

System Level Self-Healing for Parametric Yield and Reliability Improvement under Power Bound

S. Narasimhan, S. Paul, R.S. Chakraborty, F. Wolff, C. Papachristou, D. J. Weyer*, and S. Bhunia
 Dept. of EECS, Case Western Reserve U., *Rockwell Automation, Cleveland, OH, USA
 E-mail: sexn124@case.edu

Abstract

Post-silicon process compensation or “healing” of integrated circuits (ICs) has emerged as an effective approach to improve yield and reliability under parameter variations. In a System-on-Chip (SoC) comprising of multiple cores, different cores can experience different process shift due to local within-die variations. Furthermore, the cores are likely to have different sensitivities with respect to system power dissipation and system output parameters such as quality of service or throughput. Post-silicon healing has been addressed earlier at core level using various compensation approaches. In this paper, we present a system level healing algorithm for compensating SoC chips for a specific output parameter under power constraint. We formulate the healing problem as an ordinal optimization problem, where individual cores need to be assigned the right amount of healing that satisfies the target system performance and power requirement. Next, we propose an efficient solution to the problem using a priori design-time information about the relative sensitivities of the cores to system performance and power. Simulation results for example systems show that the proposed healing approach can achieve higher parametric yield and better settling time compared to conventional healing approaches.

1. Introduction

Advances in integrated circuit technology today allow the integration of multiple analog and digital functional units on the same chip, resulting in a complex mixed-signal System-on-a-Chip (SoC). However, aggressive technology scaling has resulted in increasing inter-die and intra-die process variations [1]. Variability in device parameters is directly reflected in the measurable circuit performance such as power, maximum operating frequency (F_{max}) etc. For complex SoCs, such as a Wideband mm-Wave Transceiver or a Video Compression Unit, such variations may indirectly affect performance metrics representing Quality of Service (QoS) e.g. phase noise and output Peak Signal to Noise Ratio (PSNR), respectively [2]. Apart from process-

induced variations, environmental stress and aging effects, such as Bias Temperature Instability (BTI) can severely degrade the reliability of operation for these SoCs. In such scenarios, healing for parametric shift during manufacturing test as well as during normal operation can be extremely effective for improving yield and reliability.

Traditional worst corner-based design techniques require coverage of an extensive parameter space and typically result in pessimistic designs with high area/power overhead. In this regard, large SoCs are particularly vulnerable due to the complex integration of numerous individual sub-blocks that can vary greatly in performance and power characteristics. Static design time corrective measures, such as statistical design and Design for Yield (DFY) techniques suffer from limitation in terms of amount of parametric shift that may be tolerated during fabrication under a power budget.

An alternative approach is to incorporate appropriate healing mechanism in the SoC design that can be used post-fabrication for maintaining target performance under power envelope. Such a healing mechanism would require sensing the parametric shift due to process, environment or device degradations and compensating for it using appropriate repair approach. Post-silicon repair of memory using redundant rows/columns is a classical example of such in-built healing systems. Several other post-silicon healing techniques that tolerate parameter variations are described in [3-7], [12]. These approaches either incorporate built-in redundancy in the system or modify the supply voltage, bias voltage or operating frequency for the system. However, most of these explorations target healing a single core such as a microprocessor [4, 5], signal processing module, embedded memory [6, 7] or RF circuits [12], while efficient self-healing approach in large complex SoCs comprising of multiple heterogeneous cores remains largely unexplored. Some preliminary work is presented in [10-11]. In [10], an autonomic SoC framework has been presented which considers fault-tolerance as an additional design parameter along with area, performance and power. A self-healing strategy for on-line testing and healing of SoCs is proposed in [11], where reconfigurable cores on an FPGA are used to replace the faulty cores.

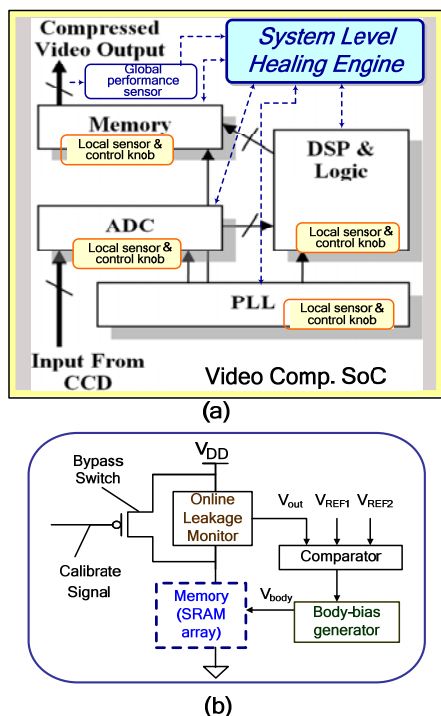


Figure 1. (a) Block diagram showing the main components of an example mixed-signal self-healing SoC used for image compression. (b) An example local sensor and control knob for the memory core.

The challenge of self healing in SoCs is complicated by the varying amounts of within-die variations in different components of an SoC and their varying sensitivities to system performance and power. Though the existing healing mechanisms for analog and digital systems may be applied to each of the cores, such adaptation typically aggravates the power overhead. Under a system power constraint, it is therefore important to define an efficient healing algorithm for complex multi-core SoCs that determine the best healing step for each core given the distribution of variation across cores. The algorithm should optimize the parametric yield while keeping the settling time as small as possible. We note that an exhaustive search through the solution space can often be expensive in terms of settling time, power and hardware resource for on-chip implementation.

In this paper, we propose an efficient heuristic-based algorithm for application of healing techniques to the components of an SoC. The proposed heuristic is based upon the concept of ordinal optimization that uses *a priori* design time knowledge about the system to decide the right amount of healing to be applied to individual cores. The algorithm takes into consideration both (a) the relative sensitivity of each unit to the overall performance of the system, and (b) the sensitivity in terms of power overhead associated with healing of the individual cores.

A global search through the solution space followed by a local search based on design-time information improves the quality of solution as well as the rate of convergence which translates to minimal settling time. Simulation results with example systems show significant improvement in yield under power bound using the proposed approach.

The rest of the paper is organized as follows. Section 2 provides a motivational example for the self-healing problem in SoC. The problem formulation and the proposed approach for solving it are described in Section 3. The simulation results for an example SoC consisting of ISCAS'85 benchmark circuits are given in Section 4. We conclude in Section 5.

2. Motivational Example

Let us consider the example of a video compression SoC as shown in Fig. 1(a). It is a mixed signal SoC where each component (ADC, RAM and DSP) is vulnerable to process variation. Fig. 1(b) shows how an online leakage monitor can sense the process variation in memory core, which is compensated by application of appropriate body-bias [6] to this core. In order to heal the whole system, the variability for each unit has to be individually sensed and appropriate correction applied. However, as we demonstrate below, an appreciable improvement in the system performance can be achieved through judicious application of the correction factor to the individual cores. The system performance is quantified in terms of the PSNR of the output image. Process variation can lead to (i) nonlinearity of the front-end ADC, (ii) increase in the minimum cycle time for the DSP unit, and (iii) read/write failure in the embedded memory [3]. The cumulative effect of these errors can severely corrupt the output image quality, which is reflected in the degraded PSNR value. In order to observe the effect of process variation on the output PSNR, SPICE simulations were carried out on an example SoC consisting of an 8-bit Successive Approximation Register (SAR) ADC (Analog to Digital Converter) followed by a 2-D DCT (Discrete Cosine Transform) unit for a 70nm predictive technology model (PTM) [8]. As inter-die variations of 10% and 20% in transistor threshold voltage (V_{th}) were successively introduced into the system, bit flips were observed on the output of both ADC and the DCT units. An Inverse-DCT on the final output, implemented using MATLAB, produced a distorted image as shown in Table 1. From Table 1, we note that under large variation, the output PSNR is dominated by the PSNR at the ADC output.

In order to restore the ADC and the DCT outputs to their original values, we applied Forward Body Bias (FBB) to the two units. However, this comes with an associated power overhead which is listed in Table 2. Table 2 also lists the improved PSNR values for the final

Table 1. Degradation in output image quality under process variations and the effect of healing.




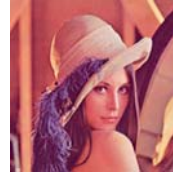



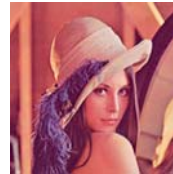
	Effect of process variation (V_{th})			After healing (for 10% variation)
	0%	10%	20%	
ADC o/p				
PSNR (dB)	∞ (no noise)	28.68	27.82	33.35
DCT o/p				
PSNR (dB)	∞ (no noise)	27.15	27.07	33.25

image. A process variation of 10% was considered for each of the units and the total power overhead as a result of FBB was constrained to 10%. We vary the bias voltage independently for each of the unit, thereby controlling the amount of healing applied to these units. The total power overhead of 10% is distributed into 2–8%, 5–5%, 8–2% and 10–0% among the ADC and the DCT blocks. The reason is to find an optimal combination of the power overhead for these blocks in order to achieve an acceptable PSNR. Although the output PSNR is largely determined by the healing applied to the ADC unit, it is to be noted from Table 2 that application of healing to only the ADC actually degrades the final PSNR to a value below 30dB. An 8–2 ratio actually achieves the highest PSNR improvement (output image given in last column of Table 1). It is evident from the above experiments:

- Process variations adversely affect the different components in an SoC and degrade the system performance. Healing can only be achieved by controlling the suitable parameters for each of these components.
- The order in which these components may be healed is determined by the sensitivity of the final output to the performance of these individual units.

Based on these observations, we formally define the problem for system level healing and introduce two algorithms for efficient estimation of the healing control in the next section.

Table 2: Improvement in output PSNR with the application of forward body biasing.

Power Overhead in ADC-DCT	2-8%	5-5%	8-2%	10-0%
ADC o/p PSNR (dB)	28.9	30.61	33.35	34.53
DCT o/p PSNR (dB)	28.7	30.53	33.25	29.23

3. System Level Healing Algorithm

3.1. Problem Formulation

The process of healing aims to restore the post-fabrication performance yield, which can be defined as the number of dies per wafer that meet all predefined performance metrics divided by the total number of testable dies per wafer. The performance metric can be as simple as the maximum operating frequency for each SoC component or as complex as the image Peak Signal to Noise Ratio (PSNR) for a video compression micro system built as an SoC. Healing on silicon necessitates an on-chip or on-board control circuitry to monitor the system performance and apply correction factors to system level parameters, like SoC supply voltage or substrate bias voltages of constituent blocks, in order to achieve a better performance yield. However, all these healing mechanisms come with an extra die area and power overhead.

Let us assume the SoC consists of N individual blocks which are designed to meet the performance metric of $f > f_T$ under an energy constraint of $E < E_T$. Due to process variations, the i^{th} IC has moved to a process point P_i denoted by $\langle p_{i,1}, p_{i,2}, p_{i,3}, \dots, p_{i,N} \rangle$, where $p_{i,j}$ is the post-manufacturing process point of the j^{th} block of the i^{th} chip instance. Even though the process variation can assume continuous values, let us assume for the sake of estimating the order of complexity of the problem that each block can move to n_p different process points. For a system with N blocks, the total number of system states due to random process variation in each block will be of the order of n_p^N . Assuming a maximum of $\pm 20\%$ variation in V_{th} at discrete steps of 0.5%, the number of steps is $n_p = 80$. For $N = 20$ blocks, the number of states becomes $80^{20} \sim 10^{38}$. If the chip, at any one of these states, fails to

meet the objective frequency metric, it will lead to a loss in parametric yield.

The task of achieving the desired performance yield under process variation, environmental stress or aging can be formulated as a linear programming problem (LPP), which can be mathematically stated as:

$$\text{Maximize } \sum_{j=1}^N a_j x_j \text{ subject to } \sum_{j=1}^N b_j x_j \leq c, \quad (1)$$

where x_j is the amount of healing applied to the j^{th} block and $j = 1, 2, \dots, N$. In the current context, a_j denotes the improvement in system performance due to the healing in unit j . b_j is the corresponding increment in the overall power requirement for the system. If we consider the application of adaptive body biasing (ABB) as a mechanism for self-healing then the goal of the control algorithm is to search for the best bias vector.

In practice, the healing control variables can only assume a set of discrete values within a given range. Hence, this problem belongs to a class of problems which have been considered recently in the context of optimization of Discrete Event Dynamic Systems (DEDS). Essentially, the approach to solve these problems is to rely on simulations in reaching the optimal or acceptable solution. However, often, the computational effort and resources necessary to arrive at a solution following a simulation-based approach are prohibitive to make it useful in a real scenario [9]. The solution to the

problem should satisfy the following objectives: (a) the control system should be stable; (b) it should have low settling time; and (c) it should be implemented in hardware efficiently with low power overhead. In this paper, we propose two approaches for designing the searching algorithm which can converge to an acceptable solution in a short time.

3.2. Greedy Heuristic-Based Algorithm

The first approach consists of a greedy heuristic based algorithm. A simple controller would search the entire space consisting of all possible combinations of bias voltages and ultimately arrive at the best assignment of bias voltages to individual blocks. But as the number of blocks (N) increases, the total time taken by the ‘‘Simple Search’’ to reach an optimal point becomes inordinately large. Hence we need to make some intelligent decisions based on *a priori* design-time knowledge, which can simplify the search procedure.

Based on the fact that the overall performance of the system is more sensitive to the performance of one or more blocks compared to the rest, we propose the following steps for our heuristic:

Step1: Determine output and power sensitivities (S_i^o, S_i^p , respectively) of each block i .

Step2: Rank the blocks based on a weighted metric of both sensitivities: $W_i = w_i^o * S_i^o + w_i^p * S_i^p$

Step3: Apply healing sequentially to each block in order of their rank.

Step4: Terminate the search as soon as the target specification is met or the constraint is violated.

This search procedure, also depicted in Fig. 2, might not lead to the globally optimal solution, but we can easily and rapidly derive an acceptable solution for compensation.

3.3. Convergence Rate and Proof of Convergence

Let the healing bias vector to be applied to the i^{th} chip be $B_i = \langle b_{i,1}, b_{i,2}, b_{i,3}, \dots, b_{i,N} \rangle$. For a simple search procedure, where the bias vector of each core is swept from 0–b at n_b discrete steps, the maximum number of search steps is of the order of n_b^N , which can be prohibitively large even for small number of steps n_b . Of course, for each chip, we need not sweep through the entire search space before reaching an acceptable point which meets the performance and energy constraints. The sensitivity-based algorithm described above will always lead to an acceptable solution, since, in the worst case, it reduces down to the simple algorithm of searching the entire search space by looking into all possible valid values of the compensating vector. The maximum number of search steps will be $N * n_b$ where N is the number of

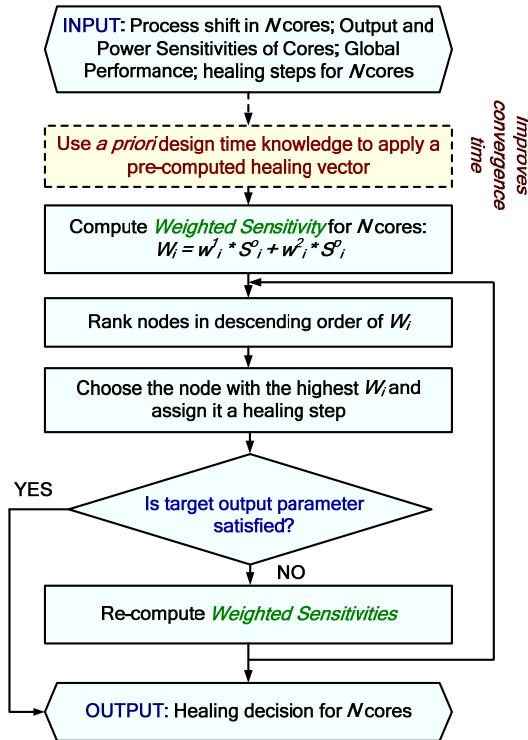


Figure 2. Flow-chart of the proposed heuristic based algorithm.

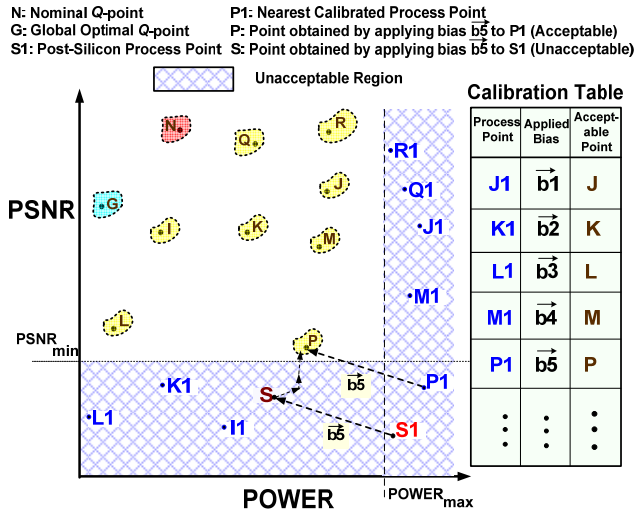


Figure 3. Illustration of the optimization problem.

blocks and n_b is the number of compensation levels. However, this case will arise only if all blocks have equal sensitivities with respect to the target output, which is a rare case when N is large.

3.4. Improving the Convergence Rate

We can improve the convergence rate of our heuristic-based search if we can estimate the values of the controlling vector. We propose a solution based on the concept of “ordinal optimization” [9], which uses *a priori* design time knowledge of the system to simplify the search process. The basic idea with the proposed healing is to apply a gradient-based search method to determine the optimal healing step for each block. We will determine a set of good initial points in the PSNR-power or F_{max} -power space at design time through simulations. During healing, once the system is calibrated and variations in individual cores are known, the system is moved to the nearest pre-defined good initial point. The nearest point is computed by using a distance metric between the sensed process corner and the calibrated (simulated) process points. This distance metric can be a simple Euclidean distance metric, which can be weighted in terms of the sensitivity of each block. Next, in the order of decreasing sensitivity of the cores, individual

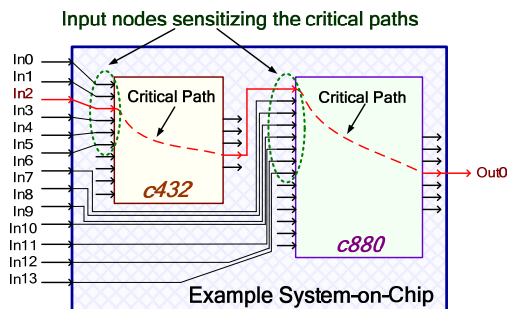


Figure 4. Block diagram of experimental setup.

cores are healed in discrete steps, following the previously described heuristic-based approach.

Ordinal optimization uses *a priori* knowledge of the nature of dependence to simplify the search process by concentrating on finding a good-enough solution, rather than the globally optimal one, thus reducing the number of iterations drastically. Fig. 3 shows an example of the application of the algorithm, where there are several acceptable Q-points. Suppose the system has shifted to a process corner different from the nominal process corner, effectively shifting the system from the nominal Q-point N to S1. At S1, neither the system power constraint is satisfied, nor is the target PSNR achieved. The algorithm finds a point P1 to be the closest process corner to S1, available in the calibration table. Assuming that applying the bias compensation (b_5), pre-computed for correcting P1, will lead us close to (if not, into) the acceptable region, we apply the same bias to S1. If the resulting Q-point S, is still not in the acceptable region, we can perform a local search in the neighborhood of S for the nearest optimal point, taking small steps and causing small changes in the power and PSNR values. After a few iterations, the search process reaches the neighborhood of point P and stops.

4. Results

4.1. Simulation Framework

To demonstrate the usefulness of the proposed algorithms in system level healing, we constructed a hypothetical SoC consisting of two ISCAS’85 benchmark circuits, *c432* and *c880* as shown in Fig. 4. A SPICE netlist of the SoC was prepared and simulated using 70nm PTM [8]. The process variation effect was modeled as a variation in the transistor threshold voltages of the two blocks. Simulations were performed for several random process corners, with and without application of the different bias voltage steps using the search algorithms. Choice of this simple SoC enables us to easily illustrate the convergence of the search algorithms. The proposed healing approach, however, can be easily applied to complex SoCs with large number of cores.

4.2. Simulation Results

The effect of process variations for the given SoC is shown in Fig. 5. With regard to the output performance of the entire system, we consider two specifications - F_{max} , the maximum operating frequency and E , the total energy consumption (considering a time period of 10ns). We use the following stopping criterion for the search - if $F > F_T$ and $E < E_T$, where F_T and E_T are the target maximum frequency of operation of the SoC and maximum allowed

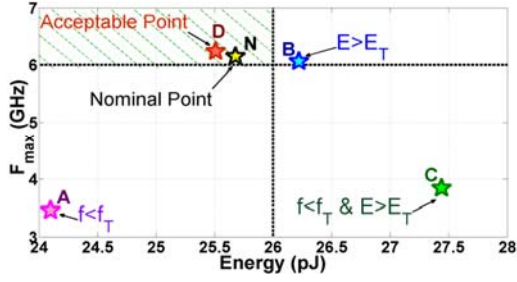


Figure 5. Four process corners and their impact on system performance.

energy consumption respectively, then we stop the search process as we have reached an acceptable region. This might not be close to the global optimum or the nominal design, but we can satisfy the target specifications, hence we stop.

To compare the convergence rates of the three algorithms, the steps of searching followed by each algorithm are shown in Fig. 6. The algorithm starts from two different process corners and searches for an acceptable solution. The nominal point is also shown in each subplot along with the post-Si process point and post-healing acceptable Q-point. The simple search algorithm requires more than 30 steps to reach an acceptable solution, as compared to 5 steps using the heuristic-based algorithm. The *c880* sub-circuit was observed to have a higher contribution to the F_{max} of the SoC as well as to the overall power consumption of the circuit. In the simple search algorithm, we follow an extensive search procedure where the body bias voltage of *c432* is swept from 0V to 0.3V in steps of 0.05V for each step of 0.05V in *c880*. Using our sensitivity-based metric, we initially apply steps of bias to *c880*, keeping the bias of *c432* fixed, and then sweep the bias of *c432*, if required.

After characterizing several random process corners with appropriate bias voltage to be applied for healing, we look at an arbitrary process corner, which has not been calibrated before. The calibrated process points along with their healing bias voltages are shown in Table 3. This *a priori* information can be used to speed-up the search process by using the ordinal optimization

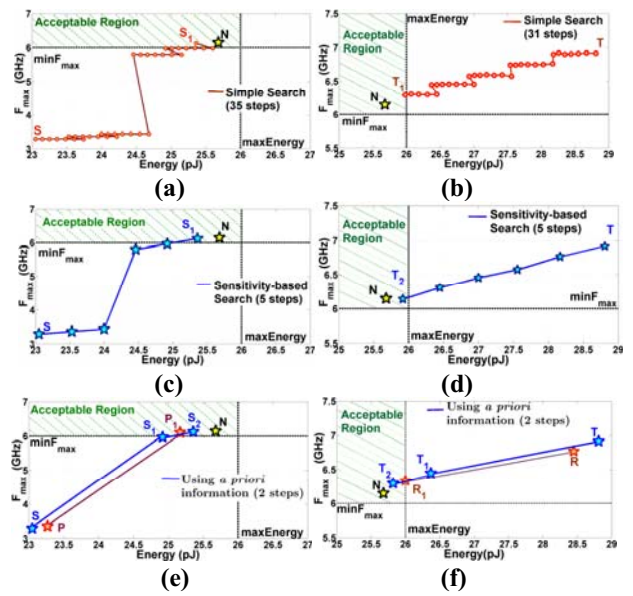


Figure 6. Simulation Results. (a, b): Simple search takes more than 30 steps to find an acceptable point starting from two different process corners S and T. (c, d): Sensitivity-based search takes less number of steps than the simple method. (e, f): The search using *a priori* information reaches near the acceptable region in one large step and then performs a local search (sensitivity-based) to reach an acceptable solution. N: Nominal Point, S and T are two different process corners non-calibrated) which are closest to P and R (calibrated), respectively.

procedure. We find the nearest calibrated process point and apply the same healing bias vector. For the point S in Fig. 6(e), the process variation is +7% and +19% for the *c432* and *c880* modules respectively, which is closest to point P in the calibration table. Similarly, point T in Fig. 6(f) has a process variation of (-9%, -19%) which makes it closest to point R. This takes us close to the acceptable region and a single step of sensitivity-based search leads us to the acceptable region, as shown in Fig. 6(e,f).

4.3. Effect of Healing on Parametric Yield

By adopting the proposed sensitivity-based SoC healing algorithm, we can increase the number of dies which pass the parametric binning post-manufacturing. If no healing algorithm is applied, parameter variations cause the system operating point shift to various points in the f_{max} -energy domain, causing many of the dies to fall beyond the performance and power bounds, as shown in Fig. 7 for 100,000 random die

Table 3: Calibration table for example SoC.

Index	Process Shift (% $\Delta V_{th,N}$)	Shifted Q-point	Bias Vector	Healed Q-point
	(<i>c432</i> , <i>c880</i>)	(Energy, F_{max})	[<i>c432</i> , <i>c880</i>]	(Energy, F_{max})
P	(14, 16)	(23.27, 3.35)	[0.00, 0.20]	(25.17, 6.10)
M	(12, 12)	(23.37, 3.30)	[0.00, 0.20]	(25.26, 6.11)
N	(16, 4)	(24.76, 5.97)	[0.00, 0.05]	(25.29, 6.14)
L	(-4, -14)	(27.83, 6.70)	[-0.05, -0.15]	(25.98, 6.25)
Q	(-20, -8)	(27.33, 6.39)	[-0.10, -0.10]	(25.88, 6.06)
R	(-12, -16)	(28.45, 6.77)	[-0.30, -0.15]	(26.00, 6.33)

Table 4: Parametric yield results for example SoC.

Process Variation (Inter-die/Intra-die)	Parametric Yield		
	No Healing	Conv. Healing	Proposed Healing
(20%, 15%)	80.88%	97.08%	99.97%
(25%, 20%)	69.66%	89.51%	99.44%
(30%, 20%)	63.12%	84.65%	98.56%
(30%, 25%)	60.42%	79.23%	97.43%

instances. The varying sensitivities of the blocks towards the overall maximum frequency and energy cause the scatter plot to take an elliptical shape. The shape will vary depending on the number of cores and their relative sensitivities. Following the proposed healing algorithm, we can heal many of the dies and bring them within the frequency and energy bounds (taken as 10% of the nominal values). This increases the parametric yield, as evident from Fig. 7. The yield values for different levels of process variation (3σ bound of Gaussian distribution centered at a mean threshold voltage of 190mV) are given in Table 4. As we can observe from Table 4, the proposed healing results in better yield compared to conventional healing which aims at bringing all cores to acceptable operating points without considering their varying impact on system performance and power.

5. Conclusion

We have presented a system level healing approach in SoC for power-constrained yield and reliability improvement. The objective of the proposed healing

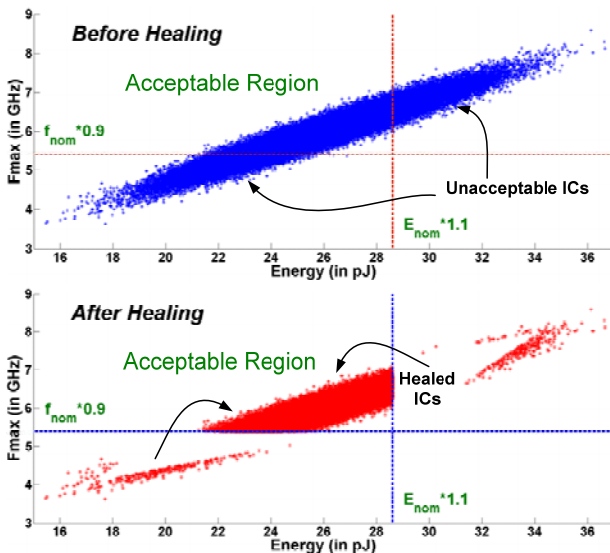


Figure 7. Scatter plot in F_{max} -Energy domain for 100,000 dies with 25% inter-die and 20% intra-die variation.

approach is to restore a SoC design shifted to an unacceptable operating point due to variations to an acceptable point with respect to target performance parameters (such as delay, PSNR) while maintaining power bound. The proposed healing approach is based on an efficient search for appropriate compensation values to constituent cores in an SoC considering their relative sensitivities in terms of system performance and power. Further, the algorithm ensures fast convergence to an acceptable point resulting in low settling time. Simulation results show that such an approach provides significant improvement in parametric yield under large within and die-to-die variations. Although in our simulation, we have used ABB as the repair mechanism, the proposed healing approach can be applied to other repair techniques (such as frequency or voltage scaling) as well as combination of multiple techniques. Future work will involve hardware implementation of the global healing algorithm and application to complex mixed-signal SoCs.

6. References

- [1] S. Borkar et al, "Parameter variations and impact on circuits and microarchitecture", *DAC*, pp. 338-342, June 2003.
- [2] Self-healing mixed-signal integrated circuits (HEALICS). Available [Online] <http://www.darpa.mil/MTO/Programs/healics/index.html>
- [3] N. Banerjee, G. Karakonstantis and K. Roy, "Process variation tolerant low power DCT architecture", *DATE*, 2007.
- [4] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: A new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation", *IEEE Transactions on Circuits and Systems*, vol. 26, no. 11, pp. 1947-1956, Nov 2007.
- [5] D. Ernst et al, "RAZOR: Circuit-level correction of timing errors for low-power operation", *IEEE MICRO*, vol. 24, no. 6, pp. 10-20, Nov 2004.
- [6] S. Mukhopadhyay, A. Agarwal, Q. Chen, and K. Roy, "SRAMs in scaled technologies under process variations: failure mechanisms, test and variation tolerant design," *IEEE CICC*, pp. 547-554, Sep 2006.
- [7] J. W. Tschanz et al, "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*, vol. 37, no. 11, pp. 1396-1402, Nov 2002.
- [8] Predictive technology model. [Online] <http://www.eas.asu.edu/~ptm/>
- [9] Y. Ho, "An explanation of ordinal optimization: Soft computing for hard problems", *Journal of Information Sciences*, vol. 113, pp. 169-192, 1999.
- [10] G. Lipsa et al, "Towards a framework and a design methodology for autonomic SoC", *International Conference on Autonomic Computing*, 2005.
- [11] S. K. Venishetti, A. Akoglu, R. Kalra, "Hierarchical built-in self-testing and FPGA based healing methodology for System-on-a-Chip", *AHS*, 2007.
- [12] A. Goyal, M. Swaminathan and A. Chatterjee, "A novel self-healing methodology for RF amplifier circuits based on oscillation principles", *DATE*, 2009.