

Area Efficient Temporal Coding Schemes for Reducing Crosstalk Effects

Jean-Marc Philippe
CEA-List DRT/DTSI/SARC/LCEI
F-91191 Gif-sur-Yvette
Jean-Marc.Philippe@cea.fr

Sébastien Pillement, Olivier Sentieys
IRISA - University of Rennes (ENSSAT)
6, rue de Kerampont 22300 Lannion, France
{pillemen, sentieys}@irisa.fr

Abstract—In this paper, we present some new crosstalk avoidance coding schemes devoted to on-chip busses. These schemes consist in encoding sequences of bits on each line of a bus transferring a packet in order to eliminate worst-case crosstalk patterns. They permit to improve the delay on the link at the cost of doubling the number of transmitted bits. The advantage of the presented solutions is that they have no wiring overhead, so they are independent from the bus bit-width. The coding schemes allow an increase of 50% of the data rate for a 1-mm bus. Moreover, the proposed solutions induce a direction in deep-submicron noise that can be used to implement a noise-tolerant system.

I. INTRODUCTION

In modern CMOS technologies, interconnects are the bottleneck of high-performance chips. Constraints (such as area or speed) on Systems-on-Chip (SoC) with deep submicron technologies require having high-speed and low area interconnects. Due to the increasing density of interconnects, to a lower voltage swing and to an increase of the aspect ratio [1], global on-chip busses suffer from large propagation delay. The main contribution to this delay is due to crosstalk, the interferences due to the coupling capacitance that exists between adjacent wires. The wire capacitances and the dimensions influencing crosstalk are summarized in figure 1. This phenomenon is directly influenced by the coupling capacitances (C_c) between the victim wire V and its two aggressors ($A1$ and $A2$). The coupling capacitance between two wires depends on technology constants but also on the dimensions of the wires, such as their length, thickness and the spacing between them.

Crosstalk increases the propagation delay on busses. It introduces a delay factor (g), as shown in Table I, where r is the ratio between the cross-coupling capacitance of two adjacent wires (C_c) and the capacitance of a wire to the substrate (C_s) as it is shown by equation 1.

$$r = \frac{C_c}{C_s} \quad (1)$$

In this table, \uparrow represents a rising transition, \downarrow represents a falling transition and $-$ means that there is no transition on the wire. In the best case, when the three wires are switching in the same direction, the delay on the victim wire is the delay without crosstalk (i.e. $g = 1$), but the bus clock cycle must be adapted exclusively regarding the worst-case delay (i.e. $g = 1 + 4.r$) to ensure the integrity of the transmitted data. For a plausible situation where $C_c = C_s$, the propagation delay

can be multiplied by five or more [2]. Other studies [3] use a parameter r up to 10.

Noise induced by crosstalk represents a second issue in deep submicron designs. The coupling capacitance between two adjacent wires introduces a permanent link between them. A transition on a wire affects the two adjacent wires by applying to them a voltage peak [4]. With the technology shrink, crosstalk has a more and more important part in the general noise level. This is due to the increasing coupling capacitance between adjacent wires. As a consequence, the voltage peak induced by cross-coupling is more and more important compared to the voltage swing on a bus line.

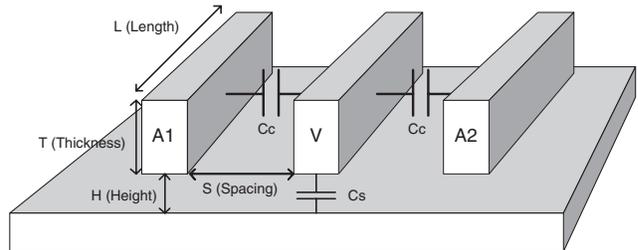


Fig. 1. A victim wire and two aggressors. The two wire capacitances are included in this figure.

TABLE I

EFFECTIVE CAPACITANCE (C_{eff}) AND DELAY FACTOR (g) OF THE VICTIM WIRE AND THE CORRESPONDING TRANSITION PATTERNS.

C_{eff}	Transition Patterns				g
C_s	($\uparrow, \uparrow, \uparrow$)	($\downarrow, \downarrow, \downarrow$)			1
$C_s + C_c$	($-, \uparrow, \uparrow$)	($-, \downarrow, \downarrow$)	($\uparrow, \uparrow, -$)	($\downarrow, \downarrow, -$)	$1+r$
$C_s + 2.C_c$	($-, \uparrow, -$)	($-, \downarrow, -$)			$1+2.r$
	($\uparrow, \uparrow, \downarrow$)	($\uparrow, \downarrow, \downarrow$)	($\downarrow, \uparrow, \uparrow$)	($\downarrow, \downarrow, \uparrow$)	
$C_s + 3.C_c$	($-, \uparrow, \downarrow$)	($-, \downarrow, \uparrow$)	($\uparrow, \downarrow, -$)	($\downarrow, \uparrow, -$)	$1+3.r$
$C_s + 4.C_c$	($\uparrow, \downarrow, \uparrow$)	($\downarrow, \uparrow, \downarrow$)			$1+4.r$

This paper introduces some simple coding schemes to avoid worst-case crosstalk patterns. The remainder of the paper is organized as follows. Section II quickly reviews some of the existing crosstalk avoidance techniques. Our approach is explained in section III. We present the experimental results in section IV and we make an analysis of noise issues in section V. Finally, section VI concludes this paper and presents future work.

II. RELATED WORKS ON CROSSTALK AVOIDANCE

The goal of crosstalk avoidance researches is to eliminate the worst-case patterns presented in Table I. The different solutions which can be found in the literature for reducing crosstalk effects can be roughly classified into two categories: passive and active solutions. Among passive solutions, shielding is the most famous technique. It consists in inserting a grounded line between every couple of wires to avoid worst-case transitions as it is shown in figure 2. In fact, all the transitions with two adjacent lines switching in opposite directions are removed. The drawback is that it doubles the needed wires. An evolution of this technique can be found in [5]. The authors propose to use a repeating pattern to route the signals instead that classical shielding. They use the pattern $VSGSVSGS\dots$, where V represents a V_{dd} wire, S represents a signal wire and finally G represents a grounded wire. Even if they propose a solution to remove wires, this scheme requires additional lines and then an increase of the interconnect area.

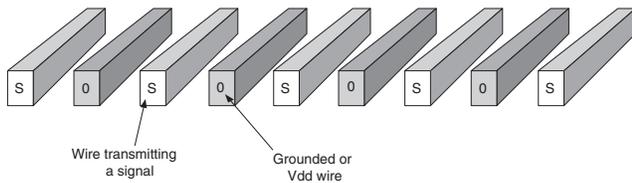


Fig. 2. Illustration of the shielding technique.

Duplication of each wire is also used to eliminate worst-case transitions. The acceleration of the signals provided by this technique is higher than the previous one because all the patterns with two invariant aggressors are removed. But this technique also doubles the number of wires.

Another solution is to skew selected bus signals to avoid switching transitions in the same time on adjacent wires [6]. It permits to avoid worst-case transitions and to speed-up busses. But the speed-up is limited due to the time extension needed to transmit a complete word and this technique requires a careful design of the transmitter and the receiver.

Active and more complex solutions have been studied in recent years. Coding is one of these promising approaches. The basic principle of coding is to transmit codewords instead of the original words. It has been shown in [7] that a 32-bit data word can be transmitted with 53 wires using partial coding with simple 3-bit to 4-bit memoryless coders separated by shielding wires. The codecs add latency to the system. In [3], the authors present a coding scheme which removes all $1 + 4.r$ and $1 + 3.r$ patterns. It has 62,5% bit overhead and a delay improvement of 50% (but the bus length is at least 1cm). In [8], the authors introduce an improvement of their coding scheme by eliminating $1 + 2.r$ patterns too. With an area overhead of about 200%, the scheme can significantly speed-up busses.

Another solution is introduced in [9]. It consists in having six groups of transition patterns with different propagation times and a fast clock. A crosstalk analyzer assigns two

consecutive words to one of the six delay groups and adapts the number of transmission cycles needed to send the word. This scheme does not eliminate crosstalk patterns but adapts the length of the transmission to them.

The methods that we propose have no wiring overhead and do not require careful routing of the clock tree. Basically, we use temporal coding to avoid opposite transitions on adjacent wires. The coding schemes are presented in the next section.

III. DESCRIPTION OF THE CODES

This paper proposes a new approach to eliminate worst-case patterns ($1 + 3.r$ and $1 + 4.r$). We use temporal coding instead of word-by-word coding. This technique codes the different wires independently and exhibits time overhead instead of wiring overhead. This simple coding can be used with very large busses as it is independent from the bus bit-width.

The main idea is to increase the number of transmitted bits on each line of a bus to code worst-case crosstalk free transitions. For example, a binary "0" can be inserted after each bit transmitted on each line as it is described in figure 3 (this technique is referred after as *Code 0*). This leads to transmit a binary word composed of only binary zeros after each word. This simple technique allows to remove the two last worst-case patterns at the cost of doubling the number of transmitted bits. As worst-case patterns are eliminated, the propagation delay is reduced, enabling the bus to transmit easily two bits instead of one. This simple method is systematic and does not require complex codecs. It is also totally independent of the bus bit-width.

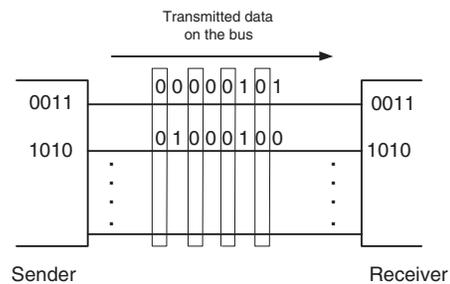


Fig. 3. Encoded data using a zero column after each bus word (*Code 0*).

The above technique can be modified to encode two bits instead of one (this technique is referred after as *Code 1*) to decrease the mean activity of the coded sequence. The packet is partitioned into 2-bit blocks which are encoded using the static correspondence described in Table II.

It is easy to see that there is no worst-case pattern inside codewords since the transitions are always rising transitions. Each coded word begins with a binary logical 0. It ensures that every transition between encoded words are falling or static transitions. So there is no worst-case pattern between encoded words. The worst-case pattern in this coding scheme is $(-, \uparrow, -)$. The corresponding effective capacitance of this pattern is $C_S + 2.C_C$ ($g = 1 + 2.r$). A drawback of this coding

TABLE II
CORRESPONDENCE BETWEEN THE ORIGINAL 2-BIT BLOCKS AND THE PROPOSED CROSSTALK AVOIDANCE CODING SCHEME CODE 1.

Original 2-bit block	Code 1
	Encoded 2-bit block
00	0000
01	0001
10	0011
11	0111

scheme is that it has a higher mean activity than the original non-coded words.

In order to decrease the mean activity of the previous code, two codes for each 2-bit group can be used (referred after as *Code 2*). The two codes are used alternatively on each transmitted block to decrease the activity and are described in Table III. An example of the proposed coding scheme is given in figure 4. The first code has only rising transitions but the second one involves only falling transitions for all 4-bit patterns. The mean activity of the random encoded sequence is the same compared to the non-coded sequence.

TABLE III
CORRESPONDENCE BETWEEN THE ORIGINAL 2-BIT BLOCKS AND THE PROPOSED CROSSTALK AVOIDANCE CODING SCHEME CODE 2.

Original 2-bit block	Code 2	
	Encoded block	Alternative encoded block
00	0001	0000
01	0011	1000
10	0111	1100
11	1111	1110

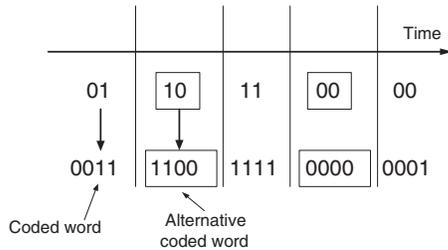


Fig. 4. Encoded data using the alternative coding scheme for one line of the bus (*Code 2*).

All these codes are easy to implement because they are totally static in the way that one (or two) 4-bit sequence corresponds always to the same 2-bit sequence. A great advantage of these coding schemes is that they are independent from the bus bit-width and there is no wiring overhead. They can be used for very wide busses without any issue since they are area-efficient. The main drawback of this approach is that it doubles the data needed to be transmitted on the bus.

IV. PERFORMANCES OF THE CODING SCHEMES

We have simulated the efficiency of our codes with SPICE using United Microelectronics Corporation (UMC) 0.13 μ m

CMOS technology and we synthesized the codecs for *Code 1* and *Code 2* with the high-speed 0.13 μ m UMC library to have the area requirements of the codecs. In our experiments, r is near 3, with $C_s=40$ fF and $C_g=14$ fF. These results were obtained using metal-2 technology parameters for a 1-mm wire.

A. Propagation delay

The wires are modeled using the π 3 distributed RC model. All transistors for the 1-mm bus are designed with common W of 8λ for the PMOS and 4λ for the NMOS. For the 10-mm bus, the transistors are 10x wider. The link was modeled using the technology rules for a metal-2 layer (width: 320nm, spacing: 280nm) with a power supply voltage of 1.2V. The drivers and capacitive loads on the wires are modeled using inverters implemented using the transistors described above. We measured the total propagation delay of a victim wire in the situation described in figure 1. In this test-bench, the input signals have the same rise and fall times of 0.1ns. The delay test bench is composed of three wires (one victim wire and two aggressors). We define the propagation delay as the delay between the time the input reaches 50% of its transition and the time the output reaches 50% of its transition. We measured the propagation delay of the victim wire for all delay groups in Table I and with two different wire lengths (1-mm and 1-cm). The worst-case results are given in Table IV. No code was used at this moment. The patterns described in this table are the inputs of the drivers modeled by the inverters.

TABLE IV
PROPAGATION DELAY IN NANoseconds OF THE VICTIM WIRE AS A FUNCTION OF THE DIFFERENT TRANSITIONS OF THE THREE WIRES AND OF THE WIRE LENGTH.

A1	V	A2	1-mm wire [ns]	10-mm wire [ns]
↑	↑	↑	0.17	0.32
↑	↑	-	0.24	0.52
↑	↑	↓	0.31	1.36
-	↑	-	0.47	1.37
-	↑	↓	0.80	2.44
↓	↑	↓	1.17	3.21
↓	↓	↓	0.18	0.34
↓	↓	-	0.29	0.55
↓	↓	↑	0.49	1.46
-	↓	-	0.60	1.42
-	↓	↑	1.05	2.51
↑	↓	↑	1.41	3.29

The worst-case propagation delay for a non-coded system is 1.41ns for a 1-mm wire, and corresponds to the pattern (↑, ↓, ↑). *Code 0* and *Code 1* permit to avoid slowest transitions and to have a worst-case propagation delay of only 0.47ns for a 1-mm wire (pattern (-, ↑, -)). In this case, we have only rising transitions at the inputs of the inverters in the interesting part of the data flow. The acceleration of the propagation delay is 3. The drawback is that our coding scheme must transmit two bits instead of one. This leads to a bandwidth improvement of 50%. The last coding scheme *Code 2* has a worst-case delay of 0.60ns because the alternative coding can lead to the pattern

($-, \downarrow, -$). In this case, the acceleration of the propagation delay is only 2.35. So, the bandwidth improvement is only 17.5%. In the case of a 10-mm wire, the improvements are respectively 20% for *Code 0* and *Code 1* and 16% for *Code 2*.

The delay improvement of our schemes is expected to grow in future technologies as the delay ratio r introduced in section I is expected to grow itself due to the increase of the aspect ratio of wires [1]. Another advantage of our approach is that it can be applied to future very large busses as it has no wiring overhead. So, the additional bandwidth improvement offered by future large busses is available to our approach.

B. Area requirements

The synthesis results of *Code 1* and *Code 2* are respectively given in Tables V and VI for a 32-bit bus. As the encoder needs two bits to perform the encoding operation and to send four bits and the decoder needs to do exactly the reverse operation, the non combinational area is much larger than the combinational one. But this non combinational area permits to deeply pipeline the codecs, enabling a low critical path between two consecutive flip-flops. We can see from this table that the combinational area requirements is larger for the *Code 2* compared to the *Code 1*. This is due to the two alternative operations needed to perform the encoding. The non combinational area is quite the same in the two cases because the number of flip-flops remains the same.

TABLE V
SYNTHESIS RESULTS OF THE CODE 1 CODEC IN THE 0.13 μ m UMC
LIBRARY FOR A 32-BIT BUS.

Device	Code 1	
	Encoder	Decoder
Combinational area (μm^2)	1635	1856
Non combinational area (μm^2)	7051	7714
Total area (μm^2)	8686	9570
Critical path (ns)	0.92	0.69

These results must be compared to the interconnect area gain provided by the independence of our schemes from the bus bit-width. For example, if we compare them to shielding, the number of needed wires is quite doubled (for a 32-bit bus, the wiring overhead is 31 bits). If we consider 32 parallel wires with the metal-2 rules, the interconnect area overhead due to shielding is about 186000 μm^2 for a 10-mm wire (including the wires and the spacing). The overhead is very much larger than the codecs area. Even for a 1-mm wire, the area overhead is quite the same.

V. NOISE ANALYSIS

Crosstalk is the origin of issues with signal integrity. The voltage peak induced by the transitions of adjacent aggressor wires can cause a failure in the evaluation of the bit transmitted by the victim wire. This effect is represented in figure 5 where $A1$ and $A2$ are the aggressors and Vic is the victim line. The two aggressors switch in the same direction, causing a voltage peak greater than $\frac{V_{dd}}{2}$ on the victim wire (UMC 0.13 μ m with

TABLE VI
SYNTHESIS RESULTS OF THE CODE 2 CODEC IN THE 0.13 μ m UMC
LIBRARY FOR A 32-BIT BUS.

Device	Code 2	
	Encoder	Decoder
Combinational area (μm^2)	2749	4417
Non combinational area (μm^2)	7086	7419
Total area (μm^2)	9835	11836
Critical path (ns)	0.77	0.79

minimum spacing for three 1-mm wires). This spike leads to a possible mistake in the received word. Even if this situation can occur in every scheme that is not especially designed to alleviate this issue, the proposed coding schemes rise the probability of appearance of such a scenario. In fact, the proposed coding schemes lead to a unification of the switching directions of the aggressors because they can switch in only one direction in the same coded 4-bit word. This increases the excitation level of crosstalk noise.

A possible solution consists in having an error-tolerant code for the link. A few lines can be added to the bus because our schemes are area-efficient. This can improve the noise tolerance of the link against noise [10]. But this solution increases the power consumption of the transmission link if low-swing signaling is not used. Moreover, low-swing signaling decreases the signal to noise ratio. A more simple solution is to increase the spacing between wires to lower the coupling capacitance between adjacent wires. We have simulated the same transitions that exists on figure 5 but with a doubled spacing. The result is shown in figure 6. We can observe that the voltage peak is clearly reduced.

Increasing the spacing between wires is also expected to decrease the propagation delay because of a decrease of C_c and hence a decrease of C_{eff} . A spacing that is 50% larger than the minimum one (420nm) and a doubled spacing (560nm) were simulated. The propagation delays are given in Table VII. We can notice that the acceleration is much lower than the one with a minimum spacing (the acceleration decrease from 3 to 2.44 for a 1-mm wire and a doubled spacing for example). This was foreseeable because these measurements depend on the comparison between the cross-coupling capacitance (C_c) and the wire capacitance (C_s). Since the spacing between wires is increased, the coupling capacitance that exists between adjacent wires becomes less important when compared to the wire capacitance. Since our schemes do not require additional wires, the area overhead induced by the increased spacing is reasonable. For example, the area requirement of shielding is higher. It will add the width of each shielding wire which is comparable to the minimum spacing.

Another way of dealing with the crosstalk-induced noise problem is to adapt the receivers. It is more interesting than increasing the spacing because the waste of area is limited and the acceleration provided by the coding schemes is conserved. We can notice that *Code 1* has only rising transitions that can cause an issue if some lines are stable at the binary zero. Falling transitions on aggressor wires are not an issue if

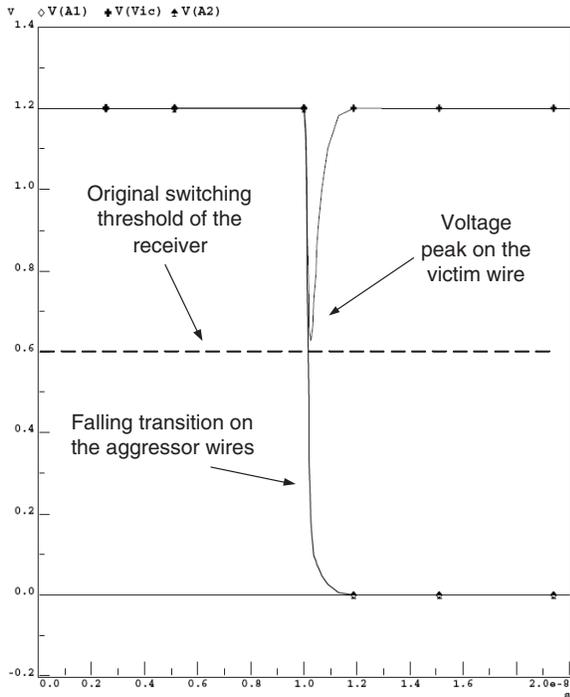


Fig. 5. Voltage peak on a victim wire due to simultaneous transition of two aggressor wires with minimum spacing.

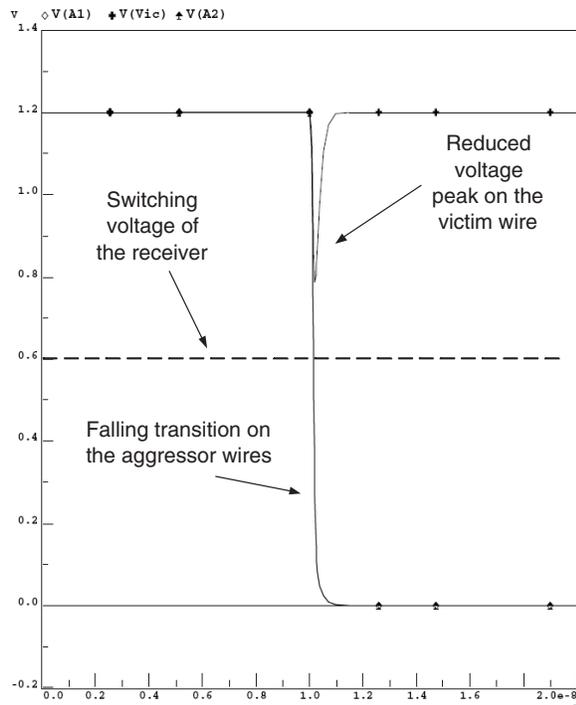


Fig. 6. Reduced voltage peak on the victim wire with two aggressors with a doubled spacing.

TABLE VII
PROPAGATION DELAY IN NANoseconds OF THE VICTIM WIRE AS A FUNCTION OF THE DIFFERENT TRANSITIONS OF THE THREE WIRES AND OF THE WIRE LENGTH WITH DIFFERENT SPACINGS BETWEEN WIRES.

			Spacing: 480nm		Spacing: 560nm	
A1	V	A2	1mm [ns]	10mm [ns]	1mm [ns]	10mm [ns]
↑	↑	↑	0.17	0.32	0.17	0.32
↑	↑	-	0.23	0.50	0.22	0.48
↑	↑	↓	0.29	1.02	0.28	0.85
-	↑	-	0.38	1.04	0.34	0.87
-	↑	↓	0.60	1.73	0.49	1.38
↓	↑	↓	0.85	2.28	0.69	1.81
↓	↓	↓	0.18	0.34	0.18	0.34
↓	↓	-	0.27	0.52	0.26	0.50
↓	↓	↑	0.42	1.09	0.41	0.90
-	↓	-	0.48	1.07	0.41	0.90
-	↓	↑	0.77	1.79	0.63	1.43
↑	↓	↑	1.02	2.34	0.83	1.86

the victim wire is stable to zero. The basic idea is to adapt the switching level of the receiver by setting the width and length of the transistors composing it (figure 7). We take into account that the transitions are always in one direction, and transform a disadvantage (a unification of the switching directions) in a fault-tolerant system by applying a unique direction to the noise. *Code 2* has the two possible directions (one per code cycle). We can replace the receiving inverter by a Schmitt trigger for example. It has the property of shifting the switching voltage level but in the two possible directions. Finally we have a faster system than can be noise-tolerant because we influence the noise in only one direction. That is possible because crosstalk noise is predominant in deep-submicron technologies. This implementation is also area-efficient as it does not require additional codecs.

VI. CONCLUSION

A new approach for avoiding crosstalk in busses is presented in this paper. This approach can increase the bandwidth of a link by 50% and does not require wiring overhead. It also has the great advantage to be independent from the bus bit-width. This approach consists in applying a temporal coding instead of a word-by-word coding. We propose another coding scheme (*Code 2*) that can reduce the activity of the codes. The proposed schemes are also simple to implement because it is a systematic approach. Another great advantage is that we control the direction of the crosstalk-induced noise to implement a noise-tolerant system. The next step of our researches is to have adaptive coding schemes to reduce the redundancy (four bits transmitted instead of two) and to improve the bandwidth enhancement.

REFERENCES

- [1] ITRS, "http://public.itrs.net/files/2003itrs/home2003.htm," International Technology Roadmap for Semiconductors, Tech. Rep., 2003.
- [2] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 2002.
- [3] C. Duan and A. Tirumala, "Analysis and Avoidance of Cross-Talk in On-Chip Buses," in *HOTI '01: Proceedings of the Ninth Symposium on High Performance Interconnects (HOTI '01)*. IEEE Computer Society, 2001, p. 133.

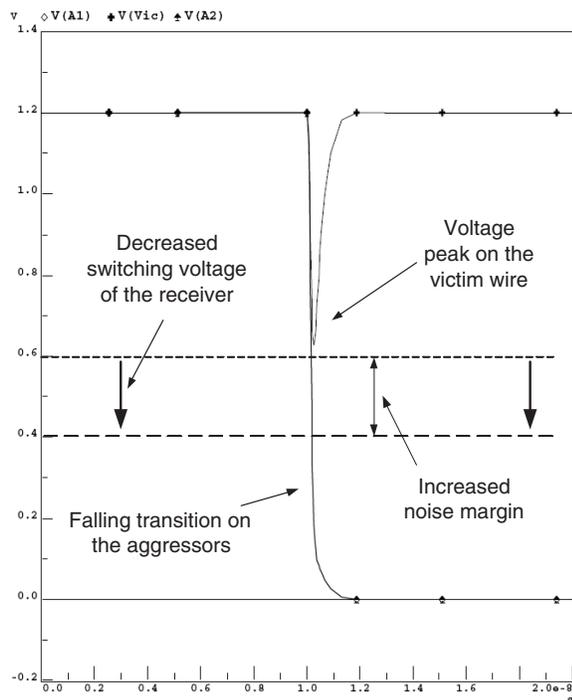


Fig. 7. Adaptation of the switching level of the receiver to increase the noise margin with minimum spacing.

- [4] A. Devgan, "Efficient Coupled Noise Estimation for On-Chip Interconnects," *IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, pp. 147–153, November 1997.
- [5] S. P. Khatri, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, *Crosstalk Noise Immune VLSI Design using Regular Layout Fabrics*. Kluwer Academic Publishers, 2001.
- [6] K. Hirose and H. Yasuura, "A Bus Delay Reduction Technique Considering Crosstalk," in *DATE '00: Proceedings of the conference on Design, automation and test in Europe*. IEEE Computer Society, 2000, p. 441.
- [7] B. Victor and K. Keutzer, "Bus Encoding to Prevent Crosstalk Delay," in *ICCAD '01: Proceedings of the International Conference on Computer-Aided Design (ICCAD '01)*. IEEE Computer Society, 2001, p. 57.
- [8] C. Duan and S. P. Khatri, "Exploiting Crosstalk to Speed up On-Chip Buses," in *DATE '04: Proceedings of the conference on Design, automation and test in Europe*. IEEE Computer Society, 2004, p. 20778.
- [9] L. Li, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "A Crosstalk Aware Interconnect with Variable Cycle Transmission," in *DATE '04: Proceedings of the conference on Design, automation and test in Europe*. IEEE Computer Society, 2004, p. 10102.
- [10] D. Bertozzi, L. Benini, and G. DeMicheli, "Low-Power Error-Resilient Encoding for On-Chip Data Buses," in *DATE'02, International Conference on Design, Automation and Test in Europe (DATE'02)*, 2002, pp. 102–109.