Semnan University

Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)

Journal homepage: https://mseee.semnan.ac.ir/journal/about /

ISSN: 2821-0786

Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)

GENSO: A Genetic Algorithm-based Optimization Framework for Lifetime Reliability Enhancement in Sequential Circuit Design

Mohsen Raji[1•], Behnam Ghavami, Saeed Keshavarzi ana Reza Mahmoudi

***Abstract*-chnology scaling becomes increasingly aggressive, lifetime reliability has emerged as a critical challenge for modern digital circuits, exacerbated by manufacturing process variations and aging effects. This paper introduces GenSO, a Genetic algorithm-based multi-objective Sequential circuit Optimization framework designed to enhance the lifetime reliability of sequential circuits modeled as Finite State Machines (FSMs), while simultaneously addressing initial delay and power consumption. The framework leverages a cross-layer approach, utilizing a gate-level delay degradation model that accounts for process variations and aging to estimate circuit lifetime reliability. A novel metric, termed Guardband-Aware Reliability (GAR), is proposed to provide a fair assessment of FSM lifetime reliability in relation to the guardband and timing yield specified by the designer. A multi-objective genetic algorithm is then employed to optimize delay, power consumption, and lifetime reliability in FSM-based sequential circuits. Experimental results demonstrate that GenSO successfully identifies non-dominated solutions for sequential circuit designs, achieving simultaneous optimization of initial delay, power consumption, and lifetime reliability. With a 15% delay overhead for a 6-year lifetime and a 10% variation ratio, GenSO improves circuit reliability by an average of 64.34%, significantly outperforming state-of-the-art reliability optimization frameworks, which typically achieve less than 30% improvement in lifetime reliability.**

***Index Terms*—Finite State Machines, Lifetime Reliability, Multi-objective Optimization, Process Variations, BTI.**

I. INTRODUCTION

WITH ever-increasing downscaling of CMOS devices, aging techniques (such as Bias Temperature Instability (BTI)) have been manifested as an important reliability issue in nano-scale technologies. [1]-[4]. BTI causes the circuit delay to increase over the operation time which in turn, may result in timing constraints violation of digital circuits [5], [6]. This affects the correct functionality of the nano-scale digital circuits and systems which leads to timing failure of the system before its targeted lifetime. Hence, lifetime reliability has been added to the traditional design challenges of digital systems in nanometer technology nodes. On the other hand, fabrication-induced Process Variations (PV) have also emerged in nano-scale digital systems. PV lead to significant deviation of timing characteristics in a chip from its initial design; i.e. the experiments showed that, PV may cause up to 30% variability in timing characteristics and up to 20X variation in leakage power of digital circuits. Importantly, the interdependency between PV and BTI mechanisms motivates the designer to consider both of them during the analysis and optimization of nano-scale digital systems.

Basically, digital systems are composed of a control-path and a data-path; i.e. the control-path controls the sequence of the arithmetic and logical operations performed on the data-path. The control-path is implemented using sequential circuits which are modeled by Finite State Machine (FSM). Sequential circuit optimization modelled as FSM through efficient state assignment and encoding is a traditional problem in sequential circuit design.

There are some previous works on FSM state assignments, most of which focused on optimizing the circuit area and power [7]-[15]. In [7], some low power techniques are investigated in FSM synthesis methods. In [8], novel approaches to increase the fault resilience FSM encoding is proposed. The problem of algebraic synthesis of a FSM is

[1] School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran.

• Corresponding author:Email: mraji@shirazu.ac.ir. Mohsen.raji@gmail.com

Cite this article as:

Raji, M., Ghavami, B., Keshavarzi, S. and Mahmoudi, R., 2024. GENSO: A Genetic Algorithm-based Optimization Framework for Lifetime Reliability Enhancement in Sequential Circuit Design. Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE), 4(4), pp. 41-50.

https://doi.org/10.22075/MSEEE.2025.37658.1212

solved by matrix approach in [9]. In [10], transforming group codes in mealy finite state machines is introduced to optimize synthesized sequential circuits. In [11], a multi-objective genetic algorithm was proposed to find the optimal state assignment considering area and power consumption in sequential circuit designs. However, the reliability metric is ignored in this work. In [12], binary particle swarm optimization algorithm was proposed to optimize the sequential circuit area by solving the state assignment problem. However, this method focuses on optimizing circuit area. In [13], a majority-based evolution (MBE) Simulated Annealing (SA) algorithm was proposed for FSM state encoding to optimize both circuit area and power. However, this work ignores the impact of PV and BTI in the characteristics of the synthesized circuit. In [14], a low power FSM synthesis approach based on a Fuzzy c-mean clustering-based decomposition method is proposed. The proposed method partitioned a set of states of FSM into a collection of c-fuzzy clusters and decomposed FSM into a set of c-sub machines and de-activate some to reduce the power consumption. One limitation of this work is that it only focuses on optimizing power consumption. This works is also focused on A Multi-Population Genetic Algorithm (MPGA)-based state assignment method was proposed in [15] for FSM synthesis to optimize both dynamic and static power consumption. MPGA consists of a set of inner-GA as a local search for finding low-power state assignment and outer loops to optimize the parameters of inner-GAs. This works is also focused only on power consumption and ignores other properties.

Despite valuable researches in sequential circuit optimization through proper FSM state assignements, less attention has been paid to design challenges of nanometer technology nodes (i.e. PV and BTI-oriented reliability issues) in the optimization of sequential circuit designs. In [16], a simulated annealing (SA) based state assignment algorithm was proposed for minimization of aging-induced degradation in the synthesized circuit. High probability states are re-encoded to minimize Negative BTI (NBTI) effects because FSM operates in these states in majority of lifetime. There are some limitations with this work: 1) no statistical model is proposed for considering the joint effects of PV and BTI which may lead to large inaccuracy due to interdependency of PV and BTI [17] they ignore the delay degradation due to Positive BTI (PBTI) occurring in NMOS transistors which is getting more important in deeper nanometer technology nodes [18] the possibility of trapping in local optimum of SA algorithm [19] which is used for FSM optimization.

In this paper, a Genetic algorithm-based multi-objective Sequential circuit Optimization framework called GenSO is presented considering the impacts of process variation and aging. The goal of GenSO is to minimize the decline in lifetime reliability, power consumption and the initial delay of sequential circuits considering the joint effect of PV and BTI. Considering the FSM model of a given sequential circuit, GenSO takes advantage of a multi-objective genetic algorithm (GA) to find the optimal state assignment for the FSM in the objective space of lifetime reliability, power consumption, and initial delay. The impacts of PV and BTI on the gate delay is modeled using a statistical model while the lifetime reliability of the circuit is computed using a metric called Guardband-Aware Reliability (abbreviated as GAR). Using an adaptive multi-objective ranking approach within GA, GenSO performs a wide design space exploration to optimize the antagonistic objectives simultaneously, instead of converting all objective functions into one based on a weighted sum approach, as is often carried out. The output of GenSO is a collection of non-dominated solutions which provides a wider picture of design space enabling the designer to pick out the best solution according to their design considerations. Experimental results show that, GenSO can find the non-dominated solutions for sequential circuit design whose initial delay, power consumption, and lifetime reliability are simultaneously optimized. In terms of reliability improvement, the results show that, GenSO averagely increases the reliability more than 69% compared to about 29% reliability improvement achieved by the only sequential circuit optimization technique. GenSO helps sequential circuit designers in making wise trade-offs in their design decisions and avoid suboptimal solutions. By imposing 15% delay overhead for 6-year life time and also 10% variation ratio, GenSO, on average, outperforms reliability of the circuit by 64.34% comparing to the state-of-the-art reliability optimization framework for sequential circuits which achieves less than 30% improvement in lifetime reliability.

Briefly, the main contributions of this paper is as follows:

1) To the best of our knowledge, GenSO is the first methodology for multi-objective optimization of sequential circuits in the objective space of lifetime reliability, power consumption, and initial delay,
2) It uses a metric called Guardband-Aware Reliability (abbreviated as GAR) to evaluate the lifetime reliability of circuits considering a guardband and timing yield specified by the designer,
3) GenSO takes advantage of an adaptive multi-objective ranking approach within GA to achieve the set of non-dominated solutions instead of converting all objective functions into one based on a weighted sum approach, its optimization process is accelerated using GPGPU to provide a faster statistical sequential circuit optimization during the sequential circuit synthesis.

The rest of this paper is organized as follows. Section II brings some backgrounds in the area of reliability concepts around the digital circuits, the power model and the statistical model for BTI an aging-aware statistical gate delay. Section III describes Guardband-aware Reliability (abbreviated as GAR. In section IV, GenSO framework is described in addition to the parallel computation of proposed framework. Section V presents the experimental results and finally, section VI concludes the paper.

II. Background

Some necessary basic information in the area of statistical static timing analysis and the aging effects on transistor delays are presented in this section.

A. Statistical Model for BTI

A common aging mechanism in nanoscale technologies is BTI [2]. Two types of BTI effects exist: Negative BTI (NBTI) and Positive BTI (PBTI), which affect PMOS transistor under negative and positive gate-to-source bias, respectively. Although NBTI was traditionally considered as the major BTI reliability issue, PBTI has recently introduced as another important problem due to the high-k metal gate dielectric used in deeper nanoscale technologies [20]. These mechanisms lead to the gradual increase in transistor threshold voltages where mainly depends of the time percentage for which the device is at stress, named as *stress probability (SP)* [5]. In order to calculate BTI-induced Vth degradation, Eq. (1) is proposed in [21]-[24], as follows:

$$\Delta V_{th,BTI} = K.t_{ox}.\sqrt{C_{ox}.(V_{GS} - V_{th0})}.e^{\frac{E_{ox}}{E_0}}.e^{\frac{-Ea}{k.T}}.a^n.t^n \qquad (1)$$

where Table I shows the definitions of the parameters.

TABLE I
The Definition of the Parameters in Eq. (1)

Parameter	Definition
K	A technology-dependent fitted constant
n	Time exponent
t_{0x}	Gate oxide thickness
E_{0x}	Vertical electric field
T	Temperature
k	Boltzmann constant
C_{ox}	Oxide capacitance per unit of area
V_th0	Initial threshold voltage value
E_a	Constant
E_0	Constant
a	Stress probability

In nano-scale integrated circuits, the presence of PV makes V_{th0} to become a random variable. A first-order Taylor approximation of Eq. (2) can be used to show the effect of PV in the long-term degradation of Vth [24]:

$$\Delta V_{th,BTI} = A.(1 - \gamma.\Delta V_{th.PV}).a^n.t^n \qquad (2)$$

where $\Delta V_{th.PV}$ shows the degradation in V_{th0} caused by PV, and A and γ are fitting parameter. Then, the total *Vth* variation of a transistor m can be obtained by summing the contributions due to BTI ($\Delta V_{th,BTI}$) and PV ($\Delta V_{th.PV}$), as given by Eq. (3) [24][25]:

$$\Delta V_{th,m} = A_m.a_m^n.t^n \\ +(1 - \gamma.A_m.a_m^n.t^n).\beta_k.\sum_i \Delta V_{th(i)} \qquad (3)$$

It is notable that, at the beginning (t = *0*), only PV causes the total variation in Vth while, as circuit ages, both the mean value and the variance of Vth is also affected by BTI mechanisms [26].

B. Statistical Aging-Aware Gate Delay Model

In order to statistically analyze the delay of the combinational circuit, the gate delay is modeled as a linear function of random variables with normal distribution describing the process parameters [25], as given in Eq. (4):

$$D_k = D_{nom(k)} + B_k.a_k^n.t^n + \\ (1 - \gamma.A.a_k^n.t^n).\beta_k.\sum_i \Delta V_{th(i)} \qquad (4)$$

where $D_{nom(k)}$ indicates the nominal gate delay, B_k is a fitting constant corresponding to the effects of Vth increase on the gate delay under nominal condition induced by BTI, and β_k is a fitting coefficient which reflects the gate delay change caused by PV-originated Vth shift without the BTI effect.

Note that Vth is consisted of two variation components; i.e. a time-zero variability component and runtime BTI originated one (See Eq. (3)). Since computational complexity remains low and the error that propagated by discarding high order terms can be ignored, this linear model is sufficient for small enough variations [27].

The PV & BTI-aware statistical gate delay can be integrated into a statistical static timing analysis tool as follows: accurate HSPICE electrical simulations is used to obtain the parameters in Eq. (4)Then, HSPICE simulations are run for each gate type at a comprehensive design conditions including different input transition time, gate sizes, load capacitances, and the operating temperatures. The nominal gate delay and sensitivities of the delay to parameters are computed. Finally, the fitting parameters of Eq. (4) are obtained using polynomials.

C. Power Model

Dynamic power is classically modeled as given in Eq. (5):

$$P_{dyn} = \alpha \times C_{tot} \times V_{dd}^2 \times .f \qquad (5)$$

where α is the average switching activity, C_{tot} shows the total capacitance, and f indicates the clock frequency.

Eq. (6) shows how the leakage power is modeled [28]:

$$P_{leak} = e^{a_0 + a_1.L + a_2.W + a_3.V_{th}} \qquad (6)$$

The total power of logic gate is computed using Eq. (7):

$$P_{tot} = P_{dyn} + P_{leak} \qquad (7)$$

D. Guardband Aware Lifetime Reliability (GAR) Metric

In order to evaluate the lifetime reliability of the circuits, our previously proposed metric called Guardband Aware Reliability (GAR) metric was used [25].

In order to have an operational combinational circuit at time 0, the value of the Critical Path (CP) delay should be less than a parameter, called timing constraint (τ). However, due to the process variations, the CP delay is modeled by a normally distributed random variable [27]. The PV-aware timing constraint for combinational circuits is traditionally considered based on the concept of p-percentile point of CP delay Cumulative Density Function (CDF) (Fig. 1.).

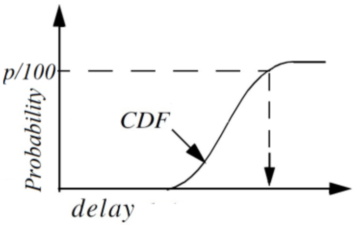


Fig. 1. p-percentile point of CP delay

In high reliable circuits, a common practice is to consider a guardband for the CP delay (i.e. p-percentile point value of CP CDF) to guarantee the given reliability. Hence, for a high reliable circuit design, τ is obtained as shown in Eq. (8):

$$\tau = (1+g) \times \varphi_0^{-1}(p/100) \qquad (8)$$

where g indicates the guardband value ($0 \leq g \leq 1$) and φ_0^{-1} is the inverse CDF of CP delay distribution. So, the GAR value for the lifetime reliability of the combinational circuit at a given time t with a given guardband of g for the p-percentile point value of CP CDF can be computed as given in Eq. (9) [25]:

$$\mathcal{R}_g^p(t) = \frac{\varphi_t\big((1+g) \times \varphi_0^{-1}(p/100)\big)}{\varphi_0\big((1+g) \times \varphi_0^{-1}(p/100)\big)} \qquad (9)$$

where, φ_t and φ_0 show the CDF of CP delay distribution at time t and 0, respectively. More details on GAR metric can be found [25].

Eq. (10) shows how to compute the GAR degradation considering a given guardband g at time t for a specific p-percentile point value ($\Delta\mathcal{R}_g^p(t)$):

$$\Delta\mathcal{R}_g^p(t) = \frac{\mathcal{R}_g^p(t) - \mathcal{R}_g^p(0)}{\mathcal{R}_g^p(0)} \qquad (10)$$

III. GenSO Framework

In this section, we present the GenSO framework, including the details of the GA implementation. First, we provide a motivational example to show the impacts of different FSM state encodings on the lifetime reliability of the synthesized sequential circuits. Then, we provide the overview of the GenSO framework. The details of the multi-objective GA-based optimization engine used in GenSO is also presented.

A. Motivational Example

A run time BTI optimization technique for finite state machines is proposed. FSMs operate in multiple states at different points of time. Output of state registers drive a number of MOSFET transistors in a circuit, some of which might be on the critical path. The number of transistors on critical paths subject to stress varies based on the encoded state representation. So, using different state coding may reduce GAR degradation. Here we provide an example. Table II shows train4.kiss2 benchmark.

TABLE II
Train4.Kiss2 Benchmark

Input	Current state	Next state	Output
00	ST0	ST0	0
10	ST0	ST1	-
01	ST0	ST1	-
10	ST1	ST1	1
01	ST1	ST1	1
00	ST1	ST2	1
11	ST1	ST2	1
00	ST2	ST2	1
11	ST2	ST2	1
01	ST2	ST3	1
10	ST2	ST3	1
10	ST3	ST3	1
01	ST3	ST3	1
00	ST3	ST0	-

We then use two different coding for synthesis as Table III.

TABLE III
Different State Encoding Assigned to Kiss4 Benchmark

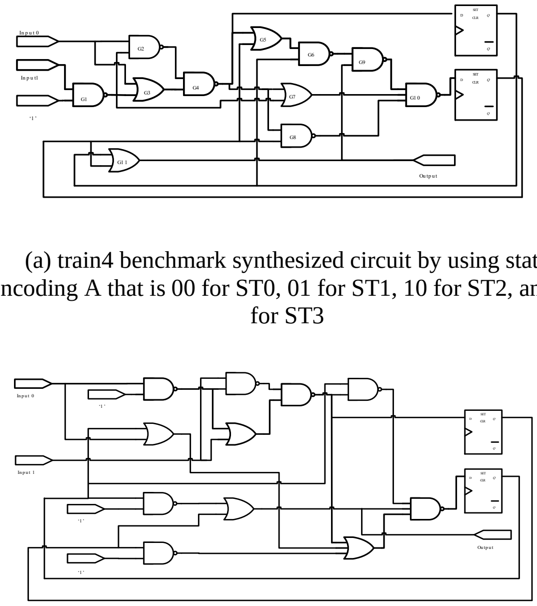
	ST0	ST1	ST2	ST3
A	00	01	10	11
B	10	01	11	00

Fig. 2.(a) and Fig. 2.(b) show synthesized circuit by using state encoding A and B according to Table III, respectively. Synthesized circuit's critical path is highlighted. For accurate analysis, we simulate both circuit in HSPICE simulation for 9 years lifetime. Table IV shows initial delay (ns) (using Eq. (4)), delay degradation (%), and average power consumption (using Eq. (5)-(7)) of these two different encodings. In this example, state encoding B has better delay, delay degradation during lifetime and power than state encoding A. So, different state encoding can result in different reliability, delay and power.

TABLE IV
Initial delay (ns), Delay Degradation (%) and Average Power Consumption (nW) of Different State Encodings in Table III

State Encoding	Initial Delay (ns)	Delay Degradation (%)	Average Power (nW)
A	64.9	42	376
B	40.2	34	279

On the other hand, the re-encoding of states changes hamming distance during the state transitions. This in turn can incur power overhead. So, in this work, we introduce multiobjective GA.



(a) train4 benchmark synthesized circuit by using state encoding A that is 00 for ST0, 01 for ST1, 10 for ST2, and 11 for ST3

(b) train4 benchmark synthesized circuit by using state encoding B that is 10 for ST0, 01for ST1, 11 for ST2, and 00 for ST3

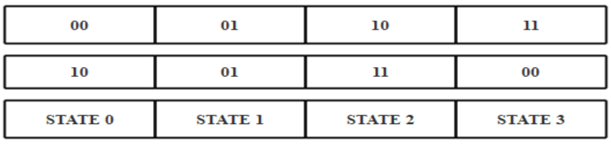
Fig. 2. train4.kiss2 benchmark synthesized circuit by different state encoding

B. Overall Overview of GenSO

GenSO is on the basis of a multi-objective GA. In GA, the solutions are modeled as an array of binary values called as chromosomes [29]. GA starts by generating an initial solution pool (step ❶). In GA, each solution is known as individual in a population and represented by a chromosome, or an array of binary values. For each population, GA evaluates the individuals based on defined fitness functions (step ❷) and then finds the optimized solutions up-to this generation using a non-dominated sorting approach (step ❸). Then, GA selects two chromosomes as the parents (step ❹) in order to reproduce chromosomes for the next generation. Then, the genetic operations (i.e. crossover and mutation) are applied to the chromosomes (step ❺). Crossover refers to partial exchange of genes so that the children chromosomes may inherit good features from both parents. In the mutation, some random alterations may happen in the genes of parent chromosome with a certain probability, known as the mutation rate. This evolution cycle may iterate for some generation to reach to a halting condition to end the GA; i.e. the algorithm will be stopped when the number of generations reaches to a pre-specified value. In the following, we explain the details of each step in the GenSO.

1) Step ❶: Chromosome representation and Initial Population

In GenSO, each child shows a possible state encoding for the FSM of the sequential circuit; i.e. genes in child shows a sequence of bits which represents current state encoding of the FSM. For an FSM which has n states, a chromosome includes n number of genes where each gene is composed of $\lceil log_2 n \rceil$ bits. Fig. 3. shows a state encoding for a sequential circuit with 2 state flip-flops (i.e. a FSM with 4 states).



00	01	10	11
10	01	11	00
STATE 0	STATE 1	STATE 2	STATE 3

Fig. 3. Chromosome structure for GenSO

The first population is generated with a number of preliminary solutions randomly.

2) Step ❷: Fitness Function Evaluation

In this paper, a multiobjective GA is used to make the opportunity for designer to choose the solution with the preferred tradeoff among various metrics. GenSO considers three fitness functions; i.e. initial delay (briefly referred as ‘delay’ in the rest of paper), power consumption, and aging-induced reliability degradation (GAR degradation based on Eq. (18)). So, the fitness functions of each child is defined as follows;

$$F(D_i) = \frac{Initial\ Delay(Child_i)}{\underset{i=1:NoP}{MAX}\left(Initial\ Delay\ (Child_i)\right)} \tag{11}$$

and

$$F(P_i) = \frac{Power(Child_i)}{\underset{i=1:NoP}{MAX}\left(Power\ (Child_i)\right)} \tag{12}$$

and

$$F(GAR_i) = \frac{GAR\ (Child_i)}{\underset{i=1:NoP}{MAX}\left(GAR\ (Child_i)\right)} \tag{13}$$

where $F(D_i)$, $F(P_i)$, and $F(GAR_i)$ respectively represent the fitness functions used for the delay, power consumption, and guardband-aware reliability degradation of the combinational circuit corresponding to the state encoding of $Child_i$. $Initial\ Delay(Child_i)$, $Power(Child_i)$, $GAR\ (Child_i)$ respectively show the delay, power consumption, and guardband-aware reliability degradation of the combinational circuit of the corresponding state encoding. $\underset{i=1:NoP}{MAX}\left(Initial\ Delay\ (Child_i)\right)$, $\underset{i=1:NoP}{MAX}\left(Power\ (Child_i)\right)$, and $\underset{i=1:NoP}{MAX}\left(GAR\ (Child_i)\right)$ are the result of applying the statistical maximum operation on the statistical distribution of the delay (using Eq. (4)), power consumption (using Eq. (5)-(7)), and guardband-aware reliability (using Eq. (8)-(10)) of all combinational circuits of the state encodings of the current population, respectively. Note that, the distributions of the parameters are evaluated by statistical analysis methods and then, p-percentile point values are used to find the fitness function value.

These fitness functions are evaluated individually and then, ranked in a 3-D space through a non-domianted sorting approach explained as following.

3) Step ❸: Non-dominated Sorting (Pareto Ranking)

In multiobjective optimizations, the goal is to optimize more than one metrics. In such cases, non-dominated sorting is used; i.e. a solution dominates another if it is not worse in any metric and is better in at least one. Based on this sorting, each solution will receive a rank called the Pareto rank, defined as the number of other solutions that do not dominate that solution.

In GenSO, we are interested in optimizing the delay, power consumption, and lifetime reliability of the sequential circuits. The Pareto rank of the solutions are computed based on the values of these metrics obtained from the defined fitness functions.

4) Step ❹: Parent selection

After finding the Pareto ranking of the solutions (individuals), a certain number of the lowest-rank solutions should be removed and new solutions (offsprings) will be reproduced from selected parents. In order to avoid local optimum at the beginning cycles of GA, GenSO takes advantage a variable called *equity* for parent selection step. A higher equity means that an individual with a lower rank has still high chance of being selected as the parent. As evolution cycles progresses, equity reduces by a ratio called *edr*. The equity variable helps GenSO to converge faster as it becomes gradually greedier.

Algorithm 1 shows the selection and reproduction of GenSO. The variables *r1*,*r2* represent a random number between 0 and population array length. The initial values for variables and initial population are addressed in Lines L1–L3. Lines L4–L11 shows the main process of selection and reproduction. In line L5 and L6 two random number are generated. Line L7 and L8 select two individuals as parents. Whatever fairness is smaller, higher-ranked individuals are selected to mate. In line L9, selected parents mate through genetic operations (i.e. crossover and mutation) to produce new offspring and replace them for the least Pareto rank individuals. Crossover and mutation are described in details in the following. Regarding the runtime complexity of this algorithm, there is a loop which iterates for the number of new children. The number of new children is a fraction of the total number of population which in its turn, is a small fraction of the number of all possible combinations of state assignment; i.e. $2^{n\log(n)}$ where n shows the number of states of the FSM.

It is notable that, the iteration number is much smaller that the number of all possible combinations; for example for the largest benchmark circuit “tbk” with 32 state flip-flops, the circuit is optimized for only 20 iterations of algorithm 1.

```
Algorithm 1: Selection and reproduction Algorithm Pseudocode
Input: Population, fdr                Output: Next Generation
Sort Population in order of decreasing Pareto rank
fairness = initial value * fdr
Set index_to_replace to maximum index of population
For i:1 to num_new_child
    r1= Random (0, max_array_index)
    r2= Random (0, max_array_index)
    parent1 = population [fairness*r1]
    parent2 = population [fairness*r2]
    population [index_to_replace]= Mutation(Crossover(parent1, parent2))
    index_to_replace = index_to_replace-1
End
```

5) Step ❺: Crossover and Mutation

Each gene in a chromosome represents a case of the state encoding in the FSM. In this work, one-point crossover is used, in which a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.

Algorithm 2 show crossover procedure. By using one-point crossover, *head* and *tail* obtained for *parent1* and *parent2* (Line L1, L2). Here, we consider *head* and *tail* length as half of total states. Then, *head1* is added to the new child (offspring). To complete child encoding, we define a set called *remained* which contains the encoding that exist in *parent2* (*ENC2*) and not used in *head1*. Finally, the encoding from *remained* will be added to new child randomly. The complexity of this algorithm is O(1) as there is no loop in it.

```
Algorithm 2: Crossover procedure
Input: parent1, parent2               Output: new offspring
L1:   Get head1,tail1 of parent1
L2:   Get head2,tail2 of parent2
L3:   Add head1 to new_child1
L4:   Let remained = {ENC2 – head1}
L6:   Add from remained to the new_child1 randomly
L11:  End
```

When a new offspring is generated, the mutation is started. We use replacing mutation in which a random encoding is selected to be replaced with another from the encodings used or unused in the selected parent. The rate of genetic operations (i.e. crossovers and mutations per generation) are pre-specified by the user in GenSO.

IV. Experimental results

In this section, the efficacy of the proposed framework in nano-scale sequential circuits is investigated using extensive experiments. First, we describe the experiment setup and then, various aspects of GenSO in multi-objective FSM optimization are investigated.

A. Experiment Setup

The proposed framework, GenSO, is implemented in C++ programming. All experiments are carried out on an Intel Core i7 quad-core, with clock frequency of 4.6 GHz and total 32G RAM. To implement the proposed parallelized approach, CUDA is used which a parallel is computing platform and application programming interface (API) model developed by NVIDIA. The CUDA device is NVIDIA GeForce GTX 1080 with 8GB memory and 20 stream multiprocessors (SMs).

FSMs of LGSYSYNTH90 suit benchmarks [30] and ITC’99 [37] are synthesized with Berkley ABC synthesis tool [31] which is an open-source tool used for synthesis and verification of binary sequential logic circuits. Scalable logic optimization is combined by ABC on the basis of And-Inverter Graphs (AIGs), optimal-delay DAG-based technology mapping for look-up tables and standard cells, and innovative algorithms for sequential synthesis and verification.

For timing analysis, parameters $A, B, \gamma,$ and β in the statistical aging model (i.e. Eq. 4) are fitted by HSPICE simulation. In the fitting process, MOSFET Model Reliability Analysis (MOSRA) is used and all HSPICE simulations are performed with PTM 22nm technology node model [32], supply voltage 1.1V, and 354°K temperature. The switching activity of the signals are calculated with regard to their signal probabilities. Values of SP for all primary inputs’ (PIs) are set to 0.5, and for internal nodes, SP is obtained with the help of approach proposed in [36].

In order to model the spatial correlation between different gates in the circuit, a 3-level quad-tree partition [33] is used. Each gate is randomly allocated on the 4×4 grid at the bottom level and then, random variables related to the gate along the hierarchy are determined. The random variables at the same level have the same probability distribution. The total PV effect on Vth is assumed 10% which is divided into 6% systematic variation and 8% random variations.

GA configuration parameters values used in the simulation are shown in Table V. The number of new solutions is the number of the low-rank solutions replaced by the newly generated children. After performing crossover and mutation, one evolution cycle finishes.

TABLE V
GA Configuration Parameters Values

Parameter	Value
Population	100
Mutation rate	0.5
#mutation	5
#cross over	10
#new child	20
#halt condition	20
fairness	1
fdr	0.9

B. Validation of GAR metric

In order to validate the GAR metric, an experiment is carried out, in which the delay degradation of two randomly selected sequential circuits form *kiss2* benchmark are analyzed using the HSPICE simulation: a circuit from the first population (SC_1) and another one from the final population (SC_2). It is notable, that the BTI-induced delay degradation of a circuit is inversely related to its lifetime reliability; i.e. a circuit with less delay degradation has higher lifetime reliability. The obtained results presented in Table V shows that the delay degradation of SC_2 is less than SC_1. This means that the GAR metric incorporated in fitness function of GenSO can effectively guide it to find the FSM encoding with better lifetime reliability.

TABLE VI
Delay Degradation, Initial Delay, and Power Consumption of Two Circuits From the First and Final Population Generated by GenSO

Circuit	Delay Deg.	Delay (ps)	Avg. Power (nW)
SC1	48%	41.3	392
SC2	22%	15.7	288

C. Lifetime reliability analysis

Fig. 4. shows the reliability of six LGSYNTH90 benchmarks (Kiss2 format) and two ITC’99 benchmarks (b12 and b15) during their lifetime under 15% PV ratio ($3\sigma/\mu$). FSM synthesized with optional coding. As shown in Fig. 4., FSM reliability is decreased during its operational lifetime. For example, the reliability of beecount is reduced to 0.8222, 0.6492, and 0.5188 for 3-, 6- and 9-year lifetime.

Fig. 5. shows circuit reliability with different PV ratios ($3\sigma/\mu$) during 6-year lifetime. It is observed that increasing PV decreases the circuit lifetime reliability. For instance, the reliability for beecount is 0.6935, 0.6492 and 0.6295 for 5%, 7% and 10% PV.

D. Multi-objective Optimization Results

In order to investigate the multi-objective optimization provided by GenSO, Fig. 6. (a-d) show the initial and final population for some representative benchmarks (i.e. *dk27.kiss2*, *sand.kiss2*, *tbk.kiss2*, and *b15*). As can be observed, the power consumption, GAR degradation, and the initial delay of the final population (shown by the blue stars) are less than the initial one (indicated by the red circles). This acknowledges that GenSO improves the populations (i.e. state encoding of sequential circuits) in terms of all objectives simultaneously.

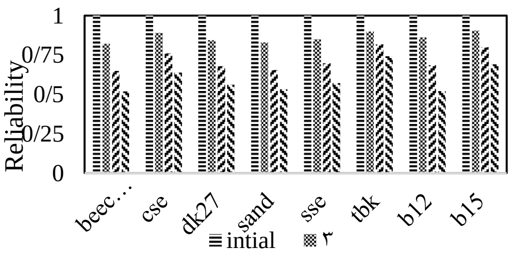


Fig. 4. FSM reliability during different lifetimes (under 15% PV ratio).

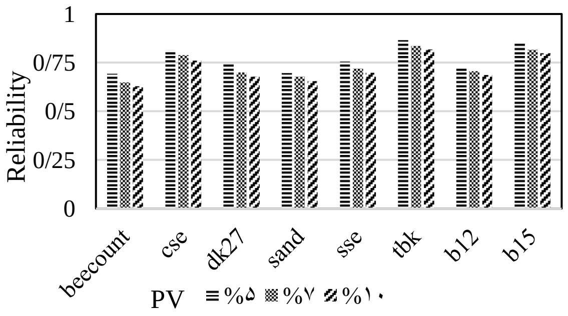
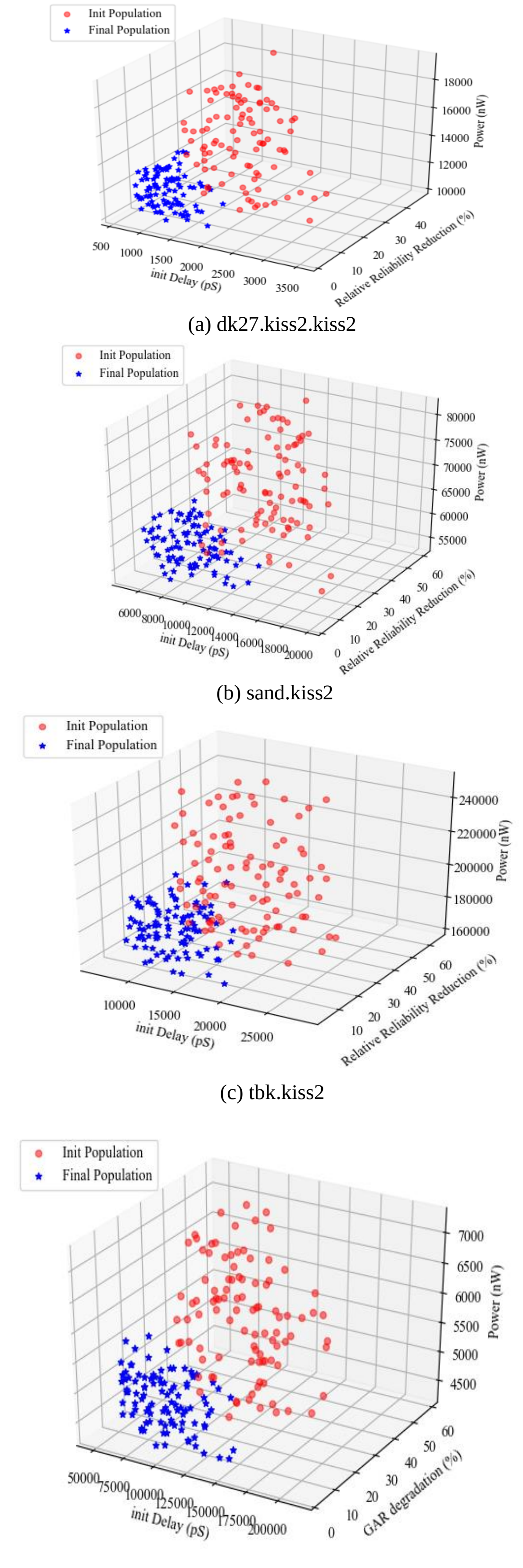


Fig. 5. FSM Reliability for different PV ratios (operation time of 6 years).



(a) dk27.kiss2.kiss2

(b) sand.kiss2

(c) tbk.kiss2

(d) b12

Fig. 6. The initial and final population generated by GenSO for four representative benchmarks

In order to compare the initial and final populations, we choose the solutions with the best fitness value from the initial and final populations. Since GenSO is a multi-objective framework, we choose the solutions with a fitness value computed by a weighted sum with the equal weights for the objectives. Table VII compare the solution with the best fitness value from the initial and final population. Column 1 shows the benchmark name and column 2 indicates whether the initial population is shown in the row or the final one. Column 3 shows the initial delay and power consumption. The following three columns show the GAR degradation for 3-, 6-, and 9-year lifetime. As can observed, GenSO improves all parameter values of the solutions and find the solutions which are better in all respect compared with initial ones.

Table VIII shows the efficacy of GenSO in terms of aging-induced timing for lifetime reliability of 99% for 6-year operation time and 15% process variation ratio. The first column indicates the benchmark circuit while the second, third, and fourth columns respectively show the timing margin without GenSO, timing margin with GenSO, and the improvement achieved by GenSO.

TABLE VII
Comparison of the Initial and Final Generation

Bench.	**Gen.**	**Initial**		$\Delta\mathcal{R}_{0.1}^{0.99}(3)$	$\Delta\mathcal{R}_{0.1}^{0.99}(6)$	$\Delta\mathcal{R}_{0.1}^{0.99}(9)$
		Delay (ps)	**Power (nW)**	**%**	**%**	**%**
beecount	Initial	2640	221974	17.76	35.07	48.10
	Final	1741	213368	5.89	15.60	25.98
cse	Initial	5289	145123	10.50	21.90	32.83
	Final	4471	128387	4.83	10.69	17.29
dk27	Initial	1227	12590	19.03	37.49	51.31
	Final	805	11801	2.42	6.31	10.56
lion9	Initial	2153	44423	9.88	22.26	33.90
	Final	1388	41777	2.71	8.25	14.70
planet	Initial	17843	528562	13.98	30.10	42.58
	Final	11267	494259	3.42	8.71	15.50
sand	Initial	9238	59861	12.44	24.17	33.68
	Final	7320	56750	3.37	7.35	12.00
sse	Initial	4210	102172	15.74	33.43	48.77
	Final	3181	91861	5.72	12.98	21.52
tbk	Initial	12310	179815	18.85	35.53	49.89
	Final	8235	173878	6.21	14.16	20.74
b12	Initial	20451	2328032	17.89	31.01	42.85
	Final	15392	1893267	6.24	10.17	18.05
b15	Initial	36740	14919274	21.67	47.70	48.01
	Final	27941	11423568	11.98	22.06	25.89

Table IX compares GenSO with the method proposed in [34] in terms of improvement in delay, power consumption, and GAR ($\Delta\mathcal{R}_{0.1}^{0.99}(6)$) for 6-year operation time and 15% process variation ratio. The temperature considered in SA-based optimization is set to 105 °C with the cooldown factor of 0.2.

TABLE VIII
Comparison of Aging-Induced Timing Margin Without and With GenSO

Bench.	Timing margin **without** GenSO (ps)	Timing margin **with** GenSO (ps)	Improvement (%)
beecount	3247	2355	27.4
cse	6082	5096	16.2
dk27	1386	985	28.9
lion9	2497	1879	24.7
planet	20697	14754	28.7
sand	10716	8251	23.0
sse	4925	3658	25.8
tbk	15018	11650	22.4
b12	22970	17512	23.7
b15	34451	25967	24.6
AVG.	-	-	**24.5**

In order to have a fair comparison, the initial solution is set to be similar in both methods and we choose the solutions with a fitness value computed by a weighted sum with the equal weights for the objectives as the first and final steps. On average, GenSO improves the circuit reliability by 30.03% while 16.04% reliability improvement is achieved by [34]. Also, 14.45% improvement in delay and 3.28% improvement in power consumption is achieved by GenSO while the SA-based optimization improves the delay and power by 6.86% and 1.70%, respectively. Although the runtime of the method presented in [34] much less that GenSO, it is based on SA algorithm which is weaker in global optimizations while GenSO is based on GA which can appropriately searches the global solution space. Moreover, [34] ignores PBTI effects on the circuit timing reliability leading to less reliability improvement than GenSO.

TABLE IX
Comparison of GenSO and the SA-based Optimization Method Proposed in [34] in Terms of Improvement in Delay, Power, and GAR

Bench.	**SA-based Opt. Method [34]**			**GenSO**		
	Delay	**Power**	**GAR**	**Delay**	**Power**	**GAR**
beecount	8.29	1.09	14.68	17.09	1.93	26.01
cse	3.62	2.76	12.75	7.72	5.74	22.78
dk27	7.05	1.56	18.94	17.14	3.13	39.94
lion9	8.63	1.88	16.32	17.73	2.97	28.71
planet	8.32	1.57	17.05	18.42	3.25	33.48
sand	3.04	0.76	11.74	10.32	2.59	32.25
sse	5.13	2.97	17.12	10.71	5.05	28.74
tbk	10.88	1.04	19.79	16.54	1.65	28.34
AVG. (%)	**6.86**	**1.70**	**16.04**	**14.45**	**3.28**	**30.03**

V. Conclusion

In this paper, a framework called GenSO is introduced for multi-objective optimization of FSM models of sequential circuits. GenSO is the first framework in which the lifetime reliability parameter is improved during the state encoding step of FSM synthesis. GenSO takes advantage of a genetic algorithm based optimization engine to explore the state encoding solutions in the objective space of initial delay, power consumption, and lifetime reliability. The efficacy of GenSO is investigated through vast experiments on LGSYNTH90 benchmark suits. As an example, the experimental results show that GenSO provides 64.3% reliability improvement for 6-year lifetime and 10% PV with 15% initial delay overhead.

A. Future Work

There are some new avenues to extend GenSO in the future. The lifetime reliability evaluation engine can be extended to consider workload-aware BTI modeling. Moreover, GenSO can be expanded to include additional parameters such as Carrier mobility, Sub-threshold slope (SS), Gate-drain capacitance (Cgd). Also, since BTI is highly sensitive to operating temperature, techniques such as

Dynamic supply voltage scaling (DVS) or factors such as temperature variations can be included in future works of this framework. Also, we plan to incorporate workload-aware BTI recovery models that account for periods of reduced stress or idle modes and include stress relaxation effects. We believe that adding these recovery mechanisms will provide a more realistic and balanced estimation of circuit lifetime reliability, especially in designs optimized for power efficiency through frequent low-power modes.

Declarations

B. Ethical Approval

Not Applicable.

C. Competing interests

The authors decline any conflict of interest.

D. Authors' contributions

M. Raji and B. Ghavami provided the problem statement and solution. R. Mahmoudi implemented the solution and S. keshavarzi and R. Mahmoudi wrote the main manuscript. All authors reviewed the manuscript.

E. Funding

No funding is received.

F. Availability of data and materials

Not applicable.

References

[1] Alam, M.A. and Mahapatra, S., 2005. A comprehensive model of PMOS NBTI degradation. *Microelectronics Reliability*, *45*(1), pp.71-81.

[2] Huard, V., Angot, D. and Cacho, F., 2015, April. From bti variability to product failure rate: A technology scaling perspective. *In 2015 IEEE International Reliability Physics Symposium* (pp. 6B-3). IEEE.

[3] V. M. van Santen, J. Martin-Martinez, H. Amrouch, M. M. Nafria and J. Henkel, "Reliability in Super- and Near-Threshold Computing: A Unified Model of RTN, BTI, and PV," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 293-306, Jan. 2018, doi: 10.1109/TCSI.2017.2717790..

[4] Jafari, A., Raji, M. and Ghavami, B., 2020. Timing Reliability Improvement of Master-Slave Flip-Flops in the Presence of Aging Effects. *IEEE Transactions on Circuits and Systems I: Regular Papers, 67(12)*, pp.4761-4773.

[5] Fang, J. and Sapatnekar, S.S., 2013. The impact of BTI variations on timing in digital logic circuits. *IEEE Transactions on Device and Materials Reliability, 13(1)*, pp.277-286.

[6] Diaz-Fortuny, Javier, Mahdi Benkhelifa, Alexander Grill, Erik Bury, Robin Degraeve, Ben Kaczer, and Hussam Amrouch. "Investigation of Cryogenic Aging in 28 nm CMOS: Suppression of BTI and HCD in Circuits and SRAM." In *2025 IEEE International Reliability Physics Symposium (IRPS)*, pp. 1-7. IEEE, 2025

[7] Sharma, M., Khanna, V., Tonk, A. and Ruhil, S., 2025. Implementation and Comparative Analysis of Various Low Power FSM Synthesis Techniques. Journal of Electronics and Power Engineering (JEPE), vol. 2, no. 1, pp.54-66.

[8] Choudhury M, Gao M, Varna A, Peer E, Forte D. TRANSPOSE: Transitional Approaches for Spatially-Aware LFI Resilient FSM Encoding. arXiv preprint arXiv:2411.02798. 2024 Nov 5.

[9] Babakov, R.M. and Barkalov, A.A., 2025. An Algorithm for Solving the Problem of Algebraic Synthesis of a Finite-State Machine with Datapath of Transitions Based on a Matrix Approach. Cybernetics and Systems Analysis, pp.1-7.

[10] Barkalov, Alexander, Larysa Titarenko, and Kamil Mielcarek. "Transforming Group Codes in Mealy Finite State Machines with Composite State Codes." Applied Sciences 15, no. 8 (2025): 4289.

[11] Al Jassani, B.A., Urquhart, N. and Almaini, A.E.A., 2011. State assignment for sequential circuits using multi-objective genetic algorithm. *IET computers & digital techniques, 5(4)*, pp.296-305.

[12] El-Maleh, A.H., Sheikh, A.T. and Sait, S.M., 2013. Binary particle swarm optimization (BPSO) based state assignment for area minimization of sequential circuits. *Applied soft computing, 13(12)*, pp.4832-4840.

[13] El-Maleh, A.H., 2015. Majority-based evolution state assignment algorithm for area and power optimisation of sequential circuits. *IET Computers & Digital Techniques, 10(1)*, pp.30-36.

[14] Tao, Y., Zhang, Y. and Wang, Q., 2018. Fuzzy c-mean clustering-based decomposition with GA optimizer for FSM synthesis targeting to low power. *Engineering Applications of Artificial Intelligence, 68*, pp.40-52.

[15] Tao, Y., Zhang, Y., Wang, Q. and Cao, J., 2018. MPGA: an evolutionary state assignment for dynamic and leakage power reduction in FSM synthesis. *IET Computers & Digital Techniques, 12(3)*, pp.111-120.

[16] Pendyala, S. and Katkoori, S., 2016, March. State encoding based NBTI optimization in finite state machines. *In 2016 17th International Symposium on Quality Electronic Design (ISQED)* (pp. 416-422). IEEE.

[17] Amrouch, H., van Santen, V.M. and Henkel, J., 2016. Interdependencies of degradation effects and their impact on computing. *IEEE Design & Test, 34(3)*, pp.59-67.

[18] Mukhopadhyay, S., Goel, N. and Mahapatra, S., 2016. A comparative study of NBTI and PBTI using different experimental techniques. *IEEE Transactions on Electron Devices, 63(10)*, pp.4038-4045.

[19] Sexton, R.S., Dorsey, R.E. and Johnson, J.D., 1999. Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. *European Journal of Operational Research, 114(3)*, pp.589-601.

[20] Zafar, S., Kumar, A., Gusev, E. and Cartier, E., 2005. Threshold voltage instabilities in high-/spl kappa/gate dielectric stacks. *IEEE Transactions on Device and Materials Reliability, 5(1)*, pp.45-64.

[21] Tudor, B., Wang, J., Chen, Z., Tan, R., Liu, W. and Lee, F., 2012. An accurate MOSFET aging model for 28 nm integrated circuit simulation. *Microelectronics Reliability, 52(8)*, pp.1565-1570.

[22] Yang, H.I., Yang, S.C., Hwang, W. and Chuang, C.T., 2011. Impacts of NBTI/PBTI on timing control circuits and degradation tolerant design in nanoscale CMOS SRAM. *IEEE Transactions on Circuits and Systems I: Regular Papers, 58(6)*, pp.1239-1251.

[23] Wenping Wang, Zile Wei, Shengqi Yang and Yu Cao, "An efficient method to identify critical gates under circuit aging," *2007 IEEE/ACM International Conference on Computer-*

Aided Design, 2007, pp. 735-740, doi: 10.1109/ICCAD.2007.4397353

[24] A. Gomez, and V.Champac, “An efficient metric-guided gate sizing methodology for guardband reduction under process variations and aging effects,” *Journal of Electronic Testing*, vol. 35, No. 1, pp. 87-100, 2019.

[25] Raji, M., Mahmoudi, R., Ghavami, B. and Keshavarzi, S., 2021. Lifetime Reliability Improvement of Nano-Scale Digital Circuits Using Dual Threshold Voltage Assignment. *IEEE Access*, 9, pp.114120-114134.

[26] Wang, W., Reddy, V., Yang, B., Balakrishnan, V., Krishnan, S. and Cao, Y., 2008, September. Statistical prediction of circuit aging under process variations. *In 2008 IEEE Custom Integrated Circuits Conference* (pp. 13-16). IEEE.

[27] Blaauw, D., Chopra, K., Srivastava, A. and Scheffer, L., 2008. Statistical timing analysis: *From basic principles to state of the art. IEEE transactions on computer-aided design of integrated circuits and systems, 27(4)*, pp.589-607.

[28] Papoulis, A. and Pillai, S.U., 2002. Probability, random variables, and stochastic processes. *Tata McGraw-Hill Education*.

[29] Goldberg, D.E., 1989. Genetic algorithms in search, optimization, and machine learning. *Addison. Reading*.

[30] http://www.pld.ttu.ee/applets/dsa/Libs/sis/

[31] https://people.eecs.berkeley.edu/~alanmi/abc/

[32] http://ptm.asu.edu/

[33] Agarwal, A., Blaauw, D. and Zolotov, V., 2003, November. Statistical timing analysis for intra-die process variations with spatial correlations. In ICCAD-2003. *International Conference on Computer Aided Design (IEEE Cat. No. 03CH37486)* (pp. 900-907). IEEE.

[34] Pendyala, S. and Katkoori, S., 2016, March. State encoding based NBTI optimization in finite state machines. *In 2016 17th International Symposium on Quality Electronic Design (ISQED)* (pp. 416-422). IEEE.

[35] Zizheng Guo, Tsung-Wei Huang, and Yibo Lin. 2020. GPU-accelerated static timing analysis. In *Proceedings of the 39th International Conference on Computer-Aided Design (ICCAD '20)*, 2020, pp. 1–9.

[36] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, “Estimate of signal probability in combinational logic networks,” *in [1989] Proceedings of the 1st European Test Conference*, 1989, pp. 132–138.

[37] S. Davidson, “ITC’99 benchmark circuits-preliminary results,” *in International Test Conference 1999. Proceedings (IEEE Cat. No. 99CH37034)*, pp. 1125–1125, IEEE, 1999