Analysis Of CPU Loading Effect On ESD Susceptibility

Omid Hoseini Izadi
Missouri S&T EMC Laboratory
Missouri University of Science and Technology
Rolla, USA
ohp63@mst.edu

Hideki Shumiya
Sony Global M&O Corporation
Tokyo, Japan
hideki.shumiya@sony.com

Shota Konno
Sony Global M&O Corporation
Tokyo, Japan
shota.kp3.konno@sony.com

Kenji Araki
Sony Global M&O Corporation
Tokyo, Japan
kenji.araki@sony.com

David Pommerenke
Institute of Electronic
Graz University of Technology
Graz, Austria
david.pommerenke@tugraz.at

DongHyun Kim
Missouri S&T EMC Laboratory
Missouri University of Science and Technology
Rolla, USA
dkim@mst.edu

***Abstract*—Two complementary approaches are presented to help to understand how CPU loading affects the sensitivity of an electronic device to ESD (electrostatic discharge) stress. Both approaches rely on synchronized noise injection while the software is running at the desired load. One of the approaches monitors the device's current consumption while the other monitors the device's electromagnetic field to synchronize noise injections. These approaches revealed that as the CPU loading increases, the device becomes more active and hence more susceptible to ESD stress. Moreover, it was observed that, in each loading condition, the device randomly became susceptible. These complementary approaches enable the capturing of high/low active intervals as well as the injection of noise voltage to the desired activity, thus, allowing for the analysis of the effect of CPU loading on ESD susceptibility.**

Keywords—Electrostatic discharge (ESD), soft failure, CPU loading, noise injection, electromagnetic interference (EMI)

I. Introduction

Soft failure investigations are necessary for evaluating the ESD (electrostatic discharge) susceptibility of an electronic device. Soft failure is a temporary upset, disturbing the normal operation of the devices. Soft failures are resolved either automatically after a short time (a few seconds) or by power-cycling the device [1, 2]. Latch-ups and permanent damages (hard failures) could also happen as a result of ESD events [3], however, they are out of the scope of this paper.

For soft failure investigation, the operating condition of the device under test (DUT) should be considered, as the DUT's susceptibility can change when the operating condition changes [4]. [5] has shown that the DUT became more sensitive to ESD when the system loading increased. It also reported that increasing or decreasing the CPU frequency can affect sensitivity. In [6], the authors observed that, while a file compression program was running, other soft failure types occurred other than those related to the display. These studies suggest that higher system loading leads to higher sensitivity.

On the contrary, other studies did not observe a similar trend. [7] reported no correlation between DUT sensitivity and system loading.

In the mentioned works, the ESD noise voltage was injected randomly, i.e., the injections were not associated with any particular activity of the DUT. As will be discussed in the following sections, random injection is not a suitable approach for evaluating the effect of system loading on device susceptibility. A better approach is to correlate ESD injections with the DUT activity; in other words, the injections should be synchronized to a particular activity, to understand the effect of system loading on the device susceptibility.

In this paper, two complementary approaches are presented for synchronizing the noise injection to the device activity. The first approach performs injections synchronous to the current consumption waveform, whereas, the second approach uses electromagnetic interference (EMI). Finally, using a smartphone as our DUT, the approaches are put to the test and compared.

II. Current Consumption-Based Synchronization Method

A. Measurement Setup

Fig. 1 shows the block diagram and the measurement setup. The current sensor is a resistor placed in series with the entire PCB of the device and is used to monitor the instantaneous current consumption of the device. The voltage drop across this resistor is monitored by an oscilloscope. The oscilloscope is set to generate a trigger signal to the TLP (transmission line pulse) generator whenever the current waveform exceeds a user-defined level, which will be henceforth referred to as the *trigger level*. Because the trigger level depends on the CPU loading and CPU frequency of the DUT, it should be set at the peak of the

This paper is based upon work supported partially by the National Science Foundation under Grant No. IIP-1916535.

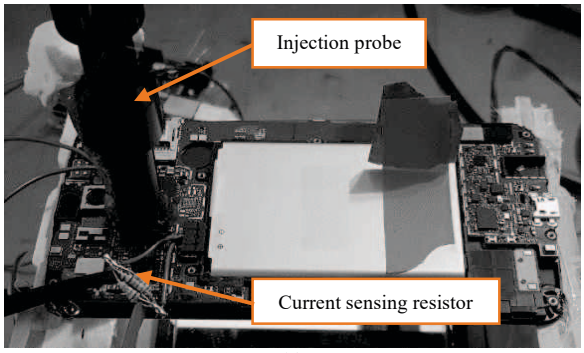
current waveform in order to trigger on high activity intervals, as shown by the horizontal dashed lines in Fig. 2. Similarly, the trigger level is set at the valley points for targeting low activity intervals. With these settings, the oscilloscope is triggered whenever the current consumption crosses the trigger level (dashed line). The generated trigger passes through the delay-control block, gets delayed, and then is fed to the TLP. The delay block compensates for the delay added by the other blocks.

An 8-mm magnetic field probe is used to inject noise into the DUT. When driven by 1 A of current, this probe can couple about 10 mA of current into a 1×0.5 mm loop placed 1 mm below the probe. A detailed explanation is provided about the injection probe in [5].

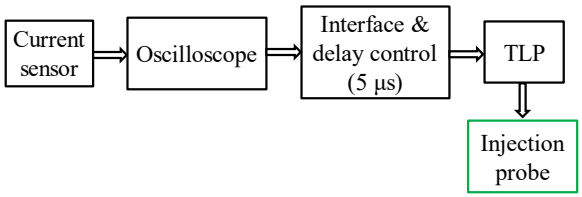
B. *Synchronization Requirements*

For a successful synchronization, the following requirements should be satisfied.

- Steady clock frequency: If the DUT clock frequency can be controlled through software, the user should fix the clock frequency. This can reduce the variations of the current consumption waveform caused by CPU frequency hops. We set the CPU frequency of our DUT at 1 GHz.



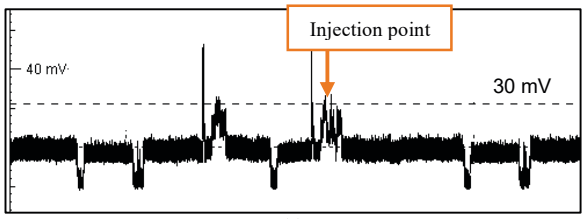
(a)



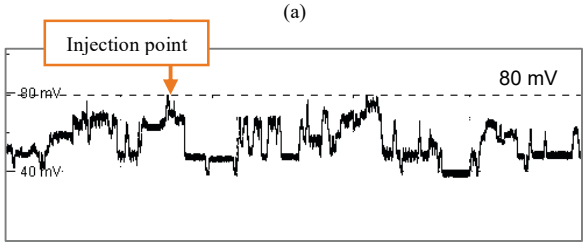
(b)

Fig. 1. Current consumption-based synchronization measurement setup; (a) injection probe and DUT, (b) block diagram of entire setup. The delay control circuitry has ~5 µs delay, which is negligible compared to the 3.4 ms delay of the TLP generator.

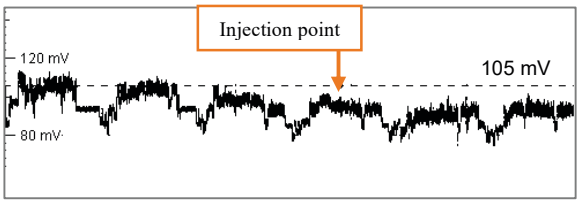
- Known delay: The delay between the moment that the oscilloscope is triggered and when the actual pulse appears at the TLP output should be known with sub-millisecond uncertainty. Most of this delay comes from the TLP relay. A mercury relay can reduce the uncertainty to less than 1 ms. The delay caused by the TLP is 3.4 ms ± 1 ms. The delay caused by the other blocks is in the range of a few micro-seconds and thus is neglected.
- An additional delay should often be added to the total delay such that the injection occurs at the next active interval. For instance, in Fig. 2c, the valley point repeats every ~6 ms, thus, an additional delay of 6-3.4 = 2.6 ms should be added so that the injection happens at the next active interval. Although the current consumption waveform is not periodic in general, especially at low CPU loadings and low clock frequencies, the waveform starts to show a semi-periodic behavior as the CPU loading and the clock frequency increase, as observed in Fig. 2a, 2b, and 2c.



(a)



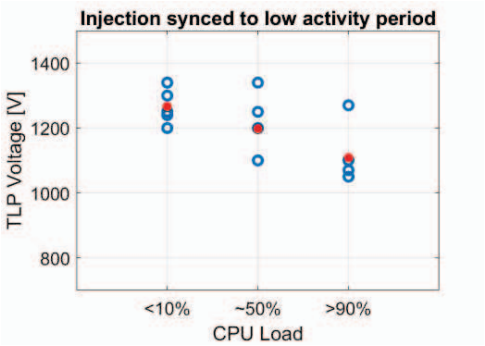
(b)



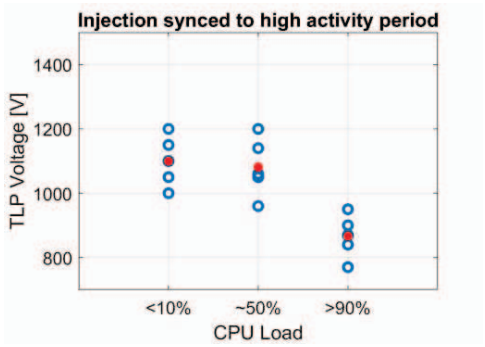
(c)

Fig. 2. Current consumption of the device under test under different CPU loading, (a) *low load* (<10%), (b) *medium load* (~50%), (c) *high load* (>90%). The CPU frequency was fixed at 1 GHz. The noise was injected during the marked low and high activity periods.

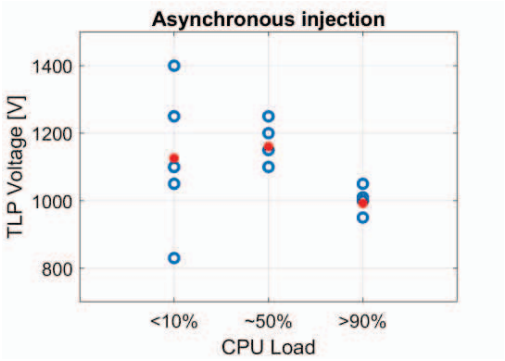
To generate different CPU loadings, it is recommended to employ a software designed for this purpose. A simple infinite loop with arithmetic calculation can intensely load the CPU; however, other parts of the system (RAM, graphic IC, etc.) may not be involved as much as the CPU. Moreover, in case of the loop, the CPU loading intensity cannot be changed – The load would always be close to 100%. In this study, a low load condition (below 10%) was created by leaving the DUT in standby without running any additional software except for the already running system-related tasks. For creating medium load (~50%), a video recording app was used, which could load the graphic IC, RAM, and CPU to some extent. Since this app had not been designed for generating a well-defined load, the activity of the CPU does not have a specific pattern (see Fig. 2b). This



(a)



(b)



(c)

Fig. 3. TLP voltage causing a soft failure vs. CPU load; (a) synchronized to low activity, (b) synchronized to high activity, (c) random injection.

lack of pattern adds uncertainty to the trigger timing. For high load (above 90%), an app called StressCPU ([8]) was used. This app could create a steady load for the CPU and RAM, as shown in Fig. 2c.

C. *Performing Synchronized Injection And Analysis*

A smartphone is used as our device under test (DUT). Its CPU frequency can hop between 300, 500, 800, 1000, and 1200 MHz, which can be controlled by software. Due to the lack of proper heat transfer between the CPU and the environment, we limited the frequency to 1000 MHz. This limitation is imposed because the injection probe directly lands on top of the CPU, which reduces the heat transfer rate. For frequencies below 500 MHz, the current consumption waveform changes irregularly and smoothly, therefore, it was difficult to achieve synchronization, hence the 1000 MHz frequency. To prevent the overheat protection circuitry from kicking in and reducing the CPU frequency by hardware (forcefully), an external fan cools down the CPU.

The TLP source voltage is increased from 0 to 5 kV until a soft failure was observed. The voltage that caused this soft failure is recorded, and the DUT is power cycled to return the DUT to its condition before the soft failure occurrence. Repeating these steps for low load, medium load, and high load conditions gives us the TLP voltage at which the DUT soft-failed vs. CPU load, as shown in Fig. 3. Fig. 3a, 3b, and 3c are obtained by synchronizing the injections to the low activity period of the CPU (corresponding to Fig. 2a), synchronizing to high activity period (corresponding to Fig. 2c), and injecting randomly (asynchronously), respectively. The red asterisk illustrates the average value (of the five repetitions). The following observation can be made from Fig. 3:

For synchronized injection (Fig. 3a and 3b), as the CPU load increases the CPU becomes more sensitive, i.e., lower TLP voltages cause a soft failure. Ideally, the CPU sensitivity should not be affected, as the CPU was stressed during a particular activity period (in low or high corresponding to Fig. 3a and 3b). However, this ideal case is not achievable because: (1) The CPU loading momentarily fluctuates due to system-related apps, housekeeping, or other system activities that are not under the user control; (2) during high load condition (see Fig. 2c), the valley point of the current consumption waveform does not revert to the low value of low load condition (see Fig. 2a); in other words, the CPU loading baseline value increases as the load increases.

Fig. 3b is obtained by synchronizing the injections to high activity periods. As expected, the CPU becomes more susceptible when it is highly active. One may also expect that the CPU sensitivity should remain steady and high, regardless of the loading condition, since the injections are synchronized to high activity periods. This contradiction can be explained using the irregular behavior of the system mentioned above.

Moreover, it is observed that the average value (red asterisk) in each loading condition in Fig. 3a is higher than the corresponding loading condition in the asynchronous scenario (Fig. 3c). This observation suggests the CPU is more robust when it is stressed during its low activity intervals.

Finally, Fig. 3c shows the results for asynchronous injection. Since the injections are performed randomly for this plot, it is expected to have a poor repeatability or in other words a large distribution, especially at lower loads. As the load increases, the idle intervals reduce and become less frequent, drastically reducing the chance of hitting a valley point. This trend can be observed in the current consumption waveforms shown in Fig. 2.

Although the current-based synchronization approach can improve repeatability and reduce the uncertainty of the results, its major downside is its requirement for monitoring the current consumption of the target IC. For a device with one CPU IC, this requirement can be met; however, if more than one CPU IC exists on the device, this approach may fail because the total

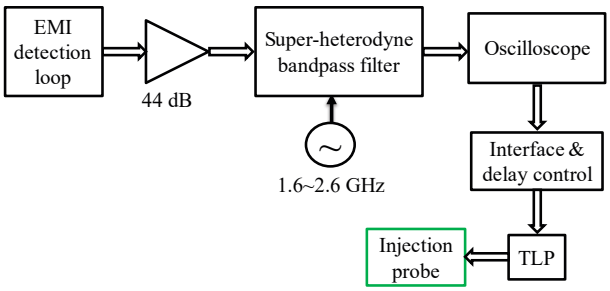


Fig. 4. Measurement block diagram for EMI-based synchronization method.

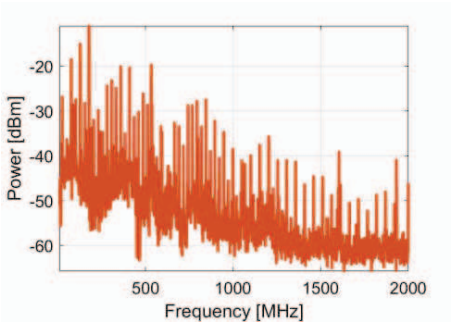


Fig. 5. Amplified spectrum of the IC under test picked up by the detection loop.

current consumption of the entire device is not a good indicator of the target IC activity. An alternative approach is to employ the EMI-based synchronization method.

III. EMI-Based Synchronization Method

A. Measurement Setup

The electromagnetic (EM) field of an IC usually has a broad frequency range. It can consist of both broad and narrow band spectral components. Some of these components may vary as the activity level of the IC changes. These components can be filtered out and used to trigger the TLP.

Fig. 4 shows the measurement block diagram. The detection loop picks up the field generated by the IC of interest. The acquired signal is then amplified and fed to a superheterodyne bandpass filter, which includes a fixed bandpass filter with 1.575 GHz center frequency and 5 MHz bandwidth, two mixers, and one synthesized source. The target frequency is mixed up to fall in the filter bandwidth and then mixed down to the baseband (0-90 MHz). This process allows sweeping through many frequencies without changing the setup. The outputted signal triggers the oscilloscope, and then the TLP after being adjusted by the delay-control block.

The selected frequency to pass through the filter and trigger the TLP should be unique for each loading condition. A frequency that is used for, namely, low load condition should not appear in the spectrum of high load or medium load. Moreover,

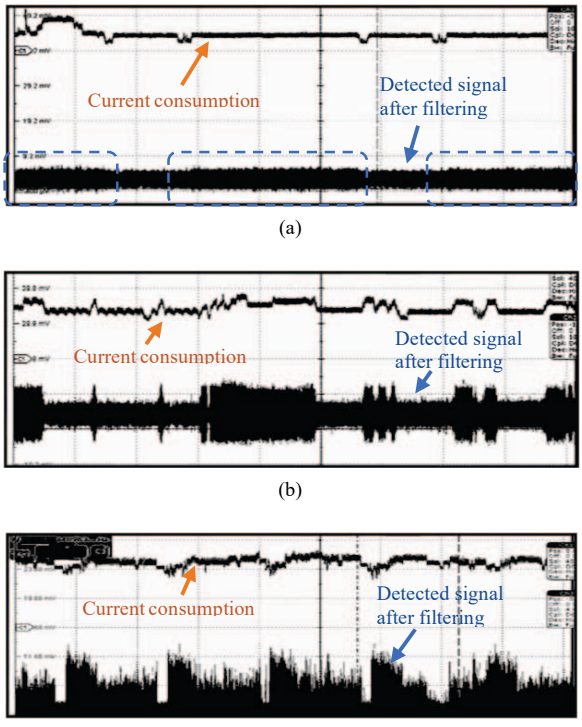


Fig. 6. Current consumption waveform compared to the picked-up signal by the detection loop after the superheterodyne filter; (a) low load, (b) medium load, (c) high load. The selected frequency components are 1.138 GHz for low load, 1.600 GHz for medium load, 1.200 GHz for high load conditions.

in each loading condition, the magnitude of the selected frequency should significantly fluctuate with activity – at least 10 dB is suggested. The biggest challenge of this method is finding a frequency that satisfies these requirements. As shown in Fig. 5, there are many frequencies to be examined. The selected frequencies in this study are 1.138 GHz, 1.600 GHz, and 1.200 GHz for low load, medium load, and high load conditions, respectively.

Fig. 6 compares the current consumption of the device (same smartphone used in the other approach) with the signal picked-up by the loop after the superheterodyne filter in each loading condition. As clearly observed in Fig. 6b and 6c, the current consumption waveform resembles that of the selected frequency, validating the frequency selection for these loads. In low load conditions, the DUT is in standby; thus, the signal picked up by the loop has a relatively constant amplitude, as there is not much change in the DUT's activity in standby; however, as encircled in Fig. 6a, a pattern with small magnitude fluctuation can be observed in the filtered signal. The TLP is triggered based on this pattern.

As for the medium load shown in Fig. 6b, as discussed before, a media recording app was employed to generate this load; thus, the CPU activity has an irregular pattern, adding uncertainty to the trigger timing. This lack of pattern can be observed both in the current waveform and the behavior of the selected frequency component.

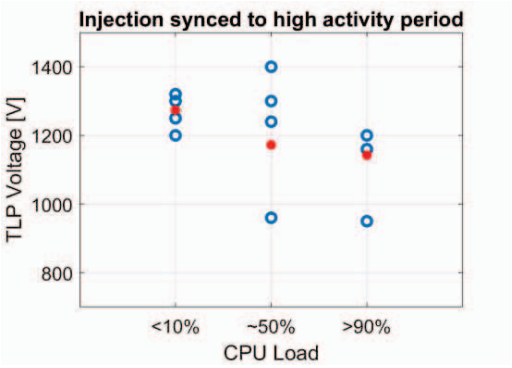


Fig. 7. TLP voltage causing a soft failure vs. CPU load.

As observed in Fig. 6c, the filtered signal not only has a semi-periodic feature, but it also has large amplitude variation. Therefore, it is expected that the failure voltages be relatively less spread out, and the IC be more sensitive in high load. This is discussed in the following section.

B. *Performing Synchronized Injection And Analysis*

Fig. 7 shows the TLP voltage at which the DUT soft-failed vs. CPU load when injections are synchronized to the high activity periods. The decreasing mean value for an increasing CPU load, suggests that the susceptibility of the device increases when the CPU loading increases, i.e., a lower TLP voltage is needed to cause a failure. This behavior is consistent between both synchronization methods. Comparing Fig. 7 with Fig. 3c (asynchronous injection), one can observe that a much better uncertainly is achieved at low load (<10%). Comparing Fig. 7 with its counterpart, Fig. 3b, one can observe that (1) the data has a large distribution especially at medium load, and (2) the failure voltages are usually higher.

These observations suggest that the EMI-based method should be used as a complementary approach for the current consumption approach, or where current monitoring is not possible.

IV. Discussion

A. *Relationship Between Level of CPU Activity And ESD Susceptibility*

The proposed test methods enable us to analyze the level of CPU activity with respect to ESD susceptibility. Some possible physical explanation behind the proportional relationship between the level of CPU activity and ESD susceptibility are:

- When the system is highly active, the system draws more power from the power distribution network (PDN), leading to increased PDN noise. As pointed out in [5], higher PDN noise can lead to higher ESD sensitivity. Higher CPU frequencies exacerbate the situation, as expressed by (1).

$$V(f) = I(f).Z(f), \quad (1)$$

where V is the voltage drop across the impedance of the power distribution network, Z, and I is the current drawn by the CPU.

- During high CPU activity intervals, more subsystems are turned on, compared to those of the low CPU activity intervals. If one or more subsystems that are only ON during high CPU activity intervals are more sensitive to ESD than others and get disturbed by ESD, the resulting soft failure can propagate throughout the system and be observed by the user. However, if the subsystem is OFF its failure may remain hidden.

B. *Multi-core CPUs*

The CPU of the smartphone under test shown in Fig. 1 is a Quad-core CPU, which consists of four ARM Cortex A7 CPUs as well as embedded peripherals such as USB, Bluetooth and Wi-Fi, Cellular Modem, GPU and Display modules. It is assumed that most of the current is consumed by the processors, not the peripherals; therefore, the active low and high intervals in the current waveform are caused by the processor activity.

Due to the multi-core architecture of such CPUs, in general, it is not clear how, namely, a 50% load is distributed between the cores. However, for the CPU tested here, a 100% load completely loads all 4 cores of the CPU. This was verified using a system monitoring app. For a 50% load, since the camera app is being used to generate this load, the load distribution is not uniform between the cores. Which is to say that the loading of the cores fluctuates. These fluctuations can be limited by preventing the CPU frequency to hop (which was done here) and/or use an app to generate a 50% load. The latter could not be achieved because of the lack of the needed skills for Android programming. In a similar study, however, a code was written in Python to generate the desired load. The DUT was BeagleBone Black with a Linux-based operating system called Debian.

C. *Level of Sensitivity For CPU And RAM*

In a different study, where a BeagleBone Black is used as the DUT, the EMI-based synchronization method was performed on the RAM IC. It is observed that the RAM is not as susceptible as the CPU. This observation may be different for different DUTs.

D. *Absolute TLP Voltage Level Vs. Trend*

The average TLP voltage obtained from the current-based approach is slightly lower than the EMI-based approach under the same loading condition. This difference is rooted in different test setups. The authors have observed that a 0.2-mm change in the injection probe height can change failure TLP voltage. Therefore, absolute TLP voltage levels can vary (and should not be compared), while the trend is comparable.

V. Summary and Conclusion

Motivated by the observed contradiction between different studies regarding the effect of CPU loading on ESD susceptibility, we presented two approaches to synchronize noise injection with CPU activity and take into account the effect of CPU loading. Using the current consumption-based synchronization method, we observed that the IC became more sensitive as its load increased. Also, we noticed that, regardless of the loading condition, the IC susceptibility increased during high activity intervals. While the former shows how the IC behaves as a function of loading condition, the latter shows how the IC behaves in millisecond windows during each loading condition. Due to these millisecond active intervals, the asynchronized injection approach could not show how sensitive the IC became under ESD stress. The main drawback of the current-based approach was the need to access and monitor the current consumed by the target IC, which could be impractical in certain devices, such as multi-layered PCBs, or if only one of the many CPU ICs is to be tested. Alternatively, the EMI-based synchronization approach was presented, which monitored the near field of the target IC, instead of its current consumption. Using the EMI-based method, we observed that the target IC became more sensitive as its load increased, a trend consistent with that of the current-based method. The most prominent advantage of the EMI-based method was at low loads (<10%) because there was less variability in the results (compare Fig. 7 with Fig. 3b). Therefore, the two methods should not be used interchangeably, but complementarily. If the load cannot generate a pattern in the current consumption waveform, the EMI-based method should be used instead.

References

[1] Nicholas A Thomson, Yang Xiu, and Elyse Rosenbaum, "Soft-Failures Induced by System-Level ESD," *IEEE Transactions on Device and Materials Reliability,* vol. 17, pp. 90-98, 2017.

[2] Omid Hoseini Izadi, David Pommerenke, Hideki Shumiya, and Kenji Araki, "Investigation of Electrostatic Discharge-Induced Soft-Failure Using 3d Robotic Scanning," in *2019 IEEE International Symposium on Electromagnetic Compatibility, Signal & Power Integrity (EMC+ SIPI)*, 2019, pp. 173-177.

[3] Ki Hyuk Kim, Jeong-Hoi Koo, Bong-Gyu Kang, Soon Jae Kwon, Yongsup Kim, and Joongho Jeong, "Systematic Design Technique for Improvements of Mobile Phone's Immunity to Electrostatic Discharge Soft Failures," in *2010 IEEE International Symposium on Electromagnetic Compatibility*, 2010, pp. 348-353.

[4] Ahmad Hosseinbeig, Omid Hoseini Izadi, Satyajeet Shinde, David Pommerenke, Hideki Shumiya, Junji Maeshima, and Kenji Araki, "A Study on Correlation between near-Field Emi Scan and ESD Susceptibility of Ics," in *2017 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, 2017, pp. 169-174.

[5] Omid Hoseini Izadi, Ahmad Hosseinbeig, David Pommerenke, Hideki Shumiya, Junji Maeshima, and Kenji Araki, "Systematic Analysis of ESD-Induced Soft-Failures as a Function of Operating Conditions," in *2018 IEEE International Symposium on Electromagnetic Compatibility and 2018 IEEE Asia-Pacific Symposium on Electromagnetic Compatibility (EMC/APEMC)*, 2018, pp. 286-291.

[6] Sandeep Vora, Rui Jiang, Shobha Vasudevan, and Elyse Rosenbaum, "Application Level Investigation of System-Level ESD-Induced Soft Failures," in *2016 38th Electrical Overstress/Electrostatic Discharge Symposium (EOS/ESD)*, 2016, pp. 1-10.

[7] Giorgi Maghlakelidze, Pengyu Wei, Wei Huang, Harald Gossner, and David Pommerenke, "Pin Specific ESD Soft Failure Characterization Using a Fully Automated Set-Up," in *2018 40th Electrical Overstress/Electrostatic Discharge Symposium (EOS/ESD)*, 2018, pp. 1-9.

[8] Frank Sabath, Daniel Nitsch, Markus Jung, and Thomas HGG Weise, "Design and Setup of a Short Pulse Simulator for Susceptibility Investigations," *IEEE Transactions on Plasma Science,* vol. 30, pp. 1722-1727, 2002.