On the problem of masking special errors by signature analyzers [1]

Lutz Voelkel [2]

TR–95–014
April 1995

---

[1] Extended version of a talk given at ICSI, February 28, 1995.
To be self-contained, some results already published in other papers are included in this report.

[2] FB Mathematik/Informatik der Universität Greifswald;
F.-L.-Jahnstraße 15 a,
D-17487 Greifswald, Germany.
e-mail: voelkel@uni-greifswald.d400.de

Abstract

Signature analysis is an important compact method in digital testing. Applying this method, a test response sequence of a device under test is compressed by a linear feedback shift register (LFSR, for short). Masking occurs if a faulty device yields the same signature as the corresponding good device. Due to the linearity of any LFSR, this happens if and only if the 'error sequence' which is obtained by the 'exor' operation from the correct and the incorrect sequence, leads to the zero signature.
The masking properties of signature analyzers depend widely on their structure which can be expressed algebraically by properties of their 'characteristic polynomials'.

There are three main directions of research in masking properties of signature analyzers:
(i) more general masking results either expressed by the characteristic polynomial or in terms of other LFSR properties;
(ii) 'quantitative' results, mostly expressed by computations or estimations of error probabilities;
(iii) 'qualitative' results, e.g. concerning the general possibility or impossibility of LFSR to mask special types of error sequences.

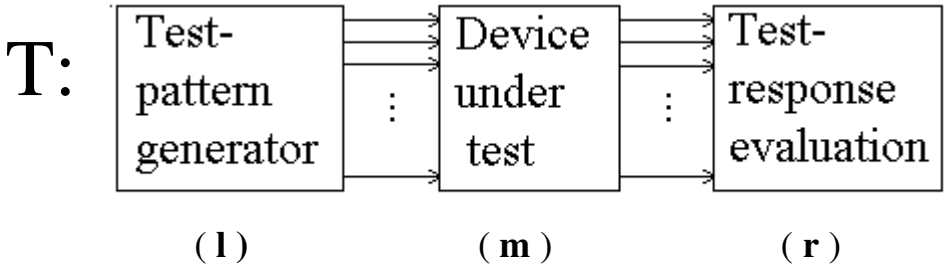
Following the third direction, we present a survey of masking properties of signature analyzers concerning error sequences having any odd weight, in the lecture. There are some results but also many open problems in this field. We have found some further insights in these problems by computer simulations.

0. Introduction

In the field of fault diagnosis of digital devices, signature analysis is well known as a special compact method used as well for testing and troubleshooting digital systems as for built-in self-testing of integrated circuits.
The basic idea of signature analysis consists in compressing any test response sequence of a device under test using a linear feedback shift register, and in comparing only the result of this compression (the so-called **signature**) with the signature of the response sequence obtained from the corresponding good device. As a consequence of the compression, it may occur that some faulty device yields a response sequence different from the sequence of the good device, but the corresponding signatures are equal. In this case the fault will not be detected by signature analysis, and this situation has been termed **masking**, or, in other references, **aliasing**.
In many articles, starting with the papers [BCA], [F], [D], [KMZ], and [S], for example, various results about masking have been published. The study of masking in signature analysis is based on an interesting mathematical background, in algebra and probability theory as well as in automata and graph theory. On the other hand, there are some open problems which can be rather easily formulated.
The report is divided into three main sections. Section 1 contains a short introduction in the field of testing digital devices. After some general remarks on compact test methods which are presented in Section 2, the masking problem by signature analysis will be considered in Section 3. Finally, some concluding remarks will be given.

1. On digital (hardware) testing

The following picture illustrates the main components of a general test system for digital hardware.



The middle component (**m**), the **device under test** (DUT, for short), can be an integrated circuit, a board, or a digital system, for example. On the other hand, from its logical structure, it can be a combinational circuit as well as a (clocked or unclocked) sequential circuit.
It is a necessary precondition for any kind of testing that there is a special fault model. Traditionally, the 'stuck at 0/1' model is widely used. In this model, faults consist in restricting the logical value of some internal wire to the constant 0 or 1, respectively, starting on a special location (such a gate input or gate output, for example).
On the other hand, some different or extended or more special (the latter one for particular kinds of circuits) fault models, respectively, are also considered in the literature of testing. More information can be found in [R], Chapter 3.
The aim of testing consists in 'stimulating' the inputs of the DUT in such a manner than many internal fault (defined by the fault model) will be detected by some corresponding output values which are said to be the test responses. In testing theory, the words „control and observe“ are used to express the relation between test inputs and test responses with respect to fault detection.

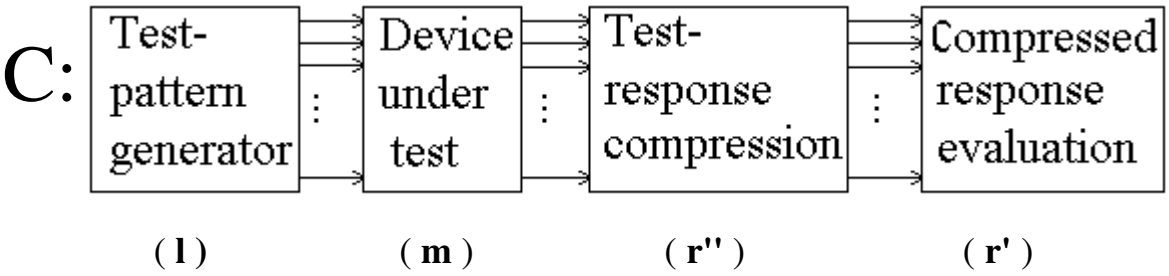
The left component (**l**) of the figure **T**, the **test pattern generator**, has to solve the problem of generating a sequence of inputs such that a maximal number of potential faults of the DUT (again, defined by the fault model) can be observed by some responses. With other words, a good 'fault coverage' is the main goal of the test pattern generation. It is easy to imagine tha this problem can be very complex. In fact, already in 1975 it has been proved by Ibarra and Sahni that there are NP-hard problems with respect to the test pattern generation even for rather restricted combinational circuits [IS]. For sequential cirquits, however, there are some PSPACE-complete testing problems [YL], and in recent papers [BJ1], [BJ2] further results can be found.
In general, the length of the test input sequences will increase if the DUT will become more complex. For practical applications, there are some rules ('design for testability'; cf. [R], Chapter 5, or, in particular, [WP], for example) concerning the circuit design which should lead to an easier testing. As an important example, we mention the modularity where some smaller parts of the DUT could be tested either exhaustively or randomly. There is a lot of papers concerning properties of exhaustive, pseudoexhaustive, or random testing.

Finally, the right component (**r**) of the figure **T** is devoted to the **test response evaluation**. The exact content of this component depends widely on the kind of the DUT, or on the background of the test method. In general, there are some different classifications.
At first, we must distinguish between a pure „go – don't go go“ decision (which is sometimes also denoted by „testing“ but in a restricted sense), and a two-level procedure which is said to be **fault diagnosis**, in general. This procedure consists in a testing stage followed by a second stage which will be applied if a fault has been found, und which has the objective of fault localization (or, equivalently, fault isolation, or fault finding). Such a diagnosis is mostly applied to boards or systems but there are also applications to integrated circuits.

Secondly, a diagnosis can be executed during several phases in the design, fabrication or even application of a digital device. We mention only the design (or prototype) test, and the production diagnosis performed by the producer as well as some diagnostic tools used in service. Latter ones are off-line techniques, in general, but there are also on-line methods for fault diagnostics.
Finally, a diagnosis procedure can be applied 'externally' controlled by some „automatic test equipment“ (ATE, for short) but there are also 'internal' test methods using (almost) only such tools which are parts of the DUT itself. Such devices are commonly referred to be „Built-In Self-Test“ tools (BIST, for short, cf. [R], Chapter 5, or, in particular, [MC1], [MC2], for example). In particular for diagnosis methods of the latter kind but also in other cases (if a fault isolation is based on 'fault dictionaries', for example), the amount of the diagnostic data can be too large such that the evaluation of the test responses would become very inefficient. For such cases, **compact methods** have been developed which lead to a reduction of this data amount.

2. Compact methods in test response evaluation

If a compact test method will be used then the structure of the general test system **T** considered in the last section will be changed in the following way.



In this figure, the right component (**r**) from the original picture **T** has been changed into a similar device (**r'**). The new device, however, does not evaluate the complete test response sequences but only the result of some data compression. This compression will be performed by the completely new component (**r''**) which is not contained in the figure **T** but which is now directly connected to the output lines of the DUT.
The response data compression can be performed either **serially**, or **in parallel**, or in any 'mixed' manner. A purely parallel compression yields a 'global' value C describing the complete behaviour of the DUT; such a method should be mainly applied for „go – don't go“ tests. On the other hand, if additional informations will be needed for fault localization then a serial compression technique has to be preferred. Using such a method, a special compressed value $C(R_i)$ will be generated for any output response sequence R_i where i ranges between 1 and the number of output lines of the DUT.

2.1. Examples

In the following we will shortly consider some examples of serial data compressions which are sufficiently well studied, and which have been successfully applied in the field.
Let $X=(x_1,...,x_t)$ be some binary sequence. Then, the sequence X can be compressed as follows.

a) By **transition counting:** $T(X) = \sum_{i=1}^{t-1} (x_i \oplus x_{i+1})$ (Hayes, 1976);

here the symbol '⊕' is used to denote the addition modulo 2, but the sum sign must be interpreted by the usual addition;

b) by **syndrome testing** (or **ones counting**)**:** $Sy(X) = \sum_{i=1}^{t} x_i$ (Savir, 1980);

c) by **accumulator compression testing:** $A(X) = \sum_{k=1}^{t} \sum_{i=1}^{k} x_i$ (Saxena, Robinson 1986).

In each one of these cases, we have the 'compression rate' n => O(log n).

The following well-known methods lead even to a constant length of the compressed value:
d) the **parity**

$$P(X) = \bigoplus_{i=1}^{t} x_i ,$$

where the bigger symbol '⊕' is used to denote the repeated addition modulo 2, and
e) the **cyclic redundancy check** (CRC, for short), performed by a linear feedback shift register of some fixed length $\mathbf{n \geq 1}$.
Such devices will be considered in the next section. Here it should only be mentioned that the parity test is a special case of the CRC for $\mathbf{n = 1}$.

Examples of parallel and/or mixed data compressions can easily be given by extending the methods discussed above on more than one output sequence. A purely parallel compression is performed by parallel signature analyzers which will be shortly mentioned in the concluding remarks.

2.2. Masking

The data compressions considered in this field have the disadvantage of some loss of information. In particular, the following situation may occur.
Let us suppose that during the diagnosis of some DUT, any 'expected' sequence X^o will be changed into a sequence X due to any fault F such that it holds $X^o \neq X$. In this case, the fault would be detected by monitoring the complete sequence X.
On the other hand, after applying some data compression C, it may be that the compressed values of the sequences are the same, i.e. $C(X^o) = C(X)$. Consequently, the fault F which is the cause for the change of the sequence X^o into X cannot be detected if we only observe the compression results instead of the uncompressed sequences.

This situation is said to be **masking** or **aliasing** of the fault F by the data compression C. Obviously, the background of masking by some data compression must be intensively studied before it can be applied in compact testing. In general, the masking probability must be computed or al least estimated, and it should be sufficiently low. Many papers on compact methods contain such estimations.

Let us consider some examples of masking by the first four data compressions considered above.
For the input sequence
X^o = (1,1,0,0,1,0,0,0), we get $T(X^o) = Sy(X^o) = 3$, $P(X^o) = 1$, and $A(X^o) = 19$.

Now consider the sequence
X = (1,0,0,1,1,0,0,0), which may be obtained from X^o by a fault which has changed as well the second as the fourth bit (and which has therefore generated a so-called two bit error).
Obviously, it holds $T(X) = Sy(X) = 3$, $P(X) = 1$, and therefore this fault would not be detected by transition counting, syndrome testing, and parity check.

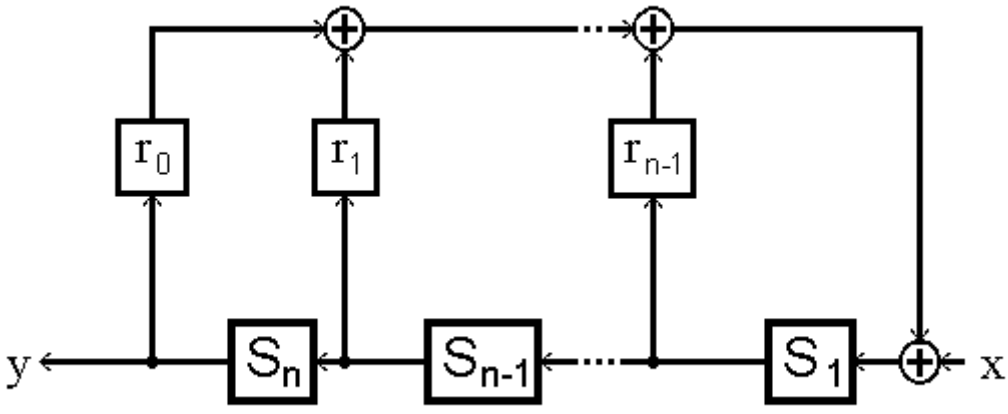
On the other hand, it holds $A(X^o) = 17$, and therefore this fault would be detected by accumulator compression testing. It is something more difficult to construct a sequence X' such that $X' \neq X^o$ but $A(X') = A(X^o)$; an example is the sequence X'= (1,0,1,0,0,1,1,0).

3. Signature Analysis

In this section, signature analyzers are introduced as special devices to perform data compressions, and their masking properties are studied.

3.1. Basic notions

A general n-stage signature analyzer (n-SA, for short) is a linear feedback shift register **S** which is built up in the following way, where the symbols $\mathbf{S_i}$ are used to denote memory elements, and the symbols $\mathbf{r_i}$ for binary multipliers (indicating the presence or absence of a feedback connection after $\mathbf{S_{n-i}}$!), i = 1,...,n. Moreover, the symbol ⊕ must be interpreted by a modulo-2 adder, and x and y denote the input and the output of the n-SA, respectively.



An n-SA is said to be **degenerated** if it holds $\mathbf{r_0} = 0$, and **non-degenerated** otherwise. Moreover, the 'extremely degenerated' n-SA without any feedback connection (i.e. $\mathbf{r_i} = 0$ for i = 0,...,n-1) will be referred to be the **trivial** n-SA. In most cases, however, only non-degenerated SAs will be considered.

For any n ≥1, an n-SA is a deterministic linear automaton $S = (V, V, V^n)$, i.e. its input space as well as its output space is the one-dimensional vector space V = {0,1} over the Galois field GF(2)), and its space of states is the n-dimensional vector space V^n.
The behaviour of S can be described by using the three matrices $B=(1\ 0\ ...\ 0)^{\mathbf{T}}$ (the input matrix), C=(0 ... 0 1) (the output matrix), and the following **system matrix** A.

$$\mathbf{A} = \begin{bmatrix} r_{n-1} & r_{n-2} & \dots & r_1 & r_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} .$$

For any state $s=(s_1,...,s_n)^{\mathbf{T}}$ of the memory elements $S_1,...,S_n$, and any binary input x, the next state s', and the output y of the n-SA are then given by the vector equations s'=As+Bx, and y=Cs. We remark that the states are represented by column vectors which is indicated by the transposition symbol „$^{\mathbf{T}}$ “.
The binary constants $r_0,...,r_{n-1}$ defining the feedback connections can be regarded as the coefficients of a polynomial $\mathbf{p_S}$ which has the degree n, and which is said to be the **characteristic polynomial** of the n-SA S**:**

$$\mathbf{p_S(x) = x^n+r_{n-1}x^{n-1}+...+r_2x^2+r_1x+r_0.}$$

As we will see later, the masking properties of any n-SA S depend widely on the „factorization state“ of its characteristic polynomial $\mathbf{p_S}$ (i. e., if p_S is reducible then they depend on the properties of its „prime factors“, or, if it is irreducible then they depend on its „primitiveness“ or „non-primitiveness“).

Using the initial state $ZERO:=(0,...,0)^{\mathbf{T}}$, the **signature S(X)** of any binary input sequence $X=(x_1,...,x_t)$ is then defined as the final state of the n-SA S which is reached after t steps, when all elements of the input sequence X have been processed. By this, the definition of the last kind of data compression presented in Section 2 is complete now. Therefore, masking could be regarded as a special case of the general definition given in Section 2 but we prefer to formulate the definition here once more explicitely.
Let X be any response data sequence produced by a device under test, and X^o the sequence generated by the same test pattern from the corresponding good device.

Then, **masking** or **aliasing** (of the **fault** which causes our device to produce the error pattern X instead of the correct sequence X^o) is defined by the conditions

(i) the sequences are different: $X \neq X^o$, and
(ii) they produce the same signature: $S(X) = S(X^o)$.

Due to the **linearity** of SAs, masking occurs if and only if $\mathbf{X \neq X^o}$, and $\mathbf{S(X+X^o) = ZERO}$.
Therefore, it is sufficient to consider only „**error sequences**“ $\mathbf{E=X \oplus X^o}$ (instead of all pairs (X, X^o) of erroneous and good sequences), and to study their „**error masking property**“ S(E)=ZERO.
It should be remarked that the data compressions T, Sy, and A defined in Section 2, are not linear, in general; counterexamples can be obtained by the examples considered above.

3.2. Masking results

Research on masking properties of signature analyzers play an important role in many papers on signature analysis. To some extent simplified, this research can be divided into three major directions.
The first one includes more general masking results which are based either on the characteristic polynomial or on other SA properties.

We illustrate this kind of studies by one example; the following fundamental result was firstly presented in [S]:

M1. Any error sequence $E=(e_1,...,e_t)$ is masked by an SA S if and only if its „error polynomial" $p_E(x) = e_1x^{t-1}+...+e_{t-1}x+e_t$ is divisible by the characteristic polynomial $p_S(x)$.

The second direction in masking studies which is represented in most of the papers concerning masking problems, can be characterized by „quantitative" results mostly expressed by some computations or estimations of masking probabilities. The following example of a result of this kind has already been presented in the early papers [BCA],[GN] and [F], and it can be easily obtained (by a counting argument) as a consequence of Smith's Theorem M1.

M2. If we suppose that all error sequences having any fixed length are equally likely the masking probability of any n-stage SA is not greater than 2^{-n}.

It is important to remark that the assumption of equal likelyhood of error sequences is not realistic, and therefore the value of this estimation is rather low. As Smith has pointed out, the result is true even for the trivial n-SA (which leads to the strange compact method to monitor only the last n bits of any test response sequence).

The third direction in studies on masking contains „qualitative" results concerning the general possibility or impossibility of SAs to mask error sequences of some special type. Examples of such a type (considered already in [S]) are burst errors, or sequences with fixed error-sensitive positions.
Traditionally, error sequences having some **fixed weight** are also regarded as such a special type, where the weight **w(E)** of some binary sequence E is simply its number of ones.
We will study masking properties for such sequences without restriction of their length. Some of the results are simple consequences of Smith's theorem but some of them have been published earlier (in some cases without any proof).
The first example of a result of this kind has in essence already been presented in [Fr] (announced yet earlier in [GN]) as an big advantage of signature analysis against transition counting. It should be remarked that the condition to be non-trivial has been mentioned only by Smith in 1980.

M3. If the SA S is non-trivial then masking of error sequences having the weight 1 by S is impossible.

Let us use this „traditional" result to divide our further discussion of masking properties into two parts with respect to error sequences which have any odd weight, or any even weight, respectively. Firstly we will consider error sequences of even weights; this case is in fact far less interesting than the other one.

The following property has been conjectured in [L] and proved in [Gö].

M4. For any SA S there exists some error sequence E such that w(e)=2, and E is masked by S.

Applying again Smith's theorem M1, this result is a consequence of the well-known fact that any polynomial p(x) divides some binomial x^m+x^k where k < m. It should be remarked that for non-degenerated SAs there is always such a binomial which has the form x^m+1.
Obviously, from the result M4 we can draw the conclusion that for any SA S and any even weight e≥2 there is some error sequence E such that w(E)=e and E is masked by S. This means, that masking of even weighted error sequences is possible by any SA.

For error sequences with any odd weight, however, this is not true; cf. the result M3 above. Moreover, the next masking property which has been observed already in [D] and [HL], characterizes a class of SAs which are generally not able to mask such error sequences.

M5. Any SA S cannot mask error sequences having any odd weight if and only if its characteristic polynomial $p_S(x)$ is divisible by the polynomial x+1.

Let S be any n-SA such that (x+1) divides the caracteristic polynomial $p_S(x)$ (obviously, this is equivalent to the fact that S contains some odd number of feedback connections). Then the result M5 is based on the decomposition of S into the „parity“ 1-SA P which has the characteristic polynomial x+1, and some other (n-1)-SA S' (cf. [Gi] or [VP] for more details). In a certain sense, this decomposition preserves the masking properties, in this special case the obvious property of the 1-SA P not to be able to mask any odd bit error sequence.
It should be mentioned that some properties of the parity are presented in the paper [C].

By the result M5, a classification of signature analyzers according to their ability to mask error sequences of any odd weigth, becomes very easy. Masking of sequences of this type by some SA S is impossible if and only if the number of feedback connections in S is odd , or, equivalently, if the number of terms in the characteristic polynomial p_S is even. It should be mentioned that such a classification of SAs has been already introduced in [HL] and [L] where it was based on observations about masking probabilities with respect to error sequences of some odd weight.

Let us now restrict our main attention to SAs with some even number of feedback connections. It depends widely on further properties of the characteristic polynomial p_S if such a SA S can mask some error sequence having any fixed odd weigth u, or not.
For u=1, a simple criterion is given by the result M3 above – one bit error sequences can be masked only by trivial signature analyzers.
But for any u ≥ 3, we do not know sufficient and necessary conditions which describe the ability of SAs to mask some error sequence E where w(E)=u.

In particular, for u=3, we have the following **open problem:**

P3. How can we characterize the class of those SAs which are able to mask some error sequence E such that w(E)=3 ?

By the result M1, this problem is equivalent to the following open question:

P3'. When does a polynomial of the form $x^n+r_{n-1}x^{n-1}+...+r_1x+1$ divide some trinomial x^G+x^K+1, where $1 < K < G$, over the field GF(2) **?**

Let us now present some weaker results related to the problem P3.

M6. The following conditions are *sufficient* that for any n-SA S *there is* some masked error sequence E such that w(E)=3:

a) the characteristic polynomial of S is

1. some power p^e of a primitive polynomial p which has a degree d ≥2, where e ≥1,

2. the product q·r of two primitive polynomials of degrees k, m such that k, m ≥2, and their exponents 2^k-1 and 2^m-1 do not have any common divisor d > 1;

b) there are at least $2^{n-1}+1$ states which are „0-successors" of the „unit state" $s_1=(1\ 0\ ...\ 0)^T$ (i.e. they are states of the form $A^k(s_1)$ for some k≥1)**;**

c) the state $(1\ 1\ ...\ 1)^T$ is a „0-successor" of s_1.

Remark: These results, in particular b) and c) are much easier to understand and to prove using the graph theoretic concepts presented in Section 3.3. We will discuss them later in that section.

On the other hand, there are also sufficent conditions for the **inability** of SAs to mask some error sequence which has, more generally, any fixed odd weight. Here, the family of SAs V_{2k} which have any even lenght 2k ≥ 4, and which are „complete" (i.e., there are feedback connections into all memory elements), play an imortant role.

M7. If for any k>1, the characteristic polynomial of any SA is some product $q\cdot v_{2k}$ including the „complete" 2k-degree polynomial $v_{2k}(x) = x^{2k}+x^{2k-1}+...+x^2+x+1$ then *it cannot mask* any error sequence E such that w(E) is odd, and $1 \le w(E) \le 2k-1$.

As a consequence of the result M1, we know that there are masked error sequences with the weight 2k+1, for the 2k-SAs V_{2k} itself (e.g., their characteristic polynomials v_{2k}).

It should be of some interest to observe that there are quite different factorizations of the polynomials v_{2k}, for k ≥ 2. For example, the polynomials v_{12}, v_{18}, and v_{28} are irreducible; v_{20},..., v_{26} are products of two factors; v_{32} of four, and v_{30} of six (all the primitive degree 5 polynomials!).

Obviously, v_2 is the only primitive polynomial of this kind; none of the other irreducible polynomials v_{2k} ($k \geq 2$) can be primitive because it holds $(x+1) \cdot v_{2k} = x^{2k+1}+1$.

Furthermore, the next statement gives a connection between different odd weights, it follows easily from the result M4 above.

M8. If for any odd integer u there is some masked error sequence having the odd weight u then there are also masked error sequences with any other odd weight v>u.

Similarly to P3 but more general, there are also problems 'Pu' regarding the other odd numbers u = 5, 7,... ; they consist in describing necessary and sufficient conditions for signature analyzers to be able to mask some error sequence E where u is the **exact** weight w(E) of the sequence E.

By the property M8, the following connection between these problems is given. For any n-SA S which has some even number of feedback connections, there is a odd number $u_0 \geq 3$, such that there are sequences masked by S which have some weight which is not less than u_0, but S is not able to mask some error sequence E such that w(E) is odd, and $w(E) < u_0$.

Therefore, the problems mentioned above can be slightly reformulated now, u = 5, 7,...:

'Pu': How can SAs be characterized which are able to mask some odd weight error sequence E if and only if w(E) ≥ u ?

3.3. Graph theoretic concepts

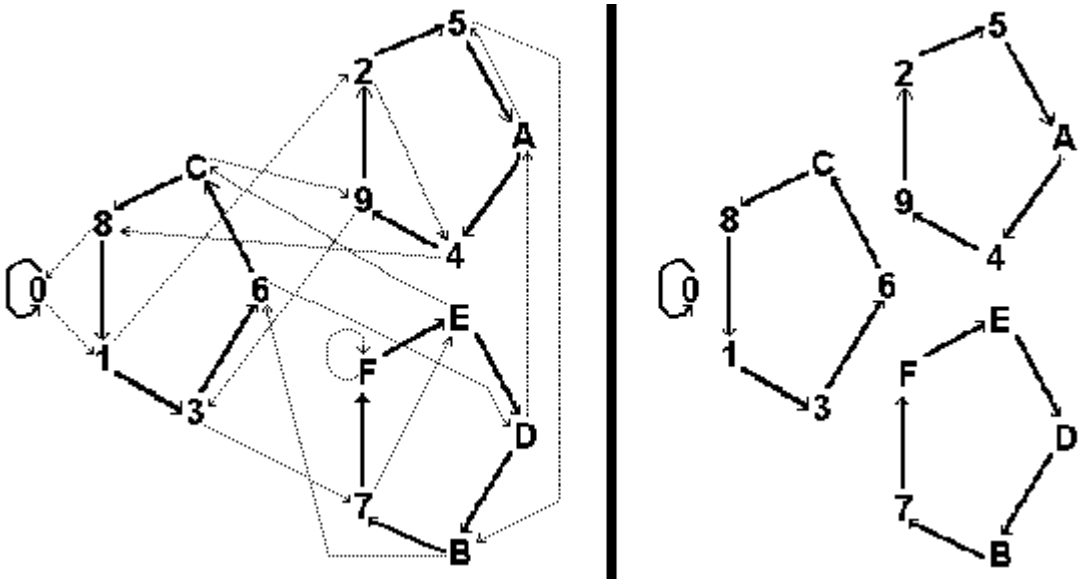
Now we introduce some types of graphs which are connected with SAs and which are useful to express results on masking by the corresponding SAs.
Let S be any n-SA, and A its system matrix. Then S can be represented by a directed graph $\mathbf{G_S=(V^n, E^0 \cup E^1)}$, which is said to be the **register graph**. In this graph, the vertices are the states of the SA S, and the „0-edges" from E^0 as well as the „1-edges" from E^1, respectively, are pairs (s,s') of states such that s'=As (i.e. s' is the immediate 0-successor of the state s), or s'=As+B (i.e. s' is the immediate 1-successor of s), respectively.
The following property is an easy consequence of definitions given above.

G0. Any error sequence which is masked by some SA S defines a uniquely determined path starting and ending in the state ZERO of the register graph G_S.

Moreover, the **autonomous graph** of any SA S is simply defined as the subgraph of the register graph G_S which contains only the 0-edges: $\mathbf{A_S=(V^n, E^0)}$.
If S is non-degenerated then the autonomous graph A_S consists of some cycles. In any case, the **zero cycle $\mathbf{C_0}$** which consists only of the state ZERO and its related zero edge which is a loop, and the **unit cycle $\mathbf{C_1}$** including the „unit state" $(1,0...,0)^T$ (where the one is stored in the first memory element S_1), are special components of the graph A_S.

To illustrate these notions by examples, we present the register graph and the autonomous graph for the „complete" 4-SA V_4 (we recall that $x^4+x^3+x^2+x+1$ is the characteristic polynomial of this SA) where the states are presented in the hexadecimal system, the 0-edges are drawn by thick lines, and the 1-edges of $\mathbf{G_{V_4}}$ are drawn by dotted lines, respectively.

Register graph $\mathbf{G_{V_4}}$ **Autonomous graph $\mathbf{A_{V_4}}$**



In general, the structure of the remaining components of the autonomous graph A_S depends widely on the characteristic polynomial of the SA. We present here only some special properties; more results and proofs can be found in [Gi], and [VP], for example.

G1. If the characteristic polynomial p_S of any n-SA S is irreducible then its autonomous graph A_S consists of the zero cycle C_0 and k other cycles where all of them have the same length l, and it holds $k \cdot l = 2^n - 1$. If, in particular, p_S is primitive then the unit cycle C_1 contains all the $2^n - 1$ non-zero states (i.e. k=1).

If any SA S has a reducible characteristic polynomial p_S then a first rough idea about the structure of the autonomous graph A_S can be obtained using the „factor polynomials" which are present in the decomposition of S. This can be done using the notion of the **cycle set** which contains informations on the number of cycles and their lengthes. The definition and further properties of cycle sets can be found in [Gi], [L], and [VP].

With regard to our problem of masking of odd weight error sequences by any non-degenerated SA S, it is important to know how the cycles of A_S are connected by 1-edges of the register graph G_S, from the following reason. While S is processing a subsequence of zero bits from any error sequence E, it will remain in the same cycle. Only by the input bit 1, it can be moved into another cycle. To get a description which is far less complicated than that one by the rather large register graphs, we now suppress the state transitions resulting from 0-inputs which are not essential with respect to the masking problem. We observe, however, all the possible transitions between cycles which are caused by 1-inputs. This leads to the following definition which is restricted here to non-degenerated SAs (cf. [He], [L], or [VP] for more details, and the latter reference for the general case).

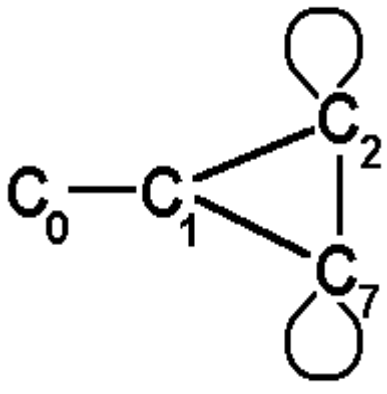
Let S be any n-stage SA. Then the **transition graph T_S** consists of the set of all cycles of the autonomous graph A_S as its vertex set, and of the edge set specified as follows. Two cycles C, C' are connected by an edge (C,C') if and only if there are states s in C and s' in C' such that the ordered pair (s,s') is an 1-edge in the register graph G_S considered above.

The following property of transition graphs is not difficult to prove ([VP]).

G2. The transition graph T_S is strongly connected for any SA S, and its edge relation is always symmetric.

As an example, we present the transition graph of the 4-SA V_4 Due to the symmetry mentioned above, all edges are drawn without directions.

The cycles C_i are named by their least member i, i = 0, 1, 2, 7. Each one of the cycles C_1, C_2, and C_7 consists of five states because the characteristic polynomial v_4 is irreducible (cf. the result G1 above).



It should be noted that graphs of this kind (undirected ones which may contain some loops) are usually not considered in graph theory. For our background in masking of odd weight sequences, however, some of these loops can be very important.

Our knowledge about further general properties of transition graphs is rather low. For any register length n such that $6 \le n \le 15$, it is mostly based on results of rather simple simulation programs. It should be of some interest to develop more efficient LFSR simulation algorithms to obtain further and better simulation results.
It must be mentioned here that one theorem (Satz 31, concerning isomorphisms between transition graphs) published in the book [VP] has been disproved by one of the very first running simulation programs! By the simulations based on this program we realized that there are different products of irreducible polynomials of the degrees 3 and 6 which lead to non-isomorphic transition graphs of the resulting 9-SAs. (In fact, the first of this graphs contains a loop on the unit cycle but the other one does not.) Fortunately, the theorem could be saved but only in a considerably weaker form (the preconditions must be similar to the formulation in the result M6a2 stated above).

To give precise notions of such special properties mentioned above, we can define the **depth $d(T_S)$** of the transition graph by the maximal level of its nodes, and the **breadth $b(T_S)$** by the maximal cardinality of all sets which include all the nodes which have the same level. Here, the **level $l(C_m)$** of any node C_m is simply its distance (which is defined in the usual way) from the zero cycle C_0. For any SA, the unit cycle C_1 is obviously the only node of the transition graph T_S which has the level 1.

It follows immediately from the property G1 that $d(T_S) = b(T_S) = 1$ for any n-SA S if its characteristic polynomial p_S is primitive. Therefore, the number 1 is a 'tight' lower bound for $d(T_S)$ as well as for $b(T_S)$ which occurs infinitely often.

On the other hand, it should be of some interest to find general nontrivial upper bounds of $d(T_S)$ and of $b(T_S)$ in terms of the register length. For two special types of SAs, which are closely related, the following simple estimations of $d(T_S)$ have been stated in [Pl], using notions and methods from combinatorics (in particular, Polya Theory).

G3 a) $d(T_S) = n+1$ if S is the „simple" n-SA such that $p_S(x) = x^n+1$;
b) $d(T_S) = m+1$ if $S = V_{2m}$ is the „complete" 2m-SA such that $p_S = v_{2m}$.

Now we return to our masking problem. The following property is a simple consequence from the result G0, and from the definition of the transition graph.

G0'. Any error sequence E can be masked by any SA S if and only if there is a path which has exactly the length w(E), and which begins and ends in the zero cycle, in the transition graph T_S.

Obviously, if the weight w(E) of a masked error sequence is a odd number then at least one edge of such a path in the transition graph T_S must connect two nodes having the same level. In general, an edge (C_i, C_k) belonging to T_S where $l(C_i) = l(C_k)$, is said to be a **cross-edge**. Moreover, the unique level of the two cycles connected by some cross-edge c will be said to be **the level of the cross-edge c** (l(c), for short).

Let now the integer u be some odd number, and $u \geq 3$, and let us return to the general problem 'Pu' considered in Section 3.2, where we asked for conditions about SAs S such that
(i) there is some error sequence E such that w(E)=u, and E is masked by S, ***and***
(ii) masking of any error sequences E by S is impossible if w(E) is odd, and w(E) < u.

Using the notion of a cross-edge, we can easily see that the existence of a cross-edge on the level $k=(u-1)/2$ in the transition graph T_S implies the condition (i).

Moreover, it is easy to prove the following result.
G4. The fact that $k=(u-1)/2$ is the lowest level in T_S where a cross-edge exists, is equivalent to the validity of the conditions (i) and (ii).

Therefore, it is important to search for the first (lowest-level) occurence of cross-edges in transition graphs.

In particular, we have to consider the level 1, if u=3. As stated above, on this level there is the unit cycle only. Therefore, a cross-edge on this level must be a loop. Using this fact, we can state a third equivalent form of the problem P3 from Section 3.2.

P3''. When does the transition graph T_S contain a loop on its first level ?

Now we can give some remarks with respect to the proof of the result M6 stated in the last section. If there are at least $2^{n-1}+1$ states in the unit cycle C_1 of any n-SA, then a cross-edge must exist there, this follows by a simple state counting argument. But this is true, in particular, for SAs S where the characteristic polynomial p_S is primitive. In this case, p_S is a divisor of some trinomial (cf. P1' above), and it can be shown that this does also hold for any power of p_S. A proof of this statement as well as a proof for the result M6a2 can be found in the book [PV], pp.150-152.
Let, finally, the state $s^* = (1\ 1\ \ldots\ 1)^T$ belong to the unit cycle C_1 in the autonomous graph A_S. We can suppose that S has some even number of feedback connections. Therefore , the 0-successor of s^* is the state $(1\ 1\ \ldots\ 1\ 0)^T$, and, consequently, s^* is its own 1-successor. But this implies that there is a loop on C_1 caused by the 1-edge (s^*, s^*) in G_S.

For any integer $k \geq 2$, cross-edges on the level k may be loops as well as 'ordinary' edges (cf. the example above), and they can occur on different locations inside the transition graph T_S. With respect to our odd-weight masking problem, however, the levels of cross-edges are much more interesting than their exact position (with respect to the breadth, for example). This can be expressed by the introduction of a fourth SA graph type which is some more reduced with respect of its size.

The **reduced transition graph** (RTG, for short) $\mathbf{R_S}$ of some SA S contains all the levels $L_0,.L_1,.L_2,...,L_r$ of the transition graph T_S, as its nodes. Edges of the RTG are all the ordered pairs (L_i, L_{i+1}) where i= 1,..., r–1, at first. Moreover, further edges may occur which are loops; a loop (L_i, L_i) must be included if and only if there is some cross-edge c in the transition graph T_S such that l(c) = i.

The general structure of any RTG can be described by some „chain" **1—X—X— . . . —X**, where **X= 1** must be interpreted by a node without loop, and **X= 0** is a node with a loop. Obviuosly, it holds R_{V_4} = **1—1—0** where the 4-SA V_4 is our example considered above.

In general, we obtain from the result G4 that the masking problem 'Pu' is obviously equivalent to the problem to find sufficient and necessary conditions of SAs such that the first (i.e. lowest-level !) loop is located exactly at the level k =(u–1)/2, in their RTG.
This means that the RTG R_S of such a signature analyzer S must have the following form

1—1— . . . —1—0—X— . . . —X.
└ exactly k ones ┘

In particular, we can now state a further description of the open problem P3 in terms of the reduced transition graphs.

P3''. When has the reduced transition graph R_S of any n-SA S the form
1—0—X— . . . —X ?

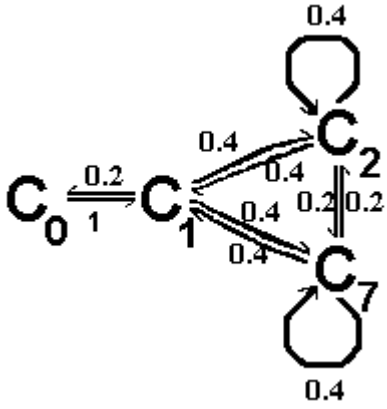
On the other hand, a more general problem caused by RTGs is the following question. Whichtypes of „binary patterns" can be in fact generated by signature analyzers as their reduced transition graphs? This problem is widely open; some first simulation results can be found in the conference paper [Vo].

4. Concluding remarks

1. The important problem of the computation or estimation of masking probabilities has only been mentioned some times in this report. An useful framework for such problems could be given by a further kind of graphs connected with SAs, the so called **weighted transition graph W_S**. This graph is simply the transition graph T_S introduced above but now again regarded to be a directed one, and, moreover, it contains an additional edge weight function.

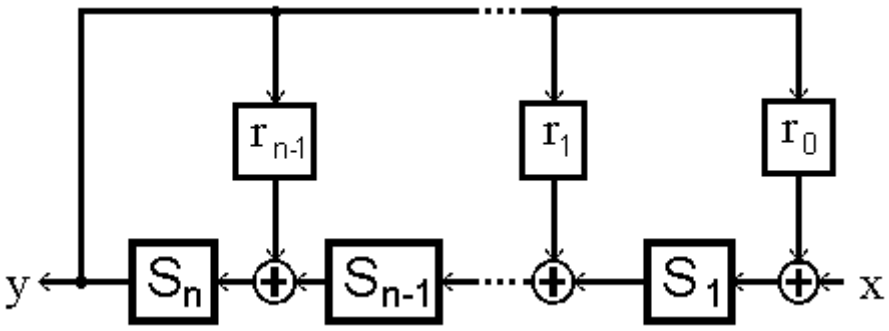
More precicely, if $e=(C_i, C_k)$ is some edge of T_S then its weight will be given by the fraction $w(e)=c/m$. In this fraction, the enumerator c is the number of all 1-edges leading from states which belong to the cycle C_i to such states belonging to the cycle C_k, while the denominator m is simply the total number of all states included in the cycle C_i which is the origin of the edge e.
The edge weight $w((C_i, C_k))$ will then be interpreted as the probability to reach the node C_k by one '1-step' which starts on the node C_i. The reference [HL] can be used as an example of an application of this concept to calculate masking probabilities.

To illustrate this notion, we include to the right the weighted transition graph of the 'complete' 4-SA V_4
As mentioned above, in comparison to the transition graph given in Section 3.3, all edges have been 'redrawn' with their original directions, and the edge weights are represented by decimal fractions.

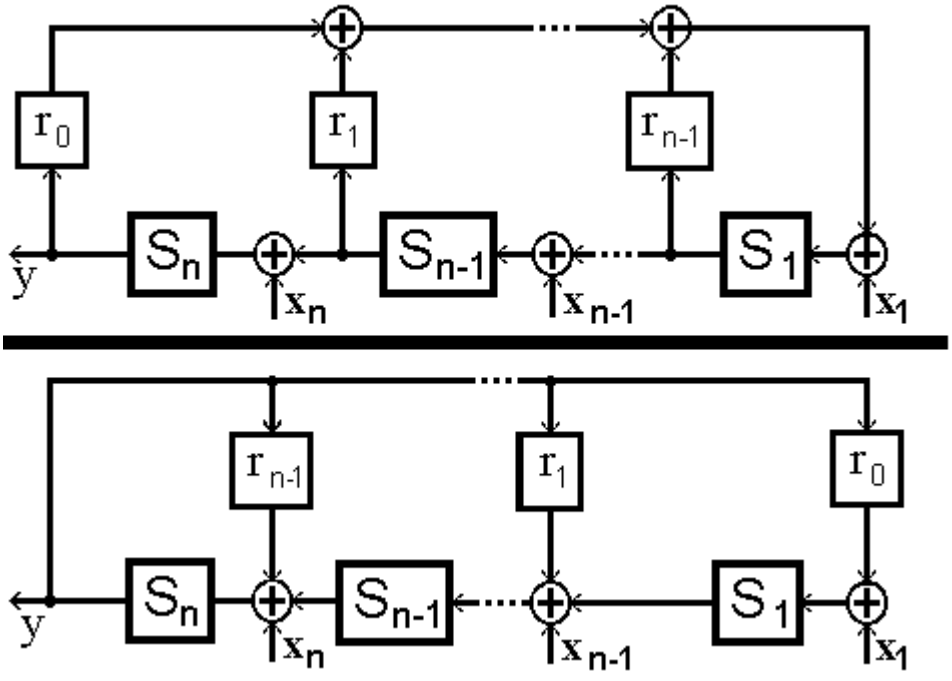


We remark that also (as the definitely last graph typeconsidered in this report) a 'weighted reduced transition graph' may be introduced which is a simple combination of the reduced and the weighted transition graph, respectively.

2. In this report, we restricted our attention only to LFSRs but also other linear devices are commonly considered as generators of signature-like data compressions. The following linear automaton is well known as a division circuit, and it is applied especially in coding theory (cf. [PW], for example). On the other hand, there are also applications in signature analysis; some basic properties, and the relationsbetween these automata and LFSRs are studied in [VP]. Moreover, it should be mentioned that also (so-called „hybrid“) combinations of these types are considered in some references (e.g. [Hl], [WM]).



3. Finally, let us shortly consider two of the parallel versions of signature analyzers. Such devices have been introduced as a built-in self-test tool in [KMZ]. We already mentioned in Section 2 that such automata can be applied to perform a parallel data compression.
For „serial signature anlyzers“ the two types considered so far are completely equivalent with respect to their masking properties. There are considerable differences, however, between the kinds of parallel SAs shown below, already with respect to the masking of odd-weight error sequences (cf. [VP], p.176).



References

[BCA] Benowitz,N., Calhoun,D.F., Alderson,G.E., Bauer,J.E., Joeckel,C.T.: An Advanced Fault Isolation System for Digital Logic. IEEE Transactions on Comp. C-24/5 (1975), 489-497.

[BJ1] Brzozowski,J.A.,Jürgensen,H.:Applications of Automata and Languages to Testing. Dept. of Comp. Science, The Unyversity of Western Ontario, Report No.420 (1994)

[BJ2] Brzozowski,J.A.,Jürgensen,H.: A Model for Sequential Machine Testing and Diagnosis. Journal of Electronic Testing 3(1992),219-234.

[C] Carter,W.C.: The Ubiquitous Parity Bit. FTCS-12(1982),289- 296.

[D] David,R.: Feedback Shift Register Testing. FTCS-8 (1978), 103-107

[F] Frohwerk,R.A.: Signature Analysis: A New Digital Field Service Technique. HP Journal 28/9(1977),2-8.

[Gi] Gill,A.: Linear Sequential Circuits. New York 1966.
[GN] Gordon,G.,Nadig,H.: Hexadecimal Signatures Identify Troublespots in Micro processor Systems. Electronics 50/5(1977),89-96.
[Gö] Gössel,M.: Bemerkung über die Existenz von Signaturregistern zur Erkennung geradzahliger Fehler. Elektron. Rechenanlagen 25 (1983), p.233.
[He] Heckmaier,J.H.: Mathematische Beschreibung und Klassifikation von Signatur registern. Diplomarbeit, München 1982.
[HL] Heckmaier,J.H.,Leisengang,D.: Fehlererkennung mit Signaturanalyse. Elektron. Rechenanlagen 25(1983),109-116.
[Hl2] Hlawiczka,A.: Hybrid Design of Parallel Signature Analyzers. 1.Europ.Test Conf. 1989, S.354-360.
[IS] Ibarra,O.,Sahni,S.K.: Polynomially Complete Fault Detection Problems. IEEE Transaction on Computers C-24(1975),242-249.
[L] Leisengang,D.: Klassifikation und Einsatz von Signaturregistern zur Fehlererkennung in digitalen Schaltungen. Dissertation, München 1983.
[KMZ] Built-In Logic Block Observation Technique. IEEE Test Conference 1979, 37-41.
[MC1] McCluskey,E.J.: Built-In Self-Test Techniques. IEEE Design& Test 2(1985), 21-28.
[MC2] McCluskey,E.J.: Built-In Self-Test Structures. IEEE Design& Test 2(1985), 29-36.
[Pl] Pliquett,J.: Mathematische Eigenschaften von Signaturregistern und Aspekte ihres Einsatzes in der Fehlerdiagnose digitaler Schaltungen. Dissertation, Greifswald 1989.
[PW] Peterson,W.W.,Weldon,E.J.: Error-Correcting Codes. MIT Press, Cambridge 1972.
[R] Reghbati,H.K.: Tutorial: VLSI Testing & Validation Techniques. IEEE Computer Society Press, 1985
[S] Smith,J.E.: Measures of the Effectiveness of Fault Signature Analysis. IEEE Tr.on Comp.C-29(1980),510-514.
[Vo] Voelkel,L.: On the Problem of masking Special Errors by Serial Signature Analyzers. 5th Workshop on New Trends in Testing, Ottawa, July/August 1991.
[VP] Voelkel,L.,Pliquett,J.: Signaturanalyse. Berlin (Akademie- Verlag und Springer-Verlag) 1988.
[WP] Williams,T.W.,Parker,K.P.: Design for Testability – A Survey. Proceedings of the IEEE, January 1983,98-112
[WM] Wang,L.T.,McCluskey,E.J.: Hybrid Designs Generating Maximum-Length Sequen ces. IEEE Trans.CAD-7(1988)1, S.91-99.
[YL] Yannakakis,M.,Lee,D.: Testing Finite State Machines. 23rd Annual ACM Symposium on Theory of Computing (1991),476-485.