Minimization of Crosstalk Noise, Delay and Power using a Modified Bus Invert Technique

Matheos Lampropoulos, Bashir M Al-Hashimi and Paul Rosinger
School of ECS, University of Southampton, United Kingdom
{ml02r, bmah, pmr }@ecs.soton.ac.uk

Abstract

Bus encoding benefits from reduced routing overhead and simpler bus drivers and receivers when compared to other techniques proposed for dealing with crosstalk problems. While all previously reported bus encoding approaches reduce crosstalk delay, they ignore the effects of inductive coupling between the bus lines, i.e. crosstalk noise, which can cause malfunctions in high speed SoCs buses. Aiming to solve this issue, this paper presents a modified bus-invert technique, which minimizes crosstalk noise, as well as delay and power, when compared to the unencoded bus, at the expense of a small area overhead.

1 Introduction

With the increasing gate densities and the shrinking geometries, System on a Chip (SoC) interconnects are getting longer and thiner while the core frequencies reach the GHz range. The result is that the effects of coupling capacitances and inductance become more significant, creating crosstalk problems which affect the reliability of digital systems.

The increased capacitive coupling leads to undesirable signal speed-ups and slow-downs on interconnects, effect known as crosstalk delay. On the other hand, extensively high overshoots and undershoots, known as crosstalk noise, can cause malfunctions due to setup and hold violations or false triggering of clock lines. Moreover, overshoots and undershoots with high amplitude can also permanently damage the protection diodes within a chip due to increased stressing of the silicon junctions [5]. Therefore, the problem of interconnect crosstalk is a critical issue in designing reliable high speed chips. Consequently, several techniques such as shielding [7, 8], crosstalk aware routing [2, 13], buffer insertion [1, 4], differential signaling [14] and bus encoding [11, 18], have been proposed for reducing the crosstalk effects. An overview of these techniques is given.

Shielding is one of the most effective ways for reducing crosstalk, in which dedicated ground or Vdd wires are inserted between the signal lines. In [7, 8] the concept of active shielding was analyzed, where the shields are not stuck at a constant voltage, but are switching in order to reduce capacitive and inductive effects. While this technique

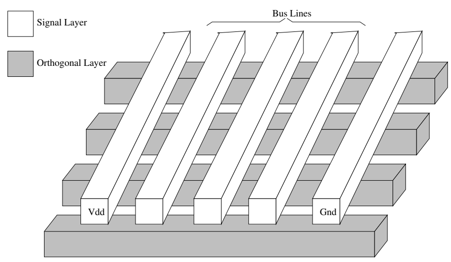


Figure 1. Coplanar structure for the bus interconnects

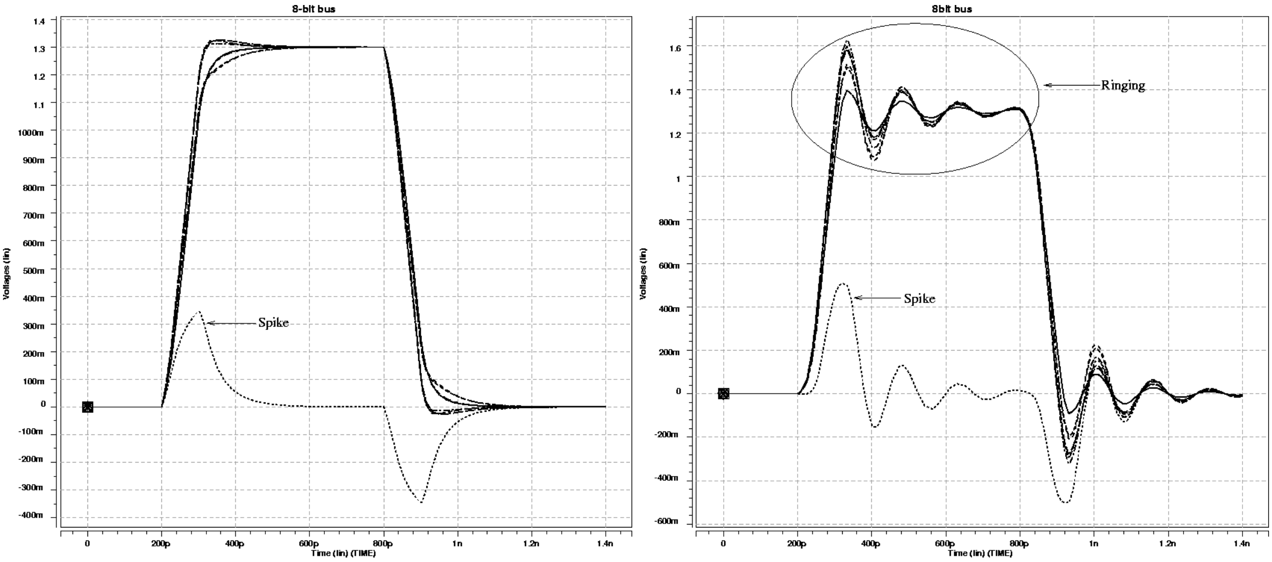
can effectively reduce crosstalk, shielding all the bus lines requires a large amount of area and also increases routing congestion.

Crosstalk aware routing is another approach which aims to optimize the relative positions of critical nets so that coupling between overlapping segments to be minimized [2, 13]. These nets can be reassigned to different metal layers or the distance between them can be increased. For large designs, crosstalk aware routing becomes very difficult, because of the complexity of the circuit and the limitations of the routing area.

Buffer insertion is another method for minimizing crosstalk in long interconnects. This technique divides the interconnects into smaller segments and inserts buffers between them to restore the signal. Although the length of parallel lines, and thus the coupling capacitances between the lines, can be minimized, there are other issues which must be also considered. A non-sofisticated use of buffers can lead to a significant increase in the final propagation delay, therefore optimized buffer insertion algorithms are necessary [1]. While this technique is suitable for critical signals (e.g clock), for long bus lines this solution may be costly in terms of area and power consumption.

Y. Massoud et al. in [14], evaluated the use of *low voltage differential signals* (LVDS) for minimizing inductive noise. Experiments with a distributed RLC model shows the superiority of this method over other techniques. However, differential signaling doubles the number of bus lines and also requires complex drivers and receivers.

Bus encoding is another approach which addresses the problem of interconnect crosstalk. The basic idea is that the patterns which are more likely to create crosstalk problems,



(a) Simulation for the pattern $(\uparrow\uparrow\uparrow - \uparrow\uparrow\uparrow\uparrow)$ in the RC model (b) Simulation for the pattern $(\uparrow\uparrow\uparrow - \uparrow\uparrow\uparrow\uparrow)$ in the RLC model

Figure 2. Comparison between the RC and RLC model. ($\uparrow$ 0->1, $\downarrow$ 1->0, - idle line)

are transformed into "crosstalk free" patterns. The bus invert technique presented in [17], was used for minimizing power consumption in the bus lines. Further attempts tried to consider also the coupling effects between lines as these increase the amount of power consumed on the bus. The coupling driven encoding scheme proposed in [9] is based on the bus invert technique but also targets the reduction of the coupled transition activity. The authors use an RC model for the interconnects to examine how the power consumption in the bus is affected by the switching relation between adjacent lines. According to the RC model, opposite transitions on adjacent lines (01->10 and 10->01) cause the biggest charging in coupling capacitances, and thus the highest energy consumption. The authors in [18] proposed the use of "shelf shielding" codes and provided a theoretical analysis for their performance. The idea is that the data patterns are encoded in such a way that no opposite skews should be found in adjacent bits. The number of redundant bus lines required in this case is less than for the total shielding approach. In [12] the authors described a shielding code which eliminates crosstalk delay and also reduces sufficiently the power consumption in the bus. However, the high complexity of the algorithm makes it unsuitable for buses wider than 4 bits. The bus encoding scheme presented in [10] uses an adaptive codebook, which makes it more effective than the bus-invert code in reducing the capacitive coupling activity, at the expense of the area overhead required for the codebook.

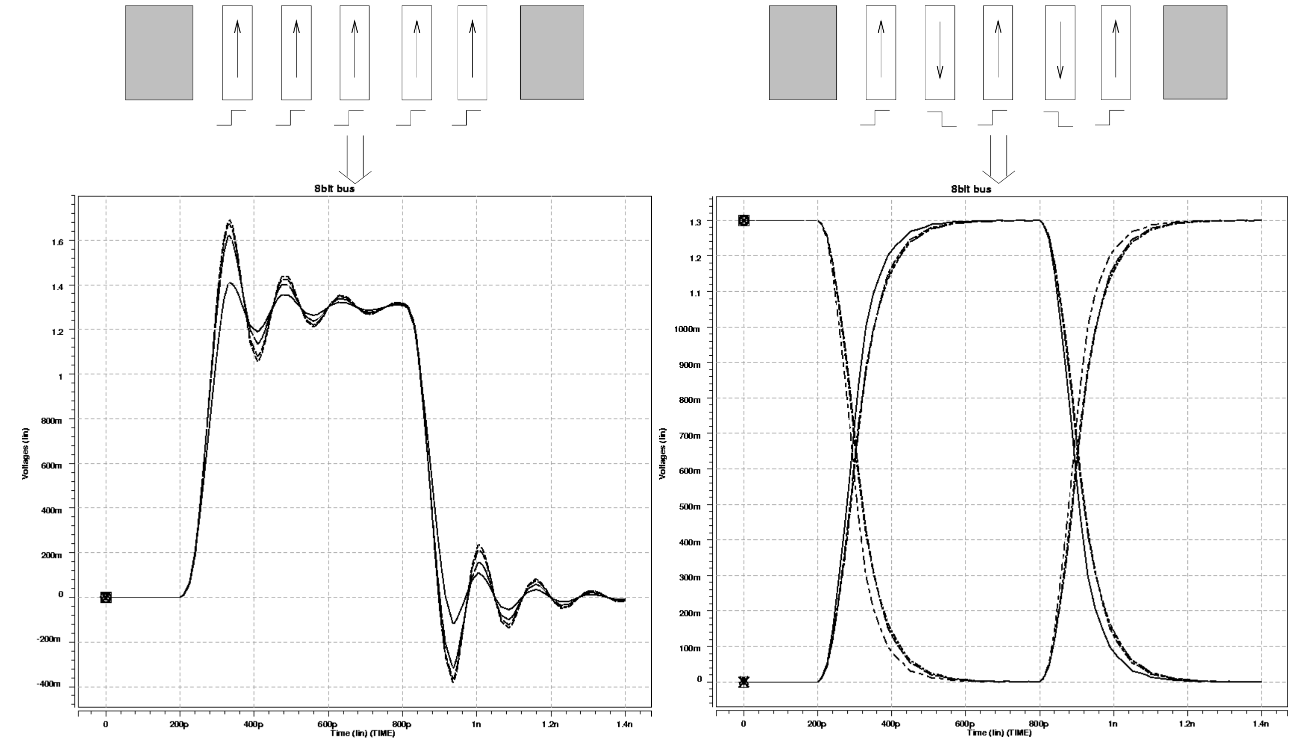
This paper proposes a modified bus invert technique for minimizing crosstalk noise. We have chosen the bus invert approach because it has the minimum routing overhead when compared to other crosstalk minimization techniques, which makes it suitable for high density chips.

2 Crosstalk noise

In order to analyze the effect of inductive coupling between bus lines at high frequencies, we have performed HSPICE simulations for an RC and an RLC model of an 8-bit bus. The coplanar interconnect structure shown in Figure 1 was used in all our simulation setups. The lack of ground plane in these structures, which corresponds to SoC designs, makes the inductive effects more severe and traditional methods for reducing it, (like increasing the spacing between lines) are not so effective any more [3].

The simulation waveforms shown in Figure 2(a) correspond to the RC model, while those shown in Figure 2(b) correspond to the RLC model. For these simulations, Vdd was set to $1.3V$ and the frequency to $1GHz$. All the bus lines are simultaneously switching from logic "0" to logic "1" and back to "0", except line 4 which is stuck to logic "0". This particular pair of patterns was chosen because it shows the effects of inductive coupling on both the switching and the stable bus lines. From the two figures it can be observed that the RC model grossly underestimates crosstalk noise due to inductive coupling between lines. The RC waveforms show only spikes on the stable signals with a peak value of almost $350mV$ and insignificant noise on the switching signals. However, the inductive coupling between lines causes a ringing with a peak value of almost $1.6V$ on the switching lines and spikes on the stable lines with a peak value up to $500mV$.

Given the small supply voltage, the high amplitude spikes and ringing due to inductive coupling between bus lines reduce the noise margin, which can cause malfunctions in the digital circuits. Therefore, in high-speed designs inductance should be considered when encoding data for crosstalk noise minimization, which represents the mo-



(a) Signals switching in the same direction (↑↑↑↑↑↑↑↑) - Even mode

(b) Opposite switching signals (↑↓↑↓↑↓↑↓) - Odd mode

Figure 3. Simulations with the RLC model for the even and odd mode transitions

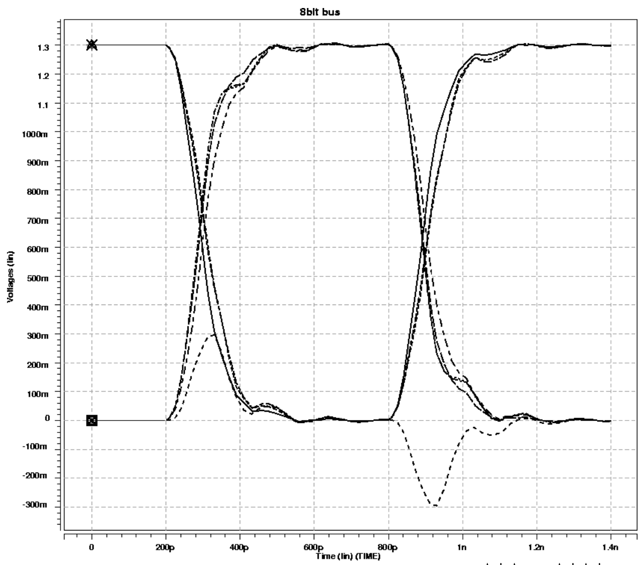


Figure 4. Simulation with the RLC model for ↑↓↑ − ↑↓↑↓

tivation of the this paper.

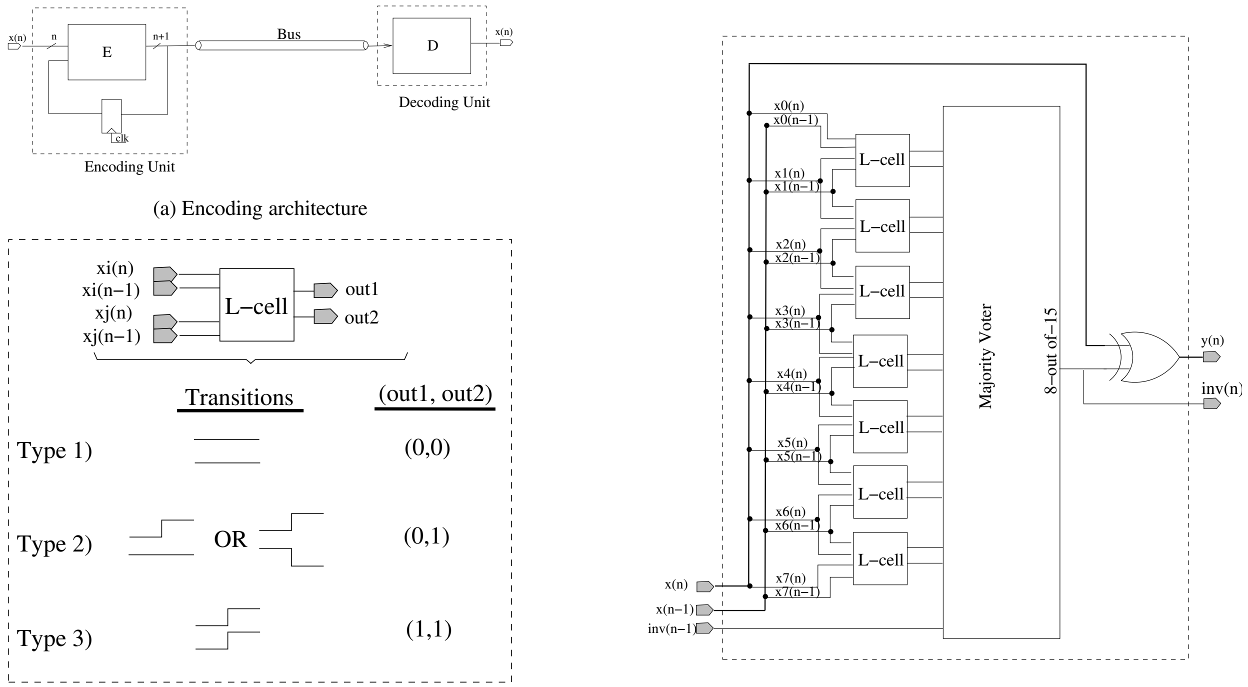
Unlike the RC model, in the RLC model the long-range inductance effects rise the necessity of considering all the switching signals and not only the adjacent lines. In order to find the relation between data patterns and crosstalk noise, the two extreme cases are examined. In Figure 3(a) all the lines switch from zero to one (even mode) and the current in the conductors flows in the same direction. In this case the magnetic fields reinforce each other, making the crosstalk noise more severe. In Figure 3(b) the lines switch from zero to one and from one to zero alternatively (odd mode). In this case the current flows in the opposite direction in adjacent lines, thus canceling the magnetic fields and reducing the crosstalk noise. The simulations with a distributed RLC model certified this observation. As shown in Figure 3(a), ringing has a peak value of almost $1.7V$, while in Figure 3(b) there is no ringing at all.

Figures 2(b) and 4 show the effect of the even and respectively odd patterns on line 4 which is held at logic “0”. The even mode causes a spike with a peak value of $500mV$ on the stable line (Figure 2(b)), while in the odd mode (Figure 4), the peak value of the spike is reduced to $300mV$.

3 Proposed bus invert technique

The above examples showed that when adjacent lines switch in the same direction (Figures 2(b) and 3(a)), the crosstalk noise is significantly high, when considering inductive effects. Opposite skews can overcome this problem thus minimizing ringing and spikes (Figures 3(b) and 4). Based on this observation we propose a modified bus invert (MBI) technique which aims to reduce crosstalk noise. The basic idea is to selectively invert data patterns such that the number of transitions in the same direction is minimized.

As illustrated in Figure 5(a), in the proposed approach the encoded data sent on the bus is a function (E) of the pre-



(a) Encoding architecture

(b) L-cell function

(c) Encoder

Figure 5. 8-bit encoder for the proposed encoding scheme

vious encoded data and the current unencoded input. If n is the number of the bits in the encoder input, the bus width is $n+1$ where the extra line carries the "invert signal" which is used by the decoding unit (D) to restore the original data. A schematic of the encoder for an 8-bit bus is shown in Figure 5(c). The bus lines are partitioned into pairs and each pair of adjacent lines as well as their values from the previous clock cycle drive the inputs of a logic cell (L-cell in Figure 5(c)), which encodes the type of events occurring on the pair of bus lines. The functionality of the L-cell is shown in Figure 5(b). If the transitions happen in the same direction then the 2-bit output of the logic cells becomes: *11* (*Type 3* in Figure 5(b)). In cases when both lines are idle the output is: *00* (*Type 1*), while for the patterns where either only one line switches, or both lines switch but in opposite directions the output is: *01* (*Type 2*). The L-cell assigns higher weights (11) to transitions which happen in the same direction and lower weights to the remaining transition types. In the case where there is no transition in any line, the weight is the minimum (00). For an n bit bus we need $n-1$ logic cells and a majority voter with $2n-1$ inputs. The majority voter takes the outputs of the logic cells and the previous invert signal and decides for the current invert signal. The invert signal is set to 0 when the count of $1's$ on the majority voter inputs is less than n and in 1 otherwise. For example, if the data patterns $x(n)$ and $x(n-1)$ in the encoder are 10101110 and 11010001 respectively and the previous invert signal is 0, then the the count of $1's$ on the majority voter inputs will be $9(>n)$ and the $x(n)$ pattern will be inverted ($inv(n)=1, y(n)=01010001$). While in the unencoded data there are five *Type 2* and two *Type 3* transitions, in the encoded data there are six *Type 1* and one *Type 2* transitions. The *Type 3* transitions are eliminated in this case, hence leading to less noise on the bus lines. On the receiver side the decoder recovers the initial values simply by inverting the data.

4 Experimental results

The validity of the proposed encoding scheme was certified with HSPICE simulations and the results will be presented in this section. Some details of the interconnect model and the setup of our experiments are given. The 3-D field solvers FastHenry [6] and FastCap [15] were used to extract the RLC values for the coplanar structure shown in Figure 1. Ground and Vdd lines are placed outside the bus in order to provide shielding from other signals. The RLC model used in our simulations is shown in Figure 6 (three lines are shown). The lines were partitioned in segments of $250um$ each, in order to have accurate results in high frequencies[1]. As there is no ground plane underneath the signal lines, no capacitances from the signal lines to the ground are included in the circuit. The capacitances between the signals and the orthogonal layer are transformed into extra parasitic capacitances between the signal lines.

[1] In [3] it was reported that the model in Figure 6 is accurate up to $4GHz$ for $250um$ long segments

More information about this model is found in [3]. The line length is $2000um$, which is a typical length for global interconnects for $0.18um$ technology [3]. Drivers with resistances of 50Ω are used for all the lines, Vdd is set to $1.3V$ and the simulation frequency to $1GHz$. The width and spacing characteristics are $W = 2.5um$ and $S = 1.25um$.

We applied MBI to a PDF and a Video file, 172 and 488 Kbytes respectively, and also to a pattern sequence containing all the possible transitions for an 8-bit bus. The PDF and the Video files represent realistic examples of data which might be transfered on a data bus. To compare the MBI with other bus-invert techniques, we have applied the same sequences for the bus-invert (BI) [17] and the coupling-driven bus-invert (CBI) [9] codes. These two encoding schemes have been selected because they have the same routing overhead with the MBI and they do not need probabilistic information for the data, thus they can be used for data buses with random data distributions. Two sets of experiments were performed, the first for measuring the crosstalk noise and delay on the bus and the second for measuring the power dissipation.

Table 1 shows the results for crosstalk[2], for the three pattern sequences (PDF, Video and all possible transitions for 8-bits). The colum NoEnc shows the absolute measured values for the unencoded data and the columns CI%, CBI%, MBI% show the relative reduction (%) with respect to the NoEnc case. As can be seen from Table 1, MBI provides significant reduction in crosstalk noise and improvement in delay performance when compared to the un-encoded bus. For example, reduction up to 33.33% in maximum ringing can be observed in the case of the All-Trans sequence. Similarly, reduction of 25.01% in maximum spike is observed for the case of the PDF file. While MBI does not specifically target crosstalk delay, improvement of up to 12.05% in maximum delay can be noted for the All-Trans sequence. This is because MBI eliminates the patterns which cause the worst case propagation delay in the bus. Such a pattern is for example the ↑↑↑↓↑↑↑, where the noise induced in the line switching from one to zero results in an extensive delay.

When compared to the BI scheme, MBI consistently attains better reduction in crosstalk noise. However, the BI scheme presents better delay performance than MBI. For example in the case of the All-Trans sequence, the BI scheme reduces the maximum crosstalk delay by up to 1.8% when compared to MBI. This is because the BI scheme reduces the maximum activity on the bus by half [17], and thus also the patterns causing the worst case delay. Moreover the BI scheme minimizes the cases where oppositely switching signals (e.g ↑↓↑↓↑↓↑) result in an increase in the

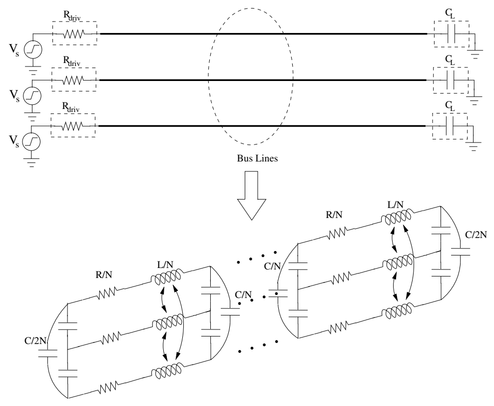


Figure 6. Distributed RLC interconnect model

propagation delay, due to the high capacitive coupling between the bus lines [8]. When comparing MBI with CBI, it can be seen that the proposed code obtains always lower delay and noise. Indeed, the CBI scheme increases the crosstalk noise when the RLC model is considered. In order to provide a fair comparison, we have used the same experimental setup for all three techniques. It should be noted that the experimental setup used in [9] was different than ours, as the authors used redundancy in time. Although time redundancy appears to be more effective than space redundancy for minimizing capacitive coupling, it increases the number of data transfer cycles.

We have also analyzed and compared the power performance of the MBI with BI and CBI. The % reduction in power with respect to the unencoded data is shown in table 2. The BI code has the best performance in terms of power, as it provides up to 19.13% reduction with respect to the unencoded data. MBI also reduces the average power dissipation on the bus by 11.21%, which is less than the BI and the CBI code. This is because the MBI does not specifically targets to eliminate transitions in opposite directions, which increase the coupling capacitances between the bus lines.

The proposed encoding scheme requires larger area overhead when compared to the BI scheme. However, the area required by the MBI encoder is smaller compared to the area of the CBI encoder. The only difference between the MBI and the CBI encoder is in the L-cell (Figure 5(b)). Implementations on the two versions of the L-cell synthesized with Synopsys Design Compiler showed that the proposed L-cell requires 13.4% less area that the CBI L-cell. Moreover the majority voter (Figure 5(c)) can be implemented with an analog circuit, which makes this technique suitable for high speed designs where the inductance noise plays a significant role in signal integrity. In [16] an analog implementation of the majority voter in a $0.5um$ technology gives a propagation delay of less than $1ns$, and therefore has a potential operating frequency higher than $1GHz$.

[2] The delay is measured between the 50% of the input and the far end value of the output, while the values for ringing and spikes are the peak-to-peak values

		NoEnc	BI%	CBI%	MBI%
PDF file	**MaxRing (V)**	0.676	23.88	-9.08	**28.89**
	MaxSpike (V)	0.598	21.67	-9.32	**25.01**
	MaxDelay (ps)	55.3	**13.40**	7.80	11.12
	AverRing (V)	0.534	24.67	-7.14	**28.14**
	AverSpike (V)	0.479	22.15	-5.18	**25.11**
	AverDelay (ps)	50.7	**10.53**	5.24	8.32
Video file	**MaxRing (V)**	0.713	26.41	-7.29	**31.67**
	MaxSpike (V)	0.641	20.73	-9.02	**24.10**
	MaxDelay (ps)	55.3	**14.08**	6.55	10.93
	AverRing (V)	0.559	26.97	-6.64	**30.39**
	AverSpike (V)	0.506	21.20	-6.15	**23.75**
	AverDelay (ps)	51.2	**11.26**	4.93	8.98
All-Trans	**MaxRing (V)**	0.720	27.36	-4.50	**33.33**
	MaxSpike (V)	0.663	21.11	-9.14	**24.33**
	MaxDelay (ps)	55.6	**14.49**	8.28	12.05
	AverRing (V)	0.541	25.14	-4.51	**29.08**
	AverSpike (V)	0.513	21.96	-5.44	**24.12**
	AverDelay (ps)	50.9	**10.94**	6.21	9.12

Table 1. Comparison in terms of crosstalk noise and delay(8-bit bus)

Patterns	BI[17] (%)	CBI[9] (%)	Proposed MBI (%)
PDF file	16.38	9.21	8.36
Video file	17.54	9.46	8.92
All-Trans	19.13	11.21	9.71

Table 2. Comparison in terms of power(8-bit bus)

5 Conclusion

We have shown that inductive coupling between bus lines creates high crosstalk noise, which can hamper the functionality of high-speed designs by reducing their noise immunity. In this paper, we have proposed a modified bus-invert technique which, unlike other bus encoding techniques, targets crosstalk noise minimization. Extensive experimental results show that the proposed technique, while performing comparably in terms of delay and power with other bus-invert techniques, achieves the highest reductions in crosstalk noise. Consequently, we belive the proposed technique represents a convenient solution for high-speed designs prone to noise-induced failures, which also adds the benefits of delay and power reductions.

References

[1] C. Alpert, A. Devgan, and S. Quay. Buffer insertion for noise and delay optimization. *IEEE Transactions on Computer-Aided Design*, 18(11):1633 –1645, Nov. 1999.

[2] Z. Chen and I. Koren. Crosstalk minimization in three-layer HVH channel routing. In *Proceedings of the IEEE Symposium on Defect and Fault Tolerance (DFT)*, pages 38 –42, Oct. 1997.

[3] C.-K. Cheng, J. Lillis, S. Lin, and N. Chang. *Interconnect Analysis and Synthesis*. 2000.

[4] S. Dubey and J. Jorgenson. Crosstalk reduction using buffer insertion. In *IEEE International Symposium on Electromagnetic Compatibility*, pages 639 –642, Aug. 2002.

[5] L. Green. Understanding the Importance of Signal Integrity. *IEEE circuits and devices magazine*, 15(6):7 –10, Nov. 1999.

[6] M. Kamon, M. Ttsuk, and J. White. FASTHENRY: a multipole-accelerated 3-D inductance extraction program. *IEEE Transactions on Microwave Theory and Techniques*, 42(9):1750 –1758, 1994.

[7] H. Kaul, D. Sylvester, and D. Blaauw. Active shielding of RLC global interconnects. In *Proceedings International Workshop on Timing issues in the specification and synthesis of digital systems*, pages 98 – 104, Dec. 2002.

[8] H. Kaul, D. Sylvester, and D. Blaauw. Active shields: a new approach to shielding global wires. In *Proceedings ACM Great Lakes Symposium on VLSI*, pages 112 – 117, Apr. 2002.

[9] K.-W. Kim, Kwang-Hyun-Baek, N. Shanbhag, C. Liu, and S.-M. Kang. Coupling-driven signal encoding scheme for low-power interface design. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 318 –321, Nov. 2000.

[10] S. Komatsu, M. Ikeda, and K. Asada. Bus Data Encoding with Coupling-driven Adaptive Code-book Method for Low Power Data Transmission. In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 18 –20, Sept. 2001.

[11] T. Lu, J. Henkel, H. Lekatsas, and W. Wolf. Enhancing signal integrity through a low-overhead encoding scheme on address buses. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 542 –547, 2003.

[12] C.-G. Lyuh and T. Kim. Low power bus encoding with crosstalk delay elimination. In *Proceedings IEEE International ASIC/SOC Conference*, pages 389 –393, 2002.

[13] L. Macchiarulo, E. Macii, and M. Poncino. Wire placement for crosstalk energy minimization in address buses. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 158 –162, 2002.

[14] Y. Massoud, J. Kawa, D. MacMillen, and J. White. Modeling and analysis of differential signaling for minimizing inductive crosstalk. In *Proceedings ACM IEEE Design Automation Conference*, pages 804 – 809, Mar. 2001.

[15] K. Nabors and J. White. FastCap: a multipole accelerated 3-D capacitance extraction program. *IEEE Transactions on Computer-Aided Design*, 10(11):1447 –1459, Nov. 1991.

[16] K. Nakamura and M. Horowitz. A 50% noise reduction interface using low-weight coding. In *Proceedings of the Symposium on VLSI Circuits*, pages 144 –145, June 1996.

[17] M. Stan and W. Burleson. Bus-invert coding for low-power I/O. 3(1):49 –58, 1995.

[18] B. Victor and K. Keutzer. Bus encoding to prevent crosstalk delay. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 57 –63, Nov. 2001.