Crosstalk-aware Domino Logic Synthesis

Yi-Yu Liu
Dept. of Computer Science
Tsing Hua University
Hsinchu, Taiwan, R.O.C.

TingTing Hwang
Dept. of Computer Science
Tsing Hua University
Hsinchu, Taiwan, R.O.C.

Abstract

We propose a logic synthesis flow which utilizes the functionality of circuit to synthesize a domino-cell network which will have more wires crosstalk-immune to each other. For that purpose, techniques of output phase flipping and crosstalk-aware technology mapping are used. Meanwhile, metric to measure the crosstalk sensitivity of domino cells in synthesis level is proposed. Experimental results demonstrate that the crosstalk sensitivity of the synthesized domino-cell network is greatly reduced by 51% using our synthesis flow as compared with conventional methodology. Furthermore, after placement and routing are performed, the ratio of the number of crosstalk-immune wire pairs to the number of total wire pairs is about 25% using our methodology as compared to 9% using conventional techniques.

I. Introduction

In deep submicron (DSM) technology, coupling capacitance grows exponentially to the down-sizing of the feature size. The crosstalk effect as such will result in performance degradation and at worst will give incorrect result. Therefore, crosstalk minimization becomes an important issue today [1].

Domino logic is a well developed circuit implementation style. As compared with the conventional CMOS implementation, domino logic has one advantage of faster speed. It is widely used in controlling logic and ALU for high performance designs. Domino logic operates in two phases: the precharge phase and the evaluation phase. In the precharge phase, output wire is precharged from low to high; while in the evaluation phase, output wire will either be discharged from high to low or remain high. Note that, once the output is discharged during evaluation, it is impossible to restore the output until next precharge phase.

The domino-cell design is especially vulnerable for crosstalk noise during evaluation phase. In the evaluation phase, if there is a phase transition in an aggressor from high to low and a victim remains high, the victim wire may be induced a signal transition from high to low due to crosstalk. Hence, the output is not correctly discharged by the victim wire.

Recently, research on crosstalk minimization focuses on optimization in physical level. The widely accepted methods are wire spacing and shielding which are considered the most effective ways to solve the crosstalk problem in domino circuit [2]. However, a lot of area overhead will be incurred by using these techniques. On the other hand, Im and Roy propose the placement of NMOS in each domino cell [3] so that the crosstalk effect of a domino cell can be minimized through its unique NMOS reordering technique. Ho, Chang, et al, develop a multilevel routing system to minimize crosstalk effect on the routing wires [4]. Kim, Jung, et al, suggest track assignment technique which takes crosstalk immunity into consideration [5]. Lou and Chen focus on crosstalk minimization before wire routing [6]. Ren, Pan, and Villarrubia try to extract the implicit information in advance to perform an incremental crosstalk-aware placement [7]. However, crosstalk minimization in physical level is far from proactive. The reason is that after the logic synthesis is performed, the functionalities of wires are fixed in physical level. The only option left is to identify the vulnerable wires and then to find a better placement and routing for them.

We observe that the crosstalk can be further reduced in physical level if a logic function is synthesized taking crosstalk effect into consideration. Figure 1 shows an example where Figure 1(a) is one technology mapped domino network with 5 cells and 4 inter-cell wires, k, n, o, and r, and Figure 1(b) shows the same circuit mapped to another 5 cells with 4 inter-cell wires, l, m, p, and q. In Figure 1(a), interconnect wires, a, d, k, n, o, and r, are crosstalk sensitive to each other. None of the pairs are crosstalk immune to each other. However, in Figure 1(b), pairs $\{g, l\}$, $\{j, m\}$, and $\{p, q\}$ are crosstalk immune to each other. (The conditions for wires to be immune to each other will be presented in Section II.) From the above example, we can see that synthesis tool plays an important role in minimizing crosstalk. On the other hand, to check if two wires induce crosstalk can be accomplished only after placement and routing are performed. To synthesize a circuit taking crosstalk into consideration, we must be able to accurately predict the wire adjacency of a layout in synthesis level.

For that purpose, we develop a cost function to predict wire adjacency in synthesis level and propose two techniques to synthesize a crosstalk-aware network. The first technique is used before technology mapping to construct a crosstalk-aware AND-OR network. The second technique is used for a crosstalk-aware technology mapping for domino cell. With a crosstalk-aware synthesized network, later placement and routing will have a better starting point.

The rest of the paper is organized as follows. The preliminary is given in Section II. Section III presents our observations and cost function to reduce crosstalk effect. The

3-9810801-0-6/DATE06 © 2006 EDAA

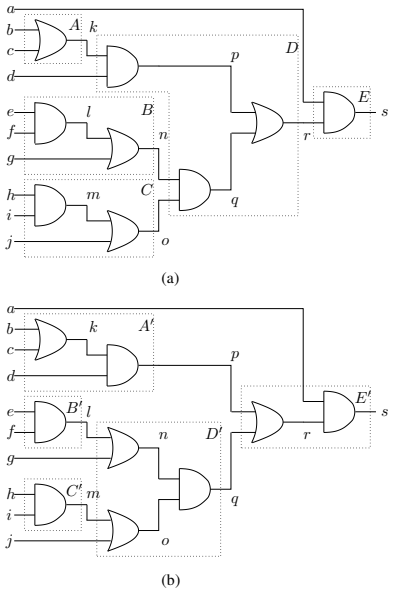

Fig. 1. Example of crosstalk effect in mapped circuits.

synthesis flow for domino circuit generation is proposed in Section IV. Our experimental results are shown in Section V. Section VI concludes this work.

II. Preliminary

A. Unateness of domino circuit

Since the phase transitions in domino circuit is required to be in monotone, a logic function to be implemented as domino circuit must be unate. To produce an unate logic, we can apply the DeMorgan's law to push all inverters toward primary inputs or primary outputs. The resultant unate network will consist of AND gates and OR gates only. Previous work addresses this issue focusing on minimizing logic duplication of the trapped inverters. Among others, Puri, Bjorksten and Rosser propose a systematic method to assign the phases of primary outputs for dynamic logic [8]. Zhao and Sapatnekar formulate the phase assignment as a 0-1 integer programming problem [9].

B. Crosstalk immunity of wires

In physical level, disregarding the functionalities of two wires, their adjacencies are viewed as the cause of crosstalk. However, if we take the functionality into consideration, two adjacent wires may not incur any crosstalk effect. Let the two adjacent wires a and v be the aggressor and victim, respectively. The crosstalk effect is induced by both the *observability* of crosstalk effect and the *satisfiability* of aggressor and victim signals [5]. They are described as follows.

Definition 2.1: (Observability): The crosstalk error on signal v can be observed only if there exists an input pattern which propagates the transition of v to any primary output. The *observability* of wire v, denoted as OBS_v, can be formulated as

$$OBS_v = \{I \in PI | \sum_{F_j \in PO} \frac{\partial F_j(I)}{\partial G_v(I)}\} \ , \qquad (1)$$

where F_j is the j-th primary output of function F, G_v is the function of v, and $\frac{\partial F_j(I)}{\partial G_v(I)}$ represents the Boolean difference of F_j with respect to G_v [10].

Definition 2.2: (Satisfiability): The crosstalk error occurs only if there exists an input pattern which satisfies a phase transition $(0 \rightarrow 1)$ of a and stable low of v. The *satisfiability* of wires v and a, denoted as SAT_{va}, can be formulated as

$$SAT_{va} = \{I \in PI | (G_v(I) = 0) \cdot (G_a(I) = 1)\} \ , \qquad (2)$$

where $G_v(I)$ is the function of wire v, $G_a(I)$ is the function of wire a, and I is a primary input pattern.

Definition 2.3: (CT-immune): A victim wire v is crosstalk immune or CT-immune to its adjacent aggressor wire a if there exists no input pattern which satisfies both *satisfiability* and *observability*. It can be formulated as

$$\textit{CT-immune}_{va} = (OBS_v \cdot SAT_{va} = \emptyset) \ . \qquad (3)$$

The *satisfiability* is a necessary condition to setup the crosstalk error in the victim, while the *observability* is a necessary condition to observe the crosstalk error in the victim at the output. Both *satisfiability* and *observability* are necessary criteria to induce and propagate the crosstalk error to the primary outputs. If any one of the above conditions does not hold, the crosstalk effect of an aggressor, a, to a victim, v, can be ignored.

We take the domino network in Figure 1(b) as an example to illustrate the CT-immune condition in domino network. There are 5 domino cells, A', B', C', D' and E'. The interconnect wires between A', D', and E' are p and q. Both wires p and q are the fanins of E'. If there is a phase transition $(0 \rightarrow 1)$ of p and stable low $(0 \rightarrow 0)$ of q, wire q may be induced an unwanted phase transition $(0 \rightarrow 1)$. The error, caused by p, of wire q will not be observed, since the output of E', s, is evaluated to 1 due to $p = 1$. Thus, q is CT-immune to p. Similarly, the error, caused by q, of wire p will also not be observed. Therefore, both p and q are CT-immune to each other.

III. Relations of wire adjacency in physical level and I/O of mapped cell in synthesis level

As pointed out in Introduction, a synthesis tool plays an important role in reducing crosstalk. However, to check if two wires induce crosstalk can be accomplished only after placement and routing are performed. To synthesize a circuit taking crosstalk into consideration, we must develop a useful cost function in synthesis level which accurately predicts wire adjacency of the final layout in physical level. In this section, we will first discuss the relations of wire adjacency and I/O of mapped domino cell in subsection III.A. Then, in subsection III.B, we will show that inputs to OR gate are CT-immune to each other in synthesis level. Finally, in subsection III.C, we develop our cost function for technology mapping based on the results in subsection III.A and III.B.

TABLE I

WIRE ADJACENCY AND I/O OF A CELL

Circuit	*I/O Adj*	*Tot Adj*	*Ratio* (%)
C1355	348	797	43.66
dalu	546	1153	47.35
C880	174	431	40.37
count	13	47	27.66
c8	20	37	54.05
C1908	344	1054	32.64
C2670	578	1328	43.52
C3540	2076	4817	43.10
C6288	2445	7608	32.14
b9	22	50	44.00
k2	1011	2698	37.47
des	4143	8072	51.33
C7552	2163	5519	39.19
t481	401	897	44.70
rot	313	696	44.97
average			41.74

A. Wire adjacency and I/O of mapped domino cells

In physical level, the adjacency of two wires are the necessary condition for two wires to induce crosstalk effect. Hence, our first challenge is to correlate the physical adjacency of wires with the interconnections of a synthesized logic. We observe a placed and routed circuit and find that many adjacent wires are the I/O signals to the same mapped domino cells. To understand if the observation is correct in general, we perform a set of experiments. First, circuits from MCNC benchmark are synthesized by *SIS* [11] and mapped to domino cells by a dynamic-programming-based domino technology mapping algorithm proposed by Zhao and Sapatnekar [12]. After that, we perform placement and routing for these circuits using *Dragon* [13] and *Multilevel Router* [14]. Finally, pairs of adjacent wires in each layout are extracted to check if they are I/O signals to the same mapped domino cell. Table I shows the results. Column *I/O Adj* is the number of adjacent wire pairs which are the I/O signals of the same mapped domino cell. Column *Tot Adj* is the number of total adjacent wire pairs of the circuit. Column *Ratio* is the ratio of *I/O Adj* to *Tot Adj*. From Table I, we can see that about 40% of adjacent wire pairs are I/O signals of the same mapped domino cell. This is an important finding. We can utilize this property in synthesis level to synthesize a domino network which is likely to have more CT-immune adjacent wires in physical level. That is, if a synthesized circuit has more cells whose I/O signals are CT-immune to each other, there will be more adjacent wire pairs in physical level to be CT-immune to each other.

B. Crosstalk immunity of OR gates

In the previous subsection, we observe that about 40% of adjacent wires are I/O signals of the same mapped domino cell from circuit layouts. Based on this observation, in order to synthesize a circuit that is more likely to be CT-immune, we can pay attention to the I/O signals of a mapped domino cell. We find that the *observability* of signals in OR gate is an empty set. Since the crosstalk error will not be propagated to primary output unless both the *satisfiability* and *observability* hold, we can exploit this property for crosstalk reduction.

Therefore, we should take advantage of OR gate network to synthesize a domino logic with more wires which are CT-immune to each other.

C. Cost function for crosstalk in technology mapping

To model the relationship of crosstalk effect between each wire, one commonly used method is the 0-1 sensitivity matrix [5] [15] [16]. A 0-1 sensitivity matrix is an n by n matrix consisting of 0 and 1. In the matrix, the element $s_{i,j}$ is 0 if the phase transition of wire i and wire j do not affect each other; otherwise $s_{i,j}$ is 1. A Boolean network with more 0 in its sensitivity matrix means that there are more CT-immune wires in that circuit. Although the immunity relationship between every pair of signals can be represented in sensitivity matrix, a lot of information contained in sensitivity matrix is redundant and irrelevant. The reason is that the most serious crosstalk effect occurs only between neighboring wires shown in Table I instead of every pair of wires in the whole circuit.

Based on our observations presented in subsection III.A and III.B, we develop a new sensitivity metric for synthesis level optimization. Our metric is developed based on the observation where I/O of a mapped cell is more likely to be routed in the same neighborhood. Therefore, we will focus on the crosstalk effect within each individual domino cell rather than the whole Boolean network in synthesis level.

By using the concept of crosstalk immunity in OR network which is discussed in Section 3.B, we define the sensitivity of each domino cell in the following.

Definition 3.1: ($SC(in_j)$): The sensitive contribution of the input signal in_j in a domino cell can be formulated as

$$SC(in_j) = \begin{cases} \frac{1}{n}, & \text{if } in_j \text{ is an input to an OR network.} \\ 1, & \text{otherwise.} \end{cases}, \tag{4}$$

where n is the number of inputs to the OR network.

Definition 3.2: ($Sen(dc_i)$): The sensitivity of a domino cell dc_i can be formulated as

$$Sen(dc_i) = \frac{\sum_{in_j \in PI} SC(in_j)}{\#fanin}, \tag{5}$$

where $\#fanin$ is the number of fanins to the domino cell dc_i.

According to our observation, input signals which belong to the same OR network are pairwisely CT-immune to each other. Therefore, these fanins can be viewed as a single net. Thus, we give a weight of $\frac{1}{n}$ to each of them. For other input signals, they are crosstalk sensitive to each other and we give a weight of 1. Then, we sum up the sensitive contribution of each fanin net in the domino cell. Additionally, we divide the summation of $SC(in_j)$ by the fanin number of the domino cell to normalize the summation of $SC(in_j)$ for different sizes of cells.

Note that the $Sen(dc_i)$ of a cell is a fractional number greater than 0 and less than or equal to 1. A domino cell with

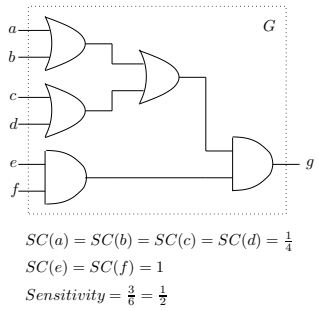

Fig. 2. Example of sensitivity.

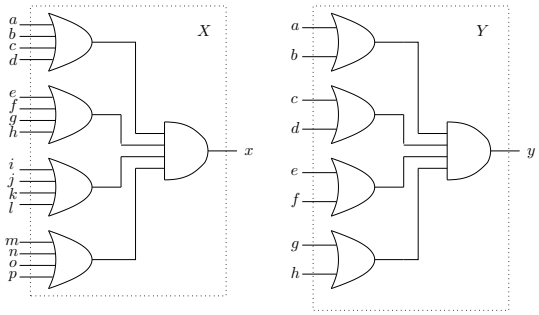

Fig. 3. Example of sensitivity measurement comparison.

large $Sen(dc_i)$ tends to be more sensitive to the crosstalk effect.

Taking the cell G in Figure 2 as an example, $Sen(G)$ is calculated as follows. Because the fanin signals a, b, c, and d belongs to the same OR network, the sensitive contribution $SC(a)$, $SC(b)$, $SC(c)$, and $SC(d)$ are $\frac{1}{4}$, respectively. For other signals, e, f, $SC(e)$ and $SC(f)$ are both 1. As a result, the crosstalk sensitivity of the domino cell G is $Sen(G) = \frac{3}{6} = \frac{1}{2}$.

To check if our metric is indeed a reasonable measurement, we compare the two measurements, 0-1 sensitivity matrix and our crosstalk sensitivity metric, for domino cells. We take Figure 3 as an example where the function of cell X is $X = (a+b+c+d)\cdot(e+f+g+h)\cdot(i+j+k+l)\cdot(m+n+o+p)$, and the function of cell Y is $Y = (a+b)\cdot(c+d)\cdot(e+f)\cdot(g+h)$. According to the aforementioned observations of OR immunity, the input wires of X can be divided into 4 CT-immune groups, $\{a,b,c,d\}$, $\{e,f,g,h\}$, $\{i,j,k,l\}$, and $\{m,n,o,p\}$. The input wires of Y can also be divided into 4 CT-immune groups, $\{a,b\}$, $\{c,d\}$, $\{e,f\}$, and $\{g,h\}$. Both X and Y have 4 CT-immune input groups. That means there are at least 3 pairs of inputs which are not CT-immune to each other. If the I/O pins is ordered alphabetically, the non-CT-immune pairs are $\{d,e\}$, $\{h,i\}$, and $\{l,m\}$ in X and $\{b,c\}$, $\{d,e\}$, and $\{f,g\}$ in Y.

Using the measurement of sensitivity matrix, there are 48 elements $(4*(4^2-4)=48)$ of 0 and 192 elements of 1 among the 240 elements $(16^2-16=240)$ of the sensitivity matrix of X. The ratio of 1 in the sensitivity matrix is $\frac{4}{5}$. Similarly, there are 48 elements $(4*(4^2-4)=48)$ of 0 and 8 elements of 1 among the 56 elements $(8^2-8=56)$ of the sensitivity matrix of Y. The ratio of 1 in the sensitivity matrix is $\frac{6}{7}$. From the above example, using the measurement of sensitivity matrix will conclude inaccurately that the crosstalk effect suffered in cell Y is worse than that in X. On the other hand, both $Sen(X)$ and $Sen(Y)$ are $\frac{1}{4}$ using our sensitivity metric which complies with the analysis of the two cells.

IV. Algorithm

A. Flow

Our crosstalk-aware domino logic synthesis flow is conducted in five steps as shown in Figure 4. First, we apply

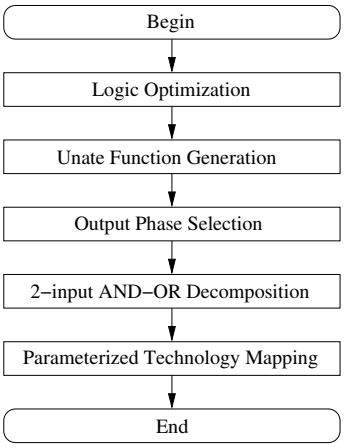

Fig. 4. Algorithm flow.

multi-level technology independent logic optimization using *SIS* [11] by a standard script [17]. After that, we transform binate functions into unate functions with minimized node duplications [8]. Based on our observations in Section III.B, we then select the output phases of the functions using DeMorgan's law to maximize the number of OR nodes. Additionally, the Boolean network is decomposed into 2-input AND nodes and 2-input OR nodes. Finally, a bottom-up parameterized technology mapping [12] [17] with crosstalk-aware consideration is performed. The details of step 3, *Output Phase Selection*, and step 5, *Parameterized Technology Mapping*, are described in the following.

B. Output phase selection

Before performing technology mapping, we optimize the Boolean network in order to have a better initial solution for technology mapping. Since the fanins of an OR function guarantee to be CT-immune to each other, we generate a Boolean network with more OR nodes. One straightforward method is to utilize the technique of output phase assignment to flip the AND nodes and the OR nodes of a Boolean network.

According to the DeMorgan's law, phase flipping on a target function can be done by three steps.

- 1. Divide the AND nodes and the OR nodes of that function into two different sets.

Procedure *Output Phase Selection* (PLA)
Begin
separate outputs into independent sets
compute the number of AND gates and OR gates
flip the output set which has more AND gates
End

Fig. 5. The *Output Phase Selection* procedure

- 2. Flip all OR nodes and AND nodes into AND nodes and OR nodes, respectively.
- 3. Add extra inverters at all primary inputs and primary outputs.

To decide if an output polarity is flipped or not, one simple algorithm is to compare the total numbers of AND nodes and OR nodes in the Boolean network. If the number of OR nodes is greater than the number of AND nodes, we keep the initial polarity of that output. Otherwise, we flip the output to generate an opposite-phase network. However, some outputs may share internal nodes, we have to duplicate the shared nodes if any two outputs are flipped into different phases.

Hence, to perform output phase selection, we group outputs, which have common internal nodes, into one set [18]. After grouping, all primary outputs are separated into several independent sets. The outputs within a set are dependent to each other; the sets themselves are independent. For those outputs in one set, their polarities must be assigned the same. This is to prevent network from duplicating extra nodes during output phase selection. Once the outputs are separated into different independent sets, we compare the number of AND nodes and the number of OR nodes in each output set. For those sets which have more AND nodes than OR nodes, we apply the aforementioned phase flipping technique to generate a complemented Boolean network. The procedure of output phase selection is drawn in Figure 5.

By means of this technique, we can represent the Boolean functions with more OR nodes than AND nodes without any duplication overhead. The time complexity of our output phase selection is, $O(X \cdot N)$, where X is the number of primary outputs and N is the total number of internal nodes.

C. Parameterized technology mapping

Technology mapping is the final step of logic synthesis to combine the optimized logic function with the process technology. Unlike traditional standard-cell CMOS design which uses a number of CMOS cells of a given cell library, the domino cell technology mapping synthesizes various types of cells on the fly. We use the parameterized technology mapping [17] to preserve the flexibility of implementing domino cells. The constraints in the mapping process are set to be the number of transistors in serial and the number of transistors in parallel for each mapped domino cell. To map a Boolean function into domino cells, the dynamic-programming-based bottom-up construction method [12] is used. Instead of minimizing area only, we introduce a new cost function taking sensitivity into consideration. The cost function of a domino cell, dc_i, is defined as,

$$Cost(dc_i) = \alpha \cdot Area(dc_i) + \beta \cdot Sen(dc_i) \ . \qquad (6)$$

In Equation (6), $Area(dc_i)$ represents the area of the domino cell in terms of the number of transistors. $Sen(dc_i)$ is the cell sensitivity discussed in Section III. Cell cost of a domino cell is a weighted sum of the area and sensitivity of the cell. The factors α and β are weighted coefficients to reflect the importance between circuit area size and crosstalk effect.

V. Experimental Results

The algorithm proposed in Section 4 is implemented as a software tool, *TM-C*, using C. The dynamic-programming-based domino technology mapping algorithm proposed by Zhao and Sapatnekar is also implemented to generate area minimized domino circuits [12]. Our experiment is performed on SUN-Blade2500 with 4 gigabytes of memory. The software platform is based upon *SIS* [11]. MCNC benchmark suite is used in our experiments. The experiment is conducted to generate a domino circuit with minimum crosstalk effect. The constraints of both the number of transistors in serial and the number of transistors in parallel for a domino cell are set to 4. All benchmark circuits are first technology independently optimized as initials using the standard script provided by *SIS* [11]. Then, the optimized networks are technology mapped by the algorithm proposed in [12] and by *TM-C*.

The experimental results are shown in Table II, and Table III. The column labeled *[12]* is the result of mapped domino circuits with minimized area. Column *TM-C* is the result of the circuits synthesized using our proposed flow, where both output phase selection and crosstalk-aware cost function for technology mapping are used.

In Table II, we compare the sensitivity of the circuits synthesized by the algorithm proposed in [12] and by *TM-C*. Column $R_{Sensitivity}$ is the ratio of average sensitivity of circuits synthesized by the algorithm proposed in [12] to that of circuits synthesized by *TM-C*. From Table II, we can see that the cell sensitivity is greatly reduced after using output phase selection and crosstalk-aware technology mapping cost function. In average, the cell sensitivity of the circuits synthesized by *TM-C* is reduced to 48.4% as compared to that of the circuits synthesized by the algorithm proposed in [12].

To understand if our synthesis tool can indeed reduce the crosstalk sensitivity in physical level, we perform placement and routing for all benchmark circuits. The experiment is conducted as follows. The networks are first technology mapped to domino cells by the algorithm proposed in [12] and by our mapper, *TM-C*. Then, *Dragon* [13] and *Multilevel Router* [14] are applied to perform cell placement and wire routing, respectively. Finally, pairs of adjacent wires are extracted to check their crosstalk immunity. Table III shows the result of crosstalk immunity. Column *Imm Adj* is the number of CT-immune wire pairs. Column *Tot Adj* is the

TABLE II

RESULT OF CELL SENSITIVITY

Circuit	*[12]*	*TM-C*	$R_{Sensitivity}$ (%)
C1355	91.90	55.44	60.32
dalu	96.27	34.04	35.35
C880	90.87	45.18	49.72
count	86.57	50.46	58.29
c8	92.50	47.00	50.81
C1908	86.21	54.27	62.95
C2670	91.93	40.86	44.45
C3540	93.82	28.99	30.90
C6288	86.03	67.35	78.29
b9	93.33	41.32	44.27
k2	93.36	30.05	32.18
des	95.02	31.40	33.05
C7552	91.51	55.22	60.34
t481	93.62	40.69	43.46
rot	93.47	39.38	42.13
average			48.43

TABLE III

RESULT OF CROSSTALK-IMMUNE WIRE PAIRS AFTER PLACEMENT AND ROUTING

Circuit	*Imm Adj*		*Tot Adj*		*Immunity (%)*	
	[12]	*TM-C*	*[12]*	*TM-C*	*[12]*	*TM-C*
C1355	70	211	797	806	8.78	26.18
dalu	54	240	1153	1331	4.68	18.03
C880	43	74	431	394	9.98	18.78
count	8	10	47	48	17.02	20.83
c8	2	13	37	39	5.41	33.33
C1908	154	280	1054	1039	14.61	26.95
C2670	104	438	1328	1385	7.83	31.62
C3540	320	1058	4817	4569	6.64	23.16
C6288	1588	2318	7608	6659	20.87	34.81
b9	6	13	50	50	12.00	26.00
k2	188	335	2698	2858	6.97	11.72
des	630	2955	8072	7883	7.80	37.49
C7552	603	1332	5519	5491	10.93	24.26
t481	65	237	897	1000	7.25	23.70
rot	50	167	696	684	7.18	24.42
average					9.86	25.42

number of total adjacent wire pairs. Column *Immunity* is the ratio of CT-immune wire pairs to the total number of adjacent wire pairs. We can see from the table, by means of using our crosstalk reduction techniques, the ratio of the number of CT-immune wire pairs to the number of total wire pairs is about 25% using our methodology as compared to 9% using area minimization algorithm. This demonstrates that our techniques are indeed effective.

VI. CONCLUSION

We have proposed a synthesis flow to minimize crosstalk for domino circuits. Output phase selection for independent sets is used to generate a Boolean network with more OR nodes. To precisely measure the crosstalk effect on domino cells, cell sensitivity is used as a metric for crosstalk minimization during technology mapping. The experimental results demonstrate that our synthesis methodology can greatly reduce the crosstalk effect by 51% in logic synthesis level as compared with conventional methodology. After placement and routing are performed, the ratio of the number of CT-immune wire pairs to the number of total wire pairs is about 25% using our methodology as compared to 9% using conventional techniques.

Acknowledgement

The authors would like to thank Prof. Yao-Wen Chang and Dr. Chung-Yi Ho for providing the source code of multi-level router and their valuable discussions.

REFERENCES

[1] A. Vittal, M. Marek-Sadowska, "Crosstalk reduction for VLSI", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, pp.290-298, vol.16, i.3, March 1997.

[2] D. Harris, T. Grutkowski, "Advanced Domino Circuit Design", Tutorial Note, *DATE*, 2004.

[3] Y. Im, K. Roy, "A Logic-aware Layout Methodology to Enhance the Noise Immunity of Domino Circuits", *ISCAS*, pp.637-640, 2003.

[4] T. Y. Ho, Y. W. Chang, S. J. Chen, D. T. Lee, "A Fast Crosstalk- and Performance-driven Multilevel Routing System", *ICCAD*, pp.382-387, Nov. 2003.

[5] K. W. Kim, S. O. Jung, U. Narayanan, C. L. Liu, S. M. Kang, "Noise-aware Interconnect Power Optimization in Domino Logic Synthesis", *IEEE Trans. Very Large Scale Integration Systems*, pp.79-89, vol.11, i.1, Feb. 2003.

[6] J. Lou, W. Chen, "Crosstalk-aware Placement", *IEEE Trans. Design and Test of Computers*, pp.24-32, vol.21, i.1, Jan.-Feb. 2004.

[7] H. Ren, D. Z. Pan, P. G. Villarrubia, "True Crosstalk Aware Incremental Placement with Noise Map", *ICCAD*, pp.402-409, Nov. 2004.

[8] R. Puri, A. Bjorksten, T. E. Rosser, "Logic Optimization by Output Phase Assignment in Dynamic Logic Synthesis", *ICCAD*, pp.2-8, Nov. 1996.

[9] M. Zhao, S. S. Sapatnekar, "Dual-monotonic Domino Gate Mapping and Optimal Output Phase Assignment of Domino Logic", *ISCAS*, pp.309-312, May. 2000.

[10] F. F. Sellers, M. Y. Hsiao, L. W. Bearnson, "Analyzing Errors with the Boolean Difference", *IEEE Trans. Computers*, pp.676-683, vol.C-17, no.7, July 1968.

[11] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, A. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis", *Electronics Research Laboratory*, Memorandum No. UCB/ERL M92/41, 4 May 1992.

[12] M. Zhao, S. S. Sapatnekar, "Technology Mapping for Domino Logic", *ICCAD*, pp.248-251, Nov. 1998.

[13] M. Wang, X. Yang, M. Sarrafzadeh, "Dragon2000: Standard-cell Placement Tool for Large Industry Circuits", *ICCAD*, pp.260-263, Nov. 2000.

[14] S. P. Lin, Y. W. Chang, "A Novel Framework for Multilevel Routing Considering Routability and Performance", *ICCAD*, pp.44-50, Nov. 2002.

[15] L. He, K. M. Lepak, "Simultaneous Shield Insertion and Net Ordering for Capacitive and Inductive Coupling Minimization", *ISPD*, pp.55-60, 2000

[16] D. A. Kirkpatrick, A. L. Sangiovanni-Vincentelli, "Digital Sensitivity: Predicting Signal Interaction Using Functional Analysis", *ICCAD*, pp.536-541, 1996

[17] M. R. Prasad, D. Kirkpatrick, R. K. Brayton, "Domino Logic Synthesis and Technology Mapping", *Intl. Workshop on Logic Synthesis(IWLS)*, 1997.

[18] K. W. Kim, S. M. Kang, "Crosstalk Noise Minimization in Domino Logic Design", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol.20, no.9, pp.1091-1100, Sep. 2001.