DOI: https://doi.org/10.46354/i3m.2019.imaaca.002

FAILURE PROGNOSIS OF EMBEDDED SYSTEMS BASED ON TEMPERATURE DRIFT ASSESSMENT

Oussama Djedidi$^{(a)}$**, Mohand A. Djeziri**$^{(b)}$**, Samir Benmoussa**$^{(c)}$

$^{(a)(b)}$Aix Marseille University, Université de Toulon, CNRS, LIS, SASV, Marseille, France

$^{(a)(b)}$Laboratoire d'Automatique et de Signaux de Annaba (LASA), University Badji Mokhtar Annaba, 23000 Algeria.

$^{(a)}$oussama.djedidi@lis-lab.fr, $^{(b)}$mohand.djeziri@lis-lab.fr, $^{(c)}$benmous2a.samir@gmail.com

ABSTRACT

The Systems-on-Chip provide a large capacity for calculation and monitoring, so they are increasingly integrated into risky processes such as aeronautical and power generation systems. However, embedded systems are subject to degradation caused by wear, that can be accelerated by the often hostile environment. This paper proposes a method of failure prognosis of embedded systems based on the estimation of the temperature drift under reference operating conditions, then the modelling of the drift trend using a support vector regression model. The remaining useful life is estimated using the integral of the probability density function of the time to failure. Experimental results, evaluated by performance analysis techniques, show the effectiveness of the proposed approach.

Keywords : Remaining Useful Life; support vector regression; Fault Diagnosis; Systems-on-Ship

1. INTRODUCTION

The development of computer science and artificial intelligence has contributed to the gradual replacement of several functions of management and control in systems at risk by embedded electronic devices. This evolution must be accompanied by the development of methods of fault diagnosis and failure prognosis to ensure the operation of these systems safely and prevent any damage or failure. In this context, part of existing work consider electronic cards as discrete systems integrating several components, and whose functioning depends on proper operation of all the components, assuming that there is a strong dependence of operation between each component (Steininger 2000). In this caser, the developed fault diagnosis methods are based on causal models such as multi-signal flow graph (Deb, Pattipati, Raghavan, Shakeri & Shrestha 1995), information flow model (Sheppard 1996), directed graph (Zhang 2005) and the fault tree (Wang, Shi & Wang 2011). A classification and a comparative analysis of existing fault diagnosis methods applied on multicore processor architectures are proposed in Gizopoulos, Psarakis, Adve, Ramachandran, Hari, Sorin, Meixner, Biswas & Vera (2011) where four categories are ditinguished: redundant execution approaches which exploit the inherent replication of processor cores and threads in a multicore processor architecture (Aggarwal & Ranganathan 2007, LaFrieda, Ipek, Martínez & Manohar 2007, Mukherjee, Kontz & Reinhardt 2002), periodic Built-In Self-Test (BIST) approaches which advocate leveraging the built-in test mechanisms of processors traditionally used for manufacturing testing (Shyam, Constantinides, Phadke, Bertacco & Austin 2006, Foutris, Psarakis, Gizopoulos, Apostolakis, Vera & Gonzalez 2010), dynamic verification approaches (Austin 1999, Meixner, Bauer & Sorin 2007), and anomaly detection approaches (Racunas, Constantinides, Manne & Mukherjee 2007, Wang & Patel 2006, Li, Ramachandran, Sahoo, Adve, Adve & Zhou 2008). The results of the comparative study highlight the effectiveness of the dynamic verification approaches in targeting both transient faults, permanent faults and design bugs.

Several methods of failure prognosis have been developed in recent years. They can be classified in three categories: modal-based (Feng, Kvam & Tang 2016), data-driven (Nguyen, Djeziri, Ananou, Ouladsine & Pinaton 2016), and hybrid approaches that combine physical knowledge with data analysis for failure prognosis (Benmoussa & Djeziri 2017, Djeziri, Nguyen, Benmoussa & M'Sirdi 2017). These methods have been applied to electrical systems, mechatronics and renewable energy systems, but to our knowledge, there has been no approach applied to processing units. So, we propose in this paper the application of a failure prognosis method to a processing unit based on the monitoring of the temperature drift generated by the SoC under reference conditions of use. The link between the drift of temperature and the degradation process is formally demonstrated and found experimentally in several works (Hong & Kim 2010, Djedidi, Djeziri & M'Sirdi 2018), it can therefore be used as a health index under reference operating conditions. To detect this drift, a reference model, data-driven, is built, then the trend of this drift is modelled by a Support-Vector Regression (SVR), and the online projection of the estimate up to a predefined maximum drift allows the Remaining Useful Life (RUL) prediction.

This paper is organised as follows: Section 2 is devoted to an overview of the proposed approach. The system, its operation, its inputs and outputs as well as its modelling for the estimation of the temperature are described in Section 3. In Section 4, the HI generation and evaluation are presented, then the tend modelling method is described in section 5 and the RUL prediction method in section 6. The results obtained are shown and analysed in Section 7, then a conclusion and perspectives are given in section 8.

2. RUL PREDICTION SCHEMA

The proposed approach is summarised in four steps as illustrated in Fig. 1. The first step is devoted to the temperature modelling using an Autoregressive-moving-average (ARMAX) model. The second step concerns the Health Index (HI) generation, which represent the difference between the measured SoC temperature online and the estimated one. The trend of the temperature drift is modelled in the third step using a Support Vector Regression (SVR) with a Gaussian kernel. The RUL expectation is then estimated in the fourth step using the time integration of the Probability Density Function (PDF) until the predefined time-to-failure.

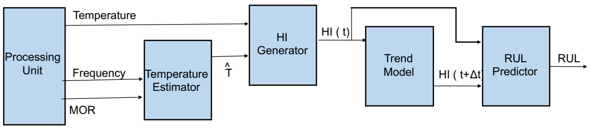

Figure 1: Overview of the RUL estimation process

3. TEMPERATURE MODEL

Experimental tests presented in (Hong & Kim 2010), show that the variation of the temperature of a Sytem-on-Chip (SoC) is influenced by two variables: the power consumed and the memory intensity (rate of access to the GDDR), and that it is the memory intensity which is the most influential on the variation of the temperature. This work also shows that the power consumed depends on the thermal voltage, and the thermal voltage is a function of the temperature, so there is an algebra-differential loop between the temperature and the power. Going back the cause-and-effect links between these two variables and the load, we find that the first common input is the frequency. Thus in the model proposed in this paper, the inputs considered are the frequencies of the CPU and the GPU, the Memory Ooccupation Rate (MOR) and the previous temperature of the SoC. The MOR is chosen because the memory intensity is not measurable.

To built a temperature estimator taking into account the regression property of the temperature, an autoregressive–moving-average (ARMAX) model is implemented. This models use the regression of inputs and previous outputs, along with the moving average to estimate the current output:

$$\begin{aligned} y(k) &= P_1 y(k-1) + \ldots + P_n y(k-n) \\ &+ Q_1 u(k-1) + \ldots + Q_m u(k-m) \\ &\qquad + e(k) + H_1 e(k-1) + \ldots + H_r e(k-r) \end{aligned} \tag{1}$$

Equation (1) is the linear difference equation of an ARMAX (n, m, r) (Orders of the model), with $y(k)$ being the output to compute, u the exogenous (X) variable or system input, and e is the moving average (MA) variable (Fung, Wong, Ho & Mignolet 2003). In the case of this work, temperature T_{SoC} is the output $y(k)$ to be estimated, and the system input is $u(k) = [f_1, \ldots, f_4, f_{GPU}, MOR]$.

4. HEALTH INDEX GENERATION

In this work, the HI is generated by comparing the measured temperature with a reference one estimated by the model of normal operation presented above. This choice is motivated by the fact that this approach is well suited to detect a progressive deviation of the temperature drift from its value in normal operation. The HI is calculated by the following equation:

$$HI = (T_{estm}(t) - T_{meas}(t))/T_{estm}(t) \tag{2}$$

The launch of the drift trend model for RUL prediction begins when a drift start is detected. To do so, HI is evaluated by a statistical method after analysing the data distribution. As the raw HI is normally distributed around its mean values, 99% of the data are inside the envelope delimited by the positive and negative value of mean plus three times the standard deviation. This property is used to calculate the normal operation thresholds.

Since the drift trend information is contained in the HI average and in order to reduce the false alarm rate, the raw HI is processed to extract its moving average HI_m. The averaged residual HI_m and the normal operating thresholds are expressed as follows:

$$\begin{cases} HI_m & = \frac{1}{n} \sum_{i=1}^{n} HI \quad , \\ th_{HI}^{+} & = \quad \|HI_m + 3 \times \sigma_{HI}\| , \\ th_{HI}^{-} & = - \; \|HI_m + 3 \times \sigma_{HI}\| \end{cases} \tag{3}$$

σ_{HI} is the standard deviation of the HI signal.

5. TREND MODELLING

The long-term prediction of the evolution of a variable whose dynamics are unknown and complex, such as the SoC ageing process, can not be done by linear models like the $ARMA$ or $ARMAX$. For efficient prediction it is necessary to choose a model that supports nonlinear dynamics and variations of initial conditions. In this context, two tools stand out and are widely used, neural networks and SVR. In this work we choose the SVM as this tool is more suitable for unsupervised learning cases. The expression of the prediction by SVR at a step of the evolution of the health index is given as follows:

$$\hat{HI}(k+1) = \sum_{[i=(m-1)\tau+1]}^{N-1} \alpha_i^* K(HI(i), HI(k)) + b^* \quad (4)$$

where α_i^* are Lagrange multipliers and τ the delay. The standard SVR toolbox is used without making any special changes to the prediction of temporal overlays. The free parameters, C, ε, the size of the kernel (Gaussian), and the dipping dimension m are selected from a comprehensive search in the parameter space to optimize the performance of the prediction on the validation set. The available N observations are therefore shared between two sets of training and validation of respective sizes Ne and Nv. Values for which the prediction error at a step on the validation set is minimal are retained for the final prediction. Once the parameters are fixed, we read the prediction using all the N observations available. The predictions at several steps, ie, for the values $(k \geq N+1)$, are realized by the ration of the prediction at one step, using the estimated vectors $\hat{HI}(k)$ at the previous iterations and not the observations themselves.

6. RUL ESTIMATION

In this case study, the RUL is expressed as the expectation of the remaining time for HI to reach the threshold Th_p and given as follows:

$$Z_{t_j} = \inf\left\{x : HI\left(t+x\right) \leq Th_p \left| HI(t) > Th_p \right.\right\} (5)$$

$$RUL\left(t_j\right) = E\left\{Z_{t_j}\right\} \quad (6)$$

The Probability Density Function (PDF) of Z_{t_j} is expressed as follows:

$$pdf_{Z_{t_j}}(x) = \frac{-\left(Th_p - HI(t_j)\right)}{\sqrt{2.\pi.x^3.\sigma^2}}.e^{-\left(\frac{\left(Th_p - HI(t_j) - \mu_{t_j}.x\right)^2}{2.x.\sigma^2}\right)} \quad (7)$$

Thus, $RUL_{t_j} = \int_0^{+\infty} pdf_{Z_{t_j}}(x)xdx$. The confidence interval from 5% to 95% probability is $[RUL_{t_j}^-, RUL_{t_j}^+]$ where $\int_0^{RUL_{t_j}^-} pdf(x)dx = 0.05$ and $\int_0^{RUL_{t_j}^+} pdf(x)dx = 0.95$.

7. APPLICATION

The development board used in this case study is shown in Fig 2. This device can run on Linux and Android, and is equipped with a ARM Cortex-A9 core processor and a core with 1 giga-byte of RAM. This card is not equipped with sensors to measure the power consumed, so an external ammeter and an oscilloscope are used to measure the power consumed (see Fig 2).

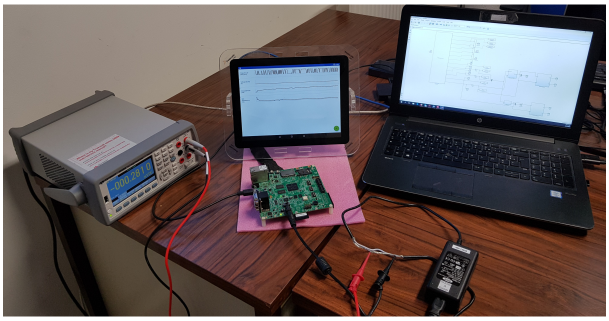

Figure 2: Overview of the application support including the development board, the display and the monitoring system.

In addition to the real SoC, a SoC simulator, developed and validated with real data in (Djedidi et al. 2018) is embedded in the monitoring system. The prediction algorithm proposed in this paper is validated using two types of degradation data: the first type are simulation data generated by the simulator. The second type is experimental data obtained on the real SoC over 4 years of use, these data are increased by data generated by the simulator to achieve the total failure. In This application case, the total failure is considered when the temperature drift is about 10 ° °C.

7.1. Validation of the diagnosis module

To validate the diagnostic part of the proposed approach, the algorithms for generating HI, moving-average filtering and generating normal operating thresholds are implemented in the supervisor. The results obtained are shown in Fig 3) for the normal operation and in Fig 4) for the case of degradation beginning emulated by overheating the SoC using a light projector of $1000W$.

7.2. Validation of the prognosis module

To validate the prediction model of the temperature drift, a SoC temperature drift was generated using the simulator. The result of the prediction is given in Fig 5. The SVM model inputs are the actual values of the drift. The result of RUL prediction follows the performance of the degradation prediction, and the real is estimated accurately. This first test shows that when the prediction is done over a short-time (with a permanent update of the measured drift) the result is very satisfactory with a negligible prediction error.

In the second test, it is assumed that the profile of HI is available until the time $t = 3000h$. From this moment, the inputs of the trend model are the values of HI estimated by the same model at the previous instants. The results are given in the Fig 6.

The results show that when the prediction is done in the medium term, the estimation error of the RUL is more important but remains acceptable as it represents $5,17\%$ of the real RUL.

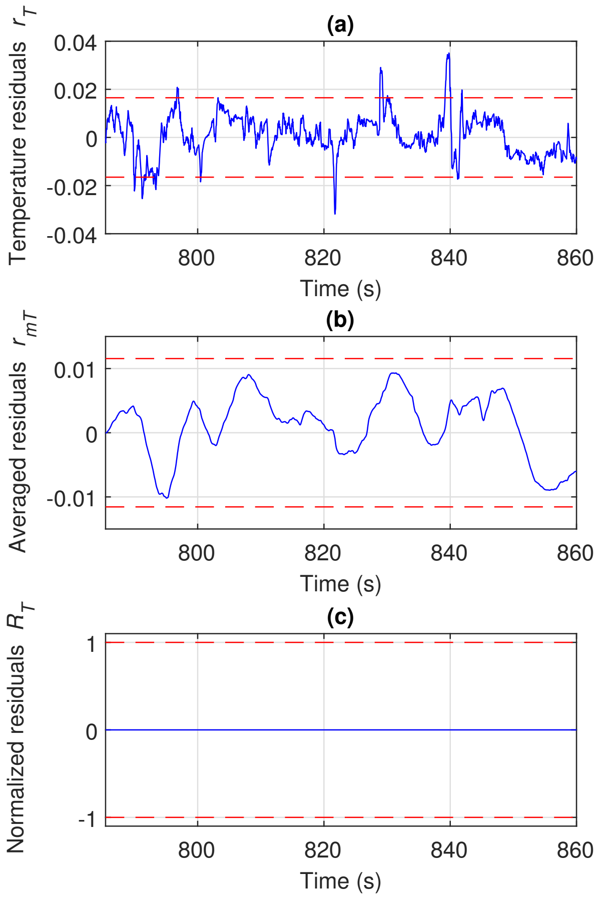

Figure 3: Health index profile in normal operation. (a): Raw HI, (b): Filtered HI, (c): Normalised HI

In the last test, we assume that HI's profile is available until $t = 2500h$. From this moment, the inputs of the trend prediction model are the values of HI estimated by the model at the previous instants. The results are given in the Fig 7.

The results show that when the prediction is long-term, the estimation error of the RUL is larger but is still acceptable as it represents 8% of the real RUL.

8. CONCLUSION

In this work, a generic method of failure prognosis of SoC is proposed in two major steps. The first step concerns the modelling of the SoC temperature and the generation of a health index characterising the SoC degradation process. This index is the drift of the temperature which represents the difference between the temperature generated by the SoC and a temperature reference generated by the ARMAX model. The second step is a prognostic one that concerns the estimation of the evolution of the short-term, medium-term and long-term temperature drift using a Support Vector Regression model. The RUL is estimated in this work using the PDF integration of the times to failure. The results obtained show the effectiveness of the proposed method on important prognostic horizons.

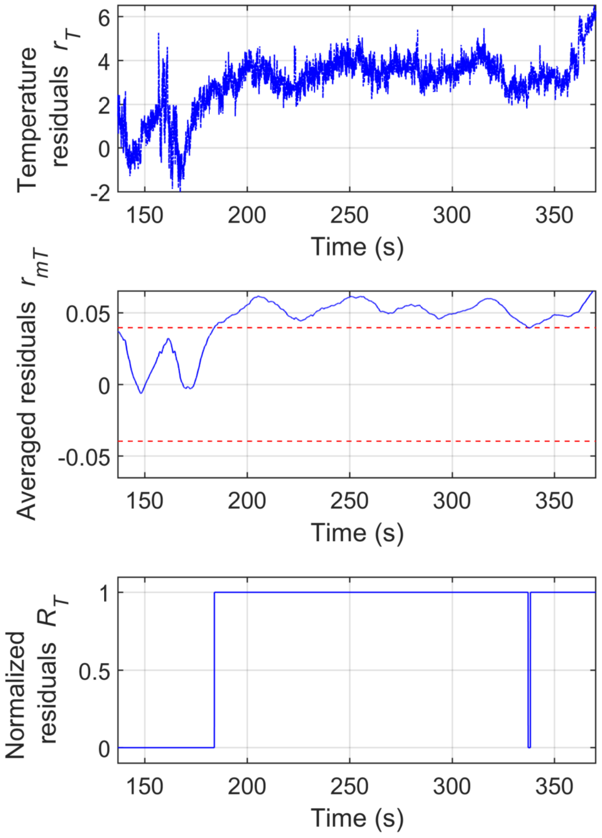

Figure 4: Health index profile at the beginning of the degradation. (a): Raw HI, (b): Filtered HI, (c): Normalized HI

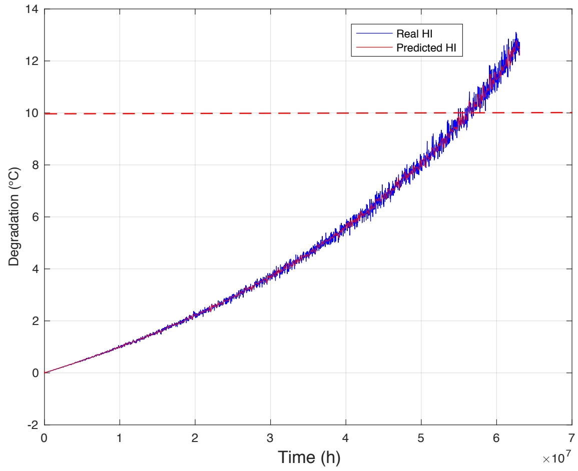

Figure 5: Degradation-trend and RUL prediction using one-step SVR model from the raw HI

REFERENCES

Aggarwal N. & Ranganathan P., (2007). Configurable isolation: building high availability systems with commodity multi-core processors. In: Acm Sigarch . . . , vol. 35, New York, New York, USA: ACM Press, vol. 35, 470–481.

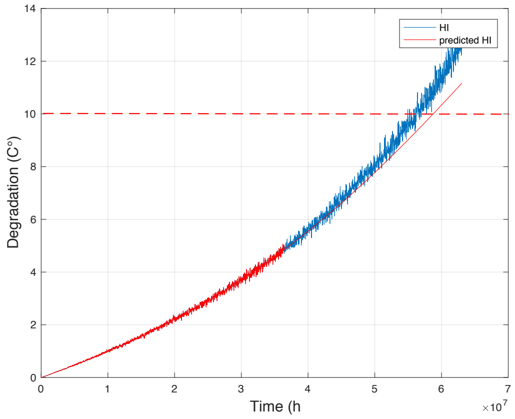

Figure 6: Degradation-trend and RUL prediction in the medium term using SVR model from the raw HI

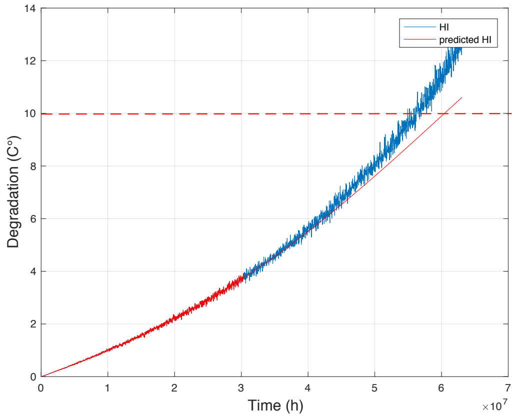

Figure 7: Degradation-trend and RUL prediction in the long-term using SVR model from the raw HI

Austin T.M., (1999). DIVA: A reliable substrate for deep submicron microarchitecture design. In: 32nd Annual Int. Symp. on Microarchitecture, (MICRO-32), IEEE Comput. Soc, November, 196–207.

Benmoussa S. & Djeziri M.A., (2017). Remaining useful life estimation without needing for prior knowledge of the degradation features. IET Science, Measurement & Technology, 11 (8):1071–1078. **URL:** *https://digital-library.theiet.org/content/journals/10.1049/iet-smt.2017.0005*

Deb S., Pattipati K.R., Raghavan V., Shakeri M., & Shrestha R., (1995). Multi-Signal Flow Graphs: A Novel Approach for System Testability Analysis and Fault Diagnosis. IEEE Aerospace and Electronic Systems Magazine, 10 (5):14–25.

Djedidi O., Djeziri M.A., & M'Sirdi N.K., (2018). Data-Driven Approach for Feature Drift Detection in Embedded Electronic Devices. IFAC-PapersOnLine, 51 (24):1024–1029.

Djeziri M., Nguyen T.B.L., Benmoussa S., & M'Sirdi N., (2017). Fault prognosis based on physical and stochastic models. In: 2016 European Control Conference, ECC 2016, IEEE, 2269–2274. **URL:** *http://ieeexplore.ieee.org/document/7810629/*

Feng J., Kvam P., & Tang Y., (2016). Remaining useful lifetime prediction based on the damage-marker bivariate degradation model: A case study on lithium-ion batteries used in electric vehicles. Engineering Failure Analysis, 70:323–342.

Foutris N., Psarakis M., Gizopoulos D., Apostolakis A., Vera X., & Gonzalez A., (2010). MT-SBST: Self-test optimization in multithreaded multicore architectures. In: Proceedings - International Test Conference, IEEE, 1–10.

Fung E.H., Wong Y., Ho H., & Mignolet M.P., (2003). Modelling and prediction of machining errors using ARMAX and NARMAX structures. Applied Mathematical Modelling, 27 (8):611–627.

Gizopoulos D., Psarakis M., Adve S.V., Ramachandran P., Hari S.K.S., Sorin D., Meixner A., Biswas A., & Vera X., (2011). Architectures for online error detection and recovery in multicore processors. 2011 Design, Automation & Test in Europe, (c):1–6.

Hong S. & Kim H., (2010). An integrated GPU power and performance model. In: ACM SIGARCH Computer Architecture News, vol. 38 of {ISCA} '10, New York, NY, USA: ACM, vol. 38 of {ISCA} '10, 280.

LaFrieda C., Ipek E., Martínez J.F., & Manohar R., (2007). Utilizing dynamically coupled cores to form a resilient chip multiprocessor. In: Proceedings of the International Conference on Dependable Systems and Networks, IEEE, 317–326.

Li M.L., Ramachandran P., Sahoo S.K., Adve S.V., Adve V.S., & Zhou Y., (2008). Understanding the Propagation of Hard Errors to Software and Implications for Resilient System Design. SIGOPS Oper. Syst. Rev., 42 (2):265–276.

Meixner A., Bauer M.E., & Sorin D., (2007). Argus: Low-Cost, Comprehensive Error Detection in Simple Cores. In: 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007), IEEE, 210–222.

Mukherjee S.S., Kontz M., & Reinhardt S.K., (2002). Detailed design and evaluation of redundant multithreading alternatives. In: 29th Annual International Symposium on Computer Architecture, 2002. Proceedings, IEEE, 99–110.

Nguyen T.B., Djeziri M., Ananou B., Ouladsine M., & Pinaton J., (2016). Fault prognosis for batch production based on percentile measure and gamma process: Application to semiconductor manufacturing. Journal of Process Control, 48:72–80.

Racunas P., Constantinides K., Manne S., & Mukherjee S.S., (2007). Perturbation-based fault screening. In: Proceedings - International Symposium on High-Performance Computer Architecture, IEEE, 169–180.

Sheppard J., (1996). Maintaining diagnostic truth with information flow models. In: Conference Record. AUTOTESTCON '96, IEEE, 447–454.

Shyam S., Constantinides K., Phadke S., Bertacco V., & Austin T., (2006). Ultra low-cost defect protection for microprocessor pipelines. In: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems - ASPLOS-XII, vol. 41, New York, New York, USA: ACM Press, vol. 41, 73.

Steininger A., (2000). Testing and built-in self-test – A survey. Journal of Systems Architecture, 46 (9):721–747.

Wang F.W., Shi J.Y., & Wang L., (2011). Method of diagnostic tree design for system-level faults based on dependency matrix and fault tree. In: 2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management, IE and EM 2011, IEEE, PART 2, 1113–1117.

Wang N.J. & Patel S.J., (2006). ReStore: Symptom-based soft error detection in microprocessors. IEEE Transactions on Dependable and Secure Computing, 3 (3):188–201.

Zhang G., (2005). Optimum Sensor Localization/Selection In A Diagnostic/Prognostic Architecture. Phd dissertation, University System of Georgia.