분할 SIMD 재구성형 병렬 프로세서를 위한 컴파일러 설계

권용인, 김용주, 윤종희, 안민욱, 최영규, 백윤흥
서울대학교 공과대학 전기컴퓨터 공학부
e-mail : yikwon, yjkim, jhyoon, mwahn, ykchoi, ypaek@compiler.snu.ac.kr

A Design of compiler for partitioned SIMD reconfigurable parallel processor

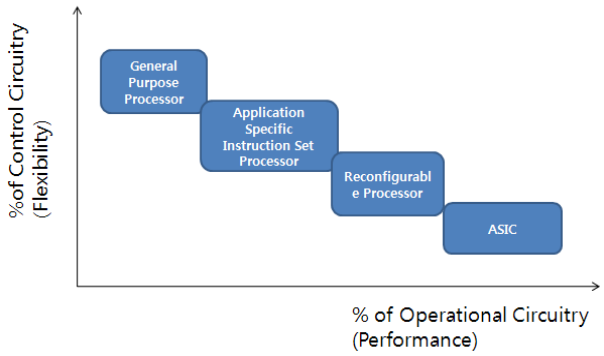
Yongin Kwon, Yongjoo Kim, Jonghee yoon, Minwook Ahn, Youngkyu Choi, Yunheung Paek
School of Electrical Engineering, Seoul National University

요 약

본 논문에서는 휴대용 단말기의 실시간 서비스 제공을 위한 재구성형 병렬 프로세서를 소개 하고, 그 기능인 분할 SIMD 를 표현하기 위한 새로운 프로그래밍 언어와 컴파일러를 설계한다. 이 방식을 이용하여 빠른 어플리케이션 개발과 각 어플리케이션의 성능 향상에 용이하다.

1. 서론

최근 휴대용 단말기에서는 동영상 재생 및 편집, 오디오 레코딩 및 재생 등의 기존의 응용프로그램부터 3D 그래픽스 등의 다양한 서비스를 제공하고 있지만 휴대용 단말기의 범용 프로세서의 성능이 턱없이 낮아 이를 위해 다양한 하드웨어가 제안되고있다.



(그림 1) Tradeoff between flexibility and performance.

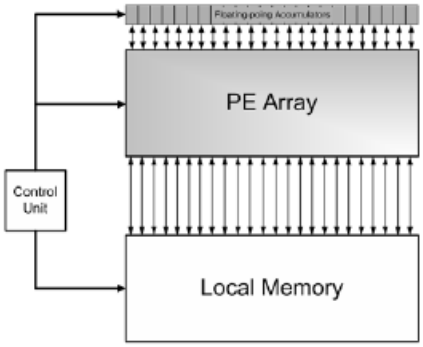
그림 1 은 유연성과 성능 사이의 관계를 보여주고 있다. 범용 프로세서를 사용할 경우 다양한 매체에 활용을 할 수 있지만 성능이 떨어지고, ASIC 은 반대로 성능은 뛰어나지만 다양한 매체에 활용하지 못하고, 변화에 대응하기가 어렵다.

범용프로세서와 ASIC 의 대안으로 병렬 프로세서를 사용하여 유연성과 성능을 향상시시키고, 비용과 전력소모는 줄일 수 있다[1].

[1]에서 제안된 아키텍쳐는 기존의 프로그래밍 언어로는 모든 명령어를 활용하지 못한다. 분할된 SIMD Array 에 서로 다른 명령어를 할당해 주려면 새로운 형태의 프로그래밍 언어와 그를 위한 컴파일러가 필요하다.

본 논문에서는 먼저 [1]에서 제안된 아키텍쳐를 소개한 뒤, 이를 위한 프로그래밍 언어와 컴파일러를 제안한다. 또한 아키텍쳐의 성능을 활용하기 위한 프로그래밍 기법을 설명한다.

2. 재구성형 병렬 프로세서의 구조

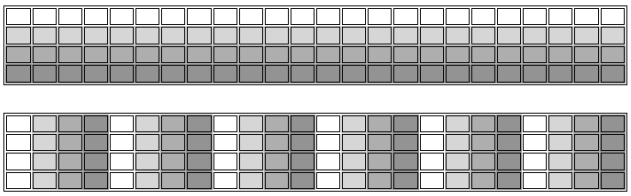


(그림 2) 재구성형 병렬 프로세서 전체 구조

재구성형 병렬 프로세서는 PE Array, Local Memory, Control Unit 으로 나누어져 있다. PE Array 는 4x24 의 PE 들의 배열이고 이들은 그물형망(mesh network)[2]으로 연결되어 있다. Local Memory 는 PE Array 와 직접 열결되어 있고, 각각의 PE 가 독립적으로 메모리 뱅크에 연결되어 있다. 각 메모리 뱅크는 2 킬로 바이트의 저장 공간을 가지고 있고 총 4x24=96 개의 메모리 뱅크를 가진다. Control Unit 은 PE Array 를 제어하여 명령어를 전달하는 역할을 한다. 또한 부동소수점 누산기 24 개가 PE Array 위쪽에 배치되어 부동소수점 덧셈을 빠르게 수행할 수 있도록 한다.

3. 기능적 특성

PE Array 를 4 그룹으로 분할하여 각 그룹이 다른 명령어를 수행하도록 할 수 있다.



(그림 3) 분할 SIMD 방식의 그룹 분할 방법

그림 3 은 그룹을 나누는 방법을 나타내고 있다. 첫 번째 방법은 0 행 PE, 1 행 PE, 2 행 PE, 3 행 PE 로 나누는 방법이다. 두 번째 방법은 0 열/4 열/8 열/… PE, 1 열/5 열/9 열/… PE, 등으로 나누는 방법이다. 이와 같이 96 개의 PE 를 24 개씩 네 그룹으로 분할을 하여 각 그룹이 mux_IN, addr_IN, addr_C 제어 신호를 서로 다르게 수행할 수 있다.

PE 는 Branch 명령어를 가지고 있지 않다. 따라서 조건부 실행을 통해 Branch 를 구현하여야 한다

<표 1> 플래그의 종류와 역할

종류	역할
<1>	항상 1
<Z>	산술논리장치의 출력이 영(Zero)
<N>	산술논리장치의 출력이 음수(Negative)
<f1>	<Z> 플래그를 유지; 사용자가 저장 여부 결정가능
<f2>	<N> 플래그를 유지; 사용자가 저장 여부 결정가능

각각의 PE 는 연산 결과를 각각의 플래그에 반영하여 그 값에 따라 다음 명령어를 수행할지를 결정할 수 있다.

4. 프로그래밍 언어 설계

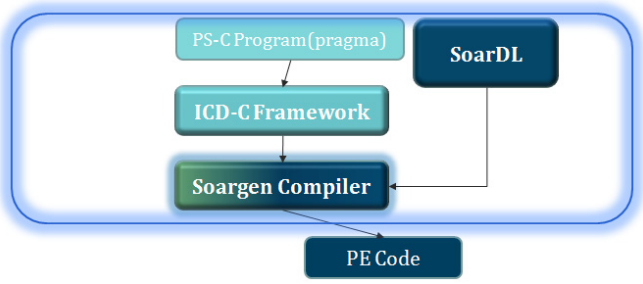
재구성형 병렬 프로세서의 코드를 고급 프로그래밍 언어를 사용하여 효과적으로 생성하기 위해 C 를 확장한 Partitioned SIMD C(PS-C)를 설계하였다. PS-C 는 PE 를 그룹별로 나누어 데이터 전달을 수행시키기 위해 MIMD_COL_MODE()함수와 MIMD_ROW_MODE() 함수를 Intrinsic function 으로 구현한다.

```
MIMD_COL_MODE(){
    b = right a;
    c = top b;
    a = bottom c;
    d = left d;
}
```

(그림 4) PS-C 예제코드

그림 4 는 PS-C 의 사용법을 보여준다. Right, left, top, bottom 은 각각 오른쪽, 왼쪽, 위쪽, 아래쪽의 PE 에서 값을 가져옴을 뜻한다. MIMD_COL_MODE()함수는 PE 를 행에 따라 네 그룹으로 나누고 첫 번째 그룹은 b = right a;, 두 번째 그룹은 c = top b; 를 수행하도록 한다. PS-C 를 이용하여 프로그램을 작성하면 C 에서는 표현하지 못하는 분할 SIMD 코드를 생성할 수 있다.

5. 컴파일러



(그림 6) 재구성형 병렬 프로세서를 위한 컴파일러 설계

새로운 프로그래밍 언어로 기술된 소스 코드를 컴파일 할 수 있는 컴파일러의 구조는 위 그림 5 와 같다. Pragma 를 포함하는 작성된 PS-C 프로그램은 ICD-C Framework 를 통해 컴파일러가 읽을 수 있는 코드 형태로 변환한다. 미리 기술된 아키텍처 정보에 맞춰 생성된 재겨냥성 컴파일러인 Soargen 컴파일러를 거쳐 PE 코드가 생성된다. 기술되는 아키텍처 정보에는 분할 SIMD 에 관한 정보와, Branch 명령어를 대체할 조건부 실행 플래그 정보가 포함된다.

6. 결론

본 논문에서는 모바일 미디어 어플리케이션을 효율적으로 프로그래밍 하고, 컴파일하기 위한 프로그래밍 언어를 설계하고 그에 따른 컴파일러 구조를 제안하였다. 제안된 프로그래밍 언어와 컴파일러를 통해 어플리케이션 개발 시간을 단축시키고 코드 성능을 향상할 수 있다.

참고문헌

[1] S. Yoo, K. Kim, Y. S. Yang, et al., “A Reconfigurable Parallel Processor for Effiicient Processing of Mobile Multimedia,” Journal of IEEK, Vol. 44-SD, No. 10, Oct. 2007, pp.23-32

[2] Abbaspour. M and Zhu. J, 2002, “Retargetable binary utilities”, In Proceedings of the 39th Design Automation Conference. New Orleans, La, 331-336

[3] S. Pees, A. Hoffmann, V. Zivojnovic, and H. Meyr., LISA – Machine Description Language for Cycle-Accurate Models of Programmable DSP Architectures, In Proc. Of the Design Automation Conference (DAC), New Orleans, June 1999

Acknowledge

본 연구는 교육과학기술부/한국과학재단 우수연구센터육성사업(R11-2008-007-01001-0), 지식경제부 출연금으로 ETRI, SoC 산업진흥센터에서 수행한 ITSoC 핵심설계인력양성사업, 서울시 산학연 협력사업(10560), 2009 년도 정부(교육과학기술부)의 재원으로 한국과학재단의 국가지정연구실사업(No.2009-0083190)의 지원을 받아 수행되었습니다.