

# 재구성 가능 SoC를 위한 시스템 수준 설계공간탐색 도구

\*안성용<sup>○</sup> \*\*심재홍 \*이정아  
\*조선대학교 컴퓨터공학부  
\*\*조선대학교 인터넷소프트웨어 공학부  
{dis<sup>○</sup>, jhshim, jalee}@chosun.ac.kr

## A System Level Design Space Exploration Tool for a Configurable SoC

\*SeongYong Ahn<sup>○</sup> \*\*JeaHong Shim \*JeongA Lee  
\*School of Computer Engineering, Chosun University  
\*\*School of Internet Software Engineering, Chosun University

### 요 약

멀티미디어 데이터 처리나 암호화 알고리즘과 같은 계산량이 많고 빠른 시간 안에 처리되어야 하는 어플리케이션들을 처리하기 위하여 재구성 가능한 논리소자와 내장형 마이크로 프로세서등이 하나의 칩에 통합된 재구성 가능한 SoC가 폭넓게 활용되고 있다. 이러한 컴퓨팅 환경의 시장적응성을 높이기 위해서는 프로토타입을 제작하기 전에 설계변수에 따른 성능수치를 이미 예측하여 최소의 비용으로 시스템의 수행 시간 및 자원제약사항을 만족할 수 있는 구조를 찾아내는 것이 필수적이다. 본 논문에서는 Y-chart 설계 방법의 기본 개념을 재구성 가능한 SoC에 적용가능하도록 확장하여, 시스템 수준의 설계공간 탐색 도구를 개발하였다. 구현된 설계 공간 탐색을 통한 시뮬레이션 결과는 시스템 설계자들에게 실제 프로토타입을 구축하지 않고 최적의 설계변수를 결정할 수 있게 하여 설계시간과 설계비용을 현저하게 줄여줄 것으로 기대된다.

### 1. 서 론

정보와 통신의 융합에 따른 급격한 기술 발달로 다양한 어플리케이션을 수행할 수 있는 시스템 설계 환경의 중요성이 최근에 급격히 부각되어 왔다. 전통적인 시스템 설계 방식은 주어진 하나의 응용 알고리즘을 수행하기 위하여, 하나의 특정 아키텍처를 고려하여 시뮬레이터를 통하여 함수측면의 성능을 확인한 후 전통적인 ASIC 설계 방식에 의존하여 시제품을 구현한다. 이 전통적인 방식은 논리 합성(logic synthesis) 및 회로 합성을 통하여 시제품을 구현하고 이의 반복적인 조정을 통하여 개발하기 때문에 제품 개발에 소모되는 시간이 길어져서, 시장이 요구하는 시간 이내에 제품을 개발하지 못하는 위험도가 높게 임은 물론 프로토타입을 개발하는 데에 추가적인 비용을 지출하게 된다.

이 외에도, 새로운 응용서비스를 추가하고자 하는 경우, 이전에 개발된 시스템을 최대한 활용하기 위한 분석이 기존의 설계환경에서는 거의 불가능하다. 이러한 문제점들을 보완하기 위해서 근간에는 하드웨어 소프트웨어 통합설계 방법을 이용하고 있다. 이러한 통합설계방법에서는 전통적인 ASIC 설계방법과는 달리, 주어진 시스템의 기능을 분화하여 하드웨어로 처리할 수 있는 부분과 소프트웨어로 처리할 수 있는 부분으로 나누어서 구현하고, 인터페이스를 사용하여 통합하는 과정을 거치게 된다.

이 과정에서 특히 하드웨어로 구현되어야 할 부분을 어느 정도의 유연성과 빠른 처리속도를 낼 수 있는 FPGA와 같은 재구성 가능한 부품을 사용하는 추세이다. 또한 SoC기술의 발달에 따라 이러한 부품들이 하나의 칩에 구현되는 재구성 가능한(Configurable) SoC가 차세대 내장형 시스템을 위한 하드웨어 플랫폼으로 각광받고 있다 [1,4,5,6].

이렇게 재구성 가능한 SoC를 사용함으로써 설계의 유연성이 증대되는 효과를 얻을 수 있지만 설계상의 구현 조건과 제약조건을 만족하는 구조를 찾아야 하는 문제가 여전히 남아있다. 이러한 문제점을 해결하기 위하여 시스템 설계의 초기단계에서 주어진 구현조건 및 제약조건을 만족하는 다양한 구조를 살펴보고, 이에 따른 최종결과물의 성능을 신뢰성 있는 정확도를 가지고 비교, 평가, 예측할 수 있는 시스템 개발 환경의 고급화 필요성이 대두되었다. 이러한 시스템 개발환경에서 핵심적인 요소로 간주 될 수 있는 부분이 시뮬레이션 관련 기술이며, 시스템 개발 제약 조건을 만족시키는 최적의 설계변수를 찾을 수 있는 시스템 설계 공간 탐색 고급 기술이다[2].

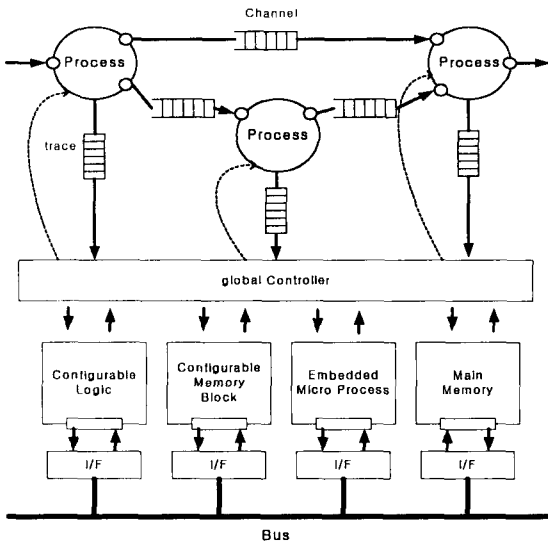
본 논문에서는 시스템의 성능분석 도구로 사용되는 Y-chart를 확장하여 시스템 수준의 설계공간탐색 도구를 개발하였다.

본 논문의 구성은 다음과 같다. 2장에서는 Y-chart를 기반으로 하는 재구성 가능한 시뮬레이터에 대하여 설명하고, 3장에서는 설계공간 탐색 환경을 소개하며 4장에서는 실험결과를 보여주고, 5장에서 결론을 기술한다.

### 2. 재구성 가능한 시뮬레이터

본 논문에서 구현된 시스템 수준의 설계 공간 탐색 도구는 재구성 가능한 SoC 설계과정에서 실제 프로토타입을 구축하지 않고 구현될 시스템의 성능을 예측할 수 있도록, 기존의 Y-chart설계환경을 보완하여 개발하였다. 구현된 재구성 가능한 시뮬레이터는 주어진 어플리케이션을 시뮬레이션하는 부분과 하드웨어를 시뮬레이션하는 부분 그리고 어플리케이션과 하드웨어를 사상(Mapping)시켜주는 사상제어기로 구성되어있다. 어플리케이션과 하드웨어간의 의미론적인 차이를 최대한 줄이기 위하여

어플리케이션의 입력은 디지털 신호처리의 모델링 방법으로 널리 사용되는 Kahn 프로세스 네트워크로 전제하였다[3]. 또한 재구성 가능한 하드웨어인 FPGA 또한 데이터 흐름(data-flow)를 활용한 스트림 기반 하드웨어로 구성되는 것을 전제함으로써 어플리케이션과 하드웨어의 사상을 용이하게 하였다. 전체적으로, 구현된 시뮬레이터는 어플리케이션 시뮬레이터가 발생시키는 트레이스를 사상제어기가 해당되는 CPU 또는 재구성 가능한 장치에 할당하는 Trace-Driven 시뮬레이션 방법을 사용하여 구현되었다. 그림 1은 구현된 재구성 가능한 시뮬레이터를 보여 준다.



(그림 1) 재구성 가능한 시뮬레이터의 구조

3. 설계 공간 탐색 도구

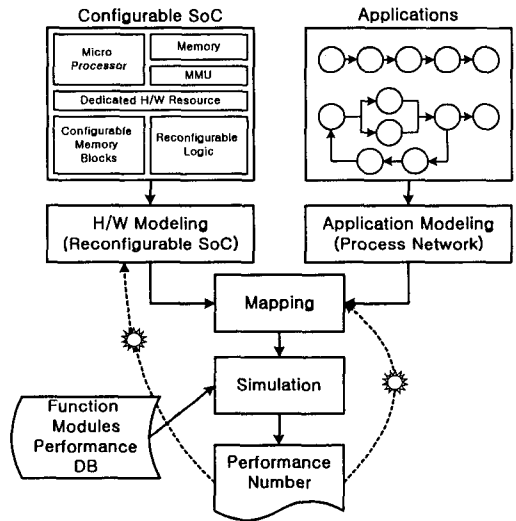
재구성 가능 SoC를 설계하기 위해서는 상위 수준의 HW/SW 통합설계(Co-Design)와 통합 설계 검증 등의 과정이 필요하다. 이러한 과정에서 설계공간 탐색 기술은 응용프로그램을 수행할 SoC의 구조를 결정짓는 핵심 기술이다. 본 논문에서 구현된 설계공간 탐색 도구는 그림 2에서 보인 바와 같이 응용 프로그램 모델링과 하드웨어 모델링, 그리고 목표 하드웨어 시스템에서의 성능을 산출하기 위한 시뮬레이션으로 구성되어 있다. 재구성 가능 SoC는 기존의 SoC 요소인 마이크로 프로세서 코어 IP, 메모리, 특정용도 HW자원 이외에 구성정보 메모리 블록(Configurable Memory Block)과 이를 이용하는 재구성 가능한 로직이 집적되어 있다. 하드웨어 모델은 SoC에 사용되는 프로세스와 FPGA와 같은 재구성 가능한 장치들의 특성과 버스구조의 특성을 고려하여 모델링된다. 재구성 가능한 SoC 위한 기본적인 구조는 2장에서 설명한 바와 같이 재구성 가능한(Retargetable) 시뮬레이터에 구현되어 있으며 사용자는 FPGA의 자원 량, 구현될 수 있는 기능요소, 각 기능요소들이 필요로 하는 자원 량만을 입력하면 된다. 이러한

방법으로 시뮬레이션 환경이 완성되면 하위 작업들이 어떤 하드웨어 자원에 사상이 되는가를 명시하는 분할 테이블(Partitioning Table)에 따라 시뮬레이션을 진행한다. 먼저 특정 어플리케이션을 하위 작업별로 나누고 소프트웨어적으로 마이크로 프로세서에서 수행하는 경우와 하드웨어적으로 재구성 가능한 로직(eg. FPGA)에서 수행하는 경우에 대하여 수행시간 분석을 수행한다. 분석 결과는 데이터 베이스 형태로 저장된다. 시뮬레이션은 하드웨어/소프트웨어 사상의 경우를 바꾸거나 재구성 가능 SoC에 적용할 수 있는 설계변수를 변경시켜가며 반복하여 진행한다.

$$Parallellism = \frac{Total\ Execution\ Time\ of\ the\ Workloads}{T_{end}}$$

$$Utilization = \frac{Time\ a\ Resource\ is\ Used}{T_{end}} \times 100\%$$

성능수치로 사용되는 병렬성(Parallellism), 하드웨어자원들의 활용도(Utilization)는 위의 수식에 의해서 구해진다. 수식에서 T\_end는 패킷처리 시뮬레이션 종료시간을 의미한다. 병렬성은 현재의 시스템이 평균적으로 한 순간에 몇 개의 작업을 처리하고 있는가를 보여주는 성능 수치로 각각의 작업들이 대기시간을 제외하고 실제로 수행되었던 시간들의 총 합계를 시뮬레이션 종료시간으로 나누어서 구해진다. 활용도는 특정한 하드웨어 자원이 얼마나 사용되었는가를 보여주는 성능수치로 각각의 자원들이 사용된 시간을 시뮬레이션 종료시간으로 나누어 구해진다.

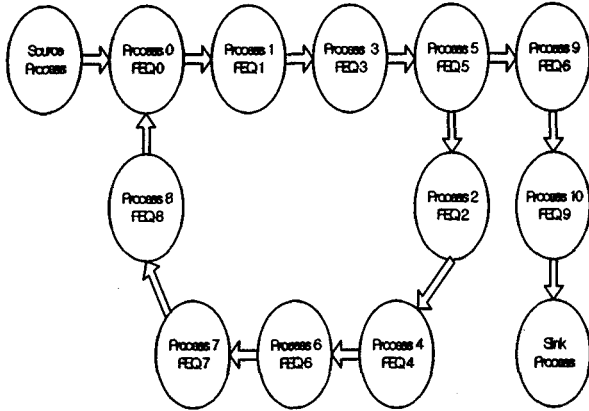


(그림 2) 설계 공간 탐색 도구의 구조

4. 실험

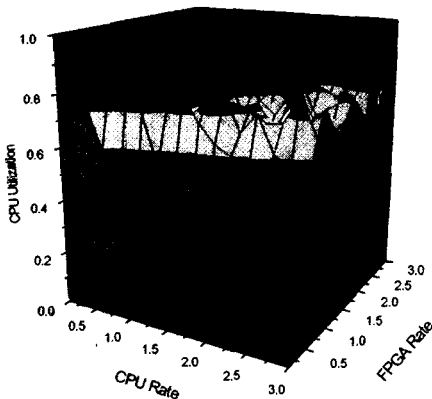
3장에서 설명된 설계 공간 탐색 환경을 적용한 실험을 위하여 그림 3에서 표현된 것과 같은 임의로 어플리케이션 모델을 사용하였다. 그림 3에서 FEQ(Function

Element Request)는 해당되는 프로세스가 실행 시 요구하는 기능요소(Function Element)를 식별하기 위한 것이다.



(그림 3) 실험에 사용된 어플리케이션 모델의 Kahn Process Network 표현

그림 4는 재구성 가능한 SoC에 포함된 하드웨어 자원 중 설계과정에서 선택 가능한 설계 변수 중 마이크로 프로세서와 재구성가능한 장치(FPGA)의 속도에 따른 CPU 활용도의 변화를 그래프로 나타낸 것이다. 그래프에서 나타난 CPU Rate과 FPGA Rate 각 하드웨어의 속도는 정형화 한 것으로 수치 1.0은 기본 Rate에 해당한다. 그림에서 보이는 것과 같이 CPU의 활용도(Utilization)은 CPU Rate이 1.0이상 되면 0.8 이하로 떨어지는 것을 관측할 수 있다. 이는 전체적으로 FPGA가 아주 느린 경우를 제외하고는 CPU 속도가 시스템의 성능에 지대한 영향을 미치게 되는데 CPU 속도가 어느 정도(실험결과에서는 1.0)이상 보장해주었을 때 CPU의 부하가 적절히 분산되는지 예측할 수 있다.



(그림 4) CPU속도와 FPGA 속도에 따른 CPU 활용도

5. 결론

본 논문에서는 FPGA와 같은 재구성 가능한 장치와 일반적인 CPU를 결합하여 재구성 가능 SoC를 위한 설계 공간탐색을 위한 도구를 설계하고 구현하였다. 구현된 도구는 특정 어플리케이션에 대한 재구성 가능 SoC의 성능수치를 산출하기 위한 시뮬레이터이고, 컴퓨팅 자원 스케줄링문제를 고려하기 위해서 Trace-driven 시뮬레이션 방법을 사용하고 설계과정에서 선택적으로 적용 가능한 설계변수들의 변화를 고려한 시뮬레이션이 가능하도록 구현되었다. 본 논문에서 구현된 설계공간탐색도구는 재구성 가능 SoC를 기반으로 하는 시스템을 설계하는 과정에서 실제 프로토타입을 개발하지 않고도 개략적인 성능을 미리 예측할 수 있도록 하여 설계 비용과 시간을 현저하게 줄여주고, 최적의 하드웨어 구성을 검색하게 해주는 설계공간탐색기술의 핵심 모듈로 활용될 것으로 기대된다.

참고문헌

- [1] E. Sanchez, M. Sipper, J.-O. Haenni, J.-L. Beuchat, A. Stauffer, and A. Perez-Urbe, "Static and Dynamic Configurable Systems," IEEE Transactions on Computers VOL.48, No.6, June 1999.
- [2] B. Kienhuis, E. Deprettere, K.A. Vissers, and P. Wolf. "An approach for quantitative analysis of application-specific dataflow architectures," In Proceedings of 11th Intl. Conference of Applications-specific Systems, Architectures and Processors (ASAP'97), pages 338-349, Zurich, Switzerland, 1997
- [3] G. Kahn, "The semantics of a simple language for parallel programming," Info. Proc., pages 471-475, Stockholm, Aug. 1974
- [4] J. Vullemin, P.Bertin, D.Roncin, M.Shand, H. Touati, P. Boucard, "Programmable Active Memories: Reconfigurable Systems Come of Age," IEEE Transactions on VLSI Systems, Vol. 4, No.1, pp56-69, March 1996
- [5] Andre DeHon. "The Density Advantage of Configurable Computing," IEEE Computer, 33(4):41-49, April 2000.
- [6] Katherine Compton, Scott Hauck, "Reconfigurable Computing: A Survey of Systems and Software," to appear in ACM Computing Surveys, 2002