Middle-East Journal of Scientific Research 20 (12): 2405-2411, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.20.12.348

System Design for Encoding and Decoding to Minimize the Crosstalk in VLSI Circuits

M. Gopala Krishnan and P. Rachelin Sujae

Bharath University, Chennai, India

Abstract: In the growing up world many technologies are growing faster and faster as they are becoming smaller and smaller, one such is the very large scale integrated design-VLSI. Many challenges are faced in VLSI, one of them is the crosstalk occurrence. Global buses in deep-submicron (DSM) system-on-chip designs consume significant amounts of power, have large propagation delays and are susceptible to errors due to DSM noise-crosstalk. Due to this, crosstalk occurrence on long on-chip buses is increasingly becoming a limiting factor in high-speed designs. Crosstalk between adjacent wires on the bus may create a significant portion of the transmission delay. Placing a shield wire between each signal wire alleviates the crosstalk problem but doubles the area used by the bus, which is an unacceptable consequence. Instead, we propose to employ data encoding and decoding for a special codes called boundary shift codes to minimize crosstalk within a bus.

Key words: Design-VLSI · Global buses in deep-submicron (DSM) system-on-chip designs · Increasingly becoming a limiting factor in high-speed designs

INTRODUCTION

VLSI: VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas. Thanks to VLSI, circuits that would have taken boardfuls of space can now be put into a small space few millimeters across [1, 7]. This has opened up a big opportunity to do things that were not possible before. VLSI circuits are everywhere eg: computer, car, brand new state-of-the-art digital camera, the cell-phones and many other. All this involves a lot of expertise on many fronts within the same field. Alongside, obeying Moore's law, the capability of an IC has increased exponentially over the years, in terms of computation power, utilization of available area, yield. Examples are embedded systems, where intelligent devices are put inside everyday objects and ubiquitous computing where small computing devices proliferate to such an extent that even the shoes you wear may actually do something useful like monitoring our heartbeats. These two fields are kind related and getting into their description can easily lead to the world of VLSI [2].

Crosstalk in VLSI: The dramatic increase in signal switching speed and density of integrated circuits leads to challenging design and test problems. Interconnection lines that were once considered to be electrically isolated can now interfere with each other and have an important impact on system performance and correctness. One such interaction caused by parasitic coupling between wires is known as crosstalk. crosstalk can be defined in many ways [2, 5]. Crosstalk can produce logic errors in the circuit. Here we have proposed the design of encoder and decoder for boundary shift codes to minimize the crosstalk and also to correct the error, it can be useful in a variety of applications including nanotechnology, low-swing signaling and radiation-hardened circuits. The detailed description of crosstalk is listed in sec4 [3].

Boundary Shift Codes: They are used for error correction and for minimizing crosstalk inVLSI. Data words are mapped to code words which are self shielding and consequently which have no invalid transitions, this allows cross talk to be minimized. The proposed design of encoder and decoder for this codes to minimize crosstalk is discussed in sec5.

Corresponding Author: M. Gopala Krishnan, Bharath university, Chennai, India.

VLSI FEATURES: VLSI technology drives the whole innovative devices and systems which effects the way we live. The integration improves the design [3]. It includes features like lower parasitics=higher speed, lower power,physically smaller, more reliable than discrete system, can design more complex system, high speed of circuits on chip due to smaller size etc., eg., the microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors.

Challenges Faced in VLSI: The increase in package density as well as the clock frequency of the VLSI circuits has lead to capacitive coupling noise-i.e., crosstalk, is one of the most challenging problems in the design and verification of modern VLSI circuits. Furthermore, the interconnect lines get thicker and narrower (and longer in case of global interconnects), which all result in the increase of crosstalk noise amplitude and duration and the circuit faults caused by such noise sources. Therefore as the VLSI technology scales down the role of interconnect parasitic effects in the signal integrity becomes increasingly more pronounced.

Occurrence of Crosstalk in VLSI: In integrated circuit design, crosstalk normally refers to a signal affecting another nearby signal. The circuit components have shrunk in size while the total number of components within a fixed chip area has increased. Here are two terms to be recognized which leads to crosstalk, first is the wire to substrate capacitance and second is the cross coupling capacitance. The enormous number of components along with small device features increases the proximity between the diffrent components of the design. Each individual component gives rise to a capacitance relative to the substrate. To know about the crosstalk coupling consider the following figures 1.1 and 1.2.

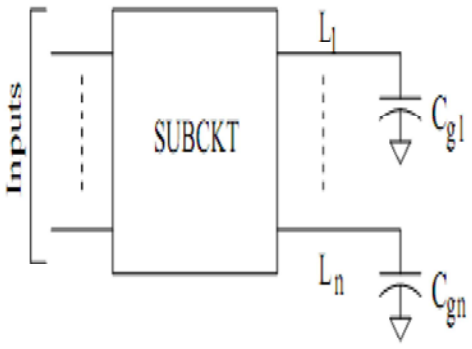


Fig 1.1: Subcircuit with substrate capacitance

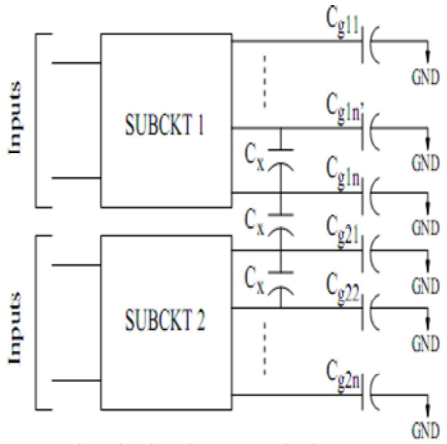


Fig 1.2: Two circuits in close proximity

For example, in Figure 1.1, 1.2 each output line of the CMOS sub circuit has a capacitance Cg, relative to the ground. Thus, n-output lines have wire-to-substrate capacitances Cg1through Cgn and Cg11 through Cg2n. On the other hand, a capacitance Cx may exist between Closely-located wires of a common sub circuit, closely-located wires of different subcircuits which are in the vicinity of one another [4]. Such a capacitance is known as the cross-coupling capacitance. Both types of cross coupling capacitances are shown in Figure 1.a, 1.2. If the wire-to-substrate capacitance Cg is much higher than the cross-coupling capacitance Cx, then each line has a high drive strength and different lines do not influence each other in any way.. The ratio of the cross-coupling capacitance to the wire-to-substrate capacitance could be as high as 3:1. As a result, the cross-coupling capacitance becomes significant as compared to the wire-to-substrate capacitance. Thus, previously unrelated events such as transitions on two closely-located lines, begin to influence one another. This phenomenon is known as crosstalk and is highly undesirable in digital circuits, since they have adverse effects on both the delay and the power consumption in the circuit, leading to signal integrity and reliability failures. This will cause the circuit to malfunction completely or worse still, function intermittently.

For example, a memory bank consisting of memory cells may be rejected because of coupling faults between adjacent memory cells, although this may have passed the design specification.

Sources of Crosstalk: With scaling down of technology, the interconnect assumes increasing importance in the design. The interconnect not only dominates the delay in the circuit but also consumes about 30% of the total

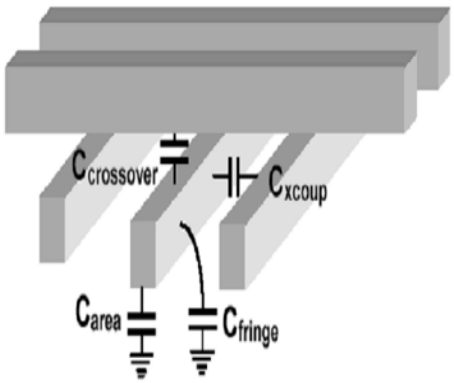


Fig 2: The various capacitance in a design

dynamic power in the design [5,7]. The most common interconnects are buses as bus-based interconnects reduces the number of connections compared to other interconnect styles. However, buses involve long lengths of wires running in parallel and in close vicinity of one another. With the decrease in the spacing between the lines compared to the thickness of the wires, the cross-coupling capacitance between the bus wires become comparable and often exceeds their wire-to-substrate capacitance.

A detailed depiction of the various capacitances related to the interconnects which run close to one another is provided in Figure 2.

The area capacitance Carea and capacitance due to fringe fields Cfringe together constitute the wire to substrate capacitance. The coupling capacitance Cxcoup exists between close wires running in the same plane. The coupling capacitance Ccrossover exists between close wires in different planes. When Cxcoup+Ccrossover becomes comparable to Carea+Cfringethe crosstalk phenomenon comes into effect. Due to crosstalk, the circuit could produce erroneous results due to unwanted delays on some of the lines [6]. Alternatively, the circuit could malfunction due to induced delays on some of the lines which could violate their timing requirements. The lines which cause delays on other lines are known as aggressors while the lines which are a?ected by these aggressors are known as victims. The crosstalk effects between aggressors and victims are twofold and are stated as follows:

- If the victim wire is at a steady state value (either logical ‘0’ or logical ‘1’) while the aggressor wire is switching (either from ‘0’ to ‘1’ or from ‘1’ to ‘0’), it could induce an unwanted positive or negative spike on the victim wire.
- If the victim wire is making a transition from ‘0’ to ‘1’ and the aggressor wire is also making a transition from ‘0’ to ‘1’, the transition of the victim wire will be hastened since the transition is being aided by the aggressor. On the other hand, if the victim wire is making a transition from ‘0’ to ‘1’ and the aggressor is making a transition from ‘1’ to ‘0’, the transition of the victim wire will be delayed since the aggressor is now opposing the victim’s transition. The transition of the victim wire from ‘1’ to ‘0’, with respect to the aggressor’s transition, can be similarly analyzed.

Effects of Crosstalk:

- Crosstalk is a concern because it can be a major contributor to the amount of jitter present in a device. Simply stated, jitter is the deviation of an edge from its expected location. A large amount of jitter in a serial communications link may cause bit errors in the received serial bit stream.
- Crosstalk noise may cause undesirable effects including excessive overshoot, undershoot, additional signal delay and even a reduction in signal delay.
- Logic speeds and clock speeds in particular, have increased significantly, thus leading to faster transition times. Also, at such high speeds the inductive properties of the wires come into play especially mutual inductance.
- Crosstalk may cause a signal to assume the wrong value. This is particularly critical when the signal is about to be latched, for a wrong value can be loaded into a storage element.
- Crosstalk may delay the settling of the signal to the correct value. This is often called noise-on-delay.

Eg: A large amount of jitter can cause a timing budget failure in a parallel system, or it can cause a clock and data recovery PLL to incorrectly recover the data in a serial system.

These effects have increased the interactions between signals and decreased the noise immunity of digital CMOS circuits. This has led to crosstalk noise being a significant problem for digital ICs that must be considered to tape out.

Design of Encoder and Decoder for Boundary Shift Codes: Since switching is one of the most important contributors to the crosstalk in VLSI circuits, it is imperative to encode the bits in such a way that the

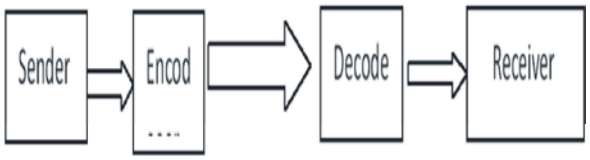


Fig 3: Model of n-bit communication channel

switching activity on the buses are reduced. Here we propose the design of encoder and decoder for boundary shift codes to minimize the crosstalk [7]. We model the bus as an n-bit communication channel as shown in eg fig 3. We say that the data words to be encoded are represented by symbols. Here there is a mapping between symbols and actual data words. The values placed on the channel by the encoder are called code words and the mapping between symbols and code words is called a codebook. If the codebook changes with time, then the encoding is said to have memory. During each signaling interval the encoding scheme is used to select and transmit an n bit word, called a codeword, from a possibly dynamic set called a codebook. The codebook is dynamic in the sense that it can be a function of previously transmitted code words. We call the overall encoding scheme a code.

A code is memory less if it uses a fixed codebook. We define the rateof the code as logbase 2mod(Cmin), where mod(Cmin) is the number of code words in the smallest codebook. This is the minimum number of bits that can be encoded during every signaling interval.

Here we want to know about the valid and invalid transitions during the encoding process. consider the following examples:

Code word at time 1:	0010	0000	0100	0100
	⇓	⇓	⇓	⇓
Code word at time 2:	0011	1111	0001	0010
	Valid	validvalid	invalid	

In the above eg four bits are used for transition from one codeword to other, the first, second, third are valid transitions since there is no adjacent bits to switch in opposite directions. But in fourth codeword the transition from the code word 1 to code word 2 causes the adjacent bits(bit 2 and 2) to switch in opposite directions. So it is invalid.

Boundary Shift Codes:

Self Shielding Codes: We say a pair of code words contains an invalid transitionif transitioning from one codeword to the other causes adjacent bits to switch in opposite directions. For example, the following code words contain invalid transition by bits 5 and 6.

C1	→	0	1	1	0	0	0	1	1
C2	→	1	1	0	1	1	0	1	0

Such transitions are undesirable because they increase cross talk noise. A code is self-shieldingif it does not allow invalid transitions. We assume that neighboring wires are routed in parallel. Therefore, the encoded bus is effectively self-shielding. In addition to avoid invalid transitions, we want to be able to differentiate our code words reliably even in the presence of errors (bit flips). This is possible if the code words in the codebook have a large enough Hamming distancebetween them, that is, if they differ in enough bit positions. For example, if the code words have a minimum Hamming distance of three between them, we can correct any single error since the "noisy" codeword must be closer to the original codeword than to any other. In general, if the minimum Hamming distance between any two code words is d, then we can either correct up to [d-1]/2 errors, or detect up to d- 1 errors. A binary code is said to be linear if the bitwise sum(mod 2) of any two codeword is a code word. A linear code can be represented by the independent set of basis code words. All other code words can then be formed by a linear combination of these. A standard representation of a linear code is the generator matrix, a matrix whose rows are an independent set of basis code words. In coding theory, a generator matrix is a basis for a linear code, generating all its possible code words. If the matrix is G and the linear code is C,

$$w = cG$$

where w is a unique codeword of the linear code C, c is a unique row vector and a bijection exists between w and c. A generator matrix for a (n, M = qk, d), q-code is of dimension k×n. Here n is the length of a codeword, k is the number of information bits, d is the minimum distance of the code and q is the number of symbols in the alphabet (thus, q = 2 indicates a binary code, etc.). Note that the number of redundant (Redundancy is the number of bits used to transmit a message minus the number of bits of actual information in the message, informally, it is the amount of wasted "space" used to transmit certain data.) bits is denoted r = n - k.

The standard form for a generator matrix is

$G=[I_k|P]$

where I k is a k×k identity matrix and P is of dimension k×r.

For example, a generator matrix for the code {0000; 0011; 0101; 0110; 1001; 1010; 1100; 1111} is

$$G=\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 1\end{bmatrix}$$

The generator matrix provides a simple way to map information bits to code words multiply the generator matrix by a column vector of information bits. A length n binary linear code encoding k bits with minimum hamming distance d is called an [n,k,d] code, Hence these codes are computationally infeasible for moderate to large bus sizes.

Boundary Shift Codes: In this section we give a general construction method for practical codes. We define a dependent boundaryin a codeword as a position where two adjacent bits differ. We denote the location by the position of the leftmost bit of the boundary. If two code words do not share any dependent boundaries, they cannot form an invalid transition. For example, consider the following code words:

C1	→	0	1	1	0	0	1	1	1
C2	→	1	1	0	0	1	1	1	0

Here *c*1and *c*2 have dependent boundaries {1; 3; 5} and {2; 4; 7}, respectively. Since there is no overlap, the transition must be valid.

Using this property, we note that if a codebook has code words with only even dependent boundaries, then performing a 1-bit circular right shift yields a new codebook with no even dependent boundaries. Since the two codebooks do not have overlapping dependent boundaries, we can alternate between the two to obtain a self-shielding code. We call this a boundary shift code.

For this construction we need an error correction code with no odd dependent boundaries. Let *C* be an [n; k; d] code and let *C*^-1be formed by duplicating each bit position in*C*.Then*C*^-1is a [2n; k; 2d] code with no odd dependent boundaries, since every bit in an odd bit position is followed by a copy. By alternating between *C*^-1and a shifted version of it, we obtain a [2n; k; 2d] self-shielding code. In addition, puncturing *C*^-1in the last bit position, i.e., removing the last bit in every codeword, yields a [2*n*- 1; *k*; 2*d*-1] code. Using a single parity check code in the above construction gives an infinite class of single error-correcting codes. In a [k+ 1; k; 2] single even parity check code the *k* data bits are appended with a final bit chosen to make the parity of the codeword even. Applying our construction, we obtain a [2k+ 1; k; 3] single error-correcting self-shielding code.

Design of Encoder: In the transmission side the input data has to be encoded before being transmitted, here the method of encoding the input data by using boundary shift codes is described.

As an eg consider the [4,3] boundary shift code with generator matrices.

Here 9 is the length (total no of encoded bits to be transmitted),4 is the dimension(actual no of input bits),3 is the minimum hamming distance.

$$G_0=\left[\begin{array}{cc|cc|cc|cc|c} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right]$$

$$G_1=\left[\begin{array}{c|cc|cc|cc|cc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

Generator matrices Go and Glare used for encoding during even and odd clock cycles respectively. Equivalently, *G*o be used for all cycles with the output then right shifted for odd cycles

From the above circuit we can infer that here x0, x1, x2, x3 are the four data inputs,y0 to y8 are the encoded data's out of given four input data's. Here the encoded bits y1,y3,y5,y7 are directly got from the input datas x0,x1,x2,x3. Here a five 2:1 Mux are used to produce the rest of the encoded bits y0,y2,y4,y6,y8 i.e., five parity bits. The following example illustrates how this code would be used to encode a 4-bit bus. The intermediate pre-shifted output is shown for clarity.

Time	input	preshifted output	output
x3	x2	x1	x0
0	1 0 1 0	110011000	110011000
1	0 1 1 1	001111111	100111111
2	1 0 0 0	110000001	110000001
3	0 1 0 0	001100001	100110000

Y8 y7 y6 y5 y4 y3 y2 y1y0

Consider first input 1010 at time 0, the preshifted output is 1 1 0 0 1 1 0 0 0

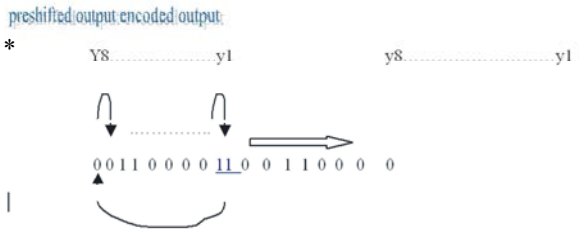
Here the hyperlinked bits (i.e. y7, y5, y3.y1) are the original bits of the input (x3, x2, x1, x0).

Then the encoded output i.e, y8,y6,y4,y2 also represent the input bits 1010. The y0 is based on the given data input. These operations are performed by duplicating each input bit, this yields the pre-shifted

output. Then append the parity check at the end i.e.y0 yielding the preshifted output. Here the time cycle is considered as even so there is no shift, therefore the encoded output is 110011000.

Consider the 1st time cycle which is odd, here the input is 0111, the preshifted output is 00111111 1, the hyperlinked bits are the original bits of the input, then the nonhyperlinked bits also represent the input bits, y0 depends on the input.

But the change comes because of the clock cycle, since it is odd, we perform a 1-bit circular right shift before transmitting.



Then the encoded bits are transmitted.

Design of Decoder: The input data has been encoded and transmitted. At the receive side, we first undo the right shift if the clock cycle is odd, then decoding can be done by majority vote, where the two "copies" of the desired bit are augmented by a third, generated by taking the sum (mod 2) of one copy of each of the other information bits and the parity check.

The above fig gives the full description of decoding, the operation is illustrated below.

The table below shoes the output at the receiver for the previous above eg.

noisy output		majority vote	data
100011000	→	(101)(000)(111)(000)	1010
001111110	→	(001)(110)(110)(110)	0111
010000001	→	(011)(001)(001)(001)	1000
011100000	→	(011)(110)(001)(001)	1100

For example, in an even cycle three independent copies of the first information bit are given by $y0$, $y1$and (y2+y4+y6+y8)mod2. A single error will affect at most one of the three copies and is therefore correctable. The following example shows how noisy versions of the code words in the previous example would be decoded after unshifting. The highlighted bits correspond to errors. [6, 2] The first three code words are decoded correctly and correct output is produced at the receiver, because the hamming distance is less than three.

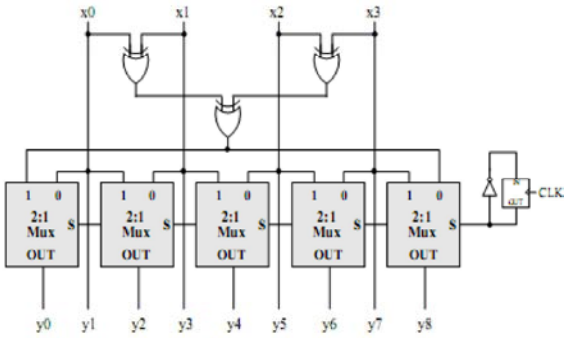


Fig 3: Encoding circuit for [,4,3] code.

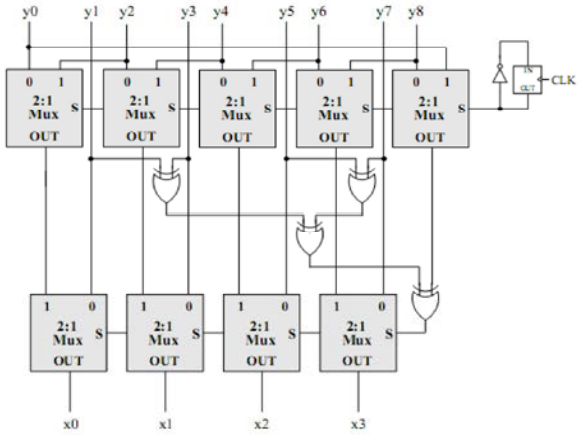


Fig 4: Decoding circuit for [9,4,3] code

The last codeword is decoded incorrectly as it contains two errors and is therefore beyond the code's error correcting capability. This reduces the switching of buses, thus minimizing crosstalk.

Illustration: For large bus sizes the increased circuit depth may lead to significant delay. This can be reduced by breaking the bus into smaller sub-busses with shielding wires inserted between them. This results in a slight increase in the number of wires and gates needed, but limits the circuit depth. In addition, it also increases the error correction capability, since single errors in each sub-bus can then be corrected independently. Since the codes constructed in the previous section are based on very simple error-correcting codes, they can be encoded and decoded efficiently. Figures 3 and 4 show an encoder and decoder respectively for the [4 ,3] code. These circuits can be generalized in a straightforward manner for larger single error-correcting codes. Gate counts and maximum circuit depths are given in Table 1. for a range of bus sizes and closed form expressions are given for the general case.

Table 1: Encoder/decoder gate counts and delay for single error correcting boundary shift codes.

Bus Size	Wires	Encoder		Decoder	
		Gates	Delay	Gates	Delay
4	9	10	3 gates	15	4 gates
8	17	18	4 gates	27	5 gates
16	33	34	5 gates	51	6 gates
32	65	66	6 gates	99	7 gates
64	129	130	7 gates	195	8 gates
n	$2n+1$	$2n+2$	$\lceil \log_2 n \rceil + 1$	$3n+3$	$\lceil \log_2 (n+1) \rceil + 1$

Advantages:

- A potentially useful feature of the proposed codes is that they are systematic, that is, the information bits are embedded in the encoded codeword and can therefore be obtained without any decoding logic. For example, the information bits of the [4, 3] code are given by bits y1, y3, y5and y7. Of course, simply using these bits rather than decoding the full codeword sacri?ces the error correction capabilities of the code. However one can imagine a scenario where error correction may only be necessary for certain destinations on the bus that are relatively far from the source, while other destinations may opt for error detection or simply picking the information bits off of the encoded codeword.
- A clear advantage of boundary shift codes is that they do not suffer from error propagation since the codebook does not depend on the choice of previous code words transmitted, rather it is only a function of the time index. As long as the source and destination are synchronized, no error propagation will occur.
- Scalable construction-simple encoder/decoder-integer rates; systematic.

CONCLUSION

In this paper we have considered the designing bus encoding schemes that provide both crosstalk minimization and active error correction. This is an important improvement and is particularly applicable to scenarios, such as nanotechnology and radiation hardened circuits, where random errors are a concern in addition to crosstalk interference.

One of the most significant contributions of this paper is a practical class of error-correcting self-shielding codes called boundary shift codes. For the specific case of single error-correcting boundary shift codes, given gate level encoding and decoding circuits. We also plan to write a verilog code for the above design, simulate and implement it.

REFERENCES

1. Victor, B. and K. Keutzer, 2001. Bus encoding to prevent crosstalk delay. Proc. IEEE/ACM Intl. Conf. on Computer Aided Design, pp: 57-69.
2. Ketann, Patel And Igorl and Marko, 0000. Error-correction and crosstalk avoidance in DSM busses.
3. Sainarayanan, K.S., C. Raghunandan, Ravindra and J.V.R. Srinivas, 2007. Bus coding to minimize redundant bit transitions M.B. TENCON.
4. Saravanan, T. and R. Udayakumar, 2013. Comparision of Different Digital Image watemarking techniques, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1684-1690.
5. Saravanan, T. and R. Udayakumar, 2013. Optimization of Machining Hybrid Metal matrix Composites using desirability analysis, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1691-1697.
6. Saravanan, T. and R. Udayakumar, 2013. Simulation Based line balancing of a single piece flow line, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1698-1701.
7. Saravanan, T. and R. Udayakumar, 2013. Comparision of Different Digital Image watemarking techniques, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1684-1690.