

On-Chip Monitoring: A Light-Weight Interconnection Network Approach

Pablo Ituero Marisa López-Vallejo Miguel Ángel Sánchez Marcos Carlos Gómez Osuna

Dpto. de Ingeniería Electrónica, ETSI Telecomunicación, Universidad Politécnica de Madrid
Avenida Complutense 30, Ciudad Universitaria, 28040 Madrid, Spain
{pituero,marisa,masanchez,cgomez}@die.upm.es

Abstract—

Current nanometer technologies are subjected to several adverse effects that seriously impact the yield and performance of integrated circuits. Such is the case of within-die parameters uncertainties, varying workload conditions, aging, temperature, etc. Monitoring, calibration and dynamic adaptation have appeared as promising solutions to these issues and many kinds of monitors have been presented recently. In this scenario, where systems with hundreds of monitors of different types have been proposed, the need for light-weight monitoring networks has become essential. In this work we present a light-weight network architecture based on digitization resource sharing of nodes that require a time-to-digital conversion. Our proposal employs a single wire interface, shared among all the nodes in the network, and quantizes the time domain to perform the access multiplexing and transmit the information. It supposes a 16% improvement in area and power consumption compared to traditional approaches.

Index Terms—Monitoring, On-Chip, CMOS, temperature, leakage, aging, dynamic adaptation.

I. INTRODUCTION

The levels of integration of current CMOS technologies are getting closer and closer to their physically feasible limits. In 2012 14 nm processes are expected to be manufactured, however, this scaling trend, that has been relentless for several decades, is expiring. Technologists are finding much trouble in dealing with the complex processes required by these extremely small device sizes. The resulting integrated circuits must face two key problems:

- Process uncertainties, that result in very low yield and reliability figures.
- Harmful effects such as high power densities, hotspots and degradation and wastage with use and time.

The traditional way of fighting against these harming effects was defining guard-bands which hid the possible variations using a corner-based design approach (typically implying that 3σ of all manufactured circuits would not exceed corner values). This safe and inefficient way of designing has ceased to be appropriate, since the area-performance-power budget is shrinking fast, and the process-induced variation and time-dependent shifts in transistor parameters are increasing rapidly [1]. Furthermore, the increase in variations within the same die (WID variations) makes existing corner-based methodologies inadequate.

This work was funded by the CICYT project DELTA TEC2009-08589 of the Spanish Ministry of Science and Innovation.

Due to these reasons, designers of current integrated circuits must be aware of all those negative effects and provide the necessary mechanisms to overcome them. Information from in situ monitors is much more trustable than indirect measures [2]. Multiuse sensors that can monitor performance, degradation, and power consumption as the circuit ages have become critical [3]. However, allocating an arbitrarily large number of such monitors will not only create a significant area overhead, but routing the data from the sensor registers to a central processing unit will also pose a challenge [4].

Our work targets this section of the new adaptive techniques based on real-time monitoring that has not been fully covered by the scientific literature yet. Specifically, we present a new low level monitoring network architecture based on the concept of digitization resource sharing. The network supposes a light weight overhead in terms of area and power consumption. Furthermore it displays the following features:

- Multi-purpose. The network can be used both for calibration and monitoring.
- Simplicity and scalability. The structure of the network makes it compatible with any existing hierarchy structure.

The structure of the paper is the following. First, other related works are reviewed. After that, the network technical requirements are analyzed. Section IV describes the network structure and the proposed topology and digitization resource sharing technique. Section V provides details on the hardware implementation that has been carried out while section VI describes in detail the characterization of the proposed signaling scheme. Finally, experimental results are presented and some conclusions are drawn.

II. PREVIOUS WORKS

Concerning previous works, since the research community has paid attention to the thermal-aware design for a longer time, it is this field that has gathered most of the previous proposals. The first works in the literature that mentioned the use of on-chip temperature sensors for DTM appeared in the mid nineties and the solution they proposed varied much from one another. Usually, they contained a single embedded temperature sensor that sometimes was capable of asserting an interruption when a thermal emergency was produced—such was the case of IBM's PowerPC—, or included a peripheral serial interface, such as I^2C , SPI or SMBus. The

most common response at that time was to reduce the power consumption via clock throttling.

Later, in the early 2000s, when temperature began to become an important design constraint, more temperature monitors were allocated on-chip and the first standard interfaces appeared. For example, Intel's 90nm Montecito [5] included four temperature sensors with four ADC point-to-point connected to a microcontroller that was compliant with the Advanced Configuration and Power Interface (ACPI). Sensor data was used to control thermal and power policies such as DVFS.

The number of monitors and the need for better thermal profiling has since become more and more important, in fact, it is not unusual to find chips with tens of monitors. Such is the case of the AMD quad-core Opteron, in which tens of thermal sensors are distributed across the chip. Each of the four cores contains a thermal evaluation circuit that point-to-point connects to the sensors in that core. However, there is not much information available about more effective ways to collect the data from the monitors.

Nowadays, even when there have been many contributions in the field of the Networks-on-Chip (NoC), the scientific literature has paid so far little attention to the topology of on-chip networks specifically designed for monitoring. Among the several papers dealing with control policies that require a certain number of sensors, such as DTM or DFVS systems, very few put forward any kind of network interface, and those that do it use a simple point-to-point connection between the sensors and the central control.

The first innovative approach that we found in the literature is that by Székely et al [6]. This pioneer work established all the basis of thermal-aware electronic design from thermal simulation to thermal monitoring. They proposed to insert the thermal test circuitry into the boundary-scan architecture and compare all the temperatures to a maximum rating. This idea of connecting all the monitors through a one-wire chain emulating a global shift-register imposes a lower bound in the routing of the network and has been employed frequently in the literature. Recent standards used in state-of-the-art processors, such as the Platform Environment Control Interface (PECI) by Intel, also make use of single-wire interfaces with the monitors.

A recent work in thermal monitoring [7] has proposed a starred network topology that connects each of the sensing nodes to a central node. The transmission is performed in parallel and to diminish the elevated number of interconnection lines that are required, the measurements undergo a compression stage —specifically they execute a reduction from eight to four bits. Still the architecture supposes a big amount of connections and furthermore there is a significant loss in precision that could not be acceptable for the DTM policies. This implies that any comparison with this work of a scheme that does not employ any compression would lead to unfair erroneous results.

The first work completely dedicated to the topic is [8], in which Madduri et al. exposed the problem and proposed a network architecture that targets multicore processors and

supports priority-based data transfer and customized interfacing. They include information of the network latency under several configurations to deliver all monitor data to the central controller. The approach is applied to a NoC.

An interesting analysis on interconnection architectures for hierarchical monitoring was presented in [9]. Based on mesh-based NoC platform experiments, the authors conclude that physically separated networks provide flexible and energy-efficient transmission for monitoring communication, with guaranteed latency. In particular, hierarchical monitoring networks are the most appropriate solution on the chosen platform. In this work we propose a low level interconnection network that can provide the lowest monitoring levels in such a hierarchical infrastructure.

In despite of all these advances, the problem of automatizing the connection of an elevated number of monitors to control not only the thermal behavior but also any other magnitudes interesting for dynamic management continues to be unsolved. Next we first identify the features that these networks must fulfill and then we introduce a network scheme that conveniently covers the lower section of the hierarchy.

III. MONITORING NETWORK OVERVIEW

Next we analyze the characteristics that a network of on-chip monitors must fulfill. Specifically we target the on-chip network that connects a set of monitors —or nodes— not necessarily evenly distributed, to a control system that processes their information.

- 1) **Multi Purpose.** Given that the network is the only way to access the different types of monitors, it has to accomplish two main functions: monitoring and calibration. The network must have the delay and latency characteristics imposed by the control policies, and also must provide with unique and differentiated access to each of the sensors to realize the calibration.
- 2) **Ultra Light Weight.** The network must suppose a small overhead in terms of area, reliability, latency, power and self-heating of the system. A solution that fulfills the system requirements, will necessarily go through a trade-off between area —especially the routing of each sensor to a processing unit—, power —mainly dependent on the data frequency and the switching activity of the interconnection lines— and both the amount and delay of the information that eventually gets to the policy controller.
- 3) **Flexibility.** The flexibility to host as many different types of monitors as possible is another key characteristic that will allow the network to be implemented in a variety of systems. A consequence of this need for adaptability is the requirement for standard interfaces that not only go towards the monitor-ends but also cover the network-OS and the network-PCB interfaces.
- 4) **Priority.** A network involving different types of monitors deals with data at several levels of importance. For instance, a quick action must be taken at general-emergencies, such as a big voltage supply droop or the

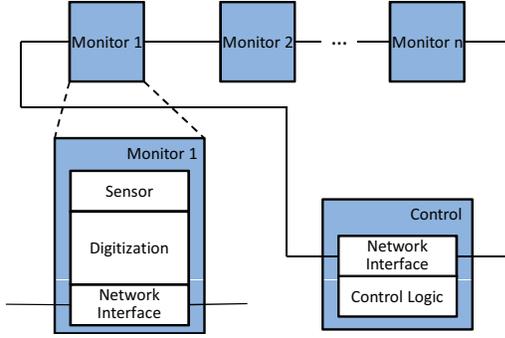


Fig. 1. Network architecture for the boundary-scan style.

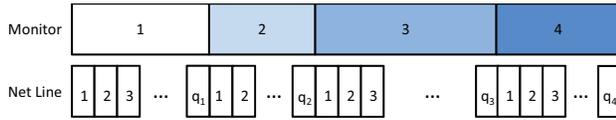


Fig. 2. Functioning of a basic boundary-scan like network.

surpassing of the safe limit in the junction temperature and this must be compatible with timely deliver of all the fixed rate information.

- 5) **Hierarchy.** A flat hierarchy network with all the monitors accessing the controller, would add a lot of complexity to the monitors and set hurdles to establish priorities. Therefore an network with at least two hierarchy levels is desirable.
- 6) **Dynamic Adaptation.** Yet another feature is the dynamic sensor-selection to avoid collecting data from those sensors that will not provide useful information, as proposed in [10]. Ideally, the network should prevent the sensor from working whenever its information is not used.

IV. PROPOSALS

In this work we target the lowest level of the network hierarchy, which is the one that has to deal with the monitors. So far, the most optimized solution for this kind of networks, concerning area and complexity, found in the scientific literature and also adopted by the industry is a boundary-scan like network similar to the one depicted in figure 1. Let us now consider a very simple implementation of this boundary-scan like network in which at each round all the bits from all the monitors are transmitted sequentially over the same connection, as depicted in figure 2. In this case, the maximum number of monitors, n , connected to the same line is limited by

$$n < \frac{f_{clock}}{f_s q} \quad (1)$$

where f_{clock} is the clock frequency of the system, f_s is the sampling frequency, and q the number of bits in each measurement.

A. Digitization Resource Sharing

For each monitor of the network there is an interface that turns an analog signal into a digital one, normally by means of an ADC or a time-to-digital converter. Let us, thus, divide each monitor into a sensing block and a digitization block. Interestingly, the part that normally occupies more area and consumes more power is the digitization block [11].

Our proposal in this area is to make several sensors share the digitization resources. In some cases, the nature of the analog signal will prevent this solution —e.g. an ADC that takes a voltage as input could have serious sensitivity issues if it is placed far from the sensing block. However, with certain types of monitors, this solution is completely feasible and actually saves area and power consumption.

Specifically, we focus on sensors whose analog varying signal is a pulse width or a ring-oscillator frequency, i.e. the digitization part is a time-to-digital converter. This kind of signals is very easy to deliver from different points of the chip to a certain spot where the digitization is performed. There is a certain dependency of the delay of the transmission lines on some of the factors that are to be measured, such as the temperature, the aging, etc. However, this variability is small enough to not affect the sensibility of the conversion, and it will be covered by our noise budget.

Every critical-path monitor has an implicit time-to-digital conversion [12] and many of the aging monitors that have been proposed are based upon the variations of the delay of a certain path. A whole generation of temperature sensors based upon time-to-digital converters have appeared in the last few years imposing a new paradigm because of their reduced power consumption and area. The common characteristics of this kind of sensors are a sensing part that produces a pulse with a varying duration as a function of the temperature and a digitization part that normally includes a counter that measures and quantizes the pulse duration. For example, [13] employs a delay line with several gates to generate a temperature-dependent pulse. Also remarkable, the sensors in [11] make use of the leakage current thermal dependencies to produce the pulse and are characterized by a very small power consumption.

Note, that although some works provide a signal with a varying frequency at the output of the sensor —such as a ring oscillator— this signal can easily be converted into a varying pulse by means of a counter, fed by this signal, that counts up to a fixed number. Furthermore, any digital signal can be converted into a PWM signal by means of a counter.

Our proposal targets a network where all the monitors in a subnetwork employ the same quantizing frequency for their time-to-digital conversion. This is not a strong limitation since normally all the sensors are of the same kind, have the same layout, and thus suffer from the same sensibility issues. With this restriction, a single count can perform the conversion for all the monitors. More precisely, all the monitors connected to a counter start their pulse at the same time and the count starts at that moment; whenever a pulse from a sensor finishes, the

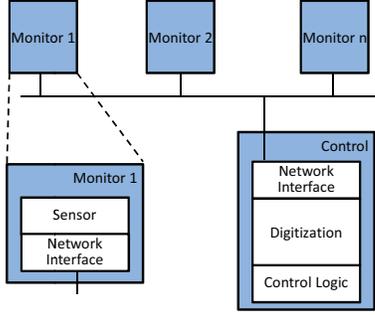


Fig. 3. Network architecture for the digitization resource sharing.

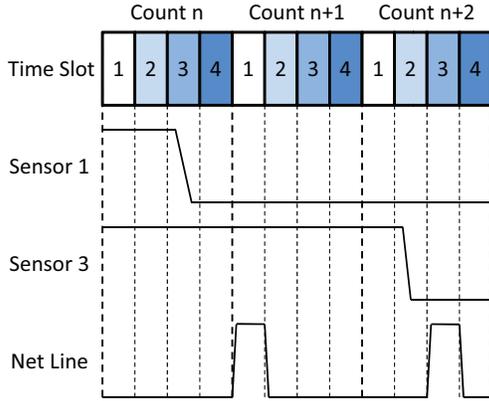


Fig. 4. Functioning example of the second proposal of digitization resource sharing.

current count is registered and associated with that particular sensor. In this way, an important power saving is produced compared with other possible implementations in which n counts were executed.

This scheme, shown in figure 3, supposes that the digitization part —i.e. the counter— is shared by the monitors and is placed at the controller. The monitors are connected to the control block through a shared line. The scheme requires that all the monitors are able to communicate their pulse-end at any counting cycle, and more importantly that the control discriminates which one sent the signal. This is achieved by the division of each counting cycle into n slots, being n the number of monitors connected to a single counter. Each monitor is assigned a slot and whenever their sensing part asserts a pulse-end, the monitor-to-network interface sends a pulse to the network in their next slot. Note that this, at most, produces an error of a counting period which compares to the one produced by the standard counter-based time-to-digital conversion. This scheme is depicted in figure 4.

The maximum number of monitors that can be connected to a single monitor is no longer limited by the sampling frequency, in this case it is the clock frequency of the system, f_{clock} what restricts the minimum time slot size, and therefore the number of words that can be sent. Particularly, the number

of monitors employing the same counter is bounded by

$$n < \frac{\Delta T f_{clock}}{2^q} \quad (2)$$

where q is the number of bits of the monitor signal; and $\Delta T = \max\{T\} - \min\{T\}$ is the difference between the minimum and the maximum pulse produced by the sensing part.

V. DETAILED HARDWARE IMPLEMENTATION

In this section we provide low-level hardware details of the implementation of the proposed network scheme including the different modules that conform it.

A. Network Architecture

The network architecture is based on the following principles:

- The network consists on one central controller, or front-end, and several monitor nodes, or back-ends, which share one physical data channel.
- Data channel access is based on a time division mechanism, with fixed slots for each of the nodes to avoid data collision. Each node knows its slot on the multiplexing scheme and all nodes are synchronized to avoid collision, no negotiation or arbitration mechanism is present to reduce complexity to the network.
- Apart from the data channel, one clock and reset lines are available for both masters and slaves within the network.
- Each back-end is connected to a monitor.
- Each back-end is responsible for the generation of the request/acknowledge interface towards its sensor.
- The central controller uses slot 0 in the time division mechanism to send the read command to the rest of the nodes.
- The central controller converts the answer of each of the nodes into a digital time measurement.

Concerning the operation, all back-ends, and the front-end share the same line on which each node has been assigned a one cycle slot to send its data. The slot that each node has been assigned is hard-coded at implementation time, with slot 0 necessarily reserved for the front-end controller to send the read command to the rest of the network. Thus, the front-end of the network will send the same pulse to all sensors/back-ends at slot 0 and then wait for the data to arrive through the serial line. Each back-end node interfaces with its corresponding monitor through a two line request/acknowledge control interface plus a data line in which the sensor will drive either the PWM signal or the parallel digital signal. In the case of PWM monitors, each back-end synchronizes the detected end of the pulse with the next available time-slot. In the case of monitors with a parallel interface, the back-end transforms the information into a PWM signal by means of a counter. The end of the count is synchronized with the next available time-slot. For this kind of monitors, we suppose that the monitor asserts the data within the first time interval, so that the first interval is used for the front-end signaling, and the second corresponds to a 0 digital value.

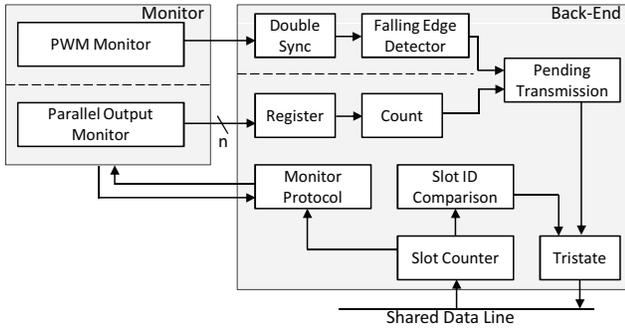


Fig. 5. Implementation of the back-end module.

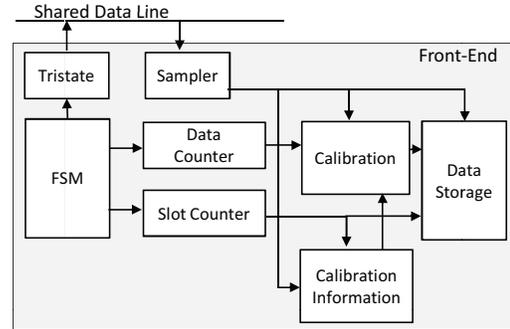


Fig. 6. Implementation of the front-end module.

A complete readout of the network would include the following steps:

- 1) The front-end node drives the shared data line high for 1 cycle during slot 0 to all sensors and back-end nodes to start the process.
- 2) Each back-end will drive the monitor request line and wait for the monitor to drive the ack line.
- 3) Each monitor produces its measurement. In the case of PWM monitors, the back-end nodes will wait for a falling edge on the PWM output. Once it is detected, the back-end will wait for its slot on the shared data line and send a 1 bit pulse to the front-end controller to signal the end of the readout. In the case of parallel interface monitors, once the data is received, it is converted into a PWM signal and asserted in the next available slot.
- 4) Whenever the front-end detects a pulse from one node, it registers the value, performs the calibration correction and stores it.

B. Network Back-End

This module realizes three main functions. First, it has to listen to the network to determine when the a monitoring round starts. Second, it is in charge of the communications with the sensor, which means it needs to inform the sensor when to start the measurement, and it has to receive the sensor outgoing data. And third, it has to synchronize the monitor data so it is asserted at the corresponding time slot. This module is depicted in figure 5.

When a pulse of one cycle is sensed at slot 0, the back-end node becomes active, sending the request signal to the sensor, waiting for its acknowledge. In the case of a PWM sensor, a falling-edge detector registers the end of the pulse from the sensor. In the case of a sensor that provides a parallel digital signal, the back-end stores the signal and activates the counter that converts this signal into a varying-width pulse.

After this, the back-end node waits for its turn in the serial line and sends a 1 cycle-long high pulse through it to indicate the end of the measurement. Slot control is achieved by an internal counter which is synchronized with all other nodes on reset.

C. Network Front-End

The front-end controller either produces a periodic start pulse at a established sampling frequency or it can also act in a on-demand mode in which it waits for an external signal activation to assert the start pulse. In either mode, the front-end waits for slot 0 and then sends the pulse over the shared data line. Subsequently, it waits until all sensors in the network have sent their end-of-measurement pulse. The front-end node keeps a count of the number of back-end nodes which have already sent their measurement. Once the front-end has received pulses from all the nodes, it stops the counting process that performs the digitization and goes back to its idle state to wait until the next start pulse is asserted.

Each time a pulse is received on the shared line, the outgoing digital value from the counter, that indicates the number of cycles from the beginning of the readout process, undergoes a linear transformation according to the stored calibration data. This corrected value along with the monitor identification, extracted from the time slot, are stored in a data storage unit.

In order to implement these functionalities, the block includes the following internal structures:

- An FSM to distinguish between the different operating modes.
- A slot counter in order to synchronize with all back-end nodes.
- A node counter to check the number of sensors from which data has already been received.
- A bank of counters, to generate the time to digital conversion measurements.
- The calibration logic that performs the transformation of the measurements.
- A data storage unit containing the current measurements.
- A memory containing the calibration information.

This module is depicted in figure 6.

VI. SIGNALING SCHEME CHARACTERIZATION

The reliability and the robustness of the architecture greatly depends on sets of single pulses traveling along a shared transmission line. Although this seems to be a very risky approach

if you think of the variability and noise issues inherent to new technologies, the truth is that the low bandwidth requirements of monitoring networks allow to reduce the working frequency down to a totally safe value while delivering all the necessary data on time.

In order to determine the working frequency that provides error-free functioning for a target line length we have performed extensive Monte Carlo simulations varying the parameters that affect the quality of the traveling pulse. For this study we have employed a 6-metal 90nm technology from UMC. We have covered process, V_{DD} and thermal (PVT) variations. We have altered the position of the pulse sender so that the differences in transmission time are accounted for. And, finally, we have considered several crosstalk scenarios. At the beginning, we also varied the type of signaling, but in the end we decided to not employ any special signaling scheme to keep a reduced complexity in our modules.

Concerning the PVT, we have taken the technology probability distribution parameters for the interconnection and transistors, for the temperature we have taken measurements in the industrial range ($-40^{\circ}C$ to $85^{\circ}C$), and for Vdd variations, we have assumed a $\pm 10\%$ uniform distribution.

The sources of crosstalk are completely determined by the layout of the circuit that the network monitors. We have assumed that the line is implemented in middle metallization layers, which are more likely to be able to accommodate some extra wiring, since lower levels are employed for the implementation of the standard cells and upper levels are normally reserved for system levels signals such as V_{DD} , ground and clocks. In particular we have carried out our experiments with a metal 3 layer (out of 6). On this layer, most capacitance is to other signal lines. In the worst scenario, which has been considered in the analysis, another data line would go in parallel so that the parasitic capacitance would be maximum. In a more realistic scenario, our transmission line could cross the bit lines of a datapath. The values on the bit lines are highly correlated and may all switch simultaneously in the same direction [14]. This case has also been accounted for.

A data line with several back-ends distributed at different locations has the extra problem of a variable transmission time from the back-ends to the listening front end. This effect also takes into account the action of the clock skew at the back-ends. The net clock is transmitted from the front-end so that all the back-ends receive a slightly delayed version of it. When a back-end transmits a pulse, it arrives at the front-end with a double delay since it was produced employing a skewed clock and then it took some time to travel through the data line. These problems were simulated by placing three back-ends, one at each end and another in the middle.

The sampling of the pulses in the front end is performed at the end of the clock period (rising edge), so that the frequency is actually bounded by the delay of the farthest back-end under the worst PVT and noise conditions. When the back-ends listen to the line expecting to synchronize with the pulse at slot 0, they take samples at the middle of the clock period (falling

TABLE I
MAXIMUM RELIABLE FREQUENCIES FOR DIFFERENT WIRE LENGTHS

Length	8mm	2mm	500 μ m
f_{max}	33MHz	111MHz	500MHz

edge) because they sense the signal with very little delay since they work with a delayed version of the clock. The results of the analysis for different interconnection line lengths are shown in table I.

VII. COMPARISON WITH PREVIOUS WORKS AND DISCUSSION

To demonstrate the advantages of our network architecture we have implemented a 32-monitor network of our proposal and another 32-monitor network of the traditional boundary-scan like approach. The implementations have the minimum circuitry that fulfills the protocols involved in each network. We synthesized the designs targeting a 90-nm standard cell library from UMC and the numeric results come from the synthesis simulation under typical conditions.

The details of the implementation are as follows. The sensing part of the monitors simulates a temperature sensor and provides a PWM signal that needs an 8-bit quantization. All the monitors need the same quantizing frequency and, in the case of the boundary-scan like network, an extra control line distributes this signal. This decision is questionable, since we could have implemented a module in each monitor that produced this frequency from the clock tree, however it would just increase their area and power consumption. The control module simply stores the data from the 32 monitors and selects the biggest, since we consider these as the indispensable functions for it. We fixed the working frequency at 10 MHz and used a sampling period of 1ms.

Figures 7 and 8 summarize the synthesis results of both architectures. As shown, we achieve an area improvement of 25% in each monitor due to the lack of digitization modules, in the complete network, we get an improvement of 16%. Concerning the power consumption, we achieve a significant reduction of 16% in the whole network. As the network adds more monitors, the improvement approaches a maximum of a 25% since the monitors represent a higher and higher portion of the network.

Note that apart from the energy saving due to the union of all digitization processes into a single one, there is also an important dynamic power reduction caused by the smaller number of loads and unloads in the shared net line. From the analysis of the boundary-scan-like implementation, we get that the number of transitions in each of the interconnection wires of an n-monitor network is upbounded by $\sum_{i=1}^n q_i$, being q_i the number of bits of the i th monitor. If we now turn to the case of our network architecture, the number of transitions in the interconnection line is upbounded by $2n$, twice the number of monitors because each monitor just transmits a pulse. The savings in dynamic power consumption on the interconnection line are significant; in fact, for very distributed networks with

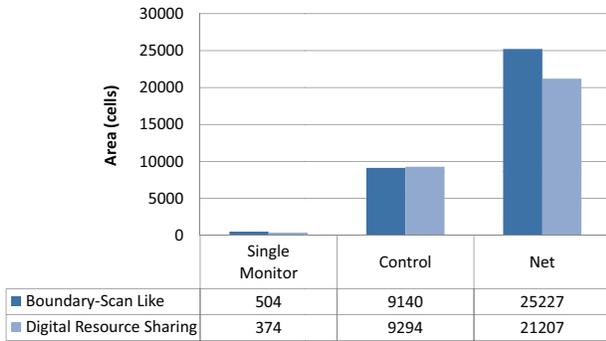


Fig. 7. Area results and comparison.

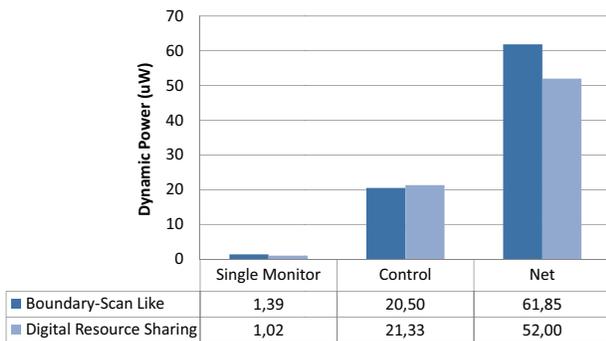


Fig. 8. Power results and comparison.

long wires, this signaling model could be extended to any kind of smart sensor, including those that do not realize a time-to-digital conversion. The idea is to turn any digital word into a width-varying pulse by means of a counter. The power overhead caused by the counter compares to that of the boundary-scan like protocol and is compensated by the energy savings in the transmission line.

Another important feature of our architecture is that the calibration process is performed employing a single digitization block, and that the write-back information of each sensor is kept at the control block rather than at each of the monitors. This simplifies the calibration stage and the linearization process since all the required logic is instantiated just once and it is easier to control.

VIII. CONCLUSIONS

In the late CMOS era, the challenges imposed by reliability, aging and thermal issues among others have highlighted the need for monitoring, calibration and dynamic adaptation. The certain possibility of systems with hundreds of monitors makes clear the need of a light weight monitoring network, able

to deliver all the information that monitors produce without supposing high area and power overheads.

We have described a network architecture based on the idea of digitization resource sharing that fulfills these requirements. The time-to-digital conversion of all the monitors is realized at the same control module and at the same time. When compared to the traditional boundary-scan like network, our architecture achieves a 16% saving in area and power consumption. Furthermore it reduces the activity on the network, it is easily scalable and simplifies the calibration process.

REFERENCES

- [1] M. Alam, "Reliability- and process-variation aware design of integrated circuits," *Microelectronics Reliability*, vol. 48, no. 8-9, pp. 1114 – 1122, 2008, 19th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis (ESREF 2008).
- [2] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Maestro: Orchestrating Lifetime Reliability in Chip Multiprocessors," in *High Performance Embedded Architectures and Compilers: 5th International Conference, Hipeac 2010, Pisa, Italy, January 25-27, 2010, Proceedings*. Springer-Verlag New York Inc, 2010, pp. 186–200.
- [3] D. Sylvester, D. Blaauw, and E. Karl, "Elastic: An adaptive self-healing architecture for unpredictable silicon," *Design Test of Computers, IEEE*, vol. 23, no. 6, pp. 484 –490, june 2006.
- [4] S. Memik, R. Mukherjee, M. Ni, and J. Long, "Optimizing Thermal Sensor Allocation for Microprocessors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 516–527, 2008.
- [5] R. McGowen, C. Poirier, C. Bostak, J. Ignowski, M. Millican, W. Parks, and S. Naffziger, "Power and temperature control on a 90-nm titanium family processor," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 1, pp. 229–237, 2006.
- [6] V. Szekeley, M. Rencz, and B. Courtois, "Tracing the thermal behavior of ics," *IEEE Design & Test of Computers*, vol. 15, no. 2, pp. 14–21, 1998.
- [7] K. Vaddina, E. Nigussie, P. Liljeberg, and J. Plosila, "Self-timed thermal sensing and monitoring of multicore systems," in *Proceedings of the 2009 12th International Symposium on Design and Diagnostics of Electronic Circuits&Systems-Volume 00*. IEEE Computer Society, 2009, pp. 246–251.
- [8] S. Madduri, R. Vadlamani, W. Bursleson, and R. Tessier, "A monitor interconnect and support subsystem for multicore processors," in *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09*. IEEE, 2009, pp. 761–766.
- [9] L. Guang, E. Nigussie, J. Isoaho, P. Rantala, and H. Tenhunen, "Interconnection alternatives for hierarchical monitoring communication in parallel socs," *Microprocessors and Microsystems*, vol. 34, no. 5, pp. 118–128, 2010, special issue on selected papers from NORCHIP 2008.
- [10] J. Long, S. Memik, G. Memik, and R. Mukherjee, "Thermal monitoring mechanisms for chip multiprocessors," *ACM Transactions on Architecture and Code Optimization*, vol. 5, no. 2, 2008.
- [11] P. Ituero, J. Ayala, and M. Lopez-Vallejo, "A Nanowatt Smart Temperature Sensor for Dynamic Thermal Management," *IEEE Sensors Journal*, vol. 8, no. 12, pp. 2036–2043, 2008.
- [12] A. Drake, *Adaptive Techniques for Dynamic Processor Optimization*. Springer, 2008, ch. 7 Sensors for Critical Path Monitoring, pp. 145–174.
- [13] P. Chen, C.-C. Chen, C.-C. Tsa, and W.-F. Lu, "A Time-to-Digital-Converter-Based CMOS Smart Temperature Sensor," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 8, pp. 1642–1648, August 2005.
- [14] W. Dally and J. Poulton, *Digital systems engineering*. Cambridge Univ Pr, 1998.