A Variation-Aware Low-Power Coding Methodology for Tightly Coupled Buses

Muroyama, Masanori
Department of Computer Science and Communication Engineering, Kyushu University

Tarumi, Kosuke
Department of Computer Science and Communication Engineering, Kyushu University

Makiyama, Koji
Department of Computer Science and Communication Engineering, Kyushu University

Yasuura, Hiroto
Department of Computer Science and Communication Engineering, Kyushu University

出版情報：ASP-DAC2005, pp.557-560, 2005-01. IEEE
バージョン：
権利関係：

A Variation-Aware Low-Power Coding Methodology for Tightly Coupled Buses

Masanori Muroyama, Kosuke Tarumi, Koji Makiyama and Hiroto Yasuura
Department of Computer Science and Communication Engineering, Kyushu University, Fukuoka, Japan
{muroyama, tarumi, makiyama, yasuura}@c.csce.kyushu-u.ac.jp

Abstract - This paper describes a novel low-power coding methodology for buses. Ultra deep submicron technology and system-on-chip have resulted in a considerable portion of power consumption on buses, in which the major sources of the power consumption are the transition activities on the signal lines and the coupling capacitances of the lines. In addition, we enter an era of considering variation of the effective coupling capacitances. We address power reduction including these phenomena by using variable length coding. Experimental results show the effectiveness of our methodology.

I. INTRODUCTION

With advancing technology process, power-aware design has become a driving force in the semiconductor industry. As CMOS processes scale to submicron dimensions, power associated with system buses and the I/O accounts for a large portion of the total system power. Reducing the power consumption in busses is a key issue to reduce the communication cost. Many encoding schemes have been studied to reduce the power dissipation on buses [1-9]. Coupling has become an important issue with scaled supply voltage when we consider signal integrity and power dissipated by coupling capacitances, referred to as coupling power. Shielding, spacing and swapping [6-8] can be ways to avoid coupling effects problem. In recent technologies, in addition to the coupling capacitance, the variability of circuit delay due to process variations has become a significant concern. As process geometries continue to shrink, the ability to control critical device parameters is becoming increasingly difficult. With increasing awareness of process variations, a number of techniques are developed [10-12]. However, most of those focus on timing analysis. This means that only timing variations due to process variations for critical path are considered. We use coupling variation due to arrival time variation for power analysis. Actually, for on-chip bus, variation of delay skew between neighbor bit lines has the potential to raise the effective coupling capacitance variation [15]. In this paper, we include power variation due to the delay skew variation in our power estimation.

Let us consider a bus coding system. A sender sends encoded data to a receiver through a bus. After receiving the data from the sender, the receiver decodes the encoded data. There are many purposes of coding, for example, which are to realize high tolerant or high performance. In this paper, we especially aim to realizing low-power with coding. The Bus-Invert method [1] and Coupling-Driven Bus-Invert [2] can be applied to encode buses without prior knowledge of data statistics. On the other hand, encoding methods considering highly correlated access patterns like address buses, the T0 method [3] and working zone method [4], or like data buses for microprocessors [5] have been proposed. Our proposed method here aggressively uses probabilistic information of input vector of buses. The basic concept proposed is based on variable length coding. Generally, variable length coding is used for data compression. We use this coding for power reduction. Some low-power coding [3-5] can be considered as variable length coding. In [3,4], scope of application is very limited such as address buses. In [5], the techniques aim to reduce only self-capacitances and are not suited for on-chip buses in terms of implementation costs. The contributions of this paper are as follows. First, for more accuracy, we propose novel power estimation including the effective coupling capacitance variation. Second, we make positive use of variable length coding with little overheads towards deep-submicron era.

The remainder of the paper is organized as follows. In Section II, along with some basic definitions, we present our power model including coupling effects considering delay skew variation. Section III gives an overview of our methodology to achieve low-power requirements. Section IV describes the experimental setup and presents the results. Section V concludes this paper.

II. POWER MODEL

In this section, we describe the power model used to estimate. In terms of physical implementation, buses can be divided into two types. The first type is off-chip implementation. In this case, the physical capacitances of a wire are only *self-capacitances Cs*. The second is on-chip one. The physical capacitances include *coupling capacitances Cx* depending on relative voltage swing between two adjacent bus lines in addition to the *Cs*. That is, those power models differ from each other. We consider all the types, because off-chip and on-chip buses are both key components of systems.

The power consumption in CMOS circuits are contributed from three parts, which are power consumed by the leakage current, by the short current during the switching and by the charging-discharging current for nodes. We focus on the power consumption relevant to the charging-discharging current, which is the dominant part in the total power consumption. Accordingly, the total dynamic power consumed by an *N*-bit bus is given by:

$$P_{bus} = \sum_{i=0}^{N-1} P_{Di},$$

where P_{Di} denote the dynamic power consumption of a bit line *i*. P_{Di} is defined as follows:

$$P_{Di} = 0.5 \cdot \left(\alpha_{Si} C_S + \alpha_{Xi} C_X\right) \cdot V_{DD}^2 \cdot f,$$

where α_{Si} and α_{Xi} denote the rates at which each capacitance is switched per one clock cycle for a bit line *i*. V_{DD} and f represent supply voltage and frequency, respectively. In cases of off-chip bus and on-chip bus in non deep sub-micron design, C_X is nearly 0.

We also define the capacitance ratio $\lambda = {}^{C_X}\!/_{C_S}$ as a weight coefficient. λ is dependent on process technologies such as interconnect width, pitch, aspect ratio and dielectric thickness. λ increases as technology shrinks towards deep sub-micron [13]. λ is about 3 for 0.18 um CMOS technology with the minimum distance between wires.

A. Effective Coupling Capacitance Estimation Considering Delay Skew Variation

The effect of coupling capacitance (C_{eff}) is no longer a constant value. There are four types of possible transitions between two adjacent lines as follows: no switching (type A), single line switching (type B), both line switching to the same states (type C) and both line switching to the opposite states (type D). In type A, no dynamic charge distribution takes place. In type B, C_{eff} is C_X. In type C and D, the effective coupling capacitance depends on signal activities of neighboring lines due to the Miller effect [14]. The Miller effect states that if two lines switch in opposite directions, the effective coupling capacitance between them is $2C_X$ because the effective voltage swing between them is doubled. On the contrary, the effective coupling capacitance becomes 0 if both lines switch in the same directions. However, when both neighboring lines switch out of synchronous (have a skew), the effect of C_X changes. If two input signals switch in opposite directions with a skew, the

effective coupling capacitance is less than $2C_X$. Contrary, in same directions, that is more than 0. Many coupling-aware power reduction methods ignore a skew between two input signals. As a matter of fact, two signals may have a skew due to layout issues, circuit designs or process variations. In this paper, we consider delay skew effects between neighboring input signals (Fig. 1).

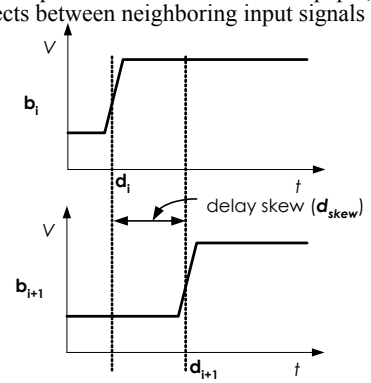

Figure 1: A delay skew

Here, we introduce how we consider coupling effect due to voltage swing between neighboring lines with a skew. If there is a sufficient skew between the bus lines, C_{eff} is equal to C_X in either opposite or same directions. We assume that a skew time T is enough for avoiding the Miller effect. T is determined by the input waveform, driving force of buffers and the output [15]. We assume following approximated equations of the effective capacitance:

$$C_{eff} = \begin{cases} 2C_X - (C_X / T)\cdot|d_{skew}|, & opposite, \ 0 \le |d_{skew}| \le T \\ (C_X / T)\cdot|d_{skew}|, & same, \ 0 \le |d_{skew}| \le T \\ C_X, & both, \ T < |d_{skew}| \end{cases} \quad (1),$$

where d_{skew} $(= d_{i+1} - d_i)$ represents an input delay skew of neighboring two lines as shown in Fig. 1. Figure 2 shows simulation results and estimated capacitance by equation (1). Simulation results are obtained by Spice simulator. C_X, T, clock frequency, transition time, and V_{DD} are 3pF, 3ns, 100MHz, 1ns and 1.8V, respectively. {R,R}, {F,F}, {R,F} and {F,R} in the figure stand for {rise, rise}, {fall, fall}, {rise, fall} and {fall, rise} transitions between neighboring b_i and b_{i+1} lines, respectively. Same and opposite in the figure represent switching in same and opposite directions, respectively. In addition, the values of same and opposite cases are calculated by equation (1). In cases of {R,R} and {F,F}, signals switch in same directions. In cases of {R,F} and {F,R}, those switch in opposite directions. As you can see in the figure, the estimated capacitance is quite corresponding to the results obtained by the circuit simulator. As delay skew increases, C_{eff} of all switching cases comes close each other (At delay skew 3ns, C_{eff} of all switching cases is 3pF, which is equal to C_X). For realizing high frequency, since driving force of buffers needs to be increased, C_{eff}'s sensitivity coefficient ($=C_X/T$) of delay skew also increases.

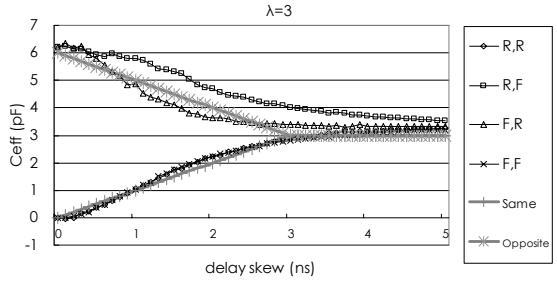

Figure 2: Effective coupling capacitance with a delay skew

Next, we will consider C_{eff} variation due to delay skew variation. Input signals consist of signal arrival time and the slope. We will focus on only variation of signal arrival time, because there is no effect of the signal slope for power consumed by the charging-discharging current. In this paper, signal arrival time is considered as normal distribution. Under this assumption, the delay skew of two neighboring lines has a normal distribution. A normal distribution in a variate X (delay skew is $X \cdot 10^{-9}$ sec) has the following probability function

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), (-\infty < x < \infty) \quad (2),$$

where μ and σ are the mean and standard deviation of normal distribution, respectively. Here we assume that μ is 0. For σ varying from 0 to 2.0, we calculate average C_{eff} by equation (1) considering delay skew variation based on equation (2) under previous experimental conditions. Results are shown in Fig. 3. As variation of delay skew increases, C_{eff} due to switching in same directions tends to be monotone increasing, and in opposite directions tends to be monotone decreasing. That means the following fact. For reducing the average effective coupling capacitance in same directions, high variability (large σ) is better. On the other hand, in same directions, low variability (small σ) is desired. When σ is 1.0, C_{eff} in same and opposite directions are 0.8 and 5.2pF, respectively. These values are used to evaluate our approach. In this manner, the impact of delay skew variation is not negligible.

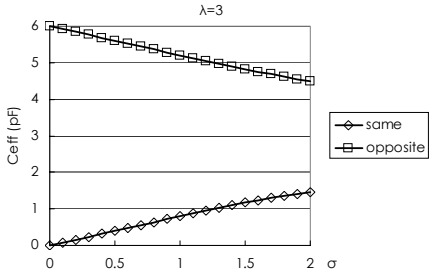

Figure 3: Average effective coupling capacitance considering delay skew variation

III. METHODOLOGY FRAMEWORK

A. Overview

As an overview, our proposed coding flow for power reduction is as follows:

1. We compress each N-bit data with variable length coding; the maximum bitwidth of compressed data is M ($>N$) and the minimum is 1.
2. The compressed data needs to be transferred via an M-bit bus even if the bitwidth is less than M. We encode remaining bits to reduce power consumption when the bitwidth of compressed data is less than M.

Applying these steps, we transfer compressed data with remaining low activity bits via an M-bit bus. Now, let us start with understanding of the flow in more detail. We first introduce some assumptions, notation and definition for discussion. We assume that an input vector must send from senders to receivers per one clock cycle. An original input vector $V_{org}^t = \left(o_{N-1}^t, o_{N-2}^t, ..., o_0^t\right)$, which is encoded to an input vector $V_{enc}^t = \left(e_{M-1}^t, e_{M-2}^t, ..., e_0^t\right)$ with our proposed encoding, is transferred through a bus $B = \left(b_{M-1}, b_{M-2}, ..., b_0\right)$, where t is the time index, N is the data width of original data, M is the encoded data width (equal to the bus width), and o_i^t and e_i^t are the value of a i-th bit at time t ($o_i^t, e_i^t \in \{0,1\}$). Orders of elements in B represent a fixed (physical) order of the bit lines of the buses. The bit lines are aligned physically by the orders from b_{M-1} to b_0. We encode original input vectors such that effective load capacitance is reduced. Next, we define an *active bit* and an *inactive bit*. Bits of compressed data are called as active bits, which are required to decode the compressed data. When the bitwidth of compressed data is less than M, remaining all bits except active bits are unnecessary, which are called as inactive bits.

B. Power Saving Mechanism

The basic concept proposed is similar to [5], which uses a variable length coding. They focus on only reducing C_S not including C_X. We utilize variable length coding for reducing power consumption including coupling power with delay skew variation in addition to C_S. They found the following characteristics based on observation of input vectors: quite long successive 0's string often appears. In consequence, encoding the length of successive 0's string, the 0's string part holds previous values to suppress switching. They seem to achieve to reduce off-chip power consumption. Since, for on-chip buses, hardware cost for encoding the length to mask active bits is inherently large, we take another variable length coding. We use string matching method as a variable length coding.

We define a matching vector $MV^j = \left(m^j_{N-1}, m^j_{N-2}, ..., m^j_0\right)$ ($1 \le j \le \omega$, $m^j_i \in \{0,1\}$) where ω is the number of matching vectors. If an original input vector V^t_{org} matches a matching vector, $e^t_i = e^{t-1}_i$ ($\forall i, M-N \le i \le M-1$) with index bits ($e^t_{M-N-1}, ..., e^t_0$). If not so, $e^t_i = o^t_i$ with the index bits. The index bits are used for specifying matching vectors matched or non-matching vectors. Note that inactive bits are e^t_i ($\forall i, M-N \le i \le M-1$) when an original vector matches matching vectors, meanwhile active bits are remaining bits. In this paper, inactive bits hold the last values just as [5] for easy implementation. In the parts of inactive bits, there are no self-switching and coupling switching in both same and opposite directions between adjacent bus lines. The bitwidth of index bits is $\lfloor log_2(1+\omega) \rfloor$ at least. The bit line e_{M-1}... e_0 are aligned physically by the orders from b_{M-1} to b_0 in terms of ordinality (see III-D.).

The above process is illustrated in the following example. Figure 4 shows the example. First, we explain the left example in the figure, where N=3, $V^{t-1}_{org} = (0,1,1)$, $V^t_{org} = (1,0,0)$, $\omega = 1$ and $MV = (1,0,0)$. At time t-1, since the original vector does not match the matching vector, an encoded vector (V^{t-1}_{enc}) consists of the original vector and an index bit where value of the index bit is 0 (indicates unmatched vector). On the other hand, at time t, the original vector matches the matching vector. Therefore, the last encoded vector except for the index bit and 1, which represents matching, make an encoded vector (V^t_{enc}). Finally, the original vectors are encoded as $V^{t-1}_{enc} = (0,0,1,1)$ and $V^t_{enc} = (1,0,1,1)$. We calculate the total effective capacitance according to parameters previously used, which is the sum of both self-capacitance and effective coupling capacitance. In type A, B, C and D transitions when self-capacitance is 1pF and σ is 1.0, the average effective coupling capacitances are 0pF, 3pF, 0.8pF and 5.2pF, respectively. In consequence, power consumption is reduced from 2.43mW to 0.648mW. A right example in the figure is another one to verify our asserting by using some input vectors. In this case, power consumption can also be reduced.

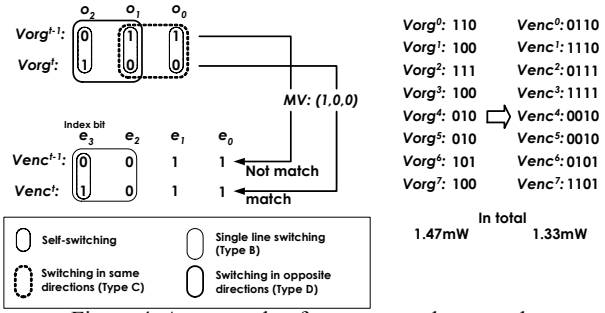

Figure 4: An example of our proposed approach

C. Matching Vector Analysis

In consideration of the following situation, our approach is suitable for narrow buses. First, the more N increases, the fewer matching ratio is. Second, the number of possible combination that needs to be evaluated for an N-bit bus is 2^N. Hence, it is hard to apply the matching vector analysis to 62- or 128-bit wide buses. Therefore, we consider narrow buses after partition the buses into some blocks. In this paper, we partition a 32-bit data bus into 16, 8, 4, 2 and 1 blocks that each block includes 2, 4, 8, 16 and 32-bit bus lines, respectively. We define each block bitwidth as bw. The following fact is obtained by simulation. In most cases, all bits of each block are 0. For example, when a data bus of an MPEG2 with a picture is traced for bw=2, 4, 8, 16 and 32, the appearance ratios of all 0 bit sequences are 76, 71, 63, 55 and 43%, respectively. These appearance ratios are extremely high. As a result of this fact, we use a matching vector $MV = (0_{bw-1}, 0_{bw-2}, ..., 0_0)$.

D. Coding Overhead

There are three factors of encoder overheads required for our approach. Those are a detector of matching vectors, multiplexers and registers. In a decoder, circuits needed are only multiplexers. Figure 5 shows block diagrams of implementation of the method for bw=4 and $\omega = 1$. The circuits are very simple. Bus coding inherently introduces area, delay, and power overheads due to the coding circuits. Since our implementation has low overheads, our approach is applicable for on-chip buses.

As you can see from the figure, there are strong symmetric properties for inputs of the bus lines. In consequent, delay skew between the bus lines has same variation as the variation of primary inputs of bus systems. When there are further more switching in same directions than opposite directions, to prevent the delay skew variation is desired in terms of power consumption (see Fig. 3).

In this paper, we do not swap wires to avoid swapping cost. In principle, the implementation of wire swaps does not require any logic. In practice, however, it will have some impact at the physical level. It consequently may cause variation of effective coupling capacitance due to input arrival time variation. A detailed discussion on the implementation of the permutation network is described in [9]. However, to swap them may lead to reduce more coupling power over hardware cost of swapping.

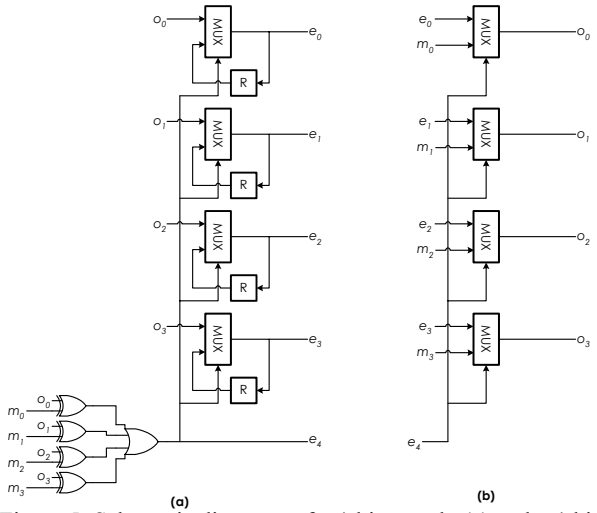

Figure 5: Schematic diagrams of a 4-bit encoder(a) and a 4-bit decoder(b) (bw=4 and $\omega = 1$).

IV. Solution Examples

We evaluate our approach using the MPEG2 and the Mibench embedded benchmark suite [16], which is a set of 35 commercially representative embedded programs divided into six categories including: Automotive and Industrial Control, Consumer Devices, Office Automation, Networking, Security, and Telecommunications. We use some applications from them, which are basicmath, bitcount dijkstra, FFT, GSM, lame, mad, patricia and tiff2rgba. The architectural simulators used in this study are derived from the SimpleScalar/ARM version 2.0 tool set [17], a suite of functional and timing simulation tools for the ARM ISA.

Experimental conditions are same as previous ones. The number of data accesses for evaluation of each benchmark is 100,000.

Table 1 shows power saving results. The simulation results indicate success of our approach. However, in the case of *bw*=2, power consumption increases. The reason for this increase is that index bit lines consume quite power over power saving of original bus parts. We will study power reduction of index bit parts.

Table 2 shows the ratio of each capacitance type. As you can see in Table 2, the effective capacitance in same directions accounts for high percentage. It demonstrates that delay skew variation must be considered. Our approach is available to prevent this effect.

Table 1: Power saving results

	Original power	Saving (%)				
	(mW)	*bw*=2	*bw*=4	*bw*=8	*bw*=16	*bw*=32
MPEG2	5.6	-25.2	4.4	10.8	13.7	9.2
basicmath	10.3	-21.2	2.2	9.9	13.1	4.2
bitcount	12.2	-9.1	19.8	15.0	-2.1	0.0
dijkstra	10.0	-15.3	4.3	4.4	5.8	2.4
FFT	11.4	-25.3	-7.4	2.4	7.2	8.6
GSM	8.4	-23.1	1.4	8.0	13.9	0.0
lame	8.3	-20.9	-2.6	4.5	9.9	6.0
mad	7.8	-17.5	4.3	9.1	11.2	2.2
patricia	9.6	-16.5	4.9	9.8	11.2	3.4
tiff2rgba	6.9	-13.9	4.1	16.5	13.3	5.7
Avg.		-18.8	3.54	9.04	9.72	4.17

Table 2: Distributions of effective capacitance (%)

		Transition type		
	Self cap.	Type B	Type C	Type D
MPEG2	31.6	45.0	11.5	11.9
basicmath	35.3	34.9	17.0	12.8
bitcount	35.9	34.3	17.8	12.0
dijkstra	34.0	38.2	15.0	12.8
FFT	32.5	39.6	13.2	14.7
GSM	36.3	37.2	17.8	8.7
lame	32.9	37.2	14.0	15.9
mad	32.9	35.9	14.2	17.0
patricia	35.1	36.4	16.5	12.0
tiff2rgba	34.1	38.6	15.1	12.2
Avg.	34.1	37.7	15.2	13.0

V. Summary and Conclusions

We have presented a novel coupling-aware coding methodology, which is based on variable length coding. Our proposed method uses probabilistic information of input vector of buses. Experimental results show that to use characteristics of the input vectors is effective for power saving. In addition, the implementation of our approach has low hardware cost.

We also have proposed new variation-aware power estimation towards deep-submicron designs, which consider the effective coupling capacitance variation due to arrival input time variation. With advancing technology process, a large variety of variations has been turning up. The variability must be a new concern about not only delay estimation but also power estimation.

From these results, the proposed methodology will be becoming a key concept for low-power bus design more and more in the future.

Acknowledgements

This work has been supported by the Grant-in-Aid for Creative Scientific Research No. 14GS0218 and the Silicon Sea-Belt Project ``Establishing Project of a Cluster for System-LSI Design and Development". We are grateful for their support.

References

[1] M. R. Stan and W. P. Burleson, ``Bus-Invert Coding for Low-Power I/O,'' *IEEE Trans. VLSI*, vol.3, no.1, pp.49-58, Mar. 1995.

[2] K. W. Kim, K. H. Baek, N. Shanbhag, C. L. Liu and S. M. Kang, ``Coupling-Driven Signal Encoding Scheme for Low-Power Interface Design,'' in Proc. *ICCAD'00*, pp.318-321, Nov. 2000.

[3] L. Benini, G. De Micheli, E. Macii and C. Silvano, ``Asymptonic zero-transition activity encoding for address buses in low-power microprocessor-based system,''In. Proc. 7th *GVLSI*, pp.77-82, Mar. 1997.

[4] E. Musoll, T. Lang and J. Cortadella, ``Working-zone method for reducing the energy in microprocessor address buses,'' *IEEE Trans. VLSI*, vol.6, no.4, pp.568-572, Dec. 1998.

[5] M. Muroyama, A. Hyodo, T. Okuma and H. Yasuura, ``A Power Reduction Scheme for Data Buses by Dynamic Detection of Active Bits,'' *IEICE Trans. Electron*., vol.E87-C, No.4, pp.598-605, Apr. 2004.

[6] R. Arunachalam, E. Acar and S. Nassif, ``Evaluation Method and Metrics of Shielding/Spacing Approaches for Coupling Avoidance,'' research report, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598, Oct. 2002.

[7] E. Macii, M. Poncino and S. Salerno, ``Combining Wire Swapping and Spacing for Low-Power Deep-Submicron Buses,'' in Proc. *GLSVLSI'03*, pp.198-202, Apr. 2003.

[8] Y. Shin and T. Sakurai, ``Coupling-Driven Bus Design for Low-Power Application-Specific Systems,'' in Proc. 38th *DAC*, pp.750-753, Jun. 2001.

[9] L. Macchiarulo, E. Macii and M. Poncino, ``Low-Energy Encoding for Deep-Submicron Address Buses,'' in Proc. *ISLPED'01*, pp. 176-181, Aug. 2001.

[10] K. Agarwal, D. Sylvester and D. Blaauw, ``Variational Delay Metrics for Interconnect Timing Analysis,'' in Proc. 41st *DAC*, pp.381-384, Jun. 2004.

[11] S. H. Choi, B. C. Paul and K. Roy, ``Novel Sizing Algorithm for Yield Improvement under Process Variation in Nanometer Technology,'' in Proc. 41st *DAC*, pp.454-459, Jun. 2004.

[12] A. Agarwal, F. Dartu and D. Blaauw, ``Statistical Gate Delay Model Considering Multiple Input Switching,'' in Proc. 41st *DAC*, pp.658-663, Jun. 2004.

[13] International Technology Roadmap for Semiconductors, http//public.itrs.net/Files/2003ITRS/Interconnect2003.pdf.

[14] H. B. Bakoglu, ``Circuits, Interconnections and Packaging for VLSI,'' Addison-Wesley, 1990.

[15] P. Chen, D. A. Kirkpatrick, K. Keutzer, `` Miller Factor for Gate-Level Coupling Delay Calculation,'' in. Proc. *ICCAD'00*, pp.68-74, Nov. 2000.

[16] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite", in Proc. *IEEE 4th Annual Workshop on Workload Characterization*, Dec. 2001.

[17] D. Burger and T. M. Austin, ``The SimpleScalar tool set, version 2.0,'' Technical Report TR-97-1342, University of Wisconsin Madison, CS Department, Jun. 1997.