

FPGA를 이용한 심전도 전처리용 적응필터 설계

한상돈 · 전대근 · 이경중 · 윤형로

연세대학교 보건과학대학 의공학과, 의공학 연구소, 연세대학교 의용계측 및 재활공학 연구센터
(2001년 2월 23일 접수, 2001년 6월 15일 채택)

Design of FPGA Adaptive Filter for ECG Signal Preprocessing

S.D. Han, D.G. Jeon, K.J. Lee, H.R. Yoon

Dept. of Biomedical Engineering, College of Health Science, Yonsei University
Research Institute of Medical Engineering,
Research Institute for Medical Instrumentation and Rehabilitation Engineering

(Received February 23, 2001. Accepted June 15, 2001)

요약: 본 논문에서는 다채널, 고해상도 사양의 심전계에서 사용되고 있는 신호처리 프로세서의 단점인 연산부하 증가문제를 최소화시킴으로 시스템의 성능을 향상시킬 수 있는 FPGA기반의 전처리용 적응필터(고역통과필터와 노치필터)를 설계하였다. 적응필터를 설계하기 위해 FPGA 설계도구(MAX-PLUS II, Altera corporation)를 이용하였으며, 테스트 데이터로 CSE 데이터 베이스를 사용하였다. 설계한 필터들의 성능평가는 기존의 신호처리 프로세서(ADSP21061, Analog Devices)와 처리속도를 비교하였으며, 그 결과 FPGA가 처리속도면에서 우수함을 보여주었으며, 고해상도 심전계의 전처리용 필터로 유용성을 보여주었다.

Abstract: In this paper, we designed two preprocessing adaptive filters - high pass filter and notch filter - using FPGA. For minimizing the calculation load of multi-channel and high-resolution ECG system, we utilize FPGA rather than digital signal processing chip. To implement the designed filters in FPGA, we utilize FPGA design tool(Altera corporation, MAX-PLUS II) and CSE database as test data. In order to evaluate the performance in terms of processing time, we compared the designed filters with the digital filters implemented by ADSP21061(Analog Devices). As a result, the filters implemented by FPGA showed better performance than the filters based on ADSP21061. As a consequence of examination, we conclude that FPGA is a useful solution in multi-channel and high-resolution signal processing.

Key words: FPGA, Adaptive filter, Digital signal processing chip

서 론

지난 수년간 고속의 실시간 처리가 요구되는 분야에서 application-specific standard products(ASSP)나 application-specific integrated circuit(ASIC)은 큰 역할을 담당해 왔는데, 그 이유는 일반적인 신호처리(DSP) 프로세서로는 구현할 수 없는 높은 성능을 발휘할 수 있었기 때문이다[1]. 그러나 ASSP나 ASIC은 특정 응용분야에 적용가능 하도록 설계된 것

이기 때문에, 범용성 측면에는 많은 제약이 있었다. 반면 신호처리 프로세서는 ASSP나 ASIC과 비교했을 때 그 성능은 떨어지지만, 거의 모든 응용분야에 사용될 수 있을 만큼 융통성이 뛰어난 장점으로 인해 널리 사용되어져 왔다. 이와 같은 기술 추세 속에 최근 몇 년간 모든 반도체 분야에서 시장성장률과 집적도 면에서 가장 높은 성장세를 보이며 등장한 것이 field programmable gate array(FPGA) 혹은 complex programmable logic device(CPLD)이며, 몇 년 전부터는 이러한 디바이스를 신호처리 분야에 응용하려는 시도가 이루어지고 있다[2]. 주로 ASIC을 제작하기 전 검증단계로 사용되던 FPGA를 신호처리 분야에 이용하려고 하는 이유는 과거 양분되었던 시스템 설계 방법론의 두 가지 상반된 측면인 높은 성능과 뛰어난 융통성을 하나의 방법론으로 통합하는 것이 가능해지기 때문이다[1][3].

본 연구는 과학기술부·한국과학재단지정 연세대학교 의용계측 및 재활공학연구센터(RRC)의 지원에 의한 것입니다.

통신저자: 윤형로 (220-710) 강원도 원주시 흥업면 매지리 234
연세대학교 백운관 RRC센터 101호 연구소장실
TEL. 033-760-2850, FAX. 033-763-1953
E-mail)hryoon@dragon.yonsei.ac.kr

대부분의 생체신호 계측분야에서 정확한 파라미터 추출을 위한 전처리 필터의 사용은 필수적이며, 점차적으로 시스템의 사양이 높아지면서, 전처리 과정에서 신호처리 프로세서를 사용하는 것이 일반화 되어가고 있는 추세이다. 그러나 신호처리 프로세서는 채널별 샘플율이 높고, 많은 수의 채널을 갖는 시스템에서 사용될 경우에는, 성능의 한계로 인해 여러개의 프로세서를 병렬로 사용하거나, 다른 방법을 이용해 시스템을 설계해야만 한다. 반면 FPGA는 신호처리 프로세서보다 빠른 연산 처리가 가능하고, 구조적으로 병렬처리가 간단히 이루어질 수 있기 때문에, 설계하려는 시스템에 가장 적합한 설계방법을 제시해 줄 수 있다. 따라서 본 논문에서는 FPGA를 생체계측 장비에 응용한 예로써, 진단용 심전계의 전처리 필터 중 고역통과 필터(highpass filter)와 노치필터(notch filter)를 FPGA에 구현하여 신호를 전처리함으로써, 신호처리 프로세서의 한계를 극복하고 시스템의 부하를 최소화하여, 전체적인 시스템의 성능향상을 이루려 한다.

본 론

1. 전처리 필터 설계

진단용 심전계의 전처리 필터 중 본 연구에서 설정한 테스트 시스템은 60Hz 전원 잡음을 제거하기 위한 노치필터와, DC 오프셋이나 기저선 변동을 제거하기 위한 고역통과 필터이다. 60Hz 노치필터의 경우 잡음이 신호의 주파수 대역과 겹쳐서 존재하며 요구되는 필터의 Q값이 높아서 일반적인 finite impulse response(FIR)나 infinite impulse response(IIR)로 필터를 설계하면 필터의 탭수가 많아져 시스템의 연산부하가 커지게 되고, 또는 필터의 비선형적인 위상응답으로 인해 신호의 왜곡을 가져올 위험이 있다[4][5]. 또 고역통과 필터의 경우에도 차단주파수가 매우 낮기 때문에, 노치필터를 설계하는 경우와 마찬가지로 실제적인 구현에 상당한 어려움이 있는데, 이러한 경우에 효과적으로 잡음을 제거하기 위한 방법으로 적응 필터가 널리 이용되고 있다[4][5][6]. 본 연구에서는 상기의 두 가지 필터를 least mean square(LMS) 알고리즘을 기반으로 설계하였으며, common standards for quantitative electrocardiography(CSE) 데이터베이스를 이용하여 성능을 테스트하였다.

1.1 고역통과필터 설계

진단용 심전계의 전처리에서 고역통과 필터의 역할은 DC 오프셋이나 기저선 변동을 제거하는 것이다. 이를 적응필터로 설계하기 위해 기저선 변동과 상관성이 있는 잡음을 기준입력으로 인가하여 주입력의 저주파 성분을 제거해야 하는데, 이때 기저선 변동은 0.05-5Hz 정도의 저주파 성분이므로 기준입력으로 상수 1을 인가하고 수렴계수 μ 를 조정함으로써 원하는 차단주파수를 갖도록 필터를 설계하였다. 실제적으로 상수는 DC 성분이지만 수렴계수를 조정함으로써 차단 주파수를 임의

적으로 설정할 수 있다. AHA(american heart association)에서는 기저선 변동을 제거하기 위한 필터를 사용할 때 ST 세그먼트의 왜곡을 방지하기 위해 0.8Hz 이하의 차단주파수를 갖도록 필터를 설계할 것을 권고하고 있다[6]. 따라서 본 절에서 설계한 필터는 식 (1)에 의해 샘플링 주파수 f_s 가 500Hz일 때, 차단 주파수 f_c 가 0.5Hz가 되도록 설계하였으며 이때 수렴계수 $\mu = \mu_h$ 는 0.00314가 된다.

$$f_c = \frac{\mu}{\pi} f_s \quad (1)$$

고역통과 필터는 LMS 알고리즘을 이용하여 식 (2)와 같이 가중치를 변경하게 된다.

$$w(k+1) = w(k) + 2\mu_n e(k) x(k) \quad (2)$$

여기서 $e(k)$ 는 주입력에서 기준입력의 필터 출력의 뺀 오차이며, $x(k)$ 는 기준입력으로 상수 1에 해당한다.

1.2 노치필터 설계

60Hz 전원 잡음을 제거하기 위한 노치필터는 다양한 생체신호 처리 분야에서 기본적인 전처리 과정으로 사용되고 있다. 이러한 노치필터를 적응필터로 설계할 때 기준입력은 60Hz 사인파를 사용한다. 본 연구에서 가정한 샘플링 주파수는 500Hz이므로 노치필터의 기준입력을 25개의 샘플로 이루어진 60Hz 사인파로 설정하였다. 여기서 수렴계수는 여러번의 실험에 걸쳐 적절한 값을 선정하였으며, 최종적으로 설계에 반영된 수렴계수 $\mu = \mu_n$ 의 값은 0.03125로 하였다. 설계한 노치필터는 LMS 알고리즘을 이용하여 식 (3)과 같이 가중치 벡터를 변경하도록 하였다.

$$\bar{w}(k+1) = \bar{w}(k) + 2\mu_n e(k) \bar{x}(k) \quad (3)$$

여기서 $\bar{w}(k)$ 는 25개의 가중치를 갖는 벡터이고, $e(k)$ 는 주입력에서 기준입력의 필터 출력의 뺀 오차이며, $\bar{x}(k)$ 는 기준입력으로써 60Hz 사인파에 해당하는 25개의 값으로 이루어진 벡터에 해당한다.

2. FPGA 기반의 필터설계

1절에서 설계한 적응필터를 FPGA에 구현하기 위해서는 필요한 연산과정을 비트단위로 세분화시키는 과정이 필수적이다. 각 연산과정별 플래그(flag)를 저장하기 위한 레지스터, 플래그의 상태에 따른 적절한 처리과정 설정, 캐시체인의 구조와 오버플로우 처리, 조합논리의 지연시간 차이를 흡수할 수 있는 버퍼의 위치와 동기화, 각 블록사이의 적절한 시퀀스 제어 등을 통한 정확한 연산과정 제어 등의 세부적인 구조가 적절히 설계되지 않는 한, 정확한 필터의 출력을 기대하기 힘들다. 또한

적응필터의 경우 특정 알고리즘에 의해 연산결과와 일부가 피드백되는 과정이 있기 때문에, FPGA로 FIR필터를 설계하는 경우와 달리 일정한 구조의 반복만으로 필터를 설계하기 힘들다. FPGA를 이용한 설계는 논리게이트 단위의 logic element(LE)를 가지고 이루어지기 때문에, 불필요한 LE의 사용을 줄이고, 각 블록사이의 지연시간을 최소화하기 위해서는, 위와 같이 모든 연산과정의 구조를 정확히 파악할 필요가 있다.

2.1 고역통과 필터의 설계

그림 1은 FPGA로 구현한 고역통과 필터의 내부 블록도이다. 고역통과 필터는 기준입력벡터가 상수 1로 구성되므로, 그림 1에 나타난 기준입력벡터와 수렴계수와와 곱셈연산이 불필요하고, 가중치 벡터도 하나의 원소로 구성되게 된다. 따라서 고역통과 필터를 구현하기 위해서는 현재 필터출력을 나타내는 뿔셈기의 출력과 수렴계수와와 곱셈연산을 할 수 있는 곱셈기, 가중치의 변경과정에 필요한 덧셈기, 변경된 가중치를 이용하여 필터의 출력을 계산하기 위한 뿔셈기가 필요하게 된다. 수렴계수는 일정한 값을 가지므로 설계한 곱셈기는 하나의 입력이 상수로 고정된 상수 곱셈기로 설계하였고, 다른 입력은 필터의 현재 출력값인 뿔셈기의 출력을 연결하였다. 이 곱셈기의 출력은 다음 샘플에 적용될 가중치를 변경하는데 사용되어지므로 설계한 덧셈기의 입력에 연결하였고, 덧셈기의 다른 입력은 이전 가중치 값을 가져야 하므로 가중치를 저장하는 레지스터의 출력을 연결하였다. 매번 입력되는 샘플은 뿔셈기의 두 개 입력중 하나에 연결되도록 하였으며, 다른 입력은 현재의 가중치가 인가될 수 있도록 하였다. 결과적으로 입력샘플에서 현재 가중치값을 빼서 필터출력을 내보내는 과정과, 다음 샘플이 들어오기 전까지 현재 출력으로 다음 가중치를 계산하는 과정이 병렬구조로 처리되어 하나의 사이클에 수행될 수 있도록 하였다.

전체적인 조합논리 회로의 동작을 일반적인 연결자원

(routing resource)보다 속도가 빠른 캐스캐이드 체인과 캐리 체인을 부분적으로 사용하여 설계하였으며, 시스템 클럭과 동기화된 레지스터를 설정하여 모든 연산과정이 하나의 클럭에 의해 이루어질 수 있도록 하였다. 또한 모든 연산과정의 경로를 하나의 연결형태로 고정하고, 멀티플렉싱에 의한 연산의 지연이 없도록 하였다. 연산목적의 회로설계에서 이러한 접근방법은 연산의 속도를 향상시킬 수 있을뿐만아니라, 부수적으로 제어리소스를 단순화할 수 있기 때문에 부가적인 LE의 사용을 피할 수 있게 된다. 최종적인 시스템은 FLEX 8000계열의 EPF8282ALC84-2 디바이스를 이용하였으며, 이 디바이스 갖고 있는 총 LE의 64%인 135개의 LE를 이용하여 설계하였다.

2.2 노치필터의 설계

노치필터는 기준입력벡터와 가중치벡터가 각각 25개의 원소로 이루어져 있기 때문에, 고역통과 필터에 비해 상당히 복잡한 연산과정을 필요로 한다. 그림 2에 점선으로 나타난 것이 25개의 원소별 연산부이다. 기준입력벡터와 가중치벡터는 25개의 별도의 레지스터 형태로 되어있으며, 수렴계수와 기준입력벡터의 곱셈은 고정된 값이므로 곱한 결과를 레지스터에 저장하여 별도의 곱셈이 필요하지 않도록 하였다.

최적의 성능을 위해 각 기준입력에 대해 상수 곱셈기를 사용하여 하나의 사이클에 25번의 곱셈이 이루어질 수 있도록 설계하였다. 상수 곱셈기의 경우 일반적인 곱셈기보다 사용하는 LE의 양이 1/8정도로 적으므로, 각 기준입력에 대해 각각의 상수곱셈기를 설계하여 전체적인 곱셈이 한번에 이루어질 수 있도록 하였다. 각각의 곱셈기에서 출력되는 결과는 덧셈기를 통해 전체적인 덧셈연산을 수행할 수 있도록 하였고, 각 덧셈기의 입출력이 고정적인 연결을 유지하므로 연산결과를 빠르게 출력할 수 있었으며, 부가적인 버퍼를 필요로 하지 않도록 하여 추가적인 LE의 소모도 줄일 수 있도록 설계하였다. 25번의 쉬프트 연산은 가중치 벡터가 변경된 후, 레지스터에 저장되는 과정에서 버퍼를 하나 두어, 가중치 변경과정과 동시에

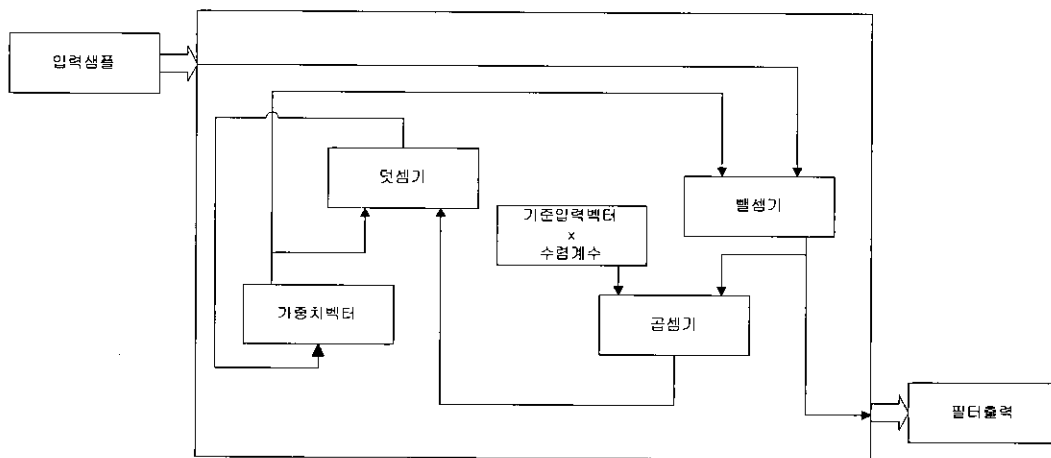


그림 1. FPGA로 구현된 고역통과 필터의 내부구조
Fig. 1. Internal structure of highpass filter implemented by FPGA

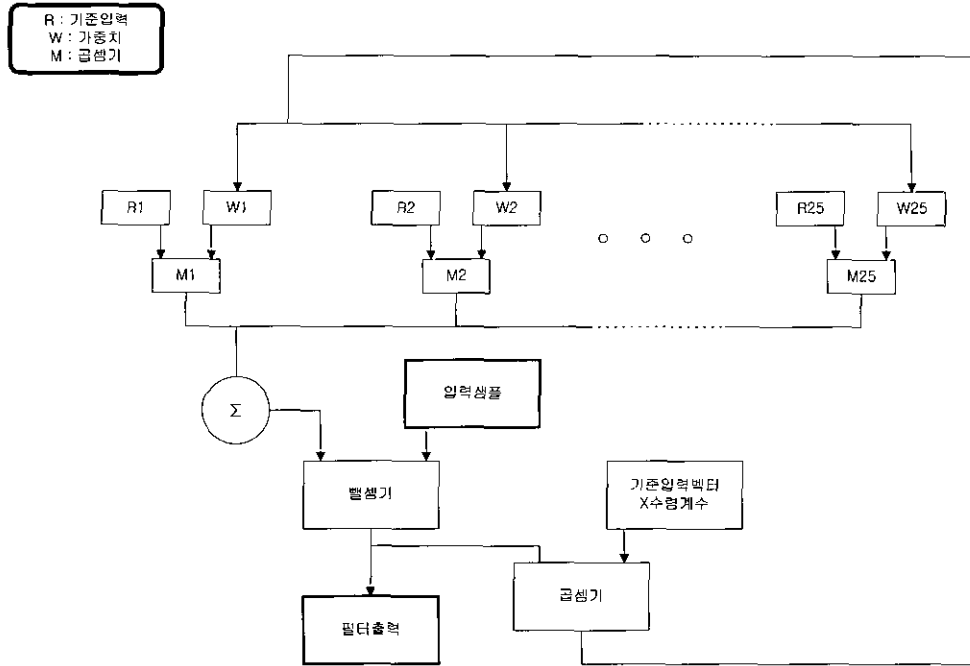


그림 2. FPGA로 구현된 노치필터의 내부구조
Fig. 2. Internal structure of notch filter implemented by FPGA

쉬프트 연산이 이루어질 수 있도록 설계하였다.
이렇게 계산된 기준입력에 대한 FIR 필터의 출력은 샘플링

된 데이터와 차이를 구해 적응필터의 출력으로 나타나게 되는데, 이 과정에서도 적응필터의 출력을 계산하기 위한 전용 16

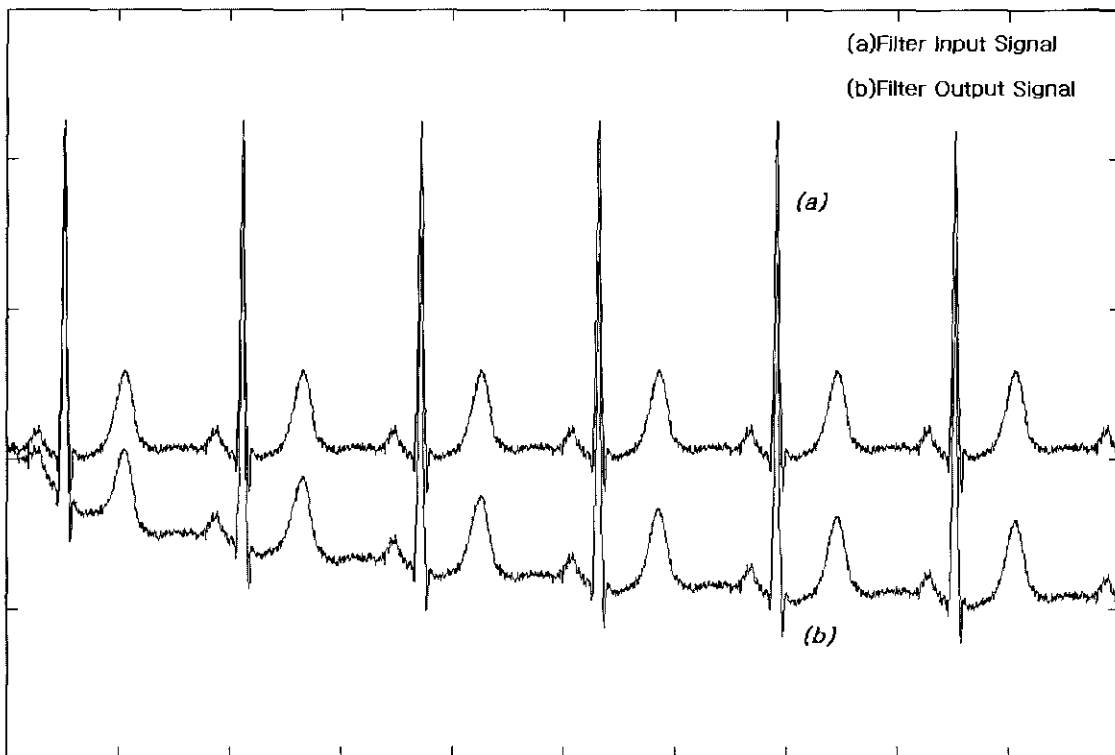


그림 3. FPGA에 구현된 고역통과 필터의 결과파형
Fig. 3. Output waveform of highpass filter implemented by FPGA

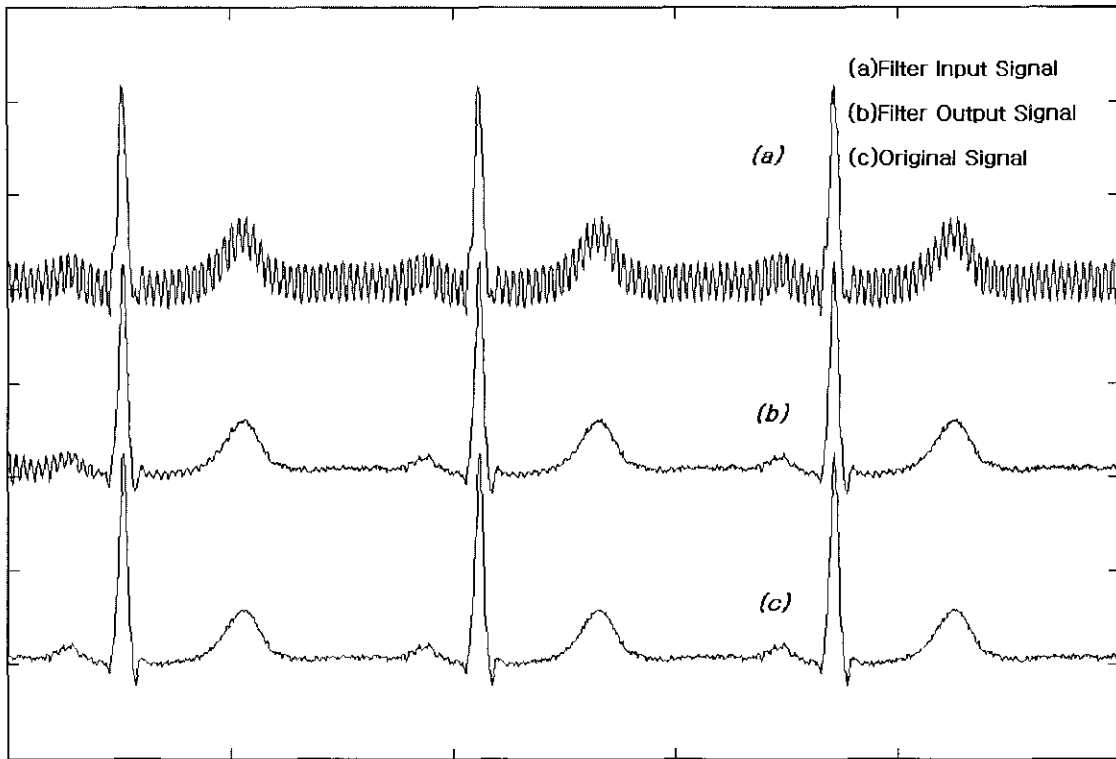


그림 4. FPGA에 구현된 노치필터의 결과파형

Fig. 4. Output waveform of notch filter implemented by FPGA

비트 샘플링을 두어 FIR 필터의 출력이 계산되면 바로 적응필터의 출력을 연산할 수 있는 구조로 설계하였다. 이러한 접근 방법은 하나의 샘플링을 더 설계함으로써 생기는 LE의 손실보다, 설계된 샘플링을 사용하기 위해 필요하게 되는 멀티플렉서와 제어로직의 복잡성으로 인한 LE의 손실이 더 크다는 결과를 고려한 것으로, 최적의 성능을 나타낼 수 있도록 하기 위해서는 필수적인 구조라고 생각된다.

기준입력별 상수곱셈기를 사용함으로써 전체적인 연산과정의 동기화에 대한 제어가 간편해졌고, 따라서 제어로직에 필요한 LE의 수를 최소화할 수 있었다. 최종적인 시스템은 FLEX 10K 계열의 EPF10K100GC503-3 디바이스를 이용하였으며, 이 디바이스 갖고 있는 총 LE의 85%인 4253개의 LE를 이용하여 설계하였다.

실험 및 결과고찰

1. 시뮬레이션 방법 및 결과

본 논문에서 설계한 두 가지 필터의 동작을 테스트하기 위해 CSE 데이터 베이스의 심전도 신호에 잡음이 섞인 데이터를 생성한 후, MATLAB을 이용하여 설계한 필터의 출력력을 계산하였다. 동일한 테스트 데이터를 MAX-PLUS II 타이밍 시뮬레이션 도구의 입력데이터로 사용하여 시뮬레이션을 수행

하였다. Altera사에서 제공하는 설계 소프트웨어인 MAX-PLUS II는 실제 디바이스에서의 모든 연결 지연(routing delay)을 고려한 타이밍 시뮬레이션을 수행하므로, 실제 하드웨어로 구현되어 동작하는 결과와 동일한 시뮬레이션 결과를 출력한다. 따라서 시뮬레이션 결과를 MATLAB의 계산결과와 비교하여 필터의 동작을 평가하였고 그 결과를 그림 3, 4에 나타내었다.

2. FPGA를 이용한 필터의 성능평가

2.1 고역통과 필터

설계한 고역통과 적응필터는 필터링 연산을 위해 각 샘플마다 한 번의 곱셈과 두 번의 덧셈, 혹은 뺄셈연산을 필요로 한다. FPGA에 구현된 필터가 동일한 연산을 수행하기 위해서는 50MHz 클럭으로 1번의 사이클에 해당하는 20ns의 시간을 소요하게 된다. 같은 연산을 Analog Devices사의 신호처리 프로세서인 ADSP21061로 수행하는 경우 13개의 명령사이클을 필요로 하며, ADSP21061은 최대 50MHz 클럭을 사용하므로 260ns의 시간을 소요하게 된다. 비교 결과를 그림 5에 나타내었다.

그림 5에 나타난 바와 같이 FPGA에 구현된 필터가 신호처리 프로세서보다 10배 이상 우수한 성능을 나타내었다. 이러한 결과는 순차적인 프로그램에 의해 연산을 수행하는 신호처리 프로세서의 단점, FPGA의 구조적 특성인 병렬처리를 통해

극복할 수 있었기 때문이다. 즉, 신호처리 프로세서가 범용적인 연산에 적합한 구조를 가지고 있고, 따라서 하나의 연산을 여러 단계에 걸쳐 수행해야 하는 반면, FPGA는 수행해야 할 연산만을 위한 조합논리회로 블록을 병렬로 처리하여, 하나의 사이클에 모든 연산동작을 수행할 수 있다. 이것은 근본적으로 시스템 클럭에 의한 신호처리 프로세서의 내부동작은 고정되어 있는 반면, FPGA는 설계자가 시스템 클럭의 positive going transition(PGT)이나 negative going transition(NGT)을 순차논리회로의 적절한 동작시점에 반영할 수 있고, 결과적으로 최적의 파이프라인 구조로 배열된 각 조합논리회로 블록을 단계별 순차논리회로에 의해 효과적으로 제어할 수 있었기 때문에 얻어진 결과이다. FPGA의 이러한 장점은 기존의 ASSP나 ASIC이 갖고 있지 못하는 융통성 측면도 많이 보완된 것이라 볼 수 있다.

2.2 노치필터

설계한 노치필터는 필터링 연산을 위해 각 샘플마다 50번의 곱셈과 51번의 덧셈, 혹은 뺄셈연산, 25번의 쉬프트 연산을 필요로 한다. 설계된 필터가 이러한 연산을 수행하기 위해서는 20MHz 클럭으로 8번의 사이클에 해당하는 0.4 μ s의 시간을 소요하게 된다. 동일한 연산을 ADSP21061로 수행하는 경우 66개의 명령사이클을 필요로 하며, ADSP21061은 최대 50MHz 클럭을 사용하므로 1.32 μ s의 시간을 소요하게 된다. 비교 결과를 그림 6에 나타내었다.

노치필터의 경우 그림 6에 나타난 바와 같이 FPGA에 구현된 필터가 신호처리 프로세서보다 3배이상 우수한 성능을 보였다. 고역통과 필터에 비해 성능의 차이가 작아진 이유는 첫

째, 노치필터의 구조적인 복잡성으로 인해 높은 시스템 클럭을 사용할 수 없었기 때문이다. 이는 고역통과 필터에 비해 설계된 회로의 크기가 크고, 사용된 LE의 연결이 복잡하여 2배이상 느린 클럭을 사용해야만 필터의 안정적인 동작을 보장할 수 있었기 때문이다. 둘째, 고역통과 필터와 노치필터에서 필요로 하는 연산과정의 차이 때문이다. 고역통과필터의 경우, 노치필터에 비해 반복적인 곱셈과 덧셈을 필요로 하지 않았고, 결과적으로 신호처리 프로세서는 구조적인 장점을 극대화시키지 못했다. 반면 노치필터는 반복적인 곱셈과 덧셈연산이 고역통과필터에 비해 25배나 많았기 때문에, 신호처리 프로세서는 구조적 장점을 극대화시킬 수 있었고, 결과적으로 고역통과필터의 경우보다 성능의 차이가 줄어들게 된 것이다.

FPGA를 이용하여 각 연산에 적합한 회로를 설계하였음에도, 이처럼 성능의 차이가 줄어들게 된 것은 FPGA를 이용한 연산회로의 설계가 어느정도의 한계를 가지고 있기 때문이다. 즉 FPGA는 LE와 같은 기본적인 회로설계단위의 반복적인 배열형태를 가지고 있기 때문에, 아무리 최적의 설계를 하더라도, 기존의 신호처리 프로세서가 가지고 있는 산술연산 회로처럼 빠른 클럭에 의해 동작되는 범용적인 연산회로를 설계하기 어렵다. 이러한 맥락에서 FPGA를 이용한 필터의 설계는 범용적인 연산회로를 각 응용에 맞게 특성화하는 과정이 필요하며, 이 과정에 대한 연구가 지속될 필요가 있다고 생각된다.

결론

본 논문에서는 많은 생체신호 처리분야에 이용되는 전처리 필터를 Altera사의 FLEX계열 FPGA에 구현하여, 샘플링과

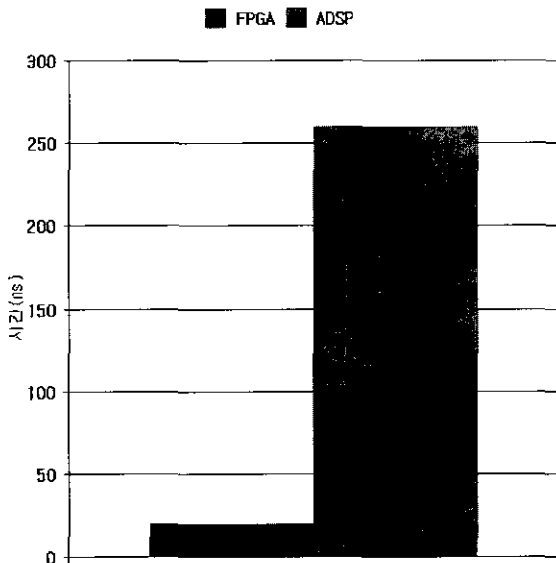


그림 5. FPGA와 신호처리 프로세서로 구현한 고역통과 필터의 성능비교
Fig. 5. Comparison of the performance for highpass filter implemented by FPGA and DSP processor

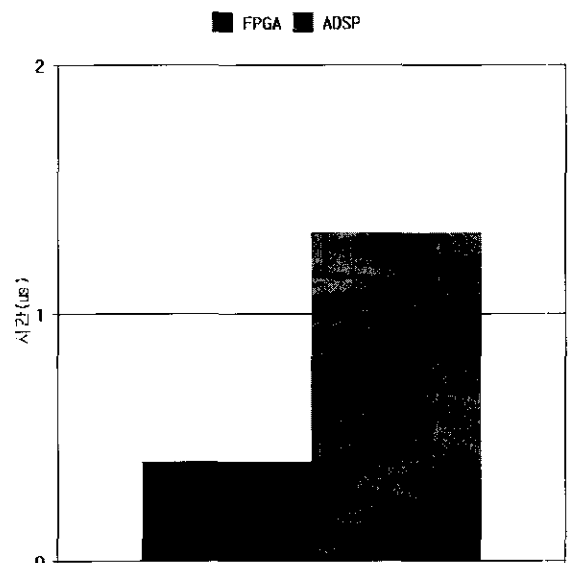


그림 6. FPGA와 신호처리 프로세서로 구현한 노치필터의 성능비교
Fig. 6. Comparison of the performance for notch filter implemented by FPGA and DSP processor

동시에 신호의 전처리를 할 수 있는 시스템을 설계하였다. 테스트 시스템으로 진단용 심전계의 전처리 필터 중에서 DC 오프셋 및 기저선 변동을 제거하기 위한 고역통과 필터와, 전원 잡음을 제거하기 위한 노치필터를 설계하였다. 설계한 시스템은 신호처리 프로세서와의 객관적인 성능평가를 위해 하나의 샘플에 대해 각 디바이스가 소비하는 연산시간을 비교평가 하였으며 얻은 결론은 다음과 같다.

1. 설정한 테스트 시스템의 경우 신호의 전처리를 FPGA를 통해 구현함으로써 전체 시스템의 연산량에 대한 부하를 최소화할 수 있었다.
2. 신호처리 프로세서로 구현할 수 없는 고속의 응용분야에 적절한 ASSP나 ASIC이 없다 하더라도 FPGA를 이용하면 적절한 성능의 시스템 설계가 가능함을 알 수 있었다.
3. 최적화된 FPGA 설계는 기존의 신호처리 프로세서를 대체하거나, 신호처리 코프로세서로서 사용되어 시스템의 성능을 극대화시킬 수 있다는 것을 확인할 수 있었다.

참 고 문 헌

1. Altera Corporation, "FLEX Devices as Alternatives to ASSPs & ASICs", Technical brief 3, 1996
2. Altera Corporation, "Improving Fixed-Point DSP Processor System Performance with PLDs as a DSP Coprocessor", Conference Paper, 1997
3. Altera Corporation, "Digital Signal Processing in FLEX Devices", Product Information Bulletin 23, 1996
4. E. C. Ifeachor, B. W. Jervis, "Digital signal processing - a practical approach", Addison Wesley, 1993
5. W. J. Tompkins, "Biomedical digital signal processing", Prentice-Hall, 1995
6. K. L. Park, K. J. Lee, H. R. Yoon, "Application of a wave let adaptive filter to minimise distortion of the ST-segment", Medical & Biological Engineering & Computing, pp.581-586, 1998
7. R. Jane, P. Laguna, N. V. Thakor, P. Caminal, "Adaptive baseline wander removal in the ECG : Comparative analysis with cubic spline technique", Computers in cardiology, pp.143-146, 1992
8. Jason Cong, Songjie Xu, "Synthesis challenges for next-generation high-performance and high-density PLDs", Design automation conference, proceedings, IEEE. pp.157-162, 2000
9. Langhammer, M., "DSP Implementation in programmable logic", ASIC Conference and Exhibit, Proceedings, IEEE. pp.211-217, 1996
10. Langhammer, M., "Pipelined adaptive filters", ASIC Conference and Exhibit, Proceedings, IEEE. pp.242-245, 1997
11. Mohammed Ferdjallah, Ronald E Barr, "Adaptive digital notch filter design on the unit circle for the removal of powerline noise from biomedical signals", IEEE Trans. Biomed. Eng, Vol. 41, No. 6, pp.529-536, 1994
12. Thakor, N. V, Zhu, Y., "Application of adaptive filtering to ECG analysis : noise cancelation and arrhythmia detection", IEEE Trans. Biomed. Eng, Vol. 38, No. 8, pp.785-794, 1991
13. K. C. Chang, "Digital design and modeling with VHDL and synthesis", IEEE Computer Society Press, 1997
14. 이승호, 이경은, 임반직, "Altera MAX+PLUS II를 사용한 디지털 시스템 설계", 북두출판사, 1999