

論文2002-39SD-2-11

# 레고 : 재구성 가능한 시스템 에뮬레이터

## (REGO: REconfiGurable system emulatOr)

金 南 道 \* , 梁 世 陽 \*

(Nam Do Kim and Sei Yang Yang)

### 요 약

다수의 FPGA로 구성된 에뮬레이터에서 FPGA간의 연결구조와 신호의 전송방법은 에뮬레이터의 확장성과 검증속도를 결정하는 중요한 요소이다. 기존의 에뮬레이터는 검증 대상이 되는 회로의 크기가 커짐에 비례하여 에뮬레이션의 속도가 현저하게 느려지는 문제점이 있다. 본 논문에서는 자원이용률을 극대화할 수 있을 뿐만 아니라 에뮬레이션의 속도도 크게 높일 수 있는 새로운 에뮬레이터 구조를 제안한다. 제안되는 에뮬레이터는 계층적인 환형 토폴로지 구조를 가지고 파이프라인의 환형으로 FPGA들을 연결하여 FPGA의 핀한계를 극복하고, 이와 같은 연결구조를 이용하여 다양한 IP들의 통합도 매우 용이하게 함으로써 설계검증 난이도가 설계자의 검증 능력을 쉽게 초과할 수 있는 시스템 수준에서의 검증도 최소한 수십 MHz 속도의 에뮬레이션으로 효과적으로 가능하게 한다.

### Abstract

For massive FPGA based emulator, the interconnection architecture and the transmission method of signals between FPGA's are important elements which decide speed of emulation and extendability of emulator. Existing FPGA-based emulation system is faced the problems of which the emulation speed getting slow drastically as the complexity of circuit increases. In this paper, we proposed a new innovative emulation architecture that has high resource usage rate and makes the fast emulation possible. The emulator with very unique hierarchical ring topology presented here has merits to overcome FPGA pin limitation by connecting each FPGA into a set of pipelined rings, and it also makes emulation speed at the tens of MHz at least even at system level where the verification complexity can easily exceed the verification capability of designers.

### I. 서 론

설계기술의 비약적인 발전에 따라 집적 회로의 복잡

\* 正會員, 釜山大學校 컴퓨터工學科  
(Dept. of Computer Engineering, Pusan National University)

※ 본 논문은 부산대학교 컴퓨터 및 정보통신 연구소  
부설 휴대형 멀티미디어 단말기 연구센터 및 반도  
체설계교육지역센터(IDEC)의 부분지원에 의한 연구  
결과입니다.

接受日字: 2001年8月24日, 수정완료일: 2002年1月18日

도는 급속도로 증가하고 있으며, 특히 사용자회로뿐만 아니라 메모리, 프로세서 코어 및 DSP(Digital Signal Processor) 코어와 같은 다양한 모듈들이 하나의 칩 내부에 집적된 시스템급 회로디자인의 설계 비중이 커지고 있는 실정에서, 이의 검증(verification)을 위한 시간과 비용의 증가는 기하급수적으로 커져가고 있다. 이같이 설계회로의 복잡도가 지속적으로 증가하고 있는 실정에서 시뮬레이션방법의 검증은 과도한 검증시간을 요구하여 제품개발사이클이 매우 짧은 현 상황에 적합하지 않다<sup>[1]</sup>. 뿐만 아니라 검증을 위해 IP와 타깃 환경을 소프트웨어적으로 모델링하는 과정에서 실제의 하드웨어 상황에서 발생하는 여러 가지 상황을 완벽하게

고려하는 것은 매우 어려우며, 이로 인하여 검증의 신뢰성이 떨어지게 된다. 위와 같은 시뮬레이션 방식의 검증시스템의 단점을 극복하기 위해, 빠른 검증속도를 제공하고 실제 하드웨어 상황에 가장 비슷한 H/W기반의 검증방식인 에뮬레이션<sup>[2~7]</sup>이 최근 들어 많이 사용되고 있다. H/W기반의 검증방식인 에뮬레이터는 다양한 응용설계에 적용할 수 있어야 하며, 이를 위해 일반적인 에뮬레이터는 재사용이 가능한 다수의 현장프로그래밍가능소자(Field Programmable Gate Array: 이하 FPGA)로 구성되어 있다.

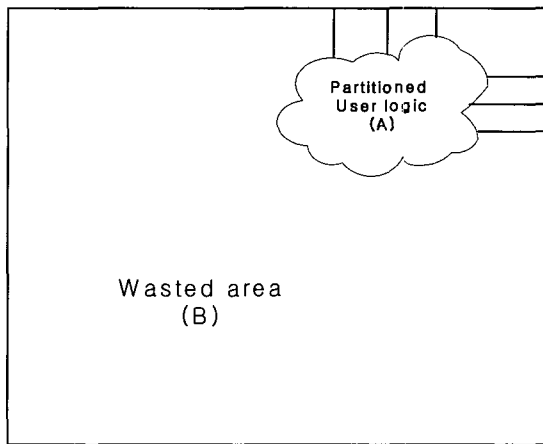


그림 1. 핀 한계를 가지는 FPGA기반의 에뮬레이션 시스템의 일반적인 FPGA사용  
 Fig. 1. FPGA usage of FPGA-based emulation system having pin limitation problem.

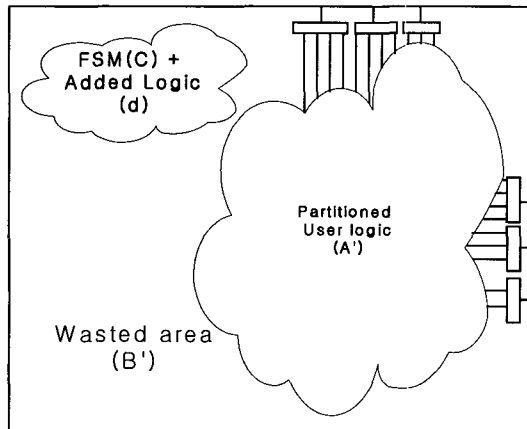


그림 2. 본 논문에서 제안한 에뮬레이션 시스템의 FPGA사용  
 Fig. 2. Expected usage of proposed emulation system.

다수의 FPGA로 구성된 에뮬레이터에서 검증대상회로는 에뮬레이터에 존재하는 다수의 FPGA들에 기존의 함수적 행위를 만족하면서 분할되어야한다. 검증대상회로의 분할시 분할된 회로는 FPGA의 핀 수와 면적을 초과하지 않도록 분할되어야 하며 대부분의 에뮬레이터는 핀 수와 면적의 불균형, 즉 가용한 FPGA의 면적을 만족하기 전에 핀 수의 제한으로 FPGA의 많은 부분을 사용하지 못하게 되는데, 이를 에뮬레이터의 핀 한정 문제(pin limitation problem)라고 한다<sup>[2,8]</sup>. 패키지 기술의 제약으로 인해 FPGA의 핀 수는 면적의 증가를 따라가지 못하는 실정이며 이는 핀 한정 문제를 수반하게 만드는 요인으로 작용한다. FPGA의 비효율적인 사용은 과도한 FPGA의 사용으로 이어지게 되며 이는 전체 에뮬레이터의 제작비용을 증가하게 만드는 결과를 초래한다. 따라서 핀 한정 문제를 극복하지 못한 FPGA기반의 에뮬레이터는 칩간의 통신대역 문제로 인하여 그림 1과 같이 FPGA내의 아주 적은 부분만을 사용하게 되며 나머지 영역은 사용하지 못하므로 FPGA 자원을 낭비하게 되고 분할된 로직 디자인을 검증하기 위해서 사용되는 에뮬레이터의 FPGA의 수가 증가하게 되며, 이는 신호가 전달되기 위해 거쳐가는 FPGA의 경계 수를 늘임으로써 전체 검증속도를 떨어뜨리는 요인으로 작용하게 된다. 또한 에뮬레이터에 존재하는 FPGA들의 상호연결형태와 신호의 전달방식은 에뮬레이터의 성능과 확장성을 결정하는 중요한 요소가 된다<sup>[1]</sup>. 따라서 에뮬레이터에 존재하는 FPGA들의 이상적인 연결구조는 검증대상시스템의 사이즈의 증가시 에뮬레이터의 확장이 요구될 때 확장이 용이한 구조를 가지고 있어야 하고 확장시 에뮬레이터의 성능이 현격히 저하되지 않아야 하며 전체 에뮬레이터의 통신용량(communication bandwidth)을 줄일 수 있는 구조를 가져야 한다. 또한 FPGA 간의 연결방식에 따라 에뮬레이터의 제작비용은 현격한 차이를 보이므로 이 또한 고려해야 할 사항이다. 본 논문에서는 기존의 FPGA기반의 에뮬레이터가 가지는 핀 한정 문제를 극복할 수 있으며, 에뮬레이터의 축소와 확대가 자유로운 새로운 계층적인 환형(ring)구조의 연결 및 통신 방법을 제안하고, 이의 유용성을 실험을 통하여 보인다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 에뮬레이터의 하드웨어 구성방식과 핀 한정 문제를 살펴보고, III장에서는 본 논문에서 제안하는 에뮬레이터의 구조와 동작방식 그리고 에뮬레이터에 적용되는 소

소프트웨어의 구성과 연결흐름을 설명한다. IV장에서는 본 논문에서 제안한 방법을 이용한 실험을 통하여 제안된 시스템의 유용성을 보인다. 그리고 마지막 V장에서는 본 논문이 기여하는 바와 향후 연구과제를 간략하게 기술한다.

## II. 관련 연구들

### 1. 에플레이터 구조

FPGA를 사용하는 에플레이터들은 수십에서 수백 개까지의 매우 많은 수의 FPGA로 구성되어 있으며 이들의 연결구조는 에플레이터의 성능과 가격, 그리고 확장성을 결정하는 중요한 요소이다<sup>[1]</sup>.

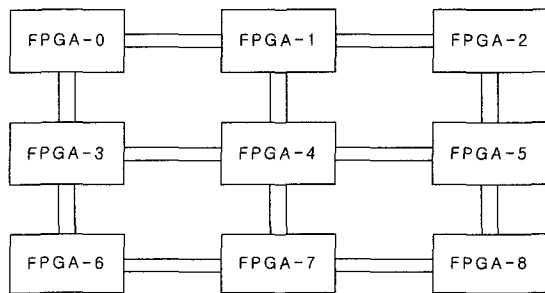


그림 3. 그물 연결도  
Fig. 3. Mesh topology.

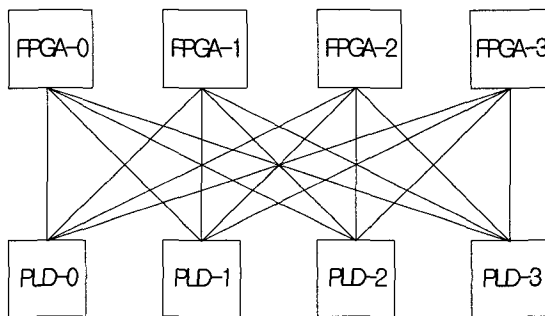


그림 4. 크로스-바 연결도  
Fig. 4. Cross-bar topology

따라서 에플레이터의 연결구조는 전체 시스템의 통신용량(communication bandwidth)을 줄일 수 있는 구조를 가지고 있어야한다. 즉, 어떤 FPGA의 신호선이 대상 FPGA에 도달하기 위해 거쳐야 하는 FPGA의 수(이를 hop이라 함)가 최소한으로 될 수 있도록 설계되

어야 한다. 또한 검증대상시스템의 크기변화에 따라 에플레이터의 확장이 요구될 때 확장이 용이한 구조가 될 수 있도록 고려되어야한다. 기존의 에플레이터의 가장 일반적인 연결구조는 그림 3의 인접한 모든 FPGA에 연결선을 가진 그물(mesh)구조<sup>[9,11]</sup>와 그림 4의 크로스바(crossbar)구조이다<sup>[3,4,6]</sup>. 그림 3의 그물 구조는 FPGA간의 거리가 가변적(FPGA-0과 FPGA-4의 거리는 2이지만 FPGA-0과 FPGA-1, 그리고 FPGA-3의 거리는 1)이어서 라우팅지연시간을 예측하기 어려운 반면에<sup>[1]</sup> 단순 확장성이 좋으며 에플레이터 자체내의 전체 연결수가 적어지고 결과적으로 PCB 자체의 연결선 수가 적어짐으로 낮은 가격으로 제작이 가능한 구조이다. 그리고 그림 4의 크로스바 형태의 연결구조를 가지는 에플레이터의 경우 FPGA 간의 직접연결은 없으며 FPGA간의 통신은 연결전용 FPGA나 ASIC을 통해서만 이루어진다. 이는 그물 구조의 에플레이터에서 발생할 수 있는 신호선 통신의 가변적 거리로 인한 타이밍 문제점을 어느 정도 보완할 수 있다. 그러나 FPGA 간의 통신거리를 줄이기 위해 별도의 연결용 FPGA들 혹은 연결전용 ASIC 칩들이 필요하게 되고 이는 PCB상에 아주 많은 수의 연결을 초래하여 결과적으로 여러층의 PCB 레이어가 요구된다. 이는 에플레이터의 가격을 높지게 되는 중요한 요인으로 작용한다. SoC(System-on-a-Chip)와 같은 초고집적시스템을 에플레이션하기 위해서는 우선적으로 에플레이터가 에플레이션 대상이 되는 시스템의 크기에 비례하여 매우 용이하게 확장될 수 있어야만 한다. 이와 같은 측면에서 그물 구조는 단순 확장성을 제공하지만 라우팅 지연시간의 예측을 어렵게 하고 크로스바 구조는 확장성을 떨어뜨리는 대신에 라우팅 지연시간의 예측을 가능케 한다<sup>[1]</sup>. 하지만 시스템의 대량 확장시에 그물 구조에서의 단순 확장성은 라우팅 지연시간 예측의 어려움을 더욱더 심화시키며, 크로스바 구조에서의 확장성을 보완한 계층적 부분 크로스바구조<sup>[6]</sup>는 크로스바를 계층적으로 구성하여 일관된 라우팅 지연시간의 예측을 제공한다. 그러나 확장성과 같이 고려되어야 하는 중요한 사항인 에플레이션의 속도에서는 지금까지 제안된 그물 구조나 부분 크로스바 구조들은 에플레이션 용량이 커지는 것에 반비례하여 에플레이션의 속도가 크게 떨어지게 된다는 문제점이 있다. 따라서 본 논문에서는 계층적 링 연결(hierarchical ring interconnection) 구조를 이용한 파이프라인 방식의 신호선 연결법을 채택함으로써

에플레이션 용량과 에플레이션 속도와의 이 같은 상관 관계를 거의 해소시킬 수 있다.

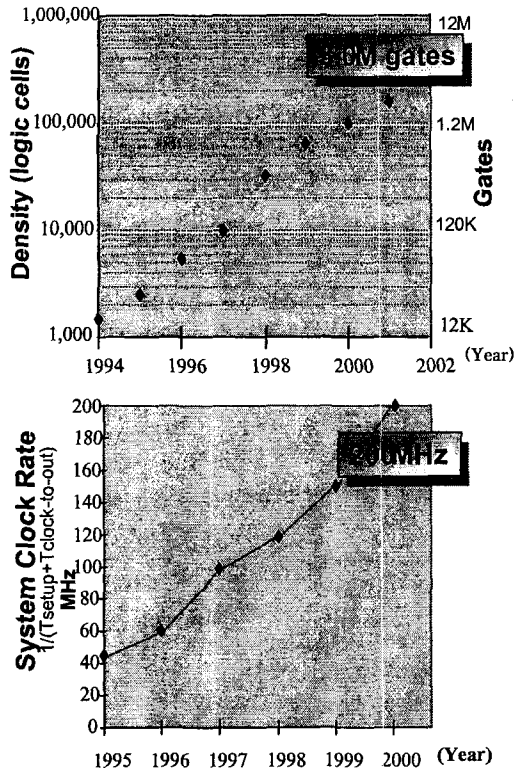


그림 5. Xilinx FPGA 집적도 & 성능 로드맵  
Fig. 5. Xilinx FPGA density & performance roadmap [Source: Xilinx, Inc.]

2. 에플레이션에서의 핀 제약 문제

다수의 FPGA 칩에 기존의 회로를 분할하여 각각의 FPGA에 배치할 때, 분할된 각 모듈은 대상 FPGA의 용량보다 반드시 작게 할당해야 하고 분할된 각 모듈은 FPGA의 통신대역(핀 수)을 초과하지 않으면서 각 FPGA에 배치되어야만 한다. VLSI기술의 비약적인 발전으로 18달마다 2배의 트랜지스터 집적도의 증가를 예상한 무어의 법칙<sup>[10]</sup>에서 예상한 것대로 그림 5와 같이 FPGA의 집적도가 증가하고 동작속도가 빨라지고 있다. 그러나 상용 FPGA 집적도가 급속도로 증가하는데 비해 핀 수의 증가는 패키지 기술의 제약으로 인해 FPGA 집적도의 증가에 비례하여 증가하지 못하는 실정이며 핀 수를 고려하지 않고 FPGA의 면적에만 맞추어 분할된 회로의 핀 수는 FPGA의 사용가능 핀 수를 쉽게 초과하게된다. 이는 이미 잘 알려진 렌트법칙<sup>[11,12]</sup>에 의해 설명되어지며 이에 대한 다양한 실험이 이루어

어져 왔다<sup>[2]</sup>. 따라서 FPGA내의 게이트 사용빈도를 넘기 전에 FPGA 핀 제한으로 인해 대부분의 FPGA 로직자원을 사용하지 못하게 되는 사용 자원 불균형을 받게된다. 이는 II.1의 에플레이터 구조문제와도 관련된다. 하나의 시그널이 한 FPGA에서 대상 FPGA로 전달되어야 할 때 거쳐야 할 FPGA수가 많은 경우 시그널의 전달을 위해 더 많은 핀 수가 필요하게 되며 분할된 회로에서 FPGA간의 전달 경로가 긴 신호선이 많으면 이는 핀 제약 문제를 더 심화시키게 된다. FPGA간의 거리가 가변적인 그물구조와, 완전 크로스바보다 FPGA간의 거리의 가변이 큰 부분 크로스바에서의 핀 제약 문제는 여전히 FPGA의 사용률을 떨어뜨리는 중요한 요인으로 작용한다. 그리고 신호의 전송이 여러 FPGA를 거쳐야 하는 경우에는 이의 정확한 동작을 위해서 클럭 사이클을 예측해야 하는데, FPGA내부의 신호전달시간보다 칩간의 신호전달시간은 예측하기 어렵고, 이 때문에 최대신호전달시간을 대략적으로 예측해야하므로 전체 클럭사이클이 실제 필요한 클럭사이클보다 더 늘어날 수 있다. 따라서 하나의 물리적 핀에 하나의 로직 신호선을 할당하는 모든 에플레이터의 경우 핀 제약 문제로 인해 많은 수의 FPGA를 필요로 한다. 이는 에플레이터에서 신호선의 전달경로를 늘려 전체 검증 속도가 느려지게 되는 주요 요소로 작용하게 된다. 이를 방지하기 위해 하나의 실제 핀에 다수의 로직 신호선을 할당하는 신호선의 시분할다중화(time division multiplexing)기반의 에플레이터에 관한 연구들이 활발히 진행되었다<sup>[2-5,13,14]</sup>. 그러나 초기의 시분할다중화기반의 에플레이터는 느린 동작속도와 비동기회로의 모델링의 어려움 때문에 실제 현장에서 널리 사용되지 못하였다<sup>[3]</sup>. 최근에는 위의 문제점들을 보완한 시분할다중화기반의 에플레이터들<sup>[2,3]</sup>이 설계현장에서 사용되고 있는 실정이지만 이들 또한 시스템 확장이 용이하지 않고 검증속도가 느려지는 구조적인 문제를 안고 있다. 따라서 핀 제약 문제를 효율적으로 극복하고 검증대상의 크기에 따라 확장이 용이한 에플레이터가 요구된다.

III. REGO: 재구성가능 시스템에플레이터

기존의 핀 한계 문제를 해결하기 위해서는 FPGA의 핀 수를 렌트법칙에 맞추어 늘여야 한다. 그러나 이는 패키지 기술의 제약과 필요한 FPGA 수의 과도한 증가

로 에뮬레이터 가격의 급격한 상승을 초래하게 된다. 따라서 제한된 핀 수에 그 이상의 신호선을 할당하는 시분할다중화 방식을 적절하게 이용해야만 한다. 예를 들어 VirtualWire<sup>[2,13]</sup>는 하나의 실제 핀에 다수의 신호선을 할당하고 이를 시분할다중화 방식으로 신호를 전달할 수 있는 구조를 제안하였다. 본 논문에서는 II장에서 기술한 문제점들을 해결하기 위하여 환형 연결구조를 갖는 에뮬레이터 REGO(REconfigurable emulatOr)를 새롭게 제안한다. 즉 신호선들의 연결을 환형 토폴로지로 구성하고 파이프라인 방식으로 하나의 핀에 여러 신호를 시분할다중화하여 전달함으로써 핀 제한 문제를 극복하여 FPGA사용률을 높이고 에뮬레이터를 구성하는 요소인 모듈, 모듈의 집합으로 구성된 박스 그리고 박스의 집합으로 구성된 캐비닛을 계층적 구조의 링 토폴로지로 구성하여 에뮬레이터의 확장을 용이하게 하는 동시에 시스템의 확장시 발생하는 에뮬레이터의 속도저하를 효과적으로 제거할 수 있다.

1. FPGA 내부 구조

분할된 검증회로의 다수의 로직 신호선을 하나의 실제 핀에 적절한 시간대에 할당하기 위하여 각 FPGA에는 그림 6에서와 같이 전체 검증대상회로에서 분할된 사용자 회로뿐만 아니라 환형으로 연결된 실제 핀을 통해 적절한 시간대에 로직 신호선을 할당하여 파이프라인 방식으로 전달하기 위한 부가회로들이 존재한다.

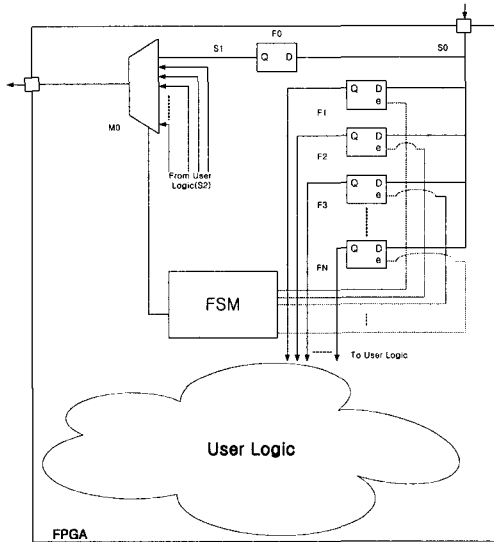


그림 6. REGO 에뮬레이션 시스템의 FPGA 내부 상세도  
Fig. 6. FPGA inside view of REGO emulation system.

분할기에 의해 분할된 사용자회로의 일부분과 환형 연결을 구성하는 두 핀 사이에 플립플롭 FO가 있고 환형 연결을 통해 들어온 신호 S0를 FPGA내부의 회로에 인가하기 위해 이를 저장하기 위한 플립플롭의 어레이 F1~FN이 있으며, 환형 연결을 통해 다른 FPGA로 전달될 내부회로의 신호선들 S2와 외부에서 환형을 통해 인가된 신호가 플립플롭에 인가되고 이의 출력 S1가 멀티플렉서 M0에 연결되어 있다.

FPGA 내부에 존재하는 모든 사용자 회로와 유한상태기, 그리고 모든 부가회로들은 에뮬레이션 클록보다 훨씬 빠른 시스템 클록에 의해 동기화된다. 이들의 동작을 제어하는 유한상태기는 초기회로의 분할된 결과로 만들어진 DDG와 에뮬레이터의 구성형태를 받아들여 이를 스케줄링한 결과에 따라 에뮬레이션을 위한 검증회로 컴파일시 자동적으로 생성되어 사용자 회로와 결합되어 동작하게 된다.

2. FPGA간 통신

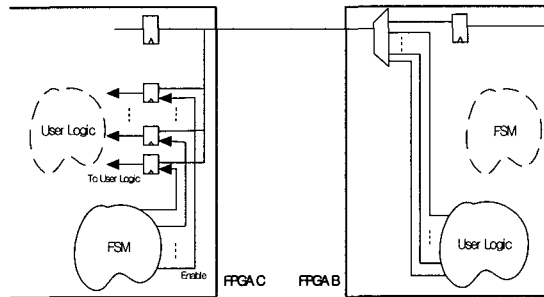


그림 7. FPGA간의 통신구조  
Fig. 7. Communication architecture of inter-FPGA.

두 FPGA간의 통신 구조는 그림 7과 같다. FPGA B에서 전달될 신호선들은 멀티플렉서의 입력단에 연결되고 이 신호들은 미리 스케줄링된 순서대로 내장된 유한상태기에 의해서 정해진 시간에 멀티플렉서의 출력단을 통해 환형을 돌다가 대상 FPGA에 도착하면 대상 FPGA의 유한상태기의 enable 신호를 받은 플립플롭에 의해 사용자 회로에 전달된다. 즉 전달된 신호의 대상 FPGA가 FPGA C라면 이는 유한상태기의 enable 신호에 의해 제어되는 플립플롭에 저장되어 사용자 로직에 인가되고, 만일 전달된 신호의 대상 FPGA가 FPGA C가 아닌 다른 FPGA라면 이는 FPGA C의 멀티플렉서 출력단을 통해 구성된 환형 연결을 거쳐 나가게 된다. 모든 신호의 전달순서는 미리 내장된 스케

줄러에 의해 스케줄링된 순서에 따라 순차적으로 보내 지게 되며 이를 통하여 하나의 핀에 다수의 신호선을 할당할 수 있게 된다.

3. 상위 구성요소의 구조와 신호전송

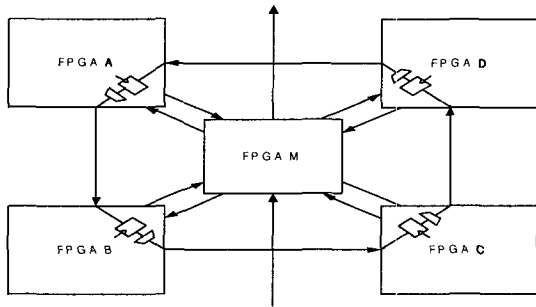


그림 8. 모듈에서의 환형구조연결도  
Fig. 8. Ring topology connections in the module.

계층적인 환형 구조를 갖는 REGO에서 하나의 모듈의 대략적인 구조는 그림 8과 같다. 그림 8에서와 같이 각 FPGA(A, B, C, D)의 핀 수는 4라고 할 때 이중 2개의 핀을 이용하여 모듈내의 FPGA간에 환형을 형성하고 나머지 2개의 핀은 다른 모듈과의 신호전달을 위해 FPGA M에 연결된다. 모듈내의FPGA간의 신호전달은 구성된 환형을 통해 이루어진다. 즉 모듈내의 FPGA, 모듈에서 모듈, 박스에서 박스로의 같은 레벨에서의 신호전달은 구성된 환형 구조를 통해 이루어지며 모듈에서 박스, 박스에서 캐비닛으로의 타 레벨간 신호전달은 상하의 연결포트와 환형구조를 동시에 이용하게 된다.

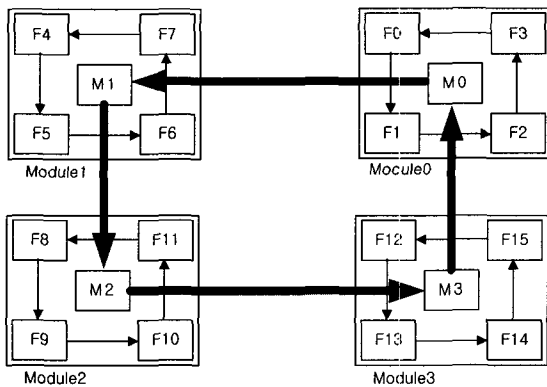


그림 9. Box내의 통신채널구성  
Fig. 9. Communication channel in a box.

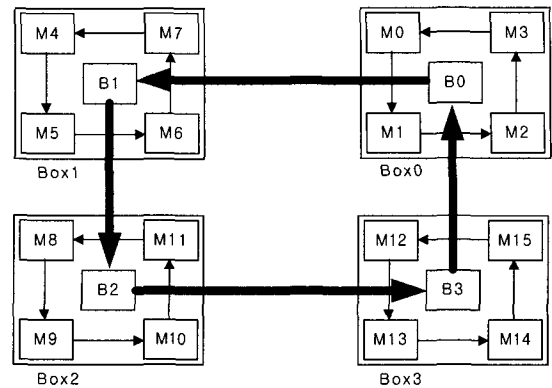


그림 10. Cabinet내의 통신채널구성  
Fig. 10. Communication channel in a cabinet.

그림 9는 4개의 모듈로 구성된 박스의 통신채널 구성도를 나타내고, 그림 10은 4개의 박스로 구성된 캐비닛의 통신채널구성을 나타내고 있다. 박스내의 각 모듈은 타 모듈 또는 박스간의 통신을 위해 모듈 FPGAs(M0, M1, M2, M3)를 가지고 있으며 이들은 같은 모듈내의 사용자 로직을 포함하는 사용자 FPGAs(0, 1, 2, 3, 4, ..., 14)와의 연결을 가지고 있다. 모듈 FPGA의 구성요소는 사용자 FPGA와 마찬가지로 환형 구성을 위한 부가회로(플립플롭과 멀티플렉서)와 이를 제어하는 유한상태기로 이루어지며 사용자 회로는 포함되어있지 않다. FPGA간의 신호전달이 환형을 통해 이루어지는 것처럼 모듈간의 통신도 환형 토폴로지(ring topology)로 이루어진다. 만일 그림 9의 모듈 0(Module 0)의 FPGA0(F0)에서 모듈 2(Module 2)의 FPGA11(F11)로의 신호전달이 필요할 시에는 아래와 같은 신호전달순서로 전송된다.

$$F0 \rightarrow M0 \rightarrow M1 \rightarrow M2 \rightarrow F11$$

그림 10은 REGO의 가장 상위 구성요소인 캐비닛의 구성도이며, 4개의 박스를 환형으로 연결한 형태로 구성되어 있다. 캐비닛에서 환형 토폴리지를 이용한 신호의 전달 방식은 위에서 설명한 박스와 모듈의 동작방식과 같다. 이와 같이, 계층적 구조를 가지는 REGO의 모든 구성요소들은 일관된 환형 구조와 상하의 연결포트를 가지고 있으며, 이를 통해 신호의 전달이 이루어짐으로써 검증대상시스템의 회로의 크기의 가변으로 인해 에뮬레이터 크기의 변화가 있을 시에도 타 에뮬레이터의 연결구조에 비해 쉽게 시스템의 크기를 변화시킬 수 있으며, 시스템의 확장시 에뮬레이터의 검증속

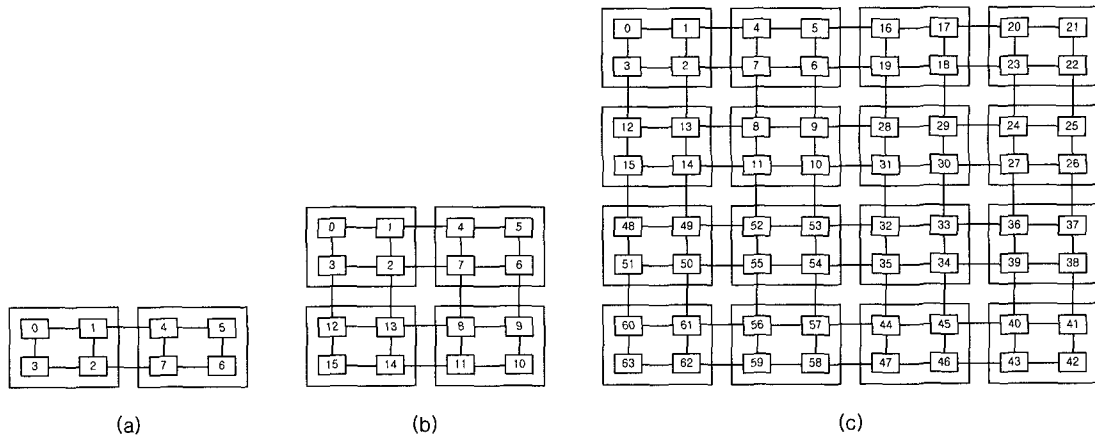


그림 11. 그물 구조에서의 에뮬레이터 시스템 확장  
 Fig. 11. Expansion of emulation system in mesh topology.

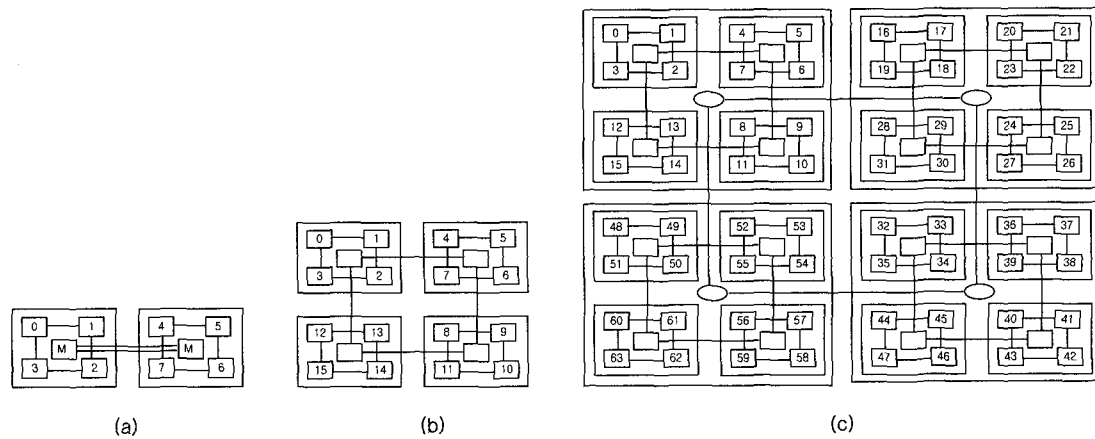


그림 12. REGO에서의 에뮬레이터 시스템 확장  
 Fig. 12. Expansion of emulation system in REGO.

도가 크게 저하되는 것을 막을 수 있다.

4. REGO의 확장성

검증대상회로의 사이즈 증가시 수반되는 에뮬레이터의 확장은 에뮬레이터의 성능과 비용면에서 아주 중요하다. 본 논문에서 제시하는 REGO는 계층적인 환형연결구조를 이용하여 에뮬레이터의 확장시에도 FPGA간의 통신대역이 급격히 떨어져 검증속도를 저하되는 문제점을 개선할 수 있다.

그림 11과 그림 12는 각각 그물구조와 본 논문에서 제안하는 REGO에서 에뮬레이터 확장의 예이다. 그림 11(a)는 최초 2개의 모듈로 구성되어 있는 그물 구조의 에뮬레이터이며 그림 11(b)와 그림 11(c)는 그림 11(a)가 확장된 구조이다. 그림 11(a)의 FPGA 0에서 모든 FPGA (1~7)까지의 최단신호라우팅의 총 합은 16 이

며 그림 11(b)의 FPGA 0에서 모든 FPGA(1~15)까지의 최단신호라우팅의 총합은 48, 그리고 그림 11(c)의 FPGA 0에서 모든 FPGA(1~63)까지의 최단신호라우팅의 총합은 504이다. 이때 그림 11(a)에서 FPGA간 최대 라우팅 거리는 4이고 그림 11(b)와 그림 11(c)의 최대 라우팅 거리는 각각 6, 14이다. 그림 12(a)는 최초 모듈의 수가 2인 REGO 에뮬레이터이다. 그림 12(b)와 그림 12(c)는 최초 모듈의 수가 2인 REGO에뮬레이터의 확장된 구조이다. 그림 12(a)의 경우 최단신호라우팅 총합은 그림12(a), 그림12(b), 그림 12(c)의 경우 각각 18, 56, 252이며 FPGA간 최대라우팅거리리는 각각 3, 5, 7이다. FPGA간 신호라우팅의 거리의 증가는 에뮬레이터의 통신대역을 감소시키며 이는 에뮬레이션 검증속도에 나쁜 영향을 미칠 수 있다.

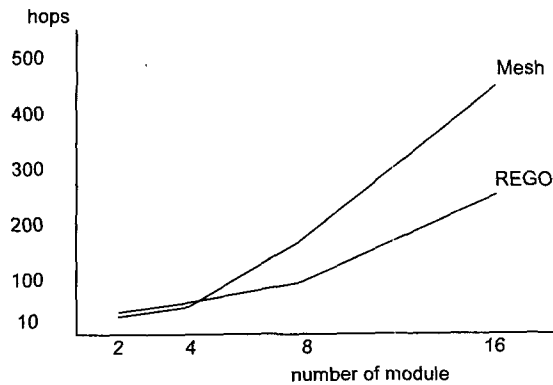


그림 13. 시스템 확장시 총 라우팅거리 비교  
Fig. 13. Comparison of total routing length in case of system expansion.

그림 13에서와 같이 Mesh의 경우 에뮬레이터의 크기 증가시 신호라우팅에 걸리는 hop수가 급격한 증가를 하고 있으며 최대 라우팅거리 또한 본 논문에서 제시 하는 구조보다 급격히 증가하고 있다. 따라서 그물 구조의 에뮬레이터는 에뮬레이터의 확장에 따라 에뮬레이터의 검증속도가 급격히 떨어질 수 있는데 반해 본 논문에서 제안하는 계층적 환형 구조의 시스템은 에뮬레이터의 확장시에도 완만하게 검증속도가 떨어져 기존의 문제점을 제거할 수 있다.

5. 에뮬레이션 클록

하나의 물리적 핀에 하나의 로직 신호선을 할당하는 방식의 FPGA기반의 에뮬레이터의 경우 대부분의 FPGA 면적을 사용하지 못함으로 인해 많은 수의 FPGA를 필요로 한다. 이는 신호선이 거쳐야 하는 FPGA의 수를 늘려 전체 검증 속도가 느려지게 되는 주요 요소로 작용하게 된다. 본 논문에서는 FPGA 간의 신호전달을 계층적인 환형 연결을 통해 파이프라인 방식으로 전달하여 핀 제한 문제를 극복하였다. 파이프라인 방식으로 신호를 전송하기 위하여 하나의 에뮬레이션 클록은 다수의 클록 집합으로 이루어져 있으며, 빠른 검증속도를 위하여 그림 14에서와 같이 고속의 시스템 클록을 사용하였다. 시분할다중화를 이용하는 에뮬레이터의 경우, 적용하는 에뮬레이션 클록은 [2]에서와 같이 여러 개의 단계(phase)로 나누어지며 각 단계는 전송시간(communication)과 평가기간(evaluation)으로 구성된다. 시스템 클록의 사이클은 FPGA내 최장 조합회로의 지연시간보다는 상대적으로 매우 빠른 FPGA 간의 최장 라우팅 지연시간의 크기로 한다.

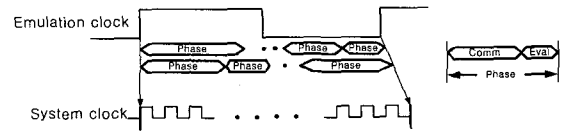


그림 14. 시분할 다중화 방식의 에뮬레이션 클록의 구성  
Fig. 14. Emulation clocking scheme which based on Time-Division-Multiple.

전송시간을 구성하는 시스템 클록의 수는 한 FPGA의 내부회로를 평가하기 위해 필요한 모든 신호선이 도달하기까지 필요한 시스템 클록의 수이며, 평가기간은 한 FPGA에 도달할 신호들이 안정적인 상태로 사용자로직에 입력되어 평가되는 FPGA내의 최장조합경로의 지연시간을 보장하는 시스템 클록의 수에 의해 정해진다. 에뮬레이션 클록의 길이는 분할된 회로의 최장 조합경로와 에뮬레이터의 FPGA 간 평균 라우팅거리에 의해서 주로 결정된다<sup>[4]</sup>.

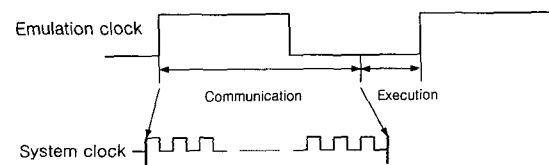


그림 15. 에뮬레이션클록의 구성  
Fig. 15. Emulation clocking scheme.

만일 분할된 회로의 모든 조합경로가 거치는 분할회로의 수가 1이 된다면 에뮬레이션클록은 전송시간과 실행시간의 하나의 단계로 구성될 수 있으며 이때 그림 15와 같은 에뮬레이션 클록으로 구성된다. 하지만 기존의 에뮬레이터에서는<sup>[2,3]</sup> 회로의 분할시 적은 연결 선을 가지고 균등하게 분할하는 기존의 분할기를 그대로 사용하였다. 따라서 기존의 고려사항을 가지고 분할하는 분할기를 사용한다면, 시분할다중화를 사용하여 핀 제한문제를 제거한다하더라도 빠른 검증속도를 기대하기 어렵다. 본 논문에서는 검증대상회로의 분할시 조합회로가 거치는 FPGA의 수를 최소로 하여 최종 에뮬레이션클록을 최소한으로 할 수 있도록 검증대상 회로를 분할하였다. 최종 에뮬레이션 클록의 길이를 결정하는데는 여러 가지 요소가 복합적으로 관련되어 있다. 검증회로를 분할하는 문제, 실제 에뮬레이터의 각 FPGA에 배치하는 문제와 회로의 종속성 분석



(dependency analysis)을 통하여 나온 그래프를 입력으로 받아들이어 신호선의 효율적인 스케줄링을 하는 단계가 그것이다. 위의 과정을 어떻게 하느냐에 따라 전체 에블레이션 클록의 길이는 정해지고, 이는 에블레이터의 성능에 큰 영향을 미친다.

6. 시스템 클록의 공급

본 논문에서 제시하는 REGO는 에블레이션 클록보다 훨씬 빠른 시스템클록에 의해 동작한다. REGO에 내장된 FPGA에 존재하는 유한상태기와 부가회로 그리고 분할된 사용자 로직은 사용자 클록에 의해 동작하는 것이 아니라 시스템 클록에 의해 동작하며 동기화된다. 전체 검증회로가 올바른 동작을 하려면 각 FPGA에 내장된 유한상태기간의 동기화는 반드시 만족하여야 한다. REGO에서는 두 단계로 이를 수행한다. 첫째로 PCB보드상에서 각 FPGA까지 시스템클록을 공급할 때 클록스큐가 최소화 하도록 하고, 두 번째로는 FPGA 내부에서 가장 속도가 빠르고 낮은 지연시간을 가지는 클록 분배구조를 이용한다.

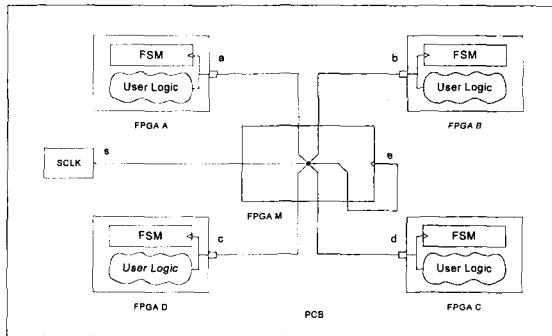


그림 16. 시스템클록 분배  
Fig. 16. distribution of system clock.

그림 16은 4개의 사용자 FPGA와 1개의 모듈 FPGA가 내장된 REGO 모듈의 간략도이다. 각 FPGA에 내장된 유한상태기와 사용자로직, 그리고 부가회로들은 그림 16의 시스템클록 SCLK에 의해 동기화되어야 한다. 이를 위해 SCLK는 FPGA의 주 로우-스큐(primary low-skew) 전역 클록 분배 구조에 연결된 그림 16의 a, b, c, d, e 클록 패드에 연결된다. 그리고 클록스큐를 최소화 하기 위하여 SCLK에서 a, b, c, d, e 까지의 PCB 트레이스는 전통적인 틀 구조로 하여 모두 거리가 같도록 하였다.

7. 소프트웨어 구성

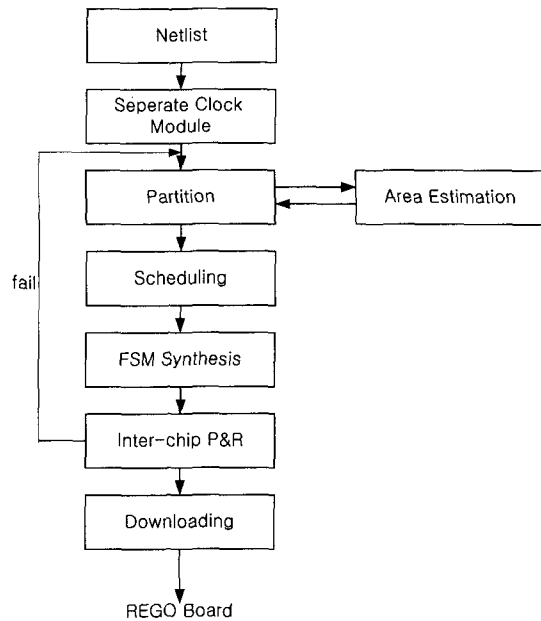


그림 17. REGO의 소프트웨어 흐름도  
Fig. 17. Software flowchart in REGO.

본 논문에서 제안하는 환형 토폴로지 구조의 에블레이터에 적용하는 소프트웨어는 그림 17과 같다. 검증대상회로의 넷리스트(netlist)를 입력받아 클록모듈을 따로 분리한다. 그리고 에블레이터의 각각의 FPGA에 적용될 다수의 부 모듈로 분할한다. 분할된 회로는 각각의 FPGA에 적용될 수 있는지 평가하기 위해 면적을 계산한다. 이때 분할된 회로뿐만 아니라 추가로 인가되는 유한상태기와 부가회로까지 고려하여 평가하도록 해야한다. 이때 분할된 회로와 추가 예상되는 회로의 합이 FPGA의 면적보다 적게 평가되면 다음 단계로 넘어가고, 만일 제공되는 면적을 초과하였을 시는 면적을 만족할 수 있도록 분할을 재 시도한다. 분할된 회로들이 초기의 합수를 만족하며 동작하기 위한 스케줄링을 위해 검증대상회로의 신호선 종속성을 조사하여 자료 의존 그래프(DDG: Data Dependency Graph)를 생성한다. 스케줄러는 자료의존 그래프를 입력받아 신호선을 스케줄링(예: 정적 리스트 알고리즘<sup>[15])하고 이에 따라 각 FPGA의 유한상태기와 환형 통신을 위한 부가회로를 합성한다. 합성이 끝난 각 FPGA의 회로는 P&R을 통해 비트스트림으로 만들어지며 이를 에블레이터의</sup>

각 FPGA에 다운로드하게 된다.

IV. 실험

본 논문에서 제안하는 환형 토폴로지를 갖는 에뮬레이터는 중앙에서 확장시 타 모듈과의 신호전달 통로 역할을 수행하는 FPGA 1개와 분할된 사용자 검증회로가 다운로드될 FPGA 4개의 총 5개의 FPGA로 구성되어 있다. 그림 21은 구현된 REGO 에뮬레이션 보드의 실물 사진이다. 실제 구현된 시스템에 사용된 FPGA는 Xilinx VIRTEX1000BG560 디바이스로 사용자 IO를 404개 가지고 있다<sup>[16]</sup>.

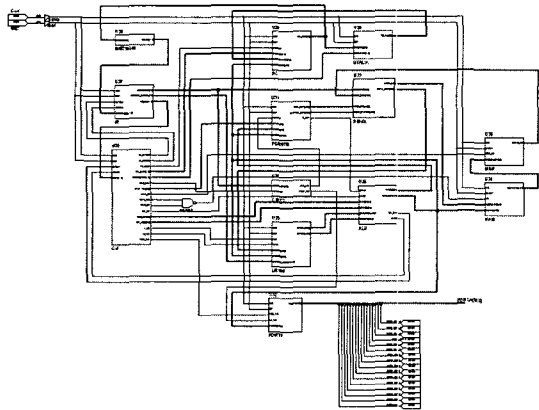


그림 18. 16비트 마이크로컨트롤러의 블록도  
Fig. 18. Block diagram of 16bit microcontroller.

그러나 검증대상회로의 크기가 제작된 REGO 모듈보드에 내장된 FPGA의 용량보다 적어 본 실험을 위하여

그림 19와 같이 환형 통신을 위해 사용 가능한 사용자 IO 핀 수를 각 디바이스마다 40핀으로 제한하고 검증대상회로를 4부분으로 분할하였다. 그림 18은 검증에 사용된 회로로서 데이터 버스가 16비트, 인스트럭션 코드 길이가 12비트인 마이크로컨트롤러이다. 검증회로는 분할은 별도의 상용품을 사용하지 않고 사용자가 수작업으로 수행하였다. 분할시 고려한 사항은 분할된 회로를 거쳐가는 조합경로를 최소화 하는 것이었고 결과로서 분할된 회로를 거쳐가는 모든 조합경로의 길이는 1이었다.

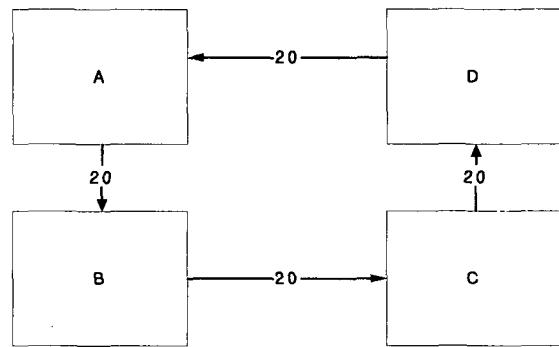
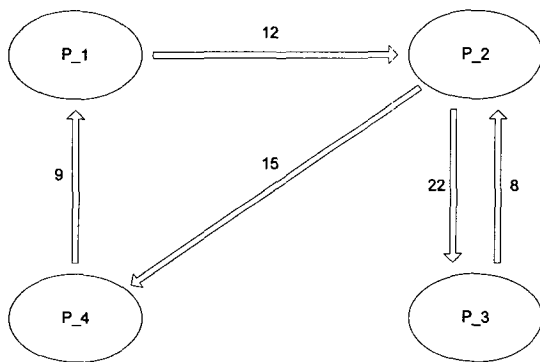
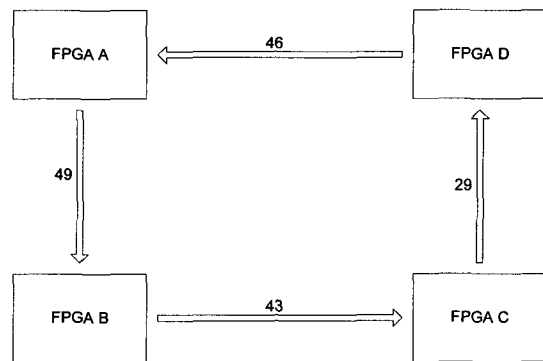


그림 19. 실험에 사용된 REGO 환형 구조  
Fig. 19. REGO ring topology used in experiment.

그림 20(a)는 그림 18의 마이크로프로세서를 분할된 부분을 거쳐가는 조합경로가 1이 되게 4개의 부분회로 P\_1, P\_2, P\_3, P\_4로 분할된 결과이다. 따라서 임의의 부분회로에서 출발한 신호선은 입력단이나 플립플롭의 출력단이며 도착점은 플립플롭의 입력단이나 출력단이다. 에뮬레이션클럭의 한 사이클은 신호선 전송시간에



(a) 분할회로의 연결도



(b) 분할회로의 임의의 할당시 연결도

그림 20. 그림 18의 분할과 전체배치 예  
Fig. 20. Example of partitioning and global placement of fig 18.

필요한 시스템클럭의 수와 FPGA의 최장조합회로의 지연시간을 만족하는 시스템 클럭의 수의 합으로 정해진다. 분할된 부분회로를 에뮬레이터에 존재하는 FPGA A, B, C, D에 배치하는 경우의 수는 총 24가지이며 이중 동일구조를 배제하면 총 6가지이다.

그림 20(b)는 부분회로 P\_1, P\_2, P\_3, P\_4를 각각 FPGA A, FPGA D, FPGA C, FPGA B에 배치한 경우이다. FPGA간 연결수는 신호선에 파이프라인을 적용하지 않고 신호선과 FPGA의 각 핀을 직접 매핑하였을 때 필요한 핀 수를 나타낸다. 이는 각 FPGA의 사용가능 핀을 40핀만으로 제한한 시스템의 핀 한계를 위배한다. 하지만 시분할다중화를 이용하여 분할된 회로의 신호선들을 종속성에 위배되지 않게 스케줄링하여 대상 FPGA에 전달하고, 전송된 모든 신호가 사용자 회로의 메모리 소자에 입력되기 위해 거쳐야하는 조합경로의 지연시간을 보장해준다면 그림 19에서 제공하는 40개의 핀 수를 가지고도 구현 가능하다.

표 1. 12비트마이크로컨트롤러의 스케줄링 결과

Table 1. Experimental result of scheduling of 12bit Microcontroller.

FPGA Placement type	Mobility based list scheduling	T <sub>clk</sub>
1	3	32Mhz
2	4	26Mhz
3	6	20Mhz
4	4	26Mhz
5	4	26Mhz
6	6	20Mhz

그림 21의 REGO 모듈보드에서 FPGA의 핀 간 신호 전달에 걸리는 지연시간은 시스템클럭 160 MHz를 사용하는 것이 가능하였으며 분할된 부분회로의 FPGA 내 최장조합경로의 지연시간은 시스템클럭 2를 사용할 수 있었다. 따라서 FPGA 간 신호전달에 기준이 되는 시스템클럭은 160MHz의 클럭이며 분할회로의 구조에 따라서 평가기간에 소요되는 시스템클럭의 수는 달라질 수 있다. 표 1에서는 그림 20(a)와 같이 분할된 각각의 분할회로 P\_1, P\_2, P\_3, P4를 그림 19의 REGO 모듈의 어느 FPGA에 배치하느냐에 따라 최종 에뮬레이션 클럭이 달라짐을 보여준다. 표 1의 2번째 열은 분

할회로의 배치 후 만들어진 DDG와 그림 19의 에뮬레이터의 연결구조와 가용자원을 입력으로 하여 Mobility를 우선순위함수로 한 리스트스케줄링<sup>[8]</sup>을 적용한 결과로서 모든 신호선의 전송기간에 필요한 시스템 클럭수를 나타낸다. 표 1의 3번째 열은 실행기간에 2개의 시스템클럭이 필요할 때 전체 에뮬레이션 클럭에 소요되는 클럭을 나타낸다. 최종 에뮬레이션 클럭은 부분회로의 배치에 따라 20~32 MHz를 얻을 수 있었는데, 이는 상용 에뮬레이터가 최대로 낼 수 있는 에뮬레이션 속도인 10MHz 보다 약 2~3.2배가 빠른 속도이다. 더욱이 검증할 회로의 크기가 커지면 커질수록 회로의 분할은 게이트 단위가 아닌 설계시에 정해지는 계층적 모듈 단위로 일어나게 되고, 대부분의 이들 모듈간의 경로는 최소 하나 이상의 메모리 소자(플립플롭 또는 래치)를 거치게 되리라는 것을 쉽게 예측할 수 있는데 이와 같은 경우에서 FPGA를 거치는 조합회로의 길이가 1이 되며 에뮬레이션 속도를 최대로 높이는 것이 가능하다. 따라서 다수의 IP들을 포함하는 SoC 수준의 초고집적 회로의 경우에 본 논문에서 제안하는 파이프라인의 신호전달을 이용한 환형 구조의 에뮬레이터를 사용한다면 기존의 부분 크로스바나 그물 구조를 가지는 에뮬레이터의 매우 느린 에뮬레이션 속도를 효과적으로 극복할 수 있음과 동시에 핀 제약에서 발생하는 문제도 동시에 해결할 수 있다.

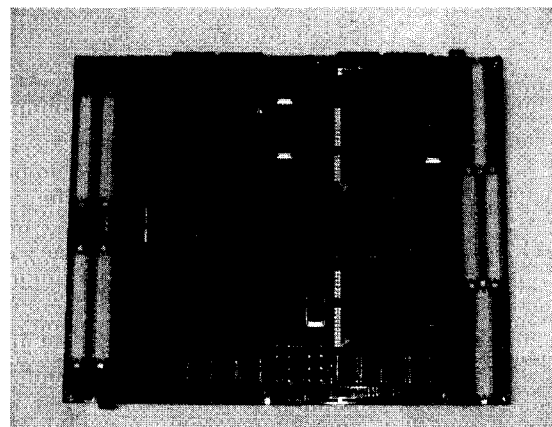


그림 21. REGO 에뮬레이션보드  
Fig. 21. REGO emulation board.

### V. 결론 및 향후 연구과제

다수의 FPGA를 이용한 FPGA 기반의 에뮬레이터에

서 핀 제한 문제는 FPGA 이용률을 크게 떨어뜨려 과도한 FPGA 사용을 유발할 뿐만 아니라 별도의 타이밍 문제를 발생시키며, 기존 방식의 연결 구조들은 SoC 설계에서 일반적인 초고집적 회로에 대한 에뮬레이션에서 연결 구조의 비효율성으로 에뮬레이션의 속도를 크게 저하시킨다. 본 논문에서는 FPGA 기반의 에뮬레이터에서 신호선들의 연결을 환형 토폴로지를 이용한 환형 통신을 이용함으로써 FPGA 핀 제한 문제를 해결하고, 고속의 에뮬레이션 속도를 검증대상 회로 크기의 증가에 독립적으로 유지시킬 수 있는 새로운 방법을 제안하였다. 본 논문에서 제안된 방법은 환형 토폴로지 구조를 형성하여 파이프라인되어진 링 통신을 가능하게 함으로써 하나의 물리적 FPGA 핀에 다수의 로직 신호선들을 할당하는 효과를 봄으로써 FPGA 이용률을 높였고 계층적 구조의 링 토폴로지 구성을 가능하게 함으로써 검증대상 회로의 크기 증가에 독립적인 고속의 에뮬레이션을 가능하게 할 수 있을 뿐만 아니라, 환형 토폴로지 구조가 가지는 가변성을 이용하여 에뮬레이터의 크기를 쉽게 조정할 수도 있으므로 다수의 IP들을 이용하는 시스템수준의 검증도 가능하다는 추가적인 장점이 있다. 향후 연구과제로서 REGO 에뮬레이터에 적합한 스케줄링 알고리즘에 관한 연구는 현재 진행 중에 있으며, 이를 통하여 에뮬레이터의 검증속도를 더욱 높일 수 있을 것으로 판단된다.

### 참 고 문 헌

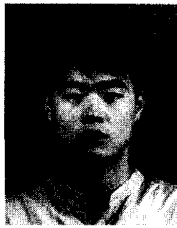
- [1] Hauck, S, Borriello, G, Ebeling, C, "Mesh routing topologies for multi-fpga systems", *IEEE Trans. on VLSI systems*, Vol.6, No.3, pp. 400-408, Sept. 1998.
- [2] Jonathan Babb, Russel Tessier, and Anant Agarwal, "Virtual wires: Overcoming pin limitations in FPGA-based logic emulators", in *Proc. IEEE workshop FPGA-based custom computing machines*, pp. 142-151, April 1993.
- [3] Quickturn Design System, Inc., "Emulation System with time-multiplexed interconnect", *US patent 005960191*, May 30 1997.
- [4] Jianmin Li, Chung-Kuan Cheng, "Routability improvement using dynamic interconnect architecture", *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol.6, Issue.3, pp. 498-501, Sept 1998.
- [5] D. Jones and D. Lewis, "A time-multiplexed FPAG architecture for logic emulation", in *Proc. 3rd Canadian workshop on Field-programmable devices*, pp. 495-498, May 1995.
- [6] Joseph Varghese, Michael Butts, and Jon Batcheller, "An Efficient logic emulation system", *IEEE Trans. on VLSI systems*, Vol.1, No.2, pp. 171-174, June 1933.
- [7] Courtoy, M, "Rapid system prototyping for real-time design validation", in *Proc. 1988 Ninth Int. Workshop on Rapid System Prototyping*, pp. 108-112, 1998.
- [8] Hsiao-Pin Su and Youn-Long Lin, "A Phase assignment method for virtual-wire-based hardware emulation", *IEEE Trans. on CAD of IC and system*, Vol.16, No.7, pp. 776-783, July 1997.
- [9] S.Matic, "Emulation of hypercube architecture on nearest-neighbor mesh-connected processing elements", *IEEE Trans. on Computer*, Vol.39, pp. 698-700, May 1990.
- [10] Bondyopadhyay, P. K "Moore's law governs the silicon revolution", in *Proc. of the IEEE*, Vol.86, Issue.1, pp. 78-81, Jan. 1988.
- [11] B.S. Landman and R.L. Russo. "On a pin versus block relationship for partitions of analogic graphs", *IEEE Trans. on Computer*, Vol.C-20, pp. 1469-1479, 1971.
- [12] Verplaetse, P, "Refinements of rent's rule allowing accurate interconnect complexity modeling", *2001 International Symposium on Quality Electronic Design*, pp. 251-252, 2001.
- [13] Jonathan Babb, Russel Tessier, Matthew Dahl, Silvian Zimi Hanono, David M Hoki, and Anant Agarwal, "Logic Emulator with Virtual Wires", *IEEE Trans. on CAD*, Vol.16, No.6, pp. 609-626, June 1994.
- [14] Chih-Chang Lin, Chang. D, Yu-Liang Wu, Marek-Sadowska. M, "Time-Multiplexed Routing for FPGA design", in *Proc. 1996 ACM*

*Int. Workshop on Field-Programmable Gate Arrays*, pp. 152-155, Feb. 1996.

[16] Xilinx Databook 2001 <http://www.xilinx.com/partinfo/databook.htm>.

[15] Robert A. Walker, Raul Camposano. A survey of high-level synthesis systems, Kluwer Academic Publishers, pp. 16-17, 1991.

저 자 소 개



金 南 道(正會員)

1994년 부산대학교 컴퓨터공학과 졸업(학사). 1996년 부산대학교 대학원 컴퓨터공학과 (공학석사). 1998년 부산대학교 대학원 컴퓨터공학과 (공학박사 수료). 1998년~현재 부산대학교 컴퓨터공학과 박사과정.

<관심분야> 논리합성, VLSI CAD, logic emulator 등



梁 世 陽(正會員)

1981년 고려대학교 전자공학과 졸업(학사). 1985년 고려대학교 컴퓨터공학과 (공학석사). 1990년 University of Massachusetts (공학박사). 1991년~현재 부산대학교 컴퓨터공학과

부교수. <관심분야> VLSI CAD, system verification 등