FPGA 임베디드 프로세서 시스템을 사용한 실시간 SONAR 신호 디스플레이 시스템의 구현

An Implementation of Real-Time SONAR Signal Display System using the FPGA Embedded Processor System

김 동 진*, 김 대 웅*, 박 영 석*
Dong-Jin Kim*, Dae-Woong Kim*, Young-Seak Park*

요 약

선박이나 함정에서 일반적으로 사용되는 SONAR 신호용 CRT 모니터 디스플레이 시스템은 벡터 주사 방식을 사용한다. 그래서 시스템의 처리회로가 복잡하고, 부품 생산이 폐쇄되어 부품 수급이 어렵고 가격이 고가이다. FPGA 기반 임베디드 프로세서 시스템은 회로를 단순화함과 더불어 코어설계를 쉽게 재구성함으로써 각종 응용 적용에 유연하고, 저가격대로 고속 성능을 제공한다.

본 논문은 기존 CRT 시스템의 문제점을 극복하기 위해서 FPGA 임베디드 프로세서 시스템을 사용하여 SONAR 신호 LCD 디스플레이 시스템을 구현하였다. 제안한 접근법은 기존 시스템에 비해 X-Y 편향과 CRT 제어 블록을 FPGA 임베디드 프로세서 시스템으로 대체함으로써 시스템 구성의 단순성과 유연성을 확보할 수 있고, 또한 저가격화를 가능하게 한다. 구현된 시스템은 SONAR 신호를 실시간으로 획득하고 LCD에 디스플레이하는 것이 가능하다.

Abstract

The CRT monitor display system for SONAR signal that are commonly used in ships or naval vessels uses vector scanning method. Therefore the processing circuits of the system is complex. Also because production had been shut down, the supply of parts is difficult as well as high-cost. FPGA-based embedded processor system is flexible to adapting to various applications because it makes simple processing circuits and its core is easily reconfigurable, and provides high speed performance in low-cost.

In this paper, we describe an implementation of SONAR signal LCD display system using the FPGA embedded processor system to overcome some weakness of existing CRT system. By changing X-Y Deflection and CRT control blocks of current system into FPGA embedded processor system, our system provides the simplicity, flexibility and low-cost of system configuration, and also real-time acquisition and display of SONAR signal.

Keywords : Nios II, real-time data acquisition, system interconnect fabric, SONAR

I. 서 론

선박이나 함정에서 일반적으로 사용되는 SONAR 신호는 자료 구조와 지도중첩이 복잡한 벡터 주사 방식으로 화면 전체를 X좌표와 Y좌표에 평행한 선분으로 나눈 격자상의 좌표계를 만들어 두 점 간의 좌표 값을 통해 직선이나 곡선 등의 도형을 표시하는 형태이다. 이러한 복잡한 신호를 구현하는 CRT 모니터 디스플레이 시스템은 회로 구성이 복잡하고, 대부분 생산이 폐쇄되어 부품 수급이 어렵고 가격이 고가이다.

현재 국내외적으로 선박, 군함정 등에 과거에 보급된 많은 SONAR CRT 디스플레이 시스템이 사용되고 있다. 시기적으로 새로운 시스템으로 대체가 요구되나 시스템이 고가이기 때문에 기존 시스템을 활용할 필요가 있다. 그러나 대부분 부품의 생산이 폐쇄되어 보수유지가 어렵기 때문에 기존 시스템을 활용하되 LCD 디스플레이 시스템으로 전환하여 사용할 필요성이 대두되고 있다. 현재 많이 사용되는 데이터 획득 시스템은 데이터를 획득하기 위한 별도의 보드를 사용하거나 또는 마이크로프로세서를 사용하여 데이터를 수집 및 저장하였다. 이러한 시스템의 단점은 매우 비싸고, 확장성이 제한되며, 경우에 따라서는 전자기의 간섭에 약하다. 특히 마이크로프로세서의 경우 CPU의 클록 속도가 느리기 때문에 입력 데이터를 실시간으로 처리하기에는 적합하지 않다. DSP의 경우 데이터 처리 속도는 빠르지만 그만큼 더 높은 비용을 지불해야 하며, 시스템의 성능 향상에도 많은 제약이 따른다[1]. 이에 반해 FPGA 임베디드 프로세서 시스템은 각종 응용에 유연하고 코어설계를 손쉽게 재구성 가능하며, 비용 대비 높은 처리 속도를 제공하기 때문에 실시간 데이터 획득 및 처리에 적합

* 경남대학교
투고 일자 : 2011. 7. 25 수정완료일자 : 2011. 10. 25
게재확정일자 : 2011. 11. 1
* 본 연구의 결과물은 2011학년도 경남대학교 학술진흥연구비 지원에 의하여 이루어졌음.

하다[2]. 본 연구에서는 기존의 SONAR 신호용 CRT 모니터 디스플레이 시스템의 단점인 복잡한 회로구성, 가격의 고가, 생산이 폐쇄되어 부품 수급 및 확보의 어려움을 극복하기 위해 범용성이 있는 FPGA 임베디드 프로세서 시스템을 사용하여 LCD 디스플레이 시스템을 구현한다. 구현 시스템은 기존 시스템의 주요 회로 모듈들을 그대로 활용하면서 시스템의 회로 구성을 단순화할 수 있고, 실시간으로 SONAR 신호의 획득 및 디스플레이가 가능하다. 실시간으로 출력되는 SONAR 신호를 획득하기 위한 SONAR 신호 획득 IP와 획득한 SONAR 신호를 RGB 컬러 신호로 변환하기 위한 그래픽 라이브러리, RGB 컬러 신호를 디스플레이 하기 위한 VGA 제어기(Controller)를 구현하였다.

II. SONAR 신호 디스플레이 시스템

일반적인 선박이나 함정에서 주로 사용되는 소나를 사용한 CRT 모니터 디스플레이 방식은 일반적으로 사용되는 래스터 주사(Raster-scan) 방식이 아니라 벡터 주사(Vector-scan) 방식을 사용한다. 벡터 주사 방식은 지정된 좌표를 직선으로 연결해 가는 선에 의한 표시 방법이다.

그림 1은 음향 탐지 장비인 SONAR를 사용한 기존 CRT 모니터 디스플레이 시스템의 회로 블록도이다. 디스플레이 인터페이스 모듈은 소나로부터 출력되는 아날로그 디스플레이 데이터를 디지털 신호인 X-Y 좌표와 8비트 컨트롤 신호로 변환한다. 변환된 X-Y 좌표 신호는 X-Y Deflection 모듈로 전달되며, 제어 신호는 CRT 제어 모듈로 전달되어 CRT 모니터를 디스플레이 하기 위한 신호로 변환되어 출력된다.

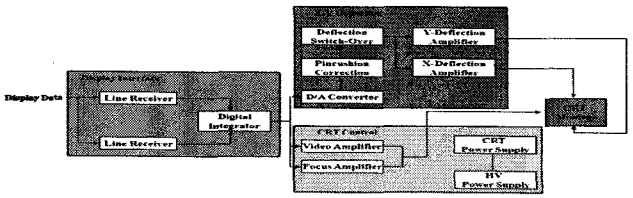

그림 1. SONAR 장비를 사용한 CRT 디스플레이 시스템
Fig. 1. CRT display system using SONAR equipment

그림 2는 기존의 SONAR 장비를 사용한 CRT 디스플레이 시스템의 단점을 보완할 수 있는 FPGA 임베디드 시스템을 사용한 SONAR 신호 디스플레이 시스템의 블록도이다. 기존의 시스템에 비해 X-Y Deflection과 CRT 제어 블록을 FPGA 임베디드 시스템으로 대체 함으로써 시스템을 간단하게 구성할 수 있고, 높은 유연성과 저 가격화가 가능하다.

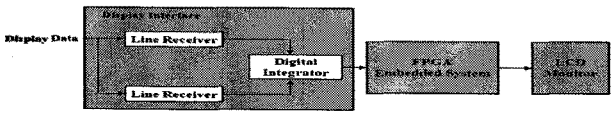

그림 2. FPGA 임베디드 시스템을 사용한 LCD 모니터 디스플레이 시스템
Fig. 2. LCD monitor display system using FPGA embedded system

III. 하드웨어 설계

하드웨어 설계의 주요 도구로는 Quartus II 설계 소프트웨어, Nios II 프로세서와 주변장치를 구성하기 위한 SOPC Builder, Modelsim 시뮬레이션 소프트웨어 그리고 구현회로 검증을 위한 SignalTap II 임베디드 논리 해석기를 이용한다[3-5]. 본 연구에서 사용한 시스템은 Terasic사의 DE2-70 보드[6]이며, SDRAM, SRAM, FLASH, VGA DAC, 확장 포트(Expansion Port) 등의 주변장치를 사용하였다.

3.1 SONAR 신호 획득 IP 구현

그림2의 FPGA 임베디드 시스템을 사용한 SONAR 신호 디스플레이 시스템을 구현하기 위해서는 디스플레이 인터페이스 모듈로부터 출력되는 신호를 실시간으로 획득하기 위한 사용자 IP 설계가 요구된다. 실시간으로 외부 데이터를 획득하기 위해 사용할 수 있는 Altera에서 제공하는 PIO(Parallel Input Output) IP로는 많은 제약사항이 존재한다. 그래서 본 논문에서는 실시간으로 데이터를 획득하기 위한 사용자 IP를 구현하였으며 이러한 사용자 IP는 재사용 가능하며, 라이센스가 없는 장점이 있다.

Nios II 시스템을 위한 IP를 만들기 위해서는 Nios II 시스템이 사용하는 시스템 인터커넥트 패브릭(System interconnect fabric)의 동작 원리와 시스템 인터커넥트 패브릭에 연결 가능한 인터페이스 6가지 타입을 사용해야 한다. 본 논문에서는 Avalon memory-mapped interface[7-8]를 사용하여 데이터 획득 IP를 구현하였다.

그림 3은 시스템 인터커넥트 패브릭의 예제 시스템이다. 프로세서와 DMA 제어기는 Avalon memory-mapped master로써 시스템 인터커넥트 패브릭을 통하여 Avalon memory-mapped slave에 연결된다. 시스템 인터커넥트 패브릭내의 Arbiter의 역할은 두개 이상의 Master가 하나의 Slaver에 연결될 때 중재자의 역할을 담당한다. 시스템 인터커넥트 패브릭은 사용자가 직접 설계하는 것이 아니고 Nios II 프로세서와 주변장치를 구성한 정보를 바탕으로 SOPC Builder가 자동으로 생성함으로써 설계에 대한 어려움을 덜어준다[7-8].

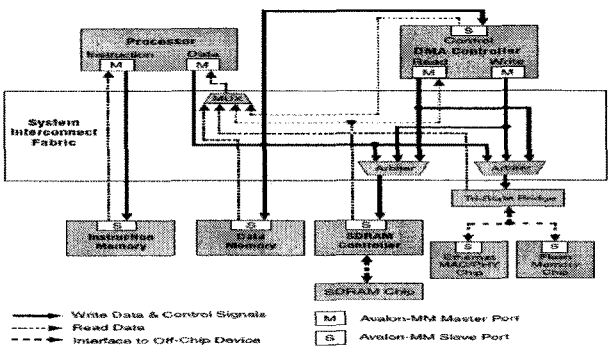

그림 3. 시스템 인터커넥트 패브릭의 예제 시스템[7]
Fig. 3. Example system of system interconnect fabric[7]

표 1은 시스템 인터커넥트 패브릭의 주요 기능을 요약한 것이다.

표 1. 시스템 인터커넥트 패브릭의 기능

Table 1. Function of system interconnect fabric

- Address decoding
- Datapath multiplexing
- Wait state insertion
- Pipelined read transfers
- Arbitration for multimaster systems
- Burst adapters
- Interrupts
- Reset distribution

표 2. Avalon-MM slave의 신호 형태

Table 2. Signal types of avalon-MM slave

Signal Type	Width[bit]	Dir
read	1	in
write	1	in
address	1~32	in
readdata	8,16,32,64,128,256,512,1024	out
writedata	8,16,32,64,128,256,512,1024	in
byteenable	1,2,4,8,16,32,64,128	in
begintransfer	1	in
waitrequest	1	out
arbiterlock	1	in
readdatabalid	1	out
burstcount	1~11	in
readyfordata	1	in
dataavailable	1	out
resetrequest	1	out

SONAR 신호 획득 IP를 설계하기에 앞서 입력되는 데이터 신호의 분석이 요구된다. 실시간으로 획득 할 데이터는 x-y 좌표 그리고 각각의 x-y 좌표를 제어하기 위한 제어 신호이다. 그림 4는 입력되는 데이터 중에서 제어 신호를 분석한 표이다. 제어 신호는 8비트로써 최상위 비트가 Clear 신호이고, F1과 F0는 색상을 판별하며, MZ와 Z H/D는 유효 데이터를 판별하며, Z MSB와 Z1과 Z LSB는 색상 밝기를 판별한다. 제어 신호는 x-y 좌표 데이터보다 2주기 지연되어서 입력이 된다. 따라서 x-y 좌표 데이터를 2주기 지연시키는 로직이 필요하다.

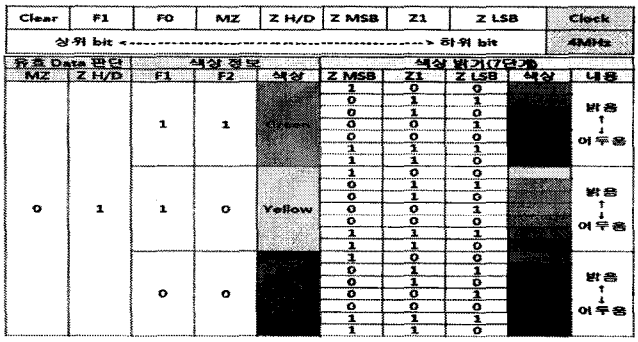

그림 4. 제어 신호 형식

Fig. 4. Control signal format

그림 5는 SONAR 신호 획득 IP의 블록도이다. 입력논리(Inout logic)에서는 클록에 동기화되어 x-y 좌표와 제어 데이터를 받아들이고, x-y 좌표는 지연논리(Delay logic)을 통해서 FIFO에 저장된다. 저장된 x-y 좌표 및 제어 신호는 Avalon memory mapped slave port를 통해서 Nios II 프로세서로 전달된다.

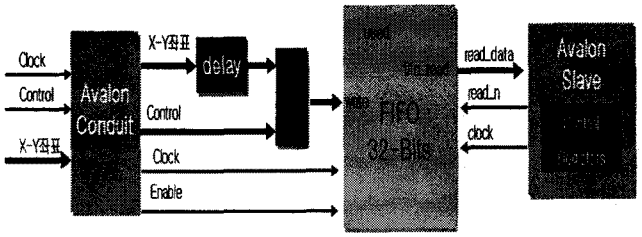

그림 5. SONAR 획득 IP의 블록도

Fig. 5. Block diagram of SONAR signal acquisition IP

그림 6은 Quartus II 설계 소프트웨어로 합성한 데이터 획득 IP에 대한 RTL 회로도이다.

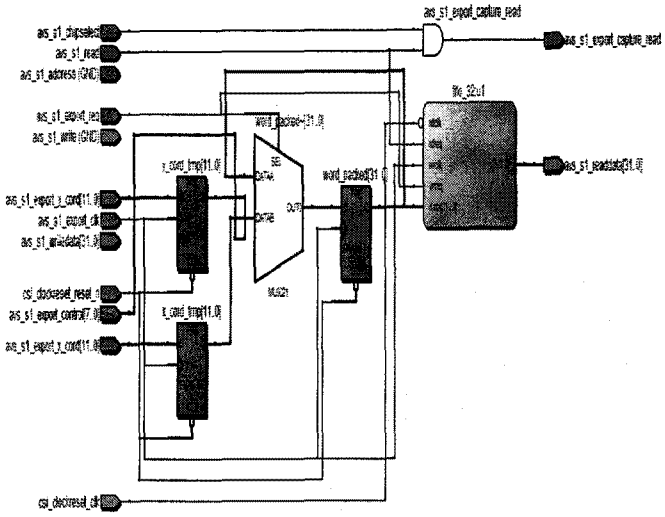

그림 6. 데이터 획득 IP의 RTL 회로도

Fig. 6. RTL circuit of data acquisition

3.2 VGA 제어기 구현

획득한 x-y 좌표 및 제어 신호를 이용하여 모니터에 디스플레이 하기 위해서는 Nios II 프로세서를 위한 VGA 제어기가 요구된다. Nios II 유저들이 만든 공개용 VGA 제어기가 있지만 대부분 지원하는 해상도가 640×480이다. 본 연구에서는 1024×768의 해상도가 요구된다.

그림 7은 본 연구에서 구현한 VGA 제어기의 블록도이다. Avalon slave port를 통해서 Nios II 프로세서로부터 프레임버퍼의 어드레스와 VGA 제어기 구동에 필요한 레지스터 설정 값을 받는다. Avaon read master는 프레임버퍼로부터 영상 정보를 VGA 제어기로 읽어들인 후 FIFO에 저

장을 한다. FIFO에 저장된 영상 정보는 VGA Sequencer 모듈에서 생성한 VGA 제어기의 타이밍 신호에 동기화되어 출력된다.

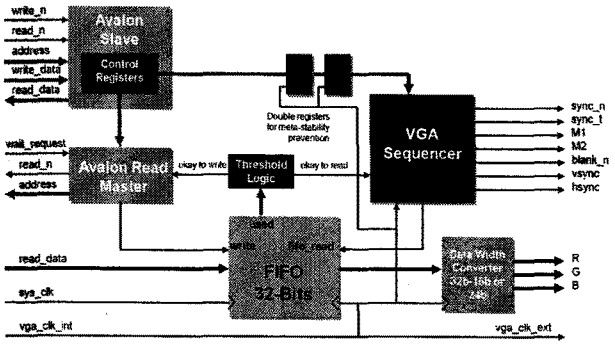

그림 7. VGA 제어기의 블록도

Fig. 7. Block diagram of VGA controller

그림 8은 Quartus II 설계 소프트웨어로 합성한 VGA 제어기의 RTL 회로도이다.

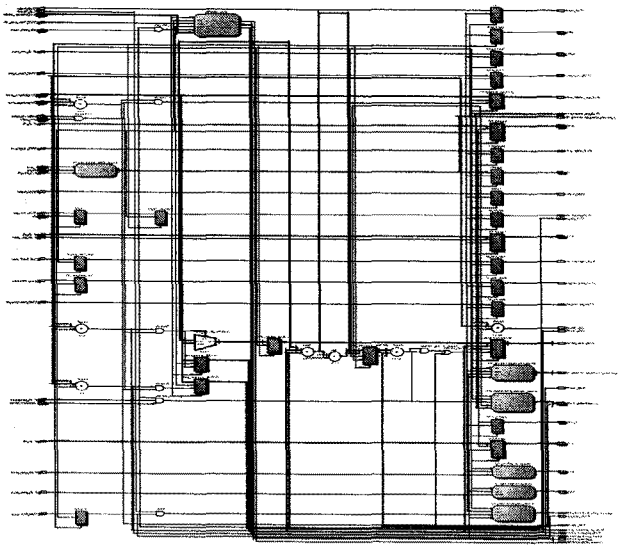

그림 8. VGA 제어기의 RTL 회로도

Fig. 8. RTL circuit of VGA Controller

3.3 Nios II 프로세서 및 주변장치 구성

Nios II 프로세서 시스템을 설계하기 위해서는 목적 시스템을 상술하고, 구성하여 시스템을 생성하는 SOPC Builder 개발도구를 사용한다. 그림 9는 SOPC Builder 설계 소프트웨어로 실시간 데이터 획득 및 디스플레이를 위한 Nios II 시스템을 구성하였다. CPU, Timer, SDRAM, SRAM, CFI Flash, EPCS Serial flash, 데이터 획득 IP(User IP), VGA 제어기(User IP) 등의 IP가 요구된다.

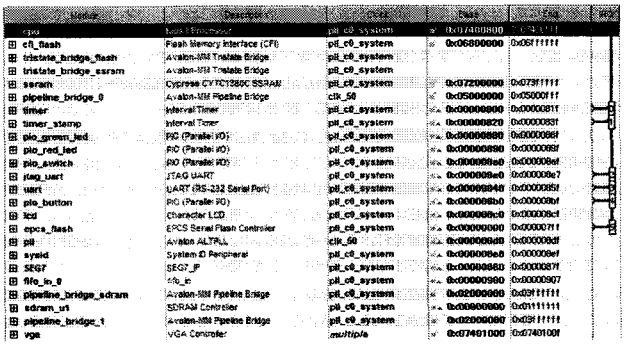

그림 9. SOPC builder를 이용한 시스템 구성

Fig. 9. Using SOPC builder system configuration

SOPC Builder를 이용하여 시스템을 구성하고 Generate를 실행함으로써 구성한 시스템이 자동 생성되고, 이러한 하드웨어 정보들은 .sopcinfo 파일에 저장된다. Quartus II에서 Compilation을 실행하여 .sof 파일을 생성함으로써 하드웨어 설계는 완료된다.

Ⅳ. 소프트웨어 설계

소프트웨어 설계의 주요 도구로는 크게 두 가지로 나눌 수 있다. 하나는 그래픽 기반의 UI를 제공하는 Nios II IDE와 Nios II Software Build Tools(SBT) for Eclipse가 있다. Nios II SBT for Eclipse는 Nios II IDE보다 Build 속도가 빠른 장점이 있다. 다른 하나는 Command Shell에서 실행 가능한 Nios II SBT Command-line이다. Nios II SBT Command-line의 장점은 사용자 스크립트, Tcl 스크립트, bash 스크립트를 사용할 수 있다[9].

4.1 Nios II Hardware Abstraction Layer(HAL)

Nios II HAL은 어플리케이션과 device-level 소프트웨어 사이에 명확하게 구분된 폼으로써 잘 정의된 소프트웨어 계층이다. HAL은 경량이고, 하드웨어를 제어하는 어플리케이션을 위한 인터페이스를 제공하는 디바이스 드라이버이다. HAL API는 ANSI C 표준 라이브러리 사용하고, printf(), fopen(), fwrite() 와 같은 C 라이브러리 함수를 사용한다. 그림 10은 HAL 기반 시스템의 계층구조이다. Nios II 프로세서 시스템 하드위어 계층은 SOPC Builder가 만들고, HAL 계층은 Nios II IDE에서 만들며, 사용자 프로그램 계층은 사용자가 직접 만든다[9].

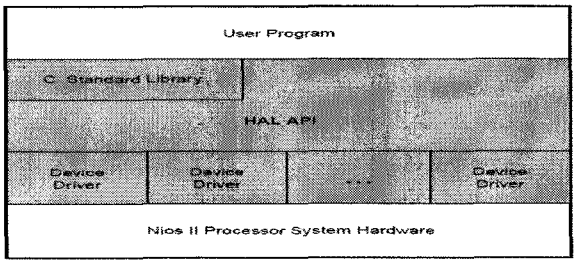

그림 10. HAL 기반 시스템의 계층

Fig. 10. Lavers of a HAL-based System

HAL이 제공하는 서비스는 표 3과 같다.

표 3. HAL 서비스

Table 3. HAL services

▸Integration with the newlib ANSI C standard library
▸Device drivers - Provides access to each device in the system
▸HAL API
▸System initialization
▸Device initialization

4.2 프로그램 설계

그림 11은 어플리케이션을 실행하기 위해서 반드시 거쳐야할 Nios II 시스템의 부팅 순서이다. 스타트업 코드와 alt_main 함수를 실행함으로써 하드웨어 초기화 및 프로그램이 사용할 메모리를 생성한 후 main 함수가 호출된다. 스타트업 코드와 alt_main 함수는 사용자가 직접 구현해야 되는 것이 아니라 Altera사에서 제공하며, Nios II 어플리케이션을 Build하고 실행할 때 자동으로 호출된다.

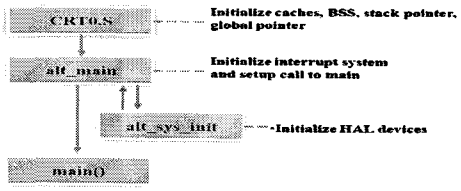

그림 11. Nios II 시스템의 부팅 순서

Fig. 11. Boot-sequence of Nios II system

그림 12는 실시간으로 출력되는 SONAR 신호의 획득 및 디스플레이를 위한 프로그램의 실행 흐름도이다. main() 함수가 실행되면 데이터를 획득하기 위한 변수와 디스플레이를 위한 프레임버퍼를 생성하고, VGA 제어기를 초기화한다. 데이터 획득 IP를 통해 x-y좌표와 제어기 신호를 실시간으로 획득한 후 제어신호의 비트 정의에 따라 각각의 신호를 분리하여 해당 변수에 할당한다. 분리된 제어 신호는 그림 4에서 정의된 값에 의해 해당 x-y좌표에 픽셀을 출력하여 한 프레임의 영상을 완성한다.

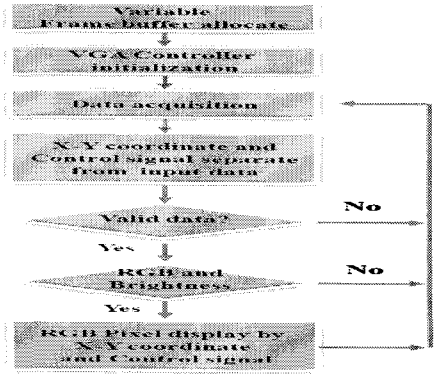

그림 12. 프로그램 흐름도

Fig. 12. Program flow-chart

V. IP 검증 및 실험결과

5.1 실험환경

본 논문의 실험을 위한 시스템 구성은 Altera Cyclone II 디바이스가 장착된 Terasic DE2-70 보드[6]를 사용하였으며, x-x좌표 및 제어 신호를 생성하기 위해서 NI사의 PCI-6534보드[10]를 사용하였다.

표 4. 실험 환경

Table 4. Experiment environment

▣ 임베디드 시스템 설계
▸하드웨어설계 : Quartus II 10.1
▸소프트웨어설계 : Nios II IDE 10.1
▣ X-Y 좌표 및 제어 신호 생성
▸하드웨어 : NI PCI-6534
▸소프트웨어 : NI LabWindows CVI 8.5

그림 13은 NI PCI-6534 보드와 DE2 70 보드를 이용한 시스템의 구성을 보인다.

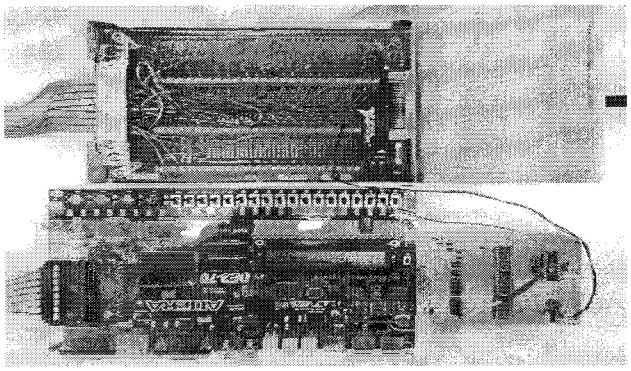

그림 13. 시스템 구성 장치

Fig. 13. System configuration

5.2 IP 시뮬레이션 검증

회로검증의 과정은 Window XP OS 환경에서 Modelsim 시뮬레이션 툴을 이용한 RTL 회로검증 그리고 Quartus II의 SignalTap II를 통한 FPGA 구현 실제 회로의 검증 단계로 진행된다. 그림 14는 SONAR 신호 획득 IP에 대한 Modelsim 시뮬레이션 결과이다. avs_s1_export_clk는 SONAR로부터 출력 되는 클럭이며, avs_s1_export_control은 유효비트와 색상을 판별하는 신호이며, avs_s1_export_x_cord 및 avs_s1_export_y_cord는 X-Y 좌표이다. SONAR로부터 출력되는 신호는 avs_s1_export_clk에 동기되어 avs_s1_export_control신호와 avs_s1_export_x_cord 및 avs_s1_export_y_cord가 SONAR 신호 획득 IP로 입력된다. 입력된 신호 중 X-Y좌표는 딜레이 로직을 거쳐 제어신호와 32비트 word_packed 신호로 결합되어 FIFO에 저장된다. 저장된 word_packed 신호는 read 신호와 클럭에 동기 되어 readdata 신호로 Nios II 프로세서로 전달됨을 확인할 수 있다.

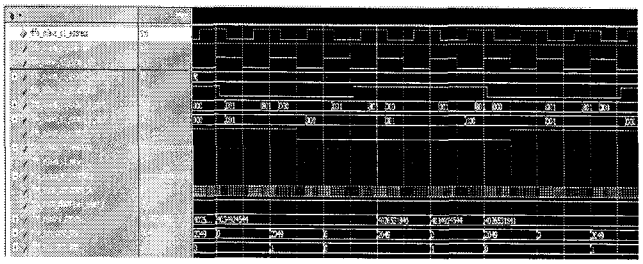

그림 14. SONAR 신호 획득 IP의 Modelsim 시뮬레이션 결과
Fig. 14. Modelsim simulation result of SONAR signal acquisition IP

그림 15는 구현 회로를 FPGA에 실장하여 실제 하드웨어 상에서 시뮬레이션한 결과이다. data_in_clk은 SONAR로부터 출력되는 클럭신호이며, x_cord 및 y_cord는 x-y좌표 신호이며, zLSB, zMSB, z1, zHD, MZ, F[1], F[0] 신호는 유효 데이터와 색상을 판별하는 제어 신호이다. SONAR로부터 출력되는 x-y좌표 및 제어 신호는 data_in_clk에 동기되어 SONAR 신호 획득 IP로 입력됨을 확인할 수 있다. oVGA_R, oVGA_G, oVGA_B와 oVGA_CLOCK, oVGA_BLANK_N, oVGA_HS, oVGA_VS, oVGA_SYNC는 VGA DAC와 모니터를 구동하기 위한 RGB 색상 및 동기 신호이다. oVGA_CLOCK에 동기되어 RGB 영상이 출력됨을 확인할 수 있다.

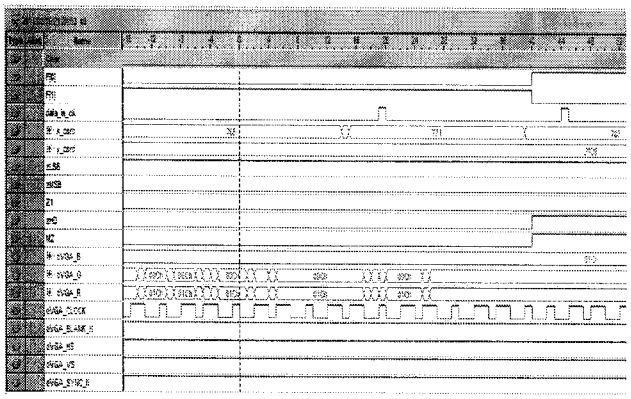

그림 15. 구현 회로의 시뮬레이션 결과
Fig. 15. Simulation result of implementation circuit

5.3 실험결과

SONAR 신호 획득 IP를 통해 입력되는 신호는 YCbCr이나 RGB 방식의 영상 정보가 아니고 x-y좌표 및 제어 신호가 입력된다. 따라서 입력 데이터를 생성하기 위해 CMOS 또는 CCD 카메라가 아니라 NI PCI-6534 보드를 사용하였다. 그림 16은 NI PCI-6534 보드로 x-y좌표 및 제어 신호를 생성하기 위한 LabWindows CVI 프로그램을 작성하여 실행한 결과이고, 가운데 그림이 NI PCI-6534 보드의 출력 포트를 통해 FPGA 보드의 확장포트로 입력된다.

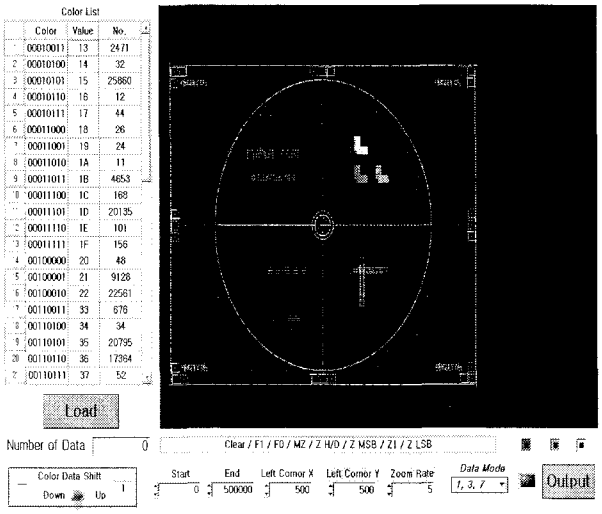

그림 16. CVI 프로그램의 실행 결과
Fig. 16. Execution result of CVI program

그림 17은 실제 FPGA 보드를 통해 실시간으로 획득한 데이터를 VGA 제어기를 통해 모니터로 출력한 그림이다. NI PCI-6534 보드와 LabWindows CVI 프로그램에서 생성한 데이터를 손실 없이 실시간으로 획득하여 모니터에 출력됨을 확인할 수 있었으며, 표 5는 구현된 IP의 성능 평가표이다. SONAR 신호 획득 IP는 입력신호를 22ms의 처리 속도를 제공하므로 초당 45회로 출력 성능을 제공할 수 있고, VGA 제어기의 경우 초당 30회의 처리 속도를 제공하며, 일반적인 디스플레이의 요구 조건을 충분히 만족하는 수준이다.

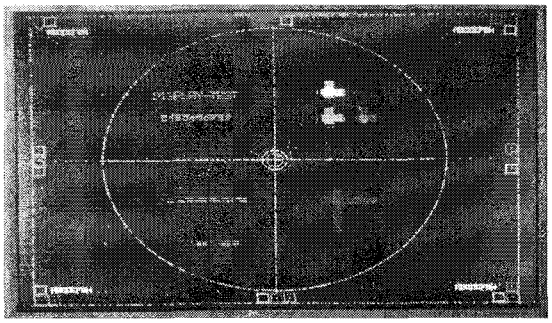

그림 17. 실험 결과 화면
Fig. 17. A screen of experimental result

표 5. IP의 성능

구 분	영상 사이즈	처리 속도
SONAR 신호 획득 IP	4096 * 4096	22 ms
VGA 제어기	1024 * 768	30 frame/sec

Ⅵ. 결 론

FPGA 임베디드 시스템은 각종 응용에 유연하고, 코어를 설정 가능하며, 높은 처리 속도를 제공함으로써 실시간 데이터 획득 및 처리에 적합하다.

본 연구에서는 기존의 SONAR 신호를 사용한 CRT 모니터 디스플레이 시스템의 단점인 복잡한 회로구성, 가격의 고가, 생산이 폐쇄되어 부품 수급 및 확보의 어려움을 극복하기 위해 FPGA 임베디드 시스템을 사용하였다. Altera Nios II 임베디드 시스템을 사용함으로써 실시간 SONAR 신호의 획득 및 디스플레이가 가능하였으며, 시스템의 회로 구성을 간단히 할 수 있었다. 실시간으로 출력되는 SONAR 신호를 획득하기 위한 SONAR 신호 획득 IP와 획득한 SONAR 신호를 RGB 컬러 신호로 변환하기 위한 그래픽 라이브러리, RGB 컬러 신호를 디스플레이 하기 위한 VGA 제어기를 구현하였다. 실시간으로 입력되는 SONAR 신호의 손실 없이 디스플레이 됨을 확인할 수 있었으며, SONAR 신호 획득 IP의 경우 4096*4096 사이즈의 데이터를 22ms 처리 속도를 보였으며, VGA 제어기는 1024*768의 영상을 초당 30프레임 출력할 수 있었다.

향후 연구 과제로는 운영체제를 포팅하여 멀티 태스킹이 가능하게 하고, 획득한 소나 신호를 디스플레이 하는 알고리즘을 하드웨어로 구현할 필요가 있다.

참 고 문 헌

[1] Chuanfei Qiu, Quanzhi Zhou, Changjun Wei, Fei Xu, "The Implement of High Speed Data Acquisition System Based-on SOPC Technology", Proceedings of the IEEE International, Automation and Logistics, pp.1528-1531, Aug 2009

[2] Wang Wei, Zhong Guidong, "The Design and Implementation of High-Speed Data Acquisition System Based on NIOS II", IEEE, Computer Society, pp.334-336, 2010

[3] Altera Corp. Embedded Design Handbook, V2.8, pp.1.1-11.15, Jan 2011

[4] Altera Corp. Simulating Nios II Embedded Processor Designs, V1.2, pp.1-9, Nov 2008

[5] Altera Corp, Using SignalTap II Embedded Logic Analyzers in SOPC Builder System, V.1.1, pp.1-21, Nov 2007

[6] Terasic Corp, DE2-70 User manual, V1.07, pp.1-89, 2009

[7] Altera Corp, Avalon Interface Specifications, V1.3, pp.1.1-7.2, Aug 2010

[8] Altera Corp, SOPC Builder User Guide, V1.0, pp.1.1-12.10, Dec 2010

[9] Altera Corp, Nios II Software Developer's Handbook, V10.0, pp.1.1-15.119, Jul 2010

[10] National Instruments Corp, NI 653X User Manual for Traditional NI-DAQ, pp.1.1-3.36, Feb 2005

[11] Altera Corp, SCFIFO and DCFIFO Megafunctions, V6.2, pp.1-27, Sep 2010

김 동 진 (Dong-Jin Kim)

2007년 2월 경남대 정보통신공학과(공학사)
2009년 2월 경남대 정보통신공학과(공학석사)
2011년 2월 경남대 정보통신공학과(공학박사수료)
※주관심분야 : FPGA설계, 임베디드 시스템

김 대 웅 (Dae-Woong Kim)

1997년 2월 경희대 컴퓨터공학과(공학사)
2011년 8월 경남대 산업대학원 정보통신공학과 석사 졸업
2005년 5월 ~ 현재 (주)한국티이에스 대표이사
※주관심분야 : FPGA설계, 임베디드 시스템

박 영 석 (Young-Seak Park)

正會員

1979년 : 영남대 전자공학과 학사 졸업.
1981년 : 한양대 전자공학과 석사 졸업
1985년 : 한양대 전자공학과 박사 졸업
1990~1991년 : 일본 우정성 통신총합연구소(관서선단연구센터) 초빙과학자
1990~1991년 : 일본 긴끼이동통신센터 객원연구원
2001년~2002년 : 미국 North Carolina 주립대학(NCSU) 교환교수
2001년~현재 : 경남대 정보통신공학과 교수
※주관심분야 : Software Engineering, Web-based Software Design & Development, Pattern Recognition, Image Processing, Computer Network & Network Computing, Embedded Processor System HW/SW