# Synthesis-for-Scan and Scan Chain Ordering

Robert B. Norwood and Edward J. McCluskey

*Center for Reliable Computing*
*Stanford University*

## Abstract

*Designing a testable circuit is often a two step process. First, the circuit is designed to conform to the functional specifications. Then, the testability aspects are added. By taking the test strategy into account during the synthesis of the circuit, the overhead due to the test features can be reduced. We present a synthesis-for-scan procedure, called beneficial scan, that orders the scan chain(s) during logic synthesis to minimize the area and performance overhead due to the scan-path by sharing the functional and the test logic. The results show that circuits synthesized with beneficially-ordered scan chains consistently have smaller area and are easier to route than circuits with traditional MUXed flip-flop scan-paths.*

## 1 Introduction

Scan chains are widely used to improve the testability of sequential designs. A scan chain provides direct access to the internal nodes, greatly improving the controllability and observability of the circuit. It is then no longer necessary to generate test patterns for a sequential circuit, a difficult and time consuming task, since the scan chain effectively turns the sequential circuit into a combinational circuit during testing.

There are a variety of scan techniques in use, each with its positive and negative aspects [1]. Scan designs, in general, have some costs associated with them. These costs are:

1. Additional circuitry added to make bistables scannable increasing the circuit area.
2. Possible performance penalty due to the increased propagation delay in the scan chain bistables.
3. Additional pins necessary for the test signals.
4. Additional interconnect that can add to area and reduce performance.
5. Increased testing time due to serialization of test patterns.

Various techniques have been proposed to reduce the costs associated with scan designs. Careful ordering of the scan chain elements can reduce the interconnect or testing time [2] [3] [4] [5]. By giving up some of the controllability and observability of a fully scanned design, partial-scan designs [6] [7] [8] [9] [10] attempt to reduce the overhead by making only a subset of the system bistables scannable. Cost-free scan [11] attempts to chose primary input vectors to establish scan paths through the combinational logic. These techniques generally approach the problem of reducing the test overhead after the circuit has been designed.

Other work [12] [13] [14] [15] considers scan chain insertion during the synthesis of the circuit. Cox's [16] [17] work on embedding scan chains through the use of test synthesis constraints is similar to the approach presented in this paper, though the actual methods used differ.

We present a technique, called beneficial scan, that combines circuit synthesis and scan chain insertion into one step. Knowledge of the circuit functions may be used to order the scan chain elements in such a fashion that the functional logic and the test logic may be shared during synthesis, reducing the cost of the scan chain. The scan chain is assumed to be implemented with MUXed flip-flops, to allow the test logic to be shared with the functional logic. This work is based on work done at the Center for Reliable Computing [18].

## 2 Scan element classifications

Since each flip-flop input equation is a function of flip-flop outputs and primary inputs, each flip-flop in a synchronous circuit has some relationship with every other flip-flop in the circuit. This relationship may be trivial, as in the case when the input to a flip-flop is not a function of another flip-flop, or the relationship may be more complex. The relationship could be as simple as whether or not the input function of one flip-flop includes the output of another flip-flop, or the relationship could be based on more complex characteristics, e.g., the input function is positive (or negative) in a variable.

Some of these relationships allow some, or all, of the test logic to be shared with the functional logic. These are *beneficial relationships*. Other relationships do not allow the logic to be shared, and a MUX must be inserted to make a flip-flop scannable. These are *non-beneficial relationships.*

For the purposes of beneficial scan, the relationship between flip-flops is based on Shannon's expansion theorem. The Shannon expansion transforms a function, $f(x_1, x_2, \ldots, x_n)$, based on residues. The $x_i$-residue, also called the positive-phase cofactor of $f$ with respect to $x_i$, is defined as

Table 1. Flip-flop classifications

***B*** indicates a beneficial relationship.

Constants are '0' or '1'.

$Q_i^*$ indicates that the flip-flop output may be inverted.

| Class | $Q_i$-residue | $Q'_i$-residue | Equation Form | New Scan Function | Overhead |
|---|---|---|---|---|---|
| Case 0 | h | h | $D_j = h$ | $D_j = T' h + T Q_i$ | MUX |
| ***B*** Case 1 | constant | constant | $D_j = Q_i^*$ | $D_j = Q_i^*$ | none |
| ***B*** Case 2 | constant | not constant | $D_j = Q_i^* + f_{Q'_i}$ | $D_j = Q_i^* + T' f_{Q'_i}$ | AND |
| | not constant | constant | $D_j = Q_i^* f_{Q_i}$ | $D_j = Q_i^* (T + f_{Q_i})$ | OR |
| ***B*** Case 3 | $f'_{Q'_i}$ | not constant | $D_j = Q_i^* \oplus f_{Q'_i}$ | $D_j = Q_i^* \oplus T' f_{Q'_i}$ | AND(OR) |
| Case 4 | not constant | not constant | $D_j = Q_i f_{Q_i} + Q'_i f_{Q'_i}$ | $D_j = T' h + T Q_i$ | MUX |
| ***B*** Case 4S | not constant (s + a) | not constant (s' a) | $D_j = s Q_i^* + s' a$ | $D_j = (s + T) Q_i^* + (s + T)' a$ | OR |

$$f_{x_i}(x_1, x_2,\ldots, x_n) = f(x_1, x_2,\ldots, x_i = 1,\ldots, x_n)$$

and the $x'_i$-residue, or negative-phase cofactor of $f$ with respect to $x_i$, as

$$f_{x'_i}(x_1, x_2,\ldots, x_n) = f(x_1, x_2,\ldots, x_i = 0,\ldots, x_n)$$

The Shannon expansion is defined as

$$f(x_1, x_2,\ldots, x_n) = x_i f_{x_i} + x'_i f_{x'_i}$$

Based on the Shannon expansion, each flip-flop *j* may be classified with respect to every other flip-flop *i*. The classifications are described in Table 1, and a more detailed example is given below. Some of these classifications are beneficial relationships and allow the test logic to be shared with the functional logic. These beneficial relationships are marked with an "***B***" in Table 1. The other classifications are non-beneficial and no logic may be shared. While other relationships exist that may be used to classify the flip-flops, these are the ones used in this technique.

Notation: $D_j$ is the input of flip-flop *j*. $Q_i$ is the output of flip-flop *i*. *T* is the scan test select signal—1 for scan mode, 0 for normal system operation. *h* is the function for normal system operation, i.e., the input equation before scan is inserted. *s* and *a* are arbitrary functions.

### 2.1 Case 2 example

Flip-flop *i* and flip-flop *j* have the following relationship, $D_j = Q_i + f_{Q'_i}$. This is a case 2 beneficial relationship since the $Q_i$-residue is a constant one and the $Q'_i$-residue is not a constant. If flip-flop follows flip-flop *i* in the scan chain, then only a single AND gate needs to be added to make flip-flop *j* scannable. The new function for $D_j$ would be $D_j = Q_i + T' f_{Q'_i}$. This modification is shown in Figure 1.

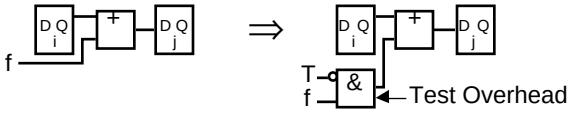


Figure 1. Case 2 flip-flop

## 3 Scan chain ordering

### 3.1 Single scan chains

Once the flip-flops have been classified, a weighted, directed graph, called a *relationship graph*, is constructed. The nodes of the graph represent the flip-flops and the primary inputs. The weighted arcs represent the relationship between the flip-flops and inputs. The head of the arc is flip-flop *i*, or input *i*, and the tail is flip-flop *j* (i.e., the arcs follow the direction of the shift during scan operation). Each flip-flop has a weighted arc to and from every other flip-flop, as well as one from every primary input. Therefore, primary inputs are found only at the head of arcs. An example relationship graph is shown in Figure 2.

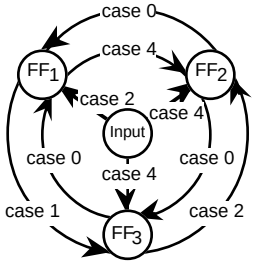


Figure 2. Relationship graph

The arcs are weighted to show the cost of adding test logic between the two flip-flops. A case 1 flip-flop needs no additional logic to perform the scan function. Case 2, case 3 and case 4s flip-flops require the addition of a single gate to add scan capability. Case 0 and case 4 flip-flops must have an entire MUX added. Based on these overheads, case 0 and case 4 arcs have a large weight, case 1 arcs have a very small weight, and the other cases are weighted in between.

The goal is to find a minimal weight path through the graph that starts at an input node and covers all the flip-flop nodes. For a circuit with *p* primary inputs and *n* flip-flops, there are $p \bullet n!$ possible scan chain orderings since there are $n!$ ways to order the flip-flops and *p* different inputs to use to scan in data. An exact solution is computationally expensive, so an heuristic is required.

Certain characteristics of the relationship graph allow heuristics to perform very well.

- Any ordering of the flip-flops is a valid order for the scan chain, so there are no dead-ends.
- The cost of choosing any particular flip-flop has little dependence on previous decisions, i.e., a poorly chosen flip-flop will only affect one other decision.

A greedy approach works well.

1. Choose the input with the lowest weight edge to act as the scan data input.
2. Add to the chain the flip-flop with the lowest weight edge remaining.
3. Repeat step 2.

This basic greedy algorithm can be modified to make better choices at each step—such as giving preference to flip-flops with only one beneficial relationship over flip-flops with multiple beneficial relationships in order to leave more options open for later, or making sure that a chosen flip-flop is not the only beneficial case for another flip-flop. These modifications, along with others, help guide the algorithm to a better result.

The algorithm tries to reduce the number of additional test pins needed by using one of the existing primary inputs as the scan-data-in pin. If there is a need to have an explicit scan-data-in pin, a new primary input may be added to the relationship graph. This new input will have a case 0 arc to every flip-flop node.

### 3.2 Multiple scan chains

The approach just described will give an ordering for a single scan chain. The possibility of finding orderings for multiple scan chains also exists. Multiple scan chain orderings take advantage of an additional characteristic of the relationship graphs:

- Beneficial relationships with a primary input are more frequent than beneficial relationships with other flip-flops.

Even if a flip-flop has no beneficial relationships with other flip-flops, it is likely that it will have a beneficial relationship with one or more inputs. By allowing multiple scan chains, a lower cost scan chain ordering can be obtained since more beneficial relationships with primary inputs will be included.

The algorithm is the same as that for single scan chains, with a slight modification to step 2:

1. Choose the input with the lowest weight edge to act as the scan data input
2. Add to an existing chain the flip-flop with the lowest weight edge remaining
   *or*
2. Add another scan chain if cost is less than adding flip-flop to existing chain(s)
3. Repeat step 2

As with the single scan chains, the algorithm tries to use existing primary inputs as the scan-data-in pins. Any additional logic sharing achieved by using multiple scan chains comes from this sharing of the inputs.

The algorithm only adds another scan chain if there is some additional logic sharing gain. A specific number of scan chains may be forced with some degradation in the results. The algorithm also tries to balance the chains without degrading the results. If there is more than one equally attractive choice during step 2, the flip-flop is chosen to balance the existing scan chains. Completely balanced chains may be forced, but the resulting order will not be optimal.

Table 2. Benchmark characteristics

| Circuit | Number Primary Inputs | Number Flip-Flops |
|---|---|---|
| dsip | 230 | 224 |
| mult32b | 34 | 62 |
| s1196 | 16 | 18 |
| s344 | 11 | 15 |
| s400 | 5 | 21 |
| s526 | 5 | 21 |
| s641 | 37 | 17 |
| sbc | 42 | 27 |

## 4 Implementation and results

### 4.1 Beneficial scan

This algorithm to order flip-flops for single, or multiple, scan chains has been implemented in SIS [19]. Existing SIS facilities are used to synthesize and technology map the resulting circuit. The circuit is mapped to the Alliance library, and the Alliance CAD system is used to do placement and routing and static timing analysis [20].

Table 3. Number of beneficial relationships in final scan chain(s)

| Circuit | Number of Scan Chains | Frequency of Various Relationships in Final Scan Chain(s) | | | | | |
|---|---|---|---|---|---|---|---|
| | | Beneficial Cases | | | | Non-Beneficial Cases | |
| | | 1 | 2 | 3 | 4S | 4 | 0 |
| dsip | 1 | 0 | 0 | 0 | 223 | 0 | 1 |
| dsip | 2 | 0 | 0 | 0 | 224 | 0 | 0 |
| mult32b | 1 | 0 | 1 | 30 | 30 | 1 | 0 |
| mult32b | 2 | 0 | 2 | 30 | 30 | 0 | 0 |
| s1196 | 1 | 0 | 1 | 0 | 5 | 0 | 12 |
| s1196 | 12 | 0 | 7 | 2 | 6 | 0 | 3 |
| s344 | 1 | 0 | 1 | 0 | 6 | 4 | 4 |
| s344 | 9 | 0 | 1 | 0 | 11 | 3 | 0 |
| s400 | 1 | 0 | 2 | 0 | 7 | 2 | 10 |
| s526 | 1 | 0 | 2 | 0 | 7 | 6 | 6 |
| s641 | 1 | 0 | 5 | 0 | 7 | 1 | 4 |
| s641 | 6 | 0 | 6 | 0 | 11 | 0 | 0 |
| sbc | 1 | 0 | 5 | 0 | 13 | 2 | 7 |
| sbc | 16 | 0 | 9 | 0 | 18 | 0 | 0 |

Table 4. Area of benchmark circuits after routing

| | MUXed Flip-Flops | Beneficial Scan | | Beneficial Scan with Free-Scan | |
|---|---|---|---|---|---|
| | Single Scan Chain | Single Scan Chain | Multiple Scan Chains | Single Scan Chain | Multiple Scan Chains |
| Circuit | %Overhead | %Overhead | %Overhead | %Overhead | %Overhead |
| dsip | 30.3 | 0.6 | 1.5 | -2.9 | -3.2 |
| mult32b | 29.0 | 18.2 | 22.9 | 23.5 | 23.5 |
| s1196 | 3.3 | -15.7 | -22.2 | -4.1 | -4.1 |
| s344 | 26.0 | 14.3 | 10.2 | 14.3 | 10.1 |
| s400 | 38.8 | 25.1 | 25.1 | 25.1 | 25.1 |
| s526 | 49.9 | 24.9 | 24.9 | 24.9 | 24.9 |
| s641 | 34.4 | 15.9 | 7.9 | 9.8 | 9.5 |
| sbc | 4.7 | -6.4 | -32.4 | -26.1 | -18.6 |

Table 3 shows the results of the scan chain ordering. Both single and multiple scan chains are shown unless the two results are the same, i.e., there is no benefit in having multiple scan chains for that particular circuit. For each benchmark circuit the number of scan chains is shown along with the breakdown of the flip-flop classifications used in the final scan chain.

Table 4 shows the area results of the circuits after placement and routing. Data is shown for five different scan-path implementations: 1) the circuit synthesized and a single scan chain, not ordered based on beneficial relationships, implemented entirely with MUXed flip-flops then added, 2) the circuit with a single beneficially-ordered scan chain, 3) the circuit with multiple beneficially-ordered scan chains, 4) the circuit with a single beneficially-ordered scan chain including cost-free scan and 5) the circuit with multiple beneficially-ordered scan chains including cost-free scan. The last two variations including cost-free scan are discussed in section 4.2. The overhead is calculated as

$$\%\text{overhead} = \left( \frac{\text{area of scan} - \text{area of nonscan}}{\text{area of nonscan}} \right) \times 100$$

The beneficially-ordered scan chain circuits achieve a significant area reduction over the MUXed flip-flop scan chains. The average area overhead for beneficially-ordered scan chains, after placement and routing, is about 12%, compared to about 27% for more traditional scan chains. The area savings before routing (table 7 in the appendix) is not quite as great since much of the area savings result from the elimination of interconnect through the logic sharing and the reduction in the length of the scan signal interconnect. Traditional scan chains are often ordered based on preliminary layouts to reduce the length of the scan signal interconnect. Beneficially-ordered scan chains have short interconnect as a result of the ordering; i.e., flip-flops that should be placed near each other in layout to produce short functional interconnect also should be placed near each other to produce short scan interconnect. All of this results in lower interconnect overhead for the beneficially-ordered scan chains.

Four of the routed circuits, two single scan chain circuit and two multiple scan chain circuits, actually end up being smaller than the non-scanned circuits. This result is a phenomenon of the place and route tool. The non-scanned circuit could achieve the same (or better) area as the beneficially-ordered circuit by using the same placement and routing layout, but with the same level of effort the place and route tool gives better results for the beneficially-ordered circuit than for the non-scanned circuit. The beneficially-ordered circuit is easier to place and route.

The data is not shown, but the average worst-path delay overhead for the beneficially-ordered scan chains is 16%, and for the MUXed flip-flop scan chains the average delay overhead is 19%. Beneficial scan can reduce the area and delay overhead due to scan.

### 4.2 Beneficial scan with cost-free scan

Cost-free scan [11] is a recent technique that allows the reuse of combinational logic for scan by exploiting the controllability of primary inputs. The circuit is analyzed and a primary input vector, called the enabling vector, is selected to enable the maximum number of free-scan flip-flops. A free-scan flip-flop does not require any additional logic to implement the scan operation. Cost-free scan is very similar to beneficial scan, but it uses different relationships in the analysis of the circuit.

Free-scan flip-flops have been added to our beneficial scan technique by adding a new class of beneficial relationship to the analysis. The beneficial ordering is now able to use free-scan flip-flops in addition to the cases presented in section 2. Tables 4 and 5 show the results of beneficially-ordering the scan chains including free-scan flip-flops.

*s344*, *s400* and *s526* have no free-scan flip-flops, so the results are the same as without free-scan flip-flops. Four of the remaining circuits (*dsip, mult32b, s1196* and *sbc*) use the maximum possible number of free-scan flip-flops in the final scan chain, while *s641* uses only three of the possible six free-scan flip-flops. The use of the free-scan flip-flops typically reduces the area overhead for single scan chains.

Table 5. Number of beneficial relationships in final scan chain(s) including free-scan

| Circuit | Number of Scan Chains | Frequency of Various Relationships in Final Scan Chain(s) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Beneficial Cases | | | | Non-Beneficial Cases | | Free Scan Flip-Flop |
| | | 1 | 2 | 3 | 4S | 4 | 0 | |
| dsip | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 222 |
| dsip | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 222 |
| mult32b | 1 | 0 | 0 | 29 | 30 | 0 | 2 | 1 |
| s1196 | 1 | 0 | 1 | 0 | 3 | 0 | 11 | 3 |
| s344 | 1 | 0 | 1 | 0 | 6 | 4 | 4 | 0 |
| s344 | 9 | 0 | 1 | 0 | 11 | 3 | 0 | 0 |
| s400 | 1 | 0 | 2 | 0 | 7 | 2 | 10 | 0 |
| s526 | 1 | 0 | 2 | 0 | 7 | 6 | 6 | 0 |
| s641 | 1 | 0 | 2 | 0 | 7 | 1 | 4 | 3 |
| s641 | 6 | 0 | 2 | 0 | 11 | 0 | 1 | 3 |
| sbc | 1 | 0 | 5 | 0 | 11 | 2 | 6 | 3 |
| sbc | 11 | 0 | 5 | 0 | 18 | 0 | 1 | 3 |

For multiple scan chains, the use of the free-scan flip-flops can reduce the effectiveness of the beneficial ordering. Multiple scan chains use additional primary inputs as test inputs to reduce the overhead, but in order to use free-scan flip-flops, certain primary inputs belonging to the enabling vector may not be used as scan inputs. This restriction reduces the number of beneficial cases available for multiple scan chains. In general, the addition of free-scan to beneficial scan can give better results than either technique by itself.

Using free-scan flip-flops has the added disadvantage of reducing the routing benefit observed with beneficial scan. Beneficially-ordered scan chains are easier to route, as discussed in section 4.1, but the addition of the free-scan flip-flops can reduce the layout tools effectiveness, resulting in slightly larger areas for some circuits.

Comparison of beneficial scan with other techniques is complicated by the fact that the most of the savings from beneficially-ordering scan chains comes after the circuit has been placed and routed. Most other techniques do not include this data. Comparisons based on the literal counts are possible, but they can be misleading. Table 6 in the appendix shows the literal counts for the benchmark circuits. The literal counts for the beneficially-ordered scan chain circuits do tend to be smaller than for the MUXed flip-flop scan chain circuits, but the literal count can increase for with beneficial scan since Shannon's expansion theorem is used to factor the logic equations. This expansion does not always reduce the literal count, but it enables the logic to be shared for the beneficial relationships, and can result in smaller circuits overall once the circuits are placed and routed.

## 5 Conclusions

This paper has presented a synthesis-for-scan technique, called beneficial scan, that orders the scan chain so as to maximize the sharing of the functional and the test logic and thereby minimize the area and delay overhead due to the scan chain. The results show that circuits synthesized with beneficially-ordered scan chains consistently have smaller area, smaller worst-path delay, and are easier to route than circuits with traditional MUXed flip-flop scan-paths. The average area overhead for beneficially-ordered scan chains, after placement and routing, is about 12%, compared to about 27% for more traditional scan chains. This area reduction results from two factors, the sharing of the functional and test logic and the reduction in the interconnect.

The class of beneficial relationships can be expanded to included free-scan flip-flops allowing more opportunities to share the functional and the test logic. Other relationships may also be found that can increase the resulting number of beneficial relationships, further reducing the test overhead.

## Acknowledgments

Many thanks to Khader Abdel-Hafez, Jonathan T.Y. Chang, Mike Lee and Irith Pomeranz for their comments and suggestions.

This research was supported in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization and administered through the Office of Naval Research under Contract No. N00014-92-J-1782, in part by the Advanced Research Projects Agency under Contract No. DABT63-94-C-0045, and in part by the National Science Foundation under Grant No. MIP-9107760.

## References

[1] McCluskey, E.J., *Logic Design Principles*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[2] Gupta, R., and M.A. Breuer, "Ordering Storage Elements in a Single Scan Chain," *Proc. IEEE Intl. Conf. Computer-Aided Design*, Santa Clara, CA, pp. 408-411, Nov. 11-14, 1991.

[3] Narayanan, S., C. Njinda, and M.A. Breuer, "Optimal Sequencing of Scan Registers," *Proc. Intl. Test Conf.*, Baltimore, MD, pp. 293-302, Sept. 20-24, 1992.

[4] Pradhan, D.K., J. Saxena, "A Design for Testability Scheme to Reduce Test Application Time in Full Scan," *Proc. IEEE VLSI Test Symp.*, Atlantic City, NJ, pp. 55-60, April 7-9, 1992.

[5] Narayanan, S., R. Gupta, and M.A. Breuer, "Optimal Configuring of Multiple Scan Chains," *IEEE Trans. Comp.*, vol. 42, no. 9, pp. 1121-1131, Sept., 1993.

[6] Agrawal, V., K. Cheng, and D. Johnson, "Designing Circuits with Partial Scan," *IEEE Design and Test* , vol. 5, no. 2, pp. 8-15, April, 1988.

[7] Chickermane, V., and J.H. Patel, "An Optimization Based Approach to the Partial Scan Design Problem," *Proc. Intl. Test Conf.*, Washington, DC, pp. 377-386, Sept. 10-14, 1990.

[8] Gupta, R., and M.A. Breuer, "The BALLAST Methodology for Structured Partial Scan Design," *IEEE Trans. Comp.*, vol. 39, no. 4, pp. 538-544, April, 1990.
[9] Lee, D.H., and S.M. Reddy, "On Determining Scan Flip-Flops in Partial-Scan Designs," *Proc. IEEE Intl. Conf. Computer-Aided Design*, Santa Clara, CA, pp. 322-325, Nov. 11-15, 1990.
[10] Chakradhar, S.T., A. Balakrishnan, and V. Agrawal, "An Exact Algorithm for Selecting Partial Scan Flip-Flops," *Proc. Design Automation Conf.*, San Diego, CA, pp. 81-86, June 6-10, 1994.
[11] Lin, C., et. al., "Cost-Free Scan: A Low-Overhead Scan Path Design Methodology," *Proc. IEEE Intl. Conf. Computer-Aided Design*, San Jose, CA, pp. 528-533, Nov. 5-9, 1995.
[12] Reddy, S.M., and R. Dandapani, "Scan Design Using Standard Flip-Flops," *IEEE Design and Test*, pp. 52-54, Feb., 1987.
[13] Vinnakota, B., and N.K. Jha, "Synthesis of Sequential Circuits for Parallel Scan," *Proc. European Conf. Design Automation*, Brussels, Belgium, pp. 366-370, March 16-19, 1992.
[14] Bhatia, S., and N.K. Jha, "Synthesis of Sequential Circuits for Easy Testability Through Performance-Oriented Parallel Partial Scan," *Intl. Conf. VLSI Design*, India, pp. 151-154, Jan., 1993.
[15] Kanjilal, S., S.T. Chakradhar, and V.D. Agrawal, "Synthesis Approach to Design for Testability," *Proc. Intl. Test Conf.*, Baltimore, MD, pp. 754-762, Oct. 17-21, 1993.
[16] Cox, H., "On Synthesizing Circuits with Implicit Testability Constraints," *Proc. Intl. Test Conf.*, Washington, DC, pp. 989-998, 0ct. 2-6, 1994.
[17] Cox, H., "Synthesizing Circuits with Implicit Testability Constraints," *IEEE Design and Test*, vol. 12, no. 2, pp. 16-23, Summer 1995.
[18] Avra, L.J., "Synthesizing for Scan Dependence in Built-In Self-Testable Designs," *Proc. Intl. Test Conf.*, Baltimore, MD, pp. 734-743, Oct. 17-21, 1993.
[19] Sentovich, E.M., et. al., "Sequential Circuit Design Using Synthesis and Optimization," *Intl. Conf. Comp. Design*, Cambridge, MA, pp. 328-333, Oct. 11-14, 1992.
[20] Hajjar, A., et. al., "TAS: An Accurate Timing Analyzer for CMOS VLSI," *Proc. European Conf. Design Automation*, Amsterdam, Netherlands, pp. 261-265, Feb. 25-28, 1991.

## Appendix

Table 6. Literal counts for benchmark circuits

| | | MUXed Flip-Flops | Beneficial Scan | | Beneficial Scan with Free-Scan | |
|---|---|---|---|---|---|---|
| | No Scan Chain | Single Scan Chain | Single Scan Chain | Multiple Scan Chains | Single Scan Chain | Multiple Scan Chains |
| Circuit | Literals | Literals | Literals | Literals | Literals | Literals |
| dsip | 2755 | 3700 | 2817 | 2828 | 2810 | 2808 |
| mult32b | 459 | 707 | 584 | 525 | 586 | 586 |
| s1196 | 757 | 829 | 809 | 806 | 801 | 801 |
| s344 | 161 | 221 | 206 | 209 | 206 | 209 |
| s400 | 169 | 253 | 232 | 232 | 232 | 232 |
| s526 | 192 | 276 | 266 | 266 | 266 | 266 |
| s641 | 377 | 475 | 467 | 460 | 440 | 461 |
| sbc | 810 | 908 | 898 | 881 | 897 | 888 |

Table 7. Area of benchmark circuits before routing

| | MUXed Flip-Flops | Beneficial Scan | | Beneficial Scan with Free-Scan | |
|---|---|---|---|---|---|
| | Single Scan Chain | Single Scan Chain | Multiple Scan Chains | Single Scan Chain | Multiple Scan Chains |
| Circuit | %Overhead | %Overhead | %Overhead | %Overhead | %Overhead |
| dsip | 22.1 | 3.5 | 2.0 | -1.0 | -1.0 |
| mult32b | 31.7 | 29.2 | 31.9 | 32.1 | 32.1 |
| s1196 | 6.0 | 0.4 | 2.6 | 1.7 | 1.7 |
| s344 | 23.2 | 23.3 | 13.1 | 23.3 | 13.1 |
| s400 | 23.2 | 21.2 | 21.2 | 21.2 | 21.2 |
| s526 | 24.5 | 16.2 | 16.2 | 16.2 | 16.2 |
| s641 | 10.9 | 11.4 | 15.0 | 13.4 | 5.3 |
| sbc | 8.4 | 2.7 | 4.0 | 8.3 | 5.9 |