A Rough Set Approach to Instruction-Level Power Analysis of Embedded VLIW Processors

Shu Xiao and Edmund M-K. Lai
School of Computer Engineering
Nanyang Technological University, Singapore 639798
shu_x@pmail.ntu.edu.sg, asmklai@ntu.edu.sg

Abstract

Current instruction-level power models of embedded VLIW processors handle the involved imprecision with average values. However, there are applications where it is critical to guarantee that the power consumption within any time frame will not exceed a certain limit. In this paper we proposes to use rough set based approach to handle the uncertainty of the values of the parameters in the power model. We illustrate how a rough power model can be formulated. The model is applied to a power-balanced instruction scheduling application based on a VLIW digital signal processor to illustrate the potential of this idea.

Keywords: Instruction-level power analysis, rough set, VLIW.

1 Introduction

Power models of embedded processors can broadly be classified into instruction-level and cycle-level models. While cycle-level models require knowledge of the detailed microarchitecture of the processor, instruction level models can be built using data obtained through power measurements as instructions are executed [1–4]. However there is always some degree of imprecision inherent in measurements. The variations in the measured values are usually handled by using the *mean* or *median* of a large number of measurements. Furthermore, in order to reduce the complexity of the power model, those instruction which consume similar amounts of power are typically given the same power figure [5]. The inter-instruction effects and other related factors also complicates the model for Very Long Instruction Word (VLIW) processors [4] and thus the power-aware instruction scheduling solutions. Complexity could be reduced at the cost of precision. Software power analysis using these models will therefore only reflect the average values. This may cause some problems when applied to, for instance, power-aware instruction scheduling. For example, one cannot guarantee that the optimal schedule obtained will not exceed a hard power variation limit for chip reliability. Therefore, it is important to explicitly model the imprecision in the instruction-level power model.

In this paper, we propose a rough set approach to model power consumption with the imprecision encapsulated. There are several approaches that can deal with parameter uncertainty. They include statistical techniques and fuzzy sets. Statistical techniques require knowledge of the probability distribution of each parameter. Fuzzy set techniques require an assignment of grade of membership. The main advantage of rough set is that it does not need such prior information on the data.

The rest of this paper is organized as follows. Related concepts in rough set theory are introduced in Section 2. In Section 3 we describe our approach to instruction-level VLIW power analysis with uncertainty using rough set theory. Our results of applying the proposed approach for modeling a VLIW digital signal processor are also given. This technique can be employed for power-aware VLIW instruction scheduling with uncertainty. Examples illustrate the potential of this idea in Section 4.

2 Rough Set Theory

We shall briefly introduce the major concepts of rough set theory needed to understand our rough power model formulation.

***Information table*:** The information about the real world is given in the form of an information table. The columns of an information table are labelled by attributes, rows are labelled by objects of interest and entries of the table are attribute values. Formally, an information system is at least a pair $S = (U, A)$, where U and A, are non-empty finite sets called the universe, and the set of attributes, respectively. Let $a : U- > V_a$ where V_a is the set of all values of a called the domain of a.

***Indiscernibility relation*:** Any subset B of A determines a binary relation $I(B)$ on U, which will be called an indiscernibility relation, and defined as follows:

$$x \quad I(B) \quad y \iff a(x) = a(y) \quad \forall a \in A, \forall x, y \in U \tag{1}$$

where $a(x)$ denotes the value of attribute a for element x. If x and y belongs to $I(B)$ we will say that x and y are B-indiscernible. Equivalence classes of the relation $I(B)$ are referred to as B-granules.

***Lower/upper approximations*:** The definitions for the two approximation operations on the set for an arbitrary concept X can be formulated as

$$B_*(X) = \bigcup_{x \in U} \{B(x) : B(x) \subseteq X\} \tag{2}$$

$$B^*(X) = \bigcup_{x \in U} \{B(x) : B(x) \cap X \neq 0\} \tag{3}$$

These two sets $B_*(X)$ and $B^*(X)$ are called the *B-lower* and the *B-upper* approximations of X respectively. The lower approximation consists of all objects from U which certainly belong to the concept X, employing the set of attributes B. The upper approximation contains all objects from U which possible belong to the concept X, employing the set of attributes B. Obviously, the difference between the upper and lower approximation constitute the boundary region of the vague concept.

***Accuracy of approximation*:** Rough set can be characterized numerically by the following coefficient called accuracy of approximation of the concept X by B,

$$\mu^B(X) = \frac{card(B_*(X))}{card(B^*(X))} \tag{4}$$

where *card*() denotes the number of elements (cardinality) of a finite set. Obviously, $\mu^B(X) \in [0, 1]$. The larger the value of $\mu^B(X)$, the more accurate the rough set approximation of X is.

3 Instruction-Level Model with Uncertainty

Suppose a program block W consists of N very long instructions

$$W =< w_1, ..., w_{n-1}, w_n, ..., w_N >$$

An estimation of the power consumed by the processor core during the execution of W can be obtained by [6]

$$P(W) \approx \sum_{\forall n \in N} (U(0|0) + \sum_{\forall s \in S} \sum_{\forall k \in K} v_s(w_n^k | w_{n-1}^k) + \sigma^n + \mu^n) \tag{5}$$

where $U(0|0)$ is the base power cost that represents the power consumed during an execution of a very long instruction constituted entirely by no-operation instructions (NOPs). S is the set of execution stages; the summations of $v_s(w_n^k | w_{n-1}^k)$ is the additional power contributions due to the change of instructions on the same functional unit (w_n^k represents the instruction executed on functional unit k in the very long instruction w_n); σ^n is the average power consumption due to a miss event on the data cache occurred when w_n is being executed; μ^n is the average power consumption due to a miss event on the instruction cache.

The number of instruction pairs to be considered for $v_s(w_n^k | w_{n-1}^k)$ in (5) could become too large to be characterized, since two instructions differ either in terms of functionality (i.e., opcode), addressing mode (immediate, register, indirect, etc.), or data differences (either in terms of register names or immediate values). The complexity is reduced by instruction clustering in existing work [5, 6], that is instructions are categorized into classes (clustered) beforehand, such that the instructions in a given class are characterized by very similar power cost.

Assuming that the Instruction Set Architecture (ISA) of the target VLIW processor is clustered, the vector of parameters in (5) are estimated as exact values indicating average cases in existing work [6]. In contrast with the above approach, our rough set theory based approach characterizes power parameters in (5) with their lower/upper approximations which can indicate the uncertainty involved. We shall now describe the procedures involved in building the rough power model.

3.1 Building the information table

Assuming that the ISA of the target VLIW processor is clustered, repeated power measurements for each parameter in (5) are made. For a cluster of instructions, we randomly choose instances with different opcodes, addressing modes, operands or the preceding opcodes. Principles for the design of experiments can be applied to reduce the impact of nuisance factors [7].

We shall denote the obtained information table by $(U, A \cup d)$, where $U = \{X_1, ..., x_n\}$ is a set of measurements; $A = \{a_1, ..., a_k\}$ is the condition attribute set, including attributes such as opcode, addressing mode, operands, preceding opcode and current reading; and d is decision attribute (power parameters). Let $a \in A$ be the attribute (current reading) which indicates power consumption.

3.2 Discretization

Rough set methods require discrete attributes. When the value set of any attribute in an information table is continuous values, it is likely that there will be few objects that will have the same value of the corresponding attributes.

In such a situation the number equivalence classes based on that attribute, defined as *indiscernibility*, will be large and there will be very few objects in each of such equivalence classes. This will lead to the generation of a large number of classification rules, therefore making rough set theoretic classifiers inefficient.

Our task involves a real value attribute, i.e. a (power consumption). Therefore, a preprocessing step is necessary to discretize this attribute. Discretization is a process of searching for partition of attribute domains into intervals and unifying the values over each interval. Hence discretization problem can be defined as a problem of searching for a suitable set cuts (i.e. boundary points of intervals) on attribute domains.

Let V_a be valuable domain of the attribute a (power consumption). Let $V_a = [l_a, r_a] \in R$ where R is the set of real numbers. We have the following definition according to [8].

Definition 3.1 *Any pair (a, c), where $a \in A$ and $c \in V_a$, defines a partition of V_a into two intervals. The pair (a, c) is called a cut on V_a.*

A set of cuts on $V_a : C_a = \{(a, c_1^a), (a, c_2^a), ..., (a, c_{k_a}^a)\}$ defines a partition P_a on V_a (for $a \in A$) into subintervals, i.e. $V_a = [c_0^a, c_1^a) \cup [c_1^a, c_2^a) \cup ... \cup [c_{k_a}^a, c_{k_{a+1}^a}]$, where $k_a \in N$ and $l_a = c_0^a < c_1^a < c_2^a < ... < c_{k_a}^a < c_{k_a+1}^a = r_a$. So the original $(U, A \cup d)$ is transformed, with $x_a = i$, into discreet $(U, A' \cup d)$, with $x_{a'} = [c_i^a, c_{i+1}^a)$, where $x \in U$, $a \in A$, $a' \in A'$, and $i \in [c_i^a, c_{i+1}^a)$.

The task of discretization is to search for a minimal set of cuts on attribute domains V_a that preserves the discernibility relation between measurements among U from different d (power parameters), i.e.

- *Consistency*: For any measurements $u, v \in U$, they are satisfying if u, v are discerned by A, then u, v are discerned by A';
- *Optimality*: For any C'_a satisfying consistency, it follows $card(C_a) \leq card(C'_a)$, then C_a is the optimal set of cuts.

3.3 Lower/upper approximations

For each power parameter in attribute d, compute its lower and upper approximations in the discretized information table $(U, A' \cup d)$ according to the definitions by 2 and 3 introduced in Section 2. Thus, for measurements in lower approximation of each power parameter, their values of attribute a are the union of intervals representing certain power consumption values of this power parameters. For measurements in upper approximation of each power parameter, their values of attribute a are the union of intervals representing possible power consumption values of this power parameters.

3.4 An Example: Analysis of TMS320C6711

Our target processor is TMS320C6711 [9] which is a VLIW digital signal processor. An instruction clustering process has been done beforehand. The instruction set of TMS320C6711 is coarsely partitioned into four clusters: *1)* memory access cluster, *2)* double-precision floating-point arithmetic function cluster, *3)* single-precision floating-point arithmetic function and fixed-point arithmetic function cluster and *4)* miscellaneous cluster. Instructions in the same cluster can vary in terms of operands, conditional registers, cross registers, functional units or inter-instruction effect. Therefore, parameters in (5) turn into p_1, p_2, p_3, p_4 respectively in this example.

3.4.1 Building Information Table

we randomly conducted repeated measurements for each parameter using the experimental setup as in [2, 10]. Different instances are considered in terms of opcode, operands, conditional registers, cross registers, functional units or inter-instruction effect.

3.4.2 Discretization and Lower/Upper Approximations

The builded information table is input into Rosetta [**?**,11]. Rosetta includes several important discretization algorithms. These discretization algorithms are separately executed on the obtained information table. Then the genetic algorithm in Rosetta is executed to produce the classification rules. According to these rules, the lower and upper approximations for each parameter are obtained according to the definition (2) and (3).

Table 1 shows the experimental results of comparing equal frequency binning, naive algorithm, semi-naive algorithm, entropy/MDL algorithm, and boolean reasoning algorithm on the obtained information table. We include comparisons with respect to both the effect of discretization to the simplicity of the produced classifier (measured in terms with the number of discretization split points and the number of rules) and the predictive accuracy of the classifier.

The simplest discretization algorithm, equal frequency binning, merely divides a continuous attribute into k bins where (given m objects) each bin contains m/k (possibly duplicated) adjacent values. k is a user-supplied parameter. Here in the experiments, k is set to be 6 and 12. This type of discretization is vulnerable to outliers that may drastically skew the range. Since these unsupervised algorithms do not utilize the decision attributes in setting partition boundaries, it is likely that classification information will be lost by binning as a result of combining values that are strongly associated with different decision attributes into the same bin. This is indicated by the low value of "average accuracy".

Table 1: Comparison of discretization algorithms.

discretization	no. of split points	no. of rules	avg. accuracy
equal freq. binning	5	25	0.04
equal freq. binning	11	47	0.26
naive	42	107	0.40
semi-naive	12	40	0.33
entropy/MDL	21	52	0.42
boolean reasoning	24	64	0.39

Table 2: Lower/upper current reading ranges of each power parameter.

p_1	$(\emptyset, [215, 233])$
p_2	$(\emptyset, [205, 233])$
p_3	$(\emptyset, [190, 213])$
p_4	$(\emptyset, [190, 207])$

Relatively, the supervised algorithms have higher values of "average accuracy". However, the result obtained by naive algorithm obviously includes many unnecessary split points. If both the predictive accuracy and the simplicity (measured by the number of the split point and the number rules) of the classifier are considered, semi-naive algorithm, entropy/MDL algorithm or boolean reasoning algorithm are good choices.

Theoretically, semi-naive algorithm or entropy/MDL algorithm is not expected to exhibit as good performance as boolean reasoning algorithm. In our results, they show almost same performance because our test data have only one continuous attribute (current reading), which makes the advantages of boolean reasoning algorithm not used: semi-naive algorithm or entropy/MDL algorithm considers only one condition attribute at a time; boolean reasoning algorithm considers all condition attributes simultaneously.

Based on our comparison results in Section 3.4.2, we can choose to use boolean reasoning algorithm. Then the corresponding lower/upper current reading ranges of each power parameter are obtained and shown in Table 2, which describe the imprecision of each power parameter.

4 Application

This technique can be employed for power-aware VLIW instruction scheduling with uncertainty. As discussed in introduction, the power-aware instruction scheduling techniques, using the conventional power models, can only optimize power consumption in the average sense. Severe difficulties may be caused in the application of the obtained optimal instruction schedule. On the contrary, new power-aware instruction scheduling techniques [12], employing the proposed power model, can find schedules which guarantee good power variations for all possible realizations of the power consumption parameters. A simple example is given to illustrate the potential of this idea.

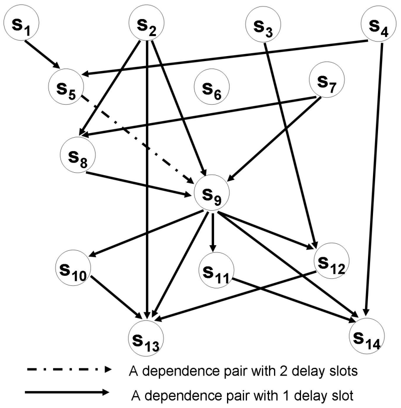
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
s11
s12
s13
s14
A dependence pair with 2 delay slots
A dependence pair with 1 delay slot

Figure 1: Data dependency graph for instructions in Example 4.1.

Example 4.1 *The target processor is TMS320C6711. The target instruction block consisting of fourteen instructions is given as* {*addaw, add, addaw, add, ldw, mv, addaw, stw, b, addaw, cmpeq, stw, ldw, b*}*. The data dependence graph is shown in Fig. 1. The task is to produce a schedule so that the power variation over the execution of the program is minimized, while the deadline constraints are met.*

In the case of the power parameters of target processor are given average values, the scheduling problem is formulated as an integer program and solved by a branch and bound algorithm as in [13], we obtain the optimal schedule $X_{MIP} = \{x_1^1, x_2^1, x_3^4, x_4^1, x_5^2, x_6^4, x_7^1, x_8^2, x_9^4, x_{10}^5, x_{11}^5, x_{12}^5, x_{13}^6, x_{14}^6\}$ *where the superscripts indicate the time slot in which the instruction is being scheduled.*

In the case of the power parameters of target processor are described as in Table 2, the scheduling problem is formulated as a rough program as proposed in [12], we obtain another schedule $X_{RP} = \{x_1^1, x_2^1, x_3^4, x_4^1, x_5^2, x_6^5, x_7^2, x_8^3, x_9^4, x_{10}^5, x_{11}^5, x_{12}^5, x_{13}^6, x_{14}^6\}$.

Rough simulations [14] of these two schedules with the *possible realization of the power parameters show*

$$P(X_{RP}, p_1, p_2, p_3, p_4)_{inf}(0.9) = 14798 \quad (6)$$

$$P(X_{MIP}, p_1, p_2, p_3, p_4)_{inf}(0.9) = 17713 \quad (7)$$

(6) means, with confidence level 0.9*, the current deviations (from the mean) of schedule* X_{RP} *are less than 14798 under all the possible realization of the power parameters in* ξ*. (7) means, with confidence level* 0.9*, the current deviations (from the mean) of schedule* X_{MIP} *are only less than 17713 under all the possible realization of the power parameters in* ξ*. We can see that the schedule obtained by integer programming is far from a globally optimal one if considering the possible realization of the power parameters.*

The optimal schedules obtained by mixed-integer programming often have larger deviations from the optimal objective function values due to the ignorance of imprecision accumulation. However, the rough programming approach takes parameter imprecision into account in a natural way.

5 Conclusion

A rough set theory based approach has been proposed to the problem of instruction-level VLIW power modelling with uncertainties. The involved imprecision in power parameters are described with their lower/upper approximations. Experiments have been applied on TI TMS320C6711. This technique can be employed for power-aware VLIW instruction scheduling with uncertainty. The main benefit is that the computed schedule is globally optimal with the possible realization of the power parameters, not only optimal in the average cases. Examples illustrate the potential of this idea.

References

[1] V. Tiwari, S. Malik, and A. Wolfe, “Power analysis of embedded software a first step toward software power minimization,” *IEEE Trans. on Very Large Scale Integration Systems*, vol. 2, no. 4, pp. 437–445, Dec. 1994.

[2] J. T. Russell and M. Jacone, “Software power estimation and optimisation for high performance, 32-bit embedded processors,” in *Proc. Int. Conf. on Computer Design*, Oct. 1998, pp. 328–333.

[3] C. Gebotys, “Power minimization derived from architectural-usage of VLIW processors,” in *Proc. Design Automation Conf.*, Los Angeles, USA, 2000, pp. 308–311.

[4] N. Julien, J. Laurent, E. Senn, and E. Martin, “Power consumption modeling and characterization of the TI C6201,” *IEEE Micro*, vol. 23, no. 5, pp. 40–49, Sep.-Oct. 2003.

[5] A. Bona, M. Sami, D. Sciutos, C. Silvano, V. Zaccaria, and R.Zafalon, “Energy estimation and optimization of embedded VLIW processors based on instruction clustering,” in *Proc. Design Automation Conf.*, vol. 39, New Orleans, USA, 2002, pp. 886–891.

[6] M. Sami, D. Sciuto, C. Silvano, and V. Zaccaria, “An instruction-level energy model for embedded VLIW architectures,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 9, pp. 998–1010, Sep. 2002.

[7] D. C. Montgomery, *Design and analysis of experiments.* New York: John Wiley, 2001.

[8] H. S. Nguyen and A. Skowron, “Quantization of real value attributes,” in *Proc. of Second Joint Annual Conf. on Information Science*, Wrightsville Beach, North Carolina, 1995, pp. 34–37.

[9] *TMS320C621x/C671x DSP Two-Level Internal Memory Reference Guide*, Texas Instruments, Aug. 2002, application Report, SPRU609.

[10] S. Nikolaidis, N. kavvadias, and T. Laopoulos, “Instruction-level power measurement methodology,” *Technical Report, EASY project funded by the European Union*, 2002.

[11] J. Komorowski, A. Skowron, and A. Øhrn, “The rosetta toolkit,” in *Handbook of Data Mining and Knowledge Discovery*, W. Kl Ed. Oxford University Press, 2000.

[12] S. Xiao and E. M.-K. Lai, “A rough programming approach to power-aware vliw instruction scheduling for digital signal processors,” in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, Mar. 2005.

[13] ——, “A branch and bound algorithm for power-aware instruction scheduling of VLIW architecture,” in *Workshop on Compilers and Tools for Constrained Embedded Systems*, Washington DC, USA, Sep. 2004.

[14] B. Liu, *Theory and practice of uncertain programming.* Heidelberg: Physica-Verlag, 2002.

Shu Xiao obtained her B.E.and M.S. degrees from Wuhan University, China in 1998 and 2001 respectively. She is currently a PhD student in the School of Computer Engineering, Nanyang Technological University, Singapore. Her research interests are in power-aware VLIW instruction scheduling.

Edmund M-K. Lai obtained his B.E.(Hons) and PhD degrees from the University of Western Australia in 1982 and 1991 respectively. He is currently an associate professor in the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests are in digital signal processing and information theory.