1. Introduction

Over the past several decades, the main factors influencing the design of digital circuits have been construction costs, circuit performance, and power consumption [1,2]. However with aggressive technology scaling, reliability of digital circuits becomes a major issue in circuit design [3,4]. Since a longtime lifespan as well as correct functionality of the circuit is clear imperative for many fields of application such as aerospace, military and also medical industries, reliability issues should be taken into account at the design phase of digital circuits.

Transistor aging is a key source of failure that threatens the reliability of digital circuits by remarkably degrading the electrical characteristics of transistor results in, increasing the device delay [2]. Among different aging phenomena such as Hot Carrier Injection (HCI) and Time-Dependent Dielectric Breakdown (TDDB), Negative Bias Temperature Instability is a major challenge [5]. The NBTI phenomenon consists of two phases: the stress phase, and the recovery phase. In the stress phase where the transistor is active at the elevated temperature [6], the electric field breaks the Si-H bonds leaving some dangling Si- bonds. These are attracted and filled with holes which are the main charge carriers in P-type MOSFETS. Hence, the charge carriers gradually get trapped. This reduces the current density over time and increases the threshold voltage of the device. However, in the recovery phase when the transistor turns off, the previous V_{th} degradation is partially compensated. This phenomenon is known as partial-recovery. Since the recovery phase cannot fully compensated for the V_{th} degradation induced by the stress phase, the degradation process is gradual and accumulative over time. Therefore, BTI manifests itself as an increased V_{th} shift which in turns causes degradation of combinational circuits delay, timing violations can occur in the field and eventually cause faster wear out of the system. For example, the increase in the circuit delay due to threshold voltage shift caused by the NBTI aging phenomenon is shown to be up to 30% for 65 nm technology [7].

Aging-aware circuit design and optimization is therefore a crucial part of the design phase of digital circuits. Over the past decades, a great number of approaches have been introduced subjected to aging-aware lifetime reliability optimization of the circuit which can be categorized into the following two categories:

1) *Sense and react approaches:* using sensors to monitor aging-induced degradations such as timing or supply current [8, 9], these methods adaptively tune circuit parameters like supply voltage [10]. These techniques are so accurate in terms of obtaining performance degradation and aging monitoring yet, tuning circuit parameters, adaptively, may have an impact on the circuit performance and aging-induced degradations [11]. For example, with increasing supply voltage, the power consumption and the temperature of the circuit increase, and since the aging is a function of temperature and supply voltage, the degree of aging of the circuit also increases. In addition to, scaling the frequency of the circuit may not be feasible in time-critical systems.
2) *Precautionary Approaches:* by predicting aging effects at the design time, these techniques optimize the circuit such that the functionality of the circuit guaranteed for the projected lifetime. Using a pre-characterize library that comprise of lifetime delay of each cell at the projected lifetime, [12, 13] proposed an aging-aware synthesis approach that map a circuit

into an aging-aware reliable one. Nevertheless, the lifetime delay for each cell has to be measured by considering different input signal probabilities, and the new library may therefore contain too much information, reducing the efficiency of the synthesis process. Duan et al. [14] proposed a lifetime reliability synthesis approach that does not require modification of cell library. At first, a logic resynthesis method to reduce the NBTI/PBTI stress duty cycle is introduced. Then, a cell mapping strategy and gate-level gate sizing approach is proposed such that the delay degradation of the circuit due to BTI is mitigated. Using logic reconstruction technique as well as input reordering for each gate, Wu and Marculescu [15, 16] attempts to reduce the aging-induced delay degradation of the circuit. An aging-aware logic resynthesis method [17] to mitigate the aging effects of digital circuits. To this end, the outputs of the circuit whose lifetime delay violates the lifetime delay constraint of the circuit are specified and their fain-in cone is extracted. Then, each of the extracted cones are optimized so that their delay at the projected lifetime does not exceed the timing constraint of the circuit. In [18, 19], considering the worst case delay degradation for all the devices, Potential Critical Paths (i.e. the paths that may exceed the timing constraint of the circuit due to aging) of the circuit are extracted and then, using a metric-guided gate sizing algorithm, the delay of potential critical paths are optimized. In [20, 21], an aging-aware gate sizing algorithm based-on selection of the most favorable gates using the results of a path-based Static Timing Analysis (STA) tool is proposed. However, all of these methods suffer from high runtime complexity which makes them infeasible for optimizing industrial-size circuits.

In this paper, a fast and scalable logic resynthesis technique to improve the lifetime reliability constraint of large scale combinational circuits is presented. In the proposed methodology, the original circuit is partitioned into a set of smaller sub-circuits using cone structures that originated from circuit outputs. The extracted sub-circuits are topologically levelized and a group of most effective sub-circuits are extracted. Then, starting from the minimum topological level until reaching the last level, all the extracted sub-circuits located at the same topological level are optimized separately and independently. In all of the previous methods, after each round of the optimization procedure, we need to re-calculate the lifetime delay of the circuit, which is time consuming. However, in the proposed method, we compute the lifetime reliability constraint of each sub-circuit and by comparison the delay of the sub-circuit at the projected lifetime with its lifetime reliability constraint, the effect of optimization on the lifetime reliability constraint of the circuit is evaluated. Since computing the delay of the sub-circuit is much faster than the circuit delay computation, computing the effect of optimization on circuit delay locally will significantly accelerating the optimization procedure.

The rest of this paper is organized as follows. Section 2 motivates the proposed partitioning-based logic resynthesis technique. In section 3, the proposed method to improve the reliability of the circuit is presented. Section 4 reports the experimental results as well as a comparison with existing optimization techniques in terms of area penalty and run time. In the end, the paper is concluded in section 5.

2. Motivation

Since aging-induced delay degradation of a gate is a function of input signal probability, gate type, temperature and transition time, the degradation of the lifetime delay will vary from one path to another. Hence, a non-critical path might become critical and vice versa over the projected lifetime due to aging. Figure 1 shows the delay of each path of a circuit with 10 paths. As shown in this Figure, at the design time, i.e. at t = 0, path P7 has the largest delay which makes it the critical path of the circuit. However, after a 10-year of a lifetime, path P4 will become the critical path of the circuit due to aging effects.

As can be seen in Figure 1, in order to optimize the circuit such that it can meet the required lifetime delay constraint, one should optimize the paths that violated the lifetime delay constraint of the circuit (paths P4, P5, P6, P7 and P8). This is the procedure which is done in many of the previous works [22-26]. In these methods, a list of paths whose delay can violate the lifetime delay constraint of the circuit is extracted (potential critical path set) and by optimizing these paths, the lifetime of the circuit is improved. However, this method is infeasible to analyze the aging effect on very large scale circuits used in industry. For example as shown in [], b19 benchmark circuit has more than 100,000,000 paths and considering the paths with only 5% relative slack time may reach $> 10^7$ paths which is intractable.

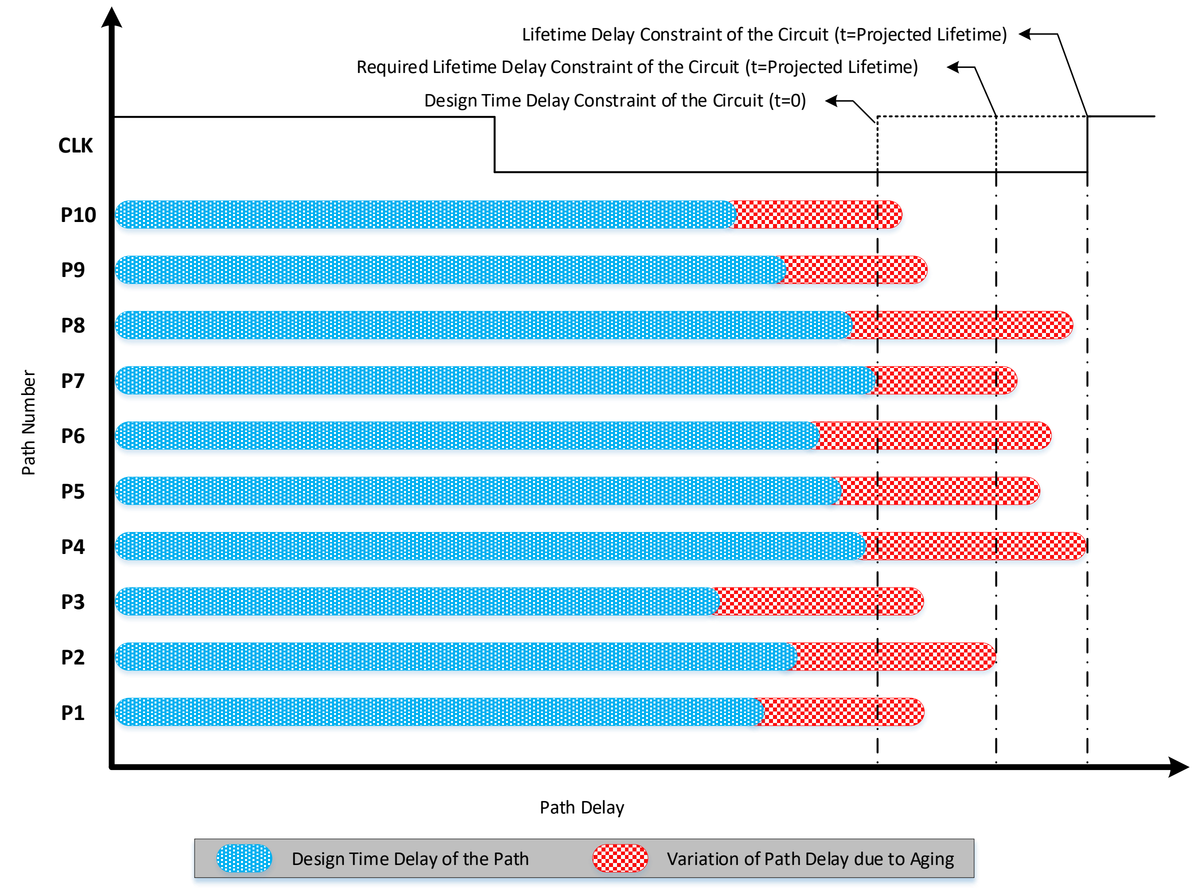

Figure 1

Besides that, in all of the aging-aware optimization techniques, after each round of optimization, the effect of optimization process on the lifetime delay of the circuit is investigated [] which causes an increase in aging-aware optimization runtime. Therefore, using these techniques is not practical for large-scale circuits.

To handle these problems, we propose a partitioning-based optimization method in which the circuit is divided into a set of smaller sub circuits. As shown in Figure 2, the proposed optimization technique consist of three steps. At first (step A), the circuit is partitioned into smaller sub circuits and sub circuit SC_i is chosen from all extracted sub circuits. Then in step B, The optimization technique applies to each sub circuit SC_i individually and independently. It is notable that using the proposed method, re-evaluating the effect of each round of optimization on the lifetime delay of the circuit is done incrementally which hugely reduce the runtime of the optimization algorithm. Finally in step C, the optimized sub circuit opt_SC_i is replaced by the SC_i in the main circuit.

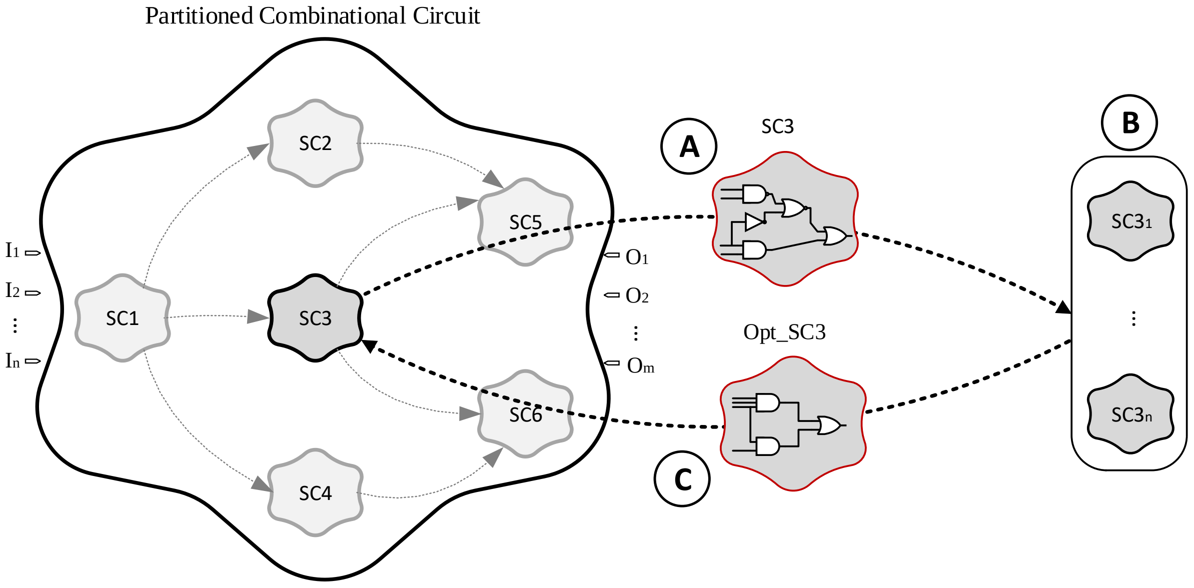

Figure 2

3. Proposed Method

Figure 3 shows the flowchart of the proposed method to reduce the performance degradation of a circuit due to aging effects. To this end, the lifetime delay of the circuit considering aging effects is first calculated using a block-based STA. Then the main circuit is divided into a set of smaller sub circuits using cone structures, and for each sub-circuit, a table including the lifetime delay from each of the sub-circuit inputs to each of its outputs is constructed. Because of aging-induced delay degradations, only the delay of a group of circuit outputs exceeds the lifetime delay constraints, thus, all the sub circuits are first levelized and then a group of the most effective sub-circuits are selected for optimization. Finally, the optimization technique is applied to each of them separately and independently.

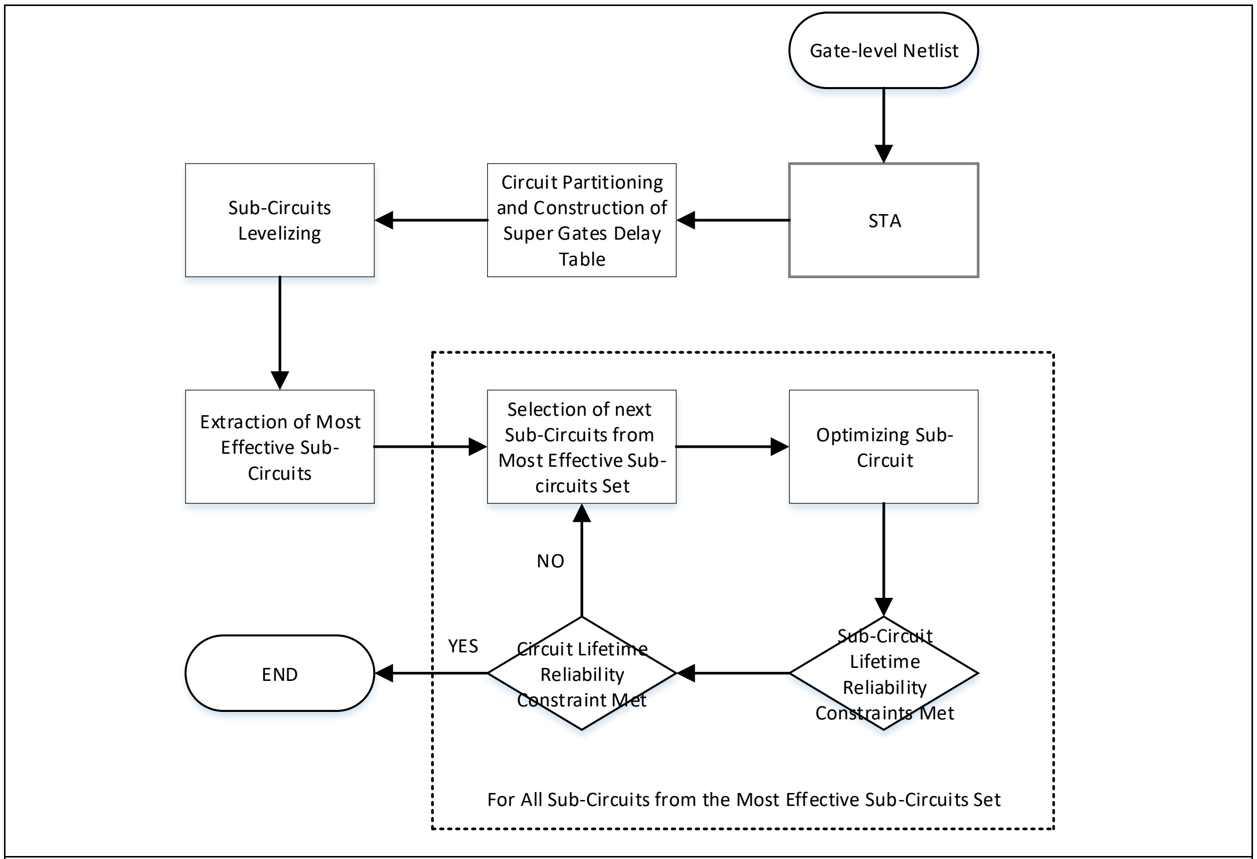
Gate-level Netlist
STA
Circuit Partitioning and Construction of Super Gates Delay Table
Sub-Circuits Levelizing
Extraction of Most Effective Sub-Circuits
Selection of next Sub-Circuits from Most Effective Sub-circuits Set
Optimizing Sub-Circuit
Sub-Circuit Lifetime Reliability Constraints Met
Circuit Lifetime Reliability Constraint Met
NO
YES
END
For All Sub-Circuits from the Most Effective Sub-Circuits Set

Figure 3

3.1. Circuit Partitioning

Partitioning the circuit into a set of smaller sub-circuits is done through cone structures that drive from the outputs of the circuit. Cone structure is defined as a set of gates that are located between a specific output and primary inputs of the circuit. Given that, in order to change the delay of each output of the circuit, the cone structure of that output should be changed. In this paper, cone structures used for circuit partitioning. Figure 4 illustrates a combinational circuit along with its cone structures.

A combinational circuit is split into a set of smaller sub-circuits using the following for steps:

1) *Graph Construction:* the circuit is modeled as a direct acyclic graph $G(V, E)$ composed of nodes (V) and edges (E). In this graph, each node $v \in V$ represents a circuit gate, and each edge $e(v_i, v_j) \in E$ denotes a path from gate v_i to gate v_j.

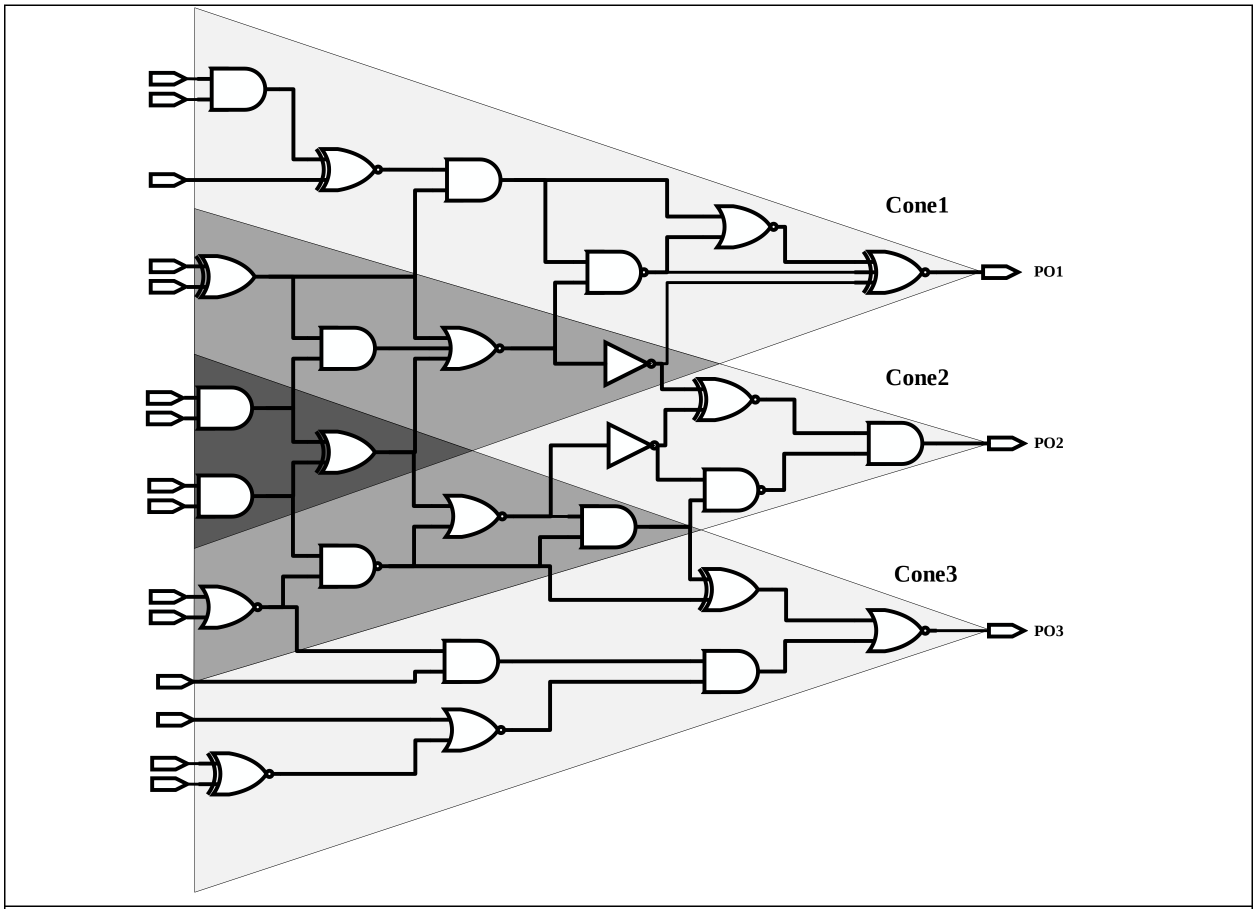

Figure 4

2) *Tagging Nodes:* the graph of the circuit is backwardly traverse from the circuit outputs. During each traverse, a tag T is assigned to each node of the graph according to the corresponding circuit output. At the end of this step, every graph node has a list of tags called tag list (TL) which can be expressed as follows:

$$\forall_{v \in V}: TL(v) = \bigcup_{g \in PL(v)} TL(g) \qquad (1)$$

Where $TL(v)$ and $PL(v)$ respectively represent the tag list of node *v* and parent list of node *v*.

3) *Graph Clustering*: a cluster is defined as a set of nodes with the same TL. In order to extract the graph clusters, a list of different TLs of the circuit is constructed which called Cluster List (CL). To this end, the nodes connected to the circuit outputs are first added to the CL. Then, the nodes whose TL is different from its parents' TL are added to the CL. These nodes are the ones located in the overlapping region between two or more cones. Thus, CL can be constructed as follows:

$$CL = \bigcup_{v\in V}\left\{\left((v\ connected\ to\ circuit\ outputs)\ OR\ \left(\exists_{g\in PL(v)}\ s.t.\ TL(v) \neq TL(g)\right)\right)\ AND\ \left(\nexists_{cl\in CL}\ s.t.\ TL(v) = TL(cl)\right)\right\} \tag{2}$$

Finally, using the CL, clustering is performed as follows:

$$\forall_{cl_i\in CL}: Cluster_i = cl_i \bigcup_{v\in V}\left\{v: \left(TL(v) = TL(cl_i)\right)\ AND\ \left(\exists PL(v)\ s.t.\ TL(v) = TL\left(PL(V)\right)\right)\right\} \tag{3}$$

4) *Cluster inputs-outputs assignment:* By assigning inputs and outputs to each cluster, the circuit is split into a set of sub-circuits. To this end, the nodes of the cluster which at least have a parent with a different TL are considered as output of the sub-circuit. Furthermore, if the node is connected to circuit output, it is considered as the output of the sub-circuit, i.e.:

$$Outputs_i = \left\{v \in V:\ (v \in V)\ OR\ \left(\exists PL(v)\ s.t.\ TL(v) \neq TL\left(PL(V)\right)\right)\right\} \tag{4}$$

where $Outputs_i$ depicts the outputs of *i*th cluster.

The nodes which are connected to the circuit primary inputs and the nodes whose children have different TLs are considered as the inputs of the sub-circuits, i.e.:

$$Inputs_i = \left\{e \in E: (e \in Circuit\ Inputs)\ OR\ \left(TL(v) \neq TL(u)\right)\right\} \tag{5}$$

where node *v* is connected to node *u* by edge *e*.

Using the proposed cone structure partitioning approach, the circuit is divided into a set of independent smaller sub-circuits. Figure 5 shows the partitioned graph of Figure 4. It is worthy to note that such cone-oriented circuit partitioning facilitates levelizing the extracted sub-circuits which prevents from excessive re-computation of circuit delay during optimization (it will be discussed in more details in the following sections).

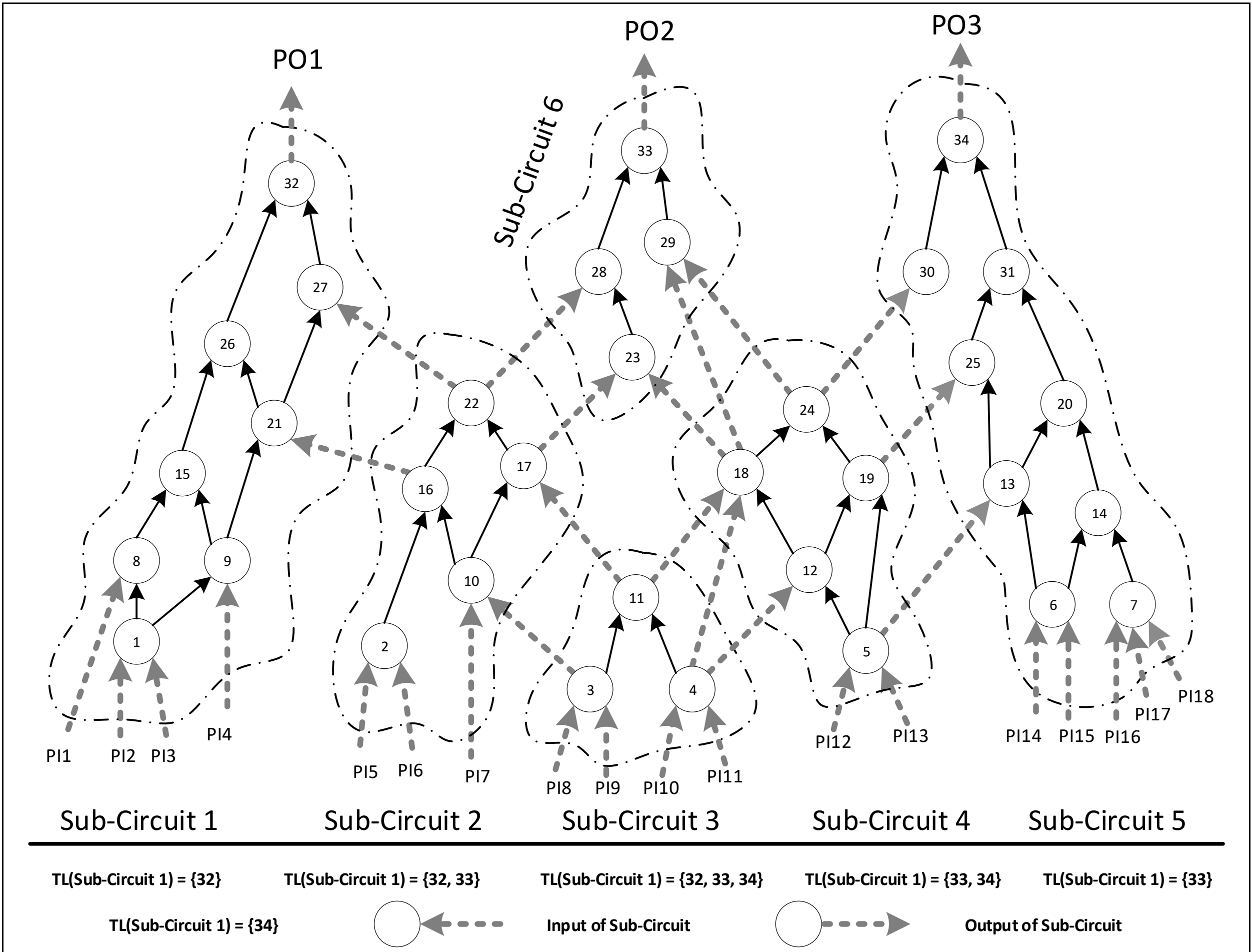

Figure 5

3.2.Static Timing Analysis (STA)

STA is a well-known computation method for calculating digital circuit delay. In order to calculate the clock period of a digital circuit at the projected lifetime, STA requires the post-aging delay information of all circuit elements, i.e. gates and interconnects.

In order to calculate the post-aging delay of each gate, we first need to compute the effect of NBTI on the V_{th} of a transistor. To this end, we use the widely accepted reaction-diffusion based predictive model [27]:

$$\Delta V_{th\_nbti} = \left(\frac{\sqrt{K_v^2 \; s \; T_{clk}}}{1 - \beta_t^{\frac{1}{2n}}} \right)^{2n} \tag{6}$$

where K_v takes the dependence of electrical field, temperature, and carrier concentration, β_t is the recovery parameter, t represents transistor lifetime (or the time in which the measurement is performed), V_{th} and s respectively represent the initial threshold voltage and signal probability (duty cycle) of transistor, T_{clk} shows input clock cycle, and n is determined by dispersion material which is around 0.16 for H_2. For a more detailed explanation of the model parameters, please refer to [27]. By considering the effects of temperature (T), signal probability (λ), width to length ratio (W/L), projected lifetime (t), and initial V_{th}, we conduct SPICE simulations to compute V_{th}degradation of a transistor under different conditions. Afterwards, using SPICE simulation results, degraded transistor models are created (N degraded transistor model for N different conditions). Then, using the degraded transistor models along with gate conditions, SPICE simulations were performed to compute the lifetime delay and slope of each gate. Gate conditions are defined as input signal slew range $[S_{min}, S_{max}]$ and output load capacitance range $[C_{min}, C_{max}]$. It is notable that computing slope of the gate requires parasitic information such as inner capacitance and resistors.

Generally STA comprises of two step: *Forward Propagation* and *Backward Propagation*. Starting from circuit inputs, the forward propagation step, at first, compute the maximum and minimum arrival time (AT_{max}, AT_{min}) of each gate and then propagates them from the gate input to the gate output, i.e.:

$$AT_{max}(gate_i) = \max_{1\le j\le m}\left\{AT_{max}\left(gate_j\right)\right\} + Delay(gate_i) \tag{7}$$

$$AT_{min}(gate_i) = \min_{1\le j\le m}\left\{AT_{min}\left(gate_j\right)\right\} + Delay(gate_i) \tag{8}$$

The second step includes backwardly propagating the maximum and minimum required arrival time (RAT_{max}, RAT_{min}) of each gate, i.e.:

$$RAT_{max}(gate_i) = \min_{1\le j\le m}\left\{RAT_{max}\left(gate_j\right)\right\} - Delay(gate_i) \tag{9}$$

$$RAT_{min}(gate_i) = \max_{1\le j\le m}\left\{RAT_{min}\left(gate_j\right)\right\} - Delay(gate_i) \tag{10}$$

3.3.Levelizing Sub-Circuits

Consider Figure 5. In this figure, if we first optimize the sub-circuit 2 and then optimizing sub-circuit 3, we should re-optimize sub-circuit 2. It is due to the fact that, the optimization process aimed to optimizing the lifetime delay of each sub-circuit considering the arrival time of the inputs of the sub-circuits such that the circuit eventually obey the specified lifetime delay constraint. However, since the optimization of each sub-circuit results in the change of arrival time in its fan-out cone, all sub-circuits must first be levelized, and then the optimization algorithm proceeds with the sub-circuits in the lowest level (level 1). After resizing the sub-circuits of the first level, the arrival time of sub-circuits in the next level is recomputed. This procedure is done level by level until the sub-circuits of last level are visited. Figure 6 shows the levelized sub-circuits of figure 5.

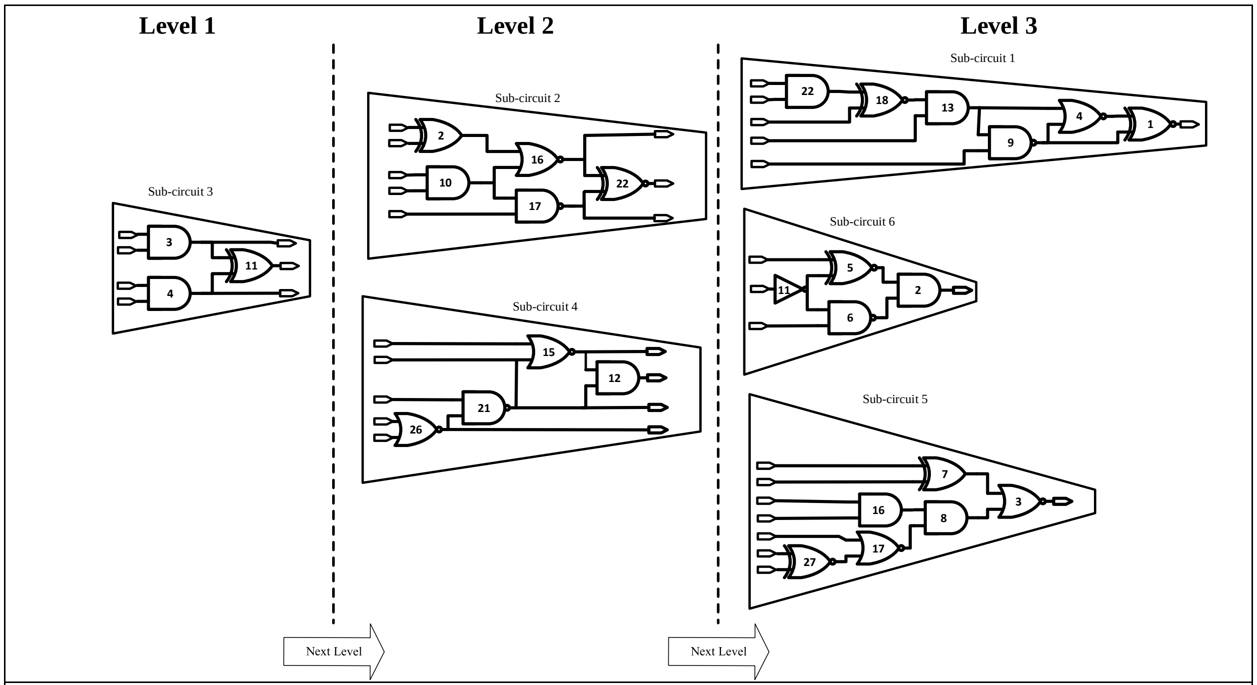

Figure 6

Due to aging effect, the delay of a group of circuit outputs exceeds the lifetime delay constraints, and the delay of other outputs does not violate these constraints. As a result, there is no need to optimize all sub-circuits derived from the partitioning algorithm and we should only optimize the sub-circuits that can ultimately lead to violations of circuit lifetime constraints, i.e. the sub-circuits from which there is a path to the outputs of the circuit whose delay violates the lifetime delay constraint of the circuit. Therefore, in the proposed aging-aware optimization method, using the results obtained from the STA, the circuit outputs that can violate the lifetime delay constraint of the circuit are specified and then, a group of sub-circuits which is in their TL, the corresponding outputs label existed are placed in the most effective sub-circuits set and are chosen for optimization. The following sub-circuits can be used to reduce the area overhead imposed by the optimization algorithm.

3.4.Sub-Circuit Optimization

Without loss of generality, a logic resynthesis method based on the AND-INVERTER graph (AIG) is used to improve the lifetime reliability of combinational circuits. AIG is a widely used method in order to implement logical functions [28]. This graph consists of only AND and Inverter gates. The corresponding AIG of a logical function is not unique, i.e. different AIG structures can be created from a logical function. For example, Figure 7 illustrates three different AIGs for the logical function F = abcd.

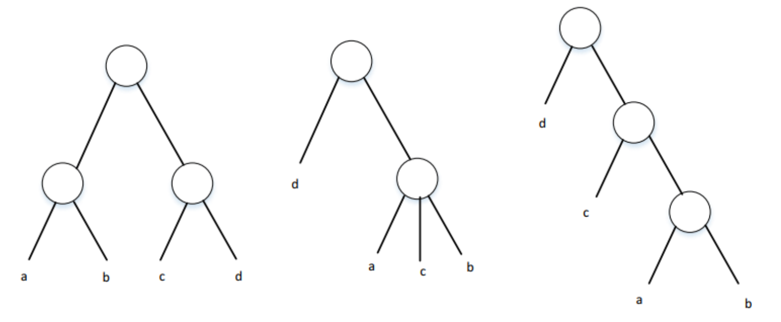

Figure 7

Since the AIG does not create canonical forms for a logical function F, one can obtain various structures from the AIG of F taking into account different design parameters such as delay, area, gate number and number of levels of the circuit. In this paper, in order to provide different implementations of a sub-circuit, at first the graph of AIG corresponding to the sub-circuit function is created and then, different implementations of the sub-circuit are extracted considering design parameters and the standard cell library. Due to the difference in the structure of each of these implementations, including the number and manner of connection between gates, the amount of aging-induce delay degradation of these structures will be different.

Using the required arrival time at each output of the sub-circuit that obtained from STA, we compute the lifetime reliability constraint of each sub-circuit and by comparison the delay of the sub-circuit at the projected lifetime with its lifetime reliability constraint, the effect of optimization on the lifetime reliability constraint of the circuit is evaluated. The optimization procedure continues until the sub-circuits met its lifetime reliability constraints. Since computing the delay of the sub-circuit is much faster than the circuit delay computation, computing the effect of optimization on circuit delay locally will significantly accelerating the optimization procedure.

3.5.Circuit Timing Update

After optimizing each sub-circuit, we need to re-calculate the lifetime delay of the circuit and compare it with its lifetime delay constraint. However, re-computing circuit delay after optimizing each sub-circuit will increase the runtime of the algorithm, significantly. In order to reduce the runtime overhead of re-computing circuit delay, we first compute the delay of each sub-circuit at the projected lifetime. After this process, each sub-circuit is considered as a super gate whose delay from the sub-circuit inputs to its outputs are determined (figure 8).

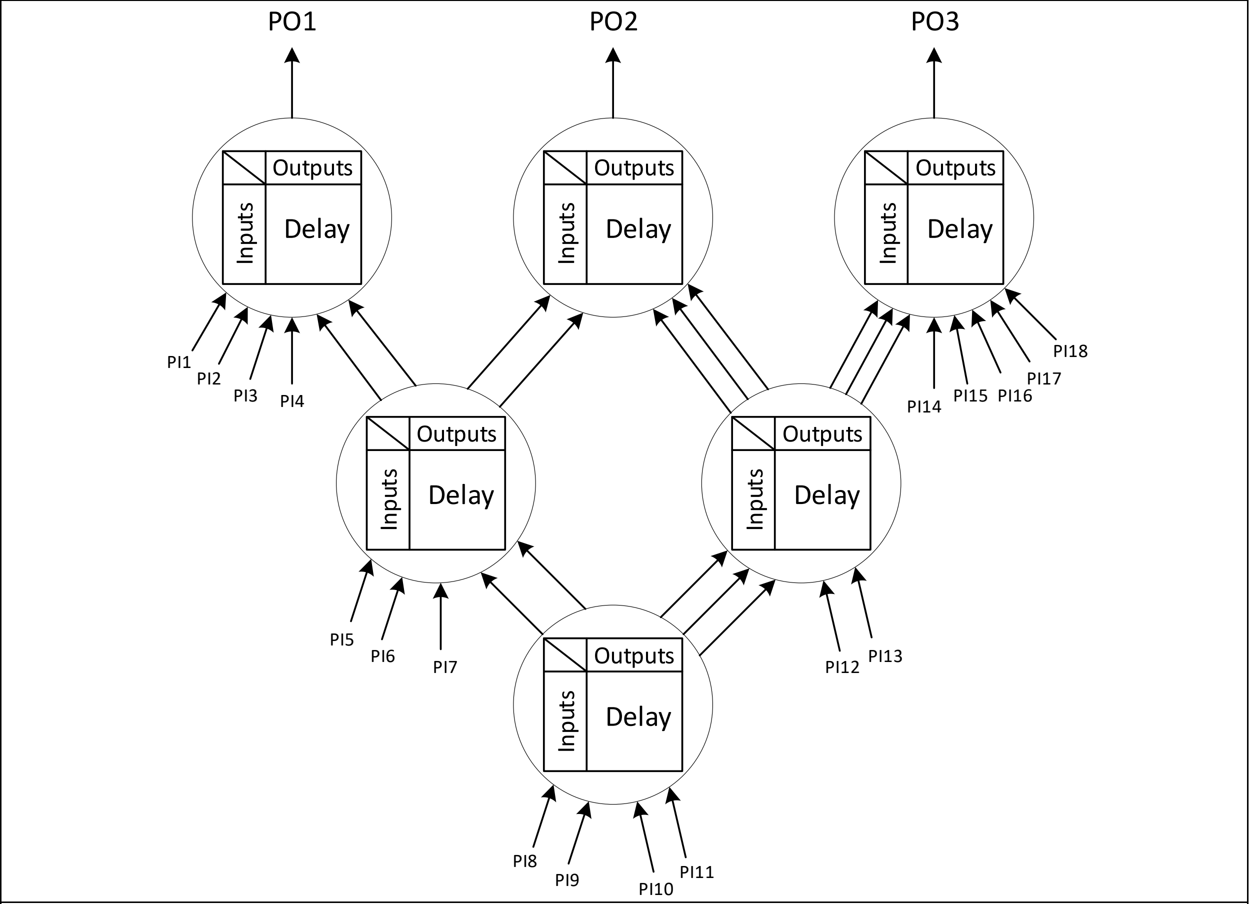

Figure 8

Then, using the new arrival times obtained from the optimized sub-circuit, we only need to re-compute the delay super gates located in its fan-out cone in order to calculate circuit delay since the arrival time of the paths going through the optimized sub-circuit are changed. In figure 9 assumes that we optimize sub-circuit 2. Since by optimizing this sub-circuit, the arrival time of super gate 1 and super gate 6 has been changed, we need to re-compute the delay of the circuit considering these super gates.

Figure 9