

# Hybrid 시스템을 위한 확장된 Y-chart를 이용한 설계 공간 탐색 환경

안성용, 이정아  
조선대학교 컴퓨터공학부

## Design Space Exploration Environment for Hybrid Systems based on Extended Y-chart

Seong-Yong Ahn, Jeong-A Lee  
Department. of Computer Engineering, Chosun University  
E-mail : dis@rain.chosun.ac.kr, jeong@rain.chosun.ac.kr

### 요약

멀티미디어 데이터 처리나 암호화 알고리즘 같은 계산량이 많고 빠른 시간안에 처리되어야하는 어플리케이션들을 처리하기 위하여 최근의 컴퓨팅 환경은 재구성가능한 시스템과 일반적인 마이크로 프로세서가 결합된 시스템을 폭넓게 활용하고 있다. 이러한 시스템의 시장적용성을 높이기 위해서는 프로토타입을 제작하기 전에 설계변수에 따른 성능수치를 이미 예측하여 최소의 비용으로 시스템의 수행시간 및 자원제약사항을 만족할 수 있는 구조를 찾아내는 것이 필수적이다. 본 논문에서는 Y-chart 설계방법의 기본 개념을 Hybrid 시스템에 적용가능하도록 확장하여, 재구성가능한 시뮬레이터를 개발하였고, 이를 기반으로 H.263 인코더 모델을 어플리케이션모델로 하고 FPGA와 일반적인 프로세서를 사용하는 가상의 시스템을 하드웨어 모델로하여 설계공간탐색을 진행하였다. 설계공간 탐색을 통한 시뮬레이션 결과는 시스템 설계자들에게 실제 프로토타입을 구축하지 않고 최적의 설계변수를 결정할 수 있게 하여 설계시간과 설계비용을 현저하게 줄여줄 것으로 기대된다.

### 1. 서론

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 용용 알고리즘을 수행하기 위하여, 하나의 특정 아키텍처를 고려하여 시뮬레이터를 통하여 합수측면의 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, 시장

이 요구하는 시간이내에 제품을 개발하지 못하는 위험도가 높게 됨은 물론 프로토타입을 개발하는데에 추가적인 비용을 지출하게 된다.

이 외에도, 새로운 응용서비스를 추가하고자 하는 경우, 이전에 개발된 시스템을 최대한 활용하기 위한 분석이 기존의 설계환경에서는 거의 불가능하다. 이러한 문제점을 보완하기 위해서 근간에는 하드웨어 소프트웨어 통합설계 방법을 이용하고 있다. 이 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 FPGA

본 연구는 한국과학재단 특별기초 연구과제(과제번호: 98-0101-04-01-3) 연구비에 의해 연구되었음

와 같은 재구성 가능한 부품을 사용하는 추세이다.[1][2][4][6]

이렇게 재구성 가능한 FPGA와 같은 부품들을 사용함으로써 설계의 유연성이 증대되는 효과를 얻을 수 있지만 설계상의 구현조건과 제약조건을 만족하는 구조를 찾아야 하는 문제가 여전히 남아있다. 이러한 문제점을 해결하기 위하여 시스템 설계의 초기 단계에서 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.

본 논문에서는 시스템의 성능분석 도구로 사용되는 Y-chart를 확장하여 재구성가능한 시뮬레이터를 개발하였다. 설계 공간 탐색을 수행하기 위하여 하드웨어 자원과 이 하드웨어를 이용한 응용프로그램을 선택하여야 한다. 응용프로그램은 어플리케이션 모델에서 분석되는데 본 논문에서는 H.263모델을 선택하였다. 또, 하드웨어 모델로, 하나의 일반적인 프로세서와 하나의 재구성가능한 FPGA를 사용하는 Hybrid 시스템을 선택하여 설계 공간 탐색을 수행하였다.

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

## 2. 재구성가능한 시뮬레이터

본 논문에서 다루는 Hybrid 시스템은 여러 다양한 하드웨어 구성방식 중, 두 가지 형태의 아키텍처의 혼합을 전제한다. 첫 번째 형태는 기존의 고정된 아키텍처인 CPU이고 두 번 째 형태는 응용프로그램에 따른 하드웨어 재구성이 가능한 FPGA로 구성되어 있다고 전제한다. 단일 응용프로그램을 효율적으로 수행하기 위한 ASIC을 개발하는 것과 달리 일련의 연관된 특정 응용프로그램들을 효율적으로 수행할 수 있는 하드웨어 구성을 찾기 위하여는 다양한 설계변수의 상관관계를 찾아내는 복잡한 문제를 해결하여야 한다. 단일형 하드웨어 구조를 전제로

응용프로그램들을 하드웨어 구조에 맵핑하여 예상 성능 수치를 측정하고 분석하여, 여러 다양한 설계 변수를 결정짓는 체계적인 방법론으로 개발된 것이 Y-chart 설계방법이다(그림 1). 이 설계방법의 효용성은 Y-Chart 설계환경의 제안자인 Delft 대학의 Bart Kienhuis와 Ed Deprettere 교수 연구팀의 TV 응용의 비디오 프로세싱에 관련된 시스템 수준의 시뮬레이션에 관련된 결과에서 잘 나타나 있다.[3][5] 본 논문에서 구현된 시뮬레이터는 Hybrid 시스템의 설계과정에서 실제 프로토 타입을 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 매핑시켜주는 매핑제어기로 구성되어있다. 어플리케이션과 하드웨어 간의 의미론적인 차이를 최대한 줄이기 위하여 어플리케이션의 입력은 디지털 신호처리의 모델링 방법으로 널리 사용되는 Kahn 프로세스 네트워크로 전제하였다. Kahn 프로세스 모델은 병행처리를 위한 프로세스 모델로 각각의 프로세스들은 blocking-read, non-blocking-write의 특성을 가진다. 또한 재구성 가능한 하드웨어인 FPGA 또한 data-flow를 활용한 스트림 기반 하드웨어로 구성되는 것을 전제함으로써 어플리케이션과 하드웨어의 mapping을 용이하게 하였다. 전체적으로, 구현된 시뮬레이터는 어플리케이션 시뮬레이터가 발생시키는 트레이스를 매핑제어기가 해당되는 CPU 또는 스트림 기반의 하드웨어자원에 할당하는 Trace-Driven 시뮬레이션 방법을 사용하여 구현되었다.

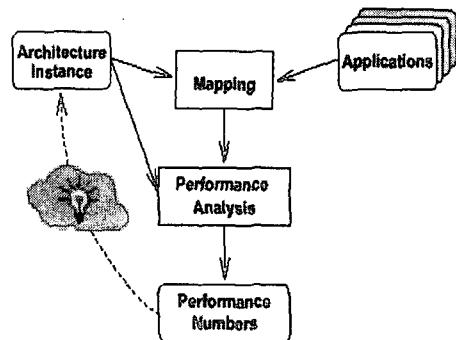


그림 1 Y-chart 설계 공간 탐색 방법

### 3. 설계 공간 탐색 환경

설계 공간 탐색은 가장 적합한 시스템 사양을 결정하기 위하여 설계변수들이 시스템 성능에 미치는 상관관계를 정량적으로 분석하고, 각각의 응용프로그램에 따른 설계변수를 결정하는 것이다. 설계공간 탐색은 설계변수가 증가함에 따라, 또 설계단계의 구체화가 추가됨에 따라, 다루어야 할 공간이 급격하게 증가한다. 시스템 제약 및 요구 조건을 만족시키는 시스템 구조의 집합은 그림 2에서 보인 것처럼 추상화(abstraction) 단계에 따라 삼각형으로 줄어든다. 이렇게 추상화 단계를 높임으로써 DSE에서 이용하는 도메인을 줄이지 않으면, 현실적인 DSE가 가능하지 않게 된다.[5]

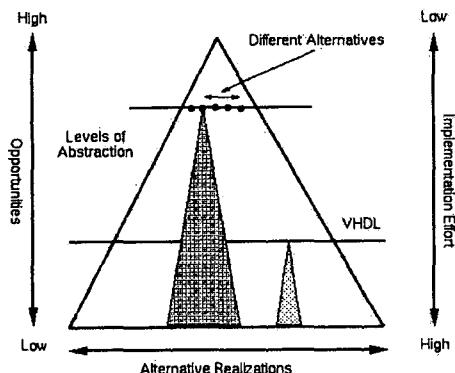


그림 2 추상화 단계에 따른 설계 공간

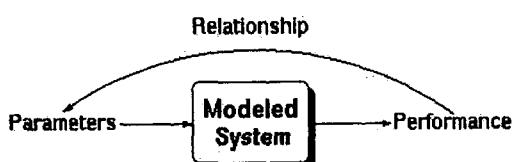


그림 3 시스템의 성능과 설계변수의 상관관계

그림 3에서 보인 것처럼 모델의 성능은 모델에 주어진 설계변수에 영향을 받는다. 설계변수가 다르게 적용되면, 설계될 시스템은 다른 성능 수치를 보이게 된다. 따라서 고수준 설계공간 탐색은 각각의 설계변수들을 적용하였을 때 성능수치가 어떻게 변하는지를 분석함으로써 이루어질 수 있다.[3]

본 논문에서는 설계변수를 변화시키면서, 위에서 언급한 바와 같이 현실적인 설계공간 탐색을 위해서

추상화단계를 시스템수준까지 높여 설계공간 탐색을 수행하였다. 기본적인 설계공간 탐색을 위한 구조는 Y-chart의 방법의 기본 구조를 용용하여 구현하였다. 그림 4는 구현된 설계공간 탐색 환경의 전체적인 구조를 보여준다. 먼저 주어진 어플리케이션을 어플리케이션 모델로 표현하고 재구성 가능한 하드웨어 템플릿을 만들어 다양한 구조를 시뮬레이션 할 수 있는 기본 환경을 조성한다. 그 다음 설계변수로 사용될 수 있는 항목들을 설계변수 테이블(Parameter Table)을 만들어 데이터베이스형태로 보유하면서 시뮬레이션의 동작을 재시작 할 때마다 설계변수 값들 중의 하나를 변경시켜가며 시뮬레이션을 계속해 나가면서 성능 수치를 산출하게 된다. 한번 시뮬레이션 하기 위해서 하드웨어 템플릿에 설계변수들을 적용하여 하드웨어 모델을 구축하고 어플리케이션 모델과 하드웨어 모델은 재구성 가능한 시뮬레이터에 의해서 매핑되는 구조를 가지고 있다.

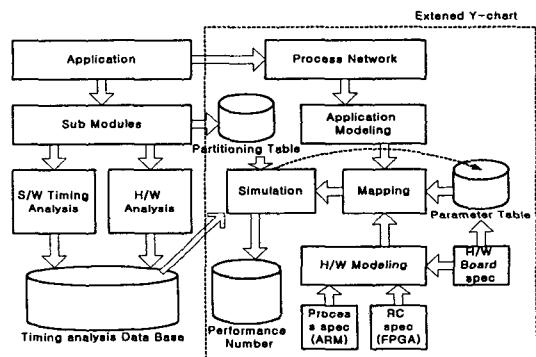


그림 4 확장된 Y-chart기반의 설계 공간 탐색 환경

### 4. 실험

용용 프로그램 모델로서는 H.263 알고리즘을 택하였고 하드웨어 구조 모델로서 일반적으로 재구성 가능한 내장형 시스템에서 주로 사용하는 하나의 범용 프로세서와 FPGA를 가진 시스템 모델을 사용하였다. H.263은 널리 쓰이는 비디오 신호 처리 모델로서 테이터 흐름을 나타내기에 적합하다. 그림 5는 Kahn 프로세스 네트워크 모델로 모델링된 H.263 해석기모델을 보여준다. 그림 5와 같이 표현된 H.263 용용 프로그램 모델과 하드웨어 모델에 대하여 FIFO와 우선순위 방법을 사용하는 간단한 스케줄링 알고리즘을 구현하여 시뮬레이션 하였다. 그리고 성능변수로서 하드웨어 소프트웨어 인터페이스를 위한 버스의 시간을 설계변수로하여 이를 변화시켜가며

각각의 스케줄링 방법이 어떤 성능을 나타내는지 살펴보았다. 이외에도 다양한 설계변수들(버스의 크기, CPU의 속도, FPGA의 속도, 패킷의 입력 간격등)을 적용하여 시뮬레이션을 진행하였으나 본 논문에서는 버스시간과 스케줄링 알고리즘의 상관관계만을 제시한다.

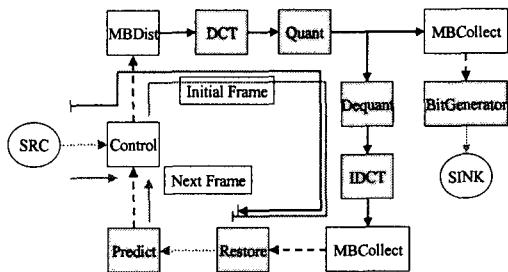


그림 5 H.263 모델

성능수치로 사용되는 병렬성(Parallelism), 하드웨어 자원들의 활용도(Utilization)와 평균처리시간(APT)은 수식 1에 의해서 구해진다. 수식 1에서  $T_{end}$ 는 패킷처리 시뮬레이션 종료시간을 의미한다.

$$\text{Parallelism} = \frac{\text{Combined Execution Time of the Workloads}}{T_{end}}$$

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

$$\text{APT} = \frac{\text{Combined Processing time of all packets}}{\text{Number of Packets}}$$

수식 1. 성능변수들의 계산식

병렬성은 현재의 시스템이 평균적으로 한 순간에 몇 개의 작업을 처리하고 있는가를 보여주는 성능수치로 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. 평균처리시간(Average Processing Time)은 어플리케이션에서의 각각의 작업들에 대하여 처리요구시간부터 처리가 종료된 시간까지의 경과시간에 대한 평균이다.

아래 그림들은 버스에서 점유하는 시간에 따른 각 하드웨어 자원의 활용도와 병렬성 그리고 패킷들이 시스템에서 종료될 때까지 머문 시간의 평균값을 보여준다. 그림 6를 보면 CPU의 활용도는 버스에서

점유하는 시간이 길어질수록 낮아지는 것을 볼 수 있고 두 가지(FCFS, 우선순위)스케줄링 방법들은 거의 차이가 없는 것으로 나타났다. 그럼 7을 보면 버스에서 점유하는 시간이 170 단위시간이하일 때는 우선순위 스케줄링 알고리즘을 사용하는 경우 FPGA의 활용도가 약간 향상되고 있음을 보여준다. 버스의 활용도와 시스템의 병렬성은 그림 6의 CPU의 활용도와 마찬가지로 거의 동일한 결과를 보여주고 있다. 반면, 버스의 점유시간이 증가하면 평균적으로 시스템에 처리되고 종료된 시간, 즉 평균적으로 시스템에 머문 시간을 살펴볼 때 우선순위 스케줄링 알고리즘을 사용한 시뮬레이션 결과가 모든 영역에서 빠른 처리를 하는 것으로 나타났다.

이러한 실험 결과는 본 논문에서 가정한 하나의 일반 프로세서와 하나의 FPGA를 사용하는 Hybrid 시스템에서는 우선순위방법과 같은 구현상의 오버헤드가 예상되는 스케줄링 방법보다는 FIFO와 같은 구현상의 오버헤드가 적은 알고리즘을 사용하는 하는 것이 바람직하다는 것을 보여준다.

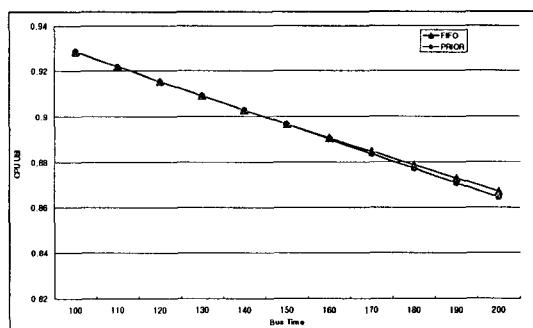


그림 6 버스 시간에 따른 CPU의 활용도

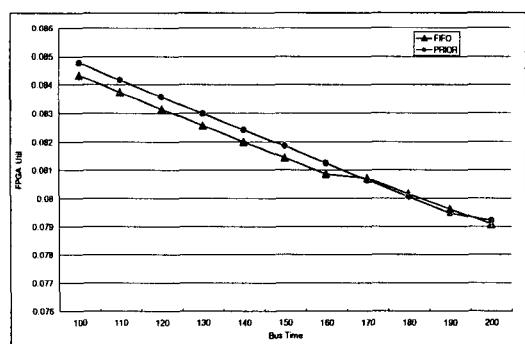


그림 7 버스 시간에 따른 FPGA의 활용도

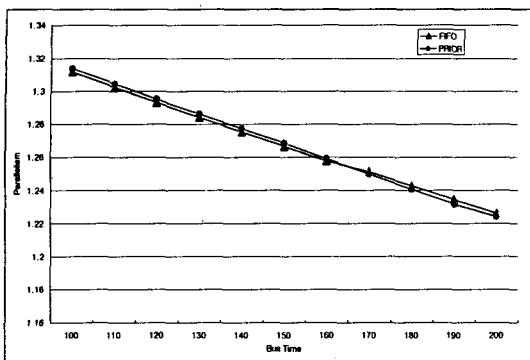


그림 8 버스 시간에 따른 병렬성

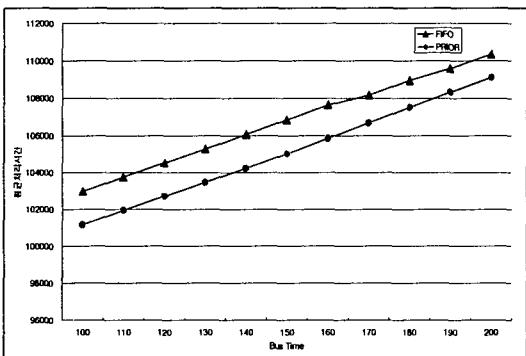


그림 9 버스 시간에 따른 종료 패킷당 평균처리시간

## 5. 결론

본 논문에서는 Hybrid 시스템을 위한 설계공간 탐색 도구를 개발하였고 H.263모델에 적용하여 실험하였다. 설계공간 탐색을 위하여 내장형 시스템 설계상에서 선택적으로 적용할 수 있는 설계변수들을 적용하여 시뮬레이션하기 위하여 Y-chart 설계 방법을 확장하여 설계공간 탐색 도구를 개발하였다. 또한 선택 가능한 설계변수들을 적용하여 다양한 하드웨어 구성을 생성하여 시뮬레이션 수행을 가능하게 하는 재구성가능한 시뮬레이터를 개발하였다.

이 도구를 이용하여 설계변수들을 변화시킴에 따라 성능 수치가 어떻게 변하는지 시뮬레이션을 통하여 산출하였다. 산출된 결과는 시스템 설계자로 하여금 최소 비용으로 시간 및 자원 제약사항을 만족하는 시스템구조를 설계하는데 기본적인 도움을 줄 수 있다.

이와 같이 본 논문에서 구현된 재구성가능한 시뮬레이터를 이용한 설계공간 탐색도구는, 시스템 설계

자로 하여금 실제 프로토타입을 구축하지 않고 설계하고자하는 Hybrid 시스템의 다양한 설계변수의 변화에 따른 성능수치를 산출할 수 있게 함으로써, 설계시간과 설계비용을 현저하게 줄여줄 것으로 기대된다.

## [참고문헌]

- [1] O.T. Alabahama, P. Cheung, and T.J. Clarke, "On the Viability of FPGA-Based Integrated Coprocessors," In Proceedings of IEEE Symposium of FPGAs for Custom Computing Machines, pp. 206-215, Apr. 1996
- [2] E. Sanchez, M. Sipper, J.-O. Haenni, J.-L. Beuchat, A. Stauffer, and A. Perez-Uribe, "Static and Dynamic Configurable Systems," IEEE Transactions on Computers VOL.48, No.6, June 1999.
- [3] 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
- [4] S. Lee, B. Jeong, S. Yoo, K. Choi, S. Hong, S. Moon, "A New Design Framework for Multiple-Application Embedded Java Systems with Reconfigurable Target Architectures," ACM SIGPLAN 1999 Workshop on Languages, Compilers, and Tools for Embedded Systems.
- [5] A.C.J. Kienhuis, Design Space Exploration of Stream-based Dataflow Architectures, PhD thesis, Delft University of Technology, Netherlands, 1998.
- [6] S. Bakshi, D. D. Gajaski, "Hardware/ Software Partitioning and Pipelining," In Proceedings of the 34th annual conference on Design Automation Conference, 1997, pages 713-716
- [7] G. Kahn, "The semantics of a simple language for parallel programming," Info. Proc., pages 471-475, Stockholm, Aug. 1974