

이기종 MPSoC 를 위한 태스크 매핑 기법 연구

조중석, 정유진, 조두산
 순천대학교 전자공학과
 e-mail : mew26@snu.ac.kr

A Study of a Task Mapping Technique for heterogeneous MPSoCs

Jungseok Cho, Youjin Jung, Doosan Cho
 Dept. of Electronic Engineering, Suncheon National University

요 약

멀티프로세서 시스템 온칩 (MPSoC) 플랫폼은 고성능 임베디드 시스템을 위한 핵심 구성요소이다. MPSoC 를 구성하는 각각의 처리요소 (processing element, PE)는 대응하는 태스크의 연산 특징에 맞추어 최적화되어 있어야 한다. 갈수록 증가하는 고성능의 요구에 따라 동종 MPSoC 는 각각의 태스크 연산 특징에 최적화된 다양한 PE 를 보유한 이기종 MPSoC 로 발전되어 왔다. 따라서 이기종 MPSoC 의 코어들은 응용에 특화된 맞춤형 명령어 세트로 설계된다. 하지만 이러한 이기종성은 다양한 태스크로 구성된 응용들을 어떻게 서로 다른 특성을 지닌 PE 들에 매핑해야 최적의 시스템을 구성할 지를 결정해야 하는 부담을 컴파일러와 같은 툴에 지우고 있다. 잘못된 매핑은 시스템 성능을 현저히 저하시킬 소지가 있다. 본 연구에서는 멀티미디어 응용 태스크의 연산 패턴을 분석하여 최적의 태스크 매핑을 결정하는 기법을 제안하고 있다.

1. 서론

MPSoC 플랫폼은 고성능 임베디드 시스템에서 사용되는 주요설계 방식이다. 이기종형 MPSoC 플랫폼은 비용 및 전력 소모 개선을 획득하면서 응용 프로그램의 다양한 요구사항을 완벽히 얻을 수 있는 유일한 해결책이 되었다. 이러한 이기종형 MPSoC 를 구성하는 처리 요소 (processing element, PE)는 다양한 형태로 설계 될 수 있다. 예를 들면, 범용 프로세서, DSPs, 응용 특화 하드웨어 가속기, 재구성형 아키텍처 등이 해당된다. 하지만 이러한 맞춤형 PEs 는 최적의 시스템 구성을 위하여 각각의 PE 들이 시너지를 생성할 수 있도록 태스크 매핑 및 스케줄링을 적절히 응용에 맞추어 조절해야 한다. 이 부분이 이기종 MPSoC 의 성패를 좌우하는 가장 어려운 문제라고 할 수 있다. 본 연구에서는 특히 태스크 매핑 문제를 고려하여 최적의 솔루션을 얻을 수 있도록 다양한 시도를 하였다.

확장 가능형 프로세서 코어 (Xtensa from Tensilica [1])는 기본 명령어 세트 아키텍처 (Instruction Set Architecture, ISA)를 갖고 여기에 특정 응용에 특화된 맞춤형 명령어들로 확장 가능하도록 구성되어 있다. 응용에 특화된 명령어들은 해당 응용에서 빈번히 계산되는 연산들의 실행 패턴을 분석하여 이를 고속으로 실행 가능하도록 하는 설계로 구형된다. 결과적으로 이러한 특화 명령어들은 성능 뿐만 아니라 전력 소모면에서도 시스템에 큰 이득을 가져온다. 따라서 스트리밍 데이터를 처리하는 멀티미디어 응용 도메인에서는 이기종형 MPSoC 는 확장 가능형 코어들을 여

러개 포함하여 성능 및 전력 소모에서 많은 이득을 얻을 수 있도록 설계 될 수 있다.

스트리밍 데이터를 입력으로 사용하는 멀티미디어 응용은 다수의 계산 집약적인 커널 혹은 태스크로 구성되고 그림 1 의 예제처럼 이는 비순환 태스크 그래프 (acyclic task graph)로 표현된다.



(그림 1) MP3 인코더 응용의 태스크 그래프

각 태스크들은 지속적으로 입력되는 데이터 스트림을 파이프라인 방식을 통하여 실행할 수 있다. 여러 단계로 구성된 파이프라인을 타고 데이터들은 병렬적으로 서로 다른 이기종 코어에서 처리된다. 우리는 각각의 태스크들이 최적으로 수행될 수 있도록 최적의 코어를 탐색하는 기법을 연구해 보았다.

2. 최적 태스크 매핑 탐색

우리가 개발하고자 하는 프레임워크의 입력은 비순

환 태스크 그래프와 각 태스크의 코드이다. 각 태스크 코드는 어셈블리 언어와 유사한 중간단계 언어 (Intermediate Representation code, IR code)를 사용하였다. 그림 2 에 중간단계 언어의 예제를 나타내었다. $a=a+1$ 의 경우 단일 정수형 (single integer type) 데이터와 상수 1 의 합으로 연산이 수행됨으로 plus:SI 연산으로 표현되었다.

```
a=a+1 의 중간단계 언어 변환 (GCC 컴파일러 사용)
(insn 12 11 10 (parallel [
  (set (mem/c/i:SI (plus:SI
    (reg/f:SI 54 virtual-stack-vars)
    (const int -4 [...])) [...])
  (plus:SI
    (mem/c/i:SI (plus:SI
      (reg/f:SI 54 virtual-stack-vars)
      (const int -4 [...])) [...])
    (const int 1 [...]))))
(clobber (reg:CC 17 flags))
]) -1 (nil))
```

(그림 2) 중간단계 언어 표현 예제

전술된 바와 같이 중간단계 언어는 각각의 응용 태스크에 포함된 연산들을 가상 기계에서 실행 가능한 기본 연산으로 변환하여 표현하기 때문에 기본 연산 패턴을 분석하기 용이한 장점이 있다. 분석된 빈번한 기본연산 패턴을 고속으로 실행시킬 수 있는 기능 유닛 (functional unit)을 보유한 코어에 해당 태스크를 할당하게 하여 최적의 성능을 보장할 수 있다. 예를 들면, FFT 필터 같은 경우 곱하기/더하기 연산이 빈번하게 사용된다. 이러한 경우 DSP 의 MAC(multiply and accumulate) 명령어로 한 번에 실행하는 것이 가능하기 때문에 DSP 코어에 할당하는 것이 시스템 성능에 가장 유리하다. 이를 결정하기 위한 목적 함수는 다음과 같다.

$$\text{Minimize}(\sum_{i=1}^n \text{ExecutionTime}(T_i))$$

T 는 태스크 그래프의 노드 집합을 나타내며 ExecutionTime(T_i)는 노드 T_i 이 코어 i 에 매핑되었을 때 실행시간을 리턴한다. 목적함수는 완전검색 기법을 사용하여 태스크 집합 T 의 모든 원소가 MPSoC 를 구성하는 이기종 코어에 빠짐없이 매핑되었을 때 최소 실행시간을 최종 매핑 솔루션으로 결정하고 탐색을 종료하도록 설계되었다.

우리가 대상으로 선택한 이기종 MPSoC 시스템은 확장 가능형 프로세서 코어들로 구성되어 있다. 이러한 코어들의 경우 각 코어들은 기본적으로 동일한 ISA 집합을 공유하고 있으며, 특정 응용에 특화된 명령어들을 실행하는 특화 기능 유닛들을 하드웨어적으로 추가하여 보유하고 있다. 따라서 태스크 매핑을 진행할 때 계산되는 실행시간은 이러한 확장

된 특화 기능 유닛의 활용도에 따라 다르게 계산되게 된다. 특화 기능 유닛으로 기본 연산 패턴을 할당하는 방식은 패턴 인식 기법을 따른다. 단순한 선형 검색을 사용하여 multiply-accumulate 연산이 연속되게 반복하여 나타난다면 MAC 명령어로 매칭시키는 방식이다.

여기서 추가 고려해야 할 사항은 데이터의 위치에 따른 통신에 대한 오버헤드를 고려하는 것이다. 우리는 문제를 단순화하기 위하여 공유 메모리형 MPSoC 를 선택하였다. 차후 분산 메모리형 시스템을 대상으로 통신 오버헤드를 추가하는 작업을 진행하고자 한다.

3. 실험

우리는 실험을 진행하기 위하여 MP3 와 MPEG2 두 개의 멀티미디어 응용 프로그램을 선택하였다. 해당 응용의 경우 계산 집약적인 커널들을 기준으로 분류하면 총 5 개의 태스크로 구성된다고 할 수 있다. MP3 의 경우 그림 1 과 같으며, MPEG2 의 경우는 다음과 같다. T1:데이터 입력, T2:motion estimation, frame prediction, T3:DCT 변환, T4:출력, T5:역 DCT 변환. 우리는 기본 PE 를 확장 가능형 tensilica xtensa LX2 프로세서 코어로 사용하였다. 이는 면적 231K gates 와 연산속도 326MHz , 0.13um 공정의 특징을 갖는다. 실험대상인 응용 프로그램이 5 개의 태스크로 구성되기 때문에 원활한 파이프라이닝을 위하여 5 개 코어를 기본으로 실험을 진행하였다. 대상 MPSoC 는 서로 다른 특화 기능 유닛으로 설계된 4 개의 DSP 와 하나의 범용 프로세서로 구성되어 있다. MPEG2 의 T2,T3,T5 가 각각의 태스크에 특화된 PE 에 매핑되는지 여부를 테스트하였다. 우리가 설계한 매핑 기법을 통하여 매핑 결과가 본래의 설계에 부합하는 매핑을 하는 것을 최종 확인하였다.

4. 결론

본 연구에서 우리는 이기종 MPSoC 환경에서 효과적인 태스크 매핑 기법을 논의하였다. 이기종 MPSoC 를 위한 태스크 그래프와 코드를 입력으로 하여, 특화된 하드웨어 기능 유닛의 활용도가 높은 연산들로 구성된 태스크들을 해당 PE 에 할당하여 전체 시스템 성능이 최적으로 구성되도록 기법을 설계하였다.

Acknowledgement

이 논문은 2010 년도 정부 (미래창조과학부)의 재원으로 한국연구재단 기초연구사업(2010-0024529), 2013 년도 정부 (교육부)의 재원으로 한국과학창의재단 (대학생 창의융합형 연구과제 지원사업)의 지원을 받아 수행된 연구임.

참고문헌

- [1] R.E. Gonzalez, "A configurable and extensible processor," IEEE Micro, vol. 20, no. 2, 2000.