

# 임베디드 시스템의 소모 전력 관리 기법

김화영, 김영길  
아주대학교 전자공학과

e-mail : wha\_young@lge.com

(The power management technique in the Embedded System)

Wha\_young Kim, YoungKil Kim  
School of Electronics Engineering Ajou University

## 요 약

배터리를 전원으로 사용하는 이동 통신 기기에서 전력 관리의 효율성은 전통적으로 중요한 요구조건 중의 하나이다. 특히 고성능과 고기능을 제공하면서도 더 긴 동작시간이 요구되는 최근의 이동 통신 기기들에서는 이러한 효율적인 전력 관리의 중요성이 더욱 크다. 본 논문은 전력 소모량과 시스템의 크기에서의 장점으로 인해 이동 통신 기기에 폭 넓게 적용되고 있는 embedded system 에서의 효율적인 전력 관리를 위한 CPU 소모 전력 관리 기법으로 적응적 동적 전력 관리 기법을 제안한다.

## Abstract

The efficiently power management is an important requirement traditionally in the mobile communication system which uses battery as their power source. Especially, it has been emphasized in the most recent devices, which has to provide high performance and various functions with an extended operating time. In this article, the adaptive power management technique for the core processor unit in embedded systems used widely for the mobile system thanks to its advantage on power consumption and physical size, is proposed.

Keywords : 이동 통신 기기, Battery life, 임베디드 시스템, CPU 전력 소모량 개선, 전력 관리 기법

## I. 서 론

이동 통신 기기의 동작 시간의 개선을 위한 연구들은 전력 공급원인 배터리에서부터 CPU 그리고 운영 체제에 이르기 까지 이동 통신 기기의 전반에 걸쳐서 진행되어 오고 있다. 그리고 이러한 연구들은 battery 용량의 확장 과 battery life 의 개선, CPU 동작 전압의 개선 및 소모 전력 개선, 그리고 Display 장치와 Memory 와 같은 주변 장치들의 소모 전력 개선 과 운영체제 동작상에서의 시나리오 개선을 통한 소비 전력 감소 등에서의 여러 개선으로 나타나고 있다.

하지만 그럼에도 불구하고 최근의 이동 통신 환경의 발달과 Ubiquitous 환경으로의 진입 그리고 추가되는 다양한 기능들로 인해 이동 통신 기기의 동작 시간의 확장에 대한 필요성은 더욱 커지고 있다.

이에, 본 논문은 CPU 소모 전력 개선에 관한 내용을 다루고자 한다. 휴대용 컴퓨터 시스템에서의 전력 소비량을 분석한 지난 연구[2]를 살펴보면, Display 의 전력 소모가 가장 크고, 이어서 Disk Drive, CPU, 그리고 Memory 의 순으로 전력 소비가 나타나고 있음을 볼 수 있다(Table 1).

비록 이 분석 결과는 Display 소자가 소비전력 개선에 매우 중요함을 나타내고 있으나 Ubiquitous 환경

Component	Manufacturer & Model	Power (watts)	Percent of Total
Display	Compaq monochrome life23c	3.5	68%
Disk Drive (105 Mbytes)	Maxtor MXL105 III	1.0	20%
CPU	3.3V Intel i486	0.6	12%
Memory (16 Mbytes)	Microz MT4C(MHA1/B1)	0.024	0.5%

Table 1: Breakdown of power consumption by components.

에서의 경향과 운영체제의 Power Management 기법의 개선 등으로 인해 최근에는 CPU 소모 전력 개선의 중요성이 더욱 커지고 있다[3].

이러한 CPU 소모 전력 개선에 있어서는 효율적인 전력 관리가 매우 중요한데 이는 다양한 기능과 높은 성능 제공에 대한 요구들을 만족시키면서도 사용자가 성능의 저하를 느끼지 못하는 범위 내에서 이루어져야 하는 기본적인 제약 조건이 존재하므로 해당 동작 순간에 제공하여야 하는 일정 성능을 가장 낮은 전력 소모량을 가지고 제공하여야 하기 때문이다.

이러한 효율적인 전력 관리에 대한 기존의 연구들을 2장에서 살펴보고, 3장에서는 CPU 소모 전력 개선을 위해 본 논문에서 제안하는 적응적 동적 전력 관리 기법 기술하며, 4장에서는 제안 기법을 구현한 시스템을 설명하도록 한다. 5장에서는 실험 및 결과 검토를 통한 결론을 기술하였다.

## II. 기존 연구들

CPU의 효율적인 전력관리에 대한 기존의 연구들은 기기의 전력 소모를 줄이는 가장 확실한 방법인 동작 전압을 낮추는 것들에 관한 것들인데 Processor 에서의 전력소모는 아래와 같이 전압의 제곱에 비례 한다.

$$E \propto CV^2$$

(C = switched capacitance, V = operating voltage, E = energy).

따라서 CPU 의 효율적인 전력 관리를 위해 기존부터 다양한 동작 전압과 속도에서 동작하는 CPU 의 설계와 이의 이용에 관한 연구들이 있어왔다[1],[3],[4].

Trevor[4]는 Hardware 적인 설계측면에서 CPU 소비 전력 감소를 위한 방법을 살펴보았는데 다음과 같이 분석된 CPU 내부 각 모듈들의 전력 소모 비율을 바탕으로 CPU 내부 Cache 에 DVS(Dynamic Voltage Scaling)의 적용이 가능함을 보여주었다.

1. Core : 58%
2. Cache: 33%
3. Processor Bus: 7%
4. SRAM: 2%

그리고 실제로 Embedded CPU[Buverde] 들은 Dynamic Voltage Scaling 과 Dynamic Frequency Scaling 을 지원하고 있다.

Software 측면에서 진행된 연구들은 모두 기본적으로 CPU 의 동작 전압을 가변시키기 위한 방법들로서 CPU의 idle cycles 과 같은 지난 동작 정보들을 추출하여 향후 동작을 예측하여 DVFM(Dynamic Voltage and Frequency Management)을 실시하는 기준을 삼는 것들이다. 이에 관한 연구들로 Weiser et al[3] 과 Edwin[4] 의 연구 내용들을 간략하게 아래와 기술하였다.

### 2.1. Weiser et al 의 연구

먼저, Weiser et al 은 자신의 연구를 통하여 아래에 기술한 3가지의 CPU DVFM 적용 정책을 위한 알고리즘 중에서 PAST를 사용가능한 알고리즘임을 기술하였다.

- \* OPT (unbounded-delay perfect-future)
- \* FUTURE (bounded-delay limited-future)
- \* PAST (bounded-delay limited-past)

각 알고리즘의 특징을 간략히 살펴보면, 다음과 같다.

#### 2.1.1 OPT (unbounded-delay perfect-future)

OPT 알고리즘은 Weiser et al 자신이 기술한 것처럼 CPU 소비 전력 감소를 위한 방법으로서는 CPU 요구 성능 예측의 가능성 여부와 smoothing 및 소비 전력 측면에서도 바람직하지도 않고 실현할 수 없는 방법이다.

이 알고리즘은 CPU 에 요구되는 향후의 모든 작업

들의 부하량을 파악하여 운영체제의 동작에 IDLE 시간이 존재하지 않도록 하는 것이다. 이는 운영체제가 동작함에 있어서 주기적으로 CPU 를 idle 상태로 진입시키고 Timer interrupt를 이용하여 CPU를 깨워서 요구되는 작업을 수행하는 동작에 있어서 CPU 의 동작 전압과 동작 주파수를 요구되는 전체 작업량을 분석하여 최적화하고 재분배 함으로써 요구되는 작업을 주어진 scheduling 시간내에 다 끝내고 CPU idle 진입을 하는 구간이 없도록 하는 것이다.

#### 2.1.2 FUTURE (bounded-delay limited-future)

이 방법은 앞으로 필요한 CPU 부하량을 예측하여 동작한다는 점에서 OPT 알고리즘과 유사하지만 OPT 알고리즘과는 다르게 미래에 요구되는 전체적인 부하량을 분석하지 않고 일정 구간만을 분석하여 운영체제의 idle time 을 조절하는 것이다. Weiser et al은 이 알고리즘 역시 미래에 요구되는 CPU 부하량을 예측할 수 없다는 점에서 사용하기에 적절치 않다고 기술하였다.

#### 2.1.3 PAST (bounded-delay limited-past)

PAST 알고리즘은 CPU 의 가장 최근의 동작 데이터들을 바탕으로 하여 향후의 동작 속도와 동작 전압을 가변시키기 위한 기준을 정하는 방법으로, CPU 동작 데이터들로서 idle cycles, non-idle cycles, 그리고 excess cycles 들을 이용하였다. Weiser et al 의 실험 결과를 보면 CPU 소비 전력의 감소에 이 방법이 효과가 있는데 이는 이 방법을 사용할 경우에 발생하는 불가피한 성능과의 Trade-off 때문이다.

상기와 같은 Weiser et al 의 연구결과는 미래에 필요한 CPU 요구성능의 예측 과 적절한 성능과의 Trade-off 가 있으면 효율적인 CPU 소비 전력 감소를 이룰 수 있음을 보여준다.

## 2.2. Edwin 의 연구

Edwin[3] 은 Weiser et al 의 연구를 더욱 발전시켰는데 그는 향후 CPU 동작의 예측과 더불어서 DVFM 적용에 있어서 smoothing 에 대한 고려 와 Weiser et al 의 실험결과에서 나타난 CPU 전력 감소와 성능과의 trade-off 의 중요성을 부각시키면서 smoothing 면에서 제안된 PAST 알고리즘이 가지고 있는 문제점을 지적하고 PAST 알고리즘을 포함하여 다양한 새로운 scheduling 알고리즘의 실험을 통하여 합리적인 smoothing 알고리즘이 훨씬 효과가 좋다는 것을 기술하였다.

즉, 이와 같은 기존의 연구들을 통하여 CPU 소비 전력 감소를 위한 확실한 방법으로서 동작 전압을 낮추는 방법으로 Dynamic voltage scaling 의 이용이 가능함이 검증되었고, 이의 사용을 위한 정책 결정은 CPU 의 동작 상태정보를 이용한 PAST 기반의 알고리즘과 smoothing 에 대한 고려를 통하여 가능함을 볼 수 있었다.

하지만 앞서 살펴본 Weiser et al 과 Edwin 은 CPU 요구 성능 예측을 하는데 있어서 오로지 CPU idle cycles, non-idle cycles 그리고 access cycles 들만

을 이용하였다. 이로 인해 Weiser et al 은 FUTURE 알고리즘은 미래를 예측할 수 없으므로 비현실적이라고 기술하였다.

하지만 본 논문에서는 CPU 의 전력 소모 감소를 위하여 CPU 상태 정보를 이용한 PAST 기반의 알고리즘과 CPU 주변 장치의 동작상태 파악을 통한 FUTURE 알고리즘의 조합을 이용한 새로운 알고리즘으로서 적응적 동적 전력 관리 기법을 제안한다.

### III. 적응적 동적 전력 관리 기법

제안하는 적응적 동적 전력 관리 기법은 기술한 바와 같이 prediction algorithm에 있어서 앞서 살펴본 기존 연구들과 다음과 같은 중요한 차이점을 가지고 있다.

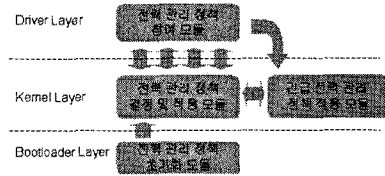
- \* FUTURE 기법 과 PAST 기법 동시 적용.
- PAST 기법 사용: 운영체제의 idle 정보를 이용.
- FUTURE 기법 적용: CPU 주변 장치의 동작 정보 이용

앞서 살펴본 기존의 연구들을 통하여 나타났듯이 CPU 전력 소모 감소를 위한 방법은 prediction 과 smoothing 이 중요하다[4]. Weiser et al[3], Edwin[4]의 연구들은 이 prediction 을 위해 CPU 의 idle cycles, non-idle cycles, excess cycles 들만을 이용하였는데, 이는 CPU 동작 이나 주변 장치들의 동작에 의해 나타나는 결과들이다. 따라서 CPU 주변 장치들의 동작 여부와 동작 유형을 알고 이 정보들을 이용할 수 있다면 CPU의 향후 요구되는 성능 예측에 대한 정확성을 크게 높일 수 있게된다. 즉 해당 순간에 요구되는 동작을 적절한 성능으로 제공하여야 하는 CPU 의 효율적인 전력 소모 관리 요구조건을 좀 더 잘 맞출 수 있는 것이다.

따라서 본 논문에서는 운영체제의 scheduling 정보와 CPU 주변 장치의 동작 상태 정보를 이용한 CPU 요구 성능을 예측 방법으로 적응적 동적 전력 관리 기법을 제안하며 제안하는 기법의 전체 구성(2.1)과 PAST 기법과 FUTURE 기법을 혼용한 CPU 요구 성능 예측 알고리즘(2.2)을 기술하였다.

#### 3.1. 적응적 동적 전력 관리 기법의 전체 구성.

이를 위해 본 논문에서 제안하는 적응적 동적 전력 관리 기법은 운영체제의 구성 드라이버들과 커널이 연계되어 여러 주변 장치들의 동작 상황과 해당 순간 요구되는 CPU 성능에 따라서 CPU 의 동작 주파수와 동작 전압을 제어하는 방식으로 되어 있다. 이를 위해 운영체제 전반에 걸쳐서 크게 동적 전력 관리 정책 결정 모듈과 동적 전력 관리 정책 적용 모듈로 구성이 되어 있다. 전체적인 구조는 아래의 그림 1에 나타나 있다.



< 그림 1. 전체 구조 >

그림 1에서 나타내어진 각 구성 모듈들에 대한 내용은 다음과 같다.

#### 3.1.1. 전력 관리 정책 초기화 모듈:

이 모듈은 reset 과 같은 시스템 초기화 시에 CPU 동작 전압 과 동작 주파수와 같은 전력 관리 정책의 초기 설정을 담당한다. 일반적으로 동작 안정성이 특히 강조되는 부팅 초기화시에는 낮은 전압에서도 동작 가능하도록 CPU 설정을 하는게 좋고 동작 안정화 이후에는 부팅 속도의 향상을 위한 CPU 동작 전압과 동작 주파수 조정이 필요하다. 이러한 이유로 인해 bootloader 단에서의 전력 관리 정책의 초기 설정이 필요하게 된다.

#### 3.1.2. 전력 관리 정책 결정 모듈:

효율적인 전력 관리를 위해 운영체제 동작 중에 운영체제의 idle 시간 정보와 CPU 주변 장치 동작 정보들을 바탕으로 CPU 동작 전압 과 동작 주파수의 변경이 필요한지 그리고 필요하다면 어떤 변경값으로 변경하여야 하는지를 결정하는 모듈이다. 이 전력 관리 정책 결정 모듈은 운영체제의 커널 내부에 위치하게 되는데 이는 유효한 idle 정보등의 파악 과 빠른 대응을 위해서 이다. 최초 부팅시의 전력 관리 정책 초기화 모듈에서 설정한 값에서 부터 시작하여 예측값에 따라 CPU 동작 전압 과 주파수를 변경하도록 한다.

#### 3.1.3. 긴급 전력 관리 정책 적용 모듈.

이 모듈은 CPU 의 주변 기기들의 동작에 따라 해당 드라이버나 시스템 어플리케이션에서 현재의 CPU 동작 전압과 주파수의 변경이 가능하도록 하는 모듈이다. 즉, 드라이버나 시스템 어플리케이션에서는 CPU 성능의 증가가 급하게 필요할 경우 이 모듈을 통하여 전력 관리 정책 결정 모듈의 동작을 기다릴 필요없이 CPU 동작을 조절할 수 있다. 이 모듈의 동작 결과에 따라 변경된 CPU 동작 전압 과 동작 주파수들은 전력 관리 정책 결정 및 적용 모듈과 상호 공유되어야 한다.

#### 3.1.4. 전력 관리 정책 참여 모듈.

전력 관리 정책 참여 모듈은 CPU 주변 기기들을 위한 디바이스 드라이버나 시스템 어플리케이션에서 전력 관리 정책 결정에 참여 할 수 있도록 하는 모듈로서 각 참여 모듈에서는 긴급 전력 관리 정책 적용 모듈을 호출하여 CPU 동작 전압 및 주파수를 변경 하거나 디바이스 드라이버가 가지고 있는 디바이스의 동작 상태 정보를 제공하여 CPU 전력 관리 정책에 반영

되도록 한다.

### 3.2 적응적 동적 전력 관리 기법에서의 CPU 요구 성능 예측 알고리즘.

본 연구에서 사용된 CPU 요구 성능 예측 알고리즘은 PAST 기법[2]을 기반으로 하며, 사용하는 정보들은 크게 운영체제의 idle 상태 정보와 CPU 주변 장치들의 동작 상태 정보를 바탕으로 구성이 된다.

이번 절에서는 운영체제의 idle 상태 정보가 뜻하는 것이 무엇인지를 먼저 살펴보고 사용한 CPU 요구 성능 예측 알고리즘을 기술하도록 한다.

#### 3.2.1 운영체제의 idle 상태 정보

본 논문에서 말하는 운영체제의 idle 상태 정보라 함은 운영체제의 동작 중에 동작 완료 후, CPU 전력 소모 감소를 위하여 주어지는 tick count 기반의 CPU idle 시간을 뜻하는 것으로서 이를 이용하여 CPU의 idle cycles 이나 excess cycles 등과 같은 정보 없이도 현재 CPU 동작에 걸리는 과부하나 여유도 등과 같은 성능에 관한 상태 정보를 파악할 수 있다. 일반적인 컴퓨터 시스템 운영체제에서의 system idle Processor 를 이와 관련한 예로 들 수 있는데, 운영체제의 system idle process 정보는 CPU가 현재 사용할 수 있는 자원의 %를 나타내는 것으로 아무 작업을 하지 않고 있는 경우에는 그림 2와 같이 CPU의 가용자원이 99%까지 올라가게 된다.

이미지 이름	사용자 이름	C...	메모리...
services.exe	SYSTEM	00	24,364 KB
windlogon.exe	SYSTEM	00	1,548 KB
csrss.exe	SYSTEM	00	12,352 KB
smss.exe	SYSTEM	00	444 KB
monsysnt.exe	wha_young	00	4,236 KB
DevObject.exe	wha_young	00	5,020 KB
PsSvc.exe	SYSTEM	00	7,354 KB
explorer.exe	wha_young	00	19,152 KB
PaTray.exe	wha_young	00	3,632 KB
zhnsd.exe	wha_young	00	3,376 KB
npkmc.exe	SYSTEM	00	1,982 KB
runDll32.exe	wha_young	00	4,572 KB
acrotay.exe	wha_young	00	3,628 KB
daemon.exe	wha_young	00	4,380 KB
mailab.exe	SYSTEM	00	30,116 KB
sqlservr.exe	NETWORK SE...	00	3,276 KB
rsapmgr.exe	wha_young	00	7,784 KB
CcmExec.exe	SYSTEM	00	20,004 KB
System	SYSTEM	00	256 KB
System Idle Process	SYSTEM	99	16 KB

<그림 2. CPU 사용율 >

결국, 이는 현재 CPU의 유휴 시간이 많다는 것을 뜻하므로 이러한 상태에서는 CPU의 동작 전압과 동작 주파수를 낮추고 CPU의 동작에 따라 주어지는 유휴시간이 줄어드는 경우는 동작 전압과 주파수를 높임으로써 직접적인 CPU idle cycles 등의 정보가 없는 상태에서도 효율적인 CPU 전력 소모 관리를 할 수 있다.

#### 3.2.2 사용한 CPU 요구 성능 예측 알고리즘.

CPU 주변장치의 동작 여부에 대한 정보는 앞으로 요구될 CPU의 연산 부하량에 상대적인 예측을 가능하게 한다. 예를 들자면, CPU 주변장치의 동작이 전혀 없는 상태에서 USB를 통한 데이터의 전송과 같은 주변 장치의 동작이 시작 되었을 때 앞으로 CPU에 걸리는 연산 부하량이 현재의 연산 부하량 보다 커질 것이라는 상대적인 예측은 충분히 할 수 있으며 반대로 USB로의 데이터 전송 중이었다가 USB의 동작이

완료되면 CPU에 걸리는 연산 부하량이 현재보다는 감소할 것임을 예측할 수 있다.

이러한 상대적인 예측치는 PAST 방식의 알고리즘이 가질 수 밖에 없는 CPU 요구 성능 예측의 오차를 줄여줄 수 있고 성능 대 전력 소모의 비율을 일정부분 제어가능하게 하여 준다.

이에 본 연구에서는 이러한 상대적 예측을 이용한 FUTURE형 알고리즘과 운영체제의 idle 상태 정보를 바탕으로 하는 PAST기반의 알고리즘을 혼합한 새로운 알고리즘을 사용하였는데 내용은 다음과 같다.

1. CPU 요구 성능 예측을 위해 운영체제의 scheduling에 의해 주어지는 CPU idle 시간을 분석.
2. Device 드라이버에서 설정한 각 주변장치의 동작 여부 정보에 따라 DVFM windows 운용.
3. 해당 window에 CPU idle 시간분석 결과가 동일 window에 연속 3회 들어오는 경우에 DVFM의 단계별 적용.

## IV. 구현

본 논문에서는 Marvell PXA320(Monahans-P processor)를 사용한 hardware platform과 Microsoft의 Windows Mobile OS를 이용하여 제안하는 적응적 동적 전력 관리 기법을 구현하였다. 제안 기법에서 CPU 소모 전력 감소를 위해서는 성능 예측 알고리즘을 이용하여 적용 Policy를 무엇으로 할 것인지를 결정 후 CPU의 DVFM을 수행하게 된다. 이때 CPU의 동작 주파수 변경은 CPU의 내부 레지스터를 통하여 할 수 있으나 CPU의 동작 전압을 변경하기 위하여서는 PMIC의 적용과 이의 제어가 함께 필요하다. 이번 장에서는 사용 시스템에 대하여 먼저 살펴보고 구현 방법 그리고 Power Policy의 기준 선정 방법에 대하여 기술하였다.

### 4.1. 구현 시스템

제안하는 CPU 요구 성능 예측 알고리즘의 구현은 LG KCI(WiB개 PDA Phone) 제품을 가지고 하였다. 이 제품에 적용한 이유는 DVFM의 지원이 가능하기 때문인데, KCI에 적용된 Marvell의 PXA320은 동작 주파수가 소모 전력이 가장 작은 Ring Oscillator 모드에서부터 최대 806 MHz까지 모두 8개의 동작 주파수 모드를 가지고 있으며 KCI에서 각 동작 주파수에 대응하는 CPU 동작 전압 조절은 CPU와 I2C 인터페이스를 통하여 연결된 PMIC를 제어함으로써 가능하게 되어있다.

한편, 본 논문에서 사용한 운영체제는 Microsoft의 Windows Mobile 5.0으로 UNIX, Linux와 마찬가지로 IOCTL 함수를 운영체제에서 지원하고 있으며 운영체제 Kernel 영역 중 일부가 개방되어 있다. 따라서 이처럼 개방된 부분을 통하여 제안 기법을 구현할 수 있었다.

### 4.2. 구현 방법

이번 절에서는 구현 방법을 모듈들의 구현 위치와

모듈들의 구현 방법 및 동작 방식 그리고 실질적으로 CPU 의 동작 전압과 주파수를 결정해 주는 Power Policy 의 기준 선정 방법에 대하여 기술한다.

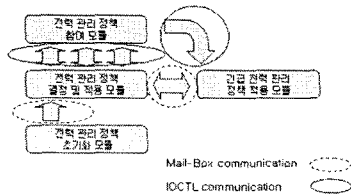
4.2.1 모듈들의 구현 위치

제안 기법은 이미 3장에서 기술한 각 모듈들을 구현함으로써 완성될 수 있는데 각 모듈들의 구현 위치는 다음과 같다.

- \* 전력 관리 정책 초기화 모듈
  - Bootloader 의 Main 함수
- \* 전력 관리 정책 결정 및 적용 모듈(DVFM 포함)
  - Kernel 의 OEMIDLE 함수.
- \* 전력 관리 정책 참여 모듈
- \* 긴급 전력 관리 정책 적용 모듈
  - Kernel 의 OALIOCTL 함수
- \* 전력 관리 정책 참여 모듈 호출자
  - 각 Driver 의 동작 Routine

한편 각 모듈들 간에 반드시 동기화되어 공유되어야 하는 전력 관리 정책 데이터들을 위하여 mail-box 통신을 통하여 공유가 되도록 하였는데, 이유는 커널 영역에서 동작하는 모듈들과 드라이버 영역에서 동작하는 모듈들 간의 전력 관리 정책의 충돌이나 비동기화를 예방하고 직관적이며 속도가 빠르기 때문이다.

전체 구조에서 각 모듈들간에 어떤 위치에서 어떤 인터페이스를 사용하였는지는 아래 그림 3에 나타내었다.



<그림 3. 모듈 인터페이스 >

4.2.1 모듈들의 구현 방법 및 동작 방식

전력 관리 정책 결정 모듈 과 전력 관리 정책 적용 모듈이 위치한 OEMIDLE은 운영체제의 scheduling 동작 중에 불리는 함수로서 기본적으로 CPU 를 idle 상태로 진입 시켜 주는 루틴이다. 또한 함수내의 코드 동작 시간이 길어지는 경우는 운영체제의 scheduling time을 뒤엎리게 만들어 매우 민감한 문제를 일으키게 된다. 따라서 OEMIDLE 함수내에 위치한 전력 관리 정책 결정 모듈과 전력 관리 정책 적용 모듈들의 구현은 Global 변수들과 분기 시에는 빠른 처리가 가능한 switch 구문을 이용하였다. 특히 시간이 제일 많이 소요되는 CPU 의 동작 주파수와 동작 전압 변경 코드는 C 코드가 아닌 ARM 어셈블리어로 구현을 하여서 소요시간을 최소화 하였다.

한편 Device Driver 들의 전력 관리 정책 참여는 각 주변장치들의 동작 상황에 대한 정보들을 IOCTL

을 이용하여 갱신하거나, 직접 OALIOCTL 함수를 통하여 CPU 동작 전압 과 주파수를 변경함으로써 이루어지도록 하였다. OALIOCTL 함수는 커널 단에서 동작을 하기 때문에 디바이스 드라이버에서 해당 OALIOCTL을 호출함으로써 이미 커널에서 동작하고 있는 전력 관리 정책 결정 및 적용 모듈과의 충돌을 쉽게 예방할 수 있다. 또한 디바이스 드라이버들의 전력 관리 정책 참여를 가능하게 하기 위해서는 현재의 전력 관리 정책이 어떤지, 즉 현재의 CPU 동작 전압과 주파수는 어떤지 그리고 어떤 동작 전압과 주파수로 변경하였는지가 기록되어 공유되어야 한다. 이러한 전력 관리 정책에 관한 데이터들은 Mail-box 통신기법을 이용하여 동기화 관리를 하였다.

4.3. Power Policy 기준 선정 방법

앞항에서 모듈의 구현 방법 과 동작 방식에 대하여 살펴보았는데, 여기서는 제안 기법의 구현 시 CPU 의 동작 전압과 동작 주파수를 결정하는 기준 산출 방법을 기술한다. 먼저 PXA 320 이 지원하는 동작 주파수 8가지 중에서 각 DVFM window에서 최대 5가지 주파수 모드만을 사용하였는데 이 때 각 주파수 모드로의 진입 여부를 결정짓는 기준은 운영체제의 scheduling 정보 상에서 미리 실험적으로 측정된 idle count 값을 바탕으로 하였다. 아래 Table 1은 사용한 주파수 모드와 각 모드의 idle count 기준을 나타내었다.

정책 유형	Policy 1	Policy 2	Policy 3	Policy 4	Policy 5
기본 동작 주파수	806 MHz	624 MHz	312 MHz	208 MHz	Ring Oscillator mode
DVFM window 1	624 MHz	312 MHz	208 MHz	Ring Oscillator mode	
DVFM window 2	806 MHz		403 MHz	208 MHz	Ring Oscillator mode
idle count	매우 바쁨	바쁨	보통	약간 여유	여유 많음

< Table 1. 사용 주파수 모드 >

상기 표에서 나타낸 3개의 DVFM window(기본, window1, window 2)들은 전력 관리 정책 참여 모듈들의 요청이 들어오면 바뀌게 되며 전체적인 DVFM windows 들의 설정 또한 전력 정책의 수립에 따라서 유동적으로 조절이 될 수 있다.

V 결론 및 향후 연구 방향

구현 시스템을 이용한 테스트를 통하여 idle count 의 기준에 따라서 CPU 의 동작 전압과 주파수의 변동 폭과 횟수가 많은 차이를 보임을 알 수 있었다. 그리고 CPU 동작 주파수나 전압의 빈번하고 큰 변동은 PMIC전압의 상승 소요 시간과 운영체제의 Scheduling 제한 시간등으로 인해 Hang 과 같은 증상을 유발할 수 있음을 알 수 있었다. 이에 실험을 통하

여 아래와 같은 idle count 값 설정을 통하여 시스템 사용 증 폭이 큰 동작 전압 과 주파수의 변경을 최소화 하게 할 수 있었다.

idle count	매우 바쁨	바쁨	보통	약간 여유	여유 많음
설정 값	1 ~ 9	10 ~ 29	30 ~ 89	90 ~ 149	150 이상

또한 시스템을 사용하지 않는 경우는 CPU 동작 전압과 주파수가 점차 떨어지면서 결국에는 Ring oscillator 모드로 진입하여 cpu idle 상태로 떨어지고 sd 메모리에서 동영상 파일을 읽어서 재생하는 것과 같은 CPU 의 성능이 좀 더 필요한 상황에서는 idle count 의 변동에 따라서 CPU 의 동작 전압과 주파수가 순간 순간 가변되어 CPU 전력 소모량을 전체적으로 줄이면서도 성능 면에서도 또한 손실이 없음을 확인 할 수 있었다.

그렇지만 각 전력 정책에 대한 idle count 의 기준 설정에 관한 연구는 추가적으로 진행되어야 한다. 이는 제안하는 기법에서 운영체제의 idle count 정보를 중요 기반으로 하여 CPU의 동작 전압과 주파수가 결정 되기 때문에 시스템에 최적화된 idle count 를 어떻게 찾아내고 증명해 낼 수 있는가 하는 방법과 최적화된 idle count 설정값을 쉽고 빠르게 찾아낼 수 있는 방법에 관한 연구가 필요하기 때문이다.

### 참 고 문 헌

- [1] Mark Weiser, Brent Welch, Alan Demers, Scott Shenker, "Scheduling for Reduced CPU Energy", manuscript, Xerox PARC, 1994.
- [2] Kester Li, Roger Kumpf, Paul Horton, and Thomas Anderson, "A Quantitative Analysis of Disk Driver Power Management in Portable Computers", Computer Science Division, University of California Berkeley, CA 94720.
- [3] Edwin Chan, Kinshuk Govil, Hal Wasserman, "Comparing Algorithms for Dynamic Speed-Setting of a Low-Power CPU", Computer Science Division, University of California, Berkeley, December 15, 1994.
- [4] Trevor Pering, Tom Burd, and Robert Brodersen, "Dynamic Voltage Scaling and the Design of a Low-Power Microprocessor System", University of California Berkeley, Electronics Research Laboratory.
- [5] K. Choi, R. Soma, and M. Pedram. "Dynamic voltage and frequency scaling based on workload decomposition". In Proceedings of the 2004 international symposium on Low power electronics and design, pages 174 ~179, ACM Press, 2004.
- [6] 이수일, "임베디드 시스템을 위한 동적 전력관리 미들웨어 설계 및 구현", 강원대학교 산업대학원 석사 학위 논문.