Study on Embedded Operating System for Reduction of Power Consumption

Hiroki Zushi (0710040)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 5, 2009

Keywords: power consumption, cache memory, embedded OS, system call, programmer.

1 Introduction

Recently mobile computing devices with complicated functions such as a mobile phone or a digital camera is spreading widely. Because these mobile devices require continuous and long battery life when using the battery, it become an considerable subject to reduce power consumption.

It is expected that the embedded processors cache memory will occupy most of the power consumption in the near future. This is because most of the power of the processor can be considered that it is consumed by the cache memory, and cache area in the processor grows as the processors performance increases. Software Self-Invalidation is one of the methods to reduce power consumption of the cache memory.

In this research, Software Self-Invalidation method is applied to embedded OS. At the same time, application developers are provided with power reduction support environment. The power reduction in embedded systems will be accomplished by using the method described above.

2 Software Self-Invalidation

Software Self-Invalidation has adopted the Last-Touch Instruction (last-touch load/store) as an instruction set. Although the Last-Touch instruction is similar to CPU as the traditional load/store instructions, it has an additional feature that invalidates the target cache block after completing the access to the target memory block. An invalidated cache block is controlled by Gated-Vdd, and the power supply is cut off accordingly.

The data on the cache will continuously consume power although by the memory access instruction, the certain cache block is not expected to be referred once again. It is possible to reduce power consumption by automatically invalidating the cache block by replacing the last reference instruction of the cache block to the Last-Touch instruction in these cases.

Embedded OS based on μITRON4.0 is targeted in this research. Embedded OS optimized for power is provided by applying the Software Self-Invalidation to the system call of the embedded OS. Furthermore, an environment which allows the Last-Touch instruction to be easily applied is provide to the application developer. In these ways, power reduction is accomplished in both the OS and the application.

3 Power optimization embedded OS and Power reduction support environment

The purpose of this research is to make the user of OS benefit in power reduction by embedding the Last-Touch instruction to the system call of OS . I focused attention on the mail box which is one of the data communication functions which ITRON specification OS offers, And additionally the Last-Touch instruction is applied to the fixed-length memory pool which is one of the memory control function which is generally used together. Normally, the data from the memory block got from the memory pool is stored in the mail box, and the memory block is transferred / received between tasks. It is thought that the timing when the data transfer / receive is finished and the memory block returns to the memory pool is optimized when using last touch memory instructions. The user of OS benefits the power reduction effect in OS level by calling the system call which returns

a memory block from the received task.

Furthermore, since the tuning of the program by an application developer is common in an embedded system, power reduction can be optimized by the programmer. In this research, the programmer is provided with an interface that embeds the Last-Touch instructions in a program. Specifically, A specifier that replaces the conventional instruction to the Last-Touch instruction is provided. Letting the programmer to embed a specifier within the code. Because the programmer can use the tool which applies the Last-Touch instruction based on such the specifier, it will become possible to control power easily.

4 Evaluation

The proposed method will be evaluated by simulation. The task set for the evaluation was written assuming that the embedded application used generally treated data processing. The task set was prepared according to multiple situations, such as when the Last-Touch Instruction is applied by the programmer according to the local arrangement of the task, or when the Last-Touch Instruction is applied by the OS according to the message communication among the task.

The total power consumption of the L1 data cache memory is measured by the simulator. Reduced power consumption was confirmable in both the OS and application by the simulation results. These results confirms that both when the proposed method is applied to the OS and application, satisfying power reduction is achievable.

5 Conclusion

In this research I have focus researched the cache memory, in which that has a probability to occupy most of the power consumption, and in addition, I have applied the Software Self-Invalidation to the embedded OS, which is one of the power reduction methods of cache memory. Moreover, the environment that supports the application developer to reduce the power consumption was proposed. Finally, the power reduction rate of L1 data

cache memory was evaluated using the task set created assuming data the embedded application used generally treated data processing.

References

[1] T.Fujita. 'Research on Low power consumption of Cache memory by Self-Invalidation', Japan Advanced Institute of Science and Technology Master thesis 2007.

[2] Tron Association. 'μITRON4.0 Specification Ver.4.02.00', http://www.ertl.jp/ITRON/home-e.html.