

Simultaneous Operation Scheduling and Operation Delay Selection To Minimize Cycle-by-Cycle Power Differential

Wei-Ting Yen, Shih-Hsu Huang, and Chun-Hua Cheng

Department of Electronic Engineering,
Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.
{shhuang}@cycu.edu.tw

Abstract. The cycle-by-cycle power differential determines the noise introduced due to inductive ground bounce. However, very few attentions are paid to minimize the cycle-by-cycle power differential in high-level synthesis stage. In this paper, we investigate the simultaneous application of operation scheduling and operation delay selection for minimizing the cycle-by-cycle power differential. An integer linear programming (ILP) approach is proposed to formally formulate this problem. Benchmark data consistently show that our approach can minimize the cycle-by-cycle power differential within an acceptable run time. Compared with previous work, the relative improvement of our approach achieves 44.8%.

Keywords: Integer Linear Programming, High-Level Synthesis, Data-Path Synthesis, Low Power, Operation Scheduling, Cycle-by-cycle Power Differential.

1 Introduction

In low-power designs for battery driven portable applications, average power, peak power, and cycle-by-cycle peak power differential are all equally important considerations. However, most previous high-level synthesis approaches [1-6] focus on the minimization of average power and/or peak power. To the best of our knowledge, [7] was the only high-level synthesis approach to the minimization of cycle-by-cycle power differential. In fact, the cycle-by-cycle power differential determines the noise introduced due to inductive ground bounce. Therefore, the minimization of cycle-by-cycle power differential is crucial in designing efficient and reliable integrated circuits.

In high-level synthesis stage [8], a behavior description is translated into 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 [7-10] is to assign each operation in the CDFG to a specific control step to start its execution. It has been recognized that operation scheduling greatly influences all quality aspects of the final

implementation. Therefore, according to Gajski [8], operation scheduling is “perhaps the most important task in high-level synthesis”. In [7], an integer linear programming (ILP) approach is proposed to formulate the problem of operation scheduling for the minimization of cycle-by-cycle power differential. In fact, up to now, [7] was the only attempt to reduce the cycle-by-cycle power differential via operation scheduling.

Different from previous work [7], in this paper, we combine operation scheduling and operation delay selection to minimize the cycle-by-cycle power differential. We find that, by slowing down non-critical operations, the cycle-by-cycle power differential can be further minimized. Therefore, based on that observation, we propose an ILP approach to formally formulate the simultaneous application of operation scheduling and operation delay selection. Compared with previous work [7], experimental results demonstrate that our approach can reduce the cycle-by-cycle power differential up to 44.8%.

The organization of this paper is as below. Section 2 gives the problem description. Section 3 describes our approach, and presents the ILP formulation. Section 4 demonstrates the experimental results. Finally, some concluding remarks are given in Section 5.

2 Motivation

Kindly assure that the Contact Volume Editor is given the name and email address of the contact author for your paper. The Contact Volume Editor uses these details to compile a list for our production department at SPS in India. Once the files have been worked upon, SPS sends a copy of the final pdf of each paper to its contact author. The contact author is asked to check through the final pdf to make sure that no errors have crept in during the transfer or preparation of the files. This should not be seen as an opportunity to update or copyedit the papers, which is not possible due to time constraints. Only errors introduced during the preparation of the files will be corrected.

This round of checking takes place about two weeks after the files have been sent to the Editorial by the Contact Volume Editor, i.e., roughly seven weeks before the start of the conference for conference proceedings, or seven weeks before the volume leaves the printer’s, for post-proceedings. If SPS does not receive a reply from a particular contact author, within the timeframe given, then it is presumed that the author has found no errors in the paper. The tight publication schedule of LNCS does not allow SPS to send reminders or search for alternative email addresses on the Internet.

In some cases, it is the Contact Volume Editor that checks all the pdfs. In such cases, the authors are not involved in the checking phase.

We use the data flow graph shown in Figure 1 to illustrate our motivation. The notation $>$ denotes the control operations, the notation $+$ denotes the addition operations, the notation $-$ denotes the subtraction operations, and the notation $*$ denotes the multiplication operations.

Assume that the power consumptions of control operation, addition operation, subtraction operation, and multiplication operation are 3mW, 3mW, 3mW, and 20mW, respectively. Figure 1(a) shows a scheduled DFG under the constraint of 1

multiplier, 1 adder, and 3 control steps. The peak power consumptions at control steps 1, 2, and 3 are 3 mW, 23 mW, and 20 mW, respectively. Analysis of the Figure 1(a) reveals that largest cycle-by-cycle is 20 mW.

For each operation, we assume that the power consumption is uniformly distributed to each control step when it executes. Suppose that the power consumption of a multiplication operation is 20mW. Then, the power consumption of each control step is 20 mW if the multiplication operation is executed within one control step; the power consumption of each control step is 10 mW if the multiplication operation is executed within two control steps; the power consumption of each control step is 20/3 mW if the multiplication operation is executed within three control step; and so on. Based on that observation, if we can slow down non-critical operations, the cycle-by-cycle power differential can be further minimized. For example, in Figure 1(a), operation o4 is a non-critical operation. Therefore, we can slow down operation o4. As a result, we obtain another scheduled DFG as shown in Figure 1(b). The peak power consumptions at control steps 1, 2, and 3 are 13 mW, 13 mW, and 20 mW, respectively. Analysis of the Figure 1(b) reveals that largest cycle-by-cycle power differential is only 7 mW.

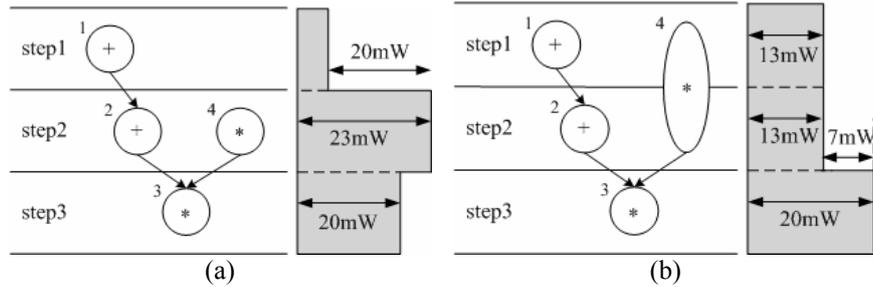


Fig. 1. A motivational example.

3 The Formulations

In our ILP formulations, we use the notation $x_{i,j,s}$ to denote a binary variable (i.e., an 0-1 integer variable). Binary variable $x_{i,j,s} = 1$, if and only if operation o_i is scheduled into control step j and the slack of operation o_i is exactly s clock cycles; otherwise, binary variable $x_{i,j,s} = 0$. Clearly, we have $1 \leq i \leq n$, $1 \leq j \leq t$ and $0 \leq s \leq t-1$, where n is the number of operations in the data flow graph and t is the total number of control steps. Thus, intuitively, the total number of binary variables is $n \cdot t^2$. However, in fact, from the ASAP (as soon as possible) and ALAP (as late as possible) schedules, we can find that a lot of binary variables are redundant since their values are definitely 0. Therefore, we can prune these redundant binary variables without sacrificing the accuracy of the solution.

The constants used in our ILP formulations are as below.

The value w_i denotes the power consumption of operation o_i .

The value s denotes delay time steps of each operation.

The value t denotes the total number of control steps.

The value n denotes the number of operations in the data flow graph.

The delay of each operation o_i corresponds to D_i clock cycles.

The value E_i denotes the earliest possible control step of operation o_i . Note that we can use the ASAP calculation to determine the value E_i for each operation o_i .

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 to determine the value L_i for each operation o_i .

We use FU_p to denote functional unit of type p , and we say $o_i \in FU_p$ if and only if operation o_i and o_k can be executed by FU_p .

The value M_p is the number of functional units of type p .

The cycle-by-cycle power differential minimization problem can be formulated as below.

Minimize $power_differential$ (Formula 1)

Subject to

For each control step c and each operation o_i, o_k and $1 \leq i, k \leq n, 1 \leq c \leq t-1$

$$-power_differential \leq \sum_{j=E_i}^c \sum_{s=c-(j+D_i-1)}^{L_i-j} \frac{w_i}{s+1} X_{i,j,s} -$$

$$\sum_{j=E_k}^{c+1} \sum_{s=c+1-(j+D_k-1)}^{L_k-j} \frac{w_k}{s+1} X_{k,j,s} \leq power_differential \quad (Formula 2)$$

For each operation o_i and $1 \leq i \leq n$

$$\sum_{j=E_i}^{L_i} \sum_{s=0}^{L_i-j} X_{i,j,s} = 1 \quad (Formula 3)$$

For each data dependency relation $o_i \rightarrow o_l$ and $1 \leq i \leq n, 1 \leq l \leq n$

$$\sum_{j=E_i}^{L_i} \sum_{s=0}^{L_i-j} (j+D+s-1) \cdot X_{i,j,s} < \sum_{j=E_l}^{L_l} \sum_{s=0}^{L_l-j} j \cdot X_{l,j,s} \quad (Formula 4)$$

For each control step c and each type of function unit FU_p

$$\sum_{o_i \in FU_p} \sum_{j=E_i}^c \sum_{s=c-(j+D_i-1)}^{L_i-j} X_{i,j,s} \leq M_p \quad (Formula 5)$$

Formula 1 defines the objective function. Formula 2 and Formula 3 describe peak power and peak power differential respectively. Formula 4 states the constraint that every operation must be scheduled to a control step. Formula 5 ensures that the data dependency relationships are preserved. Formula 6 states that the number of resources, type k , used in any control step should be less than or equal to the allocated resources M_p .

We use the HAL example [9] as shown in Figure 2 to illustrate the ILP formulations. The delay of each operation is 1 control step, i.e., $D_i = 1$ for $i = 1, 2, \dots, 11$. The timing constraint is 5 control steps; in other words, the total number of control steps is 5. Figure 2(a) and (b) show the ASAP and ALAP schedules of this example. According to the ASAP and ALAP schedules, we can prune all the redundant binary variables. Table 1 tabulates all the necessary (i.e., irredundant) binary variables associated with each operation.

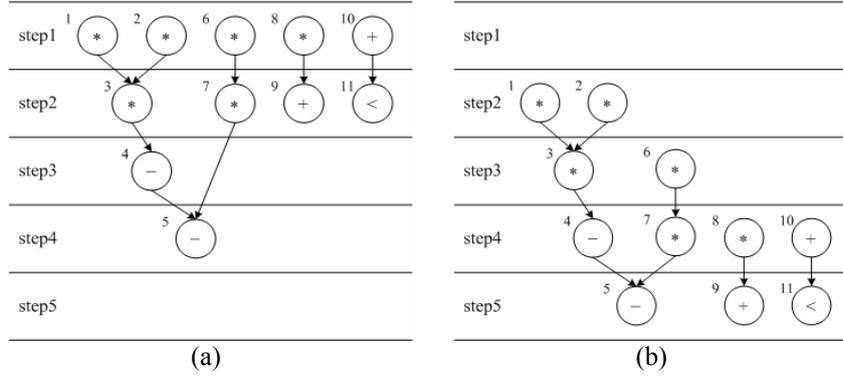


Fig. 2. HAL example. (a) ASAP schedule. (b) ALAP schedule.

Table 1. The binary variables associated with each operation.

Operation	Associated Binary Variables
O ₁	X _{1,1,0} , X _{1,1,1} , X _{1,2,0}
O ₂	X _{2,1,0} , X _{2,1,1} , X _{2,2,0}
O ₃	X _{3,2,0} , X _{3,2,1} , X _{3,3,0}
O ₄	X _{4,3,0} , X _{4,3,1} , X _{4,4,0}
O ₅	X _{5,4,0} , X _{5,4,1} , X _{5,5,0}
O ₆	X _{6,1,0} , X _{6,1,1} , X _{6,1,2} , X _{6,2,0} , X _{6,2,1} , X _{6,3,0}
O ₇	X _{7,2,0} , X _{7,2,1} , X _{7,2,2} , X _{7,3,0} , X _{7,3,1} , X _{7,4,0}
O ₈	X _{8,1,0} , X _{8,1,1} , X _{8,1,2} , X _{8,1,3} , X _{8,2,0} , X _{8,2,1} , X _{8,2,2} , X _{8,3,0} , X _{8,3,1} , X _{8,4,0}
O ₉	X _{9,2,0} , X _{9,2,1} , X _{9,2,2} , X _{9,2,3} , X _{9,3,0} , X _{9,3,1} , X _{9,3,2} , X _{9,4,0} , X _{9,4,1} , X _{9,5,0}
O ₁₀	X _{10,1,0} , X _{10,1,1} , X _{10,1,2} , X _{10,1,3} , X _{10,2,0} , X _{10,2,1} , X _{10,2,2} , X _{10,3,0} , X _{10,3,1} , X _{10,4,0}
O ₁₁	X _{11,2,0} , X _{11,2,1} , X _{11,2,2} , X _{11,2,3} , X _{11,3,0} , X _{11,3,1} , X _{11,3,2} , X _{11,4,0} , X _{11,4,1} , X _{11,5,0}

Assume that there are two types of functional units: the multiplier (FU_1), which can execute the multiplication operation, and the ALU (FU_2), which can execute other operations. Now we can construct the ILP formulations as below.

Formula 2. Using an example to illustrate peak power and peak power differential for control step 4 to control step 5. Thus, we have $-power_differential \leq (3x_{4,4,0} + 1.5x_{4,3,1} + 3x_{5,4,0} + 20x_{7,4,0} + 6.7x_{7,2,2} + 10x_{7,3,1} + 20x_{8,4,0} + 5x_{8,1,3} + 6.7x_{8,2,2} + 10x_{8,3,1} + 3x_{9,4,0} + 1x_{9,2,2} + 1.5x_{9,3,1} + 3x_{10,4,0} + 0.75x_{10,1,3} + 1x_{10,2,2} + 1.5x_{10,3,1} + 3x_{11,4,0} + 1x_{11,2,2} + 1.5x_{11,3,1} + 1.5x_{5,4,1} + 1x_{9,3,2} + 0.75x_{9,2,3} + 1.5x_{9,4,1} + 0.75x_{11,2,3} + 1x_{11,3,2} + 1.5x_{11,4,1}) - (3x_{5,5,0} + 3x_{9,5,0} + 3x_{11,5,0} + 1.5x_{5,4,1} + 1x_{9,3,2} + 0.75x_{9,2,3} + 1.5x_{9,4,1} + 0.75x_{11,2,3} +$

$1x_{11,3,2} + 1.5x_{11,4,1}$) \leq *power_differential*. All the constraints due to Formula 2 are listed in the following.

$$\begin{aligned} -\text{power_differential} \leq & (20x_{1,1,0} + 10x_{1,1,1} + 20x_{2,1,0} + 10x_{2,1,1} + 20x_{6,1,0} + 10x_{6,1,1} + \\ & 6.7x_{6,1,2} + 20x_{8,1,0} + 10x_{8,1,1} + 6.7x_{8,1,2} + 5x_{8,1,3} + 3x_{10,1,0} + 1.5x_{10,1,1} + 1x_{10,1,2} \\ & + 0.75x_{10,1,3}) - (10x_{1,1,1} + 20x_{1,2,0} + 10x_{2,1,1} + 20x_{2,2,0} + 20x_{3,2,0} + 10x_{3,2,1} + 10x_{6,1,1} + \\ & 6.7x_{6,1,2} + 20x_{6,2,0} + 10x_{6,2,1} + 20x_{7,2,0} + 10x_{7,2,1} + 6.7x_{7,2,2} + 10x_{8,1,1} + 6.7x_{8,1,2} + \\ & 5x_{8,1,3} + 20x_{8,2,0} + 10x_{8,2,1} + 6.7x_{8,2,2} + 3x_{9,2,0} + 1.5x_{9,2,1} + 1x_{9,2,2} + 0.75x_{9,2,3} + \\ & 3x_{10,2,0} + 1.5x_{10,1,1} + 1x_{10,1,2} + 0.75x_{10,1,3} + 1.5x_{10,2,1} + 1x_{10,2,2} + 3x_{11,2,0} + 1.5x_{11,2,1} + \\ & 1x_{11,2,2} + 0.75x_{11,2,3}) \leq \text{power_differential}; \end{aligned}$$

$$\begin{aligned} -\text{power_differential} \leq & (10x_{1,1,1} + 20x_{1,2,0} + 10x_{2,1,1} + 3x_{9,2,0} + 20x_{2,2,0} + 20x_{3,2,0} + \\ & 10x_{3,2,1} + 10x_{6,1,1} + 6.7x_{6,1,2} + 20x_{6,2,0} + 10x_{6,2,1} + 20x_{7,2,0} + 10x_{7,2,1} + 6.7x_{7,2,2} + \\ & 10x_{8,1,1} + 5x_{8,1,3} + 6.7x_{8,1,2} + 20x_{8,2,0} + 10x_{8,2,1} + 6.7x_{8,2,2} + 1.5x_{9,2,1} + 1x_{9,2,2} + \\ & 0.75x_{9,2,3} + 1.5x_{10,1,1} + 1x_{10,1,2} + 0.75x_{10,1,3} + 3x_{10,2,0} + 1.5x_{10,2,1} + 1x_{10,2,2} + 3x_{11,2,0} + \\ & 1.5x_{11,2,1} + 1x_{11,2,2} + x_{11,2,3}) - (10x_{3,2,1} + 20x_{3,3,0} + 6.7x_{6,1,2} + 10x_{6,2,1} + 20x_{6,3,0} + \\ & 10x_{7,2,1} + 6.7x_{7,2,2} + 20x_{7,3,0} + 10x_{7,3,1} + 6.7x_{8,1,2} + 5x_{8,1,3} + 10x_{8,2,1} + 6.7x_{8,2,2} + \\ & 20x_{8,3,0} + 10x_{8,3,1} + 3x_{4,3,0} + 1.5x_{4,3,1} + 1.5x_{9,2,1} + 1x_{9,2,2} + 0.75x_{9,2,3} + 3x_{9,3,0} + 1.5x_{9,3,1} \\ & + 1x_{9,3,2} + 1x_{10,1,2} + 0.75x_{10,1,3} + 1.5x_{10,2,1} + 1x_{10,2,2} + 3x_{10,3,0} + 1.5x_{10,3,1} + 1.5x_{11,2,1} \\ & + 1x_{11,2,2} + 0.75x_{11,2,3} + 3x_{11,3,0} + 1.5x_{11,3,1} + 1x_{11,3,2}) \leq \text{power_differential}; \end{aligned}$$

$$\begin{aligned} -\text{power_differential} \leq & (10x_{3,2,1} + 20x_{3,3,0} + 6.7x_{6,1,2} + 10x_{6,2,1} + 20x_{6,3,0} + 10x_{7,2,1} + \\ & 6.7x_{7,2,2} + 20x_{7,3,0} + 10x_{7,3,1} + 6.7x_{8,1,2} + 5x_{8,1,3} + 10x_{8,2,1} + 6.7x_{8,2,2} + 20x_{8,3,0} + \\ & 10x_{8,3,1} + 3x_{4,3,0} + 1.5x_{4,3,1} + 1.5x_{9,2,1} + 1x_{9,2,2} + 0.75x_{9,2,3} + 3x_{9,3,0} + 1.5x_{9,3,1} + 1x_{9,3,2} \\ & + 1x_{10,1,2} + 0.75x_{10,1,3} + 1.5x_{10,2,1} + 1x_{10,2,2} + 3x_{10,3,0} + 1.5x_{10,3,1} + 1.5x_{11,2,1} + 1x_{11,2,2} + \\ & 3x_{11,3,0} + 1x_{11,3,2} + 0.75x_{11,2,3} + 1.5x_{11,3,1}) - (20x_{3,3,0} + 10x_{3,2,1} + 3x_{4,3,0} + 20x_{6,3,0} + \\ & 6x_{6,1,2} + 10x_{6,2,1} + 20x_{7,3,0} + 10x_{7,2,1} + 20x_{8,3,0} + 6x_{8,1,2} + 10x_{8,2,1} + 3x_{9,3,0} + 1x_{9,2,1} + \\ & 3x_{10,3,0} + 1x_{10,1,2} + 1x_{10,2,1} + 3x_{11,3,0} + 1x_{11,2,1}) \leq \text{power_differential}; \end{aligned}$$

$$\begin{aligned} -\text{power_differential} \leq & (3x_{4,4,0} + 1.5x_{4,3,1} + 3x_{5,4,0} + 20x_{7,4,0} + 6.7x_{7,2,2} + 10x_{7,3,1} + \\ & 20x_{8,4,0} + 5x_{8,1,3} + 6.7x_{8,2,2} + 10x_{8,3,1} + 3x_{9,4,0} + 1x_{9,2,2} + 1.5x_{9,3,1} + 3x_{10,4,0} + 0.75x_{10,1,3} \\ & + 1x_{10,2,2} + 1.5x_{10,3,1} + 3x_{11,4,0} + 1x_{11,2,2} + 1.5x_{11,3,1} + 1.5x_{5,4,1} + 1x_{9,3,2} + 0.75x_{9,2,3} + \\ & 1.5x_{9,4,1} + 0.75x_{11,2,3} + 1x_{11,3,2} + 1.5x_{11,4,1}) - (3x_{5,5,0} + 3x_{9,5,0} + 3x_{11,5,0} + 1.5x_{5,4,1} + \\ & 1x_{9,3,2} + 0.75x_{9,2,3} + 1.5x_{9,4,1} + 0.75x_{11,2,3} + 1x_{11,3,2} + 1.5x_{11,4,1}) \leq \text{power_differential}; \end{aligned}$$

Formula 3. Using operation o_{10} as an example, there is exactly one binary variable is true among all the six binary variables associated with operation o_{10} . Thus, we have $x_{10,1,0} + x_{10,1,1} + x_{10,1,2} + x_{10,1,3} + x_{10,2,0} + x_{10,2,1} + x_{10,2,2} + x_{10,3,0} + x_{10,3,1} + x_{10,4,0} = 1$. All the constraints due to Formula 3 are listed in the following.

$$x_{1,1,0} + x_{1,1,1} + x_{1,2,0} = 1;$$

$$x_{2,1,0} + x_{2,1,1} + x_{2,2,0} = 1;$$

$$x_{3,2,0} + x_{3,2,1} + x_{3,3,0} = 1;$$

$$x_{4,3,0} + x_{4,3,1} + x_{4,4,0} = 1;$$

$$x_{5,4,0} + x_{5,4,1} + x_{5,5,0} = 1;$$

$$x_{6,1,0} + x_{6,1,1} + x_{6,1,2} + x_{6,2,0} + x_{6,2,1} + x_{6,3,0} = 1;$$

$$x_{7,2,0} + x_{7,2,1} + x_{7,2,2} + x_{7,3,0} + x_{7,3,1} + x_{7,4,0} = 1;$$

$$x_{8,1,0} + x_{8,1,1} + x_{8,1,2} + x_{8,1,3} + x_{8,2,0} + x_{8,2,1} + x_{8,2,2} + x_{8,3,0} + x_{8,3,1} + x_{8,4,0} = 1;$$

$$x_{9,2,0} + x_{9,2,1} + x_{9,2,2} + x_{9,2,3} + x_{9,3,0} + x_{9,3,1} + x_{9,3,2} + x_{9,4,0} + x_{9,4,1} + x_{9,5,0} = 1;$$

$$x_{10,1,0} + x_{10,1,1} + x_{10,1,2} + x_{10,1,3} + x_{10,2,0} + x_{10,2,1} + x_{10,2,2} + x_{10,3,0} + x_{10,3,1} + x_{10,4,0} = 1;$$

$$x_{11,2,0} + x_{11,2,1} + x_{11,2,2} + x_{11,2,3} + x_{11,3,0} + x_{11,3,1} + x_{11,3,2} + x_{11,4,0} + x_{11,4,1} + x_{11,5,0} = 1;$$

Formula 4. Using the data dependency relation of $o_1 \rightarrow o_3$ as an example, operation o_3 can be executed if and only if operation o_1 has completed its execution. If operation o_1 is scheduled into control step 1 with zero slack, operation o_3 can be scheduled into control step 2 with the slack of at most one clock cycle. If operation o_1 is scheduled into control step 1 with the slack of one clock cycle or operation o_1 is scheduled into control step 2 with zero slack, operation o_3 can only be scheduled into control step 3 with zero slack. Thus, we have $x_{1,1,0} + 2x_{1,1,1} + 2x_{1,2,0} < 2x_{3,2,0} + 2x_{3,2,1} + 3x_{3,3,0}$. All the constraints due to Formula 4 are listed in the following.

$$\begin{aligned}
& x_{1,1,0} + 2x_{1,1,1} + 2x_{1,2,0} < 2x_{3,2,0} + 2x_{3,2,1} + 3x_{3,3,0}; \\
& x_{2,1,0} + 2x_{2,1,1} + 2x_{2,2,0} < 2x_{3,2,0} + 2x_{3,2,1} + 3x_{3,3,0}; \\
& 2x_{3,2,0} + 3x_{3,2,1} + 3x_{3,3,0} < 3x_{4,3,0} + 3x_{4,3,1} + 4x_{4,4,0}; \\
& 3x_{4,3,0} + 4x_{4,3,1} + 4x_{4,4,0} < 4x_{5,4,0} + 4x_{5,4,1} + 5x_{5,5,0}; \\
& x_{6,1,0} + 2x_{6,1,1} + 3x_{6,1,2} + 2x_{6,2,0} + 3x_{6,2,1} + 3x_{6,3,0} < 2x_{7,2,0} + 2x_{7,2,1} + 2x_{7,2,2} + 3x_{7,3,0} + \\
& 3x_{7,3,1} + 4x_{7,4,0}; \\
& 2x_{7,2,0} + 3x_{7,2,1} + 4x_{7,2,2} + 3x_{7,3,0} + 4x_{7,3,1} + 4x_{7,4,0} < 4x_{5,4,0} + 4x_{5,4,1} + 5x_{5,5,0}; \\
& x_{8,1,0} + 2x_{8,1,1} + 3x_{8,1,2} + 4x_{8,1,3} + 2x_{8,2,0} + 3x_{8,2,1} + 4x_{8,2,2} + 3x_{8,3,0} + 4x_{8,3,1} + 4x_{8,4,0} < \\
& 2x_{9,2,0} + 2x_{9,2,1} + 2x_{9,2,2} + 2x_{9,2,3} + 3x_{9,3,0} + 3x_{9,3,1} + 3x_{9,3,2} + 4x_{9,4,0} + 4x_{9,4,1} + 5x_{9,5,0}; \\
& x_{10,1,0} + 2x_{10,1,1} + 3x_{10,1,2} + 4x_{10,1,3} + 2x_{10,2,0} + 3x_{10,2,1} + 4x_{10,2,2} + 3x_{10,3,0} + 4x_{10,3,1} + \\
& 4x_{10,4,0} < 2x_{11,2,0} + 2x_{11,2,1} + 2x_{11,2,2} + 2x_{11,2,3} + 3x_{11,3,0} + 3x_{11,3,1} + 3x_{11,3,2} + 4x_{11,4,0} + \\
& 4x_{11,4,1} + 5x_{11,5,0};
\end{aligned}$$

Formula 5. Consider that there are four multiplication operations o_3, o_6, o_7 and o_8 can be scheduled into control step 3. However, the maximum number of multiplication operations that can be scheduled into control step 3 is constrained by the number of multipliers (i.e., M_j). Thus, we have $x_{3,2,1} + x_{3,3,0} + x_{6,1,2} + x_{6,2,1} + x_{6,3,0} + x_{7,2,1} + x_{7,2,2} + x_{7,3,0} + x_{7,3,1} + x_{8,1,2} + x_{8,1,3} + x_{8,2,1} + x_{8,2,2} + x_{8,3,0} + x_{8,3,1} \leq M_1$. Suppose that we are given three multipliers and three ALUs; in other words, $M_1 = 3$ and $M_2 = 3$. All the constraints due to Formula 5 are listed in the following.

$$\begin{aligned}
& x_{1,1,0} + x_{1,1,1} + x_{2,1,0} + x_{2,1,1} + x_{6,1,0} + x_{6,1,1} + x_{6,1,2} + x_{8,1,0} + x_{8,1,1} + x_{8,1,2} + x_{8,1,3} \leq 3; \\
& x_{1,1,1} + x_{1,2,0} + x_{2,1,1} + x_{2,2,0} + x_{3,2,0} + x_{3,2,1} + x_{6,1,1} + x_{6,1,2} + x_{6,2,0} + x_{6,2,1} + x_{7,2,0} + \\
& x_{7,2,1} + x_{7,2,2} + x_{8,1,1} + x_{8,1,2} + x_{8,1,3} + x_{8,2,0} + x_{8,2,1} + x_{8,2,2} \leq 3; \\
& x_{3,2,1} + x_{3,3,0} + x_{6,1,2} + x_{6,2,1} + x_{6,3,0} + x_{7,2,1} + x_{7,2,2} + x_{7,3,0} + x_{7,3,1} + x_{8,1,2} + x_{8,1,3} + \\
& x_{8,2,1} + x_{8,2,2} + x_{8,3,0} + x_{8,3,1} \leq 3; \\
& x_{7,2,2} + x_{7,3,1} + x_{7,4,0} + x_{8,1,3} + x_{8,2,2} + x_{8,3,1} + x_{8,4,0} \leq 3; \\
& x_{10,1,0} + x_{10,1,1} + x_{10,1,2} + x_{10,1,3} \leq 3; \\
& x_{9,2,0} + x_{9,2,1} + x_{9,2,2} + x_{9,2,3} + x_{10,1,1} + x_{10,1,2} + x_{10,1,3} + x_{10,2,0} + x_{10,2,1} + x_{10,2,2} + x_{11,2,0} + \\
& x_{11,2,1} + x_{11,2,2} + x_{11,2,3} \leq 3; \\
& x_{4,3,0} + x_{4,3,1} + x_{9,2,1} + x_{9,2,2} + x_{9,2,3} + x_{9,3,0} + x_{9,3,1} + x_{9,3,2} + x_{10,1,2} + x_{10,1,3} + x_{10,2,1} + \\
& x_{10,2,2} + x_{10,3,0} + x_{10,3,1} + x_{11,2,1} + x_{11,2,2} + x_{11,2,3} + x_{11,3,0} + x_{11,3,1} + x_{11,3,2} \leq 3; \\
& x_{4,3,1} + x_{4,4,0} + x_{5,4,0} + x_{5,4,1} + x_{9,2,2} + x_{9,2,3} + x_{9,3,1} + x_{9,3,2} + x_{9,4,0} + x_{9,4,1} + x_{10,1,3} + \\
& x_{10,2,2} + x_{10,3,1} + x_{10,4,0} + x_{11,2,2} + x_{11,2,3} + x_{11,3,1} + x_{11,3,2} + x_{11,4,0} + x_{11,4,1} \leq 3; \\
& x_{5,4,1} + x_{5,5,0} + x_{9,2,3} + x_{9,3,2} + x_{9,4,1} + x_{9,5,0} + x_{11,2,3} + x_{11,3,2} + x_{11,4,1} + x_{11,5,0} \leq 3;
\end{aligned}$$

After solving these ILP formulations, we have that $x_{1,1,0} = x_{2,1,0} = x_{3,2,1} = x_{4,4,0} = x_{5,5,0} = x_{6,2,0} = x_{7,3,1} = x_{8,1,3} = x_{9,5,0} = x_{10,3,0} = x_{11,5,0} = 1$ and the values of other binary

variables are 0. Figure 3 gives the corresponding schedule. The cycle-by-cycle power differential is 10mW.

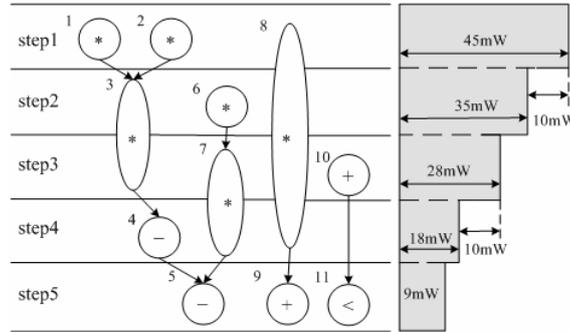


Fig. 3. Our result.

4 Experimental Results

We use the Extended LINGO Release 8.0 to solve the ILP formulations on a personal computer with P4-3.3GHz CPU and 1024M Bytes RAM. Five benchmark circuits, including EF [11], BF [12], HAL [9], AR [13], IIR [14], FIR [15], IDCT1 [16], and IDCT2 [16] are used to test the effectiveness of our approach. In our experiments, we assume that the power consumptions of control operation, addition operation, subtraction operation, and multiplication operation are 3mW, 3mW, 3mW, and 20mW, respectively. Given the number of multipliers, the number of ALUs, and the number of control steps, we can derive the ILP formulations for each benchmark circuit. The CPU time of each benchmark circuit is only few seconds.

For the purpose of comparisons, we also implement the ILP approach proposed in [7]. In the experiments of [7], we assume that the delay of each operation is 1 control step. Table 2 gives our experimental results. The column Resources denotes the resource constraints. The column Steps denotes the number of control steps. The column [7] denotes the largest cycle-by-cycle power differential obtained by the approach [7]. The column Ours denotes the largest cycle-by-cycle power differential obtained by our approach. The column Imp% denotes the percentage of improvement. The average improvement achieves 44.8%.

Table 2. Experimental results.

Circuit	Constraints		Cycle-by-cycle power differential		
	Resources	Steps	[7]	Ours	Imp%
EF	4 ALUs	16	17	14	17.7%
	4 ALUs	17	17	11	35.3%
BF	3 ALUs, 3 multipliers	9	17	14	17.7%
	3 ALUs, 3 multipliers	10	14	7	50.0%
HAL	3 ALUs, 3 multipliers	5	14	10	28.6%
	3 ALUs, 3 multipliers	6	14	7	50.0%
AR	4 ALUs, 4 multipliers	9	34	28	17.6%
	4 ALUs, 4 multipliers	10	28	20	28.6%
IIR	4 ALUs, 4 multipliers	5	31	26	16.1%
	4 ALUs, 4 multipliers	6	19	15	21.1%
FIR	2 ALUs, 2 multipliers	8	17	5	70.6%
	2 ALUs, 2 multipliers	9	17	5	70.6%
IDCT1	5 ALUs, 5 multipliers	13	6	1	83.3%
	5 ALUs, 5 multipliers	14	6	1	83.3%
IDCT2	7 ALUs, 7 multipliers	25	6	2	66.6%
	7 ALUs, 7 multipliers	26	5	2	60.0%

5 Conclusions

In this paper, we present an ILP formulation to model the cycle-by-cycle power differential minimization problem via operation delay selection. To the best of our knowledge, our paper is the first work that uses operation delay selection to reduce the cycle-by-cycle power differential. Benchmark data consistently show that our approach has significant cycle-by-cycle power differential reduction. Compared with previous work, our average improvement achieves 44.8%.

6 Acknowledgement

This work was supported in part by the National Science Council of R.O.C. under the grant number NSC 93-2220-E-033-001.

References

1. T. L. Martin and D. P. Siewiorek, "Non-Ideal Battery Properties and Low-Power Operation in Wearable Computing," Proc. of International Symposium on Wearable Computers, pp. 101—106, 1999.
2. W.T. Shiue, "High Level Synthesis for Peak Power Minimization using ILP", Proc. of IEEE International Conference on Application-Specific Systems, Architectures, and Processors, pp. 103—112, 2000.
3. L. Benini, G. Casterlli, A. Macii, and R. Scarsi, "Battery-Driven Dynamic Power Management", IEEE Design Test Computers, vol. 13, pp. 53—60, 2001.
4. 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.
5. S.P. Mohanty, N. Ranganathan, and S.K. Chappidi, "Peak Power Minimization through Datapath Scheduling", Proc. of IEEE Computer Society Annual Symposium on VLSI, pp. 121—126, 2003.
6. S.H. Huang, C.H. Cheng, C.H. Chiang, and C.M. Chang, "Peak Power Minimization through Power Management Scheduling", Proc. of IEEE Asia and Pacific Conference on Circuits and Systems, pp. 868—971, 2006.
7. S. P. Mohanty, N. Ranganathan, and S. K. Chappidi, "ILP Models for Energy and Transient Power Minimization During Behavioral Synthesis", Proc. of intl. Conf. on VLSI Design, pp. 745—748, 2004.
8. D.D. Gajski, N.D. Dutt, and B.M. Pangrle, "Silicon Compilation (tutorial)", Proc. of Custom Integrated Circuits Conference, pp. 102—110, 1986.
9. 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.
10. P. Faraboschi, J.A. Fisher and C. Young, "Instruction Scheduling for Instruction Level Parallel Processors", Proc. of the IEEE, vol. 89, no. 11, pp. 1638—1659, 2001.
11. M. Balakrishnan and P. Marwedel, "Integrated Scheduling and Binding: A Synthesis Approach for Design Space Exploration", Proc. of IEEE/ACM Design Automation Conference, pp. 68—74, 1989.
12. C.A. Papachristou and H. Konuk, "A Linear Program Driven Scheduling and Allocation Method Followed by an Interconnect Optimization Algorithm", Proc. of IEEE/ACM Design Automation Conference, pp. 77—83, 1990.
13. J. Ramanujam, S. Deshpande, J. Hong and M. Kandemir, "A Heuristic for Clock Selection in High-Level Synthesis", Proc. of IEEE Asia and South Pacific Design Automation Conference, pp. 414—419, 2002.
14. K. I. Kum and W. Sung, "Word-Length Optimization for High-Level Synthesis of Digital Signal Processing Systems", Proc. of IEEE Workshop on Signal Processing Systems, pp. 569—578, 1998.
15. D. Shin and K. Choi, "Low Power High Level Synthesis by Increasing Data Correlation", Proc. of IEEE International Symposium on Low Power Electronic Design, pp. 62—67, 1997.
16. C. Lee, M Potkonjak, and W. H. Maggione-Smith, "MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications System", Proc. of IEEE International Symposium on Microarchitecture, pp. 330— 335, 1997.