

## Parallel Self-Test and the Synthesis of Control Units

Bernhard Eschermann, Hans-Joachim Wunderlich

Universität Karlsruhe, Institut für Rechnerentwurf und Fehlertoleranz  
Zirkel 2, 7500 Karlsruhe, F. R. Germany

**Abstract** – Most self-test techniques are implemented with so-called multifunctional test registers at any specific time either used for pattern generation or for response analysis. In a parallel self-test, however, test registers are used for pattern generation and response analysis simultaneously. In this paper a novel circuit structure for controllers with parallel self-test is presented, which does not result in a loss of fault coverage. By using a dedicated synthesis procedure, which considers the self-test hardware while generating the circuit structure instead of adding it after the design is completed ("synthesis for testability"), the self-test overhead can be kept low. The structure also facilitates realistic dynamic tests. As an example to illustrate the approach, the IEEE boundary scan controller is used.

### 1 Introduction

Digital circuits are usually partitioned into data path and control unit portions. A control unit's behavior is typically modelled by a finite state machine (FSM) description, its structure by an interconnection of combinational logic and storage elements (see Fig. 1).

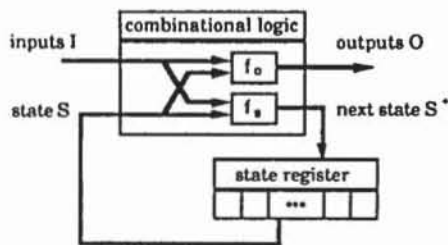


Fig. 1: General structure of control units.

One characteristic of controllers is a strong mutual dependence of state variables, which leads to a high sequential depth of these circuits. It makes testing harder compared to data paths and becomes particularly critical for self-testable designs. If the state register is replaced by a single multifunctional

self-test register [BeMa 84], e. g. a BILBO [KöMZ 79], the direct feedback lines imply that the signatures of the test responses have to be used as test patterns for the state variables (see Fig. 2).

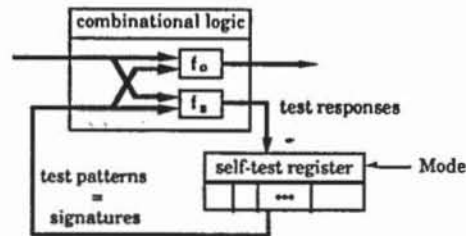


Fig. 2: Structure of controllers with parallel self-test.

In [KiHT 88] the use of a parallel self-test for data paths without such direct feedbacks was investigated. Empirical results indicated that for certain examples the concurrent use of MISR's (multiple input signature registers) for pattern generation and signature analysis does not cause a significant loss of fault coverage. Similar results were published for the circular self-test path approach, with which BIST overheads can be reduced [KrPi 89]. But both techniques cannot *guarantee* a high enough fault coverage and require extensive fault simulation. Moreover, in structures with direct feedbacks it might even be completely impossible to set the next state lines to all the values needed to detect certain faults [ChGu 89].

Several approaches are offered to overcome this problem. In [WaMc 87] the direct feedback path from storage elements to storage elements via the combinational logic is broken by doubling the number of flipflops and adding an additional self-test register only responsible for compacting the test responses. The state register itself is reconfigured as a pure pattern generator in self-test mode. In [HuPe 87] the same problem is treated for data paths, in which such "self-adjacent" registers also occur, by supplying test patterns through the scan path from an external pattern generator. Unfortunately both approaches result in significant hardware overheads.

In [ChGu 89] a special state assignment strategy is proposed, which guarantees that all the states are reachable in self-test mode. However, the state assignment is very restricted and cannot take the minimizability of the combinational logic into account. The additional amount of combinational logic thus might cost more than duplicating the number of storage elements, where it is possible to use optimized synthesis and state assignment techniques.

In this paper we present a novel circuit structure for parallel self-testable controllers. It avoids register duplication and in some cases even requires less combinational logic than the optimized synthesis techniques, while guaranteeing that all states of the circuit stay reachable in self-test mode. It decreases the speed disadvantage of self-testable designs and the self-test control effort by simplifying the self-test register. At the same time it makes it possible to perform a dynamic test to detect e. g. delay faults, which becomes more and more important. The goal is reached by explicitly considering the self-test structure during the synthesis process, instead of adding it in a special design for testability step as an afterthought, a strategy already successful in optimizing self-testable controller structures with separate pattern generators and response analyzers [EsWu 90].

In the next section we introduce some background material useful later on, before describing the target structure in section 3. In section 4 the testability of this structure is discussed and upper bounds on the aliasing probabilities are derived, as well as estimates of test length and test confidence. In section 5 we present a technique for synthesizing such a structure starting from a behavioral description. Finally, the concept is illustrated using the TAP controller of the IEEE boundary scan architecture [IEEE 90] as an example.

## 2 Background

A testable circuit has to be both controllable and observable. The observability of a sequential circuit can be increased by including its storage elements into a signature register. For the circuit to be controllable, one has to be able to force it into arbitrary states. These properties can be analyzed probabilistically by modelling the circuit as a Markov chain [Boot 67]. In the sequel we shortly summarize the used notations and facts, more details about Markov theory can be found in standard textbooks like [Fell 57].

For a sequence of random inputs  $I^0, I^1, \dots, I^t$ , the sequence of FSM states  $S^0, S^1, \dots, S^t$  can be described as a homogeneous Markov process with stationary transition probabilities

$$p_{ik} = \text{prob}\{S^t = S_k \mid S^{t-1} = S_i\}. \quad (1)$$

These probabilities depend on the probability of primary inputs and the FSM specification. Let  $(i_1 \dots i_p)$  be the vector of binary input variables and  $p_k = \text{prob}\{i_k = 1\} = 1 - \text{prob}\{i_k = 0\}$ . Then an input  $I = (i_1, \dots, i_p)$  has the probability

$$p_I = \prod_{i_k=1} p_k \cdot \prod_{i_k=0} (1 - p_k) \quad (2)$$

and the state transition probabilities are

$$p_{ik} = \sum_{S_k=f_a(I_j, S_i)} p_I \quad (3)$$

where  $f_a$  is the next state function of the FSM. If the probability of all the states is known for a certain time  $t_0$ , the state probabilities can be computed for all  $t > t_0$  with the Chapman-Kolmogorov equality

$$pS^t = pS^{t_0} \cdot P^{t-t_0}, \quad (4)$$

where  $pS^t$  represents the row vector of state probabilities  $pS^t = (\text{prob}\{S_1^t\} \dots \text{prob}\{S_n^t\})$  and  $P$  the matrix of transition probabilities  $P = (p_{ik})$ . The elements of the  $t$ -th power of  $P$  are denoted  $p_{ik}^{(t)}$ .  $P$  is a stochastic matrix, i. e. all the rows sum up to 1, whereas in general  $P$  is not doubly stochastic, i. e. the columns do not sum up to 1. A state  $S_k$  is called *reachable* from  $S_i$ , if a  $t \geq 0$  exists, such that  $p_{ik}^{(t)} > 0$ . If  $S_k$  is reachable from  $S_i$ , in the state transition graph there is a directed path from  $S_i$  to  $S_k$ . Two states  $S_i$  and  $S_k$  are called *mutually reachable*,  $S_i \sim S_k$ , if  $S_i$  is reachable from  $S_k$  and  $S_k$  is reachable from  $S_i$ . The relation " $\sim$ " partitions the state set into equivalence classes. If every state is reachable from every other state, there exists only one equivalence class containing all the states. A Markov chain is called *irreducible* if its state set consists of a single equivalence class with respect to mutual reachability. A set  $C$  of states is called *closed*, if no state outside of  $C$  can be reached from a state  $S_i \in C$ .

**Definition 1:** A sequential circuit is called *controllable*, if for every single stuck-at fault there exists an input sequence, which, starting from a specified state, leads to an incorrect next state or output.

When a circuit is controllable, the corresponding Markov chain has to be irreducible. If  $r$  memory elements are used to store  $n < 2^r$  specified states, not all the  $2^r - n$  (invalid) states added in the implementation necessarily satisfy this condition.

**Theorem 2:** If the combinational logic of a sequential circuit is irredundant and the state transition graph is strongly connected, the circuit is controllable.

**Proof:** Since the combinational logic is irredundant, for every single stuck-at fault there exists an input and a present state, which expose the fault, i. e. lead to a faulty next state or output. In particular there has to be a present state contained in the  $n$  states of the circuit specification [DMNS 90, Lemma 3]. If the state transition graph is strongly connected, there exists an equivalence class for " $\sim$ " containing all the specified states. This state set is closed, the corresponding Markov chain irreducible. Therefore all the specified states are reachable from a given valid state and there exists an input sequence, such that the sensitizing state is reached. ■

Since control units in system mode usually have strongly connected state transition graphs, they are controllable, provided that the combinational logic was made irredundant, which is necessary anyway to guarantee the detection of faults in the combinational logic.

A general problem of self-test registers is caused by dividing their functionality into a system mode and a self-test mode. In the test mode additional XOR-gates have to be in the data path, whereas in system mode these gates have to be disabled by some form of mode control logic, for which the control signals have to be provided externally. Moreover, the controllability of the circuit can be reduced in self-test mode, since the excitation function of the flipflops is changed. By switching to self-test mode, it becomes possible that a subset of states is no longer reachable, although it would be necessary to enter these states to sensitize certain faults. System mode and self-test mode give rise to two very different Markov processes.

Let  $M(s)$  be the next state of an MISR in autonomous mode. For a conventional state register consisting of D-flipflops, the next state is equal to the excitation variable (cf. Fig. 1)

$$s^+ = y = f_s(i, s), \quad (5)$$

whereas for a MISR state register the next state is generated by<sup>1</sup>

$$s^+ = y \oplus M(s) = f_s(i, s) \oplus M(s), \quad (6)$$

similar to T-flipflops, where the next state is  $s^+ = y \oplus s = f_s(i, s) \oplus s$ . If the Markov matrix in system mode is denoted  $P$ , this corresponds to a new

Markov matrix in self-test mode  $Q = (q_{ij})$  with  $p_{ij} = q_{ij} \oplus M(i)$ . Only in special cases is  $Q$  irreducible, such that the controllability is secured. One of these cases can be systematically utilized for a given FSM by assigning the state codes in a unique way [ChGu 89]. The approach also makes it necessary to assign a special behavior to the  $2^r - n$  invalid state codes not necessary to obtain the required system functionality. This way the symptoms of reduced controllability by switching to self-test mode are hidden, but the original cause is not removed.

### 3 Target Structure

To remove the cause of the controllability problem, it is necessary to avoid a second mode of operation with different state transition probabilities. This can be achieved by implementing the system functionality using the MISR in its signature analysis mode as state register. Because of the linearity of the operations involved, the necessary excitation variable  $y$  to produce a state transition from state  $s$  to state  $s^+$  can be computed easily and is

$$y = s^+ \oplus M(s) = f_s(i, s) \oplus M(s) = f_M(i, s) \quad (7)$$

compared with  $y = f_s(i, s) \oplus s = f_T(i, s)$  for T-flipflops and  $y = f_s(i, s)$  for D-flipflops (see Fig. 3). By implementing a pertinent next state function  $f_M(i, s)$  in the combinational logic, arbitrary controllers can be implemented with MISRs as state registers, which makes it unnecessary to provide a special system mode.

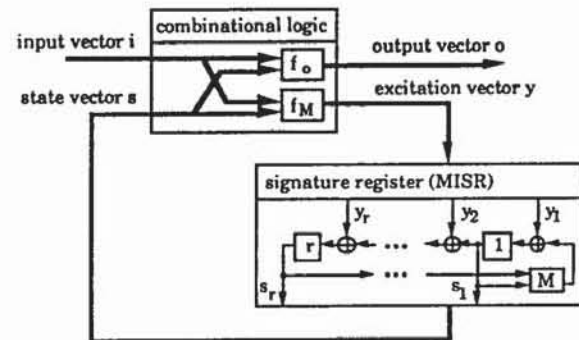


Fig. 3: Controller target structure.

The circuit structure for a parallel self-test without disjoint system and test modes has several advantages: Eliminating the D-flipflop mode reduces both the area of the self-test register and the number of signals to control it. Besides signature analysis the only other mode needed is a scan mode to initialize the flipflops and to shift out the resulting signature. As there is no reconfiguration of the

<sup>1</sup> The variables denote bit vectors,  $\oplus$  denotes a bitwise XOR-operation on these vectors.

flipflops in self-test mode, a test at the full clock frequency can be performed in order to detect dynamic faults relevant to system operation, e. g. delay faults, if only the test patterns for the primary inputs are supplied fast enough. By targeting the state assignment algorithm towards MISR state registers, the combinational logic needed to implement the system function can be optimized in the same way a controller with D-flipflops can be optimized. This way the overhead for parallel self-testable controllers is greatly reduced.

If the MISR is too short to achieve a reasonably low aliasing probability, it can be extended by combining it with other conventional signature registers or a circular self test path. This is illustrated in Fig. 4. During the test mode the multiplexer connects the registers to form one long signature register. In system mode it disconnects the state register of the controller from the rest of the circuit. With this solution the advantages of the parallel self-test structure of Fig. 3 are retained, because switching between the two modes does not require additional circuitry in the data path, the critical path is not changed, and the controllability of the parallel self-testable circuit is not reduced, provided that the contents of the signature registers combined are independent of each other.

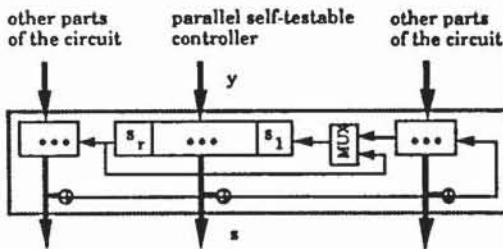


Fig. 4: Extended signature register to reduce aliasing.

#### 4 Testability Analysis

Since the state transition graph of the controller of Fig. 3 in self-test operation is identical to the system state transition graph, the corresponding Markov process contains an irreducible chain with all the specified states. If the combinational logic is irredundant, it is in principle controllable because of Theorem 1. The same fact obviously holds for the extended signature register of Fig. 4, if the values in the different parts of the signature register are independent. To use this controllability in practice, a pertinent sequence of input signals has to be supplied. This can be done by a circuit producing precomputed deterministic patterns [e.g. Daeh 83,

AkJa 89] or pseudorandom patterns with optimized input probabilities [e.g. Wund 87]. The test with pseudorandom patterns is analyzed in more detail later this section. All dynamic faults occurring in system mode can be excited in the structures of Fig. 3/4, since all the possible state pairs  $(s, s^+)$  occurring during system operation can also be produced in test mode given a pertinent input sequence.

A further question for parallel self-test structures is, whether the observability to be provided by replacing the state register with a signature register is not impaired because of increased fault masking. Let  $f$  be a fault which causes the circuit to enter a faulty state  $S_f^0$ . Assume that no more fault occurs after that point. The fault is masked after exactly  $L$  test patterns, if for the first  $L - 1$  patterns  $T^i$ ,  $1 \leq i < L$ , the circuit was in a faulty state  $S_f^i$  while producing correct outputs  $O^i$  and returns to the correct state  $S^L$  after pattern  $T^L$  (see Fig. 5).

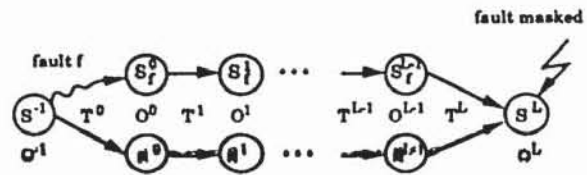


Fig. 5: Correct and faulty state sequence and fault masking.

Let the probability to produce the correct output in a faulty state be  $p_O$ , the probability to reach the correct next state from a faulty state  $p_S$  and let these probabilities be statistically independent. The probability  $P_f$  to go from a faulty state  $S_f^i$  to a faulty next state  $S_f^{i+1}$  while asserting the correct output  $O_i$  then is

$$P_f = (1 - p_S) \cdot p_O \quad (8)$$

The probability of fault masking after exactly  $L$  patterns  $P_M(L)$  is

$$P_M(L) = P_f^{L-1} \cdot p_S \cdot p_O = \frac{p_S}{1 - p_S} \cdot (1 - p_S)^L \cdot p_O^L \quad (9)$$

If after a fault was exposed  $N$  additional test patterns are applied, the fault can be masked by any of the patterns  $T^1, \dots, T^N$ . The overall fault masking probability  $P_M$  then is

$$\begin{aligned} P_M &= \sum_{L=1}^N P_M(L) = \frac{p_S}{1 - p_S} \cdot \sum_{L=1}^N [(1 - p_S) \cdot p_O]^L \\ &= \frac{p_S}{1 - p_S} \cdot \left[ \frac{1 - [(1 - p_S) \cdot p_O]^{N+1}}{1 - (1 - p_S) \cdot p_O} - 1 \right] \\ &< \frac{p_S}{1 - p_S} \cdot \left[ \frac{1}{1 - (1 - p_S) \cdot p_O} - 1 \right] \\ &= \frac{p_S \cdot p_O}{1 - (1 - p_S) \cdot p_O} \quad (10) \end{aligned}$$

**Example:** The following derivation is not necessarily applicable to concrete circuits, but should give an estimate of the order of magnitude fault masking probabilities would have. If the probability to assert the correct output in a faulty state is equal to the probability of obtaining any other output, we get  $p_0 = 2^{-q}$ , where  $q$  is the number of output variables. If the probability to reach the correct next state is equal to the probability to reach any other state, we additionally have  $p_s = 2^{-r}$ , where  $r$  is the number of state variables. For an extended signature register (Fig. 4)  $r$  is increased accordingly. Thus we get

$$P_M < \frac{1}{2^{r(2^q-1)+1}} = 2^{-(r+q)}. \quad (11)$$

For  $r = 5$  state variables and  $q = 20$  output variables, this formula gives a bound on the fault masking probability of  $P_M < 2,98 \cdot 10^{-8}$ .

In the sequel we analyze the dependence of input probabilities, test length and test confidence in the structure of Fig. 6, where weighted pseudorandom patterns are supplied at the primary inputs of the parallel self-testable circuit structure by a generator of unequiplausible random tests (GURT, [Wund 87]). For that purpose some more background about Markov processes is needed.

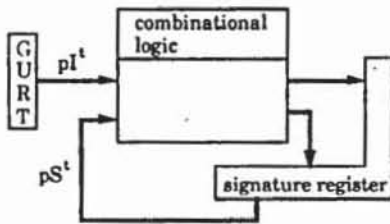


Fig. 6: Parallel self-test with pseudorandom patterns.

A vector of state probabilities  $pS^s$  in a Markov process with transition probability matrix  $P$  is called *stationary*, if  $pS^s = pS^s \cdot P$ . A state  $S_i$  has a *period*  $t_p$ , if  $p_{ii}^{(t)} = 0$  for all  $t \neq k \cdot t_p$ ,  $k \in \mathbb{N}$ , and  $t_p$  is the smallest such number. If there exists a state  $S_i$  with  $p_{ii} \neq 0$  and the Markov chain is irreducible, every state of the chain has the period  $t_p = 1$  [FrHW 79]. The chain is then called *aperiodic*. A Markov chain is called *ergodic*, if there exists a  $t$  such that for all state pairs  $S_i, S_k$   $p_{ik}^{(t)} > 0$ . Every irreducible and aperiodic Markov chain with finite state set is ergodic. For every irreducible and ergodic Markov chain the value  $\lim_{t \rightarrow \infty} pS^t$  exists and is identical with the unique stationary vector of state probabilities  $pS^s$  with

$\sum_{i=1}^n \text{prob}[S_i] = 1$  [Fell 57]. Using the Jordan normal form of matrix  $P$  and sorting the eigenvalues in descending order  $1 = \lambda_1 > |\lambda_2| \geq \dots \geq |\lambda_n| > 0$ , we have

$$\lim_{t \rightarrow \infty} P^t = \lim_{t \rightarrow \infty} \sum_{i=1}^n \lambda_i^t \cdot P_i = P_1. \quad (12)$$

Hence the speed with which the stationary distribution  $pS^s = \lim_{t \rightarrow \infty} pS^0 \cdot P^t = pS^0 \cdot P_1$  of state probabilities is reached mainly depends on the second largest eigenvalue  $|\lambda_2|$  of the Markov-Matrix  $P$ .

Since most state transition graphs of controllers contain a transition from a state to itself, e. g. from the reset state back to the reset state with a reset signal, the corresponding Markov chain is aperiodic. As it is also irreducible, instead of time-variable state probabilities after a certain number of test patterns the stationary probability distribution  $pS^s$  may be used. Finding  $pS^s$  only requires solving a system of linear equations, whose parameters are determined by equation (3).

The confidence of a random pattern test of length  $N$  corresponding to the input probabilities  $pI$  is the probability to detect all possible faults of a fault set  $f_i \in F$  by applying  $N$  patterns. Let  $p_{fi}(pI, pS)$  be the probability that fault  $f_i$  is exposed at the primary outputs  $o$  or the flipflop excitation outputs  $y$  of the combinational logic if a random vector corresponding to  $pI$  is applied to the primary inputs and one corresponding to  $pS$  is applied to the state inputs. Formula (3) shows that the stationary distribution  $pS^s(pI)$  depends on the input probabilities, so does the fault exposure at the output and next state signals  $p_{fi}(pI, pS^s(pI))$ . Hence the probability that  $N$  random patterns according to  $pI$  do not expose  $f_i$  is  $(1 - p_{fi}(pI, pS^s(pI)))^N$ , and in [Wund 90] it is shown that all faults are exposed at least once with a probability very close to

$$J_N(pI) = \prod_{f_i \in F} (1 - (1 - p_{fi}(pI, pS^s(pI)))^N). \quad (13)$$

In addition we have to consider fault masking and obtain a test confidence  $C_N(pI)$  of

$$C_N(pI) = (1 - P_M) \cdot J_N(pI). \quad (14)$$

From (13/14) the necessary test length  $N$  to achieve a given test confidence  $C$  can be computed. A fault-free simulation has to be performed to make sure that after the pattern generator for the primary inputs returns to its initial state the state of the circuit under test is different from its initial state. Otherwise the same input / state combinations occur again and no new faults can be detected.

## 5 Synthesis Process

Until now we have shown that the circuit structure for a parallel self-test without disjoint system and test modes has several advantages and that the test can be performed with high fault coverage and low aliasing. In this section it is shown that by targeting the state assignment towards MISR state registers, the combinational logic to implement the system functionality can be optimized in the same way a controller with D-flipflops can be optimized.

If a MISR instead of D-flipflops is used as state memory, a function  $f_M(i, s)$  has to be implemented, which is different from the next state function  $f_s(i, s)$ . For the minimization of the combinational logic this difference is not very important, since the flipflop excitation function can be easily obtained from the current and next state codes and a MISR description with equation (7). This function together with the output function  $f_o(i, s)$  can then be minimized with conventional programs for combinational logic synthesis.

Further optimization requires an adaptation of the state assignment procedure to the modified target structure. If a conventional procedure is used, the state assignment is optimized such that  $y = f_s(i, s)$  is well minimizable. The same state assignment used for MISRs gives results, which do not differ significantly from random state assignments (see Table 1 for a PLA target). With a state assignment targeted to make  $f_M(i, s)$  well minimizable, the combinational logic can be implemented much more efficiently. This is also shown in Table 1 for some examples from the sequential synthesis benchmark set [MCNC 88].

example	MISR solution / state assignment ...		
	... optimized <sup>2</sup> for D-FF's	... random (Ø 50 tries)	... optimal for MISR's
bbtas	14	13.6	6
beecount	21	19.9	14
dk14	41	44.8	32
dk15	25	26.5	24
dk17	23	24.7	19
dk27	12	11.2	7
ex6	33	33.7	30
lion	7	7.5	7
mc	9	9.8	9
s8	11	10.7	8
shiftreg	9	7.3	3
tav	11	10.2	9
train4	9	7.0	7

Table 1: Number of product terms for  $f_M(i, s)$  and  $f_o(i, s)$ .

<sup>2</sup> State assignment was done with the program *nova* [ViSa 89].

The sequence of code bits influences the combinational logic for MISR state registers because of the direct dependence of excitation variables on the contents of other flipflops in the MISR. For  $r$  state variables and  $n$  states the number of non-equivalent state assignments is

$$SA(r, n) = \frac{2^r!}{(2^r - n)!} \quad (15)$$

It exceeds the number relevant to D-flipflops by a factor of  $r!$ . Conventional state assignment algorithms cannot cope with the complex dependences in MISR state registers. Algorithms for T- and JK-flipflops [WeDo 69, TuBr 74] have been published, but do not help since in these cases the value of the  $i$ -th excitation variable  $y_i$  only depends on the contents  $s_i$  of the  $i$ -th flipflop. Consequently it is necessary to develop a new state assignment algorithm for this application. Since state assignment is NP-hard [WoKA 88], heuristics have to be employed.

The goal is to devise a divide-and-conquer strategy, in which the set of states is recursively partitioned into two sets, one encoded with a code bit 0, the other with a 1. When only one state is left in a partition, its encoding is different from the encoding of all other states. The partitioning and assignment is done such that a cost function reflecting the complexity of realizing the next state and output logic is minimized. To obtain such a cost function, however, is more difficult than for D-flipflops because the values of the excitation variables depend on other state variables. The idea used is that once an encoding for one state variable  $s_{i-1}$  is fixed, the excitation variable  $y_i$  can be derived from  $s_{i-1}$  and the code of the next state variable  $s_i^+$ . Alternatively,  $s_i^+$  can be chosen such that  $y_i$  becomes as simple to implement as possible, so at any point in the state assignment process the cost of the next assignment can be estimated. In this column-based approach one state variable is assigned after the other for all the states (see Fig. 7).

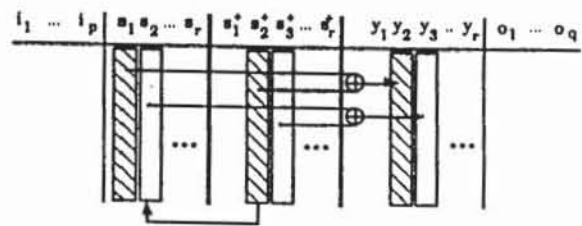


Fig. 7: FSM transition/excitation table with state assignment dependences.

Only the determination of the first state variable  $s_1$  is problematic, as the implementation effort for  $y_1$

cannot be estimated before all the other coding columns are known. However, it is possible to estimate its effect on the complexity of the output function  $f_o(i, s)$  and to use this information to choose  $s_1$ . After the state assignment is completed, an additional degree of freedom can be used to reduce the amount of logic for  $y_1$  by choosing the MISR feedback function  $M(s)$  in such a way that  $y_1 = s_1^+ \oplus M(s)$  is minimized. Even if a primitive feedback polynomial is required, in general there are still several choices. The synthesis process for controllers using the target structure of Fig. 3/4 is summarized in Fig. 8.

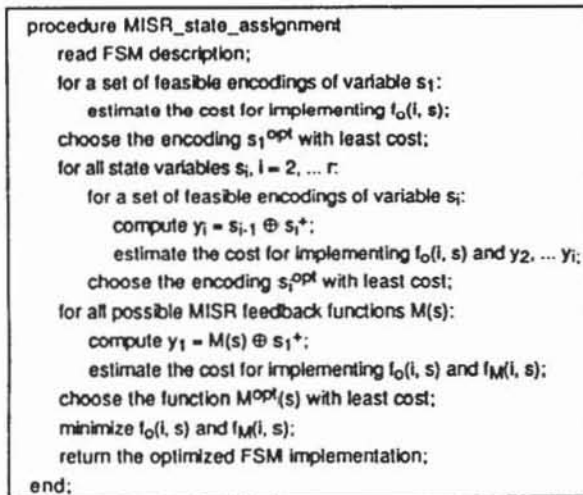


Fig. 8: Synthesis process for controllers with MISR state registers.

## 6 Example: Boundary Scan Controller

For implementing a chip conforming to the IEEE P1149.1 Boundary Scan Standard [IEEE 90], a standardized test interface for controlling the on-chip test equipment has to be provided. The test interface consists of an instruction register, boundary scan registers and a control unit to coordinate different test actions. Obviously it is desirable for this test access port (TAP) to be tested as well. An external functional test of the TAP controller requires  $589 + 23 N_i + 12 N_d$  test patterns, where  $N_i$  is the width of the instruction register and  $N_d$  the width of the data register [DaUY 89]. A structural test requires a scan path to be incorporated, e. g. an LSSD solution like the one proposed in [IEEE 90]. Since the boundary scan pins may not be used to test the TAP itself, additional pins would be necessary for that purpose.

The circuit structure of Fig. 3/4 allows to implement a self-testable TAP controller without additional pins, which tests itself while it is used without requiring a separate test phase. Using the optimized synthesis procedure it can be realized with a low hardware overhead (see Fig. 9). Since the controller is a Moore automaton, output and next state logic are realized separately, MISR<sub>1</sub> is used as state register. Like in [DaUY 89] the instruction register is included in the test. Some outputs of the controller as well as the instructions are observed in an additional signature register MISR<sub>2</sub>.

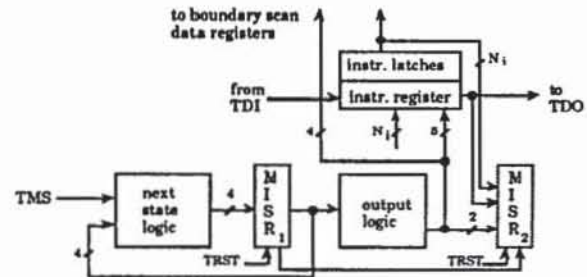


Fig. 9: Parallel self-testable boundary scan TAP controller.

While used, the circuit is controlled by the primary inputs TRST (Test Reset), TDI (Test Data In) and TMS (Test Mode Select). These inputs concurrently serve as test patterns; additional test patterns for the output logic and the next state lines are automatically produced by MISR<sub>1</sub>. The test responses of the controller's next state and output logic as well as the contents of the instruction register are compacted in the signature registers MISR<sub>1</sub> and MISR<sub>2</sub>, which are connected according to Fig. 10 to reduce the probability of aliasing.

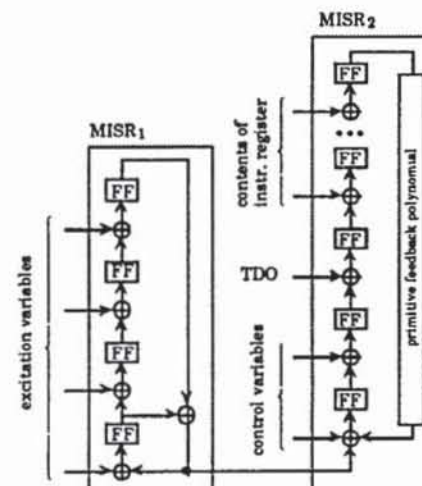


Fig. 10: Signature analysis in the TAP controller.

In order to determine the aliasing properties of these registers we use a result of [WiDa89]:

**Lemma 3:** If the transition matrix of a signature register of length  $R$  is nonsingular, its aliasing probability for long test sequences converges towards  $2^{-R}$ .

**Theorem 4:** For long test sequences the fault masking probability of two signature registers connected according to Fig. 10 converges to  $2^{-R}$ , where  $R$  is the sum of the lengths of both signature registers, if none of the signature registers is degenerated.

**Proof:** The combined signature register is described by the equation  $s^+ = M \cdot s \oplus y$ , where the transition matrix  $M$  depends on the feedback polynomials  $m^1(x) = 1 + m_1x + \dots + m_r x^r$  of MISR<sub>1</sub> and  $m^2(x) = 1 + m_{r+1}x + \dots + m_R x^R$  of MISR<sub>2</sub>:

$$M = \begin{bmatrix} m_1 & m_2 & \dots & \dots & m_r & 0 & \dots & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 & \vdots & \dots & \dots & \dots & \vdots \\ 0 & 1 & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots & \dots & \dots & 0 \\ m_1 & m_2 & \dots & \dots & m_r & m_{r+1} & m_{r+2} & \dots & \dots & m_R \\ 0 & \dots & \dots & \dots & 0 & 1 & 0 & \dots & \dots & 0 \\ \vdots & & & & & 0 & 1 & \dots & \dots & \vdots \\ \vdots & & & & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 & 1 & 0 \end{bmatrix}$$

If none of the registers is degenerated,  $m_r = m_R = 1$ . Then the determinant of  $M$  is different from 0, i. e.  $M$  is nonsingular. Therefore with Lemma 3 we get Theorem 4. ■

Most of the output signals of the controller can be observed indirectly via the instruction register, so they can be excluded from the signature analysis process. The resulting signature can be checked via the boundary scan path by connecting MISR<sub>2</sub> to the boundary scan data registers. The interface to the self-test control on the next higher hierarchical level stays the same, the test of the TAP controller only extends the length of the signature to be shifted out.

design alternative	total area in $\lambda^2$	crit. path (comb. logic) in ns	next state logic	output logic
conv. self-test	768 × 683	524544	45	21
parallel self-test	659 × 616	405944	33	25

Table 2: Layout results for the TAP controller.

In Table 2 we compare two standard cell designs of the TAP controller, one with separate pattern generation and signature analysis registers

(conventional self-test) and the combinational logic from [IEEE 90], the other with MISR state register and parallel self-test. The savings both in area and in delay are significant. (The signature register MISR<sub>2</sub> for observing the primary outputs, which would be necessary in both cases, was not included in the comparison shown below.) The state encoding used to obtain these results for the state register in Fig. 10 is given in Table 3.

Test-Logic-Reset	0000	Update-DR	1000
Run-Test/Idle	0010	Select-IR-Scan	0110
Select-DR-Scan	0100	Capture-IR	1110
Capture-DR	1010	Shift-IR	1111
Shift-DR	1011	Exit1-IR	0101
Exit1-DR	0001	Pause-IR	0111
Pause-DR	0011	Exit2-IR	1101
Exit2-DR	1001	Update-IR	1100

Table 3: State encoding for the TAP controller.

For a complete test of the instruction register it is sufficient to exercise the 4 possible state transitions  $0 \rightarrow 0, 0 \rightarrow 1, 1 \rightarrow 0, 1 \rightarrow 1$  for each register cell. While using the boundary scan hardware of a chip, this will generally be ensured. To obtain a simple model of TAP usage, we assumed an input sequence, in which the instruction EXTEST (instruction code 00...0) and the instruction BYPASS (instruction code 11...1) are read in. In between data are shifted into the boundary scan data register, are enabled, and test responses are shifted out again. This roughly corresponds to a connection test using boundary scan.

With this short input sequence it is already possible to detect over 99% of all single stuck-at faults in the circuit (controller + instruction register) including the memory elements by comparing the resulting signature with the correct signature. The other faults are potential detects and correspond to stuck-at faults of the reset signals (e. g. TRST). They can only be safely detected, if the flipflop contents are defined before the reset.

In summary a boundary scan controller with the parallel self-test structure of Fig. 3/4 was designed, which is automatically tested during the normal usage of the boundary scan facilities. Neither have special test patterns to be applied, nor are special test control signals necessary. The solution only requires moderate hardware overheads and does not slow down system operation. It facilitates the detection not only of static faults but also of dynamic and even transient faults occurring while the circuit is working.



We also validated the results concerning a test with random patterns. For a random test of the TAP controller with  $pI = \text{prob}[TMS = 1] = \frac{1}{2}$  the second largest eigenvalue is  $\lambda_2 = 0.809$  and starting in the state "Test-Logic-Reset" the vector of state probabilities  $pS^t$  satisfies the condition  $\|pS^t - pS^0\| < 0.001$  already after 15 test patterns. To obtain a probability of 99.9 % for a detection of all faults,  $N = 1349$  random test patterns are needed. This number is only 30 % larger than the 1057 patterns that would be needed if the 1-probabilities of all state variables could be independently set to  $\frac{1}{2}$ . A detailed fault simulation shows that the proposed parallel self-test technique is able to achieve a 100 % coverage of all stuck-at faults and aliasing can be greatly reduced by connecting the MISR state register to the signature register for the outputs, whereas for a state register implementation with D-flipflops fault masking would cause serious problems (see Fig. 11).

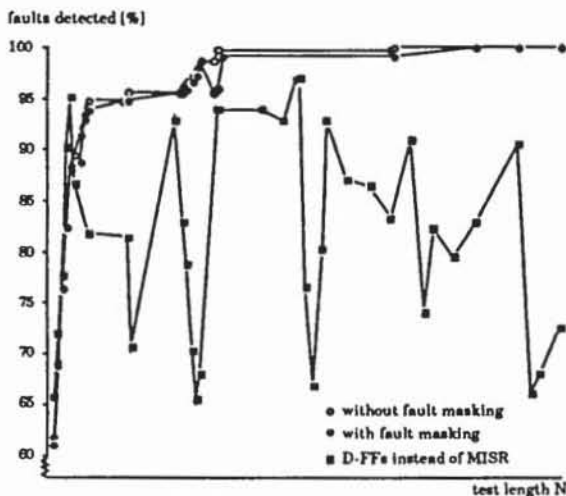


Fig. 11: Fault coverage as obtained from fault simulation.

## 7 Conclusions

A novel circuit structure for sequential circuits with parallel self-test was presented, in which the signatures can be safely used as test patterns. With this structure the self-test register is simplified as well as the self-test control. Testing for dynamic faults, e. g. delay faults, becomes possible.

The best self-test solution is to apply weighted pseudorandom patterns at the primary inputs of the circuit to sensitize possible faults. To optimally utilize the ensured controllability of the novel circuit structure then requires new algorithms, with

which the weight distributions of pseudorandom patterns can be determined such that for a given test confidence level the test length is minimized. More work is also needed to efficiently obtain closer estimates of the fault masking probability based on the structure of the circuit under test.

Hardware overheads can, however, be kept low by explicitly considering the self-test structure during the optimization steps of the synthesis process. As example application the design of a parallel self-testable boundary scan controller was presented and the possible savings were shown.

## References

- AkJa 89 S. Akers, W. Jansz: Test Set Embedding in a Built-In Self-Test Environment; Proc. Int. Test Conference, pp. 257-263, 1989.
- BeMa 84 F. Beucler, M. Manner: HILDO: The Highly Integrated Logic Device Observer; VLSI Design, pp. 89-96, June 1984.
- Boot 67 T. Booth: Sequential Machines and Automata Theory; Wiley, New York, 1967.
- ChGu 89 C. Chuang, A. Gupta: The Analysis of Parallel BIST by the Combined Markov Chain (CMC) Model; Proc. Int. Test Conference, pp. 337-343, 1989.
- Daeh 83 W. Daehn: Deterministische Testmuster-generierung für den eingebauten Selbsttest von int. Schaltungen; NTG-Fachberichte 82, Großintegration, pp. 16-19, 1983.
- DaUY 89 A. Dabhura, M. Uyar, C. Yau: An Optimal Test Sequence for the JTAG/IEEE P1149.1 Test Access Port Controller; Proc. Int. Test Conference, pp. 55-62, 1989.
- DMNS 90 S. Devadas, H. Ma, A. Newton, A. Sangiovanni-V.: Irredundant Sequential Machines Via Optimal Logic Synthesis; IEEE Trans. on CAD, vol. 9, pp. 8-18, 1990.
- EsWu 90 B. Eschermann, H.-J. Wunderlich: Optimized Synthesis of Self-Testable Finite State Machines; Proc. 20th Int. Symp. Fault-Tolerant Computing, pp. 390-397, 1990.
- Fell 57 William Feller: An Introduction to Probability Theory and its Applications; Wiley, New York, 1957.
- FrHW 79 F.-J. Fritz, B. Huppert, W. Willems: Stochastische Matrizen; Springer, Berlin, 1979.
- HuPe 87 C. Hudson, G. Peterson: Parallel Self-Test with Pseudo-Random Test Patterns; Proc. Int. Test Conference, pp. 954-963, 1987.
- IEEE 90 IEEE Std. P1149.1: Standard Test Access Port and Boundary-Scan Architecture, 1990.
- KjHT 88 K. Kim, D. Ha, J. Tront: On Using Signature Registers as Pseudorandom Pattern Generators in Built-in Self Testing; IEEE Trans. on CAD, vol. 8, pp. 919-928, 1988.

- KoMZ 79 B. Könemann, J. Mucha, G. Zwiehoff: Built-In Logic Block Observation Techniques; Proc. Int. Test Conference, pp. 37-41, 1979
- KrPi 89 A. Krasniewski, S. Pilarski: Circular Self-Test Path: A Low-Cost BIST Technique for VLSI Circuits; IEEE Trans. on CAD, vol. 8, no. 1, pp. 46-55, 1989.
- MCNC 88 R. Lisanke: Logic Synthesis and Optimization Benchmarks, Version 2.0; Microelectronics Center of North Carolina, 1988.
- TuBr 74 G. Tumbush, J. Brandeberry: A State Assignment Technique for Sequential Machines Using J-K Flip-Flops; IEEE Trans. on Computers, vol. 23, pp. 85-86, 1974.
- ViSa 89 T. Villa, A. Sangiovanni-V.: NOVA: State Assignment of Finite State Machines; Proc. 26th Design Automation Conference, pp. 327-332, 1989.
- WaMc 87 L. Wang, E. McCluskey: Built-in Self-Test for Sequential Machines; Proc. Int. Test Conference, pp. 334-341, 1987.
- WeDo 69 P. Weiner, T. Dolotta: Mixed Memory Realizations of Sequential Machines; IEEE Trans. on Computers, vol. 18, pp. 272-277, 1969.
- WiDa 89 T. Williams, W. Daehn: Aliasing Probability for Multiple Input Signature Analyzers with Dependent Inputs; Proc. 3rd CompEuro, pp. 5.120-5.127, 1989.
- WoKA 88 W. Wolf, K. Keutzer, J. Akella: A Kernel-Finding State Assignment Algorithm for Multi-Level Logic; Proc. 25th Design Automation Conference, pp. 433-438, 1988.
- Wund 87 H.-J. Wunderlich: Self Test Using Unequiprobable Random Patterns; Proc. 17th Int. Symp. Fault-Tolerant Computing, pp. 258-263, 1987.
- Wund 90 H.-J. Wunderlich: Multiple Distributions for Biased Test Patterns; IEEE Trans. on CAD, vol. 9, no. 6, pp. 584-593, 1990.