

論文2003-40CI-5-7

128채널 PC 기반 하드웨어 시뮬레이터 구현

(Implementation of a PC based Hardware Simulator with 128 channels)

丁甲天*, 崔鍾鉉**, 朴性模**

(Gab Cheon Jung, Jong Hyun Choi, and Seong Mo Park)

요약

본 논문은 디지털 회로의 검증 및 테스트에 유용한 128 채널 하드웨어 시뮬레이터의 구현에 대하여 기술하였다. 하드웨어 시뮬레이터는 로직분석기와 신호발생기의 기능을 동시에 수행한다. 각 채널에 해당하는 코어 모듈은 독립적인 메모리와 내부 모드를 가지고서 하나의 컨트롤러처럼 동작하기 때문에 코어모듈을 추가함으로써 채널 수를 쉽게 확장할 수 있다. 또한 PC를 기반으로 하고 있어 저가형 시스템으로 구현 가능하고, 편리한 GUI(Graphic User Interface) 구성을 할 수 있다. FPGA를 이용하여 구현된 시뮬레이터는 최대 50MHz에서 동작하며 평균 55W의 전력을 소모한다.

Abstract

This paper describes a 128-channel hardware simulator that is useful for verification and testing of digital circuits. It performs logic analyzer function and signal generator function at the same time. The core module, which implements one channel of the simulator, operates as a controller with independent memory and internal mode. Therefore, we can easily extend the number of channels with addition of core module. Moreover, since the simulator was implemented as a PC based system, one can construct a low-cost system and can configure convenient GUI(Graphic User Interface) environment. The simulator implemented using FPGA operates at 50Mhz and consumes 55W power as average.

Keywords : hardware simulator, verification, testing, logic analyzer, signal generator

I. 서론

ASIC을 포함한 디지털 회로를 설계하는데 있어 시뮬

* 正會員, 全南大學校 電子工學科

(Dept. of Electronics Eng., Chonnam National Univ.,)

** 正會員, 全南大學校 컴퓨터工學科

(Dept. of Computer Eng., Chonnam National Univ.)

※ 본 논문은 한국과학재단 지정 전남대학교 고품질 전기전자부품 및 시스템연구센터의 연구비 지원에 의해 연구되었음.

接受日字:2003年6月17日, 수정완료일:2003年8月21日

레이션을 통한 회로의 검증은 필수적이며, 이러한 검증을 위해서는 회로를 소프트웨어적으로 모델링하여 시뮬레이션을 하거나 회로를 FPGA에 미리 구현하여 동작 및 기능을 검증하는 하드웨어적인 방법을 사용한다. 집적 회로 기술의 향상에 따라 설계 회로의 규모와 복잡성이 날로 증가 되어지고 있어 소프트웨어적인 시뮬레이션만으로는 검증하기가 어려워지고 있고, 소프트웨어 모델은 실제 하드웨어에서 발생하는 전달 지연과 같은 요소를 정확히 고려하지 못하므로 시뮬레이션 결과에 대한 신뢰성이 떨어진다.

소프트웨어 시뮬레이션이 회로의 규모 및 복잡도에

따라 연산에 많은 시간을 소요하는 것과는 달리 하드웨어적인 시뮬레이션은 회로의 규모에 무관하게 빠른 시뮬레이션이 가능하고, 결과도 실제 상황에 가까운 검증이기 때문에 신뢰할 수 있다. 이러한 장점들로 인해 회로 검증을 위한 하드웨어 에뮬레이터와 시뮬레이션 시스템들이 많이 개발 되어지고 있다^{1, 5)}.

일반적으로 하드웨어 시뮬레이션을 위해서는 회로를 구현한 에뮬레이터와 입출력 신호를 처리하는 시뮬레이터가 필요한데 에뮬레이터에 내장되어 있는 시뮬레이터를 사용하거나²⁾, 독립된 에뮬레이터를 사용하는 경우 신호발생기(Signal Generator)와 로직 분석기(Logic Analyzer)등을 에뮬레이터에 연결하여 검증 환경을 구축하게 된다^{3, 5)}. 신호 발생기나 로직 분석기 등을 독립적으로 사용하여 검증환경을 구축할 경우는 각 장치간의 동기화 문제가 있을 뿐만 아니라 장치에 대한 비용도 많이 든다. 이러한 장치들은 대부분 자체 제어판과 CRT를 가지고 있지만 인터페이스가 편리하지 않고 입출력의 개수가 많은 경우 사용자는 출력된 결과를 한 화면에서 확인하기가 힘들다. 또한 개발자는 C 소스나 HDL과 같은 상위수준의 설계에서 사용했던 테스트벡터를 그대로 가져와 최종 개발된 하드웨어에서 테스트 할 필요가 있는데, 일반 테스트 장비의 경우 테스트 벡터를 디스크에 저장하거나 비교하는 등의 작업을 수행하기가 쉽지 않기 때문에 테스트를 위한 부가적인 회로가 요구되어진다.

본 논문에서는 신호발생기와 로직분석기 기능을 동시에 가지는 PC 기반 하드웨어 시뮬레이터의 구현에 대하여 기술하였다. II장에서는 제안된 하드웨어 시뮬레이터의 전체적인 구성을 기술하고, III장에서는 각 채널에 해당하는 코어 모듈의 동작을 기술하며, IV장에서는 시뮬레이터의 하드웨어/소프트웨어 구현 및 유사한 시스템들과의 비교를 하였으며, V장에서 결론을 맺는다.

II. 시뮬레이터의 구성

본 논문에서 제안된 하드웨어 시뮬레이터는 구현된 회로의 칩 레벨에서의 검증을 위한 시스템이다. 사용자는 시뮬레이터의 128개의 채널 중 필요한 수 만큼을 테스트 회로에 연결한 후 PC 상에서 테스트 패턴을 입력하게 되면, 시스템은 테스트 회로에 이 패턴 값들을 인가하게 되고 대응하는 테스트 회로의 출력을 다시

PC에 업로드하여 출력 파형을 디스플레이 한다. 사용자는 테스트 회로를 입출력타입에 상관없이 시뮬레이터의 채널에 연결할 수 있으며, 시스템의 각 채널은 1M 비트의 테스트 패턴 또는 출력 파형을 저장할 수 있다. 구현된 하드웨어 시뮬레이터의 내부는 크게 PC 인터페이스 카드와 시뮬레이션 보드의 두 부분으로 구성되어 있다. PC 인터페이스 카드는 PC와 시뮬레이션 보드사이의 인터페이스를 담당하며, 시뮬레이션 보드는 PC의 명령에 따라 테스트 회로의 입력에 테스트 패턴을 제공하거나 테스트 회로의 출력값을 계측하여 자체 메모리에 저장하는 기능을 수행한다.

1. PC 인터페이스 카드

일반적인 PC의 시스템 버스인 PCI(Peripheral Component Interconnect) 버스는 최고 33MHz의 가변 버스 클럭을 사용하고 32 비트의 어드레스와 데이터 버스가 같은 신호선을 사용하면서 시간적으로 multiplex 되어진다⁶⁾. 시간적으로 multiplex 되어지는 어드레스와 데이터 버스의 제어를 간단히 하기 위해서는 PCI 버스를 어드레스 버스와 데이터 버스가 각기 다른 신호선을 사용하는 로컬 버스로 변환 시켜주는 회로가 필요하며, 이를 위해 본 시뮬레이터는 PLX사의 PCI9050 칩을 사용하여 변환된 로컬 버스상에 PC호스트와 시뮬레이션 보드간의 입출력 및 제어신호 전달을 위한 인터페이스 로직을 구성하였다⁷⁾. <그림 1>은 PCI9050과 시뮬레이션 보드 인터페이스로 구성된 PC 인터페이스 카드를 나타낸다.

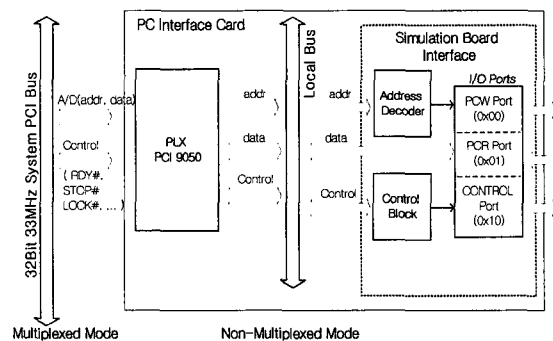


그림 1. PC 인터페이스 카드의 블록도
Fig. 1. Block Diagram of the PC Interface Card.

PC 호스트에서 시뮬레이션 보드로의 접근은 메모리 맵 I/O 접근을 통해 수행되어지도록 하였으며, 이를 위해 시뮬레이션 보드 인터페이스는 테스트 패턴 데이터

의 쓰기(PCW 포트), 측정된 출력 파형 데이터의 읽기(PCR 포트), 클럭발생기 및 시뮬레이션 보드 제어(Control 포트) 등의3개의 I/O 포트를 가진다. <표 1>은 각 포트들에 대한 상세한 기능들을 나타낸다.

표 1. 시뮬레이션 보드 인터페이스를 위한 입출력 포트
Table 1. I/O ports for simulation board interface

번지	포트	기능
0x00	PCW[7:0]	Write to simulator's data bus
0x01	PCR[7:0]	Read from simulator's data bus
0x02	CON[7](SIM_PLLB)	0: PLL setting mode, 1: Simulation mode
	CON[6](PCW_PCRB)	0: Read from simulator, 1: Write to simulator
	CON[5](SCL)	Serial Clock
	CON[4](SDA)	Serial Data
	CON[3](SMM_SEL[1:0])	Sub Module select
	CON[1:0]	Operation Control

<표 1>에서 SCL과 SDA는 PLL(Phase Lock Loop)에 원하는 주파수를 설정하거나, 각 채널의 내부 모드 비트를 설정하는데 사용되어지고, SIM_PLLB 비트는 현재 SCL, SDA 신호가 시뮬레이션 보드내의 PLL 설정하는데 사용될 것인지 또는 시뮬레이션 동작에 사용될 것인지를 결정한다. PCW_PCRB 비트는 PCW포트에 데이터를 쓸 것인지 또는 PCR 포트에서 데이터를 읽어 오는지를, SMM_SEL[1:0] 비트는 시뮬레이션 보드의 서브모듈을 선택하는데, CON[1:0]는 선택된 채널에 대한 동작 제어를 위해 사용되어진다.

2. 시뮬레이션 보드

시뮬레이션 보드는 <그림 2>에 보는 바와 같이 채널들의 어레이로 이루어진 서브 모듈 네 개(A, B, C, D)와 각 서브 모듈로의 데이터나 제어신호 분배, 그리고 클럭 발생 등의 기능을 수행하는 전역 제어부(Global Control Block)로 구성된다.

전역 제어부내 디코드 블록은 컨트롤 포트의 값을 디코드하여 클럭 발생기의 주파수 설정, 서브모듈 선택 및 제어, 데이터 버스의 제어 신호를 만들어낸다. 클럭 발생기는 391KHz~90MHz 클럭을 생성할 수 있는 Cypress사의 ICD2053 프로그램 가능한 PLL을 사용하였으며^[8], PLL은 디코드 블록으로부터 입력되는 QSDA(Serial Data)와 QSCL(Serial Clock)의해 구현된 FPGA

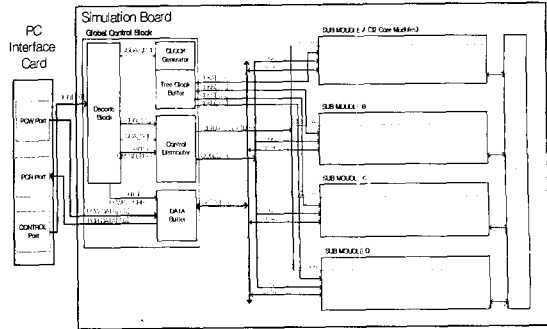


그림 2. 시뮬레이션 보드의 블록도
Fig. 2. Block diagram of the simulation board.

의 최대 동작 속도인 50MHz 클럭을 발생한다. 채널의 수가 많을 경우 PLL의 출력은 모든 채널을 구동하지 못하므로 클럭 버퍼를 트리 구조로 구성하여 채널들을 구동하도록 하였고, 이때 트리 구조는 하나의 서브모듈에 4개의 클럭을 공급하고 하나의 클럭은 8개의 코어 모듈을 구동하도록 구현되었다. 제어 신호 분배기(Control Distributer)는 디코드 블록의 제어신호를 서브모듈 각각에 분배하는 역할을 수행하며, 데이터 버퍼는 PC호스트와 서브모듈들 간의 데이터의 이동 경로와 방향을 설정한다.

각 서브모듈은 32개 채널들의 어레이로 구성되었으며, 테스트 회로와의 연결을 위한 32개의 SDATA 신호선을 가진다. 서브모듈내 각 채널들은 SDA 신호선들에 의해 직렬로 연결되어 있고, 전역 제어부와 인터페이스를 위한 데이터 버스(DATA[7:0]), 동작 제어 신호(CON[1:0], SCL(Serial Clock) 신호선들을 공유한다.

III. 코어 모듈

하드웨어 시뮬레이터의 각 채널에 해당하는 코어 모듈은 모드에 따라 신호발생기 역할을 수행하거나 로직 분석기 역할을 수행하는 실질적인 블록으로 각 채널의 테스트 패턴을 저장하고 있는 SRAM의 데이터를 테스트 회로에 출력하거나, 테스트 회로의 출력 값을 샘플링하여 SRAM에 저장하는 역할을 한다. 그리고 PC가 인터페이스 카드를 통해 SRAM의 데이터를 읽거나 쓸 수 있는 인터페이스도 제공한다. 코어 모듈은 <그림 3>과 같이 모드 레지스터, SG(Signal Generator)/LA(Logic Analyzer) 블록, SRAM 제어 블록, 코어 제어 블록으로 구성되어진다

모드 레지스터는 코어 모듈의 동작 모드를 결정한다.

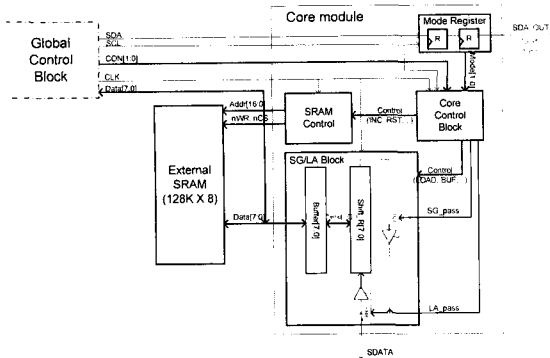


그림 3. 코어 모듈 블록도
Fig. 3. Block diagram of the core module.

설정되는 모드 레지스터 값에 따라 코어 모듈은 시뮬레이션 수행을 위한 모드인 SG(Signal Generator), LA (Logic Analyzer) 모드와 PC와의 인터페이스를 위한 모드인 PCR(PC Read), PCW(PC Write) 등의 4가지 모드를 가진다.

SG/LA블록은 병렬 입출력이 가능한 쉬프트 레지스터와 1바이트버퍼로 구성되며, 이 블록의 동작에 따라 코어 모듈은 신호 발생기 또는 로직 분석기로 각각 다르게 동작할 수 있다. <그림 4>는 코어 모듈내 SG/LA 블록의 동작을 나타낸다. 코어 모듈이 신호 발생기로 동작할 경우 SRAM으로부터의 데이터를 저장하고 있는 버퍼의 내용을 8 클럭을 한 주기로 쉬프트 레지스터에 병렬 로드하고, 쉬프트 레지스터는 한 클럭마다 한 비트 단위로 MSB(Most Significant Bit) 방향으로 출력함으로써 테스트 회로에 신호를 발생하게 된다(<그림 4(a)>). 반면 로직 분석기로 동작할 경우 쉬프트 레지스터는 테스트 회로에서 측정된 SDATA 데이터를 1 클럭마다 샘플하여 LSB(Least Significant Bit)에서 MSB 방향으로 한 클럭마다 이동시키고, 8 클럭을 주기로 하여 쉬프트 레지스터의 내용은 버퍼에 저장된다. 버퍼에 저장된 데이터는 다음 샘플 데이터가 버퍼에 저장되는 8사이클 내에 SRAM에 저장되어진다.

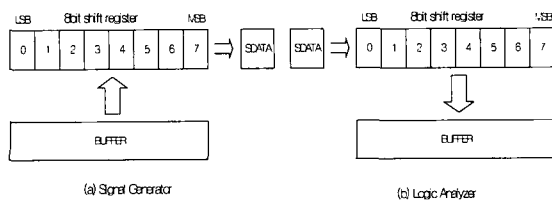


그림 4. SG/LA 블록의 동작
Fig. 4. Operation of SG/LA block.

SRAM 제어 블록은 SRAM 주소를 초기화시키거나, 시뮬레이션 사이클에 맞추어 해당 주소에 데이터를 읽고 쓰기 위한 제어 신호들을 발생시킨다. 코어 모듈 내부 블록들의 제어를 담당하는 코어 제어 블록은 코어 모듈의 내부모드와 시뮬레이션 보드의 전역제어부의 명령(CON[1:0])에 따라 해당 블록들에 대한 제어를 수행한다. <표 2>는 전역 제어부의 명령들과 각 명령에 따라 적용되어지는 코어 모듈의 모드를 나타낸다.

표 2. 전역제어부 명령
Table 2. Commands of Global Control Bloc.

CON[1:0]	명령	동작	코어 모듈 모드
00	Mode_set	Set mode registers of each channel	All
01	Run	Start simulation	SG or, LA
10	Pc_inc_address	Increase SRAM address	All
11	Pc_reset_address	Reset SRAM address to zero	All

Mode_set 명령은 각 코어 모듈의 내부 모드를 설정하는 명령으로서, 전역 제어부는 SDA에 각 채널의 모드 비트를 실어 주고, SCL을 클럭으로 하여 모드를 설정한다. 하나의 서브모듈을 구성하는 32 개의 코어 모듈들은 <그림 5>와 같이 코어 모듈들의 모드 레지스터들이 직렬로 연결되어 있어 전역 제어부는 순차적으로 각 채널의 모드를 설정하게 된다.

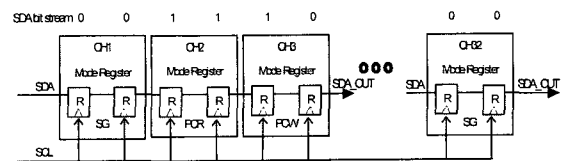


그림 5. 각 채널들에 대한 모드 레지스터 설정
Fig. 5. Setting of mode register in each channel.

시뮬레이션 동작을 수행하도록 하는 명령인 Run 명령은 각 코어 모듈들이 SCL의 상승에지에서 시작하여 SRAM 크기인 1M사이클 까지 시뮬레이션을 수행하도록 한다. Pc_inc_address와 pc_reset_address 명령들은 PC에서 각 채널에 대한 데이터들을 SRAM으로 부터 읽거나 쓸 경우 SRAM의 주소를 컨트롤 하기 위해서 사용되어진다.

해당 코어 모듈의 신호 발생기로서의 동작은 PC 호

스트에서 테스트회로에 인가될 신호 벡터들을 각 채널의 SRAM에 저장시킨후, SRAM의 데이터를 테스트 회로에 인가하게됨으로써 수행되어진다. PC 호스트로부터 데이터를 가져오기 위해서는 <그림 6>과 같이, 먼저 전역제어부가 Mode_set 명령으로 코어 모듈의 모드를 PCW 모드로 설정하고, Pc_reset_address 명령어에 의해 SRAM의 번지를 초기화 시킨다. 그리고 Pc_inc_address명령을 수행함으로써 초기화된 SRAM주소로부터 하나씩 증가하면서 순차적으로 SRAM에 8비트 데이터를 저장하게 된다.

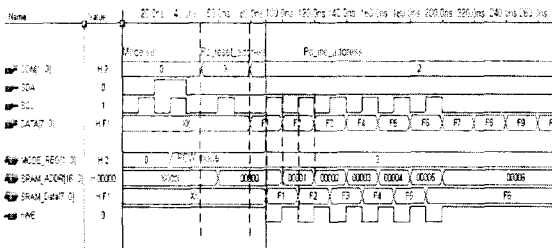


그림 6. 코어 모듈의 동작 타이밍도 (PCW 모드)
Fig. 6. Timing Diagram of Core Module's Operation (PCW mode).

SRAM에 테스트 벡터들이 저장되어진 후 전역 제어부는 <그림 7>과 같이 Mode_set 명령어에 의해 해당 코어모듈의 모드를 SG 모드로 설정하고, 어드레스를 초기화 한 후 시뮬레이션을 시작한다. SG 모드는 시뮬레이션을 시작하기 전에 쉬프트 레지스터에 1 바이트의 데이터가 미리 적재(1)되어 있어야 하며, 시뮬레이션이 시작되면 코어 모듈은 SRAM의 주소를 순차적으로 증가시키면서(2) SRAM의 데이터를 버퍼에 옮기고(3), 버퍼에 옮겨진 데이터를 다시 SHFT_REG에 저장(4)하여 출력하는 과정이 8 클럭 사이클 마다 반복하게

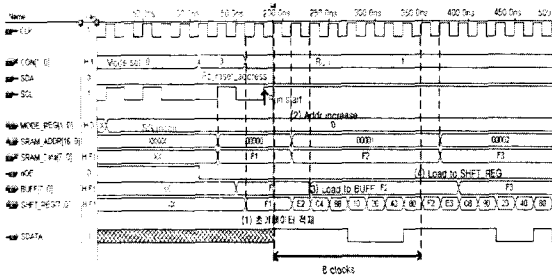


그림 7. 코어 모듈의 동작 타이밍도(SG모드)
Fig. 7. Timing Diagram of Core Module's Operation (SG mode).

된다.

코어 모듈의 로직분석기로써의 동작은 신호발생기와 반대로 테스트회로로부터 출력되는 파형을 샘플링하여 SRAM에 저장한 후, 저장된 데이터를 PC 호스트로 읽어들이므로써 동작한다. 신호 발생기와 동일한 방식으로 내부 모드를 LA 모드로 설정하고 SRAM 주소값을 초기화한 후 시뮬레이션을 시작한다. 시뮬레이션이 시작되면 코어 모듈은 <그림 8>과 같이 8 클럭 사이클 동안 데이터를 샘플링하여 버퍼에 저장(5)하고, SRAM의 주소를 증가(6)시킨 후, 버퍼의 데이터를 SRAM에 저장(7)한다.

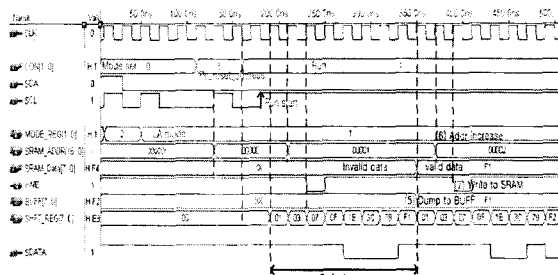


그림 8. 코어 모듈의 동작 타이밍도(LA모드)
Fig. 8. Timing Diagram of Core Module's Operation (LA mode).

시뮬레이션 수행 후 SRAM에 저장되어 있는 데이터를 호스트로 읽어오기 위해서는 <그림 9>와 같이 코어 모듈의 모드를 PCR로 설정(Mode set)하고, SRAM 주소 초기화(Pc_reset_address)한 다음, SRAM 주소를 증가(Pc_inc_address)시키면서 데이터를 PC호스트로 읽어 오게 된다.

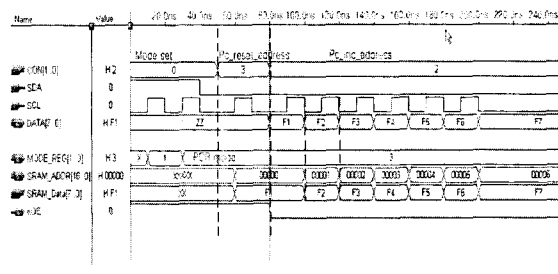


그림 9. 코어 모듈의 동작 타이밍도 (PCR모드)
Fig. 9. Timing Diagram of Core Module's Operation (PCR mode).

IV. 구현 및 비교

1. 시뮬레이터 구현

시뮬레이터의 각 채널에 해당되는 코어 모듈은 VHDL로 설계 되었으며, Altera Max-PlusII에서 제공하는 SRAM VHDL 모델을 포함하여 각 동작모드들에 대해 기능 검증 후 Altera CPLD인 EPM7064로 구현되었다^[9]. 구현된 코어 모듈은 EPM7064로직의 84%를 차지하는 약 1,200 게이트 로직을 포함하고, 37개의 입출력을 가진다. <그림 10>은 채널 메모리인 삼성 KM681002 SRAM과 함께 구현된 채널 보드를 나타낸다.

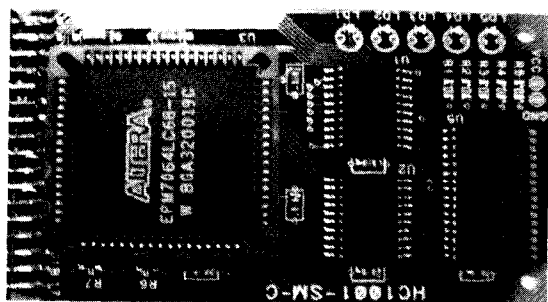


그림 10. 채널 보드
Fig. 10. Channel board.

채널 보드의 동작 검증은 두개의 채널 보드를 각각 신호 발생기와 로직분석기로 설정하고 보드들의 SDATA를 서로 연결하여, 신호발생기 채널에서 발생된 신호를 로직 분석기 채널에서 샘플링하여 각 두 채널의 메모리 내용을 소프트웨어상에서 확인해 봄으로써 각각의 채널 보드들을 검증하였다. 각 채널 보드들이 정상적으로 동작함을 확인한 후 <그림 11>과 같이 전체 시스템을 구성하였다. <그림 12>는 FLEX 10k100 Altera FPGA로 구현된 구속장 K=7의 비터비 디코더에 대해 50MHz에서 시뮬레이션 한 결과를 나타낸다. 구현된 하드웨어 시뮬레이터의 최대 동작 주파수는 50MHz 이고, 측정된 평균 전력소모는 55W이다.

하드웨어 시뮬레이터를 구동할 소프트웨어는 Win32 API기반으로 작성되었으며^[10], Waveform view, 블록설정, 시간영역에서 확대/축소, 입출력 노드를 추가, 삭제, 이동, 복사, 그룹핑 등의 GUI(Graphic User Interface)적인 기능과 시뮬레이션 run, 샘플링 주파수 설정, 입

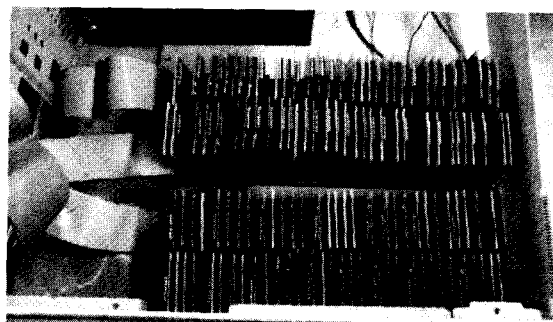


그림 11. 시뮬레이션 보드
Fig. 11. Simulation board.

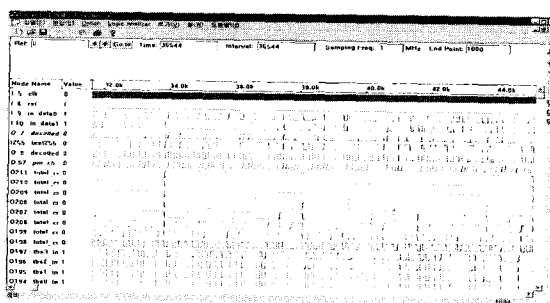


그림 12. 하드웨어 시뮬레이터의 검증
Fig. 12. Simulation result of the hardware simulator.

출력 노드의 모드 설정, 데이터의 업로드 및 다운로드 등의 시뮬레이터 하드웨어 제어 기능을 지원한다.

소프트웨어 동작은 다음과 같이 입력 설정, 시뮬레이션 수행, 출력 결과 디스플레이의 순으로 진행된다.

- ① 사용자는 테스트할 회로의 입출력 핀을 생성하고, 입출력 모드 및 테스트 벡터를 윈도우상에서 GUI (Graphic User Interface)를 통해 입력한다.
- ② 프로그램은 입력된 테스트 벡터를 파일에 저장함과 동시에 시뮬레이션 시스템의 각 채널에 다운로드 한다.
- ③ 사용자는 샘플링 주파수와 실행 시간 등을 설정한 후 시뮬레이터를 실행시킨다.
- ④ 시뮬레이션이 종결된 후에는 결과 값을 시뮬레이터의 각 채널로부터 PC로 업로드하여 파형을 보여준다.

2. 기존 시스템들과의 비교

본 논문에서 구현된 하드웨어 시뮬레이터는 신호발생기와 로직분석기의 기능을 동시에 수행하면서도 128 채널까지 편리하게 시뮬레이션 할 수 있으며, 비교적 낮은 동작주파수를 갖는 디지털 회로의 시뮬레이션에

적합한 저가형의 시스템이다.

최근에 개발된 Agilent사의 시뮬레이터 시스템(81200 Signal Generator/Logic Analyzer Platform)을 살펴보면, 이 시스템은 신호 발생기와 로직분석기의 기능을 함께 가지면서 초당 최대 2.7 기가비트 속도로 고속의 시뮬레이션이 가능하다. 그러나 이 시스템의 경우 채널에 해당하는 모듈이 로직분석기 또는 신호 발생기로 고정되어 있어 테스트할 회로의 입출력 형태와 수에 따라 해당 모듈을 추가해야 하고, 고가의 다채널 확장 슬롯에 채널 모듈을 부착하는 형태로 확장해야 한다.

Tecktronix사의 TLA721 시스템은 모듈별로 로직 분석기 또는 신호발생기로 전환 가능하고, 모듈당 64 채널을 사용하여 최대 680채널까지 지원하는 고속의 시스템이지만 샘플링 주파수와 샘플링 용량이 채널 수의 사용에 따라 저하된다.

이 두 시스템들은 자체에 운영체제가 내장된 독립형의 시스템인 반면 (주한백전자의 logic viewer는 시리얼 포트를 통하여 설계된 회로의 출력을 측정해볼 수 있는 저가형의 PC기반 시스템이다. MCU를 사용하여 최대 16채널에 대해서 초당 40 Mb 샘플링을 수행하며, 테스트 패턴을 발생 시키기 위해서는 추가적인 장비가 필요로 한다.

본 논문에서 구현된 하드웨어 시뮬레이터는 이들 시스템에 비해 샘플링 주파수가 낮은 반면 각 채널에 해당하는 코어 모듈이 독립적인 메모리와 내부 모드를 가지고서 하나의 컨트롤러처럼 동작하기 때문에 채널을 확장하더라도 샘플링 주파수와 샘플링 용량이 감소하지 않고, 물리적인 슬롯의 착탈없이 채널의 기능을 전환할 수 있는 장점을 지닌다. 또한 PC를 기반으로

구현되어 독립적인 운영체제를 가지지 않아 저가형 시스템 구현이 가능하다. <표 3>은 본 논문에서 구현된 하드웨어 시뮬레이터와 기존 상용 시스템들과의 성능 비교를 나타낸다.

V. 결 론

본 논문에서는 디지털 회로의 하드웨어 검증을 수행하기 위한 128채널 하드웨어 시뮬레이터 구현에 대하여 기술하였다. 시뮬레이터는 신호 발생기와 로직분석기 기능을 동시에 수행할 수 있으며, PC 호스트와의 인터페이스를 위해 PCI를 이용한다. 특히 각 채널 모듈은 신호 발생기와 로직 분석기를 전환하여 수행할 수 있도록 설계되어 입출력 형태에 관계없이 모든 채널을 효율적으로 사용할 수 있으며, 별도의 확장슬롯 없이도 확장이 가능하다. 또한 PC 기반으로 하기 때문에 많은 입출력에 대한 시뮬레이션 결과를 한 화면에 디스플레이 할 수 있는 편리함을 제공할 뿐 아니라, C 소스나 HDL과 같은 상위수준의 설계에서 사용했던 테스트 벡터를 최종 개발된 하드웨어에서 직접 사용하여 검증하기가 용이하다.

현재 구현된 시스템은 각 채널을 구성하는 코어 모듈 부분이 PLD로서 구현되어 있어 동작 주파수가 최대 50MHz이고, 각 채널 단위로 보드가 구현되어 있어 채널이 수가 많을 경우 시스템의 크기가 커져 이동성에 제한이 있다. 이와 같은 점을 보완하기 위해 코어 모듈을 ASIC화 하고, 여러 개의 채널을 한 장의 모듈에 집적하여 구현하는 연구를 진행 중에 있으며, 이는 시스템의 크기를 줄이고 동작 속도를 향상 시킬 수 있을 뿐 아니라 전력소비도 감소 할 수 있을 것으로 사료된다.

표 3. 기존 시뮬레이터들과의 비교
Table 3. Comparison of various simulators.

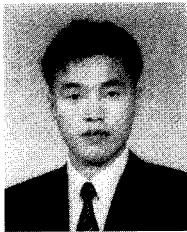
	구현된 시뮬레이터	한백전자-logic viewer[3]	Agilent-81200[4]	Tektronix-TLA721[5]
샘플링 주파수	50Mb/sec	40Mb/sec	2.7Gb/sec	2/1/0.5 Gb/sec (Quarter/half/full CHs)
샘플링 용량	1Mbits/CH	64Kbits/CH	8Mbits/CH	256/128/64 Mbits/CH (Quarter/half/full CHs)
OS 플랫폼	win98/2000 PCI Card	win95/98/me/2000 COM port	Windows2000 Stand alone	Windows2000 Stand-alone
채널 대역폭	128 CH	16 CH	Up to 128	Up to 680
확장성	Channel extension	No	Module extension	Module extension
기능	Signal Generator/Logic Analyzer	Logic Analyzer	Signal Generator/Logic Analyzer	Signal Generator/Logic Analyzer

참 고 문 헌

- [1] J. Babb, R. Tessier, M. Dahl, S. Hanono, D. Hoki, and A. Agarwal "Logic Emulation with Virtual Wires", IEEE Transactions on CAD of Integrated Circuits and Systems, Vol. 16(6) pp. 609- 626, June 1997.
- [2] 김남도, 양세양, "레고: 재구성 가능한 시스템 에뮬레이터", 대한전자공학회논문지, 제39권 SD편 제2호 pp. 91-103, 2002. 2
- [3] <http://www.hanback.co.kr/htmls/lgview.htm>

- [4] Agilent 81200 Data Generator/Analyzer Platform Technical Specification, Agilent Corporation, 2002.
- [5] TLA700 Family Datasheet, Tektronix Corporation, 2002.
- [6] PCI 버스 해설과 인터페이스 카드 설계, 국제 테크노정보연구소, 2001
- [7] PCI9050-1 DataSheet, PLX Technology Corporation, 1997.
- [8] ICD2053B Datasheet, Cypress Corporation, 1995.
- [9] MAX7000 DataSheet, Altera Corporation, 2002.
- [10] Walter Oney, Programming the Windows Driver Model, Microsoft Press, 1999.

저 자 소 개



丁 甲 天(正會員)

1996년 : 전남대학교 컴퓨터공학과 학사. 1998년 : 전남대학교 전자공학과 석사. 1999년~현재 : 전남대학교 고품질전기전자부품 및 시스템연구센터 연구원. 1998년~현재 : 전남대학교 전자공학과 박사

과정. <주관심분야 : 저전력 프로세서 구조, 영상압축, 영상통신용 ASIC 설계, DSP 설계, VLSI 설계 및 CAD 등.>



崔 鍾 鉉(學生會員)

2001년 : 전남대학교 컴퓨터공학과 학사. 2000년~2001년 : 전남대학교 반도체 설계교육센터 연구원. 2002년~현재 : 전남대학교 컴퓨터공학과 석사과정. <주관심분야 : VLSI 설계 및 CAD, 임베디드 시스템,

영상압축 등.>



朴 性 模(正會員)

1977년 : 서울대학교 전자공학과 학사. 1979년 : 한국과학기술원 전기 및 전자공학과 석사. 1988년 : 노스캐롤라이나 주립대학 전기 및 컴퓨터공학과 공학박사. 1979년~1984년 : 한국전자기술연구소 설계개발

부 선임연구원. 1988년~1992년 : 올드도미니언 대학교 전기 및 컴퓨터공학과 조교수. 1992년~현재 : 전남대학교 컴퓨터공학과교수. 2002년~현재 : 전남대학교 정보전산원 원장. <주관심분야 : 멀티미디어 프로세서 구조, VLSI 시스템 설계, 신호처리용 ASIC 설계, 영상압축, 임베디드 시스템 등.>