Minimizing Power Dissipation in Combinational Circuits During Test Application*

V. P. Dabholkar S. Chakravarty
Department of Computer Science
State University of New York
Buffalo, NY 14260
TECH REPORT NO 94-10

Abstract

Yield, Reliability and Power Supply considerations motivate the need to minimize power dissipation during test application. Two techniques for minimizing power dissipation when tests are applied to static CMOS combinational circuits are proposed. They are: (i) Test set ordering; and (ii) Repetition of test vectors. We show that: although (i) is NP-Hard good heuristics can be developed; and an optimal polynomial time algorithm exists for (ii). Experimental evaluation of these algorithms show that considerable improvement in power dissipation can be achieved using these techniques.

1 Introduction

Growing size of VLSI circuits, along with the high transistor density, is making minimization of power dissipation an important issue in VLSI design. Power dissipation issues have been addressed at various stages of circuit design like synthesis[1, 2], technology mapping[3, 4] etc.

Power dissipation issue must also be addressed during testing. Circuits are often designed to operate in two modes: *normal mode*; and *test mode*. The activation of embedded test structures during test mode can make it possible for system registers to be set to encodings of unreachable states. In addition, state transitions which are not possible during normal mode are often possible during test mode. Thus more power is dissipated during testing thereby subjecting the chip to additional stress during test mode. This in turn can reduce yield, prior to packaging, and adversely affect the reliability of systems that use periodic testing.

Battery powered systems are finding widespread use in portable communication and computing equipments. Excessive power dissipation during testing could prevent field testing of such equipments. In addition, for such battery powered systems that can provide the required power, but require periodic field testing, the power supply will drain much faster thereby adding to its down time. This can severely restrict the utilization and add to the operational cost of the product.

The problem of reducing power dissipation during testing was addressed in[5] in the context of BIST scheduling and control. In [6] this problem was addressed for scan circuits. Two techniques for reducing power dissipation during test application using scan structures were proposed. They are: Test set ordering; and Flip-Flop ordering. Here we address this problem for purely combinational circuits. It is assumed that controlling the order in which the tests

*Research partially supported by NSF Grant No.MIP-9102509.

are applied is possible in a stored pattern environment[7] or when a special built-in test pattern generator is designed for the circuit under test[8].

Two techniques for minimizing power dissipation during test application in combinational circuits are proposed. They are: (i) Test set ordering; and (ii) Repetition of test vectors. The test set ordering problem is shown to be NP-Hard. However, unlike SCAN circuits where a similar technique can be applied, for combinational circuits good heuristics can be developed. This polynomial time heuristic has a performance guarantee in that the solution computed by it will never be more than 1.5 times the optimal.

Computing the repetition sequence such that minimum power dissipation is obtained is shown to be tractable. We present a polynomial time algorithm that computes a sequence of input vectors, possibly with repetitions, so that power dissipation is minimized. However, the test set size could increase, in the worst case, to twice the original length. In practise however this increase is considerably smaller being on the average about 1.22.

Additional vectors, not in the test set, can be used to further reduce power dissipation. However, our result show that selecting such vectors is computationally very difficult.

Experimental evaluation of the proposed algorithms and techniques are presented. Two measures of power dissipation, viz. worst case (or peak) power dissipation[9, 10, 11] and average power dissipation[1, 2], were considered. Power dissipation is a function of the circuit delay. Three delay models have been studied in the literature [9] viz. unit delay, zero delay and general delay. Zero delay gives the least number of transitions while the general delay model gives the worst case transitions. We present results for the zero delay and the general-delay model. Unit delay is a special case of general delay and our algorithms can be easily modified for this case.

Experimental results show that considerable reduction in power dissipation can be obtained using the proposed techniques. In addition, the reduction in power dissipation is considerably larger with repetition than without it. These reductions are more pronounced for the general delay model.

This paper is organized as follows. Section 2 describes the power dissipation model we use. In section 3, we describe computation of node transition under zero-delay model and general-delay model. Section 4 defines the problems. Section 5 presents a proof of the intractability of the problems. Our heuristics use solutions to some “path traversal problems” on undirected graphs. In Section 6 we show how we can reduce the problems of interest to the path traversal problems. Sections 7 and 8 give the heuristics and experimental results for peak power dissipation and total power dissipation respectively. In Section 9 we present the results on repetition of test vectors. In Section 10 we consider the problem of adding new vectors to the test set to reduce power dissipation.

2 Power Dissipation Model

Two components of power dissipated in CMOS circuits are[12]: *Static power dissipation* due to leakage current or other current drawn continuously from the power supply (P_{st}); and *Dynamic power dissipation*. Dynamic power dissipation results due to switching transient current (P_{sc}: power due to short circuit current); and charging and discharging of load capacitances (P_d). Thus total power dissipation P_{total} equals $P_{st} + P_d + P_{sc}$.

Static power dissipation P_{st} is given by: $P_{st} = \sum_{1}^{n}$ leakage current $\times$ supply voltage. Here, n = number of devices and leakage current is measured per device.

The P_{sc} component of dynamic power accounts for the following. When the gate inputs are switching from '0' to '1' or from '1' to '0', both pull-up and pull-down networks may be on for a short period of time. This results in a short current pulse from V_{DD} to V_{SS} which results in power dissipation.

The P_d component of dynamic power accounts for power required to charge and discharge the output capacitive load. P_d is approximated by $1/2 \times C \times V_{DD}^2 \times N_G \times f$ where: C = output capacitance; N_G = total number of gate output transitions ($1 \rightarrow 0$ or $0 \rightarrow 1$); and f = repetition frequency. Note that this is the worst case power dissipation due to output transitions. Here it is assumed that during every transition the output is fully charged or discharged.

In general, leakage current is very small which makes P_{st} insignificant when compared with P_d. P_{sc} is proportional to the ratio t_{rf}/f where t_{rf} is rise and fall time and f is the repetition frequency. This ratio is generally small. In addition, it assumes that the pull up/pull down networks are on for t_{rf} - the entire duration of rise and fall times. This can only happen if all input transitions arrive simultaneously. In general, because of gate delays, this will not happen. Thus, the time for which all the inputs of a gate change such that the pull-up and pull-down networks are on is very small, if at all it is nonzero. So for all further discussions, following [12], P_{total} is assumed to be given by:

$$P_{total} = 1/2 \times C \times V_{DD}^2 \times N_G \times f \quad (1)$$

Equation (1) implies that power is dissipated at a node when the input vector is changed from T_i to T_{i+1}. Let $P_{i,i+1}$ be the total power dissipated in the circuit when this transition from T_i to T_{i+1} occurs. Then,

$$P_{i,i+1} = \sum_{j \in SetofNodes} 1/2 \times C_j \times V_{DD}^2 \times N_{G_j} \times f \quad (2)$$

Note that power dissipated at each node is proportional to the number of transitions taking place at that node. Number of transitions at each node depend on the gate delays. We will discuss a way to compute worst case node transitions. We assume that we have knowledge of the output load capacitance at each node.

3 Computing Node Transitions

We discuss how to compute the number of transitions at each node when input vector T_2 is applied after input vector T_1. Transitions at the primary inputs are ignored because in the model used transitions at the primary inputs cause power dissipation at the driving circuit, not in the circuit under consideration.

Under the **zero-delay model**, all gates are assumed to have zero delay. Thus, transitions are assumed to occur simultaneously. Therefore glitches (hazards) cannot occur at the outputs of any of the gates and each gate can make at most one transition: $0 \rightarrow 1$ or $1 \rightarrow 0$. Thus, for zero-delay model, the following four logic values adequately describe the signal state of a line on application of $< T_1, T_2 >$: static-zero (s0) (Logic level remains zero); static-one (s1) (Logic level remains one); rising (r) (Logic level changes from 0 to 1); and falling (f) (Logic level changes from 1 to 0). Note that s0, s1 imply no transition while r, f imply one transition. For two-input AND, OR and EX-OR gates output behavior under zero-delay model is described using Tables 1-3.

For the **General-Delay model**, gates can have finite but arbitrary delays. The following six logic values adequately describe the signal state at a line on application of an input pair $\langle T_1, T_2 \rangle$: static-zero (s0) (Logic level remains zero); static-one (s1) (Logic level remains one); hazard-rising (hr) (Initial logic level 0, final logic level 1 and one or more transitions); hazard-falling (hf) (Initial logic level 1, final logic level 1 and one or more transitions); hazard-0 (h0) (hazard with initial and final logic level 0); and hazard-1 (h1) (hazard with initial and final logic level 1).

Let GDV $= \{\mathsf{s0}, \mathsf{s1}, \mathsf{hr}, \mathsf{hf}, \mathsf{h0}, \mathsf{h1}\}$. Note that any state of the line, with any number of glitches during transitions, can be uniquely represented by an ordered pair $\{(a, b) : a \in I$ and $b \in GDV\}$, where a denotes number of glitches during transitions. Transitions $(0 \rightarrow 1 \rightarrow 0 \rightarrow 1)$ is represented as (3,hr) while transitions $(1 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 1)$ is represented as (4,h1). For two-input AND, OR and EX-OR gates output behavior under general-delay model is described by Tables 4-6.

Thus, given a combinational network C and an input pair $< T_i, T_{i+1} >$, the number of transitions at each node in C can be computed by traversing C in topological order and using these tables. This takes time $O(n)$ where n is number of nodes (or gates) in the network.

4 Problem Definitions

We first look at an example. Consider the circuit shown in Figure 1 and the test set $T = \{ T_1 = 000, T_2 = 001, T_3 = 100, T_4 = 010\}$. Here each input vector is represented in the order $(x_1\ x_2\ x_3)$. Given two vectors T_i, T_j the number of transitions at each node can be computed as stated in Section 3. The total number of transitions for the circuit is the weighted sum of all node transitions weighted against output load capacitance for each node. For the sake of this example assume output load capacitance to be unity.

Next, represent the problem using a directed graph. We call this graph *Input Transitions Graph* (ITG). ITG under zero delay model for the circuit in Figure 1 and for vectors in vector set T is shown in Figure 2. It can be seen from Figure 2 that peak power for the sequence $\langle T_1, T_2, T_3, T_4 \rangle$ is 3 and total power is 6, while for the sequence $\langle T_1, T_3, T_4, T_2 \rangle$ peak power is 1 and total power is 3. Thus substantial improvement can be obtained by properly sequencing the input vectors. This motivates the study of the following problems. Term $P_{i,i+1}$ is used as defined in section 2.

OPT_SEQ_PEAK (OSP): Given a combinational circuit C and a set of input vectors $T = \{T_1, ..., T_N\}$. Compute an optimal input order $\langle S_1, \ldots, S_N \rangle$ of T where $S_i = T_{\pi(i)}$ s.t.

$$P_{max} = \max_{i=1}^{N-1}\{P_{i,i+1}\} \text{ is minimized.}$$

The objective is to minimize the peak power dissipated while applying the set T by properly sequencing the input vectors.

OPT_SEQ_TOTAL (OST): Given a combinational circuit C and a set of input vectors $T = \{T_1, ..., T_N\}$. Compute an optimal input order $\langle S_1, ..., S_N \rangle$ of T where $S_i = T_{\pi(i)}$ s.t.

$$\sum_{i=1}^{N-1} P_{i,i+1} \text{ is minimized.}$$

To compute the sequence with the best average power dissipation solution to OST can be used.

Note that in these problems, we do not estimate the power dissipation but compute a sequence of input vectors such that power dissipation is minimized. As stated in section 2, given the library modules and the number of transitions for each pair of input vectors, estimation of the power dissipation in the circuit can be obtained by using equation 2 and solutions to these problems.

In these problems we impose the restriction that the size of the test set not be changed. Later we will consider other variants where this constraint is relaxed.

5 Hardness Results

We show that both OSP and OST are NP-Hard. Note that the proofs are independent of the delay model used. We use Hamiltonian Path (HP) Problem to prove that OSP is NP-Hard. HP is defined as follows:

HP: Given an undirected graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$, find path $\langle v_{i_1}, v_{i_2}, \ldots, v_{i_n} \rangle$ s.t. $i_j \in \{1, \ldots, n\}\ \forall j$. and every node in the graph is visited once and only once

Theorem 5.1 *OSP is NP-hard.*

Proof: HP for an undirected graph is known to be NP-Hard [13]. We reduce HP to OSP. Let $G = (V, E)$ be any undirected graph where $V = \{v_1, v_2, ..., v_n\}, E = \{e_1, e_2, ..., e_m\}$, $n = \mid V \mid$, $m = \mid E \mid$. We construct the following instance of OSP with circuit C and test set T. C consists of $(n-1)m$ 2-input OR gates. Inputs of each OR gate is tied together. Thus there are $(n-1)m$ inputs and $(n-1)m$ outputs.

Input set T consists of n input vectors $T = \{T_1, T_2, ..., T_n\}$. Each input vector corresponds to a node in G. Every input vector has m parts, one for each edge and each part has $n-1$ bits. Thus each vector T_i consists of $(n-1)m$ bits. Let $T_i^j = j^{th}$ part of T_i and let $e_j = (v_a, v_b)$ s.t. $1 \leq a < b \leq n$.

$$T_i^j = \begin{cases} 10^{n-2} & \text{if } i = a \text{ or } i = b \\ 0^i 10^{n-1-(i+1)} & \text{if } i \leq a \\ 0^{i-1}10^{n-1-i} & \text{if } a \leq i \leq b \\ 0^{i-2}10^{n-i} & \text{if } i \geq b \end{cases}$$

For example, consider the graph shown in Figure 3. The circuit C and input set corresponding to this graph is shown in Figure 4 and Table 4 respectively.

We next show that G is hamiltonian iff minimum of P_{max} for the construction is such that circuit C with input set T is $2(e-1)$. Let $p = 2(e-1)$. If there exists an edge between nodes i, k then $P_{i,j} = 2(e-1)$ else $P_{i,j} = 2e$. Therefore min $P_{max} = 2(e-1) \Rightarrow \exists$ a input sequence $\langle T_{i_1}, ..., T_{i_n} \rangle$ such that for all k, $(v_{i_k}, v_{i_{k+1}}) \in E$. This implies that $\langle v_{i_1}, ..., v_{i_n} \rangle$ is a hamiltonian path in G. Thus G is hamiltonian.

Conversely, assume that G is hamiltonian. Let $\langle v_{i_1}, ..., v_{i_n} \rangle$ be the HP. We schedule input vectors $\langle T_{i_1}, ..., T_{i_n} \rangle$. Now $(v_{i_k}, v_{i_{k+1}}) \in E$ hence $P_{i_{k,k+1}} = 2(e-1) \forall k \in \{1, ..., n-1\}$. Hence P_{max} for the given path is $2(e-1)$. Note that this P_{max} will be minimum because for every sequence $P_{max} = 2(e-1)$ or $(2e)$ as argued above. Hence the proof. □

Theorem 5.2 *OST is NP-hard.*

Proof: Using the same construction as in the proof of Theorem 5.1, it can be verified that graph G is hamiltonian iff sum of the output transitions in a sequence is $(n-1) \times 2(e-1)$. □

6 Reduction to Graph Problems

Since OSP and OST are NP-Hard, one has to resort to heuristics which yield near-optimal solutions. All the heuristics will reduce the given problem to "path traversal problems" in an *undirected graph*, as was alluded to in the example in the previous section. The graph that we use is the *Input Transition Graph* (ITG) Next, we discuss the construction of ITG from a given instance of OST or OSP.

Given an input set $\{T_1, \ldots, T_N\}$ and a multilevel combinational circuit C, we construct a complete undirected graph $ITG = (V, E)$ s.t. $\mid V \mid = N$. Let L denote the set of nodes in circuit C. Each node v_i in G corresponds to an input vector T_i. Consider a weight function w : $E \rightarrow I$, where E is the set of edges in G and I is the set of integers. Let $w(i,j)$ denote the total number of weighted transitions that occur when T_i is applied followed by T_j. Let $w_l(i,j)$ denote the number of transitions at node l when T_i is followed by T_j. Let C_l be the output capacitance of node l. Then

$$w(i,j) = \sum_{l \in L} C_l \times w_l(i,j) \tag{3}$$

We know that a pair $\{(a,b) : a \in I, b \in GDV\}$ represents the number of transitions and the kind of transitions occurring uniquely when a pair of input vectors is applied to a circuit. Let $(w_l(i,j), u_l(i,j))$ represent such a pair at node l of the circuit when input vector T_i is followed by T_j under general-delay model. The following lemma justifies why the constructed graph has to be undirected. The fact that ITG is undirected is important because it leads to the development of better heuristics with performance bounds.

Lemma 6.1 *Given a combinational circuit C, the weight function, $w_l(i,j)$, denoting the number of transitions at each node l of C when vector T_j is applied after vector T_i, is symmetric, i.e.$w_l(i,j) = w_l(j,i)$ $\forall$ $i,j \in \{1,\ldots,N\}$ and $\forall$ $l \in$ Set of Nodes in C under general-delay model.*

Proof: The proof is by induction on the depth of a circuit. *Depth* of a combinational circuit is defined as the length of a longest path from a primary input to a primary output. Let $D(C)$ denote the depth of a combinational circuit C. Similarly depth of a node l in C is the length of a longest path from a primary input to node l. Let $d(l)$ denote the depth of node l. We want to prove that $w_l(i,j) = w_l(j,i)$ $\forall$ $l s.t. d(l) \leq D(C)$. Note that for any node l, $u_l(i,j) \in$ {s0, s1, h0, h1, hr, hf} and the corresponding values for $u_l(j,i)$ are {s1, s0, h1, h0, hf, hr}. Further note that for $u_l(i,j) =$ (s0, s1, h0, h1), corresponding $u_l(j,i)$ does not change. Now consider the basis case when $d(l) = 1$, i.e. inputs to node l are primary inputs. Let the inputs be x and y. The proposition is true for x,$y \in$ {s0, s1} as the input values don't change. Note that x,$y \notin$ {h0, h1}. So the proposition needs to be verified for x or $y \in$ {hr, hf}. The cases are :

- $x(i,j)$: (0, s0), $y(i,j)$: (1, hr)
  $x(j,i)$: (0, s0), $y(j,i)$: (1, hf)
- $x(i,j)$: (0, s1), $y(i,j)$: (1, hr)
  $x(j,i)$: (0, s1), $y(j,i)$: (1, hf)
- $x(i,j)$: (1, hr), $y(i,j)$: (1, hr)
  $x(j,i)$: (1, hf), $y(j,i)$: (1, hf)

It can be verified from Tables 4, 5 and 6 that $w_l(i,j) = w_l(j,i)$ in the above cases. Thus basis case is proved. For induction hypothesis, assume $w_l(i,j) = w_l(j,i)$ $\forall$ ls.t.$d(l) \leq m$. Let n

be a node with $d(n) = m$. Let x, y be inputs to n. By an argument similar to the basis case, the proposition is true when $x, y \in \{$s0, s1, h0, h1$\}$. So we need to consider following cases:

- $x(i,j)$: (N1, h0), $y(i,j)$: (N2, hr)
  $x(j,i)$: (N1, h0), $y(j,i)$: (N2, hf)
- $x(i,j)$: (N1, h1), $y(i,j)$: (N2, hr)
  $x(j,i)$: (N1, h1), $y(j,i)$: (N2, hf)
- $x(i,j)$: (N1, hr), $y(i,j)$: (N2, hr)
  $x(j,i)$: (N1, hf), $y(j,i)$: (N2, hf)

Note that because of the induction hypothesis, $w_x(i,j) = w_x(j,i)$ and $w_y(i,j) = w_y(j,i)$. It can be verified from the Tables 4, 5 and 6 that $w_n(i,j) = w_n(j,i)$ for all the above cases. Hence the proof. □

As a consequence of Lemma 6.1 and equation 3 we get the following theorem and it's corollary.

Theorem 6.1 *Given a combinational circuit C and an input vector set T, the edge weight function, $w(i,j)$, in the corresponding instance of ITG is symmetric under general delay model, i.e. $w(i,j) = w(j,i)\ \forall i,j \in \{1,\ldots,N\}$.*

Corollary 6.1 *Given a combinational circuit C and the input vector set T, the edge weight function, $w(i,j)$, in the corresponding instance of ITG is symmetric under* **zero delay model***.*

Thus OSP and OST can be restated as path traversal problems for the constructed graph ITG as follows.
OSP: Find a hamiltonian path $\langle\ v_{i_1}, \ldots, v_{i_N}\rangle$ in ITG s.t. $\max\{w(i_k, i_{k+1})\}$ for k = 1,..., N-1 is minimized.
OST: Find a hamiltonian path $\langle\ v_{i_1}, \ldots, v_{i_N}\rangle$ in ITG s.t. $\sum w(i_k, i_{k+1})$ for k = 1,..., N-1 is minimized.

7 Minimizing Peak Power

We use the same heuristic, described below, for minimization of peak power under zero delay as well as general delay model.

GET_PEAK($ITG(V,E)$ **)**

```
Tour ← φ; i ← 0;
Sort edges by weights in increasing order;
while( Edges_In(Tour) != N-1 )
        if ( Edge[i] can be added to Tour )
                Tour ← Tour ∪ Edge[i];
        endif
        i ← i + 1;
endwhile
```

The heuristic works as follows. Tour and also the loop invariant is a set of disjoint paths such that there is no cycle. An edge can be added to Tour in the following cases. (a) Its endpoints are not connected to any paths in Tour, (b) Only one of its endpoints is connected to an endpoint of a path in Tour or (c) One endpoint is connected to an endpoint of a path and the other is connected to an endpoint of another path (different from the first path).

Consider the graph shown in Figure 5(a). Let's say the edges are sorted in the following order: $\langle\ (B,C),(C,D),(D,E),(E,C),(D,A),(D,B),(E,B),(A,C),(A,E),(A,B)\rangle$. The algorithm selects following edges $\{(B,C),(C,D),(D,E)\}$ then it drops $\{(E,C),\ (D,A),\ (E,B),\ (A,C)\}$ and selects $\{(A,E)\}$. The hamiltonian path obtained is shown in Figure 5(b).

For our experiments we used ISCAS85 circuits [14] and stuck-at test vectors for each circuit, generated using the stuck-at generator provided with Octtools [15]. Then we resequenced the test vectors using heuristic GET_PEAK and compared the peak power dissipation of the later sequence with the power disspation of the original sequence.

Computation of power dissipation for a sequence of vectors needs knowledge of output load capacitance of each gate of the network. Since we assume that technology mapping has already taken place, output load capacitance will be known for each gate. In our experiments, we assumed capacitance to be unity for the sake of simplicity. Hence power dissipation represents sum of all the transitions at each gate.

¿From Tables 4-6 it can be verified that for AND and OR operations computation of output node transitions, for the general delay model, are not an associative operations. For gates having fanin greater than 2 the tables can be constructed though they become messy. In our implementation we convert every gate with fanin great than 2 into a tree of gates each with fanin 2. Note that this is an approximation of the actual node computation. Results are listed in Table 7. Note that percentage improvement varied from 7% to 43% for the zero delay model and from 20% to 88% in the general delay mode. The experiments were run on SPARC-II workstations. Timings varied from a few seconds to 25 minutes for the largest circuit.

8 Minimizing Total Power

We use CHRFD_TOTAL, described below, for computing good input vector sequence such that total power dissipation is minimized[16]. Before we introduce the heuristic, some related concepts are explained.

Given a graph containing an even number of nodes, a *matching* is collection of edges M such that each node is the endpoint of exactly one edge in M. A minimum weight matching is one for which the sum of edge weights is minimum. Such matchings can be found in time $O(n^3)$ where n is number of nodes of the graph [17]. An *Eulerian graph* is a graph in which all the vertices have even degree. Such a graph can be traversed by traversing each edge exactly once. A hamiltonian tour can be obtained from an eulerian tour by using *shortcuts*. A shortcut is a jump from a vertex to anther vertex when the edge between these two vertices does not exist. We next discuss the CHRFD_TOTAL heuristic.

CHRFD_TOTAL($ITG(V,E)$)
(1) Find Minimum Spanning Tree of ITG, say MST(ITG);
(2) $V' \leftarrow$ odd degree vertices in MST(ITG);
(3) $M \leftarrow$ minimum weighted matching of V';
(4) $G' \leftarrow ITG \cup$ edges in M;
(5) Find an Euler Tour of MST(ITG);

(6) Convert it to a Tour using shortcuts.

Snapshots of this algorithm are shown in Figure 5(a), (c) and (d). Figure 5(c) (without dotted lines) shows a minimum spanning tree for the graph in Figure 5(a) consisting of edges $\{(B,C),(C,D),(D,E),(A,D)\}$. Odd degree vertices in this tree are $\{A,B,D,E\}$. Minimum weighted matching consists of edges $\{(B,E),(A,D)\}$. An Euler tour for this augmented graph is $\langle B,C,D,A,D,E,B\rangle$. Note that node D is visited twice in this tour which can be jumped over. A tour after shortcuts is shown in 5(d).

Lemma 8.1 *Weight function $w(i,j)$ satisfies triangle inequality under zero-delay model, i.e. $w(i,j)+w(j,k) \geq w(i,k)$.*

proof: Consider node l of a combinational circuit C. We will first show that $w_l(i,j)+w_l(j,k) \geq w_l(i,k)\ \forall l \in C$. Under zero-delay model, logic level at any node l can undergo at most one transition i.e. $w(i,j)$, $w(j,k)$, $w(i,k) \in \{0,1\}$. Consider the case when $w_l(i,k)$ is one i.e either r or f. Then either $w_l(i,j)$ or $w_l(j,k)$ will be one and the other one will be zero, or else one of them will be neither zero nor one which is a contradiction. The inequality holds when $w_l(i,k)$ is zero. Thus,

$$\begin{aligned} w_l(i,j)+w_l(j,k) &\geq w_l(i,k)\ \forall l \in C \\ C_l \times (w_l(i,j)+w_l(j,k)) &\geq C_l \times w_l(i,k)\ \forall l \in C \\ \sum_{l\in C} C_l \times (w_l(i,j)+w_l(j,k)) &\geq \sum_{l\in C} C_l \times w_l(i,k)\ \forall l \in C \\ w(i,j)+w(j,k) &\geq w(i,k) \end{aligned}$$

□

¿From Lemma 8.1 and the analysis of [16], we get the following result.

Corollary 8.1 *For zero delay model, total power dissipation for the sequence of vectors obtained by using CHRFD_TOTAL, is at most 1.5 times the optimal total power dissipation.*

Note that such a guarantee does not hold for general delay model. This is a consequence of the fact that Lemma 8.1 does not hold for general delay model. For example, consider a two-input OR gate. Let $T_1 = 01$, $T_2 = 10$ and $T_3 = 11$. $w(1,2)$ under general delay model is 2 as the output is (2, H1), while $w(1,3) = 0$ and $w(3,2) = 0$. Thus $w(1,2) > w(1,3)+w(3,2)$ which violates Lemma 8.1.

The experimental results are shown in Table 8. Similar to the peak power minimization, we computed the percentage improvement in total power dissipation in ISCAS85 circuits when the same stuck-at test vectors are applied. Code for matching was written by Dr. Rothberg [18] and was made available as public domain in ftp site *dimacs.rutgers.edu*. This program implements algorithm due to Gabow [19]. Note that the improvement can be as large as 94% !!

9 Resequencing with Repetition of Vectors

In the previous sections resequencing was equivalent to permutation of a given set of input vectors. Following example shows that, if the restriction of using a vector *once and only once* is relaxed, optimal peak power dissipation can be further reduced.

Consider the circuit shown in Fig. 6(a) and the input vector set $T = \{T_1 = 010, T_2 = 111, T_3 = 100, T_4 = 001\}$. ITG under general delay model for this circuit and input set T is shown in Fig. 6(b). If repetition of vectors is not allowed, peak power dissipation is 4 e.g. sequence $\langle T_1, T_2, T_4, T_3 \rangle$, while the peak power dissipation is 2 when repetition is allowed e.g. sequence $\langle T_1, T_4, T_2, T_4, T_3 \rangle$. One natural question to ask at this point is, can we do better in terms of quality of solution if repetition of vectors is allowed ? And if so, how much penalty do we pay in terms of length of the sequence ? Next we define these problems.

9.1 Definitions

OPT_SEQ_PEAK_VARIABLE_LENGTH (OSP_VL): Given a combinational circuit C and a set of input vectors $T = \{T_1, ..., T_N\}$. Compute an optimal input sequence $\langle S_1, \ldots, S_M \rangle$ where $M \geq N$ and for each $i \in \{1, \ldots, N\}$ $\exists$ S_j s.t. $S_j = T_i$ and

$$P_{max} = \max_{i=1}^{M-1} \{P_{i,i+1}\} \text{ is minimized}$$

OPT_SEQ_AVERAGE_VARIABLE_LENGTH (OSA_VL): Given a combinational circuit C and a set of input vectors $T = \{T_1, ..., T_N\}$. Compute an optimal input sequence $\langle S_1, ..., S_M \rangle$ where $M \geq N$ and for each $i \in \{1, \ldots, N\}$ $\exists$ S_j s.t. $S_j = T_i$ and

$$\frac{1}{M-1} \sum_{i=1}^{M-1} P_{i,i+1} \text{ is minimized}$$

Note that sequence length can be increased either by repeating some of the input vectors or by adding new vectors.

9.2 Heuristics for OSP_VL and OSA_VL

Our heuristic is based on Kruskal's algorithm for constructing Minimum Spanning Tree for an undirected graph [20]. It can be verified that Kruskal's algorithm also gives a spanning tree with maximum edge weight minimized among all the spanning trees. It is an exercise in [20] on page 510. Following theorem gives a heuristic which computes an optimal solution to OSP_VL problem with length of the sequence increased to at most twice the length of the original sequence.

Theorem 9.1 *If the length of the input sequence is allowed to increase by adding vectors from the given input set of vectors only, then OSP_VL can be solved to optimality by using Kruskal's algorithm by increasing the sequence length to at most $2 \times n$ where n is the length of the original sequence.*

Proof: Construct a minimum spanning tree using Kruskal's algorithm. Start from any node and traverse the tree in any manner (preorder, postorder etc.) along the edges of the tree. An edge might be traversed at most two times in this traversal which guarantees that the sequence can be at most twice the length of the original sequence. Thus the sequence guarantees that maximum power dissipation is minimized over all possible sequences. □

For example, if the Minimum Spanning Tree for ITG in Fig. 6(b) consists of following edges $\{(T_1, T_4), (T_2, T_4), (T_3, T_4)\}$. If T_1 is chosen as a root, then the tour produced by this heuristic would be $\langle T_1, T_4, T_2, T_4, T_3 \rangle$.

Although the theoretical upper bound for factor of increase in vector length is 2, ITG may have a number of edges which are less than or equal to the optimal peak power dissipation, giving a number of short cuts. So the heuristic for OSP_VL is as follows:

```
OSP_VL(ITG)
(M,peak) ← MST-Kruskal(ITG)
root ← some v ∈ ITG, Tour ← φ
Inorder-Numbering(root)
current_node ← root
next_node ← next_inorder_number
while ( Not all nodes visited )
        if ( edge(current_node,next_node) ∈ M )
                add it to Tour
                current_node ← next_node
                next_node ← next_inorder_number
        else if ( edge(current_node,next_node) ∈ E and
                weight(edge) < peak)
                add it to Tour
                current_node ← next_node
                next_node ← next_inorder_number
        else
                /* backtrack in the tree by one step */
                current_node ← parent(current_node)
endwhile
output Tour
```

A similar heuristic, with some modification, can be used for OSA_VL. We have not implemented such a heuristic and leave it to the interested reader.

Experimental evaluation of this heuristic is presented in Table 10. The fourth column implies that peak power can be reduced considerably by repeating the vectors. For the zero delay model this reduction is, on an average, about 47.56%. For the general delay model the average reduction is about 87%.

The seventh column implies that using "shortcuts" the increase in the size of the test set can be contained to be much less that 2. On an average, this increase in length is about 1.22. Note that sometimes (7552, General Delay Model) by marginally increasing the test length we can get a dramatic reduction in the power dissipation.

The last column gives the % improvement obtained by carefully ordering them along with possible repetition when compared with only carefully ordering them without repetition. It is given as a percentage of the later. For the zero-delay model the average improvement is about 24.57% whereas for the general delay model it is about 70.80%.

10 Addition of New Vectors

Repetition of vectors reduced peak power dissipation. Further still, the following question can be raised: Rather than just repeating vectors from the given set, can a set of new vectors be cleverly added so that peak power dissipation is reduced further? This problem is computationally difficult to solve. In fact, we show that the problem is NP-Hard even when the given sequence is of length 2! We next establish this result.

OPT_SEQUENCE_PEAK_WITH_ADDITION_OF_VECTORS (OSP_AD): Given a combinational circuit C and two input vectors T_1, T_2. Compute an input vector T_3 such that $\max(P_{13}, P_{32})$ is minimized.

Theorem 10.1 *OSP_AD is NP-Hard.*

Proof: We reduce an instance of MAX-SAT to OSP_AD. MAX-SAT is defined as follows. Given a CNF formula F consisting of m clauses and a set of n literals U. Let each clause Cl_i have k_i literals such that each literal is of the form x or $\overline{x}$ where $x \in U$. F can be represented as

$$\begin{aligned} F &= \textstyle\bigwedge_{i=1}^{m} Cl_i, \\ Cl_i &= \textstyle\bigvee_{j=1}^{k_i} x_j^i \text{ such that } x_j^i \in U \vee \overline{x_j^i} \in U \end{aligned}$$

MAX-SAT : Given a boolean formula F as shown above, find an assignment $A \in \{1,0\}^n$ to literals $x_1, \ldots, x_n$ such that maximum number of clauses are satisfied.

Given a formula F, we construct the circuit C shown in Fig. 7. Circuit C consists of three parts C_1, C_2 and C_3. Besides the inputs of F, C has two new inputs u_1, u_2.

Circuit C_1 contains m OR gates - one for each clause Cl_i in F. These OR-gates have, in addition to the literal in the corresponding clause, the input u_1. The outputs of the OR-gates are denoted by $A_1, \ldots, A_m$. The A_i's are anded with u_2 to get the m outputs $B_1, \ldots, B_m$ of C_1.

C_2 consists of m 2-input AND gates with inputs u_1, u_2. Output lines of OR gates in C_2 are denoted by $D_1, \ldots, D_m$. C_3 consists of m 2-input EX-OR gates with inputs u_1, u_2. Output lines of EX-OR gates in C_3 are denoted by $E_1, \ldots, E_m$.

Thus, given an instance of MAX_SAT, the instance of OSP_AD constructed is defined as:

$$\begin{aligned} &C \quad \text{as defined above with } U' = U \textstyle\bigcup \{u_1, u_2 2\} \text{ as its primary inputs} \\ &T_1 \;=\; u_1 = 1,\ u_2 = 0, \text{ any assignment to literals } x_1, \ldots, x_n \\ &T_2 \;=\; u_1 = 1,\ u_2 = 1, \text{ any assignment to literals } x_1, \ldots, x_n \end{aligned}$$

For the zero delay model, the number of transitions at A_i, B_i, D_i, E_i on application of $< T_1, T_2 >$ can be computed by comparing the values given in the following table on application of these two input vectors. This, denoted by $P_{1,2}$ equals $3m$.

vector	A_is	B_is	D_is	E_is
T_1	1	0	0	1
T_2	1	1	1	0

Consider any input vector T_3' of F. Let l be the number of clauses of F it **does not** satisfy. Let $T - 3$ be derived from T_3' by assigning values to u_1, u_2. Clearly, T_3 sets $(m - l)$ OR gates to 1. The next table compares $P_{1,3}$ and $P_{3,2}$ for different values of u_1 and u_2.

Values of u_1, u_2	vector change	No of A_is change	No of B_is change	No of D_is change	No of E_is change	Total No of Transitions
$T_3 : u_1, u_2 = 00$	$T_1 \rightarrow T_3$	l	0	0	m	$m + l$
$T_3 : u_1, u_2 = 00$	$T_3 \rightarrow T_2$	l	m	m	0	$2m + l$
$T_3 : u_1, u_2 = 01$	$T_1 \rightarrow T_3$	l	$m - l$	0	0	m
$T_3 : u_1, u_2 = 01$	$T_3 \rightarrow T_2$	l	l	m	m	$2m + 2l$
$T_3 : u_1, u_2 = 11$	$T_1 \rightarrow T_3$	0	m	m	m	$3m$
$T_3 : u_1, u_2 = 11$	$T_3 \rightarrow T_2$	0	0	0	0	0
$T_3 : u_1, u_2 = 10$	$T_1 \rightarrow T_3$	0	0	0	0	0
$T_3 : u_1, u_2 = 10$	$T_3 \rightarrow T_2$	0	m	m	m	$3m$

¿From the table, it can be seen that $\max(P_{1,3}, P_{3,2})$ is minimized when T_3 is chosen such that $u_1, u_2 = 00$ and it is $(2m + l)$ where l is the number of clauses that are not satisfies by T_3. Thus, peak power is minimized when number of clauses not satisfied by formula F is minimized i.e. number of clauses satisfied by F is maximized. Thus this instance of OSP_AD will be solved iff the corresponding instance of MAX-SAT is solved. Hence the proof. □

References

[1] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks," in *ICCAD*, pp. 402–407, ACM/IEEE, 1992.

[2] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," in *29th Design Automation Conference*, pp. 253–259, ACM/IEEE, 1992.

[3] C.-Y. Tsui, M. Pedram, and A. Despain, "Technology decomposition and mapping targeting low power dissipation," in *30th Design Automation Conference*, pp. 68–73, ACM/IEEE, 1993.

[4] V. Tiwari, P. Ashar, and S. Malik, "Technology mapping for low power," in *30th Design Automation Conference*, pp. 74–79, ACM/IEEE, 1993.

[5] R. M. Chou, K. K. Saluja, and V. D. Agrawal, "Power constraint scheduling of tests," *IEEE International Conference on VLSI Design*, pp. 271–274, 1994.

[6] S. Chakravarty and V. Dabholkar, "Minimizing power dissipation in scan circuits during test application," *Tech. Report No. 94-06, Dept. of Computer Science, SUNY at Buffalo, to appear in ACM/IEEE Int'l workshop on Low Power Design, Napa, California, April,* 1994.

[7] M. Abramovici, M. Breuer, and A. Friedman, *Digital Systems Testing and Testable Design.* Computer Science Press, 1990.

[8] M. Khare and A. Albicki, "Cellular automata used for test pattern gengeration," *Intl' Conference on Computer Design*, pp. 56–59, 1987.

[9] S. Devadas, K. Keutzer, and J. White, "Estimation of power dissipation in CMOS combinational circuits using boolean function manipulation," *IEEE Trans. Computer Aided Design*, vol. CAD-11, pp. 373–383, March 1992.

[10] H. Kriplani, F. Najm, P. Yang, and I. Hajj, "Maximum current estimation in CMOS combinational circuits," in *29th Design Automation Conference*, pp. 2–7, ACM/IEEE, 1992.

[11] H. Kriplani, F. Najm, P. Yang, and I. Hajj, "Resolving signal correlations for estimating maximum currents in CMOS combinational circuits," in *30th Design Automation Conference*, pp. 384–388, ACM/IEEE, 1993.

[12] H. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A systems perspective.* Addison-Wesley Publicating Company, second ed., 1992.

[13] M. R. Gary and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness.* San Fransisco: W. H. Freeman, 1979.

[14] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran," in *Int'l Symposium for Testing and Failure Analysis*, pp. 253–259, ACM/IEEE, 1992. A special session on ATPG and Fault simulation.

[15] T. Larabee, "Test pattern generation using boolean satisfiability," *IEEE Trans. Computer-Aided Design*, vol. CAD-11, pp. 4–15, January 1992.

[16] E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys, *The Traveling Salesman Problem.* Great Britain: John Wiley and Sons, 1985.

[17] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids.* New York: Holt, Rinehart and Winston, 1976.

[18] E. Rothberg, *Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs.* PhD thesis, Stanford University, 1973.

[19] H. N. Gabow, "An efficient implementation of Edmond's algorithm for maximum matching on graphs," *JACM*, vol. 23, pp. 221–234, 1976.

[20] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms.* Cambridge, Massachusetts: The MIT Press, 1991.

Table 1				
AND	s0	s1	r	f
s0	s0	s0	s0	s0
s1	s0	s1	r	f
r	s0	r	r	s0
f	s0	f	s0	f

Table 2				
OR	s0	s1	r	f
s0	s0	s1	r	f
s1	s1	s1	s1	s1
r	r	s1	r	s1
f	f	s1	s1	f

Table 3				
EX-OR	s0	s1	r	f
s0	s0	s1	r	f
s1	s1	s0	f	r
r	r	f	s0	s1
f	f	s1	s1	s0

AND	0,s1	0,s0	$N2$,h1	$N2$,h0	$N2$,hr	$N2$,hf
0,s1	0,s1	0,s0	$N2$,h1	$N2$,h0	$N2$,hr	$N2$,hf
0,s0	0,s0	0,s0	0,s0	0,s0	0,s0	0,s0
$N1$,h1	$N1$,h1	0,s0	$(N1+N2)$,h1	$\max\{N1, N2\}$,h0	$(N1+N2)$,hr	$(N1+N2)$,hf
$N1$,h0	$N1$,h0	0,s0	$\max\{N1,N2\}$,h0	$\max\{N1, N2\}$,h0	$(N1+N2\text{-}1)$,h0	$(N1+N2\text{-}1)$,h0
$N1$,hr	$N1$,hr	0,s0	$(N1+N2)$,hr	$(N1+N2\text{-}1)$,h0	$\max\{N1,N2\}$,hr	$(N1+N2)$,h0
$N1$,hf	$N1$,hf	0,s0	$(N1+N2)$,hf	$(N1+N2\text{-}1)$,h0	$(N1+N2)$,h0	$\max\{N1, N2\}$,hf

Table 4: Number of transitions for 2-input AND with general-delay

OR	0,s1	0,s0	$N2$,h1	$N2$,h0	$N2$,hr	$N2$,hf
0,s1	0,s1	0,s1	0,s1	0,s1	0,s1	0,s1
0,s0	0,s1	0,s0	$N2$,h1	$N2$,h0	$N2$,hr	$N2$,hf
$N1$,h1	0,s1	$N1$,h1	$(N1+N2)$,h1	$(N1+N2)$,h1	$(N1+N2\text{-}1)$,h1	$(N1+N2\text{-}1)$,h1
$N1$,h0	0,s1	$N1$,h0	$(N1+N2)$,h1	$(N1+N2)$,h0	$(N1+N2)$,hr	$(N1+N2)$,hf
$N1$,hr	0,s1	$N1$,hr	$(N1+N2\text{-}1)$,h1	$(N1+N2)$,hr	$(N1+N2\text{-}1)$,hr	$(N1+N2)$,h1
$N1$,hf	0,s1	$N1$,hf	$(N1+N2\text{-}1)$,h1	$(N1+N2)$,hf	$(N1+N2)$,h1	$(N1+N2\text{-}1)$,hf

Table 5: Number of transitions for 2-input OR with general-delay

EX-OR	0,s1	0,s0	$N2$,h1	$N2$,h0	$N2$,hr	$N2$,hf
0,s1	0,s0	0,s1	$N2$,h0	$N2$,h1	$N2$,hf	$N2$,hr
0,s0	0,s1	0,s0	$N2$,h1	$N2$,h0	$N2$,hr	$N2$,hf
$N1$,h1	$N1$,h0	$N1$,h1	$(N1+N2)$,h0	$(N1+N2)$,h1	$(N1+N2)$,hf	$(N1+N2)$,hr
$N1$,h0	$N1$,h1	$N1$,h0	$(N1+N2)$,h1	$(N1+N2)$,h1	$(N1+N2)$,hr	$(N1+N2)$,hf
$N1$,hr	$N1$,hf	$N1$,hr	$(N1+N2)$,hf	$(N1+N2)$,hr	$(N1+N2)$,h0	$(N1+N2)$,h1
$N1$,hf	$N1$,hr	$N1$,hf	$(N1+N2)$,hr	$(N1+N2)$,hf	$(N1+N2)$,h1	$(N1+N2)$,h0

Table 6: Number of transitions for 2-input EX-OR with general-delay

	T_i^1	T_i^2	T_i^3	T_i^4	T_i^5	T_i^6	T_i^7
a	10000	01000	01000	01000	01000	01000	01000
b	10000	10000	00100	00100	00100	00100	00100
c	01000	10000	10000	00010	10000	10000	00100
d	00100	00100	10000	10000	00010	00010	00010
e	00010	00010	00010	10000	10000	00001	00001
f	00001	00001	00001	00001	00001	10000	10000

Table 7: Input vectors corresponding to graph in Fig. 3

Circuit	Original Peak Power	Final Peak Power	Percentage Reduction
Zero Delay Model			
c432	69	44	36.23
c499	96	60	37.50
c1355	184	167	9.24
c1908	45	29	35.56
c5315	458	263	42.58
c6288	1554	1287	17.18
c7552	248	142	42.74
General Delay Model			
c432	23282	15326	34.17
c499	10143	6976	31.22
c1355	1234590	643726	47.86
c1908	744257	85673	88.49
c5315	1219804	124163	89.82
c6288	2.02×10^{20}	4.91×10^{19}	75.65
c7552	1134830	232955	79.47

Table 8: Peak Power Minimization on ISCAS85 circuits

Circuit	Original Total Power	Final Total Power	Percentage Reduction
Zero Delay Model			
c432	2296	1393	39.33
c499	5597	4487	19.83
c1355	10346	7776	24.84
c1908	3245	1586	51.12
c5315	35473	14161	60.08
c6288	28625	15266	46.67
c7552	21647	5987	72.34
General Delay Model			
c432	243260	33193	86.35
c499	570713	416134	27.09
c1355	19171113	6463078	66.29
c1908	9841097	713207	92.75
c5315	28471812	3028065	89.36
c6288	6.11×10^{20}	7.51×10^{18}	98.77
c7552	42726168	2448118	94.27

Table 9: Total Power Minimization

Circuit	Original Peak Power	Final Peak Power	Percentage Reduction (Original)	Original Sequence Length	Final Sequence Length	Penalty Factor	Percentage Reduction (No Repeat)
Zero Delay Model							
c432	69	31	55.07	73	100	1.37	29.54
c499	96	54	43.75	99	170	1.72	10.00
c1355	184	124	32.60	98	107	1.09	25.75
c1908	45	20	55.55	145	177	1.22	31.03
c5315	458	170	62.88	369	381	1.03	35.36
c6288	1554	1284	17.37	82	85	1.04	0.23
c7552	248	85	65.72	277	311	1.12	40.14
General Delay Model							
c432	23282	1095	95	73	90	1.23	92.85
c499	10143	5229	48	99	155	1.57	25.04
c1355	1234590	201779	84	98	114	1.16	68.65
c1908	744257	34931	95	145	167	1.15	59.23
c5315	1219804	33794	97	369	421	1.14	72.89
c6288	2.02×10^{20}	1.96×10^{18}	99	82	99	1.21	96.00
c7552	1134830	44249	96	277	302	1.09	81.00

Table 10: Peak Power Minimization with repeating vectors on ISCAS85 circuits

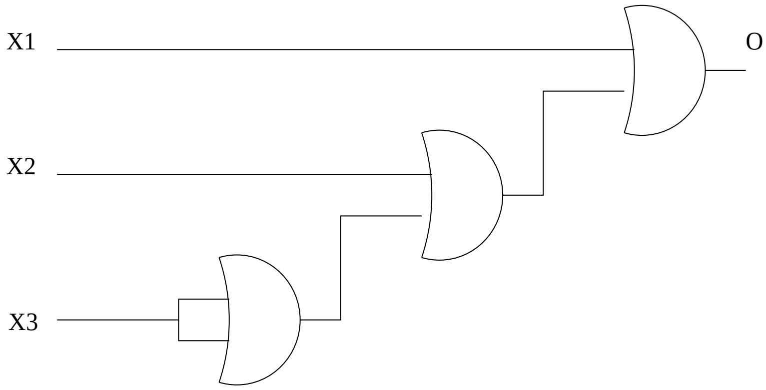

Figure 1:

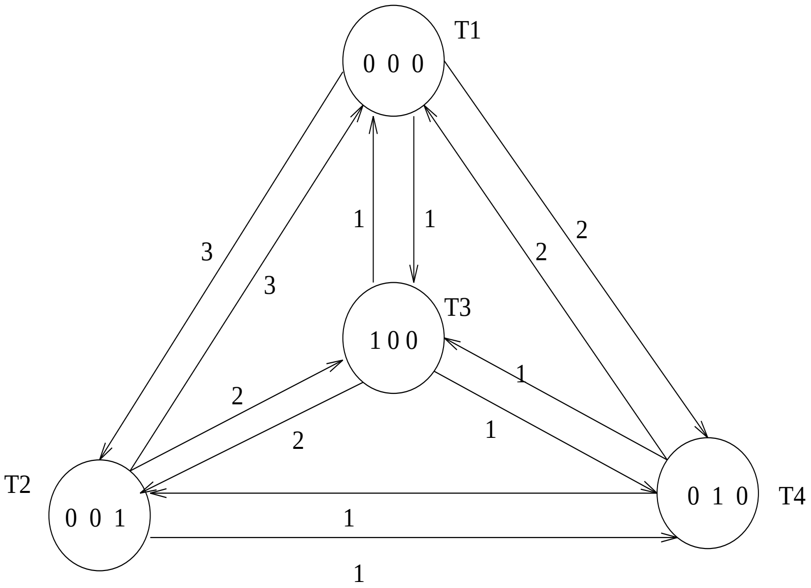

Figure 2: Input Transitions Graph for circuit in Fig 1

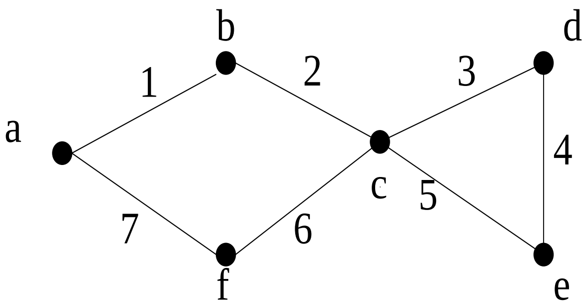

Figure 3: Graph G for finding HP

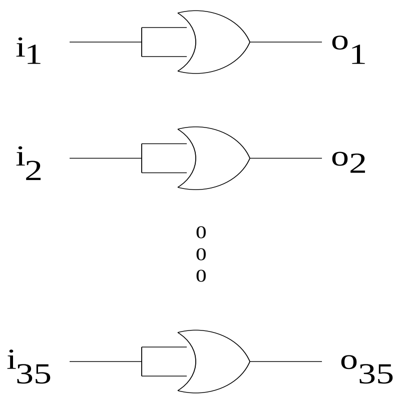

Figure 4: circuit corresponding to the graph in Figure 3

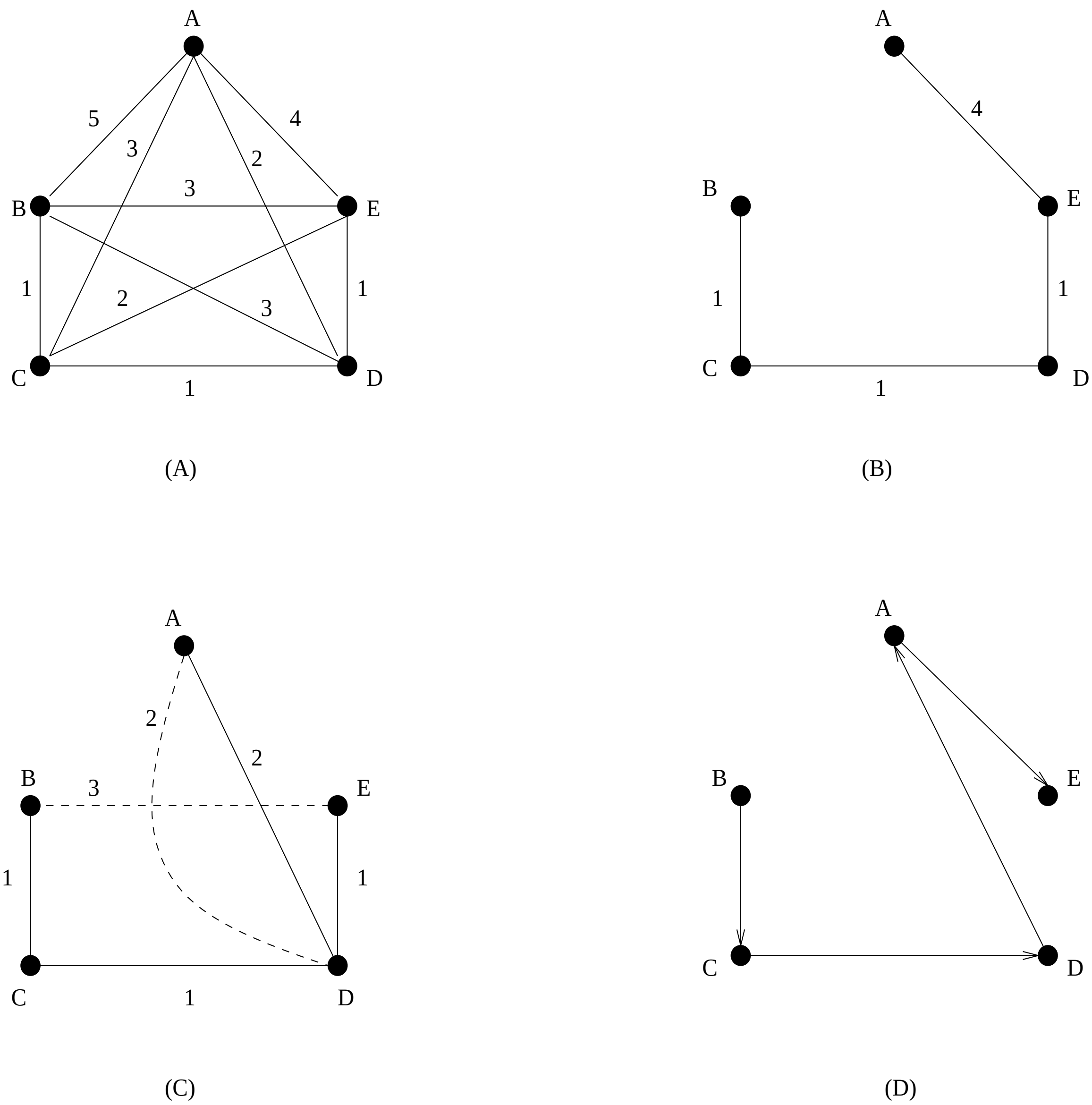

Figure 5: Examples for heuristics GET_PEAK and CHRFD_TOTAL

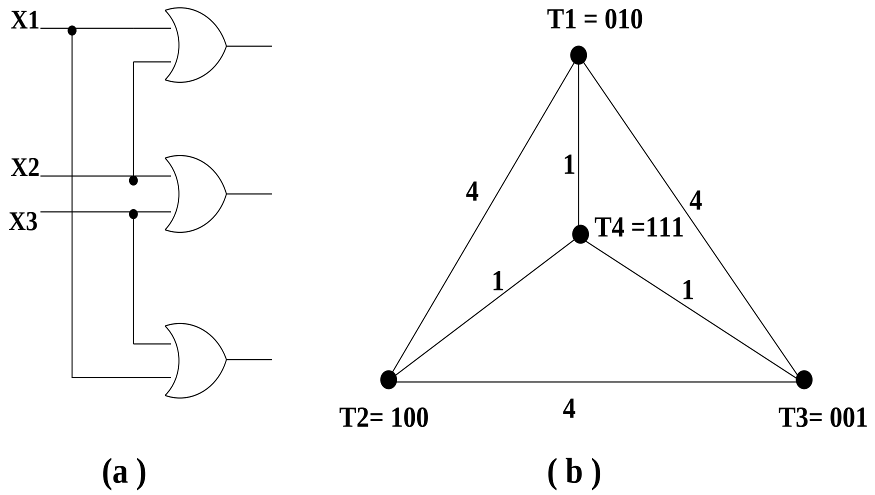

Figure 6: Circuit and ITG corresponding to vector set T

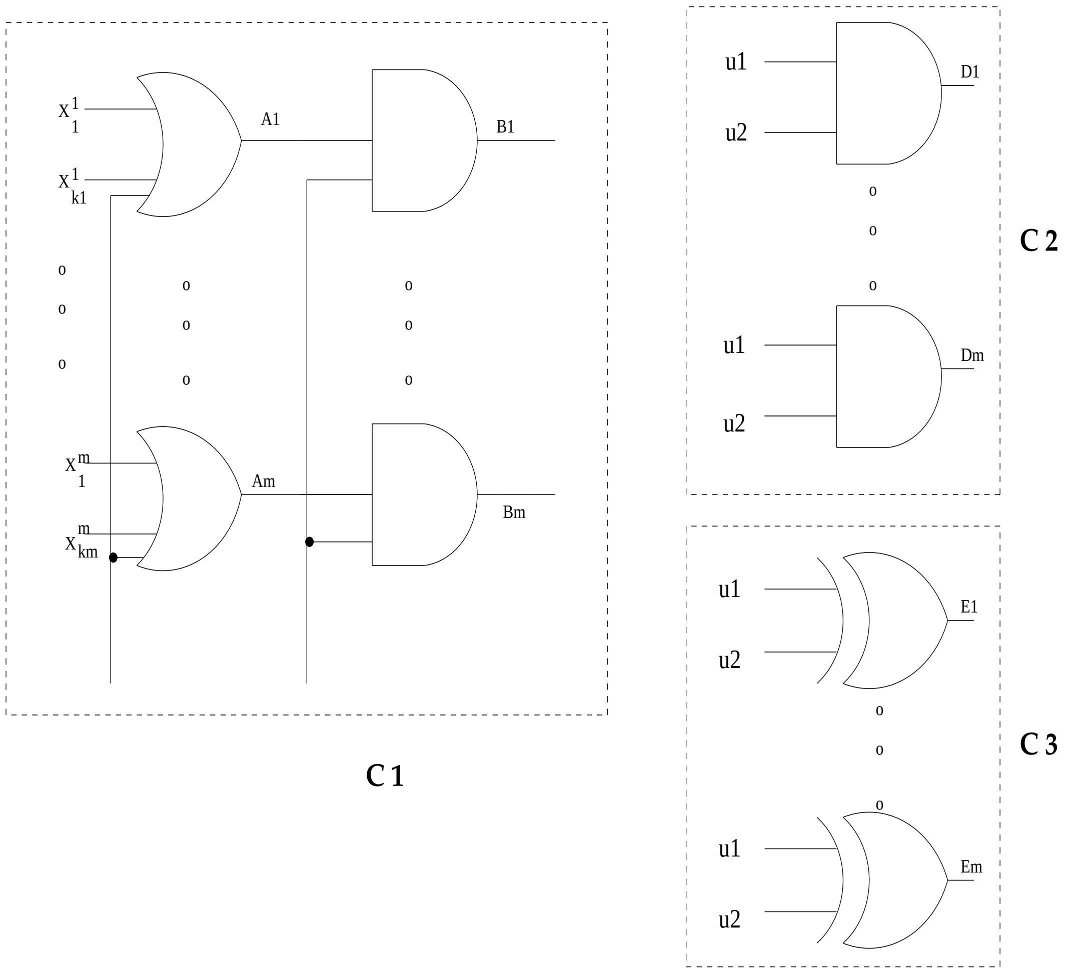

Figure 7: Circuit corresponding to instance of OSP_AD