
고해상도 잉크젯 헤더 인터페이스를 위한 IP 설계 및 구현

이 종 혁*

Design and Implementation of IP for High Resolution Inkjet Header Interface

Jong-Hyeok Lee*

이 논문은 2007년도 경성대학교 학술연구비지원에 의하여 연구되었음

요 약

시스템 전반을 제어하는 임베디드 컨트롤러는 임베디드 시스템에서 가장 중요한 부분이다. 최근에는 임베디드 컨트롤러에 SoC가 ASIC보다 많이 사용하려고 하지만 긴 개발 기간, 높은 가격으로 중·소형시스템의 적용부분에는 문제가 많다. 그래서 많은 회사들이 아직은 임베디드 프로세서와 IP기술을 이용하고 있다.

고해상도 잉크젯 마킹 시스템은 임베디드 컨트롤러를 가지는 프린팅 시스템으로 산업 현장의 많은 부분에 사용되고 있다, 그러나 인쇄 품질, 마킹 에러, 시스템 에러 등의 문제를 가지고 있다.

본 연구에서는 인쇄품질의 문제를 해결하기 위하여 IP를 설계 구현하였고, 시뮬레이션을 통하여 논리소자의 총 개수 및 타이밍에 대한 비교 분석하였다. 분석결과 출력 신호들이 기준시간을 만족함을 알 수 있었으며, 구현한 IP를 고해상도 잉크젯 마킹시스템에 적용한 결과 양질의 출력 메시지를 얻을 수 있었다.

ABSTRACT

Embedded Controller which controls whole system is most important part of embedded system. Nowadays, new technique called SoC is more using than ASIC. But SoC have some problems. Because of long development time and high cost, it is hard to applying SoC to small and medium enterprise. So many companies use IP technique combined with embedded processor.

High resolution inkjet marking system is printing system with embedded controller. It is used in various part of industry. But it has many problems such as printing quality, marking errors, system faults and so on.

In this paper, we designed and implemented IP that can solve the printing quality problems. We analyzed total-logic-elements and timing by simulation. As a result of simulation, we could verified that output signals satisfied reference timing. Applying IP to high resolution inkjet marking system, we could get good quality printing message.

키워드

IP, 임베디드 시스템, 고해상도 잉크젯 시스템, 헤더 인터페이스

I. 서 론

컴퓨터의 중요 연산을 담당하는 프로세서의 발전은

우리 사회에 큰 변화를 가져다주었다. 1970년대부터 현재에 이르기까지 꾸준하게 발전해온 프로세서는 휴대폰 같은 소형의 전자기기에서부터 슈퍼컴퓨터 같은 고

성능을 요구하는 기기에까지 광범위하게 사용되고 있다. 기술의 발전으로 범용 프로세서를 사용하던 시스템에 임베디드 프로세서를 적용시키면서 시스템의 크기를 줄이고 제작비용을 절감하는 등의 효과를 가져다주었다. 특히 최근 10년간 전세계적으로 휴대폰이나 PDA(Personal Digital Assistants)같은 휴대용 단말기의 시장이 크게 발전하면서 예전의 범용 프로세서보다 휴대성과 편의성이 두드러진 임베디드 프로세서의 시장이 크게 증가하고 있다.[1]

임베디드 시스템이란 특정 목적만을 위한 동작을 수행하도록 하는 시스템으로서 여기에 사용되는 프로세서는 범용 프로세서보다 작고 간소화되어 있으며 저 전력에서 구동되게끔 설계가 되기 때문에 이를 임베디드 프로세서라 부른다. 이런 임베디드 프로세서들은 가전 제품이나 휴대폰 등 여러 영역에서 두루 사용되어지고 있다. 산업체에서 사용하는 마킹 시스템이나 자동차 공장의 차종인식 시스템도 모두 임베디드 시스템에 속한다. 하지만 대부분의 임베디드 프로세서는 특정 목적만을 수행하게 제작함으로써 인해 대부분의 환경에서 구동될 수 있게끔 만들어진 범용 프로세서와는 달리 호환성이 떨어지는 단점을 가지고 있고 그 처리능력 또한 범용의 프로세서보다는 떨어지는 편이다. 때문에 현재의 시장에는 각각의 특성을 고려한 많은 종류의 임베디드용 프로세서들이 유통되어지고 있다.[2]

반도체 기술의 발전으로 등장한 SoC(System on Chip)는 기존 프로세서의 하드웨어적 개념에 CPLD(Complex Programmable Logic Device)나 FPGA(Field Programmable Gate Array)로 대표되는 Programmable 디바이스 위에 소프트웨어적인 개념인 IP(Intellectual Property)가 탑재되는 형태를 띠고 있다. 예전에는 서로 독립적인 칩들을 보드 상에서 연결해서 구현해야 했던 시스템을 SoC를 이용하면 하나의 칩으로 구현이 가능해진다. 이러한 장점 때문에 SoC는 현재 휴대폰등의 통신기기를 중심으로 급속히 발전하고 있는 추세이다. 하지만 현재 우리나라는 외국보다 늦은 SoC의 도입으로 인해 그다지 높은 기술력을 갖추지는 못하고 있고 주요 사업에서 사용되는 핵심 IP의 경우는 외국에서 사서 쓰는 실정이다. 게다가 실질적으로 칩 하나에 모든 기능을 구현하여 탑재한다는 것은 많은 시간과 노력을 필요로 하기 때문에 현재의 빠른 시장추세를 따라가기는 힘들다. 그래서 현재의 시스템들은 기존의 임베디드 프로세서에 IP를 부분적으로 도입하고

기타 주변소자를 갖는 형태를 많이 사용하고 있다.[3],[4]

고해상도 잉크젯 마킹 시스템은 임베디드 컨트롤러를 가지는 프린팅 시스템으로 산업 현장의 많은 부분에 사용되고 있다, 그러나 인쇄 품질, 마킹 에러, 시스템 에러 등의 문제를 가지고 있으며, 특히 인쇄 품질은 헤더 인터페이스의 출력 타이밍과 직결된다. 개발자가 원하는 타이밍을 가지기 위해서는 하드웨어의 회로가 복잡해지고, 회로 수정 시 비용이 많이 들며, 정확한 타이밍을 만들기 어려우므로 이런 부분에 IP들이 많이 사용되고 있는 실정이다.

본 연구에서는 고해상도 잉크젯 마킹 시스템에서 헤더 인터페이스용 IP를 설계하고 이를 구현 시 컴포넌트 개수에 따른 논리소자의 총개수 및 타이밍에 대한 비교 분석과 출력 데이터 처리를 시뮬레이션으로 확인하며, 실제 시스템에 적용해 보고자 한다.

II. IP 설계기법

2.1 IP의 정의

IP란 Intellectual Property의 약자로서 특허, 복사권 또는 무역기밀에 의하여 보호되어지는 상품, 기술, 또는 소프트웨어 등을 의미한다. IP는 다른 의미로 보면 메가 펄스, 매크로 블록, 또는 재사용 가능한 컴포넌트로서 불리어지기도 한다. 여기서 재사용 가능한 컴포넌트라는 의미가 중요한데 이것은 어떤 물리적 구현이 재사용 되어질 수 있도록 존재하는 컴포넌트의 형태로 참조될 수 있다. 이러한 예로서 큰 규모의 칩 안에 포함된 ALU 또는 매크로 셀들과 같은 설계객체들을 말한다.

2.2 IP의 검증

IP의 검증이 완료된 것을 입증하는 방법은 해당 IP가 탑재된 칩을 제작하여 실장 테스트까지 통과함으로써 아무런 하자가 없음을 증명하는 것이다. 즉 해당 IP에 대한 사용을 고려하는 사람의 입장에서 정상 동작하는 칩이 개발될 수 있는가가 관건이므로 이에 대한 개발 여부가 확고한 입증이 되는 것이다. 게다가 IP생성과정에서 테스트까지 고려하였고, Soft IP의 경우 코드 커버리지 등이 고려되어 개발된 것이라면 완벽하게 검증된 IP라고 할 수 있다. 또한 Synopsys사와 Mentor사가 공동으로 마련한 OpenMORE 프로그램인 RMM(Reuse Methodology

Manual: Synopsys사와 Mentor사가 공동으로 저술한 재사용 관련 매뉴얼)을 이용하여 설계된 IP의 설계규칙 부합 여부를 평가할 수 있다. 근래에는 FPGA OpenMORE을 마련하여 FPGA에 의한 방법으로 평가해 볼 수 있도록 되어 있다. 한편 검증된 IP라는 것을 확인할 수 있는 가장 간단한 방법은 실제 칩을 제작하여 판매되고 있는가의 여부를 알아보면 된다.

2.3 IP 설계기법

VHDL(Very high speed integrated circuit Hardware Description Language)[5]을 이용한 IP의 기본적인 설계 기법에는 크게 4가지가 존재한다. 부울 대수를 이용하여 회로를 정의하는 data flow description, 각각의 구성요소들을 선으로 연결하여 회로를 정의하는 structural description, 알고리즘적인 코드를 이용하여 회로를 정의하는 behavioral description, 그리고 원하는 결과 값의 파형을 이용하여 역으로 회로를 유추해 나가는 timing design의 방법이 있다.

Data flow description은 주로 병행처리문에서 사용되며, 문장의 순서에 무관하게 동시에 수행 된다. 기본적으로 부울대수식과 연산자 등을 이용하여 입력과 출력사이의 관계를 나타내게 된다. 설계의 내부 논리가 부울함수로 쉽게 표현될 때에 한해서 유용한 표현이다. 하지만 디지털 시스템의 모든 기능을 부울대수로 기술하는 것은 어려운 일이다.

Behavioral description은 가장 높은 레벨의 추상적인 표현으로서 기능적 또는 수학적인 알고리즘을 사용하여 시스템의 기능을 기술한다.(기존의 프로그래밍언어 스타일과 가장 유사) process문이 가장 대표적이며, process문 내부에는 if, case, for등의 순차 처리문이 사용되어진다. 모든 회로를 Behavioral description으로 설계하는 것이 효과적일 것처럼 보일수도 있으나 설계하고자 하는 회로의 크기와 복잡도가 증가할수록 구현이 어려워지기도 한다.

Structural description은 표현 방법 중 하드웨어에 가장 가까운 하위 레벨의 표현으로써 컴포넌트들의 상호 연결 관계를 나타내어 하드웨어를 표현한다. 즉 이미 설계된 컴포넌트를 이용하여 그 컴포넌트들을 서로 연결하여 system을 기술하는 방식이다. 대표적인 Structural description으로 component문, generate문, 그리고 generic문이 있다.

Timing design은 만족해야 하는 결과 값이나 상태의

변화들을 가지고 회로를 역으로 기술해 나가는 방법이다. state machine이나 counter, shift register를 이용한 방법이 많이 사용되어지며 간단한 출력 값이나 상태의 변화를 가지는 시스템은 쉽게 적용 가능하지만 규모가 크고 복잡한 시스템에 적용하기에는 문제점이 있다.

위에서 언급한 네가지 IP 설계방법들은 방법론적인 분류로서 실제로는 이중 두가지 이상이 동시에 사용되어 IP가 구현되어지는 경우가 대부분이다.

III. 고해상도 잉크젯 헤더용 IP 설계

프린터 헤더에 전원이 처음 들어갈 때는 언제나 configure 데이터가 프린터 헤더로 전송되어야 하며 이때의 데이터 스트로브 신호, 데이터, 데이터 출력방향에 대한 요구 타이밍을 그림 1에 나타내었다.

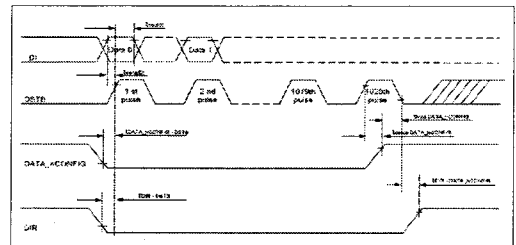


그림 1. configuration data 요구 타이밍
Fig. 1. Configuration data timing diagram

그림 2에 출력 데이터가 제대로 전송되기 위한 헤더의 요구 타이밍을 나타내었다. 하나의 라인을 구성하는 128개의 노즐을 구동하기 위해서 128bit의 데이터가 데이터 스트로브 신호와 동기되어 가해지고 128개의 데이터가 모두 전송되면 사이클 스트로브 신호와 동기되어 노즐이 개폐된다.[6]

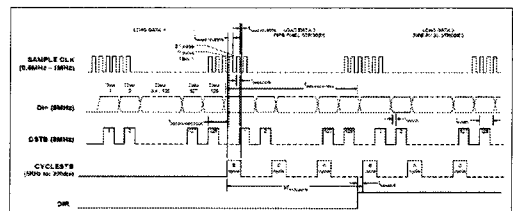


그림 2. 출력 데이터의 요구 타이밍
Fig. 2. Print data loading and control signal

그림 3은 프린팅 작업을 수행할 프린터 헤더를 직접적으로 제어할 IP의 전체적인 흐름도이다. 각각의 상황에 맞는 설정을 읽어 들여 정확한 타이밍에 헤더로 데이터를 전송하고 헤더의 상태를 확인하는 것이 주목적으로 CPLD에 탑재되어 그 기능을 수행하게 된다.[7]

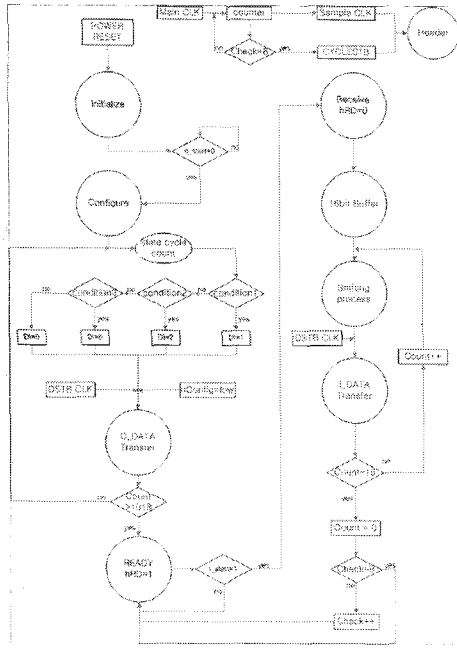


그림 3. IP의 전체 동작도
Fig. 3. Overview of IP operation

IP의 제작에 앞서 세부적인 동작의 수행을 위한 개념도를 작성해 그림 4에 나타내었다. configure 데이터 파트와 이미지 데이터 파트로 나누어 동작을 하게 되며, configure 데이터 파트의 경우는 전원 공급 시 한번만 수행을 하게 된다. 반면 이미지 데이터 파트는 지속적으로 동작하면서 MCU에서 이미지 데이터가 전송될 때마다 이를 처리하게 된다.

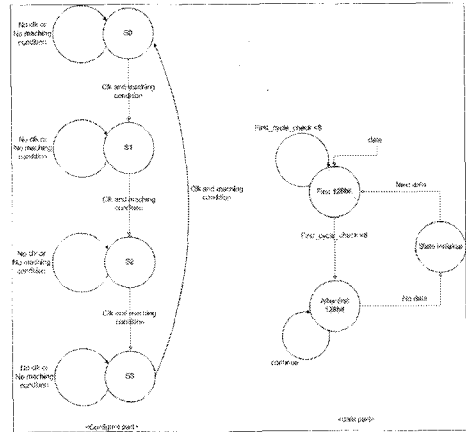


그림 4. IP의 세부 동작도
Fig. 4. Detail operation diagram of IP

IV. 프린터 헤더용 IP의 구현

4.1 IP 프로그래밍

실제 프린팅 작업을 제어하고 헤더의 상태를 확인하는 컨트롤 IP를 작성하였다. 작성 언어는 VHDL을 이용하였고 기본적인 IP 설계방식들인 data flow, structural, behavioral 방식을 혼합적으로 사용하여 설계하였다.

MCU에서 CPLD로 넘어오는 데이터는 병렬데이터이다. 하지만 프린터 헤더는 직렬데이터만을 받아들이므로 이를 변환해주는 작업이 필요하다. 그림 5는 입력된 병렬 데이터를 직렬 데이터로 변환하는 VHDL코드이다. 하나의 bit를 보내기 위해 5개의 state를 거치도록 설계하였다. 이를 통해서 시스템이 요구하는 지연 범위를 충족시킬 수 있었다. 이 외에도 IP의 내부에는 동작 타이밍 조절을 위한 클럭 설정부와 헤더의 상태에 따른 인터럽트부 등 부수적인 코드가 들어가 있다.

```

elsif (i_start = '1' and c_start = '0') then
    di(0) <= '0';
    di(1) <= '0';
    di(2) <= '0';
    if (first_cycle_check < 8) then
        READY <= '0';
        if(d_cnt<16) then
            if (check = c0) then
                di(3) <= i_data(d_cnt);
                check <= c1;
            elsif (check = c1) then
                dstb <= not dstb;
                check <= c2;
            elsif (check = c2) then
                check <= c3;
            elsif (check = c3) then
                dstb <= not dstb;
                check <= c4;
            elsif (check = c4) then
                check <= c0;
                d_cnt <= d_cnt + 1;
            end if;
        else
            first_cycle_check <= first_cycle_check + 1;
            d_cnt <= 0;
            if (first_cycle_check<7) then
                READY <= '1';
            end if;
        end if;
    end if;
end if;
    
```

그림 5. VHDL 코드(데이터 변환부)
Fig. 5. VHDL code(data converter part)

4.2 IP의 검증

코딩이 끝난 VHDL 파일을 QuartusII를 이용하여 컴파일 작업과 시뮬레이션 작업을 수행 하였다. 논리 오류나 코딩의 오류 등을 모두 체크하고 수차례의 컴파일 작업을 수행하여 컴파일 작업을 완료하였다. 컴파일 후 심벌 파일을 생성시키고 시뮬레이션 입력값을 위한 파일을 생성하는 등의 일련의 과정을 거쳐 시뮬레이션이 수행되었다.

컴포넌트 도입이 IP 설계에 미치는 영향을 분석하기 위하여 컴포넌트를 1개 또는 2개로 했을 시의 시뮬레이션 결과를 표 1에 나타내었다. 컴포넌트를 2개로 하였을 경우 총논리 소자 개수는 5.3%의 감소를 가져왔으며, 시간 분석 결과 홀드시간은 무려 48.5%의 감소를 가져왔다. 이 결과 IP 설계에서 컴포넌트를 많이 도입하는 것이 전체적으로 효과가 있음을 알 수 있었으며, 총논리 소자의 개수를 고려할 때 저사양의 Programmable Device에서도 충분히 동작할 수 있음을 알 수 있었다.[8]

표 1. 컴포넌트 개수에 따른 시뮬레이션 결과
Table 1. Simulation result by component number

	컴포넌트수 1개	컴포넌트수 2개	비고
Total logic elements	209	198	5.3%
Worst-case tsu	13.9ns	10.5ns	24.5%
Worst_case tco	8.4ns	8.1ns	3.6%
worst_case th	-3.3ns	-4.9ns	48.5%

그림 6에 시뮬레이션 후 Timing Analysis부분을 나타내었다. Timing Analysis 결과 대부분의 신호들이 5ns 안쪽의 딜레이를 가지는 것을 볼 수 있다. worst case의 경우 10ns정도를 보였다. 시스템이 요구하는 신호의 셋업 타임이나 홀드타임 등의 시간적 요인을 모두 적용시켜도 오차범위(20ns) 내에서 결과 값이 출력됨을 확인할 수 있었다.

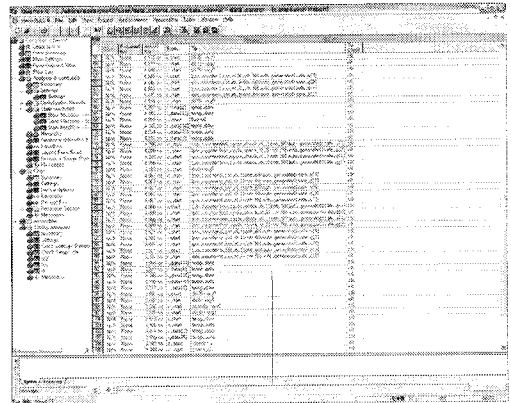


그림 6. Timing analysis
Fig. 6. Timing analysis

시뮬레이션 작업이 끝나면 Wave form 파일을 통해서 결과를 확인할 수 있다. 그림 7은 시뮬레이션에 의한 결과 값을 보여준다. CPLD에 들어온 병렬의 데이터가 클럭 이벤트에 맞춰 직렬의 데이터로 순차적으로 출력이 됨을 확인할 수 있었다. 처음 configure data가 먼저 들어간 후 I_star 핀이 high가 되고 데이터가 들어가자 dstb 사이클이 생성되고 출력 데이터 DI에 그에 상응하는 출력값이 들어가 있는 것을 확인할 수 있었다.

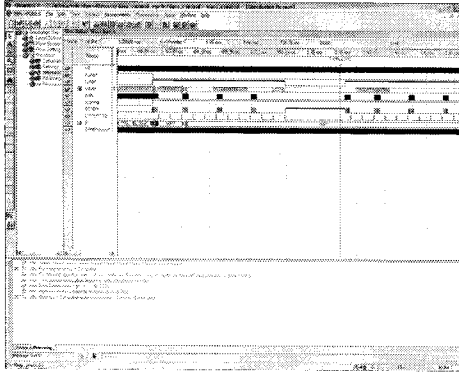


그림 7. 전체 시뮬레이션 결과
Fig. 7. Total simulation result

그림 8에 앞에서 하나의 image data가 끝나고 대기상태로 있다가 다시 image data가 들어오자 다시 동작을 시작하는 모습을 나타내었다. MCU에서 데이터를 보내지 않으면 설정 값들을 초기화 시키고 READY 상태로 대기하다가 데이터를 다시 보내면 처음부터 data 처리부가 구동되게 된다.

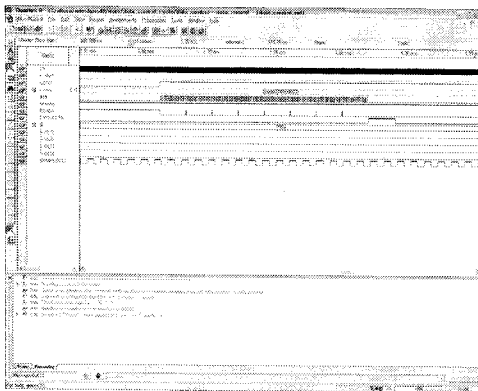


그림 8. 새로운 데이터의 처리
Fig. 8. New data processing

구현한 IP를 고해상도 잉크젯 시스템에 적용해 보았으며 이를 그림 9에 나타내었으며, 프린터 출력 결과를 그림 10에 나타내었다. 테스트 결과 일정 속도로 진행되는 물체의 표면에 고품질의 인쇄를 할 수 있었다. 그리고 인쇄메시지의 날자나 시간 같은 자동 업데이트 부분도 실시간적인 업데이트를 통한 인쇄로 만족스러운 결과를 볼 수 있었다. 하지만 적정 인쇄 거리를 벗어나 진입하는 물체의 경우 인쇄가 흩어지는 경우도 있었다. 이를

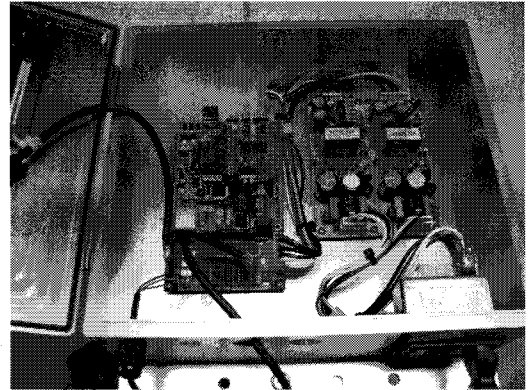


그림 9. 구현된 시스템
Fig. 9. System implementation



그림 10. 인쇄 이미지 견본
Fig. 10. Sample of printing image

V. 결 론

시스템 전반을 제어하는 임베디드 컨트롤러는 임베디드 시스템에서 가장 중요한 부분이다. 최근에는 임베디드 컨트롤러에 SoC가 ASIC보다 많이 사용하려고 하지만 긴 개발 기간, 높은 가격으로 중·소형시스템의 적용부분에는 문제가 많다. 그래서 많은 회사들이 아직은 임베디드 프로세서와 IP기술을 이용하고 있다.

고해상도 잉크젯 마킹 시스템은 임베디드 컨트롤러를 가지는 프린팅 시스템으로 산업 현장의 많은 부분에 사용되고 있다, 그러나 인쇄 품질, 마킹 에러, 시스템 에러 등의 문제를 가지고 있다.

본 연구에서는 인쇄품질의 문제를 해결하기 위하여 IP를 설계 구현하였고, 시뮬레이션을 통하여 논리소자의 총개수 및 타이밍에 대해 비교 분석하였다.

컴포넌트를 1개로 하였을 경우보다 컴포넌트를 2개로 하였을 경우 총논리 소자 개수는 5.3%의 감소를 가져왔으며, 시간 분석 결과 홀드시간은 무려 48.5%의 감소를 가져왔다. 이 결과 IP 설계에서 컴포넌트를 많이 도입하는 것이 전체적으로 효과적임을 알 수 있었으며, 총논리 소자의 개수를 고려할 때 저사양의 Programmable Device에서도 충분히 동작할 수 있음을 알 수 있었다. 또한 실제 고해상도 잉크젯 마킹 시스템에 적용시켜 테스트한 결과 일정 속도로 진행되는 물체의 표면에 고품질의 인쇄를 할 수 있었다. 그리고 인쇄메시지의 날자나 시간 같은 자동 업데이트 부분도 실시간적인 업데이트를 통한 인쇄로 만족스러운 결과를 볼 수 있었다.

하지만 인쇄 거리를 벗어나 진입하는 물체의 경우 인쇄가 흩어지는 경우가 있었으므로 향후 이에 대한 보완 방법을 모색할 계획이다.

참고문헌

- [1] 반도체 설계재산 유통센터, "<http://www.kipex.or.kr>"
- [2] 박인철, "임베디드 프로세서 동향", KIPEX 09/10 2006 NO.1, EZPEX, 2006.
- [3] Wayne wolf, "Computer as Component : principles of Embedded Computing System Design", MORGAN KAUFMANN PUBLISHERS, 2001.
- [4] Simon, "임문자를 위한 임베디드 시스템", 사이텍 미디어, 2004.
- [5] 박세현, "VHDL 기본과 활용", 도서출판 그린, 1998.
- [6] XAAR, "<http://www.xaar.co.uk>"
- [7] 조용범, "컨트롤러 설계를 위한 VHDL 활용", 전력전자 학회지, 3권 제 2호, pp.19-25, 1998.
- [8] Altera Inc., "<http://www.altera.com>"

저자소개



이 종 혁(Jong-Hyeok Lee)

1975년 부산대학교 전자공학과 학사
1980년 부산대학교 전자공학과 석사
1991년 부산대학교 전자공학과 박사

1990년~현재 경성대학교 컴퓨터공학과 교수
1998년 7월~1999년 6월 미국 Beckman Institute, Univ. of Illinois, 객원연구원

※ 관심분야: 인공지능, 음성인식, 신호처리