Redundancy Efficient Crosstalk Avoidance Scheme in VLSI Circuits

Dr. K. Padmapriya, D. Gopinadh
Associate professor, ECE dept. Jntuk UCE, Vizianagaram, India
B.TECH final year, ECE dept, Jntuk UCE, Vizianagaram, India

Abstract— In deep sub micrometer design interconnect delay has become a major factor. Crosstalk is highly evident in deep sub micrometer design. Crosstalk depends on different data patterns that are transmitted on the bus. For the avoidance of crosstalk many schemes have been proposed which boost the bus speed or reduce energy consumption. High proportion of the proposed schemes are non linear in nature and are impractical. A crosstalk avoidance code which can be implemented practically is dealt with which is based on the representation of numbers in Fibonacci numeral system. An improved version of this scheme is later presented where data is represented in Fibonacci numeral system only if forbidden patterns are present in the data.

Index Terms—Codec, crosstalk, Fibonacci number, on-chip bus, deep sub micrometer.

I. INTRODUCTION

Once the design of VLSI technology has moved on to the regime of deep sub micrometer design (DSM) interconnect delay has become a major hurdle. Crosstalk has become the major factor of the total power consumption and the time delay. Bus encoding can be used to reduce crosstalk.[1] Interconnect delay has dominated the logic delays. A simplified on chip bus model with crosstalk is considered which is shown in fig 1.

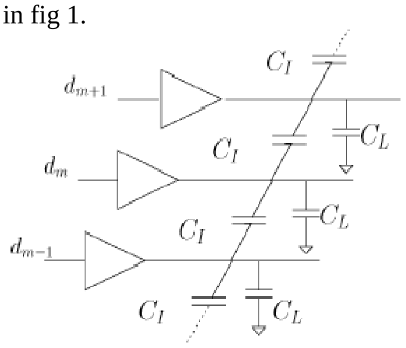

Fig 1. On-chip bus with crosstalk

Where C_L denotes the load capacitance which is seen by the driver 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 lines of the bus. Reducing the crosstalk boosts the speed of the bus significantly.[2]

Delay in a bus of the jth wire is given by equation (1)

T_j=abs $(k.C_L.\Delta V_J+k.C_I.\Delta V_{j,j-1}+k.CI.\Delta V_{j,j+1})$ ------ (1) [2]

K is the constant determined by the driver strength and wire resistance. ΔV_j is the voltage change on the j^{th} line and $\Delta V_{j,k}$ is the relative voltage between the j^{th} and the k^{th} line.

II. CROSSTALK CLASSIFICATION [3]

Crosstalk in an on chip bus is dependent on data transitions. Based on the bus model with crosstalk which is shown in fig 1 delay in the j^{th} wire of a bus is given in equation (1).

Let $C_{eff,j}$ be the effective capacitance of the j^{th} wire. Effective capacitance of the wire is given in equation (2)

$C_{eff,j}= C_L.abs(\delta_j+\lambda.\ \Delta_{j,j+1}+\ \lambda.\ \Delta_{j,j-1})$ ----- (2)

Where δ_j is the normalized voltage change on the j_{th} line and

$$\lambda = C_I/C_L.$$

Based on the effective capacitance of the wire crosstalk can be classified into four different classes as shown in table. The value of the effective capacitance ranges from C_L to C_L $(1+4\lambda)$ depending on the data transitions on the wire and also on the transitions on its neighbors.

CLASS	C_{eff}	TRANSITION PATTERNS
0C	C_L	000 to 111
1C	$C_L(1+\lambda)$	011 to 000
2C	$C_L(1+2\lambda)$	010 to 000
3C	$C_L(1+3\lambda)$	010 to 100
4C	$C_L(1+4\lambda)$	010 to 101

Table1: crosstalk classification

Different types of encoding schemes to reduce the crosstalk have been dealt with in [4]-[7]

III. FORBIDDEN PATTERNS [3,8]

Forbidden patterns are defined as 3 bit patterns "101" and"010". Any data or a code that represents the data is said to be forbidden pattern free (FPF) if there are no forbidden patterns in any 3 consecutive bits. Data transition patterns that belong to forbidden patterns are of 4C classification.

For example consider the data pattern **101**1 this pattern consists the forbidden pattern. Data pattern 011110 which is used to represent the same data using another numeral system does not consist of any forbidden patterns. For any bus that transmits only FPF patterns, the crosstalk experienced will be no more than 2C [3]. By encoding the data words to FPF patterns the bus can be speeded up by approximately 100%.[3]

For any m-bit vector $b_m b_{m-1}........b_2 b_1$ we define the following quantities:
T(m) is the total number of distinct m-bit vectors.
$T_g(m)$ is the total number of FPF vectors
$T_b(m)$ is the total number of non FPF vectors
For buses with more than three bits we have
$T_g(m)=T_g(m-1)+T_g(m-2)$. ------ (3) [8]

The relationship shown in equation (3) is same as the relationship between the elements in a Fibonacci sequence. Hence data patterns with FPF codes can be represented by a Fibonacci numeral system.
To encode an n-bit binary sequence into a FPF code, the minimum number of bits required is given by the smallest integer that satisfies the equation [8]
$n<= [\log_2(2.f_{m+1})]$ ------ (4)
.

IV. FIBONACCI BASED NUMERAL SYSTEM [1]

A numeral system is a "frame work where numbers are represented by numerals in a consistent manner"[9]. The most commonly used numeral system in all the digital design is binary numeral system which has a base2. A binary numeral system is a one to one mapping system where one number has one and only one representation. The binary numeral system is said to be complete and unambiguous. Any number can be represented in a binary numeral system as shown in (5).
$V= \sum^{n}_{k=1} b_k .2^{k-1}$ ------ (5)
Fibonacci numeral system was first used in CAC designs in [1]. Fibonacci numeral system is the numeral system that uses the Fibonacci sequence as the base. Any number can be represented as the summation of the Fibonacci numbers. A Fibonacci sequence can be generated by the equation (6).

$$f_m = \left.\begin{matrix} O \\ 1 \\ f_{m-1}+f_{m-2} \end{matrix}\right\} \quad \text{-------} \quad (6)$$

A Fibonacci numeral system is also complete but this system is ambiguous. A Fibonacci numeral system is a one to many mapping system where one number can be represented in many ways. All the Fibonacci vectors that represent the same value are defined as equivalent vectors. Any number can be represented in the Fibonacci based numeral system and the representation is given by equation (7)

$$V=\sum_{k=1}^{m} b_k.f_k \quad \text{------} \ (7)$$

Any data pattern that is not FPF can be converted into FPF by performing some or all of the operations mentioned in [9]

- If the vector ends with a forbidden pattern (101 or 010) there exists an equivalent vector that ends with 110 or 001.
- If any part in the data contains the forbidden pattern they can be replaced by a pattern that is forbidden pattern free using two corollaries that are defined for a m-bit Fibonacci vector $d_m,d_{m-1},....,d_2,d_1$.
- If a number of consecutive bits consist a pattern of 0101,…..0100(alternating 0 and 1 except last two bits) replacing it with the pattern 0011,….,1111 (all bits are one except the first two bits) produces an equivalent vector that is FPF.
- If a number of consecutive bits consist a pattern of 1010,…..1011(alternating 1 and 0 except last two bits) replacing it with the pattern 1100,….,0000 (first two bits are 1's and the other bits are 0's) produces an equivalent vector that is FPF.
- These operations implemented on data effectively convert the data into forbidden pattern free data because of the presence of equivalent vectors in Fibonacci numeral system.

V. PROPOSED SCHEME

Encoding the data based on the Fibonacci numeral system is highly efficient as it reduces forbidden patterns completely, however the number of bits used for the representation of the number based on Fibonacci numeral system is high. Encoding all the data irrespective of the presence of forbidden patterns leads to some unwanted crosstalk and also this method requires more number of bits. Hence it is advisable to encode the data which contains forbidden patterns to Fibonacci numeral system and send the other data in the usual manner. This method would be more efficient because forbidden patterns which are generally known as worst case patterns constitute a very small amount of the usual data. Consider three bit data, only two worst case scenarios are present in a total of eight data transitions. So, encoding the entire data would result in increment of the number of bits to send the data. A scheme that would encode the data into Fibonacci numeral system only if the data contains forbidden patterns or in the other case data would be sent in the normal binary system which would require less number of bits than in the Fibonacci numeral system.

Algorithm:

Let v be the number and $b_n b_{n-1} .....b_2 b_1$ be the binary representation of the number and $d_m d_{m-1}......d_2 d_1$ be the representation of the number in Fibonacci numeral system.
Based on the algorithm the given data stream can be converted into the proposed scheme of encoding.

ENCODER:

```
\\ check for forbidden patterns
For i=1 to n-2 do
 if((xor($b_i$,$b_{i+1}$)&xor($b_{i+1}$,$b_{i+2}$)==1  then
\\ if forbidden patterns are present generate code in
Fibonacci numeral system.
\\ indicate Fibonacci numeral system is used
z=1
\\  msb stage
if $v>=f_{m+1}$ then
    $d_m=1$
    $r_m=v-f_m$
else
    $d_m=0$
    $r_m=v$
end if
\\ other stages
For k= m-1 to 2 do
    If $r_{k+1} >= f_{k+1}$ then
        $d_k=1$
    else if $r_{k+1}<f_k$ then
        $d_k=0$
    else
        $d_k=d_{k+1}$
    end if
    $r_k=r_{k+1}-f_k.d_k$
end for
\\ lsb
$d_1 = r_2$
return $(d_m d_{m-1}..d_2 d_1)$
else
\\ indicate binary numeral system is used
z=0
return $(b_n b_{n-1}\ldots b_2 b_1)$
```

```
DECODER:
\\ check whether received sequence is based on
Fibonacci numeral system or binary numeral system.
If z=1 then
    v=0
    For i=m to 1 do
        $v=v+d_m.f_m$
    return (v)
else
    $v=v+b_n.2.^(n-1)$
    return (v)
```

In the given data the presence of any forbidden patterns is checked for. If any forbidden patterns are present then the encoder encodes the data based on Fibonacci numeral system. Presence of forbidden patterns is confirmed by the use of xor operation. If the xor operation yields one then the presence of forbidden patterns is confirmed. If there is no forbidden pattern present in the data then the data is encoded based on binary numeral system. An extra bit is used to indicate to the decoder whether data is encoded based on Fibonacci numeral system or binary system. Based on the info in the wire data is decoded.

The number of bits to encode the data in the Fibonacci numeral system and the proposed scheme is tabulated and is shown in table 2

Number of bits used in Fibonacci representation	Number of bits to represent FPF free data	Redundancy
4	3	1
5	4	1
7	5	2
8	6	2
10	7	3
11	8	3
13	9	4
14	10	4
16	11	5

Table2. Number of bits used for representing data in both Fibonacci and binary numeral system.

From the above table we can observe that the number of bits to represent the data in binary system when there are no forbidden patterns in the data stream is significantly less than the number of bits in the Fibonacci numeral system representation. Hence this scheme of encoding would be redundancy efficient and also would eliminate the 4C class of crosstalk.

VI. CONCLUSION AND FUTURE WORK

Crosstalk avoidance codes effectively reduce the inter wire crosstalk and hence boost the speed on the data bus. The forbidden pattern free crosstalk avoidance code proposed based on Fibonacci numeral system is very effective in reducing the crosstalk. A scheme such that data that contain forbidden patterns are encoded based on Fibonacci numeral system and other data is sent in usual manner. This would result in the reduction of power consumption and the number of bits required in sending the data which is tabulated in table2. Hence this would be a more efficient scheme of coding. This type of encoding deals with only forbidden patterns. Future work on this subject would involve a better encoding scheme that would involve gray code as the scheme for encoding data.

REFERENCES

[1] M.MUTYAM," preventing crosstalk delay using Fibonacci representation," VLSI des., 2004,pp. 685-688

[2] P.SOTIRIADIS and A.CHANDRAKASAN, "Low power bus encoding techniques considering inter wire capacitance," in proc. IEEE-CICC, 2000, PP507-510.

[3] C.DUAN, A.TIRUMALA, S. P. KHATRI, "analysis and avoidance of crosstalk in on-chip bus," in proc. ICCD, 2004, PP. 12-17

[4]B.VICTOR and K.KEUTZER, "bus encoding to prevent crosstalk delay," in proc. ICCAD,2001, PP. 57-63

[5]S.P. KHATRI, "crosstalk noise immune VLSI design using regular layout fabrics," Ph.d. dissertation. Dept. Elect. Eng. Comput. Sci. univ. California Berkeley, Berkeley, 1999

[6] C.DUAN and S.P. KHATRI, "exploiting crosstalk to spped up on chip busses "in proc. Int. conf. Autom. Test Eur., 2004,pp. 778-783.

[7] C.DUAN, K. Gulati, and S.P. Khatri, "Memory-Based cross-talk cancelling CODECs for on-chip busses", in Proc. ISCAS, 2006, pp.1119-1123.

[8] CHUNJIE DUAN, VICTOR H.CORDERO CALLE, SUNIL P. KHATRI, “efficient on-chip crosstalk avoidance code,” in IEEE TRANSACTIONS ON VLSI VOL. 17, NO.4, APRIL 2009.

[9] WIKIPEDIA, “NUMERAL SYSTEM”.