Journal of Digital Contents Society Vol. 10 No. 2 Jun. 2009(pp. 217-223)

# 임베디드 시스템에서 DFS 기법을 이용한 동적 전력 관리

권기현*, 김남용**, 변형기***

**요 약**

XScale PXA255 기반 Embedded Linux 환경에서 전력 소비를 줄이기 위해 DFS(Dynamic Frequency Scaling) 기법의 디바이스 드라이버를 제작하고 이 디바이스 드라이버가 포팅되어 있는 임베디드 타겟보드의 전력을 관리하기 위한 미들웨어 DFM(Dynamic Frequency Management)를 설계하고 구현하여 임베디드 시스템의 전력 소비를 감소하는 방법을 제시한다.

# Dynamic Power Management using Dynamic Frequency Scaling in Embedded System

Ki Hyeon Kwon*, Namyong Kim**, Hyung-Gi Byun***

**Abstract**

In order to decrease the power consumption in Embedded Linux environment based on XScale PXA255, We produce the device driver of DFS(Dynamic Frequency Scaling) technique, design and implement the middleware DFM(Dynamic Frequency Management) to scale the power of embedded target board with porting this device drive, suggest the method to reduce the Embedded system's power consumption.

**Keywords : Dynamic Frequency Management, Dynamic Frequency Scaling**

## 1. Introduction

Embedded system is built in different devices from cellular phone, PDA, digital TV to cutting-edge weapons, and a key technology deciding the value added of production, thus it is appearing as a new industry field in which the growth possibility is very high in post pc times, but at the same time the problem for power consumption of processor is on the rise[1].

As a study to reduce the power usage of embedded system, low power WPEG decoder design[2], voltage scheduling in the low power task using the analysis of static time[3], such several researches to decrease the power consumption in embedded system are fulfilled[4].

Due to the electric specific of processor, the more its performance is enhanced, the more sp ending of power is increased. However, the consumption quantity of processor is variable according to the achieved works; the maximum quantity is not always needed. Thus by controlling the voltage and operating frequency supplied to the processor according to the amount used of processor, the power consumption may be reduced. This method is named as DVS(Dynamic Voltage Scaling).

But the embedded system is operated on the basis of embedded processor to treat a specific purpose, and in order to meet this, it is necessary to plan the program not only to maximize the efficiency of target application but to decrease the power consumption. Thus this study suggests the program to induce the lower powe

※ 제일저자(First Author) : 권기현
접수일:2009년 03월 11일, 완료일:2009년 06월 09일
* 강원대학교 전자정보통신공학부
anyjava@empal.com
** 강원대학교 전자정보통신공학부(교신저자)
*** 강원대학교 전자정보통신공학부

r with maintaining the performance of system by applying DFS(Dynamic Frequency Scaling) technique on the basis of benchmark for throu ghout of embedded system according to variab le applications.

## 2. Related Study

### 2.1 DVS(Dynamic Voltage Scaling)

Dynamic voltage scaling technique is a met hod to reduce the mount used of energy by co ntrolling the supply voltage of processor comp osed of CMOS circuit. In general because the power(P) consumption of CMOS circuit has a relation of square to the supply voltage, the re duction of supply voltage is very efficient met hod to decrease the consumption of energy. H owever, when the supply voltage is decreased, the maximum operating frequency of circuit al so slows, thus the throughput of processor is reduced. The technique establishing the speed of clock and voltage in the non-damage limit of service quality demanded for given conduct by using the relations of energy consumption and the processing speed of such processor is the dynamic voltage scaling technique.[5][6][7].

### 2.2 DPM(Dynamic Power Management)

Dynamic power management is a technique to reduce the consumption of power by transfe rring to pertinent power state according to the demands of given performance among the sepa rate power states supplied with applying devic e for inactive condition on the basis of use pa ttern by composition element of system. Accor ding to the specifics of several devices organiz ing system, detailed implementations are varie d but basically the exact estimation for the us e pattern in which applying device is not used is necessary and when the transition to the st ate of low power is determined, the time and energy incidental expenses by state transition should be considered.

Next is power induction formula.

$$P = V \cdot I \quad (1)$$

$$I = \frac{Q}{s}\ [Coulomb/second] = Q \cdot f\ (\because f = \frac{1}{s}) \quad (2)$$

$$Q = C \cdot V\ [Farad \cdot Voltage] \quad (3)$$

$$\therefore P = V \cdot I$$
$$= V \cdot \frac{Q}{s} = V \cdot Q \cdot f = V \cdot C \cdot V \cdot f \quad (4)$$
$$= V^2 \cdot C \cdot f\ [V^2 F/s]$$

### 2.3 DVFS(Dynamic Voltage Frequency Scaling)

The most efficient low power method is DV FS. This is a technique to reduce the power c onsumption by converting the voltage and freq uency of IP(Intellectual Property) including var iable operating frequencies in the system accor ding to the regular task schedule, and these s ystem structures need the sync signal in the c hip and chip communication and due to the IP cores including different operating frequencies, it is consisted of various IO interfaces, thus P PL is needed to occur multi clock for the com munication among them.

### 2.4 DFS(Dynamic Frequency Scaling) of PXA255

Next is dynamic frequency scaling of PXA2 55. DFS is a technique to change the operatin g frequency variably according to the processi ng to be treated.

#### 2.4.1 PXA255 Clock Operator

When the clock of PXA255 is inputted, 3.68 64MHz crystal is connected to the pin of PXT AL and PEXTAL, and 32.768KHz crystal is co nnected to the pin of TXTAL and TEXTAL. For this, oscillators and PLLs are assembled i n the inside of PXA255, so any exterior device

s are not demanded except for the crystal to o ccur the clock.

#### 2.4.2 Core PLL(Phase Locked Loop)

Core PPL supply the clock by using the CP U core, memory controller, LCD controller, D MA controller. These are amplified according t o the following variables<Table 1>.

<Table 1> Frequency Multiplier

| | |
|---|---|
| L | Crystal Frequency to Memory Frequency Multiplier<br>Memory frequency is determined as [memory clock frequency=crystal frequency(3.6864MHz) × L], the value of L may be 27, 32, 36, 40, 45. |
| M | Memory Frequency to Run Mode Frequency Multiplier<br>RUN mode frequency is determined as [RUN mode frequency =memory frequency × M] the value of M may be 1 2 or 4. |
| N | Run Mode Frequency to Turbo Mode Frequency Multiplier<br>TURBO mode frequency is determined as [TURBO mode frequency =RUN mode frequency × N], the vlaue of N may be 1.0, 1.5, 2.0, 3.0. |

<Table 2> Core PLL Output for 3.6864 MHz Crystal

| L | M | Frequency for values "N" and CCCR values of "N" | | | |
|---|---|---|---|---|---|
| | | 1.00 Run | 1.50 | 2.00 | 3.00 |
| 27 | 1 | 99.5 @1.0V | - | 199.1 @1.0V | 198.6 @1.1V |
| 36 | 1 | 132.7 @1.0V | - | - | - |
| 27 | 2 | 199.1 @1.0V | 298.6 @1.1V | 398.1 @1.3V | - |
| 36 | 2 | 265.4 @1.1V | - | - | - |
| 45 | 2 | 331.8 @1.3V | - | - | - |
| 27 | 4 | 398.1 @1.3V | - | - | - |

In the Table 1, the generating power of Cor e PLL according to the value of L, M, N for 3.6864 MHz is presented, this generating powe r frequency is 400MHz maximally in the TUR BO mode. The output frequency selections are shown in <Table 2>.

### 2.5 Clock Configuration-Related Registers.

#### 2.5.1 Core Clock Configuration Register(CCCR)

Thorough the CCCR register, memory frequ ency multiple(L) related with Core Clock frequ ency, RUN mode frequency multiple(M), TUR BO mode frequency multiple(N) are configured.

#### 2.5.2 CCLKCFG Register of Coprocessor 14

2.5.1 Core Clock Configuration Register(CCC R) is used for beginning of the TURBO mode and frequency variation sequence. When it beg ins as the frequency variation sequence, in ord er to keep constant the Turbo Bit, Read-Modif y-Write process, that is, the stage of reading this register then modifying only applying bit and writing it again should be employed.

### 2.6 Frequency Variation Process

The frequency variation process is used wh en clock frequency of Core PLL is converted a nd in this process, CPU, memory controller, L CD controller and DRAM clocks related with Core PLL are stopped and modules except for them are operated. This process is used to ch ange the frequency configured as default at an early stage of booting, or may be employed to operate with lower frequency to spare the pow er by processor.

## 3. DFM Middleware Design

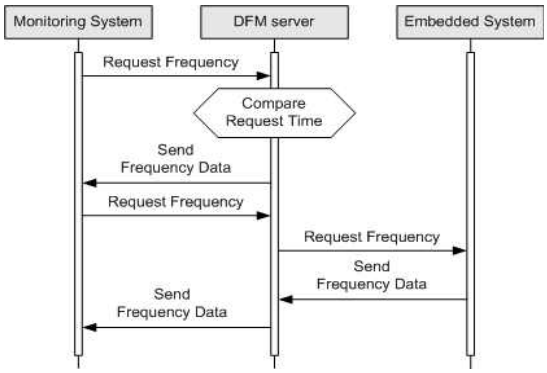
### 3.1 System Structure Design

Among some elements of this system, monito ring system regularly checks the amount of po wer and transmits a needed frequency scaling p arameter value to DFM sever and makes it pos sible to scale the power of target board(Fig. 1).

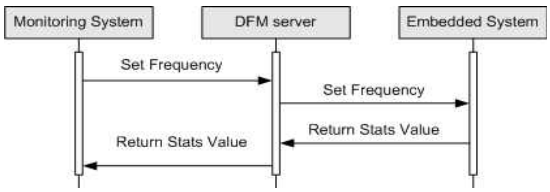

(Fig. 1) System Structure

### 3.2 State Transition Diagram

When monitoring system asks DFM server to send a frequency of specific target board, t he DRM server including all data of target sy stem gets to ask the target board to send freq uency data. Reading the current CCCR of it th e target board transmits the data to DFM ser ver, after receiving this, the DFM server trans mits frequency information to the monitoring s ystem(Fig. 2).




(Fig. 2) The Protocol Flow of Frequency Information Request

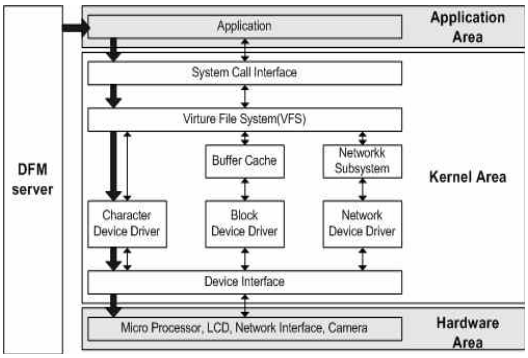



(Fig. 3) The Protocol Flow of Frequency Setting

In adverse, during the frequency setting, the number of target system, frequency values are transmitted in the monitoring system. After re ceiving this, DFM server sends the frequency and device number to the target system and a djusts the setting. Following figure is protocol flow during the frequency setting of target bo ard(Fig. 3).

### 3.3 DFS Device Driver Implementation

The module for fulfillment of DFS is imple mented in the device driver. The sort of devic e driver suited to implement DFS is character device driver, which is located in kernel level and scales the frequency by applying the data transmitted from DFM server to PXA255 proc essor through the device driver(Fig. 4).




(Fig. 4) Data Flow of Target System

Next the source is the part of Device Drive r for frequency scaling and control the CCCR register by ioctl function(Fig. 5).

The number 2, 3 lines configure CCCR regi ster and change the CPU frequency. The num ber 4, 5 lines change the coprocessor register to change the operation speed. The number 6, 7, 8 verify the existence of turbo mode. The n umber 9, 10 lines activate the turbo mode if t he turbo mode exists.

```
1  __asm__ __volatile__("
2    ldr r0, =0x41300000 @ PXA_REG_CCCR
3    ldr r1, =0x241        @ 1001000001
4    mov r0, #(1 << 1)
5    mcr p14, 0, r0, c6, c0, 0
6    and  r1, r1, #(0x7 << 7)
7    cmp r1, #(0x2 <<7)
8    beq 10f
9    mov r0, #(1 <<0)
10    mcr p14, 0, r0, c6, c0, 0
11    10:"
12    :
13    :
14    : "r0");
```

(Fig. 5) The Part of the Device Driver

# 4. Implementation & Evaluation

This paper explains the mode l of actual sy stem and investigates the results for test. And next is the implementation and evaluation of t he system.

## 4.1 Test Environment for DFM

Laptop, PC which we usually use become th e monitoring system and window, windows C E are porting as operating system. DFM serve r is implemented in general PC and the operat ing system is Linux, and the target system eq uips the embedded Linux.

<Table 3> DFM server Specification

| | |
|---|---|
| MODEL | Samsung Magic Station m2761 |
| OS | Linux 2.2.17 |
| Processor | Intel Pentium III 800MHz |
| Memory | 448 MB |

<Table 4> Embedded System Specification

| | |
|---|---|
| MODEL | X-Hyper255B |
| OS | Embedded Linux 2.4.18 |
| Processor | Intel PXA255 400MHz |
| Memory | 32MB Flash ROM, 64MB SDRAM |

The specification of PDA, DFM server and embedded System is in <Table 3> and <Tabl e 4> separately.

## 4.2 Performance Evaluation

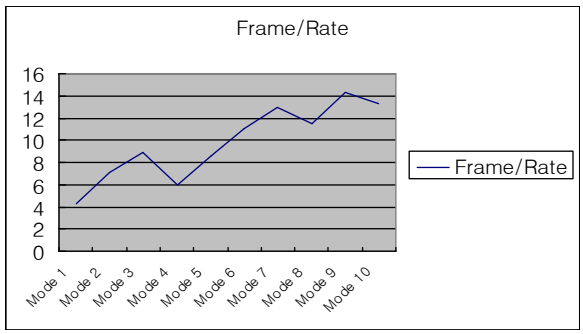
The file used in test is the video supplied b asically by embedded device-manufacture enter prise its spec is <Table 5>.

<Table 5> Testing File Specification

| | |
|---|---|
| File name | holeman_cf.mpg |
| File Size | 3.63MB |
| Average Data Rate | 170.19KB per second |
| Image Size | 352 × 240 |
| Frame Rate | 29.97 fps |

The frequencies for each modes are as follo ws: mode1:99.5MHz, mode2: 199.1MHz, mode3: 298.6MHz, mode4: 132.7MHz, mode5: 199.1MH z, mode6: 298.6MHz, mode7: 398.1MHz, mode8: 265.4MHz, mode9: 331.8MHz, and mode10: 398. 1MHz; 400MHz.

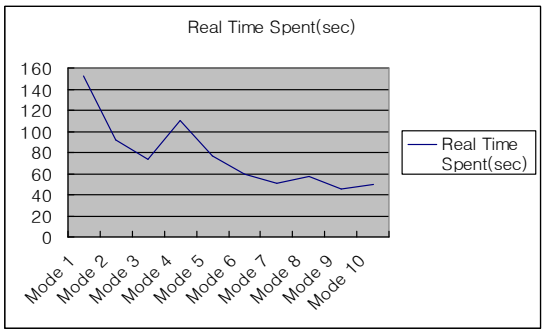
The Frame rates when the files <Table 5> are running for each CPU frequency scaling ar e as (fig. 6).




(Fig. 6) Frame Rates for Frequency Scaling

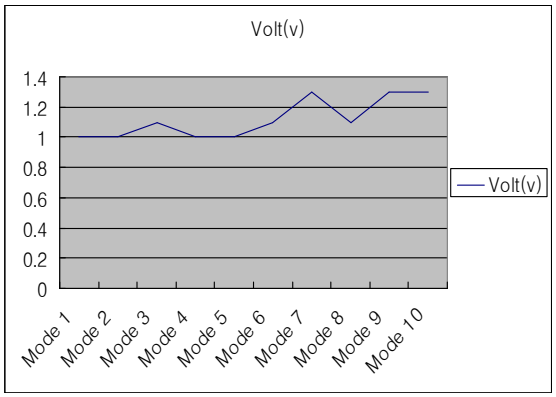
From the [fig. 6], it is noted that the frame rate of video file implemented for clock frequen cy, 99.5MHz in mode 1 is lower than the frame rate of the frequency, 398.1MHz in mode 10. It shows that mode 4, 8 and 10 has lower frame rate compared with mode 3, 7, 9 respectively.

It is found that in mode 1 the video file pro cessing time needs 152 seconds but in mode1 0, the processing time needs only 49 seconds (Fig. 7). It shows that mode 4, 8 and 10 has l ower time spent compared with mode 3, 7, 9 r espectively.




(Fig. 7) Time Spent Rates for Frequency Scaling

This chart demonstrates the voltage for each frequency mode[Fig. 8]. In mode 1, the voltage of 1V is started up, but in mode 10 1.3V is started up. It shows that mode 3 and 4 has lower voltage consumption compared with mode 7, 8 respectively.




(Fig. 8) Voltage Change by Frequency Scaling

The video processing in high clock is faster than the video processing in low clock and the frame rate is also higher. However, it is also recognized that the amount of power consumption is proportioned to that.

## 5. Conclusion

This paper studied the design and implementation of dynamic power scaling middleware using the DFS technique for reducing the power consumption, with focusing on microprocessor.

In general the power consumption may be significantly fluctuated according to the component installed in the embedded device , the service of the device, application softwares used.

Suggested low power system mode l is a software based on middleware, and the reductio n plan was suggested to change the frequency according to the processor spec then decrease the power consumption by controlling the microprocessor, and on the basis of this, not only in middleware field but also in operating system field, the reduction of power consumption should be studied.

## 참고문헌

[1] 성경모 외 “임베디드S/W: 국제경쟁력 분석을 통한 산업육성방안”, KISTI, 기술산업분석, 2004.

[2] 손동환 외 “DVS를 이용한 저전력 WPEG 디코더”, 한국멀티미디어학괴, 학술대회지, pp. 35-40, 2001.

[3] 신동군 외 “정적 시간 분석을 이용한 저전력 태스크 내 전압 스케줄링”, 한국정보과학회논문지, 시스템및 이론, pp. 561-572, 2001.

[4] 김지홍 “휴대용 임베디드 시스템에서의 전력 관리”, ITFIND 주간기술 동향, pp. 1195, 2005.

[5] K. Flautner, S. Reinhardt, and T. Mudge, “Automatic Performance-Setting for Dynamic Voltage Scaling,” in Proc. of 7th Conference on Mobile Computing and Networking, July 2001.

[6] D. Grunwald, P. Levis, C. B. M. III, M. Neufeld, and K. I. Farkas, “Policies for Dynamic Clock Scheduling,” in Proc. of the Fourth USENIX Symposium on Operating Systems Design and Implementation, Oct. 2000, pp. 73-86.

[7] A.Varma, B. Ganesh, M. Sen, S. R. Choudhury, L. Srinivasan, and J. Bruce, “A Control-Theoretic Approach to Dynamic Voltage Scheduling,” in Proc. of International Conference on Compilers, Architectures and Synthesis for Embedded Systems, July 2001, pp. 255-266.

[8] Marc A. Viredaz, Lawrence S. Brakmo, and William R. Hamburgen, “Energy Management on Handheld Devices,” ACM QUEUE, vol. 1, no. 7, Oct. 2003, pp. 44-52.

## 권 기 현

1995년: 강원대학교 대학원 컴퓨터과학과 졸업(이학석사)
2000년: 강원대학교 대학원 컴퓨터과학과 졸업(이학박사)
1996년～2002년 : 동원대학 인터넷정보과 교수
2002년～현 재 : 강원대학교 공학대학 전자정보통신공학부 교수
관심분야 : 미들웨어, 임베디드소프트웨어, 무선센서 네트워크, U-Healthcare

## 김 남 용

1988년 : 연세대학교 전자공학과 (공학석사)
1991년 : 연세대학교 전자공학과 (공학박사)
1992년～1998년 : 관동대학교 전자통신공학과 부교수
1998년～현 재 : 강원대학교 공학대학 전자정보통신공학부 정보통신공학전공 교수
관심분야: Adaptive equalization, RBFN algorithm, odor sensing systems.

## 변 형 기

1990년: Manchester University 전자전기공학과 MSc.
1995년: Manchester University 계측공학과 Ph.D.
1996년～현 재 : 강원대학교 공학대학 전자정보통신공학부 정보통신공학전공 교수
관심분야: 전자 후미각 시스템 및 패턴인식