A Role for Embedded Instrumentation in Real-Time Hardware Assurance and Online Monitoring Against Cybersecurity Threats

Alfred L. Crouch and Adam W. Ley

Our 2019 AUTOTESTCON paper [1] introduced the concept of hardware assurance for security and trust that focused on Trojan payloads and triggers and the use of embedded instruments and data collection from those instruments as one of the detection strategies. A Phase I Small Business Innovation Research grant funded the proof of concept of using a COTS board as a sandboxed emulation environment to safely study the attack scenarios associated with various Trojan attacks, and that effort is described in this article.

A provider of an electronic system has three basic strategies for the inclusion and deployment of cybersecurity techniques: do nothing; defend against the triggering or activation of Trojans; or assume all electronic systems are infected. The "do nothing" camp assumes that their electronics are inconsequential, or that the probability of an attack is minimal, or that the cost of including cybersecurity hardening is too expensive for their specific product. Do nothing has proven to be a bad strategy in that even inconsequential electronics have been used in public distributed denial of service (DDOS) attacks. The "defend" camp involves the detection of the Trojan before it can activate, which is not an easy task and definitely requires predictive embedded detection instruments. The "assume infection" position represents a mindset that is more about "recovery" during or after a Trojan activation, asking whether or not a circuit can be crafted to operate in the presence of an active Trojan, and uses a slightly different flavor of embedded detection instruments that relies on a change of the circuit from normal behavior.

Several fundamental categories of embedded instruments that provide such monitoring have been elaborated, to include those that:

- detect activity at circuit elements that have been designated as unused;
- enforce time bounds for validity of time-framed operations;
- check for unexpected analog, parametric or side-channel effects;
- guard against back-door operations; and
- raise hardware assertions or alerts in response to deviations from normal behavioral limits.

In combination, and if architected correctly, these instruments can lead to detection, identification, location and impact assessment of hardware exploits.

In this article, we summarize the aforementioned 2019 AUTOTESTCON paper and expand into some of the work done to date.

Embedded Instruments

There are many ways to categorize Trojans [2]. Our research has shown that hardware vulnerabilities, whether intentional or inadvertent, can be exploited via three fundamental means: Leaker—leak data to expose critical data or structure to allow reverse-engineering; Behavior Modifier—change operation to prevent the device or system from accomplishing its intended mission, substitute an alternate mission, or allow a non-authorized party to take over the hardware; or Reliability Impact—degrade, break, or destroy the hardware, operation, or reputation [3].

These different fundamental Trojan attacks, when coupled with stealth assumptions, require different types of instruments for detection (i.e., a stealth requirement mandates that a Trojan within a system must remain hidden, disguised, or sequestered until the ideal or scheduled opportunity arises and payload activation is triggered).

See Fig. 1 for a simple illustration of a cone of logic describing an example Trojan implementation, inclusive of payload and trigger.

In the cone-of-logic example, digital Trojans can be an inserted transistor, gate or wire route, or can be a substituted gate (e.g., a 3 input AND substituted for a 2 input AND). Other organizations, such as a team from New York University have looked into the insertion of Trojans [4]. There are many strategies for detecting these types of Trojans [5]. The most sophisticated and difficult-to-find triggers and Trojan payloads

This article is based on research originally presented at AUTOTESTCON 2019 [1].

1094-6969/20/$25.00©2020IEEE

Authorized licensed use limited to: Peter Levin. Downloaded on September 15,2020 at 18:50:33 UTC from IEEE Xplore. Restrictions apply.

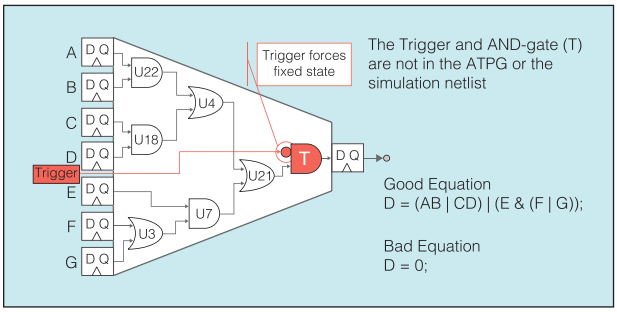

Fig. 1. A cone of logic describing Trojan implementation of a Behavior Modification Kill Switch or a Reliability Impact spoofed Stuck-at-0 Fault.

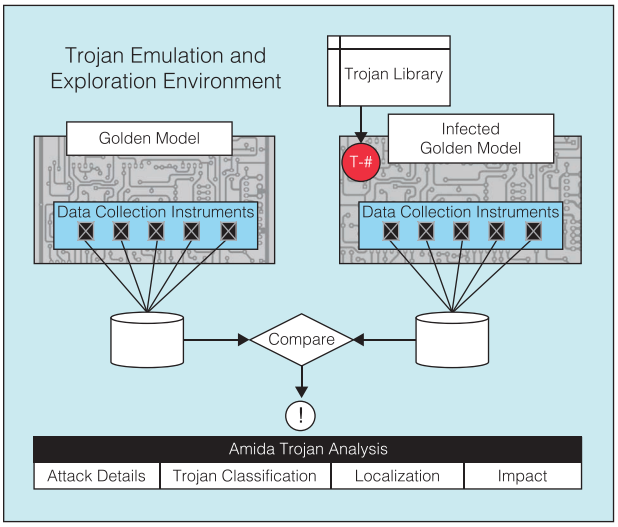

Fig. 2. Method for TE3© environment where compare is a machine learning tool.

are the side-channel or parametric effects that use temperature, voltage, power, and other non-digital effects to either trigger a Trojan payload or as a Trojan payload (for example, leaking data by modulating a temperature hot spot). Through this type of insidious behavior, a Trojan may surreptitiously exfiltrate or obfuscate critical data without impacting any of the normal circuit operation.

The best detection method is to place embedded instruments within the semiconductor device that can monitor activity and circuit behavior and can identify when suspect anomalous behavior occurs (either pre-activation or post-activation). In an ideal sense, one goal is to detect "pre-trigger" impacts, such as excess power consumption or spurious emissions or Trojan setup behavior. Circuit hardening may then quarantine, countermeasure, or even obviate Trojan activity deemed critical prior to (defending) or at the time the Trojan is activated (recovery).

From one simplified point of view, there are two broad classes of embedded Trojan detection instruments: embedded instruments that are designed specifically to target historically known Trojan triggers or payloads (known as binary detection); and embedded instruments that are designed to detect the "unknown" Trojan. Detecting unknown Trojans or a new Trojan effect that is not anticipated is accomplished by learning and monitoring the normal operative behavior of the circuit and identifying any anomalous deviation. There are many questions concerning the use of embedded instruments for hardening any given IC design: what instruments, how many, how complex, where placed, what area/power cost and what coverage is needed? This requires an exploration of the wide variety of instruments in the context of the various Trojan attacks.

Our Approach to Hardware Assurance

Toward the investigations described in this article, we employed a methodology and a novel Trojan Emulation and Exploration Environment or TE3© tool ([3]) that uses direct on-hardware applied research to emulate and assess a pristine system within a sandbox FPGA using embedded instruments. Then, by insertion of a variety of hardware Trojans and conducting a training session using a supervised machine learning tool, we can meet detection and identification goals involving both digital behavior and analog/parametric/side-channel effects (Fig. 2).

For the investigations of interest to this article, a Golden Circuit implementing a simple secure messaging facility was emulated. During the project, ten significant and relevant Trojans were inserted, several each from the Leaker, Behavior Modifier, and Reliability Impact categories; and across the spectrum of digital, analog, parametric and environmental triggers and payloads. Within this framework a number of Trojan cases were explored, as summarized in Table 1.

For initial development of the TE3©, we created a library of data collection instruments aimed at straightforward Trojan detection: targeted binary instruments that could directly detect Trojan activity (extra hidden logic, trigger, payload) and "historically known" attacks (however, binary instruments are costly at one instrument per anticipated Trojan). As the project progressed, more sophisticated side-channel instruments were also created and added to the library of instruments with the goal of migrating toward a machine learning process that could be used to detect the "unknown" Trojan (a new Trojan effect that has no historical reference). This required a change in the philosophy of data collection instruments to instruments with the ability to "learn the Golden Model/Circuit" in a normal operation sense and using machine learning to identify and classify any "anomalous" behavior(s). The security data collection instruments investigated during the proof-of-concept phase are presented in Table 2.

All implemented instruments reside within an IJTAG IEEE Std 1687-2014 [6] scan network and follow the generic IJTAG test data register (TDR) pattern illustrated in Fig. 3.

The 1687 IJTAG Standard describes the use of embedded instruments in conjunction with an IEEE Std 1149.1-2013 Test Access Port (TAP) and JTAG TAP Controller [7]. The two standards, when coupled together, create an efficient embedded instrument access architecture. In our architecture, each instrument is sequestered behind a segment insertion bit (SIB) within an IJTAG scan path to allow the instruments to be turned on and off and removed from the active serial scan

Authorized licensed use limited to: Peter Levin. Downloaded on September 15,2020 at 18:50:33 UTC from IEEE Xplore. Restrictions apply.

Table 1 – Trojans evaluated				
Trojan Name	**Trigger Type**	**Trigger**	**Payload Type**	**Payload**
GhostPort	Data Comparator	Text: "take it"	Behavior Modifier	Take Over
BitFlipper	Timer Comparator	Timer	Reliability Impact	Error Inject
PinCast	State Comparator	Trigger by pin state	Data Leaker	Leaks on unused pin
ServiceBlock	Data Comparator	Text includes "urgent"	Behavior Modifier	Blocks Receiver
FakeFault	Message Counter	Message Counter	Reliability Impact	Stuck-at ASCII bit
IR-PinCast	State Comparator	Turn on IR LED	Data Leaker	Leaks on IR LED
IllegalJTAG	IR Encoding	IR has illegal instruction	Data Leaker	Bad JTAG Instruction
ServiceBlock-AT	Temperature Change	Temperature Drop	Behavior Modifier	Blocks Receiver
VM-PinCast	State Comparator	Trigger by pin state	Data Leaker	Leaks on analog pin
VM-PinCast-AT	Temperature Change	Temperature Drop	Data Leaker	Leaks on analog pin

Table 2 – Data collection instruments evaluated			
Instrument Name	**Instrument Category**	**Attack Detection**	**Description**
Unused Pin Activity Monitor	Active Unused Elements	Digital Data Leaker	Detection of activity on unused digital pins
Timer Monitor	Time-Framing	Behavior Modifier or Reliability Impact	Operation time learning and timeout generation
JTAG Instruction Verifier	Back-Door Operations	Behavior Modifier	Verification that JTAG is used properly
Unused IR-Port Activity Monitor	Active Unused Elements	Side-Channel Data Leaker	Detection of activity on unused IR port
Temperature Monitor	Side-Channel Effects	Behavior Modifier or Reliability Impact or Data Leaker	Detection of use of temperature as a trigger condition
Voltage Monitor	Side-Channel Effects	Behavior Modifier or Reliability Impact or Data Leaker	Detection of use of simple voltage modulation as a data exfiltration method
Intermingled Voltage Monitor	Side-Channel Effects	Behavior Modifier or Reliability Impact or Data Leaker	Detection of use of complex voltage modulations on existing analog signals

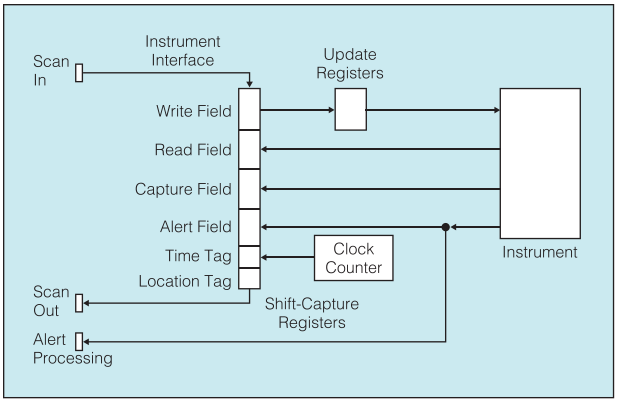

Fig. 3. Generic Trojan detection/data collection instrument that allows read, write, capture of instrument response, a time tag, a location tag, and captures as well as asynchronously (real-time) passes on an alert signal.

chain as needed to manage power and operational noise and to adjust data collection latency and data bandwidth by managing the scan chain length. To extract data from the emulation board and the embedded monitors, we used the ASSET ScanWorks® software program.

Embedded Instrumentation for Hardware Assurance

All security instruments may take on the form of simple digital or analog circuits or more complex elements that process or accumulate the variables being monitored. The ultimate goal of instrumentation in a hardware security sense, then, is to monitor activity and collect data on the well-being of a given Golden Circuit to identify any anomalous activity. The data collected may further be organized into variables and analyzed as to the

Authorized licensed use limited to: Peter Levin. Downloaded on September 15,2020 at 18:50:33 UTC from IEEE Xplore. Restrictions apply.

root cause of the anomalous behaviors—some of which may be identified as hardware malfeasance (e.g., hardware Trojans and counterfeit elements).

For Trojan detection, instruments must be brought to bear that can identify when anomalous behaviors are evident—such as the delivery of a Trojan payload that would modify the operation of a device, would impact the lifetime of a device or would leak sensitive data (exfiltrate) from a device in a stealthy manner (as stated earlier, aligned with one or more of the three canonical forms: Leaker, Behavior Modifier, Reliability Impact). For Trojan detection, and further, for Trojan identification, location and operational impact assessment, the required instruments fall into several fundamental categories that are related to the type of vulnerability that exists within the Golden Circuit. A non-comprehensive list of these categories is: active unused elements; time-framed operations; side-channel effects; back-door operations; and hardware assertions. These categories and the instruments implemented per our proof-of-concept investigation are described further below.

Detect Activity at Unused Circuit Elements

Schedule tolerance drives the use of pre-existing intellectual property (IP) cores that may include logic functions that will not be used by the end device—for example, a hardware macro of an embedded microcontroller or a high-speed IO port such as USB. These extra features will remain in the final design under some design circumstances (hard macro, legal agreement). IP Cores that contain design elements that are not required in the final design provide back-door access into and out of the IP Core logic [8] that can be used to infiltrate the IC design with code or operations, exfiltrate data or operation sequences from the IC design, or can be used to disrupt the normal behavior of the final design (e.g., Take-Over, Kill-Switch or Error-Injection). A nefarious individual or organization can easily create a Trojan by making an illicit connection to one or more of these unused ports or functional elements. The design verification and manufacturing test flows verify the "active portion" of the final design (to manage test cost and vector volume), verification that the unused elements will remain quiescent (unused) is often missed.

The types of hardware assurance instruments needed, in this case, are "unused element activity monitors" that monitor pins, ports and functions to see if they ever become active (use power or process data or support logical operations). In our investigations to date, "unused element activity monitors" were implemented as edge-detect and level-detect monitors associated with unused digital pins on the hardware design.

Enforce Bounds for Time-framed Operations

Another type of security monitoring requires learning the time involved for normal operations in a Golden Circuit. This type of monitoring can be termed "time-framing." In essence, an operation to be monitored needs to support an "initiation point" and a clear "ending point." In our proof-of-concept investigation, the Golden Circuit implemented a secure messaging facility where the "start of message" (SOM) packet reset and initiated a counter, and then the "end of message" (EOM) packet stopped the counter and stored the count value. If the EOM is delayed or does not arrive, then the counter will generate an "over-limit" alert. A goal of this particular "timer instrument" is to learn "elapsed message times" under different conditions from start to finish and to feed these numbers to a machine learning algorithm that would provide a "min" and a "max" value for message operations with an alert generated if this range is violated. Time-framing can be applied, for example, to hardware operations such as mathematical calculations (adders, dividers, multipliers, integrators, differentiators), or state transitions (state-machines and sequencers), memory or storage operations (writing, reading from memory or register files) and data transactions (data transfers internal to the IC, data transfers between external ports of the IC).

Check for Unexpected Side-channel Effects

One common issue concerning physical hardware as compared to a simulation model is that the simulation environment focuses on the digital or Boolean results, whereas hardware actually supports non-digital effects such as current, voltage, power modulations, temperature and power supply manipulations and analog signal corruption. In addition, the environment surrounding a physical electronic system or device is also subject to modifications and manipulations (note: not a complete list)—air pressure, frequency/intensity of light, type of atmosphere (e.g., nitrogen versus air, humidity level), background sound level or directed sound waves, physical location (GPS coordinates) or altitude, and many forms of emissions (light, radio-frequency, sound, heat, vibration, etc.). All of these types of considerations are lumped into the category of side-channel effects. Side-channel effects can be used as triggers or can be manipulated as the Trojan payload.

The easiest side-channel effects to include nefariously within a device are the use of temperature or radio-frequency (RF) as a trigger or as a payload (to exfiltrate data); and the use of high temperature to impact the reliability (lifetime) of a device. For example, a simple ring-oscillator can be tied to an internal circuit node such as the connection between two gates, and whenever that circuit node is at logic-1 the ring-oscillator becomes active and produces a hot spot. To view the changes between 1 and 0 traffic, a coolant spray could be used to lower the temperature of the device where the difference between a logic 1 and logic 0 can easily be seen using an inexpensive thermal camera.

One of the most dangerous side-channel Trojans is the use of temperature as a reliability impact. Any trigger could initiate a Trojan payload that would cause excessive switching, or operation that exceeds the thermal limits of the device (such as simultaneously activating multiple memory BISTs). This could cause the wire bonds or the silicon itself to melt.

Temperature monitors are useful to assess various locations within the device and to provide direct alerts or internal countermeasures if the temperature shows anomalous behavior

Authorized licensed use limited to: Peter Levin. Downloaded on September 15,2020 at 18:50:33 UTC from IEEE Xplore. Restrictions apply.

(e.g., such as that of being sprayed with coolant, being kept in an extremely cold room, or rapid rises of temperature associated with extreme activity or switching). A lesson learned in the proof-of-concept investigation was that the side-channel effects and the associated detection instruments are limited to the capabilities of the hardware emulation system used (the Trojans are emulated using programmable logic, not actual Trojans). A real physical side-channel Trojan or trigger may be programmed within a device at any location using any of several physical techniques; however, the reality is that analog/parametric triggers and Trojans represent physically large "hard to hide" circuit elements. Therefore, the most-likely scenario for nefarious individuals applying most side-channel effects is to reuse existing analog circuitry such as DACs, ADC, DSPs, and similar analog-mixed-signal (AMS) elements.

Guard Against Back-door Operations

The functional data pathways on currently available devices have a number of data checks associated with them, and they routinely encrypt the data, so there is a measure of applied security. Similarly, the command pathways for most chips have authorization steps and requested operation checking (for example, supervisor mode in the microprocessors). There are still large families of ports in today's devices that do not have any security measures associated with them: the test and debug ports. For example, the JTAG, I2C and SPI are all serial ports and they provide extensive access to the inner workings and configuration settings of many chips— especially the larger digital chips such as microprocessors (CPUs) and graphic processing units (GPUs) that make up the heart of many systems. One method of managing these utility ports is to disable them after manufacturing test. There have been many device purchasers who have complained loudly that a major methodology to assist in board integration has been removed by disabling the JTAG and its boundary-scan capability. In other cases, such as for SPI Flash chips or SPI DACs, there can be no disabling the SPI port since it is required as the preferred method used to configure and set up the memory or data conversion device.

If there is not an effort to provide locks and keys or challenge-response hardware to limit the access to these back-door ports, then instruments can be used to monitor the features of the back-door ports to provide alerts if they are being used in an illicit, nefarious, or simply incorrect manner. This generally requires identifying allowable modes or allowable physical access to internal features and then monitoring if the current operation exceeds these restrictions (such as our "illegal JTAG" instrument that monitored the instruction register for an allowable instruction such as BYPASS or the IJTAG AccessLink). Similar instruments can be used to verify that the test data registers (TDRs) that may be activated under various instructions are in fact accessing the correct logic they are intended to monitor (read) or control (write).

Raise Hardware Assertions

A final category of instruments to be discussed in this article is for those involved with the ongoing real-time verification of device operations, hardware assertions. These are instruments that are also commonly used for "verification" of the design. Assertions take on many forms: some are simple value checkers (for example, a register is allowed to contain values from 0x000F to 0x0FFF and so comparators can be used to verify that the value within the register never exceeds the allowed values); and some are sequence verifiers (for example, making sure a state machine that is meant to go from state-1 to state-2 to state-3 never transitions directly from state-1 to state-3). Many of these types of assertions are built into devices and they provide an alert signal or an interrupt when a unit gets "out of synch" with the current operations. In many cases, these are the types of instruments that would be required by "machine learning," to assess what is normal behavior during the learning phase of data collection/analysis or to identify abnormal behavior during the application or predictive phase (where learning labels are not active as part of data collection).

Progress, Conclusions and Future Work

The Phase I effort has completed, and all ten Trojans and all seven instrument types described in this article have been applied and evaluated. The emulation of a real system within an FPGA proved that the instruments and their data clearly fell into two categories: targeted Trojan data that positively indicated a detection from binary instruments; and training data suitable for a machine learning system to identify a detection from generic data. The more cost-effective instruments were the ones conducting machine learning as we saw that targeted detection instruments did require one instrument per anticipated Trojan. Another category of cost-effective instruments were ones that were already included for other purposes: debug and trace instruments, test instruments, yield-analysis instruments and functional instruments that naturally monitor items such as clocks stability, operating temperature, and time-framing of internal operations. An effort should be made by design organizations to include these into the cybersecurity framework as data collection instruments.

The data collection methodology used for the proof-of-concept phase of this project could be considered "on demand" and was controlled by the external ASSET ScanWorks® software in conjunction with the operator or investigator of the hardware. The intent of the exploration phase was to also emulate such a system to establish the data collection parameters, asking questions such as: "How often is data collection required?"; "What is the latency between data collection and detection or action?"; and "What fields or variables must the data collection instruments support to allow background operation and provide real-time or on-line value?" The result of the investigation was to create an instrument network that did not interfere with normal operative behavior and to provide an alert field in all of the data collection instruments—a path where an instrument generated alert would go directly to an alert processing function without waiting for data collection to capture the alert value within the IJTAG data collection TDR. The answer to creating the optimal "real-time" system is left as future work as we are now progressing from a proof-of-concept

Authorized licensed use limited to: Peter Levin. Downloaded on September 15,2020 at 18:50:33 UTC from IEEE Xplore. Restrictions apply.

to a product prototype stage. Future work includes a new more complex Golden Model and the software automation of the vulnerability analysis, Trojan insertion, instrument insertion and the expansion of the machine learning analysis to also recommend countermeasures and recovery techniques.

Acknowledgment

The authors would like to thank their colleagues: Amida: Sergio Orellana, Matt Ritonia, Jackie Medina, and Peter Levin, for their contributions to development of the TE3© concept and methods; and ASSET InterTech: John Akin and Dean Geerdes, for their work on the FPGA-implemented Golden Models / Circuits, IJTAG networks and embedded instrumentation logic.

References

[1] A. Crouch and A. Ley, "A role for embedded instrumentation in real-time hardware assurance and online monitoring against cybersecurity threats," in *Proc. 2019 IEEE AUTOTESTCON*, 2019.

[2] R. Karri, J. Rajendran, K. Rosenfeld and M. Tehranipoor, "Trustworthy hardware: identifying and classifying hardware Trojans," *Computer*, vol. 43 no. 10, pp. 39-46, Oct. 2010.

[3] A. Crouch, E. Hunter, and P. Levin, "Enabling hardware Trojan detection and prevention through emulation," in *Proc. 2018 IEEE Int. Symp. Technologies for Homeland Security (HST)*, 2018.

[4] V. Jyothi, P. Krishnamurthy, F. Khorrami, and R. Karri, "TAINT: tool for automated insertion of Trojans," *IEEE Int. Conf. Computer Design (ICCD)*, 2017.

[5] W. Hu, B. Mao, J. Oberg, and R. Kastner, "Detecting hardware Trojans with gate-level information-flow tracking," *Computer*, vol. 49 no. 8, pp. 44-52, Aug. 2016.

[6] *IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device*, IEEE Std 1687-2014, Dec. 2014.

[7] *IEEE Standard for Test Access Port and Boundary-Scan Architecture*, IEEE Std 1149.1-2013, May 2013.

[8] X. Zhang and M. Tehranipoor, "Case study: detecting hardware Trojans in third-party digital IP cores," in *Proc. 2011 IEEE Int. Symp. Hardware-Oriented Security and Trust*, Jun. 2011.

Alfred L. Crouch (alfred@amida.com) is Director of Hardware Engineering at Amida Technology Solutions in Austin, Texas. He is a widely published author in the field of DFT, test automation, test and test/debug security. Al participates in IEEE standards projects in these fields, including, IEEE 1838, 1687 and P1687.1. He holds degrees in electrical engineering from University of Kentucky, Lexington, Kentucky.

Adam W. Ley (aley@asset-intertech.com) is VP, Chief Technologist at ASSET InterTech, Inc. in Richardson, Texas, a leading supplier of solutions for non-intrusive test, validation and debug. Adam participates in IEEE 1149.1 and related standards for test. Previously, Adam was at Texas Instruments in Sherman, Texas. Adam earned the BSEE degree from Oklahoma State University, Stillwater, Oklahoma, in 1986.

Authorized licensed use limited to: Peter Levin. Downloaded on September 15,2020 at 18:50:33 UTC from IEEE Xplore. Restrictions apply.