

근전도 신호 처리를 위한 적응 필터의 VHDL 구현

김정섭⁰, 이석필, 박상희
연세대학교 전기공학과

Implementation of the adaptive filter for EMG signal processing using VHDL

Jung-Sub Kim⁰, Seok-Pil Lee, Sang-Hui Park
Dept. of Electrical Eng., Yonsei Univ.

Abstract

We present the implementation of the adaptive filter for EMG signal processing using VHDL. For making ASIC, the basic FPU(floating point processor), e.g., adder, multiplier and divider, are implemented with VHDL. The FPU is simulated and the controller for the RLSL(recursive least square lattice) algorithm of the adaptive filter is implemented. Then FPU and the controller are linked and simulated. Finally the models are synthesized and the gate level is implemented.

1. 서론

보철 제어를 위해서는 근전도(electromyographic : EMG) 신호 처리가 많이 쓰여져 왔다[1]. 근전도 신호는 시간에 따라 통계적 특성이 변화하는 비안정(nonstationary) 신호이다[2]. 이러한 신호들에 대하여, 서로 다른 기능에 의한 신호의 정확한 필터링과 인식을 하기 위해서는 엄밀한 통계적인 해석이 요구된다. 그러나, 이것은 최적 블록 알고리즘 필터로는 한계가 있다. 그러므로 시간에 따라, 그때 그때의 파라미터를 추출할 수 있는 적응 필터 알고리즘이 사용되어야 한다. 적응 필터는 그 동작을 회귀 알고리즘(recursive algorithm)에 의존하여 수행하며, 이는 관련 신호의 특성을 완전히 알지 못하는 환경에서 필터가 만족스럽게 동작될 수 있도록 해 준다. 본 연구에서는 회귀 알고리즘 중에서 모듈 구조로 되어 있어서 하드웨어 구현시 적용이 용이하고 수렴성이 우수한 RLSL(recursive least square lattice) 알고리즘을 채용하고, 이를 ASIC(Application Specific Integrated Circuit)으로 구현하고자 한다.

적용 필터에 적용되는 알고리즘은 정수가 아닌 수들을 처리하고, 또한 넓은 범위의 값을 표현해야 하는 정밀 연산 부분이 들어가게 된다. 따라서 이런 연산을 해주기 위해서는 부동 소수점 연산이 필수적으로 필요하게 된다. 1985년에 IEEE(Institute of Electrical and Electronic Engineers)에서는 이진 부동 소수점 연산에 대한 표준을 정하였다[6]. 이 표준안에 따르면, 부동 소수점 수치는 분수부(fraction part)와 지수부(exponent part), 그리고 부호(sign)로 저장되어 있다. 따라서, 이러한 표준에 맞도록 연산기를 만들어 주어야 한다.

그러므로, 본 논문에서는 RLSL 알고리즘을 이용한 적응 필터를 구현하기 위해서, VHDL을 이용해서 먼저 기본적인 부동 소수점 연산기를 기술한 후에, 시뮬레이션을 통하여 부동 소수점 덧셈기와 곱셈기, 그리고 계산기의 연산을 검증하고자 한다. 또한 각각의 연산기를 연결하는 제어기를 기술하여 전체 적응 필터를 기술한 후에 검증하고자 한다. 마지막으로 합성 툴(synthesis tool)을 사용하여 논리 합성하여 게이트 레벨로 구현하고자 한다.

2. RLSL 알고리즘의 분석

근전도 신호는 근본적으로 시변(time varing) 신호이며 비안정(nonstationary) 신호이다. 그러므로 시간에 따른 정확한 특징을 추출하려면 적응 필터 알고리즘이 적합하다.

적용 필터의 알고리즘의 한 종류인 RLSL(Recursive Least-Square Lattice) 알고리즘의 중요한 장점은 구조가 모듈(module)로 되어 있다는 것이다. 따라서 예측 차수를 증가시킬 때 모든 이전 값을 다시 계산할 필요가 없다. 또한 RLSL 알고리즘은 모듈 구조이므로 하드웨어 구현시 VLSI 기술에 적용이 용이하다. 본 연구에서는 하드웨어 설계에 용이한 RLSL 알고리즘을 이용하기로 한다.

순방향 a posteriori 예측 오차(forward a posteriori prediction error)를 $f_m(n)$, 역방향 a posteriori 예측 오차(backward a posteriori prediction error)를 $b_m(n)$, 순방향 반사 계수(forward reflection coefficient)를 $\Gamma_{f,m}(n)$, 역방향 반사 계수(backward reflection coefficient)를 $\Gamma_{b,m}(n)$ 라고 하면 다음과 같은 식이 성립한다[3].

$$f_m(n) = f_{m-1}(n) + \Gamma_{f,m}(n)b_{m-1}(n-1), \quad m=1,2,\dots,M \quad (2.1)$$

$$b_m(n) = b_{m-1}(n-1) + \Gamma_{b,m}(n)f_{m-1}(n), \quad m=1,2,\dots,M \quad (2.2)$$

여기서, 초기조건은 식 2.3와 같다.

$$f_0(n) = b_0(n) = u(n) \quad (\text{단, } u(n) = \text{입력값}) \quad (2.3)$$

식 2.1과 식 2.2로 표현되는 RLSL 알고리즘의 신호 흐름 그래프는 그림 1과 같다. 또한 예측 차수가 M차인 다단계 RLSL 예측기는 그림 2와 같다.

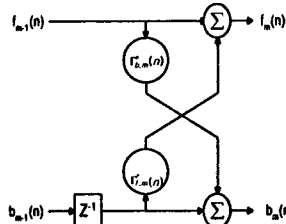


그림 1 RLSL 알고리즘의 표현

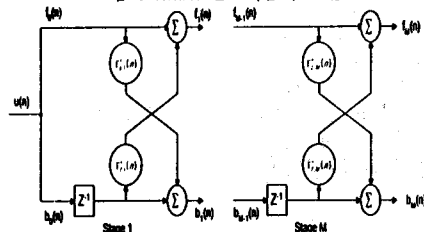
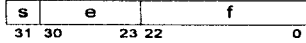


그림 2 다단계 RLSL 예측기

3. 적응 필터의 설계

3.1 부동 소수점 표준안

본 논문의 부동 소수점 연산기에서 사용되는 부동 소수점 데이터 형태는 IEEE-754 표준 중 단정도 연산을 따르는 데, 아래와 같이 세 가지 영역(field)으로 구성되어 있다[6].



- (1) s : 1 비트로 나타내어지는 부호(sign)
- (2) e : 바이어스(bias)된 지수(exponent)
= E + 바이어스(단, E = 바이어스되지 않은 지수)
- (3) f : 가수(fraction)

이러한 세 가지 영역으로 나타낼 수 있는 실재값(Value)은 다음과 같이 표시된다.

$$Value = (-1)^s \cdot 2^E \cdot (1.f)$$

또한, 반올림 모드는 반올림으로 인한 오차가 최소가 되도록 가까운 쪽으로 반올림하는 방법인 Round to Nearest 모드를 사용한다.

3.2 부동 소수점 덧셈기의 설계

부동 소수점 덧셈을 수행하기 위해서는 먼저 자리맞춤을 해주어야 한다. 즉, 첫 번째로 지수부를 계산해서 지수값이 큰 쪽에 맞게 지수값이 작은 쪽의 가수를 그 차이만큼 우로 이동(shift)해 주어야 한다. 또한 이때, 우로 이동한 비트들을 반올림 모드에 따라 반올림 여부를 결정해 주어야 한다. 그런 후에 24 비트 덧셈기에 넣어 계산을 수행하고, 그 계산 결과를 부동 소수점 표현 형태에 맞도록 다시 변형해서 최종 결과를 얻는다. 즉, 출력값 25비트 중에서 맨 처음 1이 되는 값을 찾아서 그 이후부터 가수로 표현하고, 그 자리수만큼 지수값을 조정해 주어야 한다.

3.3 부동 소수점 곱셈기의 설계

먼저 부호부에 대한 계산은 논리합을 써서 설계하고, 다음에는 가수부에 대한 계산을 해주어야 한다. 가수부는 입력이 24 비트씩 들어오고, 출력은 48비트가 나오는 곱셈기를 만든다. 곱셈기는 더하고 쉬프트하는 방식을 사용하지 않고, 수정된 booth 알고리즘을 이용하여 보다 빠른 시간에 계산을 수행하도록 한다. 마지막으로 지수부의 계산을 수행하는데, 두 지수값에 대해 덧셈을 해주고, 가수부의 계산 결과에 따라 Carry In을 결정하여 계산한다.

3.4 부동 소수점 제산기의 설계

부동 소수점 제산기의 구현은 가수부에서 곱셈기 대신에 제산기를 써야 하는 것과 지수부에서 뺄셈을 해주어야 하는 것을 제외하고는 곱셈기와 거의 구조가 비슷하다. 가수부에서의 제산기는 일련의 뺄셈과 쉬프트 방식으로 구현하는 데, 입력 48비트, 24비트에 대하여, 출력 48비트가 나오도록 설계한다. 나온 결과에 따라 역시 지수부의 자리수를 다시 계산해 주어야 한다.

3.5 RLSL 제어기의 설계

RLSL 제어기는 2장에서 설명한 알고리즘을 하드웨어로 구현하였다. 이 적응 필터는 차수가 12차이고, 총 데이터 개수는 64개이다. 따라서 이것을 제어해 주기 위하여 처음에 입력을 64개 받고, 다음에 각 계수들을 초기화시킨 후 계수 갱신 방법에 따라서 각 계수들을 갱신한다. 계수 갱신은 먼저 각 차수에 대하여 실행하여 마지막 차수까지 갱신한 후에, 다시 다음 데이터에 대하여 차수 갱신을 반복하는 형태이다. 이런 처리를 해주기 위해서 제어기에 상태 변수를 두어 갱신 순서에 따라 상태 변수를 변화시켜 가면서 제어를 한다.

3.6 적응 필터의 설계

전체 필터는 부동 소수점 가산기 2개, 곱셈기 2개, 제산기 2개를 포함하고, 그 중간에 RLSL 제어기를 넣어서 서로를 연결하고 있다. 데이터 저장용 위한 램은 128K짜리 2개를 외부에 연결하고, 또한 클럭도 외부에서 연결하였다. 각각을 2개씩 쓴 이유는 계산이 2개씩 병렬로 처리되기 때문에, 한 번에 2개의 연산을 같이 수행하여 보다 빠르게 처리하기 위함이다. 본 적응 필터의 블록 다이어그램은 그림 3과 같다.

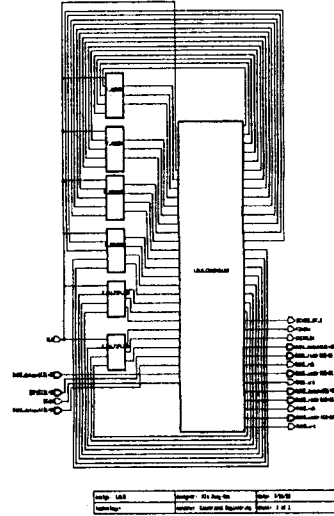


그림 3 적응필터의 블록 다이어그램

4. 시뮬레이션 및 결과 고찰

본 논문에서는 모두 4개의 블록을 설계한 후, 연결하여 전체 블록을 구성하였다. 각 블록마다 테스트 벡터에 대하여 시뮬레이션을 수행해서, 각 블록을 검증하였다. 또한, 전체 블록에 대하여 64개의 입력 테스트 벡터에 대하여 시뮬레이션을 수행하여 검증하였다. 이에 대한 소프트웨어는 Synopsys simulation tool을 사용하였다. 그 중에서 최상위 블록(적응 필터)에 대한 시뮬레이션 결과는 그림 4과 같다. 그림 4는 29번째 데이터에 대한 7차 계수를 갱신하는 과정을 RLSL 제어기의 입출력에 대해서 보여준 예이다. 이와 같은 방법으로 본 논문에서 구현한 VHDL 모델에 대한 검증하였다.

한편, 검증이 완료된 VHDL 모델에 대하여, Synopsys에서 기본으로 제공하는 library를 사용해 Synopsys synthesis tool을 이용하여 논리 합성을 수행하였다. 각 블록의 gate area는 표 1과 같다.

부동 소수점 덧셈기	6,061
부동 소수점 곱셈기	5,856
부동 소수점 제산기	16,109
RLSL 제어기	11,934
적응 필터	67,986

표 1 각 블록에 대한 Gate Area

또한, 각 블록에 대한 논리 합성 결과는 그림 5와 같다.

각 블록에 대한 검증은 마친 후에, 전체 블록에 대하여 한 테스트 벡터에 대해 시뮬레이션을 수행한 결과값을 C 언어로 기술된 RLSL 알고리즘의 결과값과 비교해 봄으로써 본 필터의 정확성을 검증하였다.

앞으로의 연구과제는 timing 시뮬레이션을 수행한 후, FPGA를 이용한 하드웨어 구현을 행하여, ASIC으로 구현하는 것이다.

참고 문헌

- [1] G. N. Saridis et al., "EMG pattern analysis and classification for prosthesis arm," IEEE Trans. on BME, Vol.29, pp.403-409, 1982
- [2] Harry. G. Kwantny et al., "An application of myoelectric signals", IEEE Trans. on BME, Vol.7, No.4, pp.303-312, Oct 1970
- [3] Simon Haykin : Adaptive Filter Theory, Prentice Hall, 1991
- [4] Keinosuke Fukunaga : Introduction to Statistical Pattern Recognition, Academic Press, 1990
- [5] 이용석, "60MHz clock 주파수의 IEEE 표준 Floating Point ALU", 전자공학회 논문지, Vol.28-A, No.11, Nov. 1990
- [6] IEEE std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic, IEEE, 1985
- [7] Zinalabedin Navabi, VHDL, Analysis and Modeling of Digital Systems, McGRAW-HILL International Editions, 1993
- [8] Douglas L. Perry, VHDL, McGRAW-HILL International Editions, 1991
- [9] K. Hwang, Computer Arithmetic : Principles, architecture and design, John Wiley and Sons, Inc., 1978

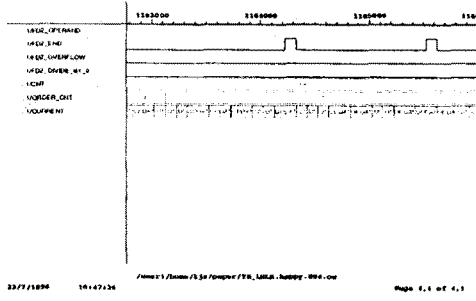
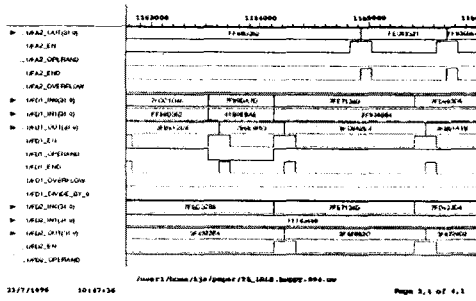
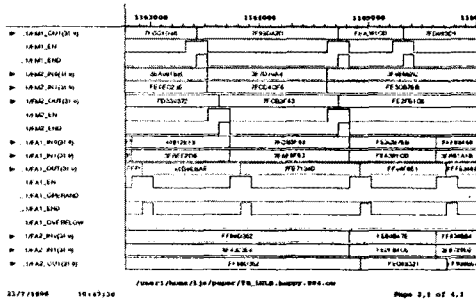
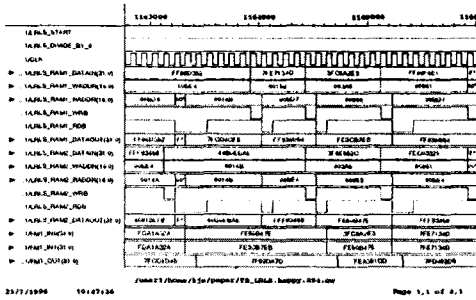


그림 4 최상위 블록에 대한 시뮬레이션 결과

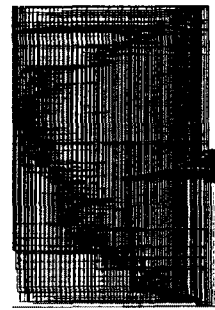
5. 결론

본 논문에서는 근전도 신호의 특징 파라메타 추출에 이용할 적용 필터를 VHDL을 이용하여 구현하였다. 실제적인 구현을 위해서 모듈형으로 되어있는 RLSL 알고리즘을 이용했으며, top-down 설계를 위하여 총 4개의 블록을 설계하였다.

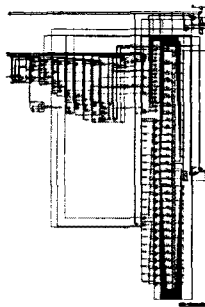
정밀연산을 위하여 부동소수점 연산기, 즉 덧셈기, 곱셈기, 계산기를 설계하고, 이런 블록을 RLSL 알고리즘에 따라 계산하게 하는 제어기를 설계했다. 각 블록에 대하여 합성(synthesis)이 가능하도록 모델링하고, 시뮬레이션과 합성을 행하였다.



(a) 부동 소수점 덧셈기



(b) 부동 소수점 곱셈기



(c) 부동 소수점 제산기



(d) RLSL 제어기

그림 5 각 블록에 대한 논리 합성

이 논문은 1994년도 한국학술진흥재단의 대학부설연구소 연구과제 연구비에 의하여 연구되었음.