

## RLC Coupling-Aware Simulation and On-Chip Bus Encoding for Delay Reduction

Shang-Wei Tu, Yao-Wen Chang, and Jing-Yang Jou

**Abstract**—This paper shows that the worst case switching pattern that incurs the longest bus delay while considering the RLC effect is quite different from that while considering the RC effect alone. It implies that the existing encoding schemes based on the RC model may not improve or possibly worsen the delay when the inductance effects become dominant. A bus-invert method is also proposed to reduce the on-chip bus delay based on the RLC model. Simulation results show that the proposed encoding scheme significantly reduces the worst case coupling delay of the inductance-dominated buses.

**Index Terms**—Bus-invert method, coupling, inductance, interconnect delay, worst case switching pattern.

### I. INTRODUCTION

With aggressive scaling of transistor size, interconnect delay increasingly dominates chip performance in deep-submicrometer designs [17], [18], [20]. Besides, as the process technology advances and the clock frequency increases over gigahertz, the inductance effects of on-chip interconnect structures have become increasingly significant [7], [18]. On-chip inductance effects in high-performance circuit designs might affect interconnect in many ways. The performance of a circuit will be reduced due to the increase of wire delay [5], [13]. The long-range inductive crosstalk can cause serious signal integrity related problems [9], [13]. Signal overshoots and undershoots due to wire inductance may damage devices. Finally, inductance in power and ground grids can increase the noise in the supply and ground voltages when large currents flow. This is also known as the ground-bounce problem. Therefore, inductance effects cannot be neglected in today's high-performance circuit designs, especially for global interconnects such as clock wires and signal buses.

Most existing works focus on reducing the effects resulting from coupling capacitance on the bus structure. There is not much work in the literature considering inductance effects on the bus structure to develop encoding schemes to reduce bus delay. Considering only the capacitive coupling effect, Victor and Keutzer [21], Baek *et al.* [1], Hirose and Yasuura [10], and Sotiriadis and Chandrakasan [19] proposed their bus encoding techniques to eliminate crosstalk delay.

Since most previous works only consider capacitance effects on the bus to reduce delay, the worst case switching pattern that incurs the largest delay is when adjacent wires simultaneously switch in opposite transition directions. However, considering the RLC circuit model for the bus structure, we find that the worst case switching pattern with the largest on-chip bus delay is when all wires simultaneously switch in the same direction. On the contrary, this worst case pattern is the best case pattern of a coupling RC model. Further, the best case pattern with the RLC model is that the central wire of the bus switches in a different direction from all other wires that all switch in the same direction.

Manuscript received March 29, 2005; revised July 19, 2005 and September 23, 2005. This work was supported by the MediaTek Research Center at NCTU under Grant Q583. This paper was recommended by Associate Editor R. Suaya.

S.-W. Tu and J.-Y. Jou are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: kuma@athena.ee.nctu.edu.tw; jyjou@faculty.nctu.edu.tw).

Y.-W. Chang is with the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: ywchang@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCAD.2005.860956

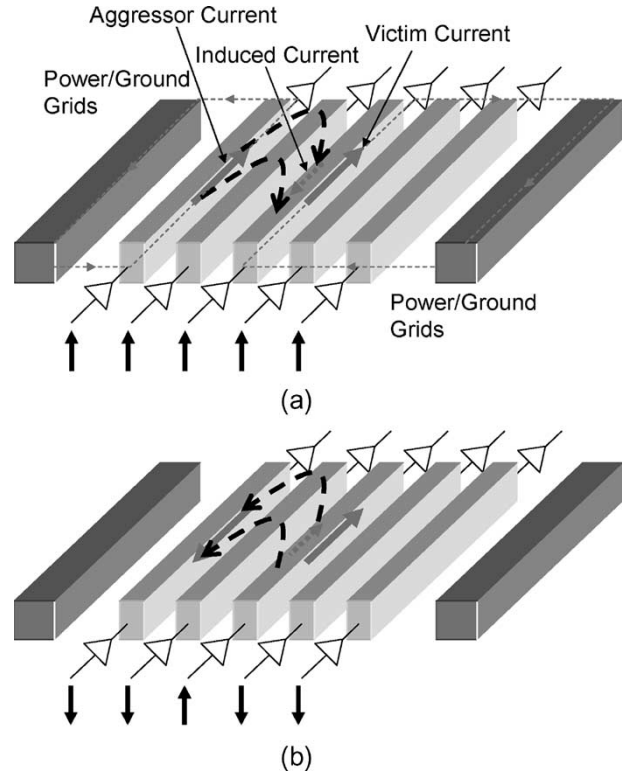


Fig. 1. LC cross-coupled 5-bit bus structure. (a) Switching pattern of the worst case delay in the RLC model. (b) Switching pattern of the best case delay in the RLC model. ( $\uparrow$ : switch from “0” to “1.”  $\downarrow$ : switch from “1” to “0.”)

However, this best case pattern is just the worst case pattern with the RC model. See Fig. 1 for examples of the worst case and best case switching patterns on a 5-bit bus. Therefore, the worst case patterns with the maximum on-chip bus delay are completely different for the RC and RLC models. Hence, as inductance cannot be neglected in today's high-performance circuit design, it is very important to consider RLC effects to develop encoding schemes to reduce the bus delay.

With the findings of the best case and worst case patterns, we propose a new encoding scheme for on-chip buses to minimize coupling delay with the dominance of inductance effects. The key idea is that inductance coupling effects should be alleviated by transforming the data sequences transmitting through on-chip buses. However, the architectures of the encoder and decoder should be of low complexity so that the power and delay overheads due to the codec circuitry can be compensated by the significant reduction of bus delay.

The rest of this paper is organized as follows. Section II describes the parameters and basic assumptions used in our study for the bus structure and then gives the working flow. Section III performs some simulations by using the RC model. Section IV gives simulations by using the RLC model. The method and circuitry of our encoding (decoding) scheme are described in Section V, and simulation results are shown in Section VI. Finally, Section VII concludes the paper and discusses our future work.

### II. PRELIMINARY

In this work, we used the bus structure shown in Fig. 1 to conduct our simulations. We assume that all drivers (receivers) have uniform size and all signal wires have uniform width, spacing, and length. The length, width, and pitch of the signal wire were 2000, 0.8, and

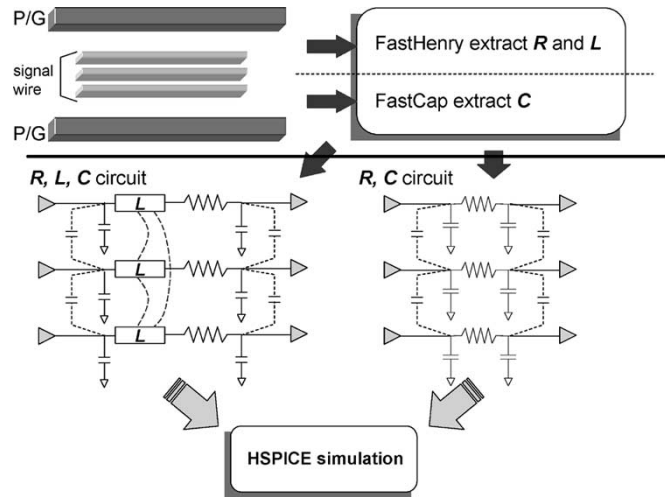


Fig. 2. Working flow.

TABLE I  
SIMULATION RESULTS OF A 5-bit BUS CONSIDERING ONLY  
RC EFFECTS (0: NO TRANSITION)

Switching pattern	50% delay of the central wire (ps)	Delay comparison with that of 00↑00
↑↑↑↑↑	33	-35.29%
↑↓↑↑↑	49	-3.92%
↑↓↑↑↓	51	0.00%
↑↓↑↓↑	73	43.14%
↑↓↑↓↓	75	47.06%
↓↑↑↑↑	35	-31.37%
↓↑↑↑↓	37	-27.45%
↓↑↑↓↓	54	5.88%
↓↓↑↑↑	52	1.96%
↓↓↑↓↓	76	49.02%
00↑00	51	0.00%

2  $\mu\text{m}$ , respectively. The respective width and pitch of the power/ground were 2 and 13  $\mu\text{m}$ . The heights of all wires are set to 2  $\mu\text{m}$ . The signal rise/fall time was set to 100 ps. With these feasible parameters [7], [8], [18], we used the famous 3-D field-solver FastCap [15] to extract the self and coupling capacitance and FastHenry [14] to extract the resistance, self inductance, and coupling inductance. Then, with these extracted *RLC* parameters, we constructed the coupling *RLC* and *RC* circuit models. Both circuit models were constructed as  $\pi$  segments using series resistance (or series resistance and inductance for *RL*) and shunt capacitance. Finally, the circuits were simulated by using HSPICE. The overall flowchart is illustrated in Fig. 2. In our simulations, we assumed that synchronous latches are located at the transmitter side. Thus, all the signals switch at the same time on the buses.

### III. SIMULATIONS WITH THE RC CIRCUIT MODEL

In this section, we simulate all switching patterns on the 5-bit bus structure considering only *RC* effects. The simulation results are listed in Table I. We should note that the number of total switching patterns is  $2^5 = 32$  (without considering nontransition cases). However, switching from “0” to “1” is symmetric to switching from “1” to “0” for bus delay computation. Therefore, the complete switching patterns can be reduced to  $2^5/2 = 16$ . Besides, the 5-bit bus structure is also a symmetric structure with respect to the central signal wire. For example, the switching patterns ↓↓↑↑↑ and ↑↑↑↓↓ have the same delay effect on the central signal wire. Hence, the complete switching

TABLE II  
SIMULATION RESULTS OF A 5-bit BUS CONSIDERING  
*RLC* EFFECTS (V<sub>dd</sub> = 1.2 V)

Switching pattern	50% delay of the central wire (ps)	Delay comparison with that of 00↑00	Max swing (V)	Noise (% of V <sub>dd</sub> )
↑↑↑↑↑	97	51.56%	1.71	42.50%
↑↓↑↑↑	88	37.50%	1.47	22.50%
↑↓↑↑↓	63	-1.56%	1.31	9.20%
↑↓↑↓↑	75	17.19%	1.17	-2.50%
↑↓↑↓↓	51	-20.31%	1.04	-13.33%
↓↑↑↑↑	78	21.88%	1.56	30.00%
↓↑↑↑↓	55	-14.06%	1.4	16.67%
↓↑↑↓↓	45	-29.69%	1.13	-5.83%
↓↓↑↑↑	66	3.13%	1.33	10.83%
↓↓↑↓↓	37	-42.19%	0.903	-24.75%
00↑00	64	0.00%	1.31	9.17%

patterns can further be reduced to ten patterns as listed in Table I (the first ten patterns).

From Table I, the three patterns ↓↓↑↑↓, ↑↓↑↓↓, and ↑↓↑↑↑ result in significantly larger delays on the central signal wire. Obviously, when we consider the resistance, self capacitance, and coupling capacitance of interconnects, the worst case switching pattern that incurs the largest delay is when adjacent wires simultaneously switch in opposite transition directions. Therefore, all the previously mentioned encoding schemes [1], [10], [19], [21] can improve the worst case bus delay.

### IV. SIMULATIONS WITH THE RLC CIRCUIT MODEL

In this section, we first simulate all switching patterns on the 5-bit bus structure considering the *RLC* effects of bus interconnects, and then increase wire capacitance to see whether the worst case switching pattern will change or not as the wire capacitance becomes dominant. The simulation results for the 5-bit bus are shown in Table II. From Table II, we observe that the worst case pattern changes from ↓↓↑↓↓ (in Table I) to ↑↑↑↑↑ and the best case pattern changes from ↑↑↑↑↑ (in Table I) to ↓↓↑↓↓. Therefore, the worst case and best case switching patterns are completely different considering *RC* and *RLC* effects. Therefore, as the technology advances and the clock frequency continues to increase, it is very important to consider *RLC* effects on the bus structure to derive encoding schemes to reduce bus delay. Otherwise, the encoding schemes might not improve or even worsen the on-chip bus delay because of the redundant logics and wires. Further, we also observe that the largest overshoot noise occurs for the pattern ↑↑↑↑↑, as shown in Table II.

Why does the worst case switching pattern ↑↑↑↑↑ result in the largest bus delay when considering *RLC* effects on the 5-bit bus? Theoretically speaking, this is mainly due to two factors. 1) Inductance becomes dominant due to higher frequency (increasing the impedance of wire inductance that is  $j\omega L$ ) and longer interconnects (longer return path). 2) It is also due to the long-range effect of inductance. From Faraday’s law [2], as shown in (1), the electromotive force induced in a closed circuit is equal to the negative rate of increase of the magnetic flux linking the circuit. We have

$$V_j = -\frac{d\Phi_{ij}}{dt} \text{ with } \Phi_{ij} = \int_{s_j} \vec{B}_i \cdot d\vec{s}_j \quad (1)$$

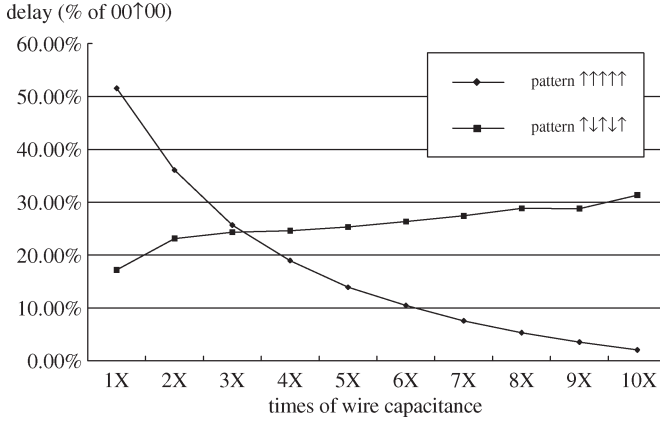


Fig. 3. Delays (percentage of that of pattern  $00\uparrow 00$ ) of the worst case switching pattern with various wire capacitances.

where  $V_j$  is the electromotive force induced in loop  $j$  due to the time-varying current  $I_i$  in loop  $i$ . Here,  $\Phi_{ij}$  is the magnetic flux in loop  $j$  due to the current  $I_i$ ,  $B_i$  is the magnetic flux density arising from current  $I_i$ , and  $S_j$  represents the surface bounded by the loop  $j$ . The orientation of  $B_i$  can be determined from the right-hand rule. Therefore, as shown in Fig. 1(a), the time-varying (increasing) current of the leftmost aggressor wire will induce a downward time-varying (increasing) magnetic field on the victim wire. Therefore, the current results in a positive mutual flux  $\Phi$ , which also increases with time. Finally, from (1), the induced voltage on the victim loop is negative; that is, the induced current on the victim wire flows in the reverse direction of the victim current. Hence, while all neighboring wires simultaneously switch in the same direction as the victim wire does, they will all induce a current of different direction on the victim wire as shown in Fig. 1(a). This implies that the charging time (delay) will increase due to the long-range coupling. We can conclude that as inductance effects dominate, the worst case switching pattern with maximum delay is when all wires simultaneously switch in the same direction. Meanwhile, these patterns will also result in the largest noise between each other.

Since Cao *et al.* [3] claimed that the worst case switching pattern for a 5-bit bus should be  $\uparrow\downarrow\uparrow\downarrow\uparrow$  considering capacitive and inductive coupling, we also conducted simulations to see whether the worst case switching pattern will change or not when capacitance effects become dominant. We simulated with the extracted *RCL* circuit model of the 5-bit bus by increasing the wire capacitance step by step. The simulation results are shown in Fig. 3, and the complete switching patterns when capacitance effects dominate (ten times of wire capacitance) are listed in Table III.

From Fig. 3 and Table III, we observe that the worst case switching pattern for the 5-bit bus changes from  $\uparrow\uparrow\uparrow\uparrow\uparrow$  to  $\uparrow\downarrow\uparrow\downarrow\uparrow$ . From Table III, while considering the worst (best) case switching pattern as wire capacitance dominates, we should first consider the immediate neighbors for the worst (best) case capacitive coupling and then consider the farther neighbors for the worst (best) inductive coupling.

To further investigate the change of the worst case switching pattern when capacitance effects dominate, we also conducted simulations with varying signal rise times. As shown in Fig. 4, the worst case switching pattern for the 5-bit bus also changes from  $\uparrow\uparrow\uparrow\uparrow\uparrow$  to  $\uparrow\downarrow\uparrow\downarrow\uparrow$  when we increase the signal rise time (i.e., decrease the working frequency). This phenomenon also conforms to the trend when capacitance effects dominate since the impedance of wire capacitance will increase as the working frequency decreases. We should note that the frequency of interest here is 583.3 MHz as the rise time is set to 600 ps,

TABLE III  
SIMULATION RESULTS OF THE 5-bit BUS WHEN WIRE CAPACITANCE BECOMES DOMINANT (TEN TIMES WIRE CAPACITANCE)

Switching pattern	50% delay of the central wire (ps)	Delay comparison with that of $00\uparrow 00$
$\uparrow\uparrow\uparrow\uparrow\uparrow$	443	2.07%
$\uparrow\downarrow\uparrow\downarrow\uparrow$	480	10.60%
$\uparrow\downarrow\uparrow\downarrow$	436	0.46%
$\uparrow\downarrow\uparrow\downarrow\uparrow$	570	31.34%
$\uparrow\downarrow\uparrow\downarrow$	530	22.12%
$\downarrow\uparrow\uparrow\uparrow$	405	-6.68%
$\downarrow\uparrow\uparrow\downarrow$	347	-20.05%
$\downarrow\uparrow\downarrow\downarrow$	348	-19.82%
$\downarrow\downarrow\uparrow\uparrow$	433	-0.23%
$\downarrow\downarrow\uparrow\downarrow$	358	-17.51%
$00\uparrow 00$	434	0.00%

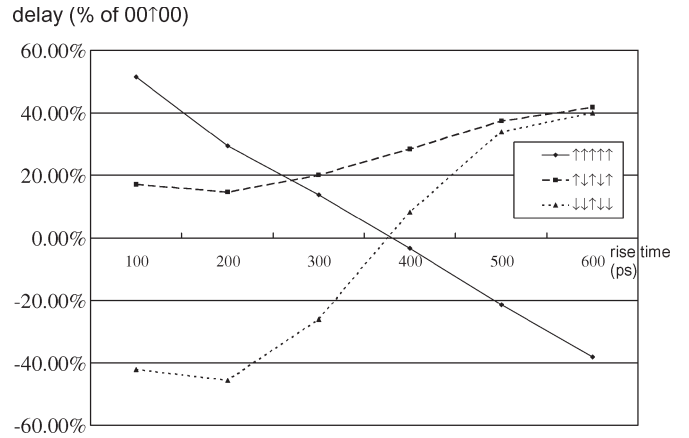


Fig. 4. Delays (percentage of that of the pattern  $00\uparrow 00$ ) of the worst case switching pattern with various signal rise times.

for which the capacitance effects dominate. (See [12] for the formula to determine whether the inductance effects are significant.)

## V. BUS-INVERT SCHEME

Inspired by Stan's low-power bus-invert method [16] for reducing the transition activities to reduce the bus transition power, we propose a bus-invert method to reduce the on-chip bus delay due to coupling effects while inductance effects dominate. Our bus-invert method inverts the input data when the number of bits switching in the same direction is more than half of the number of signal bits. The remaining problem is how to implement the coding architecture with low complexity. For the implementation, we propose an encoder architecture shown in Fig. 5.

There are three types of possible signal transitions: type I:  $\uparrow$  (switching from "0" to "1"), type II:  $\downarrow$  (switching from "1" to "0"), and type III: 0 (no switching). If we refer to  $x_i(n)$  as an input signal and to  $x_i(n-1)$  as its previous input signal, then type I is  $(x_i(n), x_i(n-1)) = (1, 0)$ , type II is  $(x_i(n), x_i(n-1)) = (0, 1)$ , and type III is  $(x_i(n), x_i(n-1)) = (0, 0)$  or  $(1, 1)$ . With the input  $x_i(n)$  and  $x_i(n-1)$ , the codeword generator generates  $(q_L, q_H) = (0, 1)$  for type I,  $(1, 0)$  for type II, and  $(0, 0)$  for type III. Then all  $q_L$ 's are inputs to the majority voter ( $L$ ) and all  $q_H$ 's to the majority voter ( $H$ ). Finally, from the output of the majority voter  $L$  or  $H$ , we can detect if the number of type I or II transitions is more than half

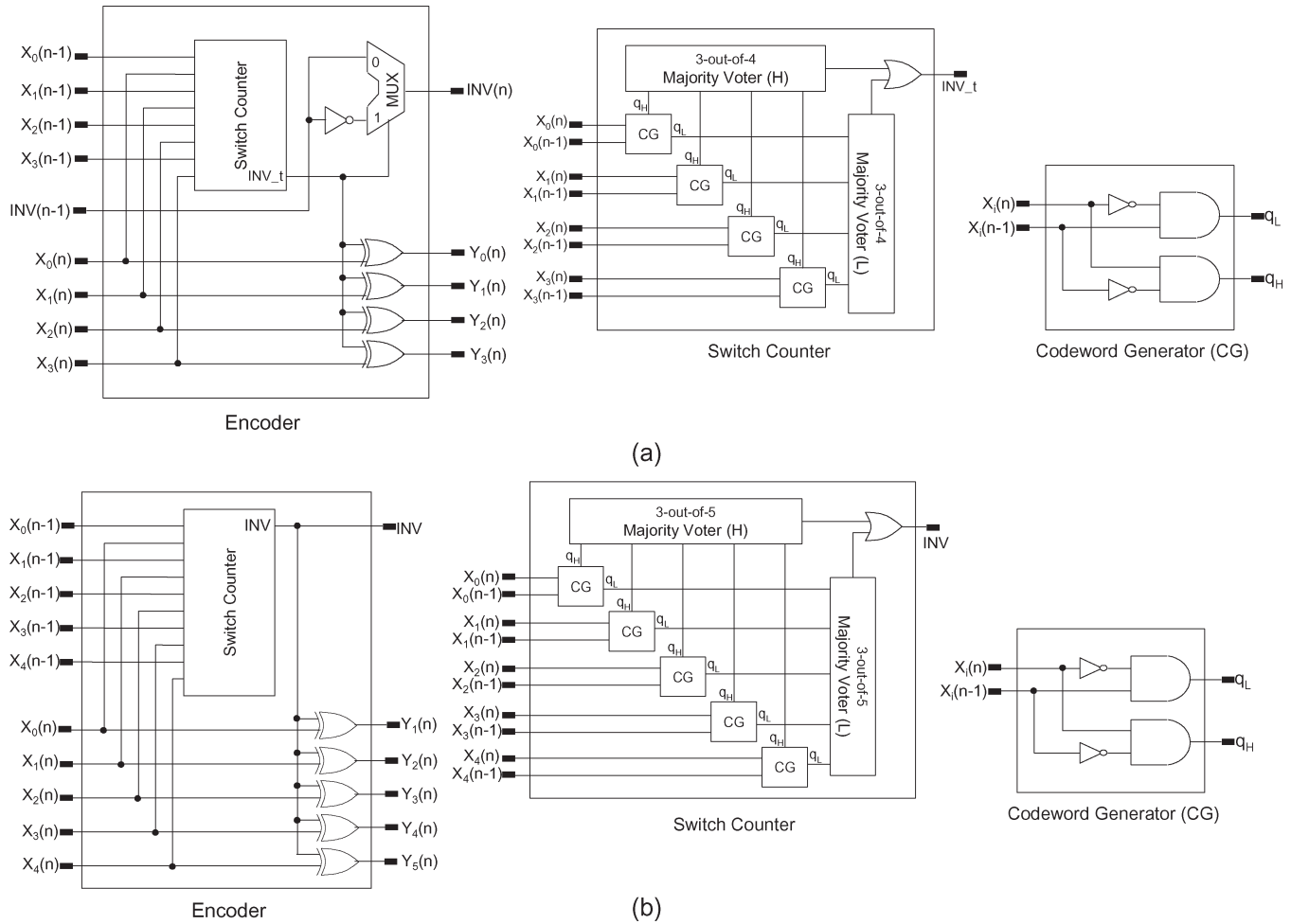


Fig. 5. (a) 4-bit bus encoder for the bus-invert scheme. (b) 5-bit bus encoder for the bus-invert scheme.

of the number of signal bits. If one of the majority voters' outputs is high, the input signal should be inverted. The majority voters can be implemented by using either a tree of full adders or resistors combined with a voltage comparator [16].

Since the additional invert line will contribute to transitions, it should also be considered. Let  $N$  be the total number of signal bits of a bus excluding the invert line. The output of the majority voter is asserted when  $\lceil (N + 1)/2 \rceil$  inputs are high. If  $N$  is odd, the example encoder architecture is just as that shown in Fig. 5(b). Hence, after encoding, the worst case switching pattern occurs when  $(N + 1)/2$  signal bits switch in the same direction, where  $N$  is odd. If  $N$  is even, the encoder architecture is somewhat different as that shown in Fig. 5(a). The major differences are that we need an extra input  $INV(n - 1)$  for our encoder and  $INV(n) = INV(n - 1)'$  or  $INV(n - 1)$ , depending if  $INV_t$  is high or low. Hence, after encoding, the worst case switching pattern is that  $N/2$  signal bits switch in the same direction, where  $N$  is even.

The circuitry of the receiver is relatively simple because it only needs to conditionally invert the receiving data to get a correct data value. If  $N$  is odd, the receiving data need to be inverted only when the invert line is high. If  $N$  is even, the receiving data need to be inverted only when the invert line has a transition.

For today's high-performance circuits, there are typically only 14 to 16 FO4 (fanout-of-four inverter [6]) delays per clock [11]. Hence, the delay overhead introduced by our encoder should be minimized. Let  $d_{AND2}$ ,  $d_{OR2}$ , and  $d_{XOR2}$  be the delay of a two-input AND gate, that of a two-input OR gate, and that of a two-input XOR gate, respectively. For

an  $N$ -bit bus, the critical path delay  $D(N)$  of our  $N$ -bit bus encoder is given by

$$D(N) = d_{\text{Codeword Generator}} + d_{\lceil \frac{N+1}{2} \rceil\text{-out-of-}N \text{ Majority Voter}} + d_{OR2} + d_{XOR2} \quad (2)$$

where  $d_{\text{Codeword Generator}}$  equals the delay of an inverter  $d_{INV}$  plus  $d_{AND2}$  (i.e.,  $d_{\text{Codeword Generator}} = d_{INV} + d_{AND2}$ ), and the delay  $d_{\lceil (N+1)/2 \rceil\text{-out-of-}N \text{ Majority Voter}}$  is given by  $\log_{3/2} N * d_{\text{full adder}}$  (delay of a full adder) since we use a full adder tree to implement the majority voter. Therefore, for a typical 8-bit bus encoder with optimized logic and the full-adder circuit implemented as a mirror-type adder, the critical path of the encoder has a delay of ten FO4, which is about two thirds of the clock cycle time. This delay overhead is similar to that of the low-power bus-invert method. Nevertheless, this delay overhead is the "worst case" scenario. Since the encoding logic could be fused with the logic of the IP block, this delay overhead could be reduced with the simultaneous optimization of encoding and the IP block logic.

Like the bus-invert method, our method can also reduce the bus transitions. The reduction of the bus transition count occurs when there are  $\lceil (N + 1)/2 \rceil$  bits that transit in the same direction. For this case, our encoder will invert the current data and the transition count will be reduced. Take an 8-bit bus as an example. For the transition pattern (00↑↑↑↑) before encoding, the transition count is five. After

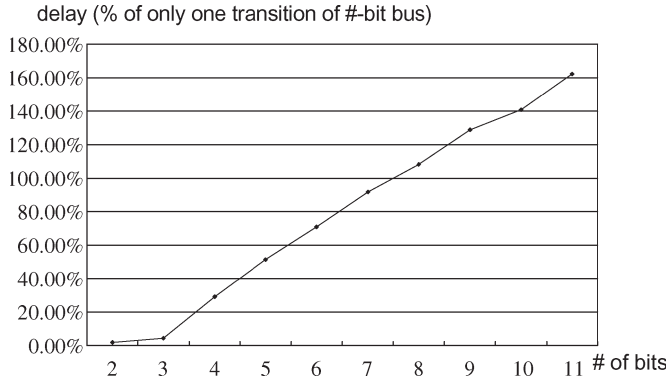


Fig. 6. Worst case #-bit bus delay (percentage of the delay of only one transition pattern of the #-bit bus) with # varying from 2 to 11.

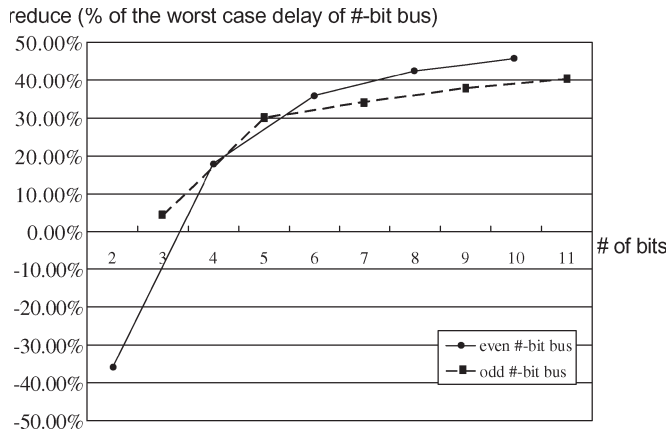


Fig. 7. Reduction of worst case delay of #-bit bus by using the bus-invert method with # varying from 2 to 11.

encoding, the transition pattern changes to ( $\uparrow\uparrow 000000(\uparrow)$ ) or ( $\downarrow\downarrow 000000(\uparrow)$ ), and the transition count is reduced to three (the additional transition ( $\uparrow$ ) is due to the signal transit on the invert line). Therefore, our encoding scheme can also reduce the average power consumed by the bus in terms of the average transition count. However, the peak power dissipation after encoding will remain the same. For the transition pattern ( $\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow$ ) before encoding, our encoder will not invert the current data (i.e., the transition pattern will remain the same after encoding, and this transition pattern causes the peak power consumed by the 8-bit bus due to the coupling capacitance between wires). Since the oppositely switching signals are good for reducing the inductive coupling delay, our encoder will keep these transitions unless there are  $\lceil(N+1)/2\rceil$  bits that transit in the same direction. However, the oppositely switching signals are the worst for the power consumption when considering the capacitive coupling effects. Therefore, the peak power will remain the same after using our encoding scheme.

## VI. SIMULATION RESULTS

### A. Bus Coupling Delay Reduction

With the parameters given in Section II, we conducted our simulations by varying bus signal bits with or without using the proposed bus-invert method. The simulation results are shown in Figs. 6 and 7.

From Fig. 6, we observe that coupling inductance has greater impacts on bus delay as the number of bus bit lines increases. For a tight LC cross-coupled bus, as shown in Fig. 6, the increase (in percent) of the worst case switching delay grows about linearly with the number of bus bit lines. Hence, for a high-frequency tight LC cross-coupled

TABLE IV  
REDUCTION OF WORST CASE NOISE BY USING THE BUS-INVERT METHOD FOR BUS WIDTHS RANGING FROM 2 TO 11

#-bit bus	Max swing (V)	Max swing after encoding (V)	Noise reduction (%)
2	1.59	1.33	19.55%
3	1.63	1.44	13.19%
4	1.67	1.41	18.44%
5	1.71	1.47	16.33%
6	1.72	1.47	17.01%
7	1.74	1.49	16.78%
8	1.74	1.46	19.18%
9	1.76	1.51	16.56%
10	1.75	1.49	17.45%
11	1.76	1.52	15.79%
Average noise reduction			17.03%

bus, the delay due to signals simultaneously switching in the same direction should be considered.

As shown in Fig. 7, our encoding method can significantly reduce the worst case switching delay; in other words, the bus performance can be improved. Besides, our encoding method can obtain an even better reduction rate as the number of bus bit lines increases. However, since the encoder architectures for even-bit and odd-bit buses are slightly different, the delay reductions are also a little different. For an  $N$ -bit bus, if  $N$  is odd, the worst case switching pattern after encoding is  $(N+1)/2$  signal bits switching in the same direction including the INV line. For when  $N$  is even, the worst case pattern after encoding is that only  $N/2$  signal bits switch in the same direction, including the INV line. Hence, the reduction curve of even-bit buses is above that of odd-bit buses when the number of bits is larger than five (see Fig. 7). We should also note that for the 2-bit bus, our encoding method will worsen the worst case delay because the additional INV line will introduce large additional coupling to the victim line. In other words, the delay of the worst case after encoding for 2-bit lines plus one INV line will be larger than the worst case for only 2-bit lines.

In addition to reducing the worst case delay, our method has the side effects of decreasing the maximum ground bounce and eliminating the maximum inductive noise. For example, as shown in Table IV, the average reduction of maximum inductive noise is about 17%. Since the ground bounce and the inductive noise are also worst when all signal wires switch in the same direction, our method can also reduce these effects.

### B. Delay Overhead of the Bus Encoder

To investigate the delay introduced by our bus encoder and the coupling delay reduction by using our encoding method, we conduct the following simulations to show the delay reduction considering the delay overhead of the encoder for different technology nodes.

The parameters used are adopted from the 1997 National Technology Roadmap for Semiconductors (NTRS'97) [17] and the simulation results in [4]. These parameters are shown in Table V. We consider a typical 8-bit bus with the total routing length of half perimeter of a chip and four times of the minimum wire width and spacing. The length of each wire segment between two buffers is  $3000 \mu\text{m}$ . In addition, the buffers are sized to maintain equal input and output transition times, which is a classical design criterion for buffer sizing. The simulation results are listed in Table VI. Column 2 shows the half perimeter of a chip according to the chip area reported in Table V, assuming that chips are of the square shape and thus the half perimeter of a chip is  $2\sqrt{\text{Area}}$ . Column 3 lists the number of required wire segments for signals passing through the half perimeter of a chip [i.e., the half

TABLE V  
INTERCONNECT AND DEVICE PARAMETERS USED

Technology (nm)	180	150	130	100	70
Across chip clock (MHz)	1200	1400	1600	2000	2500
Projected clock period (ps)	833	714	625	500	400
Area (mm <sup>2</sup> )	340	385	430	520	620
V <sub>dd</sub> (V)	1.65	1.35	1.35	1.05	0.75
Metal resistivity $\rho$ ( $\mu\Omega$ -cm)	2.2	2.2	2.2	2.2	1.8
Dielectric constant	2.75	2.25	1.75	1.75	1.5
4X min. wire width (nm)	720	600	520	400	280
4X min. wire spacing (nm)	960	840	680	560	400
Metal aspect ratio	1.8:1	2.0:1	2.1:1	2.4:1	2.7:1
Via aspect ratio	2.2:1	2.4:1	2.5:1	2.7:1	2.9:1
FO4 (ps)	51.1	48.7	45.8	39.2	21.9
Buffer input cap. (fF)	0.60	0.55	0.425	0.35	0.21
Buffer $R_d$ (K $\Omega$ )	3.72	4.52	4.50	4.78	4.84

TABLE VI  
SIMULATION RESULTS OF THE COUPLING DELAY REDUCTION BY USING OUR ENCODING METHOD AND THE DELAY OVERHEAD OF THE ENCODER FOR DIFFERENT TECHNOLOGY NODES

	Half perimeter (mm)	# of req. wire segments	Delay overhead (10FO4) (ps)	Delay gain – 1 wire seg. (ps)	Delay gain – full chip (ps)	Overall delay gain ((full chip-overhead)/overhead) (%)	Noise reduction (%)
180nm	18.44 X 2	12	511	51.324	615.888	20.53	32.57
150nm	19.62 X 2	13	487	43.594	566.722	16.37	31.97
130nm	20.74 X 2	14	458	34.464	482.496	5.35	31.68
100nm	22.80 X 2	15	392	38.333	574.995	46.68	31.19
70nm	24.90 X 2	17	219	34.442	585.514	167.36	30.34

perimeter of a chip  $2\sqrt{\text{Area}}$  (millimeter) divided by the length of a wire segment which is 3 mm]. Column 4 shows the delay overhead induced from our 8-bit bus encoder. The delay gains (the worst case delay of the bus before encoding minus that after encoding) of the signals passing through one wire segment (3000  $\mu\text{m}$ ) and through half perimeter of a chip are shown in Columns 5 and 6, respectively. The overall delay gains [((Column 6 – Column 4)/Column 4)  $\times$  100%] are given in Column 7. Finally, noise reduction is shown in Column 8.

Columns 2 and 3 in Table VI reveal the increase of the chip size as the technology advances. Since the intrinsic gate delay decreases as the feature size shrinks, the delay overhead of our encoder decreases as well (see Column 4). We report the worst case delay overhead derived in Section V in our simulations. From Columns 5–7, we observe that the overall delay gain tends to increase as the technology advances although the delay gain of each wire segment decreases. For example, as shown in our simulations, the delay gain for signals passing through the half perimeter of a chip is only about 20% for the 0.18- $\mu\text{m}$  process while this gain increases to about 167% for the 0.07- $\mu\text{m}$  process. The reasons are twofold: 1) the decrease of the intrinsic gate delay and 2) the increase of the chip size. To further improve the overall delay gain by reducing the delay overhead, designers can also use dynamic logic to implement the encoding circuit. In addition to the coupling delay reduction, our method can also reduce the maximum inductive coupling noise for long-interconnects by about 30%. The simulation results are shown in Column 8.

## VII. CONCLUSION AND DISCUSSIONS

In this paper, we have shown that the inductance effect has changed the worst case switching pattern with the maximum bus delay. For a 5-bit bus structure, the worst case switching pattern is  $*\uparrow\downarrow*$  or  $*\downarrow\uparrow*$  considering  $RC$  effects, but the worst case pattern changes to  $\uparrow\uparrow\uparrow$  or  $\downarrow\downarrow\downarrow$  considering  $RLC$  effects. Hence, we shall consider both the  $RC$  and the  $RLC$  effects to derive effective encoding schemes for bus delay optimization.

We have also conducted simulations considering  $RLC$  effects on the bus structure when the wire capacitance becomes dominant. We have observed that the worst case switching pattern is also different from the one considering  $RC$  effects. The difference is due to the long-range inductive coupling.

We have also proposed a bus-invert method to reduce the worst case on-chip bus delay with the dominance of the inductance coupling effect. Simulation results have shown that our encoding method can significantly reduce the worst coupling delay of a bus. In the future, we intend to develop a more sophisticated bus-invert scheme to further reduce the inductive coupling delay.

Our encoding scheme is recommended for cases when buses or parallel signal wires are about thousands of micrometers long and work above gigahertz frequencies. At such working frequencies, the gate delay overhead of our encoder should be small enough. If we choose the full-adder tree to implement the majority voter, the delay of the majority voter is  $O(\log_{1.5} N)^*$  (full-adder delay), where  $N$  is the total number of signal bits of a bus. In other words, if  $N$  is very large, our encoder may cause timing violations. To solve this problem, we can divide the original bus into subbuses by inserting ground wires between subbuses. Hence, the overall problem is a gate-delay-dependent (and thus process-dependent) optimization problem. Therefore, we shall solve this problem in our future work.

It should be noted that our encoding method is not optimal. However, it is very simple yet efficient, and thus the encoder and decoder logics are also very easy for implementation. Therefore, the delay and the power overhead of the encoder and decoder logics are minor compared to the delay and the power consumption of the bus. It needs further investigation for the possible optimal encoding scheme, and it could be a possible direction of our future work. We believe that the resulting “optimal” encoder and decoder would be much more complex than ours and might use more than one pipeline stage to encode/decode data.

The worst case switching pattern, as pointed out in this paper, could be varied with the dimensions and the working frequency of a bus.

Hence, to develop a flexible encoding scheme that can cope with the varying worst case patterns, one potential method is to conduct complete HSPICE simulations for all switching patterns according to the working frequency and the extracted RLC model of a bus to find the real worst case switching pattern. After simulation, all transition delays between any two data patterns can be measured. Then, we can develop an appropriate bus encoding method to avoid the patterns that violate the delay constraint. However, it is very time consuming to conduct complete HSPICE simulations and may suffer from the memory explosion problem for wide buses (for an  $n$ -bit bus, there are totally  $2^n$  data patterns and  $4^n/2$  transition patterns). Therefore, identifying the real worst case switching pattern of a bus efficiently is also a desirable research topic before the development of a flexible bus encoding scheme.

#### REFERENCES

- [1] K. H. Baek, K. W. Kim, and S. M. Kang, "A low energy encoding technique for reduction of coupling effects in SOC interconnects," in *Proc. 43rd IEEE Midwest Symp. Circuits and Systems*, Lansing, MI, Aug. 2000, pp. 80–83.
- [2] D. K. Cheng, *Field and Wave Electromagnetics*, 2nd ed. Reading, MA: Addison-Wesley, 1989.
- [3] Y. Cao, X. Huang, N. H. Chang, S. Lin, O. S. Nakagawa, W. Xie, D. Sylvester, and C. Hu, "Effective on-chip inductance modeling for multiple signal lines and application to repeater insertion," in *Int. Symp. Quality Electronic Design*, San Jose, CA, Mar. 2001, pp. 185–190.
- [4] J. Cong, "An interconnect-centric design flow for nanometer technologies," *Proc. IEEE*, vol. 89, no. 4, pp. 505–528, Apr. 2001.
- [5] M. H. Chowdhury, Y. I. Ismail, C. V. Kashyap, and B. L. Krauter, "Performance analysis of deep sub micron VLSI circuits in the presence of self and mutual inductance," in *IEEE Int. Symp. Circuits and Systems*, Scottsdale, AZ, 2002, pp. 197–200.
- [6] D. Chinnery and K. Keutzer, *Closing the Gap Between ASIC and Custom—Tools and Techniques for High-Performance ASIC Design*. Boston, MA: Kluwer, 2002.
- [7] M. A. Elgamel and M. A. Bayoumi, "Interconnect noise analysis and optimization in deep submicron technology," *IEEE Circuits Syst. Mag.*, vol. 3, no. 4, pp. 6–17, 2003.
- [8] R. Escovar and R. Suaya, "Optimal design of clock trees for multigigahertz applications," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 3, pp. 329–345, Mar. 2004.
- [9] L. He and K. M. Lepak, "Simultaneous shield insertion and net ordering for capacitive and inductive coupling minimization," in *Int. Symp. Physical Design*, San Diego, CA, 2000, pp. 55–60.
- [10] K. Hirose and H. Yasuura, "A bus delay reduction technique considering crosstalk," in *Proc. Design Automation and Test Eur. (DATE)*, Paris, France, Mar. 2000, pp. 441–445.
- [11] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wire," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.
- [12] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of merit to characterize the importance of on-chip inductance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 4, pp. 442–449, Dec. 1999.
- [13] Y. I. Ismail, "On-chip inductance cons and pros," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 6, pp. 685–694, Dec. 2002.
- [14] M. Kamon, M. J. Tsuk, and J. K. White, "FastHenry: A multipole-accelerated 3D inductance extraction program," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 9, pp. 1750–1758, Sep. 1994.
- [15] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 10, no. 11, pp. 1447–1459, Nov. 1991.
- [16] M. R. Stan and W. P. Bursleson, "Bus-invert coding for low-power I/O," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 1, pp. 49–58, Mar. 1995.
- [17] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1997.
- [18] —, *International Technology Roadmap for Semiconductors*, 2003.
- [19] P. P. Sotiriadis and A. P. Chandrakasan, "Reducing bus delay in submicron technology using coding," in *Proc. Asia and South Pacific Design Automation Conf.*, Yokohama, Japan, Feb. 2001, pp. 109–114.
- [20] L. Trevillyan, D. Kung, R. Puri, L. N. Reddy, and M. A. Kazda, "An integrated environment for technology closure of deep-submicron IC designs," *IEEE Des. Test. Comput.*, vol. 21, no. 1, pp. 14–22, Jan./Feb. 2004.
- [21] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in *Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 2001, pp. 57–63.

### Modeling the Driver Load in the Presence of Process Variations

Janet M. Wang, Jun Li, Satish Yanamanamanda, Lakshmi K. Vakati, and Kishore K. Muchherla

**Abstract**—Feature sizes of less than 90 nm and clock frequencies higher than 3 GHz calls for fundamental changes in driver-load models. New driver-load models must consider the process variation impact of the manufacturing procedure, the nonlinear behavior of the drivers, the inductance effects of the loads, and the slew rates of the output waveforms. The present deterministic driver-load models use the conventional deterministic driver-delay model with a single  $C_{\text{eff}}$  (one ramp) approach. Neither the statistical property of the driver nor the inductance effects of the interconnect are taken into consideration. Therefore, the accuracy of existing models is questionable. This paper introduces a new driver-load model that predicts the driver-delay changes in the presence of process variations and represents the interconnect load as a distributed resistance, inductance and capacitance (RLC) network. The employed orthogonal polynomial-based probabilistic collocation method (PCM) constructs a driver-delay analytical equation from the circuit's output response. The obtained analytical equation is used to evaluate the driver output delay distribution. In addition, the load is modeled as a two-effective-capacitance in order to capture the nonlinear behavior of the driver. The lossy transmission line approach accounts for the impact of the inductance when modeling the driving-point interconnect load. The new model shows improvements of 9% in the average delay error and 2.2% in the slew rate error over the simulation program with integrated circuit emphasis (SPICE) and the one ramp modeling approaches. Compared with the Monte Carlo method, the proposed model demonstrates a less than 3% error in the expected gate delay value and a less 5% error in the gate delay variance.

**Index Terms**—Driver equivalent resistance, inductance effect evaluation criteria, interconnect driving-point admittance, multiple effective capacitance, probability collocation method (PCM), process variation.

#### I. INTRODUCTION

As technologies advance beyond the deep submicrometer (DSM) regime, design for manufacturability (DFM) issues are moving into the mainstream with unexpectedly low yields starting at the 130-nm process node. At 90 nm and below, DFM issues are the major factors affecting the speed of production ramps and the profitability of semiconductor companies.

Manuscript received June 15, 2004; revised October 24, 2004 and May 2, 2005. This work was supported in part by the National Science Foundation under Grant NSF-345090. This paper was recommended by Associate Editor L. Scheffer.

J. M. Wang is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721-0104 USA (e-mail: wml@ece.arizona.edu).

J. Li is with Anova Solutions Inc., San Jose, CA 95054 USA (e-mail: junli@anova-solutions.com).

S. Yanamanamanda, L. K. Vakati, and K. K. Muchherla were with the University of Arizona, Tucson, AZ 85721, USA. They are now with Micron Technology, Boise, ID 83716 USA (e-mail: sashy@gmail.com; kalpana@email.arizona.edu; muchherla@gmail.com).

Digital Object Identifier 10.1109/TCAD.2005.862739