

임베디드 시스템을 위한 개선된 예측 동적 전력 관리 방법

준회원 김 상 우*, 정회원 황 선 영*

An Improved Predictive Dynamic Power Management Scheme for Embedded Systems

Sang-Woo Kim* Associate Member, Sun-Young Hwang* Regular Member

요 약

본 논문은 임베디드 시스템에서 불필요한 전력 소모를 감소하기 위해 개선된 예측 동적 전력 관리 구조와 태스크 스케줄링 알고리즘을 제안한다. 제안된 알고리즘은 불필요한 전력 소모를 최소화하기 위해 미리 스케줄링을 한다. 제안된 예측 동적 전력 관리는 수행 오버헤드를 경감하기 위해서 스케줄링 라이브러리를 제공한다. 실험 결과 제안된 알고리즘은 동적 전력 관리를 적용한 LLF 알고리즘과 비교하여 평균 22.3% 전력 소모 감소를 보인다.

Key Words : DPM; Task Scheduling; Embedded System; Low-power System.

ABSTRACT

This paper proposes an improved predictive dynamic power management (DPM) scheme and a task scheduling algorithm to reduce unnecessary power consumption in embedded systems. The proposed algorithm performs pre-scheduling to minimize unnecessary power consumption. The proposed predictive DPM utilizes a scheduling library provided by the system to reduce computation overhead. Experimental results show that the proposed algorithm can reduce power consumption by 22.3% on the average comparing with the LLF algorithm for DPM-enable system scheduling.

I. 서 론

모바일 기기와 같이 배터리 기반으로 동작하는 임베디드 시스템은 한정된 배터리를 가지고 장시간 동안 사용하는 것이 중요하다. 지난 몇 년간 두드러지게 발전한 고집적 반도체 기술은 시스템 내부에서 저전력 수행을 가능하게 함으로써 위와 같은 문제점을 해결한다. 그럼에도 불구하고 현재 시스템에 대한 다양한 요구를 충족하기 위해 시스템 내부 전력 소모를 최소화할 수 있는 저전력 관리가 중요하다^[1]. 대부분 모바일 기기는 LCD 빛의 광도와 스피커의 볼륨과 같은 시스템의 전력 소모를 결정하는 많은 매개변수들이 있다^[2]. 어플리케이션별 하드웨

어의 전력 관리 방법은 다양한 전력 소모의 매개변수를 고려해야 하므로 구현이 어렵다는 단점이 있다. 이를 극복하기 위해 하드웨어 자원 관리 기능과 수행할 소프트웨어의 총체적인 정보를 갖고 있는 운영 체제 관점에서 전력 관리를 하는 연구가 주목 받고 있다^{[3][4]}.

시스템 수준의 저전력 설계를 위한 연구는 정적인 기법과 동적인 기법으로 나뉜다^[1]. 정적인 기법 중 최근 연구가 활발히 진행되고 있는 ASIP(Application Specific Instruction-set Processor)에 최적화된 재가양성 컴파일러(Retargetable Compiler)는 특정 어플리케이션에 대한 실행 성능 향상과 저전력 수행이 가능한 최적화된 코드를 생성한다^{[5][6]}. 동적인 기법

※ 본 논문은 2009년도 「서울시 산학연 협력사업」의 「다노IP/SoC설계기술혁신사업단」의 지원으로 이루어졌습니다.

* 서강대학교 전자공학과 CAD & ES 연구실 (hwang@sogang.ac.kr)

논문번호: KICS2009-03-085, 접수일자: 2009년 3월 3일, 최종논문접수일자: 2009년 6월 8일

은 시스템의 수행 환경에 능동적으로 대처하는 방법으로 동적 전력 관리와 동적 전압 조절 방법이 제안되었다^{[7][8]}. 동적 전력 관리는 시스템 컴포넌트가 idle 상태를 유지할 경우에 시스템의 전력 상태를 바꿔서 전력 소모를 감소시키는 방법이고 동적 전압 조절은 사용하는 프로세서의 동작 전압과 주파수를 변경하는 방법이다. 실제로 컴파일러를 이용한 저전력을 위한 코드 재구성 방법은 불필요한 전력 소모를 감소시킬 수 있으나, 모바일 기기와 같이 사용자의 관심에 의해 수행 환경이 변하는 경우에는 정적인 기법으로 해결할 수 없으므로 동적인 기법의 적용이 필요하다.

프로세서의 동작 속도를 줄이면 프로세서의 동적 전력은 감소되지만 입출력 장치들의 사용 시간이 증가되어 이들 전력의 소모는 증가한다. 동적 전압 조절 기법은 프로세서의 동적 전력 소모 관점에서 효율적인 방법이지만 시스템의 주변 장치에 대한 동적 전력 소모를 고려하지 않는 문제점이 있다. 이를 해결하기 위해 동적 전력 관리 기법은 주변 장치를 고려한 동적 전력 관리함으로써 불필요한 전력 소모를 감소시킨다. 예를 들어, 무선 센서 네트워크 환경하의 시스템에서 이를 구성하는 각 노드의 저전력 동작이 중요하므로 노드간의 전력 소모를 감소하기 위해 동적 전력 관리 기법을 사용하나 기존의 동적 전력 관리 기법은 전력 소모를 최소화하지 못한다는 단점이 있다^{[9][10]}. 이와 같은 환경에서 제안된 예측 동적 전력 관리 기법은 어플리케이션 관점에서 전력 소모를 최소화 가능한 효율적인 방법을 찾아낼 수 있다는 점에서 의의가 있다^[10].

본 연구에서는 Power-Aware Task Library (PATL)를 이용하여 기존 예측 동적 전력 관리의 시스템을 개선한 시스템 구조와 제안된 시스템에서 효율적으로 수행하는 Predictive Low-power Device Scheduler (PLDS)를 구축하였다. 어플리케이션에 사용되는 태스크들과 입출력 장치들의 전력 소모 정보들로부터 저전력을 위한 태스크 스케줄링 알고리즘을 구축하기 위해서는 다양한 상황에 최소의 전력 소모가 가능한 선택할 수 있도록 태스크와 입출력 장치들의 전력 소모 정보를 통합해야한다. 통합된 최종 정보는 PATL로 구성되고 최대 장치 중첩을 동시에 고려하여 저전력 소모를 가진 태스크 스케줄링이 가능하도록 PLDS 알고리즘에 지원한다.

본 논문의 구성은 다음과 같다. 2절에서는 기존의 동적 전력 관리에 대한 연구에 대해 설명하고, 3절에서는 제안된 예측 동적 전력 관리 시스템,

PATL, 그리고 태스크 스케줄링 알고리즘 (PLDS)에 대하여 설명한다. 4절에서는 제안된 알고리즘의 효율성을 검증을 위해 Instrument Navigation System (INS)의 태스크와 입출력 장치인 후지쯔사의 하드 디스크, TI사의 DSP, SST사의 Flash들의 스펙을 가지고 시뮬레이션을 수행한다. 끝으로 5절에서는 결론을 제시한다.

II. 관련 연구

동적 전력 관리 방식의 목적은 프로세서에 어떤 동작 전압을 제공하는가에 상관없이 기본적으로 입출력 장치에 동작 상태와 sleep 상태의 전력 상태를 가지고 입출력 장치의 불필요한 전력 소모를 줄이는 것이다. Time-out 기반의 동적 전력 관리는 시스템이 특정 시간 T 이상 idle 상태일 때 해당 시스템의 전력을 shut-down 하는 단순한 방식이지만 불필요한 전력 소모가 많다는 단점이 있다^[11]. 특정 시간 T 동안에 낭비되는 전력 소모를 줄이기 위해 다음 idle 상태의 주기를 예측하는 전력 관리 정책이 필요하며 predictive, stochastic, adaptive 방식들이 제안되었다^{[12][13][14][15]}.

Predictive 동적 전력 관리 방식은 디바이스의 idle 시간을 예측하고 예측한 구간이 전력 상태를 전환할 것인지에 대한 기준으로 입출력 장치의 sleep 상태 경우와 전력 상태를 전환한 경우에 전력 소모가 같게 되는 시간인 break-even time보다 길면 디바이스의 전원을 차단하는 방법이다^{[12][13]}. 이 기준으로 동작하는 동적 전력 관리는 장치의 전원 차단여부를 미리 결정하여 태스크들을 수행하므로 전력 낭비를 감소시킨다. Stochastic 동적 전력 관리 방식은 입출력 장치, 장치 사용 요청, 전력 관리자 등을 Markov process를 기반으로 하여 확률적으로 모델링한 방법으로 시스템의 전력 소모가 최소화되도록 입출력 장치의 전력 상태를 결정한다^[14]. Adaptive 동적 전력 관리 방식은 현재 작업하는 상태에서 다음의 idle 상태를 예측하여 가장 적합한 sleep 상태를 선택하는 방법으로 작업량과 상관없이 동적으로 전력 소모가 적은 것을 선택한다^[15]. 이 방식들은 일반적인 시스템에서 전체 동적 전력 소모를 상당히 줄일 수 있으나 실시간 시스템의 제약 조건을 고려하지 않았으므로 태스크의 마감시간의 준수를 보장하지 못한다. 새로운 동적 전력 관리 방식은 전력 소모를 최소화하되 실시간 시스템의 제약 조건을 고려한 태스크 스케줄링 알고리즘이 필요하다.

수행할 태스크에 대해 look-ahead를 이용한 2단계

동적 전력 관리 알고리즘은 정적으로 미리 결정된 태스크의 스케줄과 입출력 장치 사용 정보를 이용하여 마감 시간을 엄수한 태스크를 가지고 각 입출력 장치의 전력 상태를 결정하고 이 결과를 입출력 장치 관리의 전력 알고리즘에 적용한다¹⁶⁾. 위와 같이 수행한 결과는 기존 알고리즘에 비해 전력 소모를 더욱 감소시킬 수 있으나 사용된 두 알고리즘은 독립적으로 수행되지 않으므로 전력 소모의 최소화에 있어서 한계를 보인다. 이 한계를 극복하기 위해 개선된 알고리즘은 최대 장치 중첩도(Maximum degree of device overlap)를 이용하여 정적으로 결정된 태스크 스케줄링 알고리즘을 제시하였으나 시스템에 새로운 태스크가 추가되는 경우는 시스템이 동작할 수 없는 문제점이 있다¹⁷⁾. 이와 같은 문제점을 해결하기 위해 제안된 방법은 기존 Earliest Deadline First (EDF) 혹은 Least Laxity First (LLF) 스케줄링 알고리즘에 적용한 태스크 스케줄링 알고리즘이다¹⁸⁾. 여기서 EDF는 마감 시간이 가장 빠른 태스크가 높은 우선 순위를 갖고 수행되고 LLF는 여유시간이 가장 적은 태스크를 먼저 수행할 수 있게 스케줄링을 한다. 위 알고리즘은 시스템에 존재하는 여유시간과 최대 장치 중첩도를 고려하여 스케줄링을 수행하기 때문에 시스템 이용도 (system utilization)의 값이 클수록 시스템의 여유시간이 감소되므로 전력 소모를 최소화할 수 있는 기회가 줄어드는 한계를 가진다.

III. 제안된 동적 전력 관리 기법

본 절에서는 제안된 예측 동적 전력 관리 시스템과 PLDS 알고리즘을 제시한다.

3.1 제안된 예측 동적 전력 관리 시스템

정확한 예측이 가능한 동적 전력 관리 시스템은 앞으로 수행할 태스크의 정보와 사용될 입출력 장치의 스펙을 동시에 고려하는 태스크 스케줄링 알고리즘을 수행할 수 있다. 그림 1은 제안된 예측 기반 동적 전력 관리 시스템을 보인다.

그림 2는 제안된 예측 동적 전력 관리 시스템의 Power-Aware Task Library (PATL) 기록 내용을 보인다. 사용될 입출력 장치 정보는 정적으로 결정된 PATL을 이용하기 때문에 동적 전력 관리 시스템 부하를 가중시키지 않는다. 차기 스케줄링할 범위 내에 최소의 전력 소모가 가능한 태스크 스케줄링을 하려면 그 범위 내에 스케줄링이 가능한 태스크들의 종합적인 정보가 필요하다. 차기 스케줄링할 범위의 태

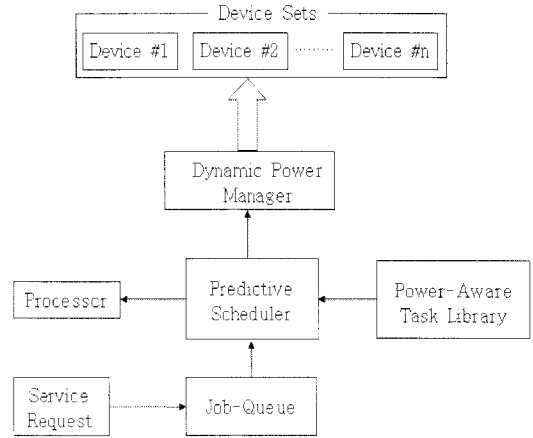


그림 1. 제안된 예측 동적 전력 관리 시스템

스크 개수가 n 이라면 스케줄링이 가능한 n 개 태스크의 집합을 모든 경우의 수만큼 생성하고, 태스크의 여유시간과 스케줄링할 태스크들이 사용할 입출력 장치의 전력 상태를 PATL에 작성한다. n 의 최대값이 클수록 예측 가능한 스케줄링의 범위가 넓어지므로 정교한 스케줄링을 수행할 수 있다는 장점이 있지만, 저장할 내용이 방대하여 라이브러리 크기가 커진다는 단점이 있다. 스케줄링의 정밀성과 저장장치 요구량과의 trade-off를 위해 n 의 최대값 결정이 중요하다.

스케줄링할 태스크 집합의 모든 태스크들이 마감 시간을 반드시 엄수하며 일부의 태스크들을 일정한 시간만큼 연기하거나 앞당겨서 수행하려면 미리 스케줄링할 범위 내에서 태스크들의 여유시간과 최대 장치 중첩도를 분석하고 스케줄링을 수행해야 한다. 위와 같은 조건을 만족시키려면 모든 태스크 중에서 마감시간이 가장 짧은 것을 먼저 수행한 후에 다른 것을 고려해야 한다. PATL에 기록되는 n 의 최대값은 스케줄링할 모든 태스크의 개수이고 기록되는 태

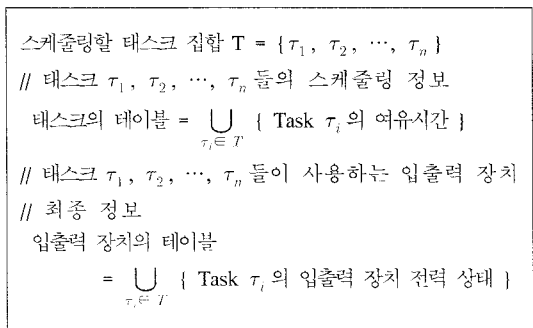


그림 2. 제안된 예측 동적 전력 관리 시스템의 PATL 기록 내용

블 내용들은 스케줄링할 태스크 집합에서 마감시간이 최소인 태스크의 마감시간까지 스케줄링된 정보를 가진다.

입출력 장치들과 태스크들 간의 상호 연관 관계 정보를 정적으로 작성한 PATL은 predictive scheduler에 정보를 제공하여 저전력을 위한 태스크 스케줄링을 수행하도록 한다. 제안된 스케줄러는 Predictive Low-power Device Scheduler (PLDS) 알고리즘을 이용하여 최소 전력 소모를 위한 태스크 스케줄링을 제공한다.

3.2 PLDS 알고리즘

차기 스케줄링할 범위 내에 있는 태스크들과 현재 입출력 장치들의 전력 상태 정보들은 PATL과 최대 장치 중첩도를 이용하여 마감시간을 엄수하며 전력 소모가 가장 적은 태스크들의 집합을 얻는다. 동적 계획법(dynamic programming)과 같이 제한된 수행 기간 동안 최소 전력을 위한 태스크 스케줄링을 시스템의 전체 수행시간으로 확장한 경우는 전력 소모를 크게 감소시킬 수 있다^[9].

동일한 입출력 장치를 수행하는 모든 태스크들의 장치 중첩에 대한 정보를 알아내기 위해서는 마감시간이 엄수된 태스크 sequence에서 장치 중첩 구간 l_j 을 구하여 이용한다. 제안된 장치 중첩 구간 l_j 는 불필요한 전력 소모가 없는 일련의 태스크들이 수행되는 구간이다. 그림 3은 동일한 입출력 장치에서 태스크 $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5$ 를 수행시 측정된 장치 중첩 구간 l_1, l_2, l_3, l_4 를 보인다. 장치 중첩 구간 기준으로 break-even time은 입출력 장치에서 불필요한 전력 소모가 없는 최대 구간 l_{be} 이다^[2]. 모든 태스크들의 동작하지 않은 구간이 l_{be} 보다 크면 불필요한 전력 소모를 하므로 장치 중첩 구간 l_j 로 구분된다. 동일한 입출력 장치를 사용하는 태스크들에 대해, 차기 스케줄링할 범위 내에 존재하는 최대 중첩 구간 l_{max} 는 $\max(l_1, l_2, \dots, l_j)$ 가 된다. l_{max} 의 태스크 집합 T 는 단일 입출력 장치 집합 K 를 고려한 최대 장치 중첩으로써 전력 소모의 최소화가 가능하지만, 이보다 더 최소화시킬 수 있으므로 모든 입출력 장치 $\kappa_1, \kappa_2, \dots, \kappa_n$ 을 고려해야 한다. 모든 입출력 장치를 고려하며 각 입출력 장치에 대한 l_{max} 의 태스크 집합 $T_{overlap}$ 은 $\{\tau_1, \tau_2, \dots, \tau_n\}$ 가 된다.

그림 4는 제안된 PLDS 알고리즘의 pseudo 코드를 보인다. 모듈에 적용되는 차기 스케줄링할 범위는 현

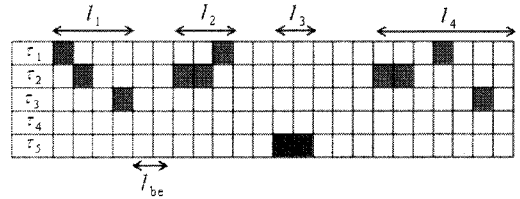


그림 3. 동일한 입출력 장치를 사용하는 태스크 sequence에서 장치 중첩 구간

재 주어진 태스크들의 마감시간이 최소인 태스크의 마감시간까지를 말한다. Device_Overlap 모듈은 차기 스케줄링할 범위 내에 $T_{overlap}$ 을 생성하고 LLF_scheduling 모듈은 차기 스케줄링할 범위 내에 LLF 스케줄링을 수행한 결과를 얻어내고 Consumed_Power 모듈은 스케줄링할 태스크 집합 T 의 모든 태스크들이 소모되는 전력 값을 얻을 수 있다. PATL_Taskset 모듈은 매개 변수인 태스크 집합 T_n 가 PATL의 스케줄링할 집합 T 로써 입출력 장치의 테이블에서 입출력 장치 κ_n 의 최대 장치 중첩 구간을 가진 태스크의 테이블에서 태스크 집합을 얻을 수 있다. 만약 태스크 집합을 얻지 못한 경우는 마감시간을 엄수하지 못한 것이다.

제안된 PLDS 알고리즘은 스케줄링 시점 전에 차기 스케줄링할 범위 내에 저전력 스케줄링이 가능한 태스크 집합을 찾아내며 이들이 마감시간을 엄수하

Procedure PLDS ()

```

begin
    T_overlap = Device_Overlap ( );
    T = LLF_scheduling ( );
    // T는 차기 스케줄링할 범위 내에 LLF 스케줄링
    // 하고 결과를 얻는다.
    bound = Consumed_Power ( T );
    // bound는 T의 모든 태스크들이 소모되는 전력 값을
    // 얻는다.
    for ( n = 1; n ≤ 입출력 장치의 총 개수; n++ )
        begin
            T_temp = PATL_Taskset ( T_n );
            // T_n은 입출력 장치 κ_n의 최대 장치 중첩을
            // 고려한 태스크 집합 T_overlap이다.
            if ( Consumed_Power ( T_temp ) < bound ) then
                T = T_temp;
                bound = Consumed_Power ( T_temp );
            end if
        end for;
    return T;
end;

```

그림 4. 제안된 PLDS 알고리즘의 pseudo 코드

면 스케줄링을 수행하고 그렇지 않으면 LLF 스케줄링을 수행한다.

IV. 실험 결과

제안된 PLDS 알고리즘의 전력 소모에 대한 효율성을 평가하기 위해 실생활에 사용되는 Instrument Navigation System (INS)의 태스크를 사용하였다^[20]. 표 1은 실험에 사용되는 INS의 태스크 집합으로 사용한 6개의 태스크들을 보인다. 표 2는 후지쓰사의 하드 디스크, TI사의 DSP, SST사의 Flash를 모델링한 것으로써 각 입출력 장치의 동작 상태의 전력(P_a), sleep 상태의 전력(P_s), shut-down 상태의 전력(P_{sd}), wake-up 상태 전력(P_{wu}), 전력 상태 전환에 소모되는 시간(t_{sd} 와 t_{wu})를 보인다^{[21][22][23]}.

표 3은 INS 태스크 집합의 모든 태스크들이 사용할 입출력 장치들을 고려하여 태스크 스케줄링 알고리즘에 수행되는 실험 목록을 보인다. 주어진 알고리즘의 우수성을 보이기 위해 네 가지 실험을 수행하였다. [실험 1]은 모든 태스크들이 용량이 가장 큰 하드디스크만 사용하고, [실험 2]는 태스크의 이용도가 클수록 용량이 큰 입출력 장치를 사용하고, [실험 3]은 태스크의 이용도가 작을수록 용량이 큰 입출력 장치를 사용하며, [실험 4]는 무작위로 생성된 입출력 장치를 사용한다.

표 4는 표 3의 모든 실험을 수행한 결과를 보인다

표 1. INS의 태스크 집합

| Task | Execution Time (ms) | Period (ms) | Deadline (ms) | Utilization |
|--------------------|---------------------|-------------|---------------|-------------|
| τ_1 | 1.2 | 2.5 | 2.5 | 48.0 % |
| τ_2 | 4.3 | 40.0 | 40.0 | 10.8 % |
| τ_3 | 10.3 | 62.5 | 62.5 | 16.5 % |
| τ_4 | 20.3 | 1000.0 | 1000.0 | 2.0 % |
| τ_5 | 100.3 | 1000.0 | 1000.0 | 10.0 % |
| τ_6 | 25.0 | 1250.0 | 1250.0 | 2.0 % |
| System Utilization | | | | 89.3 % |

표 2. 실험에 사용되는 입출력 장치 목록

| Device | P_a (W) | P_s (W) | P_{sd} (W) | P_{wu} (W) | t_{sd} (sec) | t_{wu} (sec) |
|--------|-----------|-----------|--------------|--------------|----------------|----------------|
| HDD | 0.95 | 0.13 | 0.54 | 1.61 | 0.67 | 2.72 |
| DSP | 0.63 | 0.2 | 0.4 | 0.4 | 0.5 | 0.5 |
| Flash | 0.125 | 0.001 | 0.05 | 0.05 | 0.01 | 0.01 |

표 3. INS의 태스크들이 사용된 입출력 장치

| | τ_1 | τ_2 | τ_3 | τ_4 | τ_5 | τ_6 |
|-----------------|-----------------|----------|----------|----------|----------|----------|
| Utilization (%) | 48.0 | 10.8 | 16.5 | 2.0 | 10.0 | 2.0 |
| [실험 1] | HDD | | | | | |
| [실험 2] | HDD | DSP | HDD | Flash | DSP | Flash |
| [실험 3] | DSP | HDD | DSP | Flash | HDD | Flash |
| [실험 4] | Random (50번 수행) | | | | | |

표 4. 제안한 알고리즘과 LLF 알고리즘의 전력 소모 비교

| | LLF 적용시 전력 소모 (mW) | 제안된 PLDS 적용시 전력 소모 (mW) | 전력 비교 (%) |
|--------|--------------------|-------------------------|-----------|
| [실험 1] | 29.4 | 23.8 | - 19.0 |
| [실험 2] | 19.9 | 14.7 | - 26.1 |
| [실험 3] | 17.7 | 13.9 | - 21.5 |
| [실험 4] | 18.5 | 14.3 | - 22.7 |

다. [실험 1]에서 PLDS 알고리즘이 LLF의 알고리즘에 비해 모든 실험 중에서 가장 낮은 19.0%의 전력 소모 감소를 보이고, [실험 2]와 [실험 3]에서 PLDS 알고리즘이 LLF 알고리즘에 비해 각각 26.1%, 21.5% 전력 소모가 감소를 보였으며, 실험 결과 태스크의 이용도가 높을수록 용량이 큰 입출력 장치를 사용하면 효율적임을 알 수 있다. [실험 4]는 무작위로 생성된 입출력 장치를 50번 수행한 것으로 PLDS의 알고리즘이 LLF의 알고리즘에 비해 평균 22.7% 전력 소모가 감소하였음을 보인다. 제안된 PLDS 알고리즘은 DPM이 적용된 LLF 스케줄링 결과에 비해 평균 22.3%의 전력 소모가 감소한다.

그림 5는 표 4의 결과를 그래프로 보인다. 그림 5를 통해 제안된 PLDS 알고리즘은 시스템에 사용되는 입출력 장치와 태스크의 관계에 상관없이 LLF의 알고리즘을 적용한 경우보다 최소의 전력 소모가 가능하다. 이는 태스크들과 입출력 장치들의 스펙을 중

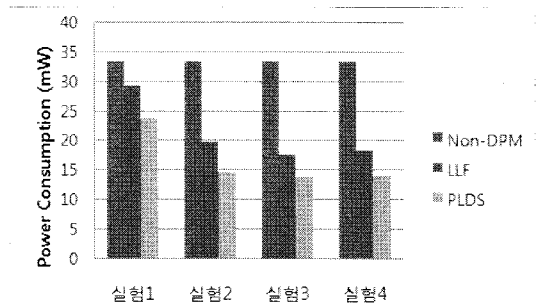


그림 5. LLF와 PLDS의 전력 소모 결과 비교

합적으로 분석하여 스케줄링을 하면 동적 전력 관리를 통해 불필요한 전력 소모를 감소시킬 수 있는 기회가 증가한다.

V. 결 론

본 논문에서는 실시간 시스템에서 다양한 입출력 장치의 전력 상태 전환을 최소화하기 위해 예측 가능한 범위 내에 있는 태스크들의 마감시간을 엄수하고 장치 중첩도를 최대화할 수 있는 예측 기반 태스크 스케줄링 알고리즘을 제안하였다. 수행되는 태스크들과 입출력 장치에 대한 종합적인 정보를 PATL(Power-Aware Task Library)에 미리 구축한다. 제안된 PLDS 알고리즘은 차기 스케줄링할 범위에서 각 입출력 장치의 최대 중첩 구간과 PATL의 태스크 집합을 적용한 결과를 이용하여 전력 소모가 적은 태스크 집합을 선정한다. 이와 같이 반복적으로 스케줄링을 하여 장치의 전력 상태 전환 횟수를 최소화함으로써 불필요한 전력 소모가 감소된다. 실생활에 사용되는 INS의 태스크 집합과 모델링된 입출력 장치에 대해 실험한 결과 동적 전력 관리 방식을 적용한 LLF 스케줄링 알고리즘의 경우보다 평균 22.3%의 전력 소모가 감소되었으며 시스템 이용도가 크더라도 전력 소모의 최소화가 가능할 것으로 기대된다.

참 고 문 헌

[1] J. Rabaey and M. Pedram, Eds., *Low Power Design Methodologies*, Kluwer Academic Pub., 1996.

[2] E. Macii, "Dynamic Power Management of Electronic Systems", *IEEE Design & Test of Computers*, vol. 18, no. 2, pp. 6-9, Mar. / Apr. 2001.

[3] Y. Lu, L. Benini, and G. De Micheli, "Power Aware Operating Systems for Interactive Systems", *IEEE Trans. VLSI Systems*, vol. 10, no. 2, pp. 119-134, April 2002.

[4] Y. Lu, L. Benini, and G. De Micheli, "Operating-system Directed Power Reduction", in *Proc. Int. Symp. Low Power Electronics and Design*, pp. 37-42, July 2000.

[5] 이성래, 황선영, "머신 행위기술로부터 Retargetable 컴파일러 생성시스템 구축", 한국

통신학회논문지, 32권 5호, pp. 286-294, 2007년 5월.

[6] 이성래, 황선영, "Application에 최적의 ASIP 설계를 위한 효율적인 Architecture Exploration 방법", *한국통신학회논문지*, 32권 9호, pp. 913-921, 2007년 9월.

[7] Y. Shin and K. Choi. "Power-Conscious Fixed Priority Scheduling for Hard Real-Time System", in *Proc. Design Automation Conference*, pp. 134-139, June 1999.

[8] S. Lee and T. Sakurai, "Run-time Voltage Hopping for Low-power Real-Time Systems", in *Proc. Design Automation Conference*, pp. 806-809, June 2000.

[9] L. Benini, A. Bogliolo, and G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management", *IEEE Trans. VLSI Systems*, vol. 8, no. 3, pp. 299-316, June 2000.

[10] A. Sinha and A. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks", *IEEE Design & Test of Computers*, vol. 18, no. 2, pp. 62-74, Mar. / Apr. 2001.

[11] R. Golding, P. Bosh, and J. Wilkes, "Idleness is not Sloth", in *Proc. Winter USENIX Technical Conf.*, pp. 201-212, Oct. 1995.

[12] C. Hwang and A. Wu, "A Predictive System Shutdown Method for Energy Saving of Event-Driven Computation", in *Proc. Int. Conf. Computer-Aided Design*, pp. 28-32, Nov. 1997.

[13] M. Srivastava, A. Chandrakasan, and R. Brodersen, "Predictive System Shutdown and other Architectural Techniques for Energy Efficient Programmable Computation", *IEEE Trans. VLSI Systems*, vol. 4, no. 1, pp. 42-55, Mar. 1996.

[14] L. Benini, A. Bogliolo, G. Paleologo, and G. De Micheli, "Policy Optimization for Dynamic Power Management", *IEEE Trans. Computer-Aided Design*, vol. 18, no. 6, pp. 813-833, June 1999.

[15] E. Chung, L. Benini, and G. De Micheli, "Dynamic Power Management Using Adaptive Learning Tree", in *Proc. Int. Conf. Computer-Aided Design*, pp. 274-279, Nov. 1999.

[16] V. Swaminathan and K. Chakrabarty, "Energy-conscious, Deterministic I/O Device Scheduling in Hard Real-time Systems", *IEEE Trans. Computer-Aided Design*, vol. 22, no. 7, pp. 847-858, July 2003.

[17] V. Swaminathan and K. Chakrabarty, "Pruning-based, Energy-optimal, Deterministic I/O Device Scheduling for Hard Real-time System", *ACM Trans. Embedded Computing Systems*, vol. 4, no. 1, pp. 141-167, Feb. 2005.

[18] 이원규, 황선영, "실시간 시스템에서 효율적인 동적 전력 관리를 위한 태스크 스케줄링 알고리즘에 관한 연구", *한국통신학회논문지*, 31권 4A호, pp. 393-401, 2006년 4월.

[19] D. Travers and R. Kaye, "Dynamic Dispatch by Constructive Dynamic Programming", *IEEE Trans. Power Systems*, vol. 13, no. 1, pp. 72-78, Feb. 1998.

[20] D. Katcher, H. Arakawa, and J. Strosnider, "Engineering and Analysis of Fixed Priority Schedulers", *IEEE Trans. Software Eng.*, vol. 19, no. 9, pp. 920-934, Sept. 1993.

[21] *MHL2300AT Hard Disk Drive Product Manual*, Fujitsu Co., 2000.

[22] *TMS320C6411 Power Consumption Summary*, Texas Instruments Co., 2003.

[23] *SST Multi-Purpose Flash SST39LF020*, SST Co., 2005.

김 상 우 (Sang-Woo Kim)

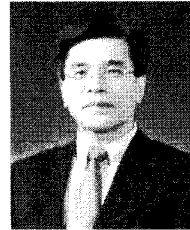
준회원



2009년 2월 서강대학교 전자공학
학과 졸업
2009년 3월~현재 서강대학교
전자공학과 석사과정
<관심분야> ASIP Design,
Retargetable Compiler for
Embedded System

황 선 영 (Sun-Young Hwang)

정회원



1976년 2월 서울대학교 전자공
학과 졸업
1976년 2월 한국과학원 전기
및 전자공학과 공학석사 취득
1986년 10월 미국 Stanford 대
학교 전자공학 박사학위 취득
1976년~1981년 삼성 반도체(주)

연구원, 팀장

1986년~1989년 Stanford 대학 Center for Integrated
Systems 연구소 책임 연구원 및 Fairchild
Semiconductor, Palo Alto Research Center 기술
자문
1989년~1992년 삼성전자(주) 반도체 기술자문
2002년4월~2004년3월 서강대학교 정보통신대학원장
1989년 3월~현재 서강대학교 전자공학과 교수
<관심분야> SoC 설계 및 framework 구성, CAD
시스템, Embedded 시스템, DSP 시스템 설계 등