

## 배터리와 태스크를 고려한 저전력 알고리즘 연구

윤충모\*, 김재진\*\*

### 요약

본 논문은 배터리와 태스크를 고려한 저전력 알고리즘을 제안하였다.

제안한 알고리즘은 배터리의 용량과 사용 목표 시간에 따른 단위 시간의 소모 전력을 설정한다. 주어진 모든 태스크들의 소모 전력을 계산한다. 태스크들 중에서 소모 전력이 가장 큰 태스크의 소모 전력과 소모 전력이 가장 작은 태스크의 소모 전력의 평균을 구한다. 태스크의 소모 전력의 평균을 단위 시간을 고려하여 다시 소모 전력을 계산한다. 태스크의 평균 소모 전력의 크기가 계산된 소모 전력의 평균보다 작거나 같을 경우 태스크의 평균 소모 전력보다 큰 태스크 들을 대상으로 저전력을 수행한다. 또한, 태스크의 평균 소모 전력의 크기가 계산된 소모 전력의 평균보다 클 경우 계산된 소모 전력의 평균보다 큰 태스크 들을 대상으로 저전력을 수행한다. 저전력은 태스크의 프로세서와 디바이스의 소모 전력을 분할하여 소모 전력이 큰 부분에 대해 저전력을 수행한다. 실험은 배터리를 고려한 저전력 알고리즘인 [6]과 비교하였다. 실험결과 [6]보다 소모 전력이 감소되어 알고리즘의 효율성이 입증되었다.

키워드 : 소모전력, 저전력, 태스크, 배터리

## A Study on the Low Power Algorithm consider the Battery and the Task

Choong-Mo Youn\*, Jae-Jin Kim\*\*

### Abstract

In this paper, we proposed the low power algorithm consider the battery and the task.

The proposed algorithm setting the power consumption of unit time consider the capacity of the battery and the target time. Calculate the power consumption of all tasks. Calculate the average power consumption by the task have maximum power consumption and the task have minimum power consumption. Recalculate average power consumption consider the unit time of task. Compare calculated average power consumption and average power consumption of task. Compared results, low power algorithm processing the average power consumption less than or equal calculated power consumption of task. Low-power algorithm is greater than the average power consumption of the task to perform targeted tasks. Low-power processors and the task by dividing the power consumption of the device in large part for the low-power consumption is performed. Experiments [6] were compared with the results of the power consumption. The experimental results [6] is reduced power consumption than the efficiency of the algorithm has been demonstrated.

Keywords : power consumption, Low power, Task, battery

### 1. 서론

※ 교신저자(Corresponding Author): Choong-Mo Youn  
접수일:2014년 05월 13일, 수정일:2014년 06월 11일  
완료일:2014년 06월 30일

\* 서일대학교 정보전자과 교수  
5477choong@hanmail.net

\*\* 강동대학교 신재생에너지과 교수

▣ 본 논문은 2013년도 서일대학교 학술연구비 지원에 의해 연구되었음.

배터리를 사용하는 이동기기의 급속한 보급으로 이동기기의 사용시간을 증가시키기 위한 연구가 지속적으로 진행되고 있다. 대부분의 연구는 저전력 소자를 개발하거나, 개발된 저전력 소자를 이용하여 회로를 구현하는 방법들이다. 클럭 게이팅(clock gating) 방법을 이용한 알고리

들이 제안되었다.[1][2][3] 배터리를 고려한 알고리즘으로 QoS를 조절하여 배터리의 사용시간을 증가시키는 방법도 제안되었다.[4] 또한 주파수를 조절하여 저전력을 구현하는 알고리즘도 제안되었다.[5][6] 배터리의 잔량을 고려한 BDLPA(Battery Driven Low Power Algorithm)은 태스크의 소모 전력이 배터리의 잔량보다 클 경우 태스크의 주파수를 조절하여 배터리의 잔량으로 태스크가 수행될 수 있도록 하였다.[6] 그러나 이러한 알고리즘들은 배터리의 전원이 허용하는 모든 태스크들을 수행한 후 태스크의 소모 전력이 배터리의 잔량보다 높을 경우에만 주파수를 조절하여 태스크를 수행하게 함으로써 수행할 수 있는 태스크의 수가 한정적이다. 또한 배터리를 사용하고자 하는 시간에 대한 정의가 없어 전체적으로 사용시간을 증가시키는데 어려운 단점을 가지고 있다.

본 논문에서는 이러한 단점을 보완하여 배터리와 사용 요구 시간을 고려하여 소모 전력을 조절해야 할 태스크를 결정한다. 결정된 태스크의 프로세서(process)와 디바이스(device) 중에서 소모 전력이 큰 것을 대상으로 소모 전력을 줄일 수 있도록 하는 방법을 제안한다.

## 2. 관련연구

배터리를 고려하여 저전력의 회로를 구현하기 위해서는 배터리의 용량을 고려하여 전체 회로의 소모 전력을 줄이는 방법을 연구한다.[5][6] 이러한 알고리즘의 배터리 용량과 연계하여 본 논문의 저전력 알고리즘을 구현하기 위해서 사용 목표 시간과 태스크의 최대 수행 시간을 이용하여 태스크의 소모 전력의 범위를 결정하여야 한다. 저전력 알고리즘을 적용해야 할 태스크의 범위를 결정하기 위한 경계는 식 (1)을 이용한다.

$$P_{T_{def}} = \frac{B_{cap}}{(T_t/T_{t_{max}})} = \frac{B_{cap} \cdot T_{t_{max}}}{T_t} \quad (1)$$

$P_{T_{def}}$  : 소모 전력 경계 값

$B_{cap}$  : 배터리 용량

$T_t$  : 사용 목표 시간

$T_{t_{max}}$  : 태스크의 최대 수행 시간

또한 알고리즘을 적용하기 위한 조건을 만족하기 위해서는 태스크의 평균 소모 전력을 구해야 한다. 태스크의 평균 소모 전력을 빠르게 계산하기 위해 식 (2)를 이용하여 구한다.

$$P_{average} = \frac{m \cdot T_{max} + n \cdot T_{min}}{m + n} \quad (2)$$

$P_{average}$  : 태스크의 평균 소모 전력

$m, n$  : 최대 소모 전력 태스크와 최소 소모 전력 태스크의 수

$T_{max}$  : 태스크의 최대 소모 전력

$T_{min}$  : 태스크의 최소 소모 전력

저전력 알고리즘을 수행하기 위한 대상을 선정하기 위해 태스크를 구별할 수 있는 경계를 필요로 한다. 경계는 단위 시간(unit time)을 이용하여 수행한다. 단위 시간은 태스크가 수행할 수 있는 최대의 시간을 의미하며 식 (3)과 같이 정의 된다.

$$T_{unit} = \frac{C_{battery}}{N_{task}} \quad (3)$$

$T_{unit}$  : 단위 시간

$C_{battery}$  : 배터리의 용량

$N_{task}$  : 수행하고자 하는 태스크의 수

태스크의 소모 전력은 프로세서와 디바이스로 나누어 정의 할 수 있다. 프로세서의 소모 전력은 주파수와 비례한다. 주파수는 식 (4)와 같이 정의 된다.

$$f \propto \frac{(V_{dd} - V_t)^\sigma}{V_{dd}} \quad (4)$$

$V_{dd}$  : 공급 전압

$V_t$  : threshold 전압

$\sigma$  : 속도 초화 계수

식 (4)에서와 같이 소모 전력을 줄이기 위한 방법으로 가장 효율적인 방법이 공급 전압을 조절하는 DVS(Dynamic Voltage Scaling) 알고리즘을 적용하여 저전력을 구현 한다. 또한 태스크의 소모 전력은 프로세서의 소모 전력과 입출력 장치를 포함한 디바이스의 소모 전력으로 나

누어 볼 수 있다. 디바이스는 입출력 장치와 여러 종류의 라이브러리(library)로 구성되어 있다. 디바이스의 소모 전력을 감소시키기 위한 방법은 일정 시간 이상 사용되지 않는 라이브러리들에 대해 sleep 상태로 변화시켜 전력 소모를 줄이는 DPM(Dynamic Power Management) 알고리즘을 적용한다.

### 3. 저전력 알고리즘

저전력 회로를 구현하기 위해서는 배터리의 용량과 요구 사용 시간을 고려하여 태스크의 소모 전력 한도를 결정한다. 계산된 소모 전력의 평균과 태스크의 평균 소모 전력의 크기를 비교한다. 비교 결과 태스크의 평균 소모 전력의 크기가 계산된 소모 전력의 평균보다 작거나 같을 경우 태스크의 평균 소모 전력보다 큰 태스크들을 대상으로 저전력을 수행한다. 또한, 태스크의 평균 소모 전력의 크기가 계산된 소모 전력의 평균보다 클 경우 계산된 소모 전력의 평균보다 큰 태스크들을 대상으로 저전력을 수행한다. 저전력은 태스크의 프로세서와 디바이스의 소모 전력을 분할하여 소모 전력이 큰 부분에 대해 저전력을 수행한다. 이러한 알고리즘은 (그림 1)에 나타내었다.

(그림 1) 알고리즘

```

Battery_Task_Low_Power( $B_{cap}, T_t, P_{library}$ )
{
    Calculate  $P_{T_{def}} = \frac{B_{cap} \cdot T_{t_{max}}}{T_t}$ ;
    Calculate  $P_{average} = \frac{m \cdot T_{max} + n \cdot T_{min}}{m + n}$ ;
    for(i=0; All Task; i++)
    {
        Calculate  $P_{Task}(i)$ ;
    }
    list_PTask(n) <= Sort from Maximum to
        Minimum Power consumption;
    if( $P_{average} \leq P_{T_{def}}$ )
    {
        for(n=0; n≠null; n++)
    
```

```

{
    if(list_PTask(n) ≥ P_average)
    {
        if( $P_{average\_process} > P_{average\_device}$ )
            Application DVS algorithm;
        elsif( $P_{average\_process} < P_{average\_device}$ )
            Application DPM algorithm;
        elsif( $P_{average\_process} = P_{average\_device}$ )
            Application DVS+DPM algorithm;
    }
}
else
    break;
}
if( $P_{average} > P_{T_{def}}$ )
{
    for(n=0; n≠null; n++)
    {
        if(list_PTask(n) ≥ P_T_def)
        {
            if( $P_{average\_process} > P_{average\_device}$ )
                Application DVS algorithm;
            elsif( $P_{average\_process} < P_{average\_device}$ )
                Application DPM algorithm;
            elsif( $P_{average\_process} = P_{average\_device}$ )
                Application DVS+DPM algorithm;
        }
    }
}
else
    break;
}
}

```

(Figure 1) Algorithm

(그림 2)는 (그림 1)의 알고리즘의 적용 예로 선정된 데이터패스이다.

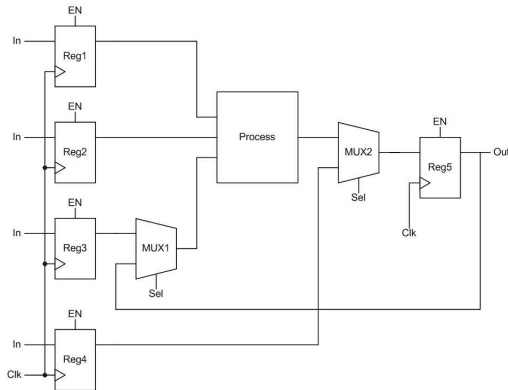
(그림 2)의 데이터 패스에서 최대의 소모 전력을 갖는 태스크의 소모 전력은 다음과 같다.

$$T_{max} = P_{Process} + 5 \cdot P_{Reg} + 2 \cdot P_{MUX}$$

최소의 소모 전력을 갖는 태스크의 소모 전력은 다음과 같다.

$$T_{\min} = 2 \cdot P_{Reg} + P_{MUX}$$

(그림 2) 데이터패스



(Figure 2) Datapath

배터리의 용량이 3.7V에 2.5Ah이고 사용 목표 시간을 3시간, 태스크의 최대 수행 시간이 0.02sec라고 가정했을 경우의 태스크의 소모 전력 경계는 다음과 같다.

$$P_{T\_def} = \frac{(3.7 \cdot 2.5) \cdot 0.02}{3} = 0.062W \cdot \text{sec}$$

전체 태스크들의 평균 소모 전력을 계산하여 태스크의 소모 전력 경계와 값의 크기를 비교한다. 두 값들 중에서 작은 값을 기준 값으로 선정하여 기준 값보다 소모 전력이 큰 태스크에 대해 저전력을 적용한다.

### 4. 실험결과

실험에서 사용하는 프로세서는 AMD's mobile K6을 사용하였다.[8] 그 외의 장치로는 하드디스크와 DSP, Flash로 구성하였다.[9][10][11] 알고리즘의 실용성을 검증하기 위해 [7]과 동일한 대상으로 실험을 수행하였다. 기존의 알고리즘과의 결과를 비교하여 본 논문의 효율성과 우수성을 비교하기 위해 가장 유사한 알고리즘인 [7]을 대상으로 실험한다.

실험 대상의 태스크에 대한 특성은 [7]과 동일한 것으로 <표 1>에 나타내었다.

<표 1> 태스크의 특성

태스크	T1	T2	T3	T4	T5
사용	○	-	-	-	○
디바이스	-	-	○	○	○
	-	-	-	○	○

<Table 1> Task character

태스크에 따른 소모 전력 결과는 <표 2>에 나타내었다.[6]

<표 2> 태스크에 따른 소모 전력

태스크	소모전력[W]	
	최소	최대
T1	5.7	8.75
T2	2.02	9.1
T3	10.1	14.8
T4	8.9	12.4
T5	15.2	20.45
합계	41.92	65.5

<Table 2> Power consumption for task

[6]의 평균 소모 전력과 본 논문의 알고리즘을 적용한 결과를 <표 3>에 나타내었다.

<표 3> 실험 결과

태스크	[6]	제안한 알고리즘
T1	7.23	7.16
T2	5.56	5.55
T3	12.45	12.41
T4	10.65	9.71
T5	17.83	13.96
합계	53.72	48.79

<Table 3> Experimental Result

### 5. 결론

본 논문은 배터리와 태스크를 고려한 저전력 알고리즘을 제안하였다.

제안한 알고리즘은 이동기기의 사용시간을 증

가시킴을 위해 저전력 알고리즘을 적용하는 방법을 제안하였다. 알고리즘은 배터리의 용량과 사용 목표 시간에 따른 단위 시간의 소모 전력을 설정한 후 태스크의 소모 전력의 평균을 단위 시간을 고려하여 다시 소모 전력을 계산한다. 계산된 소모 전력의 평균과 태스크의 평균 소모 전력의 크기를 비교하여 태스크의 평균 소모 전력의 크기가 계산된 소모 전력의 평균보다 작거나 같을 경우 태스크의 평균 소모 전력보다 큰 태스크 들을 대상으로 저전력을 수행한다. 또한, 태스크의 평균 소모 전력의 크기가 계산된 소모 전력의 평균보다 클 경우 계산된 소모 전력의 평균보다 큰 태스크 들을 대상으로 저전력을 수행한다. 저전력은 태스크의 프로세서와 디바이스의 소모 전력을 분할하여 소모 전력이 큰 부분에 대해 저전력을 수행한다.

실험결과 [6]의 알고리즘보다 소모 전력이 약 9.2% 감소되어 알고리즘의 효율성이 입증되었다.

향후 본 논문에서 제시한 알고리즘의 수식과 방식을 사용 되는 태스크들의 소모전력 형태를 유형별로 분리하여 look-up table 방식으로 분류함으로써 가장 빠른 방식의 저전력 알고리즘 구현에 대한 연구에 적용할 수 있는 기반이 될 수 있어 지금보다 좀 더 효과적인 저전력 알고리즘 연구를 수행할 수 있을 것으로 사료된다.

References

[1] Qing Wu, Massoud Pedram, Xunwei Wu, "Clock-Gating and Its Application to Low Power Design of Sequential Circuits," IEEE Custom Integrated Circuits Conference, 1997.

[2] D. Garrett, M. Stan, and A. Dean, "Challenges in clock gating for a low-power ASIC methodology," in Proc. ISLPED, San Diego, CA, pp. 176-181, August, 1999

[3] Pietro Babighian, Enrico Macii, "A Scalable Algorithm for RTL Insertion of Gated Clocks Based on ODCs Computation," IEEE transactions on Computer-Aided Design of Integrated Circuits And Systems, vol. 24, no. 1, pp.29-42, January 2005

[4] L. Benini, G. Castelli, A. Macii, R. Scarsi, "Battery-Driven Dynamic Power Management of Portable System", ISSS(International Symposium on System Synthesis), pp.25-33, 2000

[5] 김재진, "시간제약 조건하에서 태스크에 따른 저전력 알고리즘에 관한 연구", 한국정보기술학회, vol. 7, no. 6, pp. 27-34, 2009

[6] 김재진, "이동기기에서 배터리를 고려한 저전력 알고리즘 연구", 한국컴퓨터정보학회, 제16권, 제2호, pp. 193-199, 2011

[7] 윤연선, "전력소모 모델링을 통한 DVS와 DPM 동시적용 알고리즘에 관한 연구", 서강대학교 대학원, 2007

[8] "Mobile AMD'6 Processor Power Supply Application Note", [www.amd.com](http://www.amd.com)

[9] "MHL2300AT Hard Disk Drive Product Manual", [www.fujitsu.com](http://www.fujitsu.com)

[10] "TMS320C6411 Power Consumption Summary", [www.ti.com](http://www.ti.com)

[11] "SST Multi-purpose Flash SST39LF020", [www.sst.com](http://www.sst.com)

윤충모



1990년 : 단국대학교 대학원 (공학 석사)  
 2000년 : 청주대학교 대학원 (공학 박사)

1993년~현재 : 서일대학교 정보전자과 교수  
 관심분야 : CAD 알고리즘, 정보보안, 저전력 등



## 김재진

1995년 : 청주대학교 대학원 (공학 석사)

2003년 : 청주대학교 대학원 (공학 박사)

2001년~현재 : 강동대학교 신재생에너지과 교수  
관심분야 : CAD 알고리즘, 정보보안, 저전력 등