

# MAG 알고리즘을 이용한 힐버트 변환기의 설계에 관한 연구

이영석\*

## A Study on the design of Hilbert transformer using the MAG Algorithm

Young-seock 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** : 힐버트 변환, SOC

### 1. 서 론

디지털 미디어기기의 발달로 작고 빠르며 복잡한 하드웨어를 하나의 칩 위에 구현하는 SOC(system on chip) 기술이 많이 연구되고 있다. 여러 시스템이 하나의 칩 위에 설계 될 때 하드웨어가 복잡하게 되면 높은 전력 소비와 생산비용이 늘어나게 된다. 현대의 디지털 통신 및 멀티미디어기기는 높은 성능과 낮은 복잡도의 디지털 신호 처리(DSP : digital signal processing)시스템을 필요로 한다. 최근에 DSP 시스템의 성능향상을 위해 높은 샘플링 비율과 많은 데이터의 사용으로 연산량이 증가하고 이에 따라 하드웨어의 복잡도가 증가 하였다. 그러므로 DSP 시스템에서 하드웨어 복잡도를 줄이고 성능을 향상시키기 위한 다양한 연구가 수행

되고 있다[5][6][7][8]. 특히, DSP 시스템에서 널리 사용되고 있는 힐버트 변환을 하드웨어로 구현하는 경우에 곱셈연산을 효율적으로 연산하는 MAG(multiplier adder graph) 알고리즘[1][3]은 곱셈연산을 쉬프트와 덧셈으로 대체함으로써 하드웨어의 복잡도를 줄일 수 있다. 본 논문은 MAG 알고리즘을 힐버트 변환의 곱셈연산에 적용하여 하드웨어 사용을 줄이는 방법을 제안하였다. 제안된 방법은 Xilinx사의 Virtex-4 시리즈의 FPGA(xc4lx100-10)와 통합개발환경으로 Xilinx ISE 9.2i와 Modelsim6.1f를 이용하여 구현하고 검증하였다. 구현된 힐버트 변환에 대해서는 일반적인 곱셈연산을 사용하여 구현한 설계와 게이트수를 비교하여 제안한 방법이 하드웨어의 복잡도를 감소시킨다는 것을 확인하였다.

\* Corresponding Author: Electronic Engineering Professor of Chungwoon University (yslee@chungwoon.ac.kr)

Received : August 18, 2014

Revised : August 29, 2014

Accepted : September 12, 2014

## II. 힐버트 변환

실함수(real function)  $x(t)$ 의 힐버트 변환 쌍  $\tilde{x}(t)$ 는 다음과 같이 정의 된다.

$$x(t) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{y(\tau)}{t-\tau} d\tau \quad (1)$$

$$y(t) = \tilde{x}(t) = H[x(t)] = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \quad (2)$$

위 식에서  $y(t) = H[x(t)] = (1/\pi t) * x(t)$ .  
 로서  $H$ 는  $x(t)$ 의 컨벌루션(convolution) 연산  
 으로 정의 되고, 이 때 실함수  $x(t)$ 와  $\tilde{x}(t)$ 에  
 의해 구성된 복소함수는 해석함수 (analytic  
 function)가 된다. 식 (2)를 이산 힐버트 변환으  
 로 정의하면  $y(k) = IDFT[H(k)x(k)]$ 으로 정  
 의되며 이 수식에 의해 유도되는 이산 힐버트  
 변환은 다음과 같이 정의 된다.[4]

$$y(k) = \sum_{s=0}^{N-1} x(s)h(k-s) \quad (3)$$

$$\text{여기서, } h(k) = \frac{2}{\pi} \sum_{r=1}^M \sin \frac{2\pi rk}{N},$$

$$\begin{cases} M \triangleq N/2 - 1 & \text{for } N \text{ even} \\ M \triangleq (N-1)/2 & \text{for } N \text{ odd} \end{cases}$$

식 (3)은 행렬로 정리하면, 다음과 같이 나타  
 낼수 있다.

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} h_{k0} \\ \vdots \\ h_{kN-1} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} \quad (4)$$

위 식에서  $k=0,1,\dots,N-1$  및  $s=0,1,\dots,N-1$   
 에 대하여  $H \triangleq [h_{ks}]$ 로 정의된다. 이 때  $h_{ks}$  다음  
 과 같은 특징을 갖는다.

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

식 (3)에서  $N=5$ 일 때를 가정하면  $h_{ks}$  값은  
 $h(0)=0, h(1)=0.6156, h(2)=-0.1452, h(3)=0.1452,$   
 $h(4)=-0.6156$ 이며  $H$ 는 식 (5)의 특징을 사용하  
 여 다음과 같이 표현할 수 있다.

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

여기에서  $h_{ks}$ 는  $h_{01} \triangleq h_0, h_{02} \triangleq h_1$ 으로 정의 한  
 다. 식 (6)에서  $N=5$ 일 때  $H$ 와  $x(k)$ 에서 20번  
 이상의 곱셈연산이 수행된다. 그러나 Dutta  
 Roy[2]는 식 (4)와 식 (6)의 행렬을 2개의 곱셈  
 요소로 변환하여 곱셈을 줄이는 방법을 식 (7)과  
 같이 나타낼수 있다.

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

또한 Bart Rice의 결과로부터 모든 출력의 합  
 은  $y(0) + y(1) + y(2) + y(3) + y(4) = 0$ 이며,  
 식 (7)의 출력을 7번의 곱셈으로 계산하는 알고  
 리즘을 식 (9)와 식 (10)에 나타내었다.

$$\begin{aligned} m_1 &= h_1(x(1) - x(0)) \\ m_2 &= h_1(x(3) - x(4)) \\ m_3 &= h_0(x(2) - x(0)) \\ m_4 &= (h_0 + h_1)(x(3) - x(1)) \\ m_5 &= h_0(x(4) - x(2)) \\ m_6 &= h_0(x(0) - x(3)) \\ m_7 &= h_1(x(0) - x(2)) \end{aligned} \quad (9)$$

$$\begin{aligned}
 y(0) &= -(y(1) + y(2) + y(3) + y(4)) \quad (10) \\
 y(1) &= m_2 - m_3 \\
 y(2) &= m_1 - m_2 + m_4 \\
 y(3) &= m_5 - m_1 \\
 y(4) &= m_1 - m_6 + m_7
 \end{aligned}$$

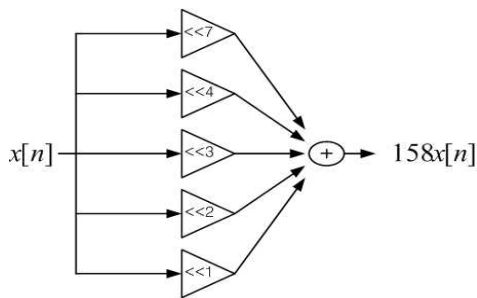
위에 정리된 식 (9)와 식 (10)의 알고리즘을 사용하여 일반적인 곱셈으로 힐버트 변환을 설계 할 수 있다.

### III. MAG 알고리즘을 적용한 힐버트 변환

그림 1에서 적용된 MAG 알고리즘에서 계수 158과 입력  $x(n)$ 의 곱셈을 만들어 내는 과정을 나타 내었다. 그림1의(a)에서 계수 158은  $128+16+8+4+2=158$ 와 같이 각 요소의 합으로 구성될 수 있다. 또한 각각의 요소는 그림 1의 (b)와 같이 <<로 표현된 쉬프트 연산으로 구현할 수 있다.

bit	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
shift	<<9	<<8	<<7	<<6	<<5	<<4	<<3	<<2	<<1	<<0
value	512	256	128	64	32	16	8	4	2	1
158	0	0	1	0	0	1	1	1	1	0

(a) 계수 158의 2진 표현 구성요소



(b) 쉬프트와 덧셈으로 구성되는 계수 158  $x[n]$

그림 1. MAG 알고리즘을 사용한 계수 158

식 (9)에 정리된 수식을 이용하여 그림 2의 블록 다이어그램으로 나타 내었다. 또한 그림 1의 (b)에서 MAG 알고리즘을 사용한 쉬프트와 덧셈으로 계수  $h_0, h_1$ 를 만들어 내고 그림 2의 부분 곱( $m1 \sim m7$ )을 사용하여 식 (10)의 출력을 얻도록 설계하였다.

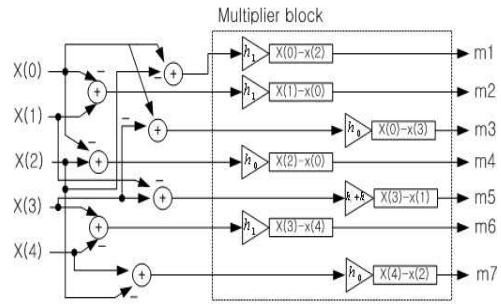
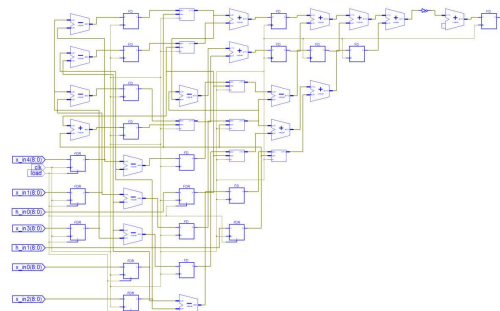


그림 2. MAG가 적용된 곱셈 연산 과정

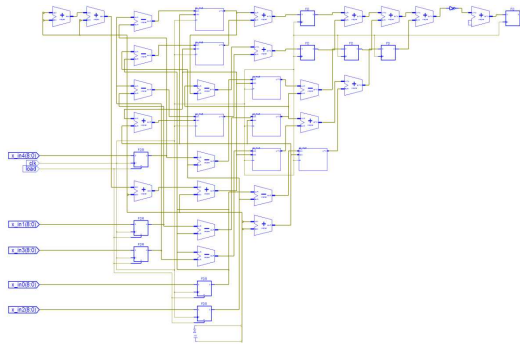
본 논문에서는 그림 2에서 나타낸 곱셈블록에 MAG 알고리즘을 적용하고 쉬프트와 덧셈으로 계수를 만들어 내는 곱셈연산으로 하드웨어 복잡도가 낮은 힐버트 변환기를 구현하였다.

### IV. 실험 및 결과고찰

본 연구에서 설계한 힐버트 변환은 Xilinx FPGA Virtex-4(xc4vlx100-10)와 Xilinx ISE9.2i에서 VHDL로 구현하였으며, Modelsim6.1f를 사용하여 시뮬레이션 하여 결과를 검증하였다.



(a) 일반 곱셈기를 사용한 경우



(b) MAG를 이용한 곱셈기를 적용한 경우

그림 3.  $N=5$  일 때 힐버트 변환의 구현

그림 3의 (a)와(b)의 비교에서 그림 3의 (b)에 적용된 MAG 알고리즘에 의해 설계된 힐버트 변환기에서 사용된 소자의 차이를 표 1를 통해 확인 할 수 있다.

표. 1 그림3에서 (a)와 (b)의 게이트 소자 비교

Logic Utilization	proposed method	General method
Number of Slices	144	200
Number of Slice Flip Flops	135	225
Number of 4 input LUTs	238	247
Number of bonded IOBs	142	160
Number of GCLKs	1	1
Number of DSP48s	7	7

### V. 결 론

본 연구에서는 MAG 알고리즘을 사용한 곱셈 블록을 힐버트 변환에 적용하여 하드웨어의 복잡도를 감소시킬 수 있는 설계방법을 제안하고 구현하였으며, 구현된 하드웨어에 대해서 일반적인 곱셈연산을 사용하여 구현한 설계와 비교하

여 시뮬레이션 결과를 검증하였다. 향후 연구 과제로서는 CSD 표현을 사용하고 곱셈기가 없는 하드웨어 구현에 중점을 두고 연구를 수행하고 있다.

### Reference

- [1] D.R. Bull and D. H. Horrocks, "Primitive Operator Digital Filters," IEEE Proceedings G, 138(3):401-412, June 1991.
- [2] Dutta Roy, S. C. 1976. Alternative matrix formulation of the discrete Hilbert transform. Proc. IEEE, 64:1435.
- [3] U. Meyer - Baese Digital Signal Processing With Field Programmable Gate Arrays. second Editon. Springer-verlag, Berlin Heidelberg New York, 2nd ed, 2004.
- [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

---

이 영 석 ( Yeong-seok Lee )

평생회원



86 - 93 서울시립대학교 공학사  
93 - 95 서울시립대학교 공학석사  
95 - 98 서울시립대학교 공학박사  
98 - 현재 청운대학교 부교수  
07 - 08 청운대학교 학과장  
2007 - 현재 조달청 영상장비도  
입 조달심위위원  
2008 - 현재 정보통신연구진흥  
원 생체계측 기술 개발  
사업 심위위원  
2009 - 현재 정보전자통신기술  
학회 정보이사  
<관심분야> 디지털 회로 설계, 신호 및  
시스템, 마이크로프로세서