

# 가변 전압 프로세서를 사용하는 실시간 시스템에서 소비 전력감소를 위한 전압조정

## Voltage Scaling for Reduced Energy Consumption in Real-Time Systems Using Variable Voltage Processor

이 용 준\*, 김 용 석\*\*  
Yong-Jun Lee\* and Yong-Seok Kim\*\*

**Abstract** - Energy consumption has become an increasingly important consideration in designing real-time embedded systems. In this paper, we propose a voltage scaling method to reduce energy consumption in fixed priority real-time systems using variable voltage processors. The Hyperperiod of tasks is divided into domains. The most suitable voltage of each domain is determined off-line and stored in a table. During task execution, the voltage of processor is adjusted according to the information of the table. A simulation result shows that the proposed method can reduce 80% of power consumption in comparison to no power management. The difference to the optimal EDF based method is only 5%.

**Key Words** : Real-Time, Embedded System, Fixed Priority Scheduling, Power, Voltage

### 1. 서 론

근래에 들어서 배터리 등의 제약된 전원 공급 장치를 사용하는 실시간 시스템에서의 에너지 소모에 관한 연구가 관심 사항으로 부각되고 있다. 하지만 대부분의 연구가 EDF (Earliest Deadline First) 스케줄링 방식의 기반 시스템을 목적대상으로 삼고 있다. EDF는 이론적으로 현재까지의 실시간 스케줄링 방식 중 가장 많은 이점을 가지고 있는 방식이며 전압관련 부분에서도 뛰어난 성능을 보여주고 있다.[1] 하지만 EDF방식은 현재 실제적인 시스템에 거의 사용하지 않고 있기 때문에 연구에 대한 실질적인 적용의 한계가 있게 된다. 이와 반대로 RM방식은 현재 대다수의 시스템에서 사용하는 고정우선순위 방식을 기반으로 한다. 본 논문에서는 RM방식을 기반으로 한 실시간 시스템에서 소비 전력감소를 위한 전압조정 기법을 소개하고, 실제적인 시스템의 적용을 용이하게 하고자 하는데 목적이 있다.

### 2. 전압 조정 기법

실시간 시스템에서 각 태스크의 주기 내에서 최악의 경우의 연산량을 실행하고 남은 시간을 여유시간이라고 한다. 실시간 태스크는 정해진 종료시한까지 실행을 종료하면 되기 때문에 여유시간을 기반으로 가변전압 프로세서의 전압을 적절히 조절함으로써 태스크의 실행시간은 여유시간(이하 슬랙) 한도 내에서 증가하지만, 전체적인 시스템의 전력 소모량

(주) 본 논문은 BK21의 지원에 의해 이루어 졌음.

저자 소개

\* 李 鎬 俊 : 江原大學 컴퓨터情報통신工學科 碩士課程

\*\* 金 龍 碩 : 江原大學 컴퓨터情報통신工學科 教授

을 감소시킬 수 있다. 실시간 시스템에서 가변전압 프로세서를 사용할 경우에 소모되는 전력은 다음과 같이 근사적으로 표현된다.[3]

$$P \approx C \cdot V^2 \cdot f \quad (1)$$

( $f$  = 클럭 주파수,  $V$  = 공급전압,  $C$  = 커패시턴스)

여기서  $f$ 는 클럭 주파수를 나타내는데 프로세서의 클럭 주파수는 대체로 공급 전압에 비례하고 소비 전력은 공급전압의 제곱에 비례한다. 따라서 전압을  $1/N$ 로 낮추면 클럭 주파수(이하 속도)도  $1/N$ 로 줄어들고 단위 시간당 소비 전력은  $1/N^3$ 로 감소하게 된다. 대신에 전체 실행시간은  $N$ 배로 늘어나므로 전체 작업을 완료하는데 소모되는 전력량(Wh)은  $1/N^2$ 로 감소하게 된다.[2]

$$W_h \propto V^2 \quad (2)$$

일반적으로 사용하는 단순한 전압 조정 기법에는 수행할 태스크가 없을 때 프로세서를 휴면상태로 전환하는 기법이 있다.(Sleep on No Task : SONT). 수행할 태스크가 있으면 정상적인 전압으로 실행을 하고 남아있는 슬랙 시간동안 프로세서는 쉬게 된다. 수행할 태스크가 없을 경우는 단순히 전압을 0으로 설정한다.

STA(Single Task Allocation)는 슬랙이 발생한 시점을 사전에 알고 있을 경우 슬랙 직전의 태스크가 슬랙을 모두 사용하는 방식이다. 슬랙의 크기가  $d$ 이고 슬랙을 할당할 태스크의 최악의 시간을  $w_{\text{ect}}$ 라고 하면 전압은 다음과 같다.[3]

$$S^{STA} = \frac{w_{\text{ect}}}{w_{\text{ect}} + d} \quad (3)$$

### 3. 제안하는 기법

본 논문에서 제안하는 슬랙공유(SSPM : Slack Sharing Power Management)는 고정 우선순위 실시간 태스크 스케줄링 시스템에서 가변 전압 프로세서를 사용하여 전력소모를 줄이기 위한 전압조정 기법이다. SSPM은 사전에 태스크들 간의 초월주기 구간에 대하여 최적의 전압으로 수행할 수 있는 범위들을 사전(오프라인)에 계산하여 테이블에 저장하고 태스크들을 실행할 때(온라인)에는 테이블의 정보를 이용한다. 제안된 기법은 상대적으로 낮은 전압으로 수행할 수 있도록 하고 테이블에 저장되는 범위를 넓게 나누어 테이블의 크기가 커지지 않도록 하였고, 온라인 시에 전압을 낮추기 위한 계산 작업을 제거함으로써 온라인상에서 오버헤드를 줄일 수 있다.

#### 3.1 연구배경

전력소모는 슬랙을 여러 태스크들에 나누어 할당함으로써 더욱 감소시킬 수 있다. 초월주기 구간은 적절한 크기의 세부구간인 도메인들로 나누어지고 하나의 도메인에는 여러 개의 태스크 인스턴스들이 포함된다. 전압은 도메인별로 결정되는데 도메인에 포함되는 태스크 인스턴스들이 마감시간을 만족하는 한도 내에서 최적의 값으로 결정된다. 도메인은 일정 구간동안 모든 태스크들이 마감시간을 만족하도록 하고 슬랙을 공유할 수 있도록 한다. 슬랙을 도메인 단위로 공유할 경우 얻을 수 있는 전력감소는 다음과 같다. 그림1에서 Task1, 2, 3은 각각 우선순위 순으로 나타난 태스크이고 주기와 최악의 실행시간을 가진다. 초월주기(0-24)는 D1, D2, D3, D4의 4개의 도메인으로 나누어 진다. 도메인 D1은 0-8까지의 시간을 가지는 구간이다. 이 구간에서 태스크는 도메인이라는 개념이 없을 경우에는 STA방법을 적용하여야 한다. STA방법을 통한 각 태스크의 소모 전력은 태스크의 실행시간 순(0-8까지 시간 내)으로 식(2)와 (3)을 통해  $1(\text{Task2}), 1(\text{Task 3}), 1/9(\text{Task1}), 1/4(\text{Task1})$ 의 전력을 보여줌에 있어서 전력은 3.36이 된다. 도메인 개념을 적용할 경우의 전력은 식(2)와 다음에 나올 식(4)을 통해  $(5/8)^3(\text{전압}) \times 8(\text{구간})$ 로 1.95이 된다.(단, 본 논문에서의 전압은 구체적인 언급이 없는 한 1을 기준으로 하고 언급된 전압은 1에 대한 상대적인 전압이다.) 즉, 도메인 개념을 사용하여 전압을 감소시킬 경우 도메인 개념이 없을 경우보다 전력감소에 더 효과적인 방안이 될 수 있다.

#### 3.2 도메인 결정

도메인의 결정에 있어서 도메인 구간의 설정은 중요한 문제가 된다. 도메인이 짧을 경우 테이블의 크기가 커지게 되고, 전력소모에 있어서 일반적으로는 좋은 성능을 보여 주지만 여러 태스크가 슬랙을 공유하지 못하게 되어 오히려 더 낮은 성능을 나타낼 수도 있다. 반대로 도메인이 길 경우는 계산 작업과 테이블 크기가 작아 질 수 있으나 전력소모에 있어서 그다지 높은 성능을 보여주지 못한다. 따라서 적절한 도메인의 구간을 나누기 위해서 본 논문에서는 가장 빈번히 발생하는 주기(예, 그림 1의 Task1)를 제외한 나머지 도메인의 주기들을 기점으로 도메인을 설정 하였다.

태스크  $i$ 는  $T_i$ 로 표현되고  $i$ 가 작을수록 우선순위가 높은

태스크를 가리킨다. 태스크  $i$ 의  $j$ 번째 주기부분 마감시간은  $T_{ij}$ 로 표현한다.

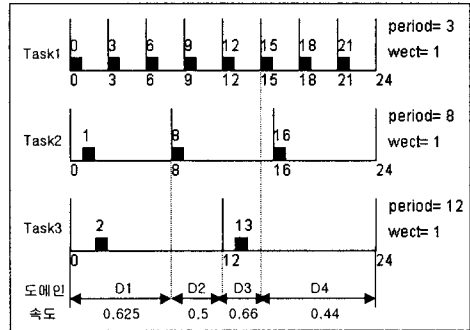


그림 1. 초월주기까지의 도메인의 예  
Fig. 1. Domains in the Hyperperiod of Tasks

도메인의 결정은 초월주기 동안 발생하는 최단 주기 태스크  $T_1$ (최 상위 우선순위)를 제외한 모든 주기들을 검색하여 발생된 주기들 사이로 도메인을 설정하고 발생된 주기 부근에서 실제적인 데드라인  $Dr$ (도메인 내의 모든 태스크들의 마감시간을 충족시키기 위한 도메인 값의 조정 변수)으로 도메인의 값을 조정한다. 실제적인 데드라인은 임의의 주기  $T_{ij}$ 에 대해서  $T_{i(j-1)} < Dr < T_{ij}$ 의 범위 내에서 결정이 되는데, 발생된 주기의 이전공간( $Dr$ 의 범위 내)들이 발생된 주기  $T_{ij}$ 의 우선순위 보다 높은 태스크일 경우 RM스케줄링 방식에 의해 우선순위가 높은 태스크들이 먼저 실행되어야 한다. 따라서 발생된 주기  $T_{ij}$ 의 이전공간중 시간상 맨 마지막 태스크 인스턴스(발생 주기와 가장 인접)를 기점으로, 역으로 인스턴스들을 검색하여 인스턴스가 슬랙이거나 현재 발생된 주기 태스크의 우선순위와 같은 경우까지 현재 태스크의  $Dr$ 은 이동하게 되고 이동된 태스크 인스턴스의 끝지점이  $Dr$ 이 된다. 만약 이전공간들이 현재 주기  $T_{ij}$ 의  $Dr$ 의 범위 내에서 현재  $i$  태스크를 만날 때까지 계속된다면 실제적인 데드라인은 현재 태스크  $i$ 의 끝 지점이 된다.

$Dr$  직전 구간이 슬랙일 경우, 이 슬랙 이전의 태스크 인스턴스의 STA에 의한 전압  $S^{STA}$  보다 식(4)을 이용한 도메인 전압  $S^{SSPM}$ 가 작을 경우 이 인스턴스는 마감시간을 만족하지 못한다.  $S^{SSPM}$ 가  $S^{STA}$  보다 클 경우 이 인스턴스를 포함한 도메인 내의 전체 태스크는 마감시간을 만족하지만 반대의 경우 이 인스턴스를 기준으로 도메인을 분리하여 각각의 전압과 도메인 시각을 테이블에 저장해서 태스크들이 마감시간을 모두 만족시키도록 한다.

#### 3.3 도메인 내의 전압(속도)

다음은 도메인이 결정이 된 후 도메인 내의 전압을 나타내는 식이다.

$$S^{SSPM} = S \cdot \sum_{i=1}^n \left( \frac{C_i \cdot (N_i)}{de - ds} \right) \quad (4)$$

$G_i$ 는 태스크  $i$ 의 한 주기 동안의 실행시간,  $d_e$ 는 도메인이 끝나는 시각,  $d_s$ 는 도메인의 시작 시각,  $N_i$ 는  $d_s$ 부터  $d_e$ 까지 태스크  $T_i$ 의 인스턴스의 개수,  $n$ 은 태스크의 총 개수이다. 그림 2는 그림1의 태스크 집합에 대해 STA와 제안하는 기법인 SSPM을 각각 적용하여 태스크를 실행시킬 때의 시간에 따른 프로세서의 전압 변동을 나타낸 그림이다. +는 문맥교환의 발생 및 문맥교환으로 뒤로 미루어진 태스크가 실행하는 것을 나타내며 태스크 위의 숫자는 프로세서의 전압(속도)을 나타낸다. STA는 슬래에 대한 공유가 없기 때문에 단순히 하나의 태스크 인스턴스가 그와 인접한 슬래를 사용하게 되므로 전압 변화가 심하다. 하지만 도메인별로 전압을 결정할 경우 상대적으로 STA를 적용한 경우 보다 전압 변동이 작게 된다. 그림2에서 STA의 전력 소모량은 7.19인데 반해 SSPM의 4.17로 개선되었다.(전압을 조절하지 않을 경우의 전력소모를 24로 설정)

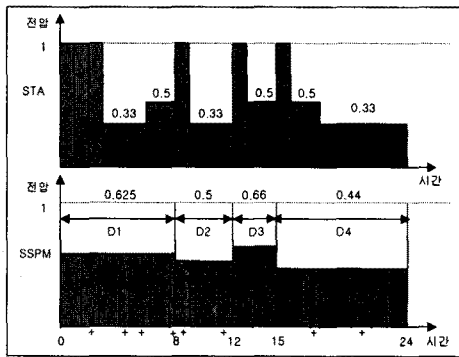


그림 2. STA와 SSPM의 시간에 따른 전압변화  
Fig 2. Voltage Scaling of STA and SSPM

오프라인 시에 도메인 값과 도메인의 전압이 결정된 후 각각은 테이블에 저장이 되고 태스크가 수행이 될 때 테이블의 저장된 값을 이용하여 프로세서의 전압을 조정한다.

#### 4. 성능 평가

성능은 제안된 알고리즘의 부하에 따른 전력소모율로 평가하였다. 모의실험의 비교 대상으로는 SONT, STA 및 EDF와 제안하는 기법인 SSPM로 하였다. 아래 그림 3는 모의실험의 결과를 나타내고 있다.

비교 대상 중 EDF(Earliest Deadline Frist)는 본 논문에서 적용하는 고정우선 순위 방식의 스케줄링 방식이 아닌 동적우선순위 방식이다. 정상 전압이 S라면 EDF에서 전압은

$$S \cdot \sum_{i=1}^n \frac{C_i}{T_i}$$

로 낮출 수 있다. 이것은 초월주기까지 슬래이 낮출 수 있는 최소의 전압이다. 때문에 전압관련에서 현재까지 최대의 전압감소를 얻을 수 있는 방법이다.[1]

부하의 변화로는 0.2부터 0.6사이의 값으로 하였고 태스크 개수는 2와 6사이로 하였다. 부하의 변화는 RM방식의 스케

줄에서 최악의 경우 부하의 상한선이 이론적으로 0.69까지 떨어질 수 있으므로 부하의 크기를 0.6까지로 제한하였다.[3] 제안된 기법인 SSPM은 SONT과 비교할 때 평균적으로 약 80%정도의 전력 감소 보여주었고, STA방식과 비교할 때 약 20%- 30%의 추가적인 전력감소를 보여주었다. 또한 SSPM은 가장 이상적인 전력감소율을 보여주는 EDF방식과 비교해서도 5%정도 성능이 떨어지기는 하지만 비슷한 정도의 결과를 보여 주고 있다.

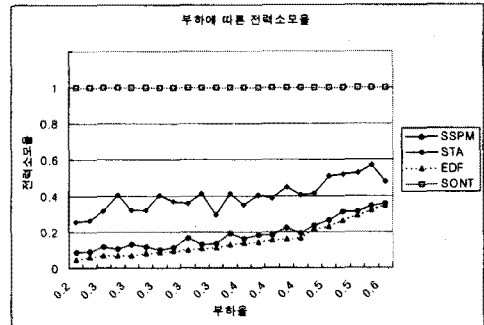


그림 3. 부하에 따른 소모전력 비교  
Fig 3. Comparison of Power Consumption

#### 5. 결론

본 논문에서는 대부분의 상용 운영체제에서 스케줄링 방식으로 많이 사용되는 RM기반의 전력감소 방안을 제시하고 있다. 오프라인으로 초월주기 동안 적절한 구간의 도메인들로 나누고 테이블에 저장한 후 테이블에 지정된 전압을 적용하여 프로세서의 전압을 조정하여 전체적인 시스템의 전력감소를 도출하였다. 또한 태스크를 실행할 때는 별도의 계산 작업 없이 테이블의 값을 이용함으로써 온라인상의 계산부하를 낮추어 시스템의 성능을 높일 수 있도록 하였다.

모의실험을 통하여 제안된 SSPM은 STA와 비교할 때 약 20-30%정도의 전력감소 효과를 볼 수 있었으며 최적의 전력감소를 나타내는 EDF와 비교해서도 약 5%정도 낮은 차이를 갖는 결과를 보여주었다.

#### 참고 문헌

- [1] Gang Quan and Xiaobo(Sharon) Hu, "Energy Efficient Fixed-Priority Scheduling for Real-Time Systems on Variable Voltage Processors," *Design Automation Conference*, vol. 38, pp. 828-833, 2001.
- [2] Y.H.Lee, C.M.Krishna. "Voltage-Clock Scaling for Low Energy Consumption in Real-Time Embedded Systems" *RTSS '2000*
- [3] F.Yao, A.Demers, and S.Shenker, "A scheduling model for reduced CPU energy" in *Proc.IEEE Annual Foundations of Computer Science*, pp. 374-382, 1995.
- [4] C.L.Liu and James Layland, "Scheduling Algorithms for Multiprogramming in a Hard -Real-Time Environment", *ACM*, 1973