

TOPS: A Target-Oriented Partial Scan Design Package Based on Simulated Annealing

C.P. RAVIKUMAR and H. RASHEED

Department of Electrical Engineering, Indian Institute of Technology, New Delhi 110016, INDIA

(Received December 3, 1992, Revised March 22, 1993)

In this paper, we describe algorithms based on Simulated Annealing for selecting a subset of flip-flops to be connected into a scan path. The objective for selection is to maximize the coverage of faults that are aborted by a sequential fault simulator. We pose the problem as a combinatorial optimization, and present a heuristic algorithm based on Simulated Annealing. The SCOAP testability measure is employed to assess the selection of flip-flops during the course of optimization. Our algorithms form a part of an integrated design package, TOPS, which has been designed as an enhancement to the OASIS standard-cell design automation system available from MCNC. We discuss the TOPS package and its performance on a number of ISCAS'89 benchmarks. We also present a comparative evaluation of the benchmark results.

Key Words: *Design for Testability, Partial Scan Design, Target Faults, Simulated Annealing*

1 INTRODUCTION

In designing digital integrated circuits, a popular method for achieving testability is *full scan design*, where the flip-flops in the circuit are threaded into a chain, which can perform as a shift register when the circuit is placed in the *test* mode. A test vector is shifted serially into the shift register, and the response of the circuit is shifted out serially for observation [1]. One of the serious drawbacks of full scan design is the area overhead that results due to

- the extra wiring required to thread the flip-flops into a chain
- the extra space occupied by flip-flops when they are modified into scan cells.

In order to reduce the overhead of full scan and still maintain the advantages offered by the scan methodology, several authors have designed techniques for *partial scan design* where only a subset of flip-flops are scanned. In order to guide the selection of as small a subset as possible, the following approaches have been considered in the literature. In all these approaches, the essential idea is to exploit the underlying structure of the circuit; a graph called *S-graph* is derived which captures the structural information. An *S-graph* has one node corresponding to each of the flip-flops in the original circuit, and a

directed edge from node i to node j if (and only if) there exists a *combinational path* from the flip-flop i to flip-flop j in the circuit. In other words, an edge (i, j) in *S-graph* represents the combinational logic that separates the output of flip-flop i from the input of flip-flop j .

- Cheng and Agrawal observed that a circuit is poorly testable if its *S-graph* has long cycles [3]. When a flip-flop in the circuit is converted into a scan cell, the operation corresponds to deleting the corresponding node in the *S-graph*. In order to achieve high testability, the *S-graph* must be rendered acyclic by deleting as few nodes as possible. We refer to this acyclic graph as *A-graph* in this paper. This problem, also called the *feedback vertex cover set problem*, is known to be NP-complete. Cheng and Agrawal gave heuristic algorithms to obtain small feedback vertex covers.
- Gupta and Breuer introduced the concept of *balance* in a sequential circuit in addition to the concept of acyclicity [6]. The advantage of transforming the circuit structure into a balanced acyclic structure is that a combinational test vector generator can be employed to generate test patterns for the partially scanned circuit.
- Some authors believe that poor testability of a circuit may be ascribed to its large *sequential*

depth. The sequential depth of a circuit is defined as the length of the longest path in the A -graph. Thus, Lin and Reddy [8] considered the following problem. Delete the fewest number of nodes from the S -graph such that the graph becomes acyclic and the length of the longest path in the A -graph is minimized. The authors described a two-step heuristic algorithm for the problem; in the first step, cycles in the S -graph are removed by deleting nodes, and in the second step, the sequential depth is reduced by deleting more nodes. Each deleted node forms part of the final partial scan chain.

- An entirely different approach may be considered for partial scan. If a sequential test generation algorithm is available, one could attempt to run this algorithm on the unscanned circuit to generate test vectors within a certain time limit to detect as many faults as possible. The remaining faults, also called *aborted faults* or *target faults*, are the only ones which need to be addressed by the partial scan mechanism. In a similar setup, one may use a random test vector generator and a sequential fault simulator to generate a list of target faults.

Chickermane and Patel observed empirically that hard-to-detect faults tend to lie in strongly connected components (SCCs) of the S -graph [4]. As a result, the partial scan system proposed by these authors examines the SCCs in the S -graph and computes a *profit function* p_i for each node i . If a node i is part of cycles i_1, i_2, \dots, i_k , the profit p_i obtained by scanning the node i is given by

$$p_i = \sum_{j=1}^k W(i_j) \quad (1)$$

where $W(i_j)$ indicates the weight of cycle i_j . In turn, the weight of a cycle is the number of hard-to-detect faults that lie on the cycle. The objective in [4] is to select cells for scanning, such that the cumulative profit is maximized without exceeding the upper bound on the cost of scanning. The authors presented heuristic techniques to obtain good solutions to the above optimization problem.

In this paper, we apply the Simulated Annealing algorithm [7] to the partial scan design problem. Unlike the references cited above, our algorithm does

not rely on structural properties such as cycles, weighted cycles, sequential depth, or strongly connected components. Instead, we regard the search space in a uniform manner when looking for a solution to the partial scan problem. In our approach, the structural information is used to evaluate the testability of a configuration during the course of annealing. In the following section, we describe the details of the algorithm. In Section 4, we discuss the results obtained by applying our technique to several ISCAS benchmark circuits. We also implemented a greedy algorithm to compare the results obtained through the annealing procedure. The greedy algorithm and its results are also discussed in Section 4. Our algorithms are integrated into a package named TOPS, which enhances the OASIS design automation system [9] available from MCNC by providing partial scan capability. We discuss the salient features of TOPS in Section 3.

2 PARTIAL SCAN FOR TARGET-FAULT DETECTION

The Simulated Annealing algorithm [7] has been widely applied to a number of optimization problems in Design Automation such as floorplanning, partitioning, placement, and routing (see [10], [11]). The annealing algorithm is similar to an iterative improvement algorithm in nature, with the exception that inferior states are also accepted with a certain probability. This probability is a function of a parameter known as the temperature, which is decreased slowly during the course of the algorithm. At higher temperatures, the annealing algorithm accepts inferior states with a high probability, hence behaving like a random search procedure. As the temperature is lowered, the acceptance probability for inferior states drops, and the procedure attains greedy characteristics at temperatures close to 0. The reader may refer to [7] for a complete description of the annealing algorithm.

The partial scan design problem may be phrased in terms of a state-space search as follows. A state, or configuration, consists of a subset of cells. Let n be the total number of flip-flops in the circuit. A subset of k flip-flops can be selected in $\binom{n}{k}$ ways, and hence there are $\sum_{k=1}^n \binom{n}{k} = 2^n - 1$ possible solutions to the partial scan problem. A perturbation of a state consists of deleting a flip-flop from the present configuration, or adding a flip-flop to the present configuration, or both. The *cost* of a configuration is the area overhead that results by scanning the flip-flops

which correspond to the present configuration. Our cost measure consists of two components.

- the increase in functional area
- an estimate of the increase in wiring area due to scan path

The *profit* of a configuration is measured by the SCOAP testability index of the configuration [5], as extended to the case of *target faults*. We will describe our cost and profit function in more detail later in this section.

The procedure *Anneal* is outlined in Figure 1. The annealing schedule is given by the initial temperature T_0 , the final temperature T_f , the cooling rate α , the number of iterations per temperature M , and the rate β at which M is increased progressively over temperatures. The initial configuration S_k is also an input to the procedure; it consists of a randomly selected k -sized subset of the flip-flops in the circuit. The function *perturb* returns a new configuration S'_k by perturbing the subset S_k as explained earlier. The new configuration is accepted under two conditions.

- If both the profit and cost parameters of the new configuration are better, then S'_k is accepted.
- When the new configuration is inferior in either the cost measure or the profit measure, or both, then the Metropolis criterion [7] is separately applied to both cost and profit terms. S'_k is accepted if both the Metropolis criteria succeed.

2.1 Calculating the Profit

The *profit* function is an implementation of the SCOAP testability analysis procedure. In the SCOAP terminology, $SC^1[x]$ indicates the sequential 1-controllability of a line x . $SC^0[x]$ is defined similarly. The sequential observability of a circuit is denoted by SO . The conventional SCOAP testability index for a sequential circuit is given by

$$U = \sum_x SC^1[x] + SC^0[x] + SO[x] \quad (2)$$

where the summation is carried over all lines x . In our work, since we are mainly interested in target faults, we define an alternate testability index T . Let $\delta^r(x)$ be a 0-1 function which evaluates to 1 if and only if there exists a target fault of the form x -stuck-at- r .

$$T = \sum_l \delta^0[l] \cdot SC^1[l] + \sum_l \delta^1[l] \cdot SC^0[l] + \sum_l (\delta^0[l] + \delta^1[l] - \delta^0[l] \cdot \delta^1[l]) \cdot SO[l] \quad (3)$$

where the summation is carried over all lines l .

```

procedure Anneal( $S_k, T_0, T_f, \alpha, \beta, M$ );
  (*  $S_k$  is the initial configuration, with  $k$  flip-flops.
    $T_0$  is the initial temperature.  $T_f$  is the final temperature.
    $\alpha$  and  $\beta$  are the cooling parameters.
    $M$  is the number of trials attempted at any temperature. *)
begin
   $T := T_0$ ;
  while ( $T_f < T$ ) do begin
    for  $i := 1$  to  $M$  do begin
       $S'_k := \text{perturb}(S_k)$ ;
       $\Delta_p = \text{profit}(S'_k) - \text{profit}(S_k)$ ;
       $\Delta_c = \text{cost}(S'_k) - \text{cost}(S_k)$ ;
      if (( $\Delta_p > 0$ ) and ( $\Delta_c < 0$ ))
        or (( $\text{random} < e^{\Delta_p/T}$ )
          and ( $\text{random} < e^{-\Delta_c/T}$ )) then
         $S_k = S'_k$ ;
    end
     $T := T * \alpha$ 
     $M := M * \beta$ 
  end
end

```

FIGURE 1 Simulated Annealing Algorithm for Partial Scan Selection

Scanning a flip-flop F affects the T index in two ways.

- the controllabilities of lines that are reachable from the output of F may improve, and
- the observabilities of lines which lead to the input of F may improve.

Let $\gamma(i, j)$ be a 0-1 function which evaluates to 1 if and only if there exists a directed path from line i to line j . Let I and O respectively indicate the input and output of flip-flop F . Let T_S^{psc} denote the testability index for the partially scanned circuit, where S is the set of flip-flops selected for scan. The expression for T_S^{psc} is given below.

$$\begin{aligned} T_S^{psc} = & \sum_{\forall F \in S} \sum_l \gamma(O, l) \cdot \delta^0[l] \cdot SC^1[l] \\ & + \sum_l \gamma(O, l) \cdot \delta^1[l] \cdot SC^0[l] \\ & + \sum_l \gamma(l, I) \cdot (\delta^0[l] + \delta^1[l]) \\ & - \delta^0[l] \cdot \delta^1[l] \cdot SO[l] \end{aligned}$$

The reader should note that while evaluating T_S^{psc} , the line observabilities $SO[l]$ and controllabilities $SC^0[l]$ and $SC^1[l]$ must be recomputed for the partially scanned circuit. For simplicity, we have used the same notation to indicate the observabilities and controllabilities for both unscanned and scanned circuits.

The *profit* function computes the difference $T_S^{psc} - U$ for a given circuit and a given subset of k flip-flops selected for partial scan. This is done in two steps.

1. a forward breadth-first-search to identify the lines whose controllabilities are affected by the partial scan. If the search process encounters a line l such that l -stuck-at- r is a target fault, then the procedure accumulates the value $SC^{1-r}[l]$ into PSC^{1-r} (sequential $1 - r$ controllability of the partially scanned circuit).
2. a backward breadth-first-search to identify the lines whose observabilities are affected by the partial scan. If, during the search, the procedure encounters a line l such that l -stuck-at- r is a target fault, then the procedure accumulates $SO[l]$ into PSO . Here, PSO indicates the sequential observability of the partially scanned circuit.

It is clear that $T_S^{psc} = PSO + PSC^0 + PSC^1$. The unscanned testability index U is computed by the

program once initially. The worst-case time complexity of calculating the profit function is linear in the number of nodes of the circuit.

2.2 Computing the Cost

The computation of the cost of a configuration depends necessarily on the implementation technology and the layout style. In our work, the target technology is 2μ SCMOS, and the layout style is standard cell. The size of the cell which implements the unscanned flip-flop is $58.0 \times 64.0\lambda^2$. The scanned cell takes an area of $58 \times 72.0\lambda^2$. The functional area overhead due to a scanning a single cell is therefore $464\lambda^2$ units. An increase in wiring area is also incurred due to partial scan; this area is estimated as follows. We generate a placement P of the unscanned circuit using the available standard-cell placement tool. We assume that the same placement of cells is used in the scanned circuit as well. Under this assumption, we have implemented an estimator for the increase in wiring area. This estimator first calculates the increase in track density for each channel due to the extra wiring required to implement the scan path. The order in which the scan cells are connected into a scan path is crucial in the above calculation. However, the problem of determining the best ordering is an instance of the Travelling Salesperson Problem, and is hence computationally difficult. We generate a good heuristic solution to the problem and use this ordering to estimate the increase in the channel track-densities. The cumulative increase in track densities, multiplied by the width of a single track, is used as an estimate of the wiring overhead.

2.3 A Greedy Algorithm

In this section, we describe a greedy approach to the partial scan design problem. We use this procedure to generate a *good* initial solution to the selection problem; this initial solution is passed on as input to the *Anneal* procedure. As a result, annealing can begin at a relatively low temperature. We found this method very effective in reducing the total computational requirements of Simulated Annealing.

The essential idea behind the procedure *Greedy* is to rank each of the flip-flops individually by its *target testability improvement index* which we define below. Given a sequential circuit with n flip-flops f_1, f_2, \dots, f_n , the target testability improvement index of a flip-flop f_i is defined as

$$t(f_i) = T_{\{f_i\}}^{psc} - U \quad (4)$$

```

procedure Greedy(n, limit, C);
    (* n is the number of flip-flops in the circuit.
       limit is the upper bound on the area overhead that can be tolerated.
       C is the circuit description. *)
begin
    for i := 1 to n do begin
        S := {i};
        R(i).ff := i
        R(i).gain := profit(S);
    end
    sort(R); (* Ascending Order *)
    P := {};
    totalcost := 0;
    for i := 1 to n do begin
        P := {R(i).ff} ∪ P
        totalcost := totalcost + FOVHD + wire_ovhd(P);
        if totalcost ≥ limit return(P);
    end
end

```

FIGURE 2 Greedy Algorithm for Partial Scan Selection. FOVHD is the functional area overhead contributed by a single scan cell. *wire_ovhd* is a procedure which estimates the wiring area overhead for a given subset *P* of flip-flops

In other words, $t(f_i)$ measures the improvement in target testability by scanning *only* the flip-flop f_i . Since the functional area overhead resulting from scanning any of the flip-flops is the same, a greedy strategy for scan selection is to pick those flip-flops with the highest values for t . The upper limit on the area overhead is used to guide the number of selected flip-flops. The complete procedure is shown in Figure 2. If the target faults are distributed uniformly over the circuit and not clustered in a small region, the greedy algorithm is likely to perform well. We discuss the experimental results on the greedy algorithm in the Section 4.

3 THE TOPS PACKAGE

The algorithms discussed in the previous section have been coded in the C programming language on a Sun SPARC workstation. In this section, we discuss the overall organization of the TOPS package. As mentioned earlier, the TOPS package is designed to interface with the OASIS design automation software from MCNC [9]. The input to the package is the structural description of a sequential circuit given in either the ISCAS format, HILO format, or the VPNR format. VPNR (Vanilla Place and Route) is a circuit description language developed by MCNC. The advantage of using VPNR is that it allows us to describe the circuit as various levels of detail [9]. If the input is provided in ISCAS or HILO format, we

internally convert it into VPNR format (see Figure 3). The VPNR description can be compiled into a layout using two programs **cplrt** and **dtglrt**, which generate a standard-cell placement and routing, respec-

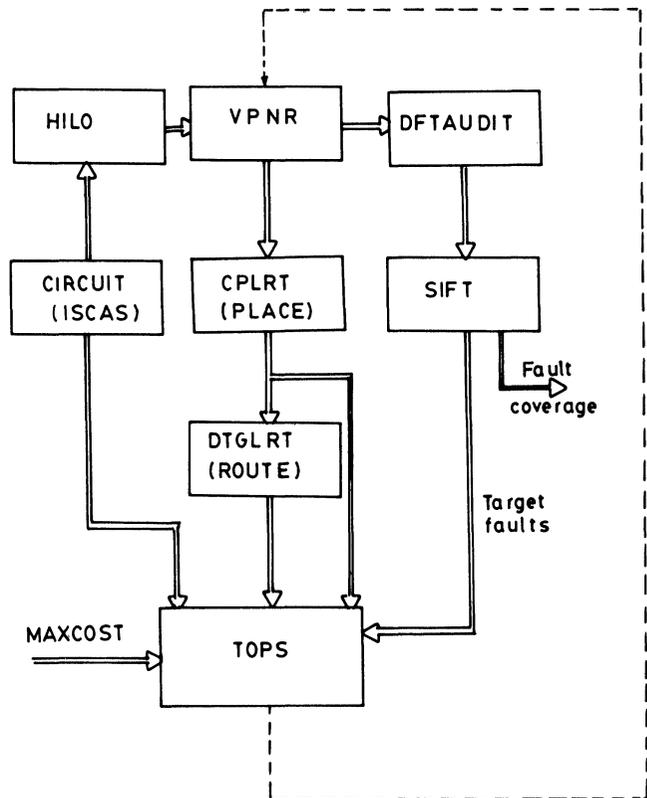


FIGURE 3 The Organization of TOPS package

tively. The layout is generated using unscanned flip-flops (cell **dr2s**). The **dftaudit** program is used to prepare a circuit description as required by the sequential fault simulator **sift**. **sift** applies a specified number of random test patterns to the circuit and reports the list faults which could not be detected. The number of random test patterns, N , plays an important role in the performance of the partial scan design system. If N is chosen large, the list of target faults may become smaller, giving less work to the partial scan selection algorithms; however, the fault simulator would then require an excessive amount of CPU-time. Of course, there are hard-to-test circuits (such as the s420 benchmark from ISCAS) for which increasing N beyond a certain limit does not help in reducing the number of target faults. Presently, we select N by a trial-and-error procedure where N is initially set to 1000 and doubled in every iteration. If two successive values of N do not reduce the number of target faults, we use the smaller value of N to generate the final list of target faults. If the final value of N selected by our procedure is N_f , it is easy to see that we need $\log_2(N_f/1000)$ runs of sequential fault simulation. Assuming linear-time performance from the fault simulator, the total time spent on fault simulation is seen to be $O(2N_f - 1000)$.

The TOPS package receives as inputs the original netlist, the placement information, routing information, and the list of target faults. After the selection process, the TOPS package modifies the layout description file (VPNR format) to convert the selected flip-flops into scan cells (cell type **dsr2s**). The **scan** program is used to thread the flip-flops into a scan path.

4 EXPERIMENTAL RESULTS

The effectiveness of the TOPS package was tested against several standard ISCAS'89 benchmark circuits. These are tabulated in Table 1. NF is the number of flip-flops in the original circuit. UFC indicates the unscanned fault coverage obtainable by running a random test pattern generator as explained in the previous section. SS is the size of the scan set (number of flip-flops selected for scan). $SFC(G)$ is the fault coverage obtained through the scan set selected by the *Greedy* procedure. $SFC(G + A)$ is the fault coverage obtained by first running the *Greedy* procedure and then improving the solution by running the *Anneal* procedure.

We compared our benchmark results with other published work, namely, [4] and [8]. Our results were better in three cases (s298, s386, and s510), and comparable in the remaining cases. It is to be noted that in [4], and [8], the authors used a deterministic sequential test pattern generator on the unscanned circuit. As a result, the unscanned fault coverage reported by these authors is significantly higher than those in Table 1. As an example, for the circuit s526, the unscanned fault coverage is 49.4% in [8]; the random test pattern generator which we used could only generate a fault coverage of 9.9%. Similarly, the unscanned fault coverage for s386 is 67.44% in our system, whereas it is 81.8% in [4]. Table 1 also throws light on the performance of the Simulated Annealing algorithm in comparison to the greedy algorithm. The greedy algorithm competes with the annealing algorithm in most cases, but the annealing algorithm performs better in three of the nine cases tested. This is to be expected, since the greedy al-

TABLE I
Benchmark Results on TOPS

Circuit	NF	UFC	$NUFC$	SS	$SFC(G)$	$SFC(G + A)$
s208	8	49.3	8K	3	67	71.16
s298	14	87.98	128K	3	100	100
s386	6	67.44	32K	2	96.61	96.61
s420	16	33.04	2K	8	64.42	83.95
s510	6	0.00	1K	1	100	100
s526	21	9.9	16K	8	94.95	98.74
s820	5	50.11	128	1	89.88	89.88
s832	5	35.17	1K	1	98.39	98.4
s1238	18	76.75	128K	2	94.70	94.70
s1488	6	61.98	512K	1	99.5	99.5
s5378	179	68.99	512K	54	97.17	98.7

gorithm may select a flip-flop f_2 following the selection of f_1 based on the testability improvement index of f_2 . However, it may be the case that many of the faults covered by f_2 are already covered by f_1 . Since the greedy procedure does not reverse its decisions, it is likely to get stuck at a local optimum solution. The annealing procedure, on the other hand, can start with the solution generated by *Greedy* and improve it further by applying local transformations.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have described a tool suite known as TOPS which we have implemented around an existing Design Automation package. TOPS is a hybrid of two heuristics for the partial scan selection problem. A greedy procedure is used to first select a good starting solution, which is improved iteratively using the Simulated Annealing procedure. The essential idea behind TOPS is to first use a sequential test generator to catch as many faults as possible. The aim of the scan selection algorithms is then to catch the remaining faults using the fewest number of scan cells. TOPS is an integrated package and interfaces to a standard-cell design automation package. We have described the performance of TOPS on several standard benchmark circuits. The benchmark results indicate that TOPS provides an effective method for partial scan selection. We are presently working on extending the TOPS package on several fronts. First, we feel that the greedy algorithm can be further improved by posing the selection problem as a *linear assignment problem*. Second, we are studying the relationship between the structural properties of the circuit (such as acyclicity) and their relationship to circuit testability. The greedy procedure and the annealing algorithm presented in this paper do not directly take into account such structural properties. Instead, they rely on the SCOAP testability measure in deciding the contribution of a flip-flop to total circuit testability. SCOAP, in turn, uses the circuit structure in assessing the controllabilities and observabilities of individual nodes. In a recent survey conducted by Chandra and Patel [2], the authors found SCOAP to be the most reliable of the existing testability measures. However, the work of Cheng and Agrawal [3], Chickermane and Patel [4], and Lin and Reddy [8] have indicated the usefulness of several structural properties in judging the testability

of a circuit. We have presently initiated an effort to improve TOPS through the use of similar structural properties such as acyclicity.

Acknowledgements

We thank Dr. Raghu Hudli of IBM Fishkill for helping us verify the correctness of our SCOAP implementation by supplying us the SCOAP testabilities computed independently.

References

- [1] M. Abromovici, M.A. Breuer, and A.D. Friedman. *Digital Systems Testing and Testable Design*. Computer Science Press, NY, 1990.
- [2] S.J. Chandra and J.H. Patel. Experimental evaluation of testability measures for test generation. *IEEE Transactions on CAD*, 8(1):93–97, January 1989.
- [3] K.-T. Cheng and V.D. Agrawal. A partial scan method for sequential circuits with feedback. *IEEE Transactions on Computers*, 39(4):544–548, April 1990.
- [4] V. Chickermane and J.H. Patel. A fault oriented partial scan design approach. In *Proceedings of the ICCAD*, November 1991.
- [5] L.H. Goldstein. Controllability/observability analysis of digital circuits. *IEEE Transactions on Circuits and Systems*, 26:685–693, 1979.
- [6] R. Gupta, R. Gupta, and M.A. Breuer. BALLAST: A methodology for partial scan design. In *Proceedings of the Int. Symp. on Fault-Tolerant Computing*, June 1989.
- [7] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 13, 1983.
- [8] D.H. Lee and S.M. Reddy. On determining scan flip-flops in partial scan designs. In *Proceedings of the Design Automation Conference*, pages 322–325, 1990.
- [9] MCNC. *Open Architecture Silicon Implementation Software—User's Manual*. Microelectronics Corporation of North Carolina, USA, 1990.
- [10] S. Sechen and A. Sangiovanni Vincentelli. The timberwolf placement and routing package. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 522–527, 1984.
- [11] M.P. Vecchi and S. Kirkpatrick. Global wiring by simulated annealing. *IEEE Transactions on Computer-Aided Design*, 2(4):215–222, October 1983.

Biographies

C.P. RAVIKUMAR obtained his B.E. degree (electronics) from Bangalore University, India (1983), M.E. (Computer Science) from Indian Institute of Science (1987) and Ph.D. (Computer Engineering) from University of Southern California (1991). Since 1991, he is with the Department of Electrical Engineering, IIT Delhi where he is an assistant professor. His research interests are in the areas of VLSI Design Automation, Parallel Architectures and Algorithms and Combinatorial Optimization.

HAROON RASHEED obtained his B.E. (Electronics) from REC, Calicut (India) in 1989, and M.TECH (Integrated Electronics and Circuits) from Indian Institute of Technology, Delhi in 1992. Since 1992, he has been employed at S.G.S. Thomson, New Delhi, as a design engineer.

