RC25266 (W1202-006) February 8, 2012
Electrical Engineering

IBM Research Report

Goldilocks Failures: Not Too Soft, Not Too Hard

Sani R. Nassif
IBM Research Division
Austin Research Laboratory
11501 Burnet Road
Austin, TX 78758
USA

Veit B. Kleeberger, Ulf Schlichtmann
Technische Universität München
Institute for Electronic Design Automation
Arcisstrasse 21
80-333 Munich, Germany

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Goldilocks Failures: not too soft, not too hard

Sani R. Nassif
Austin Research Laboratory, IBM Corporation,
11501 Burnet Road,
Austin, TX 78758
nassif@us.ibm.com

Veit B. Kleeberger and Ulf Schlichtmann
Technische Universität München
Institute for Electronic Design Automation
Arcisstr. 21, 80333 Munich, Germany
kleeberger@tum.de, ulf.schlichtmann@tum.de

***Abstract*—It is well known that circuits fail when one or more of the constituent components fails, due -for example- to phenomena such as electromigration in wires. Such *hard* failures, typically due to topological changes in circuit connectivity, are treated distinctly from *soft* failures which could be due to components drifting out of spec over time. However, in certain types of circuits, such as memory, the distinction between soft and hard failures is not clearly defined. The primary cause of the blurring between these two phenomena is manufacturing variability, which can make a topologically correct circuit behave as if it had a short or an open. This paper will show the linkage between these two failure types, and show how increasing variability in future technologies will likely exacerbate this problem further.**

I. History

There is no account[1] of the first *failure* found on an integrated circuit, similar to Grace Hopper's famous account of discovering a *bug* in the Harvard Mark II [1]. Nevertheless, over fifty years of creating ever smaller circuits naturally leads to an increasing set of mechanisms for those same circuits to misbehave. In the early days of digital integrated circuit technology, one could identify two distinct types of error-causing failures:

1) Hard failures which manifest themselves as changes in the intended topology of the circuit. Such failures are often caused by material imperfections or contamination, and cause the circuit to behave incorrectly, i.e. produce the wrong output. The reason they are characterized as *hard* is not that they are difficult to find (though one can imagine that some such defects can be maddeningly difficult to localize), but rather that they behave largely independently of the operating environment of the chip. An output which is shorted to ground will stubbornly produce a logic zero regardless of clock frequency, power supply voltage, or operating temperature.
2) Soft failures, in contrast, are related to shifts in the electrical properties of the elements that constitute the circuit. A transistor may exhibit excessively high leakage, or a resistor may have too low of a resistance. Under normal operating conditions, the circuit operates in a logically correct fashion but may fail to do so within a certain time, power budget, or over certain temperature ranges. The output of a circuit which has a soft failure is correct *sometimes*, and there is some correlation between this correctness and the current operating environment of the chip.

[1]That the authors know of at least.

For many technology generations, the major yield detractor was hard failures caused by particulate contamination, primarily affecting the lithography patterning process and causing shorts and opens in the resulting circuits.

As feature sizes were reduced; smaller defects could cause such problems, and this resulted in a strong (and very successful) push by equipment makers and fab operators to reduce particle counts by *cleaning up* the manufacturing plant. In this phase, yield management focused on reducing particle counts, and was largely independent of specific design practice.

With the advent of sub-wavelength lithography (around the 180nm node) the situation became somewhat more complex because the lack of pattern fidelity became an additional source of topology errors [2]. In addition, a host of variability-related phenomena came to the forefront as devices became so small that local atomistic effects became important. One such phenomenon is *Random Dopant Fluctuations* (RDF), which is caused by the inherent randomness of the ion implantation process used for doping the channels of modern day MOSFETs. The ion implantation process accelerates ions of impurities like Boron or Phosphorus and embeds them into a Silicon crystal (the device channel). Each accelerated ion collides with a number of lattice atoms, losing some energy with each collision before coming to rest at some location. This is a very chaotic process and the final location is quite random [3].

For past technologies with large devices, the number of atoms in a channel numbered in the tens of thousands or more, so one could employ the *law of large numbers* to get an average concentration of dopant atoms in the channel, and that average concentration would then determine the threshold voltage of the particular device in question. In current technologies, the number of atoms in the channel is *countable* and of the order of 40! This means that minor fluctuations in individual atom positions can have a large impact on the *relative* local concentration of dopants, and thus a profound impact on the threshold voltage of the device in question. Similar atomistic effects impact device dimensions (Line Edge Roughness) [4].

The upshot of these trends is that there has been a steady increase in the amount of variability incurred by heavily scaled devices. This variability is large enough to change the *character* of the impact of this variability on circuit

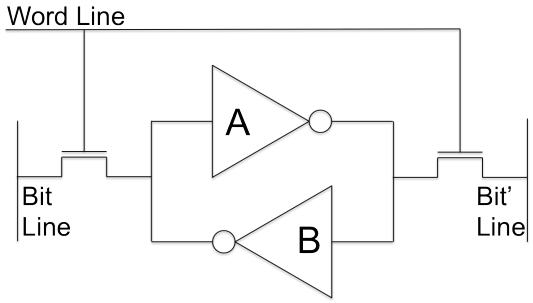

Fig. 1. An SRAM bit.

operation. While small amounts of variability can lead to modest performance fluctuations and exhibit themselves as *soft* failures, large amounts of variability can lead to significant performance fluctuations which cause the circuit to appear to have a *hard* failure.

It is precisely this overlap between soft and hard failures that we target in this paper.

II. Static Random Access Memory (SRAM)

The performance of microprocessors is heavily dependent on the amount of fast local memory (commonly referred to as the *Cache*) available for storing program instructions and the associated data [5]. This creates a strong incentive for putting as much memory as possible in close proximity to the processor. Achieving such high memory densities can only be done by using the smallest possible devices for creating that memory. Thus it is common for memory circuits to be amongst the most challenging of all components of a high performance processor or complex system-on-chip integrated circuit. The primary reason for this difficulty is that memory, by utilizing the smallest possible devices, becomes the most sensitive to manufacturing variations, and requires extensive modeling and analysis [6]!

A typical static random access memory circuit for a single storage bit is shown in Figure 1. The two inverters A and B are connected back-to-back such that they can be in one of two stable states, 1/0 or 0/1 (for the outputs of inverters A and B respectively). In addition there is an intermediate unstable state where both inverters are in their high-gain region, but that state is not useful because it is not stable and therefore exists only for short periods of time before naturally occurring noise will cause the circuit to fall into one of its two stable states.

The two *access devices* to the left and right connect the SRAM bit cell to the external bit lines so that the cell can be read or written, and are activated by the word line. While these devices are very important and contribute greatly to the performance as well as error susceptibility of the SRAM, we will set them aside for this discussion and focus on the *inner cell* composed of the inverters A and B.

To study the stability properties of the inner cell we construct a circuit composed of those inverters where we drive one side (the input to A) via a voltage source, shown in Figure 2.

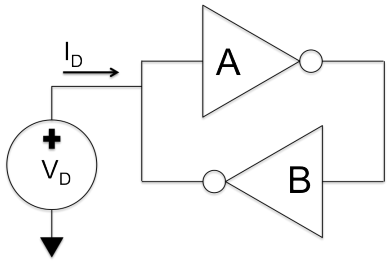

Fig. 2. An SRAM inner cell driven by a voltage source.

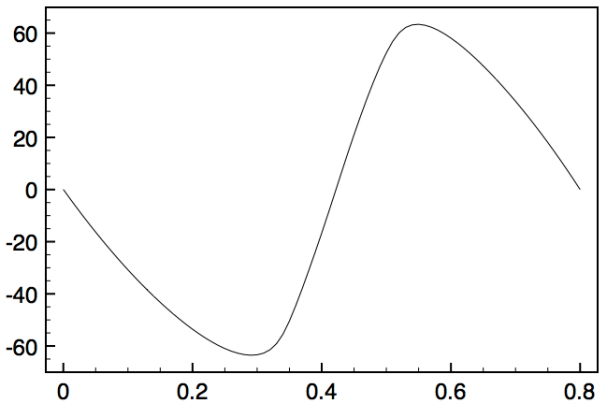

Fig. 3. A plot of driven current (μ Amps) vs. drive voltage (Volts).

For any given value of the *driving* voltage source V_D, we can simulate the circuit and measure the driving current I_D. We observe that if the driving current is zero, then this indicates that the circuit is in a *stable state*, meaning that it does not require the voltage source to be present in order to maintain its state. If the current I_D is non-zero, however, it is an indication that the voltage source is driving the circuit and that it is not in a stable state.

We perform a Spice [7] DC sweep simulation of the circuit in Figure 2 and plot the driving current I_D vs. the driving voltage V_D for a 22nm CMOS Technology with a power supply voltage (V_{DD}) of 0.8 Volts, and utilizing the 22nm Predictive Technology Model [8]. The plot is shown in Figure 3 and shows that the current is zero for three values of voltage, zero, V_{DD}, and an intermediate value corresponding to the unstable state of the SRAM bit cell.

We are interested in the behavior of the I_D vs. V_D curve as a response to manufacturing variability. We focus here on threshold voltage variability, since it is the dominant source of variability in SRAM devices of current CMOS technologies. Random Dopant Fluctuations affect each device independently from other devices, and thus our test circuit in Figure 2 would normally have four distinct values of threshold voltage for each of the P and N-channel devices comprising inverters A and B. Such a study is, of course, easily possible but our purpose in this section is to make the point that manufacturing variability can impact the operation of an SRAM bit cell in a manner similar to *hard* topological defects.

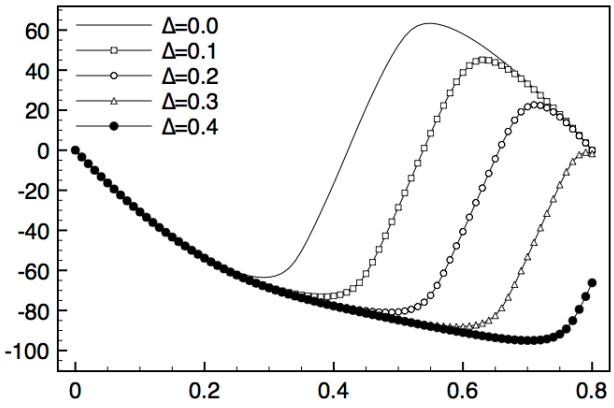

Fig. 4. A plot of driven current vs. drive voltage for various threshold voltage perturbations.

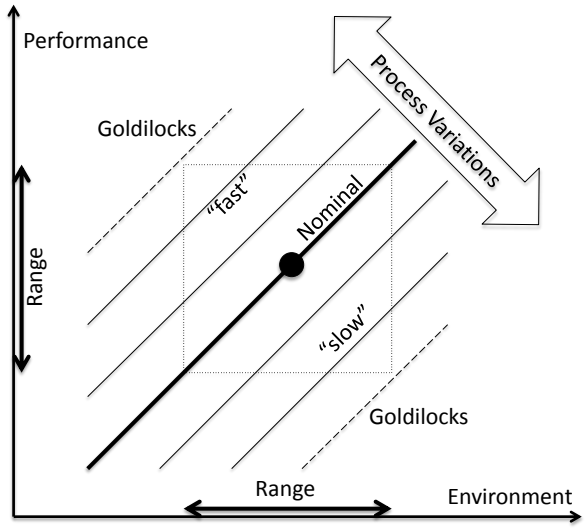

Fig. 5. Performance vs. Environment and manufacturing variability.

Based on the above, we performed a simplified simulation study where we chose to vary the threshold voltage of the transistors in inverter A only. Furthermore, we varied both the P and N-channel devices by the same amount Δ -thus making this a simple one-dimensional problem. Figure 4 shows plots of the driving current vs. voltage for various values of the threshold voltage shift Δ.

Consider the I_D/V_D curve for $\Delta = 0.4$ in Figure 4. The curve has but one single intersection with the x-axis, denoting that this SRAM bit cell has only one stable state (one where the input to inverter A is zero). Manufacturing variability has transformed a circuit with two stable states (and one unstable one) into another circuit with just one state! Whereas a normal SRAM cell can store both a zero and a one, our cell can now only store a zero no matter how long we might wait for it to store a one -so clearly not functioning as an SRAM bit cell. In fact, our cell is behaving in a manner similar to a cell where the input to inverter A is *shorted* to ground, i.e. a cell with a *hard* topological defect.

The example above showed an SRAM bit cell misbehaving due to excessive threshold voltage variability. An expert on SRAM operation, however, might claim that if the power supply voltage was raised sufficiently, our errant cell would indeed exhibit working behavior. This is a correct observation, but we know that any given technology or circuit has *limits* beyond which it cannot operate. For example, it is well known that one cannot apply arbitrarily high voltages to an integrated circuit because of the dielectric breakdown limit for gate oxides. Similarly, one cannot wait an infinite amount of time for a phenomenon to occur, thus any delay larger than some maximum can be considered as practically infinite, and -most likely- the result of a phenomenon behaving like a hard failure.

III. Goldilocks Failures

We propose that any variability-related failure which occurs over the full *practical* range of the operation of a circuit qualifies as a *Goldilocks* failure, i.e. a failure that appears hard, but is -in fact- caused by phenomena normally associated with soft failures.

Consider Figure 5 which shows a hypothetical plot of chip performance, such as frequency, vs. chip operating environment, such as the power supply voltage V_{DD}. On the x and y axes we also show the *range* for the two quantities. In a realistic example, power supply voltage usually has a lower limit below which the SRAM components in a chip cease to work, and an upper limit above which device breakdown becomes a problem. A similar argument can be made for the performance, where chips far from the desired specification are undesirable.

Consider the line labeled *Nominal* in Figure 5, which represents the behavior of a nominal chip. If the design is *centered* [9] properly with respect to process variations, we would expect it to be in the middle of the rectangle defined by the intersection of the operating and performance ranges. We will denote this rectangle, which represents the *practical* range of operation, as R_P. The slope of the line reflects the relationship between the performance and environment; in our example we know that operating frequency increases with increasing supply voltage.

Due to process variations, the performance of chips will vary and we show this in Figure 5 by plotting a family of curves with different values of performance at the nominal environmental condition. We further delineate two lines, labeled *"fast"* and *"slow"* which might represent a particular percentile of performance (e.g. the $\pm 3\sigma$ levels) and show the behavior of the chip for moderate amounts of variability. Note how chips which are far from the nominal process point have smaller ranges of operation within R_P.

With an even larger amount of variability, we would have chips which do not overlap with R_P at all. It is precisely these situations that we label as *Goldilocks* failure. As circuits become more complex, and as the range of operation becomes

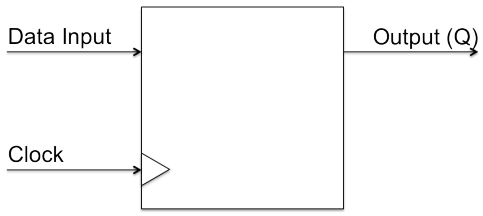

Fig. 6. A simple latch.

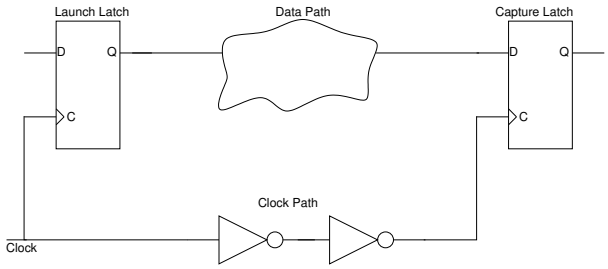

Fig. 7. A sequential circuit consisting of launching and capturing flip-flop, the combinational datapath between, and the clock path including clock buffers.

narrower due to ever lower voltage supplies, we believe that these Goldilocks failures will begin impacting other circuit structures besides SRAM. The key message of this paper is that this behavior, long observed in memories, is starting to become more common for other types of circuits [10]. Memories have the advantage of an array organization and an easily testable behavior, leading to sophisticated methods for overcoming such faulty behavior like redundancy and error detection and correction. The same cannot be said for general purpose circuits, however, hence the need for studying these failure mechanism [11].

IV. Goldilocks Failures in Latches

We are motivated to examine latches because they bear a resemblance to memory circuits, and because -like memories- they are used ubiquitously across a chip. A latch also stores a single bit of information and uses feedback to do so, but its function is somewhat more complex because it serves as a *synchronization* element as well. Specifically, the latch has a *Data* input and a *Clock* input; and its function is to *capture* the data when the clock input is asserted. Due to inevitable internal delays in the latch circuit, any latch will require that the data be *ready* for some time before the clock is asserted; this interval of readiness is referred to as the *Setup Time*. Due to these same delays, the output of the latch, denoted usually by Q as shown in Figure 6, is not asserted immediately upon capture of the data but after a delay referred to as the *Clock to Q* delay. It is these delays that we will study next.

In a sequential digital circuit, latches are used to divide a large network of logic gates into smaller parts such that the delay of any one part is less than the clock period. Consider a typical part of a sequential circuit shown in Figure 7, consisting of a set of *Launch* latches which begin the path, a cloud of logic gates comprising the combinational circuit being evaluated, and a set of *Capture* latches at the end of the path.

Ensuring the correct timing of digital circuits is a well-established field and is accomplished via *Static Timing Analysis* (STA) [12]. STA evaluates the validity of a circuit by checking timing inequalities that ensure the proper sequencing of data and clock inputs. For the circuit in Figure 7 one such inequality would be:

$$T_{c2q,L} + T_{path} + T_{setup,C} > T_{clock} + T_{skew} \qquad (1)$$

$T_{c2q,L}$ is the clock-to-q delay of the launching latch, T_{path} the delay of the combinational logic path between the launch and the capture latch, and $T_{setup,C}$ the setup time of the capturing latch. The sum of these three quantities has to be larger than the clock cycle time T_{clock} plus the clock skew T_{skew} which denotes the difference in arrival time of the clock at the launch and capture latches. Recall that an ideal situation would be for the clock to arrive at all latches simultaneously (i.e. with zero skew), but in realistic designs, inevitable asymmetries and additional delay (represented in Figure 7 by some clock buffers) will cause a non-zero skew. Clock distribution and skew minimization is an area of active research, see [13].

Assuming that the launch and capture latches are identical, we can characterize $T_{latch} = T_{c2q,L} + T_{setup,C}$ as one single latch characteristic influencing the timing of the circuit. This quantity T_{latch} is clearly dependent on variability in the process parameters, and is thus a random variable itself. In this study, we examined the distribution of T_{latch} subject to the following process variations:

- Channel length variability,
- P- and N-Channel threshold voltage variability,
- P- and N-Channel carrier mobility variability.

We desire to find the *worst-case* value of T_{Latch} due to variability for a given yield Y, since such a value would be an integral part of the *specification* of the latch for future phases of the design. This is akin to finding one of the *fast* or *slow* curves in Figure 5, but since manufacturing variability is in multiple dimensions, such a worst-case value needs to be found by solving the following optimization problem [9]:

$$\max_{\mathbf{x}} T_{Latch}(\mathbf{x}) \quad \text{s.t. } (\mathbf{x}-\mathbf{x_0})^T \cdot \mathbf{C}^{-1} \cdot (\mathbf{x}-\mathbf{x_0}) \leq \beta_{WC}^2 \quad (2)$$

The parameter vector $\mathbf{x}$ denotes the values of all process parameters. Thus, the optimization problem finds the values of all process parameters x_{WC} at the worst case point of T_{Latch} under the given yield constraints. The amplitude of process variations is given by the covariance matrix $\mathbf{C}$ and the nominal design point by the parameter vector x_0. The whole optimization constraint defines a target yield Y under which the worst case value of T_{Latch} has to be found. The target yield Y is here expressed in standard deviations β_{WC} (see Equation 3), which is generally more convenient because it is somewhat easier to comprehend than a probability.

$$Y = \int_{-\infty}^{\beta_{WC}} \frac{1}{\sqrt{2\pi}} \exp(-\frac{t^2}{2}) dt \qquad (3)$$

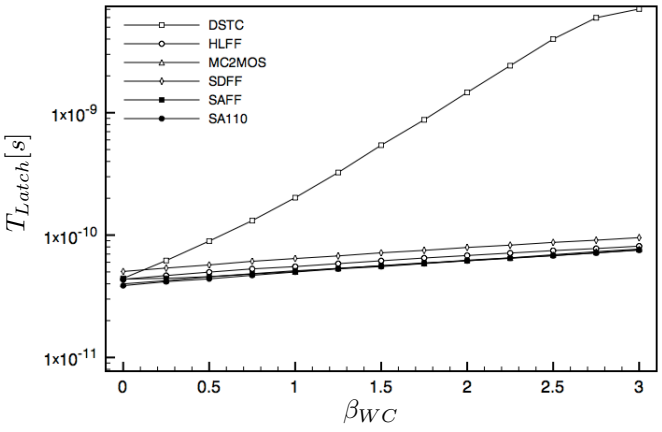

Fig. 8. T_{Latch} vs. failure sigma β_{WC} for a variety of latch types.

We can solve the optimization problem in Eq. 2 for different values of β_{WC} using sequential quadratic programming (SQP) [14] to obtain the worst-case performance value of T_{latch} as a function of β_{WC}. Such an analysis would allow us to trade off the performance of the latch against the overall expected yield of any circuit.

Using this method we compare different latch architectures in order to study their response to manufacturing variability (again using the 22nm Predictive Technology Model [8]). The relative sizings of the transistors in the latches are taken from [15]. Figure 8 shows the worst case value of the latch delay T_{Latch} as a function of β_{WC}. The plot shows that all latches degrade in performance, but one latch, DSTC, degrades much more rapidly (note the logarithmic y axis).

The slope of the curve in Figure 8 is an indication of the width of the area between the fast and slow lines in Figure 5. A higher slope means a wider band, and also means that smaller amounts of variability can result in a *Goldilocks* situation. Such a situation means in this context that some path in the chip will be so slow under this condition that it would fail even under the minimum clock frequency that we are willing to operate the chip under (i.e. the lower bound in Figure 5). Let us now assume that a latch can be considered to fail when its delay is one order of magnitude larger than the nominal delay. Using this assumption, we can see that the DSTC latch reaches this performance level with approximately 1.5 sigma of process variations, while other latches can tolerate more than twice that level and still remain operational.

Performing a comparison of alternative circuits like what we did in Figure 8 allows a designer to make decisions that can include the sensitivity of circuits to manufacturing variability, and thus reduce the probability of a *Goldilocks* failure. With manufacturing tolerances increasing with further scaling, such analysis will become more and more necessary.

V. Conclusions

Our purpose in this paper was to show the linkage between increasing levels of manufacturing-induced variability and circuit failures that are difficult to differentiate from traditional *hard* failures caused by defects that change circuit topology. This phenomenon has been observed in the SRAM area for some time now, but is at the threshold of becoming important for other types of circuits. This increasing variability is brought on by continued scaling and by the emergence of an increasing number of atomistic phenomena like random dopant fluctuations and line edge roughness. While there are some upcoming technology innovations that promise some relief, like FinFETs or thin-body SOI devices [16], [17], the overall trend is likely to continue through the end of the Silicon CMOS era.

Acknowledgment

This work was supported in parts by the German Research Foundation (DFG) as part of the priority program "Dependable Embedded Systems" (SPP 1500 - spp1500.itec.kit.edu).

References

[1] Wikipedia, *Wikipedia entry describing Rear Admiral Grace Hopper*, http://en.wikipedia.org/wiki/Grace_hopper. I

[2] L. Liebmann, "Layout impact of resolution enhancement techniques: impediment or opportunity?" in *International Symposium on Physical Design (ISPD)*, 2003. I

[3] A. Asenov, "Random dopant induced threshold voltage lowering and fluuctuationsin sub 0.1 micron MOSFETs: A 3D atomistic simulation study," *IEEE Trans. on Electron Devices*, vol. 45, no. 12, pp. 2505–2513, Dec. 1998. I

[4] Y. Ye, F. Liu, S. Nassif, and Y. Cao, "Statistical Modeling and Simulation of Threshold Variation Under Dopant Fluctuations and Line-Edge Roughness," in *Design Automation Conference (DAC)*, 2008. I

[5] J. Hennessy and D. Patterson, *Computer architecture: a quantitative approach*. Morgan Kaufmann Pub, 2011. II

[6] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *Design Automation Conference (DAC)*, 2006. II

[7] L. W. Nagel, "Spice2: A computer program to simulate semiconductor circuits," Ph.D. dissertation, University of California, Berkeley, 1975. II

[8] "Predictive technology model (ptm)," available at http://www.eas.asu.edu/~ptm. II, IV

[9] H. Graeb, *Analog Design Centering and Sizing*. Springer, 2007. III, IV

[10] S. Nassif, N. Mehta, and K. Cao, "A Resilience Roadmap," in *Design Automation & Test in Europe (DATE)*, 2010. III

[11] N. Carter, "Design Techniques for Cross-Layer Resilience," in *Design Automation & Test in Europe (DATE)*, 2010. III

[12] R. B. Hitchcock, "Timing verification and the timing analysis problem," in *Design Automation Conference (DAC)*, 1982. IV

[13] P. Restle, T. McNamara, D. Webber, P. Camporese, K. Eng, K. Jenkins, D. Allen, M. Rohn, M. Quaranta, D. Boerstler, *et al.*, "A clock distribution network for microprocessors," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 792–799, 2001. IV

[14] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*. Cambridge University Press, 1986. IV

[15] V. Stojanovic and V. G. Oklobdzija, "Comparative Analysis of Master-Slave Latches and Flip-Flops for High-Performance and Low-Power Systems," *IEEE J. Solid-State Circuits*, vol. 34, no. 4, pp. 536–548, Apr. 1999. IV

[16] B. Cheng, S. Roy, A. Brown, C. Millar, and A. Asenov, "Evaluation of statistical variability in 32 and 22 nm technology generation LSTP MOSFETs," *Solid-State Electronics*, vol. 53, no. 7, pp. 767–772, 2009. V

[17] F. Andrieu, O. Weber, J. Mazurier, O. Thomas, J. Noel, C. Fenouillet-Béranger, J. Mazellier, P. Perreau, T. Poiroux, Y. Morand, *et al.*, "Low leakage and low variability Ultra-Thin Body and Buried Oxide (UT2B) SOI technology for 20nm low power CMOS and beyond," in *Symposium on VLSI Technology (VLSIT)*, 2010, pp. 57–58. V