Abstract
Embedded systems serve diverse functionalities to meet the requirement for computer, communication equipment, consumer electronics, and car (4C) products. It results the need of embedded systems to exponential growth. While hundreds of thousand embedded systems run on every corner of daily life, the consumed power consumption is extremely huge. As a result, embedded systems consumed low power consumption become a significantly issue. This study presents model analysis approach to obtain low power consumption for embedded systems. First, embedded systems are divided into a set of tasks that are implemented with hardware circuits and software applications. Second, various models with same tasks combination and height of tree of embedded system are analyzed for power dissipation. Next, dynamic and static power consumption for each task is measured. For further calculating to gain an embedded system with low power consumption of embedded system can be assessed. From these evaluating designs, the lowest power consumption happens while task is performed. On the other hand, task consumes static power dissipation when its states being idle. For embedded system inside, the power consumption of embedded system is not achieved. It is necessary to get the power consumption of embedded system for each task execution, the power consumption of embedded system can be determined. The states of task categories power dissipation into dynamic and static power consumption. Dynamic power consumption happens while task is performed. On the other hand, task consumes static power dissipation when its states being idle.

Categories and Subject Descriptors
C.3 [Computer Systems Organization]: Special-Purpose and Application-Based Systems—real-time and embedded systems, microprocessor/microcomputer applications.

General Terms
Theory, Experimentation.

Keywords
Low power consumption, Power saving, Embedded system.

1. INTRODUCTION
Recently, low power consumption of embedded systems becomes significant issue owing that the energy of earth is gradually consumed. The worst affected products include computer, communications equipment, consumer electronics and car, etc. In the energy shortage era, the Intel Company predicts that 15 billion embedded products will surf to internet in year 2015. Once those products simultaneously serve, energy dissipation must be rapidly raised even used up if power saving or energy efficient improvement is not achieved. Executing task inside embedded systems consume power consumption. The states of task categories power dissipation into dynamic and static power consumption. Dynamic power consumption happens while task is performed. On the other hand, task consumes static power dissipation when its states being idle. For embedded system inside, the power consumption of embedded system can be determined. From these evaluating designs, the lowest power consumption of embedded system can be determined.

2. PRELIMINARY WORK
Smarter, smaller and portable characteristics make embedded systems to serve the functions becoming diversity. The products reside embedded systems that spread over computer, communication, consumer, and car (4C). However, the more embedded systems serve, the more power dissipation consumed. For minimizing the CPU power consumption for real time embedded system, Silva-Filho and Lima [2] first studies task execution in the power consumption of processor(s). Then, he finds the affection of optimal configuration processor(s) for energy consumption. Finally, he defines globally optimal scheduling which gains minimal energy consumption for homogeneous multiprocessor system. Silva-Filho and Lima [2] state memory hierarchy consumes power up to 50% in microprocessor system. Consequently, they propose an automated architecture exploration mechanism to NIOS II processor and memory hierarch with parameter variation. Experimental results show the reduction of energy consumption near to 27%. In 2008, Zeng et al. [3] present generalized dynamic energy performance scaling (DEPS) framework to hard real-time embedded systems for exploring application-specific energy-saving potential issue. Three energy performance tradeoff technologies called DHRC, DVFS and DPM are integrated into DEPS. Experiment results of simulation show the static DEPS improves 13.6% and 13.7% in DVFS and DHRC, respectively. Also, dynamic DEPS improves 5.7% than static DEPS. Qiu et al. [4] discuss the execution time of tasks with conditional instructions or operations problem. They adopt probabilistic random variable approach to model execution time of tasks. Then,
they propose practical algorithm VACP to minimize energy consumption for uniprocessor embedded systems. Gao et al. [5] present energy-efficient architecture for embedded software (EASES) and dynamic energy-saving method to solve energy-saving problem. The former uses a processor with dynamic voltage scaling capability, FPGA modules and extends directed acyclic graph to embedded system. The latter adopts pre-assignment to achieve dynamic runtime scheduling and minimizing energy consumption.

Real-time power information is a valuable data for software designer for battery-powered embedded systems. Genser et al. [6] propose power profiling approach to collect real-time power information at early design stages. Moreover, they present an emulation-based power profiling approach to achieve real-time power analysis for embedded systems. Because of the power information is collected at early design stages, the development efficiency and time-to-market is improved. In 2008, Elewi et al. [7] first discuss the real time scheduling of dependent tasks problem and then present enhanced multi-speed (MS) algorithm for energy saving. With energy consumption problem of battery-powered embedded system, Casares et al. [8] aim embedded smart camera to analyze the power consumption and performance. Not only graph of energy consumption but also instruction of collections is presented. They conclude the important of lightweight algorithm, the time of transfer data and transferred data type.

3. LOW POWER MODEL ANALYSIS

Modern embedded system executes sequentially a set of tasks to provide multimedia, social network and diverse applications. Performing these tasks consume a lot of power energy that depend on the deploying architecture of embedded system. In order to design embedded system with low power dissipation or power saving, minimizing power dissipation for every task is one intuitive approach.

Each task consume individually the power dissipation that affect the design becomes low power dissipation products or power saving equipment. Consequently, the candidate of tasks combination for embedded system becomes a significant issue. According to the hardware-software codesign theory, the candidate of tasks is either hardware circuits or software applications. As a result, the degree of power consumption that depends on the power dissipation with designed tasks with implementation via hardware circuits or software applications.

Tree topology is generally used to analysis the power dissipation for embedded systems. It consists of node, arc, control and data flow, height and level. Figure 1 shows two tree topologies called $M$ and $N$ of embedded system with 7 tasks. The embedded system executes task from node 1. Then, it departs for next nodes that depend on the arcs after run a period of time. The arc connect sink and destination node. It is used to exhibit the control and data flow. Another terminology in tree topology called height, $H$, of tree that is used to exhibit the architecture for embedded systems. The other terminology named level, $L$, is used to indicate the control or data flow in specific time for embedded systems. We apply tree topology to model embedded system and then further to analysis their power consumption.

From the appearance of view, Figure 1(a) and (b) are two kinds of design of topology for embedded system. Both of them are consist of 7 nodes, some arcs, control and data flow and levels. In particular, it is as same as $H$ in two tree topology. Besides, level 1 to 3 happen the control or data flow in time $t_1$, $t_2$ and $t_3$.

Embedded systems consume dynamic or static power consumption that depends on the status for task. Dynamic power consumption $D$ occurs while task is executed. On the contrary, the...
Low Power Dissipation Model Analysis for Embedded Systems
Yang-Hsin Fan, Jan-Ou Wu, San-Fu Wang

According to Figure 2(d) and Figure 3(d), we conclude topology M dissipates the same power consumption as topology N, i.e., both topology consume total power consumption as one dynamic power consumption, D, and two static power consumption, S, for each node. Such conclusion can be applied to other topology of embedded systems. As a result, we summary the sum of power consumption for embedded system in Equation (1). It consists of the levels of tree, L, a set of static power consumption \(P_{s,t_1}, P_{s,t_2}, \ldots, P_{s,t_n}\) and dynamic power consumption \(P_{d,t_1}, P_{d,t_2}, \ldots, P_{d,t_n}\) for each node.

\[
P = (L - 1) \times (P_{s,t_1} + P_{s,t_2} + \cdots + P_{s,t_n}) + (P_{d,t_1} + P_{d,t_2} + \cdots + P_{d,t_n}) \tag{1}
\]

where

- \(L\) is the height of tree,
- \(P_s\) is static power consumption,
- \(P_d\) is dynamic power consumption,
- \(t_1, t_2, \ldots, t_n\) is a set of tasks

4. EXPERIMENTAL RESULTS

This study is evaluated by adaptive pulse code modulation (ADPCM) system which is comprised of encoder and decoder modules. The encoder module consists of four tasks namely \(T_a, T_b, T_c\) and \(T_e\). On the other hand, two tasks called \(T_d\) and \(T_f\) are designed for decoder module. Each task is implemented separately to hardware and software form by using Verilog and C language. The measured data of each task includes dynamic and static power consumption with implementation of hardware circuits and software applications. Table 1 shows the measured data for \(T_a\) to \(T_f\).

![Figure 3. Power consumption model analysis for N topology.](image)

![Figure 4. Power dissipation of ADPCM embedded system with five hardware tasks and one software task.](image)

Table 1. Measured data of tasks for ADPCM system.

<table>
<thead>
<tr>
<th>Application</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADPCM system</td>
<td>Tasks</td>
</tr>
<tr>
<td>Encoder</td>
<td></td>
</tr>
<tr>
<td>(T_a)</td>
<td></td>
</tr>
<tr>
<td>(T_b)</td>
<td></td>
</tr>
<tr>
<td>(T_c)</td>
<td></td>
</tr>
<tr>
<td>(T_d)</td>
<td></td>
</tr>
<tr>
<td>(T_e)</td>
<td></td>
</tr>
<tr>
<td>(T_f)</td>
<td></td>
</tr>
</tbody>
</table>

The experiments are set to five scenarios which depend on the number of tasks with implementation by hardware circuits and software applications. These five scenarios are different to the number of hardware tasks and software tasks. The first scenario is that embedded system implement with five hardware tasks (i.e. 11111) and one software task (i.e. 0) and level as 6. Figure 4 illustrates the power dissipation of ADPCM with six designs with five hardware tasks and one software task. There are six kinds of designs that are 011111, 101111, 110111, 111011, 111101 and 111110. Experimental results display the implementation with 111101 which gains low power consumption as 6.01mW. On the contrary, the design of 110111 consumes the most power consumption as 6.055mW. The second scenario is that embedded system implement with four hardware tasks (i.e. 1111) and two software tasks (i.e. 00) and level as 6. Figure 5 displays these designs with four hardware tasks, two software tasks and level as 6. It has 15 types of combinations while develop an embedded system. The lowest power dissipation is 11.887mW which are developed via a set of tasks with 011101. On the other hand, the most power consumption of design as 101011 that consume 11.939mW. The third scenario is that embedded system implement with three hardware tasks (i.e. 111) and three software tasks and level as 6. Figure 6 demonstrates these embedded systems that comprises of three hardware and software tasks and level as 6. There are totally 20 types to complete the designs. Experimental results indicate the lowest power consumption as 17.7656mW which tasks are made of 010101. In contrast, the most power dissipation as 101010 design that consumes 17.8204mW.
The fourth scenario is that embedded system implement with two hardware tasks (i.e. 11) and four software tasks (i.e. 0000) and level as 6. Figure 7 illustrates the 15 results that consist of two hardware circuits and four software applications. Among 15 designs, the embedded system with 010100 performance gains low power consumption as 23.6468mW. On the contrary, the design of 100010 consumes the most power consumption as 23.699mW.

Finally, the fifth scenario is that embedded system implement with one hardware task (i.e. 1) and five software tasks (i.e. 00000) and level as 6. Figure 8 exhibits the outcomes which comprise of one hardware circuit and five software applications. The lowest power consumption is made up of 010000 design. It consumes power dissipation as 29.5311mW. In contrast, the design with 000010 consumes the power dissipation as 29.5756mW.

5. CONCLUSIONS

This study aims for power consumption of embedded systems to achieve the low power dissipation design. We categorize tasks into hardware or software form with corresponding to implementation by hardware circuits or software applications. Each task consumes dynamic and static power consumption that is taken into account. Then, we present various models with same tasks combination and height of tree of embedded system to gain low power consumption. Based on the proposed approach of low power model analysis, embedded system with minimizing the power consumption can be determined among diverse designs. Experimental results of ADPCM prove the effectiveness for gaining low power consumption of embedded systems that apply low power model analysis approach.

6. ACKNOWLEDGMENTS

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC 101-2221-E-143-002.

7. REFERENCES


