Crosstalk Delay Avoidance in Long on Chip Buses by using Different Fibonacci Codec Techniques

Savitha A.C.
Asst. Professor
E&C Department
JSSATE

Siddesh.G.K, PhD
Associate Professor
E&C Department
JSSATE

ABSTRACT

In this work, a CODEC design to eliminate/reduce the propagation delay across long on chip buses which are increasingly becoming a limiting factor in high-speed design has been proposed. Crosstalk between adjacent wires are transitioning in opposite direction create a significant portion of this delay. The coding scheme is based on the Fibonacci numeral system. The proposed CODEC design is efficient and a modular technique. Encoding and decoding algorithms are proposed for three different Fibonacci techniques. The experimental results show that the proposed CODEC reduces crosstalk delay when compared to that of existing approaches. The implementation has been done in verilog code. These codes were synthesized and verified using Cadence Encounter RTL compiler tool with geometries at 180nm.

Keywords

Crosstalk, Fibonacci, CODEC, On Chip Bus

1. INTRODUCTION

As the CMOS technology scaled down to deep submicron level, the crosstalk effects due to the coupling capacitance between interconnection lines has become one of the main performance limiting factors. The condition for crosstalk delay is that the signal at both lines switches to the opposite direction. The result is an increase in transition time. Since the main focus of this work is on eliminating delay caused by crosstalk interactions. The main idea is to prevent simultaneous opposite transitions by skewing signal transition timing of adjacent wires. In some high-speed design, the technique of shielding was used, which involves putting a grounded wire between every signal wire on the bus. But it has the effect of doubling the wiring area [1]. It requires 63 wires for 32- bit bus. Bus encoding scheme can achieve the same amount of bus delay improvement as passive shielding, with a much lower area overhead. Forbidden pattern coding (FPC) technique [3] prohibits 010 and 101 pattern from code words, which in turn eliminates crosstalk transitions. It requires 52 wires for 32- bit bus. The proposed technique is proved theoretically far better than just placing shielding wire between every adjacent wire. They showed that a 32- bit can be encoded with 46wires.

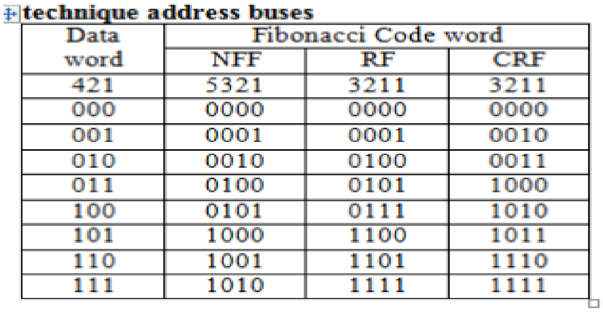
In this work, the focus is on CODEC design technique using a variant of Fibonacci representation and gives a recursive procedure to generate crosstalk delay free binary Fibonacci code words. This technique introduces an elegant and efficient mapping between data words and code words. The paper is structured as follows: Section 3 gives the proposed different encoding and decoding algorithm. In Section 4 deals with the results of experiments. Section 5, The comparison of timing report of proposed technique with those reported in [9], and the conclusion is provided in Section 6.

2. RELATED WORK

In [2], Victor et al. have given two types of self shielding codes, i.e., codes with memory and memory less codes. Since the proposed technique is closely related to memory less codes. Let symbol be the data word to be encoded and codeword be theencoded data word. The assumption made is that data words are represented in the binary number system. For every data word, code words are given using Fibonacci number system {1, 1, 2, 3, 5, 8…}. The maximum number of information bits for a m-line channel is $[\log_2(F_{m+2})]$ from paper[6].

3. PROPOSED CODEC

Table 1. Codes for different Fibonacci technique address buses



Data word	Fibonacci Code word		
	NFF	RF	CRF
421	5321	3211	3211
000	0000	0000	0000
001	0001	0001	0010
010	0010	0100	0011
011	0100	0101	1000
100	0101	0111	1010
101	1000	1100	1011
110	1001	1101	1110
111	1010	1111	1111

The above Table 1 shows the codes proposed for binary number system. All the three bit codes are encoded to four bit code words which eliminates the crosstalk

3.1 Normal-Form Fibonacci (NFF) Coding Technique

In Normal-Form Fibonacci coding technique , transition signaling (TS), which eliminates crosstalk transitions is used. Hence NFF code words are transmitted using TS technique [5], where data to be transmitted is XORed with the data present on the bus. Transmitting NFF code words using the TS technique eliminates crosstalk transitions which are shown in the table 2. Below table gives an example of crosstalk transitions (refer to the first column of Table 2).Transmitting NFF code words as it is results in crosstalk transitions as shown in the second column of Table 2. As shown in the third column of Table 2, transmitting NFF code words using the TS technique eliminates crosstalk transitions.Table 3 shows the NFF encoding algorithm in which d indicates data word, m is the code word length and fk is the Fibonacci series. The length of the code word m gives wires required to encode the data bits. For example 3bit data can be encode using 4 wires, similarly 8 bit data can be encode using 10 wires, 16 bit data can be encode using 20 wires, 32 bit data can be encode using 46 wires.

Table 2

Data word	Fibonacci code word			
	NFF	NFF TS	RF	CRF
010	0010	0010	0100	0011
101	1000	1010	1100	1011
011	0100	1110	0101	1000
110	1001	0111	1101	1110
111	1010	1101	1111	1111
000	0000	1101	0000	0000
001	0001	1100	0001	0010
010	0010	1110	0100	0011

Table 3. NFF encoding algorithm

```
Input d;
      $r_m$=d;
      for k=m-1 to do
            if $r_{k+1}$<$f_k$ then
                  $c_k$=0;
            else
                  $c_k$=1;
            end if
            $r_k$=$r_{k+1}$-$f_k$.$c_k$;
      end for
      $c_0$=$r_1$;
   Output: $c_{m-1}$ $c_{m-2}$....$c_0$;
```

3.2 Redundant Fibonacci (RF) Coding Technique

In the case of NFF technique, Fibonacci numbers fm-1….f0 are considered as the basis elements to generate m-bit code words. Similar to the NFF technique, in redundant Fibonacci (RF) coding technique, Fibonacci numbers are considered as the basis elements with the exception that f0 is used twice. That is, in order to generate m-bit RF code words, fm-2…f0, f0 are considered as the basis elements. As f0 Table 4. RF encoding algorithm is considered twice in the RF technique, and obtain two sets of RF code words; each is a complement of the other. The twosets ,redundant Fibonacci (RF) and complement redundant Fibonacci (CRF) codeword sets are considered . Table 4 and Table 5 shows the encoding algorithm of RF and CRF.

Table 4. RF encoding algorithm

```
Input d;
      $r_m$=d;
      for k=m-1 to do
            if $r_{k+1}$<$f_{2[(k+1)/2]}$ then
                  $c_k$=0;
            else
                  $c_k$=1;
            end if
            $r_k$=$r_{k+1}$-$f_k$.$c_k$;
      end for
      $c_0$=$r_1$;
Output: $c_{m-1}$ $c_{m-2}$....$c_0$;
```

3.3 Complement Redundant Fibonacci(CRF) Coding Technique

Table 5 . CRF encoding algorithm

```
Input d;
      $r_m$=d;
      for k=m-1 to do
            if $r_{k+1}$<$f_{2[(k/2)+1]}$then
                  $c_k$=0;
            else
                  $c_k$=1;
            end if
            $r_k$=$r_{k+1}$-$f_k$.$c_k$;
      end for
      $c_0$=$r_1$;
   Output: $c_{m-1}$ $c_{m-2}$....$c_0$;
```

The encoder stages implement the arithmetic function given in Equation (1). Each stage produces a single bit c_k and a remainder r_k. For the MSB stage, the input $r_{k+1}$is the input datato be encoded. For other stages, $r_{k+1}$is the remainder of the preceding stage.The encoding is carried out sequentially in m-1 stages, similar to a division operation. Starting from the MSB, each stage compares the input to a Fibonacci number and produces a coded bit as well as a remainder. The remainder from the one stage becomes the input to the next stage. $r_k = r_{k+1}-f_k.c_k$ (1)

3.4 Decoding algorithm for Fibonacci technique

Table 6

```
Input c;
      for k=0 to m
            if ($d_{(k)}$==1)
                  d=d+$f_{(k)}$
            end for
Output: $d_{m-1}$ $d_{m-2}$....$d_0$;
```

4. EXPERIMENTAL RESULTS

Table. 7: Timing Reports for Encoder

NFF			RF			CRF		
Point	Delay (ps)	Arrival (ps)	Point	Delay (ps)	Arrival (ps)	Point	Delay (ps)	Arrival (ps)
d[0]	+0	0 F	d[0]	+0	0 F	d[0]	+0	0 F
g193/Y	+37	37 R	g189/Y	+151	151 F	c reg[1]/SI	+0	0
g189/B1	+0	37	g183/BO	+0	151	c reg[1]/CK	+343	343 R
g189/Y	+88	125 F	g183/Y	+101	252 R			
c reg[1]/D	+0	125	c reg[1]/D	+0	252			
c reg[1]/CK	+181	305 R	c reg[1]/CK	+80	332 R			
Data arrival time		305	Data arrival time		332	Data arrival time		343

Table 7 and Table 8 reports the worst-case delay among the bus signals in picoseconds. The results were generated using Cadence tool, with circuit geometries at 180nm. The above timing report shows the data arrival time for the encoder of NFF, RF and CRF technique from input side to output side. The d [0] is the input port of the encoder and the name in the bracket is the reference name for that port. So, the total delay from input port, d[0] to output port is c_reg[1]/CK is 305ps, 332ps and 343ps. The F in the third column of each technique indicates a transition from 0 to1 and R indicates a 1 to 0 transition.

Table 8. Timing Reports for Decoder

NFF			RF		
Point	Delay (ps)	Arrival (ps)	Point	Delay (ps)	Arrival (ps)
c[1]	+0	0 R	c[1]	+0	0 R
g337/Y	+42	42 F	g196/Y	+113	113 F
g336/A	+0	42	g195/AN	+0	113
g336/Y	+148	190 F	g195/Y	+184	297 F
g329/A	+0	190	g191/B	+0	297
g329/Y	+58	247 R	g191/Y	+120	417 F
d reg[1]/SE	+0	247	d reg[1]/D	+0	417
d reg[1]/CK	+352	599 R	d reg[1]/CK	+88	505 R
Data arrival time		599	Data arrival time		505

The above timing report shows the data arrival time for the decoder of NFF and RF technique from input side to output side. Prime Time reports the worst delay path from input to output. c[1] is the input port of the decoder and the name in the bracket is the reference name for that port. So the total delay from input port, c[1] to output port d_reg[1]/CK is 599ps, 505ps. The F in the third column indicates a transition from 0 to 1 and R indicates a 1 to 0 transition. Table 9 and Table 10 shows the CODEC overhead summary of encoding and decoding of theses Fibonacci technique.

CODEC overhead summary of encoding

Technique	Cell Area	Power(nw)	Effective Delay(ps)
NFF	81	2160.970	305
NFF-TS	76	1808.252	368
RF	79	1471.216	332
CRF	75	1747.919	343

CODEC overhead summary of decoding

Technique	Cell Area	Power(nw)	Effective Delay(ps)
NFF	95	2233.118	599
RF-CRF	77	1796.993	505

4.1 Comparison to other technique

The encoder presented in Fibonacci technique that takes the code words of the proposed technique codes has a data arrival time of NFF, RF and CRF is 0.305ns, 0.332ns and 0.343ns respectively when compared to encoder proposed in Crosstalk-immune coding [9] has a data arrival time of 1.57ns and also compared to encoder proposed in BruteKeutzer codes [2] has a data arrival time of 1.75ns. Similarly the decoder takes data arrival time of NFF, RF and CRF is 0.599ns and 0.505ns respectively when compared to decoder proposed in Crosstalk-immune coding [7] has a data arrival time of 1.72ns. When the data arrival time is small, that creates a more positive slack. Figure1 and Figure2 show the delay comparison of different encoding and decoding technique. It shows the data arrival time for the encoder to convert 3 bit data words into 4 bit code words. Similarly for the data arrival time for the decoder to convert 4 bit code words into 3 bit data words. Based on these figures, it can be seen that the Fibonacci technique would reduce the worst case delay. This results in significant savings for longer buses.

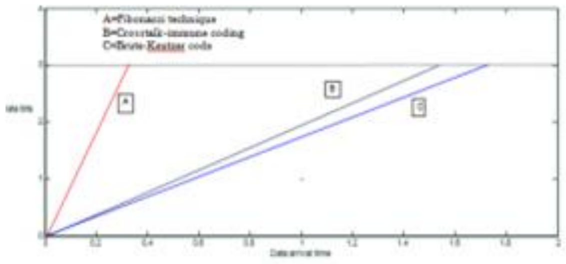


Figure 1: Delay comparison of encoding (ns)

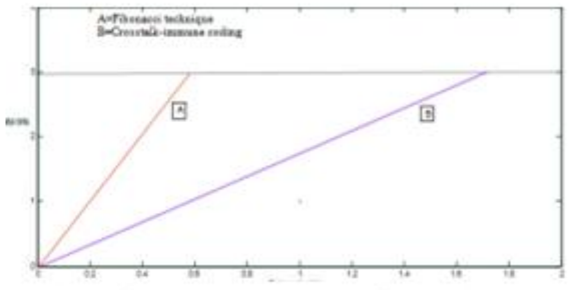


Figure 2: Delay comparison of decoding (ns)

From the analytical study authors also has compared the wiring overhead ratio for 32 bit data to encode. Hence, a 32-bit bus can be implemented with 63 wires in Shielding, 52 wires in Forbidden Pattern Coding technique, and 46 wires in the Fibonacci technique. With shielding, wiring overhead is 97%, whereas with this coding it is only 44% [6]. Figure 3 shows the comparison of different encoding technique to encode 32 bit data.

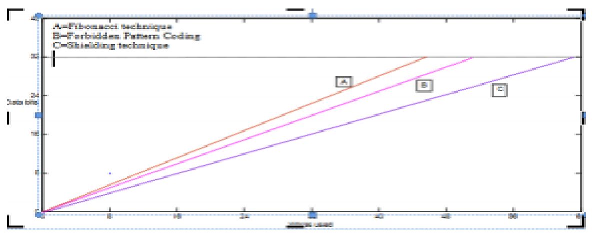


Figure 3 : Wires required to encode

5. CONCLUSIONS AND FUTURE WORK

The proposed encoder and decoder designs have the following features:Both the encoder and decoder have less arithmetic operations, Results in further complexity reduction in implementation. Bus partitioning becomes trivial and less overhead compared to the FPF-CAC CODECs.Both the encoder and decoder are constructed in a systematic Fashion. The encoder consists of multiple stages and a CODEC design for a larger bus can be extended from a CODEC of a smaller bus.This paper presents a memory less transition bus encoding technique for elimination of cross talk. An analytical study of the performance of Fibonacci code is also presented. As mentioned in [8], the crosstalk effect is maximum only when adjacent wires are transitioning in opposite direction. Implementation of encoder and decoder, the overall delays on the bus are reduced for longer buses. Future work will include designing codes such that the coding circuitry can be implemented efficiently in the case of inductance where crosstalk occurs when adjacent lines transition in the same direction.

6. REFERENCES

[1] S.P. Khatri “Cross-talk Noise Immune VLSI Design using Regular Layout Fabrics”, PhD thesis, University of California at Berkeley, Berkeley, California, 1999.

[2] S.P. Khatri, A. Mehrotra, R.K. Brayton, Ralf H.J.M. Otten, and A.L. Sangiovanni-Vincentelli. “A novel vlsi layout fabric for deep sub-micron applications”, In Proceedings of Design Automation Conference, pages 491–496. IEEE, 1999.

[3] F. Caignet, S. Delmas-Bendhia, and E. Sicard, “The challenge of signal integrity in deep-submicrometer CMOS technology,” Proc. IEEE, vol. 89, no. 4, pp. 556–573, Apr. 2001.

[4] D. Pamunuwa, L.-R. Zheng, and H. Tenhunen, “Maximizing throughput over parallel wire structures in the deep submicrometer regime,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 2, pp. 224–243, Apr. 2003.

[5] R. Arunachalam, E. Acar, and S. Nassif, “Optimal shielding/spacing metrics for low power design,” in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, 2003, pp. 167–172.

[6] B. Victor and K. Keutzer, “Bus encoding to prevent crosstalk delay,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, 2001, pp. 57–63.

[7] C. Duan, A. Tirumala, and S. Khatri, “Analysis and avoidance of crosstalk in on-chip buses,” in Proc. Hot Interconnects, 2001, pp. 133–138.

[8] P. Subramanya, R. Manimeghalai, V. Kamakoti, and M. Mutyam, “A bus encoding technique for power and cross-talk minimization,” in Proc. IEEE Int. Conf. VLSI Design, 2004, pp. 443–448.