초고속 시스템 에뮬레이터의 구조와 이를 위한 소프트웨어

김 남 도[†] · 양 세 양[††]

요 약

SoC 설계의 복잡도가 지속적으로 커짐에 따라 기존의 소프트웨어 모델을 이용한 시뮬레이션 방법으로는 이를 검증하기에는 너무 많은 시간이 소요되어 많은 문제가 있다. 이를 해결하기 위해 시뮬레이션 방법보다 훨씬 빠른 검증속도를 제공하는 다양한 FPGA 기반의 로직 에뮬레이터가 활발히 연구되어왔다. 하지만 제한된 FPGA 핀 수로 인해 FPGA 내부에서 매우 낮은 자원이용률을 초래하고 있을 뿐만 아니라, 검증 대상이 되는 회로의 크기가 커짐에 비례하여 에뮬레이션의 속도가 현저하게 느려지는 문제점이 있다. 본 논문에서는 파이프라인 방식의 신호전달을 통하여 FPGA의 자원이용률을 극대화할 수 있을 뿐만 아니라 에뮬레이션의 속도도 크게 높일 수 있는 시스템 수준의 새로운 에뮬레이터 구조와 소프트웨어를 제안한다. 파이프라인의 링을 통하여 다수의 로직신호선을 하나의 실제 핀에 할당하여 핀 제한 문제를 해결하고, FPGA 간의 신호전달 경로를 사용자회로와 분리시킴으로서 빠른 시스템 클록의 사용을 가능케 하며 분할된 회로간에 조합경로를 줄여 실제 에뮬레이션 클록의 속도를 높일 수 있었다. 또한 신호의 전달을 파이프라인 방식으로 보내기 위해 적용하는 스케줄링을 계산의 복잡도가 낮은 휴리스틱 방법을 적용하였다. 12비트 마이크로콘트롤러를 간단한 휴리스틱 스케줄링 알고리즘을 적용한 실험결과를 통하여 높은 검증속도를 확인하였다.

Topology of High Speed System Emulator and Its Software

Nam Do Kim[†] · Sei Yang Yang[††]

ABSTRACT

As the SoC designs complexity constantly increases, the simulation that uses their software models simply takes too much time. To solve this problem, FPGA-based logic emulators have been developed and commonly used in the industry. However, FPGA-based logic emulators are facing with the problems of which not only very low FPGA resource usage rate due to the very limited number of pins in FPGAs, but also the emulation speed getting slow drastically as the complexity of designs increases. In this paper, we proposed a new innovative emulation architecture and its software that has high FPGA resource usage rate and makes the emulation extremely fast. The proposed emulation system has merits to overcome the FPGA pin limitation by pipelined ring which transfers multiple logic signal through a single physical pin, and it also makes possible to use a high speed system clock through the intelligent ring topology. In this topology, not only all signal transfer channels among FPGAs are totally separated from user logic so that a high speed system clock can be used, but also the depth of combinational paths is kept swallow as much as possible. Both of these are contributed to achieve high speed emulation. For pipelined signals transfer among FPGAs we adopt a few heuristic scheduling having low computation complexity. Experimental result with a 12 bit microcontroller has shown that high speed emulation possible even with these simple heuristic scheduling algorithms.

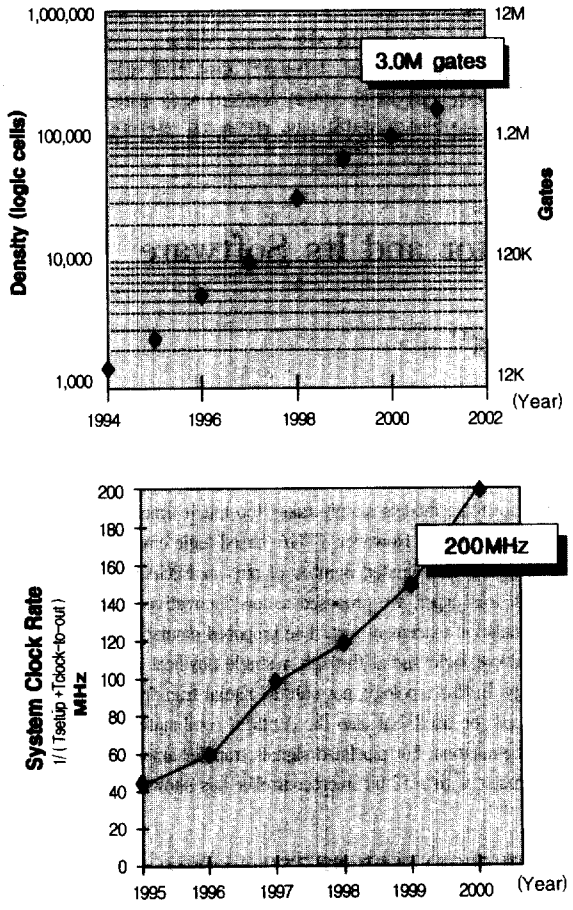
키워드 : 설계검증(design verification), 에뮬레이터(emulator), 현장프로그래밍가능소자(FPGA), 계층적 환형구조(hierarchical ring topology), 파이프라인(pipeline)

1. 서 론

사용자 디자인을 하드웨어 상에서 검증하기 위해서는 높은 집적도와 빠른 수행시간을 제공하는 MPGA(Mask Programmed Gate Array)를 사용할 수 있다. 하지만 MPGA는 사용자 디자인을 단 한번만 프로그래밍 할 수 있어 재사용이 가능하지 않으며 높은 제작비용과 긴 제작기간이 요구됨으로, 디자인 검증시 빈번히 일어날 수 있는 오류발생시 사용자 디자인의 빠른 수정으로 다양한 응용설계를 짧은 시간 안에 개발하여야 하는 현 상황에서 사용하기는 적합하지 않다. 반면에 재구성 가능한 디바이스(예 : Field Programmable Gate Array : FPGA)는 MPGA에 비해 낮은 집적도와 동작속도를 가지지만 소프트웨어 기반의 검증방식보다는 100,000~1,000,000배 빠른 속도로 동작하며[1], 원하는 사용자 디자인을 여러 번 프로그래밍 가능할 뿐만 아니라 대상회로를 디바이스에 구현하는 시간이 MPGA에 비해 아주 빠른 장점을 가지고 있다. 또한 무어의 법칙[2]에서 예상한 VLSI 집적도 증가에 따라 (그림 1)에서와 같이 FPGA의 용량과 동작속도는 매우 빠른 속도로 증가하고 있어 FPGA 기반의 에뮬레이션 시스템의 중요성이 부각되어 이의 활발

† 정 회 원 : 부산대학교 대학원 컴퓨터공학과
†† 정 회 원 : 부산대학교 컴퓨터공학과 교수
논문접수 : 2001년 9월 14일, 심사완료 : 2001년 10월 25일

한 연구가 진행되어왔다[3-7]. 집적회로 설계기술의 발달과 산업체에서 요구하는 다양한 응용에 비례하여 설계 디자인의 크기는 매년 증가하고 있지만 단일 FPGA의 적은 용량으로는 이를 만족시키지 못하는 실정이기 때문에 대다수의 에뮬레이션 시스템은 다수의 FPGA로 구성된다. 일반적으로 다수의 FPGA로 구성된 에뮬레이션 시스템의 경우, 검증하려는 사용자 로직은 에뮬레이션 시스템을 구성하는 다수의 FPGA의 제한된 용량과 핀의 수를 초과하지 않으면서 분할되어야 한다.



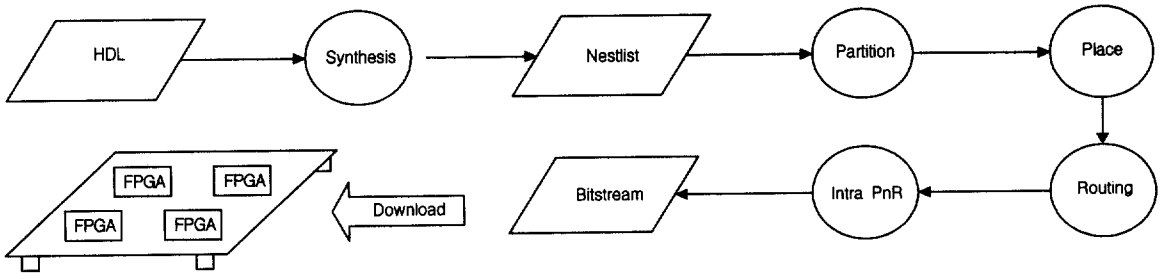
(그림 1) Xilinx FPGA 집적도 & 성능 로드맵

하지만 분할된 회로와 이의 핀 수는 선형관계(linear relationship)[8]에 있어 분할된 회로가 커질수록 필요한 핀 수는 증가하지만 집적도의 기술발전에 비해 패키지 기술발전은 이를 따라가지 못하는 실정이다. 하나의 실제 핀에 하나의 로직 신호선을 할당하는 FPGA 기반의 에뮬레이션 시스템에서 FPGA의 제공 핀 수의 제한으로 인해 FPGA 용량의 낮은 사용률(10~20%)[4]을 나타내는데 이를 에뮬레이션의 핀 제한 문제(pin limitation problem)라 한다[4]. 핀 제한 문제를 효과적으로 해결하여 FPGA 용량의 사용률을 높이기 위해 하나의 실제 핀에 다수의 로직 신호선을 할당하고 이를 시분할다중화방식(Time Division Multiplexing : TDM)으로 전달하는 에뮬레이션 시스템이 연구되었다[4, 7, 9, 11]. 시분할다중화방식의 에뮬레이션 시스템[4, 7, 9, 11]은 핀 제약 문제를 효과적으로 해결하여 FPGA 사용률을 높였지만 신호전달의 구조적인 문제로 인하여 기존 에뮬레이션 시스템의 검증속도인 1~3Mhz의 검증속도를 넘기 힘든 현실이다. 본 논문에서는 새로운 계층적인 환형 연결구조의 FPGA 기반의 에뮬레이션 시스템과 이를 위한 소프트웨어를 제안한다. FPGA 간의 신호전달을 파이프라인 방식으로 전달하여 핀 제한 문제를 해결하고 사용자로직과 분리된 환형 연결구조를 통하여 FPGA 간의 신호전달을 함으로써 빠른 시스템클록을 사용할 수 있으며 계층적인 환형 연결구조를 통해 계획된 라우팅 경로를 제공하여 검증대상회로가 커질 때 급속히 느려지는 기존 에뮬레이션 시스템의 문제점을 해결하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존의 하드웨어 기반의 에뮬레이션 시스템에 대하여 간략히 살펴보고, 3장에서는 본 논문에서 제안하는 계층적 환형 연결구조의 에뮬레이션 시스템의 구조와 신호전송의 방식에 대하여 설명한다. 4장에서는 3장에서 기술한 계층적 환형 연결구조의 에뮬레이션 시스템에서 필요한 소프트웨어에 대하여 기술한다. 5장에서는 본 논문에 적용되는 파이프라인 방식의 신호전달을 위한 스케줄링에 대해 간략히 알아보고 6장에서는 이를 실험한다. 7장에서는 본 논문이 기여하는 바를 언급하고 향후 연구과제와 문제점에 관하여 기술한다.

2. 기존의 에뮬레이터 방법 및 문제점

기존의 FPGA 기반의 하드웨어 에뮬레이터를 이용하여



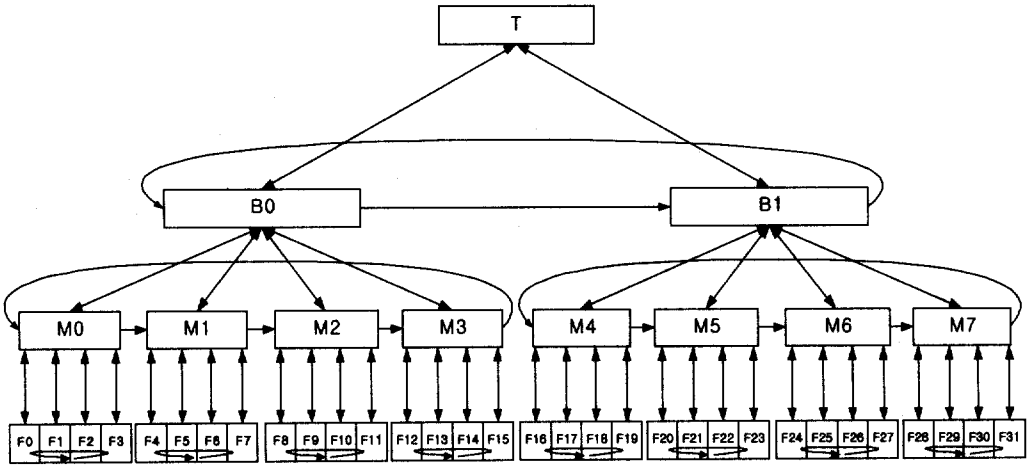
(그림 2) 일반적인 에뮬레이션을 위한 흐름도

검증대상회로를 검증하기 위하여 (그림 2)와 같은 과정을 거친다. 상위레벨에서 HDL로써 기술된 검증대상회로는 합성을 통해 게이트레벨의 네트리스트로 변환되고 분할기에 의해 다수의 부분회로로 분할된다. 분할된 부분회로는 에뮬레이터의 각 FPGA에 할당되고 검증대상회로의 연결도를 바탕으로 전체 배선을 하게 된다. 전체 배선이 완료되면 각 FPGA 내의 CLB(Cell Logic Block)에 부분회로의 게이트가 할당되고 FPGA 내부의 배선을 거치게 된다. FPGA 내부의 배치배선이 완료되면 FPGA 구성 가능한 비트스트림이 만들어지게 되며 이를 각 FPGA에 다운로드 하여 검증을 하게 된다. 이상적으로는 에뮬레이터에서 검증 가능한 검증대상의 크기는 에뮬레이터에 존재하는 각 FPGA의 면적의 합이나, 실제로는 FPGA의 핀 제한 문제로 인해 에뮬레이터 자원을 효율적으로 사용하지 못하여 과도한 FPGA의 사용이 요구되고 이에 따라 신호가 전달되기 위해 거쳐가는 FPGA의 경계 수가 늘어남으로 인해 전체 검증속도를 떨어뜨리는 요인으로 작용한다. 이를 해결하기 위해 연구된 초기의 시분할다중화기반의 에뮬레이션 시스템은 느린 동작속도와 비동기회로의 모델링의 어려움 때문에 실제 현장에서 널리 사용되지 못하였다[7]. 최근에는 위의 문제점들을 보안한 시분할다중화기반의 에뮬레이션 시스템들[4, 7]이 설계현장에서 많이 사용되고 있지만 이들 또한 검증대상회로의 크기가 증가할수록 검증속도가 느려지는 구조적인 문제를 안고 있어 1~3Mhz의 느린 검증속도를 나타내고 있다. 또한 에뮬레이터의 대표적인 연결구조인 그물연결은 라우팅지연시간을 예측하기 어려우며 신호선이 거쳐가는 FPGA 경계수를 늘여 타이밍 문제를 유발할 수 있으며[10] 크로스바연결구조[3, 7]는 값비싼 연결전용 디바이스의 증가와 보드 레벨에서의 많은 연결이 요구되어 에뮬레이션 PCB 제작에 많은 비용이 소요되며 에뮬레이션 시스템 확장시 보드레벨에서의 연결이 필요하므로 전체 에뮬레이션 시스템의 엄청난 가격상승효과를 나타낼 뿐만 아니라 확장성이 크게 떨어지는 문제점이 있다.

3. 계층적구조의 재구성가능한 에뮬레이터

3.1 에뮬레이터 구조

본 논문에서는 에뮬레이션 시스템 확장성이 뛰어나고 크로스바 연결구조보다 상대적으로 보드레벨의 연결이 적으며, 에뮬레이션 시스템 확장시 발생하는 급격한 검증속도 감소의 문제점을 해결할 수 있는 계층적인 환형 연결구조를 제안한다. 제안하는 에뮬레이터의 구조는 (그림 3)과 같이 사용자 FPGA인 F0~F31를 부분적으로 환형으로 연결하고 이들을 계층적으로 구성하였다. 에뮬레이션 시스템의 최하위 구성요소인 모듈 M0~M7은 각각 4개의 사용자 FPGA로 구성되어 있으며 M0~M3, M4~M7은 각각 환형으로 연결된다. 모듈의 상위구성 요소인 박스 B0~B1은 각각 4개의 모듈로 구성되며 이들 또한 환형으로 연결되어 있으며 두 개의 박스로 캐비닛 T가 구성된다. 환형으로 연결된 모듈, 박스, 캐비닛의 수는 검증대상회로의 크기에 따라 신축적으로 변할 수 있어 에뮬레이션 시스템의 크기를 큰 비용 없이 쉽게 확장 또는 축소시킬 수 있다. 즉 전체 시스템은 하나 또는 다수의 모듈로 구성될 수도 있으며 하나 이상의 박스 또는 캐비닛으로 구성될 수도 있다. 또한 FPGA 간의 체계적인 신호전달 방식으로 신호전달에 걸리는 FPGA 간의 평균 hop수(어떤 FPGA에서 출발한 신호선이 대상 FPGA에 도달하기 위해 거쳐야 하는 FPGA의 수)가 검증대상회로의 크기가 커짐에 비례하여 급격히 높아지는 기존의 에뮬레이션 시스템에 비해 완만히 증가하여 속도가 현저하게 느려지는 기존의 시스템에 비해 성능을 유지할 수 있다. (그림 3)의 사용자 FPGA F1에서 시작한 조합회로 신호선이 대상 FPGA F8로 전달할 때 신호선의 라우팅경로는 항상 F1 → M0 → M1 → M2 → F8이며, FPGA F1

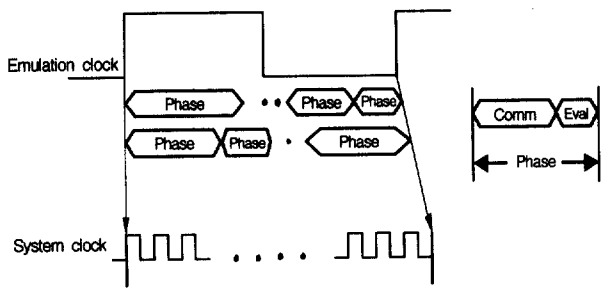


(그림 3) 계층적인 환형 구조를 갖는 시스템 에뮬레이터의 구성

에서 시작한 조합회로 신호선이 대상 FPGA F20으로 전달할 때 신호선의 라우팅경로는 F1→M0→B0→B1→M5→F20이다. 기존의 에뮬레이터에서처럼 신호선의 라우팅 결과에 따라 FPGA 간의 hop수가 정해지는 것이 아니라 FPGA 간의 신호선 전송에 필요한 hop수는 이미 정해져 있으며 계층적인 신호선 전송방식으로 검증시스템의 사이즈가 커질 경우에도 약간의 hop수만이 증가하게 되는 것이다. 이로서 검증회로 컴파일시 전체 FPGA의 배치 배선(Global Place and Routing)의 복잡도를 크게 완화할 수 있으며 검증회로의 사이즈가 커짐에도 높은 성능을 유지할 수 있다.

3.2 파이프라인 방식의 신호전송

하나의 물리적 핀에 하나의 로직 신호선을 할당하는 방식의 FPGA기반의 에뮬레이션 시스템의 경우 대부분의 FPGA 면적을 사용하지 못함으로 인해 많은 수의 FPGA를 필요로 한다. 이는 신호선이 거쳐야 하는 FPGA의 수를 늘여 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 4)에서와 같이 고속의 시스템 클록을 사용하였다. 본 논문에서 제안하는 에뮬레이션 시스템에 적용하는 에뮬레이션 클록은 [4]에서와 같이 여러 개의 단계로 나누어지며 각 단계는 전송기간과 평가기간으로 구성된다. 시스템 클록의 사이클은 FPGA내 최장 조합회로의 지연시간보다는 상대적으로 매우 빠른 FPGA 간의 최장 라우팅 지연시간의 크기로 한다.



(그림 4) 에뮬레이션 클록의 구성

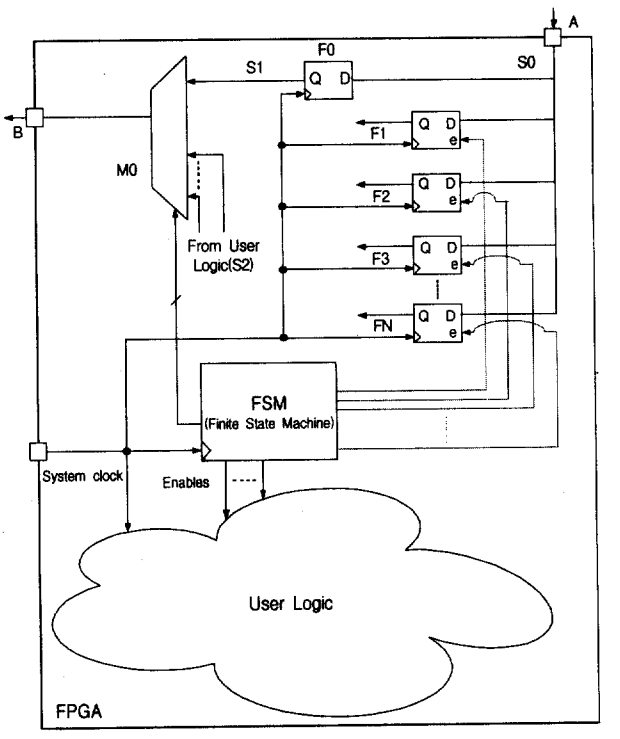
전송기간을 구성하는 시스템 클록의 수는 한 FPGA의 내부회로를 평가하기 위해 필요한 모든 신호선이 도달하기까지 필요한 시스템 클록의 수이며, 평가기간은 한 FPGA에 도달할 신호들이 안정적인 상태로 사용자로직에 입력되어 평가되는 FPGA내의 최장조합경로의 지연시간을 보장하는 시스템 클록의 수에 의해 결정된다. 따라서 전체 에뮬레이션 클록의 길이는 데이터 의존도 그래프(Data Dependency Graph : DDG)에서 FPGA 자원이 무한할 때 얻어지는 최장 지연시간 값을 하한 값(low bound)으로 한다. 이 때 다음과 같은 식이 성립한다.

$$Tclk \geq n \times tc + m \times tr \quad (1)$$

이 식에서 $Tclk$를 에뮬레이션 클록 한 사이클의 길이라고 할 때 tc는 FPGA 내의 최대 조합경로지연시간이고 tr은 FPGA 간의 최장 라우팅 지연시간이다. 이 때 n은 DDG의 노드 중 최장조합경로의 가장 높은 깊이를 나타내며, m은 DDG의 어떤 노드가 터미널노드까지 도달하기 위해 거쳐가는 FPGA 경계수 중 Tclk를 가장 크게 하는 FPGA 경계수를 나타낸다. 만일 에뮬레이션 시스템이 제공하는 핀 수가 무한히 많다면 $Tclk = n \times tc + m \times tr$ 이지만 현실적으로 FPGA의 용량에 맞게 분할하는 경우 필요 핀 수는 FPGA 핀 수를 초과하게 되며 신호선의 전달을 위한 스케줄링 적용시 자원의 부족으로 생길 수 있는 휴면시간(idle time) 때문에 대부분 $Tclk$는 $n \times tc + m \times tr$를 초과하게 된다.

3.3 FPGA 내부 구조

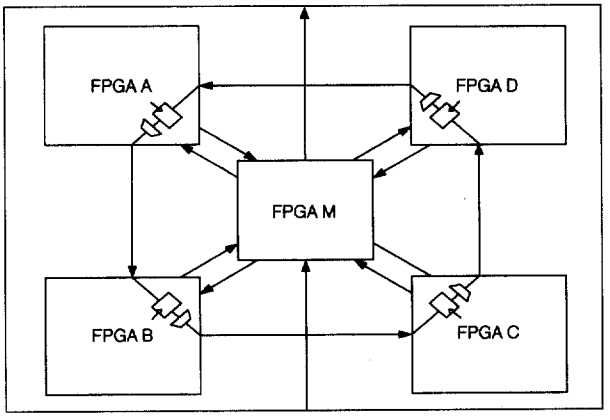
본 논문에서 제안한 에뮬레이터에는 FPGA 간의 신호전송을 고속의 FPGA 내부 클록을 이용하여 파이프라인 방식으로 전달함으로써 핀 제한 문제를 극복하여 FPGA 사용률을 높였다. 분할된 검증회로의 다수의 로직 신호선을 하나의 실제 핀에 적절한 시간대에 할당하기 위하여 각 FPGA에는 (그림 5)에서와 같이 전체 검증대상회로에서 분할된 사용자



(그림 5) 에뮬레이션 시스템의 FPGA 내부 상세도

회로뿐만 아니라 환형으로 연결된 실제 핀을 통해 적절한 시간대에 로직 신호선을 할당하여 파이프라인 방식으로 전달하기 위한 부가회로들이 존재한다. 환형 연결을 구성하는 두 핀 A, B 사이에 플립플롭 F0가 있고, 타 FPGA에서 환형 연결을 통해 들어온 신호 S0를 FPGA 내부의 회로에 인가하거나 타 FPGA로 전달하기 위해 이를 저장하는 플립플롭의 어레이 F1-FN이 있다. 환형 연결을 통해 다른 FPGA로 전달될 내부회로의 신호선들 S2와 F1-FN중에서 다른 FPGA로 전달될 신호들, 그리고 외부에서 환형을 통해 인가된 신호가 플립플롭에 인가되고 이의 출력 S1이 멀티플렉서 M0에 연결되어 있다. 사용자 회로와 유한상태기, 그리고 모든 부가회로들은 에뮬레이션 클록보다 훨씬 빠른 시스템 클록에 의해 동기화 된다. 이들의 동작을 제어하는 유한상태기는 초기회로의 분할된 결과로 만들어진 DDG와 에뮬레이션 시스템의 구성형태를 받아들여 이를 스케줄링한 결과에 따라 에뮬레이션을 위한 검증회로 컴파일시 자동적으로 생성되어 사용자 회로와 결합되어 동작하게 된다.

(그림 6)은 본 논문에서 제안한 계층적 구조의 가장 하위 구성요소인 모듈의 환형 구조 연결도이다. 각 모듈은 4개의 사용자 FPGA인 FPGA A, FPGA B, FPGA C, FPGA D와 상. 하위레벨간의 통신을 위한 모듈 FPGA M으로 구성된다. 모듈 FPGA는 타 모듈과의 신호전달을 위해 모든 사용자 FPGA와 연결을 가지고 있으며 같은 모듈내의 신호전달은 환형으로 연결된 사용자 FPGA 간의 연결로서 파이프라인 방식으로 이루어지며 이를 위해 모듈내의 모든 FPGA인 사용자 FPGA, 모듈 FPGA는 환형 구성을 위한 부가회로와 이를 제어하는 유한상태기로 이루어진다.

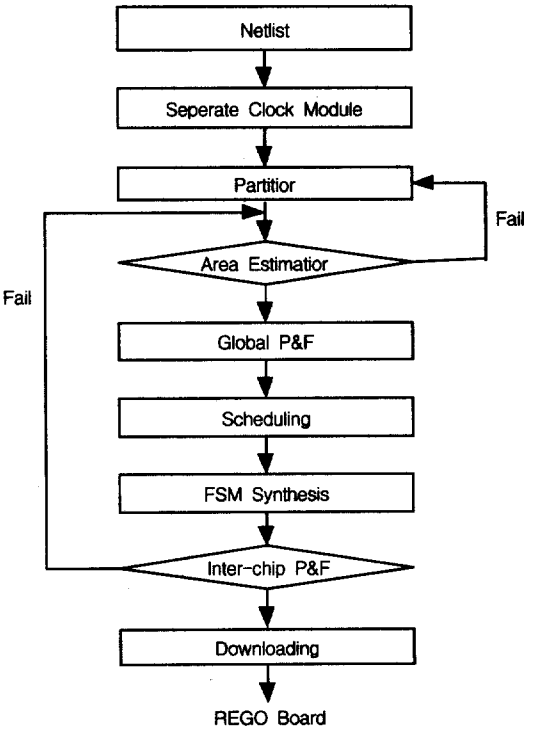


(그림 6) 모듈에서의 환형 구조연결도

모듈의 상위 구성요소인 박스, 캐비닛의 환형 연결구조와 신호전달은 모듈에서의 경우와 동일하다. 다만 모든 사용자 회로는 모듈의 구성요소인 사용자 FPGA에만 존재한다. 본 논문에서 제안하는 계층적 환형 구조의 에뮬레이션 시스템은 FPGA 간의 신호선들의 전송을 계층적인 환형 구조를 통한 파이프라인 방식을 이용함으로써 FPGA 간의 통신대역폭을 크게 증대시켜 FPGA 핀 수 보다 많은 로직 신호선을 포함하는 사용자회로를 FPGA에 포함시킬 수 있어 FPGA 사용률을 높이게 된다.

4. 에뮬레이션 소프트웨어 구성

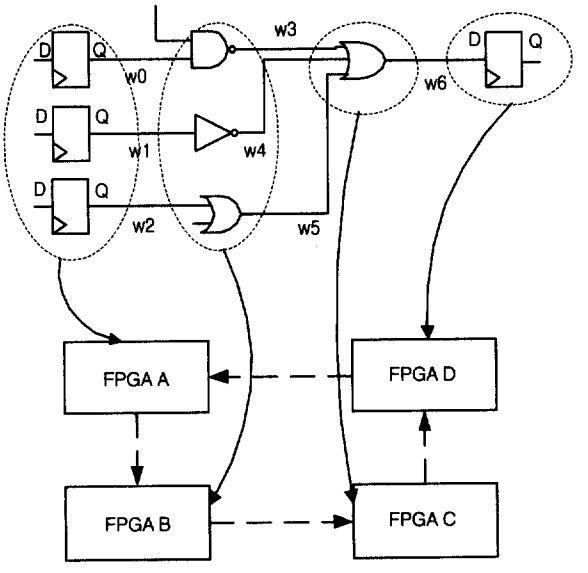
본 논문에서 제안하는 환형 토폴로지 구조의 에뮬레이션 시스템의 소프트웨어의 구성은 (그림 7)과 같다. 검증대상 회로를 합성한 후 생성된 네트리스트(netlist)를 입력받아 클록모듈을 따로 분리한다. 그리고 에뮬레이터 시스템의 각각의 FPGA에 적용될 다수의 부 모듈로 분할하고, 분할된 회로는 각각의 FPGA에 적용될 수 있는지 평가하기 위해 면적을 계산한다. 이 때 분할된 회로뿐만 아니라 추가로 인가되는 모듈인 유한상태기, 플립플롭들까지 고려하여 평가해야 하고 만일 분할된 회로와 추가 예상되는 회로의 합이 FPGA의 면적보다 적게 평가되면 다음 단계로 넘어가고, 제공되는 면적을 초과하였을 시는 면적을 만족할 수 있도록 분할을 재시도한다. 그리고 분할된 회로를 에뮬레이션 시스템의 각 FPGA에 할당한다. 할당시 고려하여야 할 점은 전체 에뮬레이션 시스템의 신호전송을 최소화하는 것이다. 에뮬레이션의 검증속도를 최소화하기 위한 효과적인 스케줄링은 필수적이다. 스케줄링을 위하여 검증대상회로의 신호선 종속성을 조사하여 DDG를 생성한다. 스케줄러는 자료의존 그래프와 에뮬레이션 시스템의 연결구조를 입력



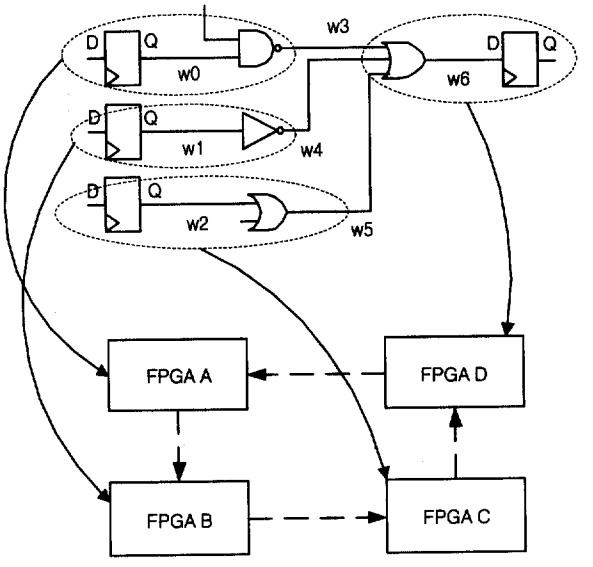
(그림 7) 제안된 에뮬레이터의 소프트웨어 흐름도

받아 신호선을 스케줄링(예 : 정적 리스트 알고리즘[12])하고 이에 따라 각 FPGA의 유한상태기와 환형 통신을 위한 부가회로를 합성한다. 합성이 끝난 각 FPGA의 회로는 배치 배선(Place and Routing)을 통해 비트스트림으로 만들어지며 이를 에뮬레이션 시스템에 다운로딩하게 된다.

기존의 에뮬레이터에서는[4, 7] 회로의 분할 시 적은 연결선을 가지고 균등하게 분할하는 기존의 분할기를 그대로 사용하였다. 에뮬레이션 시스템의 검증속도는 에뮬레이션 클록에 의해 결정되며, 에뮬레이션 클록은 식 (1)을 하한선으로 하여 주어진 자원과 스케줄링에 의해 필요한 시스템 클록수가 정해지며 분할된 회로의 최장조합경로와 에뮬레이션 시스템의 FPGA 간의 평균 라우팅거리에 의해서 주로



(a) Level - oriented Partition

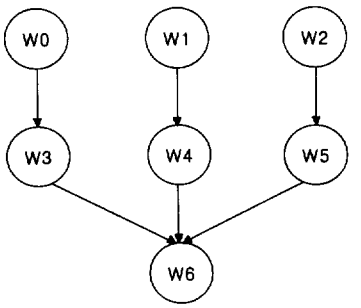


(b) Combinational depth-oriented Partition

(그림 8) 분할 예제

결정된다[4]. 따라서 기존의 고려사항을 가지고 분할하는 분할기를 사용한다면, 시분할 다중화를 사용하여 핀 제한문제를 제거한다하더라도 빠른 검증속도를 기대하기 어렵다. 따라서 본 논문에서 제안하는 에뮬레이터 시스템에 적용하는 분할기의 최대 고려사항은 FPGA를 거치는 최장조합경로를 최소로 만드는 것이다. (그림 8)은 간단한 예제회로가 본 논문에서 제안한 환형으로 연결된 4개의 FPGA인 FPGA A, FPGA B, FPGA C, FPGA D로 구성된 에뮬레이터에 각기 다르게 분할되어 할당된 모습이다.

인접 FPGA 간의 라우팅에 요구되는 시스템클록의 수는 1이고, FPGA내의 최장경로는 시스템클록 1개로 신호선의 값을 보장한다고 할 때 (그림 9)는 (그림 8)과 같이 검증대상회로를 분할하여 FPGA에 배치할 때 만들어지는 DDG를 나타낸다. DDG에 적용할 스케줄링과 제공되는 FPGA 핀 수에 따라 에뮬레이션에 필요한 시스템클록의 수가 결정된다. (그림 9) (a)에서 FPGA를 거치는 최장조합경로는 3이다. (그림 9) (a)의 w0, w3, w6의 경로만을 생각할 때 (그림 8) (a)의 예제회로에서 신호선 w0가 FPGA A에서 FPGA B로 전달되기 위한 전송기간, FPGA B에 도달한 신호에 의해 내부회로가 실행되어야하는 평가기간, w3의 전송기간 FPGA C의 평가기간, w6의 전송기간, 그리고 FPGA D에서의 평가기간이 필요하다. 하지만 (그림 9) (b)는 각 하나의 전송기간과 평가기간만을 필요로 한다. 즉 최장조합경로의 길이를 최소로 하여 에뮬레이션 클록을 구성하는 요소 중에서 큰 비중을 차지하는 평가기간의 수를 줄임으로써 최종 에뮬레이션 클록의 길이를 큰 폭으로 줄일 수 있다.



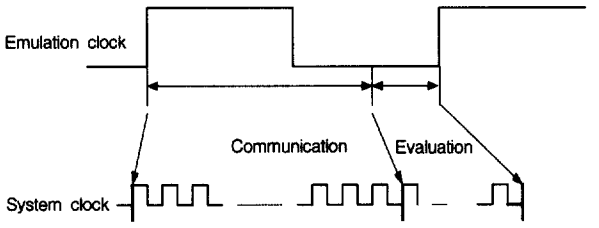
(a) DDG for Fig 8. (a)

(b) DDG for Fig 8. (b)

(그림 9) (그림 8)의 DDG

최근의 시스템 디자인은 사용자회로뿐만 아니라 메모리, 프로세서 코어 및 DSP(Digital Signal Processor) 코어와 같은 다양한 모듈들이 하나의 칩 내부에 집적된 시스템급 회로디자인의 설계 비중이 커지고 있는 실정이다. 검증할

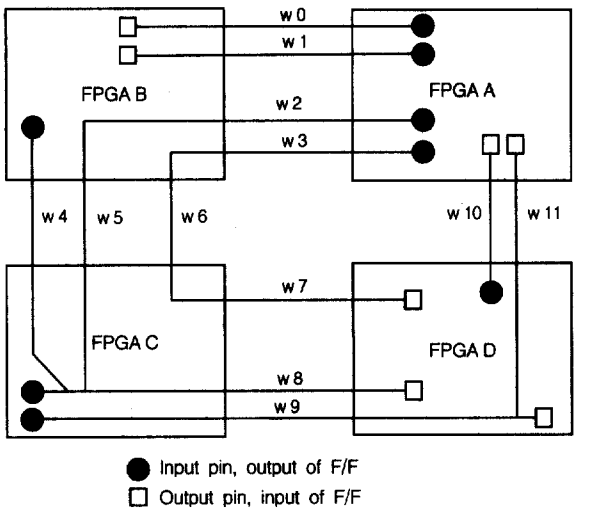
회로의 크기가 커지면 커질수록 회로의 분할은 게이트 단위가 아닌 설계시에 정해지는 계층적 모듈 단위로 일어나게 되고, 대부분의 이들 모듈간의 경로는 최소 하나 이상의 메모리 소자인 플립플롭 또는 래치를 거치게 되리라는 것을 쉽게 예측할 수 있는데 이와 같은 경우 모듈 단위로 분할된 부분회로의 조합경로는 1이 되고 (그림 10)과 같이 하나의 전송기간과 평가기간으로 구성된 에뮬레이션 클록이 구성된다.



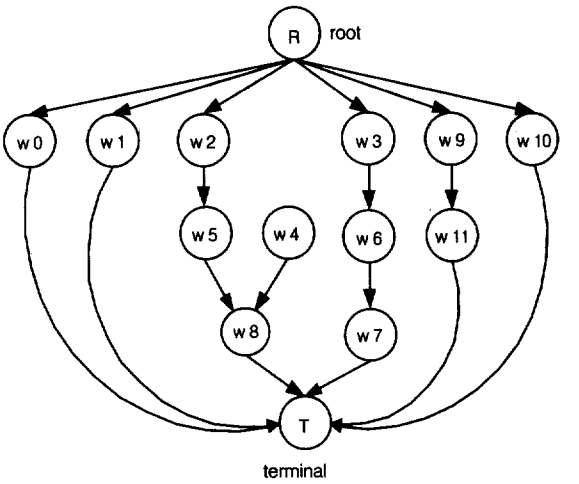
(그림 10) 에뮬레이션 클록의 구성

5. 스케줄링

본 논문에서 제안하는 에뮬레이션 시스템의 제한된 자원으로 최대의 검증속도를 만들기 위해서는 효과적인 스케줄링 알고리즘이 적용되어야 한다. 설계기술의 비약적인 발전에 따라 집적 회로의 복잡도는 급속도로 증가하고 있으며, 특히 시스템급 하드웨어기반의 에뮬레이션 시스템에서의 검증대상회로의 크기는 수 백만개 이상의 로직신호전달을 포함하고 있다. 일반적으로 jobshop 문제는 각기 다른 타입의 작업을 수행하는 $m(k=1,\cdots,m)$개의 머신을 가지고 있으며 각 머신은 어떤 시간에 오직 한가지 작업을 수행할 수 있다. 또한 $q_j(k=1,\cdots,q_j)$개의 작업으로 구성된 $n(j=1,\cdots,n)$개의 job으로 구성되며 각 job의 작업들은 순서에 맞게 수행되어야 한다. 에뮬레이션 시스템의 라우팅을 위한 FPGA 간의 연결을 머신의 타입이라 하고 DDG의 각 노드를 job이라 한다. 그리고 DDG에서 노드의 연결을 job의 작업순서라고 할 때 이는 jobshop 문제라고 볼 수 있다. jobshop 문제는 integer linear programming[12, 13]과 Branch and Bound 알고리즘을 사용하여 최적의 해를 구할 수 있으나, 계산의 복잡도가 너무 크기 때문에 문제의 사이즈가 상당히 큰 시스템급 디자인을 스케줄링하기에는 적당하지 않다. 따라서 에뮬레이션에서의 스케줄링 문제에서는 복잡도가 크지 않으면서 타 알고리즘에 비해 성능이 크게 떨어지지 않는 휴리스틱방법을 선택하여야 한다. 대표적인 휴리스틱 알고리즘으로는 리스트 스케줄링(List scheduling) 알고리즘이 있으며 우선순위 함수에 따라 다양한 응용이 있다. (그림 11) (a)는 4개의 사용자 FPGA로 구성된 검증시스템에 검증대상회로가 분할되어 각 FPGA에 배치되어 만들어진 FPGA 경계를 지나는 모든 조합경로들의 모습이다. (그림 11) (b)는 (그림 12) (a)의 모든 조합경로로부터 만들어진 DDG의 결과이다. DDG의 노드는 작업타입(operation type)을 나타내며 연결선은 작업의 의존도를 나타낸다. 조합경로 분석을 위해 검증회로의 입력단이나 플립플롭의 출력단에서 시작되어 검증회로의 출력단 또는 플립플롭의 입력단까지의 모든 조합경로를 검색한다. (그림 11) (a)의 w0, w1은 FPGA A에서 시작하여 FPGA B까지 조합경로이고, w4, w5는 FPGA B에서 시작하여 FPGA C까지 조합경로이다. 또한 FPGA A에서 출발한 w3은 FPGA B와 FPGA C의 조합회로를 거쳐 FPGA D에 인가된다. DDG는 최상위의 루트노드와 최하위의 터미널노드는 더미노드로써 모든 신호전달은 루트의 첫 하위노드들의 집합 w0, w1, w2, w3, w9, w10으로부터 시작되며, 터미널노드를 만날 때까지 계속된다. 루트노드와 터미널노드 이외의 노드는 FPGA 간의 신호전달을 나타내며 이들간의 연결선은 신호 의존을 나타



(a) Combinational path

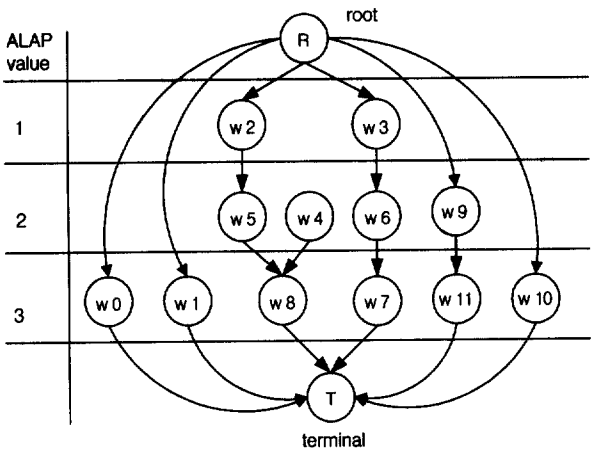


(b) DDG for (a)

(그림 11) 조합경로와 대응하는 DDG

낸다. FPGA B에서 FPGA C로의 신호전달 w5는 FPGA A에서 FPGA B로의 신호전달 w2 이후 시간에 일어나야 하며 FPGA C에서 FPGA D로의 신호전달 w8 이전시간에 할당되어야 한다. (그림 11) (a)의 각 FPGA의 핀 수를 4개라고 할 때 FPGA A는 FPGA B, FPGA D와 연결되어 있으며 FPGA C는 FPGA B, FPGA D와 연결되어 있다고 가정한다. 그리고 연결된 핀들을 이용하여 (그림 6)처럼 환형 경로를 구성한다. 인접한 FPGA 간의 실제 핀은 2개이고 연결해야 하는 로직 신호선은 이를 초과하고 있다. I_{AB}를 FPGA A에서 FPGA B로 전달해야할 신호의 수라 할 때 I_{AB}, I_{BC}, I_{CD}, I_{DA}의 수는 각각 4, 3, 3, 2이다.

ALAP(As Late As Possible) 값을 이용한 리스트 스케줄링을 사용할 때 초기 DDG의 모든 노드의 ALAP값은 -1로 할당한다. 어떤 노드의 ALAP값이 -1이라면 그 노드는 아직 ALAP값을 할당받지 못한 것을 의미한다. (그림 11) (b)의 DDG의 최대 경로 길이는 터미널노드를 제외하고 3이다. 이 값을 이용하여 ALAP값이 -1이고 하위노드가 터미널노드에 연결된 모든 노드 w0, w1, w8, w7, w11, w10에 대하여 ALAP값 3을 할당한다. 어떤 노드의 모든 하위노드에 ALAP값이 할당되면 ALAP값이 정해진다. 즉 ALAP(A) = Min(ALAP(A의 모든 하위노드)) -1 이다. 만일 어떤 노드 A의 하위노드들의 ALAP값이 4, 5, 3이라면 ALAP(A) = Min(4, 5, 3) -1 이며 ALAP(A) = 2가 된다.



(그림 12) (그림 11) (b)의 ALAP 값

〈표 1〉 리스트 스케줄링의 예

Time	R_{AB}	R_{BC}	R_{CD}	R_{DA}	Ready list
tr0	w2, w3	w4	w9	w10	w0, w1, w2, w4, w3, w9, w10
tr1	w0, w1				w0, w1, w5, w6, w11
te0	FPGA B Evaluation				
tr2		w5, w6			w5, w6, w11
te1	FPGA C Evaluation				
tr3			w7, w8		w8, w7, w11
te2	FPGA D Evaluation				
tr4				w11	w11
te3	FPGA A Evaluation				

(그림 12)는 (그림 11) (b)에 ALAP값을 할당한 모습이며 <표 1>은 (그림 11) (b)를 ALAP값을 이용한 리스트 스케줄링한 결과이다. <표 1>의 tr은 FPGA 간의 라우팅에 걸리는 기본단위를 나타내고, te는 FPGA 내부의 최장경로의 지연시간을 나타낸다. R_{AB}는 FPGA A에서 FPGA B로 연결된 자원을 나타낸다.

6. 실 험

본 논문에서 제안한 계층적 환형 구조의 에뮬레이터의 가장 하위 요소인 모듈의 실제 구현은 4개의 사용자 FPGA와 확장시 타 모듈과의 신호전달 통로역할을 수행하는 FPGA 1개로 구성되어 있으며 사용 FPGA는 VIRTEX1000BG560[14]이다. 실험에 사용된 검증대상 회로는 데이터 버스가 16비트, 인스트럭션 코드 길이가 12비트인 마이크로 콘트롤러로써 이 회로의 분할은 별도의 상용툴을 사용하지 않고 사용자가 수작업으로 수행하였다. 이때 고려된 사항은 분할된 회로를 거쳐가는 조합경로를 최소로 하는 것이었고 결과로서 조합경로의 길이는 1이다. 실험에 사용된 스케줄링 알고리즘은 ASAP와 ALAP, 그리고 이 둘을 이용한 이동성(Mobility)을 우선순위 값으로 가지는 리스트 스케줄링이다. 환형 통신을 위해 사용 가능한 사용자 IO 핀을 40핀으로 제한하였고 <표 2>는 스케줄링 방법과 분할된 회로의 FPGA의 배치에 따라 달라지는 에뮬레이션 클록의 변화이다. 분할된 회로의 모든 조합경로는 1이다. 따라서 에뮬레이션 클록은 (그림 10)에서처럼 전송기간과 평가기간으로 이루어진 하나의 단계만으로 구성된다. 또한 분할된 회로의 가능한 FPGA배치는 총 6가지이다. <표 2>는 FPGA 배치와 스케줄링에 따라 신호전송시 요구되는 시스템클록의 수를 나타낸다. <표 2>에서와 같이 ASAP, ALAP, Mobility 모두 3~6의 시스템클록이 요구되었다. 실제 FPGA 간의 라우팅 지연시간은 160Mhz의 클록을 이용할 수 있었고 FPGA내의 최장 조합회로는 160Mhz의 시스템 클록 두 개를 사용하여 전달하였다. 이 때 전체 에뮬레이션 클록을 구성하는 시스템클록의 수는 ASAP, ALAP, Mobility 모두 5~8이고, 32Mhz~20Mhz의 검증속도를 얻을 수 있었다. 이와 같은 에뮬레이션 속도는 현재 상용 에뮬레이터들의 에뮬레이션 속도인 1Mhz~3Mhz에 비하여 대략 10배 정도로 빠른 것이다.

<표 2>에서와 같이 ALAP와 Mobility를 이용한 리스트 스케줄링이 ASAP보다 우수하다. 이는 ASAP는 신호의존도 그래프에서 회로의 최장조합경로를 반영하지 못하기 때문에 ALAP, Mobility를 이용할 때에 비해 결과가 좋지 못하다. ALAP와 Mobility를 이용한 리스트 스케줄링은 회로

〈표 2〉 12비트마이크로콘트롤러의 스케줄링 결과

FPGA Place type	ASAP	*Tclk*	ALAP	*Tclk*	Mobility	*Tclk*
1	3	32Mhz	3	32Mhz	3	32Mhz
2	4	26Mhz	4	26Mhz	4	26Mhz
3	6	20Mhz	6	20Mhz	6	20Mhz
4	5	22Mhz	4	26Mhz	4	26Mhz
5	4	26Mhz	4	26Mhz	3	32Mhz
6	6	20Mhz	6	20Mhz	6	20Mhz

의 구성과 분할, FPGA 배치에 따라 결과는 달라질 수 있다. 하지만 검증할 회로의 크기는 날로 증가하는 추세이며, 이를 계산 복잡도가 큰 알고리즘인 integer linear program, force-directed scheduling을 사용하는 것은 짧은 디버깅 사이클을 요구하는 현 상황에 적합하지 않다. 따라서 본 논문에서 제안하는 에뮬레이터 시스템에는 계산 복잡도가 적으면서 비교적 좋은 결과를 나타내는 Mobility를 이용한 리스트 스케줄링을 적용하였다.

7. 결론 및 향후 연구 방향

본 논문에서 제안한 FPGA 기반의 에뮬레이션 시스템은 FPGA의 연결을 계층적인 환형 토폴로지로 구성하고 로직 신호선의 전달을 계층적인 환형 구조에서 파이프라인 통신을 이용하였다. 이로서 FPGA 핀 제한 문제를 해결하여 FPGA 사용률을 높이고, 검증대상 회로 크기의 증가 시에도 고속의 에뮬레이션 속도를 유지할 수 있으며 환형 토폴로지 구조가 가지는 가변성을 이용하여 에뮬레이션 시스템의 크기를 쉽게 조정할 수도 있어 다수의 IP들을 이용하는 시스템수준의 검증도 가능하다. 또한 에뮬레이션 속도를 증대시키기 위한 효과적인 분할 방법을 제시하였다. 12비트 마이크로콘트롤러 예제에 대하여 로직 신호선들의 전송을 파이프라인 방식으로 보내기 위한 ASAP, ALAP, Mobility를 이용한 리스트 스케줄링을 적용한 결과, 고속의 에뮬레이션 속도를 얻을 수 있음을 보였다. 에뮬레이터 시스템의 성능은 분할기와 스케줄링 방법에 따라 많은 차이를 보일 수 있다. FPGA를 거치는 조합경로의 수를 줄일 수 있도록 분할하고 최적화된 스케줄링을 적용한다면 에뮬레이터 성능향상에 크게 기여할 것이다. 하지만 현재 본 논문에서 제안하는 에뮬레이션 시스템에 적용한 Mobility값을 이용한 정적 리스트 알고리즘은 회로 분할 시 FPGA를 거치는 조합 경로 길이를 1이 되게 분할한 회로에 대하여는 최적의 결과를 나타내지만 FPGA를 거치는 조합경로길이가 2이상인 회로에 대하여는 최적의 결과를 나타내지 못한다. 따라서 이를 위한 효과적인 스케줄링 알고리즘의 연구가 필요하다. 이와 같은 경우에 Mobility와 DDG의 구조적인 연결정보를 이용한 효과적인 휴리스틱을 연구중이며 이를 이용하여 빠른 에뮬레이션 동작을 나타내리라고 예상한다.

참 고 문 헌

[1] S. Walters, "Computer-aided prototyping for ASIC-based systems," IEEE Design and Test of Computers, Vol.8, Issue.2, June, 1992.

[2] Bondyopadhyay, P. K., "Moore's law governs the silicon revolution," in Proc. of the IEEE, Vol.86, Issue.1, pp.78-81, Jan. 1988.

[3] 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, 1993.

[4] Jonathan Babb, Russel Tessier, and Anant Agarwal, "Virtual wires : Overcoming pin limitations in FPGA-based logic emulators," in Proc. IEEE Workshop on FPGA-based Custom Computing Machines, pp.142-151, April, 1993.

[5] Van den Bout. D. E, Morris. J. N, Thomae. D, Labrozzi. E, WinGo. S, Hallman. D., "AnyBoard : An FPGA-Based Reconfigurable System," IEEE Design and Test of Computers, Vol.9, Issue.3, pp.21-30, Sept. 1992.

[6] Courtoy, M., "Rapid system prototyping for real-time design validation," in Proc. 1988 Ninth Int. Workshop on Rapid System Prototyping, pp.108-112, 1998.

[7] Quickturn Design System, Inc., "Emulation System with time-multiplexed interconnect," US Patent 005960191, May, 1997.

[8] B. S. Landman, R. L. Russo, "On a pin versus block relationship for partitions of logic graphs," IEEE Trans. on comput., Vol.C-20, pp.1469-1479, 1971.

[9] 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.

[10] 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.

[11] Selvidge, C., Agarwal, A., Dahl, M., Babb, J., "TIERS : Topology IndependEnt Pipelined Routing and Scheduling for VirtualWire Compilation," in Proc. ACM/SIGDA Int. Symp. Field Programmable Gate Array, pp.25-31, 1995.

[12] Giovanni De Micheli, Synthesis and optimization of digital circuits, McGraw-Hill, Inc., pp.207-211, 1994.

[13] Thomas, E., Morton, David, W., Pentico, Heuristic scheduling systems, John Wiley & Sons, Inc., 1993.

[14] Xilinx Databook 2001 http : //www.xilinx.com/partinfo/databook.htm.

김 남 도

e-mail : ndkim@hyowon.pusan.ac.kr

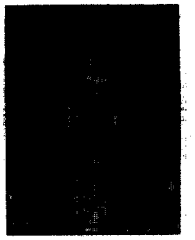
1994년 부산대학교 컴퓨터공학과 졸업 (학사)

1996년 부산대학교 대학원 컴퓨터공학과 (공학석사)

1998년 부산대학교 대학원 컴퓨터공학과 (공학박사 수료)

1998년~현재 부산대학교 컴퓨터공학과 박사과정

관심분야 : 논리합성, VLSI CAD, logic emulator 등



양 세 양

e-mail : syyang@hyowon.pusan.ac.kr

1981년 고려대학교 전자공학과 졸업(학사)

1985년 고려대학교 컴퓨터공학과(공학석사)

1990년 University of Messachusetts(공학박사)

1991년~현재 부산대학교 컴퓨터공학과 부교수

관심분야 : VLSI CAD, system verification 등