Design and Implementation of Memory-based Cross – Talk Reducing Algorithm to Eliminate Worst Case Crosstalk in On- Chip VLSI Interconnect

Souvik Singha, G. K. Mahanti

Abstract:- Cross- talk induced Delay and power consumption are two of the most important constraints in an on- chip bus design. In same metal the ratio of cross-coupling capacitance between adjacent on-chip wires is quite larger. As a consequence, cross- talk interference becomes a serious problem for VLSI design. On chip bus delay maximized by cross-talk effect when adjacent wires simultaneously switch for opposite signal transition directions. In this paper we propose a memory- based cross-talk reduction technique to minimized the cross-talk for on-chip buses based on graph representation. In this approach that represents all the illegal code words canonically generates code words efficiently. As a result, a memory-based cross-talk avoidance CODEC would need to partition large buses into small groups. Our approach is applicable for reducing the cross talk, using a unified implicit formulation. It can actually speed up the bus by exploring cross talk among neighboring wire. By using this approach, we have developed a CODEC based algorithm to minimize the cross- talk or interference in on- chip buses.

Keywords:- crosstalk, Bus Encoding, On-chip bus, Crosstalk Free Algorithm, Delay.

I. INTRODUCTION

On- chip bus is widely used in VLSI. In bus structure, cross- talk immunity is more important because long interconnect wires often run together and in parallel [2,3,4]. The inter- wire coupling capacitance between adjacent wires of the bus is relatively larger than other interconnects. Besides physical capacitance increase, simultaneous switching for opposite transition directions between adjacent wires makes the effective inter-wire coupling capacitance doubled [1,7,8]. As a consequence, signal wires exhibit a significant delay variation and noise immunity problem. This problem is aggravated for long on- chip buses [5,8,9]. As technology shrinks, the coupling capacitance exceeds the self

capacitance, which causes increased delay and power dissipation on the interconnect lines. To sum up with shrinking the feature size, increasing the die size, scaling the supply voltage, increased interconnect density and faster Clock rates, the on-chip buses suffer from higher power consumption and large propagation delay due to capacitive crosstalk[5]. Since, both power consumption and delay incurred by a system bus, with increase in the coupling and the self capacitances, minimizing these capacitances is a major challenge in modern DSM design. Bus encoding schemes can be achieve the same amount of bus delay improvement as passive shielding, with a much lower area overhead [1,4]. These codes are commonly referred to as Crosstalk Avoidance Codes (CACs). CACs can be memory- less [6,4] or memory- base [1]. Memory- based coding approaches generate a codeword based on the previously transmitted code and the current data word to be transmitted. Although these type of codes need fewer additional bus wires. On the other hand the memory- less coding approaches use a fixed code book to generate a code word. The focus of this paper is to develop memory based encoding techniques to alleviate crosstalk in on- chip buses.

Different approaches have been proposed for reducing crosstalk by eliminating specific data transition patterns. Some schemes focus on reducing the energy consumption, while other focus on minimizing the delay. Certain schemes offer improvements in both. Certain schemes incur different area overheads since they require additional wires, spacing between wires or both. Our encoding approaches allow for the selective reduction of the crosstalk effects in on- chip buses. By proving immunity from cross talk in buses, our encoding based techniques reduced the cross talk induced delay variation effect in on chip buses. This has the important benefit of reducing the maximum delay as well as reducing signal integrity problems in the bus signals. In the sequel, we refer to bus overhead as the additional number of bits required in order encoding a bus in a cross- talk free manner.

II. CROSS TALK

One of the important effects of coupling capacitances in that they may induce unwanted voltage spikes in neighboring bus wire. This is known as cross talk.

Manuscript received on April 26, 2012.

Souvik Singha, Department of Electronics and Communication Engineering, National Institute of Technology, Durgapur, India, Mobile No. 09564620695, (e-mail: singha.souvik@gamil.com).

G.K.Mahanti, Professor, Department of Electronics and Communication Engineering, National Institute of Technology, Durgapur, India,
(e-mail: gautammahanti@yahoo.com).

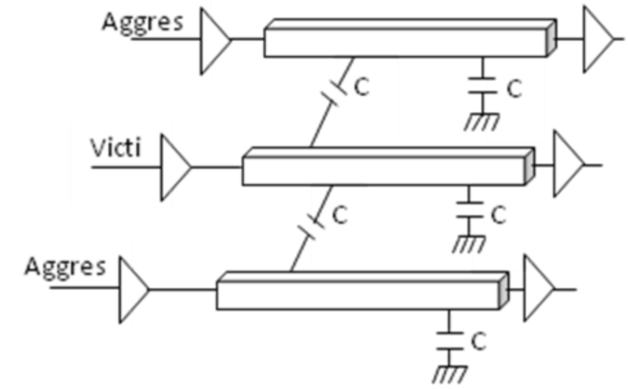


Figure 1: Delay with three type transition on a three-line directional bus.

Figure – 1 show the data bus model when no switching transition occurs in termed an aggressor and the wire on which it produces a noise spike in termed as a victim.

The problem of reducing capacitive cross talk effect on buses depends on the transition on the bus lines. The effect of an aggressor on a victim depends on a number of factors, and not every aggressor will inject an appreciable amount of noise into a victim. Typically, an aggressor wire is physically adjacent to a victim Wire and they may be modeled as being connecter by a distributed coupling capacitance [10, 11]. Hence a switching event in the aggressor wire while the victim wire is silent can result in the injection of at current in to the victim wire, causing an electrical spike [11].

For example, consider three wires that run in parallel for a significant distance with minimum space. If the middle wire switches from low to high, while its neighbors are simultaneously switching from high to low, the effective capacitance of the middle wire becomes doubled compared to the case where the neighbors are quiet. On the other hand, if all the three wires are simultaneously switching in the same direction, the coupling capacitance of the middle wire becomes zero. It has been shown that the delay and power for a long bus is strongly a function of the coupling capacitance between the wire [9, 11].

III. CROSS TALK CLASSIFICATION AND DELAY ESTIMATION

Crosstalk in an on-chip bus has been shown to be dependent on the data patterns on the bus and has a significant impact on the signal delay as well as the overall energy consumption of the bus [2,3,5,6]. Considering the introduced 3- wire bus model and examining the total effective capacitance (Ceff) driven by the driver of the central (victim) line for different transitions of two adjacent (aggressor) lines, we can distinguish 5 different delays in a first order approximation if wire spacing are equal. From figure 2 illustrate a simplified on-chip bus model with crosstalk. In the figure, CL denotes the load capacitance, which includes the receiver gate capacitance and also the parasitic wire-to-substrate parasitic capacitance. CI is the inter-wire coupling capacitance between adjacent signal lines of the bus we calculate that: (i) The victim and the aggressors switch in the same direction. The effective capacitance charged by the driver of the victim line is given by: Ceff = CL . (ii) The victim and one aggressor switch in the same direction and the other aggressor is quite: Ceff = CL + CI . (iii) The victim switches and both aggressor are quiet, or the victim and one aggressor switch in direction, while the other aggressor switches oppositely: Ceff = CL + 2 CI . (iv) The victim and one aggressor switch oppositely, while the other aggressor is quite: Ceff = CL + 3CI. (v) The victim and both aggressors switch oppositely: Ceff = CL + 4CI .

Consider a group of three wires in an on-chip bus, which are drive by signals bi+1 , bi , bi-1. The total capacitance of driver bi is dependent on the state of bi-1 and bi+1. The degree of crosstalk in an on-chip bus is depending on the transition patterns on the bus. Based on the model figure 2 the delay ti of the ith wire in a data bus is given as [7,9].

$$t_i = abs(K.C_L.\Delta V_i + K.C_i.\Delta V_{i,i-1} + K.C_i.\Delta_{i,i+1}) \qquad (1)$$

Where k is a constant determined by driver strength and wire resistance, ΔV_i is the voltage change on the ith line and $\Delta V_{i,k} = \Delta V_i - \Delta V_k$ is the relative voltage change between the ith and kth line. Since on-chip buses are generally full-swing binary buses, we can assume that the two output voltage levels are V_{out} and 0V and hence: $\Delta V_i \in \{0, \pm V_{out}\} and\ \Delta V_{i,k} \in \{0, \pm V_{out}, \pm 2V_{out}\}$. If we let $\lambda = \frac{C_I}{C_L}$, above expression can be written as:

$$t_i = K.C_L.V_{out}.abs(\delta_i + \lambda.\delta_{i,i-1} + \lambda\delta_{i,i+1}) \qquad (2).$$

here $\delta_i \in \{0,1\}$ is the normalized voltage change on ith line. $\delta_{i,i\pm1} \in \{0, \pm1, \pm2\}$ is the normalized relative voltage change on ith line .

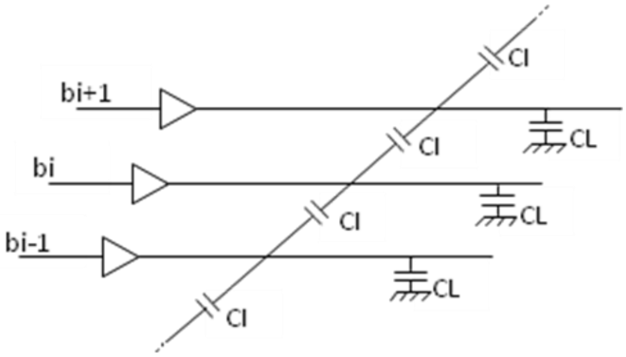


Figure 2: On-chip bus model with crosstalk.

The δ_i term corresponds to the intrinsic delay and remaining two terms correspond to crosstalk induced delay. Since $\lambda >> 1$ the first term has negligible contribution to the delay. If we define C_{eff} , i as the total capacitance of the driver of ith line, we have:

$$C_{eff,i} = C_L.abs(\delta_i + \lambda.\delta_{i,i+1} + \lambda.\delta_{i,i+1}) \qquad (3).$$

$$= C_L.abs(\Delta V_i + \lambda.\Delta V_{i,i-1} + \lambda.\Delta V_{i,i+1}) \qquad (4).$$

We know that, $\Delta V_{i,k} = \Delta V_i - \Delta V_k;\ i.e\ \Delta V_{i,i-1} = \Delta V_i - \Delta V_{i-1} and\ \Delta V_{i,i+1} = \Delta V_i - \Delta V_{i+1}.$ From the above expression, we get: $\min(C_{eff,i}) = C_L - 0.C\ and\ \max(C_{eff,i}) = (1+4.\lambda).C_L.$

Crosstalk patterns corresponding to the different $C_{eff,\ i}$ values are classified as

0.C, 1.C, 2.C, 3.C and 4.C patterns. Since speed of the data bus is determined by max{ $C_{eff,\,i}$ }overall the bit in the bus, we observe that by eliminating 4.C crosstalk on all lines in the bus, we can increase the maximum speed of the bus by ~ 38%. If 3.C and 4.C crosstalk can be eliminated on all lines, we can speed up the bus by ~100%.

IV. CROSS-TALK SEQUENCES AND BUS TRANSITION PATTERNS

In this section, we discuss a scheme to eliminate cross-talk sequences, with a low overhead than the previous scheme. In Figure 2 we consider a group of three wires in an on chip bus. In best case, $b_{i+1,}$ $b_{i,}$ $b_{i-1,}$ all simultaneously transition in the same direction. But in the worst case, b_{i+1} and b_{i-1} simultaneously transition in the opposite direction as b_i . In the best case, the total effective capacitance of b_i is $C_{min} = C_L$, and in the worst case, the effective capacitance is $C_{max} = 4.C_I + C_L$. with $\lambda >> 1$ we observe that $\frac{C\max}{C\min}$ >>1, and hence the delay of the bus signals strongly depends on the data pattern being transmitted on the bus.

A.O.C Cross-talk (000 → 111) :

On the transition pattern on the wire of interest as well as its immediate neighbors on either side we clearly state that:

$$\Delta V_i = 1,\ \Delta V_{i+1} = 1, \Delta V_{i-1} = 1.$$

$$C_{eff,i} = C_L\left[\Delta V_i + \lambda(\Delta V_i - \Delta V_{i-1}) + \lambda(\Delta V_i - \Delta V_{i+1})\right]$$

$$= C_L\left[1 + \lambda(1-1) + \lambda(1-1)\right]$$

$$= C_L\left[1+0\right] = C_L$$

B.1.C Cross-talk (011 → 000):

$$\Delta V_i = 1,\ \Delta V_{i+1} = 1.$$

$$C_{eff} = C_L\left[\Delta V_i + \lambda(\Delta V_i - \Delta V_{i-1}) + \lambda(\Delta V_i - \Delta V_{i+1})\right]$$

$$= C_L\left[1 + \lambda(1-0) + \lambda(1-1)\right]$$

$$= C_L\left[1+\lambda\right].$$

C.2.C Cross-talk (010→ 000) :

$$\Delta V_i = 1.$$

$$C_{eff,i} = C_i\left[\Delta V_i + \lambda(\Delta V_i - \Delta V_{i-1}) + \lambda(\Delta V_i - \Delta V_{i+1})\right]$$

$$= C_L\left[1+\lambda+\lambda\right]$$

$$= C_L\left[1+2\lambda\right].$$

D.3.C Cross-talk (010 → 100):

$$\Delta V_i = 1,\ \Delta V_{i+1} = 1,\ \Delta V_{i-1} = 1.$$

$$C_{eff,i} = C_L\left[1 + \lambda(\Delta V_i - \Delta V_{i-1}) + \lambda(\Delta V_i - \Delta V_{i+1})\right]$$

$$= C_L\left[1 + \lambda(1-0) + \lambda(1-(-1))\right]$$

$$= C_L\left[1+\lambda+2\lambda\right]$$

$$= C_L\left[1+3\lambda\right].$$

E. 4.C Cross-talk (010 → 101):

$$\Delta V_i = 1,\ \Delta V_{i+1} = 1,\ \Delta V_{i-1} = -1.$$

$$C_{eff,i} = C_L\left[\Delta V_i + \lambda(\Delta V_i - \Delta V_{i-1}) + \lambda(\Delta V_i - \Delta V_{i+1})\right]$$

$$= C_L\left[1 + \lambda(1+1) + \lambda(1+1)\right]$$

$$= C_L\left[1+4\lambda\right].$$

As a result of this large delay variation, the worst case delay of a signal in an on-chip bus also increased, limiting system performance. The problem due to crosstalk is aggravated in long on-chip buses, since bus signal are longer and therefore more capacitive, resulting in large worst case delay.

V. FORBIDDEN PATTERN FREE CACs

Forbidden patterns are defined as the two 3-bit patterns "010" and "101". A forbidden pattern free code is a set of code words which do not contain forbidden patterns on any 3 adjacent bus bit [7]. For example 11100110 and 11000110 are forbidden pattern free vectors while 11010011 and 11110110 are not FPF vectors. By eliminating the forbidden patterns in the code words, it is guaranteed that C_{eff} for any bit in the bus does not exceed $(1+2\lambda)C_L$ and hence the maximum delay is reduced by ~60% compared to an un-coded bus.

VI. ELIMINATING CROSSTALK USING ENCODING TECHNIQUE

Table 1 shows a particular mapping which eliminates 3.C and 4.C crosstalk sequence using forbidden pattern free encoding technique. The encoder we implemented is shown in figure 3, for a 16-bit bus. The input is divided into four 4-bit groups and the data of each group is encoded using a 4-5 logic gates.

Table 1: Encoded 4 → 5 input-output

Input	Output
$d_1\,d_2\,d_3\,d_4$	$q_1\,q_2\,q_3\,q_4\,q_5$
0 0 0 0	0 0 0 0 0
0 0 0 1	0 0 0 0 1
0 0 1 0	0 0 1 1 0
0 0 1 1	0 0 0 1 1
0 1 0 0	0 1 1 0 0
0 1 0 1	0 0 1 1 1
0 1 1 0	0 1 1 1 0
0 1 1 1	0 1 1 1 1
1 0 0 0	1 1 1 1 1
1 0 0 1	1 1 1 1 0
1 0 1 0	1 1 0 0 1
1 0 1 1	1 1 1 0 0
1 1 0 0	1 0 0 1 1
1 1 0 1	1 1 0 0 0
1 1 1 0	1 0 0 0 1
1 1 1 1	1 0 0 0 0

Algorithm 1 Forbidden pattern free encoding technique:

```
Encode (v, n).

1) Subdivide the input vector v in the set of group
   consists n bits each group
     $(V_{j,1}, V_{j,2}, \ldots\ldots V_{j,n})$
   $(V_{j+1,1}, V_{j+1,2}, \ldots\ldots V_{j+1,n})$.
2) Now add an additional signal called a group
   complement signal that define as follows:
     For the group of n = 4 bits signal

     $d_{j,1} = V_{j,1}$
     $d_{j,2} = (V_{j,2} V_{j,3} + V_{j,2} \overline{V_{j,4}}) \oplus V_{j,1}$
     $d_{j,3} = (V_{j,2} + V_{j,3} \overline{V_{j,4}}) \oplus V_{j,1}$
     $d_{j,4} = (V_{j,3} + V_{j,2} V_{j,4}) \oplus V_{j,1}$
     $V_{j,5} = V_{j,4} \oplus V_{j,1}$
3) if $(d_{j,5} \neq d_{j+1,1})$ then
     Encoded output
     $\{d_{j,1}, d_{j,2}, d_{j,3}, d_{j,4}, d_{j,5}\}\{(\overline{d_{j+1,1}}), (\overline{d_{j+1,2}}), (\overline{d_{j+1,3}}), (\overline{d_{j+1,4}}), (\overline{d_{j+1,5}})\}$

   Else

     $\{d_{j,1}, d_{j,2}, d_{j,3}, d_{j,4}, d_{j,5}\}\{d_{j+1,1}, d_{j+1,2}, d_{j+1,3}, d_{j+1,4}, d_{j+1,5}\}$

4) Exit.
```

Our experimental results shown in figure 5, that the proposed technique results is reduced delay variation due to crosstalk. As a result the overall delay of a bus actually decreases even after the use of encoding scheme. Here the input bus is split into 4-bit groups. The values of each group are compared with their previously transmitted values. The group data or its complement is transmitted based on the output of the comparator. Again one group complement bit is transmitted per group. We can see that each group is totally independent of other groups and therefore all the encoding is done in a single step, thus improving circuit speed. On the other hand, we focus on crosstalk in buses where the problem is significantly more acute since buses tend to be longer, resulting in large capacitances and therefore more aggravated worst-case delays. In this approach to determine the effective bus of width m that can be encoded in a k.c free manner, using a physical bus of width n.

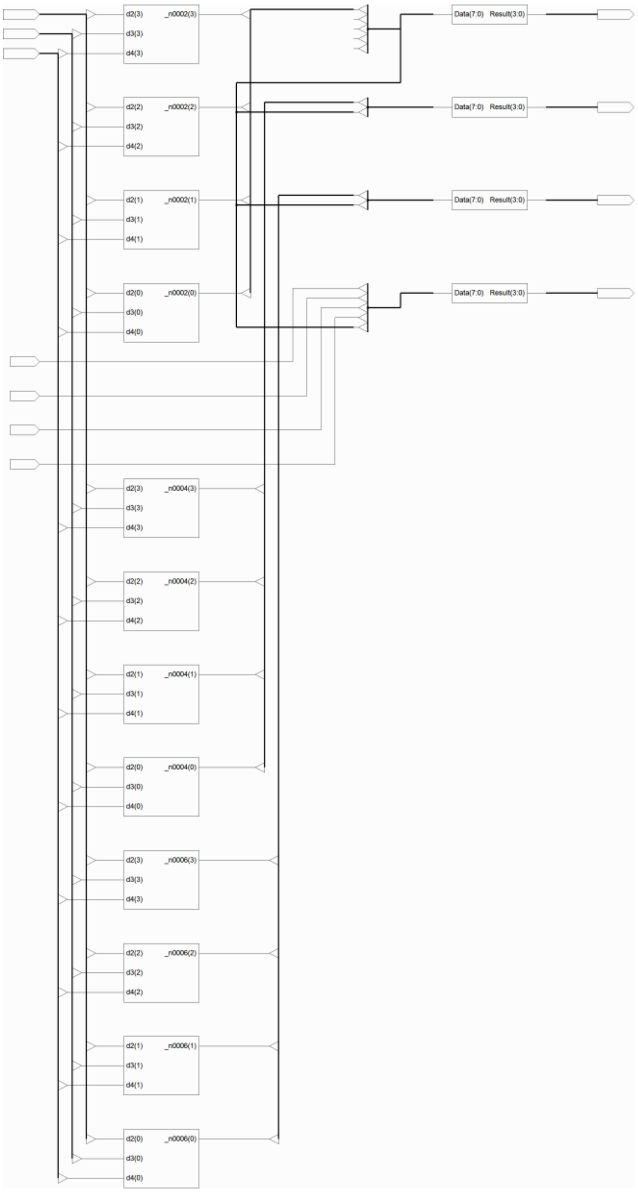


Figure 3: 16 bit codec structure.

VII. CROSSTALK REDUCTION CODING

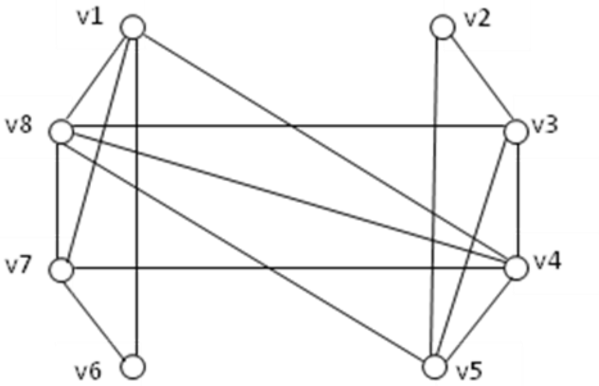


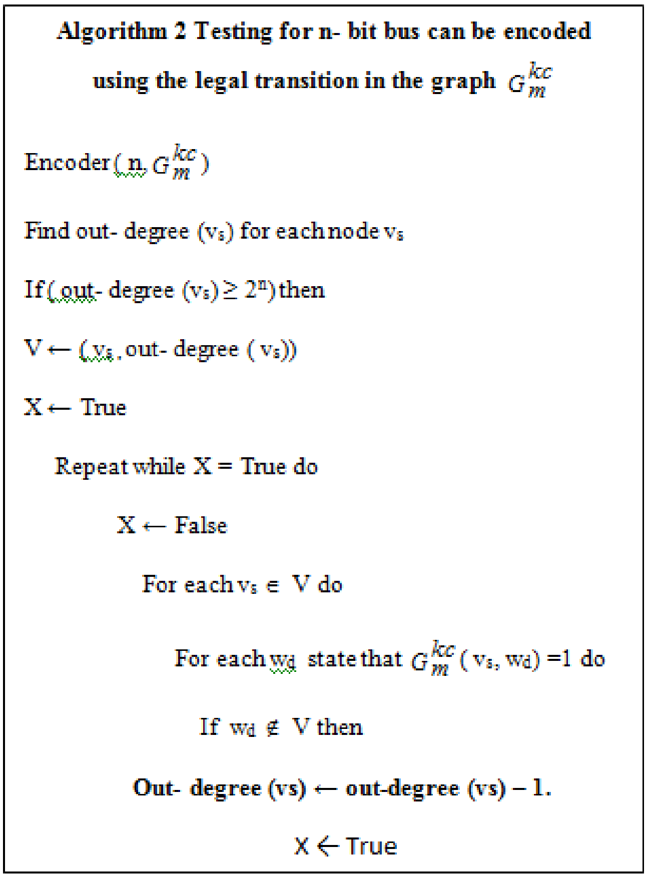
Figure 4: Graph with different cross talk constraints.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In the case of crosstalk- based bus encoding, a graph $G_m^{kc} = (V, E)$ can be constructed for an m- bit bus. G_m^{kc} Consists of a set of 2 m vertices V and a set of edges E. each v ∈ V represent one of 2m code words that may be transmitted on the bus. In this section, we present a method for memory- based crosstalk reduction code generation using Reduce Ordered Binary Decision Diagrams (ROBDDs) [1]. It is a two- step approach to determine the effective bus of width n that can be encoded in a cross- talk free manner. for (1≤ k ≤ 4) using a physical bus width m.First we construct an Algorithm, that is G_m^{kc} which encodes all vector transitions on the m- bit bus that are cross talk free [1].

Using this algorithm G_3^{kc} ((vi+1, vi+2, vi+3)← (v1, v2, v3), (wi+1, wi+2, wi+3)← (w1, w2, w3)) referred to the ROBDD variable substitution of (wi+1, wi+2, wi+3) by (v1, v2, v3) and (w1, w2, w3) respectively[1].

Now from G_m^{kc} we find the effective bus width n, such that an n- bit data bus can be encoded in a cross- talk free manner. If an n- bit (n < m) bus can be encoded the legal transitions in G_m^{kc} then each source vertex vs ∈Vc has at least 2m out going edges (vs, wd) to destination vertices wd. such that the destination vertex wd ∈ Vc. so the cardinality of Vc is at least 2n . If the number of outgoing- edges of any vs is greater than 2n, we add vs and its out- degree into V. For each vs ∈ V, we next check if each of its destination nodes wd are in V. If wd ∉ V, we decrement the out- degree of vs becomes less than 2m, we remove vs from V. This operation is performed until convergence. If at this point, the number of surviving vertices in V is 2n or more, then an n- bit codec can be constructed from the graph G_m^{kc}. We initially call the algorithm with n = m-1 where m is the physical bus size than an n- bit bus cannot be encoded using G_m^{kc}, then we determine n. We repeat this until we find a value of n such that the n- bit bus can be encoded by the graph G_m^{kc}. So the Algorithm is given below.



Algorithm 2 Testing for n- bit bus can be encoded using the legal transition in the graph G_m^{kc}

```
Encoder( n, $G_m^{kc}$ )
Find out- degree ($v_s$) for each node $v_s$
If( out- degree ($v_s$) $\geq 2^n$) then
V ← ( $v_s$ , out- degree ( $v_s$))
X ← True
  Repeat while X = True do
      X ← False
        For each $v_s \in$ V do
           For each $w_d$ state that $G_m^{kc}$ ( $v_s$, $w_d$) =1 do
             If $w_d \notin$ V then
          Out- degree (vs) ← out-degree (vs) – 1.
                   X ← True
```

Let the input of the Algorithm is n = 2, V ← ø. Then the algorithm generates the following:

Step 1: We calculate Out- degree of all the vertices of the above graph.

Out - degree $(v_1) = 4 \geq 2^n$ i.e $V = \{ v_1 \}$.
Out - degree $(v_2) = 2 > \neq 2^n$ i.e $V = \{v_1\}$.
Out- degree $(v_3) = 4 \geq 2^n$ i.e $V = \{ v_1, v_3\}$.
Out- degree $(v_4) = 5 \geq 2^n$ i.e $V = \{ v_1, v_3, v_4\}$.
Out- degree $(v_5) = 4 \geq 2^n$ i.e $V = \{ v_1, v_3, v_4, v_5\}$.
Out- degree $(v_6) = 2 > \neq 2^n$ i.e $V = \{ v_1, v_3, v_4, v_5\}$.
Out- degree $(v_7) = 4 \geq 2^n$ i.e $V = \{ v_1, v_3, v_4, v_5, v_7\}$.
Out- degree $(v_8) = 5 \geq 2^n$ i.e $V = \{ v_1, v_3, v_4, v_5, v_7, v_8\}$.
So $V \leftarrow \{ v_1, v_3, v_4, v_5, v_7, v_8\}$.

Step 2: For each v_s in V.

2.1 For v_1 , from the vertices V { (v_1, v_4), (v_1, v_6), (v_1, v_7), (v_1, v_8) }.
So $v_6 \notin$ V
Out- degree (v_1) = 4-1=3
Now out- degree of $v_1 < 2^n$ so
$V \leftarrow V – (v_1)$
$V = \{ v_3, v_4, v_5, v_7, v_8\}$.

2.2 For v_3, from the vertices V { (v_3, v_2) (v_3, v_4), (v_3, v_8), (v_3, v_5) }.
So $v_2 \notin$ V
Out- degree (v_3) = 4-1= 3
Now out – degree of $v_3 < 2^n$
So $V \leftarrow V- (v_3)$
$V = \{ v_4, v_5, v_7, v_8\}$.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

2.3 For v_4, from the vertices V {(v_4, v_3), (v_4, v_1), (v_4, v_8), (v_4, v_7), (v_4, v_5) }.
So $v_3 \notin V$
Out- degree of $(v_4) = 5-1= 4$.
Again $v_1 \notin V$
Out- degree $(v_4) = 4-1= 3$
Now out- degree of $v_4 < 2^n$.
So $V \leftarrow V- (v_4)$
$V= \{v_5, v_7, v_8\}$.

2.4 For v_5, from the vertices V {(v_5, v_4), (v_5, v_3), (v_5, v_2), (v_5, v_8)}.
So $v_4 \notin V$
Out- degree of $v_5 = (4-1) =3$
Again $v_3 \notin V$
Out- degree of $(v_5) = (3-1) = 2$
Again $v_2 \notin V$
Out- degree of $(v_5) = (2-1) =1$
Now out- degree of $v_5 < 2^n$
So $V \leftarrow V- (v_5)$
$V= \{v_7, v_8\}$

2.5 For v_7, from the vertices V { (v_7, v_6), (v_7, v_4), (v_7, v_1), (v_7, v_8)}.
So $v_6 \notin V$
Out- degree of $(v_7) = (4-1) = 3$
Again $v_4 \notin V$
Out- degree of $(v_7) = (3-1) = 2$
Again $v_1 \notin V$
Out- degree of $(v_7) = (2-1) = 1$
Now out- degree of $v_7 < 2^n$
So $V\leftarrow V- (v_7)$
$V = \{v_8\}$.

2.6 For v_8, from the vertices V { (v_8, v_1), (v_8, v_3), (v_8, v_4), (v_8, v_5), (v_8, v_7)}.
So $v_1 \notin V$
Out- degree of $(v_8) = (5-1) = 4$
Again $v_3 \notin V$
Out- degree of $(v_8) = (4-1) = 3$
Again $v_4 \notin V$
Out- degree of $(v_8) = (3-1) = 2$
Again $v_5 \notin V$
Out- degree of $(v_8) = (2-1) = 1$
Again $v_7 \notin V$
Out- degree of $(v_8) = (1-1) = 0$

So using this algorithm for n = 2 we found that V = ø. So (n = 2) cannot be transfer. We conclude that for the given graph n = 2 number of bit cannot be encoded. Further we examine n = 1, that number of bit can be encoded easily for the above encoded graph.

VIII. IMPLEMENTATION AND RESULTS

The memory – based CODEC designs are more complicated. Bus partitioning can also be used in the memory- based CODEC designs, where a wide bus is partitioned into small groups and encoded and decoded independently. For example, for the input bus size of 32, the memory based code required only 40 wires, as opposed to 46 wires needed for a memory less code. We run our algorithm to construct the graph G^{kc}_{m} and to find the effective bus width m. If two buses of width n and n+1 have an identical effective bus width of m, in that case, overhead as defined above is larger for the bus of real width n+1. For wider buses, we recommend that the bus be partitioned into smaller bus segments and each segment be encoded and decoded independently. In such a situation, we could choose a bus width n that yields the lowest overhead. In particular, the choice of 4 or 6 bit segments is preferable over 5 or 7 bit segments.

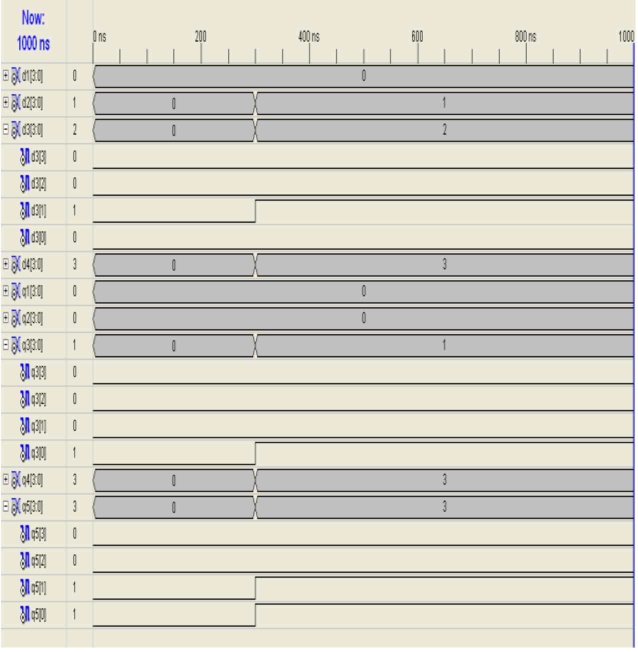


Figure 5: Experimental result.

IX. CONCLUSION

In this work, we propose to employ bus encoding to eliminate delay within a bus. In this paper present a rigorous analysis of the theory behind “Memory- Based CODEC” and give the fundamental theoretical limits on the performances of CODEC with and without the memory. In this paper, we have developed a technique for reduction of maximum bus delay caused by cross talk reduction based algorithm in an on- chip buses. By using this approach to find the effective cross- talk free bus bandwidth m of a bus with physical width n. Here we developed memory- based crosstalk free algorithm and compare to existing algorithm we get the complexity that is roughly O(m).

REFERENCES

1. C. Duan, K. Gulati and S.P. Khatri, “Memory-based Cross-talk Canceling CODECs for On-chip busses”, ISCAS 2006, pp 4-9.
2. C. Duan, A.Tirumala and S.P.Khatri, ”Analysis and Avoidance of Cross-talk in On-chip Bus”, HotInterconnects, 2001,pp 133-138.
3. C. Duan and S. P. Khatri, ”Exploiting Crosstalk to Speed up On-chip busses”, DATE 2004, pp 778-783.
4. C.Duan, C.Zhu, S.P.Khatri, “Forbidden Transition Free Crosstalk Avoidance CODEC Design” DAC 2008, June 8-13, 2008, Anaheim, California, USA, pp-986-991.

5. C. Duan, V. Cordero and S. P. Khatri, ”Efficient On-Chip Crosstalk Avoidance CODEC Design”, IEEETransactions on VLSI Systems, to appear.
6. Madhu Mutyam, ”Preventing Crosstalk Delay using Fibonacci Representation”, Intl Conf. on VLSI Design, 2004, pp 685-688.
7. Bret Victor and K. Keutzer,”Bus Encoding to Prevent Crosstalk Delay”, ICCAD, 2001, pp 57-63.
8. M. Mutyam, “ACM Transactions on Design Automation of Electronic Systems”, Vol. 14, No. 3, pp. Article 43, pp. 1-20, 2009
9. S.R. Sridhara, A. Ahmed, and N. R. Shanbhag, ”Area and Energy-Efficient Crosstalk Avoidance Codes for On-Chip busses”, Proc. of ICCD, 2004, pp 12-17.
10. J. -S Yim and C. –M. Kyeng. “ Reducing cross- coupling among interconnect wires in deep- submicron Datapath design”.36th design Automation Conference (DAC) ,june 1999 ,pp. 485-490.
11. K. Hiroes and H. Yasuura, “ A bus delay reduction technique considering crosstalk”. Design, Automation and Test in Europe (DATE), mar 2000, pp 441-445.

AUTHORS PROFILE

Mr. Souvik Singha is working as an Assistant Professor in the department of Computer Science & Informatics, Bengal Institute of Technology & Management, Santiniketan. He received his Diploma in Engineering (CST) under West Bengal Board of Technical Education in the year of 2002 and he received BE (CSE) under University of Burdwan in the year of 2005. He received his M.Tech in CSE under National Institute of Technology of Durgapur in the year of 2008. He is a Ph.D. scholar of National Institute of Technology, Durgapur.

Dr.G.K.Mahanti was born in the year 1967 in India. He obtained his Ph.D. (Engg.) from IIT, Kharagpur, India. He has more than 20 years of teaching experience. He is presently a professor in the department of Electronics and Communication Engineering, NIT, Durgapur, India.

He has published 24 research papers in international journals and 21 papers in national and international conferences. He is the reviewer of many international journals like Electronics Letter, IEEE Antennas and Wireless Propagation Letter, Progress in Electromagnetics Research etc. and many conferences. He is also the program committee member of many national conferences. His biography is listed in Marqui’s who is who in the world.

His research area is Array Antenna Synthesis, Evolutionary Algorithms & Electromagnetics.