Syndrome-Based Functional Delay Fault Location in Linear Digital Data-Flow Graphs
Abhijit Chatterjee and Manuel A. d'Abreu
GE Corporate R&D, P.O. Box 8, Schenectady, NY 12301

Abstract
Circuit speed, throughput, floating point operations per second are some of the buzz words of yesterday that are extremely critical to the survival of some of the high-tech electronic products of today. Yet, with the onset of new technology new problems have surfaced, such as those concerned with timing, that have generated new research areas. In a large system, from a timing problem in a component or interconnecting lead can cause the entire system to fail. Rather than replacing the entire system, it is logical to only replace the failed component or lead. However, the problem of determining which component or lead has failed is quite formidable. In this paper we show how timing problems identified during speed testing of a class of circuits widely used in digital signal processing and control can be isolated to individual or sets of circuit components. The fault location scheme exploits linearity properties of these circuits to identify failures.

Introduction
With the increased sophistication in VLSI design techniques, the need for operating synchronous sequential circuits at higher speeds has become critical. In digital signal processing applications, higher speed results in higher throughput of pipelined arithmetic computations. However, higher speed requirements have resulted in other problems resulting from the fact that the logic values at the outputs of the logic gates must settle down before they are clocked into the system registers. At high speeds, due to some gates (or paths in the circuit) being 'slower' than others, incorrect logic values are clocked into the system registers, resulting in what is called a delay fault. Testing for delay faults was studied by Hsieh et al in [1], by Malaiya and Narayanswamy in [2], by Smith in [3], and by Lin and Reddy in [4]. Automatic test generation algorithms for delay faults were developed by Reddy et al in [5], by Schulz et al in [6], and by Lesser and Shedletsky in [7]. The problem of logic synthesis for delay fault testability has recently been addressed by Roy and Abraham in [8], by Pramanik and Reddy in [9], by Kundu and Reddy in [10] and by Devadas and Keutzer in [11].

While the above research is significant, the problem we address in this paper is that of functional delay fault location/diagnosis. A functional delay fault is said to occur when a circuit operator (such as an adder or multiplier implemented with synchronous sequential logic) generates an incorrect result or output value when clocked at the desired clock frequency at which it is designed to operate. The incorrect result could be due to excessive delays on the leads connecting the operators or due to a 'slow' operator itself. We desire, by performing certain experiments on the system, to determine those operators (and associated interconnects) that do not function properly at the intended clock speed.

The necessity for such a methodology arises from advanced packaging technologies [12] that allow unpackaged die to be placed on a substrate to be later interconnected with high density interconnection techniques. In this packaging technology the interconnect delays are insignificant when compared with operator or chip delays. In the GE DIODES system [13], families of die containing adders, multipliers, etc, are used to realize complex digital systems. High-speed testing of all the adders/multipliers on an unpackaged die is a difficult process. In fact in the current testing process only a functional test of the die is performed. Speed testing would be an even more difficult problem if these complex systems were mass-manufactured. On the other hand, the packaging technology allows the designer to replace individual die if the packaged substrate does not work. Clearly, in the event that the substrate (overall system of interconnected dies and associated circuitry) does not function at the desired speed (but passes the functional test) we need a mechanism to identify the source of the fault or at least localize the fault to a die and its associated interconnect.

The technique we use is that of syndrome based analysis. Syndrome testing has been used in a different context (detection of stuck-at faults) by Savir in [14]. Syndrome-based diagnosis by matching test responses to fault patterns of different components has been studied by Kime in [15]. In our approach, the syndrome consists of the sequence of error values observed at the output over a range of time. By using the mathematical properties of linear digital data flow graphs, fault location/diagnosis is performed.

Preliminaries
Our approach is based on a state variable representation of the digital circuit in which the current circuit state (or memory values) can be expressed recursively in terms of its past state. The digital state variable systems (SV-systems) that we consider are interconnections of adders, multipliers, shifters and registers. The generic model of the state variable systems is indicated in Figure-1. The box L
is a digital circuit that computes a linear transformation of the values of its \( n+m \) inputs and generates \( n+w \) outputs as shown. The processing is purely arithmetic and hence each data word represents a numerical quantity. In general, the state variable system is implemented with synchronous sequential logic. The word delays \( X_1, X_2, ..., X_n \) (the system states) are symbolic. In practice delay optimization is used to totally or partially eliminate these delays by making use of the latency of the various functions of \( L \) [16].

A digital state variable system with \( m \) inputs, \( w \) outputs and \( n \) states (delays in Figure-1), can be described by a set of system state equations and system output equations namely:

\[
X(t + 1)^T = AX(t)^T + Bu(t)^T \quad \text{and} \quad \quad \quad Y(t)^T = CX(t)^T + Du(t)^T
\]

where \( A, B, C, \) and \( D \) are \((n \times n), (n \times m), (w \times n)\) and \((w \times m)\) matrices of real numbers (called the state matrices), respectively. \( X(t) \) is the vector \([X_1(t), X_2(t), ..., X_n(t)]\) of state variables, \( u(t) \) is the vector \([u_1(t), u_2(t), ..., u_m(t)]\) and \( Y(t) \) is the vector \([Y_1(t), Y_2(t), ..., Y_w(t)]\).

Our fault model assumes that a single operator can have a functional delay fault; for brevity we will not discuss timing problems with operator interconnects (this entails a simple modification of the current analysis). In the state and output equations, the index \( t \) of \( X(t), Y(t), \) and \( u(t) \) is incremented when a set of data words at the inputs \( u_1, u_2, ..., u_m \) is processed by \( L \) and the results are available at \( Y_1, Y_2, ..., Y_w \) and \( o_{p_1}, o_{p_2}, ..., o_{p_n} \). The operators of \( L \) run on a clock \( c_k \) that completes a fixed number of cycles, \( r = 1, 2, ..., \gamma \) for each increment of the index \( t \). Consider that during speed testing, the circuit-under-test fails under a specified sequence of inputs \( u(-k), u(-k + 1), ..., u(0) \) and generates an incorrect output value at some time \( t \geq 1 \). We conduct a delay fault test experiment as follows. The circuit-under-test is fed the same sequence of inputs as during the speed test but with the clock \( c_k \) running at much slower speed. At an appropriate cycle, \( r = i \) in the time frame \( t = -j \) for some \( j \leq 0 \), a single clock pulse at the desired clock speed is fed into \( c_k \). Subsequently, the clock is returned to its original slow speed. This is shown in Figure-2, in which \( T > T' \). The values of \( t = -j \) and \( r = i \) are chosen so that the delay fault test experiment excites the same functional delay fault as the speed test. In order to do this, several delay fault test experiments are conducted and the details are beyond the scope of this paper. By knowing how the error introduced by the excited delay fault at \( t = -j + 1 \) is processed by the circuit at times \( t = -j + 2, -j + 3, ... \) and by making measurements of the errors in the circuit output values at these times, fault location is performed.

Let the circuits have \( P \) operators \( O_1, O_2, ..., O_P \). Let there be \( z \) paths, \( Path_m, 1 \leq m \leq z \), from the output of the operator \( O_j \) to the output \( o_{p_i} \) (i.e. to the state variable \( X_i(t) \)), for some \( 1 \leq i \leq n \). Each path passes through a set of operators such as adders, subtractors and multipliers. The gain of an adder/subtractor form its input to its output is defined to be \( +/- 1 \). The gain of a constant multiplier is defined to be the multiplier value itself. The composite gain \( g_m \) of the path \( Path_m \), for \( 1 \leq m \leq z \), is defined to be the product of the gains associated with all operators in that path. For example the gain of the path \( p_{p_1}, O_4, O_3, o_{p_2} \) in Figure-3 is 0.7. The total gain \( g_j \) from the output of operator \( O_j \) to the output \( o_{p_i} \) (state variable \( X_i(t) \)) is \( g_j = \sum_{m=1}^{z} \theta_m \). Thus the gain from the output of \( O_1 \) to \( o_{p_2} \) (in Figure-3) is given by (gain of path \( O_1, O_4, O_3, o_{p_2} \) plus the gain of path \( O_1, O_3, o_{p_2} \)) \( 1 + 1 = 2 \).

Definition: The gain vector \( G_j \) for operator \( O_j \) is defined to be the vector \([g_{j_1}, g_{j_2}, g_{j_3}, ..., g_{j_m}]\).
Fault Location by Precomputed Syndromes

Let $X_t$ be the state vector at time $t$. Under fault-free conditions, $X_{t+1} = AX_t + Bu$. For simplicity, we assume that $i=0$ in Figure 2. At time $t=0$, the incorrect values of the state variable $X_t$ in the last column of matrix $A$ are replaced by $0$. If the operator $O_i$ is faulty and its output value has an error of magnitude $e_i$, the output is given by

$$X_t^f = AX_t + Bu + G_i^f e_i$$

However, at the next clock pulse, since the delay fault no longer exists,

$$X_{t+1}^f = AX_{t+1} + Bu + G_i^f e_i + Y_t$$

Substituting for $X_1$ in the expression for $X_2$, we get

$$X_2^f = A^2X_1 + ABu + AG_i^f e_1 + Bu + G_i^f e_2$$

Expanding the state equations recursively, we get

$$X_k^f = A^kX_1 + A^{k-1}Bu + A^{k-2}Bu + ... + A^1Bu + G_i^f e_k$$

Notice that the fault-free value $X_k$ can be expressed as

$$X_k^f = A^kX_1 + A^{k-1}Bu + A^{k-2}Bu + ... + A^1Bu + G_i^f e_k$$

Therefore, $X_k^f$ is $X_k = AX_1^f + Bu + G_i^f e_{k-1}$, and $Y_k = CX_k^f + Bu = CF_k^f + CA_i^f G_i^f e_{k-1} + Bu$.

If $e_i$ is the error in $Y_k$, then $e_k = Y_k - CF_k^f - Bu = e_k - CA_i^f G_i^f e_{k-1}$.

For a predefined syndrome length $i$, the syndrome $S_i$ for operator $O_i$ is defined to be the sequence

$$S_i = [CA_i^f G_i^f, CA_i^f G_i^f, ..., CA_i^f G_i^f]$$

Different operators, in general, have different syndromes and operators with the same syndrome are said to be in the same ambiguity class. By observing the error magnitude of the output $Y(t)$ at time $t=0, 1, 2, ..., l-1$, the observed error vector $OEV = [e_0, e_1, e_2, ..., e_{l-1}]$ is constructed. Under the single delay fault assumption, the $OEV$ must be $K \cdot S_i$, for some $1 \leq i \leq P$. By identifying $S_i$ from the precomputed syndromes, the faulty operator or ambiguity class of operators is determined. The die containing this operator or the set of operators is then replaced to allow operation of the circuit at the desired clock speed.

Example 1:

For the digital state variable system of Figure 3,

$$
\begin{bmatrix}
X_1(t+1) \\
X_2(t+1) \\
X_3(t+1)
\end{bmatrix} =
\begin{bmatrix}
0.7 & 1 & 0 \\
1.4 & 1 & 1 \\
0.7 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X_1(t) \\
X_2(t) \\
X_3(t)
\end{bmatrix} +
\begin{bmatrix}
0 \\
1 \\
1
\end{bmatrix}
u(t)
$$

and

$$Y(t) = [0 \ 1 \ 0]
\begin{bmatrix}
X_1(t) \\
X_2(t) \\
X_3(t)
\end{bmatrix}
$$

The gain vectors for the five operators are as follows:

$$G_1 = [1 \ 1 \ 0],
G_2 = [0 \ 0 \ 1],
G_3 = [0 \ 1 \ 1],
G_4 = [1 \ 2 \ 1],
G_5 = [0 \ 1 \ 1]$$

The error syndromes for $l=3$ are as follows:

$$S_1 = [1 \ 2.4 \ 5.38],
S_2 = [1 \ 2.4 \ 4.4],
S_3 = [1 \ 2.4 \ 9.98]$$

Consider a delay fault test experiment in which the error in the output $Y(t)$ at times $i=1, 2, 3$ are 0.1, 0.11 and 0.25 respectively. Since this most closely matches the syndrome $S_2$, the functional delay fault lies with the operator $O_2$.

Fault Location by Algebraic Method

There is another way of viewing the same problem. Let $c_1, c_2, ..., c_i-1$ be the observed errors in the output $Y(t)$ at times $i=1, 2, ..., i-1$ and let the erroneous state vector at time $t=1$ be $[X_1 + a_1, X_2 + a_2, ..., X_n + a_n]$ where $X_1, X_2, ..., X_n$ are the expected (fault-free) state variable values and $a_1, a_2, ..., a_n$ are the corresponding errors in them. Then

$$Y_1 = C[X_1 + a_1, X_2 + a_2, ..., X_n + a_n]^T = C[X_1, X_2, ..., X_n]^T + [a_1, a_2, ..., a_n]^T$$

Consequently, $e_i = C[a_1, a_2, ..., a_n]^T$.

We define $Z = [a_1, a_2, ..., a_n]$, so that

$$X_2^f = A(X_1^f + Z^T) + Bu_1$$

Hence $Y_2^f = CAX_1^f + CAZ^T + Du_1$, and therefore $e_2 = CAZ^T$.

By expanding the state equations recursively to time $t=k$, we can show that

$$e_k = CA^{k-1}Z^T$$

If there are $n$ state variables, then we can write $n$ linear simultaneous,

$$e_1 = CA^0Z^T,
\ldots,
e_n = CA^{n-1}Z^T$$

In this system of equations, $a_1, a_2, ..., a_n$, are known and the elements of the vector $Z = [a_1, a_2, ..., a_n]$ are the unknown variables. By solving this system of equations we obtain the vector $Z$. Under the single delay fault assumption, this vector must be the same as $G_i$ for some $1 \leq i \leq P$. Hence we can identify the operator or set of operators from which the delay fault could have originated.

Example 2:

The equations that we obtain are

$$e_1 = CA^0Z^T = a_1,
\ldots,
e_5 = CA^{4}Z^T = a_5.$$
Example 3:

The circuit of Figure-4 represents a second-order filter whose transfer function in the z-domain is 
\[(z^2a_0 + a_1z) + (z^2 - b_1z - b_2).\] The state and output equations are
\[
\begin{align*}
X_1(t+1) &= [0 \quad b_2] X_1(t) + [a_1] U(t) \\
X_2(t+1) &= [1 \quad b_1] X_2(t) + [a_0] U(t) \\
Y(t+1) &= [1 \quad b_1] X_1(t) + [a_0] U(t)
\end{align*}
\]
where \(a_0 = 0.17, \ a_1 = 0.35, \ b_1 = 0.97, \) and \(b_2 = -0.79.\)

The gain vectors for this filter are \(G_1 = [1, 0, 0], \ G_2 = [0, 1, 1], \ G_3 = [1, 0, 0], \)
\(G_4 = [0, 1, 1], \ G_4 = [0, 1, 1], \)
\(G_5 = [1, 0, 0], \ G_5 = [0, 1, 1].\)

This filter contains two ambiguity classes \(C_1 = \{O_1, O_2, O_1, O_2\}, \) and \(C_2 = \{O_1, O_2, O_1\},\) where all the operators in a class have the same error syndrome. The corresponding syndromes are \(S_1 = [0.97, 0.16]\) and \(S_2 = [1, 0.97].\)

Figure 4 Second-Order Filter

Practical Issues

In practice, the presence of overflow during computation can cause the OEV to be corrupted. In bit-serial and digit-serial systems this can be avoided by using only the least significant bit or digit for the delay experiment and setting the higher order bits of the word in such a way that overflow does not occur. There can also be errors due to finite precision arithmetic in error computation (such as illustrated in Example 2). These have to be taken into account when matching the OEV to the precomputed fault syndrome.

Conclusion

In this paper we have presented a novel approach to fault location in linear digital data flow graphs. The fault location scheme is simple and depends on the linearity property of these data flow graphs. Identification and replacement of the failed component allows operation of the circuit at the desired clock speed.

Bibliography