System Level Techniques to Improve Reliability in High Power Microcontrollers for Automotive Applications

Andrea Acquaviva and Massimo Poncino
DAUIN - Politecnico di Torino
10129 Torino, Italy
Email: [andrea.acquaviva,massimo.poncino]@polito.it

Marco Otella and Michele Sciolla
Fiat Research Center
10043 Orbassano, Torino, Italy
Email: [marco.ottella,michele.sciolla]@crf.it

Abstract—In high power microcontrollers, a decrease in circuit lifetime is often observed in safetycritical applications where circuitry is subjected to the most severe stresses and reliability has become a major concern. Thus, ad-hoc design solutions become necessary to mitigate the impact of ageing.

In this paper we discuss hardware-software approaches that exploit distributed on-chip monitoring of wear-out parameters to perform ageing-aware allocation of computation and recovery periods on the various computational units.

I. INTRODUCTION

One of the major industrial priorities in the transportation field concerns the area of reference designs and architectures in order to offer common architectural approaches (standardised and interoperable) for future electrical vehicles. The new electrical vehicle architectures based on distributed embedded computing and electronics system will allow significant energy saving, with enhanced fun-to-drive while increasing safety, and comfort, and decreasing the overall complexity of the vehicle.

From a system point of view, next generation of electrical vehicles will interconnect and aggregate different domains, as schematically shown in Figure 1

- Energy sources: power electronic modules and control algorithms that perform functions such as power management (battery, super capacitors, etc.), and recharging (grid, range extender, on-board photovoltaics, etc.).
- Propulsion: electronics modules and control algorithms performing functions such as electrical motor control and distributed traction (1, 2, 4 motors).
- Chassis (Drive dynamic): electronics modules and control algorithms that perform functions such as steering, traction control, ABS, ESP, active suspensions, etc.
- Power and Signal Distribution (PASD): on board high voltage (power train) and low voltage (auxiliaries) bus, storage recharge bus, with related high voltage management features for safety and reliability.
- Vehicle Body and On-board Control: addressing chassis/body design for optimisation against conflicting requirements such as cost and strength, or performance and energy efficiency.
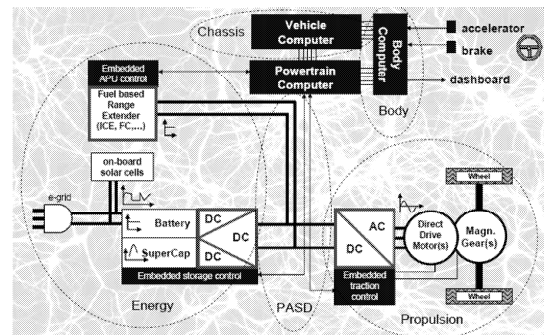
Fig. 1. Electrical vehicle domain partitioning.

Thus, realisation of energy efficient and cost-effective electrical vehicle with enhanced safety and comfort calls for dedicated development of distributed and embedded hardware and software systems for the automotive industry. According to the trends of semiconductor technologies in automotive Control Units, the forthcoming generations of microcontrollers will be required to provide large computational power capable of handling highly complex real-time Operating Systems and their applications will be even more critical and demanding in electrical vehicles, where most functions will be performed through electrical and electronic devices. Being in the context of safety-critical applications, all these capabilities must be provided in a reliable way.

II. MICROCONTROLLER RELIABILITY ISSUES

In high power microcontrollers, a decrease in circuit lifetime is often observed in safety-critical applications where circuitry is subjected to the most severe stresses and reliability has become a major concern. Reliability achieved through system-level techniques should both improve the reliability of single System on Chip (SoC) components and make it possible to exploit system level information to increase the overall chip lifetime [2, 3, 5].

Ageing aware architectural and software solutions will, in a proactive way, limit the number of faults and increase their predictability. Reliability and fault management, is one of the grand challenges of the semiconductor industry for automotive applications (in particular for safety critical systems) for which
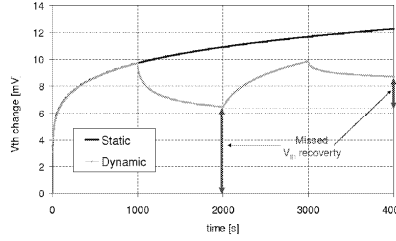
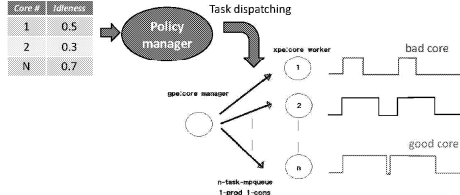Fig. 2.   NBTI effect: Threshold voltage degradation over time.

Fig. 3.   Adaptive idleness distribution policy description.

the main known degradation phenomena are the Hot Carrier Injection (HCI) and the Negative Temperature Bias Instability (NTBI) the latter being predominant. NTBI can be partially recovered by imposing a virtual ground during the idle period (Figure 2).

III. Key Directions for System Level Reliability Management

Such an integrated and collaborative approach between hardware and software will be the key to address the tight real-time requirements of automotive applications. A promising direction is the exploitation of data acquisition about system level and environmental condition (workload, EMI, temperature, humidity, and so on). More specifically, software strategies for ageing mitigation and reliability improvement will exploit the use of these data to evaluate system degradation and to perform suitable workload allocation and hardware resource management.

An ad-hoc system level policy will perform idleness distribution using a workload adaptive strategy: The policy manager will exploit information coming from each core about the level of degradation and will convert it into an idleness distribution table for each core. Based on this information the policy manager will decide task dispatching in such a way that aged cores have reduced activity.

IV. Task Allocation for Wear-out Control

The main mechanisms we exploit are the insertion of idleness (recovery) periods and the workload migration. The latter is used for two purposes: First, balancing the ageing where this cannot be obtained through the selective insertion of recovery periods, because the full utilization is required. Second, mitigating the performance impact of recovery periods by moving the computation to younger, idle cores when available.

In Figure 3 we depicts a possible scheme for the insertion of recovery periods on a multicore controller platform. The policy, using as inputs the wanted lifetime, the maximum frequency of the core and the threshold frequency (that is, the
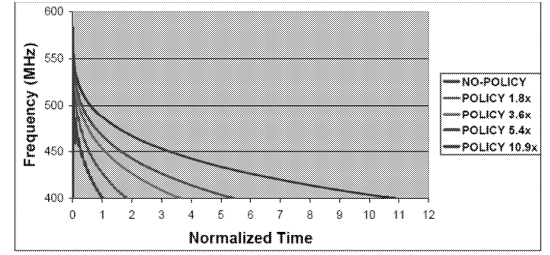

Fig. 4.   Maximum CPU frequency over the time, for different imposed lifetimes.

frequency below which the cpu is considered dead), it extracts the stress/recovery ratio that guarantees the desired lifetime. The algorithm periodically monitors the current ratio on each core and eventually forces the core to recover till the ratio is correct. Besides utilization, external factors can impact the expected lifetime. To take them into account, ageing monitors measuring the current circuit speed can be exploited [1, 4]. In Figure 4 we show how it is possible to control the degradation of the circuit speed due to NBTI by imposing recovery periods. It sketches the maximum frequency of one core over time, comparing the case without policy versus the case with the policy, for different imposed lifetimes. The time is normalized with respect to the lifetime of the case without policy as reference. The figure refers to a three cores platform running three tasks. However, recovery leads to a performance impact. Task migration can help mitigating this impact by moving the computation on an available core. Preliminary tests demonstrate that performance hit can be reduce by a factor of 2 depending on the task characteristics.

V. Conclusion

In this paper we discussed the main challenges concerning reliability in multicore microcontrollers in the context of automotive applications and we discussed a system level software approach to face them. In particular, we focused on compensation and control of wear-out effect such as NBTI through recovery insertion and task migration as viable strategy to achieve a good trade-off between lifetime and system performance.

References

[1] A. Drake, R. Senger, H. Singh, G. Carpenter, and N. James, "Dynamic measurement of critical-path timing," in *IEEE, Proceedings of the Conference on Integrated Circuit Design and Technology and Tutorial*, 2008, pp. 249–252.

[2] L. Huang, F. Yuan, and Q. Xu, "Lifetime reliability-aware task allocation and scheduling for mpsoc platforms," in *IEEE, Proceedings of the Conference on Design, Automation and Test in Europe*, 2009, pp. 51–56.

[3] F. Paterna, A. Acquaviva, F. Papariello, G. Desoli, M. Olivieri, and L. Benini, "Adaptive idleness distribution for non-uniform aging tolerance in multiprocessor systems-on-chip," in *IEEE, Proceedings of the Conference on Design, Automation and Test in Europe*, 2009, pp. 906–909.

[4] B. Rebaud, M. Belleville, E. Beigne, M. Robert, P. Maurine, and N. Azemard, "An innovative timing slack monitor for variation tolerant circuits," in *IEEE, Proceedings of the Conference on IC Design and Technology*, 2009, pp. 215–218.

[5] C. Zhuo, D. Sylvester, and D. Blaauw, "Process variation and temperature-aware reliability management," in *IEEE, Proceedings of the Conference on Design, Automation and Test in Europe*, 2010, pp. 580–585.