A ROUGH PROGRAMMING APPROACH TO POWER-AWARE VLIW INSTRUCTION SCHEDULING FOR DIGITAL SIGNAL PROCESSORS

Shu Xiao and Edmund M-K. Lai

School of Computer Engineering, Nanyang Technological University, Singapore 639798

ABSTRACT

Current techniques for power-aware VLIW instruction scheduling assumed that the power consumption parameters are precisely known. In reality, there will always be some degree of imprecision. In this paper, we propose to apply rough set theory to handle the imprecision involved. Power consumption parameters are modelled as rough variables and the power-balanced instruction scheduling problem is formulated as a rough program. The effectiveness and advantages of our approach is illustrated through examples.

1. INTRODUCTION

Advanced digital signal processors employ VLIW architectures for demanding signal processing applications. Each long instruction word consists of one or more instructions that can be executed in parallel on different functional units. These processors rely on the compiler to schedule instruction at compile time to meet deadline as well as power constraints. Average power consumption reduction is known to be an important constraint for its great impact on battery life and heat dissipation. Significant processor supply current variations cause power supply noise, degrade chip reliability and accelerate battery exhaustion. Hence power variation reduction without compromising execution speed is another important instruction scheduling constraint in embedded VLIW systems.

Power-aware instruction scheduling refers to the task of producing a schedule of these parallel instructions so that the average power consumption is minimized or the power variation over the execution of the program is minimized, while the deadline constraints are met. Previously published works in this area make use of power consumption models with parameters that are assumed to be precisely known [1–4]. However, in reality, the values of these parameters are not precise for two main reasons. Firstly, physical measurements, which has been an important approach to instruction-level power modelling and estimation for microprocessors [5–8], are always imprecise. The variations in the measured values are using handled by using the *mean* or *median* of a large number of measurements. Secondly, in order to reduce the complexity of the power model, those instruction with consume similar amounts of power are typically clustered together and given a the same power figure [9].

While these approximations with the average values allow us to optimize power consumption in the average sense, they are not enough in some important occasions. For example, the optimal schedule obtained with the average values fails to guarantee that a hard power variation limit for chip reliability is exceeded or not in the actually occurrences. Therefore, it is more desirable to formulate the optimization problem with the power parameters also modelling their deviations from the average. As a result, the obtained optimal schedule is expected to have the best power variations even considering its deviations from the average that may actually occur.

There are several approaches to deal with imprecision or uncertainty. In this paper, we propose to use the rough set theory [10] approach to model the uncertainty inherent in the power model parameters. The instruction scheduling problem can then be formulated as a rough program [11]. One of the main advantages of rough set is that it does not need any prior information on the data, such as probability distributions in statistics, basic probability assignment in the Dempster-Shafer theory [12], or grade of membership in fuzzy set theory [13].

This paper focuses on the optimization problem of VLIW instruction scheduling for power variation reduction and is an extension of our previous work [4]. The rest of the paper is organized as follows. The next two sections summarize the conventional solutions. Section 2 introduces a simple power model for VLIW architectures. The mixed integer programming formulation for this optimization problem is described in Section 3. Section 4 presents the method for modelling the power consumption parameters as rough variables based on measurements. Section 5 proposes a rough programming formulation for the optimization problem of VLIW instruction scheduling for balanced power consumption. Throughout this paper, we assume that an initial instruction schedule that meets the speed performance requirements has been obtained. Our algorithm reschedules instructions for power variation optimization in the second phase.

2. POWER MODEL

A VLIW processor with an issue width of k can execute at most k instructions simultaneously on separate functional units. Each instruction requires a different amount of time to execute. We divide the time line into equal length time slots. A power cost p_i is associated with each instruction i which represents the average power consumed by this instruction over the instruction execution.

Let the instruction schedule be $N =< N_1, N_2, ..., N_t >$ where $N_i = (n_1, n_2, ..., n_k)$ is the very long instruction word issued at the *i*-th time slot of N. The power consumption at the *i*-th time slot of schedule N is the sum of the power consumed by all the executing instructions, issued either at the *i*-th time slot or previous ones. This can be expressed mathematically as

$$P^i = \sum_{1 \leq k \leq i} \sum_{n_j \in N_{(i-k+1)}} p_{n_j} \tag{1}$$

where n_j is an executing instruction at the *i*-th time slot. Thus the average power consumption over all t time slots is given by

$$M = \left(\sum_{i=1}^{t} P^i \right) / t \quad (2)$$

The power deviation from the average value at any given time slot i is

$$PV^i = |P^i - M| \quad (3)$$

Therefore, the total power deviation for a schedule is given by

$$PV = \sum_{i=1}^{t} PV^i \quad (4)$$

This is a rather simplified power model for the VLIW instructions. However, our techniques do not depend on a particular power model. It can be easily be modified to work with more sophisticated power models if they are available.

3. MIXED-INTEGER PROGRAM FORMULATION

The conventional mixed-integer program for VLIW instruction scheduling for minimal power variations is given by **P1** and efficient techniques have been proposed to solve it [4].

P1: $\min \ PV(X, \xi)$

subject to

$$\begin{array}{ll} X = \cup x_i^k & i = 1, ..., n; k = 1, ..., t \\ x_i^k \in \{0, 1\} & i = 1, ..., n; k = 1, ..., t \end{array} \quad (5)$$

$$\begin{array}{l} G(X) \leq 0 \\ L(X) = 0 \end{array} \quad (6)$$

where $PV(X, \xi)$ is the power variation of a given schedule X over time which we seek to minimize, and ξ denotes the set of the power consumption parameters. In (5), n is the number of instructions in X and t is the number of time slots available. The binary decision variables x_i^k has a value of 1 if instruction i is rescheduled in time slot k; otherwise its value is zero. $G(X) \leq 0$ and $L(X) = 0$ in (6) denote the constraint matrix for processor-specific resource constraints, data dependence constraints and performance deadline constraints.

The basic problem with the above formulation is that the power consumption parameters ξ in the objective function need to be precise values.

4. POWER MODEL WITH POWER CONSUMPTION PARAMETERS REPRESENTED AS ROUGH VARIABLES

The imprecision of the power consumption parameters can be encapsulated by expressing them as rough variables. Based on [11] we shall define rough space and rough variables.

Definition 4.1 *Let Λ be a nonempty set, $\overline{A}$ be a σ-algebra of subsets of Λ, Δ be an element in $\overline{A}$, and π be a set function satisfying the following axioms:*

1. *$\pi\{\mathrm{A}\} \geq 0$ for any $A \in \overline{A}$.*
2. *For every countable sequence of mutually disjoint events $\{\mathrm{A}_i\}_{i=1}^{\infty}$, we have $\pi\{\bigcup_{i=1}^{\infty} A_i\} = \sum_{i=1}^{\infty} \pi\{A_i\}$.*

Then $(\Lambda, \Delta, \overline{A}, \pi)$ is called a rough space.

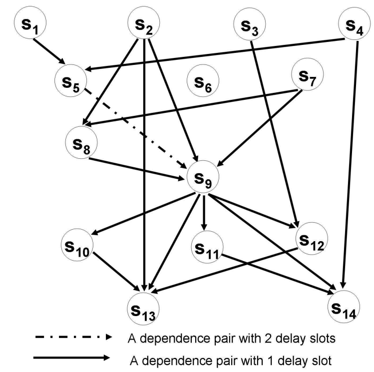

Fig. 1. Data dependency graph for instructions in Example 4.2

Definition 4.2 *A rough variable ζ on the rough space $(\Lambda, \Delta, \overline{A}, \pi)$ is a function from Λ to the real line $\Re$ such that for every Borel set O of $\Re$, we have $\{\lambda \in \Lambda | \xi(\lambda) \in O\} \in \overline{A}$. The lower and the upper approximations of the rough variable ζ are then defined as $\underline{\zeta} = \{\zeta(\lambda) | \lambda \in \Delta\}$ and $\overline{\zeta} = \{\zeta(\lambda) | \lambda \in \Lambda\}$ respectively.*

Example 4.1 *Suppose a rough space $(\Lambda, \Delta, \overline{A}, \pi)$ where $\Lambda = \{x | c \leq x \leq d\}$, $\Delta = \{x | a \leq x \leq b\}$, with $c \leq a \leq b \leq d$. Then the function $\zeta(x) = x$ for all $x \in \Lambda$ is a rough variable, also expressed as $([a, b], [c, d])$, where $[a, b]$ is its lower approximation and $[c, d]$ is its upper approximation. This means that the values within $[a, b]$ are sure and those within $[c, d]$ are possible.*

The requirement is to encapsulate the deviation range of the power consumption parameters by expressing them as rough variables. So the rough variable expression $([a, b], [c, d])$ as in Example 4.1 can be an appropriate way. A power consumption parameter p_i (see Section 2) can be expressed as a rough variable $([a, b], [c, d])$ with $c \leq a \leq b \leq d$ on the real line where $[a, b]$ is its lower approximation and $[c, d]$ is its upper approximation. Then, based on rough set theory [10] we will give our approach to determine values of a, b, c and d for each $p_i \in \xi$.

1) Building information tables: An information table is 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. Conduct repeated measurements for each $p_i \in \xi$ and collect the data. Principles for design of experiments can be applied to reduce the impact of nuisance factors [14].

*2) Discretization:*In rough set theory, when the value set of any attribute in an information table is continuous values or real numbers, 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. One solution to this problem is discretization. Nguyen proposed the named discretization approach based on rough set methods and Boonlean reasoning [15, 16]

3) Lower/upper approximations: For each $p_i \in \xi$, compute the lower and upper approximations according to their definitions in Example 4.1.

Table 1. Current readings(mA) of 50 measurements for p_{addaw}.

206	208	203	197	208	210	204	202	203
191	191	204	203	201	194	212	211	199
205	194	210	209	198	212	196	197	194
197	203	201	212	207	200	203	205	203
203	200	190	196	206	196	206	205	204
205	196	198	195	202				

Table 2. Discretized current readings of measurements for p_{addaw}

[203,207)	[207,214)	[203,217)	[190,198)	[207,214)
[207,214)	[203,207)	[202,203)	[203,207)	[190,198)
[190,198)	[203,207)	[203,207)	[198,202)	[190,198)
[207,214)	[207,214)	[198,202)	[203,207)	[190,198)
[207,214)	[207,214)	[198,202)	[207,214)	[190,198)
[190,198)	[190,198)	[190,198)	[203,207)	[198,202)
[207,214)	[207,214)	[198,202)	[203,207)	[203,207)
[203,207)	[203,207)	[198,202)	[190,198)	[190,198)
[203,207)	[190,198)	[203,207)	[203,207)	[203,207)
[203,207)	[190,198)	[198,202)	[190,198)	[202,203)

Example 4.2 *Consider the TMS320C6711 [17] which is a VLIW digital signal processor. An initial performance optimized instruction schedule X_1 consisting of fourteen instructions is given by*

$$X_1 = \{x_1^1, x_2^1, x_3^1, x_4^1, x_5^2, x_6^2, x_7^2, x_8^3, x_9^4, x_{10}^5, x_{11}^5, x_{12}^5, x_{13}^6, x_{14}^6\}$$

where the superscripts indicate the time slot in which the instruction is being scheduled. These fourteen instructions are { addaw, add, addaw, add, ldw, mv, addaw, stw, b, addaw, cmpeq, stw, ldw, b }. The data dependence graph as shown in Fig. 1.

To represent the set of power consumption parameters $\xi = \{p_{addaw}, p_{add}, p_{ldw}, p_{mv}, p_{stw}, p_b, p_{cmpeq}\}$ as rough variables, we randomly conducted fifty repeated measurements for each parameter. Table 1 shows the data set for power parameter p_{addaw} consisting of 50 repeated measurements.

Based the measured data, the possible power consumption values on real line are discretized using the Boonlean reasoning algorithm. The partial discretization results corresponding to Table 1 are shown in Table 2.

After discretization, the lower and upper approximations for each power consumption parameter are generated using the Rosetta Toolkit [18]. The categorization rules are shown in Table 3. According to these categorization rules, the lower and upper approximations for each parameter are obtained. They are shown in Table 4.

5. ROUGH PROGRAMMING FORMULATION

If ξ is a set of rough variables, then the values of the function $f(X,\xi)$ for any given X are also rough variables. The rough returns of $f(X,\xi)$ may be ranked by *1)* the expected value $E[f(X,\xi)]$; *2)* the α-optimistic value $f(X,\xi)_{sup}(\alpha)$ or the α-pessimistic value $f(X,\xi)_{inf}(\alpha)$, for some predetermined confidence level $\alpha \in (0,1]$; *3)* the trust measure $Tr\{f(X,\xi) > \overline{r}\}$ for some predetermined level $\overline{r}$. In order to indicate the deviation, we may measure the rough return $f(X,\xi)$ for any decision X by its α-pessimistic value.

Table 4. Rough power consumption parameters in Example 4.2.

p_{addaw},p_{add},p_{mv},p_{cmpeq}	p_{ldw},p_{stw}	p_b
$(\emptyset, [190, 214])$	$(\emptyset, [214, 233])$	$(\emptyset, [190, 207])$

Definition 5.1 *let ϑ be a rough variable, and $\alpha \in (0,1]$. Then*

$$\vartheta_{inf}(\alpha) = inf\{r | Tr\{\vartheta \le r\} \ge \alpha\} \quad (7)$$

is called the α-pessimistic value to ϑ, where Tr is the trust measure operator.

We now give the rough programming formulation **P2** of the VLIW power-balanced instruction scheduling problem.

P2: $\min \ PV(X,\xi)_{inf}(\alpha)$

subject to

$$PV(X,\xi)_{inf}(\alpha) = inf\{\overline{PV} | Tr\{PV(X,\xi) \le \overline{PV}\} \ge \alpha\} \quad (8)$$

$$\begin{array}{ll} X = \cup x_i^k & i = 1,...,n; k = 1,...,t \\ x_i^k \in \{0,1\} & i = 1,...,n; k = 1,...,t \end{array} \quad (9)$$

$$\begin{array}{l} G(X) \le 0 \\ L(X) = 0 \end{array} \quad (10)$$

where α is the specified confidence level and ξ is the set of rough power consumption parameters. (9) and (10) are the same as those in the conventional formulation since there are no rough variables involved. $PV(X,\xi)_{inf}(\alpha)$ is the smallest value $\overline{PV}$ satisfying $Tr\{PV(X,\xi) \le \overline{PV}\} \ge \alpha$. This means that, for a given X, the rough return of $PV(X,\xi)$ will be below the pessimistic value $\overline{PV}$ with a confidence level of α. Solving this program involves searching for the minimum α-pessimistic value $PV(X,\xi)_{inf}(\alpha)$ among all feasible schedules X.

Example 5.1 *Continuing from Example 4.2, suppose the confidence level is $\alpha = 0.9$. We have the following rough programming model for the scheduling problem:*

$$\min \ PV(X,\xi)_{inf}(0.9)$$

subject to

$$PV(X,\xi)_{inf}(0.9) = inf\{\overline{PV} | Tr\{PV(X,\xi) \le \overline{PV}\} \ge 0.9\} \quad (11)$$

$$PV(X,\xi) = \sum_{k=1}^{6} |P^k - M|$$
$$M = \left(\sum_{k=1}^{6} P^k\right)/6 \quad (12)$$
$$P^k = \sum_{i=1}^{14} x_i^k p_i + \sum_{i=1}^{14} x_i^{k-1} \varepsilon(D_i - 1) p_i$$

$$\xi = \cup p_i \quad i = 1,...,14 \quad (13)$$

$$\begin{array}{ll} X = \cup x_i^k & i = 1,...,14; k = 1,...,6 \\ x_i^k \in \{0,1\} & i = 1,...,14; k = 1,...,6 \end{array} \quad (14)$$

$$\begin{array}{l} G(X) \le 0 \\ L(X) = 0 \end{array} \quad (15)$$

The power variation of a given schedule X is computed in (12) according to the power model described in Section 2. The rough power consumption parameters in (13) are given in Table 4. $G(X) \le 0$ and $L(X) = 0$ in (15) denote the constraint matrix for processor-specific resource constraints, data dependence constraints and performance deadline constraints respectively.

Table 3. Categorization rules for each power consumption parameter in Example 4.2.

current([203, 207)) => parameter(p_{add}) OR parameter(p_b) OR parameter(p_{addaw}) OR parameter(p_{mv}) OR parameter(p_{cmpeq})
current([207, 214)) => parameter(p_{add}) OR parameter(p_{addaw}) OR parameter(p_{mv}) OR parameter(p_{cmpeq})
current([190, 198)) => parameter(p_{add}) OR parameter(p_b) OR parameter(p_{addaw}) OR parameter(p_{mv}) OR parameter(p_{cmpeq})
current([202, 203)) => parameter(p_{add}) OR parameter(p_b) OR parameter(p_{addaw}) OR parameter(p_{mv}) OR parameter(p_{cmpeq})
current([198, 202)) => parameter(p_{add}) OR parameter(p_b) OR parameter(p_{addaw}) OR parameter(p_{mv}) OR parameter(p_{cmpeq})
current([214, 234)) => parameter(p_{ldw}) OR parameter(p_{stw})

We use a hybrid intelligent algorithm [11] to solve the rough program. The optimal schedule obtained is

$$X_{op} = \{x_1^1, x_2^1, x_3^4, x_4^1, x_5^2, x_6^4, x_7^1, x_8^3, x_9^4, x_{10}^5, x_{11}^5, x_{12}^5, x_{13}^6, x_{14}^6\}$$

The objective function $PV(X, \xi)$ has an optimal 0.9-pessimistic value of 127. That is,

$$inf\{\overline{PV}|Tr\{PV(Xop, \xi) \leq \overline{PV}\} \geq 0.9\} = 127 \qquad (16)$$

According to (16), the obtained optimal schedule X_{op} has the best power variations even considering its deviations from the average that may actually occur, which is less than or equal to 127 with a confidence level above 0.9. On the contrary, the optimal schedule obtained using the mixed-integer formulation in Section 3 only has the best power variation in the average sense, without considering the deviations that actually occur. The advantages of rough programming approach are obvious in occasions such as a hard power variation limit is required to be guaranteed.

6. CONCLUSIONS

Rough set theory has been applied to the problem of power-balanced VLIW instruction scheduling. We showed how the ideas from rough set theory can be used to model the imprecise power consumption parameters as rough variables and formulated the scheduling problem as a rough program. The advantages and effectiveness of our approach has been demonstrated through examples. This work is a first attempt to apply rough set theory to this area. Future work involves the development of more efficient algorithms to solve the rough programming model by exploiting the problem specific structure.

7. REFERENCES

[1] M. T. Lee, V. Tiwari, S. Malik, and M. Fujita, "Power analysis and minimization techniques for embedded DSP software," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 5, no. 1, pp. 123–135, Mar. 1997.

[2] H. Yun and J. Kim, "Power-aware modulo scheduling for high-performance VLIW processors," in *Proc. Int. Symp. on Low Power Electronics and Design*, Huntington Beach, California, USA., Aug. 2001, pp. 40–45.

[3] C. Lee, J. K. Lee, T. T. Hwang, and S. C. Tsai, "Compiler optimization on VLIW instruction scheduling for low power," *ACM Trans. on Design Automation of Electronic Systems*, vol. 8, no. 2, pp. 252–268, Apr. 2003.

[4] S. Xiao and E. M-K. Lai, "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, Sept. 2004.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.*, New Orleans, USA, 2002, vol. 39, pp. 886–891.

[10] Z. Pawlak, *Rough Sets: theoretical aspects of reasoning about data*, Kluwer Academic Publisher, Boston, MA, 1991.

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

[12] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University press, Princeton, 1976.

[13] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. 3, pp. 28–44, Jan. 1973.

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

[15] 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.

[16] H. S. Nguyen, "Discretization problem for rough sets methods," in *Proc. of the Int. Conf. on Rough Sets and Current Trends in Computing*, Warsaw, Poland, Jun. 1998, pp. 545–552.

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

[18] 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.