

## تقليل ذروة التيار عند اختبار الدوائر الرقمية التوافقية

محمد الفيلكاوي وإمتهياز أحمد

قسم هندسة الكمبيوتر، كلية العلوم وهندسة الحاسوب، جامعه الكويت، الكويت

### الخلاصة

في عصر التكنولوجيا ذات الكثافة العالية (VLSI)، تكتظ الملايين من الترانزستورات في رقيقة واحدة، فإن الطلب على طرق مناسبة لإدارة الطاقة في هذه الرقائق لها أهمية قصوى. إختبارات الفحص ذات الطاقة المنخفضة تتخذ مركز الصدارة في الأهمية بالنسبة لكل من الدوائر التوافقية والمتابعة في السنوات الأخيرة نظرا لتأثيرها علي سلامة الرقائق عند تصنيع. في هذا البحث، نقتراح طريقة جديدة تستهدف الحد من التيار العالي خلال اختبار الدوائر التوافقية لخفض ذروة استهلاك الطاقة. وعلى عكس من الطرق السابقة، هذه الطريقة تستخدم ذروة التيار التي تحددها اتجاه التيار لإيجاد الحد الأدنى من ذروة الطاقة. ويتكون إطار الإختبار المقترح من مرحلتين رئيسيتين حيث يتم إعادة ترتيب مجموعة اختبار باستخدام معيار يجمع ذروة التيار وذروة الطاقة مجتمعة ثم يتبعها استخدام تقنية خوارزمية (Fiduccia-Mattheyses) لإعادة ملء قيم المدخلات الغير محددة. النتائج التجريبية على مجموعة من دوائر (ISCAS85) التي يتم استخدامها لهذا النوع من الدراسات تبين أن الطريقة المقترحة تقلل من تيار الذروة وذروة الطاقة وإجمالي الطاقة بنسبة ٣٣٪، ٣٢٪، و ٤٣٪ على التوالي مقارنة مع طريقة ترتيب الإختبار بطريقة حساب المسافة والملء العشوائي للمدخلات الغير محددة.

# Minimizing peak current in combinational circuit test

Mohammad G. Al-Failakawi\* and Imtiaz Ahmad

*Computer Engineering Department, Kuwait University, Kuwait*

*\*Corresponding author: alfailakawi.m@ku.edu.kw*

## ABSTRACT

In the very large scale integration (VLSI) era, where millions of transistors are packed in a single chip, the demand for proper power management is of paramount importance. Low power test is taking center stage for both combinational and sequential circuits in recent years due to its impact on overall yield. This paper proposes a technique that targets the reduction of peak current during combinational circuit test to realize peak power reduction. Unlike previous methods, this approach utilizes peak current defined by the direction of switching activity to find minimum peak power. The proposed framework consists of two main phases, where the test set is first reordered using a combined peak current/peak power cost function followed by an x-refilling technique based on Fiduccia-Mattheyses algorithm to refill unspecified bit values. Experimental results on ISCAS85 benchmark circuits show that the proposed approach reduces peak current, peak power, and total power by 33%, 32%, and 43% respectively compared to Hamming distance-based ordering with random filling.

**Keywords:** Circuit testing; logic circuit; low power; optimization; vector ordering

## INTRODUCTION

With the continuous advancement in complementary metal oxide semiconductor (CMOS) technology, more and more transistors are packed on a single die resulting into chips with tens or hundreds of millions of transistors in a single package. When testing such large designs, peak and average power dissipation become important metrics to assess increased cost and reliability concerns. As the average power of the VLSI system increases, heat dissipated by such system increases. Thus, special packaging and cooling may be needed, which leads to increased system cost. Moreover, peak power demands an instantaneous access to power/ground rails at a very short period of time which results in IR-drop or ground bounce effects that could produce logic errors and/or reliability concerns such as electro-migration (Saxena *et al.*, 2003). It is customary during the design phase to analyze peak and average power and design the system appropriately to meet functional operation of such systems. However, during testing, values for peak and average power change drastically and can severely affect system performance and reliability.

In (Li *et al.*, 2001), it was shown that power consumption of a circuit during test mode is considerably higher, i.e. 100-200% higher than during the normal mode of operation. It has been observed that test efficiency correlates with toggle rate; thus switching activity during testing is significantly higher than normal mode of operation (Girard, 2002). Another cause for increased power consumption during testing can be attributed to the fact that consecutive functional input vectors have high correlation between them, which is not the case for consecutive test vectors (Girard, 2002).

Test vector ordering has been considered by researchers as an effective method to reduce both average and peak power consumption during test (Girard *et al.*, 1997; Badereddine *et al.*, 2006; Flores *et al.*, 1999; Chattopadhyay & Choudhary, 2003; Hashempour & Lombardi, 2008; Sokolov *et al.*, 2005). The methods used in test vector ordering can be divided into two major groups; one group that uses minimum Hamming distances (Girard *et al.*, 1997), while the other performs ordering based on circuit switching activities (Chattopadhyay & Choudhary, 2003; Hashempour & Lombardi, 2008; Sokolov *et al.*, 2005; Paramasivam & Gunavathi, 2007). Previous works have empirically proved that minimal Hamming distance between test vectors translates into a lower switching activity; thus, one can reduce power consumption when compared to random ordering (Girard *et al.*, 1997; Badereddine *et al.*, 2006; Flores *et al.*, 1999). Moreover, approaches that utilize switching activity have been shown to provide better results compared to the Hamming distance approach, but they require the availability of internal details of the circuit-under-test (CUT) (Hashempour & Lombardi, 2008).

One issue that must be addressed when performing test vector ordering is how to handle unspecified bits (don't care bits). Automatic test pattern generators (ATPG) normally generate partially specified test vectors, i.e. vectors with 0, 1, and x, where the value 'x' stands for 'don't care' bits. Many studies have utilized don't care bits to reduce power by specifying them to a value that reduces number of transitions in the circuit-under-test (Badereddine *et al.*, 2006; Flores *et al.*, 1999; Chattopadhyay & Choudhary, 2003; Paramasivam & Gunavathi, 2007; Wang *et al.*, 2007; Li *et al.*, 2005; Maiti & Chattopadhyay, 2008). Thus, for any vector ordering problem, it is important to target both test ordering as well as x-filling to realize true low power test. Authors have exploited don't care bits for different objectives, which include minimizing Hamming distance among test vectors for a lesser number of transitions (Flores *et al.*, 1999; Chattopadhyay & Choudhary, 2003), reduce shift power in scan design (Badereddine *et al.*, 2006; Wang *et al.*, 2007; Li *et al.*, 2005; Eggersglub, 2014; Trinadh *et al.*, 2014), reducing peak temperature (Dutta *et al.*, 2013), improving reliability (Feng *et al.*, 2014) or reducing leakage power (Paramasivam & Gunavathi, 2007; Wang *et al.*, 2007; Maiti & Chattopadhyay, 2008).

Power consumption of a CMOS circuit is highly dependent on the amount of

switching activity. However, only minimizing the switching activity may not be a good indicator of peak currents. In recent work (Huang *et al.*, 2006; Huang *et al.*, 2009; Lee & Kim, 2011; Gu *et al.*, 2010; Wang *et al.*, 2013; Borowczak & Vemuri, 2014), researchers have shown the importance of peak current minimization and its strong correlation with switching activity direction. In (Huang *et al.*, 2006; Huang *et al.*, 2009), authors have shown that the direction of the transitions play a major role in determining true peak power, since peak power is related to peak current. In their definition, authors (Huang *et al.*, 2006; Huang *et al.*, 2009) have used maximum transitions in a given direction commonly referred to as peak current to represent peak power. Using the definition of peak current, the authors proved that although the switching activity in the circuit may be the same, different combinations of switching directions still result in different peak currents.

While many techniques have addressed the problem of reducing peak power between two consecutive test vectors (or vector pairs), none have addressed the problem of reducing peak current by considering the direction of these transitions which was shown to correlate to actual peak power (Huang *et al.*, 2006; Huang *et al.*, 2009; Lee & Kim, 2011; Gu *et al.*, 2010; Wang *et al.*, 2013; Borowczak & Vemuri, 2014). In this work, we propose a framework for test vector ordering and x-filling to minimize peak current for combinational circuit test. The methodology aims to reduce both peak and total power by utilizing minimum peak current ordering and FM-algorithm based x-refilling (Fiduccia & Mattheyses, 1982). Experimental results comparing the proposed approach and traditionally available concepts have shown the effectiveness of the proposed framework.

## PEAK POWER MINIMIZATION

In order to understand the various power issues in low power testing, it is important to have proper understanding of various terms commonly used in power related topics. The following is the power terminology that will be used throughout the paper:

*Energy*: total switching activity generated during test application.

*Average power*: ratio between energy and test time.

*Instantaneous power*: power dissipated at a given instant of time.

*Peak power*: highest value of instantaneous power for a test set.

*Peak current*: maximum current value for a test set.

The major source of power consumption in CMOS technology is dynamic power, which is attributed to the charging and discharging of load capacitances (Girard, 2002; Chandrakasan *et al.*, 1992; Wang *et al.*, 2007). The energy/power consumed at node x, when it undergoes a transition can be expressed as (Saxena *et al.*, 2003):

$$E_t = \frac{1}{2} \times C_0 \times V_{dd}^2 \quad (1)$$

Where  $C_0$  is output capacitance of node  $x$ , and  $V_{dd}$  is power supply voltage. This equation can be approximated using the number of transitions on node  $x$  as:

$$E_x = E_t \times SA_x \quad (2)$$

Where  $SA_x$  is the number of transitions node  $x$  experienced over a period of time. The term  $SA_x$  is the only variable part in equation (2), and it will be used to estimate the power consumed at node  $x$ . Therefore, total power consumed by a test vector pair  $(v_a, v_b)$  can be expressed as:

$$SA_{(v_a, v_b)} = \sum_{x=1}^N Node(x, v_a) \oplus Node(x, v_b) \quad (3)$$

Where  $N$  represents the number of nodes in the CUT, and  $Node(x, v_a)$  represents the value of node  $x$  when vector  $v_a$  is applied. In order to find the total power consumed by the complete test set, we need to calculate the power for all test vector pairs in the test set resulting in the following expression:

$$SA_{total} = \sum_{i=1}^{k-1} SA(v_i, v_{i+1}) \quad (4)$$

Where  $k$  represents the number of test vectors in the test set. It is common practice to use weighted switching activity (WSA) to calculate total power (Wang *et al.*, 2007). This is due to the fact that node fan-outs increase the capacitance of various nodes, which in turn affects the power. When such effects are of interest and to be taken into account, equation (4) must be modified by multiplying the  $SA_x$  term by a factor that represents the node's fan-outs (i.e. fan-out +1). Previous work in (Huang *et al.*, 2009) has shown that the impact of fan-out capacitances on peak current is minimal (i.e. 3% for 3x increase in capacitance), and since this work investigates peak current, we decided not to use the WSA model. Instead we utilized the simplified switching activity ( $E_x$ ) model described above for the remainder of this discussion.

Peak power is defined as the worst case transitions a CUT undergoes after the application of a test set. Therefore, peak power can be calculated using equation (3) as follows:

$$SA_{peak} = \max[SA(v_i, v_{i+1})]_{i=1}^{k-1} \quad (5)$$

Many previous works targeted the minimization of  $SA_{peak}$ , which corresponds to the number of transitions for worst case vector pairs. In (Huang *et al.*, 2006; Huang *et al.*, 2009), the authors have shown that the amount of current drawn from supply voltage ( $V_{cc}$ ) and sunk to ground is different, depending on the transition type. For example, they have shown that a 0 to 1 transition on the output of D flip/flop draws

approximately two times (2x) the current from the power supply, compared to that sunk to ground (in the worst case). Alternately, a 1 to 0 transition would result in current sunk to ground in excess of 2.5x when compared to that drawn from  $V_{cc}$ . Even though the experiments conducted by (Huang *et al.*, 2006) consider only transitions on flip/flop outputs, the same concept can be generalized to any logic gate. This means that the direction of switching activity plays a major role in the amount of current consumed by the CUT; thus, minimizing the number of transitions only does not guarantee true minimum peak power.

In this work, the maximum number of transitions in the same direction is used to estimate peak current. Therefore, to find peak current, the maximum up and down transitions for every vector pair must be calculated. The up and down transitions can be calculated for a test vector  $(v_a, v_b)$  using equations (6) and (7) respectively as follows:

$$RT(v_a, v_b) = \sum_{x=1}^N RT_x \quad (6)$$

$$FT(v_a, v_b) = \sum_{x=1}^N FT_x \quad (7)$$

where  $RT_x$  and  $FT_x$  in equations 6 and 7 refer to up and down transitions on node  $x$ , respectively. Therefore, peak current for a test vector pair  $(v_a, v_b)$  can be found as follows:

$$Dir_{peak} = \max[RT(v_i, v_{i+1}), FT(v_i, v_{i+1})]_{i=1}^{k-1} \quad (8)$$

In order to minimize peak power, all previous work has targeted minimizing  $SA_{peak}$  term defined in equation (5); however, as described above, this does not necessarily result in minimum peak current. Consider, for example, the circuit shown in Figure 1:

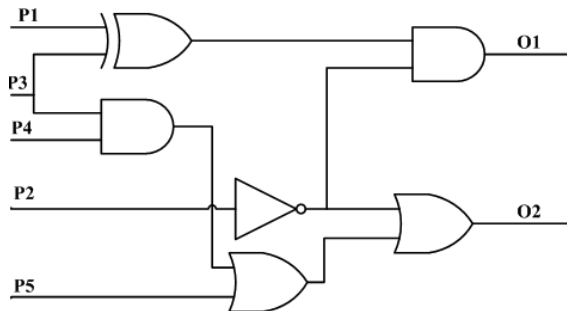


Fig. 1. Example of CUT.

Table 1 gives different test set orderings for such a circuit. The first two columns give original ordering as generated by ATPG. Columns HD, SA, and Dir give possible ordering based on Hamming distance, switching activity, and peak current respectively. For every ordered test set, a vector number is given to represent the number of the vector in the original unordered test set. The last three rows of Table 1 summarize power statistics of the various orderings. It can be seen from the results that peak current is reduced, when using the peak current concept compared to the other approaches, even for SA based ordering. For example, in columns SA and Dir, both approaches have the same number of peak transitions (i.e. 3). However, Dir approach has lower peak current (2 transitions in same direction for Dir compared to 3 for SA approach). This example demonstrates that even when peak transitions are the same, there is room to reduce peak current further.

**Table 1.** Example CUT test set.

Unordered test set		Ordered test set		
Vector no	Test vector	HD	SA	Dir
1	00001	9	9	9
2	01110	1	1	1
3	10011	3	5	5
4	11001	6	3	8
5	00010	7	4	6
6	11010	4	2	7
7	11000	8	6	4
8	01100	2	7	2
9	10101	5	8	3
Peak current		4	3	2
Peak power		4	3	3
Total		17	10	10

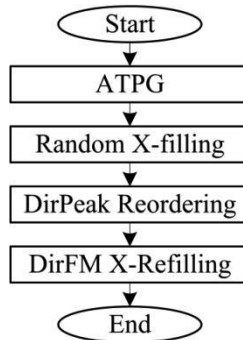
Almost all ATPG tools available today generate test sets with don't care values. Such don't care bits are normally specified in such a way to reduce power. Many approaches integrate x-filling strategy with their re-ordering algorithm to find the best solution for low power test. In this work, we follow a similar approach by utilizing don't care bits to reduce peak current by performing x-filling as a post-processing step. Our x-refilling algorithm is based on Fiduccia-Mattheyses (FM) algorithm (Fiduccia & Mattheyses, 1982) and is described in section 3. FM algorithm is a partitioning

algorithm based on the well-known KL algorithm (Kernighan & Lin, 1970) which is commonly used in the computer-aided design of digital systems, due to its simplicity and efficiency. KL algorithm belongs to a class of iterative improvement algorithms, where it initially starts with an initial partition and then moves nodes between partitions to improve partitioning. The process is repeated until no further improvement is possible. Fudduccia and Mattheyses developed an efficient variant of KL algorithm in which only a single vertex is moved across the cut in a single move by utilizing a better data structure.

FM algorithm has been used for the reduction of leakage (static) power in testing for nanometer technologies (Maiti & Chattopadhyay, 2008; Kao *et al.*, 2010) in addition to many other applications. In this work, we employed the FM algorithm to reduce peak current by assigning “don’t care” bits with suitable values of 0 or 1. Each  $x$  in a test vector was considered a node. Based on the initial random assignment value, all nodes associated with don’t care bits are placed in two partitions: 0-partition and 1-partition based on their initial fill value. Given this initial partition of nodes, we utilize the FM algorithm to reassign values to don’t care bits by moving across the partition in order to reduce peak current.

## PEAK CURRENT MINIMIZATION FRAMEWORK

In this section we describe the proposed framework for minimizing peak current in combination circuit test. The proposed peak current minimization framework consists of two phases, one targeting test vector ordering, and the other is x-refilling phase. Figure 2 gives the complete flow of the proposed framework.



**Fig. 2.** Proposed low current framework.

The proposed methodology starts by generating a test set for the CUT using a combination circuit ATPG. Next, all don’t care bits in the unordered test set were randomly filled then ordered using DirPeak algorithm that minimizes peak current. After that, the initial randomly filled don’t care bits were refilled using our proposed



DirFM algorithm, which is a FM based algorithm to perform x-refilling to minimize peak current. Even though Figure 2 shows the complete framework in which algorithms DirPeak and DirFM are used, these algorithms can be integrated with any other test flow since they are completely independent.

The problem of test vector reordering can be formulated as a completely connected graph, where every node represents a test vector and is connected to another node by an undirected edge. The weight on the edge represents the cost of applying the two test vectors one after the other (i.e. a vector pair). For example, if node 1 is connected to node 2 through an edge, then the edge represents the cost of applying a test vector pair  $(v_1, v_2)$  or  $(v_2, v_1)$ . In our implementation, edge weight is a function of the number of switching activity and peak current of the vector pair. Using this graph representation, finding a test vector ordering that reduces peak current can be formulated by finding a Hamiltonian path of minimum cost (Girard *et al.*, 1997; Badereddine *et al.*, 2006; Flores *et al.*, 1999; Chattopadhyay & Choudhary, 2003; Hashempour & Lombardi, 2008; Sokolov *et al.*, 2005). This is equivalent to the well-known travelling salesman problem and is considered as a NP-complete problem. Therefore, it is a common practice to use heuristics to find a solution to such a problem (Hashempour & Lombardi, 2008; Sokolov *et al.*, 2005). In our approach, we use greedy algorithm (DirPeak) to solve the test vector ordering problem.

Algorithm DirPeak is shown in Figure 3. The algorithm starts first by computing the number of transitions and peak current for every test vector pair according to equations (4) and (5), respectively. Second, it adds these vectors to an unordered list and picks a root randomly as the starting point of the Hamiltonian path (i.e. first\_min) and removes it from the unordered list. Third, the algorithm picks a vector with minimum edge cost with respect to the root from the unordered list, which is minimum in peak power (i.e. equation 5) with minimum transitions in the same direction (i.e equation 8). Last, the newly selected vector becomes first\_min while second\_min is deleted from the unordered list, and the process is repeated again. The algorithm will continue execution until the unordered list becomes empty. In step 6a, the algorithm finds a vector with minimum switching activity, which has a complexity  $O(n)$  for each iteration of the “while” loop. Since the loop is repeated  $n-1$  times, this results in an  $O(n^2)$  complexity of this loop. Step 6 has the worst case complexity, and the overall complexity of algorithm DirPeak is  $O(n^2)$ , where  $n$  is the number of test vectors in the test set.

## DirPeak Algorithm:

1. Compute total and peak transitions between every vector pairs;
2. `unordered_list := list of all test vectors;`
3. `first_min := randomly selected test vector;`
4. Add `first_min` to `ordered_list`;
5. Delete `first_min` from `unordered_list`;
6. while (`unordered_list.length != 0`) do {
  - a. `set_min := list of vectors with minimum transitions with respect to first_min;`
  - b. `second_min := a vector from set_min with minimum peak current with respect to first_min;`
  - c. Add `second_min` to `ordered_list`;
  - d. `first_min := second_min;`
  - e. Delete `second_min` from `unordered_list`;

Fig. 3. DirPeak algorithm.

It was mentioned earlier that many algorithms utilize don't care bits in test sets to reduce power. In our framework, we initially filled these don't care bits randomly and then reorder the test set; then, we analyze these random assignments for power improvement and modify their assignments as needed, to reduce peak current. The proposed x-refilling algorithm based on FM algorithm, also called DirFM, is given in Figure 4.

## DirFM Algorithm:

For each node with "X" value do {

1. `peak_sa_1 := Compute peak switching activity;`
2. `peak_sw_s_dir_1 := Compute peak switching with same direction;`
3. `total_sa_1 := Compute total switching activity;`
4. Tentatively move node from its partition to another partition;
5. `peak_sa_2 := Compute peak switching activity;`
6. `peak_sw_s_dir_2 := Compute peak switching with same direction;`
7. `total_sa_2 := Compute total switching activity;`
8. `cost := calculate_cost();`
9. if (`cost < 1`) {
  - a. Make tentative movement permanent;
  - b. `peak_sa_1 := peak_sa_2;`
  - c. `peak_sw_s_dir_1 := peak_sw_s_dir_2;`
  - d. `total_sa_1 := total_sa_2;`
  - e. Lock this node;
10. else {
  - a. Undo movement of this node;
  - b. Lock this node;

Fig. 4. DirFM algorithm.

The algorithm starts with calculating total, peak power, and peak current for all vector pairs and creating two partitions: one containing all x-values that were filled with 0, and the other one containing those that were filled with 1. The FM algorithm is started by selecting a member in one partition and tentatively moving it to the other partition (i.e. changing bit value to its complement). Then, cost function is calculated for this new assignment and is compared to that of the original assignment. If the cost function is less than 1, then the new assignment is better in terms of power and the bit is permanently moved to the new partition. Alternatively, if the cost function value is greater than 1; then, power is increased, and the tentative move is undone. In either case, the bit is flagged to inhibit future movement (i.e. locked into the partition). The cost function used in DirFM algorithm is as follows:

$$Cost = \frac{1}{3} * \frac{SA_{peak2}}{SA_{peak1}} + \frac{1}{3} * \frac{dir_{peak2}}{dir_{peak1}} + \frac{1}{3} * \frac{SA_{total2}}{SA_{total1}} \quad (9)$$

To clarify DirFM refilling heuristic, Figure 5 shows a graphical representation of this heuristic. Figure 5 assumes that four don't care bits that were initially filled at random. Bits that were filled with 1 are placed in partition '1', whereas those filled with '0' are placed in partition '0'. Assuming the algorithm chooses bit 2 for refilling, the heuristic starts by tentatively moving it from partition '0' to partition '1'. After calculating the cost function, the cost function of the new assignment was found to be less than 1, thereby making the move permanent. Node 2 is locked and a new node is selected repeating the process again. The complexity of algorithm DirFM is linear with respect to the number of don't care bits in the test set, i.e.  $m$ ; hence, it is an  $O(m)$  complexity algorithm.

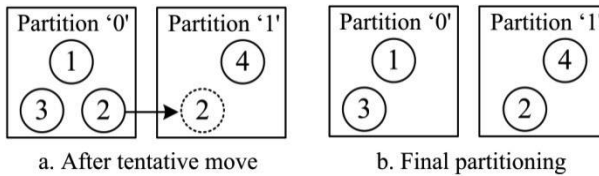


Fig. 5. FM example

## EXPERIMENTAL RESULTS

To validate the proposed algorithms, we compare the results of the proposed ordering algorithm to that of commonly used algorithms such as those that are based on Hamming distance and switching activity. In the discussion that follows, we compare the performance of the proposed algorithm to these two different approaches.

Three different ordering algorithms were implemented and they are as follows:

*MinHD*: An algorithm that utilizes minimum Hamming distance between vector pairs to find best order.

*MinSA*: An algorithm that uses peak switching activity (i.e peak power) between vector pairs to find best order.

*DirPeak*: The proposed algorithm that utilizes both, peak power and peak current to find best order.

All algorithms were implemented using Java and were run on a PC with 1.8 GHz Intel DuoCore processor with 2 GB RAM. Each algorithm uses equations (4) and (5) to calculate peak and switching activity of the test set. All algorithms were initially run without enabling the DirFM x-refilling step to analyze the effectiveness of ordering algorithm DirPeak. Then DirFM refilling algorithm was enabled for all three algorithms and the impact of the proposed x-refilling strategy is discussed. All algorithms were run on ISCAS85 benchmark circuits, and the test set was generated by ATALANTA ATPG (Lee & Ha, 1993). The characteristics of the ISCAS85 benchmark circuits are highlighted in Table 2.

**Table 2.** ISCAS85 benchmark circuits.

Circuit	Ordered test set					
	# PIs	# POs	# Gates	# Patterns	Fault coverage (%)	# Xs (%)
C17	5	2	6	10	100	38
C432	36	7	160	96	99.046	55.61
C499	41	32	202	57	96.306	3.93
C880	60	26	383	312	100	80.69
C1355	41	32	546	102	99.492	1.22
C1908	33	25	880	197	99.521	42.70
C2670	233	140	1193	606	95.741	93.73
C3540	50	22	1669	508	96.004	73.78
C5315	178	123	2307	1014	98.860	93.76
C6288	32	32	2416	53	98.954	12.50
C7552	207	108	3512	662	97.629	82.89

To analyze the performance of the proposed algorithm, algorithms MinHD, MinSA, and DirPeak were run on the unordered test set generated by ATALANTA using a random root and randomly filled. The results are shown in Table 3. Note that in both Table 3 and Table 4, the last row gives the average values. The results are given using three columns for each algorithm, where columns “Dir”, “peak”, and “tot” represent peak current, peak power, and total transitions respectively. Peak current here refers to the maximum transitions in a given direction. Moreover, only percent reductions in these values are given for algorithms MinSA and DirPeak (i.e. % columns) to ease

the analysis of the results. The percent reduction is calculated by using the values generated by algorithm MinHD as a base value.

In Table 3, the first observation is that algorithm DirPeak outperforms MinHD algorithm, when it comes to reducing peak current and peak power on average. DirPeak algorithm was able to reduce peak current and peak power by 22% and 20%, respectively. Moreover, the proposed algorithm was able to reduce total power on average by 25% compared to Hamming distance due to the fact that the Hamming distance approach does not take into account internal circuit transitions.

The second observation is that algorithm DirPeak in most instances was able to further reduce peak current and peak power found by MinSA algorithm. Table 3 shows an additional 2-3% of approximate reduction can be gained in peak current and power, when comparing the performance of DirPeak algorithm to MinSA. This demonstrates that there is some additional reduction in peak current by including switching direction in the cost function of the reordering algorithm as compared to only considering the switching activity, as is the case in MinSA.

**Table 3.** Peak current comparison.

Circuit	MinHD			MinSA (%)			DirPeak (%)		
	Dir.	Peak	Tot.	Dir.	Peak	Tot.	Dir.	Peak	Tot.
C17	3	5	16	66.7	60	50	66.7	60	50
C432	41	69	2995	24.4	10.1	34.2	24.4	15.9	33.6
C499	46	74	2173	21.7	29.7	42.5	21.7	29.7	42.5
C880	114	190	34858	29.8	25.3	28	34.2	25.3	27.7
C1355	92	184	8148	17.4	19	14.9	29.3	29.3	14.5
C1908	207	413	45595	9.2	14.3	15.6	15.5	18.4	14.9
C2670	308	565	239038	12.3	8.1	25	7.5	7.8	24.8
C3540	369	673	229005	16	9.2	32.4	26.6	20.5	32.5
C5315	538	1032	821018	6.5	2.6	17	8	9.1	16.8
C6288	535	1052	28605	1.5	0	7.4	1.5	0	7.4
C7552	894	1719	738900	11.4	8.7	18.1	9.8	7.4	18.1
<b>Average</b>				<b>19.7</b>	<b>17</b>	<b>25.9</b>	<b>22.3</b>	<b>20.3</b>	<b>25.7</b>

We analyzed the impact of using the proposed x-refilling approach as a post processing step to the three previously defined algorithms. Table 4 shows the results of applying DirFM on the ordered test set generated by algorithms MinHD, MinSA, and DirPeak. Note that DirFM algorithm only modifies don't care values that were initially filled randomly, and it does not change the order of the test set generated by these algorithms. All columns in Table 4 are given as percent reduction, compared to the base case of algorithm MinHD with no refilling heuristic (i.e. no DirFM).

From Table 4, it can be seen that utilizing the proposed DirFM algorithm for refilling don't care bits results in an improvement in almost all power values for all algorithms. DirFM algorithm was able to further reduce peak current values of MinHD, MinSA, and DirPeak algorithms by 12%, 10%, and 11%, respectively, when compared to the initial forms of these algorithms. A similar observation can be made with respect to peak and total power values as well (reduction compared to no DirFM counter parts is approximately 10-17%). Since all algorithms benefit from using algorithm DirFM, this shows the importance of the x-filling technique in providing further reduction in peak current/power. It also demonstrates the importance of using the concept of peak current in x-filling strategies used in low power test.

**Table 4.** DirFM X-filling impact.

Circuit	MinHD-DirFM (%)			MinSA-DirFM (%)			DirPeak-DirFM (%)		
	Dir.	Peak	Tot.	Dir.	Peak	Tot.	Dir.	Peak	Tot.
C17	0	20	0	66.7	60	50	66.7	60	50
C432	14.6	7.2	22.1	39	29	52.7	41.5	37.7	51.9
C499	0	0	0	21.7	29.7	42.5	21.7	29.7	42.5
C880	31.6	22.6	44.7	42.1	34.7	52	42.1	37.9	52.2
C1355	0	0	0	17.4	19	14.9	29.3	29.3	14.5
C1908	10.1	14.5	25.3	14	15.3	29.6	26.6	29.5	30
C2670	30.5	28.3	49.5	26.9	30.1	55.7	38	38.1	56.1
C3540	10.8	4.5	24.6	14.6	8	50	27.9	21.8	50.4
C5315	13	16.2	46.6	29.4	29.8	51.2	32.3	30.9	51.5
C6288	10.3	11.6	12	11.6	12.4	20.4	11.6	12.4	20.4
C7552	14.2	16.3	43.7	36.4	36.6	53.5	33.9	32.6	53.7
<b>Average</b>	<b>12.3</b>	<b>12.8</b>	<b>24.4</b>	<b>29.1</b>	<b>27.7</b>	<b>43</b>	<b>33.8</b>	<b>32.7</b>	<b>43</b>

## CONCLUSION

In this work, we proposed a framework to minimize peak current in a combination circuit test. The approach was shown to be effective in reducing peak current, when compared to other algorithms that do not consider switching direction in their cost function. Experimental results on ISCAS85 benchmark circuits showed reduction in peak current, peak, and total power values of 33%, 32%, and 43% respectively, compared to Hamming distance-based ordering with random filling. Even though the approach proposed in this work is for combinational circuits, the concept can be extended to scan-based design.

## ACKNOWLEDGEMENT

This work was supported by Kuwait University under Research Grant no. EO 02/08.

## REFERENCES

- Badereddine, N., Girard, P., Pravossoudovitch, S., Landrault, C., Virazel, A. & Wunderlich, H. 2006.** Minimizing peak power consumption during scan testing: test pattern modification with x filling heuristics. In Proceedings of International Conference on Design and Test of Integrated Systems in Nanoscale Technology. pp. 359-364.
- Borowczak, M. & Vemuri, R. 2014.** Enabling side channel secure FSMs in the presence of low power requirements. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI. pp. 232-235.
- Chandrakasan, A., Sheng, T. & Brodersen, R. 1992.** Low power CMOS digital design. *IEEE Journal of Solid State Circuits* **27**(4): 473-484.
- Chattopadhyay, S. & Choudhary, N. 2003.** Genetic algorithm based approach for low power combinational circuit testing. In Proceedings of the 16<sup>th</sup> IEEE International Conference on VLSI Design. pp. 552-557.
- Dutta, A., Kundu, S. & Chattopadhyay, S. 2013.** Thermal aware don't care filling to reduce peak temperature and thermal variance during testing. In Proceedings of the 22<sup>nd</sup> Asian Test Symposium. pp. 25-30.
- Eggersglub, S. 2014.** Dynamic x-filling for peak capture power reduction for compact test sets. *Journal of Electronic Testing* **30**: 557-567.
- Feng, Z., Jing, N. & He, L. 2014.** IPF: In-place x-filling algorithm for the reliability of modern FPGAs. *IEEE Transactions on VLSI Systems* **22**(10): 2225-2228.
- Fiduccia, C. M. & Mattheyses, R. M. 1982.** A linear-time heuristic for improving network partitions. In Proceedings of the 19<sup>th</sup> ACM/IEEE Design Automation Conference. pp. 175-181.
- Flores, P., Costa, J., Neto, H., Monteiro, J. & Marques-Silva, J. 1999.** Assignment and reordering of incompletely specified pattern sequences targeting minimum power dissipation. In Proceedings of the 12<sup>th</sup> IEEE International Conference on VLSI Design. pp. 37-41.
- Girard, P. 2002.** Survey of low-power testing of VLSI circuits. *IEEE Design and Test of Computers* **19**(3): 82-92.
- Girard, P., Landrault, C., Pravossoudovitch, S. & Severac, D. 1997.** Reduction of power consumption during test application by test vector ordering. *Electronics Letters* **33**(21): 1752-1754.
- Gu, J., Qu, G., Yuan, L. & Zhou, Q. 2010.** Peak current reduction by simultaneous state replication and re-encoding. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 592-595.
- Hashempour, H. & Lombardi, F. 2008.** Evaluation and analysis of heuristic techniques for vector ordering of VLSI test sets. *IEEE Transactions on Instrumentation and Measurement* **57**(9): 1998-2004.
- Huang, S., Chang, C. & Nieh, Y. 2006.** State re-encoding for peak current minimization. In Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 33-38.
- Huang, S., Chang, C. & Nieh, Y. 2009.** Opposite-phase register switching for peak current minimization. *ACM Transactions on Design Automation of Electronic Systems* **14**(1): Article 14.
- Kao, W. C., Chuang, W.S., Lin, H. T., Li, J. C. & Manquinho, V. 2010.** DFT and minimum leakage pattern generation for static power reduction during test and burn-in. *IEEE Transactions on VLSI Systems* **18**(3): 392-400.

- Kernighan, B. & Lin, S. 1970.** An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal* **49**: 291-307.
- Lee, H. & Ha, D. 1993.** On the generation of test patterns for combinational circuits. Tech. Rep. 12-93, Dept. of Electrical Eng., Virginia Polytechnic Institute and State University.
- Lee, Y. & Kim, T. 2011.** State encoding algorithm for peak current minimization. *IET Computers and Digital Techniques* **5**(2): 113-122.
- Li, W., Reddy, S. M. & Pomeranz, I. 2005.** On reducing peak current and power during test. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*. pp. 156-161.
- Li, X., Li, H. & Min, Y. 2001.** Reducing power dissipation during at-speed test application. In *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*. pp. 116-121.
- Maiti, T. K. & Chattopadhyay, S. 2008.** Don't care filling for power minimization in VLSI circuit testing. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. pp. 2367-2640.
- Paramasivam, K. & Gunavathi, K. 2007.** Switching activity based method for minimizing testing power in digital circuits. *ECTI Transactions on Electrical Engineering, Electronics, and Communications* **5**(1): 61-69.
- Saxena, J., Butler, K.M., Jayaram, V.B., Kundu, S., Arvind, N.V., Sreepakash, P. & Hachinger, M. 2003.** A case study of IR-drop in structured at-speed testing. In *Proceedings of the IEEE International Test Conference*. pp. 1098-1104.
- Sokolov, A., Sanyal, A., Whitley, D. & Malaiya, Y. 2005.** Dynamic power minimization during combinational circuit testing as a traveling salesman problem. In *Proceedings of IEEE Congress on Evolutionary Computation*. pp. 1088-1095.
- Trinadh, A. S., Potluri, S., Balachandran, S., Babu, C. S. & Kamakoti, V. 2014.** XStat: Statistical x-filling algorithm for peak capture power reduction in scan tests. *Journal of Low Power Electronics* **10**(1): 107-115.
- Wang, L. Y., Chu, Z. F. & Xia, Y. S. 2013.** Low power state assignment for FSMs considering peak current optimization. *Journal of Computer Science and Technology* **28**(6): 1054-1062.
- Wang, S., Chen, Y. & Li, K. 2007.** Low capture power test generation for launch-off-capture transition test based on don't-care filling. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. pp. 3683-3686.
- Wang, W., Hu, Y., Han, Y., Li, X. & Zhang, Y. 2007.** Leakage current optimization techniques during test based on don't care bits assignment. *Journal of Computer Science and Technology* **22**(5): 673-680.

**Open Access:** This article is distributed under the terms of the Creative Commons Attribution License (CC-BY 4.0) which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

*Submitted:* 15-12-2014

*Revised:* 2-3-2015

*Accepted:* 10-3-2015