# Configuring Multiple Scan Chains for Minimum Test Time*

Sridhar Narayanan, Rajesh Gupta+ and Melvin Breuer

Dept. of Electrical Engg.–Systems
University of Southern California
Los Angeles, CA 90089-2560

+IBM East Fishkill
Zip 3A1/306, Route 52
Hopewell Jct., NY 12533-0999

## Abstract

*To reduce the high test time for serial scan designs, the use of multiple scan chains has been proposed. In this paper we consider the problem of optimally constructing multiple scan chains so as to minimize overall test time. Rather than follow the traditional practice of using equal length chains, we allow the chains to be of different lengths, and show that this can lead to lower test times. The main idea in our approach is to assign those scan elements that are more frequently accessed to shorter scan chains. Given a design with $N$ scan elements, and given that $k$ scan chains are to be used for applying tests, we present an algorithm of complexity $O(kN^2)$ for configuring the chains such that the overall test application time is minimized. By analyzing a range of circuit topologies, we demonstrate test time reductions as large as 40% over equal length chain configurations.*

## 1 Introduction

Full and partial scan designs [1, 2] are used to reduce the cost of test generation for sequential circuits. However the use of scan design techniques leads to high test application cost because test data (both input patterns and output results) needs to be shifted in and out of the circuit serially through a single scan chain. Due to the ever increasing amount of test data and the mounting cost of test equipment, the importance of reducing test time assumes great significance.

The primary focus of this paper is on minimizing the shifting time for scan designs by the use of multiple scan chains. Each scan chain requires its own scan-in and scan-out pins, which may be multiplexed with system I/O pins. The popular approach to multiple scan chaining is to make all chains have equal length [3]. The scan flip-flops are assigned to specific chains based purely on geometric constraints, in an attempt to minimize the routing area overhead. In [4], heuristic rules are used to guide the assignment and ordering of scan registers in the chains. However the approach does not consider the number of test patterns required to test the various logic blocks and this can have a significant impact on the total test time.

We present a structured methodology to configure multiple scan chains so as to minimize the test application time. In most scan designs there is a broad distribution in the frequency of usage of the different scan flip-flops [5]. The assignment of flip-flops to the chains in our methodology is performed based on the number of tests applied through each flip-flop. Some flip-flops are involved in applying a larger number of tests than other flip-flops. By placing these flip-flops in shorter chains, we demonstrate test time reductions as large as 40% over equal length chain configurations. An efficient algorithm based on dynamic programming is developed to optimally configure the multiple scan chains.

*This work was supported by the Defense Advanced Research Projects Agency and monitored by the Federal Bureau of Investigation under Contract No. JFBI90092.

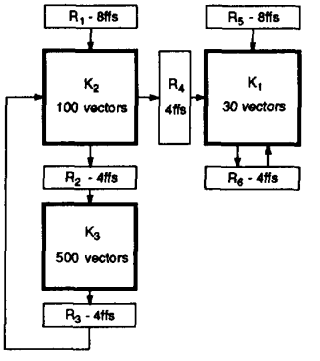


Figure 1: Circuit 1.

## 2 Test Application Scheme

Throughout the paper we will assume that a full scan methodology is used in the selection of scan flip-flops although the results are also applicable to certain partial scan designs [6]. In a full scan design of a sequential circuit, the scan flip-flops provide complete access to the combinational portions of the circuit. This combinational logic can be partitioned into a set of kernels, or disjoint portions of logic that can be tested independently. A kernel is essentially a maximal region of connected combinational logic. All inputs/outputs of a kernel are connected to either primary I/O or to scan flip-flops. Test generation can be carried out as a separate process on each kernel to determine the number of test patterns required to fully test the kernel. Note that since the kernels are disjoint, every scan flip-flop and every primary I/O can apply test patterns to at most one kernel and receive test results from at most one kernel.

Consider the circuit shown in Fig. 1 consisting of three combinational kernels $K_1, K_2$ and $K_3$ and six scan registers. The scan flip-flops in each of the registers are assigned to the four chains $C_1, C_2, C_3$ and $C_4$ as shown in Fig. 2. Note that two of the flip-flops in register $R_4$ are assigned to chain $C_1$ and the other two to chain $C_4$. If any flip-flop in a chain $C$ is used to either apply test patterns or receive test results from a kernel $K$, we say that $C$ is *involved* in testing $K$. Hence chains $C_1, C_2$ and $C_4$ are involved in testing $K_1$; $C_1, C_3$ and $C_4$ are involved in testing $K_2$; and $C_3$ is involved in testing $K_3$. Using the four chains, we



0-8186-3010-8/92 $03.00 © 1992 IEEE

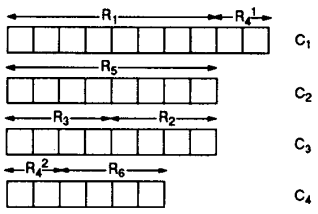


Figure 2: A four-chain configuration for Circuit 1.

| Test session | Kernels tested | No. of patterns | Chains active | Chain cycle |
|---|---|---|---|---|
| $TS_1$ | $K_1, K_2, K_3$ | 30 | $C_1, C_2, C_3, C_4$ | 10 |
| $TS_2$ | $K_2, K_3$ | 70 | $C_1, C_3, C_4$ | 10 |
| $TS_3$ | $K_3$ | 400 | $C_3$ | 8 |

Table 1: Summary of test sessions for Circuit 1

apply the appropriate number of tests to each kernel using the *overlapped application scheme* [7].

In our example, the test begins with the first test session, denoted by $TS_1$, in which 30 patterns are applied to $K_1$, which is the kernel with the fewest test patterns. Simultaneously, the first 30 patterns for both $K_2$ and $K_3$ are also applied. In order to test the three kernels, all four chains are used and are said to be "active" in this test session. The test session $TS_1$ is summarized in the first row of Table 1. To apply each test pattern in this session, the corresponding input patterns for the kernels are merged and shifted simultaneously into the four scan chains, while at the same time the previous test results are shifted out. A scan chain is said to be **flushed** if the number of clock cycles for which it is placed in the shift mode is equal to the length of the scan chain (in terms of the number of flip-flops). Since the longest active chain $C_1$ has 10 flip-flops, 10 clock cycles are required to entirely flush all the four chains; this number is called the **chain cycle** of the session $TS_1$. Note that all the chains active in a session operate in a synchronized manner. A single clock cycle is required for the test patterns to propagate through the kernels before the test results can be shifted out and the next pattern shifted in. Thus the duration of $TS_1$ in clock cycles is $30(10 + 1) = 330$.

After the first test session is completed, $K_1$ has been fully tested but there are 70 remaining patterns to be applied to $K_2$ and 470 to $K_3$. The second test session $TS_2$ is carried out in a similar manner as the first, and is summarized in the second row of Table 1. During $TS_2$, 70 patterns are applied to $K_2$ and $K_3$ and only chains $C_1$, $C_3$ and $C_4$ are active. The chain cycle of $TS_2$ is therefore 10 and its duration is $70(10 + 1) = 770$ clock cycles. Finally in the third test session $TS_3$, the remaining 400 patterns are applied to $K_3$ using chain $C_3$ and the time required is $400(8+1) = 3600$ clock cycles. Thus, the total test time is equal to $330 + 770 + 3600 + 10 = 4710$ clock cycles where the last 10 clock cycles are required to flush out the final results from all the chains.

## 3 Optimal Chains

If a single scan chain is used to test Circuit 1, the chain cycle used in each session is equal to 32 and hence the total test time is $500(32 + 1) + 32 = 16{,}532$ clock cycles. Multiple scan chains greatly reduce the time spent in shifting data; however the standard practice of configuring equal length chains does not achieve the largest possible reduction in test time. In Fig. 3, two alternative four-chain configurations are shown for the circuit of Fig. 1. The

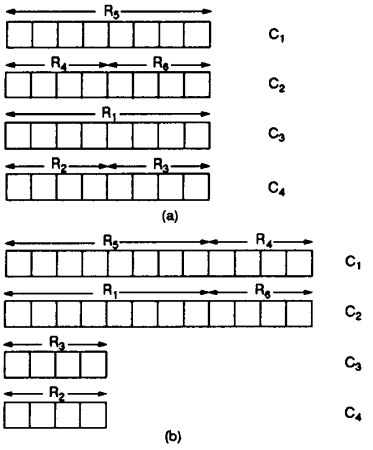


Figure 3: (a) Equal length chains (b) Optimal four-chain configuration

configuration in Fig. 3(a) has four chains of equal length. The chain cycle used in each test session will be equal to 8 and the total test time is equal to $500(8+1) + 8 = 4508$ clock cycles. Consider the four-chain configuration shown in Fig. 3(b). Test sessions $TS_1$ and $TS_2$ will employ a chain cycle of 12 since chains $C_1$ and $C_2$ are involved in testing $K_1$ and $K_2$. However in test session $TS_3$, only chains $C_3$ and $C_4$ are used and hence a chain cycle of 4 is sufficient to apply the remaining 400 patterns to $K_3$. The total test time is equal to $30(12+1) + 70(12+1) + 400(4+1) + 12 = 3312$ clock cycles. A 26% reduction in test time is obtained over the configuration in Fig. 3(a). The reduction is a direct consequence of placing flip-flops that are frequently used, i.e., flip-flops involved in testing $K_3$, in the two short chains $C_3$ and $C_4$. In fact the configuration shown in Fig. 3(b) represents an optimal four-chain configuration in terms of minimizing the test time.

Given the number of chains to be configured as an input parameter, the goal of this paper is to generate optimal configurations such as the one shown in Fig. 3(b). The test time improvements over configurations with equal length chains are clearly dependent on the number and test lengths of the kernels in the circuit as well as the number of flip-flops involved in testing each kernel. Note that the simple division of flip-flops into equal length chains does permit an arbitrary assignment and ordering of the flip-flops in the chains. This leads to improved routing area and performance optimizations. In configuring optimal chains, we constrain the assignment of the flip-flops to the chains, but place no restrictions on the order of flip-flops within a chain. In addition we will indicate ways to incorporate routing constraints in our model to generate solutions that trade off test time with area overhead.

## 4 Model of the Problem

Let us generalize the test application scheme to a circuit consisting of an arbitrary number of kernels. Without affecting the evaluation of the test time, two or more kernels with equal test lengths can be merged into a single kernel. Each scan flip-flop associated with either of the original kernels is now associated with the merged kernel. Let $n$ denote the number of kernels with different test lengths in the circuit and let $KSEQ = \{K_1, K_2, \ldots, K_n\}$ represent the set of kernels ordered in terms of strictly increasing test lengths. Let $WSEQ = \{W_1, W_2, \ldots, W_n\}$ denote the ordered set obtained by replacing every kernel in $KSEQ$

by its test length. By definition, $W_1 < W_2 < \ldots < W_n$. The application of tests using the overlapped scheme is divided into a sequence of $n$ test sessions. In each test session, a fixed number of test patterns are applied to a subset of kernels until the test set of the kernel with the smallest test length is exhausted. In the first session, $W_1$ test patterns will be applied to all kernels until the test set of $K_1$ is exhausted. Similarly, in the second test session, $(W_2 - W_1)$ test patterns will be applied to kernels $K_2, \ldots, K_n$ until the test set of $K_2$ is exhausted. In general the overlapped scheme is described by the following rule (assume $W_0 = 0$): *In test session* $TS_i$ $(1 \leq i \leq n)$, *apply* $W_i - W_{i-1}$ *test patterns to kernels* $K_i, \ldots, K_n$.

**Definition 1**
The **chain cycle** of a test session is equal to the minimum number of clock cycles required to flush all chains active in the session, i.e., the length of the longest chain used in the session.

Let $CC_i$ $(1 \leq i \leq n)$ denote the chain cycle of test session $TS_i$. In configuring multiple chains, the flip-flops in a single register are treated as being independent, i.e., they may be assigned to different chains. Let $k$ denote the number of scan chains to be configured and let $N$ denote the total number of scan flip-flops in the circuit. Let $S = \{r_1, r_2, \ldots, r_N\}$ denote the set of $N$ scan flip-flops. A weight factor $wf_m$ $(1 \leq m \leq N)$ is attached to each scan flip-flop $r_m$ by means of the following rule.

$$wf_m = max(\text{test length of kernel fed by } r_m, \text{ test length of kernel feeding } r_m).$$

For example, consider all flip-flops in register $R_2$ of Circuit 1. Each of them will have a weight factor equal to 500, i.e., max(500,100). Since 500 is the third element in $WSEQ$, each of these flip-flops will be used in all three sessions $TS_1, TS_2$ and $TS_3$.

**Definition 2**
A *multiple scan chain configuration* of $k$ chains, denoted by $\mathcal{MC}$, is defined as a $k$-way disjoint partition of the flip-flops in the set $S$ and is represented by an ordered set of scan chains $\{C_1, C_2, \ldots, C_k\}$.

In addition a chain $C_j$ $(1 \leq j \leq k)$ is characterized by the following two parameters.
a) $l(C_j)$: the length of the chain in terms of the number of flip-flops in the chain; and
b) $W(C_j)$: the weight of the flip-flop with maximum weight factor in the chain, i.e.,

$$W(C_j) = \max_{r_m \in C_j} (\text{weight factor of } r_m).$$

Without loss of generality, assume the chains in $\mathcal{MC}$ to be ordered in terms of non-increasing lengths, i.e., $l(C_1) \geq l(C_2) \geq \ldots \geq l(C_k)$. The four-chain configuration in Fig. 2 is shown again in Fig. 4(a) and the weight factors of the flip-flops and the parameters of each of the chains are displayed. Since the weight of $C_1$ is equal to 100, which is the second entry in $WSEQ$, the chain will be active in sessions $TS_1$ and $TS_2$. In the general case, if a chain has weight equal to $W_i$, it will be used in every session from $TS_1$ to $TS_i$.

Given a multiple chain configuration $\mathcal{MC}$, we can now evaluate the chain cycle $CC_i$ $(1 \leq i \leq n)$ utilized in each test session $TS_i$ of the test application scheme.

$$CC_i = max[l(C_j) | 1 \leq j \leq k \text{ and } W(C_j) \geq W_i].$$

The total test time $\mathcal{T}$ is equal to the sum of the durations of each test session. The duration of test session $TS_i$ is

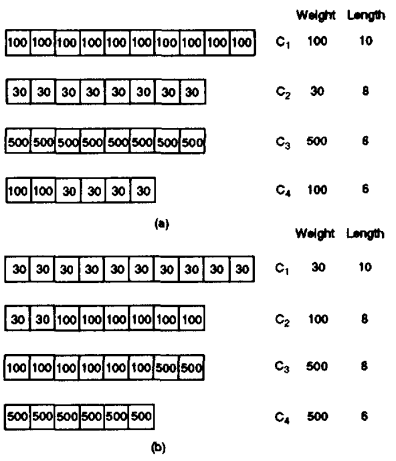


Figure 4: (a) The four-chain configuration of Figure 2. (b) Transformed configuration.

equal to $(W_i - W_{i-1})(CC_i + 1)$. Therefore the total test time is given by

$$\mathcal{T} = \sum_{i=1}^{n} (W_i - W_{i-1})(CC_i + 1) + CC_1.$$

Note that the final term $CC_1$ in the above expression represents the time to flush the final results from all chains. Since the addition of 1 in each term of the summation introduces a constant factor equal to $W_n$, the objective function to be minimized is given by

$$\mathcal{T} = \sum_{i=1}^{n} (W_i - W_{i-1})CC_i + CC_1. \quad (1)$$

## 5 Restricting the Search Space

The size of the search space involved in generating an optimal $k$-chain configuration is approximately equal to $k^N$. To reduce the search space we characterize a limited set of configurations that satisfy a certain property. Consider the four-chain configuration shown in Fig. 4(a). Without changing the length of any chain in the configuration, let us reassign the flip-flops among the chains so that every flip-flop in $C_{j+1}$ has weight greater than or equal to every flip-flop in $C_j$ $(1 \leq j \leq 3)$. Fig. 4(b) shows the transformed configuration. As a result of this transformation, the total test time decreases. To verify this, recall that the chain configuration determines the chain cycle employed in each session. Using the transformed configuration, the chain cycles in the first and third sessions are unchanged. However, in $TS_2$, only chains $C_2$, $C_3$ and $C_4$ are active in the new configuration and hence the chain cycle decreases from 10 to 8. The reassignment of flip-flops among the chains can be done for any multiple chain configuration without increasing the test time.

**Lemma 1**
For any k-chain configuration $\mathcal{MC}$, without changing the length of any chain, rearrange the flip-flops in the chains so that $W(C_1) \leq W(C_2) \leq \ldots \leq W(C_k)$. A new configuration $\mathcal{MC}_{new}$ is generated in which every flip-flop in $C_{j+1}$ has weight greater than or equal to the weight of every flip-flop in $C_j$ $(1 \leq j < k)$. The test time of the resulting configuration is less than or equal to the test time of $\mathcal{MC}$.

By only considering configurations that satisfy the conditions of the above lemma, we can significantly reduce the

search space for an optimal configuration. This intuitively satisfies our notion of assigning the frequently used flip-flops, i.e., flip-flops with large weight factors, to shorter chains.

To simplify the explanation of the final algorithm, we present an alternative method to evaluate the test time solely based on the weights and lengths of the chains in the configuration. Consider the transformed configuration shown in Fig. 4(b). Recall that $WSEQ = \{30, 100, 500\}$. In $TS_1$, chain $C_1$ will determine the chain cycle which is equal to 10. The contribution of the session to the total test time is equal to 30(10), i.e., $W(C_1)l(C_1)$. Similarly, chain $C_2$ will determine the chain cycle of $TS_2$ and the duration of the session will be $(100 - 30)8$, i.e., $(W(C_2) - W(C_1))l(C_2)$. Finally, chain $C_3$ will determine the chain cycle of $TS_3$ and the session will have duration equal to $(500 - 100)8$, i.e., $(W(C_3) - W(C_2))l(C_3)$. Without affecting the total test time we can add the expression $(W(C_4) - W(C_3))l(C_3)$ which evaluates to 0 since $W(C_4)$ is equal to $W(C_3)$. To account for the time to shift out the final results, the length of $C_1$, i.e., 10 is added to the total test time.

**Lemma 2**
Given any k-chain configuration $\mathcal{MC}$, the total test time can be evaluated from the weights and lengths of all the chains and is given by the following expression. Define $W(C_0) = 0$.

$$\mathcal{T} = \sum_{j=1}^{k} [W(C_j) - W(C_{j-1})]l(C_j) + l(C_1). \qquad (2)$$

Note that the summation in equation 2 iterates through the *chains* in the configuration as opposed to the *sessions* of the test scheme as in equation 1. All information required to calculate the total test time is embodied in the weight factors of the $N$ flip-flops used to configure the $k$ chains.

## 6 A Polynomial-Time Algorithm

Due to lack of space, we will use an example to illustrate the dynamic programming algorithm. Details of the algorithm can be found in [8]. The algorithm will be used to configure the optimal 3-chain configuration of 10 flip-flops as shown in Fig. 5(a). The weight factors of the flip-flops are shown. The test time of the configuration is $50(4) + (60 - 50)4 + (100 - 60)2 + 4 = 324$. Recall that the search space for an optimal configuration is restricted to those configurations that satisfy Lemma 1. A brute-force approach to determine an optimal configuration is to enumerate every such configuration and explicitly evaluate the test time for each of them by using equation 2.

The drawback of explicitly enumerating every possible configuration is that it results in an algorithm with time complexity exponential in the number of chains. *Dynamic programming* [9] can be used to dramatically reduce the complexity of the algorithm. It achieves this reduction in computational costs by dividing the complete problem into a set of non-overlapping subproblems. Optimal solutions to these subproblems are evaluated once and stored in a table. The optimal solution to the entire problem is then easily determined by using these stored values. The table shown in Fig. 5(b) is used to implement the dynamic programming technique. The flip-flops, sorted in order of non-decreasing weight, are shown above the table.

We employ a bottom-up approach to configuring the chains. The table is filled by first considering all single-chain subproblems involving chain $C_1$. The first row in the

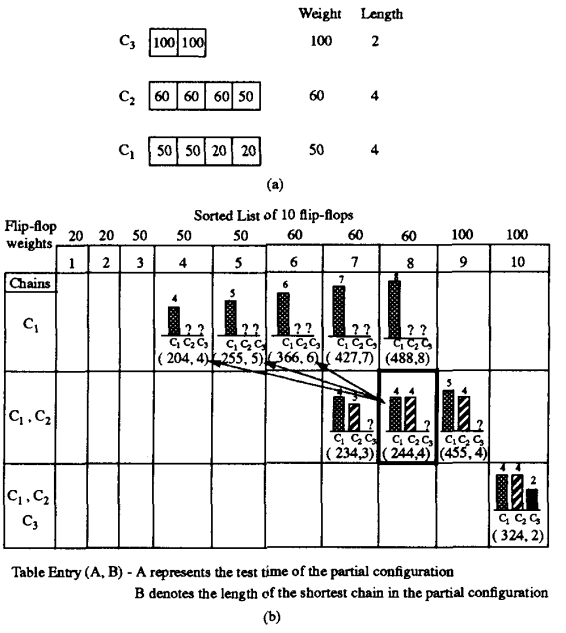


Figure 5: (a) Optimal 3-chain configuration. (b) Dynamic programming table.

table is used to store optimal solutions to these subproblems. Recall that $C_1$ will always contain the lowest-weight flip-flops. Since it is the longest chain in any configuration, its length can vary from $\lceil \frac{10}{3} \rceil$ to 8 (at least one flip-flop must exist in each of the remaining two chains). For each of these lengths, the contribution of $C_1$ is evaluated and stored in the appropriate entry. For example, if $l(C_1) = 5$, then the weight of $C_1$ is equal to 50 and its contribution to the total test time is $50(5) + 5 = 255$. This information is stored in the first row as a 2-tuple (255,5) where the first value denotes the test time contribution of $C_1$ and the second denotes the length of $C_1$.

Each entry in the second row of the table corresponds to subproblems involving the chains $C_1$ and $C_2$. The number of flip-flops used in configuring two chains must be at least 7 (if not chain $C_3$ would be of a greater length than either $C_1$ or $C_2$) and at most 9 (since $C_3$ must contain at least one flip-flop). Let us study how the table entry enclosed in a bold rectangle is derived; for this entry, we need to assign the 8 lowest-weight flip-flops to the two chains. Since 8 flip-flops are being assigned to chains $C_1$ and $C_2$, the length of chain $C_3$ must be equal to 2. Hence the length of chain $C_2$ can assume values of 2, 3 and 4 and the corresponding lengths for chain $C_1$ are 6, 5 and 4. We consider the three cases separately to determine the optimal solution to the subproblem. In each case, we add together the contribution of chain $C_2$ and the appropriate value in the first row of the table. For example, if $l(C_2)$ is equal to 4, the test time is evaluated as $(60 - 50)4 + 204 = 244$. Note that the contribution of $C_1$, i.e., 204 is directly obtained from the first row of the table. Among the three cases, the solution with least test time corresponds to the 2-tuple (244,4) where 244 denotes the optimal test time and 4 denotes the length of $C_2$ in the optimal configuration. The arrows from the highlighted entry point to the single-chain subproblems used in evaluating the optimal solution.

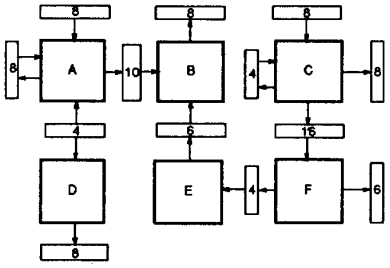


Figure 6: Circuit 2

| | CASE 1 | | CASE 2 | | CASE 3 | |
|---|---|---|---|---|---|---|
| | No. of tests | No. of ffs | No. of tests | No. of ffs | No. of tests | No. of ffs |
| A | 40 | 16 | 60 | 16 | 200 | 30 |
| B | 80 | 18 | 500 | 24 | 80 | 14 |
| C | 50 | 20 | 80 | 36 | 500 | 36 |
| D | 200 | 12 | 200 | 12 | 50 | 8 |
| E | 500 | 10 | 50 | 4 | 40 | 0 |
| F | 60 | 22 | 40 | 6 | 60 | 10 |

Table 2: Three Variations of Circuit 2

Finally, the third row in the table will possess a single entry for distributing all 10 flip-flops among the three chains. For each of the three possible lengths of $C_3$, i.e., 1, 2 and 3, the contribution of $C_3$ is added to the appropriate value stored in the 2-tuples in the second row. For example, if $l(C_3)$ is equal to 2, the test time of the configuration is evaluated as $(100-60)2+244 = 324$. The optimal test time corresponds to the minimum of the three possible cases and the optimal configuration is derived from the table. By analyzing the algorithm, it can be shown that given $N$ flip-flops to be optimally configured into $k$ chains, the worst case time complexity is of order $O(kN^2)$.

## 7 Implementation Results

To validate the concepts underlying our methodology, we utilize the circuit shown in Fig. 6. The circuit has six combinational kernels and thirteen registers with each register labeled with the number of flip-flops it contains. The primary inputs/outputs at each kernel are not shown.

A full scan methodology is used in testing the circuit. To accurately compare the test times of optimal configurations against equal length chain configurations, we use three variations of this basic circuit by permuting six test lengths in different ways among the kernels. For each of the three cases, Table 2 shows the test length and number of flip-flops associated with each kernel. Note that if a register is shared by two kernels, all flip-flops in the register are associated with the kernel having the greater test length (see definition of $wf_m$).

In Table 3, while varying the number of scan chains from 1 to 10, we compare the test times (in terms of number of clock cycles) for an equal length chain configuration and an optimal configuration for each of the three cases. The percentage reduction in all three cases represents the savings in test time of the optimal configuration as compared to equal length chains. For each case, if the kernels are listed in decreasing order of test lengths, the corresponding list of the number of flip-flops associated with each kernel is (10,12,18,22,20,16) for Case 1, (24,12,36,16,4,6) for Case 2 and (36,30,14,10,8,0) for Case 3. An interesting observation is that the test time difference between an optimal configuration and a configuration with equal length chains decreases as the number of flip-flops associated with kernels of large test lengths increases. In Case 1, the total number of flip-flops associated with the kernels of test length 200 and 500 is 20% of the total number of flip-flops and the average test time reduction is 34%. However, for Case 2 and Case 3, when the flip-flop ratio increases to 36% and 67% respectively, the average test time reduction drops to 20% and 5% respectively.

| No. of chains | Test time (eq. length) | Case 1 % red. (optimal) | Case 2 % red. (optimal) | Case 3 % red. (optimal) |
|---|---|---|---|---|
| 1 | 49598 | 0 | 0 | 0 |
| 2 | 25049 | 36.54 | 17.59 | 5.14 |
| 3 | 17033 | 34.86 | 23.20 | 0.58 |
| 4 | 13025 | 38.44 | 18.91 | 7.16 |
| 5 | 10520 | 38.04 | 23.47 | 4.45 |
| 6 | 9017 | 38.94 | 24.69 | 8.15 |
| 7 | 7514 | 36.31 | 20.40 | 4.23 |
| 8 | 7013 | 39.27 | 24.98 | 10.22 |
| 9 | 6011 | 36.13 | 21.46 | 5.29 |
| 10 | 5510 | 35.77 | 21.67 | 5.01 |

Table 3: Comparison of test times with multiple chains

## 8 Conclusion

We have presented a comprehensive methodology to efficiently configure multiple scan chains. The main idea is to assign flip-flops involved in testing kernels with larger test lengths to shorter chains. A polynomial-time algorithm has been derived to obtain a configuration that minimizes the test application time. Results clearly show the advantages of the design strategy over configuring equal length chains. Note that we retain the flexibility to arbitrarily order the flip-flops within a scan chain to minimize routing overheads. In addition, the model of our problem can be modified to reflect routing and performance constraints. For example, an upper bound can be imposed on the maximum wire length between any two adjacent flip-flops in a scan chain. This additional constraint can be used to further restrict the search space of an optimal configuration with respect to test time.

## References

[1] K.-T. Cheng and V.D. Agrawal. A partial scan method for sequential circuits with feedback. *IEEE Trans. on Computers*, 39(4):544–548, April 1990.

[2] R. Gupta, R. Gupta, and M. A. Breuer. The BALLAST methodology for structured partial scan design. *IEEE Trans. on Computers*, 39(4):538–543, April 1990.

[3] P.P. Fasang et al. Automated design for testability of semicustom integrated circuits. In *Proc., Int'l Test Conf.*, pages 558–564, November 1985.

[4] S. Bhawmik and P. Palchaudhuri. DFT Expert: Designing testable VLSI circuits. *IEEE Design & Test*, pages 8–19, October 1989.

[5] S.P.Morley and R.A.Marlett. Selectable length partial scan: a method to reduce vector length. In *Proc., Int'l Test Conf.*, pages 385–392, November 1991.

[6] R.Gupta. *Advanced Serial Scan Design for Testability.* Ph.D. thesis, Univ. of Southern California, Dept. of Electrical Engg., 1991. CEng. Technical Report 91-10.

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

[8] S. Narayanan, R. Gupta, and M.A. Breuer. Optimal configuring of multiple scan chains. Accepted for publication in *IEEE Trans. on Computers* .

[9] F. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Massachusetts, 1990.