

# A Method of Redundant Clocking Detection and Power Reduction at RT Level Design

Mitsuhisa Ohnishi, Akihisa Yamada, Hiroaki Noda, and Takashi Kambe

Precision Technology Development Center, Sharp Corporation  
2613-1, Ichinomoto, Tenri, Nara 632, Japan

## Abstract

This paper proposes a novel method to estimate and to reduce redundant power of synchronous circuits at RT level design. Because much redundant power is caused by redundant clockings which activate registers unnecessarily, we detect these clockings. They are detected from the difference of the numbers of incoming and outgoing data of a register. And then we introduce gated-clock scheme to reduce the power consumption of the circuits using our estimation results. Experimental results demonstrate the accuracy of our method and the effect on power reduction.

## 1 Introduction

It is very important for low power design to analyze the power consumption of a circuit. Several power estimation techniques for CMOS digital circuits have been proposed[2–9]. These techniques estimate the power consumption of the whole circuits. However, they cannot distinguish redundant behaviors from essential ones in circuits. Therefore, an LSI designer cannot obtain the information about which part of the circuit behaves redundantly and how much power he can reduce, although such information is very useful for low power design.

In this paper, we propose a novel method which identifies and reduces **redundant clockings**. Since these clockings activate registers unnecessarily, they are critical issue for low power design. A clock signal charges and discharges large wire load capacitance and internal capacitance of register cells at high frequency. Furthermore, the output of a register causes switchings in the consequent circuits. If the power consumed by redundant clockings is estimated, an LSI designer can know how much power each register wastes and apply low power techniques such as gated-clock scheme[11] or multi-phase clock scheme.

Hereafter, when a register  $A$  feeds data to a register  $B$ ,

we refer to register  $A$  as “source register” of register  $B$ , and register  $B$  as “destination register” of register  $A$ .

In our method, redundant clockings are detected from the difference of the number of data transferred from source registers and that of data transferred to destination registers. We regard the number of times when a condition of data-transfer becomes true as the number of transferred data. The conditions of data-transfer among registers are extracted by analyzing RT level HDL descriptions statically. Then, we dynamically count how many times the conditions are satisfied during RT level simulation.

This paper is organized as follows. In section 2, we describe redundant behaviors of a register and the basic ideas to detect redundant clockings. In section 3, we present an algorithm to estimate the power consumed by redundant clocking. In section 4, we discuss how to reduce the power consumption of the circuit using estimation results. We have experimental results to show effectiveness in section 5, and we have conclusions in section 6.

## 2 Basic Ideas

In this section, we show the basic ideas to detect redundant clockings. The redundant clockings occur when a register stores excessive data from source registers or feeds excessive data to destination registers. In order to detect the redundant clockings, we focus on the difference of the numbers of incoming and outgoing data of a register, or the balance of the numbers of incoming data and clockings for a register.

We define three types of redundant behaviors of a register which are caused by redundant clockings.

**Unused data latching:** if a register stores excessive data, some data are not transferred to any destination registers. We call this behavior as unused data latching.

**Unchanged data latching:** if source registers of a register feed data which is not updated, the register stores the same data that has already stored in itself. We call this behavior of the register as unchanged data latching.

**Redundant data holding:** if a register does not store data incoming from source registers in a certain clock cycle, the register stores data from itself. We call this behavior as redundant data holding.

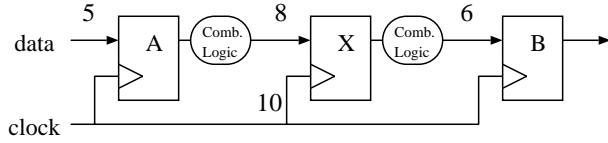


Figure 1: An example RT level circuit.

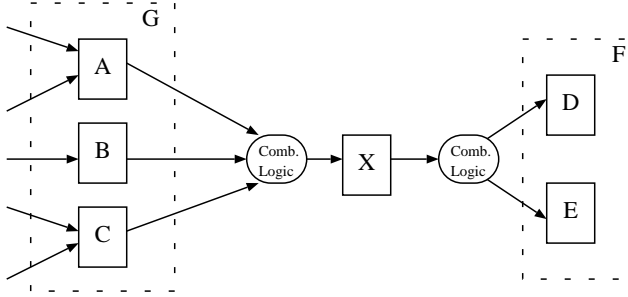


Figure 2: Multiple source registers and multiple destination registers for register  $X$ .

First, let us consider a register which has a single source register and a single destination register. Fig. 1 shows an example circuit. Registers  $A$ ,  $B$ , and  $X$  are driven by signal *clock*. We assume that the numbers of data-transfers during 10 clock cycles are as follows:

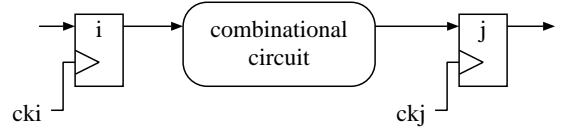
- 5 data are transferred to  $A$ ,
- 8 data are transferred from  $A$  to  $X$  through some combinational circuits,
- 6 data are transferred from  $X$  to  $B$  through some combinational circuits.

We focus on register  $X$ . The number of data incoming from  $A$  to  $X$ , eight, is larger than the number of data outgoing from  $X$  to  $B$ , six. We identify two *unused data latching*s of  $X$ . The number of clockings for  $X$ , ten, is larger than the number of data incoming to  $X$ . We also identify two *redundant data holdings* of  $X$ . Consider register  $A$ . The number of data incoming to  $A$ , five, is smaller than the number of data outgoing from  $A$ , eight. We identify three *unchanged data latching*s of  $X$  by these two numbers for  $A$ .

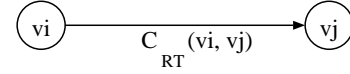
Next, let us consider a more complex case where there are multiple source registers and/or multiple destination registers. In that case, we treat the multiple source registers or the multiple destination registers as a pseudo-register.

An example of complex circuit is shown in Fig. 2. For register  $X$ , we introduce two pseudo-registers  $F$  and  $G$ . We make assumptions as follows:

- The number of data incoming from  $G$  to  $X$  is the sum of the numbers of data from  $A$ ,  $B$ , and  $C$ .
- If  $A$ ,  $B$ , and/or  $C$  stores data in a certain clock cycle,  $G$  is updated.



(a)



(b)

Figure 3: An example circuit. (a) RT level circuit. (b) Data transfer graph.

- If  $D$  and/or  $E$  stores data from  $X$  in a certain clock cycle, one data of  $X$  is transferred to  $F$ .

On these assumptions, we treat the case of multiple source registers and/or multiple destination registers in the same way as the case of a single source register and/or a single destination register.

### 3 Algorithm

In this section, we describe our algorithm to count the numbers of data-transfers among registers in detail. At first, we define data-transfer conditions which become true when data-transfers from/to the register occur. They are extracted by analyzing RT level HDL descriptions, statically. Then, we count the numbers of times when these conditions become true. These numbers are counted in RT level simulation, dynamically.

Extraction of the data-transfer conditions is shown in section 3.1. Estimation methods of the number of redundant clockings and redundant power are described in section 3.2 and 3.3, respectively. The outline of our algorithm is described in section 3.4.

#### 3.1 Extraction of the Data Transfer Conditions

We define *data transfer graph* (DTG) to capture the relationship of data-transfer among registers on data-path. A data transfer graph is a directed graph as shown in Fig. 3. A node  $v_i$  represents a register  $i$  in the circuit. A directed edge  $(v_i, v_j)$  exists if and only if data is transferred from a register  $i$  to a register  $j$  through only combinational circuits. We treat a primary input as a register which feeds data in every clock cycle, and a primary output as a register which stores data in every clock cycle.

Each edge  $(v_i, v_j)$  has a data-transfer condition  $C_{RT}(v_i, v_j)$ .  $C_{RT}(v_i, v_j)$  becomes true when a data-transfer occurs between register  $i$  and  $j$ , and is extracted from the given HDL description. We show an example of VHDL description in Fig. 4. The description means

```

if(ck'event and ck = '1') then
  if(cond = '1') then
    j <= i;
  end if;
end if;

```

Figure 4: An example of VHDL description (we assume that  $i$  and  $j$  are registers).

“when signal  $cond$  equals to '1' at the rise-edge of signal  $ck$ , a value of  $i$  is assigned to  $j$ .” Then, the data-transfer condition  $C_{RT}(v_i, v_j)$  is represented by (1).

$$C_{RT}(v_i, v_j) = (ck'event \wedge ck = '1') \wedge (cond = '1') \quad (1)$$

We represent  $\wedge$  as logical product (AND) and  $\vee$  as logical sum (OR) throughout the paper.

In addition, each node  $v_i$  has three types of data-transfer conditions,  $C_{LAT}(v_i)$ ,  $C_{USED}(v_i)$ , and  $C_{CHG}(v_i)$  as shown in Fig. 5. These conditions are defined as follows.

$C_{LAT}(v_i)$  is a condition of data-transfer between a register  $i$  and one or more source registers of register  $i$ . Let registers  $1, 2, \dots, m$  be source registers of register  $i$ , then  $C_{LAT}(v_i)$  is represented by (2).

$$C_{LAT}(v_i) = \bigvee_{r=1}^m C_{RT}(v_r, v_i) \quad (2)$$

$C_{USED}(v_i)$  is a condition of data-transfer between a register  $i$  and one or more destination registers of register  $i$ . Let registers  $1, 2, \dots, n$  be the destination registers of  $i$ , then  $C_{USED}(v_i)$  is represented by (3).

$$C_{USED}(v_i) = \bigvee_{r=1}^n C_{RT}(v_i, v_r) \quad (3)$$

$C_{CHG}(v_i)$  is a condition of data-transfer to one or more source registers of a register  $i$ . Let registers  $1, 2, \dots, k$  be the source registers of register  $i$ , then  $C_{CHG}(v_i)$  is represented by (4).

$$C_{CHG}(v_i) = \bigvee_{r=1}^k C_{LAT}(v_r) \quad (4)$$

### 3.2 Estimation of the Number of Redundant Clockings

In this section, we describe how to estimate the number of redundant clockings.

Let  $A_{CK}(v_i)$  be the number of clockings for a register  $i$ ,  $A_{LAT}(v_i)$ ,  $A_{USED}(v_i)$ , and  $A_{CHG}(v_i)$  be the numbers of times when  $C_{LAT}(v_i)$ ,  $C_{USED}(v_i)$ , and  $C_{CHG}(v_i)$  become true, respectively. These numbers are profiled during RTL simulation.

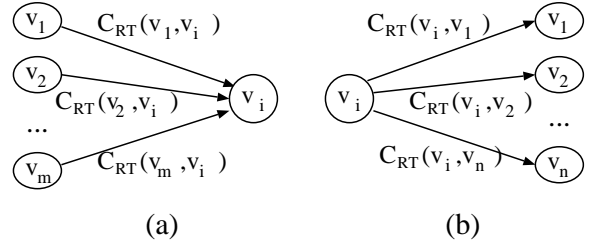


Figure 5: Examples of data-transfer conditions. (a)  $C_{LAT}$ . (b)  $C_{USED}$ . (c)  $C_{CHG}$ .

$A_{HOLD}(v_i)$ ,  $A_{UU}(v_i)$ , and  $A_{UC}(v_i)$  are the numbers of behaviors of *redundant data holding*, *unused data latching*, and *unchanged data latching* of a register  $i$ , respectively.  $A_{HOLD}(v_i)$  is estimated using (5).

$$A_{HOLD}(v_i) = A_{CK}(v_i) - A_{LAT}(v_i) \quad (5)$$

$A_{UU}(v_i)$  and  $A_{UC}(v_i)$  are estimated using (6) and (7), respectively.

$$A_{UU}(v_i) \geq A_{LAT}(v_i) - A_{USED}(v_i) \quad (6)$$

$$A_{UC}(v_i) \geq A_{LAT}(v_i) - A_{CHG}(v_i) \quad (7)$$

Equality of (6) is satisfied when the destination register of register  $i$  always receives updated data. If some data incoming to register  $i$  are transferred to the destination register more than one time, the right side of (6) underestimates  $A_{UU}(v_i)$ .

Equality of (7) is satisfied when all data incoming to the source register of register  $i$  are transferred to register  $i$ . If some data incoming to the source register are not transferred to register  $i$ , the right side of (7) underestimates  $A_{UC}(v_i)$ .

### 3.3 Estimation of the Power Consumed by Redundant Clocking

We estimate the redundant power which is caused by the redundant clocking.

The power consumption of CMOS circuits is denoted as follows:

$$P = nC_LV^2 \quad (8)$$

$$n = \alpha f, \quad (9)$$

where  $P$  is power consumption,  $V$  is supply voltage,  $C_L$  is load capacitance,  $\alpha$  is switching rate, and  $f$  is clock frequency[1].

Let  $P_{HOLD}(v_i)$ ,  $P_{UU}(v_i)$ , and  $P_{UC}(v_i)$  be power consumed by redundant behaviors *redundant data holding*, *unused data latching*, and *unchanged data latching* of a register  $i$ , respectively. Then, they are estimated as follows:

$$P_{HOLD}(v_i) = A_{HOLD}(v_i) \cdot L_{CK}(v_i) \cdot V^2, \quad (10)$$

$$P_{UU}(v_i) = P_{UU_{CK}}(v_i) + P_{UU_{FUNC}}(v_i) \quad (11)$$

$$= (A_{UU}(v_i) \cdot L_{CK}(v_i) + \frac{1}{2} \cdot A_{UU}(v_i) \cdot L_{FUNC}(v_i)) \cdot V^2, \quad (12)$$

$$P_{UC}(v_i) = A_{UC}(v_i) \cdot L_{CK}(v_i) \cdot V^2. \quad (13)$$

In equations (10)–(13),  $CK(v_i)$  is the clock driving register  $i$ .  $P_{UU_{CK}}(v_i)$  and  $P_{UU_{FUNC}}(v_i)$  are the power consumed by the clock net and the power consumed by the consequent combinational circuit of register  $i$  when *unused data latching* occurs, respectively.  $L_{CK}(v_i)$  is the load capacitance which is charged and discharged when the clock signal  $CK(v_i)$  changes.  $L_{FUNC}(v_i)$  is the load capacitance which is charged and discharged when the output of register  $i$  changes. In (12), we assume that the output of a register changes from '1' to '0'  $n/2$  times during  $n$  clock cycles.

### 3.4 Outline of Our Method

We show the outline of our algorithm.

1. Extraction of the conditions,  $C_{RT}$ ,  $C_{LAT}$ ,  $C_{USED}$ , and  $C_{CHG}$  for each register.  $C_{RT}$  is extracted by analyzing HDL description statically.  $C_{LAT}$ ,  $C_{USED}$ , and  $C_{CHG}$  are represented, as we have seen in section 3.1.
2. Profiling  $A_{LAT}$ ,  $A_{USED}$ , and  $A_{CHG}$  of each register, the numbers of times when three conditions become true during RT level simulation, dynamically.  $A_{CK}$ , the number of clocking for each register, is also profiled.
3. Estimation of the number of redundant behaviors of each register,  $A_{HOLD}$ ,  $A_{UU}$ , and  $A_{UC}$  from the numbers of data-transfers.
4. Estimation of the power consumed by redundant clockings,  $P_{HOLD}$ ,  $P_{UU}$ , and  $P_{UC}$ .

## 4 Power Reduction

Gated-clock scheme is one of solutions for low power design. Although it sometimes causes clock skew problem in timing design phase, it is still used widely for low power design of synchronous circuits due to its effectiveness. We adopt this scheme to eliminate redundant clocking.

Since modification of the clocking scheme for all registers wasting power requires long redesign time and much effort, we select registers for which clocking should be modified in the following way:

1. Record the clock cycle in which each register behaves redundantly during RT level simulation as follows:

```

if(ck'event and ck = '1') then
  if(cond = '1') then
    j <= i;
  end if;
end if;

```

(a)

```

gck <= ck and cond;
if(gck'event and gck = '1')then
  j <= i;
end if;

```

(b)

Figure 6: (a) Example description of single-clock scheme. (b) Example description of gated-clock scheme.

- i) Calculate the sum of  $A_{HOLD}$ ,  $A_{UU}$ , and  $A_{UC}$  for each register in every clock cycle.
- ii) Record  $t$  for each register if and only if the sum in clock cycle  $t$  is larger than the old sum in clock cycle  $t-1$ , since a redundant clocking for the register is detected in clock cycle  $t$ .

2. Group registers which behave redundant behaviors similarly as follows:

```

foreach  $i$  (registers which do not belong to any groups)
{
  Let register  $i$  belong to a new group  $G_i$ .
  foreach  $j$  (registers which do not belong to any groups)
  {
    Count the number of clock cycles in which both
    registers  $i$  and  $j$  behave redundantly.
    if (counted number > given threshold)
    {
      Let register  $j$  belong to group  $G_i$ .
    }
  }
}

```

3. Calculate the total redundant power for each group.
4. Select groups whose total redundant powers are more than a given threshold power. They are targets of modification of the clocking scheme for power reduction.

We introduce a single gated-clock for each selected group. An enabling condition for a gated-clock is derived from conditions of data-transfer for each register in the group. An example HDL descriptions are shown in Fig. 6. The condition of data-transfer for register  $j$  is  $cond = '1'$ . We assume that a 2-input-AND gate is used for gating clock  $ck$ . Then the enabling signal for gated clock signal  $gck$  is  $cond$ .

Because gated-clock scheme requires additional circuits for enabling clock signal, it may cause overheads of area, delay and power. In practical design, trade-off between overheads and power reduction by the optimization should be

Table 1: Dimensions of circuit A.

# of registers	# of gates	clocking scheme
47 (227 bits)	2,430	single clock

Table 2: Estimation results of circuit A.

Total power	78.6 mW
Redundant power	31.2 mW

considered. When the power reduction for a register is small, the clocking scheme for the register should not be modified.

Estimation results depend on given test patterns for RT level simulation. Consider a register which wastes power in simulation with a given test pattern. The functionality of the circuit is not changed by introducing gated clock scheme for the register, because we derive an enabling signal for a gated-clock from conditions of data-transfer. However, to reduce effort to redesign, test patterns simulating the actual behaviors of the circuit should be used for RT level simulation.

## 5 Experimental Results

We have developed a power analysis system for RT level circuits, and applied it to two example circuits. Experimental results demonstrate that our method can precisely estimate the power which can be reduced, and that information about redundant clocking is useful for low power design.

### 5.1 Example Circuit A

Circuit A is a part of a video signal processor. Its dimensions are shown in Table 1. The registers in the circuit consist of various numbers of edge-triggered flip-flops, from 1 up to 32 bits. All registers are driven by a single clock signal.

#### 5.1.1 Power Estimation of Circuit A

We use a commercial CAD tool to estimate the total power consumption of the whole circuit. Load capacitances  $L_{CK(v_i)}$  and  $L_{FUNC(v_i)}$  described in section 3.3 are also calculated by the tool.

The result of the power estimation is shown in Table 2. Using our method (in section 3), the redundant power consumed by redundant clockings is estimated as 31.2mW, which is about 40% of the total power.

Table 3 shows the distribution of registers which behave redundantly. The first column shows percentages of redundant clockings. The other columns show the numbers of registers which behave *redundant data holding*, *unused data latching*, and *unchanged data latching*, respectively. It shows that 25 registers are identified as *redundant data holding* registers during over 90% of the whole clock cycles. It is also shown that 5 registers store unchanged data during over 90% of the whole clock cycles.

Table 3: Distribution of redundant behavior of registers.

%	$A_{HOLD}$	$A_{UU}$	$A_{UC}$
91~100	25	0	5
81~90	0	0	0
71~80	1	0	0
61~70	0	0	0
51~60	0	0	0
41~50	0	0	0
31~40	0	0	0
21~30	0	0	0
11~20	1	3	0
1~10	1	0	0
total	27	3	5

Table 4: The number of registers, flip-flops, and the total redundant power of each selected groups.

	#registers	#flip-flops	redundant power
Group1	4	36	5.19 mW
Group2	2	18	5.19 mW
Group3	4	8	2.30 mW
Group4	3	6	1.84 mW
Group5	1	6	1.84 mW
Group6	1	6	1.84 mW
Group7	1	6	1.84 mW
Group8	1	6	1.80 mW
Group9	1	6	1.77 mW
Group10	2	2	1.54 mW
Group11	1	4	1.15 mW
Group12	1	1	0.31 mW
Group13	1	1	0.31 mW
total	23	106	26.9 mW

#### 5.1.2 Power Reduction of Circuit A

Using our method (in section 4), 13 groups are selected as targets of modification when the threshold of total redundant power of a group is set to 0.3mW. The number of registers and the total redundant power in each group are shown in Table 4. The estimated total redundant power of all selected registers is 26.9mW. We appended 13 gated-clocks into HDL descriptions to drive these registers, manually. Then we reduced 28.9mW which is 37% of the total power as shown in Table 5. This result shows that our method can estimate redundant power accurately and reduce the power consumption efficiently.

The number of gates of the modified circuits is 2,387, which is smaller than that of the original circuit. One of the reasons for the reduction in gate count is that individual control circuits for data-transfer for all flip-flops in a group are replaced with a single enabling circuit for a gated-clock.

### 5.2 Example Circuit B

Circuit B is another part of the video signal processor described in section 5.1. Its dimensions are shown in Table 6.

Table 5: Redundant power of 23 registers and reduced power.

Estimated power	26.9 mW
Reduced power	28.9 mW

Table 6: Dimensions of circuit *B*.

# of registers	# of gates	clocking scheme
180 (1,789 bits)	15,626	single clock

This circuit operates in two modes, recording and playback.

The power consumption of the whole circuit and the redundant power, which estimated using our method, are shown in Table 7. Table 7 shows that about 25% of the total power is consumed by redundant clockings in both modes. In this circuit, the redundant clockings are detected at 117 of 180 registers.

Gated-clock scheme described in section 4 is used for low power. We selected 66 registers out of 180 registers as the target of modifying. The redundant power of the selected registers and reduced power by using gated-clock scheme are shown in Table 8. We reduced 37mW in recording mode and 35mW in playback mode. The power reductions in each mode are 29% and 27% of the total powers, respectively.

In this case, the reduced power is larger than the estimated one. Recall our earlier example circuit in Fig. 1. If data is not transferred from *X* to *B* in a clock cycle *t*, one *unused data latching* of *X* is identified. However, a clocking for *A* in clock cycle *t* – 1 is also redundant clocking. In our algorithm, we do not identify this behavior of *A* as *unused data latching*. In the experiment, we modified HDL description of clockings for registers *A* based on the information about the redundant behavior of *X*.

## 6 Conclusion

We have proposed a method to detect redundant clocking for registers such as *redundant data holding*, *unused data latching*, and *unchanged data latching* in an RT level circuit. In order to detect the redundant clockings, the number of data-transfers is profiled by using RT level simulation techniques and the power is estimated. In the experiment, we have estimated the wasted power in circuits. We have obtained 27 – 37% power reduction by introducing gated-clock scheme using the information about redundant clockings. Our experimental results show that our method can estimate redundant power accurately.

## References

[1] A. P. Chandrakasan, R. W. Brodersen, “LOW POWER DIGITAL CMOS DESIGN,” *Kluwer Academic Publishers*, 1995.

Table 7: Estimation results of circuit *B*.

mode	recording	playback
Total power	127 mW	128 mW
Redundant power	33 mW	35 mW

Table 8: Redundant power of 66 registers and reduced power.

mode	recording	playback
Estimated power	28 mW	29 mW
Reduced power	37 mW	35 mW

- [2] M. A. Cirit, “Estimating Dynamic Power Consumption of CMOS Circuits,” in *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 534–537, Nov. 1987.
- [3] F. N. Najm, “Transition Density, A Stochastic Measure of Activity in Digital Circuits,” in *Proc. 28th ACM/IEEE Design Automation Conference*, pp. 644–649, June 1991.
- [4] P. E. Landman, J. M. Rabaey, “Black-Box Capacitance Models for Architectural Power Analysis,” in *Proc. Int. Workshop on Low Power Design*, pp. 165–170, 1994.
- [5] D. Liu and C. Svensson, “Power Consumption Estimation in CMOS VLSI Chips,” *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 6, pp. 663–670, June 1994.
- [6] Y. Uchimura, Y. Okuno, K. Kaneko, “Power Consumption Estimator by Logic Simulation (in Japanese),” in *Proc. The 5th Karuizawa Workshop on Circuits and Systems*, pp. 273–277, Apr. 1992.
- [7] T. Sato, Y. Ootaguro, M. Nagamatsu, H. Tago, “Architectural-level Power Estimation for CMOS RISC Processors (in, Japanese),” *Technical Report of IEICE*, VLD95-112, pp. 71–76, Dec. 1995.
- [8] F. N. Najm, “A Survey of Power Estimation Techniques in VLSI Circuits,” *IEEE Trans. Very Large Scale Integration Systems*, Vol. 2, No. 4, pp. 446–455, Dec. 1994.
- [9] K. Keutzer, “The Impact of CAD on the Design of Low Power Digital Circuits,” in *Proc. IEEE Symposium on Low Power Electronics*, pp. 42–45, 1994.
- [10] V. Tiwari, S. Malik and P. Ashar, “Guarded Evaluation: Pushing Power Management to Logic Synthesis/Design,” in *Symposium Proceedings of Int. Symposium on Low Power Design*, pp. 221–226, 1995.
- [11] T. Ishihara and H. Yasuura, “Some Experimental Results on Low Power Design with Gated Clock (in Japanese),” *Technical Report of IEICE*, VLD95-116, pp. 7–12, Dec. 1995.