IMTC 2004 – Instrumentation and Measurement
Technology Conference
Como, Italy, 18-20 May 2004

On Test Data Compaction using Linear Cycle Registers

Serge Demidenko[1], Eugene Levine[2], Vincenzo Piuri[3], Gourab Sen Gupta[4,5]
[1]School of Engineering & Science, Monash University, Malaysia Campus
2 Jalan Kolej, 46150 Petaling Jaya, Selangor, Malaysia
Phone: +60-3-56360600, Fax: +60-3-56329314
Email:Serge.Demidenko@engsci.monash.edu.my
[2]Rainbow Technologies Inc., Minsk, 220026, Minsk, Belarus
Phone: +375-17-249-8276, Fax: +375-17-248-8812
Email: lev@rainbow.by
[3]Department of Information Technologies, University of Milan
Via Bramante 65, 26013 Crema (CR), Italy
Phone: +39-02-503-30066, Fax: +39-02-503-30010
Email: piuri@dti.unimi.it
[4]Institute of Information Sciences & Technology, Massey University
Private bag 11222, Palmerston North, New Zealand
Phone: +64-6-350 5799, Fax: +64-6-350-2259
[5]School of EEE, Singapore Polytechnic, 500 Dover Road, Singapore
Email: G.SenGupta@massey.ac.nz

Abstract *– Compact testing (testing where data compression is employed to reduce a size of test response sequences and reference data) is widely used in the modern automated test instrumentation systems aimed at digital and mixed-signal devices. Cycle shift registers are among the most effective data compactors. They provide low hardware overhead, high operation speed and good fault coverage. The aim of this paper is to investigate single-input and multiple-input cycle registers and to derive the most general and non-recurrent analytical description of their operation. The description is then deployed for test reference data calculation. And it can also be used as a tool for error diagnosis. Another goal of the paper is to develop and research an alternative equivalent architecture for a multiple-channel cycle register allowing to produce detail error coverage analysis of this data compactor for an arbitrary bit-error multiplicity and the error configuration.*

Keywords *– Electronic Testing, Data Compression, Cycle Register, Fault Coverage*

I. INTRODUCTION

Electronic testing is an important area of the modern instrumentation and measurement technology. In the most general sense the testing includes generation and application of some special test signals onto inputs of an electronic circuit or system (often called *Device-Under-Test* or *DUT*), acquisition of DUT test responses, processing and comparing them with the reference values corresponding to the fault free operation.

Linear Cycle Registers (LCR) are simple regular-architecture circuits that for long time have been widely employed as efficient test response compactors in the area of digital testing as well as in fault-tolerant design of micro-programmed control units [1, 2].

In a wide sense LCR can be considered as a special case of a wider class of *Linear Feedback Shift Registers* (LFSR) where the feedback structure is greatly simplifies and consists only of one feedback line from the last memory element of the shift register [3-4]. Thus it would be generally possible to use the results on analytical description of operation of LFSR and evaluation of its fault coverage when employing LCR [5-6]. However in such a case the results would be approximate and they often would contain redundant information.

The aim of this paper is to obtain an analytical description of the signature compression by means of LCR, to derive equivalent architectures for the register, and to use them to research the fault coverage provided by the registers.

Two types of LCRs will be discussed in this paper: *Single-Input Cycle Register* (SICR) and *Multiple-Input Cycle Register* (MICR).

We will start with an analytical description of SICR operation (Section II) and that of MICR (Section III). We will show then their application for the test reference value calculations (Section IV). Finally we will proceed to the MICR alternative architectures (Section V) and their fault coverage evaluation (Section VI).

II. SINGLE-INPUT CYCLE REGISTER

Figure 1 below shows the Single-Input Cycle Register.

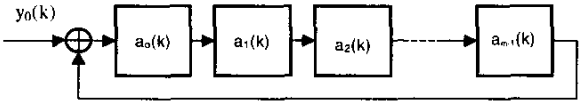

Figure 1. Single-Input Cycle Register

0-7803-8248-X/04/$17.00 ©2004 IEEE

The operation of SICR can be presented by the following expressions:

$$\begin{cases} a_0(k) = y_0(k) \oplus a_{m-1}(k-1), \\ a_i(k) = a_{i-1}(k-1) \end{cases} \quad (1)$$

where $i = \overline{1, m-1}$; $k = 1,2,3,...$; $a_i(k) \in \{0,1,\}$ - the state of the i-th bit of the register on the k-th cycle of the operation; $y_0(k) \in \{0,1\}$- the input signal on the k-th cycle; m – length (number of bits of SICR) and $\oplus$ - *mod 2* addition.

By using (1) it is possible to get an analytical expression for the bits of SIRC on an arbitrary cycle of its operation with respect to the initial state of the register

$A_0=\{a_0(0), a_1(0), a_2(0), ..., a_{m-1}(0)\}$

and its input sequence

$Y = \{y_0(k) | k = 1,2,3,...\}$.

From the second expression of (1)

$a_i(k) = a_0(k-i)$

By substituting the right part of the above in accordance with the first equation of (1)

$a_i(k) = y_0(k-i) \oplus a_{m-1}(k-i-1) = y_0(k-i) \oplus a_0(k-i-m)$

And by applying this procedure consequently the expression for $a_i(k)$ can be presented as

$$a_i(k) = a_i(k_0) \oplus \sum_{q=0}^{p-1} y_0(k - qm - i), \quad (2)$$

where $k_0 = k(\mathrm{mod}\, m)$, $p = (k - k_o)/m$, and here and further $\sum$ - is the *mod 2* summing.

Finally the expressions for the bit values of LCR on an arbitrary k-th operation cycle can be presented as follows:

$$\begin{cases} a_i(k) = a_{m-k_0+i}(0) \oplus \sum_{q=0}^{p} y_0(k - qm - i), \\ \qquad \forall i = \overline{0, k_0}, \\ a_i(k) = a_{i-k_0}(0) \oplus \sum_{q=0}^{p-1} y_0(k - qm - i), \\ \qquad \forall i = \overline{k_0 + 1, m-1}. \end{cases} \quad (3)$$

III. MULTIPLE-INPUT CYCLE REGISTER

Figure 2 shows a Multiple-Input Cycle Register used for test data compaction in digital testing of multi-output circuits.

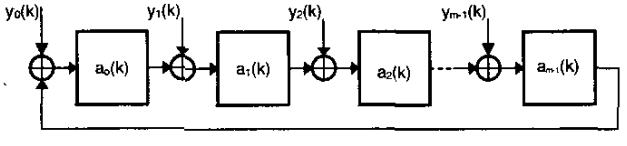

Figure 2. Multiple-Input Cycle Register

Operation of MICR can be described by the following system of expressions:

$$\begin{cases} a_0(k) = y_0(k) \oplus a_{m-1}(k-1), \\ a_i(k) = y_i(k) \oplus a_{i-1}(k-1) \\ \qquad \forall i = \overline{1, m-1};\ k = 1,2,3,... \end{cases} \quad (4)$$

where $y_i(k)$ is a bit value at the i-th input on the k-th operation cycle.

By sequentially converting the second summand (i.e., $a_{m-1}(k-1)$) in the right part of the first equation of (4) in accordance with the second equation of the system, the expression for the first bit of MICR can be rewritten to the following:

$$a_0(k) = \begin{cases} a_{m-i}(0) \oplus y_0(1); \quad k = 1, \\ a_{m-k}(0) \oplus \sum_{i=1}^{k-1} y_{m-1}(k-i) \oplus y_0(k), \\ \qquad \forall k = 2,3,4,...,m. \end{cases} \quad (5)$$

For $k>m$

(i.e., $k = pm + k_0$; $p \in \{1,2,3,...\}$; $k_0 \in \{0,1,2,...,m-1\}$)

this leads to

$$\begin{aligned} a_0(k) &= a_{m-1}(k-1) \oplus y_0(k) = \\ &a_0(k-m) \oplus \sum_{j=1}^{m-1} y_j(k-(m-j)) \oplus y_0 = \\ &a_{m-1}(k-m-1) \oplus y_0(k-m) \oplus \\ &\sum_{j=1}^{m-1} y_j(k-(m-j)) \oplus y_0(k) = \\ &a_0(k-2m) \oplus \sum_{j=1}^{m-1} y_j(k-(2m-j)) \oplus y_0(k-m) \oplus \\ &\sum_{j=1}^{m-1} y_j(k-(m-j)) \oplus y_0(k) = ... = \\ &a_0(k_0) \oplus \sum_{i=1}^{p} \sum_{j=1}^{m-1} y_i(k-(im-j)) \oplus \sum_{i=1}^{p} y_0(k-(i-1)m). \end{aligned} \quad (6)$$

For $k_0=0$ and $k_0=1$ the first member (i.e., $a_0(k_0)$) of the final expression (6) can be presented as

$$a_0(k_0) = \begin{cases} a_0(0) \ \textit{for}\ k_0 = 0, \\ a_{m-1}(0) \oplus y_0(1), \ \textit{for}\ k_0 = 1, \end{cases} \quad (7)$$

while for $k_0>1$ the expression for $a_0(k_0)$ is as follows:

$$\begin{aligned} a_0(k_0) &= a_{m-k_0}(0) \oplus \sum_{i=1}^{k_0-1} y_{m-i}(k_0 - i) \oplus y_0(k_0) = \\ &a_{(p+1)m-k}(0) \oplus \sum_{j=(p+1)m-k+1}^{m-1} y_j(k-(p+1)m+j) \oplus y_0(k-pm) \end{aligned} \quad (8)$$

Finally by combining (5)-(8) the expression for $a_0(k)$ can be presented in the following way:

$$a_0(k)=\begin{cases} a_{m-1}(0)\oplus y_0(1)\ \ for\ k=1,\\ a_{m-k}(0)\oplus\sum_{i=1}^{k-1} y_{m-i}(k-i)\oplus y_0(k),\\ \qquad\qquad \forall k\in\{2,3,4,\ldots,m\},\\ a_0(0)\oplus\sum_{i=1}^{p}\sum_{j=1}^{m-1} y_j(k-(im-j))\oplus\\ \qquad \sum_{l=1}^{p} y_0(k-(l-1)m)\ \ \forall k>m, k_0=0,\\ a_{m-1}(0)\oplus y_0(1)\oplus\sum_{i=1}^{p}\sum_{j=1}^{m-1} y_j(k-(im-j))\oplus\\ \qquad \sum_{l=1}^{p} y_0(k-(l-1)m)\quad \forall k>m,\ k_0=1,\\ a_{m-k_0}(0)\oplus\sum_{f=m-k_0+1}^{m-1} y_f(k-(p+1)m+f)\oplus\\ \sum_{l=1}^{p} y_0(k-lm)\oplus\sum_{i=1}^{p}\sum_{j=1}^{m-1} y_j(k-(im-j)),\\ \qquad \forall k>m,\ k_0=\{2,3,4,\ldots,m-1\} \end{cases} \quad (9)$$

In order to get an expression for $a_i(k)$ the variable $a_{i-1}(k-1)$ in the right part of the second equation of (4) is to be sequentially transformed:

$$a_i(k)=\begin{cases} a_{i-k}(0)\oplus\sum_{j=0}^{k-1} y_{i-j}(k-j)\ \forall k\ge i,\\ a_0(k-i)\oplus\sum_{j=1}^{i} y_j(k-(i-j)),\ \forall k<i. \end{cases} \quad (10)$$

Thus taking into account (9) the expression for $a_0(k-i)$ where $k>i$ can be written as:

$$a_0(k-i)=\begin{cases} a_{m-1}(0)\oplus y_0(1)\,\forall k=i+1,\\ a_{m-k+i}(0)\oplus\sum_{q=1}^{k-i-1} y_{m-q}(k-i-q)\oplus\\ \qquad y_0(k-i)\ \forall(k-i)\in\{2,3,4,\ldots,m\} \end{cases} \quad (11)$$

Hence

$$a_i(k)=\begin{cases} a_{m-1}(0)\oplus y_0(1)\oplus\sum_{j=i}^{i} y_j(k-(i-j)),\\ \forall i\in\{1,2,\ldots,m-1\},\ k=i+1,\\ a_{m-k+i}(0)\oplus\sum_{q=1}^{k-i-1} y_{m-q}(k-i-q)\oplus\\ \qquad y_0(k-i)\oplus\sum_{j=1}^{i} y_j(k-(i-j))\\ \forall k\in\{i+2,i+3,\ldots,i+m\}, i\in\{1,2,\ldots,m-1\}. \end{cases} \quad (12)$$

To get an expression for $a_i(k)$ for $(k-i)>m$ additional manipulations are required.

Taking into account that $k>m$ it can be presented as $k=pm+k_0$, where $p\in\{1,2,3,\ldots\}$, $k_0\in\{0,1,2,\ldots,m-1\}$. Then by taking into account the second equation of (9) the following can be written:

$$a_0(k-i)=a_0(0)\oplus\sum_{q=1}^{p}\sum_{j=1}^{m-1} y_j(k-i-(qm-j))\oplus \sum_{f=1}^{p} y_0(k-i-(f-1)m)\ \forall k_0=i \quad (13)$$

By combining (13) and (10)

$$a_i(k)=a_0(0)\oplus\sum_{q=1}^{p}\sum_{j=1}^{m-1} y_j(k-qm-(i-j))\oplus \sum_{f=0}^{p} y_0(k-fm-i)\oplus\sum_{l=1}^{i} y_l(k-(i-l))\ \forall k_0=i \quad (14)$$

and by taking into account the fourth equation of (9)

$$a_0(k-i)=a_{m-1}(0)\oplus y_0(1)\oplus \sum_{q=1}^{p}\sum_{j=1}^{m-1} y_j(k-i-(qm-j))\oplus\sum_{f=1}^{p} y_0(k-i-(f-1)m) \quad (15)$$

where

$k\in\{i+m+1,i+m+2,i+m+3,\ldots\}$, $i\in\{1,2,3,\ldots,m-1\}$.

Thus the second expression of (10) can be written as follows:

$$a_i(k)=a_{m-1}(0)\oplus\sum_{q=1}^{p}\sum_{j=1}^{m-1} y_j(k-i-(qm-j))\oplus \sum_{f=1}^{p} y_0(k-i-(f-1)m)\oplus\sum_{l=1}^{i} y_l(k-(i-l)) \quad (16)$$

where

$k\in\{i+m+1,i+2m+1,i+3m+1,\ldots\}$, $i\in\{1,2,3,\ldots,m-1\}$,

$k_0=i+1$.

If $k_0>I$, then it can be written that $(k-i)-pm=(k-pm)-i=k_0-i$.

And by employing (6) an expression for $a_0(k-i)$ can be presented in the following way:

$$a_0(k-i)=a_0(k_0-i)\oplus\sum_{q=1}^{p}\sum_{j=1}^{m-1} y_j(k-i-(qm-j))\oplus y_0(k-i-(q-1)m). \quad (17)$$

In accordance with (8) and by taking into account that $(p+1)m-k=m-k_0$, the first member in the right part of (13) can be written as

$$a_0(k_0-i)=a_{m-(k_0-i)}(0)\oplus y_0(k_0-i)\oplus\sum_{j=m-(k_0-i)+1}^{m-1} y_j(k-(p+1)m-(i-j)). \quad (18)$$

And after applying (18) to (17) and incorporating the obtained result into (10):

$$a_i(k) = a_{m-(k_0-i)}(0) \oplus \sum_{f=1}^{p} y_0(k - fm - i) \oplus$$
$$\sum_{q=0}^{p-1} \sum_{f=1}^{m-1} y_i(k-(q+1)m-(i-j)) \oplus \quad (19)$$
$$\sum_{l=m-(k_0-i)+1}^{m-1} y_l(k-(p+1)m-(i-l)) \oplus \sum_{s=1}^{i} y_s(k-(i-s)),$$

where $i \in \{1,2,3,\ldots, k_0-2\}$, $k \in \{i+m+1, i+m+2, \ldots\}$.

By using the same approach the expressions for $a_i(k)$ can be derived for other ranges of i and k:

$$a_i(k) = a_{m-1}(0) \oplus y_0(1) \oplus \sum_{q=1}^{p-1} \sum_{j=1}^{m-1} y_j(k-i-(qm-j)) \oplus$$
$$\oplus \sum_{i=1}^{p-1} y_0(k-i-(l-1)m) \oplus \sum_{s=1}^{i} y_s(k-(i-s)), \quad (20)$$
$$\forall k-i > m, \text{ and } k_0 = 0,$$

and

$$a_i(k) = a_{i-k_0}(0) \oplus \sum_{q=0}^{p-1} y_0(k-qm-i) \oplus$$
$$\sum_{f=0}^{p-2} \sum_{j=1}^{m-1} y_j(k-i-(qm-j)) \oplus$$
$$\sum_{l=1}^{p-1} y_0(k-i-(l-1)m) \oplus \sum_{s=1}^{i} y_s(k-(i-s)), \quad (21)$$
$$\forall (k-i) > m, \{k_0+1, k_0+2, k_0+3, \ldots, m-1\},$$
$$k_0 \in \{1,2,3,\ldots, m-1\}.$$

IV. TEST REFERENCE VALUES

The presented above analytical expressions (10), (12), (14), (16), (19-21) cover all the possible ranges for i and k and can be directly employed to compute test reference values for SICRs and MICRs used to compress arbitrary test response sequences.

For MICR this can be implemented as a set of the following steps:

1. For a given length of the cycle register m and a number of cycles k, values of p and k_0 are obtained (taking into account that $k = pm+k_0$);
2. By using (9), the value of $a_0(k)$ is found for the given m, p, k_0 and input sequences $y_i(j)$, $i = \overline{0, m-1}$, $j = \overline{0, k}$.
3. By using (10), (12), (14), (16) or (19-21) values $a_i(k)$, $i = \overline{1, m-1}$ are found for a given input sequences $\{y_i(j)\}$, $i = \overline{0, m-1}$, $j = \overline{0, k}$, initial state of the cycle register $a_i(0)$, $i = \overline{0, m-1}$ and values of i, m, k, k_0 and p.

For the case of SICR the calculations are significantly simplified. They include just two steps:

1. For a given length of the cycle register m and a number of cycles k, values of p and k_0 are found (taking into account that $k=pm+k_0$);
2. By using (3) values of the bits of SICR $a_i(k)$, $i = \overline{0, m-1}$ are then found.

V. EQUIVALENT ARCHITECTURES

The analytical expressions for MICR presented in the section IV can be deployed to derive an equivalent architecture of the register. As an example 4-bit MICR is considered. The initial architecture of the register corresponds to that presented in Fig. 2.

The equivalent architecture of the 4-bit MICR can be presented as a set of four SICR's with all 0 initial state and CR with an initial state corresponding to that of the original register as shown in Fig. 3.

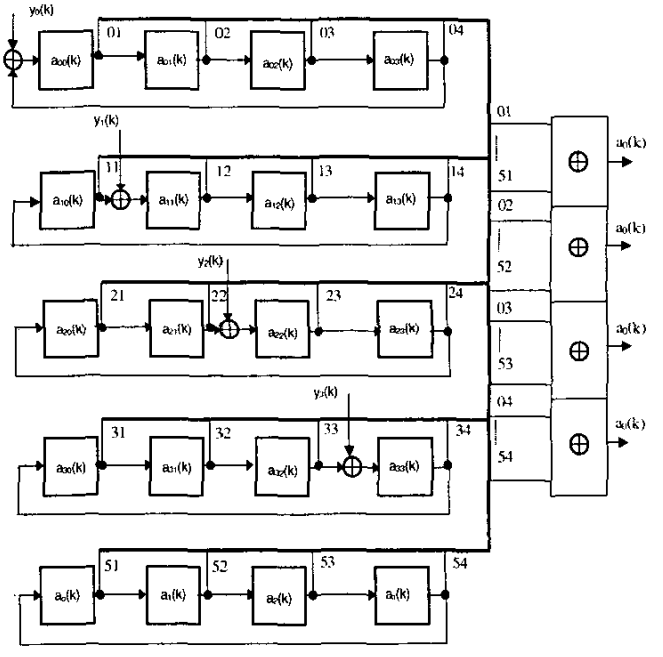

Figure 3. Alternative architecture of MICR

This architecture can be simplified thus leading to the following one that combines SICR and a *mod 2* adder with delayed inputs (Fig. 4).

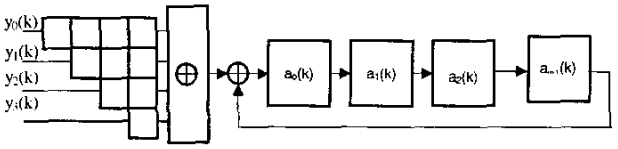

Figure 4. Simplified architecture of 4-input MICR

The state of the shift register of the simplified architecture is equal to that of traditional MICR if m all-zero vectors are applied to the primary inputs of the CR after entering the test data being compressed.

It can be seen that the architecture presented in Fig. 4 performs actually double data compression. It includes compression "in space" (CiS) that is performed by the *mod 2* adder and compression "in time" (CiT) realized by means of SICR.

In turn this implies that the error masking can occur either in CiS or in CiT or in both. And since the output of CiS circuit is connected to the input of CiT, the error detection probability evaluation has to be done for CiS and CiT, and the conditional probability has to be used to evaluate the total error detection provided by MICR.

VI. FAULT COVERAGE EVALUATION

In order to simplify the error detection analysis provided by MICR a basic CiS-CiT architecture is considered (Fig. 5).

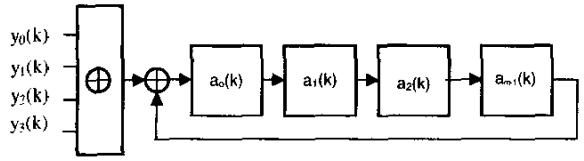

Figure 5. Basic equivalent architecture of MICR

A. Error masking evaluation by CiT circuit (SICR)

Error masking by SICR has been evaluated in the literature since rather long time ago (e.g., [7]). However the absolute majority of the research publications on the topic have been based on an assumption that all the bits of CR can take value of *0* or *1* with equal probability of 50%. By making such an assumption it has been shown that the probability of the error masking by m-bit SICR can be estimated as

$$P_m \approx \frac{1}{2^m}$$

However the above value is not based on analysis of possible error configurations on the input of CR circuit. Thus it gives just a very rough and averaged estimation that has got just a limited practical usefulness.

A substantially more accurate evaluation could be done if a number of all possible configurations of non-detected erroneous input sequences are found. And then it would be related to the total number of all possible input sequences of the given length.

In order to simplify the analysis it is assumed that an initial state of all the bits of CR is *0*. Using the approach proposed in [5] the analysis is done for the bits of CR from *0*-th to k_0-th, where $k_0 = k(\mathrm{mod}\, m)$.

The condition for the error masking (non-detection) can be written as

$$\sum_{q=1}^{p} e(k - qm - i) = 0 \qquad (22)$$

where $\{e(k)\}$ – error bit sequence and $p = (k - k_0)/m$.

It can be concluded from the expression (22) that SICR detects all single-bit errors.

For a double bit error there are two non-zero members in the sequence $\{e(k)\}$. They correspond to some cycles k_1 and k_2 of SICR operation ($k_1 > k_2$). It can be shown that

$$k_1 - k_2 = cm, \quad c \in \overline{1, p-1} \qquad (23)$$

Thus any undetectable by SIRC error sequence can be presented as mod 2 sum of elementary sequences that contain two non-zero bits satisfying the condition (23).

In order to evaluate a fault coverage of SIRC it is necessary to find a number of masked errors of multiplicity of r-bit denoted $N(r)$, and then to find the probability of error masking $P_E(r)$ as $P_E(r) = N(r)/T(r)$, where $T(r)$ is the total number of possible errors of multiplicity r.

It has been shown in [5] that for an arbitrary cycle $k = pm + p_0$ it is possible to find p portions of the sequence being compressed that satisfy the condition $k(\mathrm{mod}\, m) \le k_0$, as well as $(p-1)$ fragments of the sequence satisfying the condition $k(\mathrm{mod}\, m) > k_0$.

By analyzing all possible pairs of elements of a sequence being compressed satisfying the condition (23) the number of masked 2-bit errors is equal to

$$N(2) = k_0 C_p^2 + (m - k_0) C_{p-1}^2 .$$

The total number of possible errors of multiplicity 2 is equal to

$$T(2) = C_k^2 = \frac{k(k-1)}{2} .$$

Thus the probability of error masking is

$$P_E(2) = 2\frac{k_0 C_p^2 + (m - k_0) C_{p-1}^2}{k(k-1)} \qquad (24)$$

To simplify the derivation, let us assume that $k_0 = 0$, i.e., $k = pm$ (such an assumption does not affect the validity of the analysis). In this case the number of undetected errors is

$$N(2) = \frac{mp(p-1)}{2}$$

and the probability of the error masking can be presented as

$$P_E(2) = \frac{mp(p-1)}{k(k-1)} \approx m\left(\frac{p}{k}\right)^2 = \frac{1}{m} \qquad (25)$$

Following the same approach

$$P_E(4) \approx 3\frac{1}{m^2}\frac{m-1}{m} \tag{26}$$

$$P_E(6) \approx 3\cdot 5\frac{1}{m^3}\left(\frac{m-1}{m}\right)^2 \tag{27}$$

$$P_E(8) \approx 3\cdot 5\cdot 7\frac{1}{m^4}\left(\frac{m-1}{m}\right)^2\frac{m-2}{m} \tag{28}$$

$$P_E(10) \approx 3\cdot 5\cdot 7\cdot 9\frac{1}{m^5}\left(\frac{m-1}{m}\right)^2\frac{m-2}{m}\frac{m-3}{m} \tag{29}$$

$$P_E(12) \approx 3\cdot 5\cdot 7\cdot 9\cdot 11\frac{1}{m^6}\left(\frac{m-1}{m}\right)^2\frac{m-2}{m}\frac{m-3}{m}\frac{m-4}{m} \tag{30}$$

By analyzing the expressions (25)-(30), and by applying the mathematical induction method the general expression for the probability of masking for an arbitrary error multiplicity r (r is even number, and $r \le 2m$) can be written as follows

$$P_E(r) \approx \frac{1}{m^{\frac{r}{2}}}\frac{m-1}{m}\prod_{n=1}^{\frac{r}{2}}(2n-1)\frac{m-n+1}{m} \tag{31}$$

B. Probability of error detection by CiS

It is apparent that all 1-bit or odd-bit multiplicity errors will be detected by SiC with probability of 100%. And since they cause the change (inversion) of odd-number of bits on the CiS output sequence, these errors also will be detected by CiT with 100% probability.

In contrast, the even-bit multiplicity errors can be masked in CiS (as well as in CiT, as it was shown above). The other important feature mention is that CiS can change the error multiplicity. For example 3-bit error in the same input vector of CiS will be transferred into 1-bit error on the output of it. Thus all possible configurations of the errors on the input of CiS have to be considered. The probability of the errors occurrence and that of their detection have to be found in order to get the final error detection evaluation for a given error multiplicity.

By doing this for *2*-bit, *4*-bit, *6*-bit, *8*-bit, etc., error multiplicity and by employing the mathematical induction technique, the expression for the error detection by CiS can be presented as follows:

$$P_D(r) \approx \sum_{i=2}^{r}\frac{1}{\prod_{t=1}^{\frac{m-i}{2}}2t}\frac{r!}{i!}\left(\frac{m-1}{km}\right)^{\frac{r-i}{2}} \tag{32}$$

Finally by combining (31) and (32) the total error detection probability for MICR can be estimated by the following expression

$$P_{DT}(r) \approx \sum_{i=2}^{r}\frac{1}{\prod_{t=1}^{\frac{m-i}{2}}2t}\frac{r!}{i!}\left(\frac{m-1}{km}\right)^{\frac{r-i}{2}} \times \left(1-\frac{m-1}{m^{\frac{r}{2}+1}}\prod_{n=1}^{\frac{r}{2}}(2n-1)\frac{m-n+1}{m}\right), r = 2n,\ n = 1,2,3,... \tag{33}$$

The expression (33) is true for the even-bit errors. The odd-bit errors will always manifest themselves on the output of CiS as new errors that are also of odd-bit multiplicity type. In turn they will be detected by CiT with 100% probability. Thus

$$P_{DT}(r) = 1,\ \forall r = 2n-1,\ n = 1,2,3,... \tag{34}$$

VI. CONCLUSION

In this paper the general analytical expressions describing operation of SICR and MICR are obtained. They can be used for calculation of the test reference values (i.e., those corresponding to the error-free operation of the device being tested), as well as for the error diagnosis when the actual value obtained as a result of testing and the reference one are mismatched.

The equivalent architecture for MICR has been derived. The architecture shows that MICR can be presented by two interconnected circuits. The first one provides so-called compression in space. It combines several input lines into just one output by means of *mod2* addition. The second circuit is SICR that performs compression in space, i.e., it produces a short output word (often called *signature*) from a long bit-sequence coming to its input.

Finally by using the derived architecture the error coverage provided by MICR has been evaluated for an arbitrary error multiplicity and error configurations.

REFERENCES

[1] M. L. Bushnell, V. D. Agrawal. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Kluwer, 2000

[2] S. Demidenko, E. Levin, K. Lever, "Concurrent self-checking for microprogrammed control units: an analytical survey". IEE Proceedings-E, Vol. 138, No 6, pp. 377-388, Nov. 1991.

[3] P. H. Bardell, J. Savir, W. H. McAnney. *Built-in Test for VLSI: Pseudorandom Techniques: Pseudorandom Techniques*. Wiley, 1988

[4] V. N. Yarmolik, S. N. Demidenko. *Generation and Application of Pseudorandom Sequences for Random Testing*, Wiley, 1988

[5] V. N. Yarmolik. *Fault Diagnosis of Digital Circuits*. Wiley, 1990

[6] S. Demidenko, V. Piuri, "Signature compression: a generalised description and its applications", Journal of Microelectronics Systems Integration, Plenum Press, Vol. 1, N. 3, pp. 19-34, March 1995.

[7] R. David. "Testing by Feedback Shift register". IEEE Trans. on Computers, 1980, No 9, pp. 668-673.