論文97-34C-6-3

# 곱셈기가 없는 효율적인 가변탭 FIR 필터 칩 설계

# (Design of an Efficient Multiplierless FIR Filter Chip with Variable Length Taps)

尹 成 賢 *, 鮮 于 明 勳 *

(Sung Hyun Yoon and Myung Hoon Sunwoo)

요 약

본 논문은 종래기술의 하드웨어 크기를 감소시키며 동일한 기능을 수행하는 필터의 가변탭 구조와 곱셈기를 사용하지 않는 새로운 구조의 FIR 필터칩을 설계한다. 제안된 구조는 탭을 가변시키기 위하여 *데이터 재사용 구조 data-reuse structure*와 *계수 반복 기법 recurrent-coefficient scheme*을 사용하였다. 이 방법은 하드웨어 크기가 적은 2개의 8 × 1 MUX, 궤환루프 및 2 × 1 MUX들만을 요구하기 때문에 기존의 주소 생성 유니트와 모듈로 유니트를 이용한 구조에 비해 사용된 게이트 수를 약 20 % 이상 줄였다. 곱셈기를 사용하면 칩 면적이 증가되므로 실시간 처리가 가능한 범위에서 이를 제거하는 것이 유리하다. 본 논문은 곱셈기가 없는 필터를 만들기 위해서 bit-serial 알고리즘을 필터에 적용하여 한 사이클에 두개의 부분곱을 생성하도록 개선하였다. 제안된 구조는 약간의 하드웨어 증가로써 기존 비트 단위 곱셈처리 구조들에 비해 필터링 속도를 두배 향상시켰다. VHDL 모델을 구현하였고 0.8 ㎛ SOG(sea-of-gate) 라이브러리를 이용하여 논리 합성 및 시뮬레이션을 수행하였다. 칩의 동작 주파수는 77 MHz이며 총 게이트 수는 9,507개이다.

**Abstract**

This paper proposes a novel VLSI architecture for a multiplierless FIR filter chip providing variable-length taps. To change the number of taps, we propose two special features called a *data-reuse structure* and a *recurrent-coefficient scheme*. These features consist of several MUXs and registers and reduce the number of gates over 20 % compared with existing chips using an address generation unit and a modulo unit. Since multipliers occupy large VLSI area, a multiplierless filter chip meeting real-time requirement can save large area. We propose a modified bit-serial multiplication algorithm to compute two partial products in parallel, and thus, the proposed filter is twice faster and has smaller hardware than previous multiplierless filters. We developed VHDL models and performed logic synthesis using the 0.8 ㎛ SOG (sea-of-gate) cell library. The chip has only 9,507 gates, was fabricated, and is running at 77MHz.

## I. 서 론

디지탈 필터는 신호원에 대한 잡음 특성, 성능, 시스템 구성의 용이성 등으로 인하여 아날로그 필터에 비해 널리 사용되고 있다. 디지탈 필터는 크게 FIR (Finite Impulse Response) 필터와 IIR(Infinite Impulse Response) 필터로 구분된다. FIR 필터는 동일한 주파수 특성을 얻기 위해 IIR 필터보다 높은 차수가 요구되는 단점이 있으나 선형위상 특성을 얻을 수 있고 곱셈기와 가산기 그리고 지연소자만을 이용하여 쉽게 구현할 수 있어 널리 사용된다[1,2].

FIR 필터의 탭은 응용영역에 따라 가변되므로 고정시킨 탭을 갖는 반도체 칩[3]을 사용하는 경우에는 사용하지 않는 탭에 영(zero)을 인가하므로 필터링시 영

* 正會員, 亞州大學校 工科大學 電機電子工學府
(School of Electronical and Electronic Eng., Ajou Univ.)

接受日字:1997年4月8日, 수정완료일:1997年5月31日

의 탭수만큼 불필요한 연산을 수행하여 연산회로의 효율을 저하시킨다. 이러한 문제점을 해결하기 위해 현재 가변탭 구조를 가지는 필터가 제안되어 여러 응용분야에서 널리 사용되고 있다[4-6]. 이러한 가변탭 구조를 가지는 필터들은 공통적으로 내부 또는 외부 메모리(RAM, ROM)를 가지고 있으며, 이 메모리의 주소를 생성하기 위한 주소생성 유니트(address generation unit)와 메모리를 원환버퍼(circular buffer)처럼 사용하기 위한 모듈로 유니트(modulo unit)로 구성되어 있다. 이러한 필터들은 많은 하드웨어를 요구하기 때문에 비효율적이다.

또한 FIR 필터의 주요한 구성요소를 이루는 곱셈기는 VLSI 칩 구현시 die 면적을 많이 차지하므로 칩의 가격을 증가시킨다. 그러므로 실시간 처리를 할 수 있다면 곱셈기가 없는것이 칩의 가격면에서 효과적이다. 기존의 FIR 필터 중에서 곱셈기를 사용하지 않고 비트 단위로 곱셈을 처리하는 방식[7-11]으로는 bit-serial 방식[7,8]과 Quasi-serial 방식[10,11]이 있다. Bit-serial 방식은 add-and-shift 연산을 수행하는 방식으로 가산을 수행한 후 자리수 정렬을 위하여 쉬프트를 수행하게 된다. 이 방식은 $M$-비트간의 곱셈연산을 위해 총 $M$ 사이클이 요구되며, 연산을 위한 하드웨어로는 한 개의 $M$-비트 레지스터, 2개의 $2M$-비트 레지스터, 및 $2M$-비트 가산기가 필요하다. Qusasi-serial 방식은 2의 보수(2's complement) 곱셈 연산을 위해 add-and-shift 연산을 변형시킨 구조로 $M$-비트간의 곱셈을 위해 총 $2M-1$ 사이클이 요구되며 bit-serial 방식보다 하드웨어 소모가 큰 단점을 갖는다[10].

본 논문은 탭을 가변하기 위하여 *데이터 재사용 구조*와 *계수 반복 기법*을 제안하였다. 이 방법은 탭을 가변시키기 위하여 하드웨어 크기가 적은 2개의 8 × 1 MUX, 궤환루프(feedback loop) 및 2×1 MUX들만을 요구하기 때문에 기존의 탭을 가변시키기 위한 구조[4-6]에 비하여 사용된 게이트 수를 약 20 % 이상 줄였다. 또한 곱셈기가 없는 필터를 만들기 위해 bit-serial 알고리즘을 필터에 적용하여 한 사이클에 두개의 부분곱(partial product)을 생성하도록 개선하였다. 제안된 구조는 약간의 하드웨어 오버헤드(overhead)를 포함하나 기존 비트 단위 곱셈처리 구조들[7,9]에 비하여 필터링 속도를 두배 향상시켰으므로 성능면에서 유리하다.

본 논문은 다음과 같이 구성된다. Ⅱ절에서는 FIR 필터의 기본 연산구조 및 제안된 필터의 수식적인 모델을 기술하며 Ⅲ절에서는 설계된 필터의 VLSI 구조를 논하며, Ⅳ절에서는 제안하는 필터칩의 성능 및 확장 기능에 대해 기술한다. 끝으로 Ⅴ절에서는 연구성과 및 개선방안에 대한 결론을 기술한다.

## Ⅱ. 설계된 FIR 필터의 수식적인 모델

FIR 필터는 IIR 필터와 달리 궤환 구조가 없으므로 식 (1)과 같은 간단한 컨볼루션 연산으로 표현된다[1].

$$y[n]=\sum_{k=0}^{N-1}h[k]x[n-k] \tag{1}$$

여기서 $x[n]$과 $y[n]$은 입력 데이터 및 출력 데이터를 나타내며, $h[n]$은 필터의 유한 충격 응답(Finite Impulse Response) 특성이며 $N$은 필터의 차수를 의미한다. 이 알고리즘을 직접 구현 방식(Direct Realization Method)을 이용하여 구성하면 $N$개의 곱셈기와 $N-1$개의 가산기 및 지연소자가 필요하게 된다[2]. 그러나 이 구조는 하드웨어 크기가 큰 곱셈기를 다수 사용하므로 실시간 처리속도를 만족하는 조건아래 곱셈기를 제거하여 하드웨어를 최적화 시키는 것이 효율적이다.

곱셈기가 없는 필터를 만들기 위해 식 (1)에 bit-serial 알고리즘[7,8]을 적용하면 식 (2)와 같이 표현된다.

$$y[n]=\sum_{k=0}^{N-1}\left(\sum_{j=0}^{M-1}h_j[k]\cdot 2^j\right)x[n-k] \tag{2}$$

여기서 $h_j$, $N$, $M$은 각각 계수 $h$의 j번째 비트, 탭수, 계수 비트수를 나타낸다. Bit-serial 알고리즘은 승수(multiplier)의 LSB로부터 MSB로 쉬프트 시키면서 피승수(multiplicand)를 곱한 결과에 그전에 계산된 부분곱을 누적시키는 방법이다.

곱셈 연산을 위한 사이클 수를 줄이기 위하여 식 (2)의 bit-serial 알고리즘을 이용한 필터식의 논리곱 부분을 짝수와 홀수 승수 비트로 나누어 bit-serial 알고리즘을 적용하면 식 (3)과 같이 표현된다.

$$y[n]=\sum_{k=0}^{N-1}\sum_{j=0}^{\frac{M}{2}-1}\left(h_{2j}[k]\cdot 2^{2j}+h_{2j+1}[k]\cdot 2^{2j+1}\right)x[n-k] \tag{3}$$

이때 분리된 짝수 및 홀수 승수 비트는 LSB로부터 MSB로 쉬프트 되면서 논리곱되므로 동시에 2개의 부

분곱을 생성한다. 기본적으로 비트 단위 곱셈 알고리즘을 적용한 필터식은 $NM$ 사이클[7-9]을 요구하나 제안된 알고리즘은 $MN/2$ 사이클이 걸리므로 2배의 속도 향상을 얻는다.

## Ⅲ. 설계된 FIR 필터의 VLSI 구조

제안하는 1차원 FIR 필터 칩의 가변탭 구조는 크게 데이터 블록의 *데이터 재사용 구조*와 계수 블록의 *계수 반복 기법*을 이용한 구조로 구성된다. 제안한 구조를 설명하기 위해 한가지 예를들면, 데이터와 계수는 8-비트로 구현하였으며 필터의 탭수는 64이내에서 가변할 수 있도록 설계하였다. 그러나 필터의 탭수 및 계수가 확장 및 축소되는 경우에도 이 구조는 용이하게 적용될 수 있다.

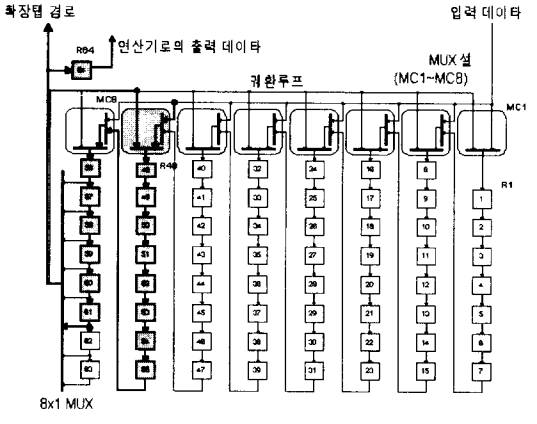


그림 1. 데이터 블록의 데이터 재사용 구조
Fig. 1. The data-reuse structure for data block.

1. 데이터 블록의 데이터 재사용 구조

데이터 블록의 *데이터 재사용 구조*는 그림 1과 같이 64개의 8-비트 레지스터 열 (R1-R64)과 한개의 8-비트 8 × 1 MUX 및 8개의 *MUX 셀*(MC1-MC8)로 구성된다. 각 MUX 셀은 2개의 2 × 1 MUX로 구성된다.

$N$차 필터링 결과를 얻기위해서는 $N-1$개의 입력 데이터가 지연선(delay line) 또는 원환버퍼에 저장되어 연산시 재사용 되야한다. *데이터 재사용 구조*는 $N-1$개의 입력 데이터를 궤환시켜 데이터를 필터 연산기와 원환버퍼로 보내주는 역할을 한다. 원환버퍼는 $N-1$개의 레지스터, 궤환루프, *MUX 셀*들로 이루어진다. 8 × 1 MUX는 레지스터 R56-R63 출력 중의 하나를 선택하여 동시에 궤환루프와 R64로 데이터를 전송한다. $N$ 탭에 의해 설정된 *MUX 셀*들중의 하나는 궤환루프를 통해 이전의 입력 데이터를 받아들이며, 매 $N$ 사이클마다 현재의 입력 데이터를 그림 1의 상위에 있는 레지스터(R1, R8, R16, R24, R32, R40, R48, R56)로 전송해준다. 다른 MUX 셀들은 레지스터들을 연결하여 필터의 지연선을 만들어준다. 그러므로 필터탭수는 원환 형태로 연결된 레지스터 수에의해 결정된다.

예를 들어, 필터탭수가 15라고 가정하자. 입력 데이터는 $D_0$, $D_1$, ..., $D_{14}$, ...이며 계수는 $C_0$, $C_1$, ..., $C_{14}$이다. 15탭 필터링을 위하여 15개의 레지스터들은 그림 1의 빗금 친 레지스터를 따라 원환버퍼를 형성한다. $k$번째 필터링 결과를 얻기위해서는 단지 15개의 입력 데이터, 즉 $D_k$, $D_{k+1}$, ..., $D_{k+14}$ 가 필요하며 $k+1$번째 필터링 결과를 얻기위해서는 $D_{k+1}$, $D_{k+2}$, ..., $D_{k+15}$ 가 필요하다. 따라서 $k$번째 결과를 얻은 후에 $D_k$는 제거되고 $D_{k+15}$는 원환버퍼로 입력된다. 이때 외부 핀이 원환버퍼 연결을 제어한다.

*데이터 재사용 구조*를 이용하면 데이터를 연산기로 자동적으로 공급하기 때문에 기존칩[4-6]의 주소생성 유니트 및 모듈로 유니트가 필요 없으며 상대적으로 하드웨어 크기가 적은 8 × 1 MUX, 궤환루프 및 8개의 *MUX 셀*들만을 이용하여 구현되므로 하드웨어 비용면에서 유리하다. 또한 제안한 구조는 8개의 *MUX 셀*을 추가하므로써 64탭에서 128탭으로의 확장이 가능하다. 이렇듯 탭수가 2의 자승배로 증가하는 것에 비해 하드웨어 크기는 선형적인 증가를 보이므로 탭수를 크게 확장할 때 매우 효율적이다.

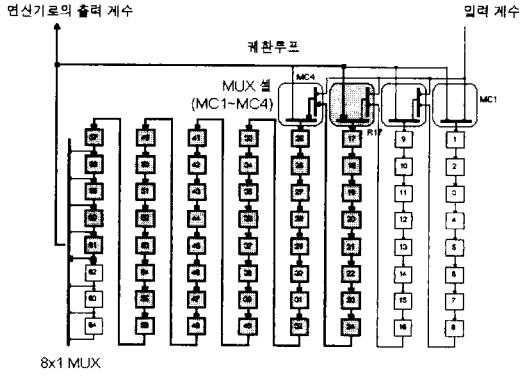


그림 2. 계수 블록의 계수 반복 기법
Fig. 2. The recurrent-coefficient scheme for coefficient block.

2. 계수 블록의 계수 반복 기법

그림 2는 64개의 8-비트 레지스터열(R1-R64)로 구성되어 있는 계수 블록을 보여준다. 계수 블록은 *계수 반복 기법*을 이용하여 *데이터 재사용 구조*에 비해 *MUX 셀*의 수를 줄였다. *계수 반복 기법*은 식 (4)를 만족시키는 범위에서 탭수가 32 이하인 경우에 계수탭 $N$을 $r$번 반복시켜 $Nr$ 탭으로 변화시켜 사용하는 방법이다. 따라서 64 이하의 모든 탭은 단지 4개의 *MUX 셀*만을 갖는 33에서 64탭을 위한 하드웨어로 표현된다.

$$33 \leq Nr \leq 64 \qquad (4)$$

예를 들어 15탭 필터링 계수를 $C_0$, $C_1$, ..., $C_{13}$, $C_{14}$라 할 때 식 (4)에 의해 15탭 계수열을 3번 또는 4번 반복 입력하면 45 또는 60 필터탭으로 변환된다. 만약 45 필터탭으로 선정하였다면 처음의 입력 계수 $C_0$, $C_1$, ..., $C_{13}$, $C_{14}$는 그림 2에 보여준 빗금 친 레지스터 R61-R47에, 두번째는 R46-R32에, 세번째는 R31-R17에 저장된다. 이 구조는 궤환루프를 통하여 R61-R17까지의 반복입력된 계수를 포함한 원환버퍼를 만든다.

그림 2에서 보였듯이 계수를 반복적으로 저장하여 사용하는 방법은 가변탭을 지원하기 위한 하드웨어를 8 × 1 MUX 및 4개의 *MUX 셀*(MC1-MC4)만으로 구현할 수 있으므로 사용된 하드웨어 크기를 줄일 수 있다. 또한 내부의 탭을 128개로 확장할 경우 추가될 레지스터 파일에 단지 4개의 *MUX 셀*만을 추가하면 가능하다.

이와 같은 데이터 블록 및 계수 블록에서의 최대 임계경로(critical path)는 8 × 1 MUX와 2 × 1 MUX로 탭의 확장에 관계없이 일정하므로 고속의 동작이 가능하다.

3. 필터의 연산기

그림 3은 파이프라인 연결된 필터 연산기를 보여준다. 필터 연산기는 2개의 4-비트 병렬입력직렬출력 쉬프트 레지스터(PISO-SR1: Parallel-Input-Serial-Output Shift Register, PISO-SR2), 2개의 부분곱 생성단(PPG: Partial Product Generator), 부분곱 합성단(PPS: Partial Product Summation), 16-비트 그룹 CLA 가산기 및 레지스터들로 구성된다.

계수값은 한번에 두개의 부분곱을 생성하기 위해 짝수와 홀수 비트로 나뉘어진다. 분리된 계수는 2개의 4-비트 PISO-SRs에 각각 저장되며, 매 사이클마다 값을 PPG로 쉬프트 시킨다. 데이터 블록으로부터 전송된 데이터는 8-비트 래치(latch)에 저장된다. 2개의 PPG는 데이터 비트와 계수 비트를 논리곱시켜 매 사이클마다 2개의 부분곱을 생성한다. PPS는 2개의 부분곱을 더하며, 이 결과는 변환로직에서 16-비트 스트림으로 변환된다. 16-비트 그룹 CLA 가산기는 변환된 스트림과 누적된 부분곱을 더해준다. 곱셈은 4 사이클이 소모되며, 하나의 필터링 결과를 얻기위해서는 $4N$ 사이클이 소모된다.

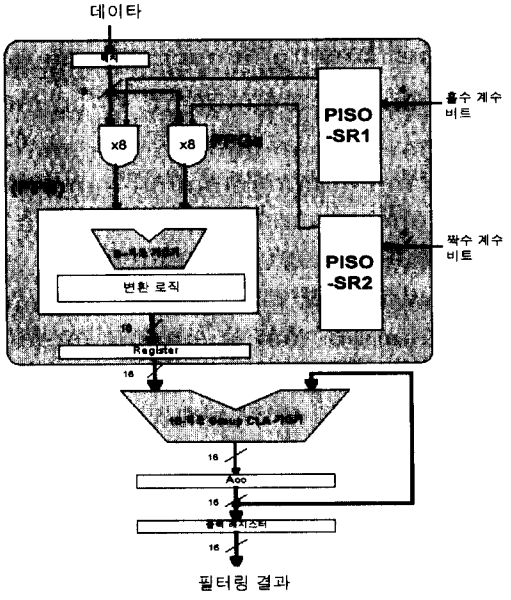


그림 3. 필터의 연산기
Fig. 3. The filter computation unit.

본 구조에서 홀수 계수 비트로 논리곱된 부분곱을 8-비트 가산기에 1-비트 쉬프트 레프트시켜 연결하는 방법을 사용하였다. 이 구조는 PPS에서 쉬프트 레지스터를 사용하지 않고 동일한 기능을 수행하므로써 하드웨어 및 타이밍 오버헤드를 최소화 할 수 있다. Acc 값은 필터링 결과값을 얻기위해 $4N$ 사이클때마다 출력 레지스터로 전송된다.

제안된 구조는 기존의 무곱셈 필터[7,9]와 비교하여 2배 빠르게 연산을 수행하므로써 필터링 사이클을 반으로 줄일 수 있다.

## Ⅳ. 설계된 필터칩의 성능 및 확장 구조

설계된 FIR 필터는 VHDL을 이용하여 모델링 하였

으며 0.8 ㎛ SOG 라이브러리를 사용하여 논리합성 하였으며 신호의 버퍼링을 고려한 전체 게이트 수는 9,507개이다. 논리합성후의 시뮬레이션 결과는 게이트 레벨의 모든 시간지연 및 타이밍 정보를 포함하고 있으므로 fanin/out, rising/falling 타임 등과 같은 실제 회로상의 특성들을 고려하여 VHDL 모델의 타이밍 시뮬레이션 결과와 동일함을 검증하였다. 설계된 필터는 실제 칩으로 구현하여 기능을 검증하였으며 칩의 동작 주파수는 77 MHz이다.

S. Ward[12]를 근거로하여 기존의 가변탭[4-6]을 게이트로 환산하면 약 890개이며 본 구조는 712개의 게이트만으로 구성되므로 약 20%의 게이트를 절약할 수 있다. 필터 연산기만을 비교하였을 때 [12]에 근거한 기존의 연산기[7]는 약 800개의 게이트로 이루어지나 제안된 연산기는 882개의 게이트로 이루어진다. 이는 하드웨어의 큰 증가없이 연산 속도를 2배 높일수 있어 유리하다.

제안된 필터는 64탭 FIR 필터로 사용하는 경우 초당 313 Ksample을 처리하며, 이를 2-D 콘볼루션 필터로 사용할 경우에는 초당 2.293 Msample을 처리한다. 이는 256 × 256 영상에 대해 35 frame을 실시간 처리한다.

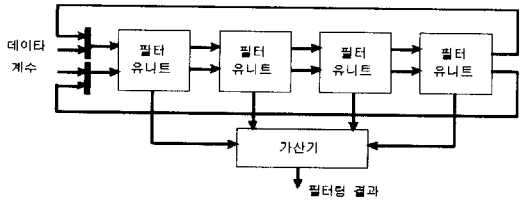


그림 4. 256탭의 1차원 FIR 필터
Fig. 4. The extended 256-tap filter.

본 구조는 탭을 확장시키거나 해상도가 높은 영상을 처리하기 위해 필터 유니트를 다단으로 연결하여 사용할 수 있다. 그림 4와 같이 필터 유니트를 4개 직렬 연결하여 256 탭 FIR 필터로 사용하는 경우 초당 313 Ksample의 데이터를 처리 할 수 있으며, 그림 5와 같이 필터 유니트를 세 개 병렬로 영상 메모리와 연결하여 사용하는 경우 초당 7.08 Mpixel을 처리한다. 이는 512 × 512의 동영상 데이터를 초당 27 프레임 실시간 처리할 수 있는 속도이다. 1-D, 2-D 신호 처리에 있어 연산 속도를 개선하기 위해 그림 6과 같이 필터 연산기를 증가시킬 수 있다. 위에서의 모든 확장구조는 필터 유니트와 연산기가 단지 9,507개와 882개의 게이트로 이루어지기 때문에 하나의 칩으로 구성될 수 있다.

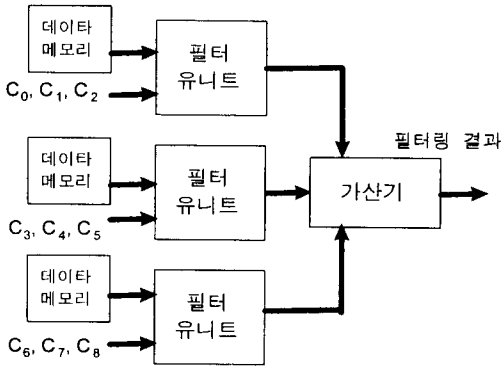


그림 5. 3 × 3 윈도우를 갖는 2차원 콘벌루션 필터
Fig. 5. The 2-D convolution filter with a 3 × 3 window.

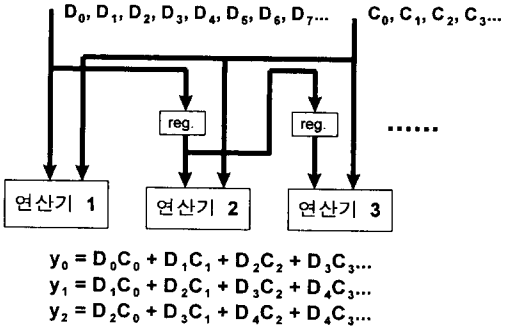


그림 6. 필터의 병렬 연산기 구조
Fig. 6. The parallel computation units.

## V. 결 론

본 논문은 곱셈기를 사용하지 않고 필터의 탭수를 가변시킬 수 있는 새로운 FIR 필터 구조를 제안하고 실제 설계 및 구현하였다. 필터탭을 가변시키기 위해 본 논문에서 제안하는 *데이터 재사용 구조*와 *계수 반복 기법*은 기존칩에 비해 확장성이 우수하며 사용된 하드웨어 크기를 약 20 % 이상 줄였다. 또한 새로이 제안하는 bit-serial 알고리즘을 사용하여 구현한 필터의 연산기는 하드웨어 증가없이 기존 무곱셈 필터에 비해 2배의 처리속도를 향상시켰다. 설계된 구조는 파이프라인을 이용하여 77 MHz의 클럭 주파수에서 동작하며 확장성을 가지고 있어 1차원 및 2차원 신호처리를 위한 다양한 목적의 필터를 구성할 수 있다. 구현

된 필터는 0.8 ㎛ SOG 셀 라이브러리를 이용하여 논리합성하였고 전체 게이트 수는 9,507개이다.

## 참 고 문 헌


[1] Sanjit K. Mitra and James F. Kaiser, *Handbook Digital Signal Processing,* WILEY-INTERSCIENCE, 1993.

[2] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing : Principles, Algorithms, & Applications,* 1996.

[3] Harris Semiconductor Inc., *Digital Signal Processing Databook,* HSP43168, 1994.

[4] Motorola Inc., *Motorola Semiconductor Technical Data,* DSP56200.

[5] Harris Semiconductor Inc., *Digital Signal Processing Databook,* HSP4312, HSP 43481, 1994.

[6] Advanced RISC Machines Limited (ARM), ARM7DM, 1996.

[7] Robert E. Morley, Jr., Gray E. Christensen, Thomas J. Sullivan, Orly Kamin, "The Design of a Bit-Serial Coprocessor to Perform Multiplication and Division on a Massively Parallel Architecture," in *Proc. IEEE, The 2nd Symposium on the Frontiers of Massively Parallel Computation*, Fairfax, U.S.A., pp. 419-422, Oct. 1998.

[8] Israel Koren, *Computer Arithmetic Algorithms,* pp. 29-32, 1993.

[9] Woo Jin Oh and Yong Hoon Lee, "Implementation of Programmable Multiplierless FIR Filters with Powers-of-Two Coefficients," *IEEE Trans. on Circuits and Syst.*, vol. 42, no. 8, Agu. 1995.

[10] Earl. E. Swartzlander, Jr., "The Quasi-Serial Multiplier," *IEEE Trans. Comput.*, vol. C-22, pp. 317-321, Apr. 1973.

[11] T. G. MaDaneld and R. K. Guha, "The Two's Complement Quasi-Serial Multiplier," *IEEE Trans. Comput.*, C-24, pp. 1233-1235, 1975.

[12] S. Ward, P. Barton, J. B. G. Roberts, and B. J. Stanier, "Figure of merit for VLSI implementations of digital signal processing algorithms," *Proc. Inst. Elec.*, vol. 131, Part F, pp. 64-70, Feb. 1984.


## 저 자 소 개


**尹 成 賢(正會員)**

1996년 2월 아주대학교 전자공학과 학사. 1996년 2월 ~ 현재 아주대학교 전자공학과 석사과정. 관심분야는 DSP 칩, 통신 및 신호처리용 ASIC 설계

**鮮牛 明勳(正會員)**

1980년 2월 서강대학교 전자공학과 학사. 1982년 2월 한국과학기술원 전자공학과 석사. 1982년 3월 ~ 1990년 8월 한국전자통신연구소(ETRI) 연구원. 1985년 9월 ~ 1990년 8월 Univ. of Texas at Austin 전자공학과 박사. 1990년 8월 ~ 1992년 8월 Motorola, DSP Chip Division, 미국. 1992년 8월 ~ 1996년 10월 아주대학교 전기전자공학부 조교수. 1996년 10월 ~ 현재 아주대학교 전기전자공학부 부교수. 관심분야는 VLSI 및 Parallel Architecture, 통신, 영상 및 신호처리용 ASIC 설계