A Prototype of A Multi-Core Wireless Sensor Node for Reducing Power Consumption

Sotaro Ohara Makoto Suzuki Shunsuke Saruwatari Hiroyuki Morikawa
Morikawa Laboratory
Research Center for Advanced Science and Technology, The University of Tokyo
{sotaro, makoto, saru, mori}@mlab.t.u-tokyo.ac.jp

Abstract

This paper presents initial experiment results toward realizing a multi-core CPU for wireless sensor nodes. The multi-core CPU reduces power consumption with enabling users to easily manage hard real-time tasks. The results show a sensor node with triple CPUs can eliminate about 76 % of power consumption compared to a single CPU sensor node.

1. Introduction

Wireless sensor nodes have to satisfy two objectives, which are reduction of power consumption and hard real-time operation. It is difficult to satisfy the objectives at the same time with a single-core CPU because of the following reasons. The single-core CPU has to process many tasks concurrently. If two or more tasks have deadlines, the deadlines increase CPU load. To guarantee all the deadlines, users have to select the CPU frequency with considering the worst case. However, higher frequency causes higher power consumption. Additionally, the users also have to care about conflict management among the tasks.

In this paper we presents a multi-core CPU can reduce power consumption with enabling users to easily manage hard real-time tasks. The multi-core CPU allows distributing tasks to several cores. The distribution decreases CPU frequency, and reduces power consumption. The distribution also enables users to easily manage hard real-time tasks. We have developed a multi-core CPU circuit prototype, and evaluated the power consumption according to a number of CPUs using the prototype. The results show a sensor node with triple CPUs can eliminate about 76 % of power consumption compared to a single CPU sensor node.

This paper is organized as follows. Section 2 describes the problems of single-core CPU design wireless sensor networks. Section 3 discusses on a multi-core CPU design and its benefits. The implementation of prototype and its evaluation are shown in section 4. Finally, section 5 concludes our work.

2. Problems on A Single-Core Sensor Node

A wireless sensor node has to achieve the following two requirements at the same time. First is a hard real-time operation. A wireless sensor node has many hard real-time tasks such as sensing, calculating, communicating, and so on. These tasks have to be completed within their deadlines for accurate operation. Second is low power consumption. Since a wireless sensor node has a battery that has limited power source, it is crucial to minimize power consumption in wireless sensor applications. However, there is a trade-off between the requirements. To satisfy the first requirement, CPU must be driven at sufficiently high frequency, but the higher frequency increases power consumption.

The trade-off is related to the CPU structure. CPUs are constructed of CMOS transistors. They consume electrical energy when CMOS transistors are switched, and required driving voltage depends approximately linearly on clock frequency. According to [4], power consumption P on CPU is approximately derived as follows:

$$P = \alpha \times C_L \times V_{dd}^2 \times f \quad (V_{dd} \geq \beta \times f) \tag{1}$$

where the α is ratio of switched transistors to all the transistors in the CPU, C_L is load capacity of the CMOS transistors of the CPU, V_{dd} is power supply voltage, f is CPU clock frequency, and β is proportional constant that varies according to the CPU circuit architecture. As we can see from the above equation, it is important to decrease voltage to preserve power consumption of the CPU.

To verify the equation (1), we evaluate the dependency of power consumption on clock frequency using Microchip PIC18 Microcontroller [3]. We calculated power consumption per instruction with division of measured power consumption on the used clock frequency. According to [3], the maximum frequency f_{max} at which a PIC18 microcontroller can run when voltage V is supplied is defined as follows:

$$f_{max} = 16.36 \times (V - 2.0) + 4 \quad (V \geq 2.0) \tag{2}$$

Table 1. Frequency v.s. Guaranteed Voltage

Frequency	Voltage	Frequency	Voltage
2 MHz	2.00 V	10 MHz	2.37 V
4 MHz	2.00 V	12 MHz	2.49 V
6 MHz	2.12 V	16 MHz	2.73 V
8 MHz	2.24 V	20 MHz	2.98 V

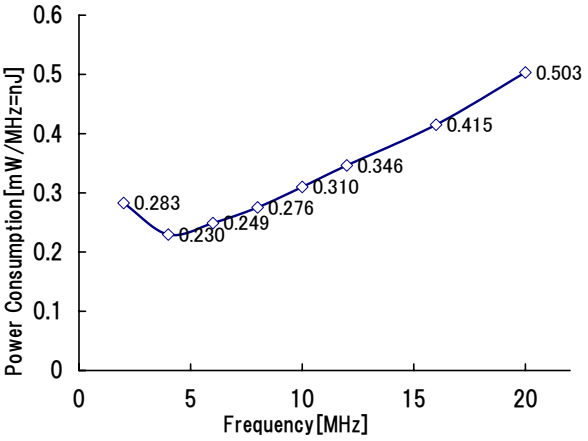

Figure 1. Power Consumption per Instruction v.s. Clock Frequency

We evaluated power consumption in case of using 2 MHz, 4 MHz, 6 MHz, 8 MHz, 10 MHz, 12 MHz, 16 MHz, 20 MHz ceramic resonators. Table 1 shows the selected frequency and the corresponding voltage.

The evaluated power consumption per instruction is shown in Figure 1. This results shows that it is possible to reduce power consumption by reducing clock frequency since it allows selecting lower voltages. For example, a 20 MHz CPU consumes about 220 % more power than a 4 MHz CPU.

3. Multi-Core CPU for Wireless Sensor Networks

A multi-core CPU allows distributing tasks among several cores. This provides an efficient solution to make CPU of wireless sensor nodes provide hard real-time operation with decreasing power consumption.

If we use a multi-core CPU, it becomes easy to implement each task with satisfying the deadline because each core performs only a few numbers of tasks (ideally only a single task). This allows users to implement each task without considering the other tasks in terms of time restriction. For this reason, we believe that the multi-core CPU for wireless sensor nodes provides us easiness of building applications.

Additionally, the multi-core CPU can decrease power consumption. If it is possible to set optimal clock frequency to each core, users can select the lowest CPU clock frequency that complete the assigned task within its deadline since the user may know the deadline of each task. Therefore, the sum of the frequency that satisfy each deadline of a task assigned to a core is lower than the frequency that satisfies all the deadlines with a single-core CPU. Since high-frequency leads to high power consumption as mentioned in section 2, a multi-core CPU can reduce power consumption of wireless sensor nodes.

The purpose of the multi-core CPU for wireless sensor nodes corresponds to existing multi-core CPU for laptop computers. However, task model of wireless sensor nodes is different to that of laptop computers. We need to quantitatively estimate power consumption on the multi-core CPU depending on task model in wireless sensor networks.

4. Experimental Setup

To verify our concept for the multi-core CPU, we estimate power consumption according to a number of CPUs by the following method. We prototype the multi-core CPU with a multi-CPU sensor node, and we develop two benchmarks assuming two kind of application for wireless sensor networks. We run the benchmarks on the prototyped multi-CPU sensor node and measure its power consumption with changing a number of CPUs.

4.1. Circuit Prototype

The developed circuit is shown in Figure 2. The circuit block diagram of the system is also shown in Figure 3. For simplicity, we employ master-slave architecture. We can measure its power consumption from the supplied voltage and the voltage indicator.

The prototype has three PIC18 CPUs and one real time clock. The circuit allows changing frequency of each CPU statically with changing a corresponding ceramic resonator. The voltage is configurable by a stabilized power supply, but the same voltage is supplied for all the CPUs. The output pin of each CPU is connected to external interrupt pins on the other CPUs so that CPUs can invoke task execution on other CPUs.

4.2. Benchmark

To emulate the behavior of applications with the circuit prototype, we measured calculation cost of various computational tasks including sensing, several calculation processing, wireless communication module handling. Table 2

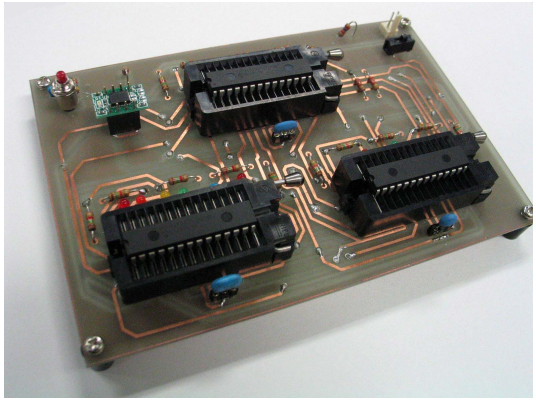

Figure 2. Multi-Core CPU Circuit Prototype

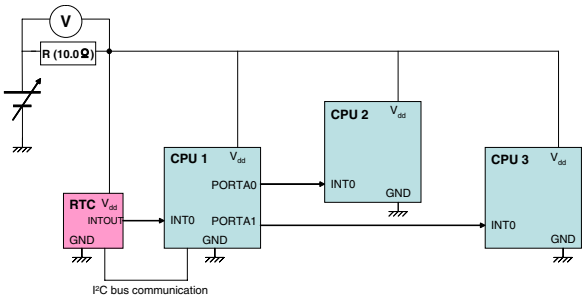

Figure 3. Circuit Block Diagram

shows the cycle, deadline and calculation cost of each measured task, and the cost will transform to computation time according to clock frequency of the CPU.

The sensing task samples acceleration of the wireless node at 100 Hz sampling rate. The RMS task calculates RMS (Root Mean Square) of sampled acceleration data. PHY (TX) is a task which sends bit-stream to a wireless communication module (TI CC1000). PHY (RX) is a task which handles bit-stream from CC1000. The CS (carrier sense) task obtains RSSI (Received Signal Strength Indicator) and estimates whether transmitting neighbor nodes exist or not. Finally, the CRC task calculates CRC value to check the correctness of received packet.

We develop two benchmarks using the measured tasks, and the benchmarks are a sampling with communication benchmark and RMS sampling benchmark. The sampling with communication benchmark assumes an earthquake monitoring [2, 6]. The earthquake monitoring needs precise time-synchronized 100 Hz sampling, and the tasks are periodically executed with strict deadline. Therefore, the sampling with communication benchmark includes four tasks: sampling, RMS, PHY (RX), and CRC.

The RMS sampling benchmark assumes human context

Table 2. Characteristics of Tasks

Task	Cycle	Deadline	Cost
Sensing	10 msec	17 usec	17 cycle
RMS	10 msec	10 msec	5957 cycle
PHY (TX)	26 usec	26 usec	39 cycle
PHY (RX)	26 usec	26 usec	39 cycle
CS	50 msec	416 usec	465 cycle
CRC	10 msec	4.2 msec	503 cycle

Table 3. Task Assignment (APP1)

CPU Number	Clock Frequency	task(s)
1 CPU	8 MHz	sensing, RMS, PHY, CRC
2 CPU	6 MHz	PHY, CRC
	4 MHz	sensing, RMS
3 CPU	6 MHz	PHY
	4 MHz	CRC
	4 MHz	sensing, RMS

recognition application [1]. The human context recognition application periodically senses acceleration and calculates its RMS value. When the calculated RMS value exceeds a threshold, the sensor node broadcasts an event packet. The event packet is relayed through sensor nodes in a multi-hop fashion. In this application, sensor nodes communicate with long-preamble MAC [5], and the preamble length is set to 50 ms. Therefore, the RMS sampling benchmark includes four tasks: sensing, RMS, CS, and PHY (TX).

5. Results

We measured power consumption on two benchmarks using the circuit prototype with changing a number of CPUs. We refer the sampling with communication benchmark as APP1, and the RMS sampling as APP2.

Table 3 and Table 4 shows the task assignment for each core. We selected the lowest frequency that meets application time restrictions on completion of tasks. The lowest frequency was determined using results of a preliminary experiment by varying a clock frequency. We selected a voltage which provides minimum power sufficient for running CPU at the determined frequency.

Figure 4 shows total current of used CPUs for three cases: single-core CPU, double-core CPU and triple-core CPU. We can see that the single-core CPU uses more power compared to other two multi-core CPUs. We can also see that the single-core CPU cannot sleep during transmission times because the PHY task is running.

Figure 5 shows the average power consumption in case

Table 4. Task Assignment (APP2)

CPU Number	Clock Frequency	task(s)
1 CPU	20 MHz	sensing, RMS, CS, PHY
2 CPU	12 MHz	CS, PHY
	4 MHz	sensing, RMS
3 CPU	6 MHz	PHY
	6 MHz	CS
	4 MHz	sensing, RMS

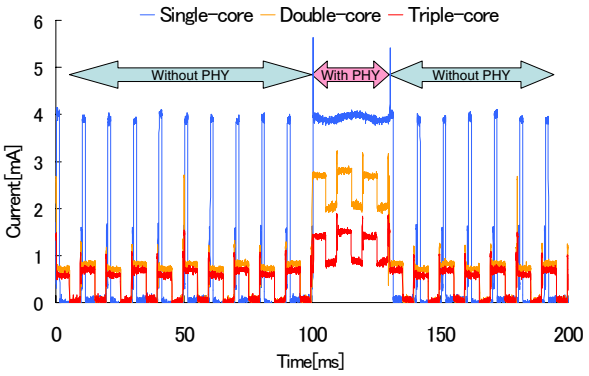

Figure 4. Current by Time Series (APP2)

of APP1, APP2 with PHY, and APP2 without PHY on single CPU, double CPUs and triple CPUs. The power consumption for APP2 with PHY was averaged between 90 ms and 140 ms in Figure 4, and that for APP2 without PHY was averaged between 40 ms and 90 ms in Figure 4. The results show that a multi-core CPU can reduce power consumption. With controlling the voltage of each individual CPU, it seems possible to reduce power consumption increasingly, since each CPU can select optimum voltage according to the sufficient frequency for running necessary tasks. In this case, applications which have large bias of processing request such as APP2 with PHY can greatly reduce power consumption. For example, the triple CPU can eliminate about 76 % of power consumption compared to the single CPU in APP2 with PHY. On the other hand, in case of low bias, the reduction is small.

6. Conclusion and Future Work

In this paper, we have described our motivation for a multi-core CPU that is reduction of power consumption and simplification of sensor networks application development. We also have showed an evaluation of power consumption with a developed multi-core CPU circuit prototype.

There are two disadvantages of multi-core CPU design. The first one is that using the larger number of cores increases the cost of a sensor node. However, we believe that extra cost is compensated by the fact that the lifetime of device is prolonged due to lower power consumption. The second one is that the multi-core CPU has communication overhead among cores. However, communication overhead can be minimized by sharing a single memory space among cores.

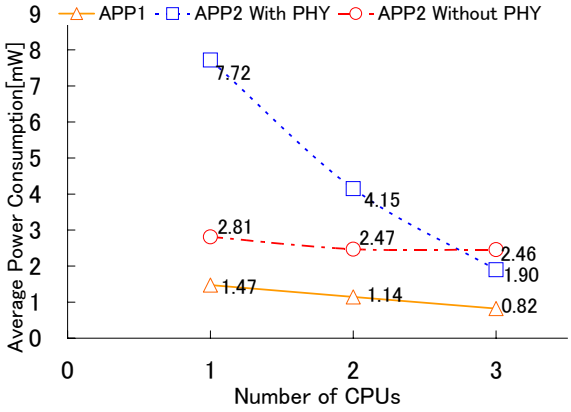

Figure 5. Power Consumption v.s. CPU Cores

In our future work, we will evaluate a multi-core CPU in terms of power consumption, cost and computational overhead. Moreover, we will design a multi-core CPU specialized for wireless sensor nodes based on these evaluations.

References

[1] Y. Kawahara, H. Kurasawa, and H. Morikawa. Recognizing user context using mobile handsets with acceleration sensors. In *Proceedings of IEEE Portable*, pages 1–5, Orland, FL, May 2007.

[2] N. Kurata, S. Saruwatari, and H. Morikawa. Ubiquitous Structure Monitoring using Wireless Sensor Networks. In *In Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems (IPACS'06)*, Tottori, Japan, December 2006.

[3] Microchip. *PIC18F2331/2431/4331/4431 Data Sheet*. http://ww1.microchip.com/downloads/en/DeviceDoc/39616C.pdf

[4] D. A. Patterson and J. L. Hennessy. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Pub, 2006.

[5] J. Polastre, J. Hill, and D. Culler. Versatile Low Power media Access for Wireless Sensor Networks. In *Proceedings of the 2nd ACM Conference on Embedded Network Sensor Systems (SenSys'04)*, Baltimore, Maryland, November 2004.

[6] M. Suzuki, N. Kurata, S. Saruwatari, and H. Morikawa. Demo abstract: A high-density earthquake monitoring system using wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'07)*, pages 373–374, 2007.