

실시간 재구성 시스템을 위한 설계 공간 탐색 시뮬레이션 도구 구현

안성용⁺⁺, *이병석⁺, 이정아⁺

⁺조선대학교 컴퓨터공학과, ⁺⁺(주)미래로가는길

e-mail : *dis@rtf.co.kr*, *novrain@chosun.ac.kr*, *jalee@chosun.ac.kr*

Implementing Design Space Exploration Simulation tool for Real-Time Reconfiguration System

Seong-Young Ahn, *Byeong-Seok Lee, Jeong-A Lee

⁺Dept. of Computer Engineering, Chosun University,

⁺⁺Road To the Future Co., Ltd.

Abstract

A simulation tool for design space exploration of a real-time reconfiguration system was developed in this paper. We described an algorithm for a partial real-time reconfiguration to utilize already existing configured functional units and applied it to H.263 encoder application. This scheme allows us to find a starting configuration for the further optimization without actually building a prototype.

I. 서론

재구성형 시스템 기반의 어플리케이션에서는 프로그램의 일부분을 재구성이 가능한 하드웨어 자원으로 구현하면 성능을 향상시킬 수 있다. 하지만 이러한 자원은 용량이 제한적이고, 하드웨어 처리 부분이 많을 경우에는 모두 다 할당할 수 없는 문제점이 있다. 이러한 경우 프로그램이 실행하는 중간에 동적으로 하드웨어 자원을 재구성하는 방법이 필요하게 된다. 이러한 동작 방식을 실시간 재구성(Run-Time Reconfiguration)이라 한다. 실시간 재구성의 기본개념은 가상 메모리(Virtual Memory)와 유사한 가상 하드웨어(Virtual Hardware)에 기반을 두고 있다[1].

본 논문에서는 재구성형 시스템의 설계 공간 탐색을 위한 시뮬레이션 도구와 설계 공간 탐색 환경을 구축하였다. 실시간 재구성 모델링을 위하여 간단한 알고리즘을 기술하였으며 H.263 인코더를 부분적 실시간 재구성 모델에 적용하여 시뮬레이션 하였다.

II. 본론

재구성형 시스템에서 FPGA는 실시간 재구성이 가능한 대표적인 하드웨어 자원이라고 할 수 있다. 일반적인 FPGA는 모든 구성을 한꺼번에 바꾸는 단일문맥(Single-Context) 구조를 사용한다. 이러한 구조는 상대적으로 느리고 제한점이 많기 때문에 효율적이지 못하다. 때문에 현재는 단일 문맥 구조 보다는 다중문맥(Multi-Context) 구조[2]나 부분적 재구성(Partially Reconfigurable) 구조[3]를 많이 적용 하고 있다.

특정 기능이 FPGA에서 실행되기 위해서는 로직 합성(Synthesis)으로 만들어진 FPGA Bit Stream이 필요하다. 이렇게 생성된 Bit Stream을 실제로 동작하기 위해서는 FPGA 배치 및 경로 설정(Place & Route)과정이 필요하다. 이 때, 매핑 모양을 제어하는 과정이 실시간 재구성형 장치에서는 매우 중요한 역할을 한다. 일반적으로 특정 기능을 수행하는 기능 요소(Functional Element)의 모양은 불규칙적이다. 또한 합성 및 구현, 배치 및 경로 설정(Place & Route) 도구에 따라서 불규칙적으로 생성된다. 이러한 문제로 FPGA에서 부분적으로 실시간 재구성을 한다는 것은 현실적으로 불가능 하다. 따라서 이러한 부분을 간략하게 하기 위한 연구가 진행되고 있다[4]. 본 논문에서는 기능 요소의 모양을 사각형으로 가정하여 문제를 간략화 하였다. 그렇지만 전체적인 FPGA Cell들은 2차원 배열 형태이고, 기능 요소 또한 인접한 Cell의 집합이기에 사각형 모양으로도 제어할 수 있다.

그림 1은 부분적으로 실시간 재구성을 위한 알고리즘을 간략하게 보여준다. 처음에는 현재 실행해야 하는 기능 요소를 대기 큐에서 읽어 오는 과정이다. 대

기 큐에는 어플리케이션 모델에서 실행을 요청하는 기능 요소들이 사상 제어기에 의해서 저장되고 관리되고 있다. 다음으로 큐에서 읽혀진 기능 요소는 FPGA에서 구성되어 있는가를 검사한다. 여기서 구성되어 있다면, 그 기능 요소가 실행중인지를 검사한다. 그 기능 요소가 실행중이 아니라면 바로 실행될 수 있도록 FPGA 자원을 할당한다. 만약 실행중이라면 똑같은 기능을 두 개 이상 구성할 것인지, 아니면 한 개만 유지할 것인지 판단한다. 같은 기능을 수행하는 기능 요소를 두 개 이상 구성하지 않을 경우에는 기능 요소를 대기 큐에 되돌린다. 두 개 이상의 같은 기능 요소를 허용하는 경우에는 현재 FPGA 전체 공간 중에 이 기능 요소를 구성할 수 있는 공간이 충분한지를 판단한다. 만약 충분하면 해당공간을 초기화한 후 재구성 과정을 실행하며, 부족할 경우에는 대기 큐로 되돌린다. 다음으로 다시 한 번 기능 요소에 대한 FPGA 구성 여부를 판단한다. 공간에 기능 요소가 없으면 할당 자원이 충분한지 검사한다. 여기서 공간이 부족하면 대기 큐로 되돌리고, 공간이 충분하면 해당 공간을 초기화 한 후 재구성하는 과정을 진행한다. 이러한 과정들은 대기 큐에 있는 모든 기능 요소에 대하여 반복적으로 수행한다.

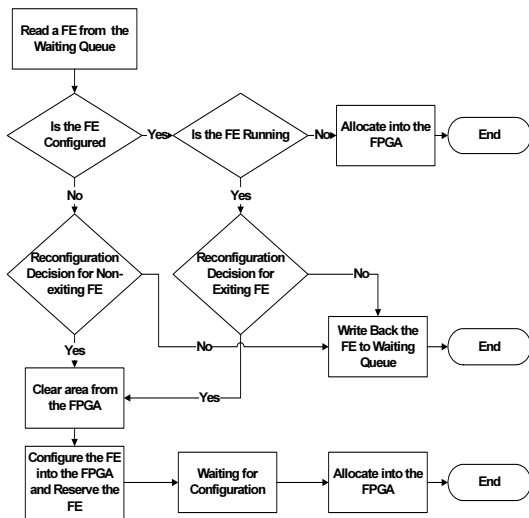


그림 1. 실행 중 부분적 재구성을 위한 흐름도

III. 실험 결과

그림 2는 H.263 인코더 모델에 대한 재구성 가능한 시뮬레이터의 수행 시간을 보여준다. 이 실험에서는 실시간 재구성 가능 SoC에서 시스템의 주요 장치인 CPU와 FPGA의 속도 변화에 따른 시뮬레이션 종료시간을 보여준다. CPU와 FPGA 속도는 기준 프로세서의 속도를 1로 정규화 하였다. 따라서 속도가 2는 기준 프로세서보다 2배 빠르다는 것이고, 0.5는 2배 느리다는 뜻이다. 그리고 CPU Rate와 FPGA Rate는 정규화된 범위를 나타낸다. 그림처럼 CPU 속도가 0.8까지는 성능 향상에 뚜렷하지만, 그 이상에서는 둔화되는 것

을 볼 수 있다. 또한 FPGA 역시 속도 증가에 따른 성능 향상이 미비함을 볼 수 있다.

이 실험 결과를 이용하면 내장형 시스템에서 작동하는 어플리케이션을 설계자가 실제 프로토타입 하드웨어를 제작하지 않아도 어플리케이션의 제약조건을 만족하는 하드웨어 구성을 개략적으로 알 수 있는 방법을 제시해준다.

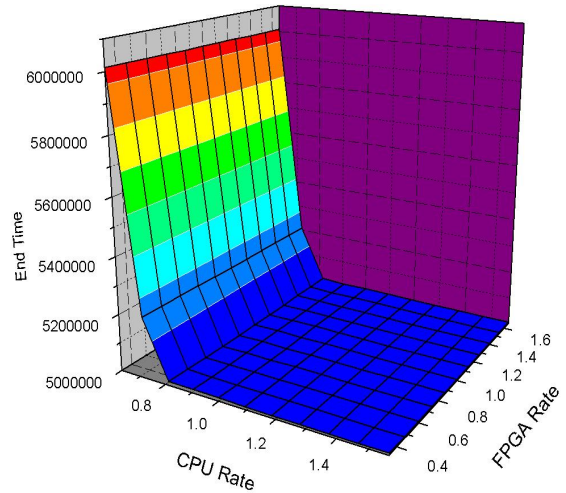


그림 2. 실행 중 부분적 재구성 적용시 시뮬레이션 종료시간

IV. 결론 및 향후 연구 방향

본 논문은 설계 공간 탐색 환경에서 시뮬레이션을 간단하게 수행하기 위한 방법으로 전체 공간을 검사하여 유휴공간을 찾는 First-Fit 방법을 사용하였다. 이 방법은 실제 시스템의 경우 오버헤드가 가중 될 수도 있는 부분이다. 따라서 실제 시스템이 실시간으로 부분적 재구성에 관한 부분은 타 연구결과와 결합되어 최적화 과정을 거쳐야 한다.

참고문헌

- [1] Katherine Compton, Scott Hauck, "Reconfigurable Computing: A Survey of Systems and Software," ACM Computing Surveys, 2002.
- [2] S. Trimberger, D. Carberry, A. Johnson, J.Wong, "A Time-Multiplexed FPGA," IEEE Symposium on Field-Programmable Custom Computing Machines, 1997.
- [3] S. Hauck, T. W. Fry, M. M. Hosler, J. P. Kao, "The Chimaera Reconfigurable Functional Unit," IEEE Symposium on Field-Programmable Custom Computing Machines, 1997.
- [4] Matthias Dyer, Chistian Plessl, and Marco Platzner, "Partially Reconfigurable Cores for Xilinx Virtex," FPL 2002, LNCS 2438, Springer-Verlag Berlin Heidelberg, 2002.