Design for Test and Reliability in Ultimate CMOS

Michael Nicolaidis
Lorena Anghel
Nacer-Eddine Zergainoh
TIMA (CNRS, Grenoble INP, UJF)

Yervant Zorian
Synopsys

Tanay Karnik, Keith Bowman, James Tschanz, Shih-Lien Lu, Carlos Tokunaga, Arijit Raychowdhury, Muhammad Khellah, Jaydeep Kulkarni, Vivek De
Intel® labs
Intel® Corporation
Hillsboro, OR, U.S.A.

Dimiter Avresky
IRIANC

Abstract— **This session brings together specialists from the DfT, DfY and DfR domains that will address key problems together with their solutions for the 14nm node and beyond, dealing with extremely complex chips affected by high defect levels, unpredictable and heterogeneous timing behavior, circuit degradation over time, including extreme situations related with the ultimate CMOS nodes, where all processor nodes, routers and links of single-chip massively parallel tera-device processors could comprise timing faults (such as delay faults or clock skews); a large percentage of these parts are affected by catastrophic failures; all parts experience significant performance degradations over time; and new catastrophic failures occur at low MTBF.**

Keywords: DfT, DfY, DfR, ultimate CMOS, single-chip massively parallel teradevice processors.

I. Introduction

Michael Nicolaidis and Lorena Anghel

Technology scaling is going on and tools, materials and device innovations should enable its continuation for one decade more. 20nm technology is around, 14nm is expected for 2014, and 11nm in 2016. Using source Mask Optimization based lithography, PDSOI and bulk silicon, and subsequently FINFET and ETSOI is in the roadmap down to the 11 nm node. Things are less precise on the subsequent nodes, but silicon nanowires and fully depleted SOI are listed as the most pertinent choices for 8nm and 5nm, while 3nm is expected to move to some kind of carbon devices (nanotubes or otherwise). This evolution leads to unprecedented densities enabling integrating more than 10^{12} devices in a die. This will enable building extremely complex chips (including single-chip massively parallel computers comprising thousands of processors) and achieving unprecedented computing power. While process innovations are in the foundations of this evolution, power dissipation, quality, yield and reliability issues may become the showstoppers. In particular, rapidly increasing defect densities and fault model complexities; extreme static and dynamic PVT variability; cross talk; accelerated aging and device degradation; and increasing soft error rates; bring us in a domain where traditional DfT, DfY, and DfR approaches could not work.

This session brings together specialists from the DfT, DfY and DfR domains that will address the key problems together with their solutions for the 14nm node and beyond, dealing with extremely complex chips affected by high defect levels, unpredictable and heterogeneous timing behavior, circuit degradation over time, including extreme situations related with the ultimate CMOS nodes, where all processor nodes, routers and links of a single-chip comprise numerous timing faults (such as delay faults or clock skews); a large percentage of these parts will be affected by catastrophic failures; all parts experience performance degradations over time; and new catastrophic failures occur at low MTBF.

II. DFT and DFY for 14nm and Beyond

Yervant Zorian

With every new generation of semiconductor fabrication process, further miniaturization is realized and more complex designs enabled. While this is meant to increase functionality and performance of the target electronic product, but it makes the new chips more susceptible to defectivity, and result in manufacturing yield and reliability challenges. The upcoming 14 nanometer technology node with what it brings in terms of miniaturization and design complexity is a natural candidate for the above challenges. That is why understanding and resolving these challenges and finding new design for testability and yield optimization solutions in advance is important, without which we will observe considerable lengthening of production ramp-up period, and thus, time to volume (TTV).

We will look briefly into the two main causes of challenges in the upcoming 14nm technology node - design complexity and process miniaturization, and their impact on design for testability and design for yield. On the design complexity side, while the trend of escalating cost of chip design is continuing with every node, the cost of yield ramp up is simultaneously increasing with the same ratio. The investment in adequate DFT and DFY infrastructure during the design stage is critical to accelerate yield ramp up. As the use of third party IP core moves ahead, we will see more usage for third party subsystems in the new SOCs. The test and yield optimization challenges would require that those third party subsystems come with their DFT and DFY infrastructure in advance and allow integration with the hierarchy of the complex SOC, such subsystems will be the

978-3-9810801-8-6/DATE12/©2012 EDAA

volume drivers for manufacturing test, diagnosis and repair. But not every application will require the same type of scaling and integration. We will see numerous approaches to continue scaling beyond the traditional Moore's approach. The multi die integration, such as 2.5D or 3D is definitely one of the viable options to enable miniaturization and high bandwidth. This approach brings with additional DFT and DFY challenges, including the modeling defects resulting from silicon thinning effect and miniature interconnects such as through silicon vias. While exploring through the different approaches to reach scalability, we will have a set of planning tools to allow trade-off analysis for early exploration of design space. While based on cost modeling, this type of exploration tools will take into account the test and yield optimization costs, and their impact on power dissipation, test time, and area investment. The design complexity of 14 nanometer era necessitates early exploration prior to investing in expensive design efforts and yield ramp.

The second main cause of challenges in the upcoming 14nm technology node is the process miniaturization, and its impact on design for testability and design for yield. The change in form factor and power constraints of mobile products (smart phone and tablets) is the major reason for adopting this advanced node. On the one hand, the use of new device technologies, such as FinFET, will bring certain advantages such as excellent channel control and lowering inter die variability, but will also bring new types of defects due its 3D nature. The fault modeling effort for the new devices is definitely one of the key challenges in process miniaturization. Several phenomena will increase the number of faults to monitor and repair. In new applications related to touch and gesture, we will see far more use of analog oriented subsystems, which will come with their own set of faults to detect and diagnose. In memories, the trapping and de-trapping of individual carriers due to active trips in the gate oxide will cause random telegraph noise induced failures; the weakness in cells will cause static and dynamic weak cell faults; the defects on data and address buses and paths will cause new data and address setup and hold issues; Also, the transistor parameter variations will cause a range of read, write, access and hold issues. All of these require augmented test algorithms and repair infrastructure. Committing to a third party IP or subsystem will necessitate that it is silicon proven and fully characterized before it being integrated in a design. In addition, growing concerns about intermittent errors, unstable storage cells, and the effects of aging will also influence further the DFT and DFY infrastructure invested in our new generation of scaled SOCs.

III. DESIGNING RELIABLE PROCESSOR CORES

Tanay Karnik, Keith Bowman, James Tschanz, Shih-Lien Lu, Carlos Tokunaga, Arijit Raychowdhury, Muhammad Khellah, Jaydeep Kulkarni, Vivek De

Static variations in the microprocessor parts are addressed by per-part binning of supply voltages (V_{CC}) and frequencies (F_{CLK}) of operation. On-die caches are typically operated on a separate supply, other than the core supply, to satisfy the per-part minimum read/write/retention voltage (V_{CCMIN}) requirements. Dynamic variations in supply voltage, temperature, and transistor aging adversely affect the operating V_{CC}-F_{CLK} point. A common method to overcome these dynamic variations is to add a safety guard band in voltage or frequency domain, which causes a severe penalty in power-performance trade-off.

Dynamic voltage-frequency scaling (DVFS) is commonly employed to adapt the V_{CC}-F_{CLK} point to deliver a desired performance level with low power. Intel®'s Core™ microprocessor integrates a dedicated microcontroller for power management with custom ISA and hardware optimized for power management arithmetic [1]. The power management algorithms are implemented via firmware. The microcontroller monitors the per-die operating environment, resolves multiple requests from on-die sensors, and arrives at a unified power-management state. If the total power consumed is less than safe limits then it raises V_{CC}/F_{CLK} to convert the power headroom into extra performance.

Itanium™ processor extended the DVFS approach to adapt the V_{CC}-F_{CLK} point in response to dynamic variations [2]. The approach was further enhanced by adding adaptive body bias (V_{BB}) control in a research processor designed for TCP/IP input processing [3]. Both processors included on-die sensors coupled with adaptive DVFS control circuits. The sensors consisted of thermal sensors inserted at local hot spots and V_{CC} droop detectors to measure global V_{CC} fluctuations which affect all circuit paths. The adaptive circuits can adjust F_{CLK}, V_{CC}, or V_{BB} in response to dynamic variations. Aging sensors can track critical path degradation from worst-case DC stress to capture the effects of aging during power up/down sequences and sleep modes [4]. Since the sensor-based design requires time to detect and respond to dynamic variations, these techniques reduce the F_{CLK}/V_{CC} guard bands for slow-changing global variations, resulting in higher average V_{CC} than deeply embedded resiliency techniques.

In comparison to the previous sensor-based adaptive techniques, the guard bands are further reduced by combining tunable replica circuits (TRC) for timing-error detection with error recovery [4-5]. The TRCs are inserted adjacent to each critical pipeline stage. They toggle each clock cycle and drive an error-detection sequential (EDS) element [6-9] to detect late timing transitions from dynamic variations. The TRCs require post-silicon tuning to accurately track the critical pipeline delays. The TRC and the local pipeline stage use the same V_{CC} and clock, enabling the TRC to detect V_{CC} droops at finer granularity and to capture the clock-to-data relationship per pipeline stage. When dynamic variations induce a late timing transition in the TRC, the EDS generates an error signal, which invalidates erroneous data. The appropriate instruction is replayed to ensure correct operation. Assuming infrequent timing errors from dynamic variations, the TRCs provide a superior power-performance trade-off as compared to sensor-based adaptive cores.

Embedding EDS circuits [5-9] in all critical paths eliminates the V_{CC}/F_{CLK} guard bands for fast and slow as well as global and local dynamic variations. When dynamic variations induce an actual timing error, the error is detected

and corrected to maintain proper system functionality. In addition to removing the guard bands for dynamic variations, further F_{CLK} benefits are possible by exploiting path-activation probabilities. Silicon measurements demonstrate that resilient circuits enable a 41% throughput gain at equal energy or a 22% energy reduction at equal throughput, as compared to a conventional design when executing a benchmark program with a 10% V_{CC} droop [5]. The EDS design detects critical-path timing failures for fast and slow as well as long-range and local dynamic variations. In contrast, the TRC design cannot detect path-specific or highly-localized dynamic variations. Although transistor aging degradation affects the individual transistors in a path depending on the gate voltage and temperature conditions, a separate DC-stressed TRC with a periodically-toggled input can track the worst-case delay of aging for critical paths and clocks. The TRC requires a delay guard band to ensure the TRC delay is always slower than critical-path delays, thus preventing the possibility of exploiting path-activation rates for higher performance as with embedded EDS circuits. Furthermore, the TRC design may initiate an error recovery when an actual error did not occur, resulting in unnecessary recovery cycles. However, in comparison to the EDS design, the TRC design significantly reduces the design complexity overhead and provides a larger error-detection window. Both designs require post-silicon calibration, which affects testing costs.

An integrated all-digital dynamic variation monitor (DVM) was also designed to continuously measure the impact of dynamic parameter variations on circuit-level performance to enhance silicon debug and adaptive clock control [10]. The DVM consists of a TRC, a time-to-digital converter, and multiplexers to measure circuit delay or frequency changes with less than a 1% measured resolution error while capturing clock-to-data correlations. Silicon measurements demonstrated the DVM capability of tracking the worst case F_{MAX} reduction to within 1% for a wide range of V_{CC} droop profiles. The DVM interfaces with an adaptive clock control circuit to dynamically adjust the clock frequency by changing the divide ratio in the phase-locked loop in response to persistent variations. This enables the microprocessor to adapt to the operating environment for maximum energy efficiency.

Finally we will address the variability problem in on-die caches. We mentioned a separate voltage regulator for the shared memory system as a static variation tolerance mechanism above [1]. We present the design of tunable replica bits (TRBs) to alleviate a significant portion of the 8T SRAM guard bands [11], similar to the use of TRCs described above. Measurements on a 16 KB SRAM array fabricated on a 45 nm CMOS process demonstrate possible microprocessor power savings as high as 26% when TRBs are employed, with insignificant loss of performance. TRBs can also be employed with DVFS to mitigate slow-changing dynamic variations such as temperature or device aging.

REFERENCES

[1] R. Kumar, and Glenn Hinton., "A family of 45nm IA processors," in IEEE ISSCC Dig. Tech. Papers, Feb. 2009, pp. pp.58-59.

[2] R. McGowen, et al., "Power and Temperature Control on a 90-nm Itanium Family Processor," IEEE J. Solid-State Circuits, Jan. 2006, pp. 229-237.

[3] J. Tschanz, et al., "Adaptive Frequency and Biasing Techniques for Tolerance to Dynamic Temperature-Voltage Variations and Aging," in IEEE ISSCC Dig. Tech. Papers, Feb. 2007, pp. 292-293.

[4] J. Tschanz, et al., "Tunable Replica Circuits and Adaptive Voltage-Frequency Techniques for Dynamic Voltage, Temperature, and Aging Variation Tolerance," in IEEE Symp. VLSI Circuits, June 2009.

[5] K. Bowman, et al., "A 45nm resilient microprocessor core for dynamic variation tolerance," IEEE J. Solid-State Circuits, Jan. 2011, pp. 194-208.

[6] P. Franco and E. J. McCluskey, "On-Line Testing of Digital Circuits," in Proc. IEEE VLSI Test Symp, Apr. 1994, pp. 167-173.

[7] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies," in Proc. IEEE VLSI Test Symp., Apr. 1999, pp. 86-94.

[8] K. A. Bowman, et al., "Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance," IEEE JSSC, pp. 49-63, Jan. 2009.

[9] S. Das, et al., "Razor II: In Situ Error Detection and Correction for PVT and SER Tolerance," IEEE J. Solid-State Circuits, Jan. 2009, pp. 32-48.

[10] K. Bowman, et al., "All-digital circuit-level dynamic variation monitor for silicon debug and adaptive clock control," IEEE TCAS-I, Nov. 2011, pp. 2017-2025.

[11] A. Raychowdhury, et al., "Tunable replica bits for dynamic variation tolerance in 8T SRAM arrays," IEEE J. Solid-State Circuits, Apr. 2011, pp. 797-805.

IV. DESIGNING SINGLE-CHIP MASSIVELY PARALLEL PROCESSORS AFFECTED BY EXTREME FAILURE RATES

Michael Nicolaidis, Lorena Anghel, Nacer-Eddine Zergainoh, Dimiter Avresky

Ultimate-CMOS and post-CMOS technologies promise integrating trillions devices in a single die, leading to single-chip massively parallel architectures comprising thousands interconnected processors, and enabling the next computation turn. But the aggressive technology scaling that paves the way to the ultimate CMOS nodes has dramatic impact to: process, voltage and temperature (PVT) variations; sensitivity to electromagnetic interferences (EMI), to atmospheric radiation (neutrons and protons) and to alpha particles; and circuit aging. It also imposes stringent power dissipation constraints. The resulting high defect levels, heterogeneous behavior of identical circuits, accelerated circuit degradation over time, and extreme complexity, affect adversely fabrication yield and also prevent fabricating reliable chips in ultimate CMOS and post-CMOS technologies. These issues are becoming the main show-stoppers in the path leading to these technologies.

In the following we address these issues in the context of massively parallel tera-device processors, by considering that:

- After fabrication, all processing and routing nodes may be affected by some temporary faults such as delay faults, and clock skews.
- Fabrication faults altering persistently the circuit behavior may affect one or more regular blocks (RAMs, FIFOs, buses) in a large fraction of nodes. Such faults may also frequently occur during product life.
- Fabrication faults altering persistently the circuit behavior may affect irregular blocks (which are difficult to repair) in

a significant portion of nodes. Such faults may also frequently occur during circuit life (e.g. every few days)..
- New timing faults induced by circuit aging, as well as soft errors (SEUs and transients) may frequently occur during circuit life (and thus during application execution).
- Circuit degradation is continuous and requires continuous self-regulation of circuit parameters (clock-frequency, voltage levels, body bias), to maintain operational each processor node.

Clearly, no existing solution can cope with such massively defective systems, which invalidate even massive redundancy schemes (e.g. duplication, TMR), as all replicated parts may be defective. Such schemes also induce high area and power penalties. An interesting proposal [1] [2], targeting the design of reliable single-chip massively parallel processors avoids massive redundancy by using self-tests (hardware implemented or software implemented [3] [4]) to detect failures and create routing tables that are used subsequently to avoid failed processing nodes or failed routes. However, this approach could not cope with the issues affecting ultimate CMOS and post CMOS technologies as:

- In highly defective technologies, the vast majority of nodes (processing elements and routers) may include one or another kind of faults (e.g. timing faults produced by process, voltage and temperature variations, EMI, or aging). Thus, declaring as defective nodes affected by any kind of faults will quickly waste the computational resources of the chip.
-Achieving high fault coverage for timing faults is very difficult. Thus, many of these faults may escape fabrication test and also periodic self-tests and produce run-time errors.
- Faults occurring during application execution can not be covered by self-tests.

A more advanced energy-aware approach for allocating the application tasks to the available, reachable, and fault-free cores of embedded NoC platforms is proposed in [24].

In this paper we present a comprehensive approach developed at the TIMA laboratory, enabling using in efficient and reliable manner all parts able to perform useful computations. Hence, we call the chips having such capabilities *Terminator* - ***Te**ra-device bio-**mi**metic **na**no**t**echn**o**logy **r**obust – Chips* [5] in reference to the android, which, after being smashed, mobilized his last functioning parts to pursue his mission.

Our framework comprises several techniques spanning at all levels of the system. Innovations are introduced at all levels of this framework, including its overall architecture, its particular components, and the way the cooperation of these components is architected to optimize the outcome. Our work concerning several of these components is already presented in various publications. In the rest of the paper we shortly present the overall strategy and architecture of our framework, which was used since several years to define and guide the development of its particular components, but was not yet published.

A. Overall Strategy and Framework Architecture

Fault Partitioning: For optimizing the fault and error detection mechanisms in terms of area, power and for improving the fault/error mitigation efficiency, we treat differently faults affecting the function of the circuit only temporally (referred hereafter as *temporary faults*), from faults modifying persistently the logic function of the circuit (referred hereafter as *persistent faults*). This partitioning avoids also wasting hardware resources. Indeed, as the vast majority of faults expected to affect ultimate CMOS are temporary faults - including timing faults (such as delay faults and clock skews induced by spot defects increasing circuit delay, PVT variability, EMI, circuit aging), and soft errors (such as SEUs and SETs) - rejecting as faulty the nodes that include such faults, will quickly exhaust hardware resources. Thus, nodes affected by temporary faults will be recycled by modifying their clock frequency/Vdd level/body bias.

Error/Fault Detection: Faults persistently modifying the circuit function (like stuck-at faults) are easy to test because they do not require special test conditions for maximizing their duration. Thus, we detect them by means of off-line self-tests. Some temporary faults (SEUs and SETs) are impossible to test by means of self-test. Other temporary faults (i.e. timing faults), are very complex to test and we can not guaranty high fault coverage. Thus, these faults are treated by a concurrent error detection scheme. Conventional concurrent error detection require high area and power costs, but more recent solutions using double-sampling [6-9], enable detecting temporary faults and reduce drastically these costs. Thus, we adopted this approach. Though our framework is also compatible with double sampling in FF-based designs, we preferably implement it by using the GRAAL architecture. While this architecture uses latch-based design style, which is less common than FF-based design, it has significant advantages such as low area and even lower power costs [11]. In addition, it does not require short path constraints (as the other double-sampling schemes) and also detects temporary faults (delay, clock, and transient faults) of large duration (e.g. delay faults increasing by 100% circuit delays) [10]. The later is very important as temporary faults are becoming the major issue in nanometric processes. Also detecting large temporary faults offers large timing margin. Thus, we can cope with practical temporary faults and at the same time reduce power by aggressively reducing Vdd under the level required for correct timing behaviour. As explained later, this power reduction is efficiently managed at the array level by our task allocation algorithm, based-on the concurrent error detection rates observed at each node.

Error/Fault Recovery: The concurrent error detection scheme allows detecting errors induced by temporary faults affecting processing nodes, routers or links as soon as they occur. Thus, instruction replay (a mature technique already used in some commercial microprocessors like the IBM Power 6 processor [12]) or message retransmission for faults affecting links and routers, will be used to recover from temporary faults at low area, performance and power penalties. They are performed at lower clock frequency [6] to achieve correction of errors induced by timing faults.

Persistent faults are detected by the self-tests. But instruction replay or message retransmission does not work for

them. The same is also the case for memory faults detected but not corrected by ECC. Rollback recovery is used in these cases, introducing much longer recovery time. Fortunately, MTBF of persistent faults is large as aging degrades circuit slowly and will infrequently lead to persistent faults (e.g. node MTBF larger than several days could comfortably considered even in ultimate CMOS). Thus, performance penalty induced by rollback recovery will be insignificant.

A next issue is related to the fact that as the state of the system is huge; we have to perform *check-pointing for partial states* of the system (e.g. states of tasks), while *preserving its overall coherence*. To cope with this issue a check-pointing algorithm for parallel processors maintaining system coherence was developed [13].

Another issue is that *errors produced by persistent faults will go undetected by the double-sampling scheme and will create system failures*. To cope with this issue we have to ensure that rollback will recover all these errors. We guaranty this by a tight cooperation between self-tests and rollback recovery. That is, the system state used for performing rollback recovery is maintained until the next self-test session (e.g. if rollback recovery is based on check-pointing, each check-pointing session is preceded by a self-test).

Yet another issue is that *errors produced by persistent faults will be reproduced during rollback* even if we reduce clock frequency. To cope with this issue we use a cooperation between the rollback recovery and the fault tolerant task allocation algorithm described later, to ensure that no tasks is reallocated to resources declared as faulty by the self-test (this will happen during the recovery session but also afterwards).

A further issue is that the state used for performing rollback recovery (e.g. the states saved at each check-pointing) must be safely stored in reliable, and thus off-chip, storage media. But single-chip massively-parallel processor grids will have only a few IO ports. Thought check-pointing usually represents a small percentage of the application execution time, *in a system comprising thousands of processing nodes and few IO ports, check-pointing the state of the nodes to off-chip storage media will result in permanent congestion of these ports*. Thus, as an alternative to check-pointing-based recovery we developed a hierarchical task distribution approach, which guaranties reliably storing within the unreliable hardware of the chip the root states of this distribution and allows rollback recovery without using check-pointing [14].

Distinguishing persistent from temporary faults. One more issue is that while self-tests are not efficient against timing faults, they will detect many of them. *As timing faults are much more frequent than persistent faults, the detectable ones will induce very frequent rollback recovery sessions, which will affect system performance*. To cope with this issue, the self-tests are partitioned into small sessions. Then, when a fault is detected by such a session, the session is repeated at reduced clock frequency to determine whether the root fault is temporary or persistent. Then, only in the later case we declare the node as faulty and activate a rollback. *This technique also avoids declaring the circuit definitely faulty due to detectable timing faults, which otherwise could waste hardware resources*.

Self-repair. As regular blocks, (e.g. RAMs, FIFOs) can be repaired at low cost [15] [16] [17] [18], then, if a fault is detected in such a block, the block is repaired and is considered fault-free by the fault-tolerant scheduling and allocation algorithm. This is also the case for persistent faults affecting interconnection links [19][20]. Thus, we significantly reduce the number of processors, routers and links that are declared unusable due to such faults.

Array-level Self-adapting Approaches: To guaranty reliable uninterrupted operation, while maximizing computing power reducing energy dissipation, and coping with circuit degradation, the above techniques are completed by array-level approaches, which guaranty fault-tolerant, congestion-free, and deadlock free routing, as well as variability-aware, power-aware and application-deadline aware fault-tolerant task scheduling and allocation and circuit parameters regulation. Due to the high complexities of both hardware (thousands of nodes) and applications; circuit heterogeneity; high fraction of unusable nodes, routers and links (i.e. affected by un-repairable persistent faults); frequent occurrence of new persistent faults; and high error rates induced by temporary faults; deterministic algorithms achieving global optimization for these tasks are computationally intractable. Thus, our array-level algorithms do not search optimal, global and centralized solutions. Instead, *distributed, "non-deterministic", and local approaches* are used, by opportunistically selecting free routers or nodes at local level and/or by making loose global decisions and leaving the final choices at the local level.

Adaptive fault tolerant routing: In complex grids comprising thousands of nodes, routing algorithms based on routing tables are congestion prone and have low adaptability to new failures. Hence, we developed distributed algorithms using local opportunistic decisions (get another path when a node/router/link is faulty or congested) [21][22]. Our simulation results show that they easily scale for grids comprising thousands nodes; tolerate multiple faulty nodes/routers/links; avoid congestions; and cope with new failures occurring at any time. But, as they are based on local decisions, they are deadlocks prone. To cope with we use virtual networks and pertinent rules acting at the local level.

Adaptive fault tolerant task scheduling and allocation: At a first step we integrate in a single algorithm our FT routing approach (discussed above) with distributed FT scheduling and allocation. We also integrate in the same algorithm check-point-free rollback recovery [23]. This is done by an approach which: uses an hierarchical task organization into parent-children trees; maintains this hierarchy during execution; and if a persistent fault occurs in a resource executing a child it goes back to the parent to redistribute the workload in fault-free resources. This way we avoid saving the internal states of the grid to external media (no check-pointing), which would congest the grid I/Os, and we can recover correct operation even after the occurrence of any multiple fault (by going up in

the tree until a fault-free parent, which redistributes the aborted workload to fault-free resources).

In further developments we extended the algorithm to enable variability aware and power aware task scheduling and allocation. Partial results in this domain are presented in [24]. Again we use a distributed non-deterministic approach to handle computation complexity. One of the developed schemes maps different groups of tasks into different regions of the grid according to the energy dissipated by each group and the power-dissipation characteristics of the regions. Then, the leader of each region maps task clusters into sub-regions, and the leader of each sub-region maps each task to a node, by using similar power dissipation considerations. The clock frequency required to meet the deadline of each task and the power/reliability priorities of the task are encapsulated in the header of each task. Each node knows its frequency/Vdd operating domains for various error occurrence rates (by monitoring its concurrent error detection signal). Thus, it can determine its clock frequency and Vdd level to reach the task deadline, minimize the energy dissipated, and achieve the target reliability level (in terms of error occurrence rate). Thus, circuit parameters (clock frequency and Vdd) are continuously regulated to minimize power, preserve reliability and adapt to circuit degradation induced by aging.

CONCLUSION

We presented a framework able to realize the Terminator chips. That is a framework able to: achieve reliable non-interrupted operation in single-chip massively parallel processors, comprising thousands of processing nodes, in which virtually all nodes/routers/links are affected by some temporary or persistent (repairable or not repairable) fault and new faults may occur with high MTBF (e.g. a node becomes non-functional every few days due to the occurrence of persistent non-repairable faults); and at the same time reduce power dissipation, increase performance, and continuously adapt circuit operating parameters to circuit degradation.

REFERENCES

[1] G. Zajac, P., Collet, J.H. & Napieralski, A., Self-Configuration and Reachability Metrics in Massively Defective Multiport Chips, in Proc. *14th IEEE International On-Line Testing Symposium*, July 2008.

[2] E. Kolonis, M. Nicolaidis, D. Gizopoulous, M. Psarakis, J.H. Collet, P. Zajac "Enhanced Self-Configurability and Yield in Multicore Grids" 15th IEEE International On-Line Testing Symposium, IOLTS 2009, Sesimbra-Lisbon, Portugal, June 24-27, 2009,

[3] D.Gizopoulos, M.Psarakis, M.Hatzimihail, M.Maniatakos, A.Paschalis, A.Raghunathan, S.Ravi, “Systematic Software-Based Self-Test for Pipelined Processors”, *IEEE Trans. on VLSI Systems*, vol.16(11), pp 1441-1453, 2008.

[4] J. Zhou, H.-J. Wunderlich, “Software-Based Self-Test of Processors under Power Constraints”, *in Proc. IEEE Design, Automation and Test in Europe Conf.*, pp. 1–6, 2006

[5] M. Nicolaidis, Designing Robust Single-Chip Massively Parallel Tera-Device Processors: Towards the Terminator Chip, Keynote –Plenary Session, IEEE VLSI Test Symposium, May 1-5, 2011, Dana Point, CA.

[6] M. Nicolaidis, “Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies,” in Proc. IEEE VLSI Test Symp., Apr. 1999, pp. 86-94.

[7] L., Anghel, M. Nicolaidis, Cost Reduction and Evaluation of a Temporary Faults Detecting Technique, Proceedings Design Automation and Test in Europe Conference, March 2000, Paris (*Best Paper Award*), also chapter in “The Most Influential Papers of 10 Years DATE”, Lauwereins, Rudy; Madsen, Jan (Eds.), Springer, ISBN: 978-1-4020-6487-6, 2008.

[8] S. Das, et al., “Razor II: In Situ Error Detection and Correction for PVT and SER Tolerance,” IEEE J. Solid-State Circuits, Jan. 2009, pp. 32-48.

[9] K. A. Bowman, et al., “Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance,” IEEE JSSC, pp. 49-63, Jan. 2009.

[10] M. Nicolaidis, Circuit-level Soft-Error Mitigation, In: M. Nicolaidis (Ed), Soft Errors in Modern Electronic Systems, Springer, 2011.

[11] H. Yu, M. Nicolaidis, L. Anghel, N.E. Zergainoh, Efficient Fault Dectection Architecture Design of Latch-based Low Power DSP/MCU Processor, In Proceedings, 16th IEEE European Test Symposium, May 23-27, 2011, Trondheim, Norway.

[12] M.J.Mack, W.M.Sauer, and B.G.Mealey, "IBM Power6 Reliability," IBM J. Research and Develop., vol. 51, no. 6, pp. 763-764, Nov. 2007.

[13] C. Rusu, C. Grecu, L. Anghel, “Improving the scalability of checkpoint recovery for networks-on-chip”, IEEE International Symposium on Circuits and Systems (ISCAS), 2008, Seattle, Washington, USA.

[14] G. Bizot, D. Avresky, F. Chaix, N.E. Zergainoh, M. Nicolaidis “Self-Recovering Parallel Applications in Multi-Core Systems”, 10th IEEE International Symposium on Network Computing and Applications (IEEE NCA11), 25 - 27 August 2011 Cambridge, MA USA

[15] Zorian Y., “Embedded Memory Test & Repair: Infrastructure IP for SOC Yield“, 2002 IEEE International Test Conference.

[16] Sawada K., Sakurai T., Uchino Y., Yamada K., “Built-In self repair circuit for High Density ASMIC”, IEEE 1999 Custom Integrated Circuits Conference.

[17] M. Nicolaidis, N. Achouri, L. Anghel, ”A Diversified Memory Built In Self Repair Approach for Nanotechnologies”, IEEE VLSI Test Symposium, April-May 2004, Napa Valley, CA. (/*Best Paper Award)*

[18] M. Nicolaidis, N. Achouri, S. Boutobza, Dynamic Data-bit Memory Built-In Self- Repair, Proc. 2003 IEEE/ACM International Conference on Computer-aided Design, November 9-13, 2003, San Jose, CA,

[19] I. Loi et al, A Low-overhead Fault Tolerance Scheme for TSV-based 3D Network on Chip Links, 2008

[20] V. Pasca, L. Anghel, M. Nicolaidis, M. Benabdenbi, CSL: Configurable Fault Tolerant Serial Links for Inter-die Communication in 3D Systems", to appear Journal of Electronic Testing: Theory and Applications, Springer.

[21] F. Chaix, D. Avresky, N. E. Zergainoh, M. Nicolaidis, Fault-Tolerant Deadlock-Free Adaptive Routing for Any Set of Link and Node Failures in Multi-Cores Systems, In Proc. 9th IEEE International Symposium on Network Computing and Applications (NCA10), July 15-17 2010, Cambridge, MA

[22] F. Chaix, D. Avresky, , N. E. Zergainoh, M. Nicolaidis, A Fault-Tolerant Deadlock-Free Adaptive Routing for On Chip Interconnects, In Proc. Design Automation and Test in Europe Conference, March 14 – 18, 2011, Grenoble, France (*Best IP Award*).

[23] G. Bizot, D. Avresky, F. Chaix, N.E. Zergainoh, M. Nicolaidis, “Self-Recovering Parallel Applications in Multi-Core Systems”, 10th IEEE International Symposium on Network Computing and Applications (IEEE NCA11), 25 - 27 August 2011 Cambridge, MA USA

[24] F.Chaix, G.Bizot, M.Nicolaidis, N.Zergainoh, “Variability-aware Task Mapping Strategies for Many-core Processor Chips”, IEEE International On-line Testing Symposium, July 2011, Athens, Greece.

[25] C.L. Chou, U. Y. Ogras, R. Marculescu, Energy- and Performance-Aware Incremental Mapping for Networks on Chip With Multiple Voltage Levels, IEEE Tr. On Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, No 10, October 2008

[26] C.L. Chou, R. Marculescu, FARM: Fault-Aware Resource Management in NoCbased Multiprocessor Platforms, In Proc. Design Automation and Test in Europe Conference, March 14 – 18, 2011,Grenoble, France.