

技術解説

고속 신호 처리를 위한 디지털 필터의 설계

김진웅, 장경희
(한국 전자통신 연구소)

요약

본고에서는 일반적인 디지털 필터의 설계 방법과 설계시 고려 사항을 살펴보고, 특히 실시간 구현이 어려운 고속 IIR 디지털 필터의 설계 방법에 대하여 고찰하였다. 현재의 발달된 VLSI 기술의 잇점을 최대한으로 활용하기 위한 병렬 및 파이프라이닝 필터들의 구조 및 특성을 비교하였으며, 실제 하드웨어를 구현하는 여러가지 방법들을 상술하였다. 또한 각 연산 소자를 고속으로 구현하기 위한 비트레벨 구조 및 수체계(Numer System)에 대해 알아보고, 이를 이용한 파이프라이닝 필터의 설계 예를 보였다. 필터의 구조에 따라 유한 길이 레지스터(FWL)의 영향이 달라지며, 제안된 새로운 구조에 대한 FWL 영향의 분석이 항상 수행되어야 한다. 디지털 필터에서의 FWL 영향과 그 분석 방법, 그리고 이를 줄이기 위한 설계 방법에 대해 기술하였다. 디지털 필터를 포함한 많은 디지털 신호 처리 알고리즘이 내제된 병렬성을 갖고 있으며, 이들의 효율적인 하드웨어 실현을 위해 본고에서 고찰한 기술들이 적용될 수 있다.

I. 서론

디지털 필터의 설계 및 구현은 다음 세가지 단계의 문제를 순서적으로 해결함으로써 이루어진다.

(1) 근사화 문제(Approximation Problem) : 주어진 규격이나 요구사항으로 부터 필터의 입출력 특성을 나타내는 전달 함수 $H(z)$ 를 구하는 문제이다. 보통 필터의 규격은 주파수 응답 형태, 즉 차단 주파수, 통과대역과 정지대역의 리플(ripple)등으로 주어진다.

(2) 실현 문제(Realization Problem) : 주어진 전달 함수로 부터 FWL(Finite Word-Length) 영향 또는 요구되는 계산량등을 고려하여 최적화된 필터 구조를 찾아내는 문제이다. 단, 필터를 범용 컴퓨터에서 소프트웨어로 구현할 경우에는 크게 중요하지 않

으며 주로 SFG(Signal Flow Graph)로 표현된다.

(3) 구현 문제(Implementation Problem) : 주어진 계산 알고리즘(SFG)으로 부터 어떻게 효율적인 하드웨어를 구현하는가 하는 문제이다. 현재의 VLSI 기술을 충분히 활용하기 위해서는 단순히 순차적인 계산 알고리즘을 하드웨어로 구현하는 것은 바람직하지 못하며, 알고리즘의 병렬성을 찾아내고 활용하여야 한다.

(1)번의 근사화 문제에 대해서는 디지털 필터 연구의 초기부터 많은 연구가 진행되었으며, 훌륭한 교재가 많이 있으므로, 본고에서는 주로 (2), (3)의 문제에 대해 자세한 검토를 하고자 한다. 현재 연구된 여러가지 필터의 구조와 그 특성들에 대해 살펴보고, 고속 또는 대량의 신호 처리에 특히 유용한 VLSI의

잇점을 최대한 활용할 수 있는 병렬화, 파이프라이닝화된 필터 구현 방법에 대해 자세히 설명한다. 또한, 주어진 전달 함수를 실제 하드웨어로 구현할 때 반드시 고려해야 할 FWL 영향과 그 분석 방법에 대해서 기술한다.

(1) 근사화 문제

일반적으로 필터의 규격은 주파수 응답 형태, 즉, 통과대역과 정지대역의 차단 주파수, 그리고 각 대역에서의 리플의 크기 등이다. 필터의 전달 함수가 식 (1.1)과 같이 나타내어질 때, 근사화 문제는 주어진 규격을 만족하는 필터 계수 a_i , b_i 를 구하는 것이다. 모든 a_i 가 zero일 경우 FIR 필터가 되고, 그렇지 않을 경우 IIR 필터가 된다. FIR 필터의 구현 방법은 윈도우 방법, 주파수 표본화 방법, 그리고 수리적인 알고리즘으로 최적의 해를 찾는 Minimax 설계 방법등이 있다 [1].

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} \quad (1.1)$$

IIR 필터의 설계 방법은 주로 주어진 규격으로부터 아날로그 필터(Butterworth, Chebyshev, Elliptic 필터)를 구하고, 이를 주파수 영역상에서의 여러가지 변환 방법(Impulse Invariance, Bilinear Transform, Matched z-transform)을 사용하여 디지털 필터를 구현하며, HPF(High-Pass Filter), BPF(Band-Pass Filter), BSF(Band-Stop Filter), 또는 multi-band 특성을 가진 필터는 이와 같이 원형(prototype) LPF(Low-Pass Filter)를 설계한 후 주파수 변환(Frequency Transformation) 방법을 사용하여 얻을 수 있다.

(2) 실현 문제

전달 함수 $H(z)$ 가 주어졌을 때 프로세서가 수행할 수 있는 기본적인 연산의 순서(SFG 또는 계산 알고리즘)를 찾아내는 것을 의미하며, 여기서 기본적인 연산이란 주로 덧셈과 곱셈을 말한다. 이에는 수없이 많은 해가 있으나, 궁극적으로 최소의 하드웨어와 최소의 FWL 영향을 갖는 구조를 찾아냄을 목적으로 한다. IIR 필터의 구조는 주어진 차분 방정식(difference equation)을 직접 실현한 Direct Form, 2차 필터로 분해하여 직렬 또는 병렬로 연결한 Cascade 또

는 Parallel Form, 그리고 State Variable Form(이것을 필터 구조의 한 형태로 보거나 또는 모든 필터의 표현도구로 보거나에는 이견이 있을 수 있다.)과 그 선형 변화에 의해 얻어지는 구조들, 아날로그 필터를 근간으로 얻어지는 Wave 디지털 필터 [2], 또는 Lattice, Ladder 구조 [3]등이 있다. Direct Form은 주어진 전달 함수로부터 손쉽게 얻을 수 있으며 가장 간단한 하드웨어로 구현할 수 있으나, 인접한 여러개의 폴(clustered pole)이 있는 경우에 성능 저하가 심하며, 이를 극복하기 위한 방법으로 인접한 폴을 각기 다른 section으로 구현한 Cascade, Parallel Form이 많이 쓰인다. State Variable Form은 선형 대수를 이용한 필터의 특성 분석이 용이하며 [4][5], 그중 MRON(Minimum RoundOff Noise) 구조 [6]는 특히 주파수 변환에 무관한 최소 roundoff noise를 갖는 구조이다 [7]. Wave 디지털 필터, Lattice, Ladder 구조는 하드웨어가 복잡해지는 대신, 양자화 영향에 대한 sensitivity를 감소시킨다.

(3) 구현 문제

디지털 필터의 구현 문제는 주어진 알고리즘을 현재 사용 가능한 하드웨어 기술을 사용하여 어떻게 원하는 성능을 만족시키면서 가장 경제적으로 구현할 수 있는가 하는 것이다. 기술이 발전하면서 이전에는 너무 비싸거나 너무 속도가 늦어서 사용할 수 없었던 구현 방법이 가장 좋은 방법으로 될 수도 있으며, 그 반대의 경우도 생겨나고 있는데, VLSI 기술의 발전이 이러한 변환의 주 원인으로 작용하고 있다. 이러한 하나의 예로서 DFT와 FFT를 들 수 있다. 실현과 구현 문제는 서로 밀접한 관계가 있으며, 필터 알고리즘을 선택할 때 보통 다음의 점을 고려하여 최적의 해를 찾게 된다.

1. 시스템의 전체 연산 속도(Throughput)
2. 알고리즘의 복잡도
3. Finite Word-Length 영향

필터의 기본 연산은 주로 덧셈과 곱셈으로 이루어지며(sum of products), 곱셈의 수가 알고리즘의 복잡도를 결정한다. 필터의 응용 분야에 따라서 비트 순차형(Bit Serial) 연산, 또는 비트 병렬형(Bit Parallel) 연산등을 선택하고, SDNS(Signed-Digit Number System) [8][9], RNS(Residue Number System) [10], 분산 연산 [1]등을 사용하여 최적의 성능을 위

한 구조 설계를 한다. 이외에 회전(rotation)연산을 기본 연산으로 채택하여, 효율적인 구조로 실현할 수 있는 CORDIC(Coordinate Rotation Digital Computer) 기법이 있다 [11]. (그림 1.1)에 고속 신호처리 디지털 필터의 설계과정을 보인다.

II. 디지털 필터의 병렬 구조

고속 디지털 필터는 알고리즘의 수정없이 값비싼 바이폴라(Bipolar)나 갈륨 아스나이드(GaAs) 등의 고속 소자 기술에 의존하거나, 주어진 알고리즘을 동시(concurrent) 처리에 적합한 형태로 변환하여 구현할 수 있다. 동시적 구조(Concurrent Architecture)는 전달 함수의 입/출력 특성을 변화시키지 않으면서 알고리즘의 내부 구조를 변화시켜 얻어지며,

이러한 과정을 'Structure Recasting'이라 한다 [12].

디지털 필터 알고리즘은 일반적으로 병렬성이 강한 것이 특징이다. 즉, 입력되는 각 신호 샘플에 대해 분지(branch)등이 없이 일정한 수의 연산 기능이 수행된다. 이러한 필터 알고리즘으로부터 최대한의 병렬성을 갖는 구조를 찾아내는 데는 많은 방법이 있지만, 여기서는 크게 블록 처리(block processing) 구조와 파이프라이닝 구조를 살펴보고자 한다. 병렬성을 이용하는 기본 목적은, 제한된 연산 소자의 속도를 가지고 병렬 처리를 함으로써 원하는 필터의 처리 속도를 얻고자 하는데 있다. FIR 필터는 그 알고리즘의 SFG로부터 'Cutset Theorem' [13]을 만족하도록 래치(latch)를 추가 함으로써 원하는 정도의 파이프라이닝 구조를 쉽게 구할 수 있으나 [14], 그 계산의 효율성 때문에 여러 응용 분야에서 우선적으로 고려

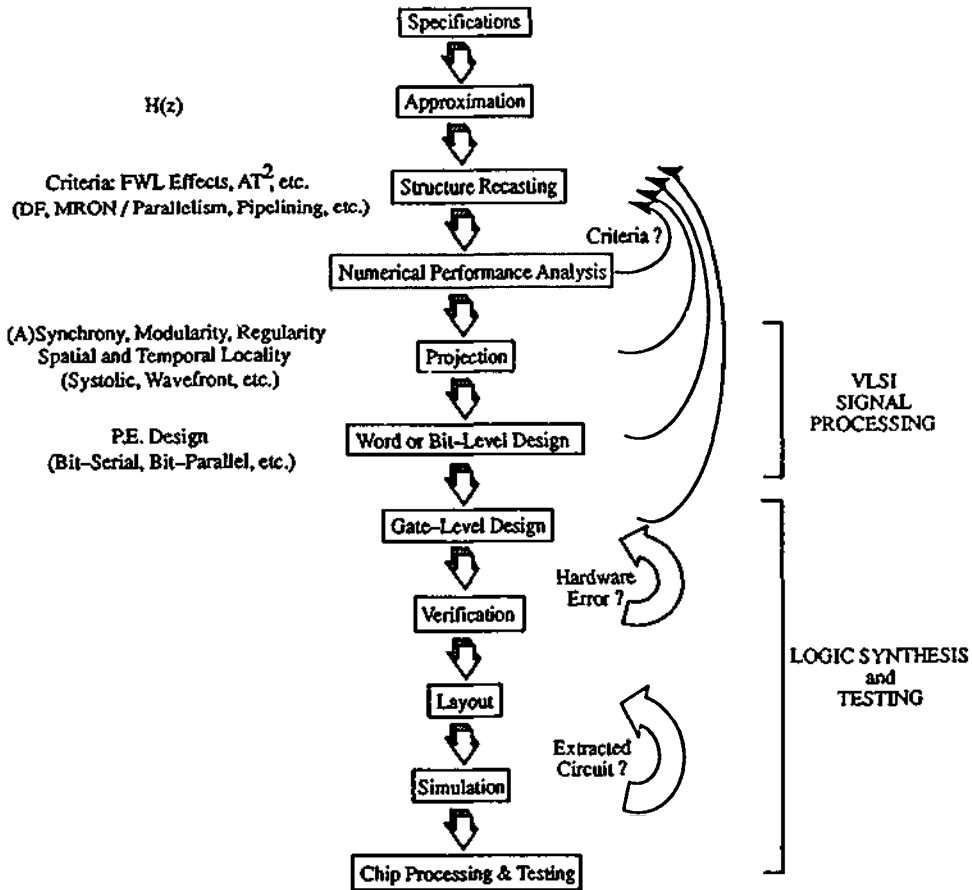


그림 1.1 고속신호처리 디지털필터 설계과정.

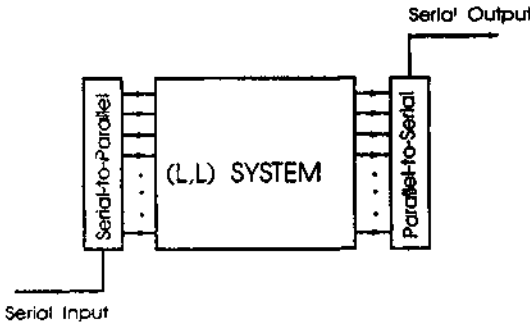


그림 2.1 블록 필터의 구조.

되는 IIR 디지털 필터는 알고리즘의 재환성(recursive) 때문에 고속으로 동작하는 회로를 구성하기가 쉽지 않다 [15]. 왜냐하면, 현재의 필터 변수나 출력 값이 입력 신호의 한 샘플 간격 동안에 계산되어야 하기 때문이다. 따라서 이절에서는 IIR 필터의 블록 처리 구조와 파이프라이닝 구조를 유도하는 방법, 그리고 그 각 구조의 특성을 알아보기로 한다.

2.1 블록 필터(Block Filter)

IIR 필터를 블록 처리에 의해 구현하는 방법은(이후 이 구조를 총칭해 블록 필터라고 한다.) Gold와 Jordan에 의해 처음 제안되었으며 [16], 그 후 유용한 병렬 처리 모델로서 많은 연구가 이루어졌다. [17]-[21]. 블록 필터의 기본 구조는 입력측에 serial-to-parallel 변환기, 출력단에 parallel-to-serial 변환기가 있고, 그 사이에 처리를 수행하는 MIMO(Multi-Input Multi-Output) 시스템이 있다(그림 2.1). 이에 대응하여 블록 처리를 하지 않는 기존 필터 구조를 SISO(Single Input Single-Output) 시스템으로도 부른다. 초기의 블록 필터 연구는 블록 convolution을 FFT를 사용하여 구현함으로써, 범용 컴퓨터에서의 계산량을 줄이는데 있었다 [17]. 그후 VLSI 기술이 발달함에 따라 MIMO 시스템은 규칙적인 구조를 갖는 프로세서 배열(array)로 구현하게 되었으며, MIMO 시스템의 병렬성을 이용함으로써 IIR 필터에 대해서도 원하는 임의의 속도를 만족시킬 수 있게 되었다. 단 이때의 구현 비용은 필요한 연산 소자의 수에 비례하게 되며, 블록 필터에서는 블록의 크기의 제곱에 비례한다. 다음에 기본적인 블록 필터의 유도 과정을 살펴본다.

원형 필터에 해당하는 차분 방정식이 다음과 같이 주어졌을 때,

$$y(k) = \sum_{i=1}^N a_i y(k-i) + \sum_{i=0}^M b_i u(k-i) \quad (2.1)$$

블록의 크기가 L인 블록 필터 알고리즘은 Look-Ahead 연산 방법을 사용하여 다음과 같이 구한다. 즉, 현재의 블록에 속한 L개의 출력 값 $y(kL)=[y(kL), y(kL+1), \dots, y(kL+L-1)]^T$ 을 이전 블록의 출력값과 현재, 그리고 이전 입력값으로 계산할 수 있는 L개의 차분 방정식을 구한다. 예를 들어 $N=2, M=1$ 인 경우,

$$y(kL) = a_{11}y(kL-1) + a_{12}y(kL-2) + b_0u(kL) + b_1u(kL-1) \quad (2.2)$$

$$y(kL+1) = a_{21}y(kL-1) + a_{22}y(kL-2) + b_{21}u(kL+1) + b_{22}u(kL) + b_{23}u(kL-1) \quad (2.3)$$

$$y(kL+L-1) = a_{L1}y(kL-1) + a_{L2}y(kL-2) + b_{L1}u(kL+L-1) + \dots + b_{L,L+1}u(kL) + b_{L,L+2}u(kL-1) \quad (2.4)$$

이 된다. 이 블록 필터의 계수 (a_{ij}, b_{ij}) 는 원형 필터 계수로부터 구해진다. 이와 같은 방법으로 계산하여 얻어진 L개의 블록 차분 방정식이 블록 필터의 알고리즘이 되며, 이를 그대로 구현하면 Direct Form 블록 필터가 된다. $N=M=2, L=4$ 인 경우의 블록 필터

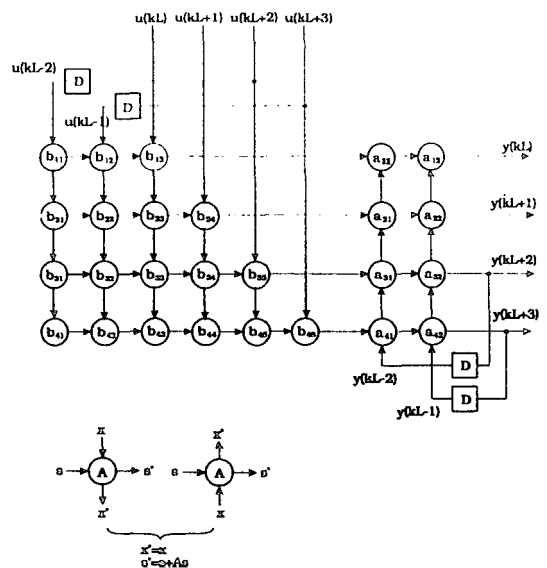


그림 2.2 Direct Form 블록 필터의 구현 예.

티구현 예를 (그림 2.2)에 보였다. MIMO 시스템 내의 각 연산 소자의 처리 속도는 원 입력 신호 속도의 $1/L$ 이 됨을 알 수 있다.

같은 방법으로 상태 공간 방정식으로 주어진 필터에 대해서도 블럭 필터를 유도할 수 있다. 원형 필터의 상태 방정식이 식(2.5)와 같이 주어졌을때,

$$\begin{aligned} x(k+1) &= Ax(k) + bu(k) \\ y(k) &= cx(k) + du(k) \end{aligned} \tag{2.5}$$

Look-Ahead 방법을 적용하여 유도한 블럭 필터에서는 L 개의 입력 신호 샘플당 한번의 상태 변수 계산을 하고, 이 추려진 (decimated) 상태 변수로부터 L 개의 출력 신호를 계산한다. 그 상태 방정식은 식(2.6), (2.7)과 같이 주어진다.

$$x(kL + L) = Ax(kL) + Bu(kL) \tag{2.6}$$

여기서,

$$\begin{aligned} A &= A^L \\ B &= [A^{L-1}b, A^{L-2}b, \dots, Ab, b] \end{aligned}$$

$$y(kL) = Cx(kL) + Du(kL). \tag{2.7}$$

여기서,

$$\begin{aligned} C &= [c', (cA)', (cA^2)', \dots, (cA^{L-1})'] \\ D &= \begin{bmatrix} d & 0 & 0 & 0 & \dots & 0 \\ cd & d & 0 & 0 & \dots & 0 \\ cAb & cb & d & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ cA^{L-2}b & cA^{L-3}b & \cdot & \dots & d \end{bmatrix} \end{aligned}$$

상태 공간 블럭 필터의 SFG를 (그림 2.3)에 보였다. 이러한 블럭 필터의 특성은 다음과 같다.

- (1) 필터의 계산 성능이 L 배 증가한다.
- (2) MIMO 시스템의 고유값(eigenvalue)은 원형 필터의 고유값의 L 제곱이 된다.
- (3) 상태 공간 블럭 필터에서는 원형 필터를 similarity 변환하여 블럭 필터를 구한 것과 블럭 필터를 바로 similarity 변환한 결과와 같다. 즉,

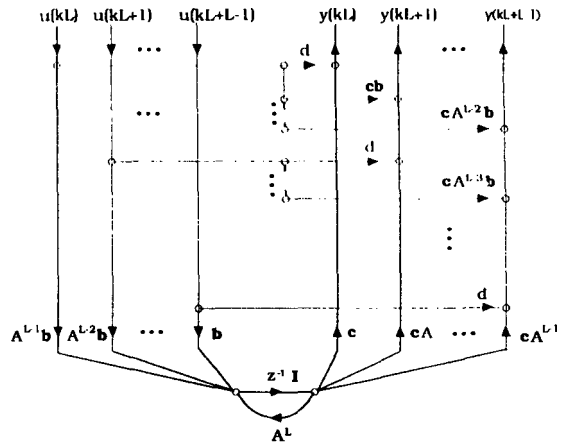


그림 2.3 상태공간 블럭 필터의 신호 흐름도(SFG).

$$(T^{-1}AT, T^{-1}b, cT, d) \rightarrow (T^{-1}AT, T^{-1}B, CT, D) \text{로 나타내어 진다.}$$

(4) MIMO 시스템은 주기 L 을 갖는 주기적 시변 필터로 해석될 수 있다. 즉, $A^l = A(k+L-1)A(k+L-2) \dots A(k+1)A(k)$ 으로 표현된다.

블럭 필터의 전달 함수는 블럭의 크기와 같은 주기를 갖는 주기 함수의 특성을 갖게 되므로 주기적 시변 필터(periodically time-varying filter)를 구현하는데 사용될 수 있다 [22]. 또한 Multi-rate 시스템, 또는 음성, 영상등의 Sub-band Coding 등에 사용되는 필터 뱅크의 구현도 블럭 필터로 용이하게 실현할 수 있다 [23].

블럭 필터의 주요한 특징중의 하나는 FWL의 영향이 대체적으로 감소한다는 것이다. 즉, SISO 시스템으로 구현한 필터에 비해 출력 샘플당 평균 roundoff 잡음이 줄어들고, 리미트 사이클(Limit Cycle)이 생길 확률이 크게 감소한다 [21]. 이는 블럭 필터의 폴의 위치가 SISO 필터의 폴 위치로부터 z -평면상의 단위원 안쪽으로 이동하기 때문이다 [24]. 그러나 협대역 필터를 Direct Form으로 구현한 블럭 필터의 경우에는, 그 필터 계수가 큰 변화를 가짐으로써 필터 계수의 양자화 영향은 오히려 증가하기도 한다 [25]. 따라서 블럭 필터의 구현은 Normal Form, 또는 NRON 구조로 구현하는 것이 바람직하다.

정형(canonic) 블럭 필터는 여러가지 잇점에도 불구하고, 블럭 크기의 제곱에 비례하여 증가하는 연산 소자의 수 때문에 블럭 크기가 매우 큰 경우에는 비효율적이 된다. 최근에 이를 개선하여 연산 소자의

수가 블록 크기에 선형적으로 증가하는 비정형(non-canonic) 구조들이 제안되었으나 [26][27], 이들 구조에서는 FWL 영향이 정형 구조에 비해 커지게 된다 [28].

2.2 파이프라이닝 필터

파이프라이닝 기술은 기본적으로 하드웨어의 복잡도를 증가시키는 대신 클럭 속도를 향상시키려는 노력의 일환으로서, 대표적으로는 ‘Look-Ahead Computation’이 있으며 그 적용 방법에 따라 ‘Clustered Look-Ahead(CLA)’[29][30]와 ‘Scattered Look-Ahead(SLA)’[31]로 나뉘어 진다. CLA 필터에서는 상태 변수 $x(n)$ 이 N 개의 연속된 과거 상태 변수 $x(n-M), x(n-M-1), \dots, x(n-M-N+1)$ 에 의하여 표시되는 반면, SLA 필터에서는 N 개의 분산된 과거 상태 변수 $x(n-M), x(n-2M), \dots, x(n-NM)$ 에 의하여 나타내어진다. 여기서 N 은 원형 필터(Prototype Filter)의 차수를, M 은 파이프라이닝 단(stage)의 수를 나타낸다. 따라서, 파이프라이닝 필터의 차수는 $M+N-1$ 보다 같거나 크게 되면, SLA에서는 NM 차의 파이프라이닝 필터를 갖게 되어 CLA에서 보다 그 차수가 커지게 된다. 하지만 SLA 필터는 근본적으로 항상 안전된 해를 갖는 반면, CLA 필터는 상쇄되는 폴과 제로의 수가 적을 때는 불안정성의 가능성을 갖고 있다. 최근의 ‘Minimum Augmented Pipelining (MAP)’ 필터 [32]에서는 최소 차수의 안정된 CLA 필터 설계 알고리즘이 제시되었다.

원형 IIR 디지털 필터의 전달 함수를 식(2.8)과 같이 나타내면

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 - \sum_{i=1}^N a_i z^{-i}}, \quad (2.8)$$

시간영역에서의 출력 샘플 $y(n)$ 은

$$y(k) = \sum_{i=1}^M a_i y(k-i) + \sum_{i=0}^N b_i u(k-i) \quad (2.9)$$

로 표시된다. CLA 파이프라이닝을 구현하기 위하여 식(2.8)의 분모와 분자에

$$G(z) = 1 + \sum_{r=1}^Q g_r z^{-r}, \quad Q \geq M-1, \quad (2.10)$$

를 곱하면, 전달함수 $H(z)$ 는

$$H(z) = \frac{D(z)}{C(z)} = \frac{B(z)G(z)}{A(z)G(z)} \quad (2.11)$$

로 된다. 다시 두개의 다항식

$$S(z) = 1 + \sum_{r=1}^{M-1} s_r z^{-r}, \quad (2.12)$$

$$U(z) = 1 + \sum_{r=1}^{Q-M} u_r z^{-r}, \quad (2.13)$$

를

$$G(z) = S(z) + z^{-M}U(z) \quad (2.14)$$

로 되도록 정의하면, $D(z)$ 는

$$D(z) = B(z)S(z) + z^{-M}B(z)U(z) \quad (2.15)$$

로 표시된다. 여기서 $S(z)$ 는 해석적으로 바로 구해지며, 다항식 $U(z)$ 의 계수 u_r 의 최적값은 MAP CLA 알고리즘의 공간 탐색(Space Searching) 방법에 의하여 얻어진다. 하지만 이러한 MAP CLA 알고리즘은 국부적으로만 최적화된 파라미터를 구하므로, M 이 큰 High-Q LPF 또는 High-Q HPF의 경우 그 최적성이 문제가 된다. 여기서 탐색 간격 K 가 중요한 역할을 하게 되며, 이러한 MAP CLA 필터의 최적성과 성능 분석, 그리고 원형 필터나 SLA 필터와의 성능 비교는 설계시 고려해야할 주요 요인이 된다 [33].

SLA 알고리즘은 [31]에서 최초로 IIR 디지털 필터에 응용되었으며, CLA 알고리즘과는 달리 기본적으로 안정성을 유지한다.(그림 2.4)에 $M=8$ 의 파이프라이닝단을 가진 SLA 필터를 보인다. 여기서 $\bar{e}_1(k), \bar{e}_2(k), \bar{e}_3(k)$ 는 모두 양자화에 의한 잡음원(Noise Source)을 나타낸다. 안정성을 유지하며 루프 파이프라이닝단의 M 의 로그에 비례하는 복잡도를 갖는 상태 공간 IIR 필터의 분해 파이프라이닝(Decomposition-based Pipelining)은

$$x(k+M) = A^M x(k) + \sum_{i=1}^M A^i b u(k+M-1-i) \quad (2.16)$$

로 표현되며, 샘플링 주기의 M 배인 ‘MAC(Multi-

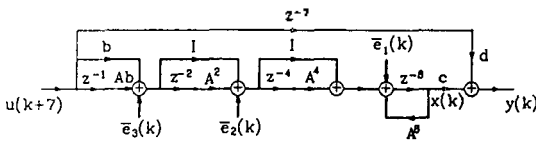


그림 2.4 M=8의 파이프라이닝단을 갖는 SLA 필터.

ply-Accumulator)’ 잠복기(Latency)를 갖는다. 따라서 원형 IIR 디지털 필터의 전달함수 H(z)의 (r, θ)에 위치하는 각 폴에 대해, SLA 파이프라이닝 필터는 (r, θ + i2π/M), i=1, 2, ..., (M-1), 에 폴들을 갖는다. 등가의 압/출력 관계를 유지하기 위하여는 첨가된 폴들에 의한 영향을 상쇄시키기 위해 N(M-1)개의 제로를 필요로 하며, 계산의 복잡성을 줄이기 위해서 분자를 N개의 계수를 가진 log₂M개의 다항식으로 인수 분해한다. 따라서, 폴만을 가진 Direct Form 필터의 경우는 N(1 + log₂M)의 곱셈을 필요로 한다. 가장 간단한 1차 IIR 필터의 경우, 전달함수 H(z)를

$$H(z) = \frac{bz^{-1}}{1-az^{-1}} \tag{2.17}$$

라 하면, 이는

$$H(z) = \frac{bz^{-1} \prod_{i=0}^{\log_2 M - 1} (1 + a^{2^i} z^{-2^i})}{1 - a^M z^{-M}} \tag{2.18}$$

로 되어, log₂M의 파이프라이닝단을 가진 SLA 필터로 구현된다.

SLA 파이프라이닝 필터의 FWL 효과는 주로 Q 값에 의존하는데, Q 값이 적은 경우는 M 값이 적을수록, Q 값이 큰 경우는 M 값이 클수록 대체적으로 보다 나은 FWL 효과를 갖는다고 말할 수 있다. Q 값이 큰 경우는 θ = ±in/M, i=1, 2, ..., M-1. 주위에서 밀집된 폴에 의해 아주 나쁜 영향을 갖는다. SLA 파이프라이닝 필터의 모든 성능 분석을 포함한 최적화는 [34], [35]에서 이루어졌다.

III. VLSI 구현

이제껏 디지털 필터는 프로그램 가능한 디지털 시그널 프로세서(Digital Signal Processor)를 이용하

여 주로 구현되어 왔으나, 서브 마이크로 단위의 VLSI 공정/설계 기술과 스탠다드 셀(Standard Cell)의 보편화에 힘입어 커스텀 IC(Custom IC)상에서의 구현도 하나의 실득력있는 대안으로 등장하게 되었다. 이러한 향상된 기술의 잇점을 누리기 위하여는 디지털 필터의 VLSI 설계시, 다음과 같은 점이 충분히 고려되어야 한다 [36][37].

1. Synchrony
2. Modularity와 Regularity
3. Spatial Locality와 Temporal Locality
4. Pipelinability

이는 동시에 동기 시스템(Synchronous System) 구조의 일종인 시스톨릭 배열(Systolic Array)의 성질이기도 하다[38]. 즉, 시스톨릭 배열 구조는 입/출력 제한 계산(I/O-bound Computation)과 산술 제한 계산(Compute-bound Computation) 중 후자에 의한 제한을 해결하기 위한 방법중의 하나이며, 그 외에 연산 소자들을 2차원으로 배열할때의 클럭 스쿠(Clock Skew) 문제를 비동기적으로 해결하기위한 웨이브 프론트 배열(Wavefront Array) 구조가 있다.

하부 설계과정에 해당되는 PE의 설계는, 요구되는 여러사양에 따라 비트 순차형(Bit Serial)과 비트 병렬형(Bit Parallel) 구조로 다시 세분되며, 이중 비트 순차형 구조는 모든 데이터 전송과 연산이 1비트의 데이터 단위로 순차적으로 이루어지기 때문에 저속이나 비교적 간단한 하드웨어로 이루어져야 하는 시스템에 적합하다 [39]. 반면에 비트 병렬 구조는 한 클럭 사이클 안에 워드내의 모든 입력 비트를 처리하여야 하기 때문에, 구현 면적이 커지는 한이 있더라도 고속의 데이터를 처리하여야 하는 곳에 주로 이용된다[40][41].

그러나 파이프라이닝을 IIR 디지털 필터에 도입하는 것은 회로의 동작이 ‘Cutset Theorem’을 정면으로 위반하는 것이 될 뿐 아니라 [13], IIR 루프내의 파이프라이닝에 의한 잠복기(Latency)의 증가로 전체 처리율(Throughput Rate)을 현저히 감소시킨다. 처리율의 감소는 일반적인 산술 연산방식을 사용할 경우, LSB로부터 MSB로의 캐리 전파(Carry Propagation)에 주로 기인한다. 이러한 단점은 비트 레벨 구조(Bit-level Architecture)를 MSD로부터 계산결과가 유효한 여유 부호 디지털 연산 방식(Redundant Signed-Digit Number System) [42]을 사용하여 해결할 수 있다. 비트 순차형 구조를 SDNS와 혼용하는

디지털 순차형(Digit Serial) 방식인 온라인 연산(On-line Arithmetic)은 디지털 레벨에서의 순차연산을 행하기 때문에, 여러 클럭 주기가 필요한 파이프라인 지연이 단지 1 또는 2 워드 지연만으로 감소되어 IIR 디지털 필터에서의 잠복기 문제를 해결하고 연산 결과가 모두 디지털 순차적으로 전송되기 때문에 기능 블록들의 연결 복잡도 또한 현저히 감소시키는 하나의 방법으로 제시될 수는 있으나 [9], 본래의 비트 직렬의 성질때문에 초고속 시스템에는 적합치 않다. 또다른 단점으로는 정규화된 연산만을 허용하는 이유로, MSD가 항상 0 값인가져서는 안된다는 것이다. 반면에 비트 병렬 구조를 SDNS와 혼용하는 디지털 병렬(Digit Parallel) 방식은 극히 높은 처리율을 요구하는 IIR 디지털 필터에 적합하다 [43] [44]. 수개의 파이프라이닝단을 지니고 있는 SLA 필터의 잠복기는 디지털 병렬 방식의 잠복기와 근원적으로 일치하기 때문에, SLA 필터의 디지털 병렬 방식에 의한 구현은 가장 최적의 방법중 하나라고 할 수 있다. 다음에 파이프라이닝 SLA 필터의 디지털 병렬 방식에 의한 구현에 대하여 상세히 설명한다.

SDNS는 1961년 Avizienis에 의해 사칙연산과 같은 기본 산술 연산에서의 캐리 전달 체인(Carry Propagation Chain)을 제거하기 위하여 제안되었다. SDNS는 각 디지털의 값이 양수와 음수값을 모두 취할 수 있다는 점에서 일반 수체계와 다르며, r진(Radix-r) SDN Z는 n+m+1개의 디지털 $z_i, i = -n, \dots, -1, 1, \dots, m$ 에 의해 다음과 같이 표시된다.

$$z = \sum_{i=-n}^m z_i r^{-i} \quad (3.1)$$

r > 2인 경우의 최소 여유 집합(Minimally Redundant Set)은 적어도 r + 2개, 최대 여유 집합(Maximally Redundant Set)은 2r-1개 까지의 원소를 갖으며, n=m인 경우는 다음과 같다.

$$\frac{1}{2} (r_0 + 1) \leq n \leq r_0 - 1, \text{ 홀수 진수 } r_0 \geq 3 \quad (3.2)$$

$$\frac{1}{2} r_e + 1 \leq n \leq r_e - 1, \text{ 짝수 진수 } r_e \geq 4$$

즉, SDNS에서는 어떤 일정한 값을 한가지 이상의 방법으로 표현할 수 있으며, 이러한 여유도를 이용함으로써 가산과 승산의 경우 각디지털의 연산결과가

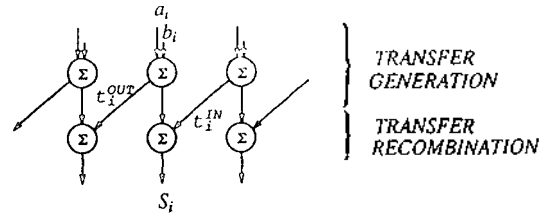


그림 3.1 병렬 r진 SDNS 가산기.

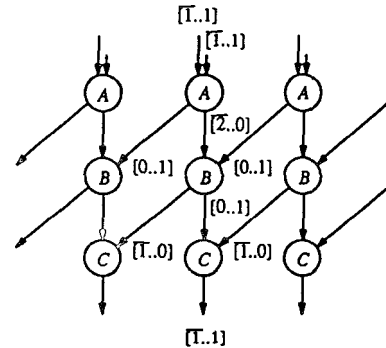


그림 3.2 병렬 2진 SDNS 가산기.

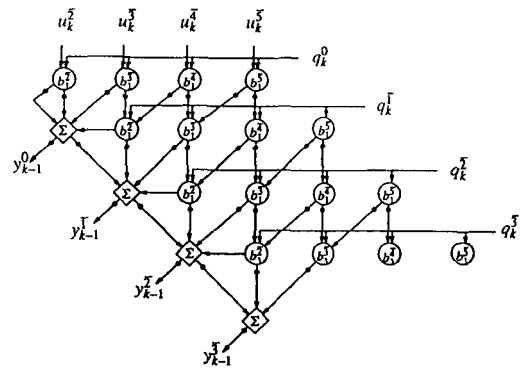


그림 3.3 저잠복 캐리절약 승산기.

인접한 수개의 디지털에만 영향을 받도록 시스템을 구성할 수 있다. 바로 이러한 점이 병렬 연산을 수행할 경우 LSD로부터 MSD로의 전체적인 캐리 전송을 불필요하게 하여, 전체연산시간이 레지스터의 워드 길이와는 무관하게 고정된 작은 값으로 되는 MSD 우선 연산(MSD-first Operation)이 된다.

진정한 의미의 병렬 가산이라 함은 t_{out} 이 t_{in} 과 무관한 일정한 값을 갖게됨을 말하며, 이는 가산과정을 (그림 3.1)에서와 같이 전송 발생부(Transfer Generation)와 전송 재결합부(Transfer Recombination)의 두단계로 나누어 수행함으로써 이루어진다. 여기서 전송 디지털트는 캐리 디지털트(Carry Digit)나 바로우 디지털트(Borrow Digit)의 역할을 하여, 양/음수의 값을 모두 갖는다. 전송 디지털트가 셀의 모든 열(Row)을 이동하는 일반 가산기와는 달리, 이러한 구조의 가산기에서는 입력전송 디지털트와 전송 생성단으로부터의 값이 결합되어 출력 합을 형성한다. 하지만 이러한 구조는 2진 경우에는 SDNS에 의한 가능 여유 디지털트 집합이 $\{1, 0, 1\}$ 로 유일하게 되어, 1개의 단에서 전송 재결합을 수행할 수 없게 된다. 따라서 2개 디지털트까지 전송을 허락하는 3개의 레벨을 갖는 (그림 3.2)와 같은 수정된 가산기가 필요하게 된다. 여기서 각 최종 합 의 디지털트는 3개의 인접한 디지털트 쌍에만 의존하게 되어, 일반 가산기에서의 수평 캐리 전송 대신 대각선 방향의 효율적인 캐리 전송이 이루어

어져, 일반 수체계로부터 SDNS로의 전환이 긴 캐리 전송 체인의 도움없이 간단히 수행된다.

캐리 절약 코어(Carry-save Core)를 사용하여 디지털 병렬, 시간축 경사, MSD 우선으로된 구조를 갖는 저잠복기 승산기(Low-latency Multiplier)를 (그림 3.3)에 보인다. 여기서의 검은점은 파이프라이닝용 래치를 나타내며, 수평 방향으로의 파이프라이닝은 오히려 전체적인 잠복기를 늘릴 수 있기 때문에 고려치 않는다.

시간축 경사를 바로 잡는 동시에 SDNS로부터 일반 수체계로 전환해주는 시스템릭 구조를(그림 3.4)에 보인다. 이러한 전환은 IIR 루프 밖에서 일어나기 때문에 필터의 처리율을 그대로 유지하면서 원하는 만큼의 파이프라이닝을 도입할 수 있다.

SLA 방식은 일반 수체계를 사용해서 설계될 수도 있으나, 이 경우 반복회수(Number of Iterations)가 파이프라이닝 레벨에 의존하게되고 따라서 레지스터의 워드 길이에 큰 영향을 받게 되어 IIR 루프에서 필요한 값 외의 모든 중간값을 계산하기 위한 별도의 MAC(Multiply-Accumulator)에 의해 하드웨어의 복잡도가 상당히 증가된다. 이러한 이유로 SLA 방식을 MSD 우선 구조와 결합하여 사용하게 되면 IIR 루프내의 잠복기가 레지스터의 워드길이와는 무관한 2 클럭 사이클로 되어, 완벽히 파이프라이닝된 2차 IIR 디지털트 필터 구조를 얻게 된다. 따라서 레지스터의 워드길이와는 무관하게 연산을 수행하는 한정된 수의 승산기로 이루어졌으며, 매 클럭 사이클마다 입력 샘플 데이터를 받아들이는 고속 IIR 디지털트 필터를 얻게된다.

M=2, 즉 2개의 루프 파이프라이닝 단을 가진 경우의 풀만으로 구성된 2차 Direct Form SLA 필터의 전달함수는 다음과 같이 된다.

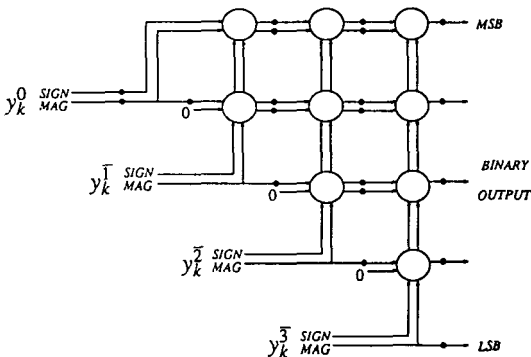


그림 3.4 SDNS를 일반 수체계로 전환해주는 병렬 변환기.

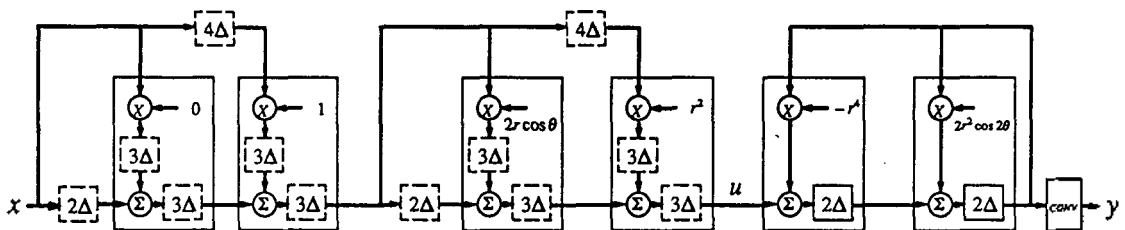


그림 3.5 M=2의 파이프라이닝단을 갖는 2차 SLA 필터의 시스템릭 구조.

$$H(z) = \frac{1 + 2r(\cos \theta)z^{-1} + r^2z^{-2}}{1 - 2r^2(\cos 2\theta)z^{-2} + r^4z^{-4}} \quad (3.3)$$

따라서

$$y(k) = u(k) + 2r(\cos \theta)u(k-1) + r^2u(k-2) + 2r^2 \cdot (\cos 2\theta)y(k-2) - r^4y(k-4) \quad (3.4)$$

M=2인 경우는 분해(Decomposed)와 비분해(Non-decomposed) SLA 필터가 동일하게 되며, 이러한 구조의 SLA 필터를(그림 3.5)에 보인다. 여기서 LSD절삭(Truncation)은 승산기에서는 'Single-length'의 레지스터에 의해, 가산기에서는 'Double-length'의 레지스터에 의해 각기 모델링된다. 이러한 디지털 필터의 처리량은 80 Msamples/sec로 추정되며, 작은 값의 M을 갖는 SLA 필터는 디지털 병렬, MSD 우선 구조와 결합되어 사용될때 효율적인 최대 처리량을 갖게 된다. 그러나, 레지스터 길이가 B인 일반 수체계를 사용하여 구현하였을 경우와 대비하여 이러한 고속 신호 처리 구조에서 같은 수준의 트렁케이션 오차를 유지하기 위해서는 B-2 만큼의 부가 레지스터가 요구되며, 또한 신호 동작 영역(Dynamic Range)이 줄어들게 되고, 오버 플로우(Overflow) 문제를 해결하기 위해 포화 연산(Saturation Arithmetic)을 사용하여야 하는 등의 단점이 있다.

IV. Finite Word-Length 효과의 분석

대부분의 실시간 처리 시스템에 사용되는 디지털 필터의 구현은 하드웨어가 훨씬 단순한 고정 소수점 연산(Fixed-point Arithmetic)을 사용하는데, 이에 따라 연산 오차가 생기고 필터의 동작이 영향을 받는다[45]. 최근에는 부동 소수점 연산(Floating-point Arithmetic)을 채택한 소자도 많이 사용되며, 이 경우의 FWL 효과는 고정 소수점 연산의 경우에 비해 필터 설계에 큰 영향을 미치지 않는다[46]. 다음은 디지털 필터 구현시 수반되는 오차의 원인들이다.

1. 입력 신호의 A/D 변환
2. 필터 계수의 양자화
3. 각 곱셈 연산후에 행해지는 반올림(Roundoff) 또는 절삭(Truncation)
4. 덧셈 연산에서 일어나는 오버플로우(Overflow)

이중 입력 신호의 A/D 변환에 의한 오차는 비상관(uncorrelated) 백색 잡음으로 모델링되며 그 영향은 필터의 구조에 무관하지만, 나머지 3가지 오차의 영향은 필터의 구조에 크게 좌우된다. 필터 계수의 양자화에 따라 전달 함수가 실제 규격으로부터 벗어나고, 연산후의 반올림 또는 절삭에 의해서는 양자화 리미트 사이클(Quantization Limit Cycles)이 야기되고 그 결과로 필터 출력의 SNR이 나빠진다. 또한 오버플로우에 의해서는 오버플로우 리미트 사이클(Overflow Limit Cycles)이 일어날 수 있다.

4.1 계수 양자화 잡음(Coefficient Quantization Noise)

상태 공간식으로 표현되는 필터 구조는 선형 대수 이론을 이용한 필터의 성능 분석이 용이하고, Similarity Transformation에 의한 최적 구조의 합성등이 가능하여, 활발히 연구되어 왔다[4][5]. 특히, 필터 계수 오차에 대한 전달 함수의 sensitivity 분석 방법에 대한 연구가 행해졌으며, 그중 대표적인 것으로 Kawamata가 제안한 deterministic 분석과 통계적인 분석 방법이 있다[6]. 식 (2.5)와 같이 상태 방정식 표현으로 주어진 필터에 대해 실제 구현된 필터의 상태 방정식을

$$x(k+1) = (A + \Delta A)x(k) + (b + \Delta b)u(k) \quad (4.1)$$

$$y(k) = (c + \Delta c)x(k) + (d + \Delta d)u(k)$$

로 나타내면(여기서 $\Delta A, \Delta b, \Delta c, \Delta d$ 는 deterministic 분석에 의하여는 상수값으로, 통계적 분석 방법에 의하여는 상관성이 없는 random variable로 모델링한다.), 통계적 분석 방법에 의하여 얻어진 전달 함수의 sensitivity 근사식은 다음과 같다.

$$S = tr[K]tr[W] + tr[W] + tr[K] + 1 \quad (4.2)$$

여기서 $tr[\cdot]$ 은 matrix의 trace를 나타내고, K와 W는 각각 필터 구조의 Controllability와 Observability Gramian이다. 이 식에 의하여 필터 sensitivity가 필터 구조의 영향을 받음을 알 수 있으며, Similarity Transformation에 의하여 최소 sensitivity를 갖는 필터를 합성할 수 있다.

4.2 Roundoff Noise

필터의 실제 구현시에는 내부에 유한 길이의 레지

스터, 누산기(Accumulator)등을 사용하므로, 곱셈 연산 후에는 항상 roundoff 연산이 필요하다. 이에 의하여 발생하는 오차를 'Roundoff Noise'라 한다. 특히 출력이 제한되는 IIR 필터에서는 이 오차의 누적에 의한 영향을 잘 분석하여야 한다. 또한 IIR 필터의 내부 summing node는 오버플로우가 일어나지 않도록 scaling 되어야 하며, 이 scaling과 roundoff noise는 밀접한 관계에 있다. 유한 에너지를 갖는 입력 신호에 대해서는 l_2 -scaling 방법이 보통 사용되며 [47], 상태 공간 방정식으로 표현되는 필터에서의 roundoff noise source는 가우시안 백색 잡음에 의해 효과적으로 모델링 된다. 다음식에서 $\varepsilon(k)$ 와 $\eta(k)$ 가 roundoff noise source이다.

$$\begin{aligned} x(k+1) &= Ax(k) + bu(k) + \varepsilon(k) \\ y(k) &= cx(k) + du(k) + \eta(k) \end{aligned} \quad (4.3)$$

필터의 scaling을 고려한 출력 전력 잡음은 다음과 같이 얻어진다[47].

$$\sigma^2 = \delta^2 \sigma_\varepsilon^2 \sum_{i=1}^N K_{ii} W_{ii} \quad (4.4)$$

여기서 δ 는 scaling 변수로서 통상 3~5의 값을 취한다. 역시 이 분석 결과로부터 최소 roundoff noise 구조를 얻을 수 있으며, MRON 구조는 주파수 변환에 의한 roundoff noise의 변화가 없으므로 협대역 필터를 구성하는데 유리하다[7]. 단지 상태 공간에서 제한 행렬 A가 full matrix로 되는 구조와 비교되었을 때만, MRON 구조가 최소 양자화 잡음을 갖는다고 말할 수 있다[48][49].

4.3 리미트 사이클(Limit Cycles)

리미트 사이클 현상은 FWL 레지스터로 구현된 IIR 디지털 필터에서 발생하며, 이는 MSB 또는 LSB에 미치는 영향에 따라 오버플로우 리미트 사이클(Overflow Limit Cycles)과 양자화 리미트 사이클(Quantization Limit Cycles)로 분류된다. 신호의 LSB에만 영향을 주는 양자화 리미트 사이클에 비해 오버플로우 리미트 사이클은 보다 심각한 오차를 유발하며, 심한 경우는 필터의 출력이 동작 영역의 최대값과 최소값 사이를 진동하게 된다. 오버플로우 리미트 사이클은 시변 비선형성을 가진 시스템에 제로

입력을 가한 경우의 특성과 동일하게 볼 수 있다. 양자화 리미트 사이클은 제로 입력과 상수 입력 양자화 리미트 사이클로 다시 세분된다. 오버 플로우 리미트 사이클과 제로 입력 양자화 리미트 사이클은 제로 입력 리미트 사이클로 함께 분류되나, 상수 입력 양자화 리미트 사이클은 내부 양자화 오차가 높은 상호 연관성(correlation)을 갖기 때문에 제로 입력 리미트 사이클의 경우에 비해 훨씬 해석이 복잡하다.

리미트 사이클을 예방하거나 감소시키기 위한 여러가지의 대책이 연구되었으며[KyungHi CAS], 기존의 1, 2차 필터에서만 해석가능하였던 오버플로우 현상이 N차의 필터에 대해 확장되었다[50]. 디지털 필터에서의 리미트 사이클(Limit Cycle)을 카이오스(Chaos)에 의한 프랙탈(Fractal) 현상으로 파악하려는 시도도 행하여지고 있다[51].

4.4 Powers-of-Two 계수 디지털 필터

FIR 필터에서의 FWL 효과는 IIR 필터에서와는 달리 비교적 문제가 없으나, 보다 나은 FWL 효과를 얻기 위하여 'MILP(Mixed Integer Linear Programming)'를 이용하여 2^x 로 표현되는 계수를 구할 수 있다[52]. 이러한 형태의 FIR 필터는 계수의 성질상 승산기를 필요로 하지않고 가산기와 천이기(Shifter)만에 의해 구현될 수 있기 때문에, 보다 빨리 연산을 수행할 수 있을 뿐만 아니라 구현 면적 또한 감소된다.

우선 'infinite precision'의 계수를 구한후, 그중 이산 계수가 아닌 계수에 대해 'Branch와 Bottom Tree'를 이용해 계수 공간을 탐색하여 'suboptimum'의 이산 계수를 구하고, 그중에서 최적 계수를 찾는다. 하지만 Remez 교환 알고리즘에 의해 얻어진 'infinite precision' 계수는, 계수값 자체에 제한을 두어 구할수가 없기 때문에 이 방법의 사용이 불가능하다. 따라서 $h(n)$ 은

$$h(n) = \sum_{i=1}^j S_i(n) 2^{g_i(n)} \quad (4.5)$$

으로 나타내어지며, 여기서 $S_i(n) = -1, 0, 1$, 정수 $g_i(n) \geq -10$ 로 된다. j 는 계수의 정확성과 계산하는 시간과의 절충 변수이다.

V. 결 론

본 고에서는 그 알고리즘의 재현성 실시간 실현이 쉽지 않은 고속의 IIR 디지털 필터설계 방법에 대하여 고찰하였다. 급속히 발달하는 VLSI 기술의 잇점을 최대한으로 활용하기 위한 병렬 및 파이프라이닝 필터 구조를 유도하는 방법과 각 구조들의 특성에 대하여 기술하였으며, 실제 하드웨어를 구현하는데 필요한 비트 레벨 구조와 SDNS를 이용한 고속 연산 소자의 구성 방법을 상술하였다. 그 예로서 설계한 SLA 파이프라이닝 필터의 구조를 보였다. 디지털 필터의 동작은 그 구현된 구조에 따라 FWL 영향이 크게 달라지므로, 새로운 구조의 제안시에는 반드시 이의 분석이 필요하다. 본고에서는 대표적인 FWL 영향의 분석 방법과 최적 합성 방법등을 소개하였다. 디지털 필터를 포함한 많은 디지털 신호 처리 알고리즘이 내재된 병렬성을 갖고 있으며, 이를 효율적으로 이용하는 연구가 앞으로 계속되어야 할 것이다.

참 고 문 헌

1. L.B. Jackson, *Digital Filters and Signal Processing*, Kluwer Academic Publishers, Norwell, MA, 1989.
2. Fettweis, A., "Wave Digital Filters: Theory and Practice," *Proc. IEEE*, vol. 74, pp. 270-327, Feb. 1986.
3. A.H. Gray, Jr., and J.D. Markel, "A Normalized digital filter structure," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp.268-277, 1975
4. V.Tavsanoglu and Thiele, "Optimal Design of State-Space Digital Filters by Simultaneous Minimization of Sensitivity and Roundoff Noise," *IEEE Trans. Circuits Syst.*, vol.CAS-31, pp.884-888, Oct. 1984.
5. M.Kawamata and T. Higuchi, "A Unified Approach to Optimal Synthesis of Fixed-Point State-Space Digital Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol.ASSP-33, pp.911-920, Aug. 1985.
6. C.T.Mullis and R.A.Roberts, "Synthesis of minimum roundoff noise fixed point filters," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp.551-562, Sep. 1976.
7. C.T. Mullis and R.A. Roberts, "Roundoff noise in Digital Filters: Frequency Transformations and Invariants," *IEEE Trans. Acoust., Speech, Signal Processing*, vol.ASSP-24, pp.538-550, Dec. 1976.
8. M.D.Ercegovac, "On-line arithmetic: An overview," in *Proc. SPIE*, pp.86-93, 1984.
9. M.D. Ercegovac and T. Lang, "On-line arithmetic: A design methodology and applications in digital signal processing," *VLSI Signal Processing III*, IEEE Press, New York, NY, 1988.
10. S.C. Knowles and J.G.McWhirter, "The application of redundant number systems to the design of VLSI recursive filters," *Mathematics in Signal Processing II*, Clarendon Press, Oxford, UK, 1990.
11. Yu Heun Hu, "CORDIC-Based VLSI Architectures for Digital Signal Processing," *IEEE Signal Processing Magazine*, pp.17-35, Jul. 1992.
12. Y.C. Lim and B. Liu, "Pipelined recursive filter with minimum order augmentation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 40, pp.1643-1651, Jul. 1992.
13. H.T. Kung and M.S. Lam, "Fault-tolerant VLSI systolic arrays and two-level pipelining," in *Proc. SPIE*, pp. 143-158, 1983.
14. Sun-Yuan Kung, "On Supercomputing with Systolic/Wavefront Array Processors," *Proceedings of the IEEE*, vol. 72, July 1984.
15. M.Renfors and Y. Neuvo, "The Maximum Sampling Rate for Digital Filters Under Hardware Speed Constraints," *IEEE Trans. Circuits and Systems*, vol. CAS-28, pp. 196-202, Mar. 1981.
16. B. Gold and K.L.Jordan Jr., "A note on digital filter synthesis," *Proc. IEEE*, vol. 56, pp. 1717-1718, Oct. 1968.
17. C.S.Burrus, "Block Implementation of Digital Filters," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 697-701, Nov. 1971.
18. C.L. Nikias, "Fast block data processing via a new IIR digital filter structure," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, Aug. 1984.
19. Cheng-Wen Wu and P. R. Cappello, "Application-Specific CAD of VLSI Second-Order Sections," *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-36, pp. 813-825, May 1988.
20. Theo A.C.M. Classen, Wolfgang F.G.Meckenbrauker, and J.B.H. Peek, "Effects of Quantization and Overflow in Recursive Digital Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 517-529, Dec. 1976.

21. C.W.Barnes and S. Shinnaka, "Finite Word Effects in Block-State Realizations of Fixed-Point Digital Filters," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 345-349, May 1980.
22. R.A.Meyer and C.S.Burrus, "A Unified Analysis of Multirate and Periodically Time-Varying Digital Filters," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 162-168, Mar. 1975.
23. P.P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proc. of the IEEE*, vol. 78, pp. 56-93, Jan. 1990.
24. S.K.Mitra and R. Gnanasekaran, "Block implementation of Recursive digital filters-New structures and properties," *IEEE Trans. Syst.*, vol. CAS-25, pp. 200-207, Apr. 1978.
25. Jinwoong Kim and W.G.Bliss, "Analysis of Coefficient Quantization Noise in Block Digital Filters," *Proceedings of Asilomar Conf. on Signals. & Computers*, Oct. 26-28, 1992.
26. C.W.Wu and P.R. Cappello, "Application-specific CAD of VLSI second-order sections," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 813-825, May 1988.
27. K. K. Parhi and D. G. Messerschmitt, "Pipeline Interleaving and Parallelism in Recursive Digital Filters-Part II: Pipelined Incremental Block Filtering," *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-37, pp. 1118-1134, Jul. 1989.
28. Jinwoong Kim and W.G.Bliss, "Numerical Performance of Non-Canonic Block State Digital Filters," *Proceedings of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1993.
29. P.M.Kogge and H.S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Computers*, vol. 22, pp. 786-793, Aug. 1973.
30. Y.C.Lim and B.Liu, "Pipelined recursive filter with minimum order augmentation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 40, pp. 1643-1651, Jul. 1992.
31. K. K. Parhi and D. G. Messerschmitt, "Pipeline Interleaving and Parallelism in Recursive Digital Filters-Part I: Pipelining Using Scattered Look-Ahead Decomposition," *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-37, pp. 1099-1117, Jul. 1989.
32. Y.C.Lim and B.Liu, "Pipelined recursive filter with minimum order augmentation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 40, pp. 1643-1651, Jul. 1992.
33. K.H.Chang and Jinwoong Kim, "Benchmark test of pipelined recursive digital filters-Clustered look-ahead and scattered techniques," in *Proc. IEEE Int. Symp. on Circuits and Systems*, May 1994, to be published.
34. K.H. Chang and W.G. Bliss, "Finite word-length effects of pipelined recursive digital filters," *IEEE Trans. Signal Processing*, to be published.
35. K.H. Chang and W.G.Bliss, "Limit cycle behavior of pipelined recursive digital filters," *IEEE Trans. Circuits Syst.*, to be published.
36. H.T. Kung, "Why systolic architectures," *IEEE Computer*, vol. 15, pp. 37-46, Jan. 1982.
37. S.Y. Kung, *VLSI Array Processors*. Prentice-Hall, Englewood Cliffs, NJ, 1988.
38. C.E.Leiserson and J.B.Saxe, "Optimizing synchronous systems," in *Proc. 22nd Ann. Symp. Foundations of Computer Science*, pp. 23-36, 1981
39. R. Hartley and P. Corbett, "Digit-serial processing techniques," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 707-719, Jun. 1990.
40. M. Hatamian and G.L. Cash, "Parallel bit-level pipelined VLSI designs for high-speed signal processing," *IEEE Proceedings*, vol. 75, pp. 1192-1202, Sep. 1987.
41. P.R. Cappello and C.W.Wu, "Computer-aided design of VLSI FIR filters," *IEEE Proceedings*, vol. 75, pp. 1260-1271, Sep. 1987.
42. A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electronic Computers*, vol. 10, pp.389-400, Sep. 1961.
43. R.F. Woods, S.C. Knowles, J.V.McCanny and J.G. McWhirter, "Systolic IIR filters with bit level pipelining," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1988, pp. 2072-2075.
44. B.P.McGovern, R.F. Woods and J.V.McCanny, "The systematic design of high performance digital filters," in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Mar. 1992, pp. V609-V612.
45. A. V. Oppenheim and C. J. Weinstein, "Effects of Finite Register Length in Digital Filtering and Fast Fourier Transform," *Proceedings of IEEE*, vol. 60, pp. 957-976, Aug. 1972.

46. A. Lacroix, "Floating-point signal processing-arithmetic, roundoff-noise and limit cycles," in *Proc. IEEE Int. Symp. on Circuits and Systems*, Espoo, Finland, pp. 2023-2030, June 7-9, 1988.
47. R.A. Roberts and C.T. Mullis, *Digital Signal Processing*, Addison-Wesley, Reading, MA, 1987.
48. K.H. Chang, "Direct form vs. minimum roundoff noise structure-Part I : Roundoff noise," *KITE J. Electronics Eng.*, vol. 4, pp. 37-42, Oct. 1993.
49. K.H. Chang, "Direct form vs. minimum roundoff noise structure-Part II : Statistical coefficient quantization noise," *KITE J. Electronics Eng.*, vol. 4, pp. 43-48, Oct. 1993.
50. K.H. Chang and C.G. Park, "Explicit sufficient condition for overflow-free design in n-th order digital filters," *IEEE Trans. Signal Processing*, being reviewed.
51. L.O. Chua and T. Lin, "Chaos in digital filters," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 648-658, Jun. 1988.
52. 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, Jun. 1983.

▲金 鎮 雄

1981년 2월 : 서울대학교 전자공학과(학사)
 1983년 2월 : 서울대학교 전자공학과(석사)
 1993년 8월 : Texas A&M University 전기공학과
 (박사)
 1983년~현재 : 한국전자통신연구소 선임연구원
 ※주관심 분야 : Structures and Numerical Performance Analysis of Digital Filter, VLSI Signal Processing, Image and Speech Processing

▲張 慶 熙

1985년 2월 : 연세대학교 전자공학과(학사)
 1987년 2월 : 연세대학교 전자공학과(석사)
 1992년 8월 : Texas A&M University 전기공학과
 (박사)
 1989년~1990년 : 삼성 종합기술원 주임연구원
 1990년~1992년 : Texas A&M University VLSI Signal Processing Lab. R.A.
 1992년~현재 : 한국전자통신연구소 선임연구원
 ※주관심 분야 : High-speed Digital Filter Design & Performance Analysis, VLSI Signal Processing, CDMA System VHDL Modeling & Implementation