

## 테스트 및 제조

# ASIC 코어의 검증 위한 대규모 디자인 에뮬레이션

By Alan Singletary  
Advisory Engineer  
IBM Microelectronics

회로내 에뮬레이션은 최초 합격 ASIC의 품질을 높이기 위한 효과적인 기법이다. 미국 텍사스주 오스틴시에 소재한 IBM Microelectronics社 설계팀에서는 몇몇 복잡한 코어의 검증에 이 기술을 시험하였다. 이 설계팀이 이용한 에뮬레이션 기법은 디자인을 몇 개의 커다란 FPGA 칩으로 분할한 다음, 이 에뮬레이트 로직을 기성품 칩과 결합해 전체 시스템을 완성하는 것이다.

내장 컨트롤러 주변장치 설계팀이 에뮬레이션 기술을 이용하여 IBM Microelectronics社의 블루 로직 라이브러리로 용으로 다수의 ASIC 코어를 개발했다. IBM社의 디자이너들이나 외부 고객들이 모두 이용할 수 있는 이 코어들은 프로세서 로컬 버스(PLB)나 온칩 주변장치 버스(OPB)와 같은 이 회사의 코어연결 온칩 버스를 이용하여 개발되었다.

디자인 품질을 높이고 코어에 대한 고객들의 요구를 충족시키기 위해 설계팀은 최근 검증 전략에 에뮬레이션 기법을 추가했다. 에뮬레이션을 통해 검증된 최초의 코어 제품이 1998년 초 블루 로직 라이브러리에 추가되었다. 이 기

사에서는 이들 코어를 제조하는 데 이용된 설계 과정과, FPGA 기반 에뮬레이션 기술을 시험할 때 유용한 몇 가지 요소들에 대해 살펴 본다. 에뮬레이션/프로토타이핑 기술은 Aptix社 것을 선택했기 때문에 이 기술과 관련된 문제들을 중점적으로 살펴 보자.

### 에뮬레이션의 이점

에뮬레이션의 이점에 대해 살펴 보기 전에, 경제적 측면 뿐만 아니라 그 밖의 측면에서도 이 기술의 한계와 비용에 대해 이해해야 할 것이다. 여기서 설명하는 에뮬레이션 기법이 경제적인 대안 중의 하나이기는 하지만, 장비 비용과 엔지니어링 작업도 만만치 않다. 그러한 엔지니어링 작업으로서, ASIC 설계팀들의 에뮬레이션별 작업이 어느 정도 있을 수 있고 디자인 자체를 다소 변경해야 할 수도 있다.

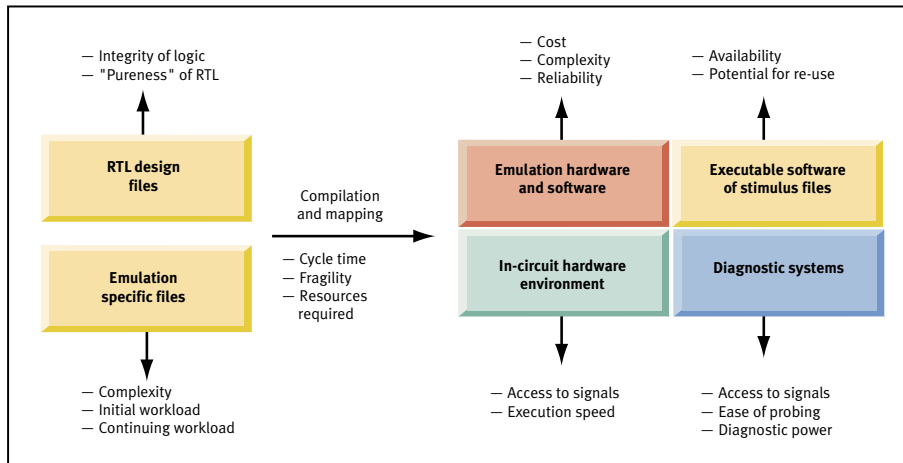


그림 1: 에뮬레이션 과정의 각 단계에서 여러 가지 단계의 기준들을 고려해야 한다.

뿐만 아니라 회로내 에뮬레이션 환경에서 정확히 디자인의 어느 부분이 시험되는지 이해하기가 어렵다. 에뮬레이트 디자인이 실제 소프트웨어 및 주변 장치와 동작하는 것을 보고 설계팀이 상당히 자신감을 얻기도 했지만, ASIC의 동작 공간이라는 측면에서 얻은 성과는 크지 않았다. 일부 유형의 버그를 찾아내고, 격리시키고, 교정하고, 검증하는 작업 또한 기존의 시뮬레이션 툴에 비해 더 많은 시간을 소모할 수 있다. 버그 문제는 주로 회로내 하드웨어 및 소프트웨어와 테스트 대상 로직 간의 복잡한 상호 작용에 기인한다. 여기에서 매우 높은 클럭 사이클이 문제를 심화시킨다. 많은 경우 에뮬레이션은 시뮬레이션 테스트보다는 시스템 수준 검증이나 디버깅과 더 많은 공통점을 갖는다. 이러한 이유에서 에뮬레이션은 그 자체로서 완벽한 검증 수단이 아니라 시뮬레이션의 효과적인 보완 수단으로 이해되어야 한다.

이러한 문제에도 불구하고 하드웨어 에뮬레이션은 여러 특수한 검증 및 개발 작업을 가능케 한다. 에뮬레이션을 통해 이용할 수 있는 높은 사이클과 실제적인 시스템 환경의 예측 불가능한 특성은 시뮬레이션으로 해결할 수 없는 현실적 시나리오 검증에 효과적이다.

### 기술의 평가

내장 컨트롤러 주변장치 설계팀에서는 세 가지 에뮬레이션 기법을 검토했다. IBM社 연구원들은 에뮬레이션 아키텍처의 선택시 여러 요인들을 고려했는데, 대부분의 아키텍처는 몇 가지 단순한 기준에 의해 제외되었다.

먼저 연구진은 디자인을 다수의 FPGA 또는 주문형 프로세서로 자동 분할하는 블랙박스 시스템을 검토했다. 이 아키텍처의 가장 큰 장점은 아주 큰 디자인을 처리할 수 있다는 것이다. 반면 이 시스템은 비용이 너무 비싼 것으로 판단되었다. 뿐만 아니라 상대적으로 낮은 실행 속도가 다양한 표준 주변 장치와의 회로내 동작 가능성을 제한할 수 있다는 염려가 제기되었다.

둘째, 디자인을 작은 토막으로 분할하고 각각의 토막을 FPGA를 탑재한 주문형 회로 기판을 이용해 에뮬레이트하는 기법을 검토했다. 이 기법은 IBM사의 다른 프로젝트에 성공적으로 이용되었으며 비용도 경제적이고 실행 속도도 높았다. 단점은 FPGA 간의 인터커넥트가 고정배선화되어 있어 FPGA 입출력 경계가 고정되어 있어야 한다는 점이다.

셋째, Aptix사의 MP4 FPCB(field programmable circuit board)을 검토하였다. 이 24 x 18 기판은 Aptix사 FPGA 모듈을 연결할 수 있는 플러그인 구역을 갖고 있다. 외부 장치를 이 플러그인 구역에 직접 연결하거나 기판 가장자리의 입출력 연결 구역을 통해 연결할 수 있다. 이 아키텍처의 가장 큰 특징은 기판 중심에 네 개의 FPIC(field programmable interconnect component) 칩이 있다는 것이다. 이들 SRAM 기반 칩을 재빨리 프로그래밍하여 936 개의 입출력중 두 개 이상을 연결할 수 있다. 이러한 연결들은 전기적으로는 수동형의 저항-정전용량성 부하로 보이며, 2 지점간 연결시에 일반적인 5(15 ns의 지연을 갖는다.

### 에뮬레이션의 원리

모든 에뮬레이션 전략에 동일한 기본 절차가 포함된다. 하지만 자세히 살펴보면 하드웨어 아키텍처, 디자인 구성, 대상 애플리케이션에 있어서 크게 다르다. 세 경우 모두 컴파일 과정에서 ASIC의 하드웨어 설계 언어(HDL) 코드가 에뮬레이션별 라이브러리와 결합되어 에뮬레이션 플랫폼에 해당하는 이미지를 작성한다.

에뮬레이션 작업이 효과적이기 위해서는 에뮬레이트되는 로직이 시뮬레이션에서 실행되는 로직 및 최종 칩과 기능적으로 동일해야 한다. Aptix사의 소프트웨어 공정은 원래의 디자인 데이터베이스에서 데이터를 인출함으로써 이러한 동일성을 유지한다. 각각의 FPGA 칩은 데이터베이스에서 필요한 HDL 모듈을 사래화하는 랩퍼(wrapper) 파일을 필요로 한다. FPGA 랩퍼 파일에는 양방향 드라이버와 같은 필요한 인터페이스 로직이 포함된다.

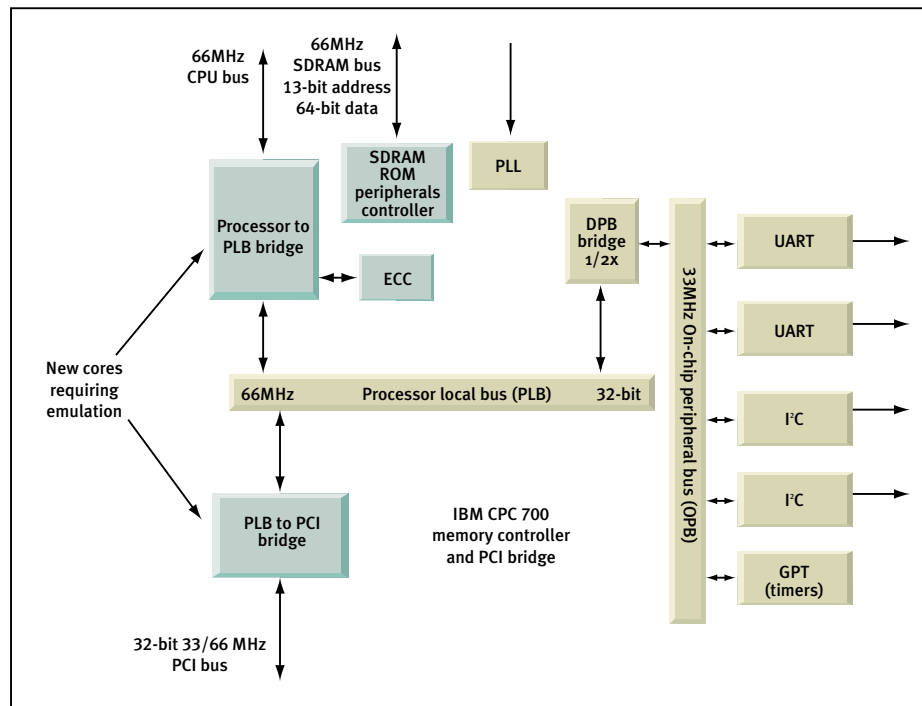


그림 2: 다섯 개의 FPGA 소자를 갖춘 32비트의 PLB 내 PCI 인터페이스가 에뮬레이션에 사용된다.

FPGA 간의 연결은 로직이 추가되지 않은 순수한 네트리스트인 에뮬레이션 상위 파일에서 정의된다. 에뮬레이션 기판에 FPGA가 아닌 다른 칩을 포함시키면 그 칩의 입출력 구성을 정의하는 HDL 모듈을 작성해야 한다.

에뮬레이션의 다음 과정은 해당 FPGA 기술을 겨냥해서 표준적인 합성 절차를 통해 디자인 파일과 FPGA 랩퍼 파일을 동시에 실행하는 것이다. 합성 이후에는 실제 과정이 몇 단계로 나뉜다. 소프트웨어는 디자인의 FPGA 별 네트리스트 파일과 동작한다. 매핑 프로세스에서부터 시작해 소프트웨어가 입력 파일이 모순되지 않는지 검사한다. 그 다음 클럭 계획을 결정하고 FPGA 핀의 위치를 정한다.

소프트웨어는 그 다음 FPGA를 배치 및 배선하기 위한 제약조건 파일을 작성한다. 이 시점에서 소프트웨어가 대상 FPGA를 위한 해당 업체 툴을 실행한다. 이 툴이 모든 칩을 성공적으로 배치 및 배선하면 마지막 단계는 회로 기판 배선을 작성하는 것이다.

이러한 설계 흐름에는 약간의 복잡성과 오류 가능성이 내포되어 있다. 디자인 파일이 서로 다른 여러 소프트웨어 패키지를 왔다갔다 하기 때문이다. 이러한 모든 인터페이스를 잘 정의하기만 한다면 프로그램들은 서로를 거의 인식하지 않게 된다. 일관성을 정확히 유지하기 위해서는 세심한 주의가 필요하다. 어떤 한 부품을 변경하면 다른 부품에도 영향이 미치기 때문이다.

### 3개 코어의 에뮬레이션

IBM사의 에뮬레이션 프로젝트는 1998년 3/4분기에 PowerPC 603 CPU를 채택한 내장 분야에 이용하기 위한 3개 코어에 대해 시작되었다. 이 디자인은 그 밖에도 PowerPC 60X 프로세서 로컬 버스 메모리 컨트롤러 인터페이스, 동기식 DRAM 및 ROM 컨트롤러를 이용한 64비트 메모리 컨트롤러, 32비트 PLB 내 PCI 인터페이스를 포함했다.

이 에뮬레이션 프로젝트의 최대 목표는 코어를 가장 실제적인 시스템 환경에서 시험하는 것이었다. 이러한 목표를 달성하기 위해 코어를 완벽한 전기능 시스템의 일부로 구현하였다. 시스템의 CPU와 DRAM이 Aptix사 기판에 연결된 주문형 회로 기판에 상주하는 한편, 에뮬레이트 코어(여러 개의 FPGA)가 이들 부품과 PCI 버스 간에 브리지를 제공했다. IBM사 네트워크 컴퓨터의 표준 주기판을 수정한 것을 PCI 버스를 통해 연결했다. 이 기판이 네트워크 접속을 포함해 PCI 및 ISA 주변장치를 액세스할 수 있도록 한다.

Aptix사 에뮬레이션 아키텍처가 출시되어 있는 다른 아키텍처와 가장 다른 점은 개별 FPGA로 구현하기 알맞게 디자인을 수작업으로 여러 개의 모듈로 분할해야 한다는 점이다. 어떤 에뮬레이션 기법의 경우에는 사용자가 관여하지 않고 소프트웨어 툴이 전체 디자인을 여러 개의 작은 FPGA로 분할한다.

수작업 분할은 자연스러운 경계를 따라 분할하는 등의 이점이 있기도 하지만 엔지니어링 시간과 숙련도를 필요로 한다. 어떤 디자인은 FPGA 자원에 따라 이용할 수 있는 분할 크기가 제한적일 수 있다. FPGA 게이트의 수와 속

도는 놀라운 속도로 증가하고 있지만 입출력의 수는 그렇지 않다. 입출력 수가 분할 병목 지점이 될 수 있기 때문에 이것이 문제가 될 수 있다. 하지만 오늘날 대규모의 표준 FPGA 패키지에서 이용할 수 있는 400 개 내지 (440 개)의 입출력은 코어 또는 모듈 경계를 따라 직접 분할이 가능토록 한다.

FPGA 간에 신호를 전송할 때의 이점은 신호를 손쉽게 디버깅 목적으로 이용할 수 있다는 점이다. 시스템에서 FPIC를 경유하는 모든 신호는 Hewlett-Packard社의 로직 분석기 포트에 연결된 외부 FPIC에 이용할 수 있다. 소프트웨어가 탐침검사할 신호를 선택할 수 있도록 하고, 직렬 또는 이더넷 인터페이스를 통해 로직 분석기 채널을 자동으로 프로그램한다.

이 기술과 관련된 또 다른 하드웨어 문제는 신호 지연과 시스템 동작 주파수이다. 지연 문제로는 FPGA 내에서의 종합적인 로직 지연, 신호가 FPGA 경계 상에서 이동하는 데 필요한 시간, FPIC를 통한 지연 횟수와 길이, 단일 네트의 긴 배선 토폴로지 등을 들 수 있다. 최상의 경우 단일 네트는 하나의 FPIC 경로만을 필요로 하며 일반적인 지연은 10 MHz 이상의 동작 주파수를 발생시킨다.

## 하드웨어와 소프트웨어의 통합

에몰레이트 코어와 기타 부품을 테스트하기 위해 두 개의 코드를 이식하고 에몰레이션에 이용할 수 있도록 수정했다. 예를 들어 입출력 주기판용 부트 펌웨어의 경우에는 타이밍 종속성을 일부 제거하였다. 이 펌웨어가 SDRAM 컨트롤러 코어 테스트를 위한 플랫폼으로 이용되었다. 이 에몰레이션 플랫폼은 비교적 속도가 높아서 시뮬레이션에서는 불가능했을 테스트를 메모리 컨트롤러 타이밍을 바꿔가면서 손쉽게 수행할 수 있다.

대부분의 테스트 기법이 BUD (bring-up driver)라고 하는 IBM社의 하드웨어 연습기 프로그램을 이용했다. 이 프로그램은 수년 동안 PowerPC 기반 칩과 시스템 검증에 이용되어 왔다. BUD를 이용하여 모든 주변 장치에 동시에 장시간 연습기 프로그램을 실행함으로써 디자인에 상당한 피로를 누적시킬 수 있었다.

이 에몰레이션 프로젝트는 진화론적 방식으로 약 6개월 동안 계속되었다. 디자인을 레이아웃 단계로 넘기기 바로 직전에 새 디자인 기능과 테스트를 추가했다. 돌이켜보면 이 방식은 설계 주기 양 끝에서 에몰레이션을 효과적으로 이용하지 못한 것 같다. 로직이 미성숙한 상태에서는 에몰레이션 환경에서 버그를 격리시키는 것이 썩 효과적인 방법은 아니다. 다양한 변수가 개입되기 때문이다.

에몰레이션은 시뮬레이션을 효과적으로 보완할 수 있는 것으로 입증되었다. 에몰레이션 시스템을 이용해 몇몇 심각한 버그를 찾아낼 수 있었다. 로직이 어떤 레지스터 셋업을 실행하기 전 수천 클럭 사이클 동안 실행될 때만 발생하는 것과 같은 스타트업 문제가 그러한 버그이다.

그 밖의 버그로는 리프레시 사이클과 관련된 메모리 컨트롤러 문제가 있었다. DRAM 리프레시가 공칭 주파수로 실행되는 시스템에 비해 더 많은 메모리 대역폭을 차지했기 때문에 에몰레이션 환경이 이러한 작업에 대해 매우 피로한 상태가 되었다. 어떤 경우에는 이러한 문제를 발견함으로써 타이밍 문제를 더 잘 해결할 수 있도록 시뮬레이션 환경을 개선할 수 있었다.

어떤 ASIC 검증 프로젝트에서나 실제로 칩을 장착하기 전까지 선부른 판단은 금물이다. 최초의 실리콘 구현시 (IBM CPC 700 메모리 컨트롤러 및 PCI 브리지) 코어에 심각한 문제가 발견되지 않았으며, 첫 패스에서 합격한 칩들을 출하할 수 있었다. 추가적으로 이 에몰레이션 플랫폼에서 실시된 소프트웨어 이식 및 테스트 개발이 BUD에 직접 적용할 수 있는 것으로 판단되어 디자인 품질을 신속하게 검증할 수 있었다. 그 결과 BUD 시스템 소프트웨어를 실행시킨 최초의 칩이 사흘 내에 작동하기 시작했다.

이러한 성공을 거둔 후 설계팀은 다섯 개의 새로운 코어와 40만 게이트를 에몰레이트하는 보다 거대한 프로젝트에 대해 동일한 에몰레이션 기법을 이용했다. 기대 수준이 적절하고 사전에 필요한 계획이 이루어지기만 한다면, 실행 속도가 높고 회로내 동작이 뛰어난 FPGA 기반 에몰레이션이 제조에 앞서 ASIC 코어를 검증할 수 있는 무한한 가능성을 제공할 것이다. □