

임베디드 시스템 설계에서의 전력 소모 최소를 고려한 메모리 접근 코드 스케줄링

황동욱
과학영재학교
tonghwang@hanmail.net

김태환^o
서울대학교, 전기.컴퓨터공학부
tkim@ssl.snu.ac.kr

Memory Access Code Scheduling Considering Energy Minimization in Embedded System Design

Tongwook Hwang
Busan Science Academy

Taewhan Kim^o
School of EECS, Seoul National University

요 약

메모리 관련 수행은 일반적으로, 임베디드 시스템에서의 전체 전력 소모에서 상당한 양을 차지하고 있으며, 임베디드 시스템에서 사용되는 메모리 중 상당수가 에너지 절약을 용이하도록 다중의 작동 모드(예: 액티브, 스탠바이, 뎀, 파워다운) 기능을 지니고 있다. 본 논문에서는 메모리 접근 코드의 스케줄링이 전력 소모에 미치는 관계를 이용함으로써 메모리의 작동 모드를 최대한 활용하는 문제에 대한 해결책을 제시한다. 기존의 방법에서는 (1) 제한된 레지스터 조건을 만족함과 (2) 효율적인 메모리 접근 모드(예: 페이지, 버스트 모드) 활용 등을 후반부의 별개 작업으로 고려한 반면, 본 제안한 방법은 (1)과 (2) 두 요소를 메모리 접근 코드 스케줄링에 긴밀히 결합시켜 전력 소모를 효과적으로 줄이고자 하였다. 벤치마크를 사용한 실험에서 우리가 제안한 방법을 사용하면, 기존의 일방적 우선순위를 기반으로 한 그리디 방식보다 평균 32.13% 더 적은 전력 소모를 가짐을 입증하였다.

1. 서론

주로 전력의 용량이 제한된 전지 등을 이용하는 임베디드 시스템의 설계 시 가장 중요한 것은 성능을 너무 크게 희생시키지 않으면서 전력의 소모를 최대한 줄여서 제한된 전원 용량을 효과적으로 사용하는 것이다. 메모리는 시스템의 성능에 매우 큰 영향을 끼치면서도 전력을 많이 소모하는 부품이다. 따라서 메모리의 전력 소모를 효과적으로 줄이는 것은 임베디드 시스템의 개발에서 매우 중요한 일이다. 최근에는 전력 소모를 줄이기 위해 메모리가 여러 가지 상태 중 하나를 선택할 수 있도록 설계되고 있다. 예를 들면 On/Off의 두 상태 외에도, Standby 모드를 만들어서 On 상태로의 딜레이 감소 등을 피하는 경우가 있다. 이러한 상태들을 최대한 이용해서 메모리의 전력 소모를 최소화하는 분야에 대한 연구가 활발히 진행되고 있다.

본 논문에서는 전력 소모 감소를 위해 개발된 메모리 상태를 이용함과 더불어, 효율적인 메모리 접근 모드를 이용해서 메모리를 저전력 상태로 전환할 수 있는 시간을 늘리고 필요 없는 전력 소모를 더욱 줄이는 것을 목표로 하였다. 본 논문에서는 절전 메모리 상태와 효율적인 메모리 접근 모드를 고려한 전력 소모

최적화 알고리즘을 제안한다.

2. 연구 동기

On/Off 모드를 제외하고도 Standby 모드를 가지고, 연속으로 작은 범위 내의 자료를 Access할 경우 Buffer 등을 통해서 메모리를 통하지 않고 자료를 읽어올 수 있는 Rambus DRAM을 연구 대상으로 하였다.

이 메모리는 On 상태일 때 3.57nJ/cycle의 전력을 소모한다. 그리고 Standby 상태일 때에는 0.83nJ/cycle의 전력을 소모한다. On 상태에서 Standby 상태로의 전환은 전력 소모 없이 순식간에 가능하다. 그러나 Standby 상태에서 On 상태로의 전환은 2 cycle의 시간과 3.57nJ/cycle의 전력을 소모한다. 이 메모리에는 페이징 버퍼가 있다. 접근할 데이터가 버퍼 내에 있으면 메모리 대신 버퍼에서 데이터를 읽으며, 이 때 버퍼는 Standby 상태여도 동작한다. 단, 메모리가 Standby 상태이면 버퍼 액세스 시 전력을 1.43nJ/cycle 소모한다.

1개의 RDRAM을 이용한 임베디드 시스템에서 예제 프로그램을 실행한 경우의 메모리 전력 소모량은 다음

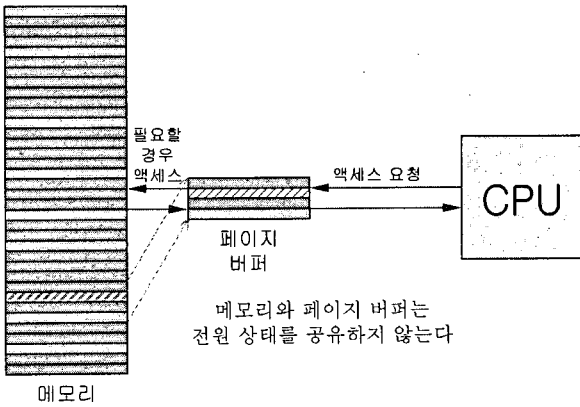


그림 1a

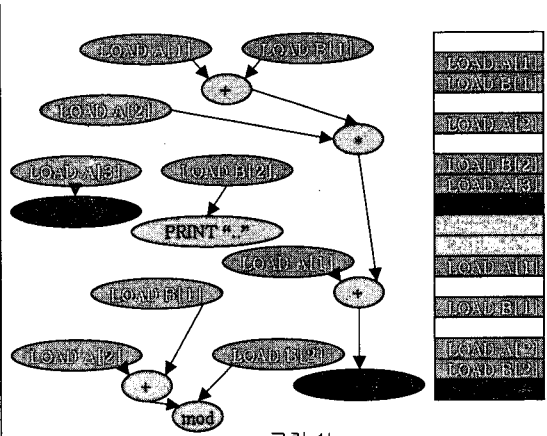


그림 1b

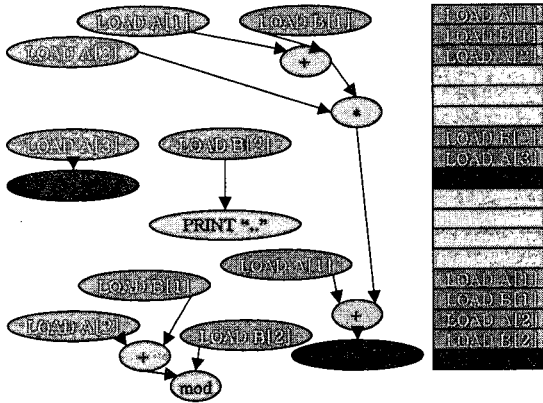


그림 1c

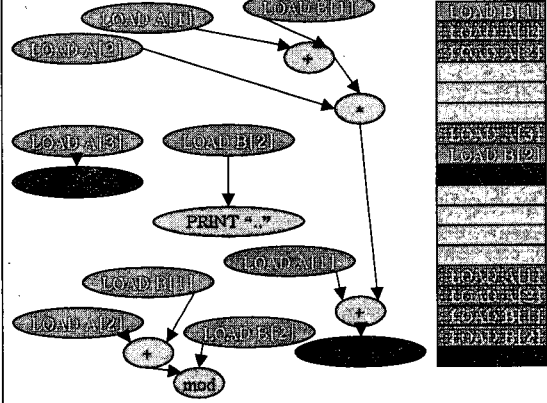


그림 1d

그림 1 Access 위치 조절을 통한 에너지 절약 기법

과 같이 분석할 수 있다. <그림 1b>는 프로그램을 최적화하기 전의 초기 프로그램으로서 총 64.26nJ의 전력을 소모한다. <그림 1c>는 Standby 상태로의 전환만을 고려해서 최적화한 결과이다. 이 경우, 소모되는 총 전력은 56.04nJ으로, 최적화를 수행하기 전과 비교하여 12.79%가 감소했다. 마지막으로, <그림 1d>는 paging mode까지 고려해서 최적화를 수행한 결과이다. 이 경우 소모되는 전력은 48.42nJ로서, 최적화를 수행하기 전보다 전력 소모량이 24.65%가 감소하였음을 알 수 있다.

이 결과는 임베디드 시스템의 CPU에 데이터를 저장해 둘 수 있는 레지스터의 수가 제한이 없다고 계산한 결과이다. 현실적으로 레지스터 개수는 제한이 있다. 그리고 레지스터가 정보를 임시로 저장하는 동안 그것이 손상되면 안 된다. 이런 레지스터 제한 조건을 고려해서

최적화를 수행해야만 실제 임베디드 시스템에서 수행 가능한 프로그램을 구현할 수 있다. 이 논문에서는 메모리 뱅크 수를 1개로 가정한다. 다수의 메모리 뱅크를 사용하는 경우는 간단한 전처리 과정을 통해 쉽게 구현할 수 있다.

3. 제안 알고리즘

제안된 알고리즘은 초기해 생성과 최적화의 이단계로 구성된다.

1단계: 기존의 알고리즘을 사용해서 모든 주어진 제약 조건을 만족하는 초기 Solution을 구한다.

2단계: 이 초기 Solution을 최적화한다. 기본적으로 조금씩 해답이 개선되는 과정을 여러 번 수행하여 최적의 해를 얻게 된다. Memory Access 1개를 다른 위치로 바꾸는 동작을 Move Operation이라 하고, Access 2

개의 위치를 서로 교환하는 것을 Swap Operation이라 한다 (그림 2참조). 이러한 기본 Operation 중 메모리 에너지 소모를 최소화하면서 합법적인 해를 찾아서 선택하고, 방금 움직인 Access는 다시 움직이지 못하도록 고정한다. 고정되지 않은 Access가 없을 때까지 이 과정을 반복하고 그 결과를 현재의 해로 저장한다. 이 과정을 Inner Loop이라고 한다. 이 최적의 해를 초기 해로 설정하여 다시 Inner Loop를 적용한다. 반복 과정은 해가 더 이상 개선되지 않을 때까지 계속된다.

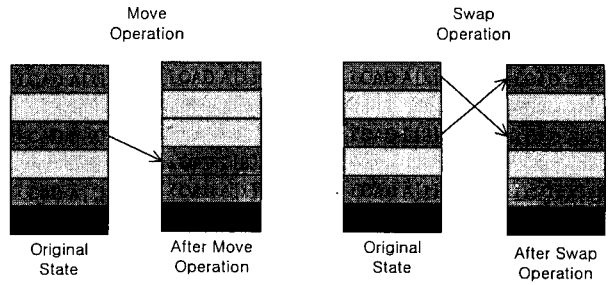


그림 2 Move와 Swap Operation의 개념

4. 실험 결과

논문에 제언한 알고리즘과 그리디 알고리즘을 비교한 결과를 그림 4에 그래프로 나타내었다. 모든 데이터는 레지스터 개수가 2, 3, 4, 5, 무한개인 경우에 대해 실험하였다. 레지스터의 개수가 무한인 경우는 메모리 Access 수보다 레지스터의 수를 더 많이 설정하는 방법으로 수행하였다. 실험 결과 표 2와 같이 레지스터 개수가 무한일 경우는 최대 44.06%, 평균 32.13%의 전력 소모량이 감소되었음을 알 수 있다.

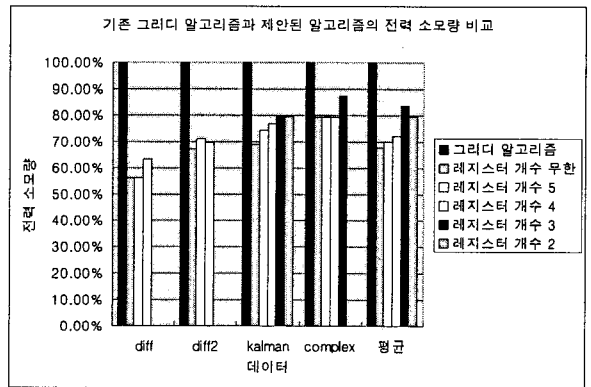


그림 4 그리디와 제안 알고리즘의 전력 소모량 비교

	그리디	제안된 알고리즘				
		INF	5	4	3	2
Diff	76.96	58.38	58.38	66	-	-
Diff2	145.62	97.96	103.24	101.33	-	-
Kalman	104.36	72.08	77.56	80.3	83.04	83.04
Complex	65.92	52.22	52.22	52.22	57.7	-
평균						

표 1 그리디와 제안 알고리즘 전력소모 비교(단위 nJ)

	전력 소모 감소율 (%)				
	INF	5	4	3	2
Diff	44.06	44.06	36.76	-	-
Diff2	32.73	29.1	30.41	-	-
Kalman	30.93	25.68	23.05	20.43	20.43
Complex	20.78	20.78	20.78	12.47	-
평균	32.13	29.91	27.75	16.45	20.43

표 2. 그리디와 제안알고리즘의 전력 소모 감소량

레지스터의 개수가 5, 4, 3개일 경우는 각각 평균 29.91%, 27.75%, 16.45%의 전력 소모 감소율을 보였다. 레지스터의 개수가 2개인 경우는 수행 가능한 프로그램이 Kalman 밖에 없었으므로 평균에 포함하지 않았다.

5. 결론

본 논문에서 메모리 전력 상태와 메모리 접근 모드를 동시에 고려하여 메모리의 전력 소모를 최소화하는 알

고리즘을 제안했다. 이 알고리즘의 실행 시간 복잡도는 Inner Loop가 $O(T^2N \log N)$ 으로 느린 편이나, 이 최적화 과정은 임베디드 시스템의 소프트웨어가 하드웨어에 탑재되기 전 한 번만 수행하면 되므로 실행 시간이 중요한 성능 요소는 아니다. 본 알고리즘은 Paging Mode와 Standby 상태를 지원하는 메모리를 기반으로 하였지만, 다른 메모리 절약 상태와 접근 모드를 가진 메모리에도 쉽게 적용할 수 있을 것이다. 또한 다중 메모리 बैं킹도 적절한 전처리 과정을 수행하여 본 알고리즘을 적용할 수 있을 것이다.

감사의 글: 본 연구는 KAIST 과학영재 교육원의 R&E 프로그램 지원을 받았다.

참조문헌

[1] T. Kim and C. Lyuh, "Memory Access Scheduling and Binding Considering Energy Minimization in Multi-Bank Memory Systems", *DAC*, 2004
 [2] Kris Kaspersky 저, 여인춘 역, "임베디드 메모리 최적화 기법", 에이콘출판사, 2004
 [3] 128/144-MBit Direct RDRAM Data Sheet, Rambus Inc., May 1999