

## 영상신호의 전처리를 위한 CSD 필터의 설계 및 VLSI 구현

正會員 梁鎭機\*, 趙南翊\*\*, 金鱗澈\*\*\*, 李商郁\*

## Design and VLSI Implementation of CSD Filter for Pre-processing of Image Signals

Yoon Gi Yang\*, Nam Ik Cho\*\*, Rin Chul Kim\*\*\*, Sang Uk Lee\* Regular Members

---

본 연구는 한국과학재단의 특정연구과제 연구비 지원으로 이루어진 것입니다.  
(94-0100-06-01-3)

---

## 要 約

본 논문에서는 고속 디지털 필터링을 위한 CSD (canonic signed digit) 계수 필터의 설계 및 VLSI 구현에 대하여 연구하였다. 특히, 영상 압축 시스템인 MPEG2 TM5 (test model 5)의 전처리 단계에서 포맷 변환을 위한 FIR 필터 대신에 사용될 수 있는 CSD 필터를 설계하였다. 그리고 이의 성능을 기존의 필터와 비교하여 하드웨어 복잡도가 더 작으면서도 성능이 우수하다는 것을 보였다. 또한 제시된 필터를 파이프라인, 비트 직렬 구조 및 비트 병렬 시스틀릭 구조로 구현하여 그 동작을 VHDL 시뮬레이션으로 확인하고 기존의 구조와 비교하였다.

## ABSTRACT

In this paper, we investigate on the design and VLSI implementation of CSD(canonic signed digit) coefficient filters for high speed digital filtering. Specifically, CSD coefficient filters are designed, which can be used as format conversion filters in place of the ones employed for the pre-processing of images in the MPEG2 TM5(test model 5). It is shown that the proposed CSD filters perform better than the conventional one, while having lower hardware complexity. Also, the pipelined bit-serial and bit-parallel systolic architectures for the CSD filter are proposed and verified through the VHDL simulation. They are also compared with the existing architectures for CSD filters.

---

\*서울대학교 제어계측공학과  
Dept. of Control and Instrumentation Eng.,  
Seoul National University.

\*\*서울시립대학교 제어계측공학과  
Dept. of Control and Instrumentation Eng.,  
Seoul City University.

\*\*\*한성대학교 정보공학과  
Dept. of Information Eng., Hansung University.  
論文番號 : 95221-0620  
接受日字 : 1995年 6月 20日

### 1. 서 론

여러 영상 압축 시스템에서는 서로 다른 영상 포맷 사이의 변환을 위하여, 또는 압축 효율 향상을 위하여 전처리 필터를 사용하고 있다. 예를 들어 MPEG2 TM5 (test model 5)에서는 여러 크기의 영상 및 서로 다른 색도 포맷 간의 변환을 위하여 샘플링 구조 변환필터를 제시하고 있으며<sup>(1)</sup>, 잡음을 제거하고 영상신호의 상관성을 높여 압축 효율을 높이기 위한 적응 필터와 움직임 보상 필터링에 대한 연구도 진행되었다<sup>(2)-(5)</sup>. 그러나 기존에 설계된 필터는 특별히 VLSI 구현을 고려한 필터라고 할 수는 없다. 예를 들어 MPEG2 TM5에서 사용되는 포맷 변환 필터의 경우를 보면 계수에 별다른 제한 조건이 없는 일반적인 고정 소수점 연산 필터이며 이에 대한 구체적인 VLSI 구조는 제시되지 않았다.

한편, 일반적인 고정 소수점 연산보다 더 빠르고 효율적인 방법으로서 각 계수가 둘 또는 세개를 제외한 나머지 비트가 모두 0이 되도록 제한하며 계수 표현도 CSD (canonic signed digit) 값으로 하는 CSD 계수 필터에 대한 설계와 구현 방법이 최근에 많이 연구되었다<sup>(7)-(13)</sup>. 이와 같은 제한조건에서도 정수 프로그래밍을 사용하여 최적 설계를 하면 그 성능이 실계수 또는 일반 정수 계수를 갖는 필터에 비하여 크게 떨어지지 않으면서도 VLSI 구현시 복잡도를 하나 또는 두개만의 덧셈으로 대체할 수 있어서 필요 면적도 작아진다는 장점이 있다.

앞으로 디지털 통신, 영상 압축 시스템, 특히 HDTV 등에서도 여러 요소에 고속 필터링이 많이 필요할 것으로 예상되므로 이러한 고속 필터를 보다 빠르고 작은 면

적을 필요로 하는 VLSI 구조로 구현하는 것이 매우 중요한 과제이다. 따라서 본 논문에서는 MPEG2 TM5에서 사용된 포맷 변환 필터를 더 효율적인 CSD 계수로 설계를 하고 필터의 성능을 비교한다. 또한 설계된 필터의 파이프라인/비트 직렬 구조 및 비트 병렬 시스틀릭 구조를 제안하여 VHDL 시뮬레이션으로 그 동작을 확인하고 기존의 다른 CSD 구현 기법과 비교한다. 우선 2장에서는 4:2:2 포맷을 4:2:0으로 바꾸는 경우와 CCIR 601의 표준 크기를 SIF 크기로 바꾸는 경우에 많이 사용되는 2:1 추림 (down sampling) 필터를 CSD 계수 필터로 설계하여 MPEG2 TM5에서 사용된 필터와의 성능비교를 한다. 3장에서는 CSD 계수 필터를 비트 직렬 파이프라인 구조로 구현하고 VHDL 시뮬레이션을 통하여 이의 동작을 확인하며 2장에서 설계된 필터의 전체적인 VLSI 구조를 보인다. 4장에서는 비트 병렬 시스틀릭 구조로 CSD 계수 필터를 구현하고 VHDL 시뮬레이션을 통한 동작확인을 한다. 마지막으로 5장에서는 결론을 제시한다.

### 2. 영상 신호의 포맷 변환을 위한 추림 필터의 설계

본 절에서는 MPEG2 TM5에서 4:2:2 → 4:2:0 및 CCIR601 → SIF 포맷 변환에 사용되는 추림 필터를 대체할 수 있는 CSD 계수 필터를 설계하였다. 표 1 이 MPEG2 TM5에서 사용되는 2:1 추림 필터의 계수로서, 특히 비월 주사의 경우 홀수 필드에 적용되는 것이다. 또한 표 1에는 이 필터의 VLSI 구현시 복잡도를

표 1. MPEG2 TM5에서 사용되는 2:1 추림 필터의 계수 (2:1 decimation filter employed in MPEG2 Test Model 5)

Coefficient	Integer Representation	Sign Magnitude Rep.	Two's Complement Rep.
$h_{-3}$	-29/256	1.00011101	1.11100011
$h_{-2}$	0/256	0.00000000	0.00000000
$h_{-1}$	88/256	0.01011000	0.01011000
$h_0$	138/256	0.10001010	0.10001010
$h_1$	88/256	0.01011000	0.01011000
$h_2$	0/256	0.00000000	0.00000000
$h_3$	-29/256	1.00011101	1.11100011

추정할 수 있도록 각 계수의 2진수 표현을 부호, 크기 표현 및 2의 보수로 나타내었다. 본 논문의 연구 목적 중의 하나가 표에서 주어진 필터보다 하드웨어 구현이 간단하면서도 성능이 좋은 CSD 계수 추립 필터를 설계하는 것이다. 이상적인 2:1 추립 필터는 샘플링 주파수를 1.0으로 정규화했을 때 차단 주파수가 0.25이어야 하므로 실제 필터 설계에서는 통과대역(pass band) 주파수  $f_p$ 는 이보다 조금 작고 정지대역(stop band) 주파수  $f_s$ 는 이보다 약간 크도록 설계되어야 한다. MPEG2 TM5에서 사용된 필터는 표 1에서 보는 바와 같이 7탭이며 이의 주파수 응답은 그림 1과 같이 나타난다. 그림으로부터 이 필터의  $f_p$ 는 0.2~0.23,  $f_s$ 는 약 0.3이며 영상의 밝기가 변하지 않도록 하기 위하여 DC 이득은 1.0으로 설계되었음을 알 수 있다.

따라서 본 논문에서는 이와 똑같은 통과 및 차단 주파수를 가지며 고정소수점 구현의 복잡도는 이보다 더 간단하거나 거의 같은 필터를 두가지 설계하였다. 구체적으로,  $f_p = 0.23$ ,  $f_s = 0.3$ , DC gain = 1.0의 제한조건을 가지며 각 계수는 두개 이하의 비트만이 0이 아닌 값을 갖도록 제한하는 7탭과 9탭 CSD 계수 필터를 설계한다. 즉, 7탭과 9탭의 CSD 계수 필터를 설계하여 이를 MPEG2에서의 필터와 비교한다.

필터 설계 방법은 {8}~{10}의 방법을 이용하였다. 이는 BaB(Branch and Bound) 방법을 이용하여 선형 계획을 반복적으로 사용함으로써 최적의 이산계수를 찾는 것으로서 정수형 프로그래밍의 한 방법이다. 이 방법을 이용하는 경우, 특히 DC 이득이 1.0이 되도록 하

기 위하여 주파수가 0인 곳에는 LP(linear programming) 제한 조건에 가중치를 크게 주어야 한다. 그리고, 여러 실험 결과 통과대역을 더 중시하여 이곳의 가중치가 정지대역에서의 가중치의 두배가 되도록 하는 것이 더 성능이 좋음을 알 수 있었다<sup>(10), (14)</sup>. 그 결과 얻어진 필터의 계수를 표 2에 나타내었으며, 각각의 주파수 응답을 그림 2에 나타내었다. 표 2와 표 1을 비교해보면 9탭 CSD 필터의 계산 복잡도는 7탭의 MPEG2에서 사용된 필터와 비슷하며, 7탭의 CSD 필터는 이보다 더 작음을 확인할 수 있다. 본 논문에서는 편의상 표 1의 필터를 "MPEG 필터", 그리고 표 2의 7탭 필터를 "CSD7 필터", 9탭 필터를 "CSD9" 필터로 표시하기로 한다.

위에서 보는 바와 같이 CSD 필터의 하드웨어 복잡도가 더 작으면서도 성능은 좋다는 것을 보이기 위하여 우선, 4:2:2 → 4:2:0 변환에 대한 각 필터의 성능을 비교하였다. 이를 위하여 원 영상을 성능이 좋고 차수가 높은 필터로 수직 방향으로 두배 늘리고 이를 MPEG 필터 및 CSD7, CSD9 필터로 다시 수직 방향의 down 샘플링을 하여 이들 각각을 원래의 영상과 PSNR로 비교한다. 실험에 사용된 영상은 512×512의 "Lena" 영상과 실제 MPEG 영상인 180×120 크기의 "Mobile and Calendar" 영상의 Cr 또는 Cb 신호이다. 실험 결과 Lena 영상에 대해서는 모든 경우 54.0 dB로 높았고, Mobile 영상의 경우도 모두 똑같이 36dB로 나와 세 필터가 같은 결과를 보였다. 그런데 CSD7의 하드웨어 복잡도가 가장 작으므로 4:2:2 →

표 2. 7탭 및 9탭 CSD 필터의 계수 (a) 7탭 (b) 9탭  
(Coefficients of CSD filters of length 7 and 9 (a) Length 7 (b) Length 9)

(a) 7탭 필터 ( $h_1 = h_{-1}$ ,  $h_2 = h_{-2}$ ,  $h_3 = h_{-3}$ )

coefficient	$h_{-3}$	$h_{-2}$	$h_{-1}$	$h_0$
real	-0.062500	-0.078125	0.375000	0.531250
CSD	$-2^{-4}$	$-2^{-4} - 2^{-6}$	$2^{-2} + 2^{-3}$	$2^{-1} + 2^{-5}$

(b) 9탭 필터 ( $h_1 = h_{-1}$ ,  $h_2 = h_{-2}$ ,  $h_3 = h_{-3}$ ,  $h_4 = h_{-4}$ )

coefficient	$h_{-4}$	$h_{-3}$	$h_{-2}$	$h_{-1}$	$h_0$
real	0.05859375	-0.125000	-0.02734375	0.3125000	0.5622500
CSD	$2^{-4} - 2^{-8}$	$-2^{-3}$	$2^{-5} + 2^{-8}$	$2^{-2} + 2^{-4}$	$2^{-1} + 2^{-4}$

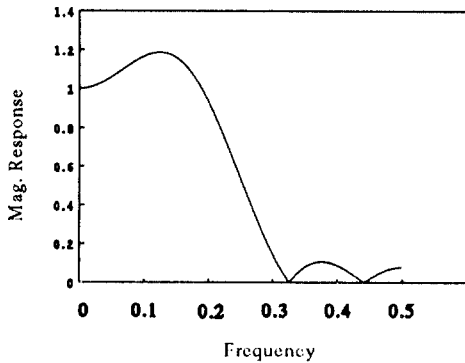


그림 1. MPEG2 TM5에서 사용된 2:1 추림 필터의 주파수 응답 (Frequency Response of 2:1 decimation filter employed in MPEG2 TM5.)

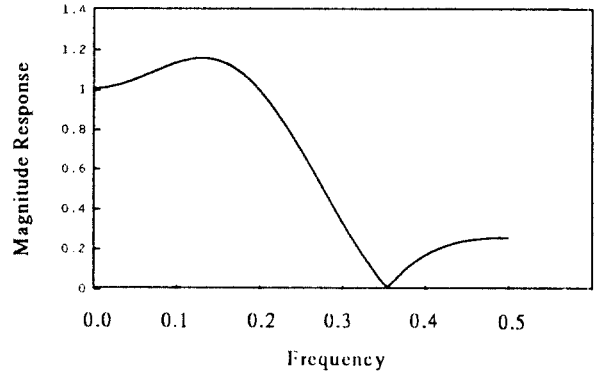
표 3. HDTV -> 디지털 TV 변환을 위한 2:1 down 샘플링 결과의 비교 (Comparison of 2:1 down sampling results from HDTV to digital TV)

Filter	Lena	Mobile
MPEG	43.6 dB	32.1 dB
CSD7	45.8 dB	33.9 dB
CSD9	51.1 dB	39.3 dB

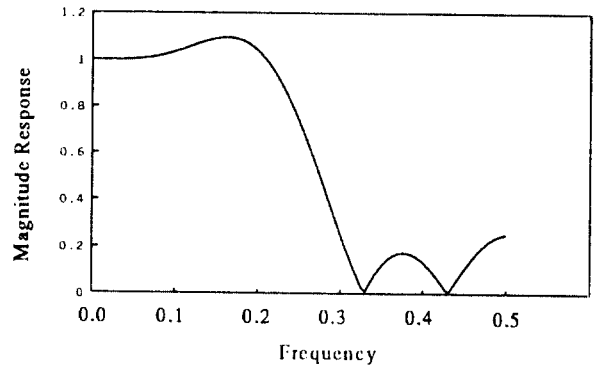
4:2:0의 영상 변환시 본 연구에서 제시된 CSD7 필터를 사용하는 것이 기존의 MPEG 필터를 사용하는 것보다 유리하다고 판단된다.

다음은 CCIR601 -> SIF 변환 또는 HDTV -> CCIR601 변환을 위한 수직, 수평 각 방향의 2:1 추림 필터링에 대한 각 필터의 성능을 비교한다. 이를 위하여 360×240 크기의 Mobile 영상과 512×512 의 Lena 영상을 고차의 성능이 좋은 보간 필터를 이용하여 가로 세로 방향 모두 2배로 늘인 후 이들을 MPEG, CSD7, CSD9 필터를 이용하여 각각 가로 세로 방향으로 다시 2:1 추림을 하고 이들을 원영상과 비교하여 그 결과를 표 3 에 나타내었다. 결과를 보면 본 연구에서 제시된 CSD 필터가 하드웨어 복잡도가 작으면서도 PSNR의 결과는 MPEG 필터보다 좋다는 것을 알 수 있다.

### 3. CSD 계수 필터의 파이프라인, 비트 직렬 구조



(a) 7th order CSD



(b) 9th order CSD

그림 2. CSD 계수 필터의 주파수 응답(Frequency Response of CSD coefficient filter)

본 절에서는 앞에서 설계한 CSD 계수를 갖는 1차원 FIR 필터의 VLSI 구조를 제시한다. VLSI의 면적과 수행시간을 고려하여 연산을 비트 단위로 수행하는가 워드 단위로 수행하는가에 따라 비트 직렬 (bit-serial) 구조 및 비트 병렬 (bit-parallel) 구조로 구분할 수 있는데, 우선 본 절에서는 파이프라인, 비트 직렬 구조를 제시하고 이를 VHDL 시뮬레이션을 통하여 검증한다.

#### 3.1 CSD FIR 필터의 비트 직렬 파이프라인 구조

디지털 신호처리에서 필터의 구현은 매우 중요하므로 이의 여러가지 구현 방법 및 VLSI 구조에 관하여 많은 연구가 수행되었다. 일반적으로 다음과 같이 필터의 식

이 주어졌을 때

$$H(z) = h_0 + h_1z^{-1} + h_2z^{-2} + \dots + h_Nz^{-N} \quad (1)$$

이를 직접 구현하면 많은 덧셈 전파시간이 필요하므로 일반적으로 그림 3 (a)와 같은 전치 구조를 사용한다. 그림으로부터 파이프라인 형태로 출력이 얻어진다는 것을 확인할 수 있다. 특히 이 필터가 선형 위상을 만족하는 경우 그림 3 (b)와 같은 구조로 하드웨어를 줄일 수 있다.

따라서 거의 모든 FIR 필터는 그림 3의 형태로 구현되며, CSD 계수 필터의 구현 역시 전체적인 구조는 이와 같은 전치 구조를 갖는다. 이러한 전치구조를 기본으로 계수 표현 방법에 따라 앞에서 언급한 바와 같이 부동 소수점, 고정 소수점, CSD 연산 필터링으로 구분할 수 있으며 이러한 계산을 워드 단위로 하는가 비트 단위로 하는가에 따라 각각 비트 병렬, 비트 직렬 방법으로 구분된다. 각 계수 표현 방법의 장단점은 앞에서 설명한

바와 같이 CSD의 경우가 복잡도가 가장 작으며, 실계수 필터나 정수 계수 필터에 비하여 계수가 가질 수 있는 값의 범위가 작으므로 각 필터가 최적 설계된 경우 FWL(finite word length) 효과에 의한 잡음은 가장 크다<sup>(6,7)</sup>. 비트 직렬 방법과 병렬 방법을 비교하면 병렬 방법은 한 워드 단위로 계산이 수행되므로 빠르지만 VLSI 면적을 매우 많이 차지한다는 단점이 있다. 반면에 비트 직렬 방법은 비트 단위로 연산이 수행되므로 비트 병렬 방법에 비하여 속도는 느리지만 집적도가 매우 높다는 장점이 있다. 따라서 속도를 최대한 빠르게 하기 위해서는 비트 병렬 구조를 사용해야하고 집적도를 최대한 높이기 위해서는 비트 직렬 구조를 사용해야 한다. 기존의 방법들 중에도 CSD 계수 필터를 비트 직렬 구조로 구현한 바가 [12]등에 있으나 이는 완전한 파이프라인 방식이 아니며 펄스 연산에 대한 구체적인 설명이 없다. 또한, 그림 4가 기존의 비트 직렬 구조를 보이고 있는데, 여기에서 보듯이 이 구조는 덧셈기의 출력이 다

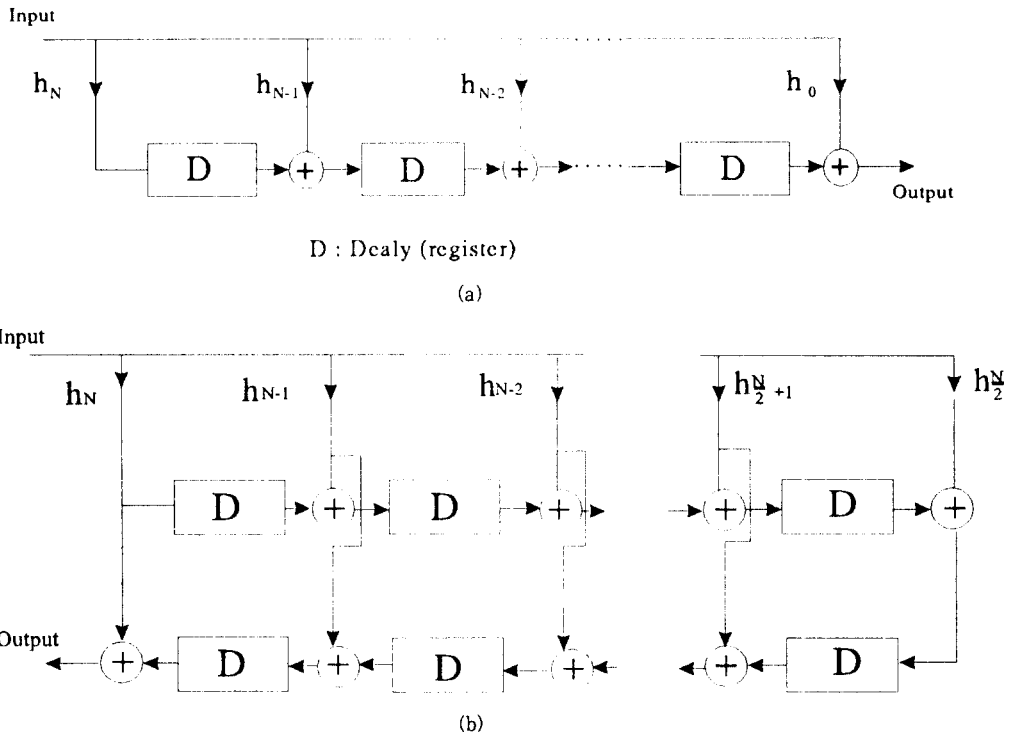


그림 3. (a) FIR 필터의 전치 구조 (Transposed form of FIR filter) (b) 선형 위상 필터인 경우의 구조

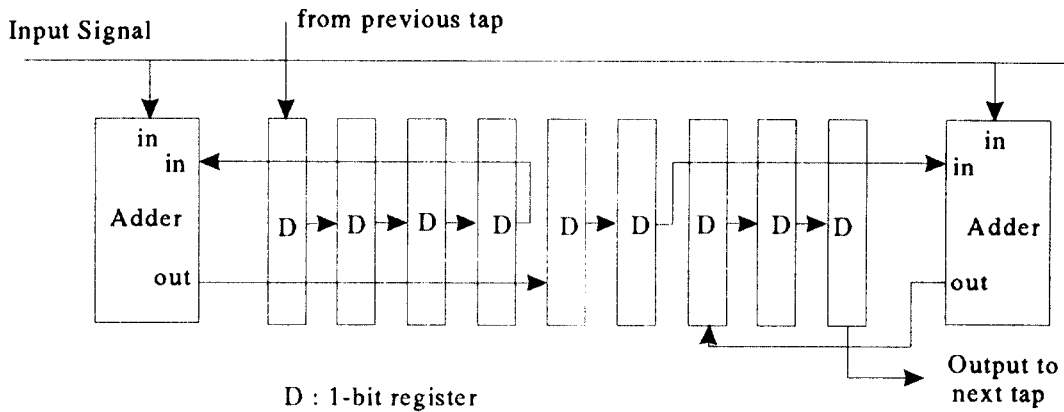


그림 4. 기존의 CSD 계수와 입력 곱셈의 구조. 출력=입력×(2<sup>4</sup>+2<sup>6</sup>)(Structure of existing multiplier for input and CSD filter coefficients, where output = input×(2<sup>4</sup>+2<sup>6</sup>))

시 앞으로 돌아가는 형태이므로 VLSI 구조의 집적도 면에서 좋지 않은 영향을 미친다. 따라서 본 연구에서는 이러한 단점을 개선한 구조를 제시한다. 즉, VLSI 집적도 면에서 유리하도록 인접한 PE (processing element)끼리만 데이터를 주고 받는 국부 연결 파이프라인 구조를 제시하고 계수가 음수인 경우에 대한 구체적인 방법을 제시한다.

본 논문에서 제안하는 구조는 그림 3 과 같은 전치 FIR 구조에 기초하며 계산은 비트 단위로 수행되므로 결국 그림 3 과 전체적으로는 같으면서 여기에서의 일반적인 레지스터 D가 슈프트 레지스터로 된다는 점만 다르다. 즉, 입력과 계수의 곱이 k 비트로 표현되는 경우 그림 5 (a)에서와 같이 전체적인 구조는 그림 3 과 같고 각 D가 k 비트 슈프트 레지스터로 된다는 것이 다른 점이다.

더 구체적으로 곱셈 방법을 보면, 그림 5 (a)의 점선 안의 계수가 2<sup>3</sup>+2<sup>6</sup>라고 했을 때 입력과 이 계수와의 곱셈이 그림 5 (b)와 같은 구조를 통하여 수행된다. 입력을 x<sub>i</sub> 라 하고 이것이 sign bit를 포함하여 01110111 이라 할 때 그림에서 보는 바와 같이 실제 입력은 x<sub>i</sub>\* = 01110111.00000000 으로 하여 결국 x<sub>i</sub>\* 가 각각 3비트 및 5비트 슈프트된 결과들이 더해지게 된다. 기존의 구조인 그림 4 와 비교하면 기존의 구조에서는 덧셈 결과가 다시 앞으로 보내지는 구조인 반면에 제안된 구조에서는 이러한 구조가 없으므로 특히 VLSI 구조에 유

리하다. 또한 기존의 구조에서는 한 입력에 대한 수행이 완전히 끝난 다음에 다음 입력이 들어와야 하지만 제안된 구조에서는 현재 입력의 MSB에 잇달아 다음 입력의 LSB가 입력되는 파이프라인 구조가 가능하다는 장점이 있다.

제안된 구조의 더 자세한 동작 설명을 그림 6 에 나타내었다. 이 그림은 그림 5 의 A, B, C 포인트에서 각 시간에 따른 상태가 어떤 값인가를 나타낸 것이다. 여기서 첫째 입력 x<sub>1</sub> = 01110111이고 둘째 입력 x<sub>2</sub> = 01010101로서 시간 t=16을 기준으로 보면 현재 덧셈기에 2<sup>5</sup>x<sub>1</sub>과 2<sup>3</sup>x<sub>1</sub>이 입력되고 있음을 알 수 있다. 또한 그림에서 알 수 있는 바와 같이 x<sub>1</sub>\* 에 잇달아 x<sub>2</sub>\*가 입력되면 t = 32를 기준으로 (2<sup>3</sup>+2<sup>6</sup>)x<sub>2</sub>가 출력됨을 알 수 있다. 그림 7 은 계수 값의 한 비트 이상이 음수가 되는 경우의 구조로서 계수가 2<sup>3</sup> - 2<sup>6</sup>인 경우의 예를 보이고 있다. 이와 같이 계수가 음수인 곳에서는 입력 x<sub>i</sub>\*가 2의 보수로 되어 슈프트되고 다른 입력과 더해져야 하므로 그림에서와 같이 인버터를 이용하여 우선 x<sub>i</sub>\*를  $\overline{x_i^*}$ 로 만든다. 이와 더불어 2의 보수는  $(\overline{x_i^*}+1)$  이므로 매 16회마다 carry를 1로 만드는 컨트롤러가 필요하다. 그러나 1을 더하는 것은 여기에서는 2<sup>6</sup>을 더하는 것과 같으므로 이를 수행하지 않아도 약간의 오차만이 생기므로 생략할 수 있다. 제안된 필터의 수행 속도는 한 비트의 출력을 얻는데 두개의 1-bit 덧셈기를 통과해야 하므로 입력 x<sub>i</sub>\*가 k 비트인 경우 다음과 같다.

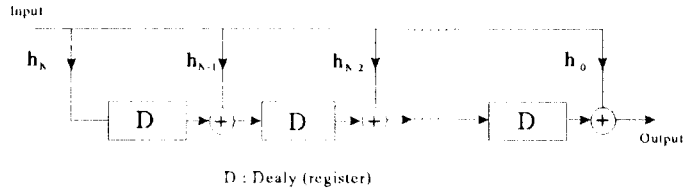
$$T=2kT_{add1} \quad (2)$$

여기서  $T_{add1}$ 은 1 비트 덧셈기의 속도로서 기존의 비트 병렬회로에서 사용되는 일반적인 비트 병렬 덧셈기에 비하여 속도가 빠르므로 비트 직렬 회로가 비트 병렬 회로보다 비트수 배만큼 느리다고 할 수는 없다.

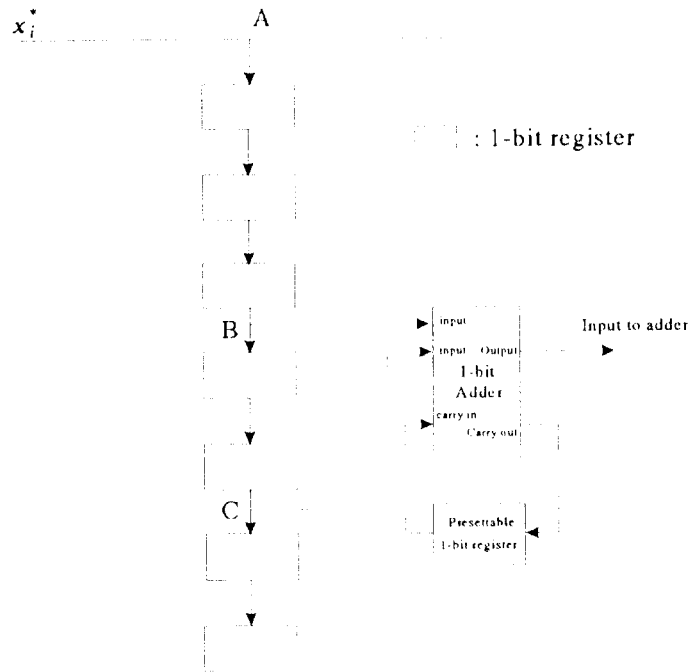
### 3.2 VHDL 시뮬레이션

앞에서 제시된 구조의 검증을 위하여 C 프로그램 및 구체적인 회로와 VHDL을 이용한 시뮬레이션을 수행하

였다. 모의실험에서는 2장에서 영상의 전처리를 위하여 제시된 표 2의 7탭 CSD 필터계수를 예로 하였다. 앞에서 언급한 바와 같이 7탭 CSD 필터의 전체적인 구조는 파이프라인이 용이한 그림 8 과 같은 전치구조이다. 그림 8 의 핵심 부분은 블록  $k_m$  부분인데,  $k_m$ 의 회로도를 그림 5 와 7 에 근거하여 그림 9 와 같이 만들었다. 본 논문의 회로도에는 부분적으로는 VHDL 언어로 합성한 것도 있고 직접 그린것도 있다. 블록  $k_m$ 의 역할은 입력열  $x_{in}$  과 이전의 덧셈결과  $sin$ 으로 부터  $m$ 번째의 필터계수를 입력열과 곱해서 한 워드 지연된  $sin$ 과의 합



(a)



(b)

그림 5. 제시된 필터이 곱셈 구조. 출력=입력×(2<sup>3</sup>+2<sup>5</sup>) (a) 전체구조 (b) 그림(a)의 점선안의 곱셈기 구조 (Architecture of multiplier for the proposed filter, where output = input×(2<sup>3</sup>+2<sup>5</sup>)) (a) Overall structure (b) Structure of the multiplier in dashed box of (a)).

을 출력시켜 다음단으로 출력시키는 것이다. 즉, 그림 9의 H0는 CSD 계수를 이용한 곱셈부를 나타내고 있고, DELAY16은 그림 8의 D로서, 곱셈의 중간과정의 길이 16에 해당하는 길이 16의 레지스터를 나타낸다. 한편 RSTIN 부분은 비트 직렬의 동기가 각 필터단에 맞게 전달되게 하는 일종의 동기 신호이다.

그림 9에서 가장 핵심이 되는 H0라고 표시된 부분

을 그림 10에 더 구체적으로 보였다. 그림 10에서 CSDLAT8 부분은 입력비트열을 지연하여 8 비트까지 저장하는 부분이다. 그림 10은 그림 5에서 제안한 구조를 VHDL로 구현한 부분으로서, 표 2에 제시된 CSD7의  $h_0$ 인  $+2^1 + 2^5$ 의 CSD 계수의 곱셈이 이루어짐을 알 수 있다. 이러한 블록  $k_m$ 을 종합하여 표 2에 제시된 7 tap의 필터링을 구현하는 전체적인 회로의 블

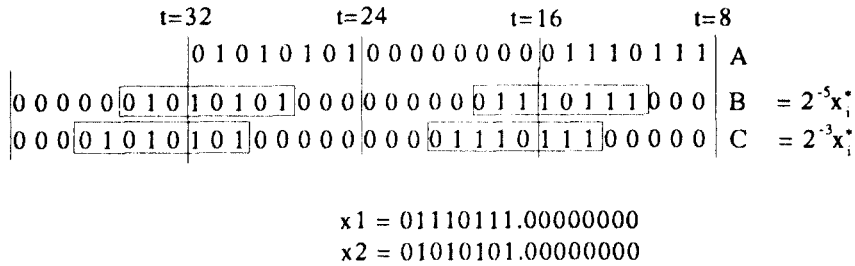


그림 6. 제시된 구조의 동작 설명 (Description of the proposed architecture)

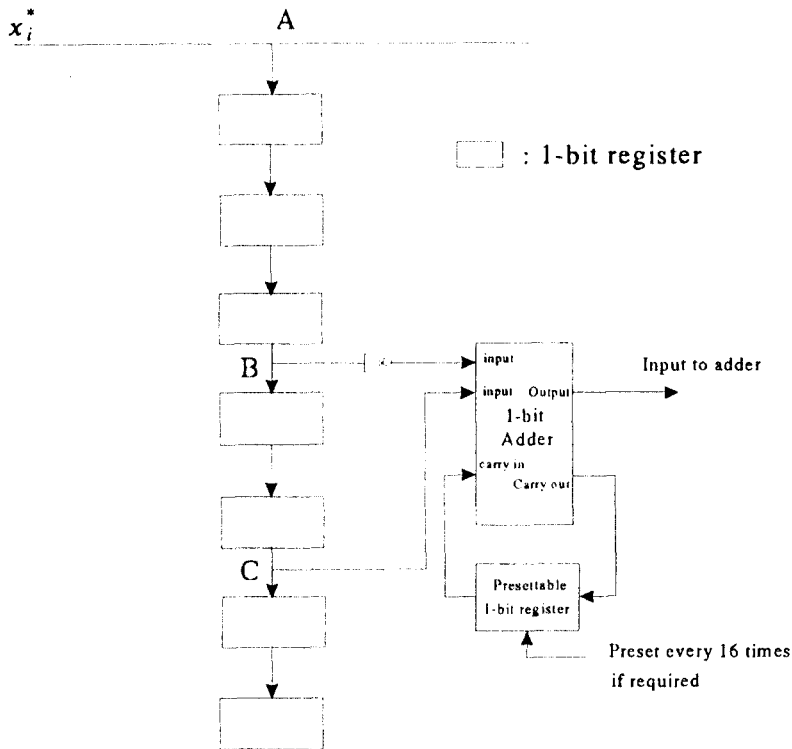


그림 7. 음의 계수가 있을 때의 곱셈 구조. 출력=입력×(2<sup>3</sup>-2<sup>5</sup>) (Architecture of the multiplier when there are negative coefficients, where output=input×(2<sup>3</sup>-2<sup>5</sup>))



력선도가 그림 11에 제시되어 있다. 그러나 본 예에서와 같은 선형 위상 필터의 경우 그림 3 (b)에서 보는 바와 같은 형태로 하여 그림 11에서의 하드웨어를 더 줄일 수 있다.

3.3 기존의 구조와의 비교

그림 5에서 제시한 구조는 비트 직렬 구조로서 일반적인 비트 병렬 연산에 비하여 비트 수 배의 계산 시간이 필요하지만 하드웨어 구현시 콘트롤 논리 회로나 덧셈기 등이 매우 간단해지므로 하드웨어의 복잡도는 비트

수 배 이상이 줄어든다. 따라서 비트 병렬 구조에 비하여 필요한 VLSI 면적이 매우 작아진다는 이점이 있다. 기존의 비트 직렬 구조인 그림 4와 본 논문에서 제시하는 그림 5의 구조를 비교하면, 기존의 구조에서는 쉬프트 결과가 뒤로 돌아가 다시 덧셈기의 입력으로 사용되는 구조를 가지므로 VLSI 구현시 각 요소 사이의 연결선이 길어진다는 단점이 있다. 이는 VLSI 구조에서는 각 요소를 연결하는 metal line이 매우 많은 면적을 차지하므로 이를 가능한 줄이기 위하여 각 요소들은 인접한 것에만 연결되어 있어야 한다는 원칙에 위배된

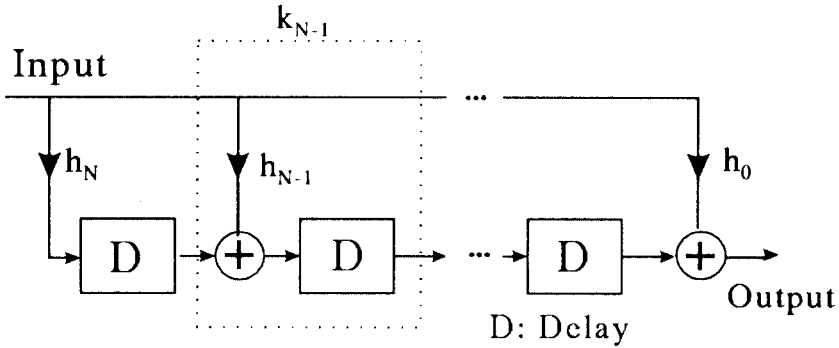


그림 8. VHDL 모의실험을 위한 FIR 필터의 전치구조 (Transposed FIR filter for VHDL simulation)

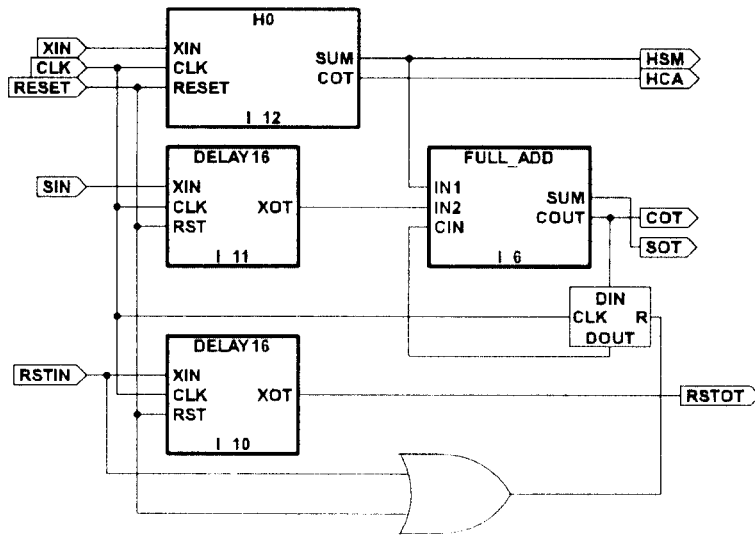


그림 9.  $k_m$  블록의 회로도 (Circuit of the block  $k_m$ )

다. 반면에 제시된 구조에서는 밀리까지 연결되는 선이 없이 국부적으로만 연결되므로 VLSI 구현에 더 유리하다. 또한 기존의 구조는 한 입력에 대한 출력이 완전히 다음 탭으로 전달된 후에 다음 입력을 받아들여 계산이 수행되는 반면에 제시된 구조에서는 현재 입력의 MSB에 이어 곧바로 다음 입력의 LSB가 입력될 수 있으므로 시간을 절약할 수 있다는 장점도 있다.

4. 비트 병렬, 시스틀릭 구조 및 구현

4.1 필터 계수의 시스틀릭 곱셈 구조

CSD 계수 필터의 비트 병렬 구조도 여러가지가 제안되어 왔으며 모두 그림 3과 같은 구조를 기본으로 하고 있고 거의 모두가 덧셈기로서 CSA (carry save adder)를 사용한다<sup>(13)</sup>. CSA는 그림 12와 같은 구조를 갖는 덧셈기로서 출력의 합(sum)과 캐리(carry)를

별도의 덧셈기에서 더하도록 되어있는 구조이다<sup>(11)</sup>. CSA에서는 CSR (carry save register)에 캐리를 저장해 두었다가 각 1 비트 덧셈기에서 나오는 합의 결과와 더한다. 따라서 전체적인 속도는 캐리와 합을 더하는 덧셈기의 속도에 크게 영향을 받는다. 이와 같은 CSA는 일반적인 덧셈보다는  $\sum N_k$ 와 같은 형태의 여러 입력의 연속적인 덧셈에 더 유리한 구조이다. [11]에서 주어진 알고리즘을 따르면 이와 같은 형태의 덧셈을 다른 덧셈기보다 빠른속도로 수행할 수 있다. 그러나 그림 3과 같은 파이프라인 형태의 필터에서는 일반적인 부동소수점 연산이 수행되는 경우 CSA가 다른 구조의 덧셈기보다 유리하지만 계수가 CSD 값일 때에는 덧셈기의 입력 수가 새개정도에 불과하므로 다른 덧셈기를 사용하는 경우보다 절대적으로 유리하지는 않다. 따라서 본 연구에서는 시스틀릭 형태, 즉 병렬, 파이프라인 형태의 CSD 계수 곱셈 구조를 제시하였다. 부분적으로 데이터

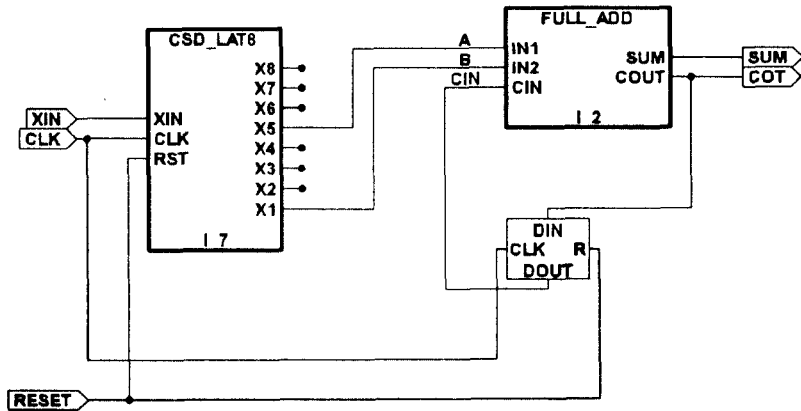


그림 10.  $H_m$  블록의 회로도 (Circuit of the block  $H_m$ )

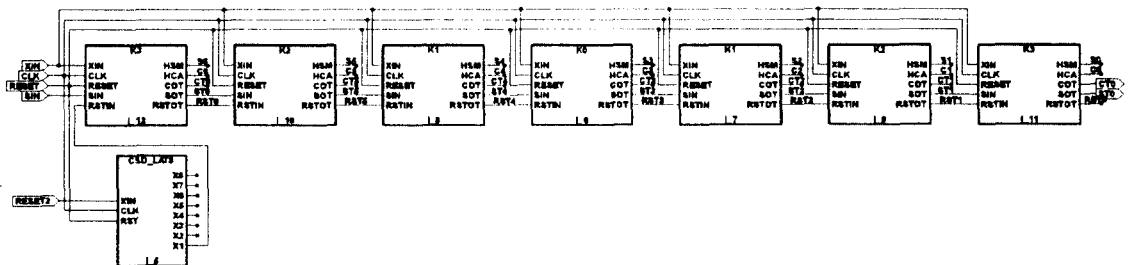


그림 11. Tap CSD 필터의 회로도 (Circuit of the 7 tap CSD filter)

가 멀리 전파되는 부분이 있으므로 엄밀히 말해 시스틀릭 구조라 할 수 없으나 부분적으로 시스틀릭 특징을 가지므로 편의상 이 구조를 시스틀릭 CSD 곱셈기라 부르고 한다. 전체적인 구조는 물론 그림 3의 전치 필터 구조이고 입력이 시스틀릭 구조에 필요한 비트 스큐(bit skew) 형태로 들어간다는 것만이 다르다. 앞 절의 예에서와 같은 계수를 가질 때 제시된 시스틀릭 CSD 곱셈기의 구조를 그림 13에 나타내었다. 그림 13 (a)는 입력  $x_i$ 가 k 비트일 때 그림 3의 입력 부분에 들어가 입력을 bit-skew 형태로 만들어 주기 위한 구조이다. 그림 13 (b)는 bit-skewer를 포함한 전체적인 CSD 계수 곱셈기의 구조로서 m 번째 덧셈기가 n 번째 입력의 LSB의 덧셈을 수행하는 동안 그 상위 비트의 m+1 번째 덧셈기는 n-1 번째 입력의 m+1 번째 비트의 덧셈을 m번째 덧셈기가 기억시켜둔 캐리와 함께 더하는 것이다. 그리고, 그림 3의 출력에는 그림 13 (a)와 반대의 bit-skewer를 삽입하여 바른 출력을 얻어야 한다. 결국 제시된 구조에서는 전형적인 시스틀릭 구조와 같이 첫 입력이 들어간 후 곱셈 비트수 만큼의 시간이 지나야 첫 출력이 얻어지며 일정 시간이 지난 후부터는 1 비트 덧셈기에 소요되는 시간마다 하나의 출력이 얻어지

므로 매우 빠른 속도의 필터링이 가능하다. 이 때 그림 3의 곱셈기 밑의 덧셈기 또한 그림 13 (b)에서의 덧셈 구조와 같은 형태로 설계되어야 한다.

4.2 VHDL 시뮬레이션 및 VLSI 구조

그림 14에는 CSD7 필터를 그림 13의 시스틀릭 CSD 곱셈기로 구현한 회로도가 제시되어 있다. 그림에서 CSAH0부터 CSAH3는 각각 CSD7의 계수 h3부터 h0를 구현한 것이다. 직렬 구조와 마찬가지로 선형 위상 필터의 경우에는 그림 3 (b)와 같은 형태로 하여 하드웨어의 양을 더 줄일 수 있다. 블록 CSADELAY는 전치구조에서 시간 지연 부분과 하단의 덧셈기를 구현한 부분이다. 그림에서 보듯이 최종적인 출력은 역 skewer인 INVSKEW를 통하여 얻어진다. 그림 15에는 그림 14 부분에서 CSAH0 부분의 회로를 더 구체적으로 나타내었다. 그림에서 LATCH16은 곱셈 중간 과정 길이에 해당하는 16 비트의 레지스터이다. 각각 1 비트와 5 비트씩 쉬프트된 입력은 시스틀릭 구현을 위하여 bit-skewer인 SKEW 블록에 입력된다. Bit-skewer를 통과한 16 비트의 신호는 16 개의 1-bit adder로 구성된 블록 SYSHEX에서 시스틀릭의 구조에 알맞게

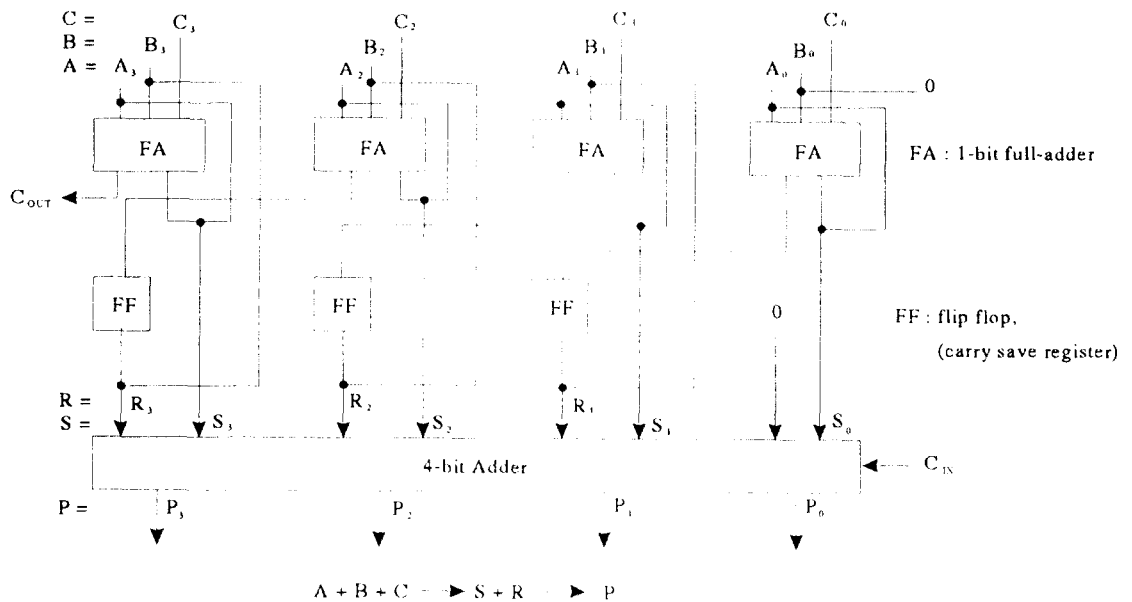


그림 12. CSA (carry save adder)의 구조 (Structure of carry save adder)

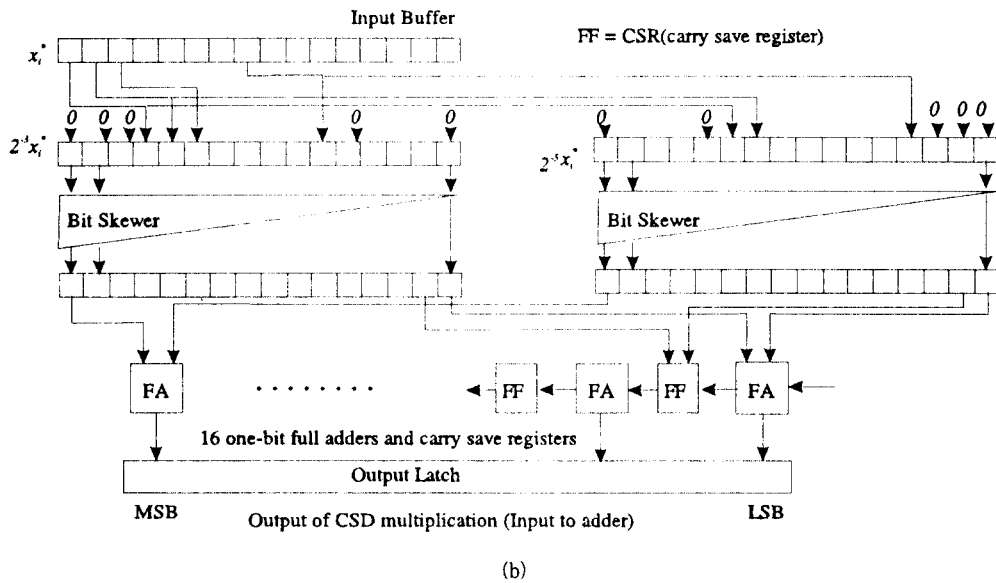
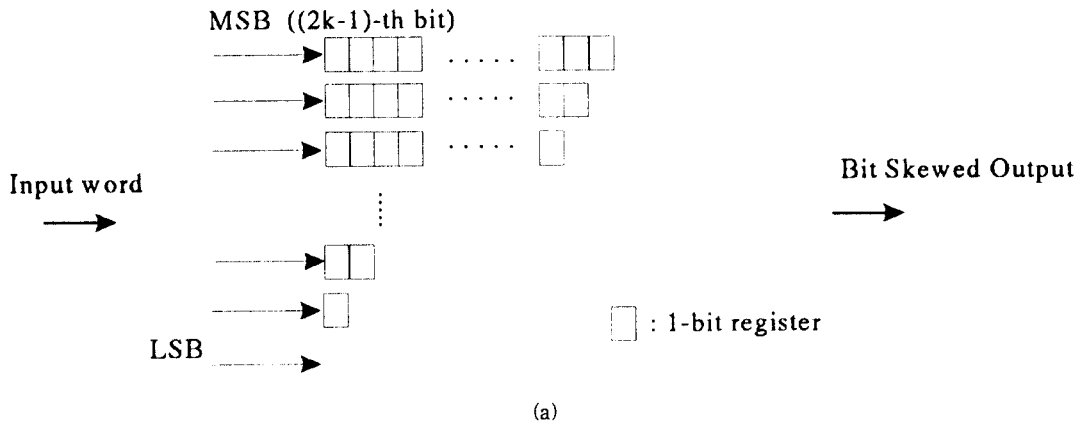


그림 13. 시스템릭 덧셈기를 이용한 시스템릭 CSD 곱셈기의 구조 (a) bit-skewer의 구조 (b) CSD 곱셈기의 구조 (Architecture of bit-skewer (b) Architecture CSD multiplier.)

더해지고 올바른 출력을 얻기 위하여 INVSKEW에서 역 bit-skew의 작용이 이루어진다. 비트 직렬방식과 마찬가지로 병렬방식의 검증에서도 C 프로그램 및 VHDL 시뮬레이터로 출력을 검증하였다.

### 5. 결 론

본 논문에서는 4:2:2 -> 4:2:0 및 CCIR601 ->

SIF 또는 HDTV -> 디지털 TV의 변환을 위한 전처리 필터의 설계 및 구현에 대하여 연구하였다. 이를 위하여 2:1 추림에 사용되는 기존의 필터, 특히 MPEG2 TM5에서 사용한 필터에 대하여 알아보았고, 기존의 필터보다 하드웨어 구조가 더 간단하면서 성능이 더 좋은 CSD 계수 필터를 설계하여 영상 필터링 실험 및 비교를 하였다. 또한 이 필터의 비트 직렬, 파이프라인 구조 및 비트 병렬 시스템릭 구조를 제시하고 각각을 VHDL

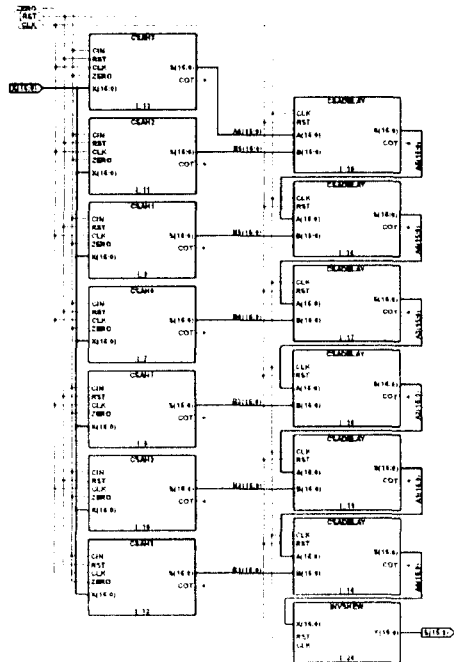


그림 14. 시스톨릭 곱셈기로 구현한 CS7 필터의 블럭선도(Block diagram of CSD7 filter implemented with systolic multiplier)

시뮬레이션을 통하여 동작을 확인하였으며 다른 구조와 비교하였다.

설계된 필터를 이용하여 영상의 추림을 수행해본 결과 본 연구에서 제안된 7탭 및 9탭 CSD 필터는 기존의 MPEG 필터보다 하드웨어 복잡도가 작으면서도 더 우수한 결과를 나타내는 것을 확인할 수 있었다. 또한 설계된 비트 직렬 구조는 비트 병렬 연산에 비하여 비트 수 배의 계산 시간이 필요하지만 하드웨어 구현시 콘트롤 논리 회로나 덧셈기 등이 매우 간단해지므로 하드웨어의 복잡도는 비트수 배 이상이 줄어든다. 따라서 비트 병렬 구조에 비하여 필요한 VLSI 면적이 매우 작아진다는 이점이 있다. 제시된 구조를 기존의 [12]의 구조와 비교해본 결과, 기존의 구조에서는 쉬프트 결과가 뒤로 돌아가 다시 덧셈기의 입력으로 사용되는 구조를 가지므로 VLSI 구현시 각 요소 사이의 연결선이 길어지는 반면에 제시된 구조에서는 멀리까지 연결되는 선이 없이 국부적으로만 연결되므로 VLSI 구현에 더 유리하다는 것을 알 수 있었다. 또한 기존의 구조는 한 입력에 대한 출력이 완전히 다음 탭으로 전달된 후에 다음 입력을 받아들여 계산이 수행되는 반면에 제시된 구조에서는 현재 입력의 MSB에 이어 곧바로 다음 입력의 LSB가

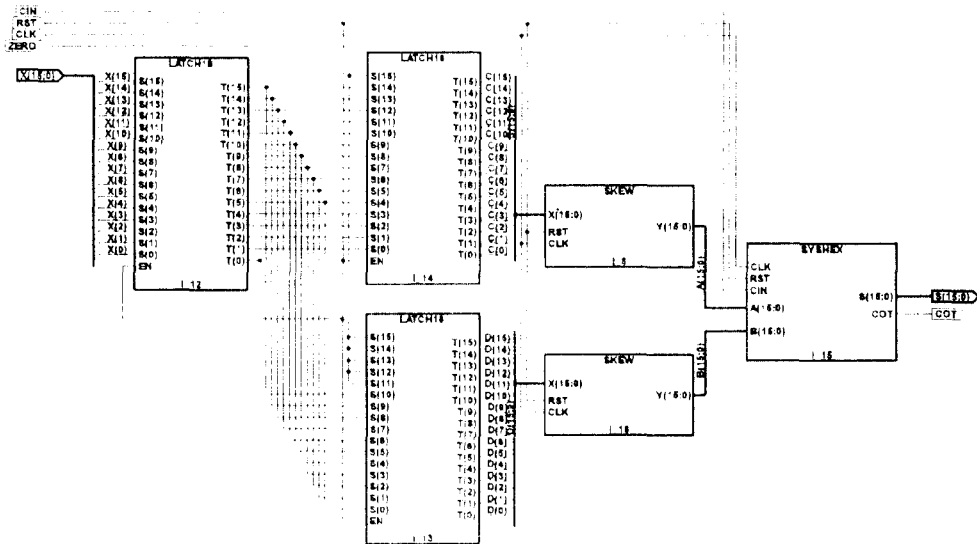


그림 15. 시스톨릭 CSD 곱셈기로 구현한 CSD7 필터의 계수중 h0 계수의 블럭선도 (Block diagram of coefficient h0 of CSD7 filter implemented with CSD multiplier)

입력되어도 되므로 시간을 절약할 수 있다는 장점도 있다. 또한 VLSI 면적이 커지는 대신에 계산속도를 최소한의 시간인 1비트 덧셈 시간 정도로 줄이기 위한 시스템의 형태의 CSD 곱셈구조도 제안하였다. 즉 병렬 연산을 하되, 비트 단위의 파이프라인이 가능하도록 입력을 비트 스큐 형태로 입력시키고 이를 캐리를 저장해 두는 덧셈기에 입력시켜 어느 정도 시간 지연 후부터는 1비트 덧셈기의 전파 시간마다 하나의 출력이 얻어지는 필터 구조를 제시하였다. 이 구조는 앞의 비트 직렬 구조에 비하여 속도는 비트 수배 정도로 빨라지지만 그 제곱에 비례하는 만큼의 VLSI 면적이 필요하다는 단점이 있다.

앞으로 할 일은 제시된 구조를 더 발전시키고 여러 설계 도구를 이용하여FPGA로 만드는 경우, 다양한 기술의 full custom IC로 만드는 경우 등에 대하여 최대 동작 속도와 게이트 수, 칩 면적 등을 구하고 기존의 구조들과 비교를 하는 것이다. 또한 CSD를 사용했을 때 덧셈, 곱셈기를 매우 효율적으로 구현하는 [15]의 방법을 본 연구와 결합하여 더 효율적인 구조로 발전시킬 예정이다. 이러한 연구들을 바탕으로 최종 목표는 전처리 필터 및 일반적인 CSD 필터의 칩을 만들어 여러 고속 디지털 필터링 분야에 활용하는 것이다.

### 참고문헌

1. Motion Picture Expert Group, "Coded representation of picture and audio information," MPEG Video Test Model 5, April, 1993.
2. J. O. Drewery, "Digital filtering of television signals," in *Digital Television*, C. P. Sandbank Ed., John Wiley & Sons, 1990.
3. T. Viero and Y. Neuvo, "3-D median structures for image sequence filtering and coding," in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk Eds., Kluwer Academic Publishers, 1993.
4. E. Dubois and S. Sabri, "Noise reduction in image sequences using motion compensated temporal filtering," *IEEE Trans. Commun.*, vol. COM-32, pp.826-831, July, 1984.
5. M. K. Okzan, M. I. Sezan, and A. M. Tekalp, "Adaptive motion compensated filtering of noisy image sequences," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3, pp. 277-290, August, 1993.
6. H. Samuelli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits and Systems*, vol. 36, pp.1044-1047, July, 1989.
7. Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers of two coefficient space," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, pp.583 - 591, June, 1983.
8. P. Siohan, "2-D filter design for sampling structure conversion," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 1, Dec, 1991.
9. N. I. Cho, S. U. Lee, K. H. Kim, "Design of FIR filter over a discrete coefficient space with applications to HDTV signal processing," *Proc. IEEE ISCAS*, pp.76-79, Chicago, May, 1993.
10. 조남익, 이상욱, "선형계획을 이용한 유한 정밀도 계수 FIR 필터의 설계," 한국통신학회논문지, 제 19권, 12호, pp.2386-2396, 1994년, 12월.
11. K. Hwang, *Computer Arithmetic, Principles, Architecture, and Design*, Reading, John Wiley & Sons, 1979.
12. J. B. Evans, Y. C. Lim, B. Liu, "A high speed programmable digital FIR filter," *IEEE ICASSP '90*, pp.969-971, April, 1990.
13. P. Jain, P. T. Yang, T. Yoshino, "FIRGEN: A computer-aided design system for high performance FIR filter integrated circuit," *IEEE Trans. Signal Processing*, vol. 39, pp.1653-1668, July, 1991.
14. L. R. Rabiner, "The design of finite impulse response digital filters using linear programming techniques," *The Bell System Technical Journal*, vol. 21, pp.1177-1198, July~August, 1972.
15. R. Hartley, K. K. Parhi, *Digit-Serial*

*Computation*, Kluwer Academic Publishers,

1995.

梁 鎰 機(Yoon Gi Yang)      정회원

한국통신학회 논문지 제19권 제10권 참조.

趙 南 翊(Nam Ik Cho)      정회원

한국통신학회 논문지 제19권 제12권 참조.

金 麟 澈(Rin Chul Kim)      정회원

한국통신학회 논문지 제20권 제7권 참조.

李 商 郁(Sang Uk Lee)      정회원

한국통신학회 논문지 제19권 제6권 참조.