A New Optimal Algorithm for Energy Saving in Embedded System with Multiple Sleep Modes

Abstract:

For embedded systems with multiple sleep modes, it is interesting to understand how to maximize the energy saving potential by choosing the suitable sleep mode(s) during the idle period. In this paper, we establish a sufficient condition to narrow down the search space of sleep policy and propose a new algorithm: optimal-idle-threshold-policy-algorithm under more realistic setting than the existing works. Theoretical proofs and experimental results justify the benefits of our approach. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Enhancement of the project:

Existing System:

Irani et al. analyzed optimal offline algorithm (called lower envelope algorithm (LEA)) and their deterministic online algorithm. Optimal offline algorithm knows idle period and the arrival time of every task beforehand, then it is able to determine whether such idle period is long enough to make the energy savings from staying in sleep modes outweigh the cost of transition back to the running mode. Optimal online algorithm does not know what optimal offline algorithm knows so that it need certain thresholds to decide if transition to sleep modes incurs more cost. The behavior of optimal offline algorithm is in fact choosing the best assignment of power states because of knowing the input (tasks and idle period) in advance, and the basic idea of the deterministic online algorithm is in fact trying to mimic the behavior of the optimal offline algorithm. Based on deterministic online algorithm, those authors further presented their probability-based online algorithm that consists of two parts: 1) optimizing power based on a probability distribution and 2) dynamically adjusting certain parameters by learning the probability distribution.

Swaminathan and Chakrabarty proposed an offline algorithm, LEDES, to decide if embedded system with two modes (one is running mode and the other is sleep mode) should transition from
sleep mode to running mode to guarantee the timely completion of tasks and as the extension of LEDES, the authors further proposed another offline algorithm, MUSCLES, that targets embedded system with multiple sleep modes to decide if embedded system should transition from one sleep mode to another to respond to tasks timely. Baptiste presented an optimal offline algorithm targeting to schedule a set of unit tasks, with release dates and deadlines, on a single given embedded system to minimize the number of idle.

**Disadvantages:**

- Power consumption is high

**Proposed System:**

The algorithm proposed in this paper provides a more realistic setting for maximizing energy saving for embedded system with multiple sleep modes in idle periods. In our model, the state transition energy cost between different modes only needs to follow the triangle inequality instead of following the symmetric assumption or being additive.

Fig. 1 presents a certain embedded system with two sleep modes. We use sr, s0, s1, and s2 to denote the running mode, idle, the first sleep, and the second sleep, respectively.

Fig. 1. Transition diagram of embedded system with two sleep modes.

We have the following principles for an embedded system.
1) When there are tasks to be processed, it will stay in $s_r$.
2) When $s_r$ ends, it will select to enter $s_0$, $s_1$, or $s_2$.
3) It is able to switch its states among $s_0$, $s_1$, and $s_2$ during idle period.
4) Its state transition is wholly controllable to guarantee the timely completion of tasks to avoid delays.

![Diagram](image)

Fig. 2. Three possible scenarios when embedded system has two sleep modes. (a) Embedded system stays in the idle mode during the period of idle. (b) Embedded system stays in the first sleep mode during the period of idle. (c) Embedded system stays in the second sleep mode during the period of idle.

There are many possible ways for embedded system to switch its states during idle period. Fig. 2 shows three possible scenarios of state transition sequences for embedded system with two sleep modes (other scenarios are presented in the Appendix). For example, Fig. 2(a) means entering $s_0$ directly when $s_r$ ends and returning to $s_r$ before next $s_r$ comes.

**OITPA ALGORITHM**

We divide OITPA into offline part and online part.

**Offline Part**

Some parameters such as optimal thresholds should be obtained and then stored in the embedded system before OITPA works. All related parameters except optimal thresholds (offline-part results) may be obtained from data sheet or experimental measurement, and all the steps except storing such offline-part outcomes may be executed by some other computing tools.

**Online Part**

After obtaining the optimal thresholds in offline part and I with the help of BIP, we formally list OITPA now.

Purpose of OITPA: OITPA shows that when embedded system owns some features, the number of action sequences may be decreased by omitting the sleep policies that switching states among
different sleep modes during idle period. Most energy saving action sequences only contain simple sequences.

Analysis of Overhead and Latency

Overhead of OITPA consists of offline part and online part. As for online-part overhead, in offline part, except optimal thresholds, we can resort to data sheet to get the other related parameters. Then, we can execute the procedure in Algorithm 1 via other computing tools except storing such thresholds in the embedded system, which means the workload for embedded system only includes storing its thresholds, so offline-part overhead comes from optimal thresholds storage.

Advantages:

- Save the energy

Software implementation:

- Modelsim
- Xilinx ISE