

An ILP Approach to the Simultaneous Application of Operation Scheduling and Power Management

Shih-Hsu Huang, Chun-Hua Cheng, Chung-Hsin Chiang, and Chia-Ming Chang
Department of Electronic Engineering,
Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.

Abstract

At the behavioral level, large power saving is possible by shutting down unused operations, which is commonly referred to as power management. However, on the other hand, operation scheduling has a significant impact on the potential for power saving via power management. In this paper, we present an integer linear programming (ILP) model for the simultaneous application of operation scheduling and power management in high level synthesis. Our objective is to maximize the power saving under both the timing constraints and the resource constraints. Compared with previous work, experimental data consistently show that our approach has significant improvement in the power saving.

Keywords: Average Power, Power Management, Integer Linear Programming, High-Level Synthesis, and Scheduling.

1. INTRODUCTION

A behavioral description can be represented by a control-data flow graph (CDFG), where each node corresponds to an operation, and each directed edge corresponds to data dependency or control relation. Under specified design constraints (timing and resource), operation scheduling [1-6] is to assign each operation in the CDFG to a specific control step to start its execution. If there is no power management, all the operations in the CDFG will always be executed under all the conditionals. However, in fact, the outputs of some operations are not used under some conditionals; thus, not all operations are necessarily executed under all the conditionals.

However, we cannot shut down an operation, unless we can identify the output of this operation is unused. In other words, to enable the power management of an operation, all the operations involved in identifying the control/data flow of this operation must be scheduled at least one control step before this operation. Therefore, operation scheduling has a significant impact on the potential for power saving via power management. As a result, in order to maximize the power saving, it is necessary to take the power management into account during the stage of operation scheduling.

Monteiro, Devadas, Ashar, and Mauskar [7] proposed the first heuristic algorithm to consider power management during the stage of operation scheduling. However, their approach ignores the active probability and computation complexity of operations, which can significantly affect the power. Thus, Chen and Sarrafzadeh [8] proposed a heuristic algorithm to improve the drawback of [7]. In their approach [8], an operation with higher potential power saving has higher priority to be shut down (i.e., all the operations involved in identifying the control/data flow of this operation have higher priority to be scheduled earlier).

In the paper, we present an integer linear programming (ILP) model for the simultaneous application of operation scheduling and power management. Our objective is to maximize the power saving under the design constraints (timing and resource).

2. MOTIVATION

The input is a CDFG, where each node corresponds to an operation, and each directed edge corresponds to a dependency constraint (data dependency or control dependency). All the conditionals in the design (CDFG) are represented by comparison nodes and multiplexer nodes, and a directed edge from comparison node to multiplexer node corresponds to a control dependency. In the following, we use the CDFG shown in Figure 1 as an example.

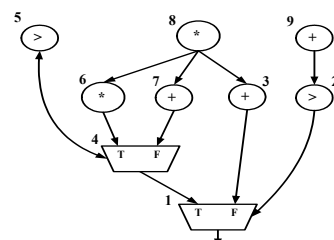


Figure 1: A CDFG.

This CDFG has nine operations. Operations o_2 and o_5 are comparison nodes. Operations o_1 and o_4 are multiplexer nodes. The directed edge from operation o_2 to operation o_1 and the directed edge from operation o_5 to operation o_4 correspond to control dependencies.

If there is no power management, we need to execute all the operations in the CDFG. Let's use the CDFG shown in Figure 1 as an example. Suppose that the power

consumptions of adder (for the execution of addition operations), multiplexer (for the execution of multiplexing operations), comparator (for the execution of comparison operations), and multiplier (for the execution of multiplication operations) are 3, 1, 4, and 20, respectively. If there is no power management, the power consumption is $3*3 + 2*1 + 2*4 + 2*20 = 59$.

However, in fact, the outputs of some operations are not used under some conditionals. Using the CDFG shown in Figure 1 as an example, operation o_7 need not to be activated, if the output of comparison operation o_2 is false or the output of comparison operation o_5 is true. Suppose that, for each multiplexer node, the probability of taking its truth input (T) part is 50 %, and the probability of its false input (F) part is also 50 % (note that the probabilities can be estimated through behavior-level simulation). If operation o_7 can be shut down according to the output of comparison operation o_2 , the power saving is $50\%*3$; if operation o_7 can be shut down according to the output of comparison operation o_5 , the power saving is $50\%*3$. If operation o_7 can be shut down according to both the output of comparison operation o_2 and the output of comparison operation o_5 , the power saving is $50\%*3+50\%*3-50\%*50\%*3$, in which the term $50\%*50\%$ denotes the probability of the condition that the output of comparison operation o_2 is false and the output of comparison operation o_5 is true.

For the convenience of presentation, we use dotted line to represent these added extra directed edges. In [8], these added extra directed edges are referred to as *soft edges*.

Obviously, inserting soft edges reduces the solution space of operation scheduling. Thus, inserting a soft edge is not always possible; i.e., we cannot add a soft edge if the design constraints (timing and resource) are violated. In this paper, we integrate power management (i.e., inserting soft edges) into the operation scheduling stage. Our objective is to maximize the power saving under the design constraints (timing and resource). Let's use the CDFG shown in Figure 1 as an example. Assume that the delay of each operation is 1 control step, the timing constraint is four control steps and the resource constraints are one adder, one multiplier, and one comparator. Following the same assumption in [7,8], there is no constraint on the number of multiplexers; in other words, the number of multiplexers is not minimized until the resource allocation stage. Figure 2 gives our scheduled CDFG in which the power saving is maximized under the given design constraints. Compared with the original CDFG as shown in Figure 1, three soft edges are added: a soft edge is added from operation o_5 to operation o_6 , a soft edge is added from operation o_5 to operation o_7 , and a soft edge is added from operation o_2 to operation o_3 . The power saving of the soft edge from operation o_5 to operation o_6 is $50\%*20$, the power saving of the soft edge from operation o_3 to operation o_7 is $50\%*3$, and the power saving of the soft edge from operation o_2 to operation o_3 is $50\%*3$. Assume that the extra power consumption caused by a soft edge is 1. Therefore, the extra power consumption due to

the insertion of three soft edges is 3. As a result, compared with the original CDFG, the total power saving is 10 (i.e., $10+1.5+1.5-3=10$). The power consumption of this scheduled CDFG is 49 (i.e., $59-10=49$)

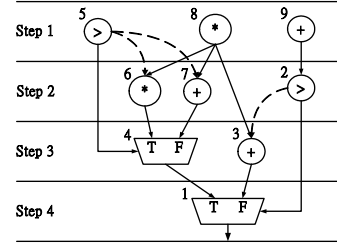


Figure 2: A scheduled CDFG in which the power saving is maximized under design constraints.

3. ILP MODEL

The notations used in our ILP model are as below.

- (1) The notation n denotes the number of operations.
- (2) The notation t denotes the number of control steps.
- (3) The delay of each operation o_i is D_i clock cycles.
- (4) The notation x_{ij} denotes a binary variable (i.e., a 0-1 integer variable). Binary variable $x_{ij} = 1$, if and only if operation o_i is scheduled into control step j ; otherwise, binary variable $x_{ij} = 0$.
- (5) The value E_i denotes the earliest possible control step of operation o_i . Note that we can use the ASAP calculation [3] to determine the value E_i for each operation o_i .
- (6) The value L_i denotes the latest possible control step of operation o_i . Note that, given the total number of control steps, we can use the ALAP calculation [3] to determine the value L_i for each operation o_i .
- (7) The value M_k is the number of functional units of type k .
- (8) The set C_i includes all the comparison operations that may shut down operation o_i .
- (9) The notation $|A|$ represents the number of elements in the set A .
- (10) The notation W_i denotes the power consumption of operation o_i .
- (11) The notation W_{soft} denotes the power consumption caused by a soft edge.
- (12) The notation $Y_{A,i}$ is a binary variable to model the insertion of soft edges. We have $Y_{A,i} = 1$, if and only if soft edges are inserted from all comparison operations in the set A to operation o_i . In other words, if $Y_{A,i} = 1$, all the comparison operations in the set A must be executed before operation o_i ; otherwise, $Y_{A,i} = 0$.
- (13) $P_{A,i}$ represents the probability that operation o_i can be shut down by the comparison operations in the set A .

Our optimization goal is to maximize the power saving. Therefore, the objective function is

$$\sum_{i=1}^n \sum_{A \subseteq C_i} [(-1)^{|A|+1} P_{A,i} \cdot Y_{A,i} \cdot W_i] - \sum_{i=1}^n \sum_{o_j \in C_i} Y_{\{o_j\},i} \cdot W_{soft} \quad (\text{Formula 1})$$

Every operation must be scheduled to a control step. Therefore, for each operation o_i , we have the following constraint:

$$\sum_{j=E_i}^{L_i} x_{i,j} = 1 \quad (\text{Formula 2})$$

The dependency constraints in the CDFG must be preserved. Therefore, for each dependency constraint $o_i \rightarrow o_l$ in the CDFG, we have the following constraint:

$$\sum_{j=E_i}^{L_i} (j + D_i - 1) \cdot x_{i,j} < \sum_{j=E_l}^{L_l} j \cdot x_{l,j} \quad (\text{Formula 3})$$

The number of resources, type k , used in any control step should be less than or equal to the allocated resources M_k . Therefore, for each control step c and each type of function unit FU_k , we have the following constraint:

$$\sum_{o_i \in FU_k} \sum_{E_i + D_i - 1 \geq j} x_{i,j} \leq M_k \quad (\text{Formula 4})$$

If a soft edge is added, an extra dependency constraint is enforced. Therefore, for each comparison operation o_l that may shut down operation o_i , we have the following constraint:

$$\sum_{j=E_i}^{L_i} (j + D_i - 1) \cdot x_{i,j} < \sum_{j=E_l}^{L_l} j \cdot x_{l,j} + (1 - Y_{\{i\},l}) \cdot t \quad (\text{Formula 5})$$

The binary variable $Y_{A,l} = I$, if for each comparison operation in the set $A \subseteq C_l$, there is a soft edge from it to operation o_l . Therefore, for each operation o_i , we have the following constraint:

$$\sum_{o_i \in A} Y_{\{i\},l} \leq Y_{A,l} + |A| - 1 \quad (\text{Formula 6})$$

The binary variable $Y_{A,l} = 0$, if and only if there exists a comparison operation o_i in the set $A \subseteq C_l$ and there is no soft edge from comparison operation o_i to operation o_l . Therefore, for each operation o_l and each comparison operation $o_i \in A \subseteq C_l$, we have the following constraint:

$$Y_{A,l} \leq Y_{\{i\},l} \quad (\text{Formula 7})$$

Let's use the CDFG shown in Figure 1 to illustrate our ILP model. Assume that the timing constraint is four control steps, and the delay of each operation is one control step (i.e., $D_i=1$ for $i = 1, 2, \dots$, and 9). From both the ASAP calculation and the ALAP calculation [3], we can determine the control steps that an operation may be scheduled into. If operation o_i is impossible to be scheduled into control step j , the binary variable $x_{i,j}$ is definitely 0. Therefore, from both the ASAP calculation and the ALAP calculation, we can prune a lot of redundant binary variables without scarifying the exactness (optimality) of the solution.

There are two comparison operations: o_2 and o_5 . We have the following observations for the insertion of soft edges. First, comparison operation o_2 may shut down operations o_3 , o_6 , and o_7 for power saving. Secondly, comparison operation o_5 may shut down operations o_6 and

o_7 for power saving. Note that operation o_6 (operation o_7) may be shut down by both the two comparison operations.

For each multiplexer node, we assume that the probability of taking its false input (F) part is 50 %, and the probability of its truth input (T) part is also 50 %. The resource constraints are one adder, one multiplier, and one comparator. The power consumptions of adder, multiplexer, comparator, and multiplier are 3, 1, 4, and 20, respectively. On the other hand, the extra power consumption caused by the insertion of a soft edge is 1. Our optimization goal is to maximize the power saving. Therefore, our objective function is as below:

$$\text{Maximize } \{ 0.5*3*Y_{\{2\},3} + 0.5*20*Y_{\{2\},6} + 0.5*20*Y_{\{5\},6} - 0.25*20*Y_{\{2,5\},6} + 0.5*3*Y_{\{2\},7} + 0.5*3*Y_{\{5\},7} - 0.25*3*Y_{\{2,5\},7} - I*(Y_{\{2\},3} + Y_{\{2\},6} + Y_{\{2,5\},6} + Y_{\{5\},6} + Y_{\{2\},7} + Y_{\{2,5\},7} + Y_{\{5\},7}) \}$$

Due to the page limit, we cannot list all the constraints of our ILP formulation for this CDFG. In the following, for each formula, we use an example to explain its meaning.

Formula 2. Using operation o_2 as an example, exactly one binary variable is true among all the two binary variables associated with operation o_2 . Thus, we have the constraint $x_{2,2} + x_{2,3} = I$.

Formula 3. Using the dependency constraint $o_9 \rightarrow o_2$ as an example, operation o_2 can be executed if and only if operation o_9 has completed its execution. If operation o_9 is schedule into control step 1, then the operation o_2 can be schedule into the control step 2 and 3. If operation o_9 is schedule into control step 2 then the operation o_2 only can be schedule into control step 3. Thus, we have the constraint $x_{9,1} + 2x_{9,2} < 2x_{2,2} + 3x_{2,3}$.

Formula 4. Consider that there are three addition operations o_3 , o_7 , and o_9 that can be scheduled into control step 2. However, at each control step, only one adder can be utilized. Thus, we have the constraint $x_{3,2} + x_{7,2} + x_{9,2} \leq I$.

Formula 5. Consider the insertion of a soft edge from operation o_5 to operation o_6 . We have $Y_{\{5\},6} = I$, if and only if a soft edge from operation o_5 to operation o_6 is inserted. Note that, if there is a soft edge from operation o_5 to operation o_6 , operation o_5 must complete its execution before the execution of operation o_6 . Thus, we have the constraint $x_{5,1} + 2x_{5,2} < 2x_{6,2} + (I - Y_{\{5\},6}) * 4$, where 4 is the number of control steps.

Formula 6. Consider the binary variable $Y_{\{2,5\},6}$. We have $Y_{\{2,5\},6} = I$, if and only if $Y_{\{2\},6} = I$ and $Y_{\{5\},6} = I$. Thus, we have the constraint $Y_{\{2\},6} + Y_{\{5\},6} \leq Y_{\{2,5\},6} + I$.

Formula 7. Consider the binary variable $Y_{\{2,5\},6}$. We have $Y_{\{2,5\},6} = 0$, if $Y_{\{2\},6} = 0$. Thus, we have the constraint $Y_{\{2,5\},6} \leq Y_{\{2\},6}$.

After solving the ILP model, we find that the maximum power saving is 10 when $x_{1,4} = x_{2,2} = x_{3,3} = x_{4,3} = x_{5,1} = x_{6,2} = x_{7,2} = x_{8,1} = x_{9,1} = Y_{\{2\},3} = Y_{\{5\},6} = Y_{\{5\},7} = I$, and the values of other binary variables are 0. Figure 2 gives our results. Three soft edges are inserted under the design

constraints (timing and resource). Compared with the power consumption of the original CDFG, the power consumption of the modified CDFG is reduced from 59 to 49.

4. EXPERIMENTAL RESULTS

We use the Extended LINGO Release 8.0 as the ILP solver on a personal computer with P4-2.4GHz CPU and 512M Bytes RAM. Four benchmark circuits, including Jian [9], Mult [10], G2 [11], and G5 [12], are used to test the effectiveness of our approach. In addition, we also randomly generate two larger circuits, called R1 and R2, for experiments. The characteristics of these six test circuits are given in Table 1. we follow the same assumption of [8], we assume that: (1) the power consumptions of ALU (for the execution of addition operations and subtraction operations), multiplexer, comparator, multiplier, soft edge are 3, 1, 4, 20, and 1, respectively; (2) for each multiplexer node, the probability of taking its truth input (T) part is 50 %, and the probability of its false input (F) part is also 50 %. The column *Power Consumption* denotes the power consumption without power management.

Table 1: Characteristics of test circuits.

Circuit	Operations					Power Consumption
	+	-	#	>	*	
Jian	10	0	3	3	0	45.0
Mult	7	3	3	3	0	45.0
G2	9	0	3	3	9	222.0
G5	16	8	2	2	0	82.0
R1	28	27	6	6	15	495.0
R2	45	29	4	4	26	757.0

Table 2 demonstrates our experimental results. To demonstrate the effectiveness of our approach, we also implement the heuristic approach proposed in [8] for comparisons. The column *Design Constraints* gives the design constraints (timing and resource). The column *ALU* gives the number of ALUs. The column *C* gives the number of comparators. The column *M* gives the number of multiplier, which can execute the multiplication operations. The column *Steps* gives the number of control steps. The column *Power Saving* denotes the power saving. The column *[8]* gives the power saving obtained by the heuristic approach proposed in [8]. The column *Ours* gives the power saving obtained by our approach. The column *Imp%* gives the percentage of relative improvement of our approach over [8], i.e., $(\text{the power saving of ours}) / (\text{the power saving of [8]}) - 100\%$. Experimental data consistently show that our approach has significant improvements over [8].

5. CONCLUSIONS

In this paper, we present an ILP model for the simultaneous application of operation scheduling and power management. Our objective is to maximize the power saving under the design constraints (timing and resource). The major advantage of our work is that it guarantees

achieving the optimal solution. Compared with previous work that heuristically improves the power saving, experimental data consistently show that our approach has significant improvements.

Table 2: Experimental results.

Circuit	Design Constraints				Power Saving		
	ALU	C	M	Steps	[8]	Ours	Imp%
Jian	3	1	0	6	18.5	20	8.11
Mult	3	1	0	6	7	9	28.5
G2	2	1	2	8	88.5	108.5	22.6
G5	4	1	0	8	12	20	66.6
R1	7	1	8	11	48.5	79	62.8
R2	7	1	8	15	322	344	6.83

REFERENCES

- [1] S. Davidson, D. Landskov, B.D. Shriver, and P.W. Mallett, "Some Experiments in Local Microcode Compaction for Horizontal Machines", IEEE Trans. on Computers, pp. 460—477, 1981.
- [2] P.G. Paulin and J.P. Knight, "Force-Directed Scheduling in Automatic Data Path Synthesis", Proc. of IEEE/ACM Design Automation Conference, pp. 195—202, 1987.
- [3] C.T. Hwang, J.H. Lee, and Y.C. Hsu, "A Formal Approach to the Scheduling Problem in High Level Synthesis", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, vol. 10, no. 4, pp. 464—475, 1991.
- [4] S. Chaudhuri, S.A. Blthye, and R.A. Walker, "A Solution Methodology for Exact Design Space Exploration in a Three Dimensional Design Space", IEEE Trans. on VLSI Systems, vol. 5, no. 1, pp. 69—81, 1997.
- [5] S.H. Huang, C.H. Chiang, and C.H. Cheng, "Three-Dimension Scheduling under Multi-Cycle Interconnect Communications", IEICE Electronics Express, vol. 2, no. 4, pp. 108—114, 2005.
- [6] S.H. Huang and C.H. Cheng, "A Formal Approach to the Slack Driven Scheduling Problem in High Level Synthesis", Proc. of IEEE International Symposium on Circuits and Systems, pp. 5633—5636, 2005.
- [7] J. Monteiro, S. Devadas, P. Ashar and A. Mauskar, "Scheduling Techniques to Enable Power Management", Proc. of IEEE/ACM Design Automation Conference, pp. 349—352, 1996.
- [8] C. Chen and M. Sarrafzadeh, "Power-Manageable Scheduling Technique for Control Dominated High-Level Synthesis", Proc. of IEEE Design, Automation, and Test in Europe Conference and Exhibition, pp. 1016—1020, 2002.
- [9] J. Li and R.K. Gupta "An Algorithm to Determine Mutually Exclusive Operations in Behavioral Descriptions", Proc. of IEEE Design Automation and Test in Europe Conference and Exhibition, pp. 457—463, 1998.
- [10] K. Wakabayashi and H. Tanaka, "Global Scheduling Independent of Control Dependencies Based on Condition Vectors", Proc. of IEEE/ACM Design Automation Conference, pp. 112—115, 1992.
- [11] C.J. Tseng, R.S. Wei, S.G. Rothweiler, M.M. Tong, and A.K. Bose, "Bridge: A Versatile Behavioral Synthesis System", Proc. of IEEE/ACM Design Automation Conference, pp. 415—420, 1988.
- [12] T. Kim, J.W.S. Liu, and C.L. Liu, "A Scheduling Algorithm for Conditional Resource Sharing", Proc. of IEEE International Conference on Computer Aided Design, pp. 84—87, 1991.