

논문 2008-45CI-1-9

주파수 평활화 기법을 이용한 전력 관리 알고리즘

(A Power-Aware Scheduling Algorithm by Setting Smoothing Frequencies)

권혁성*, 안병철**

(Hyek Seong Kweon and Byoung Chul Ahn)

요약

대부분의 전력 관리 기법에 대한 연구는 동작 주파수나 동작 전압을 변화시켜 시스템의 가용성을 증가시키는데 집중되었다. 이 방법은 전력소모를 줄이기 위해 구동 주파수를 수시로 변경하므로 실 시스템의 성능을 저하시킨다. 소비 전력을 줄이기 위한 다른 방법은 구동 주파수의 개수를 제한하거나 실행 시간 동안 전체 주파수를 완만하게 관리하여 시스템의 성능을 높이는 것이다. 허나 기존 연구는 실시간 응용 프로그램에는 적합하지 않다. 본 논문에서는 실시간 시스템에서 소비 전력의 손실을 줄이고 시스템의 성능을 향상시키기 위해 구동 주파수 레벨의 할당 방법과 주파수 평활화 기법을 이용한 전력 관리 기법을 제안한다. 이 기법은 연속적인 주파수 간격에서 알고리즘이 예측한 주파수를 양자화된 주파수로 매핑한다. 이러한 주파수 평활화 기법은 좁은 주파수 대역에서 잦은 주파수 변경으로 발생한 시스템의 오버헤드를 줄일 수 있다. 매핑된 주파수는 넓은 주파수 간격에서 주파수 불일치로 발생하는 전력소모를 줄일 수 있다. 모의실험을 통하여 제안한 알고리즘이 기존의 CC RT-DVS 기법에 비해 최대 40%, 평균 15% 정도 소비 전력을 줄일 수 있었다.

Abstract

Most researches for power management have focused on increasing the utilization of system performance by scaling operating frequency or operating voltage. If operating frequency is changed frequently, it reduces the real system performance. To reduce power consumption, alternative approaches use the limited number of operating frequencies or set the smoothing frequencies during execution to increase the system performance, but they are not suitable for real time applications. To reduce power consumption and increase system performance for real time applications, this paper proposes a new power-aware schedule method by allocating operating frequencies and by setting smoothing frequencies. The algorithm predicts so that frequencies with continuous interval are mapped into discrete operating frequencies. The frequency smoothing reduces overheads of systems caused by changing operating frequencies frequently as well as power consumption caused by the frequency mismatch at a wide frequency interval. The simulation results show that the proposed algorithm reduces the power consumption up to 40% at maximum and 15% on average compared to the CC RT-DVS.

Keywords : Real time System, Low Power Management, Dynamic Voltage Scaling, Scheduling

I. 서론

최근 무선 인터넷 환경의 확산과 더불어 스마트폰 및 PDA(Personal Digital Assistant)와 같은 이동용 내장 시스템이 대중화 되고 있다. 이러한 시스템은 기존의

고유 기능 외에 멀티미디어 시청과 같은 복잡한 응용 프로그램을 수행한다. 장시간 사용하기 위해서 대용량의 전원 장치를 사용해야 한다. 그러나 배터리의 용량이 제한되어 있어 효율적으로 전력을 관리할 방법이 필요하다. 이러한 저전력 관리 기법 중 동적 전력 관리 기법(Dynamic Power Management)과 동적 전압 조절 기법(Dynamic Voltage Scaling)은 현재까지 많이 이용되고 있다^[1~2]. 특히 동적 전압 조절 기법은 실행하는 프로그램의 동작 유형이나 특성을 파악하여 시스템의 이

* 학생회원, ** 평생회원, 영남대학교 컴퓨터공학과
(Dept. of Computer Science and Engineering,
Yeungnam University)

접수일자: 2007년10월24일, 수정완료일: 2008년1월11일

용률을 예측하고 이를 바탕으로 전력 소모량을 줄이는 기법이다. 일반적으로 CMOS 회로의 전력 소비는 공급 전원의 제곱과 구동 주파수에 비례하며 회로의 구동 주파수는 공급 전압에 비례한다^[7]. 현재 널리 사용되고 있는 프로세서들은 CMOS 회로를 많이 사용하고 있어서 동적 전압 조절 기법은 시스템의 이용률이 높지 않은 시스템에서는 소비 전력을 많이 줄일 수 있다.

동적 전압 조절 기법을 태스크의 엄격한 마감 시간을 가지는 실시간 시스템에 적용할 경우, 태스크의 유휴 시간을 활용하는 한도 내에서 구동 주파수를 설정해야 한다. 실시간 시스템에서 동적 전압 조절 기법의 효율성은 마감 시간을 만족하면서 태스크의 유휴 시간을 최대한 이용하는 것에 의해 좌우 된다. 이러한 기법은 1994년 Weiser에 의해 처음 제시 되었으며^[1], 이후 이 개념을 바탕으로 한 많은 알고리즘이 제시되고 모의실험이나 실제 구현을 통하여 검증 되었다^[3]. Yao는 EDF Scheduling 기법에서 정적 분석 방법을 이용하여 최적의 주파수를 설정하는 방법을 제시하였다^[4]. Pering은 각 Thread별 마감 시간과 예측된 시스템 이용률을 바탕으로 구동 주파수를 설정하는 동적 분석 기법을 제시하였다^[5]. Pillai는 동적 전압 조절 기법을 실시간 시스템에 적용하였고, Static, Cycle-Conserving 그리고 Look-Ahead RT-DVS 알고리즘을 제시하였다^[6]. 이 중 Look-Ahead RT-DVS 알고리즘은 현재까지 알려진 동적 전압 조절 기법 중에서 소비 전력을 최대 줄일 수 있는 알고리즘으로 알려져 있다^[11]. 그러나 현재까지의 전력 관리 연구는 마감 시간을 만족하면서 태스크의 유휴 시간을 최대한 이용하는 방법에 많이 집중되어 있다.

동적 전압 조절 기법에서 소비 전력을 줄이기 위해 몇 개의 공급 전압과 구동 주파수가 필요한지 그리고 공급 전압과 구동 주파수를 어디에 배치해야 하는지에 대한 연구도 이루어졌다^[8~9]. Ishihara는 최소 2개의 공급 전압과 구동 주파수가 있으면 최적의 소비 전력을 구현할 수 있다고 제시하였다^[8]. 그러나 이 기법은 태스크의 실제 사용 시간을 먼저 예측해야 하는데, 프로그램의 실행에 따른 태스크의 주기나 태스크의 실제 사용 시간이 자주 변하여 초기 예측 값과 차이가 있을 수 있어서 구현할 경우 소비 전력을 줄이는데 한계가 있다. Govil은 구동 주파수를 예측하는 것 보다 전체 주파수를 완만하게 관리 하는 방법이 전력 소비를 줄일 수 있음을 모의실험을 통하여 증명하였다^[9]. 그러나 이 기법은 실시간 특성을 가진 시스템에서는 사용할 수 없다는

단점이 있다^[11].

구동 주파수 변경에 대한 Overhead가 없는 이상적인 시스템에서는 구동 주파수 레벨의 개수가 많으면 시스템의 평균 이용률을 바탕으로 예측한 값과 유사한 구동 주파수를 설정할 수 있어서 태스크에 의해 사용하지 않고 소비되는 전력을 줄일 수 있다. 적용되는 전력 관리 알고리즘의 성능에 따라서 절감할 수 있는 소비 전력에는 많은 차이가 있고, 어떤 전력 관리 알고리즘은 구동 주파수 레벨의 개수가 많을수록 소비 전력이 증가하는 경우도 있다^[6]. 반면에 구동 주파수 레벨의 개수가 적은 경우, 실제 태스크에 의해 사용되지 못하고 소비되는 전력이 증가한다^[11].

본 논문에서는 동적 전압 조절 기법에서 자주 발생하는 서로 밀접하게 관련되고 정량화되어 있는 구동 주파수의 구간에 따라 발생하는 잦은 구동 주파수의 변경을 막고, 일정한 개수의 구동 주파수 레벨을 시스템의 평균 이용률을 기준으로 배치하여 넓은 구간의 구동 주파수 레벨로 인해 발생하는 소비 전력의 손실을 막는 전력 관리 기법을 제안한다. 이는 경성 실시간 시스템에서 사용 가능하다.

본 논문의 구성은 다음과 같다. II장에서는 본 논문에서 사용될 시스템과 전력 모델에 대하여 논한다. III장에서는 본 논문에서 제시하는 전력 관리 알고리즘에 대하여, IV장에서 III장에서 언급된 알고리즘의 성능을 평가하기 위한 시뮬레이션 환경 및 결과에 대하여 논한다. 마지막으로 V장에서는 결론과 향후 연구 방향에 대하여 논한다.

II. 시스템 전력 모델

1. 프로세서 모델

본 논문에서는 프로세서의 CMOS 회로 특성을 바탕으로 한 전력 모델을 이용한다^[7]. 소비 전력은 식(1)과 같이 공급 전압(V_{dd})의 제곱에 비례하고 구동 주파수(f)에 비례하며, 구동 주파수는 식 (2)와 같이 공급 전압에 비례한다. 따라서 구동 주파수를 낮추면 공급 전압을 낮출 수 있다.

$$P_{dynamic} \propto C_{ef} \cdot V_{dd}^2 \cdot f \quad (1)$$

$$f \propto \frac{(V_{dd} - V_{th})^2}{V_{dd}} \quad (2)$$

소비 전력은 식 (3)과 같이 gate switching에 따른 동적 전력 소비(dynamic power consumption)와 누설 전류(leakage current)에 의한 정적 전력 소비(static power consumption)로 구성된다. 시스템이 Idle 상태일 때, 구동 주파수를 낮추기 위해 공급 전압(V_{dd})을 임계 전압(V_{th}) 이하로 낮추면 누설 전류(leakage current)가 발생한다^[7].

$$P_{total} = P_{dynamic} + P_{static} \quad (3)$$

최저 공급 전압(V_{dd})를 0.9V 까지만 낮추면, 누설 전류에 의한 정적 전력 소비는 발생하지 않는다고 가정한다. 따라서 본 논문에서는 gate switching에 의한 동적 전력 소비에 대하여 논한다.

2. 태스크 모델

본 논문에서 사용되는 태스크 모델은 주기를 가진 실시간 태스크 집합을 이용한다^[10]. 태스크 집합은 $\Gamma = \{\tau_1, \tau_2, \dots, \tau_n\}$ 으로 표기한다. 각 태스크는 주기(T), 마감 시간(D) 그리고 최악 실행 시간(WCET, worst case execution time)(C)로 구성되고, $\tau_i = \{T_i, D_i, C_i\}$ 로 표기한다. 각 태스크는 주기를 가지며, 선점적(Preemptive)이며, 상호 독립적이다. 시간 t_0 에서 시작되고 마감 시간은 태스크의 주기와 일치 한다고 가정한다. $V(t_i)$ 와 $F(t_i)$ 는 각각 시간 t_i 에서의 공급 전압이고 구동 주파수를 의미한다. Hyper Period는 H 로 표기하며, 모든 태스크 주기의 최소 공배수로 정의한다. 각 시스템의 성능 평가는 0에서 H 까지를 기준으로 한다. 따라서 시간 t_0 에서 시간 t_H 까지의 소비 전력은 식 (4)와 같다. 이 때 k 는 상수이고, C_k 는 시스템의 컨텐서 용량이다.

$$P[0 \dots H] = \sum_{i=0}^H k \cdot C_k \cdot V(t_i)^2 \cdot F(t_i) \quad (4)$$

III. 제안된 전력 관리 기법

동적 전압 조절 기법은 각 태스크의 실행 시작과 종료 때 시스템의 최대 이용률을 바탕으로 구동 주파수를 설정하며, 전체 소비 전력은 식 (4)과 같이 변경된 구동 주파수와 공급 전압의 제곱의 곱으로 이루어진다. 따라서 소비 전력이 최소가 되기 위해서는 공급 전압의 $V(t_i)$ 값이 같거나 차이가 최소가 되어야 한다. 즉, 구동

전압은 구동 주파수에 비례하므로, 동일한 주파수로 시스템을 구동하거나 변경되는 주파수의 차이가 적어야 한다^[5]. 동적 전압 조절 기법은 구동 주파수의 차이가 적을 경우, 구동 주파수를 자주 변경해야 하므로 실제 시스템에서는 구동 주파수 변경에 따른 Overhead가 증가하므로 구동 주파수 변경 횟수를 적절히 제한하여 Overhead(SW)을 줄여야 한다.

본 논문에서는 동적 전압 조절 기법에서 자주 발생하는 조밀한 양자화된 구동 주파수의 구간에 따라 발생하는 잦은 구동 주파수의 변경을 막고, 일정한 개수의 구동 주파수 레벨을 시스템의 평균 이용률을 기준으로 배치하여 넓은 구간의 구동 주파수 레벨로 인해 발생하는 소비 전력의 손실을 막는 전력 관리 기법을 제안한다.

1. 연구 배경

동적 전압 조절 기법을 이용하여 시스템의 소비 전력을 줄일 경우, 시스템의 소비 전력은 <그림 1>과 같이 전력 그래프를 이용하여 나타낼 수 있다. 시스템에서 사용되는 전체 소비 전력은 실제 태스크가 사용하는 소비 전력(P_U)과 정량화된 구동 주파수에 의해 실제 태스크에서 사용되지 않고 소비되는 전력(P_W)과 구동 주파수 변경(SW_n)에 따른 Overhead에 의해 소비 되는 전력(P_O)으로 구성된다.

동적 전압 조절 기법을 이용하는 시스템에서 소비 전력을 줄이는 방법은 크게 두 가지가 있다. 하나는 정량화된 구동 주파수에 의해 태스크가 사용하지 않고 소비되는 전력(P_W)을 감소시키는 것이다. P_W 를 줄이기 위해서는 구동 주파수 레벨의 구간 간격을 줄여야 한다. 즉, 구동 주파수 레벨의 개수를 증가시키는 것이다. 다른 하나는 구동 주파수의 변경으로 인해 소비되는 전력(P_O)을 감소시키는 것이며, P_O 를 줄이기 위해서는 구동

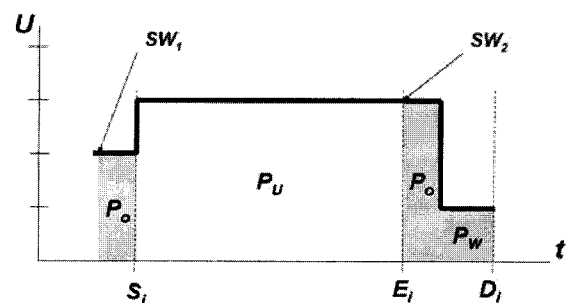


그림 1. 동적 전압 조절 기법에서의 소비 전력
Fig. 1. The classified power consumption in Dynamic Voltage Scaling.

주파수 레벨의 구간을 적절히 배치하여 구동 주파수의 변경 횟수를 줄이는 것이다. 그러나 P_W 을 감소시키기 위해 구동 주파수 레벨의 개수를 증가시키면, 구동 주파수의 변경 횟수가 증가하여 P_O 가 증가한다. 이상적인 시스템에서는 P_O 값이 없거나 적어서 무시 할 수 있지만, 실제 시스템에서는 P_O 의 값을 무시할 수 없다. 따라서 실제 시스템에서 단순히 구동 주파수 레벨의 개수를 증가시키는 방법은 시스템의 소비 전력을 줄이는데 큰 효과가 없음을 알 수 있다.

본 논문에서는 P_W 을 감소시키기 위해 전체 주파수 레벨의 개수를 증가시키지 않으면서 소수의 구동 주파수 레벨을 시스템의 평균 이용률을 중심으로 설정하여 구동 주파수변경에 따른 Overhead도 같이 줄여서 전체 소비 전력을 줄일 수 있는 전력 관리 알고리즘을 제안 한다.

2. 알고리즘

본 논문에서 제안하는 전력 관리 알고리즘은 크게 두 부분으로 나누어진다. 하나는 과거 시스템의 이용률을 바탕으로 현재 시스템 이용률의 평균/최대값을 예측하는 부분과 다른 하나는 예측된 평균 이용률을 바탕으로 구동 주파수 레벨을 평활하게 다시 설정하는 부분이다.

<그림 2>는 본 논문에서 제안하는 전력 관리 알고리즘이다. 제안한 전력 관리 알고리즘은 Pillai가 제시한 Cycle Conserving RT-DVS(이하 cc-EDF) 알고리즘에 일정한 시간(Hyper Period) 마다 구동 주파수 레벨을 다시 설정하는 부분을 추가한 것이다.

시스템의 이용률을 예측하는 부분은 태스크가 시작 될 때, 제안한 전력 관리 알고리즘의 실행 가능성을 위하여 태스크의 최악 실행 시간(WCET)을 이용하여 시스템의 이용률을 계산한다. 이러한 각 태스크의 최악 실행 시간의 합을 U_{max} 라 하고 이를 저장한다. 태스크 종료의 경우, 태스크의 실제 실행 시간을 이용하여 시스템의 평균 이용률을 계산하여 저장한다. 이러한 각 태스크의 평균 이용률의 합을 U_{ave} 라 한다.

예측된 정보를 바탕으로 구동 주파수의 레벨을 평활화 하는 방법은 <그림 3>과 같이 시스템의 이용률의 정보가 없을 경우와 있을 경우로 구분된다. t_0 에서 H_1 까지 실행될 경우, 시스템의 이용률에 대한 정보가 없으므로, 구동 주파수는 시스템에서 제공하는 구동 주파수 레벨을 사용한다. $F = \{f_1, f_2, f_3, f_4\}$ 이고, $f_4 = f_{max}$ 이고 $f_{i+1} - f_i = f_{max}/n$ 이다(n 은 주파수 수).

H_1 에서는 t_0 에서 H_1 까지 저장된 시스템의 이용률 병

```

// 시스템 이용률(오버헤드 포함)에 적합한 최저 주파수 선택
set_speed () {
    use lowest frequency
     $f_i \in \{f_1, \dots, f_m \mid f_i < \dots < f_m\}$ 
    such that  $U_1 + \dots + U_n + m \cdot SW \leq f_i / f_m$ 
    ( $0 \leq m \leq 2$ )
}

// 매 Hyper Period 마다 주파수 테이블을 다시 조정함.
adjust_frequency_table (Max, Ave) {
     $F_0 = Ave; F_1 = (Max + Ave)/2;$ 
     $F_2 = Max; F_3 = f_m$ 
}

// 최초 사용될 주파수 테이블
set_default_frequency_table () {
     $F_0 = 25; F_1 = 50; F_2 = 75; F_3 = f_m$ 
}

// 태스크 시작 할때
On_task_start () {
    set  $U_i$  to  $C_i/T_i$ 
    if (time = 0)
        set_default_frequency_table();
    else if (!(time mod HyperPeriod))
        adjust_frequency_table(Max, Ave);

    set_speed();
}

// 태스크 마감 할때 각 태스크의 실행 Profile을 마련함
on_task_end () {
    set  $U_i$  to  $cc_i/T_i$ 
    Save the Max/Ave. of  $U_i$ 
    set_speed();
}
    
```

그림 2. 제안한 전력 관리 알고리즘
Fig. 2. A proposed power-aware algorithm.

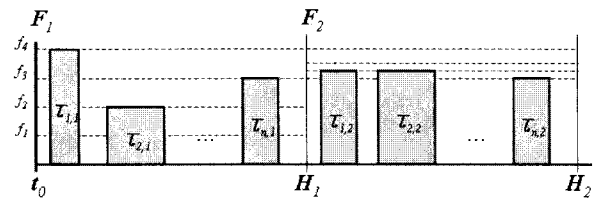


그림 3. 제안된 알고리즘에서 주파수 평활화 기법 예
Fig. 3. An example of frequency smoothing in the proposed algorithm.

보, U_{max} 와 U_{ave} 를 바탕으로 주파수 테이블을, $F = \{f_1, f_2, f_3, f_4\}$, $f_4 = f_{max}$, $f_3 = U_{max}$, $f_2 = (U_{max} + U_{ave})/2$, $f_1 = U_{ave}$ 로 다시 설정한다. 최저 구동 주파수인 f_1 을 시스템의 평균 이용률인 U_{ave} 로 설정하는 것은 조밀한 구동 주파수 구간으로 인한 잦은 구동 주파수 변경을 막는다. 이를 주파수 평활화 기법이라 한다.

태스크 i 의 시작의 경우, 구동 주파수 설정을 위해 식 (5)을 이용하여 시스템 이용률($U(t_i)$)을 산출한다. 따라서 구동 주파수 변경은 태스크의 시작과 종료에 각각 1번씩 발생할 수 있으며, 최대 2회까지 발생할 수 있다. 이 경우 <그림 1>과 같이 구동 주파수 변경에 따른 Overhead(SW), 즉, 실행 시간이 지연된다. 따라서 구동 주파수 변경으로 인한 Overhead를 반드시 고려해야 한다. 이를 위하여 식 (5)과 같이 구동 주파수 변경에 따른 Overhead를 시스템의 이용률에 더하여 계산한다.

$$U(\tau_i) = \sum_{1, k \neq i}^n U_k + \frac{C_i(\text{or } cc_i) + SW}{T_i} \quad (5)$$

제안한 전력 관리 알고리즘에서 시스템 평균 이용률을 산출하는 방법은 구동 주파수 배치의 기준이 되므로 매우 중요하다. 따라서 시스템 평균 이용률은 식 (6)과 같이 Hyper Period를 기준으로 산출된 시스템의 이용률 정보를 누적하여 사용할 수도 있다. 그러나 본 논문에서는 이전에 정보만을 이용하여 시스템의 이용률을 산출한다.

$$A = \left(\sum_1^N \left(\sum_{i=1}^n \left(\sum_{j=H/T_i, (n-1)}^{H/T_i, n-1} U(\tau_{i,j}) \right) \right) \right) \cdot 100 \quad (6)$$

IV. 모의실험

본 논문에서 제안한 전력 관리 기법의 성능을 모의실험을 통하여 평가한다. 성능 평가는 시스템의 이용률 변화에 따른 평가와 구동 주파수 변경에 따른 Overhead 증가에 따른 평가로 이루어진다.

1. 실험 환경

본 논문에서 제안한 전력 관리 알고리즘의 성능을 평가하기 위하여 RTSim 0.5.1을 이용한다^[12]. RTSim은 RETIS Lab에서 실시간 시스템의 스케줄링 평가를 위하여 개발되었으며 GPL에 의거하여 누구나 이용하거나 수정이 가능하다^[12]. 모의실험에서 사용하는 구동 주파수 레벨은 <표 1>와 같이 PXA 270 프로세서에서 제공하는 구동 주파수 레벨을 이용한다^[13]. PXA 270 프로세서는 휴대용 멀티미디어 시스템과 PDA 등 많은 휴대 장비에 사용되며, 교육용 장비로도 많이 보급되어 본 논문에서 제시하는 알고리즘에 대한 구현이 향후 쉽게 이루어 질 수 있다. 모의실험에서 본 논문에서 제안하는 전력 관리 알고리즘의 성능 평가를 위해 Pillai가 제안한 cc-EDF 기법을 이용한 두 가지의 구동 주파수 레벨 배치 방법을 사용한다^[6]. 이 방법은 <표 1>의 PXA 270에서 제공하는 48개를 이용하여 배치한 방법과 <표 1>에서 제공하는 레벨 중 25%, 50%, 75%, 100%의 4개를 이용하여 배치 방법으로 구성된다. 이를 DVS-48과 DVS-4로 각각 표시한다. 제안한 전력 관리 알고리즘은 <표 1>에서 제공하는 구동 주파수 레벨 중에 4개를 선택하여 설정한다. 이를 A-DVS로 표시한다.

표 1. 모의실험에서 사용될 구동 주파수 레벨*
Table 1. The set of frequency level used in the simulations.

주파수 ¹	전압 ¹	주파수	전압	주파수	전압
13	0.85	221	1.16	429	1.36
26	0.86	234	1.17	442	1.37
39	0.87	247	1.19	455	1.39
52	0.87	260	1.20	468	1.40
65	0.88	273	1.21	481	1.41
78	0.89	286	1.22	494	1.42
91	0.89	299	1.24	507	1.44
104	0.9	312	1.25	520	1.45
117	0.93	325	1.26	533	1.46
130	0.96	338	1.27	546	1.47
143	0.99	351	1.29	559	1.49
156	1.02	364	1.30	572	1.50
169	1.06	377	1.31	585	1.51
182	1.09	390	1.32	598	1.52
195	1.12	403	1.34	611	1.54
208	1.15	416	1.35	624	1.55

¹ 주파수의 단위는 MHz이고 전압의 단위는 V이다

표 2. 모의실험에서 사용되는 테스트 집합
Table 2. The set of task used in the simulations.

TASK	PERIOD ¹	WCET ²	AET
A	10	2	(50%~90%) * WCET
B	20	3	(50%~90%) * WCET
C	30	6	(50%~90%) * WCET
D	40	6	(50%~90%) * WCET

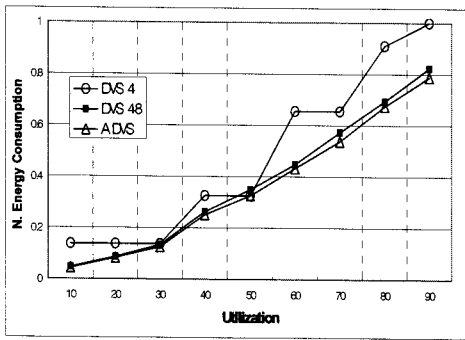
¹ 단위는 ms임.

² 전체 이용률 70% 기준임.

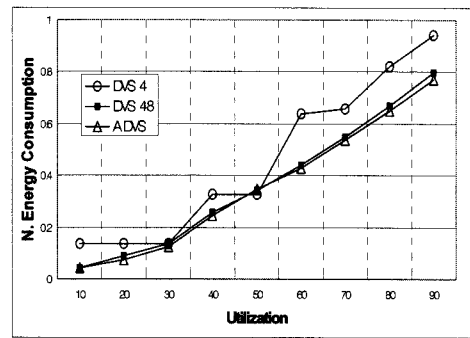
본 논문에서 사용되는 태스크의 집합은 <표 2>와 같으며, 4개의 실시간 태스크로 구성되어 있다. 각 실시간 태스크의 마감 시간은 주기와 일치한다고 가정한다. 모의실험에서는 시스템의 이용률을 10% ~ 90%까지 변화 시켜서 알고리즘의 성능을 평가한다. 이 경우, 최악 실행 시간은 시스템 이용률에 따라 변하며, 실제 사용 시간은 최악 실행 시간의 50%에서 90% 사이의 값으로 적용한다.

2. 모의실험 결과 평가

<그림 4>은 시스템 이용률의 변화에 따른 각 전력 관리 알고리즘의 성능을 평가한 것이다. 실제 사용 환경에 근접한 모의실험 환경에서 성능을 평가하기 위해 시스템의 이용률을 10%에서 90%까지 변화 시키고, 실제 이용 시간을 최악 실행 시간의 80%와 60%로 정한다. 또한, 식 (5)에서 사용될 구동 주파수 변경에 따른 Overhead(SW)는 PXA 270의 구동 주파수 변경 Overhead와 외부 전압 정류 장치인 Texas Instrument사의 TSP62400를 이용한 전압 변환 Overhead를 적용



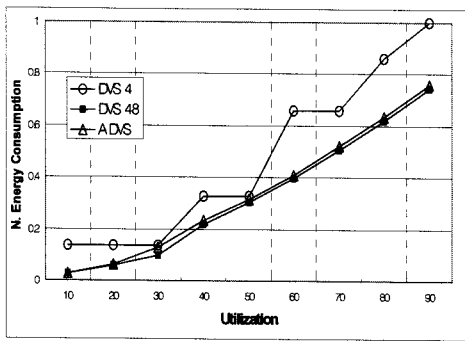
(a) AET = 80% WCET



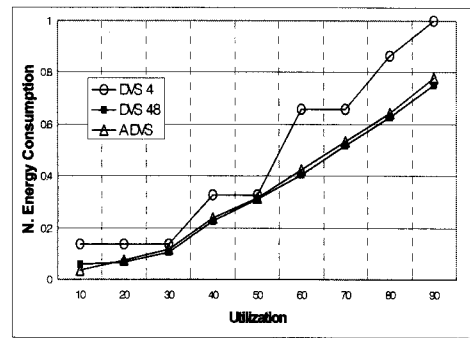
(b) AET = 80% WCET

그림 4. 시스템 이용률 변화에 따른 성능 평가

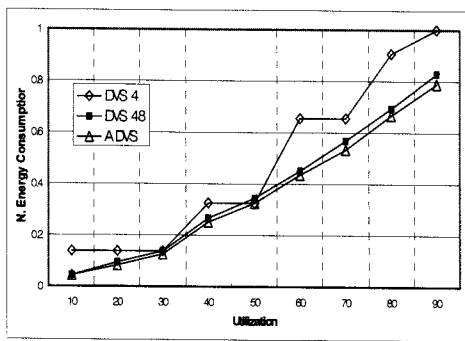
Fig. 4. The performance of the proposed algorithm varying the system utilization.



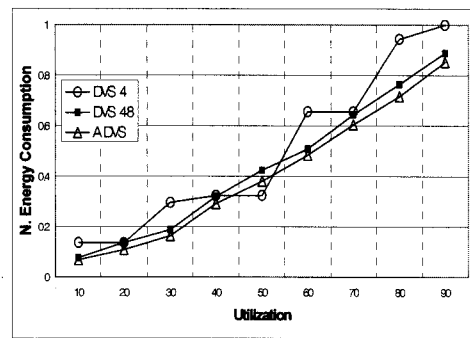
(a) Overhead = 0µs



(b) Overhead = 100µs



(c) Overhead = 500µs



(d) Overhead = 1ms

그림 5 Transition overhead에 따른 성능 평가

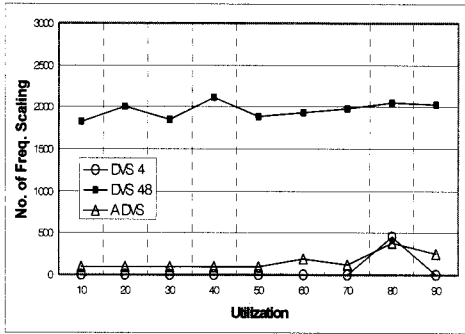
Fig. 5. The performance of the proposed algorithm varying a transition overhead.

한다^[14]. 각 overhead는 150µs와 350µs이다. 그러나 모 의실험에서는 구동 주파수가 변할 경우 공급 전압도 같 이 변한다고 가정하고 구동 주파수 변경에 따른 Overhead를 500µs로 정한다.

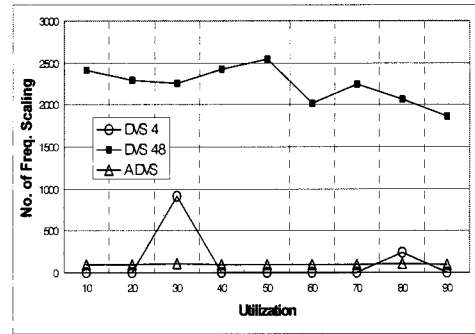
실제 이용 시간이 80%인 그림 4(a)에서는 실제 이용 시간이 60%인 그림 4(b)보다 전력을 많이 소비하는 것 을 알 수 있다. 이는 시스템의 이용률에 따른 것이다. 제안한 전력 관리 알고리즘이 DVS-48 보다 전력을 적 게 소비하는 것을 알 수 있다. 이는 DVS-48 알고리즘

에서 사용하는 구동 주파수의 간격이 좁아서 시스템의 평균 이용률을 기반으로 구동 주파수를 산출 할 경우, 구동 주파수의 변경이 잦아 이로 인한 Overhead에 따 른 소비 전력(P_o) 증가에 따른 것이다.

이를 증명하기 위해서 <그림 5>는 각 전력 관리 알 고리즘에서 구동 주파수 변경에 따른 Overhead에 의한 소비 전력(P_o)에 대한 평가한 것이다. 이 때, 구동 주파 수 변경에 따른 Overhead는 0µs, 100µs, 500µs, 1ms로 각각 정한다. 시스템의 이용률을 10% ~ 90%까지 변화



(a) Overhead = 500µs



(b) Overhead = 1ms

그림 6. Transition overhead에 따른 전압 변경 횟수

Fig. 6. The number of voltage scaling varying a transition overhead.

시키고 실제 이용 시간은 최악 실행 시간의 70%로 정한다. 다른 실험 조건은 <그림 4>의 실험 조건과 같다.

그림 5(a), 그림 5(b)와 같이 구동 주파수 변경에 대한 Overhead가 없거나 작을 경우, DVS-48 방식을 이용한 알고리즘의 소비 전력이 제안한 전력 관리 기법보다 적음을 알 수 있다. 그러나 그림 5(c), 그림 5(d)와 같이 Overhead가 증가할수록 DVS-48 방식을 이용한 전력 관리 기법의 소비 전력도 증가함을 알 수 있다. 이는 주파수를 자주 변경하여 발생하는 소비 전력(P_o)의 증가로 인한 것이다.

<그림 5>에서 DVS-48을 이용한 전력 관리 알고리즘의 소비 전력 증가는 구동 주파수를 자주 변경하여 소비 전력(P_o)의 증가에 따른 것이다. 따라서 <그림 6>는 각 전력 관리 알고리즘의 구동 주파수 변경 횟수에 대하여 조사한 것이다.

모의실험의 조건은 <그림 5>의 실험 조건과 같다. DVS-48 방식을 이용한 전력 관리 알고리즘은 많은 수의 구동 주파수 레벨 때문에 구동 주파수를 자주 변경해야 한다. 그러나 본 논문에서 제안한 알고리즘은 DVS-4를 이용한 전력 관리 알고리즘 보다는 구동 주파수 변경 횟수가 많지만, DVS-48 방식보다는 구동 주파수 변경 횟수가 적다는 것을 알 수 있다. 따라서 구동 주파수 레벨의 수를 증가시킬 경우, 구동 주파수 변경에 따른 Overhead가 없는 이상적인 시스템에서는 소비 전력이 감소하지만, 실 시스템과 같이 Overhead가 있는 경우는 오히려 소비 전력이 증가하는 것을 알 수 있다.

본 논문에서 제안하는 전력 관리 알고리즘은 시스템의 이용률을 바탕으로 구동 주파수 레벨을 다시 설정하여 소비 전력을 줄이며, 최저 구동 주파수를 시스템의 평균 이용률로 설정하여 시스템의 이용률 변화에 따

른 낮은 구동 주파수 변경을 방지하여 실 시스템에서 소비 전력을 줄일 수 있음을 확인하였다. 소비 전력은 DVS-4와 DVS-48 알고리즘과 비교할 때, 최대 40%, 평균 15% 정도 줄일 수 있었다.

V. 결론

본 논문에서는 구동 주파수의 구간이 조밀할 경우 구동 주파수를 자주 변경해야 한다는 점과 실 시스템에서는 구동 주파수 변경에 따른 Overhead가 크다는 점을 바탕으로 주파수를 완만하게 관리하는 평활화 기법을 이용하여 구동 주파수의 변경 횟수를 줄여 시스템의 소비 전력을 줄이는 기법을 제안하였다. 이 기법은 시스템의 평균 이용률을 기준으로 구동 주파수를 배치함으로써 P_w 의 전력 손실을 줄일 수 있다. 또한, 적은 수의 구동 주파수를 이용하므로 구동 주파수의 변경 횟수를 줄여 P_o 의 전력 손실을 감소시킬 수 있다. 따라서 전체 시스템의 소비 전력을 줄여 시스템의 사용 시간을 연장할 수 있다. 모의실험의 결과에서 실 시스템에서 이용되는 구동 주파수 변경에 의한 Overhead를 적용한 경우, DVS-48을 이용한 전력 관리 기법 보다 최대 10%까지, DVS-4를 이용한 전력 기법 보다 최대 40%까지 소비 전력을 감소시켰다.

향후 본 논문에서 제안한 전력 관리 알고리즘을 휴대용 내장 시스템에서 많이 사용되는 PXA 270 프로세서와 embedded Linux에서 구현하여 성능을 평가할 필요가 있다.

참 고 문 헌

[1] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," Proc. of the 1st USENIX Symposium on Operating Systems Design and Implementation (OSDI '94), 1994.

[2] IBM and Montavista Software, "Dynamic Power Management for Embedded Systems," <http://www.research.ibm.com/ar1/projects/dpm.html>, November 2002.

[3] D. Grunwald, P. Levis, C.B.M. III, M. Neufeld, and K.I. Farkas, "Policies for dynamic clock scheduling," Proc. of the 4th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2000), 2000.

[4] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU Energy," IEEE Annual foundations of computer science, 1995.

[5] T. Pering, T. Burd, and R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," 1998.

[6] P. Pillai and K.G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," Proc. of the 18th Symposium on Operating Systems Principles (SOSP 2001), 2001.

[7] S. Borkar, "Design Challenges of Technology Scaling," IEEE Micro, Vol. 19 No. 4, pp. 23-29, July/August, 1999.

[8] T. Ishihara, and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," Proc. of 1998 Int'l symposium on Low power electronics and design, 1998.

[9] K. Govil, E. Chan, and H. Wasserman, "Comparing algorithms for dynamic speed-setting of a low-power CPU," Proc. of the 1st ACM International Conference on Mobile Computing and Networking, 1995.

[10] C. L. Liu, and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a hard real time environment," Journal of the Association for Computing Machinery, v.20, no.1, 1973.

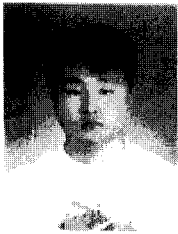
[11] H. Kweon and B. Ahn, "An Efficient Power-Aware Scheduling Algorithm in Real Time System," IEEE PACRIM, August, 2007.

[12] RTsim homepage, <http://rtsim.sssup.it/>

[13] Intel, PXA270 Processor Electrical, Mechanical, and Thermal Specification

[14] TEXAX INSTRUMENT INC, TPS62400 user's guide.

저 자 소 개



권혁성(학생회원)
1995년 영남대학교 전산공학과
학사 졸업
1997년 영남대학교 전산공학과
석사 졸업
1997년~2006년 LG전자
DDC연구소 책임연구원

2005년~현재 영남대학교 컴퓨터공학과 박사과정
<주관심분야 : 실시간시스템, 저전력 관리, 디지털 TV>



안병철(정회원)
1976년 영남대학교 전자공학과
학사 졸업
1986년 오레곤주립대 전기 및
컴퓨터공학 석사 졸업
1989년 오레곤주립대 전기 및
컴퓨터공학 박사 졸업

1976년~1984년 국방과학연구소연구원
1989년~1992년 삼성전자 수석연구원
1992년~현재 영남대학교 전자정보공학부 교수
<주관심분야 : 임베디드시스템, 실시간운영체제, 멀티미디어처리>