킨 것이라는 점에서 더욱 의미가 있다고 할 수 있다. 따라서, 본 논문에서 제안된 저전력 DCT 구조가 멀티미디어 및 통신시스템의 성능향상에 큰 영향을 줄 수 있는 구조라고 본다.

본 논문의 2장과 3장에서는 CSHM 알고리즘 및 구조와 CSHM 알고리즘을 적용한 DCT 구조에 대하여 설명하였다. 4장에서는 CSHM의 구현, 5장에서는 제안된 저전력 DCT 알고리즘 및 구조를 논한다. 6장에서는 실험을 통한 소모전력의 측정 및 결과를 평가하며, 마지막으로 7장에서 결론을 맺는다.

## II. CSHM 알고리즘 및 구조

벡터 스케일링 연산[18, 19]은 스칼라와 벡터들의 곱셈연산이다. 예를 들어 계수벡터  $C=[C_0, C_1, \dots, C_{M-1}]$ 과 스칼라  $x(n)$ 의 곱셈이 벡터 스케일링 연산으로 DSP(Digital Signal Processing)의 필터링과 행렬곱셈에 흔히 사용되어진다.

벡터 스케일링 연산에서, 적은 수의 연속된 비트를 신중히 선택하면 동일한 곱셈결과를 덧셈과 시프트(shift) 연산만으로 얻을 수 있다. 예를 들어 간단한 벡터 스케일링 연산  $[c_0, c_1] \cdot x$ ,  $c_0 = 01100111$ ,  $c_1 = 10001011$  는 다음과 같이 구성되어 질 수 있다.

$$C_0 \cdot x = 2^5 \cdot (0011) \cdot x + (0111) \cdot x \quad (2)$$

$$C_1 \cdot x = 2^7 \cdot (0001) \cdot x + (1011) \cdot x \quad (3)$$

만약,  $x$ ,  $(0001) \cdot x$ ,  $(0111) \cdot x$ ,  $(1001) \cdot x$ 가 이용

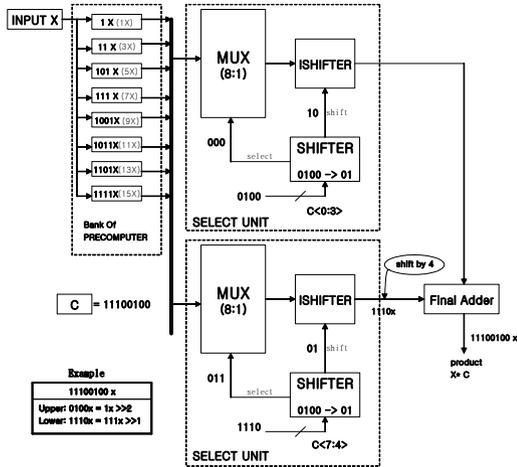


그림 1. CSHM의 구조

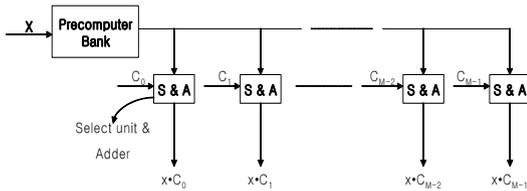


그림 2. 벡터 스케일러와 벡터 스케일링 연산

되어질시 전체적인 연산이 덧셈과 시프트로 줄어든다. 여기서 선택된 이런 비트들을 알파벳(alphabet)이라고 한다. 알파벳 집합이란 벡터 C의 계수에 해당하는 모든 알파벳을 모아놓은 집합이다. 예를 들어 알파벳 집합이란 {0001, 0011, 0101, 0111, 1001, 1011, 1101, 1111}이다.

그림 1은 CSHM의 구조를 보여준다. CSHM은 위에서 설명된 벡터 스케일링 연산을 기반으로 하였다. CSHM은 precomputer, select unit, final adder의 세부 블록으로 구성된다. Precomputer는 X를 입력으로 받아 알파벳을 가지고 곱셈을 수행한다. 알파벳은 위에서 설명한바와 같이 작은 비트순서로 결과값의 손상 없이 곱셈을 수행한다.

Select unit과 final adder는 곱셈결과를 얻기 위해 필요한 합당한 select/shift, 덧셈연산을 수행한다. 모든 가능한 상수를 포함시키고 일반적인 곱셈 연산을 수행하기 위해서 여기서는 {1, 3, 5, 7, 9, 11, 13, 15} 8개의 알파벳을 이용하였다. 그러나 C 벡터의 상수에 따라 CSHM의 알파벳의 숫자는 바뀔 수 있다. 위의 vector scaling 예제에서는 {1,3,7,11} 4개의 알파벳만이 필요하였다.

그림 2는 CSHM을 이용한 벡터 스케일러의 구

조를 나타낸다. 벡터 C의 모든 상수들에 대하여 precomputer연산은 한번만 수행되었으며, precomputer의 결과가 모든 select unit과 adder에 의해 공유되었다. 결과적으로 벡터 스케일링 연산은 덧셈과 시프트 연산으로 간단해졌으며, 이는 향상된 성능과 저전력 구조로 이어진다.

위에서 언급된 것과 같이 알파벳의 수는 벡터 C의 상수에 따라 결정된다. 상수를 약간 수정함으로써 알파벳의 수를 줄일 수 있다. 위의 예제에서 벡터 C의 상수들이  $[c_0, c_1]$ ,  $c_0=01101000$ ,  $c_1=1001100$ 으로 바뀌면 곱셈은 식 (4)와 (5)로 나타낼 수 있다.

$$C_0 \cdot x = 2^5 \cdot (0011) \cdot x + 2^3 \cdot (0001) \cdot x \quad (4)$$

$$C_1 \cdot x = 2^7 \cdot (0001) \cdot x + 2^2 \cdot (0011) \cdot x \quad (5)$$

알파벳 셋은 {0001, 0011}으로 바뀐다. 그 결과 단지 두 개의 precomputer bank와 간단한 select unit만이 필요하다. 이를 통하여 적당한 상수 수정을 이용하여 복잡도가 낮은 벡터 스케일러를 얻을 수 있다. 그러나, 미세한 계수조정으로 인하여 화질의 질이 상당히 미세하게 떨어질 수 있기 때문에 본 논문에서는 8개의 precomputer bank를 사용하였다.

### III. CSHM을 적용한 DCT 구조

DCT를 기반으로 한 시스템의 기본연산은  $8 \times 8$  이미지 블록을 공간좌표에서 DCT 좌표로 바꾸는 것이다. 1-D DCT 변환은 다음과 같다.

$$z_k = \frac{c(k)}{2} \sum_{i=0}^7 x_i \cos \frac{(2i+1)k\pi}{16} \quad (6)$$

$$k = 0, 1, 2, 3, 4, 5, 6, 7$$

$$c_k = 1/2 \quad \text{if } k=0 \\ = 1 \quad \text{otherwise.}$$

식 (6)은 벡터-행렬형태로 다음과 같이 나타낸다. 단, T는  $8 \times 8$  행렬이며 성분이 cosine 함수이다. x와 z는 각각 행과 열의 벡터이다. 본 논문에서 적용한 8-bit DCT 계수는 표 1과 같다.

$8 \times 8$ 상수 행렬 T가 대칭이기 때문에 1-D DCT 행렬은 식 (7), (8)과 같이 그 순서를 바꿀 수 있다.

$$\begin{bmatrix} z_0 \\ z_2 \\ z_4 \\ z_6 \end{bmatrix} = \begin{bmatrix} d & d & d & d \\ b & f & -f & -b \\ d & -d & -d & d \\ f & -b & b & -f \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix} \quad (7)$$

표 2. DCT 계수와 알파벳 값

| 계수 | 값    | 바이너리 값    | 알파벳·x   |
|----|------|-----------|---------|
| a  | 0.49 | 0011 1111 | 3x, 15x |
| b  | 0.46 | 0011 1011 | 3x, 11x |
| c  | 0.42 | 0011 0101 | 3x, 5x  |
| d  | 0.35 | 0010 1101 | 1x, 13x |
| e  | 0.28 | 0010 0100 | 1x      |
| f  | 0.19 | 0001 1000 | 1x      |
| g  | 0.10 | 0000 1100 | 3x      |

$$\begin{bmatrix} z_1 \\ z_3 \\ z_5 \\ z_7 \end{bmatrix} = \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \quad (8)$$

단,  $C_k = \cos n\pi/16$ ,  $a=c1$ ,  $b=c2$ ,  $c=c3$ ,  $d=c4$ ,  $e=c5$ ,  $f=c6$ ,  $g=c7$ 이다. 식 (7)은 쉽게 식 (9)와 같이 바꿀 수 있다.

$$\begin{bmatrix} z_1 \\ z_3 \\ z_5 \\ z_7 \end{bmatrix} = (x_0 + x_7) \begin{bmatrix} d \\ b \\ d \\ f \end{bmatrix} + (x_1 + x_6) \begin{bmatrix} d \\ f \\ -f \\ -b \end{bmatrix} + (x_2 + x_5) \begin{bmatrix} d \\ -d \\ -d \\ d \end{bmatrix} + (x_3 + x_4) \begin{bmatrix} f \\ b \\ -f \end{bmatrix} \quad (9)$$

식 (8)도 비슷한 방식으로 표현될 수 있다. 식 (9)에서와 같이  $8 \times 8$  DCT 행렬곱셈은 벡터 스케일링연산의 덧셈으로 표현되므로 이는 DCT 구현에서 CSHM의 이용을 가능하게 한다.

### IV. CSHM의 구현

#### 4.1 Precomputer 블록 설계

본 절에서는 CSHM의 Precomputer 블록의 설계에 대해 논한다. precomputer 블록을 설계할 때, 하드웨어 면적을 줄이기 위하여 하이브리드 구조의 리플 캐리 덧셈기(Ripple Carry Adder)를 기반으로 하는 캐리 선택 덧셈기(Carry Select Adder)를 사용하였다.

##### 4.1.1 리플 캐리 덧셈기(Ripple Carry Adder)

리플 캐리 덧셈기는 가장 기본적인 방법으로 전가산기(full adder)를 직렬적으로 연결하여 구성하는 것이다. 전가산기는 세개의 입력 ( $X_i, Y_i, C_i$ (Carry-in))과 두 개의 출력( $S_i, C_{i+1}$ (Carry-out))을 가진다.

$$S_i = X_i \oplus Y_i \oplus C_i \quad (10)$$

$$C_{i+1} = X_i \cdot Y_i + Y_i \cdot C_i + C_i \cdot X_i \quad (11)$$

이 덧셈기는 쉽게 구현이 가능하고, 면적이 작은 장점을 가지지만, 결과를 연산하는데 필요한 시간이 선형적으로 증가하므로, 고성능 하드웨어에 이용할 수 없다. 리플 캐리 덧셈기는 전가산기를 직렬 연결하여 구현을 하였다. 그림 3은 1비트 전가산기와 4비트 리플 캐리 덧셈기의 구조를 보여준다.

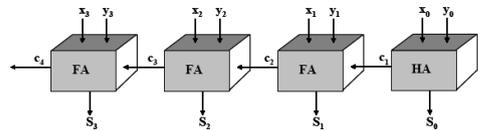
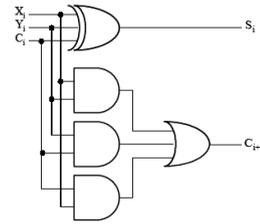


그림 3. 리플 캐리 덧셈기의 구조

##### 4.1.2 하이브리드 구조의 캐리 선택 덧셈기

캐리 선택 덧셈기는 입력되는 캐리의 값이 '0' 또는 '1'의 값을 가지므로 이 두 가지 경우에 대해 덧셈을 미리 수행한다. 그리고 최하위 블록의 캐리 출력에 결정되면, 이 결과를 이용하여 미리 계산한 결과들 중의 하나를 선택한다. 캐리 선택 덧셈기는 병렬적으로 연산을 하기 때문에, 리플 캐리 덧셈기보다 약 2배의 면적을 사용 하지만, 약 4배의 속도를 향상시킬 수 있다<sup>[20]</sup>. 캐리 선택 덧셈기는 멀티플렉서를 통과하는 시간을 고려하여 마지막 그룹으로 갈수록 많은 비트를 할당할 수 있다. 그러나 본 논문에서는 그룹별 비트가 다를 경우 너무 많은 경우의 수가 발생할 수 있으므로 각 그룹별 비트수를 동일하게 처리하였다. 그림 4는 8비트 캐리 선택 덧셈기의 구조이다.

하이브리드 구조 덧셈기는 소그룹의 캐리 출력을 캐리 전달 유닛으로 연결하여 연산한 후 캐리 전달 유닛의 출력을 다음 소그룹의 캐리 입력으로 연결하는 구조이다. 본 논문에서는 소그룹으로 저면적화를 위해 4bit 리플 캐리 덧셈기 구조를 채택하였다. 캐리 전달 유닛은 모든 부분이 리플 캐리 덧셈기로 되어 있는 경우보다 약 4배의 속도를 향상을 가져오

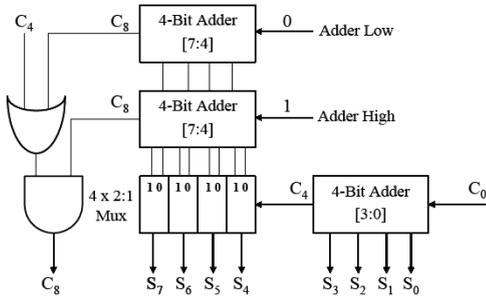


그림 4. 8비트 캐리 선택 댕셈기의 구조

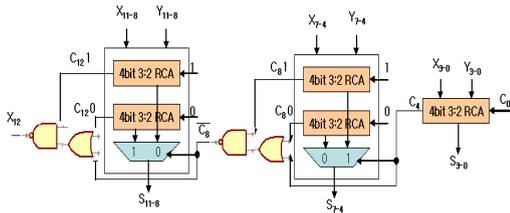


그림 5. 12비트 하이브리드 캐리 선택 댕셈기의 구조

는 캐리 선택 댕셈기로 구현하였다.

소그룹은 4, 8, 12비트로 나누어서 연산을 할 수 있다. 그림 5는 본 논문에서 적용한 하이브리드 구조의 6개 소그룹으로 나뉘어진 12bit 캐리 선택 댕셈기의 구조이다.

### 4.2 Precomputer Bank 구현

본 논문에서는 8비트 DCT 계수와 8비트 입력의 곱셈에 있어서 CSHM 알고리즘을 적용하여 불필요한 중복 연산을 제거하였고 곱셈을 댕셈기만으로 연산하였다. 구현한 precomputer bank는 CSHM에 있어 핵심 연산프로세서로서 곱셈기를 사용하지 않고 댕셈기만을 사용하여 DCT를 효과적으로 수행하는 저전력 연산기 구조이다. 또한 고속처리가 가능하면서도 구현 하드웨어의 크기를 최소화한다.

댕셈기만을 사용하여 곱셈 연산을 수행하기 때문에 곱셈기의 수를 줄여서 저전력화가 가능하고, 하드웨어의 복잡도도 떨어뜨린다. 또한 앞 절에서 설명한 바와 같이 CSHM 알고리즘에서는 precomputer 단에서 인접한 비트들의 8개의 알파벳과 입력값의 곱셈을 하기 때문에 실제적으로 공유곱셈이 가능하다. 왜냐하면 원래 모든 곱셈의 경우에는 15개의 알파벳을

필요로 하지만 간단한 shifter를 이용하여 설계하면 8개의 알파벳으로 결과 값의 변화 없이 동일한 연산 값을 얻을 수 있어서 결과적으로 8개의 알파

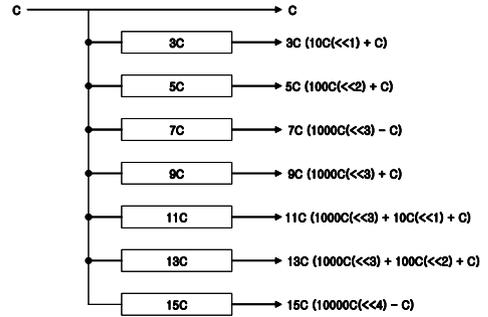
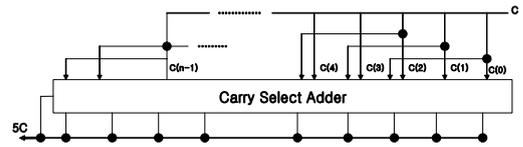
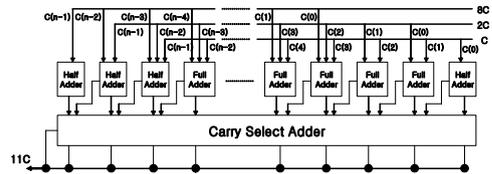


그림 6. Precomputer bank의 구조



(a) 5c



(b) 11c

그림 7. Precomputer bank의 구현

벳으로 간소화하여 이를 공유해서 사용하기 때문인 것이다. Precomputer bank의 구조는 그림 6과 같다. 그림 7은 precomputer단 내부의 연산과정을 나타낸다. 아래의 5C, 11C의 곱셈과정의 구조이다. 알파벳이 0101(5), 1011(11)이고 계수가 C일 때 계수와 알파벳의 곱은 (식 12), (식 13)과 같다.

$$5c(0101c) = 100c(\ll 2) + 1c \quad (12)$$

$$11c = 8c(1000c) + 2c(10c) + c \quad (13)$$

### 4.3 제어부 설계

제어부는 지금까지 설명한 precomputer bank단의 곱셈 (1x, 3x, 5x, 7x, 9x, 11x, 13x, 15x) 중 1개를 선택하게 하는 선택 제어신호 발생기 및 shift 제어신호 발생기로 구성되어 있다. 즉 8:1 MUX에 제어신호를 가해 precomputer bank단의 값 1개를 선택한 후 IShifter 블록에서 shift를 하여 곱셈 결과의 자리 수를 맞춘다.

제어부는 그림 8에서와 같이 Select Unit으로 나타내었고 제어 신호 발생기는 Shifter이다.

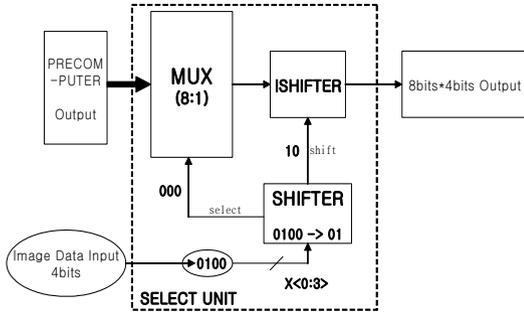


그림 8. 제어부의 구조

표 2. Select 및 shift 제어신호

| Shifter input | Select signal | Shift signal |
|---------------|---------------|--------------|
| 0000          | 1ddd          | dd           |
| 0001          | 0000          | 00           |
| 0010          | 0000          | 01           |
| 0011          | 0001          | 00           |
| 0100          | 0000          | 10           |
| 0101          | 0010          | 00           |
| 0110          | 0001          | 01           |
| 0111          | 0011          | 00           |
| 1000          | 0000          | 11           |
| 1001          | 0100          | 00           |
| 1010          | 0010          | 01           |
| 1011          | 0101          | 00           |
| 1100          | 0001          | 10           |
| 1101          | 0110          | 00           |
| 1110          | 0011          | 01           |
| 1111          | 0111          | 00           |

표 2는 Shifter에서 선택 제어 신호와 shift 제어 신호를 보여준다. 여러 다른 이미지 데이터의 입력 중에 0000이란 것이 들어올 때 선택 및 시프트 제어 신호가 don't care 조건으로 되고 IShifter에서 0을 출력으로 보내도록 설계하였다.

#### 4.4 Final Adder

앞서 설명한바와 같이 final adder는 select unit1, select unit2에서 나온 결과를 최종적으로 더해  $X \cdot C$  를 구하는 것이다. 여기서 주의할 점은 select unit1의 결과와 select unit2를 더할 때 select unit2의 결과를 오른쪽으로 4비트 left shift를 해서 최종 덧셈기로 보내야한다. 왜냐하면, 이미지 데이터 입력 4LSB와 계수와의 곱이 select unit1의 결과이고 이미지 데이터 입력 4MSB와 계수와의 곱이 select unit2이기 때문에 select unit2의 결과를 4비트 left shift해서 최종적으로 덧셈해야 되기 때문이다. 최종 덧셈기는 RSA(Ripple Carry Adder)로 간단히 설계하였다.

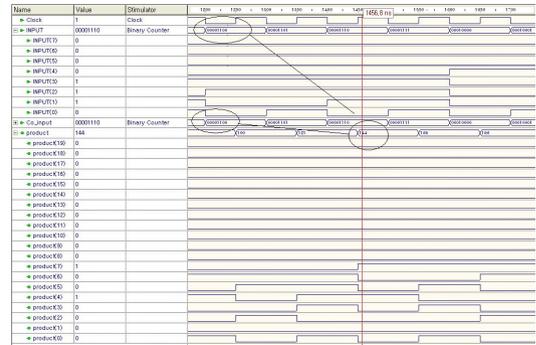


그림 9. CSHM 연산 결과 (12×12)

그림 9는 설계된 기존의 CSHM 연산기에 10만 개의 테스트벤치를 적용하여 시뮬레이션 한 것이다. 그림 9에서는 Input 12와 Coefficient 12와의 곱셈 결과를 표시하였다.

### V. 제안한 저전력 DCT 구조

제안한 저전력 DCT 구조는 그림 1의 기존 CSHM 구조를 제안한 알고리즘에 맞게 수정하였다. 수정된 구조는 그림 12에 나타내었다.

우선 제안한 알고리즘의 목표는 DCT 연산 수행 시 precomputer bank단과 select unit단의 연산시 스위칭 액티비티를 감소시켜 화질의 손상 없이 소비전력을 줄이는 것이다. 따라서, 계수나 영상 데이터 입력값의 변화 없이 DCT 알고리즘에 적용했다. 적용한 결과 소비전력을 줄 일수 있었다.

제안한 알고리즘은  $8 \times 8$  블록내의 인접한 픽셀의 값이 비슷하다면 precomputer bank단과 select unit에 들어가는 영상 신호 8비트 입력이 4MSB 와 4LSB 각각에 대해 같은 값의 입력으로 들어갈 가능성이 커진다. 따라서, 4비트 2그룹은 간단한 비교기에 의해서 이전 입력값과 이후 입력 값이 비교된다. 4MSB와 4LSB 각각의 입력값을 비교하여 이전 입력 값과 동일한 경우 연산을 생략 하게 된다. 예를 들어, 8비트 영상 데이터 01110010과 01110011 이 입력으로 들어갈 때에 상위 4비트가 동일하다. 따라서 단지 4LSB 비트만이 곱셈 되고 4MSB비트는 생략된다.

$$\begin{aligned}
 X_i &= 01110010 \\
 X_{i+1} &= 01110011
 \end{aligned}
 \tag{14}$$

식 (14)에서  $X_{i+1}$ 의 상위 4비트와 전 입력  $X_i$ 의 상위 4비트가 동일하다. 따라서, 동일한 상위 4비

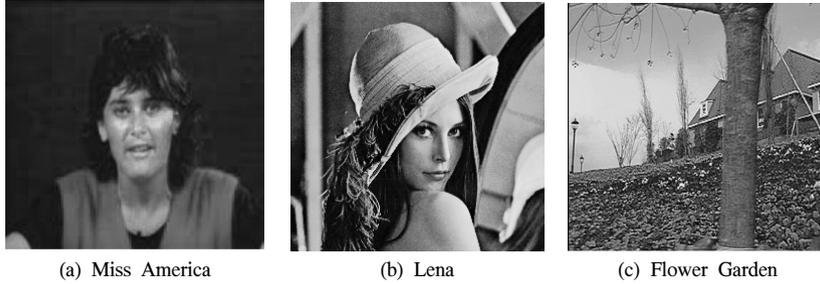


그림 10. 실험영상

트는 연산을 할 필요없이 생각하고 이전의 결과 값을 재사용하면 precomputer bank와 select unit의 연산을 안 해도 되기 때문에 스위칭 파워를 줄일 수 있다.

본 논문에서는 QCIF 파일 영상 3개를 가지고 위와 같은 중복된 비율을 C 언어를 이용하여 카운트 하였다. 표 3에서 보듯이 real-life image들을 사용하여 제안된 알고리즘에 의한 skipping 비율을 시뮬레이션 하였다.

먼저 제안된 알고리즘의 타당성을 검증하고자 C 언어를 이용하여 제안된 방법을 실제영상에 대하여 적용하여 얼마만큼 중복 연산이 필요 없는지를 확인하였다. 실험결과는 그림 11과 같으며 Miss America와 같은 저주파 성분이 많은 이미지는 Flower Garden과 같은 고주파 성분이 많은 이미지보다 중복 되는 연산이 더 많다는 것을 알 수 있다. 이는 저주파 성분의 경우 연속되는 화소의 값이 비슷하기 때문이다.

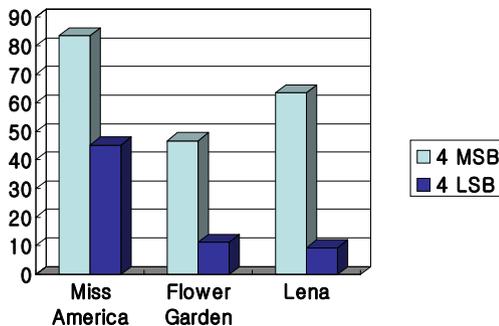


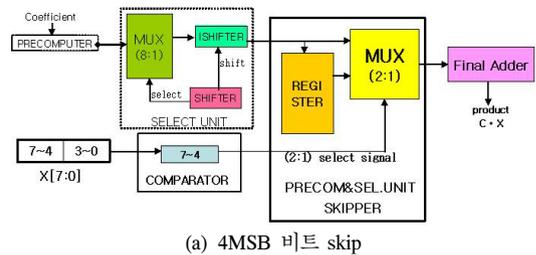
그림 11. CSJM 연산시 중복연산 비율

표 3. 제안된 알고리즘의 skipping 비율

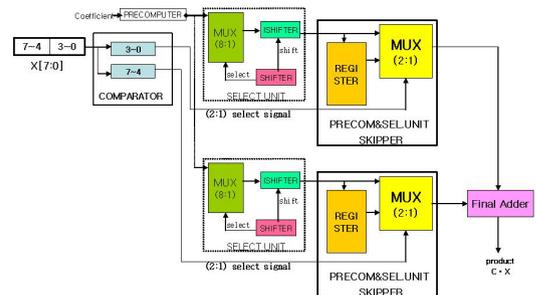
|                   | Miss America     | Lena              | Flower Garden    |
|-------------------|------------------|-------------------|------------------|
| 4MSB skip (Ratio) | 70,629 (83.604%) | 166,843 (63.646%) | 29,746 (46.805%) |
| 4LSB skip (Ratio) | 38,298 (45.334%) | 24,341 (9.285%)   | 7,310 (11.154%)  |
| Total Calculation | 84,480           | 262,144           | 65,536           |

제안한 알고리즘을 이용한 저전력 DCT 프로세서의 구조는 그림 12과 같고, 그 구조에 있어서는 이미지 데이터 두 개의 입력 값이 동일한 값인지 아닌지를 판단하는 비교기와 2:1 멀티플렉서(MUX)를 추가하였다. 이를 통하여 불필요한 중복 연산을 줄일 수 있었다.

그림 12에 사용된 비교기의 자세한 구조는 그림 13과 같다. 비교기는 저전력 CSHM 프로세서의 전 입력 값과 후 입력 값을 비교한다. 만약 중복될 경우 레지스터와 멀티플렉서를 이용하여 불필요한 반복연산을 수행하지 않도록 설계한다. 4MSB와 4LSB의 각각의 4비트가 중복되는 특성을 이용하여 모든 4MSB, 4LSB를 비교하여 불필요한 중복비트들의 연산을 수행하지 않고 CSHM 연산을 수행할 수 있게 되므로 CSHM 프로세서의 스위칭 액티비티를 감소시킬 수 있다.



(a) 4MSB 비트 skip



(a) 4MSB, 4LSB 비트 skip

그림 12. 제안된 DCT 구조

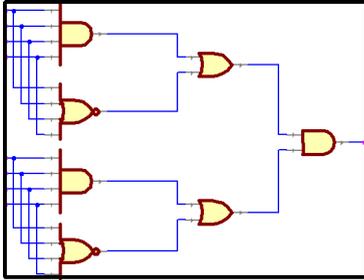


그림 13. 비교기의 구조

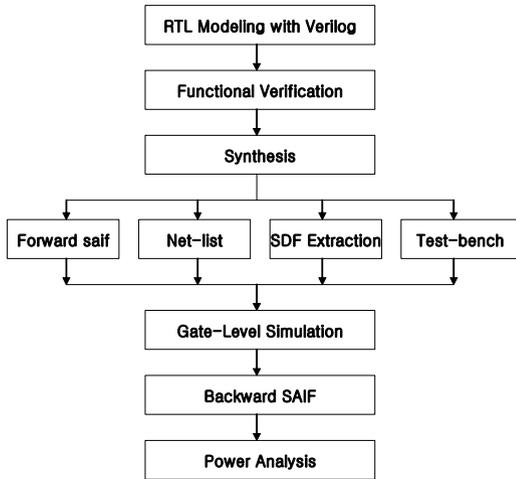


그림 14. 소모전력 측정 과정

```
Running PRESTO HDLC.
Presto compilation completed successfully.
Current design is now 'pwr_test'
1
/* read saif file */
read_saif -input pwr_test_backward.saif -instance tb_pwr_test/ut
Warning: The -unit option is not specified. By default,
it is assumed to be ns. (PIR-62)
Warning: The -scale option is not specified. By default,
the timing unit is scaled by 1. (PIR-63)
1
/* report power information */
report_saif > power_saif.rpt
1
report_power -analysis_effort high > power.rpt
1
report_power -cell -net -flat -analysis_effort high > power_cell.rpt
1
report_power -hier -analysis_effort high > power_hier.rpt
1
exit
1
do_shell>
Thank you...
```

그림 15. 최종 전력 report하는 과정

### VI. 소모 전력 측정 및 결과

본 논문에서 제안된 DCT 구조를 EDA 도구를 이용하여 배치과정을 수행한 후 소모전력을 측정하였다. 구현 및 실험과정은 그림 14와 같다. 실험환경은 SUN Microsystems의 Ultra SPARC의 유닉스 시스템을 이용하였다.

먼저, Verilog-HDL로 기존의 DCT 구조와 제안된 구조를 모두 모델링하고 합성한 후, 기능이 검증되면 네트리스트(Netlist)와 SDF(Standard Delay Format)를 추출한다. 추출된 네트리스트와 테스트벤치(Test-bench)를 가지고 Gate-level Simulation을 수행한다. Gate-level Simulation이 끝나면 네트리스트와 SDF 파일(각 Gate 마다 power 및 지연을 저장하고 있는 라이브러리 파일) 그리고 공정에서 제공하는 Verilog 셀 라이브러리 파일을 이용하여 Backward SAIF을 추출한 뒤 그 전력을 분석하였다. 본 전력측정 시 power.scr 파일을 실행하였고, 그림 15에 나타내었다. 생성된 Backward SAIF, SDF 파일, 네트리스트 그리고 공정에서 제공하는 캐패시턴스 정보가 포함된 라이브러리를 이용하여 Synopsys사의 Power Compiler 툴로 입력 영상에 대한 전력 소모를 측정한 후 비교하였다.

본 실험에서는 Hynix 0.25um 공정 라이브러리를 이용하였다.

여러 이미지에 대하여 기존의 CSHM 기반의 DCT 구조와 제안된 CSHM 기반의 저전력 DCT구조의 전력 소모를 측정하였다. 파워 컴파일러를 이용하여 각 하위 모듈의 전력 소모까지 측정할 수 있다.

아래 표는 여러 이미지에 대하여 기존의 CSHM 기반의 DCT 구조와 제안된 CSHM 기반의 전력 소모를 측정한 결과이다.

표 4. 측정된 소모전력

(단위 : mW)

|               | CSHM         | 제안된 구조         |                |
|---------------|--------------|----------------|----------------|
|               |              | 4MSB skip      | 4MSB&4LSB skip |
| Miss America  | 16.28 (100%) | 15.01 (92.20%) | 14.93 (91.71%) |
| Lena          | 16.17 (100%) | 15.09 (93.32%) | 15.01 (92.84%) |
| Flower Garden | 16.76 (100%) | 15.67 (93.51%) | 15.64 (93.32%) |

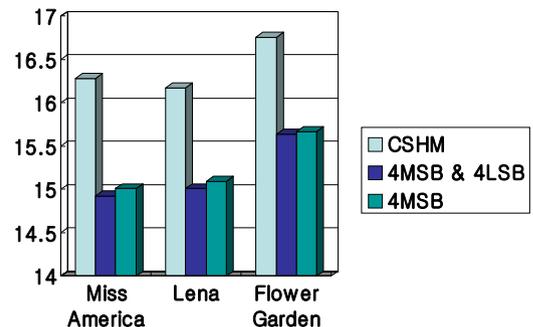


그림 16. 소모전력 비교

표 3과 표 4를 살펴보면 Miss America, Lena, Flower Garden 영상의 평균 중복 연산이 각각 다르지만 모두 제안된 아키텍처가 적용되기 이전의 CSHM 구조 보다 전력이 감소되었다. 여기에서 위의 세 가지 영상은 중복되는 연산 정도에 따라 그 전력 감소 비율이 다르게 나타났을 뿐 본 실험에서 추가되는 하드웨어와는 그다지 큰 연관관계가 없었다. 왜냐하면, 이 경우 추가되는 하드웨어는 CSA 곱셈기가 적용된 Precomputer단을 포함한 전체 아키텍처에서 볼 때 그것이 차지하는 비율이 상당히 작았기 때문이다. 즉 추가되는 하드웨어를 전체 하드웨어의 size에 비추어 보면 그 블록의 size가 훨씬 작기 때문에 이로 인해 소모되는 전력이 아주 미세하여 중복연산 skip에 의한 전력소모 감소비율에 거의 영향을 미치지 않는 것이다.

그림 16의 그래프를 통해 평균적으로 영상에 대한 소모전력이 감소했음을 쉽게 확인할 수 있다. Miss America의 경우 가장 큰 약 8.29%의 소모전력 감소율을 얻을 수 있었다. Miss America 영상은 저주파 성분이 많아서 CSHM 프로세서로의 입력값들이 중복되는 경우의 수가 많다. 즉 중복되어 연산이 되지 않는 경우가 많기 때문이다. 저주파 성분이 많은 Miss America에 비해서 저주파 성분이 적은 Lena 영상의 경우 약 7.16%의 소모전력 감소율을 나타내었다. 또한 Flower Garden 영상은 약 6.68%의 소모전력 감소율을 얻었는데 이것은 Flower Garden 영상이 Lena 영상과 비교하여 볼 때 고주파 성분이 조금 더 많기 때문이다. 그 결과, 제안된 알고리즘은 대부분의 영상에서 전력 감소를 얻을 수 있지만 배경이 단순한 영상의 경우가 가장 높은 소모전력 감소율을 얻을 수 있음을 확인할 수 있었다.

## Ⅶ. 결론

DCT는 영상 압축 시스템의 핵심 기능 중 하나이며, 최근 모바일 기기까지 그 응용범위가 확대되었다. 이러한 기기들은 높은 처리량과 동시에 배터리의 한계로 인하여 저전력 소모를 요구한다.

본 논문에서는 상기와 같은 분야의 응용을 위하여 새로운 저전력 DCT 알고리즘 및 하드웨어 설계를 연구목표로 설정하였다. 이를 위하여 CSHM 기반의 저전력 DCT 알고리즘 및 하드웨어 구조를 제안하였다. 설계된 DCT 프로세서는 다음과 같은 특징을 가진다.

- 기존의 다른 DCT 구조보다 빠른 처리속도 및 저

면적화와 VLSI 설계가 쉬운 장점을 갖는 CSHM 기반의 DCT 구조를 제안하였다.

- CSHM 기반의 DCT 구조의 특징과 영상의 저주파 영역의 DCT 입력 값의 유사성에 착안하여 제안된 저전력 DCT 프로세서의 중복 연산수를 줄였다.
- 추가되는 하드웨어 부담을 최소화하였다.
- 중복되는 연산수를 대폭 줄여 회로 내부의 스위칭을 최소화 하여 소모전력이 최소화 되도록 설계하였다.

본 논문에서 제안한 구조를 설계하고 실험한 결과 기존의 CSHM 기반의 DCT 구조보다 최대 약 9%의 소모전력을 줄일 수 있었다. 따라서, 제안된 DCT 구조는 휴대용 멀티미디어 단말기, IMT-2000 단말기 등 높은 처리량과 동시에 저전력 소모를 요구하는 휴대용 정보단말에 적용될 수 있다.

## 참 고 문 헌

- [1] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete cosine transform," *IEEE Trans. on Computer*, vol. C-23 pp. 90-93, 1974.
- [2] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Magazine*, pp. 4-19, July 1989.
- [3] K. K. Parhi, and D. G. Messerschmitt, "Static Rate Optimal Scheduling of Iterative Data Flow Programs via Optimum Unfolding," *IEEE Trans. On Computers*, pp. 178-195 Vol. 40(2), February 1991.
- [4] K. K. Parhi, "VLSI Digital Signal Processing Systems: Design and Implementation," *Wiley, NY* 1999.
- [5] Liu, K.J.R. Jie Chen, "Algorithm-Based Low-Power and High Performance Multimedia Signal Processing," *Proc. of the IEEE*, pp. 1156-1158 Vol. 86, 1998.
- [6] Yi-Wen Wu, Chen, O.T.-C and Ruey-Liang Ma, "A low-power digital signal processor core by minimizing inter-data switching activities," *Circuits and Systems, Proc. of the 44th IEEE 2001 Midwest Symposium on*, Volume: 1, 2001 pp. 172-175 vol.1.
- [7] B. G. Lee, "A New Algorithm To Compute

- The Discrete Cosine Transform,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 6, pp. 1243-1245, 1984.
- [8] H. S. Hou, “A Fast Recursive Algorithm for Computing The Discrete Cosine Transform,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-25, no. 10, pp. 1455-1461, 1987.
- [9] K. R. Rao and P. Yip, “Discrete Cosine Transform, Algorithm,” *Advantages, Applications, Academic Press*, 1990.
- [10] F. A. Kamangar and K. R. Rao, “Fast algorithms for the 2-D discrete cosine transform,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol.37, pp.1415-1424, Sept. 1989.
- [11] M. A. Haque, “A two-dimensional fast cosine transform,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 1532-1539, 1985.
- [12] H. R. Wu and F. J. Paoloni, “A two-dimensional fast cosine transform algorithm - A structural approach,” *Proceedings International conference Image Processing(Singapore)*, pp. 50-54, Sept. 1989.
- [13] H. R. Wu and F. J. Paoloni, “A two-dimensional fast cosine transform algorithm based on Hou’s approach,” *IEEE Trans. on Signal Processing*, vol. 39, no. 2, Feb. 1991.
- [14] W. H. Chen, Smith. C. and Fralick. S., “A fast computational algorithm for the discrete cosine transform”, *IEEE Trans. Commun.* vol. COM-25, no. 9, pp. 1004-1009, Sep. 1977.
- [15] A. Chandrakasan, R. Brodersen, “Low power digital CMOS design,” *Kluwer Academic Publishers*, 1995.
- [16] 조준동, 임세진, “소자의 스위칭 동작 최소화를 통한 디지털 회로 저전력 상위 레벨 최적화에 대한 연구,” *서울대 반도체 공동연구소*, 10월, 1997년.
- [17] 조준동, “알고리즘 및 아키텍처 수준 저전력 설계자동화,” *전자공학회 CAD 기술 특집*, 12월, 1997.
- [18] J. Park, H. Choo, Muhammad, K. and Roy, K., “Non adaptive and Adaptive Filter implementation based on sharing multiplication”, *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, Istanbul, Turkey, June 2000.
- [19] K. Muhammad, “Algorithmic and Architectural Techniques for Low Power Digital Signal Processing,” *Ph. D. thesis, Purdue University*, 1999.
- [20] Peter M. Kogge, “The Architecture of Pipelined Computers,” *New York :Hemisphere*, 1981.
- [21] 경종민, 박인철 외 공저, “고성능 마이크로프로세서 구조 및 설계 방법,” *대영사*, pp.316338.
- [22] 이용석, “고성능 마이크로 프로세서 곱셈기의 구조(High Performance Microprocessor Multiplier),” *비디오 및 온라인 강좌 시리즈*, <http://mpu.yonsei.ac.kr>
- [23] Jongsu Park, San Kim and Yong-Surk Lee, “A Low-Power Booth Multiplier Using Novel Data Partition Method,” *4th IEEE Asia-Pacific Conference on Advanced System Integrated Circuits*, August 4, 2004.
- [24] Jongsun Park, Muhammad, K., Roy, K., “High-performance FIR filter design based on sharing multiplication,” *Very Large Scale Integration Systems, IEEE Transactions on*, 11, 2, 2003 pp. 244-253.

김 산(San Kim)

정회원



2003년 8월 충남대학교 전자 공학과 학사

2005년 8월 연세대학교 전기 전자공학과 석사

2005년 9월~현재 LG 필립스 LCD 연구원

<관심분야> 영상처리, 마이크로

프로세서, SoC

박 종 수(Jongsu Park)

정회원



2002년 2월 경희대학교 전자 공학과 학사

2004년 2월 경희대학교 전자 공학과 석사

2004년 3월~현재 연세대학교 전기전자공학과 박사과정

<관심분야> 영상처리, 마이크로

프로세서, SoC

이 용 주 (Yong-joo Lee)

정회원



1999년 8월 연세대학교 전자 공  
학과 학사  
2001년 8월 연세대학교 전자 공  
학과 석사  
2001년 9월~현재 연세대학교 전  
기전자공학과 박사과정  
<관심분야> 영상처리, 마이크로

프로세서, SoC

이 용 석 (Yong-Surk Lee)

정회원



1973년 2월 연세대학교 전기 공  
학과 학사  
1977년 2월 University of Michi-  
gan, Ann Arbor 석사  
1981년 2월 University of Michi-  
gan, Ann Arbor 박사  
1993년~현재 연세대학교 전기

전자공학과 교수

<관심분야> 마이크로프로세서, 네트워크 프로세서, 압  
축화 프로세서, SoC