Design and Implementation of Memory-less Forbidden Transition Free Crosstalk Avoidance CODECs for On-Chip Buses

J.Venkateswara Rao[1] and P.Sudhakara Rao[2]

[1] Electronics & Communication Engineering Department, Vignan Institute of Technology & Science, Deshmukhi(V), Nalgonda Dist., Andhra Pradesh-508284, India

[2] Electronics & Communication Engineering Department, Vignan Institute of Technology & Science, Deshmukhi(V), Nalgonda Dist., Andhra Pradesh-508284, India

Abstract

Recently, reducing crosstalk noise delay is an important issue in VLSI design. As circuit geometries become smaller, wire interconnections become closer together and taller, thus increasing the cross-coupling capacitance between nets. At the same time, parasitic capacitance to the substrate becomes less as interconnections become narrower, and cell delays are reduced as transistors become smaller. In this work, we present a CODEC design for the forbidden transition free crosstalk avoidance CODEC. Our mapping and coding scheme is based on the Binary number system. In this paper, we investigate and propose a bus forbidden transition free CODECs for reducing bus delay and our experimental results show that the proposed CODEC complexity is orders of magnitude better compared to the existing techniques. Compared to the best existing approaches, we achieved a 3 times faster design and improvement in logic complexity.

Keywords: *Crosstalk, Crosstalk Avoidance Codes, Forbidden Transition Free, Encoding, System on Chip, Parasitic, Coupling Capacitance, Deep-submicron.*

1. Introduction

With shrinking device sizes, increasing chip complexity and faster clock speeds, wire delay is becoming increasingly significant [11, 12]. The propagation delay through long cross-chip buses is proving to be a limiting factor in the speed of some designs, and this trend is only expected to get worse. It has been shown that the delay through a long bus is strongly dependent on the coupling capacitance between the wires. In particular, the crosstalk effect when adjacent wires simultaneously transition in opposite directions is particularly detrimental to the delay. When the cross-coupling capacitance is comparable to or exceeds the loading capacitance on the wires, the delay of such a transition may be twice or more than that of a wire transitioning next to a steady signal. This delay penalty is commonly referred to as the capacitive crosstalk delay. The capacitive crosstalk delay strongly depends on the transition activities of the adjacent signals, hence the crosstalk type. Type-4 and type-3 crosstalk have the worst delay characteristics, followed by type-2 and then type-1. A few techniques involving selective skewing of bus data signals [13], transistor sizing [14], and repeater sizing [15] to reduce capacitive crosstalk induced delay have been proposed. Encoding is one of the more effective ways to reduce capacitive crosstalk delays. Here we present encoding techniques that focus on reducing crosstalk delay. The rest of the paper is organized as follows. Section 2 explains about Crosstalk Classification, Section 3 discusses about forbidden transition free crosstalk avoidance codes (FTF-CAC). In Section 4, we discuss about circuit implementation and experimental results. In Section 5 we compare the results of experiments that we have performed to quantify the CODEC performance. We conclude the paper in Section 6.

2. Crosstalk Classification

Figure 1 illustrates a simplified on-chip bus model with crosstalk. In the figure, C_L denotes the load capacitance, which includes the receiver gate capacitance and also the parasitic wire-to-substrate parasitic capacitance. C_I is the inter-wire coupling capacitance between adjacent signal lines of the bus. In practice, this bus structure is typically modelled as a distributed RC network, which includes the non-zero resistance of the wire as well.

The on-chip bus crosstalk is classified into five types [3, 4] as shown in Table 1.

Table 1: Transition pattern crosstalk classification Margin specifications

Crosstalk class	C_{eff}	Sample *transition* patterns
0C	C_L	000→111
1C	$C_L(1+\lambda)$	011→000
2C	$C_L(1+2\lambda)$	010→000
3C	$C_L(1+3\lambda)$	010→100
4C	$C_L(1+4\lambda)$	010→101

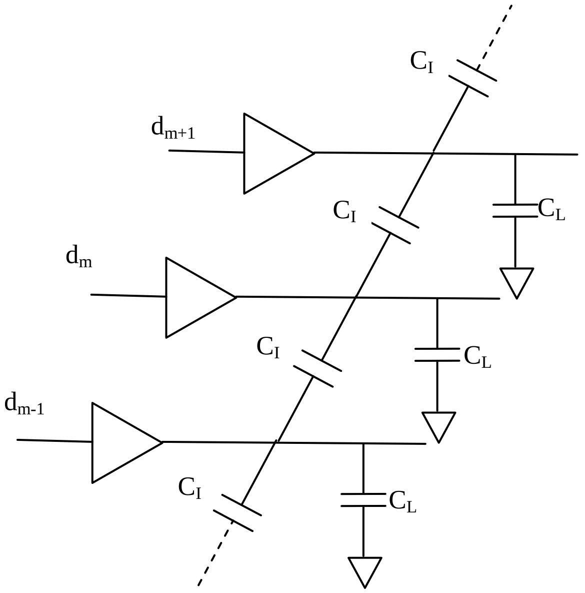

Fig. 1. On-chip bus model with crosstalk

This classification is based on the effective capacitance, in the jth line in a bus as, $C_{eff, j}$

$$C_{eff, j} = C_L [1 + \lambda ((1 - \delta_{j, j-1}) + (1 - \delta_{j, j+1}))] \qquad (1)$$
$$= C_L + C_{lw, j} + C_{rw, j}$$

It separates $C_{eff, j}$ into three components: the intrinsic capacitance C_L, the crosstalk capacitance to the wire on the left side, $C_{lw, j} = \lambda (1-\delta_{j, j-1}) C_L$, and the capacitance to the wire on the right side, $C_{rw, j} = \lambda (1-\delta_{j, j+1}) C_L$. It is easy to see that $C_{lw, j}, C_{rw, j} \in \{0, 1C_I, 2C_I\}$.

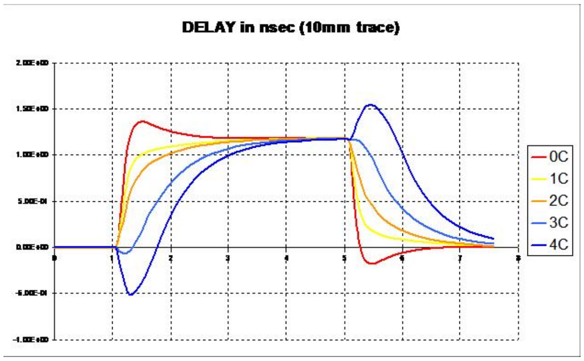

Fig. 2. Delay impact of different sequences confirmed by SPICE simulations-0.1µm CMOS process.

3. Forbidden Transition Free Crosstalk Avoidance Codes

The forbidden transition free Crosstalk Avoidance Codes (FTF-CAC) are an efficient 3*C*-free memory-less codes. It was first proposed by Victor and Keutzer in 2001 [7], and we will use the FTF-CAC to derive an efficient CODEC. The basic idea of the FTF code is to prohibit two adjacent bits from transitioning in opposite directions, i.e., the forbidden transitions 01 → 10 or 10 → 01 are not allowed. This guarantees that $\delta_{j, j+1} \geq 0$, therefore $(2 - \delta_{j, j-1} - \delta_{j, j+1}) \leq 2$ and the bus satisfies $\max_j (C_{eff, j}) = 2$ from equation 1. Hence the transition is 3*C*-free. An FTF code is a set of codewords such that transitions among these codewords do not produce forbidden transitions on any two adjacent bits.

A forbidden transition is defined as the simultaneous transition (in opposite directions) on two adjacent bits, i.e., 01 → 10 or 10 → 01. We first observe that to guarantee forbidden transition freedom on the boundary djdj+1 between any two codewords in an FTF-CAC, the 01 and 10 patterns cannot coexist in the same set of codewords. This can be easily confirmed by examining the transitions among codes in {00, 01, and 11}, or {00, 10, 11}. If we eliminate 01 or 10 from all the boundaries in the codewords in a set of codewords R, we can guarantee that R is forbidden transition free. Therefore, once again, the problem of eliminating forbidden transitions is transformed into a problem of eliminating specific patterns.

Theorem-1 The largest sets of codewords satisfying the forbidden transition free condition is the set of codewords that can transition to a class 1 codeword (defined as a codeword with alternating 0 and 1 bits) without generating forbidden transitions [7].

For any given size bus, there are two class 1 codewords: "1010. . ." and "0101 . . ." From Theorem-1, we can see that there exist two different sets with the same maximum cardinality. In one set (set A), in all codewords, the 01 pattern is eliminated from $d_{2j+1}d_{2j}$ boundaries and the 10 pattern is eliminated from $d_{2j}d_{2j-1}$ boundaries. In the second set (set B), the 10 pattern is eliminated from $d_{2j+1}d_{2j}$ boundaries and the 01 pattern eliminated from $d_{2j}d_{2j-1}$ boundaries. Table-2 lists all set-A FTF codewords for 2, 3, 4 and 5-bit busses. Take the 5-bit bus as an example, we can see that 10 is not present in d_2d_1 and d_4d_3, and 01 is not present in the boundaries d_3d_2 and d_5d_4. If one of these two sets is known, the other set can be produced by simply complementing all the codewords in the first set. There are multiple methods to produce all the n-bit codewords in the set that satisfy Theorem-1.

- Start with a complete set of 2^n vectors and remove codewords that do not satisfy the boundary constraints.

• Start with a set consisting of a single class 1 codeword and grow the FTF-CAC codewords by adding compatible codewords to the set.

• Start from a small FTF set (say 2-bit FTF codes) and inductively append bits to the codewords in the set until the codeword length reaches *n*-bits. Clearly, the first (pruning) method is impractical when *n* is large, since a complete set of codewords have 2*n* entries and searching through n − 1 boundaries requires O (2(n−1) n) searches. Both the second and the third methods listed above actually "grow" the FTF codewords instead of "pruning", and therefore require less computation. The method of "growing" codewords by appending bits to codewords in an existing set is given in Algorith-1.

Algorithm-1 is the pseudo code for generating the FTF codewords.

Table 2: FTF-CAC codewords for 2, 3, 4 and 5-bit busses

2-bit	*3-bits*	*4-bits*	*5-bit*	
00	000	0000	00000	10100
01	001	0001	00001	10101
11	100	0100	00100	10111
	101	0101	00101	11100
	111	0111	00111	11101
		1100	10000	11111
		1101	10001	
		1111		

```
Algorithm-1 FTF codeword generation
$S2 = \{00, 01, 11\}$
for $m > 2$ do
    if $m$ is odd then
        for $\forall V_{m-1} \in S_{m-1}$ do
            add $1 \cdot V_{m-1}$ to $S_m$;
            if $d_{m-1} = 0$ then
                add $0 \cdot V_{m-1}$ to $S_m$;
            end if
        end for
    else
        for $\forall V_{m-1} \in S_{m-1}$ do
            add $0 \cdot V_{m-1}$ to $S_m$;
            if $d_{m-1} = 1$ then
                add $1 \cdot V_{m-1}$ to $S_m$;
            end if
        end for
    end if
end for
```

The inductive codeword generation method given in Algorithm-1 can be used to derive the cardinality of the FTF codes. We first define,

Definition-1

$T_{t\,(m):}$ number of m-bit FTF vector,

$T_{t1\,(m):}$ number of m-bit FTF vector with the MSB being 1,

$T_{t0\,(m):}$ number of m-bit FTF vector with the MSB being 0.

The following relationship can be derived from Algorithm-1:

$$T_{t\,(m)} = T_{t0\,(m)} + T_{t1\,(m)} \qquad (2)$$

$$T_{t\,(2m)} = T_{t1\,(2m-1)} + T_{t\,(2m-1)} \qquad (3)$$

$$T_{t\,(2m-1)} = T_{t0\,(2m-2)} + T_{t\,(2m-2)} \qquad (4)$$

and with some simple manipulation of Eqs. (2), (3) and (4), we get

$$T_{t\,(2m)} = T_{t\,(2m-1)} + T_{t\,(2m-2)}, \qquad (5)$$

$$T_{t\,(2m+1)} = T_{t\,(2m)} + T_{t\,(2m-1).} \qquad (6)$$

and they can be combined into a single recursive equation

$$T_{t\,(m)} = T_{t\,(m-1)} + T_{t\,(m-2)} \qquad (7)$$

Given the initial condition of $T_{t\,(2)} = 3 = f_4$ and $T_{t\,(3)} = 5 = f_5$, we get

$$T_{t\,(m)} = f_{m+2} \qquad (8)$$

Compare Eq. 8 with the maximum cardinality of FPF codes, we find that the FTF codes have slightly lower cardinality because $2f_{m+1} > f_{m+2} = f_{m+1} + f_m$ for all m > 0. However, the asymptotic overhead percentage of the FTF code is still ∼44%, the same as the FPF code cardinality. Therefore for large busses, the coding gain for the FTF-CAC is the same as the coding gain for FPF codes $G_{FTF} \approx 39\%$.

4. Circuit Implementation and Experimental Results

The coded busses in the simulation were 6-bits wide. A 4-to-6-bit encoder and a 6-to-4-bit decoder logic were manually implemented using an arbitrary mapping of data words to codewords. Our simulations using Synopsys Design Compiler show that the maximum delay of both the encoder and the decoder was 8.21 ns.

Table-3: 6 bit FTF code words for 4 bit data words

4 bit data word	***6 bit code word***
$(d_3d_2d_1d_0)$	***$(c_5c_4c_3c_2c_1c_0)$***
0000	000000
0001	000001
0010	000101
0011	010001
0100	110100
0101	110101
0110	011101
0111	011111
1000	110000
1001	000100
1010	010000
1011	110111
1100	110001
1101	111101
1110	111100
1111	111111

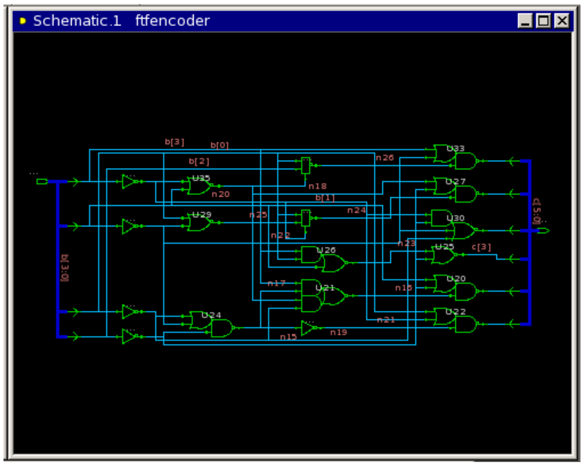

Fig. 3. Gate level schematic for FTF encoder

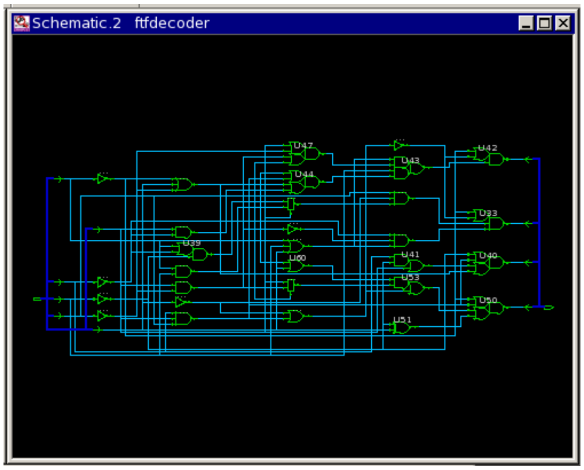

Fig. 4. Gate level schematic for FTF decoder

Table 4: Timing Reports for FTF Encoder

Point	*Incr. Delay*	*Path Delay*
input external delay	0.00	0.00 r
b[3] (in)	0.00	0.00 r
U36/Z (IV)	0.53	0.53 f
U35/Z (NR2)	1.88	2.41 r
U34/Z (MUX21L)	0.65	3.06 f
U33/Z (AO7)	0.66	3.72 r
c[0] (out)	0.00	3.72 r
Data arrival time		3.72

The above timing report shows the data arrival time for the FTF Encoder from input side to output side. Prime Time reports the worst delay path from input to output. The b [3] is the input port of the encoder and the name in the bracket is the reference name for that port. Incr. path is the incremental path delay. The delay from input port, b [3] to the output of not gate, (U36/Z) is 0.53 ns. The delay from the output of not gate, (U36/Z) to the output of next nor gate, (U35/Z) is 1.88 ns. The delay from the output of nor gate, (U35/Z) to the output of next mux, (U34/Z) is 0.65 ns. The delay from the output of mux, (U34/Z) to the output of next AOI gate, (U33/Z) is 0.66 ns. The output of AOI gate, (U33/Z) is the output port of encoder, c(0). So, the total delay from input port, b [3] to output port c [0] is 3.72 ns. The f in the third column indicates a transition from 1 to 0 and r indicates a 0 to 1 transition.

Table 5: Timing Reports for FTF Decoder

Point	*Incr. Delay*	*Path Delay*
input external delay	0.00	0.00 r
c[5] (in)	0.00	0.00 r
U62/Z (IV)	0.47	0.47 f
U46/Z (OR3)	1.44	1.92 f
U44/Z (AO4)	1.32	3.24 r
U43/Z (AO2)	0.60	3.83 f
U42/Z (AO7)	0.66	4.49 r
b[1] (out)	0.00	4.49 r
Data arrival time		4.49

The above timing report shows the data arrival time for the FTF Decoder from input side to output side. So, the total delay from input port, c [5] to output port b [1] is 4.49 ns.

5. Comparison to Other Techniques

This paper is based on the concepts proposed in [1]. The encoder and decoder presented in fig.3 and fig. 4 has a data arrival time of 3.72 ns and 4.49 ns. The encoder and decoder proposed in [1] have a data arrival time of 16.38ns and 7.63 ns. Thus the total delay for the FTF CODEC is 8.21 ns compared to encoder proposed in [1] have a total delay of 24.01 ns. So, we can say that our CODEC is around 3 times faster than the CODEC proposed in [1]. When the data arrival time is small, it creates a more positive slack.

6. Conclusions

The 1C-free bus does not require CODECs, or we can say that these CODEC designs are trivial (repeating the input bit by N times). On the other hand, 3C-free and 2C-free codes need CODECs. For these codes to be used in practice, efficient CODEC designs are necessary. In the case of crosstalk avoidance codes, the complexity and speed of the CODEC are both critical for overall bus performance. Also, the power consumption of the CODECs should be factored in when the overall power consumption is evaluated. In this paper, we present efficient CODEC design technique for the memory less CACs. The advantages of these CODECs are their low complexity, high speed as well as minimum area overhead.

Our CODEC is around 3 times faster than existing efficient CODECs.

Acknowledgments

We grateful to Synopsys Tools providers and the management of Vignan Institute of Technology & Science for providing the Back end Tools for carrying out the research.

References

[1] Duane .C, Cordero. V and Khatri. S. P. ”Efficient On-Chip Crosstalk Avoidance CODEC Design”, IEEE Transactions on VLSI Systems, April 2009, pp 551 – 560.

[2] Sotiriadis .P and Chandrakasan .A, “Low power bus coding techniques considering inter-wire capacitance”. Proc. of IEEE-CICC, 2000, pp 507-510.

[3] Chem. J, Huang .J, Aldredge .L, Li .P, and Huang .P, “Multilevel metal capacitance models for CAD symbol design synthesis systems”. In IEEE Electron Device Letter, 1992, pp 32–34.

[4] Arora .N, Raol .K, Shcumann .R, and Richardson .L, “Modeling and extraction of interconnect capacitance for multilayer VLSI circuits”. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1996, 15(1) pp 58–67.

[5] Sridhar .S.R, Ahmed .A, and Shanbhag .N. R.,”Area and Energy-Efficient Crosstalk Avoidance Codes for On-Chip busses”, Proc. of ICCD, 2004, pp 12-17.

[6] Duan .C, Tirumala .A and Khatri .S.P.,”Analysis and Avoidance of Cross-talk in On-chip Bus”, Hot Interconnects, 2001, pp 133-138.

[7] Bret Victor and Keutzer .K, ”Bus Encoding to Prevent Crosstalk Delay”, ICCAD, 2001, pp 57-63.

[8] Duan .C and Khatri S. P. ”Exploiting Crosstalk to Speed up On-chip busses”, Design, Automation and Test in Europe Conference and Exhibition, 2004, pp 778-783.

[9] Duan .C, Gulati .K and Khatri .S. P,”Memory-based Cross-talk Canceling CODECs for On-chip busses”, ISCAS, 2006, pp 4-9.

[10] Ma .J and He .L,”Formulae and applications of interconnect estimation considering shield insertion and net ordering”, ICCAD, 2001, pp 327-332.

[11 Davis J. A, al. et, “Interconnect limits on giga-scale integration (GSI) in the 21st century” in *Proceedings of the IEEE*, 2001, 89, pp. 305–324.

[12] Bohr .M. T, “Interconnect scaling—The real limiter to high performance ULSI” in *Proceedings of IEEE Electron Devices Meeting*, 1995, pp. 241–244.

[13] Hirose .K and Yasuura .H, “A bus delay reduction technique considering crosstalk” in *Proceedings of Design, Automation and Test in Europe*, 2000, pp. 441–445.

[14] Xiao .T and Sadowska .M, “Crosstalk reduction by transistor sizing” in Proceedings of Asia and South Pacific Design Automation Conference (ASPDAC), 1999, pp. 137–140.

[15] Li .D, Pua .A, Srivastava .P and Ko .U, “A repeater optimization methodology for deep submicron, high-performance processors” in IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD), Austin, TX, 1997, pp. 726–731.

First Author J.Venkateswara Rao received B.Tech Degree from, VRSEC, Nagarjuna University, Guntur, in 2000. M.Tech. Degree from NITW, Warangal, India in 2003, pursuing Ph.D in the Department of Electronics and Communication Engineering, JNT University, Hyderabad, INDIA. Currently he is working as Associate Professor in the Department of Electronics and communication Engineering, VITS, Hyderabad, INDIA, His research interest includes on chip crosstalk noise reduction in VLSI Circuits**.** He published 4 papers on “on-chip crosstalk noise reduction in VLSI circuits” in international journals and international conferences.

Second Author Dr.P.Sudhakara Rao Completed **Ph.D.,** Information and Communication Engineering from Anna University, India, **Masters** in Electronics and Communication Engineering from Anna University, India, **Bachelors** in Electronics and Communication Engineering from Mysore University, India. Worked as deputy **Director, “Central Electronics Engineering Research Institute centre, India”** for over 25 years**, 2 years** as **Vice-president, “Sieger Spintech Equipments Ltd., India”,** established an electronic department for the development of electronic systems for nearly 2 years**.** Presently working with Vignan Institute of Technology and Science, Nalgonda District, AP, INDIA as DEAN R&D, HOD ECE. He has one patent and 55 technical publications/ conference papers to his credit. Conducted many international and national conferences as chairman.