Fault Simulation Requirements for Security Fault Analysis

STATUTORILY EXEMPT

[redacted]

Editor's Note: This paper was awarded second prize in the applications category of the 1986 Computer and Information Sciences Institute Essay Competition.

Recent changes in digital design technology and the approach to developing communications security (Comsec) systems accentuate some limitations of the digital simulation software tools currently used in security fault analysis (SFA). Since SFA is performed on each new Comsec equipment design, the accuracy and efficiency of these tools is significant. To maximize the effectiveness of the SFA tools, some enhancements to the existing capabilities are needed. This paper explains why the enhancements are needed, what they are, and how to best implement them.

I. INTRODUCTION

Security fault analysis (SFA) determines how potential electronic failures affect the operation of the security-related functions of a communications security (Comsec) equipment. Security evaluation analysts use digital simulation techniques as an aid to improve the efficiency and accuracy of SFA, which is performed on each Comsec equipment design. The current SFA software tools help the analyst to model failure modes and analyze their effects at the transistor level, the logic gate level, and the system level.

The current SFA tools are based on simulation tools which were developed or acquired by Comsec designers over the last 20 years. When efforts to demonstrate the feasibility and utility of simulation as an SFA aid were begun about seven years ago, the capabilities of existing gate level and system level logic simulation tools were extended to include the fault simulation capabilities required for SFA: fault specification, monitoring, and reporting criteria. The efforts were successful and the tools are now used in the SFA of some but not all Comsec equipment designs. This limited application of the SFA tools exists mainly because the SFA tools are based on 20-year-old logic simulation tools which cannot accurately and efficiently model current technology. With the increasing complexity and volume of Comsec designs, the analyst needs more effective SFA tools to continue to obtain meaningful SFA results. Thus, an effort to enhance the existing SFA fault simulation capabilities was begun by the Y321 SFA workcenter. During the initial phase of this effort, from June 1985 to December 1985, a four-person fault simulation team developed and investigated several issues:

- why existing fault simulation capabilities must be enhanced,
- what system level and analytical capabilities are needed,
- why integrated fault simulation capabilities are needed,

- how to implement SFA capabilities with commercial products, and
- how to best implement integrated fault simulation capabilities.

Subsequent phases of the enhancement effort are planned to

- implement the required integrated fault simulation capabilities,
- develop the methodology for using these capabilities as an SFA aid,
- apply and validate the methodology for some typical Comsec equipment designs, and
- use the integrated fault simulation capabilities to support the Y321 SFA evaluations and all in-house SFA efforts.

II. WHY EXISTING CAPABILITIES MUST BE ENHANCED

The existing fault simulation capabilities include two digital logic simulation software packages customized to consider SFA fault specification, monitoring, and reporting criteria at the logic gate and system levels. Both tools resulted from independent efforts to demonstrate the feasibility and utility of fault simulation as an SFA aid. The DELFAS package was developed in R group specifically for the SFA task. The LOGICV SFA capabilities were added, on a contracted effort, to an existing digital simulation package, LOGICIV, which evolved through in-house and contracted development efforts. These currently used in-house simulation tools are a boon to the analyst, but they do lack some capabilities that the analyst needs to cope with recent changes that have produced both more complex technology and more Comsec designs to evaluate.

1. Current technology cannot be accurately and efficiently modeled.

a. Neither DELFAS nor LOGICV has inherent structures for modeling the microprocessor-based architectures, gate arrays, or programmable logic devices being used in current Comsec designs. Both simulators lack a tri-state capability that is necessary to model bus architectures to support these technologies.

b. DELFAS does not have inherent structures for hierarchically modeling a digital system. Hierarchical modeling capabilities give an analyst an effective means for modeling large system designs where he must analyze a few small circuits that are driven and sensed by numerous large circuits. In such cases, detailed gate level models could be used for the small circuits that must be analyzed and less detailed functional models could be used to define the operations of the numerous large circuits that are necessary to drive or sense the small circuits. This approach produces faster simulation turnaround, saves computer storage required for models, and eases the analyst's job of creating simulation input and analyzing simulation output. These savings are particularly significant with the increasing use of very large scale integrated (VLSI) circuits and the Commercial Comsec Endorsement Program (CCEP) designs, where a total system design may contain tens of thousands of logic gates but the analyst needs to analyze failure effects on only hundreds of gates and a few system level (functional) signals.

c. Even if the current tools included bus structures and hierarchical capabilities, modeling some current designs would still be impossible, and modeling many others could still be inefficient. For example, if the detailed design of a commercial microprocessor is proprietary, the analyst cannot model it. Also, for known but complex designs, modeling one component may take more time (several man-months for a microprocessor) than the remaining SFA effort (several man-weeks).

2. Circuit design information is not efficiently implemented.

a. In both DELFAS and LOGICV, modeling the interconnections of any circuit schematic is a slow and error-prone process. The analyst enters circuit design models in the form of a netlist where he assigns numbers for each node in the net and codes the connections for each node using these numbers. The inefficiencies and error-prone nature of the netlist approach have always been a problem for the analyst and become less affordable as system complexity increases. Now, the analyst spends time entering the schematic that could be more effectively used analyzing the fault monitoring criteria and the simulation results, tasks which provide confidence that all significant system effects are considered. Also, the netlist is useless to the analyst who needs circuit design information in the form of a logic drawing for understandability. Logic drawing format is also useful for SFA document processing, which should be integrated with the simulation software in which the design information is originally specified.

b. Configuration control of design information is often a problem because DELFAS and LOGICV are not integrated with computer-aided design (CAD) systems used by Comsec designers. Moreover, this problem is increasing as CCEP involvement increases. If the analyst could obtain controlled design information in an input format readable by his simulation package, he could avoid having to recode or reenter models and could be assured that the design he is analyzing is consistent with that of the designer. Being able to model the right design version the first time will give the analyst more time for analysis, a critical need as Comsec systems become more complex.

3. User support for SFA tools is increasingly insufficient. Neither DELFAS nor LOGICV is supported by enough staff to make enhancements for new technology and to provide adequate user services to Comsec designers and analysts. As Comsec designs become more complex and numerous, the analyst requires more complex tools and greater support services.

4. The existing tools are not easy for an analyst to use.

a. Both DELFAS and LOGICV lack appropriate and efficient user interfaces. Design information is difficult to enter, as described above. Fault specification in DELFAS is hard-coded by its maintainers and therefore cannot be changed by the analyst using new technologies which have unique failure modes. Fault monitoring in DELFAS requires the analyst to be a programmer familiar with the DELFAS internal data structures, another task which decreases the time available for analysis.

b. Both DELFAS and LOGICV lack effective user reference documents. Determining how to sequence simulation input commands to build models is not well documented; inputs are described in terms of syntax, not usage. Further, to merely find syntax descriptions of a particular command in a particular input category, the analyst must scan the document until he finds the information. Such documentation deficiencies persist due mainly to the limited support staff, as described above.

c. Debugging models is difficult because of noninformative error messages and ineffective descriptions in the user manual. Also, minor input problems can only be solved by recompiling and rerunning the entire simulation, rather than by applying interactive corrections to the portions in error.

Thus, with regard to these analytic and operational capabilities, enhancements are needed both to improve the performance of the existing capabilities and to extend the existing capabilities to meet new requirements. These enhancements are vital to regaining the proven effectiveness of fault simulation as an SFA aid.

III. WHAT CAPABILITIES ARE NEEDED

The fault simulation capabilities required are based on SFA-unique analytic requirements, the need for solutions to problems with the existing tools, and the desire to develop SFA capabilities integrated with a flexible set of CAD tools used by Comsec designers. The specific integrated fault simulation capabilities that are required are characterized by both system level criteria and analytical criteria, as listed and explained below.

System Level Criteria

1. *Hardware configuration.* The system must include these networking and peripheral capabilities:

a. The integrated software must either run on stand-alone hardware which can be networked to the Y321 VAX-11/780 or run directly on this VAX. If the CAD system is a stand-alone system, additional terminal ports and an Ethernet Local Area Network (LAN) connection to the VAX are needed so that it can be integrated with other SFA capabilities. A stand-alone system networked to the VAX is more desirable than a resident system because the analyst avoids contention for the disks between his edit process and the simulation number-crunching process by doing schematic capture on the stand-alone system and uploading the simulation to the VAX. A stand-alone system also remains usable when the VAX is unavailable due to system problems or periodic maintenance.

b. The system must include a high resolution graphics terminal (e.g., 19-inch color monitor, at least 1022 x 824 pixels), a keypad/mouse device, and a high resolution printer/plotter. These peripherals are necessary to enable the analyst to enter and produce design drawings which are effective references for analysis and are useful in SFA documentation.

2. *Run environment.* The integrated software should run in an environment that includes these operational considerations:

a. The integrated software must run in both interactive and batch modes. The analyst needs to run large simulations in a batch environment, but an interactive environment increases efficiency in debugging initial schematic entry or simulation of small circuits.

b. The integrated software should have a restart capability, so that if the system goes down during a simulation, the entire simulation does not have to be rerun. This is important for large system fault simulations which take many hours or several days to run.

c. The system should support a multitasking run environment to allow the analyst to continue to be productive doing other system operations such as schematic capture or documentation while a large simulation is running.

d. The ability to interface in-house applications software to the fault simulation software should exist to enable integrating the existing software tools used in determining the logical faults which must be specified to the SFA fault simulation software. The complete automation of the fault determination and specification process has been a goal since 1978 but cannot be realized until a suitable fault simulator is available on the VAX. This complete automation increases analyst efficiency and decreases errors associated with manually handling large amounts of similar looking data such as the Boolean equations used in fault specification.

3. *User environment.* The analyst capabilities must be menu driven and must include schematic capture, online help, constructive error messages, syntax and connectivity checking prior to runtime with interactive error correcting, graceful runtime error handling, and effective user documentation. All of these features increase the efficiency of the analyst in his associated tasks. For example, with a schematic capture capability, the analyst uses symbols on a video display unit, draws the actual connections, and adds appropriate labeling and modeling information for each element as he draws it. This eliminates the time-consuming and error-prone netlist entry process.

4. *Throughput.* The fault simulation algorithm, the automation of all fault simulations for a design, and the run and user environments should contribute to simulation throughput that provides timely results for the analyst. This is difficult to quantify for two reasons. First, different analytic efforts require different amounts of resources within the same batch or interactive process. Second, any interactive or batch process contains some operations that are inherently fast and others that are inherently slow. In general, throughput for interactive processes should be on the order of seconds to minutes and throughput for batch processes should be on the order of hours rather than days.

5. *Customer service.* Hardware/software maintenance and training provided by the CAD system vendor relieves in-house support requirements, freeing more time for analysis.

Analytical Criteria

1. *Hierarchical modeling.* The system must support hierarchical modeling at the chip, cell, and transistor levels for accurate and efficient modeling of current technology.

2. *Mixed mode simulation.* The system must support fault simulation for any combination of hierarchical models for accurate and efficient modeling of current technology.

3. *Libraries.* The system must include a library of primitives for CMOS combinational and sequential gates. Additionally, libraries for ROMs, RAMs, PLAs, PALs, and for elements in other technologies, such as bipolar, should be available. Also, the analyst must be able to construct and add his own models to libraries and the libraries must accommodate hierarchical models. These library capabilities contribute to accurate and efficient modeling of current technology and to configuration control of design information and models.

4. *User-defined failure effects.* The system must allow the analyst to define cell level failure effects which are combinational or sequential, since SFA research indicates such effects do occur.

5. *User-selected fault simulation.* The system must allow the analyst to specify which components to include in any fault simulation run, since the analyst generally does not need to fault the entire simulation model.

6. *User-defined monitors.* The system must allow the analyst to trigger observation of a node or set of nodes at specific times or under specific logic conditions during fault simulation. Each observable effect must be reported in the simulation output using descriptive text provided by the analyst.

7. *Hardware modeling.* The system should include a hardware modeling capability. This capability interfaces actual hardware to the simulation environment through connector pins and sockets and can be used to either replace a software model or compare a software model to the actual hardware. One advantage of a hardware modeling capability is to avoid building models for circuits which do not need to be evaluated or are proprietary designs but which are needed to drive or sense circuits that must be evaluated. A typical application is a system in which the Comsec device is controlled by a commercial microprocessor whose design is either proprietary or too complex to efficiently model. Another advantage of a hardware modeling capability is to provide a means of verifying software simulations of devices that must be evaluated. After the analyzed device is built, it can be interfaced to the rest of the simulation environment and its actual operation can be verified. These capabilities contribute to accurate and efficient modeling of current technology.

8. *Simulation sizing.* The system must be able to accommodate VLSI designs and their fault models. This requires a capacity for modeling tens of thousands of gates, for simulating a minimum of four logic states (logic 0, logic 1, don't know, and high impedance), and for fault simulating at least 20 failure effects per cell on hundreds of cells.

9. *Common data base.* The system should use a common data base for design verification, fault simulation, hardware modeling, and an integrated word processing facility.

IV. WHY INTEGRATED CAPABILITIES ARE NEEDED

As discussed above, the SFA analyst has unique analytic requirements, which is why in-house tools were originally developed instead of using commercial fault simulation tools. Because commercial simulators have advanced in technology modeling capabilities and in ease of use while DELFAS and LOGICV have not, integrating the desirable features of commercial tools with the analytical features of the SFA tools is a logical solution for providing the necessary enhancements to the existing SFA fault simulation capabilities. Although the desirable commercial simulator capabilities are obvious –

- state-of-the-art technology modeling,
- hierarchical modeling,
- schematic capture,
- hardware interfacing as an alternative to software modeling,

- effective user support, including documentation, and
- configuration control –

and the required SFA capabilities are obvious –

- fault specification criteria,
- fault monitoring criteria, and
- fault reporting criteria –

the most desirable approach for implementing these capabilities is not.

V. HOW ENHANCEMENTS CAN BE IMPLEMENTED

Deciding the best way to implement the integrated capabilities involved considering the SFA fault simulation requirements and the resources available to satisfy them. Once the possible implementation approaches were identified, they were assessed to determine which was best.

To determine if implementation using a commercial product was possible, the fault simulation team began a preliminary investigation of commercial fault simulation capabilities. About 20 major vendors and less than 10 industry standard simulation packages exist. Initially, ten vendors felt that their hardware and/or software would satisfy the basic system and analytical requirements. These vendors marketed six standard simulation packages:

- CADAT (by HHB-SOFTRON, also marketed by Mentor, CADNETIX, and Futurenet)
- DAISY (by DAISY)
- ZYCAD (by ZYCAD)
- TEGAS (by CALMA)
- HILO-3 (by GENRAD, marketed by Computervision)
- LASAR6 (by TERADYNE, also marketed by VALID-LOGIC)

Of these vendors, only three were interested in pursuing benchmarks which we designed to demonstrate whether their product could perform the basic fault simulation task needed for SFA. This limited interest was due to two major factors:

1. The commercial fault simulation packages do not include the fault specification capabilities required for SFA. Most of the valuable fault simulation packages perform only "stuck-at-zero" and "stuck-at-one" analysis and have no capability to model user-defined gate level combinational and sequential failure effects.

2. Most vendors do not believe that a product with more than "stuck-at" faulting capability would have enough of a market to make its development and maintenance profitable.

In spite of these limitations, a few of the commercial logic simulation packages do have modeling capabilities which enable including faults other than the "stuck-at's." An acceptable fault simulation benchmark run (by HHB-SOFTRON) confirmed that using a commercial fault simulation product was a possible implementation approach.

With these realizations, several implementation approaches are possible for achieving integrated fault simulation capabilities:

1. Develop a custom simulation package that meets only the exact SFA fault simulation requirements, as either an in-house or a contracted effort.

2. Find a vendor who will customize his commercial CAD package to meet the SFA fault simulation requirements.

3. Find a commercial CAD package with enough flexibility to accommodate the SFA-unique fault simulation requirements.

4. Find commercial CAD packages that can be integrated with in-house simulation packages to develop a complete tool set for SFA.

5. Find commercial CAD packages that can be integrated with in-house simulation packages to solve some of the major problems with the existing SFA fault simulation tools.

Given these possible implementation approaches, the feasibility of each is then assessed:

1. To develop a custom simulation package requires extensive manpower for either an in-house or a contracted effort. An in-house development means continuing in-house maintenance and support, which is a current problem that should be eliminated, not continued. Both in-house and contracted efforts require writing detailed design requirements, work which would not be necessary if a commercial package were available. Also, any complete development effort would not provide as timely a product as an existing commercial product, even if it required some custom modifications. A contracted effort depends on finding qualified software developers and willing bidders for limited application software. Further, administering a contract for the full development of as complex a product as is needed requires long-term dedicated manpower, which is not available. For these reasons, this approach was discarded.

2. To customize a commercial CAD package requires dedicated manpower but to a more manageable degree than developing a custom CAD package, because much less than the entire package is involved. However, most vendors do not want to invest in a product that will not serve a large customer base, which is the general perception about the product required. This approach is possible only if a willing vendor is found and his product has the flexibility to accommodate the SFA requirements.

3. To find a commercial CAD package with enough flexibility to accommodate all of the requirements has proven futile, especially in the area of postprocessing the simulation output according to SFA fault reporting criteria. No commercial package has the flexibility for the user to write his own reporting criteria for each unique failure effect, or even for the simulator to report any information on unique effects. Commercial CAD packages are designed to report on singular events in terms of logic zero and logic one levels helpful to the designer and not in terms of descriptive text helpful to analysts performing and evaluating SFA. For this reason, this approach was discarded.

4. To integrate commercial and in-house SFA tools as a complete tool set is possible only if the commercial packages have accessible data structures where interfaces are

needed. Several packages do allow user access to the necessary netlist and simulation results data structures. Therefore, integrating the most desirable capabilities from both commercial and in-house tools is possible. This would enable us to develop a complete tool set without the disadvantages of a full custom software development effort.

5. To integrate commercial and in-house SFA tools to solve some of the major problems (schematic entry/configuration control) with the current SFA tools is worthwhile if a complete tool set cannot be achieved or as an initial investment toward the complete tool set.

Since the full custom and full commercial approaches (approaches 1 and 3) were discarded, the only remaining implementation approaches are ones which merge in-house capabilities with commercial products (approaches 2, 4, and 5).

Considering the results of the vendor survey and the possible approaches, approaches 2, 4, and 5 all are viable because

1. Customizing a commercial CAD package (approach 2) is a reality, although HHB-SOFTRON is the only vendor willing to do so, and

2. Commercial packages can be integrated with in-house SFA tools (approaches 4 and 5) since two vendors offered solutions in these areas:

a. HHB-SOFTRON is willing to customize CADAT code to enable fault monitoring and reporting capabilities similar to DELFAS or LOGICV.

b. DAISY schematic capture software produces a data base and contains a utility that can interface the design data base to any netlist format such as those in DELFAS or LOGICV. Another DAISY utility can interface the fault simulation results data base to in-house postprocessing software.

VI. WHAT IMPLEMENTATION IS BEST

Given the viable implementation approaches, the most expedient and flexible solution is accomplished in two phases:

1. Phase 1. In a near term effort, the current SFA tools are integrated with existing commercial tools to increase the efficiency of specifying the circuit. DAISY has an outstanding library of components, schematic capture package, and netlist interface which can quickly provide a front-end to DELFAS to solve the circuit specification problem. The netlist interface also provides a future capability to interface other CAD manufacturers' tools with the SFA tool set, integrating the SFA tool set with the Comsec designers' tool sets. The schematic capture package also includes a logic simulation package that could be used for fault simulation in cases where DELFAS cannot model certain current technologies. Another DAISY interface utility could provide simulation results to DELFAS for postprocessing according to the SFA fault reporting criteria.

2. Phase 2. In a longer term effort, a commercial tool with the full capability to model current Comsec design technologies and approaches is customized for SFA and integrated with the schematic capture capability implemented in Phase 1. This can be accomplished with HHB-SOFTRON's CADAT package, which already provides all but two of the integrated fault simulation capabilities. These two capabilities are available, as HHB-SOFTRON is willing to customize CADAT to add the SFA fault monitoring and reporting criteria. The other integrated fault simulation capabilities were demonstrated

in the benchmark run and other independent demonstrations provided by HHB-SOFTRON.

VIII. CONCLUSION

More effective analyst aids are needed for SFA. The SFA fault simulation requirements are unique but can be implemented and integrated with industry standard CAD tools.

The proposed implementation method provides SFA fault simulation capabilities which consider current Comsec design technologies and approaches and are integrated with industry standard computer-aided design (CAD) tools used by Comsec equipment designers. The implementation is accomplished in two phases. First, in the near term implementation, the current SFA tools are integrated with some industry standard tools which increase the efficiency and accuracy of specifying the circuit to be analyzed. Second, in a longer term effort, an industry standard fault simulation package capable of modeling current Comsec design technologies and approaches is customized for SFA and is integrated with the circuit specification capabilities previously implemented.

The result is a set of SFA tools which are accurate, efficient, and compatible with the CAD tools used by Comsec designers.

STATUTORILY EXEMPT

[redacted]

REFERENCES

DeLand, R. W. *DELFAS User Manual*, Rev. 2. NSA, R56, 1 July 1985.

CAD/CAA System LOGICV User's Reference Manual. March 1981.

Schuster et al. "Fault Simulation Example." NSA, Y321, 23 October 1985.

NSA Functional Fault Modeling Benchmark. Mahwah, New Jersey: HHB Systems.

CHARGERFMS.REV2 Computer Listing. Columbia, Maryland: Daisy Systems Corporation.

Product brochures and demonstrations provided by vendors:

CADNETIX Corporation, Vienna, Virginia
Computervision Corporation, McLean, Virginia
Daisy Systems Corporation, Columbia, Maryland
Futurenet Corporation, Pasadena, Maryland
HHB Systems Corporation, Mahwah, New Jersey
Mentor Graphics Corporation, Rockville, Maryland
VALID Systems, Greenbelt, Maryland
ZYCAD Corporation, Morristown, New Jersey