http://dx.doi.org/10.3745/KIPSTA.2012.19A.1.017

데이터를 고려한 저전력 소모 CGRA 매핑 알고리즘

김 용 주[†] · 윤 종 희[††] · 조 두 산[†††] · 백 윤 흥[††††]

요 약

모바일 시장 및 소형 전자기기 시장의 발달에 따라 고성능 프로세서에 대한 요구 또한 커지게 되었다. 재구성형 프로세서(CGRA)는 고성능과 저전력 소모를 동시에 만족시키는 프로세서로 ASIC의 고성능 저전력을 대체하면서도 하드웨어를 쉽게 재디자인 할 수 있도록 구성된 프로세서이다. 어플리케이션의 구조에 따라 CGRA의 전체수행시간이 프로세서 자체의 수행시간보다 데이터의 전송시간에 종속되는 경우가 있다. 이 논문에서는 데이터 전송시간에 따라 수행에 사용되는 자원을 최적화 함으로써 전력소모를 줄이는 매핑 알고리즘을 제안하였다. 제안된 알고리즘을 사용한 경우, 기존의 방식보다 최대 73%, 평균 56.4%의 전력소모를 줄일 수 있었다.

키워드 : 재구성형 프로세서, 매핑 알고리즘, 저전력

Low Power Mapping Algorithm Considering Data Transfer Time for CGRA

Yongjoo Kim[†] · Jonghee Youn[††] · Doosan Cho[†††] · Yunheung Paek[††††]

ABSTRACT

The demand of high performance processor is soaring due to the extending of mobile and small electronic device market. CGRA(Coarse Grained Reconfigurable Architecture) is the processor satisfying both of performance and low-power demands and a great alternative of ASIC that can be reconfigured. This paper presents a novel low-power mapping algorithm that optimizes the number of used computation resource in the mapping phase by considering data transfer time. Compared with previous mapping algorithm, ours reduce energy consumption by up to 73%, and 56.4% on average.

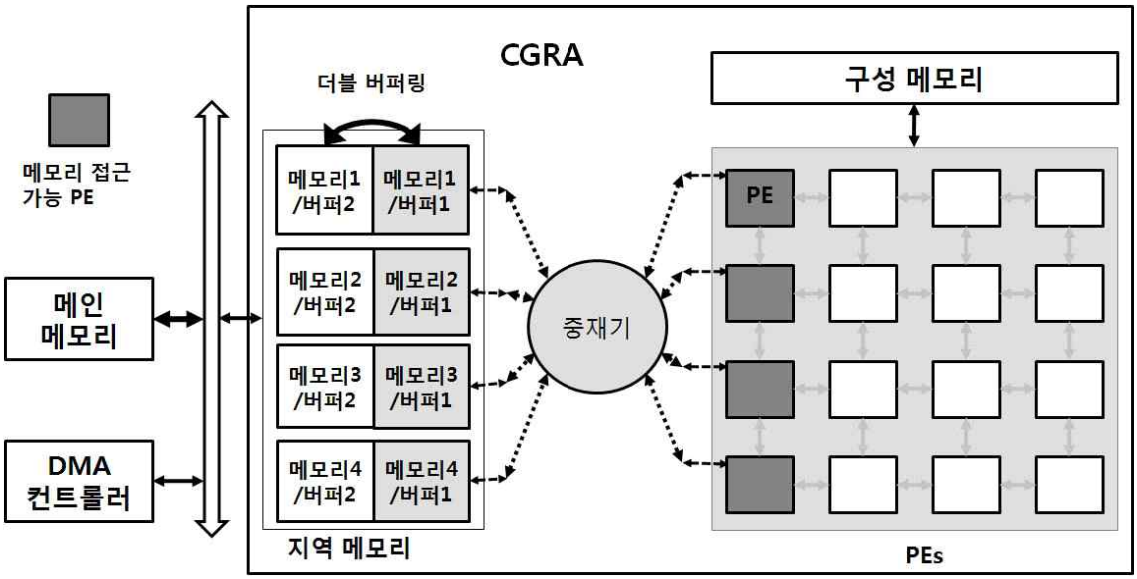
Keywords : CGRA, Mapping Algorithm, Low-power

1. 서 론

오늘날 스마트폰 및 소형 전자기기 시장이 폭발적으로 발전하면서 멀티미디어, 통신, 게임 등의 다양한 어플리케이션이 요구되고 있으며, 그 복잡도 역시 빠르게 증가하고 있다. 이전까지는 단일 프로세서를 사용하여 이러한 응용을 처리하여 왔으나 단일 프로세서의 성능적 한계로 인해 점점 더 복잡하고 다양해지는 어플리케이션에 대한 욕구를 만족시키기가 어려워지고 있다. 이에 대한 해결책으로 멀티 프로세서로의 전환이 제시되었고 많은 연구가 진행되고 있다. 하지만 소형전자기기의 경우 일반적인 프로세싱 환경과는 달리 전력의 제한이 있기 때문에 많은 전력을 소모하는 프로세서를 여러 개 넣는 방식은 아직 원하는 만큼의 효과를 보이지 못하고 있다. 성능과 전력 소모 절감을 둘 다 만족시키기 위하여 ASIC을 이용한 방식이 많이 이용되고 있으나 어플리케이션이 바뀔 때마다 하드웨어를 새로 설계해야 하는 ASIC 방식은 급변하고 있는 어플리케이션 시장에서 활용되기가 어렵다.

이러한 요구를 만족 시키기 위해 강력한 성능과 저전력 수행을 동시에 만족 시키면서 다양한 어플리케이션에서 사용할 수 있는 프로세서인 재구성형 프로세서 CGRA(Coarse Grained Reconfigurable Architecture)가 제안 되었다. CGRA는 일반 프로세서에 비해서 최소 열배에서 많게는 수십배에 달하는 PE(processing element)를 가지고 있음으로 많은 수의 연산을 동시에 수행할 수 있다. 하지만 분기제어부(branch unit)이나 캐쉬와 같은 일반적인 프로세서에 있는

※ 본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업(No.2011-0000975), 2011년도 정부(교육과학기술부) 재원의 한국과학재단의 국가지정연구실사업(No.2011-0018609) 및 지원(No. 2011-0012522), 2011년 순천대학교 학술연구비 공모과제 및 IDEC의 지원을 받아 수행되었음.
† 준 회 원 : 서울대학교 전기정보공학부 박사과정
†† 정 회 원 : 강릉원주대학교 컴퓨터공학과 강의전담교수(교신저자)
††† 정 회 원 : 순천대학교 전자공학과 전임강사
†††† 종신회원 : 서울대학교 전기정보공학부 교수
논문접수 : 2011년 5월 18일
수 정 일 : 1차 2011년 9월 22일
심사완료 : 2011년 9월 27일



(그림 1) CGRA 프로세서 구조

부분 중 연산에 필요없는 부분을 제거 하였기 때문에 강력한 성능에 비해 크기도 작으면서 저전력의 수행을 보여준다. ASIC과 달리 CGRA는 각 PE들이 프로그램 가능하기 때문에 어플리케이션을 바꿀 때 마다 각 PE들을 새로운 어플리케이션에 맞게 연산을 재배치 할 수 있으며, 이로 인해 하드웨어를 다시 디자인 해야 하는 수고를 덜 수 있다.

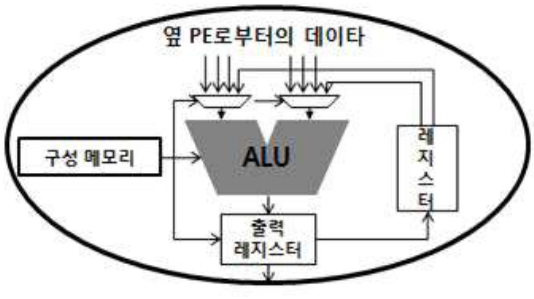
CGRA를 모바일 기기에 사용하게 될 경우 제한적인 배터리를 효율적으로 사용하기 위해 전력소모를 줄일 수 있도록 설계하는 것이 중요하다. 하드웨어 적으로, 그리고 소프트웨어 적으로 CGRA를 설계하고 활용하는 방법에 대한 여러가지 논문이 제시되었지만 CGRA의 전력소모를 줄이기 위한 연구는 상대적으로 많이 이루어 지지 않았다. 단지 몇 가지의 논문[1][2]만이 전력 소모를 줄이기 위한 기법을 제안하였다. 본 논문에서는 버스의 대역폭과 그로 인해 소모되는 데이터 전송 시간을 고려하여 CGRA의 일부를 사용 정지시키는 방식을 사용하여 CGRA의 전력소모를 줄이는 방식을 제안한다. CGRA의 성능을 떨어뜨리지 않는 최대한의 범위를 분석하고 그에 맞게 CGRA의 구성을 수정함으로써 CGRA 본연의 성능은 유지시키면서 전력 소모를 줄였다. 실험결과 CGRA를 수행하는데 걸리는 시간은 전혀 감소하지 않았고 전력 소모는 56%만큼 감소하였다.

2. 배경지식

2.1 CGRA 프로세서

(그림 1)은 CGRA의 프로세서 구조를 보여주고 있다. CGRA는 PE(processing element) 라고 불리는 작은 수행장치가 여러 개가 배열되어 있는 구조로 각 PE는 격자모양으로 연결되어 있다. 각 PE는 산술/논리연산을 수행할 수 있으며 일부 PE는 특수 기능으로 곱셈연산이나 메모리 접근 등을 할 수 있다. 각 PE들의 수행 내용은 매 cycle마다 바뀔 수 있으며, 그 수행할 내용은 구성 메모리에 저장되어 있어 PE의 구성을 바꿔주게 된다. CGRA는 ASIC과 마찬가지로 하드웨어 가속기의 역할을 하며 버스에 IP 형태로 붙어서 수행된다. 수행에 필요한 데이터는 DMA 컨트롤러를 통해서 CGRA 내부에 있는 지역 메모리로 전달되어 수행에 사용된다. 지역메모리는 여러 개의 작은 메모리로 구성되어 있으며, 그 안의 데이터는 중재기를 통해서 PE로 전달된다. 메인 메모리와의 데이터 전송을 위해서는 더블 버퍼링 기법이 사용된다. 더블 버퍼링은 수행과 데이터 전송을 동시에 하기 위해서 사용하는 기법이다. 더블 버퍼링을 위해서 각 메모리는 2개의 버퍼로 구성되어 있는데 그 중 하나는 PE 계산에 사용되고, 나머지 하나는 bus를 통해서 다음 수행을 위한 데이터를 준비하는 형태로 수행된다. PE에 데이터를 제공하던 버퍼에서 데이터를 다 사용한 경우 이 버퍼는 이제 메인 메모리에서 데이터를 들고 오는 버퍼로 전환되고, 데이터를 미리 준비해 놓은 다른 버퍼가 다시 PE에 데이터를 제공하게 된다.

(그림 2)는 각 PE의 구조를 보여준다. PE는 구성 메모리에서 매 cycle마다 수행해야 할 내용을 전달받아서 수행한



(그림 2) PE의 구조

다. 수행에 필요한 데이터는 자신의 레지스터에서 들고 오거나, 자신 주변의 다른 PE에 있는 출력 레지스터로부터 데이터를 받아와서 계산을 수행한다.

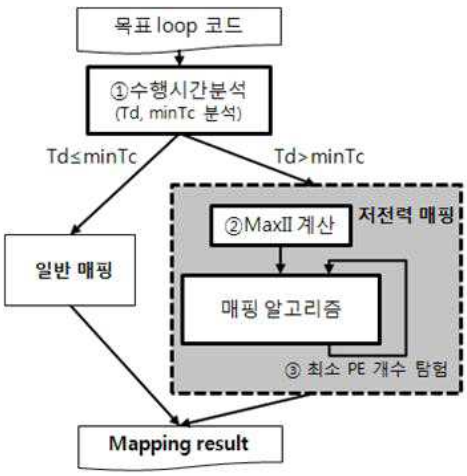
2.2 어플리케이션 매핑

CGRA는 일반적으로 loop을 가속 하는 데에 사용된다. CGRA에서 원하는 loop을 가속을 하기 위해서는 먼저 가장 안쪽 loop 부분을 DFG(data flow graph) 형태로 바꿔준다. DFG는 코드의 병렬성을 쉽게 확인하기 위해서, 소스 코드를 연산을 나타내는 노드와 데이타 종속성을 나타내는 엣지로 그린 그래프이다. 코드를 DFG로 변환 한 뒤에는 이 DFG를 CGRA에서 수행될 수 있는 형태로 스케쥴링 하는 작업이 필요하다. CGRA의 PE 집합도 PE 각각을 뜻하는 노드와 그들간의 연결을 뜻하는 엣지로 연결된 그래프로 그릴 수 있는데 이 그래프를 CGRA 그래프라고 한다. 어플리케이션 매핑이란 DFG의 노드들을 CGRA 그래프의 노드들에 매핑하는 작업이다. DFG의 노드가 CGRA의 노드에 매핑이 된다는 것은 그 연산이 해당 PE에서 수행되도록 할당한다는 것을 의미한다. DFG의 엣지는 데이터의 흐름을 의미하므로, 매핑 할 때에는 CGRA 그래프에서 데이터가 흘러갈 수 있는 길을 보장해 줄 수 있도록, 즉 엣지의 연결 상태를 고려하면서 노드를 매핑한다. 효율적인 어플리케이션 매핑을 위해서는 컴파일러는 loop 매핑 때 소프트웨어 파이프라이닝을 활용 해야 한다. 모듈로 스케줄링[3]은 가장 많이 사용되는 소프트웨어 파이프라이닝 방법이다. loop의 반복을 II(Initiation Interval) 이라고 정의한 시간마다 이전 반복과 겹쳐서 수행함으로써 병렬성을 최대한 살려서 수행할 수 있도록 스케줄하는 방법이다. CGRA에서 모듈로 스케줄링을 사용한 매핑 연구는 이미 많이 연구되어있다[4][5][6].

3. 전력 소모 최소화 매핑

3.1 데이터 전송시간과 전력 소모

어플리케이션의 loop은 코드 특성에 따라서 몇 가지로 분류 할 수 있다. loop이 분기문이 많은 경우에는 컨트롤 집중 loop이라고 하고 분기문이 없이 계산을 많이 하는 경우에는 계산 집중 loop이라고 한다. 계산 집중 loop의 경우에도 계산량에 비해서 많은 데이터를 읽고 쓰는 경우에데이터 집중적 loop으로 세분화하여 정의할 수 있다. CGRA는 보통 계산 집중적 loop을 가속하는데 많이 사용된다. 그런데 데이터 집중적 loop의 경우에는 CGRA에 데이터를 전달하는 시간이 오래 걸려서 데이터 전송 시간에서 병목 현상이 발생해서 전체 수행시간이 느려지는 경우가 있다. 실존하는 CGRA 프로세서 RSPA[2]에서 더블 버퍼링 기법[1]을 사용하여 Compress 어플리케이션의 loop을 수행해본 결과, 메모리의 버퍼셋 하나를 사용해서 CGRA에서 수행하는 시간(이후 Tc로 언급한다.)을 1이라고 할 때, 메인 메모리에서 버퍼셋으로 데이터를 채우는 시간(이후 Td로 언급한다)은 2.6이 걸리는 것을 확인할 수 있었다. 즉 데이터 집중적 loop의 경우에는 CGRA의 계산속도가 빠르더라도 데이터 전송이 그 속도를 못 따라 가기 때문에 전체수행 시간이 데이터 전송 속도에 맞춰진다는 뜻이다. 이 경우에는 PE들은 수행이 다 끝난 뒤 다음 버퍼에 데이터가 채워질 때 까지 동작을 정지하고 기다리게 된다. 하지만 PE들이 수행을 멈추고 정지되어 있다고 전력소모가 일어나지 않는 것이 아니다. 전력 소모는 크게 2가지로 나눌 수 있다. 동적 전력 소모와 누설 전력 소모로 나눌 수 있는데 동적 전력은 트랜지스터들을 변경하는 작업을 할 때, 즉 작업을 수행할 때 사용되는 전력이다. 반대로 누설 전력은 하드웨어에 기본적으로 흐르면서 소모되는 전력으로 동적 전력과는 다르게 아무런 일도 하지 않더라도 소모된다. 즉 데이터 집중 loop에서 계산이 다 끝나고 다음 데이터를 위해서 대기를 하고 있는 동안에도 전력 소모가 일어나는 것이다. 우리는 사용하는 PE의 갯수를 줄여서 누설전력을 최소화 하는 저전력 매핑 방법을 제안한다.



(그림 3) 저전력 매핑 개관

3.2 저전력 매핑

저전력 매핑은 데이터 집중적 loop인 경우를 찾아내고, 그경우에 Td>Tc를 만족하는 범위내에서 사용하는 PE의 개수를 줄여서, 전체 성능은 유지하면서 PE에서 소모되는 누설 전력을 줄이는 매핑 방법이다. 단순히 PE의 개수를 줄일 경우 전체 성능이 나빠질 수 있지만, 저전력 매핑에서는 Td를 분석하여 Tc가 Td를 넘지않는 범위에서 최대한 사용하는 PE를 줄이므로[2] 전체 수행시간이 늘어나지 않으면서도 전력 소모를 줄일 수 있다. (그림 3)은 저전력 매핑의 전체

1) 데이터 전송과 계산 수행을 동시에 수행하게 함으로써 데이터 전송 시간을 숨기는 방법을 더블 버퍼링이라고 한다. 각 지역 메모리를 두 개의 버퍼로 나누고, 하나의 버퍼가 메인메모리와 데이터를 주고받으며 데이터 전송을 하는 동안 다른 하나의 버퍼는 PE들의 계산에 사용되게 하여 데이터를 전송하면서 계산을 수행하도록 한다. 각 버퍼의 역할이 끝나면 서로 역할을 바꿔서 데이터를 준비한 버퍼가 PE의 계산에 사용되고, 데이터를 다 쓴 버퍼는 메인메모리에서 새로운 데이터를 준비함으로써 계산 수행이 끊임이 없도록 한다.

2) 사용하는 PE를 줄이면 CGRA의 계산 능력이 떨어져서 버퍼 하나를 소모하는데 걸리는 시간 Tc가 늘어나게 된다.

흐름을 보여준다. 저전력 매핑은 수행시간 분석 단계와 저전력 매핑 단계로 구분된다.

수행시간 분석 단계(그림 3에서 ①)에서는 Td와 모든 PE를 다 사용했을 때의 Tc, 즉 최대 성능에서 나오는 가장 빠른 계산시간(minTc)를 분석한다. Td는 목표 loop 코드의 데이터 접근 패턴을 분석한 뒤, 전송될 데이터 양과 시스템 버스 속도를 비교 계산해서 측정한다. minTc는 매핑 알고리즘을 사용해서 선행 매핑을 한번 해 보는 방식으로 측정한다. 만약 Td가 Tc보다 큰 경우에는 데이터 집중적 loop이므로 저전력 매핑을 수행하고 그렇지 않은 경우에는 일반 매핑 알고리즘을 수행한다.

저전력 매핑의 처음 단계는 Max II 값을 계산하는 단계(그림 3에서 ②)이다. II는 앞서 언급한 것처럼 loop의 반복 하나를 수행하는데 걸리는 시간으로, II가 커질 수록 전체 수행에 걸리는 시간은 길어진다. 하지만 II가 커지면 하나의 loop을 수행하는데 보다 오랜 시간을 소모해도 되므로 필요한 PE의 개수는 줄어든다. 이 단계에서는 Td를 넘지 않는 범위에서 최대 Tc값을 가질 수 있는 II를 구하는데, 그 II값을 Max II라고 한다. Max II를 계산한 후에는, 계산된 Max II를 만족하면서 주어진 목표 loop 코드가 매핑이 가능한 최소한의 PE를 구한다.(그림 3에서 ③) 이 과정은 최소 PE 개수 탐험을 통해 매핑과 동시에 이루어진다. 앞에서 구한 MaxII로 II를 고정하고 PE를 하나만 쓰는 경우부터 매핑알고리즘을 수행한다. 매핑이 실패하면 PE의 갯수를 하나씩 늘려가면서 매핑이 성공할 때까지 매핑을 시도한다. 매핑이 성공한다면 그때의 PE의 개수가 Max II를 만족시킬 수 있는 최소한의 PE 개수이다. 저전력 매핑의 결과물을 저전력 매핑이 아닌 경우와 비교했을 때 Tc는 증가하게 되지만 그 수치는 Td를 넘지 않는 내에서 증가하므로 전체 수행시간에서는 변화가 없다. 또한 사용하는 PE의 개수가 줄어들게 되므로 소모되는 누수 전력을 줄일 수 있다.

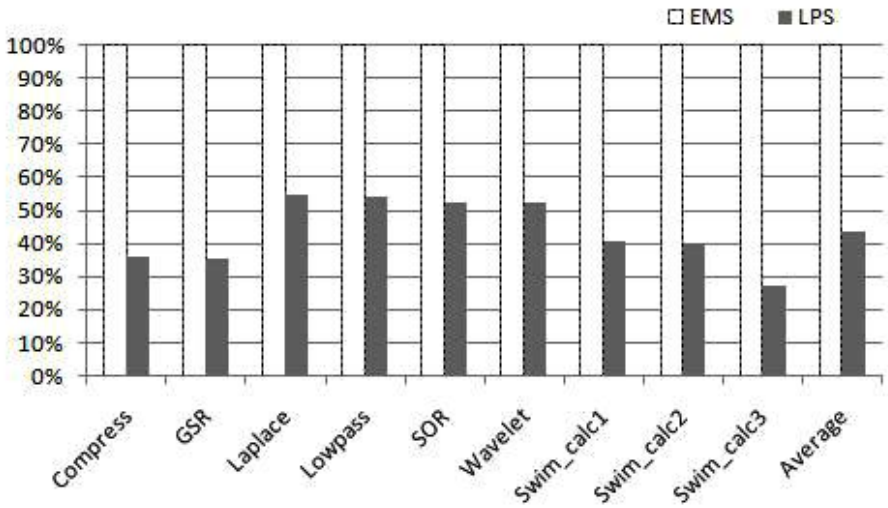
4. 실 험

4.1 실험 세팅

우리의 방식을 검증하기 위해 MMI(MultiMedia Integer) 벤치마크와 SPEC2000의 Swim 어플리케이션을 사용하였다. CGRA는 4x4 CGRA를 사용하였고, 각 PE는 산술논리연산장치와 곱셈기를 가지고 있다. 이중 첫번째 행의 PE 4개는 메모리에 접근할 수 있는 장치도 가지고 있다. 그리고 각 PE는 모두 2개의 분산된 레지스터를 가지고 있다. 버스의 데이터 전송 속도, 즉 데이타 대역폭은 실존하는 CGRA인 RSPA[2]와 같이 16bit/2cycle로 하였다. 각 PE의 전력소모량은 RSPA에서 실측한 결과를 사용하여 모델링 하였다. 누수 전력량은 동적 전력량의 20%로 하여 모델링 하였으며, 이 수치는 ARM에서 발표한 누수 전력과 동적 전력의 상관관계 문서를 참조한 값이다[7]. 비교를 위해 최신 CGRA 매핑 알고리즘인(하지만 전력소모를 고려하지 않은) EMS (Edge-centric Modulo Scheduling)[5]를 사용했을 때의 전력소모를 같이 측정하였다. 우리의 저전력 매핑 알고리즘은 LPS(Low-Power Scheduling)으로 표현하였다.

〈표 1〉 LPS와 EMS의 II와 사용한 PE의 개수

	EMS		LPS	
	II	사용한PE개수	II	사용한PE개수
Compress	3	16	6	2
GSR	4	16	12	3
Laplace	3	16	9	4
Lowpass	8	16	9	4
SOR	8	16	12	4
Wavelet	2	16	9	4
Swim_calc1	5	16	30	6
Swim_calc2	6	16	48	6
Swim_calc3	3	16	39	4



(그림 4) LPS와 EMS의 전력소모비교

4.2 전력 소모 비교

<표 1>은 각 어플리케이션의 매핑 결과를 보여준다. EMS의 II는 전력소모를 고려하지 않고 구한 최소 II 값을 보여준다. LPS의 II는 Td와 Tc를 고려하여 계산한 Max II를 보여준다. LPS는 수행시간은 변화시키지 않으면서 전체 PE 16개 중 일부만 사용하고, 사용하지 않는 부분은 꺼 놓으면서 전력 소모를 최소화 하였다.

(그림 4)는 CGRA의 PE들에서 소모되는 전력을 비교한 그래프이다. 각 내용은 EMS를 사용한 매핑의 전력소모량을 1로 봤을 때, 상대적인 저전력매핑을 전력 소모량을 그래프에 나타내었다. 사용한 모든 어플리케이션에서 전력 소모량이 감소하였다. 전력소모량은 최대 73%까지 줄었으며(swim 어플리케이션의 calc3 loop의 경우), 평균 56.4%의 감소량을 보여주었다.

5. 결 론

우리는 이 논문에서 데이터 전송 속도를 고려하여, 성능 하락이 일어나지 않는 최대한의 범위에서 PE의 소모량을 줄이는 매핑 방법을 제안하였다. CGRA에서는 전체 성능이 PE의 수행시간에 종속적인 경우도 있지만, CGRA로 데이터를 옮기는 시간에 전체시간이 종속적인 경우도 많다. 이런 경우 PE의 일부만 사용하더라도 전체 수행시간이 늘어나지 않는다. 우리는 어플리케이션을 분석하여 데이터 전송 시간과 수행시간의 관계를 분석한 후 데이터 전송 시간이 큰 경우, 전체 수행시간을 증가시키지 않는 범위 내에서 사용하는 PE의 갯수를 줄이는 방법을 제안하였다. 이 방법에서는 사용하지 않는 PE는 아예 꺼놓고 수행을 하기 때문에 주요한 전력소모의 원인 중 하나인 누수 전력을 최소화 할 수 있었다. 이로 인해 최대 73%, 평균으로 56.4%의 전력 소모를 줄일 수 있었다.

참 고 문 헌

[1] Y. Kim, J. Lee, A. Shrivastava, J. W. Yoon, and Y. Paek, “Memory-aware application mapping on coarse-grained reconfigurable arrays,” in HiPEAC, 2010, pp.171 - 185.

[2] Y. Kim, M. Kiemb, C. Park, J. Jung, and K. Choi, “Resource sharing and pipelining in coarse-grained reconfigurable architecture for domainspecific optimization,” in DATE ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp.12 - 17.

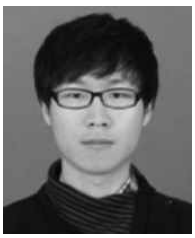
[3] B. R. Rau, “Iterative modulo scheduling: an algorithm for software pipelining loops,” in MICRO 27: Proceedings of the 27th annual international symposium on Microarchitecture. New York, NY, USA:ACM, 1994, pp.63 - 74.

[4] H. Park, K. Fan, M. Kudlur, and S. Mahlke, “Modulo graph embedding:mapping applications onto coarse-grained reconfigurable architectures,”in CASES ’06. New York, NY, USA: ACM, 2006, pp.136 - 146.

[5] H. Park, K. Fan, S. Mahlke, T. Oh, H. Kim, and H. Kim, “Edge-centric modulo scheduling for coarse-grained reconfigurable architectures,” in PACT ’08. New York, NY, USA: ACM, 2008, pp.166 - 176.

[6] B. Mei, S. Vernalde, D. Verkest, H. De Man, and R. Lauwereins 2002. Dresc: a retargetable compiler for coarse-grained reconfigurable architectures. Field-Programmable Technology, 2002. (FPT). Proceedings. 2002 IEEE International Conference on, 166 - 173.

[7] L. Schuth and J. Binney, “Managing power in 45nm and 65nm designs,”Presentation material, ARM and Synopsys. [Online]. Available: http://www.dianzichan.com/anonymous/ic/arm07conf_mgm_pwr45nm.pdf



김 용 주

e-mail : yjkim@optimizer.snu.ac.kr
2006년 서울대학교 전기공학부(학사)
2006년～현 재 서울대학교 전기정보공학부 박사과정
관심분야 : CGRA, MPSoC, 임베디드 시스템, 메모리 최적화

윤 종 희

e-mail : jhyoun@gwnu.ac.kr
2003년 경북대학교 전자전기공학부(학사)
2011년 서울대학교 전기컴퓨터공학부 (박사)
2011년～현 재 강릉원주대학교 컴퓨터공학과 강의전담교수
관심분야 : 임베디드 시스템, 컴파일러 최적화, 소프트웨어 최적화, 아키텍처, MPSoC, GPGPU.

조 두 산

e-mail : dscho@sunchon.ac.kr
2001년 한국외국어대학교 전자제어공학과 (학사)
2003년 고려대학교 전기전자공학과(석사)
2009년 서울대학교 전기컴퓨터공학부 (박사)
2008년～2010년 (주) 리코어스 이사
2010년～현 재 순천대학교 전자공학과 전임강사
관심분야 : 임베디드 시스템 설계/최적화, 메모리 시스템 최적화, 컴파일러, 병렬분산처리

백 윤 흥

e-mail : ypaek@snu.ac.kr

1988년 서울대학교 컴퓨터공학과(학사)

1990년 서울대학교 컴퓨터공학과(석사)

1997년 UIUC 전산과학(박사)

1997년~1999년 NJIT 조교수

1999년~2003년 KAIST 전자전산학교 부교수

2003년~현 재 서울대학교 전기정보공학부 교수

관심분야 : 임베디드 소프트웨어, 컴파일러, MPSoC, CGRA