

## 태스크에 따른 저전력 알고리즘에 관한 연구

김재진\*

### 요약

본 논문에서는 태스크에 따른 저전력 알고리즘을 제안하였다.

태스크는 시스템의 작업 수행에 필요한 프로세서의 내부와 외부의 자원을 의미한다. 태스크에 따라 저전력 회로를 구현하기 위해서는 각각의 태스크에 대한 생존시간과 호출횟수를 분석한다. 회로 전체의 소모 전력을 감소하기 위해서는 소모 전력이 가장 높은 태스크의 소모 전력을 우선 줄여 저전력 회로를 구현할 수 있다. 따라서 소모 전력이 최대인 태스크를 우선 선별하여야 한다. 소모 전력이 최대인 태스크는 태스크의 생존시간과 호출횟수를 고려하여 순위를 선정한다. 태스크의 생존시간이 길면서 호출횟수가 많은 태스크의 경우 가장 큰 소모 전력을 발생시키는 태스크이므로 소모 전력을 감소시킬 최우선 순위가 된다. 소모 전력이 최대인 태스크로부터 생존 시간과 호출횟수를 이용하여 저전력 회로로 구현하기 위한 주파수를 결정하여 회로 전체의 소모 전력을 감소시킨다. 또한, 생존 시작 시간에서 생존 마지막 시간까지 계속해서 최소의 소모 전력으로 태스크를 유지시켜 전체 소모 전력을 감소시킨다.

실험 결과 [7] 알고리즘에 비해 5.43%의 전력 소모가 감소된 결과를 나타내었다.

키워드 : 소모전력, 저전력, 태스크, 생존시간, 호출횟수

## A Study on the Low Power Algorithm for a Task

Jae-Jin Kim\*

### Abstract

In this paper, we proposed low power algorithm for a task.

The task means the inside of a necessary processor and external resources to work accomplishment of a system. Each task analyzes a life time and a number of called for implement a low power circuit. First of all, reduce power consumption of a task have maximum power consumption for low power circuit implementation. Therefore, first selecting a task had maximum power consumption. The task had a maximum power consumption ranking consider a life time and a number of called for each task. While a life time of task is long, top priority ranking to decrease power consumption to the task that the number of call generates the power consumption how a disguise is large in case of a lot of task becomes. Frequency decision to have minimum power consumption, and decrease power consumption all the circuit by a change of frequency of the task which the minimum task that a wasting past record is the maximum becomes. Also, keep continuously minimum power consumption, with every effort task until last life time in opening life time, and decrease gets total power consumption. Experiments results show reduction in the power consumption by 5.43% comparing with that [7] algorithm.

Keywords : power consumption, Low power, Task, The Life time, The number of call

### 1. 서론

※ 교신저자(Corresponding Author): Kim, Jae Jin  
접수일:2013년 02월 26일, 수정일:2013년 03월 13일  
완료일:2013년 03월 25일

\* 강동대학교 신재생에너지과

Tel: +82-43-879-3374, Fax: +82-43-879-3374

email: dkimjj@gangdong.ac.kr

■ 2012학년도 강동대학교 교내학술연구비 지원으로

이동기기의 발전은 수요와 적용분야가 증가되면서 지속적으로 발전되어 가고 있다. 이러한 발

계재함

전은 고사양의 제품에 대한 기대를 높여가고 있다. 고성능의 제품을 제공하기 위해서는 고사양의 프로세서(processor)와 주변기기를 필요로 한다. 그러나 고사양의 제품은 높은 소모 전력이 요구되어 제한적인 배터리의 용량으로 사용시간을 증가시키기 위해서는 개발된 저전력 프로세서를 이용하여 이동 기기를 개발하여야 한다. 프로세서의 소모 전력을 줄이기 위한 방법으로 클럭 게이팅(clock gating)을 이용한 방법과 스케줄링(scheduling) 방법, 실시간 시스템 기반의 외/내부 슬랙(external/internal slack)을 활용한 DVS(Dynamic Voltage Scaling) 방법 등이 제안되었다.[1][2][3][4][5][6] 이러한 방법들은 프로세서의 주파수를 고려하여 최소 소모 전력을 갖는 주파수를 결정하는 방법들이 대부분이었으며 주어진 시간제한 조건에 맞도록 프로세서의 동작 시간을 제한한 후 제한된 시간 제약 조건 중에서 가장 소모 전력이 작은 주파수를 찾아 프로세서의 소모 전력을 줄이는 방법이었다.[7][8][9]

그러나 이러한 방법들은 주어진 시간제한이나 저전력 프로세서를 개발하는데 적용되는 방법으로 범용적인 저전력 회로를 설계하는 방법으로 적절하나, 개발된 저전력 프로세서를 이용하여 개발된 회로에 대한 저전력을 구현할 수 없다는 단점이 있다. 또한 주파수를 감소시켜 저전력의 회로를 구현할 수 있으나 주파수의 변화 값을 정확하게 제한 할 수 없어 주어진 태스크에 따라 최소 주파수 값을 결정할 수 있는 방법을 제안하여 저전력 회로를 구현할 수 있는 알고리즘을 제안하였다.

## 2. 관련연구

개발된 저전력 프로세서를 이용하여 개발된 회로에 대한 저전력을 구현하기 위해서는 태스크에 대한 분석을 필요로 한다. 태스크는 시스템의 작업 수행에 필요한 프로세서의 내부와 외부의 자원을 의미한다. 태스크에 대한 분석은 다음과 같다.

$i$  번째 태스크  $T_i$ 에 대한 특성은 다음과 식 (1)과 같이 정의할 수 있다.

$$T_i = \{MT_i, R_i, LT_i\} \quad (1)$$

$MT_i$  : 태스크 수행에 필요한 최소 시간

$R_i$  : 태스크의 호출횟수

$LT_i$  : 태스크의 생존 시간

$LT_i = \{LT_{S_i}, LT_{L_i}\}$

$LT_{S_i}$  : 태스크의 생존 시작 시간

$LT_{L_i}$  : 태스크의 생존 마지막 시간

태스크의 생존 시간은 동일한 주파수를 기준으로 분석하여 정의된 것이다. 따라서 전력 소모가 가장 큰 태스크는 생존시간이 가장 긴 태스크와 호출횟수가 가장 많은 태스크이다. 식 (1)에서 정의된 태스크에서 전력 소모가 가장 큰 태스크를 찾기 위해 다음 식 (2)와 식 (3)을 이용한다.

$$T_t = \{T_{t1}, T_{t2}, \dots, T_{ti}\} \quad (2)$$

$T_{t1}$  :  $LT_i$ 의 값이  $t$ 인 최소  $LT_i$  값을 가진 태스크

$T_{t2}$  :  $LT_i$ 의 값이  $t$ 인  $T_{t1}$  다음으로 작은 태스크

$T_{ti}$  :  $LT_i$ 의 값이  $t$ 인 최대  $LT_i$  값을 가진 태스크

$$T_r = \{T_{r1}, T_{r2}, \dots, T_{rj}\} \quad (3)$$

$T_{r1}$  : 호출 횟수가  $r$ 인 최소 호출횟수 값을 가진 태스크

$T_{r2}$  : 호출 횟수가  $T_{r1}$  다음으로  $r$ 인 호출횟수 값을 가진 태스크

$T_{rj}$  : 호출 횟수가  $r$ 인 최대 호출횟수 값을 가진 태스크

식 (2)은 생존시간에 따른 태스크들의 집합이며, 식 (3)은 호출 횟수에 따른 태스크들의 집합이다. 이들 집합들을 이용하여 식 (4)로 소모 전력에 따른 태스크들의 집합을 만든다.

$$T_{lij} = T_t \times T_r \quad (4)$$

$$= T_{(t \times r)ij}$$

$$= \{T_{l_{1ij}}, T_{l_{2ij}}, \dots, T_{l_{mij}}\}$$

$T_{lij}$  : 태스크의 생존시간과 호출 횟수를

고려하여 연산된 태스크들의 집합  $T_{lij}$  : 집합  $T_{ij}$ 의 태스크 값들 중 최소 값을 가진 태스크  
 $T_{mij}$  : 집합  $T_{ij}$ 의 태스크 값들 중 최대 값을 가진 태스크

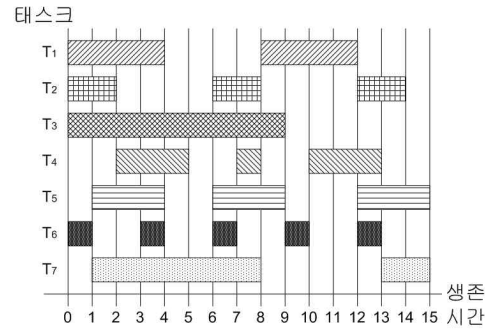
식 (4)를 이용하여 생성된 태스크 집합의 값들 중에서 최대의 값인  $T_{mij}$ 의 소모 전력이 가장 큰 태스크로서 소모 전력을 줄이는 최우선의 대상이 된다. 이때 태스크는  $j$ 번째 태스크와  $j$ 번째 태스크를 동일 대상으로 간주한다.

### 3. 저전력 알고리즘

저전력 회로를 구현하기 위해서는 주어진 태스크의 소모 전력을 분석하여 최대의 소모 전력을 가지고 있는 태스크를 대상으로 소모 전력을 감소시킬 수 있도록 알고리즘을 적용한다. 제안한 저전력 알고리즘은 주어진 태스크들 중에서 최대의 소모 전력을 갖고 있는 태스크에 대해 주파수를 변화시켜 소모 전력을 줄인다. 태스크에 따른 소모 전력의 계산 방법은 (그림 1)에 나타내었다.

[7] 알고리즘의 소모 전력 계산은 생존 시간당 태스크의 소모 전력과 생존시간의 합에 호출횟수를 곱하여 전체 소모 전력을 계산한다. 이러한 방법은 소모 전력이 가장 큰 태스크의 기동 시간이 호출 횟수와 동일하게 되어 소모 전력이 증가되는 결과를 나타낸다. (그림 1)에서 각각의 태스크에 대한 소모 전력은 태스크가 시작되는 시간에서 끝나는 시간까지의 생존시간과 호출 횟수를 고려하여 식 (4)를 이용하여 계산된다.

(그림 1) 태스크에 따른 소모전력 계산  
 (a) 태스크 특성



(a) task characteristic

(b) 태스크의 소모 전력 지수

태스크	최대 생존시간	호출횟수	소모 전력지수
T <sub>1</sub>	12	2	24
T <sub>2</sub>	14	3	42
T <sub>3</sub>	9	1	9
T <sub>4</sub>	11	3	33
T <sub>5</sub>	14	3	42
T <sub>6</sub>	13	5	65
T <sub>7</sub>	14	2	28

(b) The index number of power consumption for task

(c) 태스크의 소모 전력(생존 시간당 태스크의 소모 전력=100mW 일때)

태스크	[7]	
	소모 전력 계산	소모 전력(mW)
T <sub>1</sub>	$[100+(21.47*3)]*2$	328.82
T <sub>2</sub>	$(100+21.47)*3$	364.41
T <sub>3</sub>	$[100+(21.47*8)]*1$	271.76
T <sub>4</sub>	$[100+(21.47*2)]*2+100$	385.88
T <sub>5</sub>	$[100+(21.47*2)]*3$	428.82
T <sub>6</sub>	$100*5$	500
T <sub>7</sub>	$[100+(21.47*6)]+(100+21.47)$	350.29
합계		2,629.98

태스크	본 논문	
	소모 전력계산	소모 전력(mW)
T <sub>1</sub>	100+(21.47*11)	336.17
T <sub>2</sub>	100+(21.47*13)	379.11
T <sub>3</sub>	100+(21.47*8)	271.76
T <sub>4</sub>	100+(21.47*10)	314.7
T <sub>5</sub>	100+(21.47*13)	379.11
T <sub>6</sub>	100+(21.47*12)	357.64
T <sub>7</sub>	100+(21.47*13)	379.11
합계		2,417.6

(c) The power consumption of task(case power consumption of task have one life time = 100mW)

(Figure 1) Power consumption calculation for task

계산된 소모 전력들 중에서 가장 큰 값을 가지고 있는 태스크인 T<sub>6</sub>를 우선 대상으로 소모 전력을 줄일 수 있도록 알고리즘을 적용한다. 적용한 알고리즘은 (그림 2)에 나타내었다.

(그림 2)에서 저전력 회로를 구성하기 위해 소모 전력이 최소가 되는 주파수를 찾아 주파수 변화에 따른 소모 전력을 구해야 한다. CMOS의 소모 전력은 일반적으로 식 (5)를 이용하여 구한다.[8]

$$E \propto C_L * N_{cycle} * V_{DD}^2 \quad (5)$$

$C_L$  : CMOS 회로의 부하 캐패시턴스

$N_{cycle}$  : 프로그램이 실행된 사이클 수

$V_{DD}$  : 공급 전압

(그림 2) 저전력 알고리즘

```

Low_Power(List_Task(i))
{
    for(i=0;i;i++)
    {
        make LT_Task(i);
        make R_Task(i);
        calculate P_Task(i);
        // LT_Task(i):태스크(i)의 생존시간
        // R_Task(i):태스크(i)의 호출횟수
        // P_Task(i):태스크(i)의 소모전력
    }
    for(i=0;i;i++)
    {
        Find
        maximum(P_Task(k))<=P_Task(i);
    }
    LP_f = ( LT_Task(k) / R_Task(k) ) - 1;
    if(LP_f <= 0)
        LP_f=1;
    calculate PC_Task(i);
    /* PC_Task(i); i번째 태스크의
        소모전력 */
    els
    calculate PC_Task(i, LP_f);
}
    
```

(Figure 2) Low power algorithm

CMOS의 소모 전력은 정적전력과 동적전력, 회로단락이 주된 원인이며, 특히 동적 전력은 약 90%의 원인이 된다. 동적 전력은 식 (6)을 이용하여 구할 수 있다.[8]

$$P_{dynamic} = C_L \cdot N_{SW} \cdot f \cdot V_{DD}^2 \quad (6)$$

$C_L$  : CMOS 출력노드 부하 캐패시턴스

$N_{SW}$  : 클럭 주기당 평균 스위칭 횟수

식 (6)에서  $C_L$ 과  $V_{DD}$ 가 일정하다고 가정하면  $N_{SW}$ 와  $f$ 의 값에 의해 소모 전력이 결정된다.

주파수는 시간에 반비례하기 때문에 주기가 커지면 주파수가 감소하고 소모 전력은 감소하게 된다. 그러나 주파수 값을 감소시키는 데에는 한계가 필요하다. 주파수의 증가 범위는 소모 전력이 가장 큰 태스크의 평균 생존 시간에서 기본값인 1을 뺀 값의 범위로 한정하여 저전력을 구현한다. 주파수 감소 범위는 식 (7)을 이용하여 계산한다.

$$LP\_f = \left( \frac{LT\_Task(k)}{R\_Task(k)} \right) - 1 \quad (7)$$

$LP\_f$ : 저전력 주파수

$LT\_Task(k)$ : 태스크 k의 생존 시간

$R\_Task(k)$ : 태스크 k의 호출 횟수

(그림 1)의 태스크들 중에서  $T_6$ 의 태스크가 소모 전력이 가장 크며 식 (7)의 식을 이용하여 주파수 범위를 구하면 다음과 같다.

$$\begin{aligned} LP\_f &= \frac{13}{5} - 1 \\ &= 2.6 - 1 \\ &= 1.6 \end{aligned}$$

따라서 주파수를 1.6배까지 감소시켜 저전력의 회로를 구현한다.

#### 4. 실험결과

본 논문에서 제안한 알고리즘의 효율성을 입증하기 위해 [7]의 주파수 선택 알고리즘과 비교하였다. 실험 대상은 [7]과 같은 환경을 만들기 위해 Altera의 Quartus II를 이용하였으며 Stratix-EP1S10F484C5로 회로를 구현하여 실험하였다. 실험 결과는 <표 1>에 나타내었다.

실험결과 주파수를 30.6MHz일 경우 [7] 알고리즘을 적용했을 때보다 본 논문에서 제안한 알고리즘을 적용했을 때의 소모 전력이 평균 5.11% 감소된 결과를 나타내었으며, 주파수가 19.1MHz일 경우 평균 5.74% 감소된 결과를 나타내었다. 두 가지 주파수에서의 소모 전력 감소율 평균은 5.43% 감소된 결과를 나타내었다.

<표 1> 소모 전력 비교

주파수 (MHz)	태스크	소모 전력 mW	
		[7]	본논문
30.6	T <sub>1</sub>	427.48	422.66
	T <sub>2</sub>	403.74	379.86
	T <sub>3</sub>	311.64	311.64
	T <sub>4</sub>	443.32	416.72
	T <sub>5</sub>	522.48	491.13
	T <sub>6</sub>	475	424.37
	T <sub>7</sub>	467.06	448.38
	합계	3,050.72	2,894.76
19.1	T <sub>1</sub>	266.83	256.16
	T <sub>2</sub>	252	236.88
	T <sub>3</sub>	194.52	194.52
	T <sub>4</sub>	276.71	260.1
	T <sub>5</sub>	326.12	306.55
	T <sub>6</sub>	296.49	260.91
	T <sub>7</sub>	291.53	279.87
	합계	1,904.2	1,794.99

<Table 1> Compare for power consumption

#### 5. 결론

본 논문에서는 태스크에 따른 저전력 알고리즘을 제안하였다.

태스크에 따라 저전력 회로를 구현하기 위해서는 각각의 태스크에 대한 생존시간과 호출횟수를 분석한다. 회로 전체의 소모 전력을 감소하기 위해서는 소모 전력이 가장 높은 태스크의 소모 전력을 우선 줄여 저전력 회로를 구현할 수 있다. 따라서 소모 전력이 최대인 태스크를 우선 선별하여야 한다. 소모 전력이 최대인 태스크는 태스크의 생존시간과 호출횟수를 고려하여 순위를 선정한다. 태스크의 생존시간이 길면서 호출횟수가 많은 태스크의 경우 가장 큰 소모 전력을 발생시키는 태스크이므로 소모 전력을 감소

시킬 최우선 순위가 된다. 소모 전력이 최대인 태스크로부터 생존 시간과 호출횟수를 이용하여 저전력 회로로 구현하기 위한 주파수를 결정하여 회로 전체의 소모 전력을 감소시킨다. 또한, 생존 시작 시간에서 생존 마지막 시간까지 계속해서 최소의 소모 전력으로 태스크를 유지시켜 전체 소모 전력을 감소시킨다.

실험 결과 [7] 알고리즘에 비해 5.43%의 적력 소모가 감소된 결과를 나타내었다.

추후로 [7] 알고리즘과 본 논문의 알고리즘을 이용하여 호출횟수에 따라 저전력을 구현할 수 있는 알고리즘을 개발하여 더욱 더 효율적인 저전력 알고리즘 개발이 필요하다.

### References

[1] A. Chandrakasan, R. Brodersen, "Low power digital CMOS design," Kluwer Academic Publishers, 1995.

[2] Qing Wu, Massoud Pedram, Xunwei Wu, "Clock-Gating and Its Application to Low Power Design of Sequential Circuits," IEEE Custom Interated Circuits Conference, pp.479-482, 1997.

[3] 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

[4] T.Mudge, "power: a first-class architectural design constraint," IEEE COMPUT., vol. 34, no. 4, p.52-58, April, 2001

[5] 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, Jaunary 2005

[6] Padamnabhan Pillai, Kang G. Shin, "Real time Dynamic voltage scaling for low power embedded operating system", In Proceeding of the 18th ACM symposium on Operating System Principles(SOSP-01), pp. 89-102, 2001

[7] Kim. Jae Jin, Kang. Jin Gu, Hur. Hwa Ra, Yun. Choon Mo," A Frequency Selection Algorithm for Power Consumption Minimization of Processor in Mobile System", The Journal of Korea Society of Digital Industry and Information Management, vol. 4, No. 1, pp. 9-16, 2008

[8] Kim. Jae Jin, "A Study of Low Power Algorithm for a Task under the Time Constraint", The Journal of Korea Institute of Information Technology, vol. 7, No. 6, pp. 27-34, 2009

[9] Youn. Choong Mo, Kim. Jae Jin, "A study of FPGA Algorithm for consider the Power Consumption", Journal of Digital Contents Society, vol. 13, No. 1, pp. 37-41, 2012



### 김재진

1995년 : 청주대학교 대학원 (공학 석사)

2003년 : 청주대학교 대학원 (공학 박사)

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