CODING TECHNIQUE FOR INFORMATION SENT THROUGH A PCB PARALLEL DATA BUS FOR AVOIDING CROSSTALK

Mihai DARABAN Dan PITICA
Applied Electronics Department of UTCN
26-28 Baritiu Street, Cluj-Napoca, mihai.daraban@ael.utcluj.ro

Abstract: This paper presents a crosstalk avoidance code (CAC), developed by analyzing the integrity of the signals sent through a printed circuit board (PCB) bus. There was also determined the influence of trace impedance dynamic variation regarding the switching pattern. In CAC are integrated rules related to switching patterns that help creating a forbidden transition channel (FTC). A FTC has the advantage that does not only reduce the crosstalk effects but also resolves the inter-symbol interference (ISI) propagation problems. The CAC developed has as purpose an error free transmission at high speeds on a PCB build with low cost technologies, without using high speeds manufacturing processes.

Keywords: *switching pattern, bus coding, crosstalk, characteristic impedance.*

I. INTRODUCTION

One problem that affects parallel data buses is crosstalk, which influences the propagation time delay and characteristic impedance for each trace that make up the bus. Crosstalk effects are not constant and depend on the switching patterns that appear on the bus when sending data, [1 - 4]. There can be used traditional physical methods to resolve the crosstalk problems, like introducing guarding traces or by increasing the distance between the traces, [1, 3, 5]. Even if these methods are the most efficient in resolving crosstalk problems, they increase the surface of the bus thus increasing the cost of the PCB. Other physical methods are described in [6 and 7], but the disadvantages of these methods is that the design of the bus is based on statistical information of the sent data. When the statistical proprieties regarding how often the traces switch, the layout of the bus has to be changed.

Another approach to resolve the perturbation effects of crosstalk is to encode the information carried by the parallel bus. Some of the codes proposed in other papers, are trying to resolve the number of transitions needed to send a new symbol through the parallel bus, [2 and 8]. By reducing the number of transitions, there is reduced the number of parasitic capacitances that are charged or discharge when new data is sent. However, these algorithms do not take into account the position and direction of the switching. Because of these there are switching pattern whit reduced number of transitions that still can produce errors in the transmission of data,[9, 10].

There are many codes with good results developed for on chip buses. Appling those onto PCB transmission do not yield such good result. This is because the effects of inductance coupling it has a bigger impact then the capacitive coupling (on chip buses), thus resulting different switching patterns that introduces errors in the sent data.

In the paper the analysis of the switching patterns were made on a bus model obtained from a layout design. There were taken into account the routing imperfections that appear when realizing a bus on a real PCB (the traces do not have the same length, the corner effects). For the logic gates, their analog behavioral was described by using IBIS files obtained from the producer site. Analyses of the bit patterns were made by sending data in a specific order so that inter-symbol interference (ISI) could also be taken into account when implementing the crosstalk avoidance code (CAC), [4, 5, 9, 12]. In the paper is also presented a study regarding the impedance dynamic variation when data is sent through a parallel bus. The purpose of the study is to show for what kind of stackup can be used the proposed CAC.

For the same impedance dynamic variation of the traces there were obtained different sets of switching patterns when changing the logic families that driven the bus. Switching patterns differ because there are other electrical proprieties that characterize 74AC and 74ALVC logic families when are sending or receiving signals. Because of this reason, there does not exist is a universal coding algorithm.

As will be shown in section III, there can be realized some analysis on the data sent so that the algorithm can be adapted to the logic gates electrical proprieties. For example, smaller rise and fall times increase the coupling between adjacent traces, [1, 5]. Another influence on the switching patters has the stackup as will be shown in section III. The paper is not only proposing a coding method, but also a way to analyze the sent data through the parallel bus so that the coding method can be adapted to the bus topology.

The purpose of the crosstalk avoidance code presented in the paper is to make possible sending of data without errors on a parallel bus at speeds of 0.83Gbps (the bit interval is 6 ns), without using special stackup dimensions and high speed dielectric material that would resolve the crosstalk problems. The coding algorithms address the problem of sending data on a low cost PCB, implemented using low cost technologies and materials.

Manuscript received January 31, 2012; revised May 8, 2012

II. RELATED WORK

Crosstalk appears because of the parasitic coupling between adjacent traces that form the parallel bus. The electric and magnetic field lines are inducing capacitive and inductive currents depending on the switching patterns. A solution is to guard the traces of the bus or increase the distance between them. Doing so there will be less field lines that reach from one trace to another, affecting signal integrity.

The increase in dimension caused by the classic methods can be reduced by using the swizzling method or by placing the traces at different distances depending on statistical information, [6 and 7]. Both methods are based on algorithms that need to know which traces switch more often. For a data bus, where information is random, such approach does not give good results. The two methods are more suitable for memory address buses.

By encoding the information, there is possible to determine online the statistical information regarding the data sent and adapt the coding method to the new specification without redesigning the bus topology. Coding the information, also resolves the ISI problem, by imposing rules regarding how often traces can change their state.

There are two approaches in coding the information sent through a bus, by using memory-base or memory-less code, [3 and 10]. Memory-less algorithms are more simple to implement, because they only need the current symbol sent through the bus to code/decode the sent/received data. Usually these kinds of codes are implemented based on logic operations or a look up table. The coding of the data is signaled through additional traces, thus increasing the bus area. In some implementations there are added so many traces that becomes more feasible to guard each trace then apply the coding algorithm.

Bus-invert is a memory-base coding method that reduces the number of switching when transmitting data through a parallel data bus, [8]. It guarantees that only half the traces of the bus will switch at a moment. Because the algorithm does not take into consideration the position and direction of the transitions in the bus, there can appear adjacent traces that switch in opposite directions. Other situations are possible because there are no rules regarding the switching patterns, [9 and 10]. Improvements were made to the algorithm, but increase to much the bus area, [2].

Algorithms that are implemented by the memory-based method generate the symbol that is sent onto the parallel bus based on the previously and the current data that has to be sent, [10]. The main advantage of this method is that does not increase the space need by the bus. The signaling and the coding process are done by introducing other bits. Memory-less codes were using spatial redundancy, instead memory-based codes are using temporal redundancy. A disadvantage of the method is that as the algorithm requires more and more rules on coding the data patterns, there are introduced more bits of signaling, reducing communication throughput.

The purpose of using any of the two methods presented above is to create a forbidden transition channel. This is done by imposing rules regarding the switching patterns that may appear when transmitting data, [3 and 10].

The coding method proposed in this paper impose conditions so that all the signals transmitted are having almost the same propagation delay, and that there are no passing over the guaranteed thresholds for the signals that don't switch. In addition, the coding algorithm is taking into account the ISI effect.

III. PARALLEL DATA BUS MODEL AND ANALYSIS METHODS

A. *Parallel data bus model*

The coding algorithms were developed by observing the effects of switching patterns over the signals carried by a five-trace parallel data bus model. The bus model was obtained by transposing a layout design of a parallel bus into SPICE design. By analyzing the bus obtained from a layout design, there are taken into account all the imperfections that appear in a real design when routing a bus and placing the circuits on the PCB. For example, as can be observed in fig. 1, there has to be a distance between the components so that the machines can place them automatically. In the SPICE model, there are used transmission lines to represent the effects of corners. The model simulates also the fact that in some regions off the bus traces are presenting a weaker coupling because of the routing process.

We choose a five-trace bus, because it offers a lot of switching patters and is large enough to analyze the effects of inductive coupling on the signals sent. A smaller bus size then the one analyzed would not have revealed the effects of inductive coupling, [4 and 6]. By using a larger bus, the inductive coupling from one trace would affect only some of the traces, as the rest will be too far away to be influenced by the magnetic field lines. In the case of a larger bus, there would be too many switching patterns, increasing too much the complexity of the algorithm.

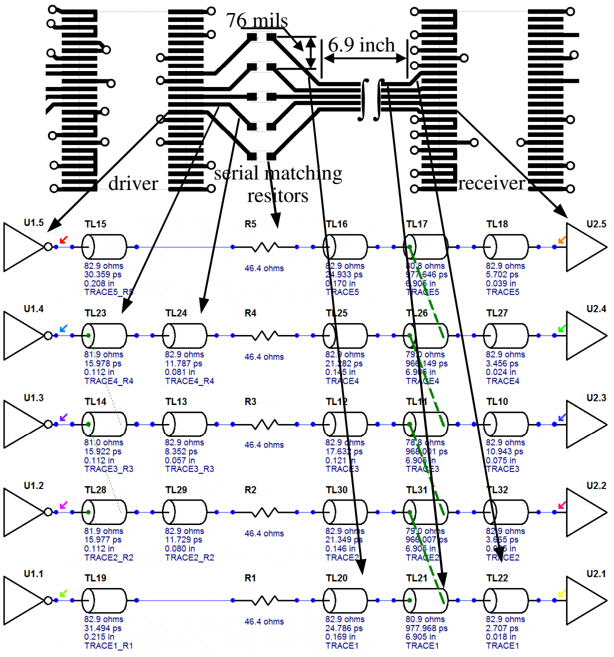


Figure 1. Layout design of the parallel data bus

In the Spice model traces were considered to be coupled based on the electrical proprieties of the driver (rise/fall time, output impedance, input capacitance and switching range). For example, the forward component of the crosstalk is strongly affected by the driver-IC switching time, [5].

The coupling is determined based on the amplitude of the coupling voltage. This approach reflects much better which traces are coupled instead of deciding on the distance between the traces. The data needed to determine if traces are coupled is stored in the IBIS files that describe the logic families gates used as drivers and receiver in the bus topology.

IBIS description offers result with better precision regarding the propagation of signals through the gates, compared to structural models. This happens, because producers are more willing in giving accurate data for the implementation, as the files do not contain information regarding the manufacturing process. The IBIS file also contains information related to the parasitic elements of each pin of the gates used in the bus. Consequently signals sent on the bus have different delays depending on the pins from where are transmitted.

To determine if there is possible to generate a universal CAC, we analyzed the signals from two different logic families (74ALVC16240 and 74AC04). Both logic families are implemented by using CMOS technology, but 74ALVC is a low voltage bus driver, that has asymmetrical threshold levels (Valim = 3.3V, ViH = 2V and ViL = 0.8V) compared to 74AC (Valim = 5V, ViH = 3.5V and ViL = 1.5V). As the analysis in section V will show, because of the reduced immunity in case of gates 74ALVC, the following switching pattern ↑L↑ can bring the victim line voltage over the threshold level. For 74AC there will be needed more traces that switch in the same direction to affect the signal on the victim line. Because of different electrical proprieties between the two logic families, there were obtained different switching patterns that produce signal integrity problems.

The microstrip traces from the layout designed are modeled by using transmission lines whose electrical proprieties are computed based on a stackup model that takes into account the distance between the layers of the PCB and the proprieties of the dielectric material. The impedances of traces were computed based on the capacitive and inductive matrix obtained from SPICE field solver.

Layer Name	Type	Thickness (mils)	ε_r	Test width (mils)	Z_0 [Ω]
TOP	Metal (cooper)	1.4 (1 oz)		10	82.9
Substrate	Dielectric (FR4)	16	4.45		
GND	Metal	1.4			
Substrate	Dielectric	28	4.5		
VCC	Metal	1.4			
Substrate	Dielectric	16	4.45		
Bottom	Metal	1.4		10	82.9

Table 1. Description of the stackup structure

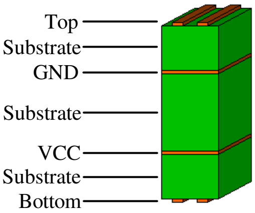


Figure 2. Stackup representation

B. Analysis methods

The effects of bit patters on the integrity of the signals carried by the parallel bus were determined by sending the information in a specific order (frame). The structure of the frame used in analyzing the 32 cases is presented in fig. 3.

For each case represented by a combination of five bits, there were analyzed 32 frames, corresponding to switching from the case analyzed to each possible symbol. Between frames there was introduced a time for the signals to settle, so that the frames would not affect each other.

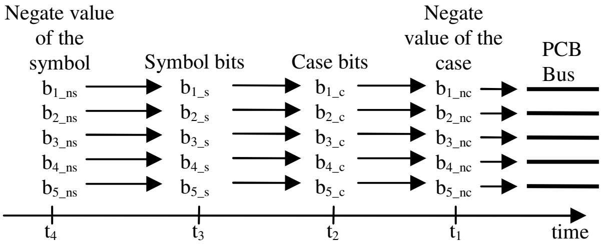


Figure 3. Structure of the frame used in analyzing a case.

ISI influence on the signals was simulated by sending before the bits corresponding to the case analyzed their inverted state. Also this is the state in which the drivers are switching after transmitting the inverted value of the symbol, preparing for the transmission of the next frame. In this state signals are left to settle for 3 bit duration before sending the next frame.

The signals from the 32 frames were superimposed over each other to create eye diagrams. In the first eye can be observed the switching pattern created by the inverted bits when commutates in the case bits. There are superimposed also the signals from the five traces of the bus. The eye diagram in which measurements were done is the second one that contains all signals from the bus when switching from the case analyzed in all possible symbols.

The analysis and creation of eye diagram was done in MatLab, by importing the wave forms obtained in the SPICE simulator. By constructing the eye diagram in this manner we could tell which switching patterns were responsible for the eye diagram collapsing or closing. For eye diagrams obtained from random bits is very difficult, if not almost impossible, to determine the bit patterns that are responsible for the signal integrity problems, [5 and 12].

For each analyzed case, there were done measurements to determine the flight time of signals corresponding to traces that switched. When signals didn't change their logic value when a new symbol was transmitted, there were done measurements to determine if the voltage swing exceeded the thresholds, causing errors in the sent data. A switching pattern was declared to affect the quality of the sent data, when the difference in propagation delay was larger than 150ps compared to the flight time of the reference signals. Another criterion was to check that the voltage swing did not pass over the threshold levels corresponding to the logic state of sent signal. For each type of switching (low to high, high to low), a reference signal was generated at the receiver, [9].

The propagation time delay was measured in such a way that was taken into account the effect of the trace impedance and the internal propagation delay from the chip, [11]. Technique used in realizing the measuring also allows in comparing the flight times for different bus topology and stackup proprieties, but when is used the same logic family.

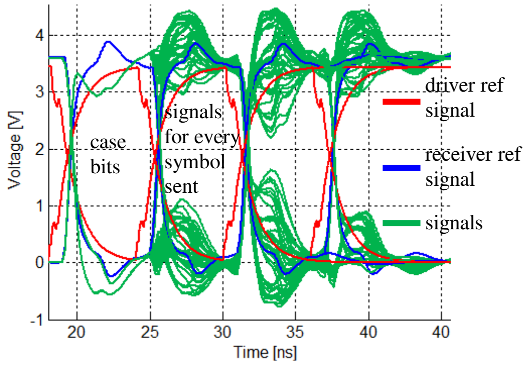


Figure 4. Eye diagram example for case 01110 (74ALVC).

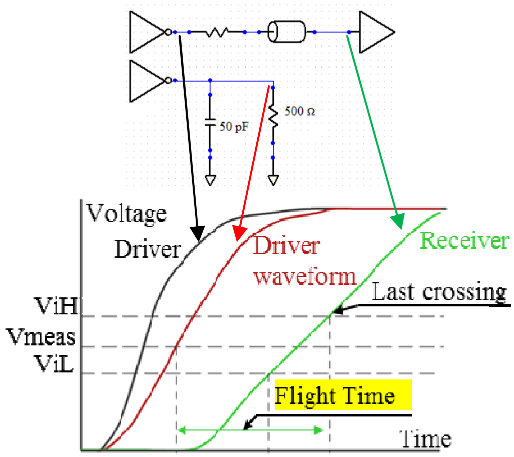


Figure 5. Flight time measurement example when switching from L→H.

IV. IMPEDANCE DYNAMIC VARIATION

Depending on the switching pattern, crosstalk modifies the propagation time delay and impedance of each trace from the parallel bus. When signals are changing their value, there are generated electrical and magnetic field lines that influence the propagation through the parallel data bus. Electromagnetic field lines influence how the traces of the bus are coupling to each other.

Because of the coupling effect that appears between the traces the impedance of each trace from the bus is different and changes with each transmitted symbol. In some cases the trace impedance during the switching gets so small/high that the matching technique is no longer able to reduce the reflections that manage to affect the next sent bits.

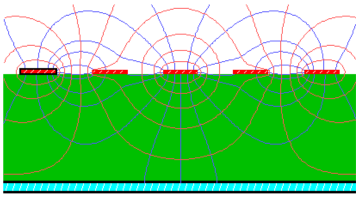


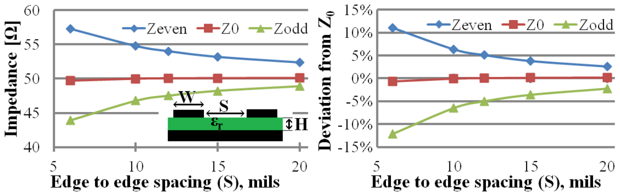
Figure 6. Electric and magnetic field lines corresponding to switching pattern ↓↑↑↑↓.

To determine how much the switching pattern can influence the impedance of the traces, we analyzed a simplified model made of two traces. For this model, there are only odd (↑↓) and even (↑↑) propagation modes, [1 and 5]. The impedance of the traces during the propagation in one of the modes mentioned above, can be computed by specifying the right relations between the voltages and currents from the equations bellow $\left(V_1 = -V_2; I_1 = -I_2 \text{ - odd mode; } V_1 = V_2; I_1 = I_2 \text{ - even mode}\right)$:

$$\begin{aligned}
&V_1 = L_{11}\frac{dI_1}{dt} + L_{12}\frac{dI_2}{dt} \quad V_2 = L_{22}\frac{dI_2}{dt} + L_{21}\frac{dI_1}{dt}\\
&I_1 = C_{1g}\frac{dV_1}{dt} + C_{12}\frac{d\left(V_1 - V_2\right)}{dt} = C_{11}\frac{dV_1}{dt} - C_{12}\frac{dV_2}{dt}\\
&I_2 = C_{2g}\frac{dV_2}{dt} + C_{21}\frac{d\left(V_2 - V_1\right)}{dt} = C_{22}\frac{dV_2}{dt} - C_{21}\frac{dV_1}{dt}\\
&Z_{odd} = \sqrt{\frac{L_{11} - L_{12}}{C_{11} + C_{12}}} \qquad Z_{even} = \sqrt{\frac{L_{11} + L_{12}}{C_{11} - C_{12}}}
\end{aligned} \tag{1}$$

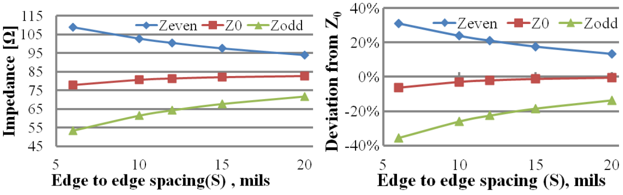
, where C_{1g}, C_{2g} - capacitance of line 1 to ground; C_{12}, C_{21} - mutual capacitance from line 1 to line 2; $C_{11} = C_{1g} + C_{12}$; L_{11}, L_{22} - self-inductance for line 1; L_{12}, L_{21} - mutual inductance between lines 1 and 2; the values of the parasitic elements can be obtained from the inductance and capacitance matrices of a transmission line system.

In figure 7, 8 and 9 are presented the traces impedance variation for different stackup. As the traces are getting further apart from the reference plan, there are fewer field lines that close to it and so increasing the coupling between adjacent traces. The impedance variation may show a deviation of up to 50% compared to the characteristic impedance (Z_0), figure 9. Creating a PCB with traces very close to the reference plane, like in figure 7, would be very expensive and represents a physical method of resolving crosstalk coupling. The analysis concentrates on resolving crosstalk problems for stackup were the distance between the traces and reference plane is more than 16 mils.



a. *b.*

Figure 7. Impedance variation for two coupled traces (H = 4 mils, W = 6.7 mils, ε_r = 4.45, Z_0 = 50 Ω).

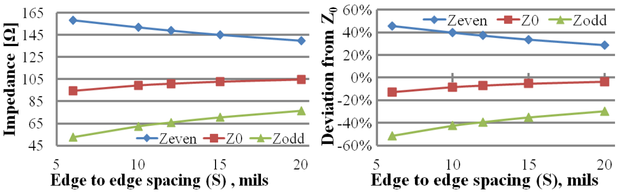


a. *b.*

Figure 8. Impedance variation for two coupled traces (H = 16 mils, W = 10 mils, ε_r = 4.45, Z_0 = 83 Ω).

The situations presented in figures 8 and 9 are less expensive and more likely to be used in mass production.

We adapted the equations (1), to our model of five traces to compute the impedance of each trace for the sent switching patterns as described in section III of the paper.



a. *b.*

Figure 9. Impedance variation for two coupled traces (H = 40 mils, W = 12 mils, ε_r = 4.6, Z_0 = 108.5 Ω)

V. CODING ALGORITHM BASED ON STATISTICAL INFORMATION OF THE SWITCHING PATTERN

During the research, the switching patterns were analyzed on a simplified bus model that did not take into account the imperfections of routing as presented in figure 1. The simplified model was used in analyzing the effect of impedance variation, when *H= 40 mils, S= 12 mils, W=12mils, ε_r = 4.6, Z_0 = 108.5Ω*. The bus was driven by using both types of logic gates mentioned in section II.

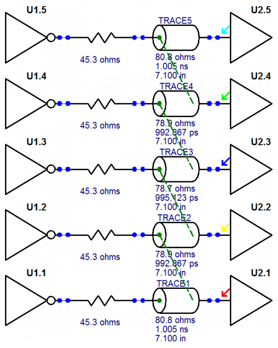


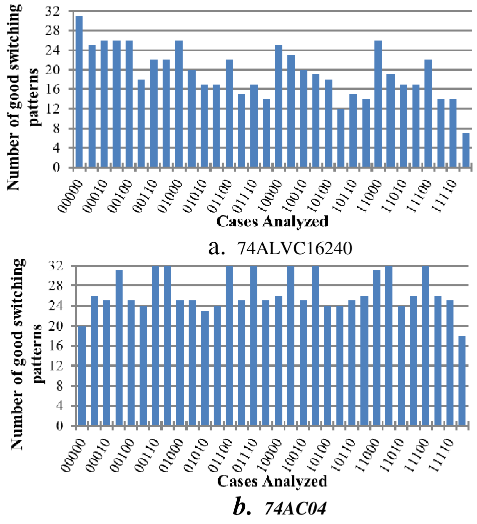
Figure 10. Simplified bus model.

Analyzing the result presented in figure 11.a and figure 11.b, there can be observed that 74ALVC has fewer cases that have a large number of symbols with which can form good switching patterns. The difference in results is because of the asymmetrical levels of thresholds for 74ALVC. There were also obtained statistical information regarding the number of cases that a symbol can follow in order to generate a good switching patterns as mentioned in section III (figure 12).

Analyzing the results from figure 11b., can be observe that for each case there is a number of switching patterns greater then 16, representing half the number of total possible switching patterns that can be created with five bits.

The proposed method of coding is to use a lookup table that contains symbols to replace the ones that would create signal integrity problems if they appear after the data sent previously. To signal the coding there will be added a new line to the bus, which if it changes its value it means that the symbol received is from the lookup table and the decoder has to identify the correct symbol that was replaced. If the signaling trace does not change its value, it means that the data received was not coded and no need to search in the lookup table. The coding algorithm exploits both temporal and spatial redundancy.

For each case there will be symbols that exclusively code the symbols that can produce bad switching patterns. When a good symbol was selected as a coding alternative for a bad symbol in a case, it can no longer be used to code other symbol. This rule guarantees that the information can be retrieve correctly, and there will be no ambiguity when decoding the information. For different cases the same symbol can be used to code different bad symbols, because the search in the lookup table is based on what was previously transmitted.



a. 74ALVC16240

b. ***74AC04***

Figure 11. Statistical information concerning what number of switching patterns can be created from a case.

From the statistical data presented in figure 11.a the coding algorithm can't be applied to resolve the crosstalk problems that appear when data is transmitted using 74ALVC16240. The stackup structure for the simplified bus model is worse when it comes to impedance variation compared to the stackup of the model obtained from the layout design (table 1). In figure 12 are presented the statistical information regarding the switching patterns for 74ALVC16240 when is used the bus model presented in figure 1.

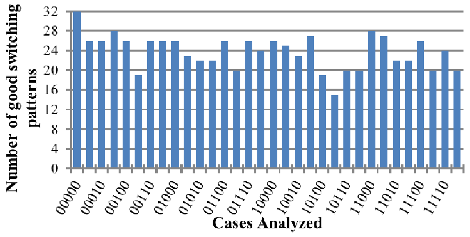


Figure 12. Statistical information regarding the number of switching patterns for each case (74ALVC16240).

Comparing the results of figure 11a and figure 12 can be observed some improvements, such as for case 11111. The case 11111 has the higher increase in good switching patters (from 7 to 20), compared to the other cases which present smaller increase. Even if the numbers are better than the previous ones, there are still cases where the number of switching patterns is below 16.

The coding technique can be applied by eliminating the possibility of switching from any case to the symbol 10101. By creating a FTC, there is eliminated the problem of coding the symbols that appear after was sent through the bus case 10101. There can be eliminated more cases, as long as the cases that remain have enough switching patterns to compensate for the removed ones.

Even if the second time there was used a more accurate model of the bus (figure 1), the improvements were brought by the stackup structure. For the case obtained from the layout design, there was used a different stackup, that managed to decrease the values of the parasitic elements (coupling capacitance and mutual inductance). As can be observed from table 2 the biggest reduction was for mutual inductance. By reducing the mutual inductance, is also reduced the effect of crosstalk produced by traces that switch in the same direction. The reduction in mutual inductance, justifies the improvements for case 11111 and also why other cases only had fewer improvements.

The difference in the values of mutual inductance between the two stackup designs it shows that on PCB busses, inductive coupling has greater influence compared to capacitance coupling.

Parasitic elements value for the simplified model (X)	Parasitic elements value for the model obtained from layout design (Y)	$\frac{X-Y}{X}\%$
C_{11} = 11.2 pF	C_{11} = 13.8 pF	-23.2%
C_{12} = 4.2 pF	C_{12} = 4 pF	4.7%
L_{11} = 110.8 nH	L_{11} = 84 nH	24.18%
L_{12} = 50.53 nH	L_{12} = 33 nH	34.7%

Table 2. Comparing the parasitic elements from the two stackup models.

VI Conclusion

Motivated by the purpose to guarantee an error free transmission on a parallel data bus, the paper propose a coding method that generates a forbidden transition channel that mitigates the crosstalk effect between adjacent traces in long PCB buses. Also by using the associated analysis of the bus presented in the paper, there that can be determined the switching patterns and so creating the lookup table needed to code/decode the transmitted information. There is needed a method to analyze the bus, because it does not exist a universal coding method. The coding algorithm has to be adapted to the electrical proprieties of the logic gates used, and to that of the used stackup.

The impedance dynamic variation revealed that the coding algorithm is more suitable for resolving crosstalk problems for low cost PCB boards that have a substrate thickness, between the traces and reference plan, greater than 16 mils.

The coding algorithm has the advantages that the analysis of switching patterns was done on a larger cluster of traces, and so determining the influence of inductive coupling on the transmission of information. Because of the proposed method of analyzing the switching patterns, there was taken into account and the ISI effect when transmitting information.

Even if the coding method was proposed for a bus that has five traces, there is possible to code larger buses by dividing them in clusters of five nets. For each cluster will be needed an additional signaling trace. Each cluster and its signaling trace will be separated by using guard traces from the other pairs of traces and their signaling wire.

Several research directions are worth further investigation.

(i) The influence of the signaling wire on the propagation of information. When there is signaled a coding symbol, is possible that the symbol and signaling transitions to interfere and affect the sent data.

(ii) To develop a coding algorithm that tries to mitigate the crosstalk problems even for asymmetrical logic gates. The new algorithm should be exploiting the advantages to switch from H → L, as is the case for 74ALVC. By imposing coding rules there should be avoided as much as possible switching from L → H. This rule can be implemented based on the bit-stuffing algorithm, [10]. The coding algorithm should be able to guarantee an error free transmission even for stackup with large impedance dynamic variation as presented in section IV.

ACKNOWLEDGMENT: This paper was supported by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

REFERENCES

[1] S.H. Hall, G.W. Hall, J.A. McCall, *High-Speed Digital System Design – A Handbook of Interconnect Theory and Design Practices*, John Wiley & Sons, Inc., New York, 2000.

[2] Fan CP, Fang CH, "Efficient RC low-power bus encoding methods for crosstalk reduction", *Integration – The VLSI Journal*, vol. 44, pp.75-86, 2011.

[3] Chunjie Duan, V.H.C. Calle, S.P. Khatri, "Efficient On-Chip Crosstalk Avoidance CODEC Design", *IEEE Transactions on Very Large Scale Integration (VLSI)*, vol. 17, pp551-560, 2009.

[4] SW Tu, YW Chang, JY Jou, "RLC Coupling-Aware Simulation and On-Chip Bus Encoding for Delay Reduction", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp.2258-2264, 2006.

[5] E. Bogatin, *Signal and Power Integrity*- Simplified Second Edition, Pretince Hall International, 2009.

[6] B. Soudan, "Controlling Inductive Coupling in Wide Global Signal Busses Through Swizzling" ICM 2002, *The 14th Internation Conference on Microelectronics*, pp.181-184, 2002

[7] E. Macii, M. Poncino, S. Salerno, "Combining Wire Swapping and Spacing for Low-Power Deep-Submicron Buses", *Proceedings of the 13th ACM Great Lakes Symposium on VLSI*, pp.198-202, 2003.

[8] M. R. Stan, W. P. Burleson, "Bus-invert coding for low power I/O", *IEEE Transaction on Very Large Scale Integration (VLSI)*, Vol. 3, pp.49-58, 1995.

[9] M. Daraban, D. Pitica, "Statistical Information of Crosstalk on Parallel Bus", *IEEE 17th International Symposium for Design and Technology in Electronic Packaging*, pp.211 – 216, 2011.

[10] CS Chang, J Cheng, TK Huang, "A Bit-Stuffing Algorithm for Crosstalk Avoidance in High Speed Switching", INFOCOM, *IEEE Proceedings*, pp.1-9, 2010.

[11] P. Carrier, "*Managing PCB Trace Lengths for Optimal System Timing",* Mentor Graphics Web Seminar, 2011. [Online]. Available:
https://mentor1.adobeconnect.com/_a781163502/p31872616/
[Accessed: June 16, 2011].

[12] J. Hancock, "*Jitter – Understanding it, measuring it, eliminating it*" Part 1-3, High Frequency Electronics, 2004