FPGA Implementation of Codec Design for Optimal Code Rate Crosstalk Avoidance Codes

K. Ramesh[1], E. Srinivas[2]

[1]M. Tech Student
CVSR College of Engineering,
Hyderabad, India
rameshkrs1008@gmail.com

[2]M. Tech, (PhD)
Department of ECE
CVSR Engineering College,
Hyderabad, India
edem.srinivas@gmail.com

Abstract: ***In deep submicron (DSM) technology, the coupling capacitance is comparable to or exceeds the self or substrate capacitance, which in turn causes the delay of a transition in a wire to be twice or more than that of a wire transitioning next to a steady signal. In this paper, the authors propose a new coding technique which minimizes both coupling and self transition activities in the bus lines using the CODEC design of all classes of CACs based on binary mixed-radix numeral systems and spatial redundancy respectively. Using this framework, we then propose novel CODEC designs for three important classes of CACs; one lambda codes (OLCs), FPCs, and forbidden overlapping codes (FOCs). Our CODEC designs have area complexity and delay that increase quadratically with the size of the bus, while achieving optimal or nearly optimal code rates. Using an FPGA kit we can observe the CACs results on it.***

Keywords: CODEC, crosstalk avoidance codes (CACs), interconnect, numeral systems, FPGA kit

1. Introduction

The advancement of very large scale integration (VLSI) technologies has been following Moore's law for the past several decades: the number of transistors on an integrated circuit is doubling every two years and the channel length is scaling at the rate of 0.7/3 years. It was not long ago when VLSI design marched into the realm of Deep Submicron (DSM) processes, where the minimum feature size is well below 1µm. These advanced processes enable designers to implement faster, bigger and more complex designs. With the increase in complexity, System on Chip (SoC), Network on Chip (NoC) and Chip-level Multiprocessing (CMP) based products are now readily available commercially. In the meanwhile, however, DSM technologies also present new challenges to designers on many different fronts such as (i) scale and complexity of design, verification and test (ii) circuit modelling and (iii) processing and manufacturability.

As VLSI technology has marched into the deep sub-micrometer (DSM) regime, new challenges are presented to circuit designers. As one of the key challenges, the performance of bus based interconnects has become a bottleneck to the overall system performance. In large designs [e.g., systems-on chip (SoCs)] where long and wide global busses are used, interconnect delays often dominate logic delays. Once negligible, crosstalk has become a major determinant of the total power consumption and delay of on-chip busses. The impact of crosstalk in on-chip busses has been studied as part of the effort to improve the power and speed characteristics of the on-chip bus interconnects. Fig. 1 illustrates a simplified on-chip bus model with crosstalk. Denotes the CL load capacitance seen by the driver, 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, In practice, this bus structure is electrically modelled using a distributed resistance-capacitance (RC) network, after including the parasitic resistance of the wire as well (not shown in Fig. 1). For DSM processes, is much greater than [7]. Based on the energy consumption and delay models given in [1], the energy consumption is a function of the total crosstalk over the entire bus. The delay, which determines the maximum speed of the bus, is limited by the maximum crosstalk that any wire in the bus incurs. It has been shown that reducing the crosstalk can boost the bus performance significantly [1], [5].

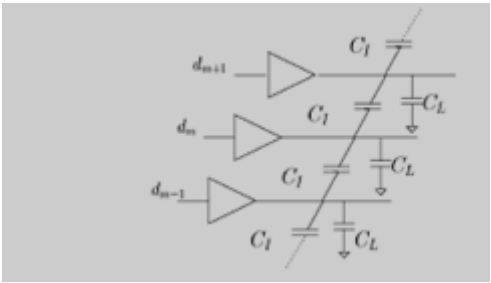

Figure 1: On Chip Bus model with Crosstalk

Since the crosstalk delay is the major part of the delay, different solutions have been proposed to reduce it, e.g. skewing the timing of signals on the bus [2], bus interleaving, pre charging, or using repeaters. These solutions have varying degrees of success. Unfortunately, these solutions are often technology-dependent, power consuming, or susceptible to process variation. A technology-independent solution to this problem is shielding, which cuts the worst case crosstalk delay by half, but it nearly doubles the wiring area; hence it is unattractive since the routing resource on a chip is scarce.

Although most CACs in the literature require less area and power overhead due to wires than shielding, extra logic

circuits have to be implemented at both ends of the bus as encoders and decoders(CODEC) .Unfortunately, most CODEC designs in the literature have very high complexities, rendering CACs-based solutions Impractical for wide buses. For example, the CODEC in [9] has an exponential complexity with respect to the size of the bus.

Researchers have made a lot of effort in finding an efficient way to implement the CODEC of CACs, leading to solutions such as partial coding [8]. In partial coding, a bus is first broken into sub-buses, which are encoded by using CACs with smaller sizes; then a shielding wire is inserted between each pair of adjacent sub-buses to avoid transition patterns with long crosstalk delay. Forbidden transition overlapping codes (FTOCs) and forbidden pattern overlapping codes (FPOCs) [7] combine partial coding with FTCs and FPCs, respectively. At the expense of a lower code rate and hence larger area and power consumption for the bus, partial coding reduces the complexities of CODECs by keeping the numbers of wires in sub-buses small.

Recently, CODECs based on a Fibonacci based numeral system (FNS) have effectively solved the complexity problem for FPCs and FTCs [10], [11]. Two FPC CODEC designs are proposed based on an FNS [11], and both CODECs have quadratic complexities with the size of the bus. One CODEC in [11] is suboptimal due to its potentially lower code rate, but has a simpler CODEC; the other CODEC in [11] is optimal in its code rate, but requires a more complex circuit. In [10], the FNS is used to encode FTCs. All CODECs in [10] and [11] have quadratic complexities.

In this paper, we generalize the idea in [10] and [11] and establish a generic framework for the CODEC design of all classes of CACs based on binary mixed-radix numeral systems. Using this framework, we propose CODECs for OLCs and FPCs with optimal code rates as well as CODECs for FOCs with near-optimal code rates. Our implementation results show that all our CODECs in this paper have area complexity and delay that increase quadratically with the number of wires and a new coding technique which minimizes self transition activities in the bus lines using spatial redundancy. Our main contributions are as follows.

- In Section 4, generalizing the idea in [10] and [11], we propose a generic encoding algorithm for CACs based on numeral systems.
- In Section 5, we define a modified Fibonacci numeral system, and propose an FPC CODEC based on it. Our FPC CODEC achieves the same code rate as the optimal FPC CODEC in [11] and has a simple circuit, similar to the near-optimal FPC CODEC in [11], integrating the advantages of the two FPC CODECs in [11]
- In Section 6, we define a numeral system for OLC CODECs, and propose an OLC encoding algorithm based on this numeral system. Our CODEC also has a quadratic complexity, which are novel to the best of our knowledge.
- In Section 7, we first prove that we cannot use the generic CAC encoding algorithm based on numeral systems to encode to the whole codebook of an FOC with maximal size. Then we propose an encoding algorithm based on a numeral system that encodes to a subset of an FOC with maximal size. For small, the code rate loss of our suboptimal encoder is small. Our FOC CODECs are also novel to the best of our knowledge. In section 8, the proposed self transition coding scheme is explained, while the results and discussions are provided in section 9 and conclusions are made in section 10.

1.1 Definitions

- Coupling Transition (CT): A Coupling Transition is defined as a transition from 0 – 1or 1 - 0, between two adjacent bus wires.
- Self Transition (ST): A Self Transition is defined as a transition from 0 -- 1 or 1 - 0 on buses with reference to the previous data on it.
- Bus Width (BW): The number of bits in the data is called the Bus Width.

2. Literature Survey

To address interconnect delay effect cross talk avoidance CODEC's are best example, they plays a key role in data transmission and reception.

2.1 Introduction of CODEC's

A codec is a device or computer program capable of encoding or decoding a digital data stream or signal. The word codec is a portmanteau of "coder-decoder" or, less commonly, "compressor- de compressor". A codec encodes a data stream or signal for transmission, storage or encryption, or decodes it for playback or editing. Codec's are used in videoconferencing, streaming and editing applications.

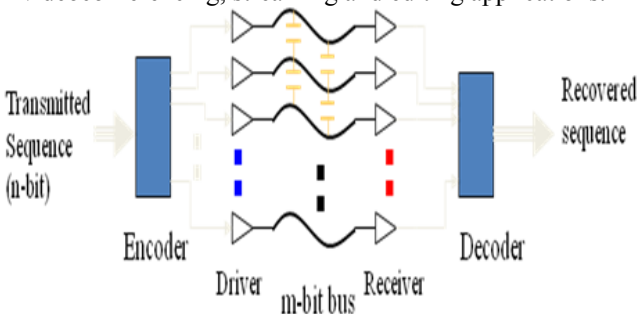

Figure 2.1: Block Diagram of CODEC Design

2.1.1. Encoder

An encoder is a device, circuit, transducer, software program, algorithm or person that converts information from one format or code to another, for the purposes of standardization, speed, secrecy, security, or saving space by shrinking size. For example a compressor encodes data (e.g., audio/video/images) into a smaller form.

2.1.2. Decoder

A decoder is a device which does the reverse operation of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode. It is a combinational circuit that converts binary information from n input lines to a maximum of 2n unique output lines.

2.2 Cross talk

Crosstalk (XT) is any phenomenon by which a signal transmitted on one circuit or channel of a transmission system creates an undesired effect in another circuit or channel. Crosstalk is usually caused by undesired capacitive, inductive, or conductive coupling from one circuit, part of a circuit, or channel, to another.

Cross talk leads cross talk delay and cross talk noise which degrade the system performance and May it leads to functionality failure respectively. When two nets are in parallel (one is aggressor and other one is victim) passing the data in the same direction leads to reduced delay of bus and when passing the data in opposite direction leads to increased delay of bus. Similarly when aggressor is switching and victim is static, cross talk noise is introduced between two net which leads to functionality failure of the design.

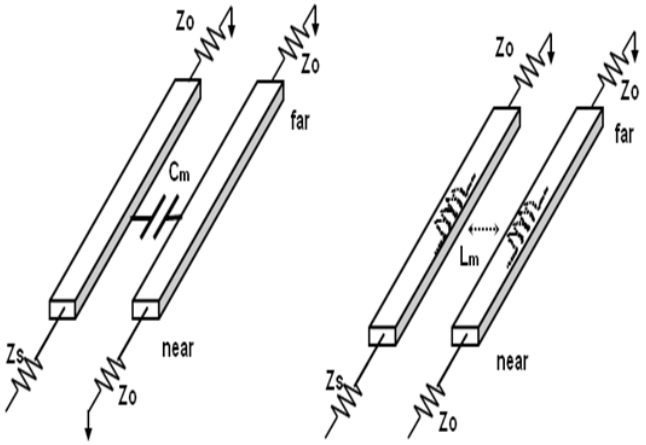

Figure 2.2: Cross talk between two wires due to induced capacitance and inductance

2.3 On-chip bus delay calculation

Deep sub micrometer system-on-chip designs suffer from the delay of global buses, which increases while the gate delay decreases with the shrinking feature size. The delay of the ith wire of an m -bit bus is given by

$$T_i = \begin{cases} \tau_0 \left[(1-\lambda)\Delta_1^2 - \lambda\Delta_1\Delta_2\right] & i = 1 \\ \tau_0 \left[(1-2\lambda)\Delta_i^2 - \lambda\Delta_i(\Delta_{i-1} + \Delta_{i-1})\right] & i \neq 1, m \\ \tau_0 \left[(1-\lambda)\Delta_m^2 - \lambda\Delta_m\Delta_{m-1}\right] & i = m \end{cases}$$

where $\boldsymbol{\lambda}$ is the ratio of the coupling capacitance between adjacent wires and the loading capacitance between the ith wire and the ground, T0 is the delay of a transition on a single wire, and $\boldsymbol{\Delta_i}$ equals 1 for 0 -> 1 transition, -1 for 1 -> 0 transition, or 0 for no transition on the i^{th} wire. As the feature size shrinks, the ratio $\boldsymbol{\lambda}$ increases, and the crosstalk delay may be several times more than the delay of a single wire and thus dominates the delay of a bus. The crosstalk delay has become a bottleneck in deep sub micrometer system-on-chip designs. This problem is so significant that global wiring scaling issues have been identified as Grand Challenges in recent International Technology Roadmap for Semiconductors (ITRS).

3. Previous Work

Despite the availability of the codes, no systematic mapping of data words to code words has been proposed for CODEC design. This is mainly due to the nonlinear nature of the crosstalk avoidance codes (CAC). The lack of practical CODEC construction schemes has hampered the use of such codes in practical designs. This work presents guidelines for the CODEC design of the "forbidden pattern free crosstalk avoidance code" (FPF-CAC). We analyze the properties of the FPF-CAC and show that mathematically, a mapping scheme exists based on the representation of numbers in the Fibonacci numeral system. Our first proposed CODEC design offers a near-optimal area overhead performance. An improved version of the CODEC is then presented, which achieves theoretical optimal performance. We also investigate the implementation details of the CODECs, including design complexity and the speed. Optimization schemes are provided to reduce the size of the CODEC and improve its speed.

4. Generic CAC Codec Designs Based On Numeral Systems

4.1 Introduction to Numeral Systems

A numeral system is a linguistic system and mathematical notation for representing numbers of a given set by symbols in a consistent manner [13]. The most commonly used numeral systems are positional numeral systems [13], where given a positive natural number, a string $(a_1 m \ldots . a_1(2,)\ a_1 1\ )_1 b$ represents a number $\sum_{i=1}^{m} a\ i^{b^{i-1}}$. For example, the binary and decimal numeral systems use powers of two and powers of ten, respectively, as bases. A binary mixed-radix numeral system is that given a basis set of non-negative numbers $\{Y_m, \ldots, Y_2, Y_1\}$, a binary string $(d_m \ldots . d_2 d_1)$ represents a number $\sum_{i=1}^{m} d\ iY_i$. In this paper, we focus on binary mixed-radix numeral systems hence forth.

A numeral system is complete if any integer $\mu e[\mathbb{J}, 2_1(i=1)^{\mathbb{N}} m \blacksquare Y$ i] Can be represented by at least one binary string $(d_m, d_{m-1} \ldots . d_2 d_1)$.

4.2 Generic CAC Encoding Algorithm

Suppose we want to transmit a -bit data message over a bus With m(m>=k) wires in one clock cycle. These k bits are first encoded into an -bit CAC codeword so that the transition patterns with long crosstalk delays are avoided. The k-bit CAC codeword is then transmitted over the bus and received by the decoder. Then the -bit message is recovered at the decoder. The idea of numeral system based CAC CODEC is that the k -bit data message can be viewed as an integer v such that $0 \leq v \leq 2^k - 1$ in the binary numeral system, and the goal of encoding algorithm is to convert into an -bit binary string, which represents under another numeral system and has no transition pattern with

long crosstalk delay. Since the encoded codeword contains only 0 and 1 and the numeral system needs to be complete, we have to use a binary mixed-radix numeral system.

Consider an m-bit CAC codebook $C(m)$ with size $C(m)$. If a numeral system is used to encode, we consider an encoder, which is essentially a mapping from all integers in to with the following properties:

• All the codeword's can be mapped from an integer in, which implies that is subjective;
• Different codeword's represent different integers under the mapping.

We propose a generic CAC encoding algorithm based on a numeral system in Algorithm 1 below. In Algorithm 1, $\{\gamma_m, \cdots, \gamma_2, \gamma_1\}$ is the basis set of the encoding numeral system, $\{\alpha_i\}, \{\beta_i\}$ and Θ are some constants depending on the CACs. $d_m, d_{m-1} \cdots d_2 d_1$ is the output of the encoding algorithm; also it is a codeword in the CAC. It is easy to see that the data message is recovered by computing

$$v = \sum_{i=1}^{m} d_i \gamma_i .$$

The CODEC for a CAC based on Algorithm 1 is shown in Fig. 1. The encoder consists of processing elements, and all processing elements have the same circuit, shown in Fig. 2. The top processing element is slightly different from the others in that $\alpha_m = \beta_m = \Theta$, which renders the input d_{m+1} to the top processing element is don't care in Fig. 1). Each processing element consists of two comparators, one subtractor, and one multiplexer. Each processing element has three parameters α_k, β_k and γ_k two inputs d_{k+1} and r_{k+1} , and two outputs d_k and r_k .

Algorithm 1 Generic CAC encoding algorithm

Input: code length m, integer v $(0 \le v \le \sum_{i=1}^{m} \gamma_i)$.
  for $k = m$ downto 2 **do**
    if $k = m$ **then**
      if $v \ge \Theta$ **then**
        $d_m = 1$;
      else
        $d_m = 0$;
      end if
      $r_m = v - d_m \cdot \gamma_m$;
    else
      if $r_{k+1} \ge \alpha_k$ **then**
        $d_k = 1$;
      else if $r_{k+1} < \beta_k$ **then**
        $d_k = 0$;
      else
        $d_k = d_{k+1}$;
      end if
      $r_k = r_{k+1} - d_k \cdot \gamma_k$;
    end if
  end for
  $d_1 = r_2$;
  Output: $d_m d_{m-1} \ldots d_1$.

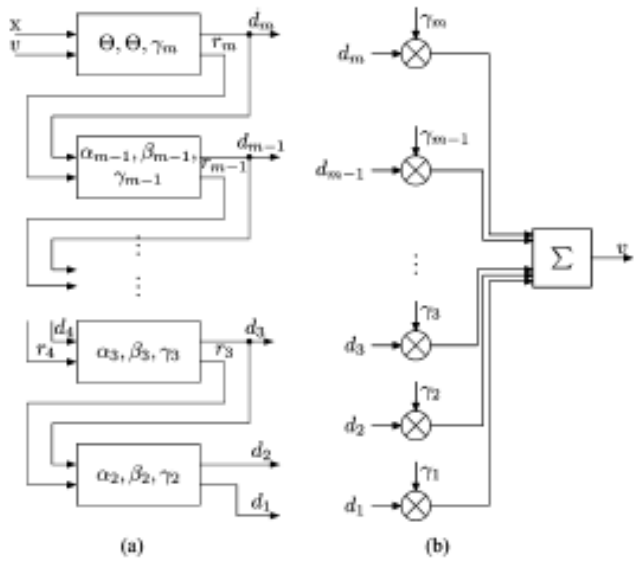

Figure 3.1: Generic CODEC of an m-bit CAC based on Algorithm 1 (note the similarity to the CODEC shown in [11.fig.3]. a) Encoder. b) Decoder.

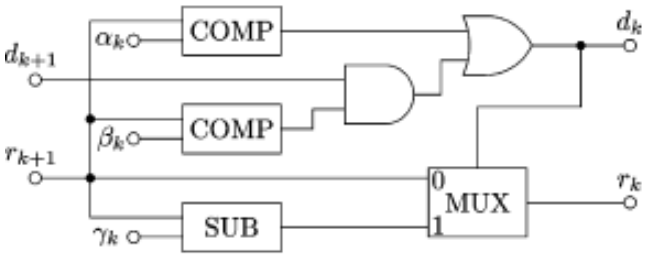

Figure 3.2: Processing element of the encoder in fig.1 (note the similarity to [11, fig.4].

5. Forbidden Pattern Free CAC

FPC is familiar to avoid (1+2λ) codes the codebook size of an m-bit FPC is given by 2Fm+1, slightly greater than that of an m –bit FTC. Since the number of codeword's needed is a power of two, an m-bit FPC leads to a higher rate than an m-bit FTC.

5.1 Numeral systems for FPC Codec's

Let {Fk} be a Fibonacci sequence. We have the following proposition.

FPC CODEC Design

With the help of the MFNS, the FPC CODEC can be designed as a special case of our general CAC by choosing

γk= Pk, Θ=Fm+1, αk =Fk+1, βk =FK .

Generic CAC encoding algorithm:

Input: code length m, integer v (0≤V ≤∑i=1m γi)

For k=m downto 2
do if k=m then
if v ≥ Θthen
dm=1;

else
dm=0;
endif
rm=v-dm.γm;
else
if rk+1≥ αk then
dk=1;
elseif rk+1< βk then
dk=0;
else
dk=dk+1;
end if
rk= rk+1- dk.γk;
end if
end for
d1=r2;
output: dmdm-1dm-2d…..d1

As a special case of our generic CAC CODEC, the circuitry of our FPC CODEC design has a quadratic complexity.

The below table represent FPF-CAC code words for 2, 3, 4 and 5-bit buses.

Table 1: FPF-CAC codeword's

2-bit	*3-bits*	*4-bits*	*5-bit*	
00	000	0000	00000	10000
01	001	0001	00001	10001
10	011	0011	00011	10011
11	100	0110	00110	11000
	110	0111	00111	11001
	111	1000	01100	11100
		1001	01110	11110
		1100	01111	11111
		1110		
		1111		

6. One lambda codes (OLC)

The (1+λ)T0 codes can achieve a worst case delay of (1+λ)T0 OLCs, are a kind of (1+λ) codes. In an OLC, no adjacent wires can transition in opposite directions when transitioning from one codeword to another. Thus the transition patterns 01 → 10 and 10 → 01 are avoided. Consider a boundary between two adjacent wires. If in all codeword's, there are only 00, 01, and 11 across this boundary, it is referred to as 01-type boundary. Otherwise if only 00, 10, and 11 appear this boundary, it is referred to as a 10-type boundary. All of the boundaries in an OLC are either 01-type or 10-type. That the OLC codebook with maximal size satisfies the following two conditions: 1) The codebook has alternating 01- and 10-type boundaries and 2) the bit patterns 010, 101, 1001, and 0110 cannot appear in any of the codeword's. The maximal cardinality of an m-bit OLC codebook, satisfies following recursion relation

gm= gm-1 + gm-5 for m≥6

6.1 Numeral systems for OLC Codec's

The OLC's codec can be designed by choosing the following parameters the maximal cardinality of an m-bit OLC codebook is given by gm. The numeral system defined by {fi}i=1m can be used to encode an m-bit OLC.

γk= fk, Θ=G2[m/2]+4
αk =G2l+2, for k=2l-1 and infinity for k=2l
βk =0, for k=2l-1 and G2l+2 for k=2l-1

6.2 OLC CODEC Design

Generic CAC encoding algorithm:

Input: code length m, integer v (0≤V ≤∑i=1m γi)

For k=m down to 2
do if k=m then
if v ≥ Θthen
dm=1;
else
dm=0;
endif
rm=v-dm.γm;
else
if rk+1≥ αk then
dk=1;
elseif rk+1< βk then
dk=0;
else
dk=dk+1;
end if
rk= rk+1- dk.γk;
end if
end for
d1=r2;
output: dmdm-1dm-2d…..d1

The bit patterns 1001 and 0110 violate the alternating boundary type constraint, and thus they cannot appear in the output vector. Thus the output vector is an OLC codeword. Since the largest codebook size of an -bit OLC is , and we can get different codeword's satisfying the constraints of OLC codeword's by the OLC encoding algorithm, the algorithm gives a bisection from integers in to the -bit OLC codebook, implying that the algorithm is optimal.

The One lambda codes efficiently work and yield good codec designs based on data bit transitions by making data transition in one direction as mentioned in introduction of one lambda codes.

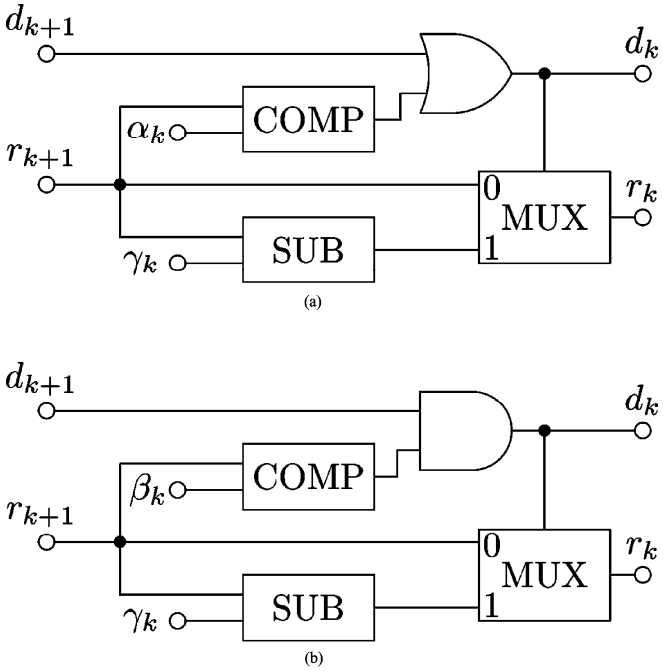

Figure 4: OLC CODEC processing element. (a) Processing element circuit when k is odd. (b) Processing element circuit when k is even

7. Forbidden Over Lapping Codes (FOC)

The (1+3λ)T0 codes can achieve a worst case delay of (1+3λ)T0 FOCs, are a kind of (1+3λ) codes. A 3-bit pattern b1b2b3 around a bit di if di+1didi-1 = b1b2b3. The FOC codebook satisfies the following constraint: the codebook cannot have both 010 and 101 appearing around any bit position. The maximal size of an m-bit FOC is given by, Tm where
Tm=Tm-1 + Tm-2 + Tm-3 for m ≥4 and T1=2, T2=4, and T3=7.

Generic CAC encoding algorithm:

Input: code length m, integer v (0≤V ≤∑i=1m γi)
For k=m downto 2
do if k=m then
if v ≥ Θthen
dm=1;
else
dm=0;
endif
rm=v-dm.γm;
else
if rk+1≥ αk then
dk=1;
elseif rk+1< βk then
dk=0;
else
dk=dk+1;
end if
rk= rk+1- dk.γk;
end if
end for
d1=r2;
output: dmdm-1dm-2d…..d1

The processing element for FOC CODEC is shown in the following figure 3.4

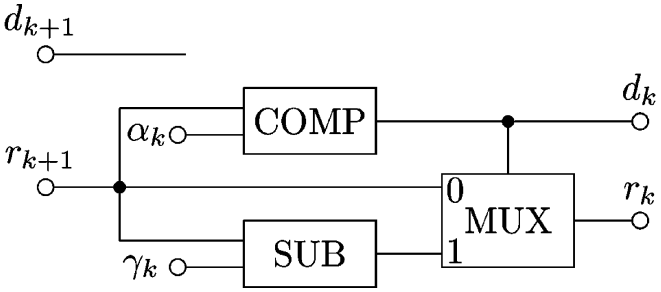

Figure 4.1: Processing element for FOC codec's

8. Self Capacitance Reduction

A self capacitance is defined as the Capacitance associated with the net and body or substrate of the device. A common form of energy storage device is a parallel-plate capacitor. In a parallel plate capacitor, capacitance is directly proportional to the surface area of the conductor plates and inversely proportional to the separation distance between the plates.

The self capacitance reduction is done by comparing the present input with previous input and selecting appropriate encoded data pattern corresponding to the minimum value of self transitions is transmitted on the bus.

Self capacitance reduction is done by the following procedure.

Let the data on an n bit wide bus, at time instant to be denoted as At = {atn-1, atn-2, atn-3……, at1, at0}. The data transmitted on the bus is denoted as A (t) enc. The function calculates (datal, data2) finds the number of self transitions between (datal, data2). The function swapAdj_n (At) swaps the adjacent bus lines in At and gives the output Asw(t) ={atn-1,atn-2,atn-3……,at0,at1}.

The energy efficient coding scheme is as follows:

- Let A(t-1)enc be the previously coded data which was transmitted on the bus and let At be the present data which should be encoded and transmitted.
- Find XOR of At with A(t-1)enc and affix it with 00, for decoding purposes. Let this new data be denoted as At(xor) Evaluate stxor=calculateST_n(At(xor) ,A(t-1)enc)
- Likewise, find XNOR of in At with A(t-1)enc and affix it with 01. Let this new data be denoted as At(xnor). Evaluate stxnor=calculateST-n (At(xnor),A(t- 1)enc).
- Let Asw(t)=swapAdj_n(At). Suffix ASW (t) with 10 and let this new data be denoted as At(swP).
- Evaluate stswp=calculateSTLn(At(swP) A(t-1)enc)
- Suffix At with 11 and let this new data be denoted as At(unc) Evaluate
- stunc=calculateST-n(At(unc) ,A(t-1)enc).
- Find min(stLxor,stxnor,stswp,stunc)
- The coded value corresponding to the minimum value in step six is transmitted.

The block diagram for self capacitance reduction encoder is depicted as

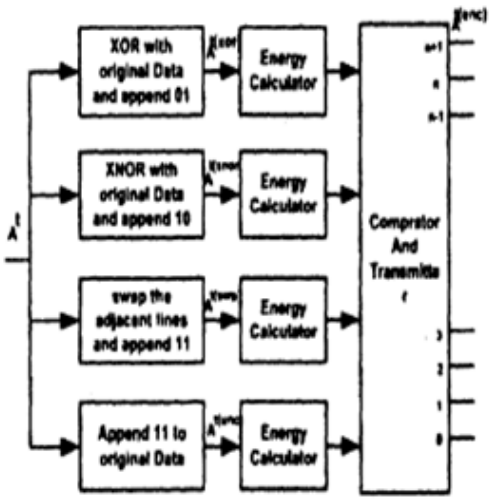

Figure 4.2: Self capacitance reduction encoder

The block diagram for self capacitance reduction decoder is depicted as

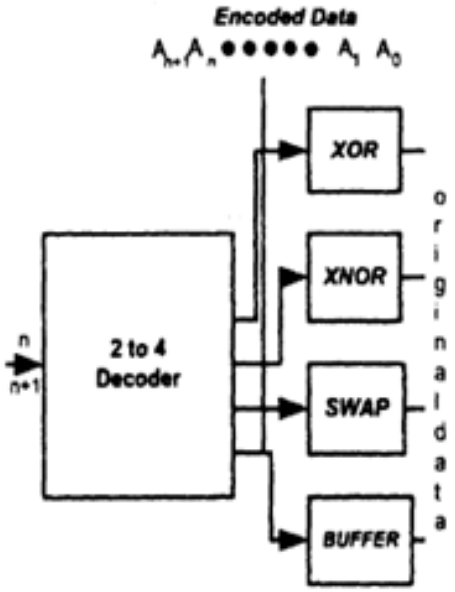

Figure 4.3: Self capacitance reduction decoder

9. Results and Discussion

Comparison of different CAC's

The following table3.3 depicts the constants which are used in different CAC's

CAC	Θ	α_k	β_k	γ_k
OLC	$G_{2\lfloor \frac{m}{2} \rfloor+4}$	G_{2l+2} $k=2l-1$ ∞ $k=2l$	0 $k=2l-1$ G_{2l+2} $k=2l$	f_k
FTC	$F_{2\lfloor \frac{k}{2} \rfloor+1}$	$F_{2\lfloor \frac{k}{2} \rfloor+1}$	$F_{2\lfloor \frac{k}{2} \rfloor+1}$	F_k
FPC	F_{m+1}	F_{k+1}	F_k	F_{m+1} $k=m$ F_k $1 \le k < m$
FOC	$\sum_{i=2\lfloor \frac{m}{2} \rfloor}^{m} h_i$	$\sum_{i=2\lfloor \frac{k}{2} \rfloor}^{k} h_i$	$\sum_{i=2\lfloor \frac{k}{2} \rfloor}^{k} h_i$	h_k

Table 2: Constants used in different CAC's

Comparison between CACs:

Table 3: Comparison between CACs

CACs	*Path delay (ns)*	*Power Req. (mw)*	*No. of slices (out of 8672)*
FPC	41.760	152.43	52
OLC	45.546	158.02	48
FOC	56.975	148.43	77

The delay of different CAC's represented in the figure 4.4

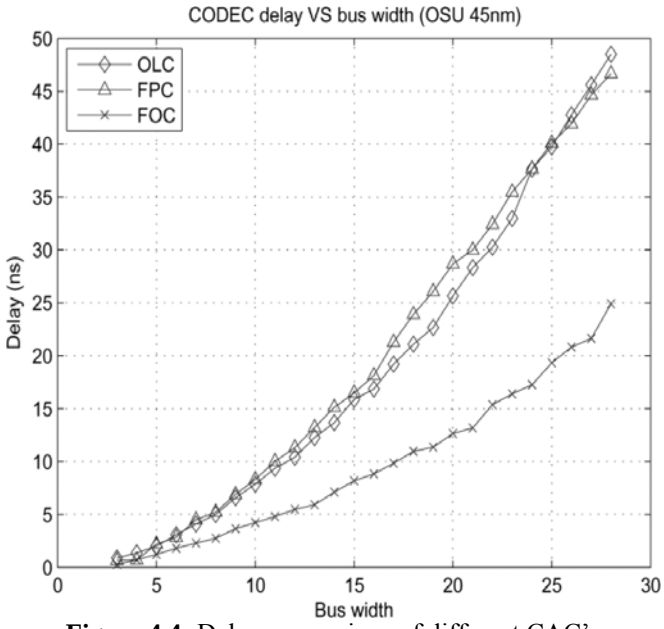

Figure 4.4: Delay comparison of different CAC's

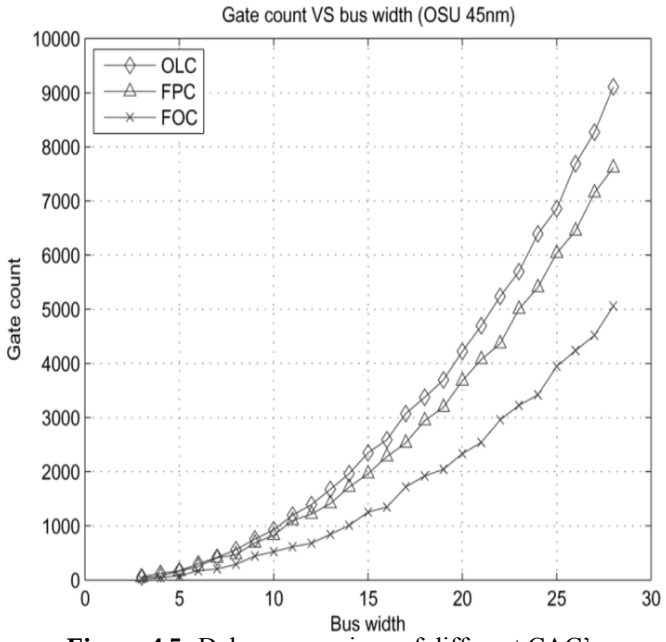

Figure 4.5: Delay comparison of different CAC's

To quantify the delay as well as area and power overheads introduced by our CAC CODECs, we implemented our OLC, FPC, and FOC CODECs based on numeral systems without pipelining. Our CODECs are simulated on Modelsim2 and synthesized by Cadence Encounter RTL Compiler3 with an OKSU Free PDK 45-nm process 4; the figures for power consumption of our CODECs are derived by the power analysis tool in Encounter.

To measure the CODEC power consumption, we assume the clock rate is 100 MHz and set the input switching rate to be 0.5 for all CODECs. The clock rate is selected merely for the purpose of demonstrations, and is inconsequential to our conclusions below. In practice, the clock rate should be determined by bus delays as well as CODEC delays. Our CODEC delays are not likely to be the bottleneck of achievable clock rates for two reasons. First, the delays of our CODECs can be easily improved by pipelining or partitioning the bus, as discussed in the paper. Second, since the technology trend indicates increasing bus delays and decreasing gate delays, the bus delays will be more likely to be the bottleneck.

The implementation results are shown in Figs. 4–4. Fig. 4.5 shows the delay introduced by our CODECs, CODEC complexities in terms equivalent gate count. The result of area consumption includes the cell area only. The power consumption of our CODECs, include the leakage power and the estimated internal and switching power. Our simulation results show that the delay and the area complexity as well as the power consumption of our CODECs increase quadratically with the bus width.

10. Conclusion

In this paper, we establish a framework for the CAC CODEC design based on numeral systems, and devise efficient CODECs for OLCs, FPCs, and FOCs by choosing appropriate numeral systems and constants. The results are summarized in Table I. Implementation results show that our CODECs all have area and delay that increase quadratically with the bus width. Used together with partial coding, our efficient CODECs help make CACs a viable option in combating crosstalk delay, which is a bottleneck in deep sub-micron system-on-chip designs.

11. Future work

The analytical models and noise reduction techniques were analyzed for use with past, present, and future IC packaging in order to predict and improve performance. The experimental results illustrated that the techniques were successful and made significant improvement in the performance of the pack- aging. While these techniques were demonstrated to have an immediate impact when applied to commonly used VLSI packages, the current trends in IC technology make these techniques even more invaluable. In addition, since all of the modelling and performance techniques were described using a common mathematical framework, the work in this monograph can be easily applied to a wide variety of electronic applications.

References

[1] P. P. Sotiriadis, "Interconnect modelling and optimization in deep submicron technologies," Ph.D. dissertation, Dept. Elect. Eng. Compute. Sci., Massachusetts Inst. Technol., Cambridge, 2002
[2] K. Hirose and H. Yasuura, "A bus delay reduction technique considering crosstalk," in Proc. Des. Autom. Test Eur. Conf. Exhibition, 2000, pp. 441–445.
[3] B. Victor, "Bus encoding to prevent crosstalk delay," M.S. thesis, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, 2001
[4] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Des., 2001, pp. 57–63.
[5] P. P. Sotiriadis and A. Chandrakasan, "Reducing bus delay in submicron technology using coding," in
[6] Proc. Conf. Asia South Pacific Des. Autom., 2001, pp. 109–114.
[7] C. Duan and A. Tirumala, "Analysis and avoidance of cross-talk in on-chip buses," in Proc. 9th Symp. High Perform. Interconnects (HOTI), 2001, pp. 133–133.
[8] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag, "Area and energyefficient crosstalk avoidance codes for on-chip buses," in Proc. IEEE Int. Conf. Comput. Des.: VLSI Comput. Processors, 2004, pp. 12 17
[9] S. R. Sridhara and N. R. Shanbhag, "Coding for reliable on-chip buses: A class of fundamental bounds and practical codes," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 26, no. 5, pp. 977–982, May 2007.
[10] S. R. Sridhara, "Communication inspired design of on-chip buses," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Illinois, Urbana, IL, 2006
[11] C. Duan, C. Zhu, and S. P. Khatri, "Forbidden transition free crosstalk avoidance CODEC design," in Proc. DAC, Jun. 8–13, 2008, pp. 986–991.
[12] Spec 95 benchmark suite: http:l/spec.com

Author Profiles

K. Ramesh received the B. Tech degree in Electronics and Communication Engineering from Srinivas Redddy Institute of Technology, Munipally, Nizamabad (Dist.), India, affiliated to Jawaharlal Nehru Technological University Hyderabad, India, in 2009, currently pursuing Master of Technology in VLSI System Design at CVSR Engineering & Technology, Venkatapur, Hyderabad, India. His research interests include Low Power VLSI Design (ASIC) and FPGA based Digital Design using Verilog HDL.

E. Srinivas received the B. Tech degree in Electronics and Communication Engineering from Anurag Engineering College, Kodad, Nalgonda (dist), India, affiliated to Jawaharlal Nehru Technological University Hyderabad, India, in 2007, Master of Technology in VLSI System Design at CVSR Engineering & Technology, Venkatapur, Hyderabad, India in 2010. He is currently pursuing PhD in J.N.T.U. Hyderabad. His research interests include Low Power VLSI Design (ASIC) and FPGA based Digital Design using Verilog HDL.