情報処理学会研究報告
IPSJ SIG Technical Report

自動クロックゲーティング生成における電力最適化制御信号選択手法

満　欣[†a)]　堀山　貴史[††]　木村　晋二[†]

クロックゲーティングは、レジスタへのクロック供給を制御することで電力を削減する手法で、順序回路の動的電力削減に広く用いられている。これまでハードウェア記述言語における新しい値のレジスタへの代入条件を用いる手法や、状態遷移の解析からレジスタに代入する条件を抽出する手法などが知られているが、より効果的な自動化手法が求められていた。レジスタの現在の値と新しい値の EXOR がクロック停止確率最大となる信号であることが知られているが、個別にゲーティング回路を付加することは非効率で、共有が不可欠である。そこで本稿では、論理関数処理に基づき制御信号候補から最適なゲーティング回路の最適な共有を行う手法を提案する。本手法は二分決定グラフ (Binary Decision Diagram, BDD) を用いて実現され、カウンタや ISCAS 89 ベンチマーク回路で効果を確認した。カウンタでは 37% ~ 76 % の電力削減が得られ、また ISCAS ベンチマーク回路では 2% ~ 18 % の電力削減が確認できた。

Automatic Clock Gating Generation through Power-optimal Control Signal Selection

Xin MAN[†a)] Takashi HORIYAMA[††] and Shinji KIMURA[†]

Clock gating is an effective technique to reduce dynamic power consumption for sequential circuits. There have been proposed clock gating generation methods using the condition specified by designers or the extracted condition by the analysis of state transitions. EXOR of the current value and the new value of a register is the control signal which can minimize the probability of clock supply to the register, but it is infeasible to add one clock gating logic for each register. In our research, we propose a method for automatic clock gating generation through control signal candidates extraction and power-optimal control signal selection based on the optimum sharing. The method is implemented based on BDD (Binary Decision Diagram). The method is applied to counters and ISCAS89 benchmark circuits. There have been found 37% ~ 76% power reductions on counter circuits and 2% ~ 18% power reduction on benchmark circuits.

1. Introduction

With the proliferation of low-power requirements and thermal limitations, power reduction becomes one of important themes in VLSI design. Among the methods of reducing dynamic power consumption [1][2], clock gating technique [3]-[10] is one of the most efficient and widely used techniques, where the clock signal is selectively gated by the control signal for registers in the design when the values stored in the registers have not been changed so as to save the power consumption of the registers and the whole circuits.

The most common approach in previous research on clock gating generation [6][11] is to specify the gating condition under which the clock signal could be safely blocked based on the current state value and the next state function of a register by designers using structural gating approach. An automatic technique has been proposed recently [12] using candidate extraction and control signal selection. The method shows the reduction compared to the structural gating approach, however the method may cause overlapping problem when there are some AND gates of the original control candidates and some other signals.

In the research, we focus on automatic clock gating generation and propose an optimization algorithm through power-optimal control signal selection based on BDD. The method includes two phases, gating control signal candidates extraction phase based on [12] and a newly formalized power-optimal control signal selection process. Since the inserted clock gating element itself causes extra power dissipation, the sharing of control signals by different registers has been taken into consideration for power optimization. By experiments, our method is useful concerning sharing conditions of control signals by several registers on power minimization. We modified the BDD package by adding a mechanism to cope with the probability of input variables and a function to compute the minimum cost based on the input probability. The method is applied to counter circuits to check the co-relation with power simulation results, and ISCAS89 benchmark circuits.

The rest of this paper is organized as follows: Section 2 introduces clock gating technique. Section 3 presents the optimization algorithm. Section 4 describes BDD based method. Section 5 shows the implementation of the optimization algorithm. The experimental results and conclusions are shown in Section 6 and Section 7.

2. Clock Gating

Clock gating control is inserted to register banks by which clock signal is gated during some clock cycles when the values stored by these register banks are the same so as to reduce the power consumption of the whole circuit.

Without clock gating, synthesis tools in general implement register banks by using a

† 早稲田大学 大学院情報生産システム研究科 (Grad. School of IPS, Waseda University)
†† 埼玉大学 理工学研究科 (Grad. School of Science and Technology, Saitama University)

feedback loop and a multiplexer as shown in Fig. 1.

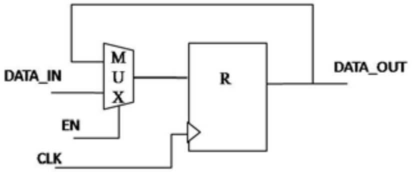

Fig. 1 Registers with Multiplexer.

Latch-based clock gating style consisting of a latch and an AND gate is widely adopted to avoid glitches on the clock gating control signal (EN) which can corrupt the clock signal to the register as shown in Fig. 2. Using structural gating approach, the gating condition under which clock signal could be safely blocked without violating the functional correctness of the circuits is identified based on the current state value and the next state function of a register by designers as shown in Fig. 3.

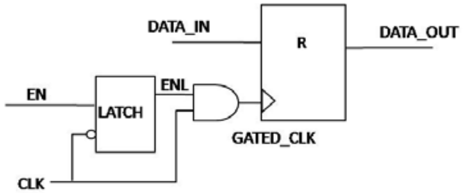

Fig. 2 Latch-based Clock Gating Style.

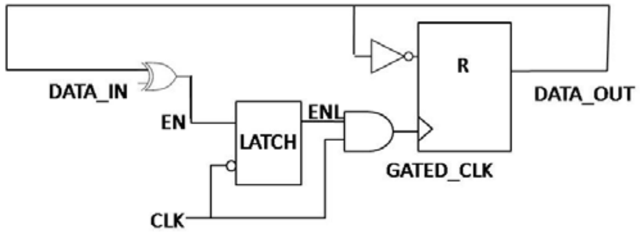

Fig. 3 Structural Gating Approach.

A register r should acquire a new value (DATA_IN) only when the value is not the same as the current state value (DATA_OUT), so the maximum possibility to stop the clock can be obtained by taking XOR of the new value and the current state value. If the XOR is 0, clock signal could be gated without violating the functional correctness of the circuit. However, since the clock gating element consumes extra power consumption, it is not effective to insert a clock gating for each register and the sharing of control signals is very important. Therefore, in the following section we propose an optimization algorithm considering the cost of gating control circuits in order to achieve the optimum power reduction of the circuit.

3. Optimization Algorithm

3.1 Clock Gating Control Signal Candidates Extraction

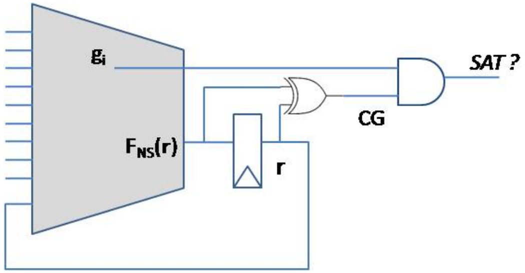

Fig. 4 Candidates Extraction [12].

In this section we present the clock gating control signal candidates extraction method of the optimization algorithm based on paper [12]. Let *r* be the current state value and $F_{NS}(r)$ be the next state function of a register as shown in Fig. 4. As mentioned in the previous section, when the current state value *r* and the next state value $F_{NS}(r)$ of the register are the same, we can switch off the clock signal. To maintain the functional correctness of the circuit, the gating condition CG is described in Eq. 1. If CG is 1, the clock signal should be applied.

$$CG = F_{NS}(r) \oplus r \tag{1}$$

The clock gating control signal candidates are extracted using CG as shown in Fig. 4. In the figure, the satisfaction of the logic AND of CG and a gate output g_i is checked. If CG AND gi is always 0, then g_i is 0, when CG takes the value of 1. In this case, $\overline{g_i}$ can be used as a clock gating control signal so that we can check that by using SAT procedure or BDD. Note that the on-set of $\overline{g_i}$ includes the on-set of CG. Also note that $\overline{g_i \cdot g_k}$ is also a candidate where g_k is another gate output and "·" represents logical AND.

By the method, we have a list of clock gating control signal candidates for each register. For each candidate $\overline{g_i}$, we can compute the 1-probability P_i by using BDD which corresponds to the probability applying the clock signal. Note that some candidates might be included in the candidate lists of different registers.

In [12], they show a method to select the clock gating control candidates based on covering problem. However, this method may cause overlapping problem when there are some AND gates of the original control candidates and some other signals. To avoid such overlapping problem, in the next section we propose a new selection method useful when the same signal might be candidates on many registers.

3.2 Clock Gating Control Signal Selection

Table 1 Cost Evaluation.

control \ register	C_0	C_1	…	C_j	…	C_m	
	P_0	P_1	…	P_j	…	P_m	
r_0	x_{00} 0/1	x_{01} 0/1	…	x_{0j} 0/1	…	x_{0m} 0/1	z_0
r_1	x_{10} 0/1	x_{11} 0/1	…	x_{1j} 0/1	…	x_{1m} 0/1	z_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
r_i	x_{i0} 0/1	x_{i1} 0/1	…	x_{ij} 0/1	…	x_{im} 0/1	z_i
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
r_n	x_{n0} 0/1	x_{n1} 0/1	…	x_{nj} 0/1	…	x_{nm} 0/1	z_n
	y_0	y_1	…	y_j	…	y_m	

By arranging registers and the full list of candidates, we can obtain a table as shown in Table 1, where each line represents the information of clock gating control signal candidates for each register in a given circuit, while each column shows the information of each clock gating control signal candidate.

At line i and column j, we put a variable x_{ij}, taking a value of 0 or 1. x_{ij}=1 denotes that the register r_i accepts C_j as a clock gating control. Note that the value of some x_{ij} can be set to 0 at the candidate extraction step. For each line i, we put a variable z_i and z_i=1 shows the case when the register r_i has no clock gating. Since each register can have only one gating control signal or no control signal, the summation of x_{ij} ($0 \le j \le m$) and z_i should be 1. We represent this constraint by Eq. 2.

$$\sum_j x_{ij} + z_i = 1 \tag{2}$$

For each column j, variable y_j is added to note where there needs a clock gating circuit of C_j. If some of x_{ij} ($0 \le i \le n$) is 1, y_j should be 1, otherwise y_j is 0. This is represented by Eq. 3.

$$\text{If } \sum_i x_{ij} > 0, \text{ then } y_j = 1 \tag{3}$$

In Table 1, P_j denotes the 1-probability of each candidate C_j. If x_{ij} is 1, the register r_i's switching activity can be P_j, while if z_i is one, the switching activity of r_i is 1. When x_{ij}=1, we need clock gating circuit for C_j and the switching activity of the clock gating circuit is measured with coefficient α, which shows the power consumption of clock gating logic with respect to that of a flip-flop. By experiments using the power simulation, α is measured as 0.8 on VDEC library. We would like to minimize the cost as shown in Eq. 4.

$$\text{cost} = \sum_j \alpha y_j + \sum_i \sum_j (x_{ij} P_j) + \sum_i z_i \tag{4}$$

The optimization method can be formalized as follows. The object of the optimization method is to minimize the cost presented by Eq. 4, under the condition defined by Eq. 2 and Eq. 3.

$$\begin{aligned} &\text{Minimize} && \sum_j \alpha y_j + \sum_i \sum_j (x_{ij} P_j) + \sum_i z_i \\ &\text{Under} && \sum_j x_{ij} + z_i = 1 \\ & && \text{If } \sum_i x_{ij} > 0, \text{ then } y_j = 1 \\ & && x_{ij}, y_j, z_i \in \{0,1\} \end{aligned}$$

4. BDD Based Method

Based on the above formulae, we show an optimization method based on BDD.

For a circuit of *n* registers with *m* potential candidates of gating control signal, the number of variable x would be *m*n*.

The flow of the BDD based method is as follows: (1) Extract clock gating control signal candidates and compute its corresponding probabilities based on BDD. (2) Construct BDD's of each port of a circuit satisfying the constraints. (3) Evaluate cost from the extracted probabilities of the candidates and select clock gating control signals using BDD.

In the following part we focus on the step (2) and (3) for control signal selection.

4.1 Logic Functions for Constraints

Before describing cost evaluation using BDD based method, we rewrite the two constraints by the following logic functions. The constraint formula by Eq. 2 is described as:

$$F_{lc\_i}(\mathbf{x},\mathbf{z}) = x_{i0}'x_{i1}'\ldots x_{ij}'\ldots x_{im}'z_i + x_{i0}x_{i1}'\ldots x_{ij}'\ldots x_{im}'z_i' + \ldots + x_{i0}'x_{i1}'\ldots x_{ij}\ldots x_{im}'z_i' + \ldots + x_{i0}'x_{i1}'\ldots x_{ij}'\ldots x_{im}z_i' \quad (5)$$

$$F_{lc}(\mathbf{x},\mathbf{z}) = F_{lc\_0}(\mathbf{x},\mathbf{z})F_{lc\_1}(\mathbf{x},\mathbf{z})\ldots\ F_{lc\_i}(\mathbf{x},\mathbf{z})\ldots F_{lc\_n}(\mathbf{x},\mathbf{z}) \quad (6)$$

where $F_{lc\_i}(\mathbf{x},\mathbf{z})$ is the logic function for constraint of each line in Table 1, and $F_{lc}(\mathbf{x},\mathbf{z})$ is the logic function for constraints of all lines with variables $\mathbf{x}=(x_{00}, x_{01}, \ldots, x_{0m}, x_{10}, x_{11}, \ldots, x_{1m}, \ldots, x_{n0}, x_{n1}, \ldots, x_{nm})$ and variables $\mathbf{z}=(z_0, z_1, \ldots, z_n)$ in Table 1. The symbol " ' " denotes logical NOT.

The constraint formula by Eq. 3 is described as

$$F_{cc\_j}(\mathbf{x},\mathbf{y}) = x_{0j}'x_{1j}'\ldots x_{ij}'\ldots x_{nj}'y_j' + x_{0j}y_j + x_{1j}y_j + \ldots + x_{ij}y_j + \ldots + x_{nj}y_j \quad (7)$$

$$F_{cc}(\mathbf{x},\mathbf{y}) = F_{cc\_0}(\mathbf{x},\mathbf{y})F_{cc\_1}(\mathbf{x},\mathbf{y})\ldots\ F_{cc\_j}(\mathbf{x},\mathbf{y})\ldots F_{cc\_m}(\mathbf{x},\mathbf{y}) \quad (8)$$

where $F_{cc\_j}(\mathbf{x},\mathbf{y})$ is the logic function for constraint of each column in Table 1, and $F_{cc}(\mathbf{x},\mathbf{y})$ is the logic function for constraints of all columns with variables $\mathbf{x}$ and variables $\mathbf{y}=(y_0, y_1, \ldots, y_m)$ in Table 1.

4.2 Cost Evaluation

After constructing a BDD of the AND of the two constraints, we can compute the minimum cost on BDD. For the computation, we modified our BDD package by adding a mechanism to cope with the probability of input variables and a function to compute the minimum cost based on the input probability in accordance with Eq. 4.

Fig. 5 shows a pseudo-code of the recursive function "*cost_calculation*". In the function *(bddptr == TRUE)* represents the 1-leaf, while *(bddptr == FALSE)* represents the 0-leaf in BDD. The basic idea of the cost_calculation function is that for each given node in BDD pointed by *bddptr*, we define variables to store the cost and the direction information (*direction*) for each given node respectively. As shown in Fig. 5, *cost* stores the minimum cost to go to 1-leaf. In order to obtain the minimum cost for each given node, we compute the cost of the node connected with 0-edge (*cost_low*) and that with 1-edge (*cost_high*) in BDD. If we follow the 1-edge, we should add the cost corresponding to the variable (*var_prob*). Note that we should compute both the cost to 0-leaf and that to 1-leaf since there are negative edges. We also use the computed cost for the repeated traversal to the same node. These mechanisms are implemented in one function.

```
double cost_calculation(BDD *bddptr){
  if ( bddptr == TRUE ) then return ( 0 );
  if ( bddptr == FALSE ) then return ( ∞ );

  if ( bddptr → cost is defined ) then return ( bddptr → cost );

  cost_low = cost_calculation(bddptr → low);
  cost_high = cost_calculation(bddptr → high) + var_prob(bddptr → variable);

  if ( cost_low < cost_high ) { bddptr → direction = 0;
                                bddptr → cost = cost_low; }
  else { bddptr → direction = 1;
       bddptr → cost = cost_high; }

  return ( bddptr → cost );
}
```

Fig. 5 Function for Minimum Cost Calculation.

5. Implementation of Optimization Algorithm

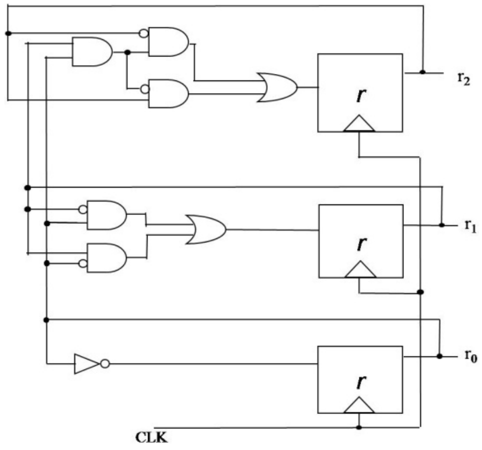

Fig. 6 3-bit Counter Circuit without Clock Gating.

The proposed method is implemented in the BDD package and applied to counter circuits with power simulation. At first, we take the 3-bit counter circuit as shown in Fig. 6 as an example to explain the special features of counters and how to apply the optimization algorithm and the BDD-based method for cost evaluation.

5.1 Clock Gating Control Signal Candidates Extraction

Table 2 Cost Evaluation with 3-Bit Counter Case.

C \ r	r_0 **C_0** (P_0=0.5)	r_1 **C_1** (P_1=0.5)	r_0r_1 **C_2** (P_2=0.25)	$r_0r_1r_2$ **C_3** (P_3=0.125)	
r_0	x_{00} 0	x_{02} 0	x_{04} 0	x_{05} 0	**z_0**
r_1	x_{10} 1	x_{12} 0	x_{14} 0	x_{15} 0	**z_1**
r_2	x_{20} 1	x_{22} 1	x_{21} 1	x_{22} 0	**z_2**
	y_0	**y_2**	**y_4**	**y_5**	

The first step consists of extracting a set of clock gating control signal candidates for each register, which satisfies the correctness condition (Eq. 1). In Table 2, registers and all clock gating control signal candidates have been listed. Let (r_2, r_1, r_0) be registers (and the outputs of the registers) of a 3-bit counter, where r_2 is the MSB (Most Significant Bit) and r_0 is the LSB (Least Significant Bit). (r_2, r_1, r_0) takes the value of (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0) and so on repeatedly. For register r_0, we have no clock gating control signal candidate. For register r_1, we have the control candidates $C_0 = r_0$. For register r_2, we have the control candidates C_0, $C_1 = r_1$ and $C_2 = r_0\ r_1$. Note that when C_2 is 1, then C_0 is 1. In the counter case, there are a lot of AND gates of a control candidate and some other signals, which is defined as the conjunction of up to i registers, as shown in Eq. 9, where $\prod$ is the logical AND. P_0, P_1, P_2 are the 1-probabilities of gating control candidates.

the gating control signal C_i is defined as

$$C_i = \prod_{k=0...i} r_k \tag{9}$$

5.2 Clock Gating Control Signal Candidates Extraction

After extracting a set of clock gating control signal candidates as shown in Table 2, we define the condition constraints in the optimization method according to Eq. 2 and Eq. 3 with variable z_i and y_j listed in each line and column respectively. As we explained in the previous section, for the 4-bit counter case, we have the following constraints

for each line:

$z_0 = 1$
$x_{10} + z_1 = 1$
$x_{20} + x_{21} + z_2 = 1$

and the following ones for each column:

if $x_{10} + x_{20} > 0$, then $y_0 = 1$
if $x_{21} > 0$, then $y_1 = 1$

Based on these constraints, we construct a BDD and compute minimum cost in our BDD package.

6. Experimental Results

We implemented the optimization method in C, tested on counter circuits with power simulation and applied to ISCAS89 benchmark circuits. In the power simulation for counter circuits, we use VDEC 0.18µm library as technology library and Synopsys Design Compiler as synthesis tools. All experiments were done on 2.66Ghz x64 machines.

Table 3 Optimization Results and Power Consumption for Counter Circuits.

			Dynamic Power After Synthesis		
Bit	Min Cost	Min-Cost Grouping	Dynamic Power of Original Counter	Dynamic Power with CG	Power Reduction
8	4.23	5 2 1	38.7	24.2	37.3%
10	4.48	6 3 1	45.4	25.2	44.6%
	4.48	7 2 1		25.2	44.6%
16	4.69	11 3 2	65.9	25.8	60.8%
20	4.82	14 4 2	79.6	26.4	66.8%
	4.82	15 3 2		26.4	66.8%
30	4.98	24 4 2	114.0	26.7	76.6%

Table 3 shows the optimization results and corresponding power consumption for counter circuits after logic synthesis. Column 1 shows the bit-width of the counter circuits. Column 4 presents the dynamic power consumption of the original counter circuits, while column 5 shows the dynamic power consumption after clock gating applied. Column 3 presents the circuit structure with minimum cost. For example, (11 3 2) is the optimum at 16-bit counter circuit. This means upper 11 registers (r15-r5) are controlled as one group and the next 3 registers (r4-r2) are controlled as another group. The lower 2 registers (r1r0) remain no control. By the experiments, we confirmed that the evaluation method with switching activity has the same tendency with the power estimation after logic synthesis of no concern with wire-load and buffers. On 8-bit to 30-bit counter circuits, 37.3% to 76.6% power reduction has been found.

We also applied our method to ISCAS89 (s344 ~ s1512) benchmark circuits. Table 4 shows the optimization results. Columns 1 and 2 show the name and the number of flip-flops of a benchmark circuit. Column 3 presents the number of product terms in the BDD's. Column 4 and 5 show the optimum costs after clock gating being applied based on our optimization algorithm and their corresponding reduction compared with that without clock gating. The cost reduction reaches from 2.3% to 18.0% for ISCAS89 benchmark circuits using our optimization method.

Table 4 Optimization Results for Benchmark Circuits.

Circuit	# F.F.s	# Product Terms in BDD's	Optimum Cost	Cost Reduction
s344 / s349	15	922	12.3	18.0%
s526/s526n	21	512	20.1	4.5%
s382 /s400	21	292	20.5	2.3%
s444	21	236	20.5	2.3%
s1269	37	730	33.2	10.3%
s1512	57	1698	54.7	4.0%
AVG	29	732	26.9	6.9%

7. Conclusions

In the paper we focus on automatic clock gating generation and propose an optimization algorithm through power-optimal control signal selection based on BDD. The method includes two phases, gating control signal candidates extraction and power-optimal control signal selection. Since the inserted clock gating element itself causes extra power dissipation, the sharing of control signals by different registers has been taken into consideration for power optimization. By applying to counter circuits and a set of benchmark circuits, the minimum cost has been obtained. Power simulation has been implemented for counter circuits which confirmed the co-relation with our method. On counter circuits, 37.3% to 76.6% power reduction has been found. And for benchmark circuits, 2.3% to 18.0% cost reduction has been reached.

Acknowledgments

We would like to thank to Professor Takeshi Yoshimura and Professor Takahiro Watanabe as well as all members of Kimura lab. of Waseda University for their comments and discussions. The work is supported in part by CREST ULP Project of JST and by Waseda University Ambient Global COE Program of MEXT. The work is also supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and Kyoto University.

References

[1] J. Montiero, S. Devadas and A. Gosh, "Retiming Sequential Circuits for Low Power", Proc. of ICCAD, pp. 398-402, Nov. 1993.

[2] L. Benini and G. De Micheli, "State Assignment for Low Power Dissipation", IEEE J. Solid State Circuits, vol. 30, no. 3, pp. 258-268, March 1995.

[3] L. Benini, and G. De Micheli, "Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines", IEEE Trans. on CAD, vol. 15, no. 6, pp. 630-643, June 1996.

[4] H. Kapadia, L. Benini, and G. De Micheli, "Reducing Switching Activity on Datapath Buses with Control-Signal Gating", IEEE J. Solid-State Circuits, vol. 34, no. 3, pp. 405 – 414, March 1999.

[5] M. Onishi, A. Yamada, H. Noda, and T. Kambe, "A Method of Redundant Clocking Detection and Power Reduction at RT Level Design", Proc. of ISLPED, pp. 131-136, Aug. 1997.

[6] L. Benini, G. De Micheli, E. Macii, M. Poncino, and R. Scarsi, "Symbolic Synthesis of Clock-Gating Logic for Power Optimization of Synchronous Controllers", ACM Trans. on Design Automation Electronic Systems, vol. 4, no. 4, pp. 351-375, Oct. 1999.

[7] Y. Luo, J. Yu, J. Yang, and L. Bhuyan, "Low Power Network Processor Design Using Clock Gating", Proc. of DAC, pp. 13-17, June 2005.

[8] H. M. Jacobson, "Improved Clock-Gating through Transparent Pipelining", Proc. of ISLPED, pp. 26-31, Aug. 2004.

[9] N. Banerjee, K. Roy, H. Mahmoodi, and S. Bhunia, "Low Power Synthesis of Dynamic Logic Circuits Using Fine-Grained Clock Gating", Proc. of DATE, pp. 6-10, March 2006.

[10] P. Babighian, L. Benini, and E. Macii, "A Scalable Algorithm for RTL Insertion of Gated Clocks Based on Odcs Computation", IEEE Trans. on CAD, vol. 24, no. 1, pp. 29-42, Jan. 2005.

[11] Q. Wu, M. Pedram, and X. Wu, "Clock-Gating and Its Application to Low Power Design of Sequential circuits", IEEE Proc. of CICC, pp. 479-482, May 1997.

[12] Aaron P. Hurst, "Automatic Synthesis of Clock Gating Logic with Controlled Netlist Perturbation", DAC, pp. 654-657, June 2008.

[13] J. Chen, X. Weil, Y. Jiang and Q. Zhou, "Improve Clock Gating Through Power-Optimal Enable Function Selection", DDECS, pp. 30-33, April 2009.

[14] Randal E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation", IEEE Trans. on CAD, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[15] Randal E. Bryant, "Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams", ACM Computing Surveys, vol. 24, no. 3, pp. 293-318, Sep. 1992.