JTAG-Based Robust PCB Authentication for Protection Against Counterfeiting Attacks

Andrew Hennessy
Department of EECS
Case Western Reserve Univ.
Cleveland, OH, 44106
e-mail: ajb200@case.edu

Yu Zheng
Department of EECS
Case Western Reserve Univ.
Cleveland, OH, 44106
e-mail: yxz402@case.edu

Swarup Bhunia
Department of ECE
Univ. of Florida
Gainesville, FL, 32611
e-mail: swarup@ece.ufl.edu

Abstract— A Printed Circuit Board (PCB) provides the backbone for the interconnection of diverse electronic components into an electronic system. Unfortunately, the increased use of untrusted third-party PCB design/fabrication facilities and the long, distributed, supply chain of a PCB makes it extremely vulnerable to variety of integrity violation attacks, primarily different forms of counterfeiting, including cloning and recycling. In this paper, we propose a novel low-overhead and robust method to authenticate PCBs that utilizes an existing industry standard — IEEE 1149.1 or JTAG test infrastructure to extract high-quality signatures with high entropy. Since the signature encapsulates the intrinsic properties of the PCB components, it can be effectively used to identify malicious modifications of PCB components (e.g. replacing a chip) in a supply chain or during field operation. Measurement results with 30 custom fabricated test boards are promising in terms of uniqueness (48.34% inter-PCB hamming distance) and robustness (1.22% intra-PCB Hamming Distance) of 128-bit authentication signature.

I. Introduction

Printed Circuit Boards (PCBs) are used extensively in nearly every electronic system to provide mechanical support and electrical connections to the components [1]. The annual world-wide revenue associated with the production and support of PCBs is estimated to 59.1 billion dollars in 2011. State-of-the-art PCBs often integrate number of integrated circuits (ICs) with high pin complexity into a miniature layout. Moreover, a survey shows that 14 percent of today's PCBs are currently operating in the 1-10GHz frequency range to support high-speed data communication [6]. As a result, the complexity and cost of PCB design is rising rapidly. With increasing complexity of PCBs including hidden vias and multiple layers (6–20), system integrators are relying on an increasing number of third-party vendors for designing, producing, and acquiring PCBs for different electronic systems.

The long, distributed, supply chain for PCBs makes them vulnerable to counterfeiting attacks of various types, including cloning, overproduction and recycling scavenged boards [2]. Conventional hardware security analysis and countermeasures predominantly target ICs. These techniques are not readily applicable at the PCB level. For example, existing anti-counterfeiting solutions for ICs such as Physical Unclonable Functions (PUF) [9] [4], or aging sensors (to protect against recycled or reclaimed chips) cannot be easily extended to the PCB level. These methods would either be ineffective (e.g., a separate component in PCB for signature generation which can be easily removed or replaced by an adversary) and/or would incur significant cost and overhead associated with incorporating and validating dedicated hardware resources.

Counterfeit PCBs in a supply chain can come various sources, including outright clones, overproduction and recycling [10], [13]. The adversary can clone the PCB layout through reverse engineering [13] and subsequently sell the counterfeit PCB as an authentic product, potentially with built-in Hardware Trojans. A PCB can readily be reverse engineered through optical means, such as x-ray analysis. Another type of attack is that if the PCB manufacturing partner is not trusted they could produce additional PCBs above what a contract was signed for, selling the overproduced PCBs against the wishes of their clients. It is often true that these counterfeit PCBs are of a lower quality than the original ones, resulting in reliability and performance issues, stemming from a lack of Quality Control. The final type of illicit PCBs is simply recycling them — selling a used PCB as a new PCB. All three methods, if combined, could be very profitable to competitors due to a loss of reputation stemming from the lower quality products seemingly being sold under a trusted brand name. With an increase in the vulnerability of PCBs against counterfeiting attacks and a rise in the reporting of incidents, there is a critical need to reliably verify the authenticity of PCBs before they are deployed in the wild.

Current industrial practices do not provide an adequate solution to effectively protect PCBs against the aforementioned diverse counterfeit attacks. One of the existing practices in industry is to store a 64-bit (or longer) unique ID in non-volatile memory for PCB authentication [5]. Moreover, a SHA-1 signature ensures the integrity of the critical signature once programmed. Murata Americas has introduced a Radio Frequency Identification Device (RFID) tag into each PCB for unique authentication, which has the advantage of being completely wireless in nature [8]. The issue with this system is that it increases the cost and the design complexity of a PCB. Furthermore, it cannot be easily integrated into legacy PCBs. PCB design solutions to prevent reverse engineering through obfuscation have been explored [7] [13]. However, such design solutions cannot protect against all forms of counterfeiting and are not applicable to existing PCBs. A recent PCB authentication approach creates hard-to-clone signature from PCB trace impedances [10]. It requires expensive test equipment for signature extraction and is limited by availability of adequate number of measurable traces. Moreover, it cannot be effective for remote authentication.

The authentication of individual ICs has been investigated to great lengths through various methods. Physical Unclonable Functions (PUF) play an important role in IC authentication [9]. However, PUFs often require embedding a separate structure in ICs during the design phase. An SRAM-PUF exploits a common existing on-chip structure and can generate a unique signature by utilizing the power-up state of memory [4]. However, these approaches cannot be readily employed at PCB level, as discussed earlier. The active metering approach in [11] that provides remote control of a design for authentication has been explored to prevent counterfeiting attacks for ICs, this system, like the aforementioned RFID-based one requires effort on the part of the circuit designer as well as an additional per unit cost.

In this paper, we present a novel counterfeit PCB detection approach

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

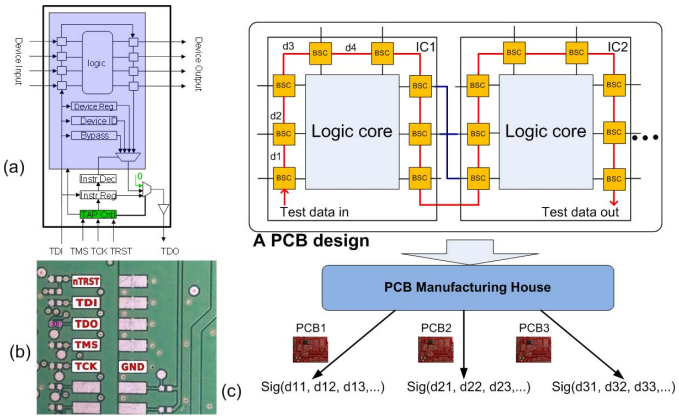

Fig. 1: (a) JTAG interface for a chip; (b) JTAG ports from a commercial PCB; and (c) Illustration of JTAG-based authentication on a PCB.

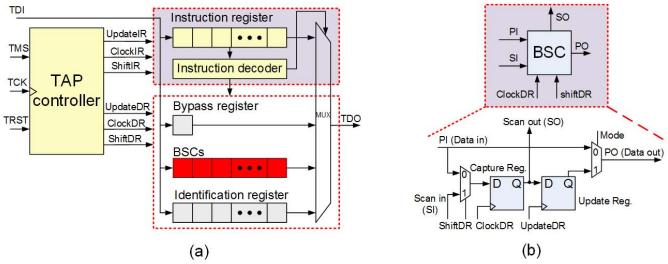

Fig. 2: Typical JTAG based boundary scan architecture in a PCB; (b) Structure of a Boundary Scan Cell.

that can effectively address different forms of counterfeiting attacks in PCBs and thus mitigate a major security risk associated with the potentially untrusted PCB supply chain. The proposed approach utilizes the Boundary Scan Chain Architecture (BSA) inherent in most ICs — a prevalent design-for-test (DFT) structure used by majority of PCBs used today. We use the industry-standard BSA design based on the Joint Test Action Group (JTAG), or IEEE 1149.1, protocol to create a unique signature from each PCB for authentication. BSA is a widely accepted DFT structure that allows for the testing and subsequent debugging of potentially every signal path in a PCB.

Fig. 1 (a) illustrates common boundary scan interface for an IC while Fig. 1 (b) shows the common boundary scan ports from a commercial off-the-shelf PCB. The overall approach for using boundary scan for signature generation is illustrated in Fig. 1 (c). It shows the connection of Boundary Scan Cell (BSC) during the PCB testing process, in which the system designer can shift of capture information from each input pin as well as the core logic of the IC. Let us take three BSC paths with the nominal delay of d_1, d_2 and d_3, respectfully, as an example. During the manufacturing process of both the IC and the PCB, subtle variations in the processes can induce a variable delay in the scan path from their nominal values. As shown in Fig. 1, we can measure the delay of the BSC paths on the PCB and generate a unique signature for authentication.

The remainder of the paper is organized as follows. Section II provides brief description on JTAG infrastructure in PCB. Section III describes the JTAG-based PCB authentication procedure in detail. Section IV and Section V present the experimental evaluation setup and measurement results. Section VI concludes the paper.

II. Background

JTAG offers an efficient boundary test structure for all PCBs, especially those tight on space (e.g. cell phones and laptops). It provides capabilities to test the PCB *in vivo*, without any modification

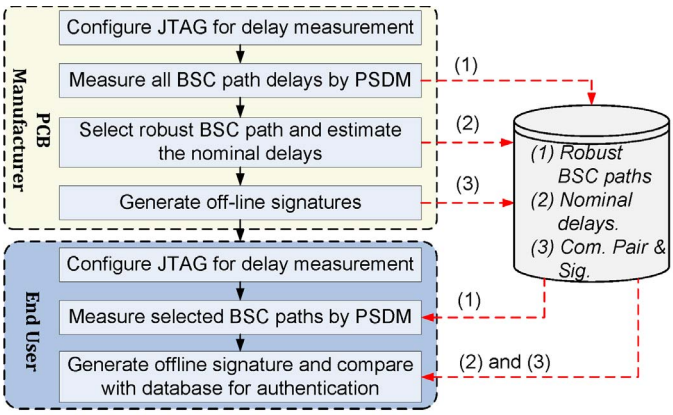

Fig. 3: Major steps in the proposed JTAG-based PCB authentication process.

required for the ICs and the only requirement for the PCB is six exposed connections. A typical implementation of JTAG is shown in Fig. 2(a) with four input pins ('TCK', 'TMS', 'TDI', 'TRST'), one output pin ('TDO'), and a ground connection ('GND'). Typical components of a JTAG implementation include the Test Access Port (TAP) Controller, the instruction register, the instruction decoder, the boundary-scan register, the bypass register and finally the identification register. The TAP controller can be modeled as a finite state machine, where state changes are triggered by the rising edge of the clock signal, 'TCK', and the next state is determined by the status of the signal 'TMS'. The outputs of the TAP include the clocks and control signals for each register. By properly navigating the TAP, instructions such as 'SAMPLE', 'PRELOAD' and 'EXTEST' can be stored in the instruction register (See Fig. 4). Only one instruction can be active at a time in the TAP. Some instructions are mandatory, such as 'EXTEXT', while others, such as 'IDCODE' are optional extensions of the standard [3].

In a JTAG system all of the BSCs are connected to one another in a manor similar to a shift register to form the Boundary Scan Register, a core part of the BSA. A BSC can force a signal onto an output pin, capture data from an input pin or modify the core internal logic of an IC. The basic structure can be seen in Fig. 2 (b). It includes the capture register and update register, triggered by separate clocks which are both derived from the 'TCK' clock fed into the TAP. A test vector can be scanned into each BSC by the port 'SI' and shifted out through the port 'SO'. The capture registers can access core data or I/O pins via a multiplexer controlled by 'ShiftDR'; the update registers provide the data to external through I/O pins. Signals 'Mode' and 'ShiftDR' are generated by the decoding of instruction register. Generally, a BSC can work in four different modes: (1) in normal mode, the data of 'PI' is passed directly to 'PO', (2) in update mode, the content of the update register is passed through to 'PO', (3) in capture mode, the signal of 'PI' is routed to the input of capture register, which is captured in the next 'ClockDR' cycle. 'ClockDR' is a derivative clock of 'TCK' by TAP controller, (4) in shift mode, the 'SO' of a capture register is passed to the 'SI' of the adjacent capture register via a hard-wired path [3].

III. JTAG-based PCB Authentication

As shown in Fig. 3, the proposed JTAG-based PCB authentication is separated into two stages. In the first stage a PCB manufacturer configures the JTAG device(s) on a PCB into an appropriate state needed to measure the delay of the BSC paths on all authentic PCBs. Then the robust BSC paths are selected with an estimate of the nominal delays for each path. Afterwards the signatures are produced off-line. The locations of the BSC paths, the nominal delays as well as signatures (including the comparison pairs defined in Section III-D)

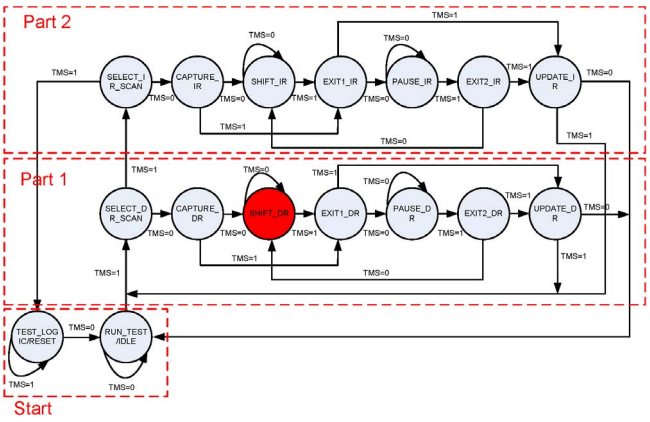

Fig. 4: The state machine of TAP controller.

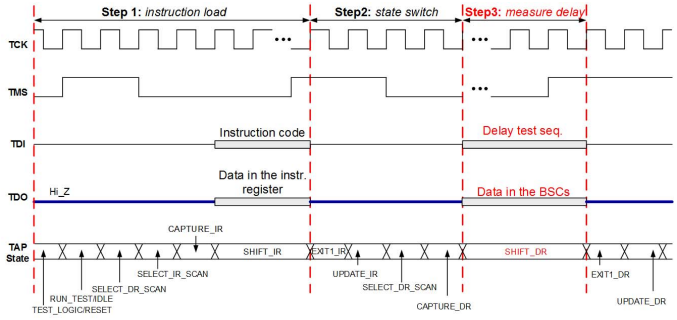

Fig. 5: The timing of the JTAG connections for the proposed authentication.

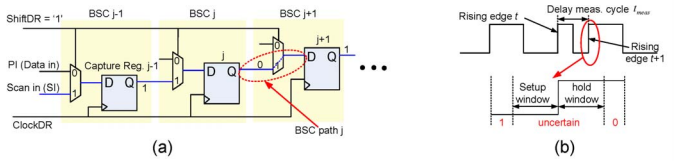

Fig. 6: (a) The connection of BSCs in state 'SHIFT_DR'; and (b) creating a clock pulse with period t_{meas} for precise scan path delay measurement.

are stored in a central database. In the second stage, an end user configures the JTAG on the suspected PCB in the same way and measure the delays of the selected BSC paths. Then the signature is computed, which is compared with the signatures stored in the database. The PCB is judged as counterfeit if the produced signature by the end user is not found in the database. In the following subsections, each step in Fig. 3 will be explained in more depth.

A. The State of the TAP Controller

In order to measure the delay of the selected BSC paths, the TAP controller needs to be forced into the proper state, or 'ShiftDR' in Fig. 2(b) is set to '1'. This configures the system to connect the 'SI' port to the input of the capture register. Additionally the proper instruction must be loaded into the Instruction Register to allow the BSCs to be connected in series with one another. Once both of these steps are finished the authentication protocol can proceed.

The state transition diagram of the TAP controller is shown in Fig. 4. The state transition is determined by signal 'TMS' on the rising edge of the clock signal 'TCK'. The machine can be split into two different sections, one dealing with the instruction register and the other interfacing with the data register. In the 'SHIFT_DR' state (highlighted in red in Fig. 4), the test data registers can shift data from the input ('TDI') one stage at a time toward the serial output ('TDO') on the rising edge of 'TCK' [3]. Hence, the test vectors can be scanned sequentially into the capture registers of BSCs.

The JTAG specification includes a handful of mandatory instructions, including: 'EXTEST', used to test the connections between two ICs on a PCB; 'SAMPLE', used to take a snapshot of the nominal operation of the IC in question; and 'PRELOAD', used to serially shift data into the BSC. All of these instructions shift data serially through the BSCs, using the 'PI' and 'PO' ports in Fig. 2(b), ultimately shifting data through the chain using the external 'TDI' and 'TDO' connections. As a result, as long as the TAP controller is in the 'SHIFT_DR' state and one of the aforementioned instructions is loaded into the instruction register the proposed delay measurements of the BSC paths can be completed.

B. Delay Measurement of BSC Paths

The timing of input and output pins is shown in Fig. 5. Three steps can direct the TAP controller to the state that measures BSC path delay for authentication. The first step is to load 'EXTEST' into instruction register. After reset, the TAP controller is in the state 'TEST_LOGIC' which is kept the same when 'TMS' is kept at '1'. As shown in Fig. 4, in the following five cycles, 'TMS' is forced to '0', '1', '1', '0' and finally '0' to make the state machine go through the 'RUN_TEST', 'SELECT_DR_SCAN', 'SELECT_IR_SCAN', 'CAPTURE_IR' and 'SHIFT_IR' states, respectively. In the state 'SHIFT_IR', the binary code of 'EXTEST' can be shifted into the Instruction Register through the 'TDI' connection. The exact code needed to shift 'EXTEST' into the Instruction Register is dependent on the ICs used on the PCB. After decoding, the boundary-scan register is placed between the 'TDI' and 'TDO' connections. The second step is to switch the state of TAP controller from 'SHIFT_IR' to 'SHIFT_DR'. Similarly, based on Fig. 4, 'TMS' is sequentially changed to '1', '1', '1', '0' and finally '0' in five clock cycles. At the end of this sequence the TAP controller is in the 'SHIFT_DR' state, which begins the third step to measure the BSC path delays.

After being setup, the output of BSC register $j-1$ is connected with input of BSC register j, as can be seen in Fig. 6(a). Hence, the BSCs are connected as a shift register, which can be used by the Parallel Scan Delay Measurement (PSDM) algorithm to measure the delay of each of the BSC paths. In this algorithm, register j is initialized to '0'. It is changed to '1' on the rising edge t of the clock signal 'ClockDR', which is derived from the external 'TCK' clock signal, to generate a 0→1 transition on the BSC path j. After an interval of t_{meas}, the rising edge $t+1$ in the clock signal 'ClockDR' is seen in Fig. 6 (b).

The Flip-Flop in the BSC register $j+1$ of PCB i outputs:

$$O_{i,j+1} = \begin{cases} 1 & t_{meas} \geq d_{i,j} \\ 0 & otherwise \end{cases} \quad (1)$$

In following equation, the resolution of t_{meas} is Δt and the initial period of measurement is t_{init}. k is called the switch point of path j, if $t_{meas} = t_{init} + k\Delta t$ and $t_{init} + (k+1)\Delta t$ lead to $O_{i,j+1} = 0$ and 1 respectively. Using this, the delay of path j is estimated as:

$$d_{i,j} \doteq t_{init} + (k+k+1)\Delta t/2 = t_{init} + (k+0.5)\Delta t \quad (2)$$

Note that the 0→1 transition occurs on the selected BSC paths to measure the delay in parallel, which allows for testing of half of all BSC paths in a single iteration, resulting in only two iterations needed to test every single BSC path. The measurement can be repeated to average out the effect of environmental noise, such as temperature

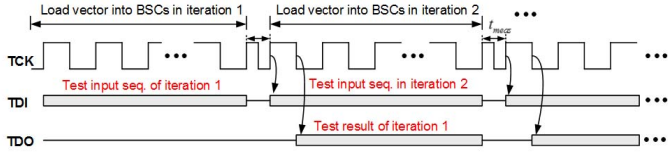

Fig. 7: The timing of test signals for delay measurement in two successive iterations.

and power supply fluctuations. The procedure of PSDM is shown in Algorithm A, including multiple iterations.

Algorithm A: PSDM Procedure [12]
Input: Location of N_{path} BSC paths.
Initialization: $t_{meas} \leftarrow t_{init}$, $sw\_num \leftarrow 0$ and $k \leftarrow 0$
while $(sw\_num < N_{path})$
Generate $0 \rightarrow 1$ transition on all selected scan paths after the rising edge t.
Produce the rising edge $t+1$ after t_{meas}.
Identify sw new switch points.
$sw\_num \leftarrow sw\_num + sw$
$k \leftarrow k+1$
end of while
Output: Path delays as (2)

In the beginning, t_{init} should be less than all the BSC path delays to be measured in Algorithm A. As a result, the number of detected switch points, denoted by sw_num, should be initialized to zero. In each iteration, the delay-measurement cycle of t_{meas} identifies sw switch points among N_{path} paths. The switch-point number sw_num is increased by sw. If sw_num is less than N_{path}, it goes into a subsequent iteration with $k \leftarrow k+1$ and $t_{meas} \leftarrow t_{meas} + \Delta t$; otherwise Algorithm A ends and computes the delays as (2). Fig. 7 shows the particular timing of pins when applying Algorithm A in measuring BSC path delays. 'TDI' and 'TDO' are respectively triggered by the rising and falling edge of 'TCK'. The adjacent two iterations can be interleaved with each other to reduce the test time. For example, loading the test vector of iteration 2 through 'TDI' can be done with storing the test results of iteration 1 from 'TDO' simultaneously. This allows for rapid testing of all BSC paths.

C. Identification of Robust BSC Path

Algorithm B: Robust BSC Path Identification
Input: N_{pcb} PCBs and N_{path} BSC paths.
Initialization: $(\alpha)_{1\times N_{path}} \leftarrow (0)_{1\times N_{path}}$, the measured repetition time n
For $i = 1$ to N_{pcb}
For $j = 1$ to N_{path}
For $k = 1$ to 2
Measure path k by n times.
Decode rep. code $(n,1)$ as $d_{i,j,k}$.
EndFor
$d_{i,j} \leftarrow d_{i,j,1}$
If $d_{i,j,1} \neq d_{i,j,2}$
$\alpha(j) \leftarrow \alpha(j)+1$
EndIf
EndFor
EndFor
Output: $\{d_{i,j}\}$ and robust indicator α

A PCB manufacturer measures all BSC path delays for all authentic PCBs. Experimental results show that the delay value of some switch points is unstable, changing in value due to things such as temperature and voltage level. As a result, for some BSC paths, the value resulting from (1) can change due to environmental factors. Consequently, we need to identify robust BSC paths that, even when factoring in environmental noises, do not change value. This is accomplished by the Robust BSC Path Identification Algorithm, or Algorithm B.

In Algorithm B, the delay of the jth BSC path is measured n times according to Algorithm A. Then its delay $d_{i,j,1}$ (regarded as $d_{i,j}$) is obtained by decoding a $(n,1)$ repetition code. For example, if $n=7$ with the delay vector $(1,2,1,2,2,2,2)$, the delay of path j is decoded as 2, since it has the largest probability of occurrence. The procedure is repeated one more time and we obtain $d_{i,j,2}$. If $d_{i,j,1} \neq d_{i,j,2}$, it means that path j is not stable in the PCB i. Hence, the robust indicator α is updated as $\alpha(j) \leftarrow \alpha(j)+1$. When Algorithm B is completed on all the PCBs, we can obtain the delay matrix on each BSC path as $\{d_{i,j}\}$, as well as the vector α. To ensure a stable signature, we only select path j with small value $\alpha(j)$ under temperature fluctuations. Based on $\{d_{i,j}\}$, we estimate the nominal delay D_j $(j = 1,2,...,N_{path})$ of path j by averaging as

$$D_j = \sum_{i=1}^{N_{pcb}} d_{i,j}/N_{pcb} \tag{3}$$

The robust path locations and corresponding nominal delays are stored together in the database, which is used by both the PCB manufacturer and the end user to produce, and subsequently verify, the signature.

D. Off-line Signature Generation

The signature is generated off-line after obtaining all the delay values. The PCB manufacturer and end user calculate it based on the nominal value $\{D_j\}$ $(j = 1,2,...N_{path})$. For PCB i $(i = 1,2,...,N_{pcb})$, the delay $d_{i,j}$ of path j is updated as

$$d_{i,j} \leftarrow d_{i,j} - D_j \tag{4}$$

After (4), the mean value of $d_{i,j}$ $(j = 1,2,...,N_{path})$ becomes zero. By removing the average from the calculation, the deviations due to process variation during manufacturing are exaggerated. Hence, when generating signature bit s, paths j and j' are used as follows, specifically where $j \neq j'$:

$$s = \begin{cases} 1 & d_{i,j} > d_{i,j'} \\ 0 & else \end{cases} \tag{5}$$

All of the comparison pairs (j, j') should be stored in the database, along with the signature of each PCB. One reason to choose an off-line signature generation method is that the manufacturer can select the BSC paths on a per-PCB basis to generate high-quality signature for each PCB, since not all PCBs will have the same Robust BSC paths. (5) is similar to the signature generation of RO-PUF. The difference, however, is that RO-PUF requires each ring oscillator to be identically implemented, with the same nominal frequency. However, in our off-line signature generation method, all of the stable BSC paths can be employed as the source of signature generation. As a result, it can incorporate all of the stable BSC paths to generate a high quality signature.

E. Authentication by System Integrator

The authentication of a suspected PCB is completed by end user. First, the location of robust BSC paths are obtained from the database of manufacturer. The actual delay of each selected BSC path in the PCB is measured according to Algorithm A and the stable value is extracted after decoding the $(n,1)$ repetition code. Based on the same database, (4) is carried out to eliminate the affect from nominal delay. Finally, end user produces a signature according to (5) with knowing the selected comparison pairs. The PCB is regarded as authentic if the obtained signature matches the database's.

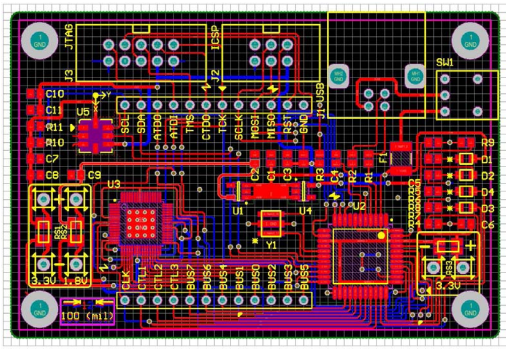

Fig. 8: The fabricated PCB for testing. The main features of the board are the microcontroller, U2, an Atmel ATMEGA16U4 and the CPLD, U3, an Altera MAXV 5M160ZE64C5N. Other features not used for this experiment include a two-axis accelerometer and current shunt resistors for measuring current.

IV. Experimental Setup

A custom PCB was designed and fabricated to evaluate the effectiveness of the proposed authentication method, the layout of which is shown in Fig. 8. The custom PCB includes two commonly used Integrated Circuits (ICs), both with JTAG interfaces. The first IC, referenced as U2 in Fig. 8, is an Atmel ATMEGA16U4. The ATMEGA16U4 is an eight-bit microcontroller capable of operation at a clock frequency of up to 16MHz. It's bigger brother, the ATMEGA32U4 (twice as much Flash and SRAM), is used in the common Arduino Leonardo development board.

The second IC, referenced as U3 in Fig. 8, is an Altera MAXV 5M160ZE64C5N Complex Programmable Logic Device (CPLD), a "C5" speed grade part, which is capable of operating at speeds up to 200MHz. CPLDs, and their bigger brothers, Field-Programmable Gate Arrays (FPGA) are commonly used in many applications to provide customizable hardware at a reasonable price, for example, to encode video. U2 contributes a total of 89 BSCs while U3 contributes the remaining 240 BSCs. The final BSC is the physical trace between U2 and U3. The total amount of BSCs for the custom PCB is 330. The JTAG interface is exposed to the outside world on connector J3, a 10 pin, 2x5, 0.1" pitch header in the standard JTAG pinout. Additionally, there is a two-axis accelerometer and current-shunt resistors on the board, however, neither feature is used for the proposed JTAG-based authentication method.

Our test setup is shown in Fig. 9. A total of thirty custom PCBs were fabricated and tested. A test fixture consisting of an HP 8082A Pulse Generator, a HP 54616C 500MHz oscilloscope, and a custom perf-board setup was used to conduct the experiment. The HP 8082A was modified to allow for digital control of the pulse width. This was accomplished by substituting the vernier potentiometer on the front of the device that controls the pulse width with a sequence of resistors controlled by relays, in a structure resembling an R-2R DAC. This enabled an Arduino Pro board to control the width of the pulse, or t_{meas}. This setup enabled a Δt of 0.02ns. This level of control was needed because the resulting D_j of all BSC paths in the custom PCB was 1.80ns.

V. Results and Discussion

The results of the proposed authentication method are shown in Fig. 10. For each test PCB, we measured the BSC path delays and produced 128 bit authentication signature. Since each authentic PCB generates a unique ID through measuring BSC path delays, the end user can identify a cloned PCB by producing its signature and compare it with the database of manufacturer. To evaluate the uniqueness and robustness of the signatures, we used the conventional Hamming Distance (HD) based metrics—inter- and intra-PCB HD, respectively.

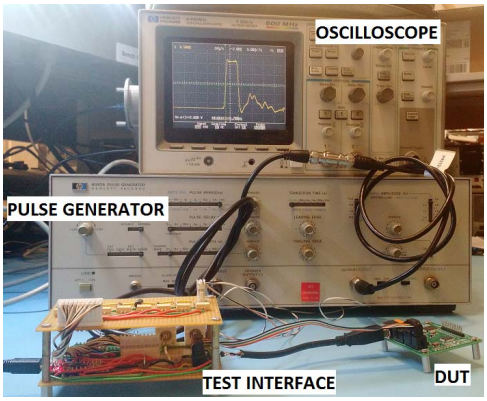

Fig. 9: The measurement setup for the custom PCB analysis. The test interface contains a microcontroller programmed as a JTAG controller, a variable resistor, and interface between the Pulse Generator and the Device-Under-Test (DUT). The Oscilloscope verifies and measures output of Pulse Generator.

Assuming $HD_{p,q}$ stands for the intra HD of all boards between the pth measurement and qth measurement, the average intra-PCB HD for n times measurements on m PCBs, denoted by $IntraHD_{avg}$ was calculated as:

$$IntraHD_{avg} = \frac{2}{mn(n-1)} \sum_{1}^{m} \sum_{p=1}^{n-1} \sum_{q=p+1}^{n} HD_{p,q} \tag{6}$$

Assuming $HD_{i,j}$ stands for the Inter-PCB HD between PCB_i and PCB_j, the average Inter-HD for m PCBs, denoted by HD_{avg}, was calculated as:

$$InterHD_{avg} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} HD_{i,j} \tag{7}$$

To compute the robustness of the signature, we changed the operating voltage of the PCB by 10% and extracted the signature. Next, we calculated the intra-PCB HD which represents the average percentage bit flips. The average inter-PCB HD for 30 boards and 128-bit signature for each board was observed to be 48.34%, which shows that the signatures have high uniqueness. The intra-PCB HD was 1.22%, which shows the signature is robust under environmental variations. The quality of the signature (both inter- and intra-PCB HD) can be further improved by eliminating BSC paths with less variations. For example, if we reduce the signature length to 64 bit, both inter- and intra-PCB HD improved to 49.19% and 1.03%, respectively.

A recycled PCB can be detected based on signature comparison. Different forms of aging effects in a PCB (e.g. electromigration on the metal traces, aging of transistors in the chips) are expected to cause large variation in signatures that makes the signature fail to match with the database. Next, we analyze the unclonability of signature in a PCB. First, we assume that an attacker can only buy the chips to clone a PCB, since the design and manufacturing of a chip may incur unacceptably high cost. The signature is generated from the BSC path delay of chips with statistical variation inherent in manufacturing processes. To clone the authentic PCB successfully, an attacker should find all the chips used on the PCB, which have the identical delays of BSC paths to those on an authentic PCB. Nowadays, each chip may have more than 1000 BSC paths, the probability that the chip manufacturer can produce two authentic chips with identical delay for all the BSC paths is extremely low (e.g., 2^{-1000}). Moreover, with the increased number of chips on a PCB, such cloning work becomes more and more infeasible. Hence, the JTAG-based authentication is an effective and secure method.

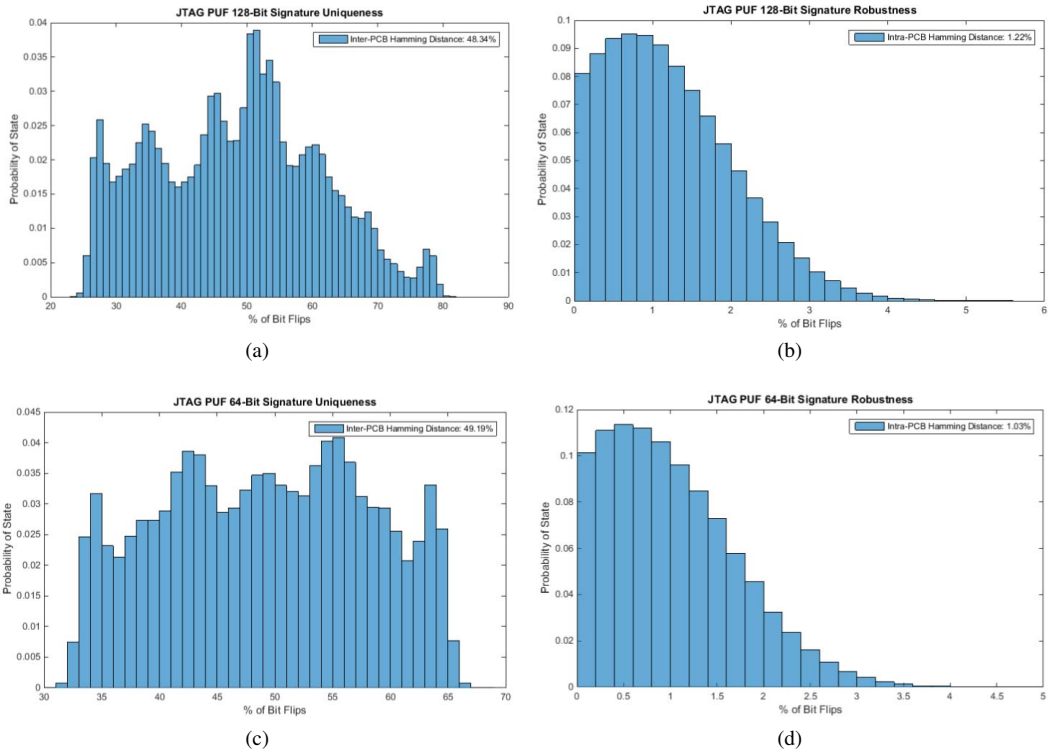

Fig. 10: The measurement results from 30 custom-fabricated test PCBs. (a) inter-PCB Hamming Distance and (b) intra-PCB hamming distance for 128 bit signature; (c) inter- and (d) intra-PCB Hamming Distance for 64-bit signature.

The proposed authentication approach is highly amenable to remote authentication. It can be accomplished by incorporating the PSDM based low-overhead delay measurement inside a system, which can create signature from the PCB on demand using the JTAG port logic with test clock control. It can then transmit the signal through a wired or wireless channel (e.g. using radio signals or WiFi) or the system can verify itself. Hence, in addition to a static integrity check during system integration, such an approach can be effectively used to dynamically verify the integrity of the components in a PCB (e.g. ICs) as well as interconnections among them. Thus, it can protect against potential physical attacks in field that can tamper a PCB (e.g. by replacing a chip with a counterfeit one) to maliciously alter its functionality or gain illegal access to a system.

VI. Conclusion

We have presented a novel low-overhead approach for PCB integrity validation. It utilizes random variations in boundary-scan path delay in the industry-standard JTAG-based DFT structure. Through experimental measurements, we have shown that the proposed scheme can produce high-quality signatures (with good uniqueness and reproducibility) for a wide range of PCBs and can be used to reliably authenticate them. We have also presented an efficient low-overhead method to measure the BSC path delays at fine resolution. We have shown that the quality of signature can be improved through choice of BSC paths, which can be done during off-line signature generation. Since the authentication approach does not require specialized hardware resources or design modifications, it can be applied to any legacy PCB that incorporates boundary scan. The proposed authentication approach provides a low-cost robust method to mitigate supply chain risk associated with counterfeit PCBs.

VII. Acknowledgment

The work is supported in part by National Science Foundation Grants 1054744 and 1603480.

References

[1] W. Custer and J. Custer-Topai, “Global electronic market data”, *Global SMT & Packaging 12.11*, 2012.
[2] “IPC PCB executive agent task force optimistic about efforts: Two government reports recognize vital role of US PCB industry.”, http://www.ipc.org/, 2012.
[3] “IEEE 1149.1 JTAG boundary scan testing in Altera devices”, http://www.altera.com/literature/an/an039.pdf, 2005.
[4] D.E. Holcomb, W.P. Burleson, and K. Fu, “Power-up SRAM state as an identifying fingerprint and source of true random numbers”, *IEEE Transactions on Computers*, 2009.
[5] Maxim Integrated, “Printed circuit board identification (PCB ID) and authentication”, http://www.maximintegrated.com/en/products/comms/one-wire, 2015.
[6] J. Isaac and D. Wiens, ”The future of PCB design”, http://www.creativebits.ca/downloads/The Future of PCB Design.pdf
[7] S. Ghosh, A. Basak, and S. Bhunia, “How Secure Are Printed Circuit Boards Against Trojan Attacks?”, *IEEE Design &Test*, 2015.
[8] Murata Manufacturing Co. RFID MAGICSTRAP®. http://www.murataamericas.com/rfid, 2015.
[9] G.E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation”, *Design Automation Conference*, 2007.
[10] F. Zhang, A. Hennessy, and S. Bhunia, “Robust counterfeit PCB detection exploiting intrinsic trace impedance variations”, *VLSI Test Symposium*, 2015.
[11] Y. Alkabani and F. Koushanfar, “Active hardware metering for intellectual property protection and security,” in USENIX Security Symp., 2007, pp. 291-306.
[12] Y. Zheng, X. Wang, S. Bhunia, “SACCI: Scan-Based Characterization Through Clock Phase Sweep for Counterfeit Chip Detection”, *IEEE Transactions on VLSI Systems*, 2014.
[13] Z. Guo, J. Di, M. Tehranipoor, and D. Forte, “Investigation of Obfuscation-based Anti-Reverse Engineering for Printed Circuit Boards”, *Design Automation Conference*, 2015.