Reliability Concerns in Embedded System Designs

Vijaykrishnan Narayanan and Yuan Xie
The Pennsylvania State University

Designing a dependable embedded system atop a less reliable hardware platform poses great challenges.

Embedded computing systems have become a pervasive part of daily life, used for tasks ranging from providing entertainment to assisting the functioning of key human organs.

While mission-critical embedded applications raise obvious reliability concerns, unexpected or premature failures in even noncritical applications such as game boxes and portable video players can erode a manufacturer's reputation and greatly diminish widespread acceptability of new devices. The advent of more sophisticated embedded systems that support more powerful functions, and the reliance on deep submicron process technologies for their fabrication, have brought reliability concerns to the forefront.

Variability in fabricated circuit primitive parameters due to the statistical nature of manufacturing process, signal integrity issues arising from internal and external noise sources, and accelerated aging of the devices are three important categories of reliability concerns facing the design of embedded systems hardware. While these concerns also affect other computing system types, embedded systems pose unique challenges.

First, these systems are cost sensitive and often work with limited resources such as smaller memory size or diskless designs. These constraints can make it difficult to apply many traditional computer design methodologies for improving embedded systems reliability. For example, executing multiple redundant versions of the same thread to ensure reliable operation, while common in server environments, can be quite costly in embedded systems.

Second, providing reliable embedded systems operation while satisfying other stringent constraints such as power consumption and real-time throughput is essential. Consequently, reliability-oriented designs that rely on aggressive redundancy, such as triple-module redundancy, might not be viable from an energy viewpoint. Similarly, approaches such as checkpointing and rollback, used in many server environments, might have no relevance in real-time embedded applications if the recovery happens after a deadline.

Third, embedded systems often have reduced noise margins thanks to aggressive power optimizations or deployment in harsh environments. For example, the use of subthreshold circuits in ultralow-power environments such as sensor nodes significantly lowers the amount of external charge needed to upset a circuit node.

In addition to hardware reliability concerns, the increasing amount of software content in embedded systems also poses a major challenge. Many documented cases of embedded systems failures have been ascribed to software malfunction, including the recent incident of a hybrid car stalling suddenly at highway speeds.

PROCESS VARIABILITY

The challenges in fabricating small feature size transistors have resulted in significant variation in transistor parameters such as channel length, gate-oxide thickness, and threshold voltage across identically designed neighboring transistors and across different identically designed chips.

Parameter variations derive from different phenomena such as wafer misalignment, random dopant fluctuations, and imperfections in planarization steps. The divergence between designed and fabricated transistor parameters creates significant functional correctness concerns such as stability problems in memory cells, creation of new critical paths in the design, and increased leakage currents.

Although designing for worst-case process margins has been used as a traditional option when dealing with outliers, the degree of variability encountered in the new process technologies makes this a nonviable option. Process variations can cause up to 20-times variation in chip leakage and up to 30 percent variation in chip frequency. Cost sensitivity makes designing for the worst case unacceptable for embedded systems.

Process variations present a particular concern in memories that constitute a significant portion of embedded processors' silicon budget. Given that, for density reasons, developers typically design memories using minimum feature sizes, process variation effects

are most significant. On memory circuits, these effects can result in different reliability concerns for read stability, write failures, hold failures, or access-time increases.

Addressing these concerns will require techniques beyond traditional memory fault tolerance. Further, operating the entire memory based on the worst-case process-variation delay would be overly pessimistic and cause performance loss. Consequently, developers need new techniques that can recognize and adapt to the resulting variation. For example, boosting the supply voltage on word lines, using a larger configurable word line driver, or reducing the access transistors' threshold voltage by body biasing can partially compensate for the access-time increase in memory cells that process variation causes. Letting the software map latency-critical data to memory portions with shorter access times, while mapping the rest to portions that have higher access times, might possibly leverage the inherent tolerance to slower memory operations for certain data in an application.

Datapath circuits and interconnects also require techniques for coping with process variation. In datapath circuits, process variation can shift the critical path, causing timing failures. Further, designing for the worst-case corners of process variation requires large design margins. Therefore, developers need adaptive techniques that can permit the design to tune its frequency based on the encountered process variation. The temporal sampling technique proposed in the Razor chip developed at the University of Michigan offers a good example of such systems.

Interconnect variability can also cause significant performance deviations. Because the scattering effect makes wiring delays vary nonlinearly with wire widths for very small features, even small variations can lead to large timing differences. Consequently, adaptive techniques for dealing with reliability should also account for interconnect issues.

TRANSIENT ERRORS

Transient faults are errors caused by temporary conditions on the chip (such as when power supply noise or interconnect noise exceed a certain threshold) or by external noise (such as soft errors caused by neutrons striking the chip). The circuit itself is not damaged even though computational errors are introduced.

Developers knew about the *soft error*—also called a single-event upset—phenomenon in memory as early as the 1970s. As supply voltages diminish and feature sizes become smaller in future technologies, soft-error tolerance will become a significant challenge when designing future embedded systems.

Adaptive techniques for dealing with reliability should also account for interconnect issues.

Further, energy constraints could force embedded systems to incorporate aggressive power optimizations and deploy techniques such as dynamic voltage scaling, which can reduce the amount of critical external charge required to upset the value stored at the circuit node. Our accelerated soft-error testing experiment on a 4-Mbit SRAM memory for one hour showed that the number of errors increases from 57 to 658 when the operational voltage drops from 5 V to 4 V.

In the nanometer era, technology scaling means soft errors are no longer confined to memory cell upsets and can impact field-level product reliability for logic and latches. Additionally, with deeper pipelining, the number of logic stages between latches becomes smaller, increasing the probability of a single-event upset making its way into a latch. Soft errors in memory can be protected by error-correcting code, but cheap solutions for logic and latch soft errors are not well understood yet. Approaches such as checkpointing and rollback may not be viable for embedded applications because they might violate real-time constraints.

Switching activities and power dissipation across the die result in power-supply voltage variation, which can introduce performance degradation and timing violations to the design. Techniques to mitigate power supply noise can be divided into three classes: using circuit techniques that generate less noise, improving circuit noise immunity, or suppressing noise without circuit modification. However, these techniques can incur extra penalties. For example, adding on-die decoupling capacitors can help suppress power supply variation, at the expense of extra silicon area and leakage power penalties.

On-chip temperature variation can be as large as dozens of degrees. Because of cost sensitivity, embedded systems usually cannot afford expensive package and cooling mechanisms to help heat dissipation. Temperature can have a dramatic impact on circuit performance and power. For example, interconnect delay increases approximately 5 percent and MOS-current drive capability decreases approximately 4 percent for every 10°C temperature increase, which can cause transient faults due to timing violations. Leakage power increases exponentially with a temperature increase, which in turn causes further temperature increases and might incur the well-known thermal runaway problem.

To reduce the hotspot temperature and balance the thermal profile, embedded system developers are turning to temperature-aware design techniques. For example, in multiprocessor system-on-chip embedded systems, a developer can allocate and schedule tasks alternatively on different processing elements, while still meeting the real-time constraint, so that the computation activities are evenly distributed across the chip to achieve a balanced temperature profile. Runtime thermal-management techniques such as activity migration and dynamic voltage scaling and dynamic frequency scaling (DVS/DFS) also help mitigate thermal challenges. Developers must

keep the real-time constraint in mind, however, because techniques such as DVS/DFS can affect performance and cause real-time violation.

ACCELERATED AGING EFFECT

Embedded system designs using new process technologies cause higher on-chip temperatures, which result from higher power densities. This significantly accelerates various failure mechanisms, including electromigration, stress migration, time-dependent dielectric breakdown, and thermal cycling, leading to an overall decrease in reliability. Accelerated aging creates a serious risk that devices will fail within an embedded system's warranty period, which poses the key challenge of finding countermeasures that effectively prolong a system's lifetime. Two common failure mechanisms highlight the design challenges this problem poses.

Electromigration is one of the main aging-related failure mechanisms caused by ion migration in the direction of the current flow. The ion migration creates voids upstream and mounds downstream, resulting in open lines and shorts to adjacent wires. Power supply rails are particularly susceptible to this failure mechanism, but signal wires can also be affected by migration. Techniques to counter the impact of electromigration can span different design levels. These include introducing new interconnect material, such as silicon nanowires, which have much higher current-carrying capacity, and coding techniques to reduce switching activity and consequently the signal-wire current. Techniques that reduce the temperature of interconnects also reduce the electromigration rate.

As the supply voltage scaling in new process generations reaches a plateau, due to the need for keeping the difference between supply voltage and threshold voltage large, electric fields are increasing. Along with higher temperatures, this mechanism can result in electrons (or holes) tunneling into the gate oxide and being trapped there. This phenomenon, known as the *hot carrier effect*, can lead to a drift in a device's threshold voltage as it ages, leading to timing failures. As this hot carrier effect becomes prominent, providing redundancy for periodically switching out spares becomes necessary. To avoid the cost of making an entire design redundant, selective redundancy can be provided for those subparts prone to failure. For example, an increase in threshold voltage in noncritical paths is unlikely to raise performance concerns.

As technology scales, designing a dependable embedded system atop a less reliable hardware platform poses great challenges for designers. Cost and energy sensitivity, as well as real-time constraints, make some fault-tolerant techniques unviable for embedded system design. Many techniques to improve reliability can incur performance, energy, or cost penalties. Further, some solutions targeted at a specific failure mechanism could negatively affect other mechanisms. For example, lowering operational voltage can help mitigate thermal problems but increases vulnerability to soft errors. Developers must understand the tradeoffs when designing reliable embedded systems.

Vijaykrishnan Narayanan *is an associate professor in the Computer Science Engineering Department at The Pennsylvania State University. Contact him at vijay@cse.psu.edu.*

Yuan Xie *is an assistant professor in the Computer Science Engineering Department at The Pennsylvania State University. Contact him at yuanxie@cse.psu.edu.*

Editor: Wayne Wolf, Dept. of Electrical Engineering, Princeton University, Princeton, NJ; wolf@princeton.edu