2014년 8월 전자공학회 논문지 제 51권 제 8 호
Journal of The Institute of Electronics and Information Engineers Vol. 51, NO. 8, August 2014 http://dx.doi.org/10.5573/ieie.2014.51.8.148

논문 2014-51-8-16

Zynq SoC를 이용한 초음파 신호처리 시스템 HW/SW co-design

(HW/SW Co-design For an Ultrasonic Signal Processing System Using Zynq SoC)

임 병 규*, 강 문 호*

(Byung gyu Lim and Moon ho Kang©)

요 약

본 연구에서는 Xilinx의 Zynq SoC(system on chip)를 이용하여 초음파신호의 포락선을 검출하기 위한 신호처리 시스템을 설계하였다. 설계 툴로 Vivado IDE(integrated design environment)를 이용하여, 초음파 신호처리를 위한 전체 과정을 계층적 블록의 형태로 설계하였다. 제안된 시스템은 Zynq-7010의 내장 ADC, FIR(finite impulse response) 밴드패스 필터, 절대값 계산모듈, FIR 로우패스 필터 및 Kalman 필터 등으로 구성되며, 최종 단으로서 FIR 로우패스 필터를 사용하는 HW design 방식과 Kalman 필터를 사용하는 HW/SW co-design 방식에 대해 성능과 유효성을 비교하였다. 비교결과, 포락선 검출 성능에 있어서는 두 방식이 서로 유사한 특성을 갖지만, 시스템 개발에 소요되는 시간 측면에서는 HW/SW co-design 방식이 HW design 방식에 비해 훨씬 더 효율적임이 확인되었다.

Abstract

In this research a signal processing system is designed for detecting the ultrasonic signal envelope using Xilinx's Zynq SoC(system on chip). As a design tool, Vivado IDE(integrated design environment) is used to hierarchically design the whole signal processing system. The proposed system consists of a Zynq-internal ADC, an FIR(finite impulse response) BPF(band pass filter), an absolute value calculator, an FIR LPF(lpw pass filter), and the Kalman filter. Under this configuration, two design schemes, HW design scheme with LPF as a final stage and HW/SW co-design scheme with a Kalman filter as a final stage, are compared in terms of the performance and efficiency. As a result, envelope detecting performances of the two schemes are proved to be almost same, but the HW/SW co-design is verified to be much more efficient than the HW design considering the much smaller time consumption during system design.

Keywords : Ultrasonic signal, Zynq SoC, Vivado, Kalman filter, HW/SW co-design

Ⅰ. 서 론

디지털 신호처리 알고리즘의 발전에 따라 더욱 고성능의 고속 연산이 가능한 장치가 요구된다. DSP를 사용하는 종래의 알고리즘 구현방식은 이러한 고성능 신호처리 요구조건을 충족시키지 못하여 최근에 FPGA (Field Programmable Gate Array)를 사용하는 사례가 증가하고 있다. 이 경우 고속처리 성능뿐만이 아니라 필드의 요구에 따라 FPGA 패브릭을 재구성하여 유연하게 적용할 수 있는 장점이 있지만, 하드웨어 기술 언어인 HDL(Hardware Description Language)를 사용해야하기 때문에 프로그래밍이 쉽지 않다. 한편, FPGA와 DSP 또는 마이크로프로세서를 하나의 칩내에 통합하여

* 정회원, 선문대학교 정보통신공학과
(Department of Information and Communication Engineering, Sunmoon University)
© Corresponding Author(E-mail: mhkang@sunmoon.ac.kr)
접수일자: 2014년06월03일, 수정일자: 2014년07월09일
수정완료: 2014년07월29일

HW/SW co-design이 가능한 장치들이 개발되고 있는데, 하드웨어와 밀접한 부분은 FPGA에 구현하고 프로세서에는 종래의 c-프로그래밍에 의한 소프트웨어 어플리케이션을 구현하여 최적의 성능을 얻을 수 있다.

이들 중에서 Zynq-7000 SoC 시리즈는[1] Xilinx의 7 Series FPGA 패브릭과 듀얼코어 Cortex-9 ARM 프로세서를 통합한 것으로서 고성능 디지털 신호처리가 요구되는 분야에서 다양하게 적용되고 있다. 논문 [2]와 [3]은 각각 실시간 시스템과 내장 시스템에 Zynq SoC (system on chip)를 적용하여 얻어지는 장점들을 보이고, [4]는 도로상의 신호판 인식에, [5]는 크루즈 제어를 위한 프로토타이핑 장치에 Zynq를 적용한 결과를 보인다. [6], [7]은 각각 초음파 이미지 신호처리와 모션-캡쳐링 장치에 Zynq를 적용한 결과를 보이며, [8]은 이더넷 네트워크의 노드설계에 적용한 결과를 보인다.

본 연구에서는 모바일 장치의 이동 궤적을 추종하기 위한 초음파신호 전처리 과정으로서 초음파신호의 포락선[9]을 검출하기 위한 신호처리 시스템을 Zynq를 이용하여 제작하였다. 모바일 장치에 부착된 초음파 발신모듈로부터 발생된 초음파 신호가 초음파 수신모듈에 인가되면, 모듈에 연결된 Zynq-7010의 내장 12비트 ADC에 의해 일차적으로 샘플링이 되고, FIR(finite impulse response) BPF(band pass filter)와 절대값 계산기를 통과한다. 끝으로 FIR LPF(low pass filter) 또는 Kalman 필터[10]를 통과하면, 이 두 경우에 대한 초음파 포락선이 최종 검출된다. 여기서 첫 번째 방식은 ADC부터 FIR LPF까지의 모든 구성요소가 모두 Zynq의 FPGA 측에 설계되는 HW design에 해당하고, 두 번째 방식은 FIR LPF를 대체하여 Kalman 필터에 의해 포락선을 검출하는 것으로, Zynq의 ARM측에 Kalman 필터를 위한 c-코드가 탑재되어 HW/SW co-design에 해당된다. 본 연구에서는 이 두 가지 설계방식에 대해 성능과 유효성을 비교하여 포락선 검출 성능에 있어서는 두 방식이 서로 유사한 특성을 갖지만, 시스템 개발에 소요되는 시간 측면에서는 HW/SW co-design이 훨씬 더 효율적임을 보인다.

시스템 설계 툴로서 Xilinx의 Vivado IDE(Integrated Design Environment)[11]를 이용하여 IP(Intellectual property)[12] 들을 기반으로 한 계층적 블록의 형태로 전체 시스템을 설계한다. 이 경우 시스템의 구조 변경이 간편하여 복잡한 시스템 설계에 매우 유용하다. 시스템 각 블록의 출력을 실시간 확인하기 위하여 로직분석기용 IP인 ILA (Integrated Logic Analyzer)를 Zynq 내에 포함시켜 별도의 모니터링 장치 없이 모든 신호처리와 분석과정이 Zynq 만으로 이루어지도록 하였다.[13]

Ⅱ. HW/SW co-design 시스템 구성

그림 1은 초음파 송수신장치 전체 시스템에 대한 블록도를 보인다. 초음파 발신기는 이동체에 부착되어 40kHz의 초음파 신호를 발생하고, 초음파 수신기에서 수신되어 차동증폭기를 통해 증폭된 후, Zynq 7010의 내장 12비트 ADC에서 샘플링 된다. ADC의 최대 샘플링 주파수는 1MHz이고 최대 변환전압은 1V이다. 샘플링된 신호는 첫째 단인 40kHz의 중심주파수를 가지는 FIR BPF에 인가되어 dc-오프셋 값이 제거되고, 둘째 단인 ABS에서 정류된 후, 셋째 단인 70kHz의 차단 주파수를 갖는 FIR LPF에서 포락선이 검출된다. 이 경우는 ADC로부터 FIR BPF에 이르는 모든 블록들이

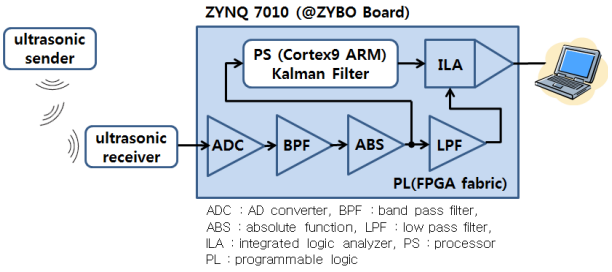

그림 1. Zynq SoC를 이용한 초음파 신호처리 시스템
Fig. 1. Ultrasonic signal processing system using Zynq SoC.

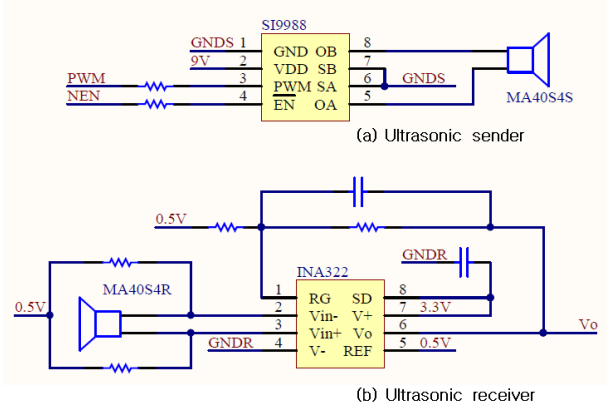

그림 2. 초음파 발신/수신모듈 인터페이스 회로
Fig. 2. Ultrasonic sender/receiver module interface circuits.

Zynq의 FPGA 패브릭영역에 설계되기 때문에 HW design 이라 할 수 있다. 한편, 1차원 Kalman 필터를 사용하여 FIR LPF를 대체할 수 있도록 하였다. 이 경우는 ADC 로부터 ABS 모듈까지는 FPGA에 설계되고 Kalman 필터는 c-코드로 ARM 내에 프로그래밍되기 때문에 HW/SW co-design이 된다.

그림 1의 모든 블록들은 Xilinx의 Vivado(2014.1)을 이용하여 설계된다. 각 블록의 출력 데이터는 로직분석기용 IP인 ILA에 의해 실시간 수집되어 최종적으로 PC로 전송되어 PC상에 확인된다. 각 블록들에 대한 내부 구조와 기능에 대한 자세한 설명은 Ⅲ장에서 다루었다.

그림 2는 초음파 송수신기에서 사용된 초음파 발신 모듈과 수신모듈의 인터페이스 회로를 보이고, 그림 3은 초음파 발신기와 수신기 및 신호처리 보드의 실제 모습을 보인다.

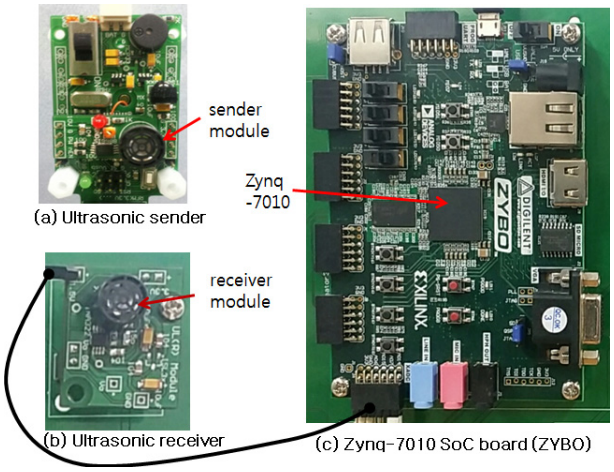

그림 3. 초음파 송수신 장치 PCB 보드
Fig. 3. Ultrasonic sender/receiver device PCB boards.

Ⅲ. HW/SW co-design 시스템 설계

그림 4는 Xilinx Vivado IDE 상에서 설계한 초음파 신호처리 시스템의 스키매틱을 보인다. 여기서, 가장 앞단의 XAdc_SysMon 블록은 Zynq가 내장한 12비트 ADC와 인터페이스하여 초음파 수신신호를 샘플링하기 위한 것이고, FIR_BPF 블록은 FIR 대역필터를 구현한 것이다. ABSfnc 블록은 절대값을 계산하기 위한 것이고, FIR_LPF 블록은 FIR 저역필터를 구현한 블록이다. Kalman_Filter 블록은 Kalman 필터링을 수행하는 블록이다. Drive_LD0 블록은 시스템이 동작중임을 확인하기 위해 주기적으로 LED를 구동하기 위한 것이다. 그림 4는 시스템이 요구하는 각 기능들을 수행하기 위해 IP들을 계층적으로 상호 연결되어 구성한 것으로, 다음에 각 블록들의 내부설계와 기능에 대해서 설명한다. 표 1은 시스템 각 구성부의 주요 설계사양들을 보인다.

표 1. 시스템 설계 사양
Table 1. System design specification.

ADC (@Z-7010)	sampling	ext, analog input channel		
	333[kHz]	14		
FIR filters	type	BW/cutoff freq.	sampling	order
FIR BPF FIR LPF	hamming window	35~45[kHz](BW) 70[kHz](Fc)	1[MHz]	100
Kalman Filter	order	R	Q	
	1	900, 2500, 10000	1	
Ultrasonic modules sender(MA40S4S) receiver(MA40S4R)	nominal freq. 40[kHz]		input volt. 18(square)[Vp-p] -	

R : measurement noise cov. Q : process noise cov.

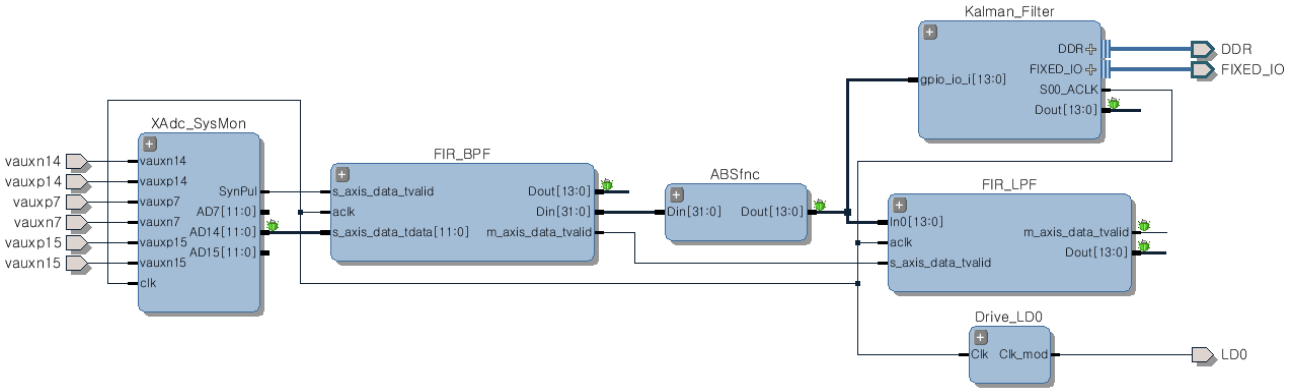

그림 4. 초음파 신호처리 시스템 Vivado 스키매틱
Fig. 4. Vivado schematic of the ultrasonic signal processing system.

1. ADC 인터페이스 설계

XAdc_SysMon 블록의 내부 구조를 보이면 그림 5와 같다. 코어 IP인 XADC Wizard[14]와 여러 부수적인 IP들이 서로 연결되어 Zynq가 내장한 12비트 ADC와 인터페이스를 수행한다. 여기서 SelectXadcZybo_0는 사용자가 제작한 IP로, 입력채널 7, 14, 15에 대한 AD 변환 결과를 레지스터에 저장하고 이들 중에서 초음파 신호가 입력되는 채널14에 대한 변환결과를 FIR BPF 쪽으로 출력한다.

2. FIR 필터 설계

FIR_BPF 블록과 FIR_LPF 블록은 FIR 디지털 필터를 구현하기 위한 블록으로, 그림 6에 FIR_BPF 블록의 내부 구조를 보인다. 여기에서 FIR Compiler[15]가 코어 IP로, FIR 필터링을 위한 여러 파라미터들이 설정된다. 그림 7은 FIR Compiler IP를 이용하여 101 스텝 밴드패스 필터의 파라미터들을 설정하는 모습을 보인다. 이들 중에서 필터의 계수 벡터 값은 Matlab의 디지털 필터 설계 툴인 fdatool[16]을 이용하여 미리 구하고, FIR

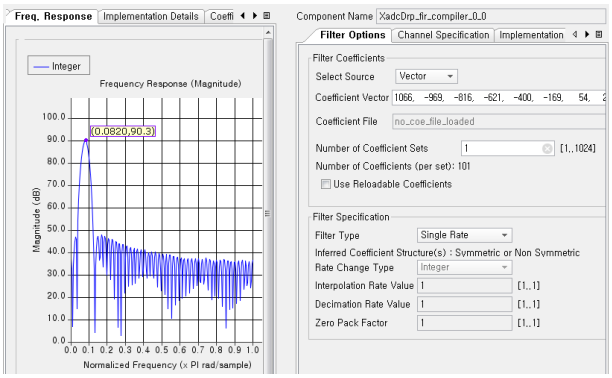

그림 7. FIR Compiler IP의 BPF 설계 윈도우
Fig. 7. BPF design window of FIR Compiler IP.

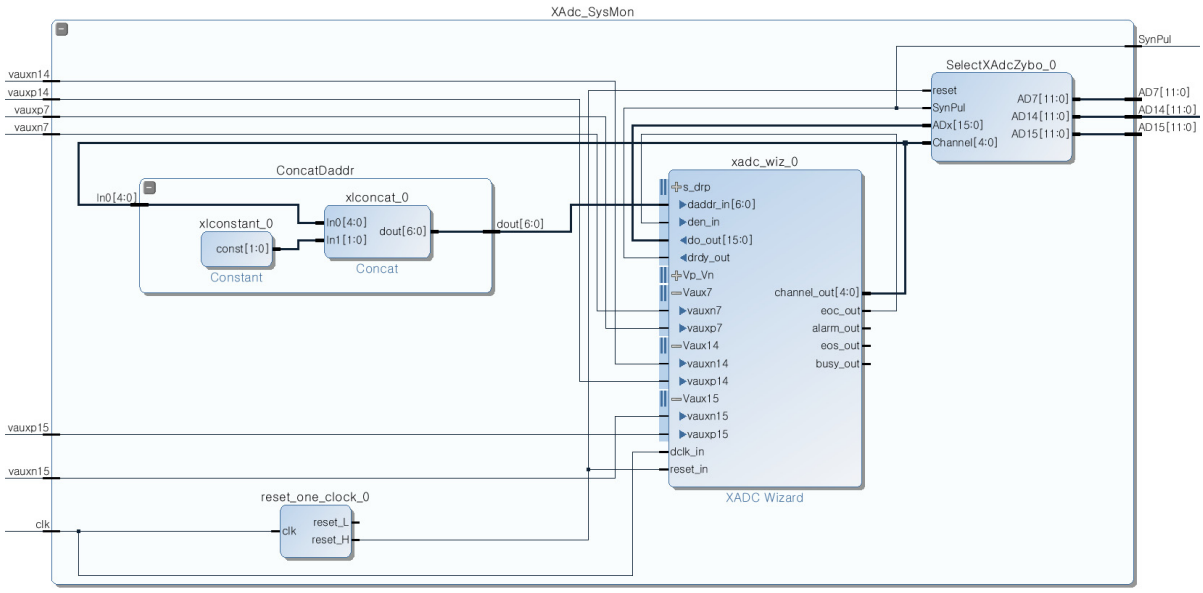

그림 5. XAdc_SysMon 블록의 내부 구조
Fig. 5. Internal configuration of XAdc_SysMon block.

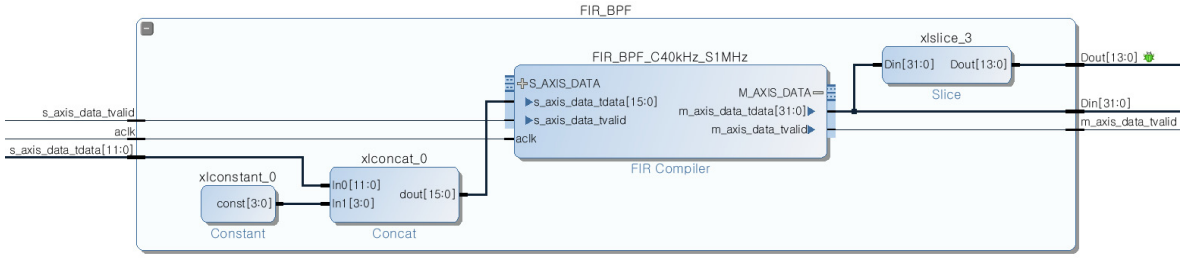

그림 6. FIR_BPF 블록의 내부 구조
Fig. 6. Internal configuration of FIR_BPF block.

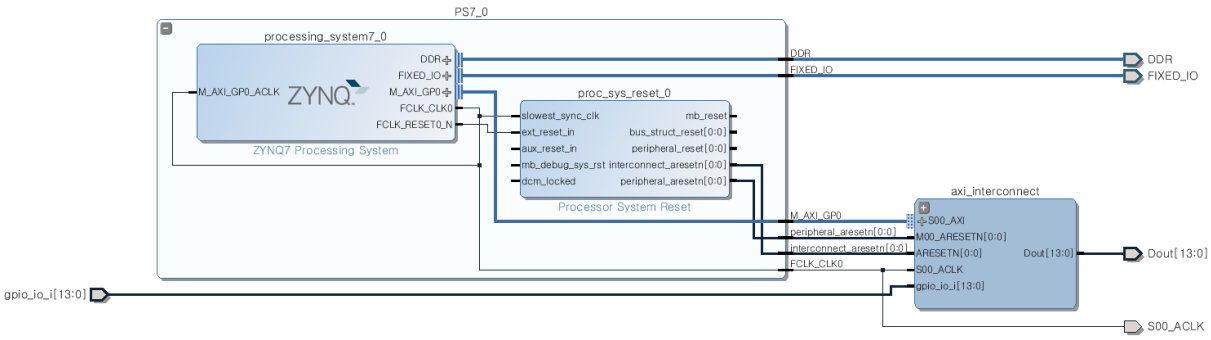

그림 8. Kalman_Filter 블록의 내부 구조
Fig. 8. Internal configuration of Kalman filter block.

Compiler IP 파라미터 설정에 활용하였다. FIR_LPF 블록도 FIR_BPF 블록과 동일한 과정을 통해 설계하였다.

3. Kalman 필터 설계

Zynq의 Cortex9 ARM 프로세서(PS)에 1차 Kalman 필터를 프로그래밍하기 위한 Kalman_Filter 블록의 내부 구조를 보이면 그림 8과 같다. 그림에서 보이는 바와 같이 프로세서는 GPIO(general perpose in/out)를 통해 FPGA 패브릭(PL)에 설계된 ABSfnc 블록으로부터의 출력을 받아 필터링을 수행한 후, 다시 GPIO를 통해 PL측으로 출력한다. ZYNQ7 Processing System 모듈은 PS의 내부 기능들을 설정하기 위한 IP이고, axi_interconnect 모듈은 GPIO를 통해 PS와 PL간 인터페이스를 수행하기 위한 여러 IP들이 결합된 블록이다. Processor System Reset 모듈은 PS 주변 IO들에게 동기 리셑 신호를 제공하기 위한 IP이다. 그림 9는 Kalman_Filter c-코드의 코어 부분을 보인다.

```
while(1)
{
 //R : measurement noise covariance, Q : process noise covariance,
 //K : Kalman gain, P : error covariance

 measure = (float)(XGpio_DiscreteRead(&Gpio, CHANNEL_GPIO)&0x03fff);
                                        //get measurement(state) from GPIO

 K = P/(P+R);                           //compute Kalman gain
 est1 = est1 + K*(measure - est1);      //update state estimate
 P = (1-K)*P + Q;                       //update error covariance

 XGpio_DiscreteWrite(&Gpio, CHANNEL_GPIO, (u32)(est1*16384)); //output estimation to GPIO
}
```

그림 9. Kalman_Filter C-코드 코어 부분
Fig. 9. Core section of Kalman filter c-code.

4. ILA를 이용한 데이터 모니터링

로직분석기용 IP인 Vivado ILA를 사용하여 별도의 모니터링 장치가 필요 없이 시스템 각 요소의 신호처리 결과를 실시간 확인한다. 이를 위해서 그림 4에 보이는 것과 같이 시스템에서 분석이 필요한 지점에 간단히 디버그 마크( )를 부착하면 되는데, ILA는 시스템 운전 도중에 디버그 마크가 놓인 지점에서 발생되는 결과 값들을 FPGA의 블록-램에 저장하여 두고 USB를 통해 외부 PC로 전송하여, 시스템의 동작과정을 실시간으로 분석할 수 있도록 한다.

Ⅳ. 실험 및 고찰

본 연구의 유효성을 보이기 위하여 실험을 수행하고 결과를 분석하였다. 실험에 사용되는 각 요소의 설계 사양을 보이면 Ⅲ장의 표 1과 같다. ADC의 샘플링 주파수는 333[kHz]로 설정하여 40kHz의 초음파신호를 샘플링하고, FIR BPF의 밴드폭은 35~45kHz로, LPF의 차단주파수는 70kHz로 설정하였다. 각 필터의 샘플링 주파수는 1MHz로 설정하였다. Kalman 필터의 R은 900, 2500, 10000으로 하고 Q는 1로 설정하여, R값 변동에 따른 결과를 분석하였다. Kalman 필터 적용에 앞서 그림 9에 보인 수행코드에 대해 수행 주기를 계산하여 수행 사이클이 약 2MHz 임을 확인하였다. 이를 통해 기본파 80kHz인 초음파 정류신호의 필터링에 Kalman 필터를 적용할 수 있음을 알 수 있다.

그림 10은 그림 4의 시스템 블록으로부터 Zynq-7010 SoC 타겥 보드에 다운로드할 실행파일인 bitstream 파일과 c-application 파일이 생성되는 단계를 보이는 흐름도이다. 여기서 HW design에 해당되는 경로는 점선으로 나타내고, HW/SW co-design에 해당되는 경로는 실선으로 나타내었다. FPGA상에 설계된 FIR LPF의 파라미터를 변경하는 경우는 HW design에 해당되어, 합성-구현-비트스트림 생성-비트스트림 다운로드 과정

표 2. 파일생성 작업별 소요시간
Table 2. Time lapse of each file-generation task.

type		time lapse[s]	sum[s]
HW design	Synthesis	186	479
	Implementation	197	
	Bitstream gen.	96	
HW/SW codesign	Rebuild	3	3

Xilinx Vivado 2014.1, Intel Core CPU/2.5GHz

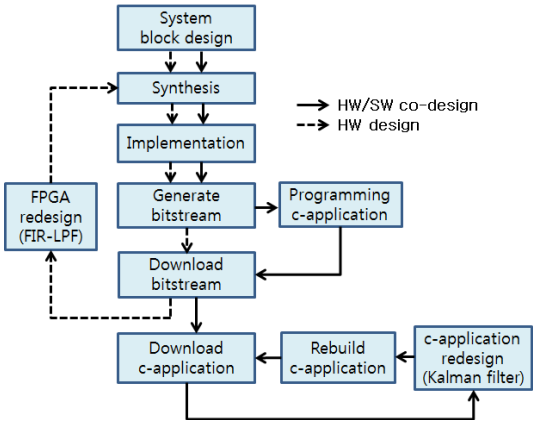

그림 10. 파일생성 작업 흐름도
Fig. 10. Flow chart of file-generation tasks.

이 수행된다. 반면에, c-코드에 의해 Kalman 필터의 파라미터를 변경하는 HW/SW co-design의 경우에는 c-application 파일의 빌드와 다운로드 과정만 수행된다. 표 2는 bitstream 파일과 c-application 파일이 생성되는 데 소요되는 시간을 측정한 것으로, bitstream 파일의 경우 약 8분이 소요되고 c-application 파일의 경우는 수초이내에 완료되어, 실행파일 생성 시간측면에서 HW/SW co-design이 HW design에 비해 훨씬 더 효율적임을 알 수 있다. 참고로, 이 과정에서 사용된 노트북 PC의 CPU 클록은 2.5GHz이다.

그림 11은 시스템 요소의 신호처리 결과 파형을 보인다. ILA IP에 의해 실시간으로 결과 데이터를 수집한 후 PC로 전송한 결과 파형이다. 파형 (a)는 Ⅲ장의 그림 4에서 XAdc_SysMon 블록의 출력 파형으로 초음파 신호를 12비트 ADC로 샘플링한 결과를 보이고, (b)는 FIR_BPF 블록의 출력 파형으로 초음파 신호에 대한 대역필터링 결과를 보인다. (c)는 그림 4의 ABSfnc 블록의 출력 파형으로 초음파 신호가 전파 정류된 결과를 보이고, (d)는 FIR_LPF 블록의 출력으로, 전파 정류된 초음파 신호의 포락선을 검출한 결과를 보인다. (e)는 그림 4의 Kalman_Filter 블록의 출력으로, R과 Q를 각각 900과 1로 설정하여 전파 정류된 초음파 신호를 필

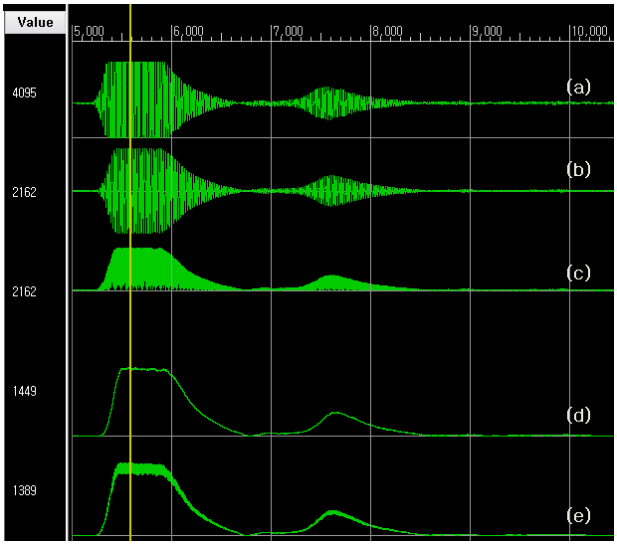

그림 11. 시스템 출력파형 ((a)ADC (b)BPF (c)ABS모듈 (d)LPF (e)Kalman 필터(R=900, Q=1))
Fig. 11. System output waveforms ((a)ADC (b)BPF (c)ABS module (d)LPF (e)Kalman filter(R=900, Q=1)).

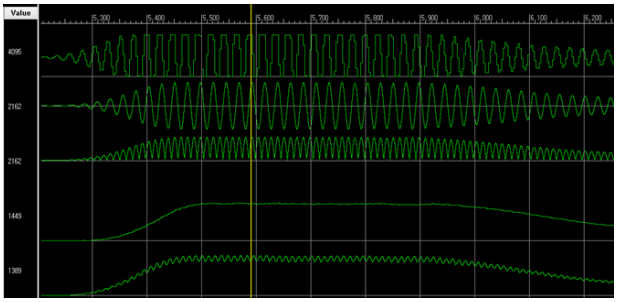

그림 12. 그림 11 확대 파형
Fig. 12. Magnified waveforms of Fig. 11

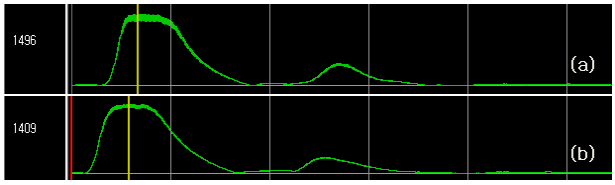

그림 13. Kalman 필터 출력파형
((a)R=2500, Q=1 (b)R=10000, Q=1)
Fig. 13. Kalman filter output waveforms.
((a)R=2500, Q=1 (b)R=10000, Q=1)

터링하여 포락선을 검출한 결과이다. (d)와 (e)에 의해, LPF뿐만이 아니라 Kalman 필터에 의해서도 초음파 포락선을 효과적으로 검출할 수 있음을 알 수 있다. 그림 12는 각 출력파형을 더욱 면밀히 관찰하기 위하여 그림 11을 확대한 것이다. (e)를 보면 Kalman 필터 출력에

리플이 확인된다. 그림 13에서 (a)와 (b)는 각각 900에서 2500과 10000으로 Kalman 필터의 R값을 증가시킨 경우로, R이 증가함에 따라 출력 리플이 감소하지만 과도특성도 같이 저하하는 모습을 보인다. 따라서, R값을 적절히 조절하면 필요한 특성을 얻을 수 있을 것으로 판단된다. 이상의 결과들로부터 HW design과 HW/SW co-design를 비교하면, Kalman 필터를 사용하는 HW/SW co-design이 FIR 필터를 사용하는 HW design에 비해서 필터의 속응성은 좋지만 응답 리플이 상대적으로 커지는 것을 알 수 있다. 한편, 시스템 개발기간 측면에서는 HW/SW co-design이 훨씬 더 효율적임을 확인하였다.

V. 결 론

본 연구에서는 Xilinx의 Zynq-7010 SoC를 이용하여 초음파신호의 포락선을 검출하기 위한 신호처리 시스템을 제작하였다. 설계 툴로 Vivado IDE를 이용하여 초음파 신호처리 전체 과정을 IP들을 기반으로 한 계층적 블록의 형태로 설계하였다. 또한, 시스템 각 요소에서의 신호처리 결과를 실시간 확인하기 위하여 Vivado ILA를 Zynq SoC 내에 포함시켜 별도의 모니터링 장치 없이 전체 신호처리와 분석이 Zynq SoC 만으로 이루어지도록 하였다. 신호처리 시스템은 Zynq-7010의 내장 12비트 ADC, FIR 밴드패스 필터, 절대값 계산모듈, FIR 로우패스 필터 및 Kalman 필터 등으로 구성되며, 최종단으로서 FIR 로우패스 필터를 사용하는 HW design 방식과 Kalman 필터를 사용하는 HW/SW co-design 방식에 대한 성능과 유효성을 비교하였다. 비교결과, 초음파신호 포락선 검출 성능측면에서는 두 방식이 서로 유사한 특성을 갖지만, 시스템 개발기간 측면에서는 HW design에 비해 HW/SW co-design이 훨씬 더 효율적임을 확인 할 수 있었다. 향후에는 본 연구결과를 이용하여 초음파 TOF를 구하고 이동체의 궤적을 추종하는 연구를 수행할 것이다.

REFERENCES

[1] “Zynq-7000 All Programmable SoC Overview,” DS190 (v1.6) Xilinx, December 2, 2013.
[2] Fleming, S. T. and Thomas, D. B., “FPGA based control for real time systems,” 23rd International Conference on Field Programmable Logic and Applications(FPL), pp. 1 - 2, 2013,
[3] Eberli, F., “Next Generation FPGAs and SOCs-How Embedded Systems Can Profit,” IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPRW), pp. 610 - 613, 2013.
[4] Russell, M. and Fischaber, S., “OpenCV based road sign recognition on Zynq,” 11th IEEE International Conference on Industrial Informatics (INDIN), pp. 596 - 601, 2013.
[5] Wehner, P., Ferger, M., Gohringer, D., and Hubner, M., “Rapid prototyping of a portable HW/SW co-design on the virtual zynq platform using SystemC,” IEEE 26th International Conference on SOC(SOCC), pp. 296 - 300, 2013.
[6] Gilliland, S., Govindan, P., Gonnot, T., and Saniie, J., “Performance evaluation of FPGA based embedded ARM processor for ultrasonic imaging,” IEEE International Ultrasonics Symposium (IUS), pp. 519 - 522, 2013.
[7] Bruckner, H. P., Spindeldreier, C., and Blume, H., “Energy-efficient inertial sensor fusion on heterogeneous FPGA-fabric/RISC System on Chip,” Seventh International Conference on Sensing Technology (ICST), pp. 506 - 511, 2013.
[8] Astarloa, A., Lazaro, J., Bidarte, U., Zuloaga, A., and Idirin, M., “System-on-Chip implementation of Reliable Ethernet Networks nodes,” 39th Annual Conference of the IEEE Industrial Electronics Society, IECON, pp. 2329 - 2334, 2013.
[9] J. Y., Cho and M. H., Kang, “Design of a Ultrasonic Oil Level Meter Using a FPGA,” *Journal of The Institute of Electronics Engineers of Korea*, Vol. 49, no. 11, pp. 169-174, November 2012.
[10] Ramsey, F., “Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation,” *IEEE SIGNAL PROCESSING MAGAZINE*, pp. 128-132, SEPTEMBER 2012.
[11] Vivado Design Suite User Guide, Using the Vivado IDE, Xilinx, Dec. 18, 2013.
[12] Vivado Design Suite User Guide, Designing with IP, Xilinx, May 1, 2014.
[13] Vivado Design Suite User Guide, Programming and Debugging, Xilinx, Apr. 2014.
[14] LogiCORE IP XADC Wizard v3.0 Product Guide for Vivado Design Suite, Oct. 2, 2013
[15] LogiCORE IP FIR Compiler v7.1 Product Guide

for Vivado Design Suite, Dec. 18, 2013
[16] Signal Processing Toolbox For Use with MATLAB version 6, the MathWorks, 2013.

저 자 소 개

임 병 규(정회원)
2014년 선문대학교 정보통신공학과 학사 졸업.
2014년 선문대학교 정보통신공학과 석사 재학.
<주관심분야 : 회로 및 센서, 모터 제어>

강 문 호(정회원)
1990년 고려대학교 전기공학과 석사 졸업.
1995년 고려대학교 전기공학과 박사 졸업.
현 재 선문대학교 정보통신공학과 교수.
<주관심분야 : 센서 네트워크, 지능제어>