Low Cost Error Monitoring for Improved Maintainability of IoT Applications

Mauricio D. Gutierrez, Vasileios Tenentes, Tom J. Kazmierski
University of Southampton, UK
Email: {mdga1g11, V.Tenentes, tjk}@ecs.soton.ac.uk

Daniele Rossi
University of Westminster, UK
D.Rossi@westminster.ac.uk

***Abstract*—Electronic systems with power-constrained embedded devices are used for a variety of IoT applications, such as geo-monitoring, parking sensors and surveillance. Such applications may tolerate few errors. However, with the increasing occurrence of faults in-the-field, devices that exhibit systematic erroneous behaviour must be eventually identified and replaced. In this paper, we propose a novel low cost error monitoring technique to assist the maintainability planning of low power IoT applications by ranking devices based on the systematic erroneous behaviour they exhibit. Small on-chip monitors are used to collect the signal probability information at the outputs of each device which is then transmitted to the system software via the communications channel of the system to rank them accordingly. To evaluate the error monitoring capabilities of the proposed technique, we injected multiple bit-flips and stuck-at faults on a set of the EPFL and the ISCAS benchmarks. Results demonstrate an average error coverage of 84.4% and 73.1% of errors induced by bit-flips and stuck-at faults, respectively, with an average area cost of 1.52%. A maintainability planning simulation shows that the proposed technique achieves a reduction of 26x to 263x in area cost and static power, and consumes over 625x less power for communications when compared against duplication and comparison.**

I. Introduction

The maintenance of electronic devices used in low-power Internet-of-Things (IoT) applications often requires physical access which might be impractical and has to be planned in advance [1]. Devices that exhibit systematic erroneous behaviour (SEB) must be identified and replaced. Thus, the maintainability of low power IoT applications can be assisted by monitoring the behaviour of those devices in-the-field.

Concurrent error detection (CED) techniques may be used to monitor SEB. CED techniques using duplication and comparison (D&C) are applicable to any circuit, and detect almost 100% of single errors with a low error detection latency, as they target an immediate detection of errors as they occur [2], but incur an area and power overhead of more than 100% [3]. CED techniques using error detecting codes achieve a lower error coverage, but with less overhead compared to D&C. However, they may have an impact on system performance and are traditionally used for memories or control logic [4], [5]. As a result, a low cost solution for monitoring devices used in low power IoT applications is required.

However, many applications for low power embedded devices such as geo-monitoring, parking sensors, or surveillance, can tolerate some errors during normal operation [6]. Such devices are not constrained by a strict error detection latency requirement, thus detecting errors immediately as they occur may not be required, as long as they continue to offer their intended service. Therefore, using error detection mechanisms such as D&C results expensive for such applications. Signal Probability Monitors (SPMs) have been recently proposed as a low cost error detection technique of SEB for applications where errors can be tolerated [7]. These monitors measure deviations of the online signal probabilities at the outputs of circuits and are capable of detecting when SEB has occurred.

In this paper, we propose a novel low cost error monitoring technique to assist the maintainability planning of IoT applications by ranking devices based on the amount of errors they exhibit. The proposed technique detects SEB in circuits used in power constrained error-tolerant applications with loose error detection latency requirements. On-Chip SPMs collect the signal probability information at the outputs of each device concurrently to normal operation. This signal probability information is transmitted to the system software through the communications channel of the system where a software module analyses the SEB exhibited by each device and ranks them accordingly. The proposed technique has been evaluated considering the SEB detection capabilities of the SPMs and by performing a maintainability simulation to compare the cost and error coverage of the proposed technique compared to D&C. To evaluate the SEB detection capabilities of the proposed technique, we injected multiple bit-flips and stuck-at faults on a set of the EPFL and the ISCAS benchmarks. We demonstrate an average error coverage of 84.4% and 73.1% of errors induced by bit-flips and intermittent stuck-at faults, respectively, with an average area cost of 1.52%. Furthermore, the maintainability simulation shows that when compared against D&C, the proposed technique achieves an area cost and static power reduction of 26x to 263x, and consumes over 625x less power.

This paper is organized as follows: Section II presents the motivation of this work. Section III presents the proposed SPM based error monitoring technique. Section IV presents the results of the two evaluations of the proposed technique followed by the conclusions in Section V.

II. Motivation

Figure 1a presents an IoT system using duplication and comparison (D&C). D&C enables the IoT system to detect all

978-1-5386-0362-8/17/$31.00 ©2017 IEEE

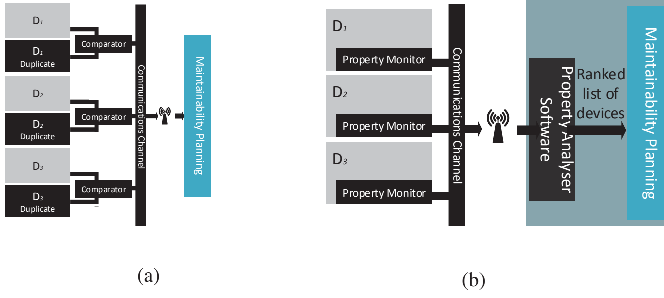


Fig. 1: (a) IoT system of devices using D&C error detection, (b) Proposed IoT system using a property monitor and analyser

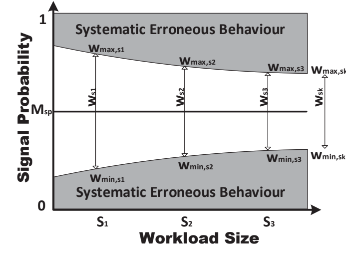


Fig. 2: Online signal probabilities

single errors as they occur on each of the terminal devices. The error information is sent through the communications channel to the system software where the maintainability planning takes place. The area and power cost of this error detection mechanism however, may result too expensive for low power IoT applications and a less costly solution may suffice.

A. *Signal probability monitors*

Signal probability monitors (SPMs) were recently introduced as a low cost error monitoring technique of the online signal probabilities [7]. Figure 2 presents the concept of online signal probabilities. The set of input patterns are referred to as the workload and the number of input patterns in a workload as workload size, denoted by S. During an error-free normal operation, the online signal probabilities at a given node may vary depending on the workload. The smaller the size of the workload, the higher the variation of the online signal probability. As the S increases, the variation of the online signal probability at the output decreases and starts to converge. The value to which the signal probability converges is the *mean signal probability*, denoted by M_{sp}. The variation of the signal probability during an error-free operation is referred to as *signature window* (w), with W_{max} and W_{min} as the *upper and lower bounds* respectively. The expected M_{sp} and the W_{max} and W_{min} signature window bounds are dependent on the input signal probabilities.

Systematic erroneous behaviour (SEB) is defined as the event in which, for a particular workload size S, systematic errors occur at a high enough rate, that the online signal probability of an output falls outside the signature window w. SEB may occur in-the-field due to intermittent faults caused by defects escaping manufacturing testing, process variation, wearout and aging [8]. Intermittent faults may manifest as *multiple bit-flips* or exhibit a behaviour similar to *permanent faults* under specific operating conditions [9]. In the presence of a fault, the circuit may produce enough errors that the online signal probability at the output falls outside the signature window w (lower than W_{min} or higher than W_{max}).

In the presence of a fault, for a given input pattern, an error is considered to have occurred only when the output of the circuit is different from the error-free case. That is:

$$error = \begin{cases} 1 & \left[o_k^{if}, p_k\right] \neq \left[o_k^{ff}, p_k\right] \\ 0 & \left[o_k^{if}, p_k\right] = \left[o_k^{ff}, p_k\right] \end{cases} \tag{1}$$

where o_k^{if} is the output when a fault is present, o_k^{ff} is the output of the error-free case and p_k is the input pattern.

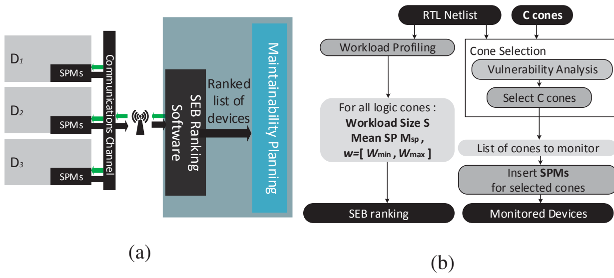
In application-specific ICs, where the workload may be known during design time, the signal probabilities of the workload tend to be *biased* towards the application. This causes the input patterns to be heavily correlated and the expected behaviour to be known. In such cases, the M_{sp} is known and the width of w might be small. In the case of general purpose devices, the workload is unknown during design time as it may vary substantially in-the-field and its patterns appear uncorrelated. If a workload is unknown, all its input patterns are considered to be random and equally likely to occur. The workload is *unbiased* towards a particular application, which makes it necessary to profile the signal probabilities to compute the M_{sp} and w. The analysis of online signal probabilities described in Figure 2 is applicable for either a biased or unbiased workload.

III. Proposed Low Cost Error Monitoring

To reduce area cost and power consumption, we propose an IoT system where instead of detecting all single errors, a property of the behaviour of each device is monitored (Figure 1b). This property information is sent to the system software where it is analysed. The property analyser provides the maintainability planning with a list of devices in the system, which are ranked according to a metric defined as a function of the property that is being monitored. Figure 3a shows the proposed low cost error monitoring technique using SPMs as property monitors and a SEB ranking software as the property analyser. The proposed technique consists of on-chip SPMs that communicate with a SEB ranking software module through the communications channel of the system. The SEB ranking software analyses the signal probability data and ranks the devices according to the number of SEB detections over a predefined interval.

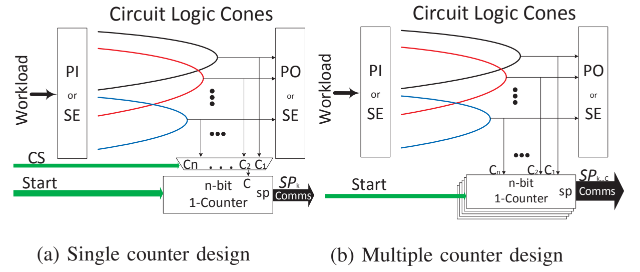
A. *Monitoring technique design flow*

Figure 3b presents the proposed design flow of the SEB ranking module and for the insertion of the SPMs on the chip. The process of workload profiling is performed depending on whether the workload is biased or unbiased [10], [11]. For a biased workload, the correlation and variations of the input patterns are known, which makes the mean signal probability M_{sp} and the signature window bounds W_{min} and W_{max}



(a) (b)

Fig. 3: (a) Proposed maintainability planning with error monitoring using SPMs, (b) Technique design flow



(a) Single counter design (b) Multiple counter design

Fig. 4: Monitoring architectures.

simple to identify. For an unbiased workload, where its input patterns are uncorrelated and considered to be equally likely to occur, an error-free simulation of a large number of unbiased workloads is required to compute the M_{sp}. The workload size S is determined once the online signal probabilities have converged. The signature window bounds W_{min}, W_{max} may be set to $M_{sp} \pm 3\sigma$ for a 0.3% probability of having a false alarm [7]. However, the signature window can be narrower, which produces a higher error coverage but increases the probability of having false alarms. This trade-off is explored in Section IV. The SEB ranking module uses the S, M_{sp} and w defined for all logic cones to determine when SEB has occurred and to rank the devices of the IoT system accordingly.

A logic cone selection process is carried out to define the C cones to monitor. Logic cones of any size and any number of inputs that are bounded by either primary inputs and outputs (PI/PO), or by sequential elements (SE) may be selected. The simplest cone selection process consists of selecting the C cones that exhibit the highest number of errors. This selection may also be based on different vulnerability analysis methodologies [12], [13]. Once the list of cones is defined, the SPMs are are synthesized and inserted into the netlist.

B. Architecture of the signal probability monitors

Two SPM-based architectures are proposed in this paper. A single counter design which provides a lower area cost but a higher monitoring time, and a multiple counter design that enables the monitoring of multiple cones at the same time but with a higher area cost.

1) Single counter design: Figure 4a shows the single counter design. When the *start* signal generated by the SEB ranking module at the backend of the system is asserted, the *n-bit 1-Counter* increases on the rising edge of the clock if the input C is asserted, effectively counting the number of logic 1's. The incoming *CS* signal selects the cone to monitor the multiplexer. The counters send the *SP* data over the communications channel when the S patterns of the workload have appear at the inputs. The value of n is determined by the workload size S according to equation (2). That will result in the minimum n required to count up to S.

$$n = \lceil log_2(S) \rceil \tag{2}$$

2) Multiple counter design: The multiple counter design (Figure 4b) enables monitoring of all the cones simultaneously. It consists of an *n-bit counter* per monitored cone. All the *SP* data is sent in parallel through the communications channel. Note that a single counter incurs a lower area cost compared to a multiple counter architecture, however is only able to monitor signal probabilities for a single cone at a time, resulting in an increased error detection latency for the other cones. On the other hand, the multiple counter architecture allows to monitor all the selected cones at the same time, reducing monitoring time and error detection latency, but increasing the area cost. Both the single and multiple counter designs may be clock or power gated, enabling the monitors only when they are requested by the SEB ranking module.

C. SEB ranking software

The comparison to determine if SEB has occurred is performed off-chip in a software module. The SEB ranking software (Figure 3a) sends the *start* signal over the communications channel to the SPMs in each of the devices. The SEB ranking module receives the signal probability data from the terminal devices after S clock cycles, which is then compared to the error-free signature window of each of the monitored logic codes of each device. If the received data is outside the corresponding signature window (SP$< Wmin$ or $Wmax <$SP) an alarm is raised for that logic cone. When the SPMs consist of a single counter, the *counter select* (*CS*) signal is increased after S clock cycles have passed, to monitor the next logic cone in a round-robin fashion. After the SP data of all the logic cones has been received, the *start* signal is de-asserted and the SPMs are disabled to save power and the number of alarms raised for each device is stored. After a predefined number of iterations the devices are ranked based on their accumulated number of alarms.

IV. Simulation Results

The proposed technique has been evaluated considering the SEB detection capabilities of the SPMs and by performing a maintainability simulation to compare the cost and error coverage of the proposed technique compared to D&C. The SEB detection evaluation consists of a simulation to compute of the error coverage achieved by the SPMs for a subset of the ISCAS'89 and EPFL'15 benchmarks [14], as well as the estimated area costs associated with them. The maintainability

TABLE I:
ibBF and ibSSA EC and area cost of different monitor designs

Benchmark	Logic Cones	Circuit gates	Workload Size S / EDL	Monitored Cones C	ibBF EC (%) of selected cones						ibSSA EC (%)				Area Cost (%)	
					$w = M_{sp} \pm 3\sigma$			$w = M_{sp} \pm \sigma$			$w = M_{sp} \pm 3\sigma$		$w = M_{sp} \pm \sigma$		Single counter	Multiple counters
					1 bit-flip	2 bit-flips	3 bit-flips	1 bit-flip	2 bit-flips	3 bit-flips	Whole Circuit	Selected Cones	Whole Circuit	Selected Cones		
c6288	32	2437	7000	1	30.07	39.98	70.76	70.37	69.74	69.74	2.84	50.37	5.11	90.65	4.66	4.66
				5	35.86	29.25	64.18	66.94	63.48	58.13	9.00	55.24	22.51	78.64	7.64	23.32
				10	20.13	20.74	55.41	59.36	56.81	59.97	15.32	36.79	43.88	75.97	9.52	46.65
				15	18.24	18.86	31.21	51.35	51.39	54.09	18.53	31.25	55.44	74.99	11.39	69.97
c7552	108	1897	7000	1	28.45	97.42	100	100	100	100	1.70	98.22	3.95	60.86	5.99	5.99
				5	23.99	86.56	99.97	82.73	94.13	99.91	9.93	91.24	18.39	62.55	9.81	29.96
				10	17.54	51.68	94.65	78.78	80.05	98.23	15.29	83.68	27.79	70.43	12.23	59.93
				15	12.22	39.33	87.92	60.47	75.22	91.07	18.98	79.71	34.46	74.04	14.64	89.89
s9234	275	4090	7000	1	100	100	99.94	100	100	100	0.80	76.41	3.13	43.53	2.78	2.78
				5	98.21	99.57	98.95	95.08	99.12	99.98	3.72	79.56	7.92	61.79	4.55	13.90
				10	96.76	97.18	98.58	94.45	97.66	99.53	7.90	82.86	12.66	70.81	5.67	27.80
				15	96.39	96.95	98.46	92.66	95.87	96.98	11.01	82.34	16.32	70.29	6.79	41.69
sin	25	5416	7000	1	32.91	67.11	43.08	50.91	80.94	66.83	1.37	38.64	4.77	70.78	2.10	2.10
				5	22.32	34.75	37.68	49.97	72.75	63.87	5.76	26.67	26.80	71.51	2.78	10.49
				10	18.27	25.39	28.43	46.84	65.46	61.39	9.19	22.31	43.82	71.44	3.62	20.99
				15	12.19	19.04	22.86	42.22	54.67	56.07	11.40	22.28	56.44	70.53	4.47	31.48
voter	1	13758	10000	1	42.31	65.15	82.78	48.32	73.57	90.74	90.12		95.91		0.83	0.83
log2	32	32060	10000	1	39.56	50.54	76.23	69.49	79.97	79.97	1.94	70.84	3.60	82.09	0.35	0.35
				5	41.92	42.29	47.95	62.35	66.95	68.15	9.72	62.86	19.25	74.88	0.47	1.77
				10	36.96	39.03	42.23	57.83	59.77	61.92	13.88	44.78	33.39	74.07	0.61	3.55
				15	33.00	37.85	39.68	51.17	57.46	55.48	18.88	35.29	43.78	72.93	0.75	5.32

simulation consists on injecting faults in the devices of an IoT system using D&C and another using the proposed technique and comparing the error coverage, the area, and power costs.

A. Error coverage simulation

The evaluation of this technique was performed for errors induced by stuck-at faults and multiple bit-flips, as these error models produce a behaviour similar to that of long duration intermittent faults occurring in-the-field [8], [9]. Unbiased workloads of different sizes of uncorrelated random patterns were applied during simulations. Single stuck-at injection simulation of all possible faults sites is performed to calculate the error coverage (EC) of errors induced by single stuck-at faults (ibSSA). Additionally, multiple bit-flips are injected to emulate upsets in sequential elements at the inputs of the monitored logic cones. Errors at the output are those where the bit-flip bypasses the inherent logic masking of the cone from the input to the output. These errors are used to compute the EC of errors induced by bit-flips (ibBF). The cones selected to monitor were those that exhibited the highest number of errors. However, as mentioned in section III.A, this selection may be based on different vulnerability analyses.

Table I presents the results obtained by applying the proposed monitoring technique to a subset of the ISCAS'89 and EPFL'15 benchmarks. The first column shows benchmark circuit, followed by the number of logic cones and the area given in number of gates in the circuit. The next column shows the error detection latency (EDL), which is given by the workload size S required for the online signal probabilities to converge. Following is the number of monitored cones $C = [1, 5, 10, 15]$. The next columns present the EC of errors induced by 1, 2, or 3 input bit-flips in the selected cones, which are calculated as shown in (3). Similarly, the ibSSA EC of the selected cones and of the whole circuit, which is calculated according to (4), are also shown.

$$Selected\ Cones\ EC = \frac{\sum_{k=1}^{C} Sel(EC)_k \cdot Sel(E)_k}{\sum_{n=1}^{C} Sel(E)_n} \quad (3)$$

$$Whole\ EC = \frac{\sum_{k=1}^{C} Sel(EC)_k \cdot Sel(E)_k}{\sum_{n=1}^{T} E_n} \quad (4)$$

where C is the number of selected cones, T is the total number of cones, E_n is the number of errors at each cone, $Sel(E)_k$ is the number of errors in the selected cones and $Sel(EC)_k$ is the EC of each of the selected cones obtained with the different signature windows. The last columns in Table I show the area cost for both the single and multiple counter designs.

The EC of the whole circuit increases as more cones are monitored. When all cones are monitored, the EC of the selected cones and of the whole circuit converges to the maximum EC observable for each signature window. Using a signature window $w = M_{sp} \pm \sigma$ for circuit *log2*, Table I shows an ibBF EC on the selected cones of 79.97% when monitoring 1 cone, with an area cost of 0.35%, and an EC of 55.48% when monitoring 15 cones, with an area cost of 0.75%. Note that the ibBF EC is higher for 3 input bit-flips than for 1 input bit-flip. This is expected, as more bit-flips are more likely to propagate errors to the output, producing a more observable SEB. Additionally, Table I also shows an ibSSA EC of 43.78% and 72.93% on the whole circuit and the 15 selected cones respectively, with the same area cost of 0.75%. If all 32 logic cones are monitored, the ibSSA EC of the whole circuit increases to 71.85% with an area cost of 1.24% using a single counter monitor. The results using a signature window $w = M_{sp} \pm 3\sigma$ of the four largest circuits show an average ibBF and ibSSA EC of 75.5% and 69.1% respectively, with an average area cost of 1.52%, when monitoring the logic cone that exhibits the most errors. Using a signature window $w = M_{sp} \pm 3\sigma$, we can see an average ibBF and ibSSA EC of 84.4% and 73.1% respectively, with the same average area cost of 1.52%. An error detection latency estimation for these circuits synthesizing them with a standard 90nm cell library results in operating frequencies in the range of [3MHz, 1.1GHz], which produces an error detection latency in the range of [0.01, 3.3] milliseconds when detecting SEB

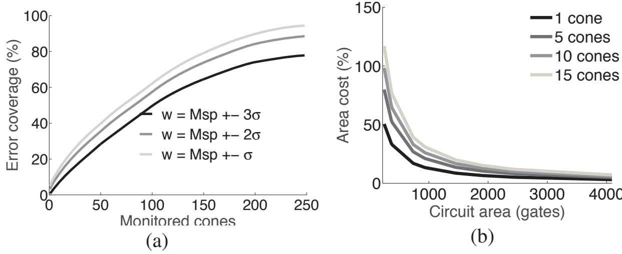


Fig. 5: (a) ibSSA EC vs monitored cones for circuit s9234. (b) Area cost of single counter monitors vs the size of the monitored circuit.

after 10000 clock cycles.

Figure 5b presents the trend of the area cost of the monitoring architecture versus the size of the monitored circuit. For the larger circuits, the area cost percentage is lower than for smaller circuits. The area cost of the monitors is only dependent on the number of monitored cones (C) and the area of each monitor. The area of each monitor is determined by the workload size S, denoted by $m(S)$ (Fig. 4). The area cost of a multiple counter design is presented in (5), where $size$ is the size of the monitored circuit.

$$Cost = \frac{C \cdot m(S)}{size} \tag{5}$$

The number of POs of logic circuits is bounded due to physical constraints, therefore, monitoring only POs would incur in a relatively low area cost. An estimation of the implementation of this technique for the circuit *twentythree* of the EPFL benchmarks with more than 23 million gates, indicates that monitoring all 68 of the PO would incur an approximate area cost of 0.0031% using a single counter design, and 0.033% using a multiple counter design.

The EC for narrow signature windows w is higher than for wide windows. The three signature windows $w = M_{sp} \pm \{3, 2, 1\}\sigma$ shown in Figure 5a, have a 0.3%, 4.5% and 31.7% respective probability of raising a false alarm. Narrower windows are stricter on the signal probability variations that can be detected, resulting in higher EC. Narrower windows detect SEB at a higher rate than wider windows, however, some of these detections may be false alarms. For maintainability planning purposes, a device that exhibits SEB at a higher rate than other identical devices in-the-field, may be prioritized for maintenance even if some SEB detections are false.

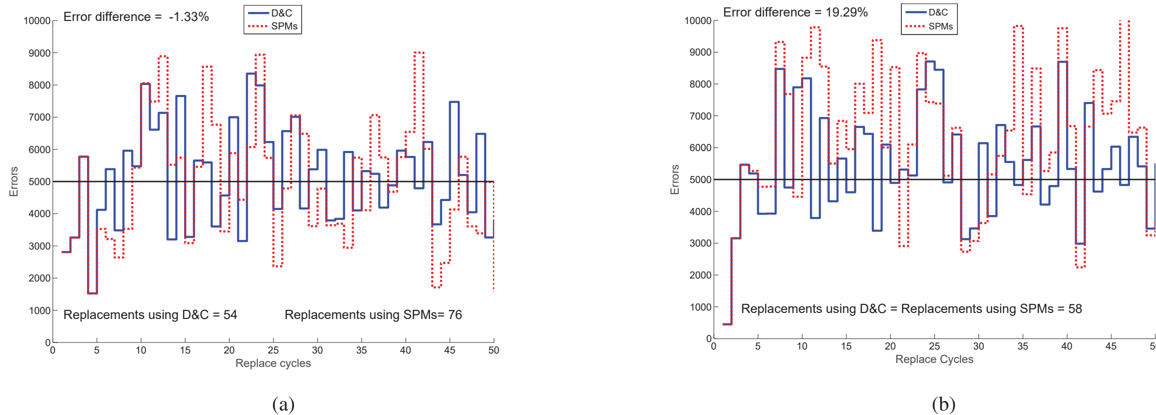
B. Maintainability planning simulation setup and results

Two maintainability simulations of an IoT systems, consisting of *six* voter circuits of the EPFL benchmarks were performed. The simulations consist of injecting a different random fault in each of the circuits while executing an unbiased workload of 10000 random patterns. The errors produced by these faults are detected using duplication and comparison (D&C) and the proposed technique with a signature window w=$M_{sp} \pm 3\sigma$. To the best of our knowledge, this is the first work of low cost error monitoring for IoT applications, thus we compare with D&C. For the D&C error detection, the maximum number of tolerable errors before a replace is necessary has been set to 5000 errors. When the total number of errors in the system (all six circuits) is greater than 5000, the *two* circuits with the highest number of errors are replaced in the next replace cycle. In the case of the proposed technique, the maximum number of SEB detections was set to different values according to the target number of replacements. When the maximum number of SEB detections have occurred, the *two* circuits with the most SEB detections are replaced in the next replace cycle.

Figure 6 presents the results of the maintainability simulations. The error difference is defined as the difference in the number of errors in the system when using the SPM technique compared to using D&C, averaged over the 50 replace cycles. Figure 6a shows the errors in the system when the maximum SEB detection number is set to 5 in order to observe a similar number of errors between the SPMs and D&C. The error difference of -1.33% indicates that the system using SPMs exhibits marginally less errors than D&C, to achieve this however, the system using SPMs must replace 76 circuits compared to 54 when using D&C to meet the error constraints. On the other hand, Figure 6b, shows the results produced by setting the maximum number of SEB detections to 8. In this case, an error difference of 19.29% indicates that the system using SPMs exhibits nearly a fifth more errors than a system using D&C, while requiring the same number of circuit replacements.

1) Area cost and Power Considerations: The area cost can be calculated by considering the number of replacements necessary by each technique to comply with the constraints set, multiplied by the cost associated with each technique. Note that the area cost per device of the D&C technique is greater than 100%, and the cost of the proposed technique using SPMs is of 0.83%. For the first simulation with a similar number of errors (Figure 6a), the area cost of the D&C technique is greater than 5400% for the 54 replacements, while the area cost of the proposed technique results in 63.1% for the 76 device replacements required, a reduction of over 85x the cost of D&C. For the second simulation (Figure 6b), the D&C technique results in an area cost greater than 5800% and the proposed in 48.14%, a reduction of over 120x the cost of D&C over the 50 replace cycles. The power consumption of the proposed technique is similarly reduced when compared against D&C.

The communications power required to transmit the error or signal probability data must be taken into consideration when comparing both techniques. Using D&C, a single error bit per logic cone per transaction is enough to provide the required error data. With the proposed technique, two bytes (16 bits) per cone are required to send the signal probability data required by the SEB ranking software. For a workload of 10000 random patterns, the D&C technique must transmit 10000 bits. The proposed technique must send the two bytes that contain the signal probability data. This results in 625x less bits transmitted using the proposed technique over D&C. Furthermore, if the D&C is adapted to count the number of errors on-chip and send that number as two bytes of data, the required power to transmit the data would be the same,



(a) (b)

Fig. 6: Maintainability simulation of D&C and SPMs with (a) similar number of errors in the system (b) and with equal number of replacements of D&C and SPMs

TABLE II:
Area and power cost of D&C and SPMs.

Circuit	Similar Error Diff.		Same Replacements		Comms. Power Red.
	Area Red.	Error Diff. (%)	Area Red.	Error Diff. (%)	
sin	26.1x	-1.1	47.6x	53.8	437.5x
voter	85.6x	-1.3	120.5x	19.3	625.0x
log2	158.7x	1.4	263.2x	40.6	625.0x

but with an even greater area cost. Table II shows the results after applying the proposed technique to the cone that exhibits the most errors of the three EPFL benchmarks examined. The area cost reduction and the error difference for the case with similar error coverage and with the same number of replacements are presented as well as the estimated reduction in communications power. The area cost is reduced by 26x to 263x and the communications power by 427x to 635x.

V. Conclusions

In this paper, we presented a novel low cost error monitoring technique to assist the maintainability planning of low power IoT applications that may tolerate some errors, by ranking devices based on the amount of errors they exhibit. On-chip signal probability monitors were used to collect the signal probability information at the outputs of each device which is then transmitted to the system software through the communications channel where the SEB ranking module ranks them according to the SEB they exhibit. The proposed technique was evaluated considering the SEB detection capabilities of the SPMs and by performing a maintainability simulation to compare the cost and error coverage of the proposed technique compared to D&C. For the SEB detection evaluation we injected multiple bit-flips and stuck-at faults on a set of the EPFL and the ISCAS benchmarks. Results demonstrate an average error coverage of 84.4% and 73.1% of errors induced by intermittent bit-flips and intermittent stuck-at faults, respectively, with an average area cost of 1.52%. The maintainability simulation showed that the proposed technique achieves a reduction of 26x to 263x in area cost, and requires over 625x less power for communications, when compared against a technique based on D&C.

Acknowledgments

This work has been supported by the Mexican CONACYT and by the EPSRC (UK) under grant no. EP/K034448/1.

References

[1] J. Paradells, C. Gómez, I. Demirkol, J. Oller, and M. Catalan, "Infrastructureless smart cities. Use cases and performance," *International Conference on Smart Communications in Network Technologies*, 2014.

[2] H. Al-Asaad, "Efficient techniques for reducing error latency in on-line periodic built-in self-test," *IEEE Instrumentation and Measurement Magazine*, vol. 13, no. 4, pp. 28–32, 2010.

[3] N. Touba and E. McCluskey, "Logic Synthesis Techniques for Reduced Area Implementation of Multilevel Circuits with Concurrent Error Detection," vol. 16, no. 7, pp. 651–654, 1997.

[4] C. Metra, D. Rossi, M. Omaña, A. Jas, and R. Galivanche, "Function-inherent code checking: A new low cost on-line testing approach for high performance microprocessor control logic," *Proceedings - 13th IEEE European Test Symposium, ETS 2008*, pp. 171–176, 2008.

[5] N. Karimi, M. Maniatakos, A. Jas, C. Tirumurti, and Y. Makris, "Workload-cognizant concurrent error detection in the scheduler of a modern microprocessor," *IEEE Transactions on Computers*, vol. 60, no. 9, pp. 1274–1287, 2011.

[6] Y. Qassim and M. E. Magana, "Error-tolerant non-binary error correction code for low power wireless sensor networks," *The International Conference on Information Networking 2014 (ICOIN2014)*, pp. 23–27.

[7] M. D. Gutierrez, V. Tenentes, D. Rossi, and T. Kazmierski, "Low power probabilistic online monitoring of systematic erroneous behaviour," in *22nd IEEE European Test Symposium 2017. ETS. Proceedings.*, 2017.

[8] J. Gracia-Moran, J. C. Baraza-Calvo, D. Gil-Tomas, L. J. Saiz-Adalid, and P. J. Gil-Vicente, "Effects of intermittent faults on the reliability of a reduced instruction set computing (RISC) microprocessor," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 144–153, 2014.

[9] D. Gil-Toms, J. Gracia-Morn, J. C. Baraza-Calvo, L. J. Saiz-Adalid, and P. J. Gil-Vicente, "Injecting intermittent faults for the dependability assessment of a fault-tolerant microcomputer system," *IEEE Transactions on Reliability*, vol. 65, no. 2, pp. 648–661, June 2016.

[10] M. D. Gutierrez, V. Tenentes, and T. J. Kazmierski, "Susceptible workload driven selective fault tolerance using a probabilistic fault model," in *Proc of 22nd IEEE International Symposium on On-Line Testing 2016*.

[11] M. D. Gutierrez, V. Tenentes, D. Rossi, and T. J. Kazmierski, "Susceptible workload evaluation and protection using selective fault tolerance," *Journal of Electronic Testing*, vol. 33, no. 4, pp. 463–477, Aug 2017.

[12] S. S. Mukherjee, C. Weaver, J. Emer, S. K. Reinhardt, and T. Austin, "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor," *Proc. International Symposium on Microarchitecture*, vol. January, pp. 29–40, 2003.

[13] C. Zhao, S. Dey, and X. Bai, "Soft-spot analysis: Targeting compound noise effects in nanometer circuits," *IEEE Design and Test of Computers*, vol. 22, no. 4, pp. 362–375, 2005.

[14] "The EPFL combinational benchmark suite." [Online]. Available: http://lsi.epfl.ch/benchmarks