Low Power Dynamic Bus Encoding for Deep Sub-micron Design

Kun-Lin Tsai[1]
Department of EE[1]
National Taiwan University
Taipei 106, Taiwan.
kunlin@orchid.ee.ntu.edu.tw

Shanq-Jang Ruan[2]
Department of ET[2]
National Taiwan University
of Science and Technology
Taipei 106, Taiwan.

Li-Wei Chen[3] and Feipei Lai[1,3]
Department of CSIE[3]
National Taiwan University
Taipei 106, Taiwan.
Email: flai@ntu.edu.tw

Edwin Naroska[4]
Department of CEI[4]
University of Dortmund
Dortmund, Germany.

Abstract— **The use of very deep sub-micron (VDSM) technology increases the capacitive and inductive coupling between adjacent wires. It also leads to severe crosstalk noise, which causes power dissipation and may also lead to malfunction of the chip. In this paper, we propose a method to reduce power consumption on buses based on dynamic coding scheme. The proposed method considers capacitive and inductive effects at the same time by using the realistic RLC table and segment invert method. The experimental results show that our approach not only reduces bus power consumption up to 29% but also reduces signal delay up to 34%.**

I. Introduction

Lately the solutions to the low power design and the signal integrity problems have become the center of wide interests. Most previous low power researches focus on how to reduce the circuit power consumption. However, as the technology scales down to the nanometer dimensions and SoC (System-On-a-Chip) designs, addressing low power consumption on on-chip interconnect becomes critical to meet design constraints.

Over the past few years, a large number of studies have been made on low power bus design. However, many of them focused on reducing bus switch activity of each signal line ([1], [2], [3], [4], [5]), and did not take coupling effects into account. Hence, they did not directly related to the power consumption caused by crosstalk. Furthermore, some of these methods are only applicable to data buses [2], [5] or address buses [4].

Basically crosstalk is caused by capacitance and inductance effects. Some approaches address coupling capacitance at physical design level by optimizing routing and gate/wire sizes [7], [8]. In [8], [9], similar approaches are introduced to find the permutation for the bus lines to minimize coupling capacitance effects. These approaches are based on profiling the bus line activity and are suited to special embedded system design; however they can not dynamically adjust to the current signals on the bus.

Another way to reduce crosstalk effect is to minimize the inductance. The inductive effect impacts signal integrity and performance, and is recognized as significant. Several techniques have been proposed to deal with this effect. [10] proposed a shielding technique, in which the signal lines are interdigitated with power or Ground lines. This approach is to isolate signal lines from their neighbors, and can be used to minimize coupling capacitive and inductive noises when combined with ordering nets [11]. Dedicated ground planes is another useful method to reduce inductive effect where the layers above and below the signal lines are dedicated to Vdd or Ground [12], [13].

In this paper, we propose a new dynamic bus encoding scheme for reducing bus power consumption and improving the signal integrity, called RLC-LBE (RLC table based Low-power Bus Encoding). Unlike previous approaches, we consider not only capacitive but also inductive effects by using realistic measure. The proposed approach uses real RLC value of the bus to dynamic optimize signal transition over the bus with segmented inversion scheme. Although bus-inverted scheme is not a new idea [6], to our best knowledge this is the first article to take both capacitive and inductive effects into consideration with bus-invert method. Especially in very deep sub-micron design, the designer should take care not only capacitive effect but inductive effect. In this paper, our approach simultaneously optimize line and coupling effects by quantifying the capacitive and inductive effects. It is advantageous to optimize line RLC and coupling RLC effects at the same time.

II. Coupling Effects

Usually, modern SoCs are equipped with multiple processing units, memories and functional components. These components communicate with each other by various buses: data bus, instruction bus, and address bus. When signals frequently propagate on these narrow-spaced buses, power is dissipated and crosstalk is generated. Our target is to minimize the power consumption caused by crosstalk effects of the on-chip buses.

There are two coupling effects on buses: coupling capacitance effect and coupling inductance effect. Considering a pair of bus wires, during the operation there are three types of possible transitions when we just take dynamic charge distribution over coupling capacitances into account. Figure 1(a) shows single line switching, where the coupling capacitance is charged (or discharged) with C_xV. In Figure 1(b), the coupling capacitance will be charged with $2C_xV$ if the signals on both wires transit into opposite directions. If both wires concurrently switch to the same voltage level, as shown in Figure 1(c), then the charge on the capacitance is not affected. The power is dissipated during the capacitance charge and discharge.

With the technology trend, on-chip inductance effects, such as delay increase, overshoot, and inductive crosstalk, have become increasingly significant. One situation is as the clock

0-7803-8935-2/05/$20.00 ©2005 IEEE.

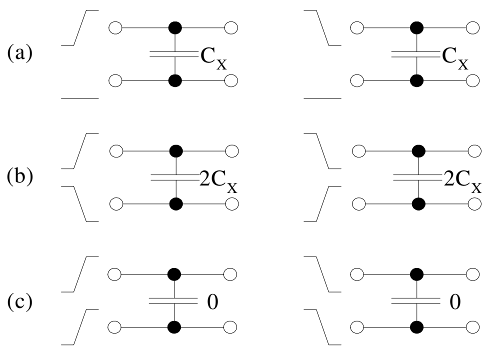


Fig. 1. Line switching for two buses interconnects.

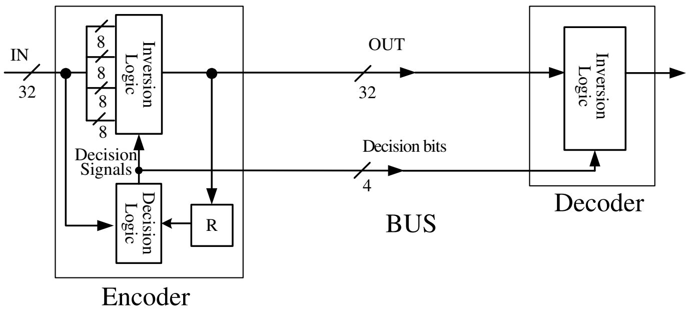


Fig. 2. The RLC-LBE encoder/decoder architecture.

frequency increases and the rise time decreases, electrical signals comprise more and more high-frequency components which makes the inductance effects more significant. Besides, in order to increase the performance, more and more wires are designed in parallel, and the inductive crosstalk is also increased. To avoid the malfunction caused by inductive effect,some error correction or data retransmission techniques are put together into the bus design. Thus, the power and delay of buses are both increased.

III. Dynamic Bus Encoding Scheme

In this section, we first introduce our bus encoder/decoder architecture. Secondly, we show how to reduce the crosstalk effect by using RLC table, and discuss the implementation of encoder/decoder. Finally, an example is given to illustrate our method.

A. RLC-LBE Architecture

The encoder of RLC-LBE (RLC table based Low-power Bus Encoding) method consists of two primary components, *inversion-logic*, and *decision-logic* as shown in Figure 2. When data transmit on the bus, we first partition the data into m groups ($m = 4$ in Figure 2). The *decision-logic* compares the current input data with the previous output data of each group according to the realistic RLC table. And then the *decision-logic* generates m decision bits and sends them to the *inversion-logic*. The *inversion-logic* decides which group of the input data needs to be inverted according to the decision signals so that it can obtain a low crosstalk effect on the bus. Finally, the encoded data and the decision bits are put on the bus for recovering data at the receiving end.

The complexity of the decoder is much simpler than that of the encoder. Only *inversion-logic* is required to convert the encoded data into the original data. The *inversion-logic* of the decoder inverts the data by the decision bits, which are generated by the RLC-LBE encoder. It can be done in a very short time.

Note that we partition the original input data into four groups in the RLC-LBE architecture, and feed them into *inversion-logic* and the *decision-logic* as shown in Figure 2. However, the number of partition groups is not fixed, it needs to be decided by the designer so that it can satisfy the design constraints. Fewer partitions result in more prominent crosstalk effect reduction due to better locality; however, it increases the size of RLC table. That means it has larger area overhead.

B. Reducing Crosstalk Effect by RLC Table

To solve the signal integrity problem caused by crosstalk, we apply the methods of *table look-up* and *segment data inversion*. First, when creating the RLC table, we build the bus model by SPICE, and give some related parameters to generate the power value. Detailed experimentation discussion will be presented in section 4. The size of RLC table would be increased with the number of bits of each group. Each record in the RLC table keeps track of the power value on the change of two consecutive transmissions. A bigger value indicates that there exists more serious crosstalk effect between bus wires.

In the following, let us discuss segment data inversion using the RLC table. In order to describe the RLC-LBE method clearly, three definitions are made:

Definition 1: Assume the bus width is n, and partition them into m groups $G_1 \sim G_m$. Let G_k denote as group k, and there are N_k bits in G_k. That is $N_1 + N_2 + ... + N_m = n$. N_i of G_i need not be equal to N_j of G_j.

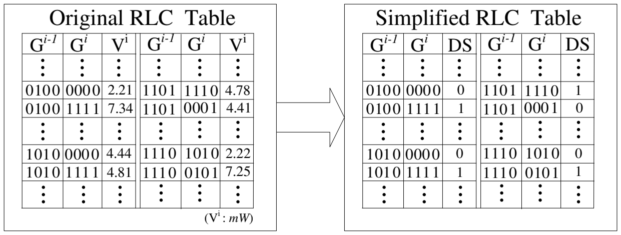
Definition 2: Let G_k denote as group k, $G^i_{k\_ori}$ as the original data of G_k at i^{th} transmission, $\overline{G^i_{k\_ori}}$ as the inverse signal of $G^i_{k\_ori}$, and $G^{i\;\;1}_{k\_enc}$ as the $(i-1)^{th}$ encoded data of G_k. Define $V^i_k = G^{i\;\;1}_{k\_enc} \Phi G^i_{k\_ori}$, where Φ is the operation of calculating the power value when data switch from $G^{i\;\;1}_{k\_enc}$ to $G^i_{k\_ori}$, and V^i_k is the power consumption between two times of transmission. Likewise, $\overline{V^i_k} = G^{i\;\;1}_{k\_enc} \Phi \overline{G^i_{k\_ori}}$ is the power value when the data switch from $G^{i\;\;1}_{k\_enc}$ to $\overline{G^i_{k\_ori}}$.

By definitions 1 and 2, we can create an original RLC table for each group as shown in the left-hand side of Figure 3 ($N_i = 4$).

Definition 3: By definition 2, define

$$DS^i_k = \begin{cases} 0 & ,\text{if } V^i_k \leq \overline{V^i_k} \\ 1 & ,\text{if } V^i_k > \overline{V^i_k} \end{cases} \quad \text{, and}$$

$$TP_i = \sum_{k=1}^{m} \{V^i_k + DS^i_k(\overline{V^i_k} - V^i_k)\}$$



Original RLC Table

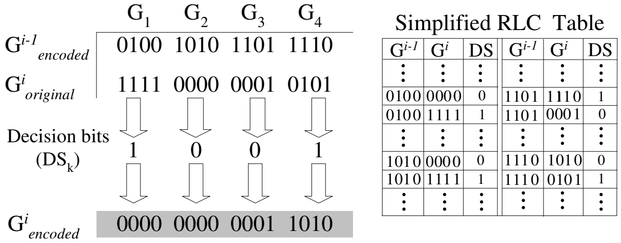
G^{i-1}	G^i	V^i	G^{i-1}	G^i	V^i
⋮	⋮	⋮	⋮	⋮	⋮
0100	0000	2.21	1101	1110	4.78
0100	1111	7.34	1101	0001	4.41
⋮	⋮	⋮	⋮	⋮	⋮
1010	0000	4.44	1110	1010	2.22
1010	1111	4.81	1110	0101	7.25
⋮	⋮	⋮	⋮	⋮	⋮

(V^i : mW)

Simplified RLC Table

G^{i-1}	G^i	DS	G^{i-1}	G^i	DS
⋮	⋮	⋮	⋮	⋮	⋮
0100	0000	0	1101	1110	1
0100	1111	1	1101	0001	0
⋮	⋮	⋮	⋮	⋮	⋮
1010	0000	0	1110	1010	0
1010	1111	1	1110	0101	1
⋮	⋮	⋮	⋮	⋮	⋮

Fig. 3. Simplification of the RLC table.


G_1 G_2 G_3 G_4
$G^{i-1}_{encoded}$ 0100 1010 1101 1110
$G^{i}_{original}$ 1111 0000 0001 0101
Decision bits (DS_k) 1 0 0 1
$G^{i}_{encoded}$ 0000 0000 0001 1010

Simplified RLC Table

G^{i-1}	G^i	DS	G^{i-1}	G^i	DS
⋮	⋮	⋮	⋮	⋮	⋮
0100	0000	0	1101	1110	1
0100	1111	1	1101	0001	0
⋮	⋮	⋮	⋮	⋮	⋮
1010	0000	0	1110	1010	0
1010	1111	1	1110	0101	1
⋮	⋮	⋮	⋮	⋮	⋮

Fig. 4. An example of RLC-LBE operation.

where DS_k^i is the group k's decision bit at i^{th} transmission, and TP_i is the total power value at the i^{th} transmission.

By definition 3, we can further simplify the original RLC table into a simple form as shown in the right-hand side of Figure 3. Thus, the process of our dynamic encoding method could be regarded as finding DS_k^i, $k = \{1, 2, \ldots, m\}$, such that TP_i is the minimum.

From the simplified RLC table, the DS bits can be generated for each group each time. The *inversion-logic* decides which group's data need to be inverted according to the DS bits. Only *XOR* gates are needed in the *inversion-logic*. Consequently, the RLC-LBE encoder is much simpler than the other complex methods.

It must be noted that the inversion of partial input data is not for minimizing the Hamming distance between two consecutive transmissions, which is proposed in [17]. It is used to select a better solution for crosstalk problem.

C. An Example

We illustrate the dynamic coding method with an example, as shown in the left side of Figure 4, the bus width is 16 bits, and we partition them into four groups, G_1 to G_4, and 4 bits in each group. The $(i-1)^{th}$ encoded output signal is $\{0100\ 1010\ 1101\ 1110\}$, and the i^{th} original input signal is $\{1111\ 0000\ 0001\ 0101\}$. Now, considering the data switch in G_1, the previous output is $\{0100\}$ and the current input is $\{1111\}$. Referring to the RLC table, shown in the right side of Figure 4, when the data switch from 0100 to 1111, the DS value is 1. It means if the data switch from 0100 to 1111, the power consumption is bigger than that of switching from 0100 to 0000 (as shown in Figure 3), and we need to invert the original data from 1111 to 0000 so that we can obtain a smaller power result. The same method can also be applied to G_2, G_3 and G_4, thus the decision bits from DS_1^i to DS_4^i are $\{1, 0, 0, 1\}$. According to the decision bits, the inversion logic inverts the input signals in G_1 and G_4. Finally, the encoded output signal with low crosstalk effect is $\{0000\ 0000\ 0001\ 1010\}$.

IV. Experimental Result

To show the effectiveness of our method, we estimated the power consumption, and signal latency during bus operation. We also implemented the encoder and decoder architecture to evaluate the overhead. An essential step of design encoder was to construct the RLC table circuit (decision logic). The inductance and capacitance model we used was generated by FastHenry [15] and Fastcap [16] with TSMC 0.18 μm process. The table was simplified and implemented as logic circuitry instead of ROM-based design. We used SPICE to simulate a 1000μm-long bus, with 0.24μm wide and 0.6 μm thick. Besides, we partition each bus wire into several segments, and each segment is 200μm long. Furthermore, the coupling effect between each two wires were calculated. The whole bus width was partition into several groups, and each group was set to 8 bits. The important point to note is the bit number of each group can be modified to meet the design constraint. The more wires in each group, the more power saving it can have, however, it results in more area overhead in the encoder. The area, delay, and power of the 32 bits encoder and decoder for *Bus-Invert* Encoding [17] and our RLC-LBE method are summarized in Table I.

TABLE I

The area, delay, and power of a 32 bits encoder and decoder for different methods.

	Encoder		Decoder	
	Bus-Invert	RLC-LBE	Bus-Invert	RLC-LBE
Area (μm^2)	5921	9951	2950	2950
Delay (ns)	4.804	6.403	0.286	0.286
Power (μW)	21.35	23.20	6.46	6.46

Table II presents the signal delay and bus power consumption of non-encoded, bus-invert method [17], and our RLC-LBE method. The 100,000 random sequences were used as the input signals for bus. As can be seen in Table II, our RLC-LBE method shows the best power and delay results in among three methods. This also implies our proposed method can save more energy than the others. The average delay reduction is 32.68% compared with the original implementation (non-encoded method), and average power saving is 27.94%. The result also shows that our method is better than the non-encoded and bus-invert method in both aspects of delay and power. Figure 5 and Figure 6 show the delay and power reduction versus the non-encoded bus. These two figures indicate that the proposed method can achieve almost the same crosstalk reduction rate for different sizes of bus width.

From the results, we conclude that the RLC-LBE method is very effective in reducing crosstalk effect up to 34% of delay (32 bits) and 29% of power (64 bits). This method can be applied to various bus widths and bus lengths, and is suitable

TABLE II

Delay and power of Non-encoded, bus-invert, and RLC-LBE method in different bus widths.

Bus Width	Delay (ns)(transmission time + circuit delay)			Power (μW)		
	Non-Encoded	Bus-Invert[17]	Proposed method	Non-Encoded	Bus-Invert[17]	Proposed method
8	291.3	261.8	206.3	52	50	40
16	705.4	575.3	484.9	102	100	74
24	1104.0	882.1	743.3	149	147	108
32	1464.1	1182.2	966.3	196	196	142
64	2862.3	2589.0	1926.0	385	383	274
Reduction		14.58%	32.68%		0.88%	27.94%

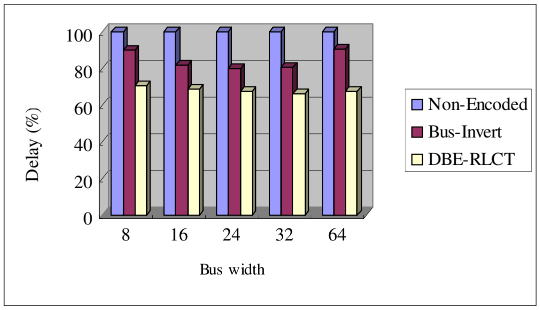


Fig. 5. Delay reduction percentage.

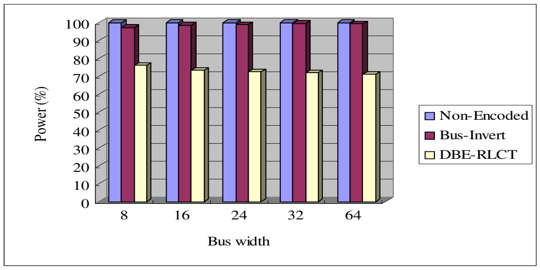


Fig. 6. Power reduction percentage.

for current SoC bus design. The encoder and decoder circuits can be easily designed by using basic NAND, NOR, and XOR gates. The RLC-LBE method is useful in the real-world design since it is based on the realistic RLC table.

V. Conclusion and Future Work

In this paper, the RLC-LBE dynamic bus encoding architecture based on the realistic RLC table is proposed. When data is prepared to transmit on the bus, we compare it with the previous output data and invert partial input data bits according to the value of RLC table so that we can obtain low crosstalk effect and low power consumption on bus. Note that both capacitive effect and inductive effect are considered in our architecture. The experimental results verified our method is feasible for very deep sub-micron design. What should be noticed is that the RLC-LBE architecture is very flexible since it can be used on arbitrary bus width and the value of the original RLC table can be adjusted for different situations. It indicates that RLC-LBE method is effective for high performance and low power transmission especially for the very deep sub-micron bus design. Though the proposed method gained a good result from the experiment, there still have some portions need to be improved. For example, the number of bits in each group needs to be optimized and generated automatically. Otherwise, the RLC table should be combined with other techniques to provide better bus design.

References

[1] U. Narayanan, K. S. Chung, and T. Kim, “Enhanced Bus Invert Encodings for Low-Power,” *In IEEE ISCAS’2002*, vol. 5, pages 25–28, May 2002.

[2] M. Madhu, V. S. Murty, and V. Kamakoti, “Dynamic coding technique for low-power data bus,” *In IEEE ISVLSI’03*, pages 252–253, Feb 2003.

[3] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, “A coding framework for low-power address and data busses,” *IEEE Trans. on VLSI Systems*, vol. 7, issue 2, pages 212–221, June 1999.

[4] L. Benini, G. D. Micheli, E. Macii, M. Poncino, and S. Quer, “Power optimization of core-based systems by address bus encoding,” *IEEE Trans. on VLSI System*, vol. 6, issue 4, pages 554–562, Dec 1998.

[5] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, “A Dictionary-Based En/Decoding Scheme for Low-Power Data Buses,” *IEEE Trans. on VLSI System*, vol. 11, no. 5, pages 943–951, Oct 2003.

[6] Y. Zhang, J. Lach, K. Skadron, and M.R. Stan “OddEven Bus Invert with Two-Phase Transfer for Buses with Coupling,” *In Proc. of ISLPED’02*, pages 80–83, Aug. 2002.

[7] H. Zhou and D. Wong, “Global routing with crosstalk constraints,” *In Proc. of DAC-1998*, June 1998.

[8] I. H.-R. Jiang, Y.-W. Chang, and J. Y. Jou, “Crosstalk-driven interconnect optimization by simulataneous gate and wire sizing,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, 19:999–1010, Sep 2000.

[9] E. Naroska, S. J. Ruan, F. Lai, U. Schwiegelshohn, and L. C. Liu, “On optimizing power and crosstalk for bus coupling capacitance using genetic algorithm” *In Proc. of ISCAS’03*, pages 277–280, May 2003.

[10] A. Deutsch, G. Kopcsay, P. Restle, H. Smith, G. Katopis, W. Becker, P. Coteus, C. Surovic, B. Rubin, R. Dunne, T. Gallo, K. Jenkins, L. Terman, R. Dennard, G. Sai-Halasz, B. Krauter, , and D. Knebel, “When are transmission-line effects important for on-chip interconnections?” *IEEE Trans. on Microwave Theory Tech*, 45:1836–1846, Oct 1997.

[11] K. Lepak, I. Luwandi, and L. He, “Simultaneous shield insertion and net ordering under explicit rlc noise constraint,” *in Proc. of DAC, 2001*, pages 199–202, 2001.

[12] D. Bailey and B. Benschneider, “Clocking design and analysis for 600-mhz alpha microprocessor,” *IEEE J. Solid-State Circuit*, 33:1627–1633, Nov 1998.

[13] L. Vakanas, S. Hasan, A. Cangellaris, and J. L. Prince, “Effects of floating planes in three-dimensional packaging structures on simultaneous switching noise,” *IEEE Trans. on Components, Packaging, and Manufacturing Technology*, 21:434–440, Nov 1998.

[14] Magma Design Automation Inc., Magma design automation inc, deep-submicron signal integrity white paper, 2002.

[15] M. Kamon, M. J. Ttsuk, and J. K. White, “FASTHENRY: A multiple-accelerated 3-D inductance extraction program,” *IEEE Trans. on Microwave Theory and Techniques*, vol. 42, issue 9, pages 1750–1758, Sep 1994.

[16] K. Nabors and J. White, “FASTCAP: A multiple accelerated 3-d capacitance extraction program,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pages 1447–1459, Nov 1991.

[17] M. R. Stan and W. P. Burleson, “Bus-invert coding for low-power i/o,” *IEEE Trans. on VLSI System*, 3:49–58, Mar 1995.