

論文2000-37SD-11-11

면적 절약형 고속 FIR 필터의 설계 및 응용 (An Area Efficient High Speed FIR Filter Design and Its Applications)

李光鉉*, 林種錫**

(Kwang Hyun Lee and Chong Suck Rim)

요 약

FIR 디지털 필터는 DSP에서 사용되는 중요한 회로 중에 하나이며, 보다 효율적인 연산을 위한 여러 가지 구조가 제안되었다. 본 논문에서는 필터 연산을 고속으로 수행하면서도 면적을 줄일 수 있는 필터 구조를 제안한다. Transposed 구조를 적용하여, 고속의 연산이 가능토록 하는 기본 구조를 사용하였다. 여기에, 이중 경로 레지스터 라인이라는 두 개의 연산 패스가 존재하여 다양한 종류의 필터 연산이 가능하며, 이 필터를 연속적으로 이어 사용할 수 있는 cascade 구조도 지원한다. Truncated Booth 곱셈기라는 면적 절약형 곱셈기를 사용하여 회로 크기를 줄일 수 있었다. 이중 경로 레지스터 라인과 truncated 곱셈기를 사용하여 주어진 조건에 최적화된 필터를 설계할 경우에 회로의 크기가 더 줄어들 수 있음을 확인하였다.

Abstract

FIR digital filter is one of important blocks in DSP application. For more effective operation, lots of architecture are proposed. In our paper, we proposed a high speed FIR filter with area efficiency. To fast operation, we used transposed form filter as basic architectute. And, we used dual path registers line to support variation of filter operation, and filter cascade is also considered. To reduce area, we adopted truncated Booth multiplier to our filter design. As a result, we showed that filter area is reduced when filter optimization using of dual path registers line and truncated multiplier with same constraints against previous method.

I. 서 론

FIR(Finite Impulse Response) 디지털 필터는 DSP(Digital Signal Processing) 응용 회로에서 사용되는 중요한 회로이다. DSP에서 필터란 계속적으로 들어오는 디지털 입력 신호에 대해서 특정한 주파수 영역에 대해서만 통과시키고 그 이외의 주파수 영역에 대해서는 무시하도록 하는 회로이다. 이런 필터는 내부의 피드백(feedback)이 있는지 없는지에 따라서 두

가지로 분류가 되는데, 피드백이 있는 경우는 IIR(Infinite Impulse Response) 필터라 하며, 없는 경우는 FIR 필터라고 부른다. FIR 필터는 IIR 필터에 비해서 오차의 영향이 적으며, 쉽게 선형(linear phase) 특성을 만족하는 등 여러 가지 유리한 점이 있다. 그러나, FIR 필터는 같은 성능의 IIR 필터에 비해서 필터 연산을 위해서 많은 양의 하드웨어를 필요로 한다.

FIR 필터 연산을 수식으로 나타내면 다음과 같이 표현된다.

$$Y_i = \sum_{j=0}^{N-1} h_j \cdot X_{i-j} \quad h_j(0 \leq j < N) : \text{필터의 계수,}$$

X_i : i 번째 입력,

Y_i : i 번째 FIR 필터의 출력

* 正會員, (주) 서두로직 MyCAD 開發部

(Seodu Logic, Inc, MyCAD development dept.)

** 正會員, 西江大學校 컴퓨터學科

(Dept. of Computer Science, Sogang University.)

接受日字:2000年5月19日, 수정완료일:2000年10月25日

필터는 계속적으로 들어오는 입력 X_i 를 받아서 계

수들과 곱한 후 더해져서 계속적으로 결과 값을 내보낸다. 식으로부터 알 수 있듯이 N 개의 계수를 가지는 FIR 필터에서 하나의 출력을 얻기 위해서는 매번 N 번의 곱셈과 $N-1$ 번의 덧셈이 필요하다. 그러나, 만약에 필터의 계수가 대칭형(symmetric) 구조를 가지고 있다면, 계수 중에서 h_i 와 h_{N-1-i} 가 같은 값을 가지며, 이런 특성을 사용해서 $N/2$ 번의 곱셈만으로 계산이 가능하다.

FIR 필터 연산을 위한 기본적인 구조로 direct 구조(그림 1)와 transposed 구조(그림 2)가 있다. Direct 구조는 입력 값을 쉬프트 레지스터를 사용하여 저장하고 이 값들과 계수들을 곱한 후 더해져서 출력을 얻는 구조이다. Transposed 구조는 곱셈기의 출력을 레지스터를 거치면서 차례대로 누적해서 필터의 출력을 얻는 구조이다. 이 구조는 입력에 대한 fan-out이 크고, 많은 레지스터가 필요하다는 단점이 있지만, 상대적으로 고속 동작이 가능하다는 장점이 있어 고속 필터 구성에 유리하다.

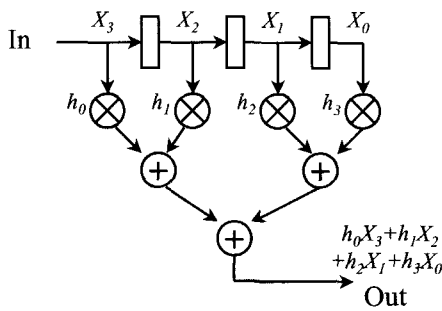


그림 1. Direct 구조 필터
Fig. 1. Direct form filter.

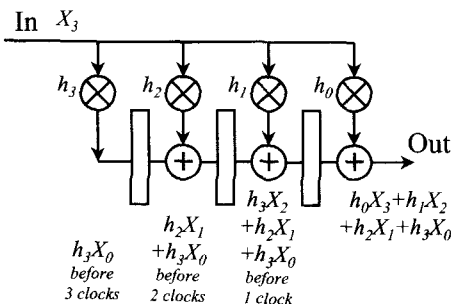


그림 2. Transposed 구조 필터
Fig. 2. Transposed form filter.

FIR 필터는 출력을 얻기 위해서 많은 곱셈과 덧셈을 필요로 하는 회로로써 이들의 연산량을 줄이기 위

한 많은 연구가 이루어 졌다^[5,7,8,9,12,13,14,16,19]. 이를 위해서 제안된 가장 일반적인 방법은 푸리에 변환을 사용하여 곱셈 연산을 줄이는 방법이다^[5]. 그리고, FIR 연산식으로부터 직접 곱셈연산 횟수를 줄이는 방법들도 제안되었다. 행렬 변환을 통해서 곱셈 연산 횟수를 줄이는 방법^[12,14,19]이 제안되었으며, multirate 구조를 사용하는 방법^[7,9,13]도 제안되었다. 기본적인 구조 하에서 bit-serial 구조^[8]를 사용하여 회로의 연산량을 줄이는 방법이 제안되었고, 계수간의 값 차이를 사용하여 연산을 줄이는 방법^[16]도 제안되었다.

본 논문에서는 이미 제안한 방법과는 다른 구조를 이용하여, 고속의 동작이 가능하면서도 적은 양의 하드웨어만을 가지는 필터를 제안한다. 본 논문에서는 기본적인 구조 자체로 고속 동작이 가능한 transposed 구조를 채택하였으며, 이중 경로 레지스터 라인을 사용하여 곱셈기의 사용을 효율적으로 할 수 있도록 하였다. 그리고, 본 논문에서는 면적 절약형 곱셈기를 사용하며, 곱셈 연산을 위해 필요한 하드웨어를 줄였으며, 출력 비트 수를 제한하여 레지스터와 덧셈기 등에 필요한 하드웨어를 감소시켰다. 그리고, 제안하는 구조는 곱셈기의 출력 비트 수를 조정하여 원하는 필터 특성에 맞추도록 조정하는 것도 가능하다.

본 논문에서 제안한 필터 구조는 면적 절약형 곱셈기의 사용으로 인해서 동작 속도의 손해 없이 회로의 크기에서 많은 이득을 얻을 수 있었다. 그러나, 곱셈기 오차의 영향으로 인해서 기존의 구조에 비해서 주파수 특성 등이 나빠지는데, 이를 보완하는 방법을 제시한다. 이러한 접근 방법은 같은 제약 조건을 가진 필터를 구성할 경우, 기존의 접근 방식 보다 훨씬 효율적으로 필터를 구성할 수 있음을 확인하였다.

II. FIR 필터 구조 설명

1. Transposed 구조

Direct 구조는 입력 데이터를 레지스터에 저장되어 전달되는 구조이지만, transposed 구조는 이와 달리, 필터 연산의 중간 결과 값이 레지스터에 저장되어 전달되는 구조를 가지고 있다. Direct 구조에서는 결과를 얻기 위해서 곱셈기를 거친 후에, 여러 단계의 덧셈기를 거쳐야만 결과를 얻을 수 있는 반면에 transposed 구조에서는 곱셈기를 거친 후에 한번의 덧셈기만 통과하면 계산이 가능하다. 다시 말해서

transposed 구조는 direct 구조에 비해서 구조적으로 고속 동작이 가능하다는 장점을 가지고 있다. 그러나, transposed 구조는 입력에 대한 팬 아웃(fan-out)이 크다는 문제점이 있는 데, 이는 적절하게 버퍼를 사용함으로써 쉽게 해결될 수 있다. 그리고 이 구조의 중요한 단점으로 레지스터의 크기가 커진다는 문제점이 있다. Direct 구조의 경우 입력 값을 레지스터에 저장하기 때문에 레지스터의 비트 수가 적지만, transposed 구조는 곱셈 결과를 저장해야 하기 때문에 비트 수가 크게 증가하며 필요한 레지스터의 양이나 덧셈기의 크기도 같이 증가한다. 본 논문에서는 이런 문제점을 해결하기 위해서 곱셈기의 출력 비트 수를 적절하게 제한하여, 필요한 레지스터의 크기와 덧셈기의 크기를 줄이는 방법을 제안하여 사용하였다.

그리고 곱셈기의 출력을 제한함에 따라서 곱셈기의 하위 비트 부분을 계산하지 않아도 된다. 본 논문에서는 이런 경우에 적용이 가능한 면적 절약형 곱셈기를 사용하여, 필터 회로의 크기를 더욱 줄였다. 면적 절약형 곱셈기를 사용하는 경우, 곱셈기의 출력을 반올림하는 방법에 비해서 오차가 증가하는 문제가 생기는데, 이는 출력의 비트 수를 조절하는 방법을 사용하여서 오차를 줄이는 것이 가능하다.

2. 이중 경로 레지스터 라인

본 논문에서 제안하는 필터의 구조는 transposed 구조를 기본으로 하면서 정방향과 역방향의 두 개의 레지스터 라인(그림 3)을 가지고 있다. 두 개의 레지스터 라인은 정방향과 역방향을 서로 연결해 주는 방법으로 대칭형 계수를 처리하도록 하였다. 홀수 탭 대칭형과 짝수 탭 대칭형을 모두 지원하도록 한 단계 앞선 결과와 출력으로 나가게 될 결과를 뽑아서 MUX로 연결하여 구성하였다. MUX를 제어하여 원하는 필터 연산을 수행하도록 하는 구조이다.

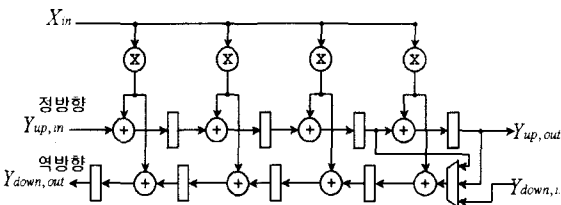


그림 3. 이중 경로 레지스터 라인
Fig 3. Dual path registers line

이중 경로 레지스터 라인은 이런 대칭형 계수 뿐만

아니라, 서로 계수의 값은 같지만 순서가 반대인 계수 쌍에 대해서 동시에 계산이 가능한 구조이다. CDMA 변조기^[17]에 사용되는 필터와 같이, 계수의 값은 같은 데 순서가 서로 반대인 구조를 가지는 계수 쌍이 존재하는 경우가 있다. 예를 들어 하나의 필터의 계수는 h_0, h_1, h_2, h_3 이고, 다른 하나의 필터는 그 계수 값이 h_3, h_2, h_1, h_0 인 경우이다. 본 논문에서는 이러한 두 계수에 대해 시간 역전형 계수 관계(time-reversal coefficient relation)이라고 정의하였으며, 이중 경로 레지스터 라인을 사용하여 효율적으로 구성할 수 있도록 하였다.

이러한 경우에는 그림 4와 같은 구조로 동작하게 된다. 이 경우는 대칭형 구조일 때와는 다르게, 계수의 순서는 정방향 레지스터 라인을 위해서 대칭형 구조와는 반대 순서를 가지게 된다. 필터의 동작을 확인하여 보면, 3번째 클럭 전에 들어오는 입력은 X_0 이라 가정하고, h_3 과 곱해져서 레지스터 a에 저장되며, 또 h_0 과 곱해져서 레지스터 e에 저장된다. 한 클럭 후에 입력 X_1 은 h_2 와 곱해진 후 레지스터 a의 값과 더해져서 $h_3X_0 + h_2X_1$ 이 되어 레지스터 b에 저장된다. 그리고, h_1 과도 곱해진 후 레지스터 e의 값과 더해져서 $h_0X_0 + h_1X_1$ 이 되어 레지스터 f에 저장된다. 한 클럭 후에는 레지스터 c에 $h_3X_0 + h_2X_1 + h_1X_2$ 가 저장되며, 레지스터 g에는 $h_0X_0 + h_1X_1 + h_2X_2$ 가 저장된다. 마지막으로, 한 클럭 후에는 레지스터 d에 $h_3X_0 + h_2X_1 + h_1X_2 + h_0X_3$ 이 저장되며, 레지스터 h에는 $h_0X_0 + h_1X_1 + h_2X_2 + h_2X_3$ 이 저장된다. 그리고, 그 다음 클럭에는 두 개의 출력으로 레지스터의 값이 나간다. 정방향 레지스터 라인의 출력인 $Y_{up,out}$ 은 첫 번째 필터 계수에 대한 출력이며, 역방향 레지스터 라인의 출력인 $Y_{down,out}$ 은 역전된 계수를 가지고 있는 두 번째 필터 계수에 대한 출력이다. 한번의 곱셈으로 두 개의 레지스터 라인에서 동시에 사용하므로, 4탭 계수를 가지는 두 개의 필터를 동시에 계산하기 위해서 8번이 아닌 4번의 곱셈만으로 수행이 가능하다.

3. Cascade 방안

두 개의 필터를 직렬로 연결하여 탭 수가 큰 필터를 구성하는 방법이 cascade이다. 이렇게 두 개의 필터를 연결하여 구성할 때, 입출력 레지스터가 없을 경

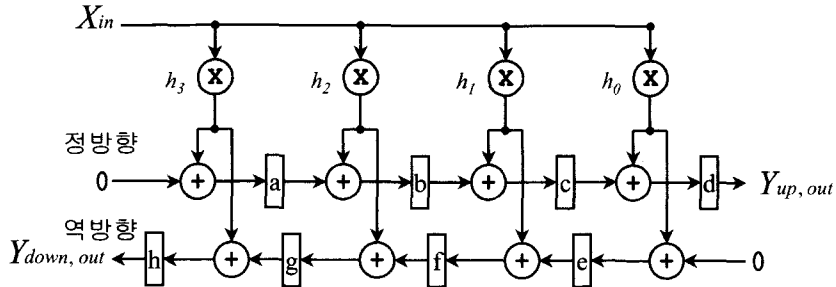


그림 4. 시간 역전형 계수 관계를 가지는 필터의 동작
 Fig. 4. Filter operation with time reversal filter coefficients.

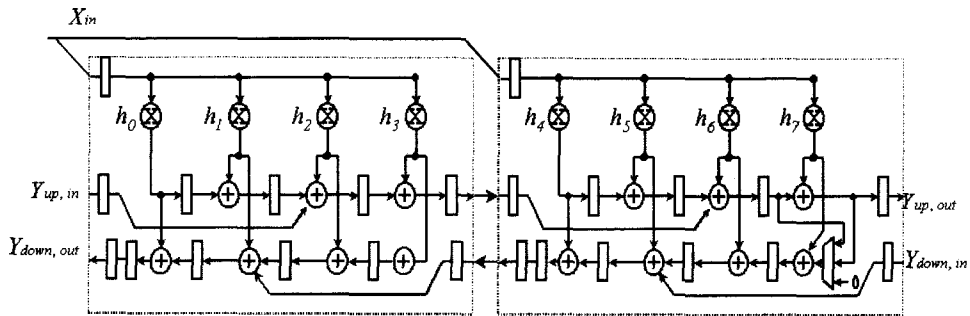


그림 5. 이중 경로 레지스터 라인과 입출력 레지스터가 있을 경우의 cascade
 Fig. 5. Filter cascading with dual path registers line and in/out registers.

우, 필터와 필터 사이에 필요한 외부 배선에 의한 딜레이로 인해서, 필터의 전체 동작 속도가 늦추어 질 수 있다. 그렇기 때문에, 레지스터와 버퍼를 두고 외부 배선에 의한 딜레이를 최소화시킴으로써, 필터 동작 속도를 가능한 빠르게 구성해야 한다. 그러나, 이런 경우에는 중간 결과를 더해주는 순서가 바뀌기 때문에, 필터가 정상적으로 동작하지 않게 된다. 정상적인 동작을 위해서는 레지스터에 의해서 바뀐 순서를 다시 맞추어 주어야 하는데, 입출력 레지스터에 의해서 2 클럭 만큼 지연되었으므로, 2개의 레지스터를 지나친 후에 더해 주어 순서를 맞춘다. 이런 식으로 두 개의 4 탭 필터를 cascade한 회로를 그림 5에 보인다. 왼쪽의 필터는 슬레이브 필터로 시간역전형 계수를 처리하는 경우와 같은 구조로 동작하며, 오른쪽의 필터가 마스터 필터가 되어 cascade된 전체 필터의 동작을 제어하게 된다.

4. Carry Save Adder를 사용한 데이터 패스의 고속화
 보다 빠른 동작을 하게 하기 위해서, 내부에 사용하는 덧셈기를 모두 CSA(Carry Save Adder)를 사용하였다. 그리고, 이를 위해서, 곱셈기는 CPA부분을 없애고 sum과 carry를 그대로 출력으로 내보내었다(그

림 6, 7). 이것이 두 단계, 또는 세 단계의 CSA를 거쳐서 레지스터로 들어가게 된다. CSA를 사용하지 않은 경우, CPA가 있는 곱셈기를 거친 후에, 다시 CPA를 지나서 레지스터로 들어가게 된다. 이 경우에는 고속의 CPA를 사용하지 않을 경우 덧셈 연산에 필요한 지연 시간이 많이 필요하게 된다. 이에 반해, CSA를 사용할 경우, CPA가 없는 곱셈기를 거친 후, 두 번 또는 세 번의 CSA를 통과하여 레지스터로 들어가게 된다. CSA의 지연 시간은 하나의 덧셈기와 같으므로, 상당한 시간 이득을 얻는다.

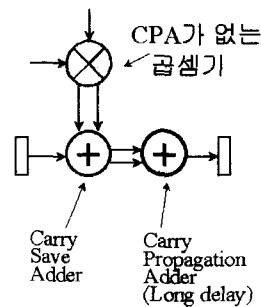


그림 6. CPA를 사용한 데이터 라인
 Fig. 6. Data path with CPA.

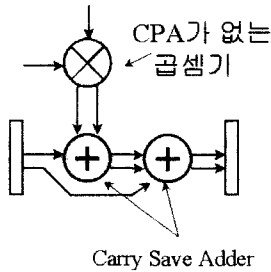


그림 7. CSA를 사용한 데이터 라인
Fig. 7. Data path with CSA.

다만, 덧셈기와 레지스터의 수가 많아진다는 점이 문제가 된다. 덧셈기의 경우, 고속의 CPA를 사용하는 경우와 비교한다면, 추가되는 덧셈기의 양은 어느 정도 상쇄된다. 그리고, 곱셈기에 사용되어지는 CPA가 없어지므로 필요한 하드웨어의 양은 좀더 줄어들게 된다. 하지만, 레지스터의 크기는 두 배로 증가하며, 이는 빠른 동작을 의한 부가적으로 증가하는 하드웨어의 양이다.

내부적으로 CSA를 이용하여 계산된 중간 결과는 마지막 탭에서, 고속의 CPA인 carry selector adder를 사용하여, 두 개의 중간 결과를 합해 최종 결과를 만들어 낸다. 여기서, carry selector adder를 사용한 이유는 CPA에 필요한 지연 시간이 곱셈기와 세 번의 CSA를 거치는 필터의 지연 시간보다 짧아야 하고, 이를 위해서 적절한 지연시간을 가지는 CPA중에서 선택한 것이다. Cascade를 위해서 하단에 연결되는 두 개의 Y_n 에 대해서는 carry selector adder를 거친 출력이 연결되며, 멀티 비트 데이터 라인을 적용하지 않는다.

5. Truncated Booth 곱셈기

Kidambi는 DSP 응용회로에서 사용 가능한 truncated 곱셈기^[18]를 제안하였다. IIR 필터 같은 피드백이 있는 회로에서는 곱셈기의 모든 출력을 사용하는 것이 아니라, 그 중 일부분만 사용하여 회로를 구성하게 된다^[11]. 이 곱셈기는 이런 경우에 사용하지 않는 하위 비트 출력을 계산하는 부분을 잘라 내어 면적을 절약할 수 있는 곱셈기이다. 하지만, 이 곱셈기는 회로의 손실로 인해서 결과에 오차가 발생하게 되는데, 이 오차를 확률적인 접근을 통해서 계산한 뒤에, 이를 교정함으로써 오차를 최소화하여 사용하였다.

본 논문에서는 Booth 곱셈기의 출력 중 상위 비트 부분의 일부만을 내보내도록 하고, 하위 비트를 계산

하는 부분을 잘라 낸 truncated Booth 곱셈기를 사용한다^[20]. Booth 곱셈기는 Booth 곱셈 알고리즘^{[21][15]}을 회로로 구현한 것인데, array 곱셈기 구조를 사용하여 곱셈기 구성이 가능하다. Booth 곱셈기는 일반적으로 승수 값 2비트를 한번에 처리하는 방법을 사용한다. 2비트를 한번에 처리하기 때문에, 전체적인 구조는 승수가 N 비트 일 경우 $N/2$ 개의 row로 구성된다. 각 Row는 승수 2비트와 하위 1비트 값으로부터 덧셈 또는 뺄셈 연산을 결정하는 Booth encoder와 이 encoder로부터 나온 신호로부터 계산에 사용할 값을 만들어 내는 Booth selector가 있으며, 그리고 Booth selector로부터 나온 값을 사용하여 실제로 계산을 하는 Booth adder가 있다. 특별히 최상 단의 경우 상위로부터 나오는 중간 결과가 없기 때문에, 이를 위한 Booth first회로가 있다. 그리고, 최하 단에는 각 row로부터 얻어진 중간 결과 값으로 최종 결과를 얻어내는 CPA가 있다. 예를 들어 8×8 비트 곱셈기 회로 중 상위 비트인 $Z[7] \dots Z[14]$ 를 계산하는 부분만 남겨 두고, 하위 비트인 $Z[0] \dots Z[6]$ 를 계산하는 부분을 cut line을 기준으로 잘라 내어 곱셈기를 재구성한다. 몇 번째 row인지에 따라서 적절한 비트 수만큼 Booth selector와 Booth adder 그리고 Booth first 회로를 잘라 낸다. 그리고, CPA도 필요 없는 비트 수만큼만을 잘라 내어서 면적의 이득을 얻도록 구성한다. 이런 식으로 하위 비트 계산 부분을 잘라 내어 곱셈기 면적에 대한 이득을 얻을 수 있으며, 하위 비트 계산하는 부분을 잘라낸 truncated Booth 곱셈기의 구조는 그림 8에 보인다.

하위 비트 부분을 계산하는 부분을 잘라 내었기 때문에, 이 곱셈기는 결과 값에 항상 오차가 포함되어 있다. 입력의 각 비트들이 1일 확률을 0.5라고 가정하고, 각 서브 블록들의 진리표를 이용하여, 각 블록들의 출력들에 대한 확률을 계산할 수 있다. 이 확률들을 이용하여, 곱셈기에서 발생하는 예상 오차를 계산할 수 있다. 이렇게 계산된 예상 오차만큼을 곱셈 결과에 교정하여, 오차를 최소화하여 사용할 수 있다. 그러나, 이 방법만으로는 오차의 분포까지는 바뀌지 않으며 1 이하의 값에 대해서는 다른 교정 회로가 필요하다는 문제점이 있다. 이런 방법 외에 곱셈기의 오차를 줄이기 위해서 출력 비트 수를 확장하는 방법이 가능하다. 곱셈기의 출력 비트 수가 증가함에 따라서 정밀도는 그만큼 높아지며, 오차의 분포는 좁아지게 된다. 필요

한 만큼 비트 수를 증가시켜, 원하는 만큼의 정확도를 만족하도록 할 수 있다. 비트 수가 증가하여 정확도가 증가하는 만큼 필요한 회로는 증가하는 정확도와 면적 사이의 반비례 관계를 가지며, 설계시 적절하게 선택하여 사용하는 것이 가능하다

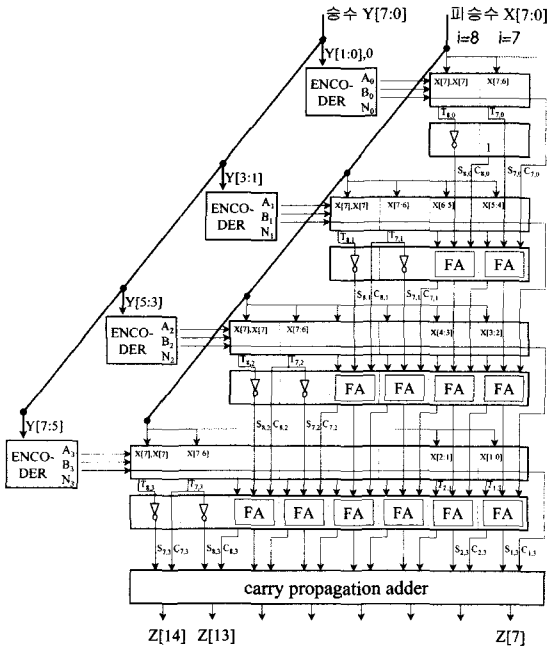


그림 8. 8×8 비트 truncated Booth 곱셈기
Fig. 8. 8×8 bits truncated Booth multiplier.

6. FIR 필터의 전체 구조의 설명

그림 9는 본 논문에서 제안하는 FIR 필터의 구조를 상세하게 표현한 것이다. 이중 경로 레지스터 라인을 두었으며, 입출력 단자에는 레지스터를 두었다. Cascade를 위해서 두 단계의 레지스터를 지나친 후에 입력을 중간 결과에 더해 주었다. CSA를 사용하여 데이터 라인을 구성하였고, 마지막 탭에는 carry selector adder를 사용하였다. 홀수 탭 대칭형인 경우에는 (1)번 경로를 통해서 데이터가 흐르며, 두 개의 Y입력은 0으로 세팅한다. 짝수 탭 대칭형인 경우에는 (2)번 경로를 통해서 데이터가 흐르며, 역시 두 개의 Y입력은 0으로 세팅한다. 이중 경로 레지스터를 사용하여 계수 쌍에 대해서 처리할 경우 (3)번 경로를 통해 하단의 역방향 레지스터는 0으로 세팅시킨다. 그리고 Cascade를 적용한 필터로써 사용되는 경우 역시 (3)번에 의해서 0으로 세팅되나, 두 개의 Y입력은 cascade 된 좌우 회로의 Y출력과 서로 연결되어, 중간 결과 값을 전달해 준다.

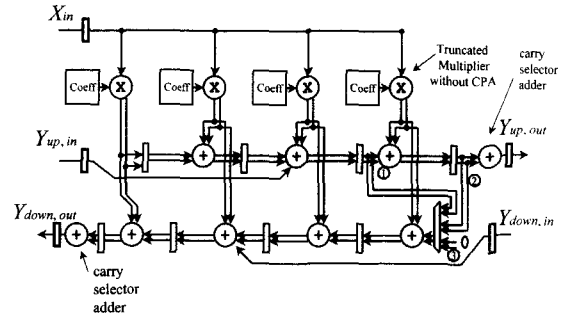


그림 9. 본 논문에서 제안하는 FIR 필터의 구조
Fig. 9. Whole FIR filter architecture.

III. 성능 평가 및 응용

1. Truncated 곱셈기 사용에 대한 비교

Truncated 곱셈기를 사용할 경우, 일반 곱셈기를 사용할 경우에 비해서 어느 정도의 오차 범위를 가지는지 비교하였다. 이를 위해서 48탭 계수를 가지는 필터를 구성하였다. 입력은 8비트, 계수는 12비트를 사용하였으며, 비교를 위해서 5가지의 필터를 구성하였다. 첫 번째는 곱셈기의 출력인 19비트를 모두 사용한 필터이며, 두 번째는 19비트 곱셈기 출력 중 12비트 위치에서 반올림하여 12비트 출력으로 사용한 경우이다. 세 번째는 12비트 출력을 얻는 truncated 곱셈기를 사용한 경우이며, 네 번째, 다섯 번째는 각각 1, 2비트 씩 곱셈기의 출력을 확장한 필터이다. 정리하면 표 1과 같다.

표 1. 실험을 위해 사용된 필터들의 구성
Table 1. Filter variations.

	입력	계수	곱셈기 출력	반올림 위치	가드 비트	레지스터
필터 1	8	12	19	X	4	23
필터 2	8	12	19	12	4	16
필터 3	8	12	12	X	4	16
필터 4	8	12	13	X	4	17
필터 5	8	12	14	X	4	18

이렇게 구성된 5가지 필터에 대한 주파수 특성과 필터 1의 출력에 비교한 오차 값의 평균, 최대/최소값, 분산 등을 정리하였다. 표 2는 3가지 임펄스에 대한 필터의 주파수 특성이다. 첫 번째 경우는 임펄스를 입력의 최대 값인 0111111(=0.9922)를 넣었을 때이며, 두 번째는 임펄스로서 01000000(=0.5)를 넣었을 때,

세 번째는 00100000(=0.25)를 넣었을때에 대한 각각의 stopband attenuation이다. 임펄스가 적은 값을 가질수록 출력 비트수에 따라 크게 영향을 받게 된다. 그러나, 두 경우 모두, truncated 곱셈기를 사용한 경우가, 곱셈기를 반올림하여 사용한 경우 보다, 주파수 특성이 좋다는 것을 확인할 수 있었다.

표 2. Impulse 신호에 대한 주파수 특성(단위 : dB)

Table 2. Stopband attenuation of impulse response(unit : dB).

	stopband atn. (impulse=1.0)	stopband atn. (impulse=0.5)	stopband atn. (impulse=0.25)
필터 1	44.58	44.58	44.58
필터 2	44.34	44.33	43.12
필터 3	44.42	44.39	43.12
필터 4	44.48	44.46	44.03
필터 5	44.54	44.52	44.57

표 3. Chirp 신호에 대한 필터의 주파수 특성 및 오차

Table 3. Error comparison about chirp signal.

	stopband attenuation	E (error)	MAX (error)	MIN (error)	Var (Err)
필터 1	41.82	0.0	0.0	0.0	0.0
필터 2	41.57	0.0	0.00153	-0.00162	0.00093
필터 3	41.63	-0.00054	0.00417	-0.00508	0.00542
필터 4	41.68	-0.00029	0.00157	-0.00208	0.00116
필터 5	41.74	-0.00017	0.00063	-0.00104	0.00026

표 3은 1024개의 입력을 가지는 8비트 chirp 신호에 대한 필터의 출력을 확인한 것으로 임펄스 신호에 비해서는 stopband attenuation이 낮게 나왔다. 이는 chirp 신호의 주파수 특성이 필터에 영향을 주었기 때문이다. 각 필터와 모든 비트 출력을 사용한 필터 1과의 오차를 비교하면, truncated 곱셈기를 사용한 필터의 오차가 반올림하여 사용한 필터보다 훨씬 오차가 크다는 것을 확인하였다. 그러나, 2비트를 확장한 필터 5의 경우, 반올림하는 경우보다 오차가 훨씬 줄어들 수 있다.

2. 이중 경로 레지스터 라인의 사용에 대한 비교
 (1) 필터의 분할과 시간 역전형 계수 관계

오버샘플링을 수행하는 필터의 경우 필터를 분할하여 재구성할 수 있었다^[17]. 오버샘플링이란 샘플링 주파수를 높이기 위해서 입력들 사이에 '0'을 넣어 주는 방법이다. 예를 들어 X_1, X_2, X_3, X_4 와 같은 입력이 들어 올 때, 이를 4배 오버샘플링 한다면, $X_1, 0, 0, 0, X_2, 0, 0, 0, X_3, 0, 0, 0, X_4, 0, 0, 0$ 과 같이 입력들 사이에 3개의 '0'이 들어가며, 샘플링 주파수는 4배 빠르게 된다. 이런 식으로 오버샘플링에서 '0'이 입력들 사이에 들어가는데, '0'은 필터의 출력에 영향을 주지 않는다. 이런 이유로 '0'을 제외한 나머지 입력들만을 필터의 입력으로 사용하는 여러 개의 서브 필터를 사용하여 재구성할 수 있다. 예를 들어 4배 오버샘플링하는 48탭 필터는 4개의 12탭 필터로 분할하고 이를 MUX를 사용하여 신호를 받는 방법을 사용하여 재구성하는 것이 가능하다. 그림 10은 분할되지 않은 필터이고, 그림 11은 4개의 서브필터로 분할된 구조이다.

그런데, 이렇게 필터를 분할하여 재구성할 경우, 원래의 필터는 대칭형 구조이지만 분할된 4개의 필터는 모두 비대칭형 구조가 된다. 대칭형 구조라면 절반의 곱셈만으로 필터 연산을 수행할 수 있지만, 비대칭형 구조라면 필터의 탭수만큼 곱셈 연산이 필요하게 된다. 그러나, 이렇게 분할된 계수들 사이에는 특별한 관계가 존재한다. 12탭의 계수의 값은 서로 같지만, 순서가 반대인 구조를 가지는 계수쌍이 나타난다. 본 논문에서는 이와 같이 계수의 값은 같고 순서가 서로 반대인 계수 쌍에 대해서 시간 역전형 계수 관계라고 정의한다.

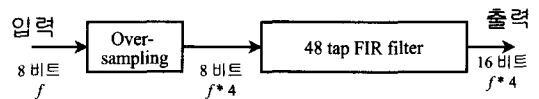


그림 10. 분할되지 않는 FIR 필터 구조
 Fig. 10. Unpartitioned FIR filter structure.

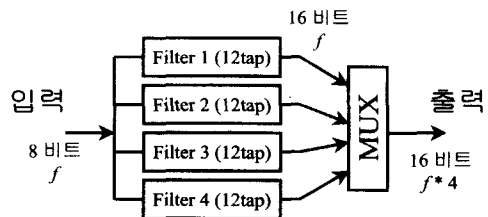


그림 11. 4개의 서브필터로 분할된 FIR 필터 구조
 Fig. 11. Partitioned filter with 4 subfilters.

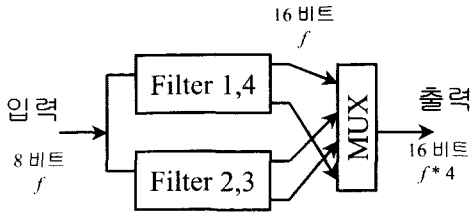


그림 12. 시간역전형 계수 관계를 이용하여 재구성한 구조

Fig. 12. Reformed structure using time reversal filter coefficients.

이러한 시간 역전형 계수 관계는 원래의 필터 계수가 대칭형이고, 오버 샘플링으로 필터를 분할하여 구성하는 경우 항상 나타난다. 이런 시간 역전형 계수 관계를 본 논문에서 제안한 이중 경로 레지스터 라인을 가지는 transposed 구조를 사용하여 그림 12와 같이 묶어 재구성하는 것이 가능하다. 이렇게 구성할 경우, 절반의 곱셈기만으로 구성이 가능하기 때문에, 필터를 더욱 효율적으로 구성하는 것이 가능하다.

(2) 비교

필터를 분할하였을 경우 비대칭형 계수에 그대로 구현하였을 경우와 이중 경로 레지스터 라인을 사용하여 구성하였을 경우에 대해서, 필요한 회로의 크기를 비교하였다. 그리고, 분할하지 않은 필터도 구성하여 회로의 크기를 비교하였다. 필터 A는 분할하지 않은 필터이며, 필터 B는 4개의 서브필터로 분할하여 재구성한 경우이다. 그리고, 필터 C는 시간 역전형 계수 관계를 가지는 두 개의 필터를 묶어서 재구성한 경우이다.

모든 회로는 VHDL을 사용하여 구현하였으며, LG 0.6 um 공정의 라이브러리를 사용하여, Synopsys synthesizer를 이용해서 gate level로 합성된 결과이다. 입력은 8비트, 계수는 12비트를 사용하였으며, 회로 크기는 NAND 게이트 기준의 게이트 개수이고, 동작 속도는 각각 입력과 출력의 동작 속도이다.

실험 결과는 표 4과 표 5에 정리되어 있다. Truncated 곱셈기를 사용하지 않은 경우 44.1dB의 stopband attenuation을 가지며, 필터 A에 비해 2%의 회로 크기가 증가했고, 필터 B에 비해서는 24%의 면적이 감소하였다. 그리고, 필터 B에 비해서 약간의 속도 향상 효과도 얻을 수 있었다. Truncated 곱셈기를 사용한 경우에는 42.7dB의 stopband attenuation을

보였으며, 필터 A'에 비해서 3%의 면적이 감소하였고, 필터 B'에 비해서는 24%의 면적이 감소하였다. 필터 C'에 비해서도 속도가 향상되었다.

표 4. 일반 곱셈기를 사용한 48탭 필터의 특성 비교

Table 4. Filter comparison with 48taps filters (normal multiplier).

	필터 A	필터 B	필터 C
회로 크기	47348	63136	48413
곱셈기 수	24	48	24
동작 속도(MHz)	5/20	20/80	22/88
stopband attenuation(dB)	44.1	44.1	44.1

표 5. Truncated 곱셈기를 사용한 48탭 필터의 특성 비교

Table 5. Filter comparison with 48taps filters (truncated multiplier).

	필터 A'	필터 B'	필터 C'
회로 크기	32376	41254	31637
곱셈기 수	24	48	24
동작 속도(MHz)	7/28	20/80	24/96
stopband attenuation(dB)	40.27	40.27	40.27

3. 필터 최적화에 대한 비교

CDMA 변조기에 사용되는 pulse shaping 필터는 다음과 같은 요구 조건을 만족해야 한다. 입력으로 들어오는 데이터의 샘플링 주파수는 4915KHz이고, passband 주파수는 590KHz 이하이고, stopband 주파수는 740KHz 이상이다. Passband에서 ripple 허용치는 ±1.5dB로 3.0dB이내이어야 하며, passband와 stopband의 차이인 stopband attenuation은 40 dB이상이다. 그리고, impulse response에 대한 mean square error에 대한 조건은 다음과 같은 수식으로 표현된다.

$$Mean\ Square\ Error = \sum_{k=0}^{\infty} [as(kT_s - \tau) - h(k)]^2 \leq 0.03$$

s(t)는 필터의 시간 도메인에 대한 출력이며, T_s는 샘플링 간격인 203.451ns이고, α와 τ는 이 오차를 최소화 사용될 수 있는 값이다. 필터의 impulse response를 가능한 오차가 적게 만들어지도록 α와

τ 를 정의하였을 때, 필터의 impulse response와 계수의 차이에 대한 누적 값이 0.03 이내가 되어야 한다는 의미이다.

그리고, 위상(phase) 특성에 대해서 passband영역에 대해서 위상 값을 1차 방정식에 fitting 하였을 때, 이에 대한 조건을 수식으로 표현하면 다음과 같다.

$$\sum_i [Arg(H(f_i)) - (a \cdot f_i + b)]^2 < 0.001 \text{ radian}$$

$H(f_i)$ 는 필터의 주파수 도메인 출력이며, $Arg(H(f_i))$ 는 필터의 phase를 의미한다. 이 값을 1차 방정식 $Y = aX + b$ 에 피팅(fitting)시키고, 각 부분에서 피팅된 1차 방정식과의 차이의 제곱에 대한 누적 값이 0.001 라디안 이하이어야 한다.

주어진 조건을 만족하는 최적의 필터를 구성하고자 할 경우, 필터의 탭 수를 최소화하여, 하드웨어를 최소화 할 수 있다. 따라서, 본 논문에서는 truncated 곱셈기를 사용하여 하드웨어를 최소화한 경우와 주어진 조건을 만족하는 최소 탭 수의 필터를 설계한 경우를 비교하였다. 이를 위하여 적절한 탭 수의 필터 계수를 사용하고, truncated 곱셈기를 사용하여, 주어진 조건을 만족하는 한도까지 곱셈기의 출력을 잘라 내어서, 필터를 구성한다.

표 6. 탭 수 조절과 truncated 곱셈기 사용의 trade off 비교
Table 6. Comparison between 2 filter optimization method

	필터 1	필터 2
요구 조건	passband ripple : 3dB 이하 stopband attenuation : 40dB 이상 mean square error : 0.03 이하 phase response error : 0.001rad 이하	
입력	8 비트	
계수	12 비트	
레지스터	14 비트	
곱셈기	곱셈기 출력을 반올림하여 사용	Truncated 곱셈기 사용
탭 수	44	48
회로 크기	30885	26254
Passband ripple (dB)	2.24	2.26
Stopband attenuation(dB)	41.63	40.59
Mean square error	2.695×10^{-3}	2.576×10^{-3}
Phase response error	0.0	0.0

그리고, 입력은 8비트, 계수는 12비트이며, 레지스터와 필터의 출력은 14비트로 하였으며, 이에 대한 실험 결과는 표 6에 정리되어 있다. 필터 1은 탭 수를 최소화한 경우인데, 40dB 특성을 만족하기 위해서는 최소한 44탭은 필요하였다. 이때 stopband attenuation은 41.63dB가 나왔다. 필터 2의 경우 48탭 필터의 곱셈기를 truncation 할 수 있는 최대 한도에서 자른 것이며, 이 때 40.59dB 특성을 가졌다. 회로의 크기를 비교할 경우, 필터 2는 필터 1에 비해서 회로의 크기가 15%의 감소하면서도, 같은 요구 조건을 만족할 수 있었다.

IV. 결론

FIR 필터는 그 자체의 연산 량이 많기 때문에, 회로로 구성하기 위해서는 많은 양의 하드웨어를 필요로 한다. 보다 효율적인 필터를 설계하기 위해서 많은 연구들이 진행되어 왔다. 본 논문에서는 이중 경로 레지스터 라인을 사용하는 transposed 구조의 필터 구조를 사용하고, 면적 절약형 곱셈기를 사용한 면적 절약형 고속 필터를 구성하였다. CDMA transmit 변조기의 pulse shaping 필터의 동작 특성으로부터, 오버샘플링으로 인해서 필터를 분할하여 구성할 수 있다. 이런 경우에 발생하는 시간 역전형 계수 관계를 본 논문에서 제안하는 이중 경로 레지스터 라인을 사용하여 효율적으로 구성하였다. 필터를 분할할 경우 고속의 동작이 가능하지만, 계수의 특성상 면적이 증가한다는 문제점이 있었는데, 본 논문에서 제안하는 구조는 계수들의 특성을 살려서 회로의 면적을 줄이는 것이 가능하였다.

또한, 본 논문에서 면적을 절약할 수 있는 truncated Booth 곱셈기를 제안하였다. 이 곱셈기의 오차를 확률적인 접근으로 분석하였고, 곱셈기의 면적과 전력 소모 이득을 확인하였다. 그리고, 이 곱셈기를 본 논문에서 제안한 FIR 필터 구조에 적용하여, 필터의 주파수 특성에 대한 손실을 적은 범위 이내로 유지하면서, 필터의 면적의 이득을 얻을 수 있었다.

CDMA 변조기에 사용되는 pulse shaping 필터를 최소의 하드웨어만으로 구성하기 위해서 기존의 접근 방식인 탭수를 줄이는 접근보다, 본 논문에서 제안하는 구조하에서 truncated 곱셈기를 사용하는 것이 훨씬 적은 하드웨어만으로 구성이 가능함을 확인하였다.

본 논문에서 제안하는 필터 구조를 다른 종류의 필터에 적용 가능 여부에 대해 더 연구할 필요가 있으며, 기존에 제안된 구조와 혼용하여 사용하는 방법을 통해 더욱 더 효율적인 필터를 구성하는 방법을 고려할 수 있다.

참 고 문 헌

- [1] Andreas Antoniou, Digital Filters :Analysis, Design, and Applications, 2nd Ed. McGraw-Hill, 1993.
- [2] Alan V. Oppenheim, Ronald W. Schaffer, Discrete-Time Signal Processing, Prentice-Hall, 1989.
- [3] Baugh, C. R. and Wooley, B. A., "A Two's Complement Parallel Array Multiplication Algorithm," IEEE Trans. on Computers, Vol.C-22, No.1-2, Dec. 1973, pp. 1045-1047.
- [4] Booth, A. D., "A signed Binary Multiplication Algorithm," Quart. J. Mech. Appl. Math., Vol. 4, Pt. 2, 1951, pp. 236-240.
- [5] E. Bidet, J.M. Cardin, M. Djoko Kouam, C. Joanblanq, J. Palicot, "FDF, a 512-TAP FIR Filter Using a Mixed Temporal-Frequential Approach," IEEE 1995 Custom Integrated Circuits Conference, pp. 173-176, 1995.
- [6] Eero Pajarre, Tapio Saramaki, "Efficient VLSI Implementation Techniques for FIR Filters," DSPx 1994, pp. 560-566, 1994.
- [7] Ernst Lueder, "Generation of equivalent block parallel digital filters and algorithms by a linear transformation," IEEE ISCAS, Chicago, pp. 495-498, 1993.
- [8] Hwan-Rei Lee, Chein-Wei Jen, Chi-Min Liu, "A New Hardware-Efficient Architecture for Programmable FIR Filters," IEEE Trans. on C&S-II, vol. 43, pp. 637-644, NO. 9, SEP, 1996.
- [9] Ing-Song Lin, Sanjit K. Mitra, "Fast FIR Filtering Algorithms Based on Overlapped Block Structure," IEEE ISCAS, Chicago, pp. 363-366, IEEE 1993.
- [10] Johnny R. Johnson, Introduction to Digital Signal Processing, Prentice-Hall, 1989.
- [11] Jun Rim Choi, Seong Wook Jeong, Lak Hyun Jang, Jin Ho Choi, "Structured Design of a 288-tap FIR Filter by Optimized Partial Product Tree Compression," IEEE Custom Integrated Circuits Conference, 1996.
- [12] Mahesh Mehendale, S. D. Sherlekar, G. Venkatesh, "Low Power Realization of FIR Filters using Multirate Architectures," IEEE 9th Int'l Conf. on VLSI Design, pp.370-375, JAN 1996.
- [13] Markus Wintermantel, Ernst Luder, "Increasing the speed and saving multipliers in block parallel digital filters by a linear transformation," IEEE ISCAS, London, vol 2. pp. 81-84, 1994.
- [14] Martin Vetterli, "Running FIR and IIR Filtering Using Multirate Filter Banks," IEEE Trans. on ASSP, vol.36, pp.730-738, NO. 5. MAY 1988.
- [15] Neil H. E. Weste, Kamran Eshraghian, Principles of CMOS VLSI Design : A systems Perspective 2nd Ed., Addison-Wesley, 1993.
- [16] S. Sankarayya, Kaushik Roy, W. Lafayette, "Algorithms for Low Power FIR Filter Realization Using Differential Coefficients," IEEE 10th Int'l Conf. on VLSI Design, pp.174-178, JAN 1997.
- [17] Samuel Sheng, Rober Brodersenm, Low-Power CMOS Wireless Communications : A Wideband CDMA System Design, Kluwer Academic Pub., 1998.
- [18] Sunder S. Kidambi, Fayez El-Guibaly, Andreas Antoniou, "Area-Efficient Multipliers for Digital Signal Processing Applications," IEEE Trans. on C&S-II, vol. 43, pp. 90-95, NO. 2, FEB, 1996.
- [19] Zhi-Jian Mou, Pierre Duhamel, "Short-Length FIR Filters and Their Use in Fast Nonrecursive Filtering," IEEE Trans on Signal Processing, vol 39, pp.1322-1332,

NO. 6, JUN 1991.

[20] Lee, Kwang Hyun, Rim, Chong Suck, "A Hardware Reduced Multiplier for Low Power Design", Proceeding of The Second IEEE Asia Pacific Conference on ASICs, pp. 331-334, Aug 2000.

저 자 소개

李 光 鉉(正會員)

1974년 7월 22일생. 1997년 2월 서강대학교 전자계산학과 졸업(학사). 1999년 2월 서강대학교 컴퓨터학과 대학원 졸업(석사). 1999년 2월~현재 (주) 서두로직 전자기술연구소 연구원 재직중

林 種 錫(正會員) 電子工學會 論文誌 第36卷 第1號

1999년 1月號 參照

