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 Sarafzadeh [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 this 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.](image_url)

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), multiplexor (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 $3 \times 3 + 2 \times 1 + 2 \times 4 + 2 \times 20 = 59$.

However, in fact, the outputs of some operations are not used under some conditions. Using the CDFG shown in Figure 1 as an example, operation $o_6$ 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 $o_5$ 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 both the output of comparison operation $o_2$ and the output of comparison operation $o_5$, the power saving is 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 $o_i$ is $D_i$ clock cycles.

The notations used in our ILP model are as below.

(1) The notation $n$ denotes the number of operations.
(2) The notation $i$ 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 $Y_{AI}$ is a binary variable to model the insertion of soft edges. We have $Y_{AI} = 1$, if and only if soft edges are inserted from all comparison operations in the set $A$ to operation $o_i$. In other words, if $Y_{AI} = 1$, all the comparison operations in the set $A$ must be executed before operation $o_i$; otherwise, $Y_{AI} = 0$.
(13) $P_{AI}$ 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

$$\sum_{i=1}^{n} \sum_{d \in D_i} \left( -1 \right)^{x_{id}} \cdot P_{AI} \cdot Y_{AI} \cdot W_i - \sum_{i=1}^{n} \sum_{a \in A_i} Y_{AI} \cdot W_{soft}$$

(Formula 1)
Every operation must be scheduled to a control step. Therefore, for each operation $o_i$ we have the following constraint:

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

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

$$\sum_{j \in E_j} (j + D_j - 1) \cdot x_{i,j} < \sum_{j \in E_i} j \cdot x_{i,j} \quad \text{(Formula 3)}$$

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:

$$\sum_{o_i \in F(c), k \in D(o_i)} x_{i,j} \leq M_k \quad \text{(Formula 4)}$$

If a soft edge is added, an extra dependency constraint is enforced. Therefore, for each comparison operation $o_9$ that may shut down operation $o_i$, we have the following constraint:

$$\sum_{j \in E_i} (j + D_j - 1) \cdot x_{i,j} < \sum_{j \in E_i} j \cdot x_{i,j} + (1 - Y_{i,j}) \cdot t \quad \text{(Formula 5)}$$

The binary variable $Y_{i,j} = 1$, if for each comparison operation in the set $A \subseteq C_j$, there is a soft edge from it to operation $o_i$. Therefore, for each operation $o_i$, we have the following constraint:

$$\sum_{o_j \in A} Y_{i,j} \leq \sum_{o_j \in A} |A| - 1$$

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

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

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., $D = 1$ for $i = 1, 2, ..., 9$). 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 sacrificing 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 $o_9$ (operation $o_9$) maybe 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:

Maximize \[ 0.5*3*Y_{2,3} + 0.5*20*Y_{2,6} + 0.5*20*Y_{2,7} + 0.5*3*Y_{2,8} + 0.5*3*Y_{5,6} - 0.25*20*Y_{2,9} + 0.5*3*Y_{2,9} + 0.5*3*Y_{5,7} - 0.25*3*Y_{2,9} \]

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 binary variables associated with operation $o_2$. Thus, we have the constraint $x_{2,1} + x_{2,2} = 1$.

Formula 3. Using the dependency constraint $o_2 \rightarrow o_5$ as an example, operation $o_2$ can be executed if and only if operation $o_5$ has completed its execution. If operation $o_5$ is schedule into control step 1, then the operation $o_2$ can be schedule into the control step 2 and 3. If operation $o_5$ is schedule into control step 2 then the operation $o_2$ only can be schedule into control step 3. Thus, we have the constraint $x_{2,1} + 2x_{2,2} < 2x_{2,3} + 3x_{2,4}$.

Formula 4. Consider that there are three addition operations $o_4$, $o_5$, 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 $x_{2,1} + x_{2,2} + x_{2,3} \leq 1$.

Formula 5. Consider the insertion of a soft edge from operation $o_1$ to operation $o_6$. We have $Y_{2,6} = 1$, if and only if a soft edge from operation $o_1$ to operation $o_6$ is inserted. Note that, if there is a soft edge from operation $o_1$ to operation $o_6$, operation $o_1$ must complete its execution before the execution of operation $o_6$. Thus, we have the constraint $x_{2,1} + 2x_{2,2} < 2x_{2,3} + (1 - Y_{2,6}) \cdot 4$, where 4 is the number of control steps.

Formula 6. Consider the binary variable $Y_{2,5,6}$. We have $Y_{2,5,6} = 1$, if and only if $Y_{2,6} = 1$ and $Y_{5,6} = 1$. Thus, we have the constraint $Y_{2,6} + Y_{5,6} \leq Y_{2,5,6} + 1$.

Formula 7. Consider the binary variable $Y_{2,5,6}$. We have $Y_{2,5,6} = 0$, if $Y_{5,6} = 0$. 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 $x_{1,1} = x_{2,2} = x_{4,3} = x_{5,1} = x_{6,2} = x_{5,2} = x_{8,3} = x_{9,4}$, $Y_{2,3} = Y_{2,6} = Y_{2,7} = Y_{5,6} = Y_{5,7} = 1$, 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], Multi [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, multiplier, comparator, multiplier, soft edge are 3, 1, 4, 20, and 1, respectively; (2) for each multiplexer node, the probability of taking its true 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.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Operations</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jian</td>
<td>+ - # &gt; *</td>
<td>45.0</td>
</tr>
<tr>
<td>Multi</td>
<td>3 3 3 3</td>
<td>45.0</td>
</tr>
<tr>
<td>G2</td>
<td>9 0 3 3</td>
<td>222.0</td>
</tr>
<tr>
<td>G5</td>
<td>16 8 2 2</td>
<td>82.0</td>
</tr>
<tr>
<td>R1</td>
<td>28 27 6 6</td>
<td>495.0</td>
</tr>
<tr>
<td>R2</td>
<td>45 29 4 4</td>
<td>757.0</td>
</tr>
</tbody>
</table>

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 multipliers, 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 Imp% gives the percentage of relative improvement of our approach over [8], i.e., (the power saving of ours) / (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.

<table>
<thead>
<tr>
<th>Design Constraints</th>
<th>Power Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>C</td>
</tr>
<tr>
<td>Jian</td>
<td>3</td>
</tr>
<tr>
<td>Multi</td>
<td>3</td>
</tr>
<tr>
<td>G2</td>
<td>2</td>
</tr>
<tr>
<td>G5</td>
<td>4</td>
</tr>
<tr>
<td>R1</td>
<td>7</td>
</tr>
<tr>
<td>R2</td>
<td>7</td>
</tr>
</tbody>
</table>

REFERENCES