

최소 가산 그래프 알고리즘에 의한 힐버트 변환기 설계에 관한 연구

이영석*

Using MAG Algorithm for Reducing Hardware in Hilbert Transformer Design

YoungSeock Lee*

요 약

DSP 시스템에서 널리 사용되는 힐버트 변환에서 곱셈 연산은 반드시 필요한 요소이며 변환에 사용되는 계수의 차수가 높아질수록 하드웨어는 복잡하고 많은 양의 게이트를 필요로 한다. 본 연구에서는 힐버트 변환에 사용되는 곱셈연산에 MAG 알고리즘이 적용된 쉬프트와 덧셈을 사용한 곱셈블록을 구현하여 하드웨어의 복잡도를 줄일 수 있다.

ABSTRACT

A hardware implementation of Hilbert transform is indispensable element in DSP system, but it suffers from a high complexity of system level hardware resulted in a large amount of the used gate. In this paper, we implemented the Hilbert transformer using MAG algorithm that reduces the complexity of hardware.

Key Word

SOC, Hilbert Transform, MAG algorithm, DSP, Reducing Hardware

1. 서 론

최근 디지털 멀티미디어기기는 작고 빠르면서 높은 성능의 다양한 부가기능들로 인해 좁은 면적의 칩 위에 복잡한 설계를 구현하는 SOC(system on chip)분야에서 좀 더 효율적이고, 높은 성능과 낮은 복잡도를 갖는 설계에 관한 기술이

연구되고 있다. 특히 현대의 디지털 통신 및 멀티미디어기기는 많은 데이터량의 처리를 위해 디지털 신호처리 (DSP: digital signal processing)시스템을 반드시 필요로 한다. 이에 따라 높은 샘플링 비율은 연산량을 증가시키고, 또한 하드웨어의 복잡도는 증가하며, 많은 전력을 소비하게 된다. 그러므로 DSP 시스템에

* 청운대학교 디지털방송공학과(yslee@chungwoon.ac.kr)

* 본 연구는 2007학년도 청운대학교 학술연구조성비의 지원으로 연구되었음.

#논문번호 : KIIECT2009-04-06

#접수일자 : 2009.10.15

#최종논문접수일자 : 2009.11.23

다. 이 때 h_{ks} 다음과 같은 특징을 갖는다[4].

$$h_{ks} \begin{cases} -h_{sk} \\ h_{\langle k+1 \rangle \langle s+1 \rangle}, \langle k \rangle \triangleq k \pmod N \\ h_{ks} = 0, k = s \end{cases} \quad (5)$$

식 (3)에서 $N=5$ 일 때를 가정하여 계산된 h_{ks} 값은 $h(0)=0, h(1)=0.6156, h(2)=-0.1452, h(3)=0.1452, h(4)=-0.6156$ 이며 계수의 값이 대칭을 이루는 특징을 나타낸다. 이 때 H 는 식 (5)의 특징을 사용하여 다음과 같이 식 (6)으로 표현할 수 있다[4].

$$H = \begin{bmatrix} 0 & h_0 & h_1 & -h_1 - h_0 \\ -h_0 & 0 & h_0 & h_1 - h_1 \\ -h_1 - h_0 & 0 & h_0 & h_1 \\ h_1 & -h_1 - h_0 & 0 & h_0 \\ h_0 & h_1 & -h_1 - h_0 & 0 \end{bmatrix} \quad (6)$$

여기에서 h_{ks} 는 $h_{01} \triangleq h_0, h_{02} \triangleq h_1$ 으로 정의한다.

이에 따라 식 (4) 와 식 (6)의 행렬의 곱셈 연산에서 2개의 곱셈 요소로 변환한 행렬연산[3]을 식 (7)과 같이 이산 힐버트 변환기의 설계를 위한 일반화된 행렬식으로 나타 낼 수 있다[4].

$$\begin{bmatrix} y(0) \\ y(1) \\ \cdot \\ \cdot \\ y(4) \end{bmatrix} = \begin{bmatrix} x(1) - x(4) & x(2) - x(3) \\ x(2) - x(0) & x(3) - x(4) \\ x(3) - x(1) & x(4) - x(0) \\ x(4) - x(2) & x(0) - x(1) \\ x(0) - x(3) & x(1) - x(2) \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} \quad (7)$$

일반적인 설계에서 $N=5, 6$ 일 때의 각 설계를 위와 같은 과정을 통하여 일반화된 h_{ks} 와 입력 $x[(0 \dots (N-1))]$ 의 곱셈 연산으로서 구현하였다.

III. 제안된 힐버트 변환기의 설계

3.1. MAG 알고리즘

MAG 알고리즘은 식 (8)과 같이 각 단계에 의한 연산 과정을 통하여 계수를 만들어 낸다[1],[2].

$$\begin{aligned} \text{cost1: } 1) A &= 2^{k_0} (2^{k_1} \pm 2^{k_2}) & (8) \\ \text{cost2: } 1) A &= 2^{k_0} (2^{k_1} \pm 2^{k_2} \pm 2^{k_3}) \\ & 2) A = 2^{k_0} (2^{k_1} \pm 2^{k_2}) (2^{k_3} \pm 2^{k_4}) \\ \text{cost3: } 1) A &= 2^{k_0} (2^{k_1} \pm 2^{k_2} \pm 2^{k_3} \pm 2^{k_4}) \\ & \cdot \\ & \cdot \end{aligned}$$

또한 CSD표현[6]이나 Bull과 Horrock[1]이 사용한 곱셈블록의 그래프 표현방식 역시 2^n 의 쉬프트연산을 사용해 계수를 구성하는 요소를 만들 수 있다. 그러나 다른 알고리즘과 MAG 알고리즘의 차이는 덧셈연산과정을 효율적으로 줄이는데 있다. 본 논문에서 실제로 적용된 계수 158을 생성하는 과정을 2진 표현과 MAG 알고리즘으로 그림 1에 비교하여 나타냈다.

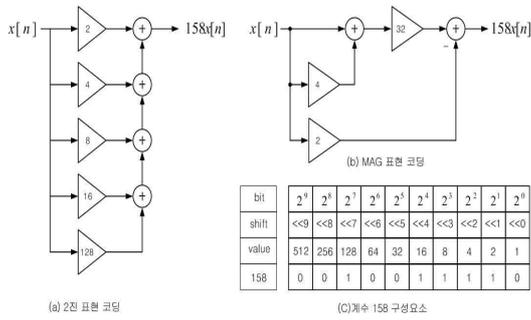


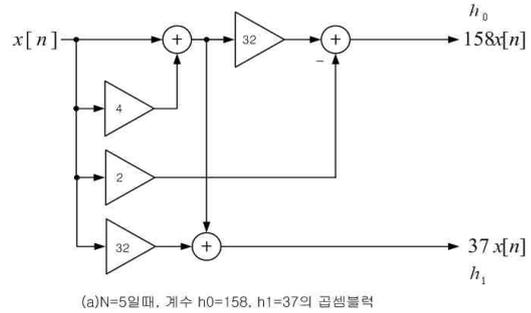
그림 1. 계수 158의 MAG 표현과 2진 표현의 비교 (a) 2진표현 (b) MAG 표현 (c) 계수 158의 구성요소

Fig 1. comparison of coefficient 158 in representation of MAG and binary. (a) representation of binary. (b) representation of MAG.

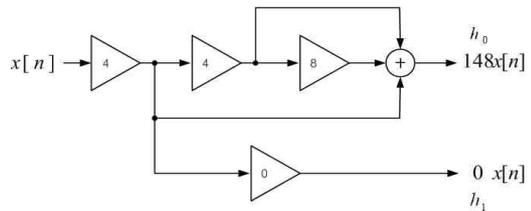
그림 1의 (a)는 계수 158에 대한 2진 표현 방식의 각 구성요소들 합으로 구현되며, (b)는 식 (8)에 의한 단계로 만들어 진다. 여기에서 (a)와 (b)의 각 구성요소는 그림 1의 (c)에서 2^n 의 쉬프트로서 구성된다. 그러므로 (a)와 (b)는 덧셈연산에서 차이를 보이며, 또한 계수를 구성하는 각 요소의 수에서도 MAG 알고리즘의 효율성을 확인 할 수 있다.

3.2. 제안된 알고리즘이 적용된 힐버트 변환기

식 (7)의 일반화적인 설계는 곱셈기를 사용하여 구현되지만 제안된 알고리즘에 의한 설계는 곱셈기를 곱셈블록으로 대체하여 설계된다. 본 논문에서 N=5, 6일 때 계수를 사용하여 설계한 힐버트 변환기의 각 곱셈 블록도는 다음에 나타낸 그림 2와 같이 설계된다.



(a)N=5일때, 계수 h0=158, h1=37의 곱셈블럭



(b)N=6일때, 계수 h0=148, h1=0의 곱셈블럭

그림 2. N=5, 6 일 때, 각 곱셈 블록도 (a) N=5 일 때, 계수 $h_0 = 158, h_1 = 37$ 의 곱셈 블록 (b) N=6 일 때, $h_0 = 148, h_1 = 0$ 의 곱셈 블록

Fig 2. each of multiplier block diagram, where N=5, 6 (a)coefficient h_0, h_1 of multiplier block, where N=5 (b)coefficient h_0, h_1 of multiplier block, where N=6

그림 2에서 사용된 계수는(N=5 일 때, $h_0 = 158, h_1 = 37$, N=6 일 때, $h_0 = 148, h_1 = 0$) 식 (3)의 $h(k)$ 값을 구하는 수식에 의해 계산된 계수 값으로 정수연산을 위해 256을 곱하여 얻어진 값이다. 그림 2의 (a)와 (b)의 곱셈 블록을 식 (7)의 수식에 각각 적용하면 아래와 같은 식 (9)와 식 (10) 같이 N=5, 6 일 때, 힐버트 변환을 설계할 수 있다.

$$\begin{bmatrix} y(0) \\ y(1) \\ \cdot \\ \cdot \\ y(4) \end{bmatrix} = \begin{bmatrix} (h_0x(1) - h_0x(4)) + (h_1x(2) - h_1x(3)) \\ (h_0x(2) - h_0x(0)) + (h_1x(3) - h_1x(4)) \\ (h_0x(3) - h_0x(1)) + (h_1x(4) - h_1x(0)) \\ (h_0x(4) - h_0x(2)) + (h_1x(0) - h_1x(1)) \\ (h_0x(0) - h_0x(3)) + (h_1x(1) - h_1x(2)) \end{bmatrix}$$

N=5 (9)

$$\begin{bmatrix} y(0) \\ y(1) \\ \cdot \\ \cdot \\ \cdot \\ y(5) \end{bmatrix} = \begin{bmatrix} (h_0x(1) - h_0x(5)) + (h_1x(2) - h_1x(4)) \\ (h_0x(2) - h_0x(0)) + (h_1x(3) - h_1x(5)) \\ (h_0x(3) - h_0x(1)) + (h_1x(4) - h_1x(0)) \\ (h_0x(4) - h_0x(2)) + (h_1x(5) - h_1x(1)) \\ (h_0x(5) - h_0x(3)) + (h_1x(0) - h_1x(2)) \\ (h_0x(0) - h_0x(4)) + (h_1x(1) - h_1x(3)) \end{bmatrix}$$

N=6 (10)

식 (9)와 (10)에서 제안된 알고리즘에 의해 대체된 곱셈 블록의 연산으로 h_0 와 h_1 은 입력 $x[0,1,\dots,N-1]$ 와 곱셈연산과 같은 결과를 보인다. 즉, N=5일 때, 입력 $x(0)$ 은 곱셈블록의 연산과정을 거치면 $x(0) \cdot h_0(158)$, $x(0) \cdot h_1(37)$ 의 2번의 곱셈기를 사용한 결과와 동일한 출력을 1번의 곱셈블록과정을 통해 출력하게 된다.

IV. 결과 및 고찰

본 연구에서 설계한 이산 힐버트 변환기의 개발환경은 Xilinx FPGA Virtex-4(xc4vlx100-10)와 Xilinx ISE9.2i에서 VHDL로 구현하였으며, Modelsim6.1f을 사용하여 시뮬레이션 하였고, 검증된 결과를 일반적인 방법과 제안된 방법으로 설계된 N=5, 6 일 때의 힐버트 변환기의 게이트 사용에 대한 비교를 표 1과 표 2에 나타냈다. 또한 그림 3의 (a)와 (b)에 제안된 설계에 사용된

게이트 소자를 그래프로 비교하였다.

표 1. N=5일 때, 제안된 알고리즘이 적용된 힐버트 변환기의 게이트 소자 사용 비교

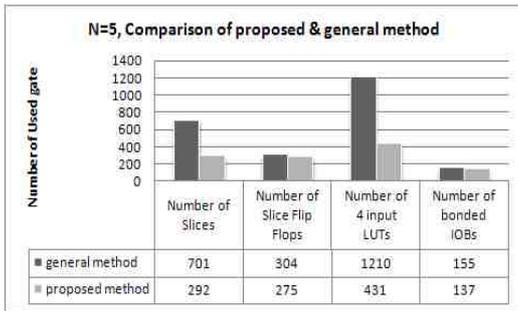
Table 1. comparison of used gate which is Using the proposed algorithm in Hilbert transformer, where N=5.

N	General method		Proposed method	
		Number of Slices	701	Number of Slices
N=5	Number of Slice Flip Flops	304	Number of Slice Flip Flops	275
	Number of 4 input LUTs	1210	Number of 4 input LUTs	431
	Number of bonded IOBs	155	Number of bonded IOBs	137

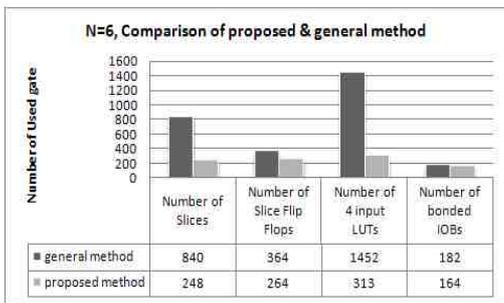
표 2. N=6일 때, 제안된 알고리즘이 적용된 힐버트 변환기의 게이트 소자 사용 비교

Table 1. comparison of Used gate which is Using the proposed algorithm in Hilbert transformer, where N=6.

N	General method		Proposed method	
		Number of Slices	804	Number of Slices
N=6	Number of Slice Flip Flops	364	Number of Slice Flip Flops	264
	Number of 4 input LUTs	1452	Number of 4 input LUTs	313
	Number of bonded IOBs	182	Number of bonded IOBs	164



(a) N=5일 때, 제안된 설계의 게이트소자 사용 비교 그래프



(b) N=6일 때, 제안된 설계의 게이트소자 사용 비교 그래프

그림 3. N=5, 6의 게이트소자 비교 그래프

Fig 3. comparative graph of used gate, where N=5, 6. (a) comparative graph in used gate of proposed design. (b) comparative graph in used gate of proposed design.

N=5, 6 일 때, 결과에서 알 수 있듯이 N값이 증가함에 따라 힐버트 변환기에 사용되는 곱셈기의 수는 증가 한다. 반면에 제안된 힐버트 변환기는 곱셈 블록을 사용함으로써 곱셈기의 효과를 내지만 곱셈기는 사용하지 않는다. 그 결과 곱셈기에 의한 게이트 소자의 사용을 크게 줄일 수 있으며, 이에 따라 동일한

출력결과를 내는 힐버트 변환기 설계에서 MAG 알고리즘에 의한 하드웨어의 감소를 확인 할 수 있다.

V. 결론

본 연구에서는 MAG 알고리즘을 사용한 곱셈 블록을 힐버트 변환에 적용하여 N=5, 6 일 때, 제안된 설계방법에 의해 하드웨어의 복잡도와 게이트 사용을 감소시킬 수 있는 방법을 제안하고 구현하여 시뮬레이션 결과를 통해 실제 하드웨어 복잡도와 게이트사용을 줄이는 것을 확인하였다. 제안된 결과를 바탕으로 SOC 분야의 하드웨어 설계에서 복잡도와 연산량 및 게이트 소자 사용의 감소에 의해 개발비용 절감과 저 전력 설계에 큰 효과를 가져 올 것으로 판단된다.

참고문헌

- [1] D. R. Bull and D. H. Horrocks, "Primitive operator digital filter," *IEEE Proc. G.*, vol. 138, no. 3, pp. 401 - 12, June 1991.
- [2] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst.*, vol. 42, no. 9, pp. 569 - 77, Sept. 1995.
- [3] Dutta Roy, S. C. 1976. Alternative matrix formulation of the discrete Hilbert transform. *Proc. IEEE*,

64:1435.

[4] Digital Filters Theory and Applications
N.K. Bose Professor of Electrical
engineering and Mathematics
University of Pittsburgh Pitts Burgh,
Pennsylvania, 103:108.

[5] D. Kodek and K. Steigliz, "Comparison of Optimal and Local Search Methods for Designing Finite Wordlength FIR Digital Filters," IEEE Transactions on Circuits and Systems, Vol. 28, pp.28-32, Jan. 1981.

[6] A. de la Serna and M. A. Soderstrand, "Tradeoff Between FPGA Resource Utilization and Roundoff Error in Optimized CSD FIR Digital Filters," IEEE Asilomar Conference, Vol. 1, pp. 187-191, 1994.

[7] I. Richard and Hartley, "Subexpression Sharing in Filters Using Canonic Signed Digit Multipliers," IEEE Transaction on Circuits and Systems, Vol. 43, No. 10, Oct. 1996.

[8] P.K. Dutta and P.B. Dutttagupta, "Optimization Method for Broadband Modem FIR Filter Design Using Common Subexpression Elimination," IEEE Instrumentation and Measurement Technology Conference, Vol. 3, P. 1321-1324, 1994.

저자약력

이영석(Youngseock Lee)



1993년 서울시립대학교
전자공학과(공학사)

1995년 서울시립대학교
전자공학과
(공학석사)

1998년 서울시립대학교
전자공학과
(공학박사)

1998년~ 현재
청운대학교

디지털방송공학과 교수

<관심분야> SOC, 임베디드 시스템, 의용생체시스템, VLSI 신호처리