

# DESIGNING PLATFORM-BASED SYSTEM POWER MANAGEMENT ON A SMART TABLET APPLIANCE

*Hung-Ming Chen<sup>1,3</sup>, Po-Hung Chen<sup>1</sup>, Tai-Jee Pan<sup>3</sup> and Feipei Lai<sup>1,2</sup>*, Senior Member, IEEE

Dept. of Electrical Engineering<sup>1</sup> & Dept. of Computer Science and Information Engineering<sup>2</sup>  
National Taiwan University  
Center for Research of Advanced Information Technologies (CRAIT)<sup>3</sup>, Tatung Company  
Taipei 106, Taiwan

## ABSTRACT

For the design of portable systems, power consumption is as important a design criterion as performance. Low power design starts at the system level and a top down approach will yield the greatest results. Previous researches only address on partial stages of platform-based power management. Hence, in this paper, we consider the low power design on the whole system to explore the methodology and implementation from the circuit design stage including selection of key components, analysis of hardware architecture and the platform design to dynamic power management on battery management techniques, operating systems and software design for various Hardware/Software co-design issues. We will take the tablet appliances platform as an example to illustrate how to design the power management system of the handheld device of medium to large sizes, and how to implement the advanced software power management mechanism on embedded OS like Microsoft Windows CE.NET.

## 1. INTRODUCTION

The Internet is widespread from work needs to family and personal leisure use. There are more and more indications that Internet will comprehensively apply to the personalized information appliances in near future.

In order to satisfy the application areas of these smart handheld devices, we can conclude the trends as follows.

1. Using color and high-resolution display.
2. Using high performance, low power and highly integral chip.
3. Open standard software ranges from standalone applications to cross-platform, open connectivity standard, and highly componentized ones.
4. Devices built-in or integrated with various wireless modules.

It is known that, from the above, to achieve the goals for highly integral chip, high performance CPU, high-resolution display and wireless applications for

This research was supported in part by grants from CRAIT of Tatung Company.

multimedia and Internet is considerably power demanding for portable devices with limited battery capacity. Since it will greatly affect the usage time for the users to operate these applications, the design for low power system becomes the main consideration on those devices.

In a whole platform design for the system power management, most of recent researches only address on techniques in the run-time stage such as CPU speed-setting [14], dynamic voltage scaling [1][9], battery management [5][6][7] etc. or key components design such as CPU, memory system [1] and battery [8] etc.

Hence, in later sections, we will explore the methodology and implementation of a platform-based power management from the circuit design stage to run-time stages.

## 2. THE METHODOLOGY FOR DESIGNING A PLATFORM-BASED SYSTEM POWER MANAGEMENT

As to a platform design from scratch, we must consider the design techniques of system power management on both SW/HW co-design from design stage to system run-time stage. Therefore, we will discuss the considerations on designing the platform-based power management on portable devices from various aspects. In Figure 1, we enumerate several design methodologies for achieving energy efficiency in a system from platform point of view.

### 2.1. Low power hardware platform design

Specification required by customers will often affect the hardware platform design, including price, functionality, and performance. In addition, handheld devices require long battery life. Hence, low power hardware design implies the consideration of low power consumption in the initial design stage, including low power components selection, architecture design, finding out the most power consumption parts, circuit design and dynamic power management schemes in HW/SW co-design optimization. Illustrations are as follows.

1. Low power components selection
2. Hardware architecture analysis and design
3. Circuit design

## 2.2. Battery characteristics consideration

From platform designer's point of view, they are usually only concerned about the output voltage and capacity of the battery. For an ideal battery, its voltage and discharging rate is a fixed value. Another battery characteristic is so-called recovery effect. Intermittent discharge of a battery can have bigger capacity than the continuous discharge one [5]. Besides, the factors that will affect the effective capacity of a battery are based on the temperature, charge/discharge current and the cycle count. Figure 2 is an example of the effects of different discharge current and battery cycle counts. From our experiments, we use 2 single cells of Li-polymer battery with nominal capacity 2000mAh for two charge/discharge conditions: Figure 2 uses one charge/discharge condition is 0.5C (1.0A) and the other is double charge/discharge current (1C). From the result, effective battery capacity are affected by battery cycle times and charge/discharge currents

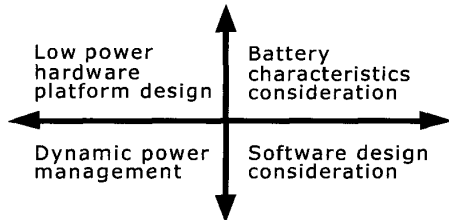


Figure 1. Various ways to achieve platform-based system power management.

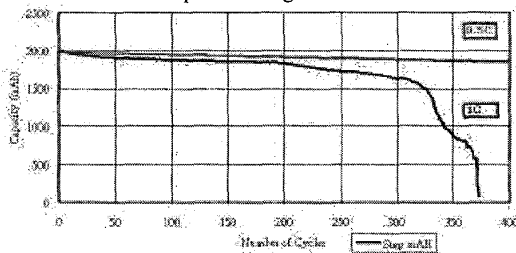


Figure 2. The 0.5C and 1C charge/discharge life cycle test of a single Li-Polymer battery

## 2.3. Dynamic Power Management

Usually systems with dynamic power management are provided with the characteristic of changing the power state of the system components based on the performance requirement of the system. Components can transfer between active and low power states depending on the power consumption and the transition time. The components and related circuits must support the control of components in different power states, and the software controls the state transition according to system run-time status.

Except for the dynamic power management, another way to achieve the low power consumption during system

run-time is dynamic adjustments of CPU frequency and voltage [2]. There are many CPUs provided with this technology no matter in PCs or embedded systems, e.g. Intel Pentium processors with SpeedStep technology, Transmeta Crusoe processors with LongRun™ technology, Intel Xscale with dynamic voltage and frequency scaling capability, and National Semiconductor IA on chip SC3200 with clock throttling capability etc. These techniques dynamically adjust the speed levels based on the performance and power consumption when the device is on duty.

## 2.4. Software design consideration

Due to the different application software, it makes handheld platforms apply to different domains. If the usage of the software can be profiled, it will be possible to design a more power-saving device during run-time.

Here, we illustrate some kinds of implementation methods from previous researches from high-level software point of view.

1. Design and implement a power-aware compiler [4] [15].
2. Establish a set of coding guidelines based on the system characteristics, and let the programmers write the code according to the guideline [3].
3. Collaborate the characteristic of application software such as loading with hardware to dynamically adjust the system processing performance [1].

## 3. DESIGNING PLATFORM-BASED SYSTEM POWER MANAGEMENT ON A SMART APPLIANCE SYSTEM

Next, we will describe the design and features of a platform-based system power management using a smart tablet appliance platform named VIVA developed by us. The goal is to build up assessment architecture of system power management for the tablet type platforms, and utilize the embedded OS architecture like Microsoft CE.NET architecture to establish a power management system to satisfy the user's requirement with different user scenarios.

VIVA platform is a wireless tablet appliance, which belongs to the market of 8.4"-12.1" median sizes display devices, and mainly apply to commercial markets for education, medical equipment, retailer or offices market segments.

### 3.1. Designing the architecture of a smart tablet appliance

Figure 4 depicts the hardware block diagram of a tablet appliance system.

### 3.2. Defining power states and transition

Next, we will describe the methodology of defining the power states and transition diagram. Besides that the system enters suspend or OFF mode, the platform should be capable of defining more power states to adjust dynamically. Thus, we define more states according to the requirement as in Figure 5.

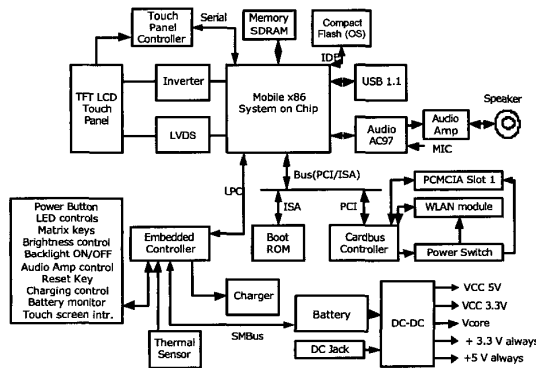


Figure 4. Hardware block diagram of a tablet appliance system

There are two idle modes defined in addition to suspend and OFF modes. We define all power states of system and major components in Table 1. In the Suspend mode the system have to keep memory data, hence besides the embedded controller (EC), which controls the power button and battery states, SDRAM and CPU will be set into the low power state, the other peripheral components will be shut down. As for the two idle modes, the idle time can be defined, and which devices will enter low power mode can be selected based on the usage modes.

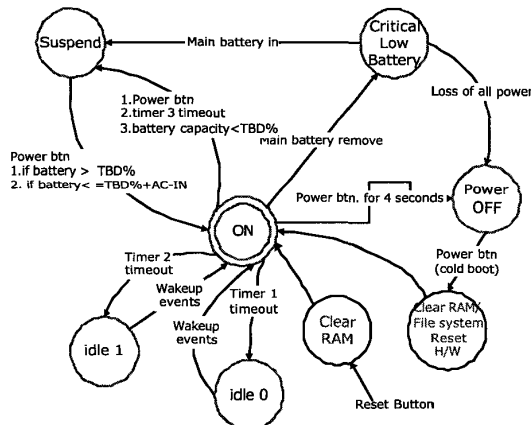


Figure 5. State diagram of system power transition

### 3.3. Design of the advanced software power management on embedded OS

We implement an advanced dynamic power management mechanism through the Microsoft Windows CE.NET platform and take it as the real case study here.

Just like other embedded operation systems, Windows CE.NET does not support APM/ACPI power management system as in general windows systems. Its subsystem architecture of power management defines three major power states, ON, Suspend to RAM and IDLE. Designers can customize and extend the power states based on the platforms. Figure 6 depicts the architecture

and flow of the power management driver. System can dynamically unload the power states supported by components. The application software in the upper layer can register through the power management program interface (API). The system will monitor each component, and broadcast to inform the driver of each component and the registered application software when the state has been changed.

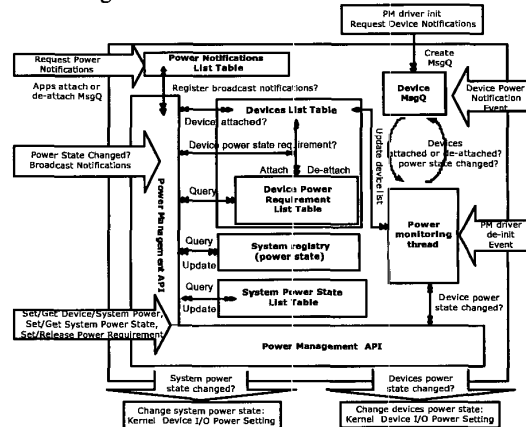


Figure 6. The power management architecture of Windows CE.NET

## 4. EXPERIMENTAL RESULTS

The devices under test were: VIVA board with built-in 802.11b WLAN, access point, gas gauge interface board and PC. Figure 7 illustrates the experimental environment. The gas gauge interface and software provides connections for the communication lines of a smart battery with a serial interface port of a Windows-based PC. It can measure values of the discharge current, temperature, and voltage and let us to calculate the power consumption of the system supplied from battery.

Users can set two idle timers and select which devices to be shut down via the advanced power management program developed by us.

In Table 2, we observe the ON mode and two defined IDLE modes, where in IDLE 0 mode, backlight brightness reduces to its half as well as in IDLE 1 mode, backlight is turned off and CPU speed throttles to 33MHz. Table 2 shows the system power consumption in each mode. It shows that the power will be saved about 10% in IDLE 0 and 47% in IDLE 1.

In addition, as Table 3, we also observe the system power consumption for turning ON/OFF WLAN module and enabling the power saving setting of the WLAN module. It shows that about 10% power saving with power saving mode enabled.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we explore platform-based system power management techniques of smart tablet appliances. The proposed methodology has used from the platform design to power states definition based on the real platform.

**Table 1.** Each power state of the system and major components

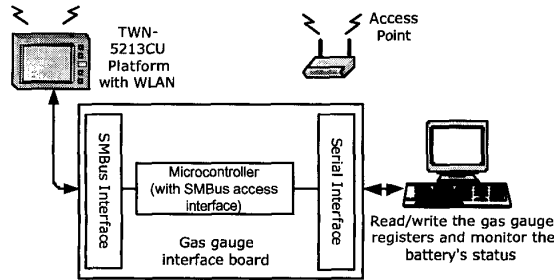
STATES	DESCRIPTION	DEVICE POWER STATES							
		CPU (SC3200)	Memory (SDRAM)	LCD Backlight	Wireless (WLAN)	Embedded Controller	LED Indicators	Audio Amp	PC card controller
OFF	All power removed	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
ON	Normal operation	ON	ON	ON	ON	ON	ON	ON	ON
IDLE0	User defined - Reduced power level	ON/Throttling	ON	Partial Mode	ON/Power Saving	ON	ON	ON	ON
IDLE1	User defined - Power saving level	ON/Throttling	ON	Backlight OFF	ON/Power Saving	ON	ON	OFF	ON
SUSPEND	System suspended and device is nonfunctional	Deep Sleep	Idle	Backlight OFF	OFF	Low On	OFF	OFF	OFF

**Table 2.** The comparison table of each system power state

POWER STATES	Voltage V (Volts)	Current A (mA)	Power (W)=V × A
ON mode	11.104	755	8.38352
Idle 0 mode (Backlight - Partial mode: 50% of max; CPU Speed: 266MHz)	11.104	683	7.584
Idle 1 mode (Backlight - OFF mode, CPU throttling to 33MHz)	11.104	401	4.452

**Table 3.** The power consumption of system with wireless module being shut down or in power saving mode

POWER STATES	Current A (mA)		Power (mW) = V × A (V=16Volts)	
	Max	Min	Max	Min
ON mode with WLAN OFF	511	450	8176	7200
ON mode with WLAN ON	590	524	9440	8384
ON mode and WLAN in power saving mode	534	471	8554	7536



**Figure 7.** Experimental Environment

From some experimental results, it shows that the advanced power management has remarkable power saving effects.

In the future we will keep on developing more advanced power control schemes such as battery control and management is in progress to enhanced battery life.

## 6. REFERENCES

- [1] M. Pedram and J. Rabaey, *Power Aware Design Methodologies*, Kluwer, 2002.
- [2] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, G. De Micheli, "Dynamic Voltage Scaling for Portable Systems," *DAC 2001*.
- [3] T. Simunic, L. Benini, G. De Micheli, M. Hans, "Source Code Optimization and Profiling of Energy Consumption in Embedded Systems," *ISSS 2000*.
- [4] V. Tiwari, S. Malik, and A. Wolfe, "Compilation Techniques for Low Energy: An Overview," *Symposium on Low-Power Electronics, San Diego, CA, Oct. 1994*.
- [5] L. Benini, G. Castelli, A. Macii, E. Macii, R. Scarsi, "Battery-Driven Dynamic Power Management of Portable Systems," *ISSS-00: IEEE International Symposium on System Synthesis*, pp. 25-30, Madrid, Spain, September 2000.
- [6] L. Benini, G. Castelli, A. Macii, R. Scarsi, "Battery-Driven Dynamic Power Management," *IEEE Design and Test of Computers*, Vol. 18, No. 2, pp. 53-60, April 2001.
- [7] Peng Rong, Massoud Pedram, "Battery-Aware Power Management Based on Markovian Decision Processes," in *Proc. of Int'l Conference on Computer Aided Design*, Nov. 2002.
- [8] M. Pedram and Q. Wu, "Design Considerations for Battery-Powered Electronics," *IEEE Transactions on VLSI Systems*, 2002.
- [9] K. Choi, K. Dantu and M. Pedram, "Frame-based Dynamic Voltage and Frequency Scaling for a MPEG Decoder," in *Proc. of Int'l Conference on Computer Aided Design*, Nov. 2002.