A State-of-the-art Approach to Develop Built-In-Test Diagnosis

Yoo, Wang-Jin Ph.D.
Assistant Professor
Department of Industrial Engineering
HAN NAM UNIVERSITY

初期段階의 製品 檢査方法은 單純히 製品이 제대로의 機能을 遂行 하는지 안하는지에 대해서만 檢査하고 決定을 내릴 뿐이었다. 대부분의 檢査란, 檢査者에게 最終的으로 製品이 引受된 후에 製品의 狀態與否를 限定的이고, 制限的으로만 判斷하여 檢査結果를 提供하여왔다. 現代의 製品特性이 構造的 복잡성의 增加, 製品不良 現狀把握의 難易度 增加 등으로 製品의 使用節次, 檢査, 維持保守, 部品交換 등이 점점 더 難解해지고, 檢査者의 個人的인 熟練度도 增加하게 되어, 製品을 維持保守하는데 많은 費用과 時間을 必要로하게 되었다. 이러한 問題들의 解決方案의 하나로 製品의 自體에 스스로 檢査할 수 있는 體系的인 시스템을 設置하게 되어 BIT(Built-In-Test) 가 誕生하게 되었다. BIT는 現存하는 製品뿐만아니라, 生産될 製品의 設計段階에서도 많이 應用되어 製品의 RAM(Reliability, Availability, Maintainability)에 많은 寄與를 해왔다. 이 PAPER는 지금까지 BIT가 주로 Military System에 適用되어 온것을 Commercial System으로의 變換을 위한 基礎作業을 提示하고 여러 代案을 列擧하였으며, BIT가 갖고 있는 問題點들을 把握하여, 向後 高度化 되가는 産業社會의 要求에 副應할 수 있는 土臺를 마련코자 Survey 하였다.

I. The State of Built-In Testing

Early generations of products had little or no inherent capability to test themselves. Their simplicity was such that it was easy to tell whether or not they were functioning properly. The technologies involved often required only visual inspection and some limited probing to troubleshoot the system.

However, as the complexity of military and commercial systems grew, symptoms of failure became less obvious to the operator. Therefore, the procedures to access, inspect, repair, and replace a component became complicated; requirement for skilled personnel and sophisticated test equipment increased; thus, too much time could be needed to maintain a system. As the requirement for availability became more mission-critical, system maintainability became very important.

The obvious solution was to design in-system circuits or devices to self-test the primary system; consequently, Built-In-Test (BIT) was born. This approach continues to be utilized and is an integral part of systems now being designed.

BIT now plays an increasingly important role in military and commercial systems as maintenance becomes a greater factor in system development. The cost of maintenance contributes a large portion to the life cycle cost of a sophisticated system. The development of effective BIT capability is essential to minimizing maintenance time, improving availability, and reducing the cost of system maintenance. Moreover, BIT, which monitors the primary system on-line and reports the system's status and failure information to the decision making device, takes part in automatic control configuration as well.

In general, BIT has found numerous applications in the real world. However, the capabilities of BIT and especially of BIT diagnosis, have not kept up with the complexity of their target systems. The process of fault detection and isolation is often limited or incomplete, and the false alarm probability is too high.

In many cases, BIT can only distinguish whether the subject equipment or unit under test is fully functional or completely broken down. Once a fault is declared, there is rarely any attempt to corroborate

its validity or determine its duration. Decisions are based on limited data input from the unit being tested, even though that data may vary over time or be affected by stimuli outside of the unit itself. No allowance is made for environmental factors, the past performance of the system, previous phenomena indicated by the BIT, or the degraded capabilities of the unit or its components. Real-time correlation between the observed response of the unit under test and current operational and environmental parameters is not performed. Also, lessons learned from one diagnosis are rarely retained for future comparison.

These shortcomings of BIT cause serious problems: wasted time, manpower and resources, failure to complete a mission, and psychological problems --high false alarm rates annoy people and cause them to doubt BIT.

In recent years, attempts at improving BIT have received considerable attention. A number of theoretical and applied papers have been published. The BIT session and exhibition have for years been an important and interesting features of the Annual Reliability and Maintainability Symposia and other notable international RAM conferences.

II. General Concept of Built-In-Test

The term "Built-In-Test" refers to a subsystem whose major purpose is to test the operating state of the primary system. Briefly, BIT is the hardware and software that are integrated into a system to perform fault detection, diagnosis and isolation, and failure recording, along with possible reconfiguration or failure management.

1. BIT Structure Diagram

Generally, BIT is a microprocessor-based testing equipment. The structure of BIT includes hardware and software. The hardware includes detection circuits or electrical (or mechanical) sensors and processors, including CPU, ROM, RAM, an Input/Output unit, and other units. In addition, other hardware such as analog-to-digital converters, comparators, and a display subsystem may also be included.

"Software" refers to micro-programs which can detect, diagnose, and isolate the fault in the system. The micro-programs are installed into the RAM of the microprocessor. The software can also control the system's operation. Usually, BIT software development depends on Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA), and, of course, on the user's objective. One must thoroughly analyze all potential failure modes as a basis for planning the BIT system. Fig. 1 is a block diagram of a typical BIT structure.

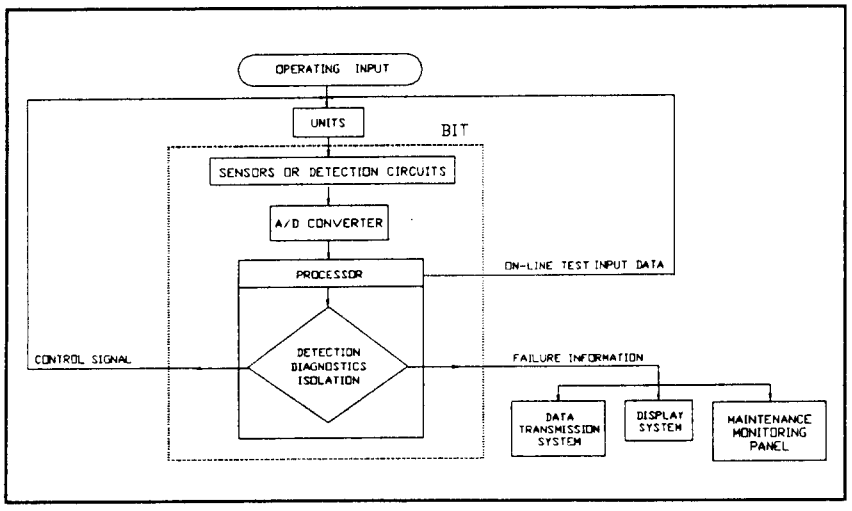

Fig. 1. Built-In-Test Block Diagram (revised from reference [25])

2. How Does BIT Work?

When a system is operating with BIT or is being tested by BIT, sensors obtain the necessary signals concerning system status. These signals are transformed into digital form through analog-digital convertors. Then the microprocessor analyzes the signals to detect a failure, diagnoses the failure mode and time, isolates the failure in a certain component, and assesses the effects of the fault on the system.

If the failure is critical to the system, warning signals are given. The BIT system may then be used as a basis for automatic system re-configuration. Finally, all the failure information will be transmitted to and stored in a database for system maintenance and repair analysis.

To apply BIT to a system, failure modes must be determined, and some important questions answered, such as: "How does a BIT detect a failure? How can we obtain and record the necessary fault information? How do we locate sensors in the right places? How many kinds of failure modes should be detected by BIT? How does a BIT isolate a failure in the proper Line Replaceable Unit (LRU)?"

The system designers should also decide what kind of signals they need, what kind of sensors will be used, and how I/O data will be processed. Therefore, software to help in making these decisions must be developed.

III. BIT Diagnosis Problems

Generally, BIT diagnosis problems include undetection (type 1 mistakes) and false alarms (type 2 mistakes). In addition to these two types of basic mistakes, there is unisolation, that is, the BIT detects a real fault but fails to isolate the fault in the proper LRU.

From available field data, it is clear that existing test techniques are capable of reaching a high detection goal (99%), and a high isolation goal (98%), but fail to meet a moderate false alarm criterion [9,18,29,30,31,32,34]. Malcolm [30] provided field data indicating that typically 30% of in-flight BIT-detected faults could not be duplicated on the ground, and typically 20% to 30% of the units faulted by BIT were found to be fault-free in the shop. Needless to say, the cost of these inefficiencies has been enormous. Therefore, considerable attention has recently been given to BIT false alarms, which waste manpower and resources on unnecessary maintenance activities.

IV. Literature Review

In the past decade, many research papers on BIT have appeared in journals and proceedings. A number of the papers referenced in this paper are summarized in Table 1. These references include BIT design parameters, diagnosis problems, approaches to improving BIT diagnosis, applications of BIT technique, the impact of BIT on systems' RAM, BIT subsystem configuration, etc. This subsection is a brief review of the relevant literature.

1. BIT Diagnosis Parameters and Assessment

Definition and specification of BIT diagnosis parameters are very important top level tasks for both conceptual definition and for a good BIT system design. Improper specification and assessment of BIT diagnosis parameters will greatly influence performance, BIT hardware and software design, the time required and the cost of developing a BIT, etc. Presenting consistent definition, specification, and assessment of BIT diagnosis parameters is the first step in improving BIT diagnosis. Unfortunately, this task is often overlooked.

1.1 Definitions of BIT Diagnosis Parameters

A number of research papers on BIT diagnosis parameters have been published (see Table 1).

Unfortunately, almost every paper uses its own terms and definitions of BIT diagnostic performance. This is not unusual for a new and developing field that is application-driven, and is due to the different industrial and military organizations working on BIT systems. Accepted standards for notations and terms have not yet been clearly established.

It is easy to ascertain that the terminology, forms, and notations chosen to describe BIT parameters in the literature are inconsistent.

Two terms are used to describe BIT false alarm performance. One is false alarm probability F_a, and the other is the false alarm rate λ_{FA}. One point of confusion in many papers is that the false alarm probability F_a (as defined in Table 2) is also called the "false alarm rate." A false alarm occurs when the

Table 1. Classification of References

CLASSIFICATION	REFERENCES
Design Parameters and Assessment	1,2,3,6,8,9,13,18,24,25,26,27,28,30,32,34,35,37,39,40,42,43,45,46,49,50,51
BIT Diagnosis Problems	2,6,8,16,18,27,28,29,30,31,32,35,39,41,43,45,46,50
Approaches to Assess and Improve BIT Diagnosis	1,2,3,6,9,16,18,19,20,25,38,39,41,42,44,45,46,47
Bayesian Processor	21,29,30,31,32
Expert Subsystem	5,7,10,15,17,32
BIT Applications	4,11,12,14,18,23,24,25,33,34,46,47,48
Impact of BIT on System	2,3,13,14,16,24,25,26,27,39,41,45,46,51
BIT Subsystem Configuration	4,8,12,14,19,22,23,25,45,46,48

BIT indicates a fault but no actual malfunction exists in the primary system.

The false alarm probability is useful for maintainability engineering and is convenient for BIT demonstration, but it may not be convenient for BIT system analysis. The false alarm probability is the probability that BIT will indicate a failure when there is no actual system malfunction.

The false alarm rate λ_{FA} is the rate at which the BIT issues false alarms in a certain time interval for the BIT surviving at the start of the interval.

Some papers defined the detecting and isolating capabilities of a BIT as one of two boolean values, "Yes" or "No," for each failure mode. Although this is true for some cases, a BIT has certain detection and isolation probabilities for every failure mode.(see Table 2 for the definitions) Hence, a more precise definition is necessary.

Table 2. Definitions for some BIT Diagnosis Parameters

NOTATION	TERM	DEFINITION
F_a	False Alarm Probability	The probability that BIT will indicate a failure when there is no actual system malfunction.
λ_{FA}	False Alarm Rate	The rate at which the BIT issues false alarms in a certain time interval for the BIT surviving at the start of the interval.
F_d	Fault Detection Probability	The probability that BIT will detect an existing functional failure in the system.
F_i	Fault Isolation Probability	The probability that BIT will isolate a failure that has been detected by BIT down to the specified level (usually a single LRU).
F_{di}	Fault Detection and Isolation Probability	The probability that BIT will detect an existing functional failure and isolate a failure to the specified level.

1.2 Assessment of BIT Diagnosis Parameters

When the BIT technique is employed in a system, a number of benefits will be obtained. However, a BIT is just another subsystem which increases the complexity of the total system, and BIT diagnosis mistakes will cause serious problems.

In references [1,3,6,8,9,18,25,27,39,44,45,46] two kinds of assessments for BIT diagnosis parameters are presented. One is a qualitative analysis method and the other is quantitative.

For purely quantitative assessment, as in the case where a BIT simply detects, isolates and indicates faults but does not obstruct or influence the system operation (i.e., the BIT only improves system maintainability), we can simply treat the BIT as an additional subsystem on a series reliability analysis model. That is, the BIT or several BITs and the primary system form a series reliability model. The mission reliability and the basic reliability models will be same. (For detailed definitions of mission and basic reliabilities, see MIL-STD-765B [36].) Because this case is fairly easy to handle, we only need to consider the effectiveness of BIT for maintainability, E_b. The formulas for calculating E_b have been

given by Shao and Lamberson [45], Bozic and Shaw [3], Lord [27], and Palazzo and Rosenfeld [39].

However, this is not always an accurate case. It is likely that BIT takes part in system control or decision making; therefore, the effect of BIT diagnosis on system mission reliability must also be considered.

Unfortunately, little previous research can be found from the literature. Shao and Lamberson [45] presented the block diagramming concept to reduce a complex system with BIT to simple and basic blocks. They also applied the compound-events method [21] to model a block with BIT.

Because the compound-events method is not suitable for a large or sophisticated system, further research is needed to decompose a system.

2. Present Methods to Deal with BIT False Alarms

There are two potential problems with BIT false alarms which must be solved. One is how to estimate the impact of BIT false alarms on the system's Reliability Availability Maintainability (RAM), and the second is how to reduce the number of BIT false alarms. To present a clear understanding of these two problems, it is necessary to analyze the false alarm (FA) phenomenon and develop a better expression. There are random undulations of failure and threshold signals, such as tolerance effects, operational environment changes (including temperature, humidity, pressure, shock, vibration, etc.), random noise interference, electromagnetic interference, excessively low detecting thresholds, poor filtering capability, lack of self-adaptability, incomplete FMEA, graceful degradations or tuning, normal system variability, sneak circuit effects, and possible software errors. Thus, false alarms can be caused by a number of random factors, and the law of rare events might be applied. We can treat $\lambda_{FA}(t)$ as a constant λ_{FA}.

It should be pointed out that there is a difference between false alarms and common failures. A false alarm can occur more than once. For instance, if a false alarm occurs, it leads to the rejection of a good LRU which is operating normally, but we do not necessarily consider the BIT to be failed, since the user and builder have agreed that a certain false alarm rate is allowed by the BIT specifications. Then BIT is not considered to have failed, even though it indicated a false alarm, since it can work continually and normally until the next false alarm occurs. Based on the analysis of these characteristics of false alarms, many papers [3,16,18,27,28,29-32,39,41,43,45,50] have studied the impact of BIT false alarms on the system, and the methods of reducing the false alarm probability.

Bozic and Shaw [3] incorporated the BIT false alarm rate into their mathematical models for system maintainability and availability (M&A). Shao and Lamberson [45] investigated the impact of BIT false alarm on a system's RAM and concluded that BIT false alarms influence system maintainability, availability, and reliability.

As the major problem in BIT diagnosis, false alarms limit the effectiveness and applications of Built-In-Test.

A number of methods to reduce BIT false alarms have been proposed in the literature.

Kreuze [24] presented FMEA-derived BIT analysis. However, Irwing [18] questioned FMEA as a useful tool for designing BIT. Experience shows that FMEA is almost always optimistic in terms of assessing BIT effectiveness, and is particularly optimistic if carried out at high level.

Specifying more reasonable or optimal thresholds for BIT is suggested by Jager [19] and Malcolm [31,32]. However, simply shifting the threshold does not help much with BIT diagnosis, as it may reduce BIT false alarm and detection probability simultaneously.

Malcolm [30-32] also suggested some other methods, such as continuous monitoring of key signals and data recording, and failure analysis from recorded data. He presented his new approach of using the Bayesian processor as a false alarm filter. As false alarms are also caused by variations in environmental conditions and random electro-noise, Kalman filtering is helpful [45].

Many papers have proposed Artificial Intelligence (AI) as the ultimate solution [3,20,32,45], but more studies are needed to judge its feasibility.

Recently, a knowledge-based system was utilized in a BIT system. Buswell and Sesto [5] investigated the incorporation of a knowledge-based expert system into the operational program of an electronic system to serve as the fault isolation executive.

Buswell and Sesto wrote that fault isolation requirements for large electronic systems are becoming increasingly more stringent. The software implementation of traditional algorithmic solutions can be a

time-consuming and costly process. An expert system promotes a separation between diagnostic knowledge and intricate algorithms. The testability design engineer can encapsulate knowledge gathered from system design data, as well as empirical knowledge gained from hands-on experience, into a database that will be inherently separate from the processing elements which constitute the applications software of the system. Furthermore, if new maintenance information is uncovered, or if the system is modified, the knowledge base can be updated to reflect the added expertise. Many books and papers [7,10,15,17,32] have introduced the applications of expert system in medical diagnosis. From the viewpoint of test data processing, there is considerable similarity between engineering test diagnosis and medical diagnosis. It might be possible, therefore, to incorporate an expert system into a BIT configuration.

In order to prevent a system from BIT mistakes, some compensating techniques have been introduced. These include: BIT circuitry redundancy [41,45], BIT condition assessment [1,41], multiple detecting [45], not responding to a failure indication [41], an overlapping self-test technique and self-test indication analyzer [46], etc.

3. The Bayesian Processor -- A False Alarm Filter

Malcolm, a senior scientist in the advanced programs support laboratory of Hughes Aircraft Company, presented a new look at solving BIT false alarm problems. His basis is the Bayesian processor -- a false alarm filter.

In [31,32], Malcolm describes a test problem that the essence of the diagnostic error problem is that fault-free systems have a certain probability of failing a test (the test result is False Positive) and faulty systems have a certain probability of passing a test (the test result is False Negative).

Using the Bayes theorem, he provided formulas for the probability that the system is faulty, given the test result being positive or negative:

$$P(F \mid T^{+}) = \frac{P(F \text{ and } T^{+})}{P(T^{+})}$$

$$= \frac{P(T^{+} \mid F) \cdot P(F)}{P(T^{+} \mid F) \cdot P(F) + P(T^{+} \mid \bar{F}) \cdot P(\bar{F})}$$

$$= K_1 \cdot P(F) \qquad (1-1)$$

and

$$P(F \mid T^{-}) = \frac{P(F \text{ and } T^{-})}{P(T^{-})}$$

$$= \frac{P(T^{-} \mid F) \cdot P(F)}{P(T^{-} \mid F) \cdot P(F) + P(T^{-} \mid \bar{F}) \cdot P(\bar{F})}$$

$$= K_2 \cdot P(F) \qquad (1-2)$$

The derived posterior odds after the n^{th} test :

$$O_n = \frac{P_n(F \mid D_n)}{P_n(\bar{F} \mid D_n)} = \frac{P_{n-1}(F \mid D_{n-1})}{P_{n-1}(\bar{F} \mid D_{n-1})} \times \frac{P(D_n \mid F)}{P(D_n \mid \bar{F})}$$

$$= (Prior \quad Odds)*(Likelihood \quad Ratio) \qquad (1-3)$$

where D_j is the j^{th} diagnosis result ($j = 1, 2, \cdots, n$), and the probability that the primary system fails after the n^{th} test, P_n:

$$P_n = \frac{O_n}{1 + O_n} \qquad (1-4)$$

Fig. 2 shows that the representative output of a Bayesian processor.

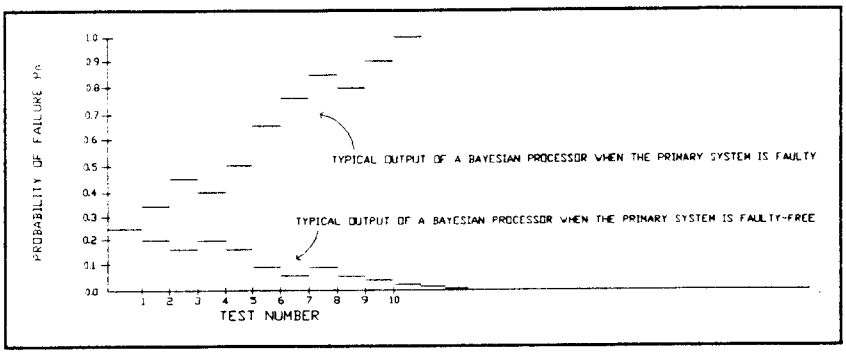

Fig. 2. Representative Output of a Bayesian Processor

He also mentioned that the Bayesian processor has been investigated via computer simulation, using a Monte Carlo-type approach. Because of the complex nature of the results, he did not describe the simulation in [32]. Prior odds and the likelihood ratios are two very important aspects for Malcolm's method. He considered the prior odds to be determined from field data. On the other hand, the likelihood ratios can be estimated by the cutting score and are validated by repeating BIT numerous times on equipment that is known to be fault-free (or faulty) and noting the percentage of times that a BIT fail indication is generated under varying environmental conditions.

The likelihood ratio indicates the extent to which the newly collected datum point (either positive test result T^+ or negative test result T^-) supports the hypothesis of failure. In the event of a BIT failure (T^+), the likelihood ratio is a number larger than 1, and in the event of a BIT pass (T^-), the likelihood ratio is a fractional number. Otherwise, the Bayesian processor has no significance. The initial odds (start-up point) get washed out as more and more tests are performed. The posterior odds will then approach either reach zero or unity.

Obviously, Malcolm's method is very interesting. However, it is still not sufficiently refined for use in practice. There are some shortcomings in Malcolm's Bayesian processor model: (1) He uses only one cutting score, which does not match most cases in the real world as a test threshold. (2) He assumes that the status of the primary system is fixed, either functioning or failed. (3) He does not list the necessary assumptions and limitations of the Bayesian processor. (4) He does not consider problems with the data recording and transmitting equipment and interfaces of the Bayesian processor to other devices and with human beings. (5) He does not develop analytical approaches to compensate for the likelihood ratios and the prior odds, etc. Thus, his Bayesian processor method must be improved in order to be employed in engineering practice.

4. BIT Subsystem Configurations

BIT is the testing equipment integrated into a primary system. It is relatively straightforward to design the individual testing circuits or components. References [4,48] provide the logic for the design of BIT hardware and software. Our interest is in the configuration of the BIT subsystem that is incorporated into the primary system, and the relationship between BIT diagnosis and BIT subsystem configuration.

Usually there are two ways to structure a BIT subsystem. One way is to structure it as a centralized system; the other is to structure it as a distributed processing system.

Krause [22] compared the distributed and the centralized BIT processing concepts. He concluded that the distributed BIT processing concept is a state-of-the-art approach to testability. Harris [14] further presented two kinds of distributed BIT processing: the star network and the data bus. Both Krause and Harris assumed that there is a central control unit in distributed BIT processing structure.

Lamberson and Shao [25] extended the concept of distributed processing and labeled it Distributed-Centralized BIT processing. In addition to a central control unit, Shao and Lamberson have also considered display and data transmission equipment, which are necessary and important interfaces between the BIT system and operators.

Daugherty and Steinmetz [8] described four basic types of BIT architectures; they concluded that the confederated BIT architecture (very similar to the Distributed Centralized BIT structure) is more appropriate for large systems where control is shared in many locations.

V. Further Topics To Improve BIT Diagnosis

1. Algorithm for Redundant Systems with BIT

Shao & Lamberson have developed a set of mathematical models to deal with cold standby, k-out-of-n, shared load and majority voting system with BIT. A complete algorithm for the mathematical models and sensitivity analysis is needed in this topic.

Some typical sets of BIT parameter data will be input to the computer program and the corresponding reliability, maintainability and availability results of redundant systems can be readily obtained, and a series of curves can be drawn.

The algorithms would be very applicable for engineers, designers and reliability analysts of any redundant system because a redundant system without BIT actually is the same system with a perfect BIT.

In addition, this is a good way to validate and refine upon Shao and Lamberson's mathematical models if necessary. It is possible that this task can be extended as a commercial software package because I have not seen a satisfied mathematical model which represents a redundant system considering both undetection and false detection.

2. Reducing BIT False Alarm Probability

Considerable attention has been focused to BIT false alarms, which cause series problems such as decrement of mission reliability, great amount of waste with manpower, time and resources, etc.

A couple of methods can be suggested as well such as :

* Using more complete FMEA and FTA to find out as many failure modes as possible, otherwise the BIT does not know how to diagnose some phenomena which were not defined in FMEA, then a false alarm may occur.
* More reasonable specifications for the threshold values of BIT.
* Applying Bayesian analysis technique in diagnostics to fully utilize the prior failure data to improve or modify the Built-In-Testing results. Malcom [30,32] has proposed this concept, but the main job is to develop applicable mathematical model and solve it.
* Expert system and self-adjustment technique. It is helpful for diagnosis and decision making, and it is also possible to filter off the random noise interference and self-adjust the threshold values of BIT.

3. Protecting a System from BIT Mistakes

BIT mistakes include undetection, false detection (false alarm) and the failure of BIT itself.

It is an interesting task because nobody can assure BIT does not have fault. In the case that BIT has made a mistake, what can we do to protect the primary system, to avoid mission abortion and to avoid waste of time and resources for useless maintenance activities?

In order to improve system reliability and maintainability. We have to develop some BIT design approaches that can compensate for BIT mistakes and hardware failure such that a system can continue full or partial operation. Following compensation approaches can improve BIT design as well :

* BIT condition assessment technique
* Not responding to a diagnostic mistake
* Overlapping BIT technique
* Conditional Diagnosis
* Multiple detecting
* BIT indication analyzer

VI. Conclusions

In the last decade, Built-In-Test (BIT) has found wide applications in military and commercial systems, because maintenance has become an important factor in systems development. BIT plays an important role in minimizing maintenance time, improving availability, and reducing the cost of system maintenance. In modern electronic systems, BIT also monitors the primary system on line: when a fault is detected, BIT recovers the fault by controlling and reconfiguring the system.

However, the capabilities of BIT have not kept pace with the complexity of its target systems. The process of fault detection and isolation is not ideal, and the false alarm probability is too high. Reducing false alarms is receiving considerable attention. It is well known that BIT diagnosis problems are very complicated.

Clearly, someone needs to widen the state-of-the-art of Built-In-Test, improve design and evaluation methodology for BIT, provide an applied algorithm and analysis for redundant systems with BIT and develop some new approaches to reduce false alarm rate and protect the system from BIT mistakes.

REFERENCES

[1] Albert, J. et al., "Built-In-Test Verification Techniques," Proceedings Of Annual R&M Symposium, pp. 252-257, 1986.
[2] Aly, A. A., et al., Norman School of Industrial Engineering, "Analytical Procedures for Testability," AD-A126167, 1983.
[3] Bozic, S. and Shaw, L., "Is BIT a Toy, Blessing or Annoyance," Proceedings Annual R&M Symposium, pp. 270-275, 1985.
[4] Buswell, S. and Graves, F., "Self-Test of Timing and Control Circuitry," Annual R&M Symposium, pp. 108-111, 1988.
[5] Buswell, S. and Sesto, P., "An Expert System to Facilitate Fault Isolation," Annual R&M Symposium, pp. 74-81, 1989.
[6] Carroll, W. H. et al., "Diagnostics Specification--A proposed Approach," Annual R&M Symposium, pp. 227-231, 1981.
[7] Chee, C.W. and Power., "Expert System Maintainability," Proceedings of Annual R&M Symposium, pp. 415-418, 1990.
[8] Daugherty, G. and Steinmetz, G., "BIT Blueprint Toward more Effective Built-In-Test," Proceedings of Annual R&M Symposium, pp. 353-360, 1990.
[9] Gleason, D., "Analysis of Built-in-Test Accuracy," Proceedings Annual R&M Symposium, pp. 370-372, 1982.
[10] Goodall, A., The Guide to Expert Systems, Learned Information (Europe) Ltd. Oxford, England, 1985.
[11] Gordon, T. W., "F/A-18 Hornet Reliability Program: Status Report," Annual R&M Symposium, pp. 228-231, 1986.
[12] Haedtke, J. E. and Olson, W. R., "Multilevel Self-Test for the Factory and Field," Proceedings of Annual R&M Symposium, pp. 274-278, 1987.
[13] Harris, D. E., "Built-In-Test for Fail-Safe Design," Proceedings of Annual R&M Symposium, pp. 361-366, 1986.
[14] Harris, D. E., "Built-In-Test to support Remote System Maintenance," Annual R&M Symposium, pp. 422-427, 1984.
[15] Hart, A., Knowledge Acquisition for Expert System, Kogan Page Ltd. London, 1989.
[16] Holbrook, R. O., "Correction For BIT Detectability Maesurement Error," Proceedings of Annual R&M Symposium, pp. 243-247, 1986.
[17] Hollnagel, "The Reliability of Expert Systems," ELLIS HORWOOD LIMITED, Chichester, West Sussex, England, 1989.
[18] Irwing, M. H., "Built in Test--Past Mistakes, Present Problems and Future Solutions," Reliability Engineering, 15(4), pp. 245-261, 1986.
[19] Jager, R., "A System Approach to Designing for Testability," Proc. of Annual R&M Symposium, pp. 520-524, 1984.
[20] Jager, Rhine, "Integrated Diagnostics: Extension of Testability," Annual R&M Symposium, pp. 248-251, 1986.
[21] Kapur, K. C., and Lamberson, L. R., Reliability In Engineering Design, John Wiley & Sons, Inc., NY, 1977.
[22] Krause, G. S., "Distributed vs. Centralized BIT/FIT Processing," Annual R&M Symposium, pp. 291-295, 1985.
[23] Krause, G. S., "Microprocessing to Reduce MTTR of Analog Systems," Annual R&M Symposium, pp. 505-509, 1984.
[24] Kreuze, F. J., "BIT Analysis and Design Reliability," Proceedings of Annual R&M Symposium, pp. 328-332, 1983.
[25] Lamberson, L. R. and Shao, J., "Built-In-Test Technology in Commercial Systems," Proceedings of International Industrial Engineering Conference, Toronto, Canada, pp. 357-362, 1989.
[26] Locurto, C. A., "Impact of BIT on Avionics Maintainability," Annual R&M Symposium, pp. 328-332, 1983.
[27] Lord, D. and Gleason., "Design & Evaluation Methodology for Built-In-Test," IEEE Trans. R-30, pp. 222-226, 1981.
[28] Lord, D. N., et al., Grumman Aerospace Corp, "Design Guidelines and Optimization Procedures for Test Subsystem," AD-A087059, 1980
[29] Malcolm, J. G., "Analysis of Built-In-Test(BIT) False Alarm Conditions," AD-A108752, 1981.
[30] Malcolm, J. G., "BIT False Alarm: An Important Factor In Operational Readiness," Proceedings of Annual R&M Symposium, pp. 206-212, 1982.
[31] Malcolm, J. G., "Practical Application of Bayes' Formulas," Proceedings of Annual R&M Symposium, pp. 180-186, 1983.
[32] Malcolm, J. G. and Foreman, G. L., "The Need: Improved Diagnostics--Rather than Improve R," Proceedinga of Annual R&M Symposium, pp. 315-322, 1984.
[33] McGee, R. L, "R&M Contribution to Hornet (F/A-18) Mission Success," Annual R&M Symposium, pp. 338-341, 1979.
[34] Merlino, D. H. and Hadjilogiou, J., "Built-In-Test Strategies for Military Systems," Annual R&M Symposium, 1989.
[35] Meth, M. A., "Finding Faults--A Dilema For Statisticians," Proceedings of Annual R&M Symposium, pp. 351-355, 1983.
[36] MIL-STD-765B, MILITARY STANDARD, Reliability Modeling and Prediction, pp. 101-1, 1981.
[37] Moreno, F. J., "BITE Availability Under Operations Constraints," Proceedings of Annual R&M Symposium, 1989.
[38] Moreno, F. J. and Reinert, D. L., "Fault Isolation Methodology For Supportability," Proceedings of Annual R&M Symposium, pp. 50-53, 1985.
[39] Palazzo, C. J. and Gleason, D., "Avionics Built-In-Test Effectiveness and Life Cycle Cost," AIAA-83-2448, pp. 1-6, 1983.
[40] Pliska, T. F. et al., "BIT/External Test Figures of Merit and Demonstration Techniques," AD-A081128, 1980.
[41] Ramirez, M. A., "Compensation for Built-In-Test Failures," Proceedings of Annual R&M Symposium, pp. 258-261, 1986.
[42] Ramirez, M. A., "Relevancy in Fault Detection Analysis," Proceedings of Annual R&M Symposium, pp. 118-121, 1988.
[43] Richard, D. W., "Smart BIT," Proceedings of Annual R&M Symposium, pp. 31-33, 1987.
[44] Rosin, A., "An Approach to the Selection of Built-In-Test Devices," Annual R&M Symposium, pp. 346-350, 1990.

[45] Shao, J. and Lamberson, L. R., "Impact of BIT Design Parameters on Systems RAM," Reliability Engineering and System Safety, Vol. 23, pp. 219-246, 1988.
[46] Shao, J. and Yoo, W., "Effect of Self-Test on the Reliability of an Automatic Control system," Proceedings of China-Japan International Symposium on Instrumentation, Measurement and Automatic Control, Beijing, 1989.
[47] Smith, G. M. and Schroeder, J. B., "Advanced Maintenance Diagnostics for Air Force Flight Control," Proceedings of Annual R&M Symposium, pp. 394-399, 1990.
[48] Smoot, S., "Designing Built-In-Test for Microprocessors," Proceedings of Annual R&M Symposium, pp. 412-415, 1984.
[49] Sylvestre, L. M., "ADINTS: Moving Toward Standardization of Automatic Test Equipment," Proceedings of Annual R&M Symposium, 1989.
[50] Trakas, R. C., "Establishing Realistic Requirement for Reliability, Maintainability, and Built-In-Test," Proceedings of Annual R&M Symposium, pp. 103-107, 1984.
[51] Tuttle, D. E., "Built-In-Test and External Tester Reliability Characteristics," AD-A083488, 1980.