Bridging RHA methodology from component to system level applied to System-on-Modules

I. da Costa Lopes, Vincent Pouget, F. Wrobel, A. Touboul, F. Saigne, K. Roed

Bridging RHA methodology from component to system level applied to System-on-Modules

I. Da Costa Lopes, V. Pouget, F. Wrobel, A. Touboul, F. Saigne, K. Roed

***Abstract*—this works presents an RHA methodology that combines both component and system level data to predict system-level reliability. The methodology is illustrated by its application to two system-on-module embedding an avionic application irradiated with protons.**

Keywords — Single-event effects, System-on-Module, Radiation Hardness Assurance

I. Introduction

New Space applications embedding components off-the-shelf (COTS) are requiring fast and cost-effective system design solutions and radiation hardness assurance (RHA) methodologies to qualify low-risk systems. In this context, system-on-modules (SoMs) know a growing interest since they provide a ready-to-use processor module reducing time-to-market for implementing simple digital systems and sub-systems.

In the standard component-level RHA approach, each component within a system is typically characterized individually using a benchmark software application, then the system reliability and availability are estimated with high margins increasing the design cost [1-4]. Alternatively, in the emerging system level approach [5-7], the whole system embedding the final application is characterized simultaneously enabling direct obtention of system reliability. Benchmarks allow the reuse of component–level results, typically for memory components. For more complex components, like processors, testing with the final application is preferable [4] to determine accurate error rates. On the other hand, application-specific system-level results cannot be reused and require whole system implementation [5].

Thus, a methodology that combines Single-Event Effects (SEE) cross-sections and Total-Ionizing Dose (TID) degradation extracted at both component and system level would provide a realistic system reliability estimation in a cost-efficient manner and provide partial data reuse [8]. Nonetheless, there is no straightforward methodology to cope with uncertainties in experimental data, simplifications of current models [9] and the complexity of analyzing fault propagation in modern systems [10].

This work proposes a methodology that provides a first step towards that objective. The bridging methodology and some guidelines are presented in the next section. Then, high-energy proton results on two SOM case studies are presented and discussed. Finally, the methodology is exploited to perform rate prediction on critical events.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie-Skolodowska-Curie grant agreement number 721624.
I. Lopes, V. Pouget, F. Wrobel, A. Touboul, F. Saigne are with IES, University of Montpellier, CNRS, France.
K. Roed is with Department of Physics, University of Oslo, Norway.

II. Elements of a Bridging RHA Methodology for SOMs

A. Targets under study

In this work, we considered digital SoMs based on programmable SoCs that include a Processing System (PS) and Programmable Logic (PL). Digital SoMs also include power regulators, transceivers and volatile and non-volatile memories, as shown in Figure 1.

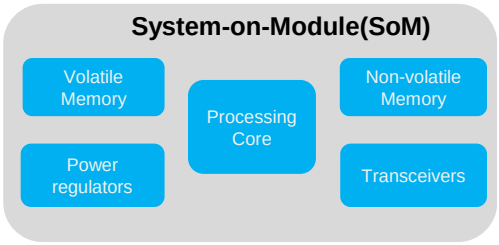

Figure 1. Block diagram of typical digital System-on-Module embedding different components

B. Bridging methodology general flow

The general flow of the proposed bridging methodology is illustrated in Figure 2.

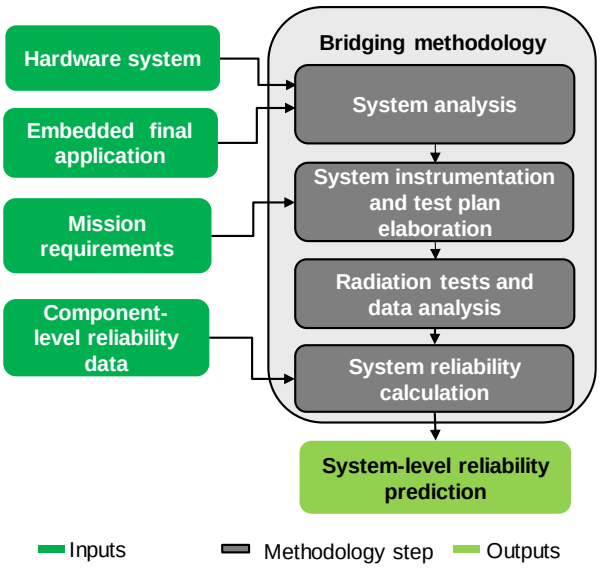

Figure 2. Bridging methodology general flow

As it illustrated in the block diagram, the first step is the system analysis taking the hardware system and the final embedded application as inputs. Afterwards, the system is instrumented and the test plan is elaborated according to the mission requirements. Then, the radiations experiments are performed, the results are analyzed and the reliability of the system is calculated in order to provide system-level reliability prediction. Optionally, component-level reliability data can be used as input for the system reliability calculation. In the following subsections the details of each step will be described.

C. System analysis

The system analysis step is presented in Figure 3.

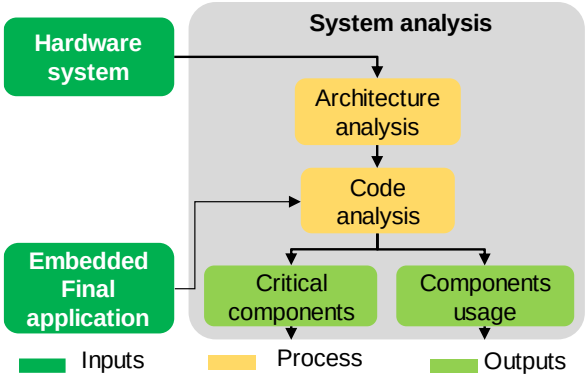

Figure 3. System analysis step

The objective of this step is to provide the identification of critical components and their usage. The hardware system description includes its description and documentation such as block-diagrams, schematics, datasheets, etc. And the embedded final application includes information required to understand the customization of the hardware system. That information can be implementation files (source-codes, executable files, boot images…), test files (representative application dataset) and documentation (functional and parametric specification).

The first process is to analyze the hardware architecture of the system using its available description and documentation. In this process, the assembled components on the system can be identified. Then, by means the code analysis it will be possible to obtain which of those components are used by the application and how they are used. The components usage information includes implementation results (power dissipation, performance and resources utilization), memory spaces (the start addresses of the different memories on the system and its sizes) and device customization (selected voltages, frequencies, operation mode, etc…).

The process of obtaining that information will depend on the availability of the file and documentation. If source-code is available, such as C, C++ and VHDL languages, the all components usage can be defined by implementation. However, if executable files are provided some components usage can be obtained by doing measurements or reverse engineering, being a more complicated process.

The component criticality analysis methodology depends on the application. For the ISO 26262 [11], which is related to safety on electronics embedded on ground vehicles, critically of a component or task depends on its severity, exposure and controllability. The severity of a component or task failure also depends on the application. However, some components are critical in any application because they compromise the availability or survivability of the whole system such as DCDC converters that provides power supply and SoCs that controls the whole system.

The component exposure s is defined by the components usage such as a large memory region that is frequently used by the application and one example of a component with high controllability, is a memory that contains an error correction mechanism, this way an error is this memory becomes less critical.

Additional, the criticality of a component can also be defined by the component SEE sensitivity that can be obtained *a priori*. The use of the components usage and critical components will be defined in the next subsection.

D. Instrumentation and test plan elaboration

The instrumentation and test plan step is presented in Figure 4.

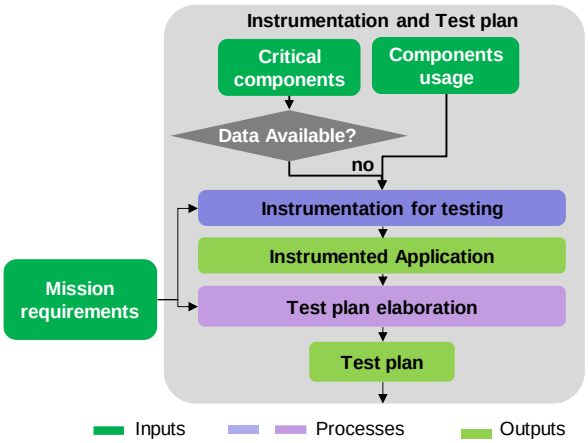

Figure 4. Instrumentation and test plan step

The critical components, components usage and mission requirements such as mission profile, budget and fault coverage will define which additional instrumentation should be added to the final application. Additional instrumentation stands for adding software, Intellectual Property (IP) cores or electrical circuitry to the system's design in order to increase the events observability. As the event observability is reduced in system-level tests, adding instrumentation is fundamental for that approach.

However, adding instrumentation to the system design adds test design time, resources, complexity, code-size, performance and power overhead. Thus, it is important to define different instrumentation levels (ILs) associated to different overheard so that the acceptable overhead will be defined by the mission budget and fault coverage. For instance, when complex applications have a worst combination of states that are rare and demand a high IL to be detected, the mission budget and fault coverage will define weather to overhead associated to that IL will be accepted or not. Additional, depending on the mission profile (duration and environment), health monitors should be added to the design in order to monitor its parametric degradation.

The components usage and available final embedded application will define the instrumentation implementation. For instance, if the final application source-code is available and there are spare resources, a code-instrumentation can be added directly to it. However, if only executable files are provided, the instrumentation can be implemented in other cores, if available, or added directly to the binary file. Hardware instrumentation will depend on hardware description availability such the schematics requirement for implementing a delacther circuit or current monitoring. After, the instrumentation be implemented it is important to validate it by using Fault Injection (FI). For instance, software-level FI [12] or FI by emulation [13] approaches are the most cost-efficient, however pulsed laser FI [14] and focused X-ray at resource level [25] are more realistic approaches.

Once the system is instrumented and validated, the test plan is elaborated according to mission profile, instrumentation and hardware system description. The particle type and spectrum should be defined according to the mission profile and hardware system description so that the active layer of the system components having different thicknesses are reached. Considering we're focusing on digital SoMs made from recent CMOS technologies, and given the limited range constraints associated with heavy ion testing in most facilities, the test plan will most probably start with a high-energy (~200MeV) proton campaign [5]. Alternatively, ultra-high energy (1500 MeV/u) heavy-ions could be used to test digital SoMs without requiring sample preparation at the NSRL facility. However, the facility availability, cost and low LETs should be taken into account when doing such non-conventional experiments.

The total fluence and target fluence per run are defined according to mission profile (environment and duration), off course, and the required fault coverage. By using 200MeV, for instance, a fluence of 1E10 p/cm2 could be used for estimating heavy-ion upper bond rates of a LEO equatorial mission, however a GEO mission would require a fluence of 1E+11 p/cm2 for reproducing the same assurance degree [16]. Those minimum fluences can be used to observe the most frequent events. However, when rare events caused by the worst state combination in complex applications are critical to the mission, the fluence should be increased according to the estimated rate.

The test plan may also define how to set the beam layout in a smart way, e.g. isolating or excluding a component of the beam to analyze its influence on the system reliability. A well instrumented system and good test plan will be essential for the next methodology step.

E. Testing and system-component correlation

The testing and system-component correlation step block diagram is presented in Figure 5.

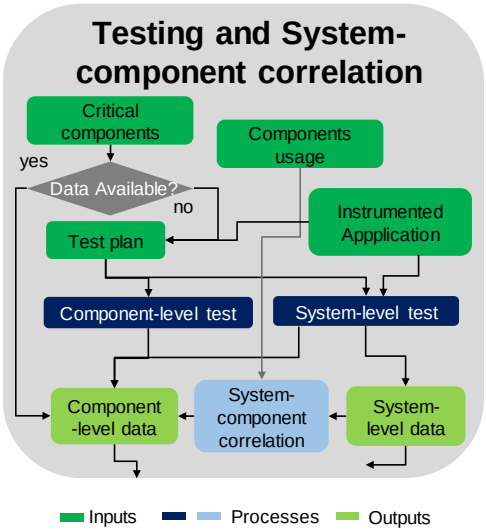

Figure 5. Testing and system-component correlation

As it can be seen in the figure, the critical components are prioritized. Thus, their radiation sensitivity should be determined from data obtained from the literature or data-bases, otherwise dedicated component-level experiments should be performed.

During the system-level testes, the events observability is defined by the instrumentation provided in the previous step. However, it is important to have a detailed and redundant reporting to guarantee the precision of the measurements. In addition, a flexible remote-controlled test setup is required to react to possible unforeseen events such as a low error capturing capability. In this case the instrumentation level could be changed on-the-fly and another version of the final application with higher resources exposition could be loaded.

The system-component correlation process includes elements from the component-level approach such as fault propagation and root cause analysis from system-level approach.

System-level event cross-sections can be used, to some extent, to calculate system availability and reliability. However, additional information is required to make them useful. For instance, if a software application crashes, one should be able to identify if it was generated by a DDR memory or the SoC processor in order to be able to estimate accurately the system reliability. Thus, it is important to perform component or resource root cause analyses of system failure where its fault mechanism and component or resource root cause is estimated.

In addition, by doing root cause analysis, it is possible to obtain component-level cross-sections from system-level cross-sections. However, it is not always trivial and requires careful SoM instrumentation. When high observability is acquired, precise root cause component analysis can be performed in which component level cross-section can be obtained from system-level events.

It also depends on the components usage knowledge. When the final application implementation details are known, structural analysis can be performed where the data path until the system output is analyzed as well the probability of different state combinations and its implication on the system output. Otherwise, only a less precise first order root cause is possible.

However, a first order root cause analysis can be validated by obtaining the component-level or resource level fault propagation to the system level. It can be performed by the same FI approaches for instrumentation validation, previously mentioned or by using fault propagation simulation tools such as SEAM [17]. Regarding SEE, common fault propagation approaches is to compute the Architecture Vulnerability Factor (AVF) [18] and critical bits [19] out of the memory bits used by a software application or FPGA design, respectively. The system-component correlation is important for performing system reliability calculation.

F. System reliability calculation

The system reliability calculation step is presented in Figure 8.

It consists in combining the available and obtained data to perform a first order prediction of the reliability of the system. The inputs of this step, in a generic approach including TID effects, are critical events, and component and system cross-sections and parametric degradation. The objective of combining data from different levels is to complete, when needed, missing system-level data and confirm the system-level rate estimations.

Regarding SEE, it basically consists in performing event rates calculations using common tools such as OMERE [20], making reasonable assumptions when required to cover missing information such as component cell depth, threshold LET or proton energy, saturation cross-section, etc.

When component-level data is used to replace missing system-level data and it was not possible to obtain a precise fault propagation of component-level failures, a worst case fault propagation can be taken into account, thanks to the components usage information, where each bit or resource used by the design is considered to generate a system failure. When the fault propagation is well known, propagated component-level data can be used to identify under or over system-level rate estimations from imprecise system-level data measurements.

More generally, the total system reliability depends on the failure rate due to SEE, TID [21] and aging [22]failure probabilities that increase with the time among other

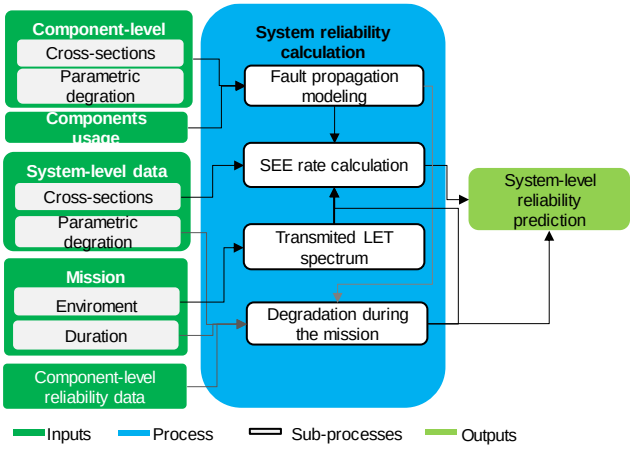

Figure 8. System reliability calculation step

reliability issues. Additionally, it has been also reported in the literature that TID [23] and aging mechanisms [24] can affect SEE sensitivity. Thus, optionally for long missions, component reliability data should also be taken into account in the SEE failure rate estimations.

G. *Methodology summary*

The bridging methodology is summarized in Figure 7. A bridging RHA methodology applied digital SoMs was proposed. It is centered in the idea of adding instrumentation to the final application in order to increase events observability so that the correlation between system-level and component-level data can be performed increasing the system-level reliability calculation precision.

However, this methodology relies in some inputs that, depending on the situation, are not always available. For instance, the schematics or datasheets required to implement hardware instrumentation cannot be provided due to intellectual property reasons. In another situation, the embedded final application cannot be ready during the radiation characterization. Thus, a representative benchmark of the final application should be used decreasing the system reliability estimation.

Regarding the use of both component-level and system-level data for predicting system-level reliability, a first step towards that direction was taken. However, the optional use of both data for system-level reliability prediction is still a field in progress and there is room for improvements.

This methodology was designed for digital electronic systems, in order to transpose the proposed methodology to

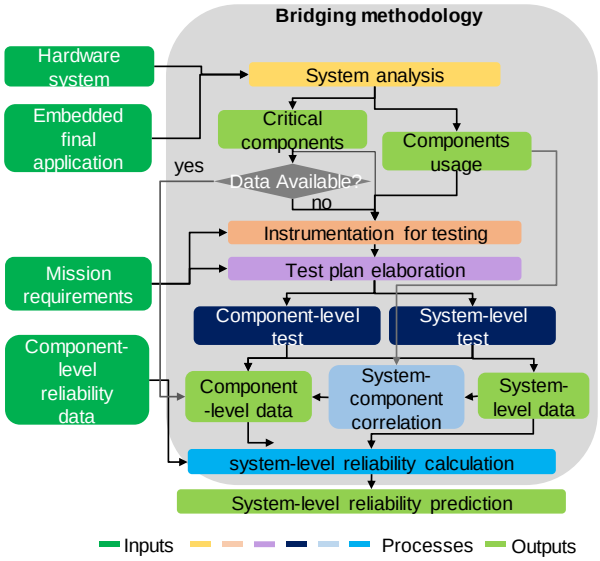

Figure 7. Bridging methodology general flow

board-level or even equipment-level, some extensions and improvements should be performed in order to address different components or subsystems embedding analog, mixed-signal, Radio-Frequency (RF) and optoelectronic components. Currently, the main challenge for digital systems is regarding the assurance of embedded software and firmware reliability. Thus, code and IP-core instrumentation has a significant importance on it. However, a bridging methodology including the other components mentioned would rely more on the hardware instrumentation.

III. CASE STUDY SOM

A. *Systems under test*

The target hardware used as case studies were commercial industrial SoMs of two different generations. The first SoM under test includes a 28nm planar Zynq7000 (Z7) SoC and 1GB DDR3. The second one includes a 16nm FinFET ZynqUltrascale+ (ZU+) SoC and a 2GB DDR4 memory. Both SoMs include the same Quad Serial Peripheral Interface (SPI) 64MB Flash memory. Three different SoM board types were used, as illustrated in Figure 6.

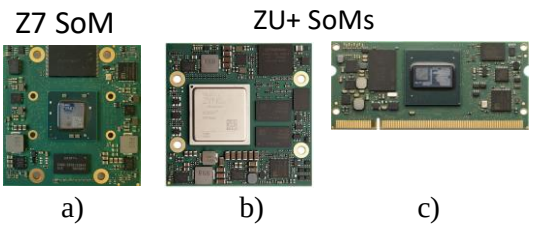

Figure 6. Target SoM pictures. a) small-form factor Z7 SoM b) small-form factor ZU+ SoM c) SO-DIMM ZU+ SoM

The Z7 SoM (Figure 6.a) comes in a small-form factor size (56x54 mm) with baredie SoC, the first ZU+ SoM (Figure 6.b) also comes in a small-form factor size but with metalid package, and finally the second ZU+ SoM (Figure 6.b) comes in a Small Outline Dual In-line Memory Module (SO-DIMM) size (67.6 × 30 mm) board with baredie SoC.

B. Embedded application and instrumentation

As case study a representative application of space and avionic simple embedded digital systems was developed. It consists in a control-loop starting for sensor out Advanced Encryption Standard (AES) decryption, Finite-Impulse-Response (FIR) filtering, Proportional-Integral-Derivative (PID) controlling and Pulse Width Modulation (PWM) actuating, as illustrated in Figure 9.

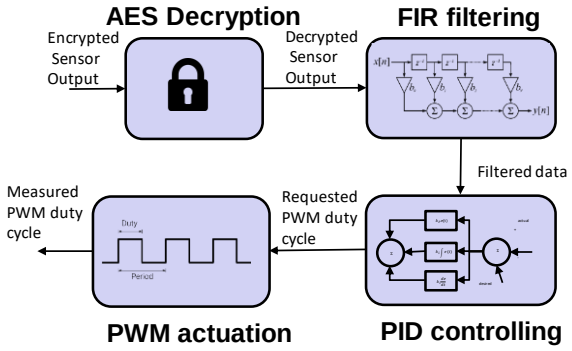

Figure 9. Control loop application block diagram

The same application was implemented in both SoM technologies, with small adjustments to take advantage of the available resources in each SoM. Details about the Z7 case study implementation are given in [25]. The ZU+ case study implementation is presented in Figure 10.

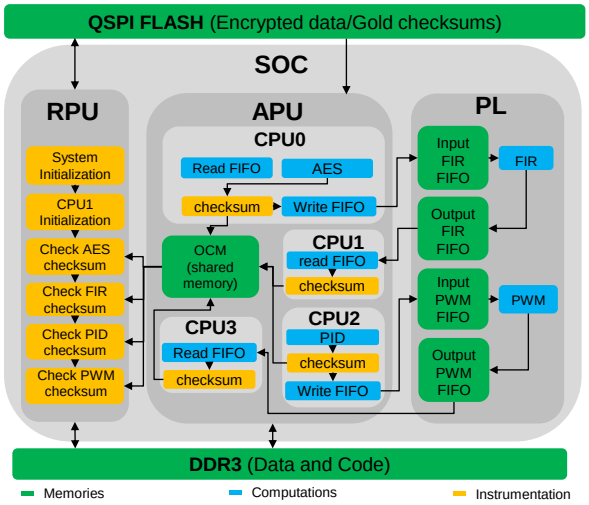

Figure 10. ZU+ case study application including instrumentation

In both SoMs encrypted sensor output was emulated by data stored on the Flash AES and PID computations were performed in the Processing System (PS) of the SoC and FIR and PWM computations were performed in the Programmable Logic (PL) of the SoC. The PS-PL communications were performed by using the Advanced eXtensible Interconnect (AXI) bus and FIFOs implemented with Block RAMs (BRAMs). The main difference is in the number of cores each SoC implementation used.

In order to improve the events observability, a flexible software instrumentation layer consisting of different Instrumentation Levels (ILs) has been developed and tested [25] for the SoMs under study. The objective of the instrumentation is to verify the computations performed in different resources, the application flow and the data integrity of the different memories. Computations were verified by using checksum, application crashes were verified by using watchdog counters, the right application sequence (control flow) was reported by using state flags and the memory integrity was verified by using built-in Error Correction Code (ECC) or checking mechanisms such as parity check.

In IL 0, the SoC application is monitored by detecting errors in the application output computation (PWM) and flow (application crashes and control flow errors). In IL1, memory upsets on the components external to the SoC (DDR and FLASH) are reported and intermediate computations are verified (AES, FIR and PID) in order to detect fault propagation and masking. Finally, in IL2, data integrity and fault status of internal resources of the SoC are monitored such as memory upsets in the On-chip Memory (OCM) and the PL FIFOs and fault registers of SoC that can indicate cache failure. Those different levels can be dynamically activated depending on the overhead constraints and the required observability.

C. High-energy protons test methodology

The system level experiments were performed in the AGRO-FIRM instrument of the KVI-CART facility by using 184MeV protons and fluxes from 1 to 3E+06 p/cm2/s. Achieving a maximum effective fluence of 4.20E+09 p/cm2 for the Z7 experiment and 1.79E+10 for the ZU+ experiment. The whole SoMs were irradiated and system output was reported by using the UART protocol.

A picture of two SoMs mounted in front of the beam line is presented in Figure 11.

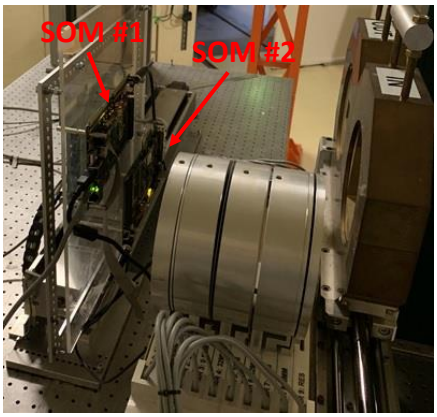

Figure 11. Picture of the test setup in the irradiation area showing

D. Laser fault injection methodology

In order to improve the root cause analysis precision of the proton result, laser FI was performed in the Single Photon Absorption (SPA) IES laser facility that has a laser wavelength of 1064nm and pulse width of 30 ps focused spot size of 1um by using 100X lens. The energy range used was 189-310pJ and the maximum number of pulses was 4,25E5 and 2,8E+5 for the Z7 and ZU+ SoMs respectively.

The experiments were conducted on the leadless back-side of the SoCs without any preparation with the laser

engraved letters written by the manufacturer that prevented the optical access to some sources. Thus, the zones irradiated were selected resources from the PL and PS of the SoCs. A picture of the laser test setup including the ZU+ SO-DIMM SoM is presented in Figure 12.

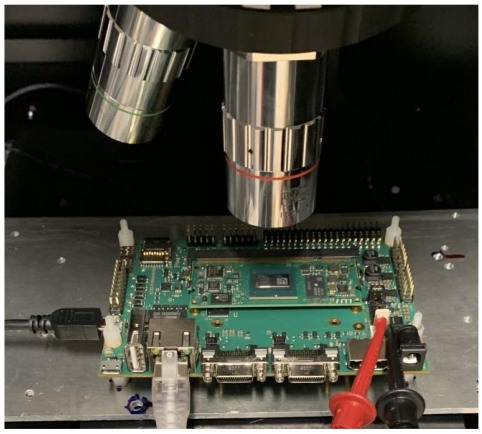

Figure 12. Laser test setup picture

IV. RESULTS AND DISCUSSION

A. *Results presentation*

Different system level events were observed in the proton and laser experiments. Software timeouts were classified into application crashes when the application stops sending data and control flow errors when the application continues running but loops in an unexpected region of the code. Checksum mismatches were classified as checksum errors when corrected in the next loop iteration, soft SEFIs when corrected after a reconfiguration and hard SEFIs when corrected after a power cycle.

No hardware instrumentation was implemented for detecting SELs, thus non-destructive SELs are seen by the application as Hard SEFIs and destructive SELs lead to a permanent loss power, as happened to one of the ZU+ SoMs after a fluence of 2,26E+09 p/cm2.

The events acronyms were defined according to the following logic:

Control Flow Errors (CTRF). Application crashes XY, where X is the processing unit and Y the core index (e.g. APU3, RPU0…). Computation checksum mismatches XY, where X is type of event and Y the type of computation. X can be an Error (E), Soft SEFI (S) and Hard SEFI (H). Examples EAES, HPWM, etc…

Memory upsets XY, where X is the memory (Flash, DDR and OCM) and Y the type of upset such as Single Bit Upsets (SBU and Multi-Bit Upsets (MBUs), for instance DDRM, FLAM and OCMS. FIFO upsets XYZ, where X is the computation (FR- FIR or PW - PWM), Y the direction (I – Input and O - Output) and Z the type of upset (S – Single and D – Double). Example PWIS and FROD. Finally, exception aborts XAB where X is the type of abort (DA – Data, PR – Prefetch and UN – Unexpected).

The system level events observed by IL in the Z7 experiment are plotted in Figure 13.

As it can be seen for both proton and laser results, the application crashes have the highest cross-sections. According to the literature [26], those hangs are most probably generated by Prefetch and Data abort, and Unexpected exceptions that were not handled in the first version of the software. In the laser results that were handled and it was possible to observe Data aborts mainly when irradiating the L1 data cache and Prefetch Aborts when irradiating the L1 instruction and L2 cache.

Regarding checksum SEFIs, it is possible to observe, thanks to the IL1, that the same rate was observed to all the checksum errors (EAES, EPID and EPWM) except the EFIR, which could probably indicate a fault propagation from AES to FIR and fault masking in the PWM computation. In the laser results those events were mostly generated when irradiating the PL FIFO Configurable Logic Block (CLB) and BRAMs.

It is also possible to note that no MBU was observed except on PL FIFO thanks to the IL2. It is probably due to the fact the Error Correction Code (ECC) registers were only checked when a checksum mismatch occurred. A way to provide a more accurate memory cross-section would be to add a parasitic benchmarks or ECC checking that is independent of the final application flow.

System level events observed by IL in the ZU+ experiment are presented in Figure 14.

It is possible to observe, that different from the Z7, proton APU timeouts were not observed and RPU timeouts has lower cross-section than some hard SEFIs. This is due to the fact that exceptions were treated in the new version of the software, as it can be observed by the occurrence of Data and Prefetch aborts. However, in the laser results only RPU timeouts and APU3 timeous were observed when irradiating the PWM FIFO BRAMs and an unidentified resource of the PS.

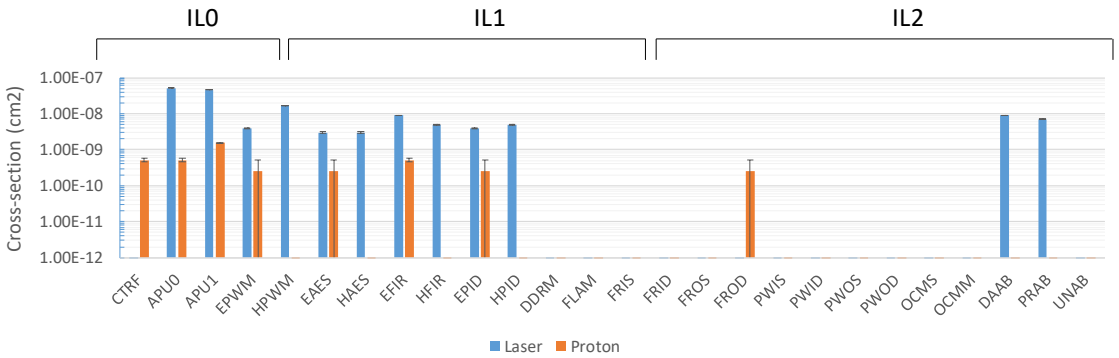

Figure 13. Z7 system level events during the proton and laser experiments

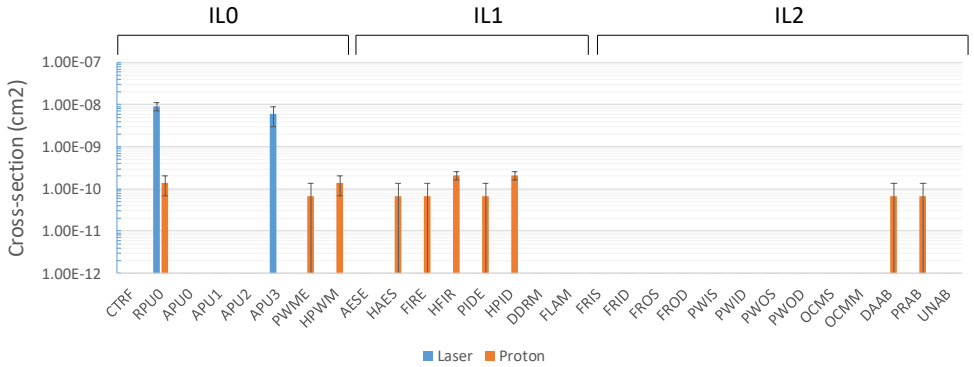

Figure 14. ZU+ system level events during the proton and laser experiments

Regarding hard checksum SEFIs, it is possible to note that there is a lower rate in the output (PWM) than in the intermediate computation steps (FIR and PID) that can indicate fault masking between the different computation steps. No memory MBU was observed in the ZU+ experiment.

B. Methodology case study discussion

As an example of the system-level reliability calculation process of proposed methodology, rate predictions for events considered as the most critical for the tested application were calculated with OMERE [20] and are presented in Table 1.

Table 1. SEE error rates for a LEO ISS mission (800km/50.6°)

SEE rate prediction			
Predictions	**SoM**	**Event**	**SEE rate (events/day)**
Optimistic	**Z7**	**Soft failures**	2.47E-04
		Hard failures	7.12E-06
		Resettable failures	6.63E-03
	ZU+	**Soft failures**	2.91E-04
		Hard failures	5.51E-03
		Resettable failures	1.14E-03
Conservative	**Z7**	**Soft failures**	4.94E-04
		Hard failures	2.85E-05
		Resettable failures	1.99E-02
		Soft failures	5.82E-04
		Hard failures	1.65E-02

The system-level rates were calculated by a combination of component-level and system level data. Component-level cross-sections per bit were extracted from the literature and multiplied by the bits used by the design. And the system-level cross-sections were obtained from the measured 184MeV proton results.

For the component-level results MBUs were classified as soft failures, SEFIs and MBU in the DDR code region and PL Configuration RAM (CRAM) were classified as (resettable failures) and non-destructive SEFIs were classified as hard failures. As no SEL instrumentation was applied in the system level experiments, checksum errors were classified as soft failures and SEFIs were classified as resettable failures while destructive failures including destructive SELs were classified as hard failures. For the measured results a threshold proton energy of 10MeV was defined [30].

In the optimistic estimation, when there was missing system-level proton measured data or the rates were lower than the ones estimate from component-level, component-level estimations were used. In the conservative estimation, safety margin factors from 2 to 4 were applied according to the criticality of the event (soft and hard) and confidence-level of the measurements.

DDR [27], DDR4[28] and Z7 SoC [29], ZU+ SoC[30] heavy-ion, proton and neutron cross-sections were extracted from the literature, when needed proton rates were calculated from heavy-ion data using the SIMPA model [31] and heavy-ion data were obtained from neutron and proton data considering the maximum secondary LET as the saturation LET.

Results in Table 1 show that the event rates are relatively low. The optimistic estimation could validate a short duration (0.25 years) mission since the highest rate event would take (0.41) to happen. The same mission would not be validated by a conservative estimation based on safety margin factors

However, that approach enables only Worst Case Analysis (WCA) and the mission could be validated trough precise error rate estimation by obtaining PL critical bits out of essential bits and AVF out of DDR code/data region, OCM, cache and register bits.

Moreover, if the system availability (Hard SEFI and timeout) requirements are not meet, one should know which component caused those errors that could be, for instance, DDR or SoC PL MBUs that were not captured by the added instrumentation. Thus, root cause component analysis rather than only WCA, should be performed, when possible, by improving the system instrumentation or performing additional laser fault injection. Since specific instrumentation design and validation increases test design time and complexity, it seems interesting to develop a generic (and possibly standard) instrumentation layer that could cover different families of digital SoMs and that could provide enhance observability of resources-level events independently of the embedded application.

An important limitation of the presented case study is the lack of observability of SEL and analog parts. That could be achieved by minor hardware modifications but requiring

full SoM schematic and slightly increasing the test complexity. This way, a trade-off between observability and overhead should be leveraged. Another option for SEL observation might be to use an infrared camera as performed in [32].

In addition, even reaching a relatively high fluence in the ZU+ experiment (1.79E+10 p/cm2), low statistics events with high error bars were observed (Figure 13) and only proton energy cross-section saturation and estimated energy threshold were used. Those uncertainties should be considered when transposing the SEE rates to other environments.

V. CONCLUSION

A bridging methodology making use of both component and system level data for predicting system-level reliability was proposed including component level SEE cross-section gathering from system level experiments through the addition of system instrumentation. As a case-study, two SOM technologies, embedding similar representative final applications and a flexible instrumentation layer, were tested at system-level with protons. An example of system-level reliability prediction was performed by combining system-level events observed and component-level cross-section from literature. Applicability and limitations of the methodology applied to the case study were discussed including the importance of root cause component analysis.

REFERENCES

[1] ESA, ESA, and SCC Basic specification No. "25100: Single event effects test method and guidelines." ESA, Noordwijk, Netherlands 1005 (1995).

[2] MIL-STD-750-1 Method 1080.1, “Single-event burnout and single-event gate rupture,” Department of Defense, USA.

[3] Schwank, James R., Marty R. Shaneyfelt, and Paul E. Dodd. "Radiation hardness assurance testing of microelectronic devices and integrated circuits: Radiation environments, physical mechanisms, and foundations for hardness assurance." IEEE Transactions on Nuclear Science 60.3 (2013): 2074-2100.

[4] JEDEC. (2006). Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Error in Semiconductor Devices. JEDEC Standard JESD89A, (October), 1–85. Retrieved from http://www.jedec.org/standards-documents/docs/jesd-89a

[5] Guertin, Steven M. Board level proton testing book of knowledge for NASA Electronic Parts and Packaging Program. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2017, 2017.

[6] Julien, Connor R., Brock J. LaMeres, and Raymond J. Weber. "An FPGA-based Radiation Tolerant SmallSat Computer System." 2017 IEEE Aerospace Conference. IEEE, 2017.

[7] Secondo, R., et al. "System level radiation characterization of a 1U CubeSat based on CERN radiation monitoring technology." IEEE Transactions on Nuclear Science 65.8 (2018): 1694-1699.

[8] Rousselet, M., et al. "Use and benefits of COTS board level testing for radiation hardness assurance." 2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS). IEEE, 2016.

[9] Sukhaseum, Nicolas, et al. "Statistical estimation of uncertainty for single event effect rate in OMERE." 2011 12th European Conference on Radiation and Its Effects on Components and Systems. IEEE, 2011.

[10] Austin, Rebekah A., et al. "A CubeSat-payload radiation-reliability assurance case using goal structuring notation." 2017 Annual Reliability and Maintainability Symposium (RAMS). IEEE, 2017.

[11] ISO. “ISO 26262 – Road Vehicles – Functional Safety Standard”, International Standard, ISO, 2011.

[12] G. S. Rodrigues, F. Rosa, Á. B. de Oliveira, F. L. Kastensmidt, L. Ost and R. Reis, "Analyzing the Impact of Fault-Tolerance Methods in ARM Processors Under Soft Errors Running Linux and Parallelization APIs," in IEEE Transactions on Nuclear Science, vol. 64, no. 8, pp. 2196-2203, Aug. 2017, doi: 10.1109/TNS.2017.2706519.

[13] I. C. Lopes, F. Benevenuti, F. L. Kastensmidt, A. A. Susin and P. Rech, "Reliability analysis on case-study traffic sign convolutional neural network on APSoC," 2018 IEEE 19th Latin-American Test Symposium (LATS), Sao Paulo, 2018, pp. 1-6, doi: 10.1109/LATW.2018.8347234.

[14] Rodrigues, G. S., Barros, A., Lopes, I., Pouget, V., Bosio, A., & Fernanda, L. “An Approximate Error-Detection Technique for Multi-Core Real-Time Systems”. Proceedings of the European Conference on Radiation and Its Effects on Components and Systems, RADECS, (in-press) 2019.

[15] I. C. Lopes, V. Pouget, K. Roed, F. Wrobel, A. Touboul, F. Saigné, J. Boch, T. Maraine. “Comparison of TID-induced Degradation of Programmable Logic Timings in Bulk 28nm and 16nm FinFET System-on-Chips under Local X-ray Irradiation”. IEEE Nuclear & Space Radiation Effects Conference, (in-press) 2020.

[16] O'Neill, P. M., G. D. Badhwar, and W. X. Culpepper. "Internuclear cascade-evaporation model for LET spectra of 200 MeV protons used for parts testing." IEEE Transactions on Nuclear Science 45.6 (1998): 2467-2474.

[17] A. Witulski et al., "Development of a Flight-Program-Ready Radiation Model-Based Assurance Platform," 2020 IEEE Aerospace Conference, Big Sky, MT, USA, 2020, pp. 1-8, doi: 10.1109/AERO47225.2020.9172762.

[18] Mukherjee, Shubhendu S., et al. "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor." Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.. IEEE, 2003.

[19] XILINX: PG036 - Soft Error Mitigation Controller v4.1. URL https://www.xilinx.com/support/documentation/ip_documentation/sem/v4_1/pg036_sem.pdf

[20] OMERE 5.3. [Online]. Available: http://www.trad.fr/en/download/omere-us/

[21] R. Ladbury and B. Triggs, "A Bayesian Approach for Total Ionizing Dose Hardness Assurance," in IEEE Transactions on Nuclear Science, vol. 58, no. 6, pp. 3004-3010, Dec. 2011, doi: 10.1109/TNS.2011.2172461.

[22] Bernstein, Joseph B.: Aerospace electronics reliability: Could it be predicted in a cost-effective fashion? In: IEEE Aerospace Conference Proceedings Bd. 2015-June (2015), S. 1–6 — ISBN 9781479953790

[23] A. A. Novikov, A. A. Pechenkin and A. I. Chumakov, "The Behavior of SEE Sensitivity at Various TID Levels," 2014 IEEE Radiation Effects Data Workshop (REDW), Paris, 2014, pp. 1-4, doi: 10.1109/REDW.2014.7004599.

[24] El Moukhtari, I., et al. "Analysis of short-term NBTI effect on the Single-Event Upset sensitivity of a 65nm SRAM using two-photon absorption." 2013 14th European Conference on Radiation and Its Effects on Components and Systems (RADECS). IEEE, 2013.

[25] Lopes, I. C., Pouget, V., Roed, K., Wrobel, F., Touboul, A., Saigné, F., et al. “Development and evaluation of a flexible instrumentation layer for system-level testing of radiation effects”. Latin American Test Symposium, LATS, (in press) 2020.

[26] Peña-Fernandez, M., et al. "The Use of Microprocessor Trace Infrastructures for Radiation-Induced Fault Diagnosis." IEEE Transactions on Nuclear Science (2019).

[27] Grürmann, Kai, et al. "Heavy ion sensitivity of 16/32-Gbit NAND-flash and 4-Gbit DDR3 SDRAM." 2012 IEEE Radiation Effects Data Workshop. IEEE, 2012.

[28] Guertin, Steven M., and Matthew Cui. "SEE test results for the Snapdragon 820." 2017 IEEE Radiation Effects Data Workshop (REDW). IEEE, 2017.

[29] Amrbar, Mehran, et al. "Heavy ion single event effects measurements of Xilinx Zynq-7000 FPGA." 2015 IEEE Radiation Effects Data Workshop (REDW). IEEE, 2015.

[30] Davis, Philip, et al. "Single-Event Characterization of the 16 nm FinFET Xilinx UltraScale+ TM RFSoC Field-Programmable Gate Array under Proton Irradiation." 2019 IEEE Radiation Effects Data Workshop. IEEE, 2019.

[31] Doucin, B., et al. "Model of single event upsets induced by space protons in electronic devices." Proceedings of the Third European

Conference on Radiation and its Effects on Components and Systems. IEEE, 1995.

[32] De Bibikoff A., Lamberbourg P., “Method for System-level testing of COTS electronic board under High Energy Heavy Ions”, RADECS 2019, to be published.