Power-gating for leakage control and beyond

Power-Gating for Leakage Control and Beyond

Andrea Calimera, Alberto Macii, Enrico Macii, and Massimo Poncino

Abstract The need of reliable nanometric integrated circuits is driving the EDA community to develop new automated design techniques in which *power consumption* and *variability* are central objectives of the optimization flow.
Although several *Design-for-Low-Power* and *Design-for-Variability* options are already available in modern EDA suites, the contrasting nature of the two metrics makes their integration extremely challenging. Most of the approaches used to compensate and/or mitigate circuit variability (e.g., Dynamic Voltage Scaling and Adaptive Body Biasing) are, in fact, intrinsically power inefficient, as they exploit the concept of redundancy, which is known to originate power overhead.
In this work, we introduce possible solutions for concurrent leakage minimization and variability compensation. More specifically, we propose *Power-Gating* as a mean for simultaneously controlling static power consumption and mitigating the effects induced by two of the most insidious sources of variability, namely, *Process Variations* (PV) due to uncertainties in the manufacturing and *Transistor Aging* due to Negative Bias Temperature Instability (NBTI).
We show that power-gating, when implemented through the insertion of dedicated switches (called *sleep transistors*), has a double effect: On one hand, when sleep transistors are enhanced with tunable features, it acts as a natural supply-voltage regulator, which implements a control knob for PV compensation; on the other hand, during the idle periods, it makes the circuits immune to NBTI-induced aging.
We describe optimization techniques for the integration of a new concept of power-gating into modern sub-45nm design flows, that is, *Variation-Aware Power-Gating*. The experimental results we have obtained are extremely promising, since they show 100% timing yield under the presence of PV and circuit lifetime extension of more than 5X in the presence of NBTI.

The authors are with Politecnico di Torino, Dip. di Automatica e Informatica, 10129 Torino, Italy

1 Introduction

MOS devices are approaching the size of atoms, which is a fundamental barrier for the scaling process of the bulk technology. The research community is thus actively pursuing alternative materials, fabrication processes, devices and architectures to be adopted in mainstream circuit and system manufacturing. While several new solutions have shown good potential (e.g., carbon nanotubes f and graphene, memristors, spin-based devices, ferromagnetic logic, atomic switches, NEMS), the debate of which of them will prevail is still open; therefore, nanometer CMOS will remain the dominant technology on the electronics market for a few years.

The 2009 ITRS Roadmap [17] reported that *static power consumption* and *variability* are the most serious concerns for the design of nanometer ICs below 45nm. Static power due to internal leakage mechanisms represents the main source of power consumption in modern CMOS circuits [28], which have shown to be power-hungry even when not switching. Variability, instead, refers to the marked tendency of a manufactured circuit to show a deviation from its nominal behavior. Main sources of variability include random and systematic process variations [4], environmental condition variations [33] (e.g., temperature and V_{dd} fluctuations) and aging effects (e.g., Negative Bias Temperature Instability and Hot Carrier Injection) [1]. While static power translates to low energy efficiency, variability originates lower reliability and lower fabrication yield; both factors make electronic circuits less reliable.

Static power and variability are not new in the EDA field, and various options for their management are already available. However, while in the past considering the two as independent variables in the design space was accurate enough to obtain reasonable results, advanced technology nodes are showing the need of holistic design strategies that could provide *concurrent* static power optimization and variability compensation. Unfortunately, this challenge is complicated by the fact that most of the design solutions for compensating variability are intrinsically power inefficient: Fault-tolerance approaches, such as “fail and correct”, or adaptive strategies, such as “dynamic voltage regulation” or “forward body biasing”, are based on the concept of redundancy, which is in contrast with low-power requirements.

In this work, we address this critical issue and we propose the use of power-gating [2] as the enabling technology for achieving simultaneous leakage optimization and variability compensation.

Power-gating (PG) is based on the insertion of a dedicated MOS switch, called *sleep transistor*, between the gated block and the actual ground rail. This provides the circuit with two power modes: A low-power mode, during which the sleep transistor is turned-off and the leakage power is reduced by more than one order of magnitude, and an active mode, during which the sleep transistor, which is turned-on, guarantees a normal connection to the global power-rails. What is interesting to note is that, in terms of variability, the sleep transistor shows important and unique properties, that is: During the active periods, when enhanced with tunable-width features, it can act as a natural supply-voltage regulator usable to implement adaptive control schemes for process-variation mitigation [14]; during the low-power mode, its effect is to make the gated circuit immune by the aging mechanisms (NBTI and HCI) [9].

Based on these observations, we claim that PG can represent a viable solution for the implementation of low-power, aging-free reliable ICs. Obviously, the beneficial effects that PG can provide must be weighted against the amount of overhead it introduces. A direct Sleep Transistor Insertion (STI) methodology, in fact, may originate excessive timing, area and power overhead, which can off-set the beneficial effects that PG offers in terms of aging and variability compensation.
To overcome this drawback, we propose new optimization techniques based on the concept of *Variation-Aware Clustered PG*, which consists of a methodology for clustering and power-gating critical cells only, that is, to apply variation-aware PG only to cells whose process variation-induced and/or aging-induced variations have a direct impact on the overall performance of the circuit. To enable such a strategy, we opted for an STI flow in which sleep transistors are inserted in layouts with row granularity. This allows a finer control of variability/aging compensation, while reducing the design overhead. Experimental results performed on a set of benchmark circuits and mapped to an industrial 45nm CMOS library prove the effectiveness of the proposed solutions, as well as their integrability to industrial design flows.
The remainder of the chapter is organized as follows. In Section 2, we discuss the main challenges to be faced during nanometric IC design; we introduce models for sub-threshold leakage power consumption, also describing the sources of variability in scaled nanometric technologies. Section 3 addresses the key design issues related to power-gating, with particular emphasis on automated solutions for physical sleep-transistor insertion. In the last two sections, we provide detailed background and models for process variation and NBTI effects and we present automated power-gating strategies for process variation compensation (Section 4) and NBTI mitigation (Section 5). Finally, Section 6 gives some concluding remarks.

2 Design Issues for Nanometric CMOS Circuits

2.1 Sub-Threshold Leakage Power Consumption

Among all the leakage current mechanisms induced by Short Channel Effects (SCEs), the *sub-threshold current* I_{sub-th} has proven to be the major contributor to the total static power consumption [28].
I_{sub-th} is defined as the drain-to-source current which flows when the transistor operates in the weak inversion region, i.e., when the gate voltage V_g is below the threshold voltage V_{th}. Under this condition, the channel shows a small, but non-zero concentration of minority carriers that are diffused from the drain to the source terminal whenever a potential greater than 0 is applied between drain and source, i.e., $V_{ds} > 0$.
A well-known model for the I_{sub-th} of a single nMOS transistor is given by the following equation [28]:

$$I_{sub-th} = \mu C_{ox} \frac{W}{L}(m-1)v_T^2 \cdot e^{\frac{V_g - V_{th}}{mv_T}} \cdot (1 - e^{\frac{-V_{ds}}{v_T}}) \quad (1)$$

with

$$m = 1 + \frac{C_{dm}}{C_{OX}} \quad (2)$$

where $v_T = KT/q$ is the thermal voltage, C_{OX} is the gate oxide capacitance; μ is the carrier mobility; m is the sub-threshold swing coefficient, with C_{dm} representing the capacitance of the depletion layer.
The magnitude of the sub-threshold current is a function of several parameters, such as the operating temperature (I_{sub-th} increases as T increases), the supply voltage (I_{sub-th} increases for larger V_g), and the device size (I_{sub-th} increases as the transistor gets larger and shorter). However, the parameter that affects most (i.e., exponentially) I_{sub-th} is the threshold voltage V_{th}: Decreasing the V_{th} by 100mV increases the leakage current by a factor of 10. That is why scaled CMOS technologies (characterized by ever smaller V_{th}) suffer of sensible sub-threshold leakage.

2.2 Sources of Variability

With variability we commonly refer to the marked tendency of a manufactured CMOS circuit to show a deviation from its nominal behavior. The sources of variation can be broadly classified according to their nature (statistical vs. deterministic), their spatial reach (local vs. global), and their temporal rate of change (static vs. dynamic). Figure 1 summarizes the typical variations arising in nano-scale CMOS circuits and systems.

		Local	Global
Statistical	Static	**Intra-die (WID) random process variations**	**Inter-die (D2D) random process variations**
	Dynamic	-	-
Deterministic	Static	**Systematic process variations**	**Lifetime degradation (NBTI, HCI, TDDB), Systematic process variations**
	Dynamic	**IR drop, clock jitter, coupling noise (capacitive and inductive)**	**Temperature and Vdd fluctuations**

Fig. 1 Types of Variability in CMOS-based Circuits and Systems.

Under the label *statistical* it is possible to include all those variations that are induced by stochastic events; they differ from *deterministic* variations, which can be somehow predicted at design time. *Global* variations affect all the transistors on the die, while *local* variations are limited to a few transistors in the immediate vicinity of each other. Finally, the classification between *static* and *dynamic* depends on the actual rate of change with time. Static variations, e.g., process variations, remain effectively invariant over the entire lifetime of the manufactured chips, while dynamic variations change over the lifetime of the chips. The changes can manifest on a large time-sale (that is the case of slow-variations like aging effects: NBTI, HCI and TDDB) or in a short time-scale (fast-variations like IR-drop, clock jitter, coupling noise, temperature and V_{dd} variations). Although all these sources of variability have deleterious effects on the reliability of CMOS digital circuits, two of them have been recognized as particularly critical, thus worth specific consideration: Process variations and aging.

Process variations (PV) [5, 4] are mostly due to random fluctuations of dopant atoms, which result in the mismatching of the electrical characteristics of transistors in the same die, and to systematic or non-systematic impreciseness of the manufacturing process (like lithography, etching, and chemical-mechanical polishing). Process variations have a significant impact both on the power dissipation and performance of a design: 20X variation in leakage power for a 1.5X variation in delay between fast and slow dies has been reported in the literature. Given the power/performance tradeoff, those figures translate into an increase of dies that must be discarded because either too slow or too power consuming. Therefore, as a relevant side effect, increased variability decreases yield, with important cost implications.

Aging, instead, includes all those wear-out mechanisms that induce *time-dependent* degradation of the operating characteristics of devices [1]. Two are the main sources of aging in active devices: Bias Temperature Instability (BTI), and Hot Carrier Interface (HCI) [26]. Both these physical/chemical effects result in the generation of interface traps at the silicon/oxide interface and cause a drift of the threshold voltage over time. These irreversible effects, and BTI in particular, have traditionally been regarded as ”reliability issues”, and have only recently received some consideration in the CAD community as a factor affecting performance of digital circuits. Traditional VLSI design, in fact, bypasses the analysis and optimization of such dynamic networks by approximating the problem to the optimization of uniform static networks with certain guard band. This may induce unacceptable design overheads.

3 Power-Gating for Leakage Power Reduction

3.1 Power-Gating Basics

Power-gating has proven to be a very effective approach to reduce standby leakage, while keeping high speed in the active mode. It is based on the principle of adding devices, called *sleep transistors*, in series with the pull-up and/or the pull-down of logic gates, and turning them off when the circuit is idle, thereby decreasing the leakage power component due to I_{DS} sub-threshold currents. When a nMOS sleep transistor is used on the pull-down path, a SLEEP signal controls its active/standby mode (i.e., SLEEP=1 during standby and SLEEP=0 during active mode). In the standby mode, the sleep transistor is off, thus disconnecting its insertion point, called *virtual ground*, from the physical ground. In active mode the gated circuit operates normally, but it incurs a delay degradation due to the series resistance of the sleep transistor. Figure 2 shows a logic block with a nMOS sleep transistor connected.

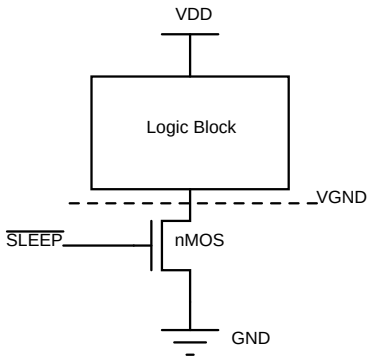

Fig. 2 A Logic Block with nMOS Sleep Transistor.

The source terminals of the logic gates in the logic block are connected to the virtual ground which is, in turn, connected to the drain terminal of the sleep transistor.
Effective use of power-gating requires a proper sizing of the sleep transistor, since that affects the performance of all the gates connected to it. While a small transistor unacceptably slows down the circuit in active mode due to its high resistance, a large one implies a significant overhead in area and a non-negligible energy (i.e., power and delay) for ON↔OFF transitions. One additional difficulty is that sleep transistor sizing is determined by the *maximum* current injected by the circuit, which leads to maximum drop across the sleep transistor drain-source path and, as a consequence, causes worst-case delay degradation during active operation.
Several power-gating styles have been proposed, differing in the *granularity* of the blocks to which sleep transistors are applied. Granularity may range from individual cells (the *fine-grained* sleep transistor insertion approach [21]) to large chip sub-units, (the *block-level power-gating* scheme), in which very large sleep transistors

are placed on the root of the power distribution networks of large chip areas. While the fine-grained approach suffers from high area overhead and an excessive buffering of the sleep signal due to very high capacitance load to be driven by this signal, block-level power-gating has the disadvantage of having long transition delays between sleep and active state, caused by the large RC time constant of the sub-unit's power distribution network.

Moreover, and this is the most important aspect, its coarse granularity reduces the degrees of freedom available to the designer. If the decision of power-gating a block is taken, then all gates in the block are gated. This may not be desirable if, for instance, there is a critical subset of gates for which parametric variations (typically due to fabrication process or aging mechanisms) induce speed degradation of the whole circuit. In this case, a finer control of the sleep transistor insertion could guarantee a perfect match between leakage savings and design overhead, and, as we demonstrate afterwards, it also allows to exploit the beneficial effects of power-gating in terms of variability only where necessary, i.e., only for those subsets of gates that are more critical.

We refer to this finer strategy as the *clustered sleep transistor insertion*, a very effective solution where multiple subsets of cells (i.e., the clusters) are connected to dedicated sleep transistors distributed across the layout. It is worth mentioning that clustered power-gating is not new in the low-power design domain, and many clustering solutions have been proposed. In [20], the authors propose a solution in which gates having mutually exclusive current discharge patterns are grouped together; the resulting clusters allow optimal sleep transistor sizes. Instead, in [31], the authors show an effective timing-driven clustering strategy that is able to handle simultaneously timing and area constraints. Differently from previous works, in Sections 4 and 5 we propose variation-aware clustering methods in which gates are grouped based on variation-induced timing criticality. Independently of the clustering algorithm, physical design details as well as constraints posed by the adoption of industrial EDA frameworks need to be considered while developing a strategy for sleep transistor insertion.

Using layout rows as atomic clustering objects greatly simplifies the physical-level management of virtual ground distribution, which is the major practical obstacle in clustering approaches that work on a cell-by-cell basis. In fact, having a mix of power gated and normal cells on the same row imposes drastic changes in power routing within a single row. The ensuing disruption of routing regularity makes it very difficult to control congestion and to ensure fast design convergence. Furthermore, since all the sleep transistors are placed in dedicated rows the sleep transistor placement is simplified and their overhead can be easily estimated and, consequently, traded for leakage reductions.

In order to fully integrate clustering and sleep transistor insertion with a state-of-the art physical design flow, placement and routing information have to be taken into account within the core clustering algorithm that selects gates to be power-gated, as well as in sleep transistor sizing and insertion.

Section 3.2 describes a common sleep transistor insertion flow suitable for the variation-aware clustered strategies proposed in this work.

3.2 Clustered Row-Based Sleep Transistor Insertion Methodology

State-of-the-art power gating methodologies follow a well known design flow, depicted in Figure 3.

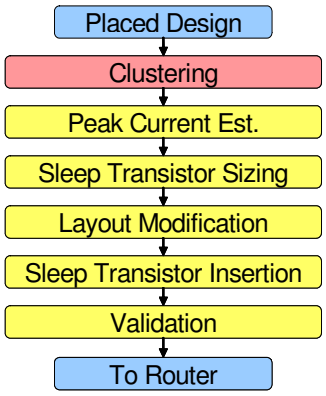

Fig. 3 Power-Gating Design Flow.

The entry point is a standard-cell placed design. The first step is *clustering*, in which cells are grouped together in order to be controlled by the same sleep transistor. Next, maximum current estimation is performed for each cluster. This information is essential to drive the selection of the appropriate sleep cells to be connected to the various clusters (*sleep transistor sizing*). The layout is then modified in order to accommodate the sleep transistor cells and the routing of the sleep signals. Finally, the modified layout is validated before it is fed to the routing tool.
This flow is fully compliant with industry standard back-end tools and it supports various power gating strategies (i.e., different sleep transistor insertion types), differing in the *granularity* of the blocks to which sleep transistors are applied, as described in Section 3.2.3.

3.2.1 Peak Current Estimation

This section describes a methodology for estimating the maximum current drawn by a cell cluster [30]. We define as Switching Window, T, of a gate under an input pattern the interval between the arrival time and the output transition of the gate. There is one switching window for each path through a particular gate. The width of the switching window of a gate is equivalent to the propagation delay of the gate for a rising or falling transition. For a given gate in the cluster, its Maximum Switching Window (MSW) is the time interval encompassing all its possible switching windows. If a gate has n switching windows, the set $T_1, \ldots, T_n$ is called the Full Switching Window (FSW) of the gate.

The peak current estimation algorithm consists of two steps. The first one computes the MSW for each gate in the cluster. The motivation behind extracting the MSW and not all the switching times of a gate is that, for a complex circuit and for a gate very deep in the logic, there is possibly a very large number of switching time intervals, each one corresponding to a path being activated through a gate. Moreover, it is very time consuming to extract all these windows for all the gates, which form a cluster. Conversely, it is very fast to extract the MSW for each gate. In fact, it only requires the calculation of the first (earliest) switching window T_1 and of the last (latest) one T_n; the MSW is simply obtained as $(T_1 \bigcup T_n)$. We thus need to extract only two switching time intervals for each gate in the cluster, which can be accomplished very fast.

The second step consists of the construction of a current plot over time which records the gates that switch at a particular time interval and, hence, the total discharge current in that time interval. This plot is the superposition over time of rectangles whose bases correspond to the MSWs of the individual gates of the cluster (which will, in general, partially overlap), and whose heights correspond to the currents drawn by the gates (e.g., derived by a technology library). From this current plot, the time interval during which the maximum current discharge occurs and the gates that contribute to this current discharge are obtained.

Figure 4 shows an example of current plot, in which five MSWs (from *a* to *e*) are shown. The interval in which the current drawn is maximum corresponds to the overlapping of the *c*, *d*, and *e* MSWs.

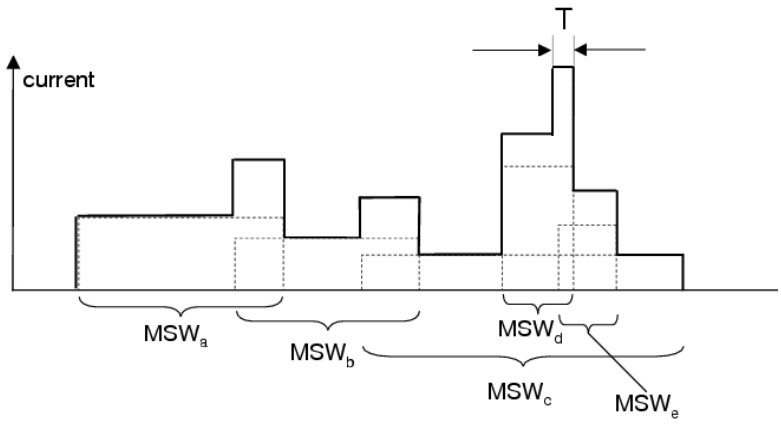

Fig. 4 Example of Current Plot.

The current plot allows the identification of the time interval during which the maximum current discharge could occur. To tighten this upper bound, we need to extract the detailed FSW information only for those gates which contribute to this maximum value, which are normally a very small percentage of the gates in the cluster.

The algorithm enters an iterative phase; it proceeds by finding, based on the current plot, the subset of gates that contribute to this maximum current value. Gate ordering on this subset of gates is executed first, so as to maximize the probability of non-overlapping switching of these gates. After ordering the gates, gate-by-gate FSW extraction is performed on the set of gates which contribute to the maximum current. Once all the possible switching time intervals for this small subset of gates are extracted, the current plot is updated and the process is repeated until convergence is reached.
Convergence is guaranteed since, in the worst case, we have to extract the FSW for all the gates in the cluster and compute their maximum currents. When full FSW extraction is computationally too expensive, we may terminate the iteration early, for example by using a time bound. Since the maximum current estimate is monotonically non-increasing as the iteration proceeds (i.e., the upper bound is progressively tightened), the use of a time bound provides us with a fine-grained control of the accuracy vs. time trade-off.

3.2.2 Sleep Transistor Sizing

An effective use of power-gating requires a proper sizing of the sleep transistor. In fact, while a small sleep transistor may unacceptably slow down the circuit in the active mode due to its high resistance, a larger one implies a large area and a significant energy cost to drive it [15]. Designers usually define an IR-drop threshold (e.g., 10% of V_{DD}) that is used as a constraint that must be met when sizing the sleep transistor resistance.
The maximum sleep transistor channel resistance can be computed using Equation 3:

$$R_{st} = \frac{V_{DD} \cdot \alpha_{drop}}{I_{on}} \tag{3}$$

where $V_{DD} \cdot \alpha_{drop}$ is the allowed voltage drop across the sleep transistor, expressed as a percentage of the supply voltage, while I_{on} is the maximum discharge current that power-gated cells inject into the sleep transistor during the active mode. The estimation of the active current is not a trivial task. In fact, an erroneous estimation of I_{on} translates to a sub-optimal sleep transistor sizing, which may result in area and power increase or, even worse, in timing violations during the active mode [30]. Considering that the sleep transistor operates in the resistive region and knowing R_{st}, its size can be properly evaluated using Equation 4:

$$\left(\frac{W}{L}\right)_{st} = \frac{1}{R_{st} \cdot \mu_{st} C_{OX} (V_{DD} - V_{th_{st}} - V_{DD} \cdot \alpha_{drop})} \tag{4}$$

where W/L is the ratio between width and length of the transistor channel, μ_{st} is the carrier mobility, C_{OX} is the oxide capacitance and $V_{th_{st}}$ is the threshold voltage.
The size of the sleep transistor is strongly influenced by its physical implementation, where carrier mobility, threshold voltage and gate length are key parameters. Hence,

for a proper sleep transistor sizing, indicating the type of MOS device is mandatory: pMOS (i.e., *header*) or nMOS (i.e., *footer*). Although both devices can be used as power-switches without distinction (i.e., maintaining the same leakage reduction), nMOS transistors are usually more suitable. In fact, pMOS devices are less leaky than nMOS ones, but they show a lower carrier mobility, which limits their ON-current. As a result, in order to guarantee a certain current capability, pMOS sleep transistors require more silicon area compared to nMOS.

In a typical power-gating approach, switch devices are high-threshold-voltage transistors (i.e., HVT). Since a HVT transistor is less leaky, this choice helps in reducing the total static consumption when the circuit is in stand-by mode. Unfortunately, a larger threshold voltage reduces the current capability of the transistor and, as demonstrated by Equation 4, more area is required to achieve the optimal resistance. On the contrary, with a low-threshold-voltage (i.e., LVT), the silicon area taken by the switch device can be sensibly reduced, but its leakage becomes significant.

Another design parameter that should be taken into account is the gate length. Usually, the sleep transistors are designed with minimum channel length, but for today's nanometric technologies, this implies high leakage currents due to Short Channel Effects (i.e., SCEs) and more power consumption. Increasing the gate length allows to reduce the power consumption [15], but the current sinking capability of the sleep transistor is drastically reduced and more area is required. Moreover, since the majority of the fab processes are optimized for a single channel length (typically, the minimum length), using devices with multiple lengths may drastically increase process variation and sensibly reduce fabrication yield.

As discussed above, the sleep transistor has to be sized in order to limit the IR-drop on the virtual rail when the maximum discharge current of the power-gated cells is injected into the virtual ground. Under this IR-drop constraint, designers can trade off channel length L and threshold voltage V_{th} (see Equation 4) to achieve area and power optimization. In principle, maximum area efficiency (i.e., minimum W) is obtained using transistors with the largest current density capability, namely, low-V_{th} transistors with minimum gate length. Obviously, this causes larger leakage consumption due to an increase of the sub-threshold current. On the other hand, in order to achieve minimum power overhead, high-V_{th} devices with larger gate length are the best choice. This helps in reducing leakage power consumption at the cost of a larger area. It is worth emphasizing that by increasing the equivalent sleep transistor area, also the load capacitance that the driving circuit has to charge and discharge increases, thus negatively impacting the dynamic power. Clearly, depending on the system constraints, designers can play with the values of W and V_{th} to achieve the required power/performance constraints. Most recent works propose optimum sleep transistor synthesis in which, for a given IR-drop and area constraints, the threshold voltage is selected to minimize the power [29]. To achieve the optimum V_{th}, both body-bias and multi-V_{th} transistors can be exploited.

3.2.3 Power-Gating Strategies

As anticipated in Section 3.1, when power gating is applied to a generic logic block, an important dimension of the problem concerns whether the gating is applied to: (i) The entire logic block (block-level or full power-gating); (ii) a subset of the cells, typically, the non-timing critical ones (partial or clustered power-gating); (iii) all the cells individually (cell-level power-gating). The block-level approach has the drawback of having an high reactivation period since it may have long transition delay between sleep and active state. The cell-level strategy is characterized by high area overhead and huge sleep signals buffering. The most promising power-gating style is then the clustered one, in particular the row-based style, which offers a reduced overhead with respect to the cell-level approach and limited active/sleep transition times with respect to the block level strategy.

3.2.4 Layout Modification

The modifications to the layout required to accommodate sleep transistor insertion depend on the chosen clustering granularity. Figure 5 shows an example of a modified layout in the case of row-based clustering; the sleep transistor is placed in a dedicated row indicated as ST row in the figure. Open channels indicate where the ground lines of adjacent rows are split to accommodate for clustering rows which share a common ground line.

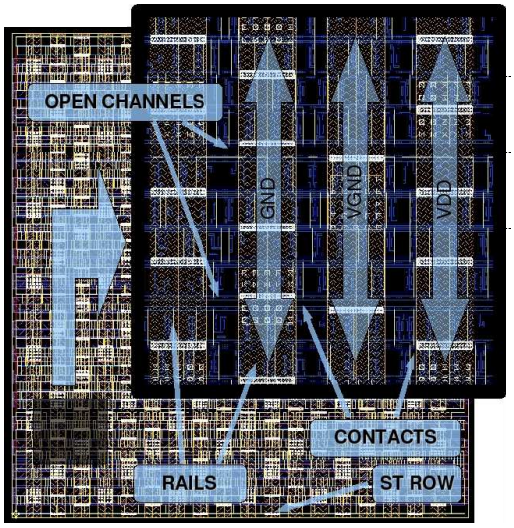

Fig. 5 Example of a Complete Layout after a Row-Based Sleep Transistor Insertion.

In the case of block-level power-gating, the layout must be modified so that the rows are extended on both sides of the layout to make space for the sleep transistor cells. All side pins must be moved and an extra pin must be added to connect the SLEEP signal with the outside. For row-based power-gating, the clustering phase returns a set of rows to be disconnected from the ground and connected to Virtual Ground. If those rows share the ground line with any non-power-gated row they need to be spaced. If two rows sharing the ground line are both to be clustered, we can simply connect the ground line to the Virtual Ground. Besides opening channels, extra rows needed to host the sleep transistor cells are added. As in the block-level case the pins have to be moved and the SLEEP pin added.
For any clustering strategy, the power grid and its connection to the standard cells are created. The smallest region that can be power-gated is limited by the distance from the V_{gnd} power rail. The maximum distance from the V_{gnd} is equal to the distance between two power stripes of the same type divided by two. To avoid having regions of different sizes the rows are always cut in the same points and those are determined by the position of the vertical stripes. Each cut, if needed, is done at the maximum distance from the V_{gnd} lines. To avoid wrong substrate polarizations the distance between two stripes of the same type has been kept equal to the one without power gating, while the width of the stripes has not been changed in order to avoid the IR drop increase.
In the remainder of this chapter, we will consider row-based sleep transistor insertion as the reference power-gating strategy. However, the solutions we present could be successfully adapted to other kinds of sleep transistor insertion approaches.

4 Clustered Tunable Power-Gating for PV Compensation

Parametric variations introduced by manufacturing represent the main cause of performance variability and yield degradation in modern VLSI circuits.
As discussed in Section 2, process variations may range from few percent to several orders of magnitude, following a deterministic scheme or a random distribution, also depending on the spatial-scale they reach: Wafer-to-wafer (W2W), die-to-die (D2D) and with-in-die (WID) variations. While the binning method has been successfully adopted for tackling W2W and D2D variations, considering WID variations (which are random, more localized and thus harder to manage) require dedicated countermeasures that must be taken since the early phases of the design flow.
Several design methodologies have been proposed in the last years, from those based on *Design for Manufacturability* approaches [13, 25, 18] (like litho-friendly and Restricted Design Rules layout design), where variability of a given design is either mitigated or amortized, to *Adaptive* strategies, which attempt to solve the variability issues by sensing and correcting the desired parameters using various knobs that affect them. These schemes are also called *Monitor & Control* (M&C) strategies, to emphasize their analogy with closed-loop control systems.

M&C approaches have proven to be extremely efficient and they are usually preferred to other strategies for their flexibility. Different embodiments of the M&C paradigm have been proposed recently. The main differentiating factor is represented by the strategy used to control the circuit. Most solutions use Dynamic Voltage Scaling (DVS) [35], or Adaptive Body Biasing (ABB) [34] as control knobs. Although both DVS and ABB are effective in adjusting circuit performance, as a side-effect they have a dramatic impact on the power consumption of the circuit; in fact, dynamic power is quadratically related to the supply voltage, while subthreshold leakage current shows an exponential relationship to the body voltage. This makes their implementation energy inefficient, thus less effective for leakage-dominated CMOS technologies.
In this section, we show how power-gating may represent a viable solution to achieve concurrent power reduction and performance control for mitigating process variations (and random WID variation in particular) and increase the timing yield. We show that, when enhanced with tunable features, the sleep transistors can act as natural supply-voltage regulators for the power-gated circuit, thus providing a low-power, yet low-cost, control knob.

4.1 Modeling Process Variations and Timing Yield

WID variations are mostly due to systematic and random variations of several physical device parameters, such as the concentration of doping atoms in the substrate (N_i), the effective channel length (L_{eff}) and width (W_{eff}) and the oxide thickness (T_{ox}). The resulting effect is the shift of the electrical characteristics of the transistors, like the threshold voltage (V_{th}) and the maximum current density (I_{ds}/W), with a significant impact on the power dissipation and the performance of a manufactured circuit [6].
Under these conditions, device parameters (and design metrics) have to be treated as random variables, whose probability distribution function (PDF) depends on the actual fabrication process. As pictorially summarized in Figure 6, deviations from the ideal case (i.e., where a given parameter X has a nominal, deterministic value) are represented by a PDF with a given shape with average value $avg(X)$ coincident with the nominal value X_{nom} of the parameter; variability is related to the variance of the PDF (e.g., the range of values between $\pm 3\sigma$). Under the typical assumption that values slightly exceeding the nominal value (X_{max} in the figure) can be considered as acceptable, we define the timing yield as the probability that $X < X_{max}$, which can be immediately obtained by the cumulative distribution function of X (area below the PDF and delimited by X_{max}; striped region in the figure).

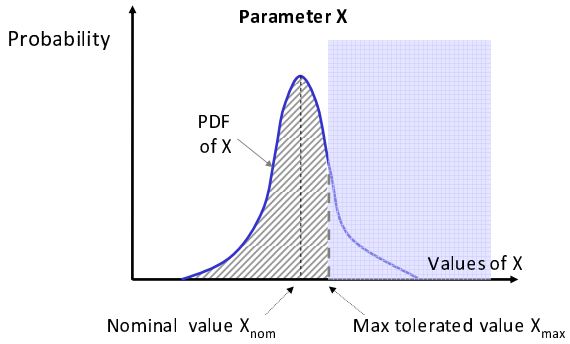

Fig. 6 Effects of Parameter Drifts due to Process Variations.

4.2 Controlling Performance with Tunable Sleep-Transistors

The active current I_{on} drained by a circuit from the power supply during normal operation may represent a suitable metric for quantifying the performance degradation induced by process-variation. In fact, the imperfections of the fabrication process may affect several electrical parameters (such as channel dimensions and threshold voltage), which in turn alter the actual current capability of the active transistors, and thus, the intrinsic speed of the devices. Therefore, the larger the speed degradation due to process variations, the smaller the resulting I_{on}.
During the active periods, the sleep transistor, that is turned-on and that operates in the linear region (in Figure7, R_{on} represents the channel resistance in the linear region), behaves as a current-to-voltage transducer that transforms the flowing current I_{on} into a voltage drop (V_{drop} in Figure7). The virtual-ground rail is now at a potential higher than ground, and the circuit operates at a scaled supply voltage ($V_{dd} - V_{drop}$). By modulating R_{on}, and thus V_{drop}, it is therefore possible to change the operating point of the entire circuit: A lower R_{on} implies a faster circuit; a larger R_{on}, on the contrary, will increase V_{drop} thus making the circuit slower.

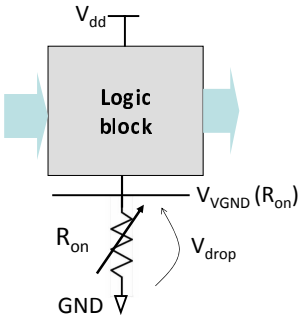

Fig. 7 Voltage Drop Across the Sleep Transistor During Active Periods.

In summary, circuits made slower by PV can recover their speed by means of R_{on} reduction, where R_{on} is proportional to the width of the sleep transistor.
The key aspect of this strategy is that the voltage regulation obtained through R_{on} appears only temporarily while the circuit is switching. Once all the internal switchings are completed, I_{on} falls to zero and so does V_{drop}. In this sense, the supply voltage V_{dd} is automatically adapted to the load, and there is no need of an external voltage regulator. A quantitative evaluation of the above dependency is given in Table 1, which reports delay values of a sample design (10 power-gated chains in parallel, each consisting of 10 inverters) as a function of the total sleep transistor width, in the presence of process variations.

d_{nom} [ps]	d_{WC} [ps]				
	W	$2W$	$4W$	$8W$	$16W$
130.2	152.7	147.1	140.8	138.7	134.9

Table 1 Quantitative Effect of Sleep Transistor Sizing on Delay.

Column d_{nom} reports the nominal delay when the circuit is power gated with a transistor of a given size W (in the example 20 "fingers" of 0.8μm each for a total of 16μm) and ignoring the presence of variations. Columns d_{WC} report worst case delay values for different values of W. By worst case, we mean the extreme value of the delay distribution obtained by randomly selecting 1000 different instances of the circuit with Monte-Carlo. The column d_{WC}–W shows then the slowest circuit instance (152.7 ps) when a transistor of size W is used for gating, i.e., the same conditions for which d_{nom} is measured. This instance is approximately 17% slower than the nominal one. If we then upsize the sleep transistor for such circuit instance, we see that, as expected, we can progressively speed up the instance, asymptotically reaching the nominal delay d_{nom}.

4.3 Design Issues and Architectures

4.3.1 Design Issues

As mentioned above, R_{on} modulation allows to compensate delay variations with a simple mechanism and with an arbitrarily fine granularity. However, such powerful control mechanism does not come for free, and designers must take into consideration the amount of overhead introduced by a tunable power-gating architecture.
Upsizing the sleep transistor may result in a significant area overhead, which causes extra static and dynamic power consumption. In fact, the sleep transistor itself leaks when turned off, and its leakage is proportional to its size. Moreover, driving a huge sleep transistor during power-mode transitions would imply additional load capacitance and larger logic effort. Such overhead may nullify the leakage power savings obtained by gating the circuit.

As an example, consider again the data of Table 1. If a 5% delay increase can be tolerated (i.e., $d_{max} = 130.2 \cdot 1.05 = 136.71ps$), then the sleep transistor must be as large as $16W$ to achieve the required speedup; this will cause the circuit to have a delay of $134.9ps < d_{max}$, but a power consumption due to the sleep transistor that is 16 times larger than in the nominal case. Clearly, without a dedicated architecture the use of power-gating as a M&C strategy may not be feasible.

4.3.2 Architectures

As described in Section 3, a key design variable in determining the area/power trade-off of a power-gated circuit is the granularity at which the sleep transistor is inserted. This is true also when considering tunable power-gating architectures.
A first option consists of power-gating the whole circuit using a single tunable sleep transistor (left configuration in Figure 8). In this case, even if the transistor is set to the proper width according to the detected delay value, its size must be the largest of the range W_{max}. Therefore, although effective in mitigating the process variation effects, large area and power overheads can not be avoided.

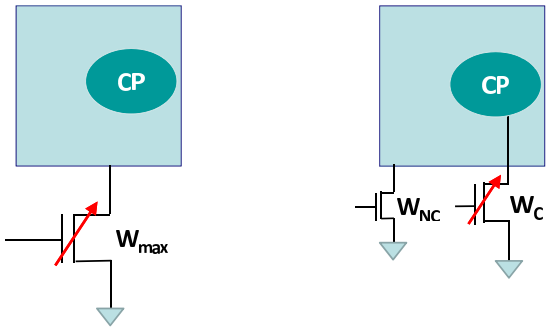

Fig. 8 Tunable Power-Gating (TPG) Options: Full-TPG (Left) and Clustered-TPG (Right).

A clustered sleep transistor insertion, on the other hand, may represent the most appropriate architecture. In fact, tunability is required only for the cells that determine the critical paths (CPs). On the contrary, all the other cells (i.e., cells for which the delay increase due to process variations do not slow down the critical paths) can be power gated with a regular, non-tunable, small-sized sleep transistor. We call this approach Clustered-Tunable Power-Gating (right configuration in Figure 8). Under this scheme, two distinct sets of cells (i.e., clusters) use separate sleep transistors: The critical cells identify the critical cluster C, which is gated by a tunable transistor of size W_C; the rest of the cells form the non-critical cluster NC, which is gated by a regular transistor of size W_{NC}. Concentrating tunability only where needed allows to reduce the total sleep transistor width ($W_{NC} + W_C \ll W_{max}$), and guarantees power savings while keeping the same delay compensation capability.

4.3.3 Design Flow and Results

The design methodology for automated implementation of clustered TPG does not differ from that described in Section 3. Starting from a (row-based) placed design, clustering, peak-current estimation, sleep transistor sizing and sleep transistor insertions are the main phases of the flow. However, working with clustered tunable power-gating architectures requires some changes and additions to the algorithms.

Design of the Tunable Sleep Transistor Cell: Tunable sleep transistor cells are not available in standard CMOS libraries. Then, it is important to support design kits with new customized cells that contain parallel sleep transistors of different sizes and driven by dedicated control signals. Figure 9 shows the schematic of a possible architecture, as described in [14] and [32]. Each parallel sub-transistor is driven by a NAND gate that receives an external configuration bit (*b3*, *b2*, *b1*, *b0*, in Figure 9), and whose value can enable (in case of 1-logic) or disable (in case of 0-logic) the corresponding transistor. An additional sleep signal provided by an external power-management unit is in charge of defining the operating mode of the gated circuit.

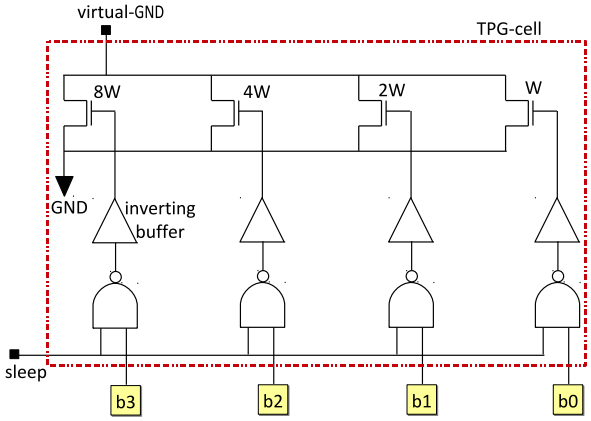

Fig. 9 Architecture of the Tunable Sleep Transistor Cell [32].

Statistical Static Timing Analysis: At design time, characterizing the effects induced by process variations is key. In fact, this allows the identification of the critical paths in presence of process variations. To do so, an option consists of integrating probabilistic models into standard Static Timing Analysis (STA) engines. For instance, it is possible to resort to Monte Carlo statistical sampling, where device parameters are stochastic variables described by process-dependent PDF. During each sample, the circuit timing is computed using traditional STA tools. The output of a Monte Carlo analysis includes the path delay distributions. From that, we can extract the list of statistical critical paths, i.e., the list of paths that show a certain probability to have a slack smaller than a user defined threshold.

Clustering and Sleep Transistor Sizing: The clustering phase is crucial, since it defines which portion of the circuit has to be assigned to which cluster (critical *C* or non-critical *NC*). At this stage, it is important to identify the granularity at which the sleep transistors are inserted into the layout. As described in Section3, different options are available, but row-based insertion can guarantee the finer granularity at the minimum cost in terms of area and layout disruption. Hence, the clustering problem comes down to the selection of which rows have to be considered as critical or non-critical. An effective solution to this problem is to mark as critical those rows that host at least one gate belonging to a statistical critical path [32]. Once the clusters *C* and *NC* are formed, it is possible calculating their maximum active currents, I_C and I_{NC}, respectively, and performing the sleep transistor sizing which returns the actual width of the two transistors W_C and W_{NC}.

Sleep Transistor Insertion: During this stage, the sleep transistor cells belonging to a customized power-gating library are placed into the layout. If a row-based approach is adopted, one can follow the strategy proposed in [31], where the sleep transistors cells are placed in dedicated layout rows, called sleep rows. It is worth mentioning that the sleep transistor of the critical cluster *C* is implemented by means of modular tunable sleep transistor cells connected in parallel. All the programmable sleep transistor cells are centered in their middle configuration, namely, the configuration word is set in the middle of the dynamic range (1000 for the 4-bits cell shown in Figure 9). This guarantees the maximal extension of the compensation range. For the non-critical cluster one can use the same scheme, but using non-tunable sleep transistor cells of fixed size [8].

Figure 10 offers a quantitative analysis of the timing yield and leakage savings that a clustered tunable power-gating architecture (*CLUSTERED-TPG*) may guarantee w.r.t. standard power-gating (*FULL-PG*) and block-level tunable power-gating (*FULL-TPG*). The results are obtained from the simulation of a subset of the ITC'99 benchmarks mapped onto an industrial 45nm technology. Only the averages are reported.
The tunable approach (i.e., *CLUSTERED-TPG* and *FULL-TPG*) originates a timing yield increase from 80,67% (*FULL-PG*) to 100%; in other terms, tunable power-gating allows to fully recover any speed degradation induced by process variation, thus making the number of circuits that must be discarded because they incur timing violations falls to zero.
Concerning leakage power, we observe that a *FULL-TPG* architecture can successfully tackle the problem of variability compensation at the price of a drastic reduction of the achievable leakage savings, due to a larger sleep transistor (leakage savings go from almost 100% for the *FULL-PG* to a mere 24.36% for *FULL-TPG*). On the contrary, a clustered scheme (i.e., *CLUSTERED-TPG*) guarantees the same timing yield as *FULL-TPG*, while maintaining reasonable leakage savings (72.1%). Clearly, applying a clustered architecture is what makes the tunable approach applicable to real-life circuits.

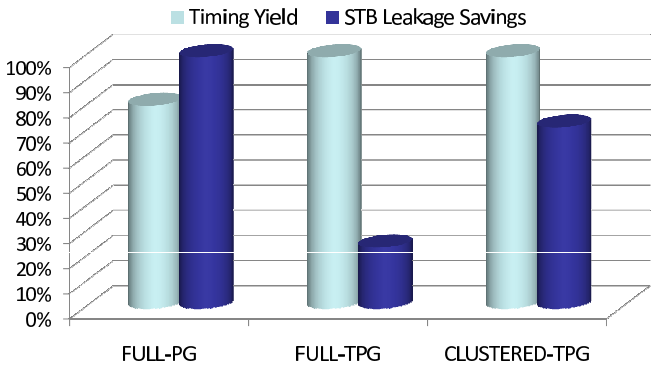

Fig. 10 Quantitative Comparisons of Various Power Gating Options.

5 Clustered Power-Gating for NBTI-Induced Aging Minimization

Besides the non-deterministic variations described in Section 2, another, and possibly more insidious, type of non-ideality of scaled devices concerns *time-dependent* deviations in their operating characteristics [1].

There are two types of sources of such time-dependent variations: Bias Temperature Instability (BTI), and Hot Carrier Interface (HCI) [26]. Both phenomena cause the generation of traps at the interface between the silicon and the oxide, resulting into an increase over time of the threshold voltage of the transistors.

BTI affects both pMOS (Negative BTI – NBTI) and nMOS transistors (Positive BTI – PBTI); in current technologies, the impact of PBTI is much lower than that of NBTI, although its importance is expected to increase with the adoption of high-k dielectrics in the gate-oxide interface [24]. Conversely, HCI effects are much more significant in nMOS transistors, and are at least two orders of magnitudes larger than for pMOS devices [12].

Between NBTI and HCI, NBTI is regarded as the most significant effect, because the surface along which interface traps are created (i.e., the whole silicon-oxide interface) is much larger than that of HCI (i.e., in the neighborhood of the drain area) [1].

We can thus consider NBTI as the dominant source of aging of transistors in current sub-45nm bulk CMOS technologies. As we will show in the rest of this section, the peculiar properties of NBTI and, in particular, its state-based manifestation will allow to use power-gating as a powerful knob for aging control, while keeping the usual benefits in static power reduction.

5.1 Background and Models

NBTI effects occur when a pMOS is negatively biased (i.e., negative V_{gs}, or a logic 0 is applied – the stress state) and originate an increase of the threshold voltage. When a zero-bias voltage is applied (i.e., a logic '1'), NBTI stress is actually removed, resulting in a partial recovery (i.e., a decrease) of the threshold voltage (the recovery state). While there is no full consensus on the exact quantum-mechanical mechanisms that govern the NBTI effects, the reactivation-diffusion model is accepted as accurate enough for pMOS NBTI aging [1]. A simplified version of such a model is the following:

$$\textit{Stress State}: \quad \Delta V_{th} = k_s e^{\frac{-E_a}{kT}} (t - t_{str})^{\frac{1}{4}} \tag{5}$$

$$\textit{Recovery State}: \quad \Delta V_{th} = k_r (1 - \sqrt{\frac{t - t_{rec}}{t}}) \tag{6}$$

where k_s and k_r are two parameters whose magnitude depends on few technological parameters (such as channel strain and nitrogen concentration), k is the Boltzmann constant, T is the device temperature, E_a is the activation energy, and t_{str} and t_{rcv} denote stress and recovery times, respectively. t is the free variable and expresses the temporal evolution of the threshold voltage drift.

Equations 5 and 6 show the qualitative behavior of NBTI stress and recovery: V_{th} increases during stress with $t^{\frac{1}{4}}$ dependency, it decreases during recovery with $1 - \sqrt{1/t}$ dependency, and it depends exponentially on temperature during stress. The most relevant feature of the model, however, is that there are two different behaviors based on the *state* of the device. This observation, coupled with the experimental evidence that NBTI aging is *frequency-independent* [1, 22], implies that it is the total stress time that matters rather than the actual dynamics of stress-recovery.

A generic signal applied to a pMOS transistor can then be modeled as a periodic waveform with equivalent amount of stress time, paving the way to a probabilistic modeling of NBTI aging. This allows to lump the models of Equations 5 and 6 into a single macromodel suitable to circuit-level simulation:

$$\Delta V_{th} = K \cdot \beta \cdot t^{\frac{1}{4}} \tag{7}$$

where K includes all the technological and environmental (e.g.,temperature, supply voltage) constants and β denotes the stress probability of the signal connected to the gate input of the pMOS transistors, that is, the probability of a logic '0'. This macromodel is also more suitable for translating the increase of threshold voltage into a delay degradation, which better corresponds to the intuition of "aging".

Under the alpha-power law, we can write the delay of a logic gate as:

$$d = \frac{C_L \cdot V_{dd}}{(V_{gs} - V_{th})^{\alpha}} \tag{8}$$

where C_L is the load capacitance, V_{gs} is the gate voltage and α is a technology-related exponent that can be approximated to 1 for sub-90nm technology.

Because of the NBTI-induced threshold voltage increase, the "aged" delay $d' > d$ then becomes:

$$d'(t) = \frac{C_L \cdot V_{dd}}{(V_{gs} - (V_{th} + \Delta V_{th}(t)))} \tag{9}$$

which can be expressed as a penalty with respect to d as:

$$d'(t) = d \cdot (1 + \frac{K \cdot (\beta \cdot t)^{1/4}}{V_{GT} - K \cdot (\beta \cdot t)^{1/4}}) \tag{10}$$

where $V_{GT} = V_{gs} - V_{th,0}$ and $V_{th,0}$ is the threshold voltage at time 0.
Equation 10 allows translating circuit operations (in terms of signal probabilities) to aging (i.e., delay increase over time) for the individual gates, similarly to what it is done for estimating dynamic power based on switching probabilities.

5.2 Power-Gating and Aging Reduction

Equation 7 clearly shows that, for a specific manufactured device and for specific operating conditions (temperature and V_{dd}), there is one single knob that can be used for mitigating the aging effects: The stress probability β.
Ideally, one would like to make β as small as possible. This would imply having as many 1's in the logic network as possible for the largest possible fraction of time. Obviously, under normal conditions, this is infeasible, because: (i) To implement meaningful functions, circuits require logic inversion, which by definition prevents achieving arbitrarily small probability for a predetermined signal value (0 or 1). (ii) Circuit structure affects probability. On the other hand, information theory suggests that a signal probability of 0 or 1 carries no information – entropy is 0, and it is maximum for a 0.5 probability. Therefore, to implement a realistic function, there must be a fair distribution of 0's and 1's.
Although the ideal objective is impossible to reach, technology-independent and technology-dependent synthesis techniques can be used to *minimize* the 0-probability of internal signals. Kumar *et al.* [23] proposed multi-level synthesis and technology mapping algorithms that adopt a modified, NBTI-oriented metric, while Wu and Marculescu [19] used logic restructuring and pin reordering to exploit the fact that not all transistors are identically important in determining the delay of a gate in the pull-up network. These strategies can reduce the aging in the *typical* state of the circuit (i.e., determined by the implementation and by the most common input patterns) and achieve a sort of local optimum.
We propose power-gating as a knob for improving over these results, aiming at a global optimum. We suggest leveraging a well-known weakness of power-gating: The logic values of the nodes of a power-gated block are lost when the block is disconnected from the power supply or the ground nets. This poses serious problems when storing values in memory elements and when interfacing power-gated to non power-gated regions. Solutions based on the usage of special types of memory elements [3] or by proper design of the sleep transistor cell [16] do exist.

For the combinational portion of a circuit, the behavior of the internal nodes depends on the type of switch (footer or header) implementing power-gating. If a footer switch is used (and thus the block is disconnected from the ground), as described in Section 3, an interesting behavior occurs. Nodes that are at logic value 1 before opening the switch do keep their values, whereas nodes with value 0 become floating. Both the virtual ground line and the 0 nodes then get charged to 1 by the leakage current of the pull-up network of the cells [27]. The speed of this charge process depends on the design of the sleep transistor cell, and can be sped up by using a proper pull-up boosting mechanism, as shown in [7].
For the sake of illustration, Figure 11 plots the signals of a simple two-inverter circuit connected to a footer switch after the latter has been turned off (0.0 on the timescale) [10]. We observe that the signals *a* and *c*, originally at logic value 1, stay unchanged, while signal *b*, at 0 when the sleep signal is activated, goes to 1 quite abruptly, and it reaches about 85% of V_{dd} in a few tens of nanoseconds.

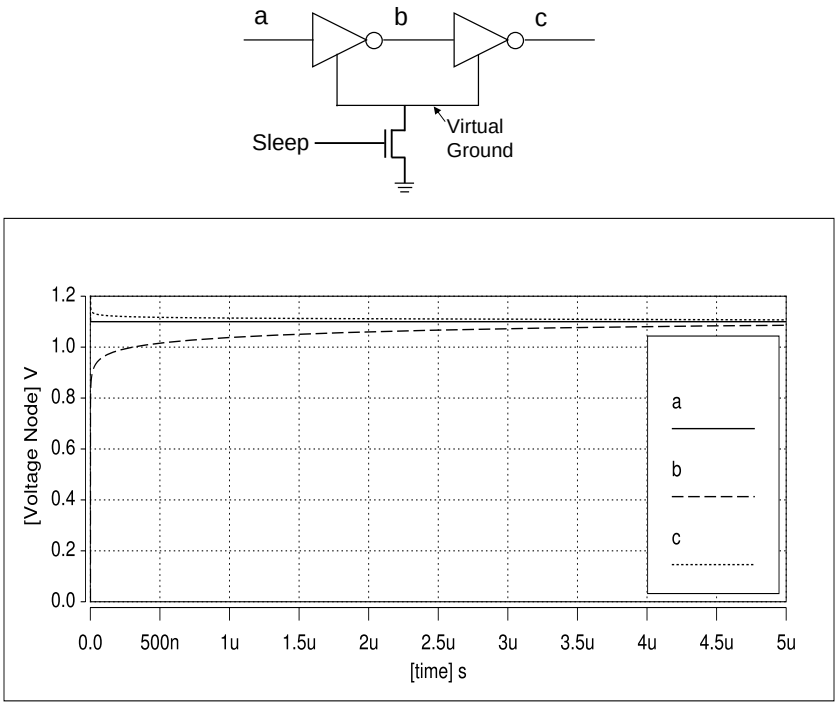

Fig. 11 Internal Signals during Standby Intervals [10].

What it matters for our purposes is that all the nodes inside the gated block region (more or less quickly) reach a logic value of 1, that is, *the gated block recovers from aging during the standby intervals*. This is clearly a non-logical state that cannot be obtained by any mechanism that operates on the circuit function or the input data, and is therefore orthogonal to such design approaches.

5.3 Design Issues and Architectures

5.3.1 Design Issues

From the analysis of the previous section, two key issues deserves a deeper evaluation. First, it is evident that the benefit in terms of aging goes together with the potential leakage reductions: If the gated block is put in a standby state sporadically, the aging reduction will be marginal. Clearly, the amount of time spent in the standby state is not a quantity that is controllable by the designer and it is determined by the environment. Therefore, it is important to parameterize the aging in terms of this quantity. Second, the implementation of power-gating does not come for free. As discussed in Section 3, the addition of a footer switch in series with the pull-down network increases the on-resistance of each cell and results into a (nominal, i.e., at time zero) delay penalty. Since by tackling aging we are trying to compensate the increase of delay over time, it is fundamental considering the fact that in a power-gated circuit we start from a larger nominal delay value.
Equation 10 can be adapted so as to incorporate the above described effects, by adding two more parameters: P_{sleep}, the probability of the sleep signal (0 means always active, 1 always in standby), and γ, the delay penalty the due to the addition of the sleep transistor (i.e., $d_s = d \cdot (1+\gamma)$ is the time-zero delay of the gated block). The new expression of the delay model of a gated block becomes:

$$d'_s(t) = d_s \cdot (1 + \frac{K \cdot (\beta \cdot (1 - P_{sleep}) \cdot t)^{1/4}}{V_{GT} - K \cdot (\beta \cdot (1 - P_{sleep}) \cdot t)^{1/4}}) \tag{11}$$

Notice that the stress probability β is now modulated by the complement of the sleep probability $(1 - P_{sleep})$; this is equivalent to an "effective" stress probability ($\beta' = \beta \cdot (1 - P_{sleep})$, with $\beta' \cdot t$ is now the effective stress time.
Figure 12 plots Equations 10 and 11; the curve for $d'_s(t)$ is parameterized with respect to P_{sleep}, and it refers to a specific value of γ. Both curves for $d(t)$ and $d'_s(t)$ refer to a value of $\beta = 0.5$. The plot is qualitative just for the sake of illustration of how the aging profiles change based on the various parameters.
The aging curves for power-gating start at a higher time-zero delay value (d_s), but they exhibit an amount of recovery that is proportional to the percentage of standby time P_{sleep}. The diamonds in correspondence to the intersections between the non power-gated curve and the power-gated ones show the breakeven points, that is, the points in time after which the power-gated circuits start aging more slowly than their non power-gated counterparts. Clearly, the intersection moves earlier in time as P_{sleep} increases, denoting better aging profiles.
There are three parameters that affect the above analysis: P_{sleep}, β, and γ. The first two are not under the designer's control and depend on the functional and environmental characteristics of the circuit; for a given implementation, they assume a well defined value. On the contrary, γ is a design variable; its magnitude is related to the sleep transistor size [2]–[31]: A larger (smaller) γ implies a smaller (larger) sleep transistor. The ideal case of $\gamma = 0$ would correspond to a transistor of infinite size.

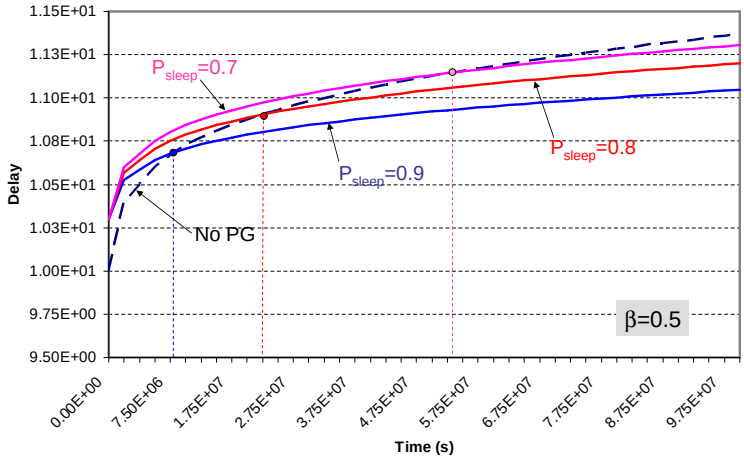

Fig. 12 Delay Degradation as a Function of Sleep Probability.

5.3.2 Architectures

The discussion above assumes the entire logic block being power-gated by a single sleep transistor. But, the aging benefits of power-gating are proportional to γ; a large value of d_s may result in a breakeven point which might be beyond the typical lifetime horizon of the circuit. Therefore, it is essential to keep γ as small as possible. One way for achieving this is to selectively apply power-gating. A first option is *partial power-gating*, in which power-gating is applied only to the non-critical gates of a circuit (left schematic of Figure 13). Assuming a coarse-grain implementation of power-gating, gating only a portion of the circuit requires the interaction with the physical placement of the cells [11, 31]. By not slowing down critical gates, this solution has the same delay as the nominal one ($d_s \equiv d$) and it requires a sleep transistor width, W, smaller than that required for power-gating the entire circuit. However, from the aging viewpoint, partial power-gating is almost identical to the non power-gated case. In fact, keeping the critical gates un-gated means that they will age as in the original circuit. Then, partial power-gating allows achieving leakage reductions at zero delay overhead, but it does not yield any aging benefit.

To get concurrent leakage and aging benefits, we can adopt a clustered architecture similar to the one of Section 4. The entire circuit is power-gated using two sleep transistors (see the right part of Figure 13): One for the critical gates (of size W_C) and another one for the non-critical gates (of size $W_{NC} \ll W_C$) [10]. The idea is to decouple the sleep transistor problem by using a small transistor for the non-critical gates (with the only purpose of saving leakage) and a larger one with smaller performance penalty for critical cells. If $W_C \ll W$, with the whole circuit gated, we can achieve leakage savings comparable to the case of full power-gating.

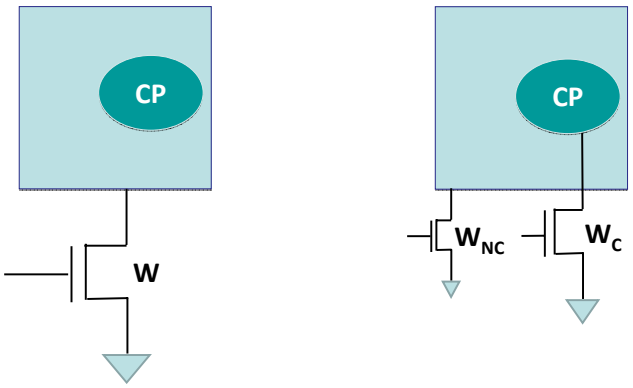

Fig. 13 Power-Gating Options: Non-Critical Gates Only (Left) and Clustered (Right).

5.3.3 Design Flow and Results

As for the case of PV-aware power-gating, the methodology for an automated implementation of an aging-aware clustered power-gating requires specialized algorithms to be integrated with the standard flow shown in Section3.

Library Characterization: Cell libraries are usually not characterized for aging. To this purpose, the designer has to fill look-up tables containing the delay degradation of each cell, parameterized with respect to the static 0-probabilities of the inputs, stress voltage (i.e., V_{gs}), and temperature. This task could be split into two phases by first characterizing the pMOS transistors (and modeling the corresponding ΔV_{th} variation) and then incorporating it into the cell as a negative voltage-source on the gate-terminal of pMOS transistors.

Aging-Aware Timing Analysis: This step consists of a probabilistic simulation that uses the parameterized cell delay models and statistics of the input signals (static probabilities) to determine the paths with a delay that is within a given percentage of the nominal critical path d.

Clustering: Clustering entails the determination of the critical and non-critical clusters, i.e., two subset of the cells that maximize the leakage and aging benefits. The selection of which cells go in which cluster strongly interacts with the type of sleep transistor insertion strategy. In particular, the granularity of the gating unit coincides with the granularity of the cluster assignment. For instance, if the row-based strategy of Section 3 is adopted, clusters consist of *rows*, and therefore the assignment to clusters is done on a row-by-row basis. In [10], the clustering problem is formulated as a 0-1 nonlinear program, where the unknowns are the membership of a gating unit (in that work, layout rows) to one of the two clusters.

Sleep Transistor Sizing: Sleep transistor sizing is dictated by the timing (γ) constraints, and involves, as for clustering, the estimation of the maximum current drawn by each cluster, as discussed in Section 3.

The results of leakage power savings and lifetime extension achieved with the application of the clustered power-gating methodology are quite promising, and they are summarized in Figure 14. Three power-gating schemes are compared: *NO-PG* (no power-gating), *PG* (the entire circuit is power-gated) and *CLUSTERED* (the two-cluster architecture in the right part of Figure 13. Data are normalized with respect to the *NO-PG* case (both leakage and lifetime are assumed to be 1).
The lifetime of a circuit is defined as the time at which the circuit degrades its performance by 15% beyond its nominal value (i.e., performance of the non power-gated circuit). The results in the chart represent an average over a set of standard benchmarks and some medium-sized industrial designs, and refer to a 45nm industrial CMOS technology.

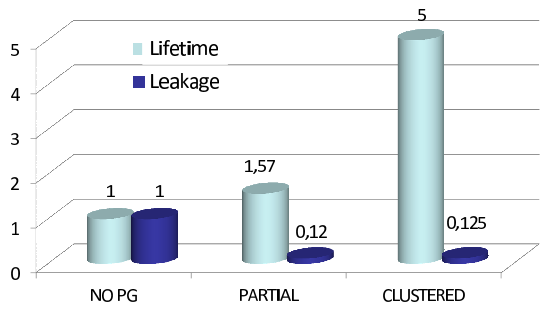

Fig. 14 Quantitative Comparisons of Various Power-Gating Options.

We observe that the *PG* scheme does not fully exploit the huge leakage reduction (about 88%) for lifetime extension; in fact, only a 57% increase is achieved. Conversely, the *CLUSTERED* architecture yields a 5X increase with respect to the non-gated version, with a negligible loss in leakage reduction (about 0.5% penalty with respect to *PG*). These numbers demonstrate experimentally the effectiveness of clustered power-gating as a tool for simultaneous leakage power minimization and aging effects mitigation.

6 Conclusions

There is wide consensus within the electronics industry that the scaling process of the CMOS technology will continue as far as the fundamental barrier of the atom size will be reached. Therefore, CMOS-based circuits and systems represent the future vehicle for digital applications of the next decade. Unfortunately, nano-scale CMOS technologies show intrinsic mechanisms, like internal leakage consumption and parametric variations, which make their use extremely challenging.
In this work, we explored the possibility of exploiting low-power techniques for concurrent leakage optimization and variability compensation, and both static (i.e., due to process variation) and dynamic (i.e., due to NBTI-induced aging mechanisms) variability have been considered. More specifically, we have shown how power-gating, when implemented through a clustered strategy, offers a suited performance control knob to compensate process variations, as well as a natural solution for reducing NBTI effects.
New design methodologies have been implemented to support variation-aware clustered power-gating, and experimental results, conducted on benchmark circuits mapped onto an industrial 45nm technology, have highlighted their effectiveness: 100% of timing yield in the presence of process variations have been achieved, with substantial (i.e., 5X) lifetime extensions w.r.t. non power-gated circuits.

References

[1] M. Alam (2008) “Reliability- and process-variation aware design of integrated circuits,” in *Microelectronics Reliability*, 48(8):1114–1122.
[2] M. Anis, S. Areibi, M. Elmasry (2003) “Design and optimization of multi-threshold CMOS (MTCMOS) circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(10):1324–1342.
[3] P. Babighian, L. Benini, A. Macii, E. Macii (2006) “Enabling fine-grain leakage management by voltage anchor insertion,” *IEEE Design, Automation and Test in Europe (DATE’06)*, pp. 868–873.
[4] D. Boning, S. Nassif (2000) “Models of process variations in device and interconnect,” in *Design of High Performance Microprocessor Circuits*, Wiley.
[5] S. Borkar (2005) “Designing reliable systems from unreliable components: The challenges of transistor variability and degradation,” *IEEE Micro*, 25(6):10–16.
[6] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, V. De (2003) “Parameter variations and impact on circuits and microarchitecture,” *ACM/IEEE Design Automation Conference (DAC’03)*, pp. 338–342.
[7] A. Calimera, A. Pullini, A. Sathanur, L. Benini, A. Macii, E. Macii, M. Poncino (2007) “Design of a family of sleep transistor cells for a clustered power-gating flow in 65nm technology,” *ACM/IEEE Great Lakes Symposium on VLSI (GLSVLSI’07)*, pp.501–504.

[8] A. Calimera, L. Benini, A. Macii, E. Macii, M. Poncino (2009) "Design of a flexible reactivation cell for safe power-mode transition in power-gated circuits," *IEEE Transaction on Circuits and Systems - Part I, Regular Papers*, 56(9):1979–1993.

[9] A. Calimera, E. Macii, M. Poncino (2009) "NBTI-aware power gating for concurrent leakage and aging optimization," *International Symposium on Low Power Electronics and Design (ISLPED'09)*, pp. 127–132.

[10] A. Calimera, E. Macii, M. Poncino (2010) "NBTI-aware clustered power gating," *ACM Transactions on Design Automation of Electronic Systems*, 16(1):3.1–3.25.

[11] L. Changbo, L. He (2004) "Distributed sleep transistor network for power reduction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(9):937–946.

[12] Z. Chen, K. Hess, J. Lee, J. Lyding, E. Rosenbaum, I. Kizilyalli, S. Chetlur, R. Huang (2000) "On the mechanism for interface trap generation in mos transistors due to channel hot carrier stressing," *IEEE Electron Device Letters*, 21(1):24–26.

[13] C. Chiang, J. Kawa (2007) *Design for manufacturability and yield for nano-Scale CMOS*, Springer-Verlag.

[14] H.-S. Deogun, D. Sylvester, R. Rao, K. Nowka (2005) "Adaptive MTCMOS for dynamic leakage and frequency control using variable footer strength," *IEEE International SOC Conference (SoCC'05)*, pp. 147–150.

[15] D. Flynn, R. Aitken, A. Gibbons, K. Shi (2007) *Low Power Methodology Manual*, Springer-Verlag.

[16] S. Henzler, G. Georgakos, M. Eireiner, T. Nirschl, C. Pacha, J. Berthold, D. Schmitt-Landsiedel (2006) "Dynamic state-retention flip-flop for fine-grained power gating with small design and power overhead," *IEEE Journal of Solid-State Circuits*, 41(7):1654–1661.

[17] ITRS (2009) "Process integration, devices & structures," in *International Technology Roadmap for Semiconductors*, ITRS.

[18] T. Jhaveri, V. Rovner, L. Liebmann, L. Pileggi, A. Strojwas, J. Hibbeler (2010) "Co-optimization of circuits, layout and lithography for predictive technology scaling beyond gratings," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(4):509–527.

[19] W. Kai-Chiang, D. Marculescu (2009) "Joint logic restructuring and pin reordering against NBTI-induced performance degradation," *IEEE Design, Automation and Test in Europe (DATE'09)*, pp. 75–80.

[20] J. Kao, S. Narendra, A. Chandrakasan (1998) "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," *ACM/IEEE Design Automation Conference (DAC'98)*, pp. 495–500.

[21] V. Khandelwal, S. Srivastava (2007) "Leakage control through fine-grained placement and sizing of sleep transistors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(7):1246–1255.

[22] S. Kumar, C. Kim, S. Sapatnekar (2006) "An analytical model for negative bias temperature instability," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD'06)*, pp. 493–496.

[23] S. Kumar, C. Kim, S. Sapatnekar (2007) "NBTI-aware synthesis of digital circuits," *ACM/IEEE Design Automation Conference (DAC'07)*, pp. 370–375.

[24] S. Kumar, C. Kim, S. Sapatnekar (2009) "Adaptive techniques for overcoming performance degradation due to aging in digital circuits," *IEEE Asia and South Pacific Design Automation Conference (ASPDAC'09)*, pp. 284–289.

[25] M. Lavin, F. L. Heng, G. Northrop (2004) "Backend CAD flows for restrictive design rules," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD'04)*, pp. 739–746.

[26] S. Mahapatra, D. Saha, D. Varghese, P. Kumar (2006) "On the generation and recovery of interface traps in mosfets subjected to NBTI, FN, and HCI stress," *IEEE Transactions on Electron Devices*, 53(7):1583–1592.

[27] E. Pakbaznia, F. Fallah, M. Pedram (2008) "Charge recycling in power-gated CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(10):1798–1811.

[28] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand (2003) "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, 91(2):305–327.

[29] A. Sathanur, A. Pullini, L. Benini, A. Macii, E. Macii, M. Poncino (2008) "Optimal sleep transistor synthesis under timing and area constraints," *ACM/IEEE Great Lakes symposium on VLSI (GLSVLSI'08)*, pp. 177–182.

[30] A. Sathanur, L. Benini, A. Macii, E. Macii, M. Poncino (2011) "Fast computation of discharge current upper bounds for clustered power-gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19(1):146–151.

[31] A. Sathanur, L. Benini, A. Macii, E. Macii, M. Poncino (2011) "Row-based power-gating: A novel sleep transistor insertion methodology for leakage power optimization in nanometer CMOS circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19(3):469–482.

[32] L. D. Silva, A. Calimera, A. Macii, E. Macii, M. Poncino (2011) "Power efficient variability compensation through clustered tunable power-gating," *IEEE Journal of Emerging and Selected Topics in Circuits and Systems*, 1(3):242–253.

[33] D. Sylvester, K. Agarwal, S. Shah (2008) "Variability in nanometer CMOS: Impact, analysis, and minimization," *Integration - The VLSI Journal*, 41(3):319–339.

[34] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, V. De (2002) "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, 37(11):1396–1402.

[35] J. Tschanz, S. Narendra, R. Nair, V. De (2003) "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *IEEE Journal of Solid-State Circuits*, 38(5):826–829.