

Module binding for low power clock gating

Chun-Hua Cheng, Shih-Hsu Huang^{a)}, and Wen-Pin Tu

Department of Electronic Engineering,

Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.

a) shhuang@cycu.edu.tw

Abstract: In synchronous sequential circuit design, clock gating is recognized as a useful technique to reduce the power consumption. Conventionally, the clock gating is synthesized after high-level synthesis. In this paper, we point out that the module binding in high-level synthesis has a significant impact on the power consumption of gated clock tree. Based on that observation, we use an integer linear program (ILP) to formally formulate the problem. Our objective is to find a module binding solution so that the power consumption (of gated clock tree) can be minimized. It is noteworthy to mention that our work is the first attempt to synthesize the clock gating in the high-level synthesis stage. Benchmark data consistently show that our approach can greatly improve the existing design flow.

Keywords: electronic design automation, high-level synthesis, gated clock, and low power

Classification: Science and engineering for electronics

References

- [1] G. E. Tellez, A. Farrahi, and M. Sarrafzadeh, “Activity Driven Clock Design for Low Power Circuits,” *Proc. of IEEE/ACM International Conference on Computer Aided Design*, pp. 62–65, 1995.
- [2] A. Farrahi, C. Chen, A. Srivastava, G. Tellez, and M. Sarrafzadeh, “Activity Driven Clock Design,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 20, no. 6, pp. 705–714, 2001.
- [3] C. Lee, M. Potkonjak, and W. H. Maggione-Smith, “MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems,” *Proc. of IEEE International Symposium on Microarchitecture*, pp. 330–335, 1997.
- [4] S. H. Huang and C. H. Cheng, “An ILP Approach to the Simultaneous Application of Operation Scheduling and Power Management,” *IEICE Trans. Fundamentals*, vol. E91-A, no. 1, pp. 375–382, 2008.
- [5] G. D. Micheli, “Synthesis and Optimization of Digital Circuits,” Kluwer Academic Publishers, 1994.

1 Introduction

In a synchronous sequential circuit, the clock signal is the most active signal in the circuit. Thus, it is important to distribute the clock signal with low power. In fact, it is very often that only a portion of the circuit is active. As a result, clock gating has been recognized as one of the most effective techniques to reduce the power consumption. By shutting down the idle modules, clock gating can prevent the circuit from consuming unnecessary power. Furthermore, clock gating can also prevent wasteful switching in the clock tree by masking off the clock at the internal node of the tree.

In [1, 2], the relation between clock gating and high-level synthesis is pointed out. In fact, the synthesis of clock gating should know which modules are idle, when, and for how long. Tellez et al. [1] and Farrahi et al. [2] find that this information can be obtained from the results of high-level synthesis. From that observation, they [1, 2] derive the clock control logics, called activity driven clock tree, for low power based on the results of high-level synthesis.

Since the synthesis of clock gating is dependent on the results of high-level synthesis, we raise the following question: can we consider clock gating directly in the stage of high-level synthesis? To the best of our knowledge, no attention has been paid to synthesize the clock gating in the stage of high-level synthesis. Due to this motivation, this paper presents the first attempt to deal with this problem. Note that our framework is still based on the activity driven clock tree design methodology [1, 2]. However, different from previous researches [1, 2], we synthesize the clock gating during the module binding in high-level synthesis; in other words, compared with previous researches [1, 2], we synthesize the clock gating in an earlier stage.

In this paper, we use integer linear programming (ILP) to formally formulate our problem — the simultaneous application of clock gating and module binding. Given a scheduled data flow graph (DFG), our objective is to find a module binding solution so that the power consumption (of gated clock tree) can be minimized.

2 Motivation

In high-level synthesis, a behavior-level description is represented by a DFG, in which each node corresponds to an operation, and each directed edge corresponds to a dependency relationship. A scheduled DFG is a DFG in which each operation is scheduled into a proper control step to start its execution. Take the scheduled DFG, called *ex*, shown in Figure 1 (a) for illustration. This scheduled DFG *ex* has eight operations. Operations o_1 , o_2 , and o_3 are scheduled into control step 1, operation o_4 , and o_5 is scheduled into control step 2, and so on.

Module binding is to assign each operation in the scheduled DFG to a module that can execute it. Two operations can share the same module if they are not executed in the same control step. Take the scheduled DFG *ex* as an example. Suppose that we are given three adders, called A1, A2

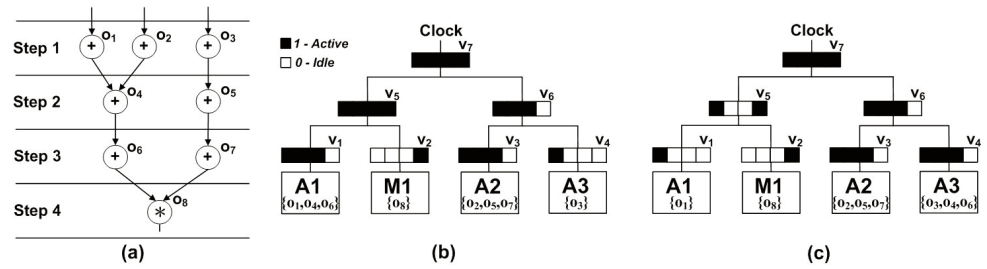


Fig. 1. (a) Scheduled DFG ex. (b) Clock tree CT1.
(c) Clock tree CT2.

and A3, and one multiplier, called M1. Operations o_1 , o_4 , and o_6 can share adder A1, operations o_2 , o_5 , and o_7 can share adder A2, operation o_3 can be assigned to adder A3, and operation o_8 can be assigned to multiplier M1. We use the following form to describe this module binding solution: $A1 = \{o_1, o_4, o_6\}$, $A2 = \{o_2, o_5, o_7\}$, $A3 = \{o_3\}$, and $M1 = \{o_8\}$. For brevity sake, in the following, we say this module binding solution is *sol1*.

If a module executes an operation at a control step, then the module is active at that control step; otherwise, the module is idle at that control step. Given a scheduled DFG and a module binding solution, we can derive the activity patterns of modules. Take the scheduled DFG *ex* for illustration. Suppose the module binding solution is *sol1*. We can derive the activity pattern of each module as below: since module A1 is active at control step 1, control step 2, and control step 3, the activity pattern of module A1 is 1110; since module M1 is active at control step 4, the activity pattern of module M1 is 0001; and so on.

In a gated clock tree, the activity pattern of each clock gate is defined as below. For a leaf node, its activity pattern is the activity pattern of the module that it controls. For an internal node, its activity pattern can be calculated by OR-ing (bitwise OR operation) the activity patterns of its children. Let PG_v be a constant that denotes the power consumption of clock gate v . Then, the total power consumption of clock gate v corresponds to the constant PG_v multiplies the number of active control steps. Further, the power consumption of clock tree is the summation of the power consumptions of all clock gates. By assuming the clock tree is binary tree structure, Tellez et al. [1] and Farrahi et al. [2] propose a methodology to build a clock tree in which the power consumption is minimized.

Take the scheduled DFG *ex* for illustration. Suppose that the module binding solution is *sol1*. After applying the activity driven clock tree design methodology [1, 2], a gated clock tree CT1 is obtained as displayed in Figure 1(b). The activity patterns of clock gates v_1 , v_2 , v_3 , v_4 , v_5 , v_6 , and v_7 are 1110, 0001, 1110, 1000, 1111, 1110, and 1111, respectively. Suppose that the power consumptions of clock gates v_1 , v_2 , v_3 , v_4 , v_5 , v_6 , and v_7 are 20, 20, 20, 20, 10, 10, and 10, respectively. Thus, since clock gate v_1 has three active control steps, the total power consumption of clock gate v_1 is $20 \times 3 = 60$. Similarly, the total power consumption of clock gates v_2 , v_3 , v_4 , v_5 , v_6 , and v_7 are 20, 60, 20, 40, 30, and 40, respectively. As a result, the

power consumption of clock tree CT1 is $60 + 20 + 60 + 20 + 40 + 30 + 40 = 270$.

Note previous researches [1, 2] minimize the power consumption of clock tree under the constraint that both the scheduled DFG and the module binding solution are given. Therefore, the impact of module binding is not considered in previous researches [1, 2]. However, in fact, the module binding solution greatly affects the power consumption of clock tree. Take the scheduled DFG *ex* for illustration. As discussed earlier, if the module binding solution is *sol_1*, the power consumption of clock tree CT1 is 270. On the other hand, let's consider another module binding solution *sol_2*: $A1 = \{o_1\}$, $A2 = \{o_2, o_5, o_7\}$, $A3 = \{o_3, o_4, o_6\}$, and $M1 = \{o_8\}$. After applying the activity driven clock tree design methodology [1, 2], a gated clock tree CT2 is obtained as displayed in Figure 1 (c). With an analysis, we find the power consumptions of clock tree CT2 is only 250.

From the above discussion, we have the following observation: different module binding solutions can lead to different clock tree power consumptions. Thus, we have the motivation to derive a module binding solution so that the power consumption of clock tree is minimized.

3 Our ILP Approach

In this section, we use integer linear programming (ILP) to formally draw up the problem of module binding for low power clock gating. First, we elaborate the notations used in our ILP approach as below.

- (1) The notation $A_{v,c}$ is a binary variable. If clock gate v is active at control step c , then $A_{v,c} = 1$; otherwise, $A_{v,c} = 0$.
- (2) The notation PG_v is a constant that denotes the power consumption of clock gate v .
- (3) The notation V denotes the set that includes all the clock gates in the clock tree. The notation B denotes the set that includes all the clock gates at the bottom level (i.e., the level nearest to the modules) of the clock tree. The notation I denotes the set that includes all the modules. The notation N denotes the set that includes all the operations. The notation C denotes the set that includes all the control steps. The notation $N(c)$ denotes the set that includes all the operations scheduled at control step c .
- (4) The notation $X_{n,t}$ is a binary variable. If operation n is assigned to module t , then $X_{n,t} = 1$; otherwise, $X_{n,t} = 0$.
- (5) The notation $\text{parent}(v)$ denotes the parent of clock gate v .
- (6) The notation $\beta_{i,c}$ is a binary variable. If module i is active at control step c , then $\beta_{i,c} = 1$; otherwise, $\beta_{i,c} = 0$.
- (7) The notation $Y_{i,v}$ is a binary variable. If module i is the child of clock gate v , then $Y_{i,v} = 1$; otherwise, $Y_{i,v} = 0$.
- (8) The notation $s(n)$ is a constant that denotes the control step into which operation n is scheduled. Note, for each operation n , its $s(n)$ can be obtained directly from the given scheduled DFG.

Next, we introduce the objective function and the constraints. Our objective

is to minimize the power consumption of clock tree. Therefore, the objective function is:

$$\text{minimize } \sum_{v \in V} \sum_{c \in C} A_{v,c} \times PG_v. \quad (\text{Formula 1})$$

Each operation n must be assigned to a module. Thus, for each operation, we have following constraint:

$$\sum_{i \in I} X_{n,i} = 1. \quad (\text{Formula 2})$$

Due to lifetime constraint, two operations cannot share the same module at the same control step. Thus, for each module i at control step c , we have the following constraint:

$$\sum_{n \in N(c)} X_{n,i} \leq 1 \quad (\text{Formula 3})$$

If clock gate v is active at control step c , then its parent is also active at control step c . Thus, for each clock gate v and each control step c , we have the following constraint:

$$A_{v,c} \leq A_{\text{parent}(v),c}. \quad (\text{Formula 4})$$

Each module i is the child of one clock gate at the bottom-level. Thus, for each module i , we have the following constraint:

$$\sum_{v \in B} Y_{i,v} = 1. \quad (\text{Formula 5})$$

If operation n is assigned to module i , then module i is active at control step $s(n)$. Thus, for each operation n and each module i , we have the following constraint:

$$X_{n,i} \leq \beta_{i,s(n)}. \quad (\text{Formula 6})$$

Suppose that clock gate v (at the bottom level of the clock tree) is the parent of module i . If module i is active at control step c , clock gate v is also active at control step c . Thus, for each clock gate v (at the bottom level of the clock tree), each module i , and each control step c , we have the following constraint:

$$Y_{i,v} + \beta_{i,c} \leq A_{v,c} + 1. \quad (\text{Formula 7})$$

In the following, we use the scheduled DFG *ex* for illustration. Suppose that the power consumptions of clock gates $v_1, v_2, v_3, v_4, v_5, v_6$, and v_7 are 20, 20, 20, 20, 10, 10, and 10, respectively. Then, our objective function is to minimize $\{(A_{v_1,1} + A_{v_1,2} + A_{v_1,3} + A_{v_1,4} + A_{v_2,1} + A_{v_2,2} + A_{v_2,3} + A_{v_2,4} + A_{v_3,1} + A_{v_3,2} + A_{v_3,3} + A_{v_3,4} + A_{v_4,1} + A_{v_4,2} + A_{v_4,3} + A_{v_4,4}) \times 20 + (A_{v_5,1} + A_{v_5,2} + A_{v_5,3} + A_{v_5,4} + A_{v_6,1} + A_{v_6,2} + A_{v_6,3} + A_{v_6,4} + A_{v_7,1} + A_{v_7,2} + A_{v_7,3} + A_{v_7,4}) \times 10\}$.

We list the constraints as below.

Formula 2. Operation o_1 must be assigned to a module. Thus, we have the constraint $X_{o_1,A1} + X_{o_1,A2} + X_{o_1,A3} = 1$. Similarly, we have the constraints $X_{o_2,A1} + X_{o_2,A2} + X_{o_2,A3} = 1$, $X_{o_3,A1} + X_{o_3,A2} + X_{o_3,A3} = 1$, and so on.

Formula 3. In the scheduled DFG *ex*, operations o_1, o_2 and o_3 are scheduled into control step 1. Due to lifetime constraint, only one operation can be

assigned to module A1. Thus, we have the constraint $X_{o1,A1} + X_{o2,A1} + X_{o3,A1} \leq 1$. Similarly, we have the constraints $X_{o1,A2} + X_{o2,A2} + X_{o3,A2} \leq 1$, $X_{o1,A3} + X_{o2,A3} + X_{o3,A3} \leq 1$, and so on.

Formula 4. Since clock gate v_5 is the parent of clock gate v_1 , we have the constraints $A_{v1,1} \leq A_{v5,1}$, $A_{v1,2} \leq A_{v5,2}$, $A_{v1,3} \leq A_{v5,3}$, and $A_{v1,4} \leq A_{v5,4}$. Similarly, we have the constraints $A_{v2,1} \leq A_{v5,1}$, $A_{v2,2} \leq A_{v5,2}$, $A_{v2,3} \leq A_{v5,3}$, $A_{v2,4} \leq A_{v5,4}$, and so on.

Formula 5. Since clock gate v_1 is at the bottom level of the clock tree, we have the constraint $Y_{A1,v1} + Y_{A2,v1} + Y_{A3,v1} + Y_{M1,v1} = 1$. Similarly, we have the constraints $Y_{A1,v2} + Y_{A2,v2} + Y_{A3,v2} + Y_{M1,v2} = 1$, $Y_{A1,v3} + Y_{A2,v3} + Y_{A3,v3} + Y_{M1,v3} = 1$, and so on.

Formula 6. If operation o_1 is assigned to module A1, then module A1 is active at control step 1. Therefore, we have the constraint $X_{o1,A1} \leq \beta_{A1,1}$. Similarly, we have the constraints $X_{o1,A2} \leq \beta_{A2,1}$, $X_{o1,A3} \leq \beta_{A3,1}$, and so on.

Formula 7. For clock gate v_1 and module A1, we have the following constraints for all the control steps: $Y_{A1,v1} + \beta_{A1,1} \leq A_{v1,1} + 1$, $Y_{A1,v1} + \beta_{A1,2} \leq A_{v1,2} + 1$, $Y_{A1,v1} + \beta_{A1,3} \leq A_{v1,3} + 1$, and $Y_{A1,v1} + \beta_{A1,4} \leq A_{v1,4} + 1$. Similarly, we have the constraints $Y_{A1,v2} + \beta_{A1,1} \leq A_{v2,1} + 1$, $Y_{A1,v2} + \beta_{A1,2} \leq A_{v2,2} + 1$, $Y_{A1,v2} + \beta_{A1,3} \leq A_{v2,3} + 1$, and so on.

After solving the ILP formulation, we find the power consumption can be minimized to be 250 when $\beta_{A1,1} = \beta_{A2,1} = \beta_{A2,2} = \beta_{A2,3} = \beta_{A3,1} = \beta_{A3,2} = \beta_{A3,3} = \beta_{M1,4} = X_{O1,A1} = X_{O2,A2} = X_{O3,A3} = X_{O4,A3} = X_{O5,A2} = X_{O6,A3} = X_{O7,A2} = X_{O8,M1} = A_{v1,1} = A_{v2,4} = A_{v3,1} = A_{v3,2} = A_{v3,3} = A_{v4,1} = A_{v4,2} = A_{v4,3} = A_{v5,1} = A_{v5,4} = A_{v6,1} = A_{v6,2} = A_{v6,3} = A_{v7,1} = A_{v7,2} = A_{v7,3} = A_{v7,4} = 1$, and other binary variables are 0. The corresponding module binding solution is *sol_2*. The corresponding clock tree is displayed in Figure 1 (c).

4 Experimental Results

We use Extended LINGO Release 10.0 as the ILP solver, and our platform is Windows 2003 x64 running on Intel Xeon E5355 CPU. Seven benchmark circuits are used to test the effectiveness of our ILP approach. Circuits Jian, BF, G2, and G5 are popular DSP applications, circuit IDCT1 is adopted from the MediaBench suite [3], and circuits R1 and R2 are adopted from [4]. The scheduled DFG of each circuit is obtained by the list scheduling technique [5]. For the purpose of comparison, we also implement the existing design flow: first, we use the left edge algorithm [5] to derive a module binding solution; then, we use the activity driven clock tree design methodology [1, 2] to minimize the power consumption.

In our experiment, we use TSMC 0.18 μm cell library to implement each circuit. A two-step process is used to estimate the power consumptions of clock gates. In the first step, by using the wire load model provided in the cell library, we estimate the wire load of each clock gate according to the areas of modules and the numbers of fan-outs. Then, in the second step, for each clock gate, we calculate its power consumption according to its output

Table I. Our experimental results.

Circuit	Design Constraints	Power Consumption		Improvement
		Existing Flow (mW)	Our Approach (mW)	
Jian	(5, 4, 0, 0, 0, 1, 1)	1.928	1.698	11.93%
BF	(8, 2, 2, 3, 0, 0, 0)	2.209	2.117	4.16%
G2	(8, 3, 0, 4, 0, 1, 1)	2.218	2.086	5.95%
G5	(7, 7, 0, 0, 0, 1, 1)	2.582	2.002	22.46%
IDCT1	(12, 4, 2, 6, 1, 0, 0)	2.740	2.484	9.34%
R1	(5, 17, 0, 4, 0, 3, 4)	9.453	8.731	7.64%
R2	(6, 24, 0, 8, 0, 2, 2)	10.857	9.206	15.21%

load (i.e., the summation of the wire load and the input pin capacitances of its successors).

Table I tabulates our experimental results. Note, in each circuit, both the existing design flow and our ILP approach can obtain results within only few minutes. The column *Design Constraints* gives 6-tuple (#step, #add, #sub, #mul, #div, #sel, #com), where #step, #add, #sub, #mul, #div, #sel, and #com are the numbers of control steps, adders, subtractors, multipliers, divisors, selectors, and comparators, respectively. We report the power consumptions obtained by the existing design flow and our ILP approach, respectively. The column *Improvement* denotes the relative improvement of our ILP approach over the existing design flow. With an analysis, we find the average improvement of our ILP approach achieves 10.96%.

5 Conclusions

In this paper, we point out that the module binding in high-level synthesis has a significant impact on the power consumption of gated clock tree. Based on that observation, we formally formulate the problem of module binding for low power clock gating. It is noteworthy to mention that our work is the first attempt to synthesize the clock gating in the high-level synthesis stage. Experimental data consistently show that our approach can greatly improve the existing design flow.

Acknowledgments

This work was supported in part by the National Science Council of Taiwan, R.O.C., under grant number NSC 96-2628-E-033-004-MY3.