Tuning Logic Simulator for Estimation of VLSI Timing Degradation under Aging

Miljana MILIĆ
Universiy of Niš, Faculty of Electronic Engineering
Aleksandra Medvedeva 14, 18000 Niš, Serbia
miljana.milic@elfak.ni.ac.rs

***Abstract*—The importance of aging effects analysis in VLSI circuits increases with nowadays fast scaling of integrated circuits manufacturing technologies. Delays along paths in a digital circuit are crucial parameters that define the circuit working frequency. They degrade over time resulting in delay faults and circuit failures. The prediction of circuit long-term behavior is useful mechanism for ensuring a VLSI's lifetime reliability. In particular, paths in a digital circuit that have the largest delays are the most sensitive to gates' delay fluctuations, and consequently aging. Delay of those paths can be obtained using either aging sensors or through statistical analysis of accelerated aging experiments, but such approaches can be very difficult, time consuming and expensive for implementation. This paper suggests a new methodology capable to estimate the aging effect to digital circuit delays along multiple paths, simultaneously. The proposed technique has been developed for circuits described at a gate level, and implemented within a standard logic simulator, which enables aging analysis in initial phases of system design process. Results show that proposed methodology can efficiently estimate the long-term timing behavior of the digital circuit in a very early design stages with a small computational effort, helping the designer in selection of most reliable design choices.**

***Index Terms*—accelerated aging, circuit simulation, integrated circuit modeling, integrated circuit reliability, very large integration.**

I. INTRODUCTION

Current VLSI development does not tolerate any unnecessary power, area and delay overheads. It requires fast, reliable and fault tolerant design solutions, insensitive to aging induced parameter drifts. Design solutions with predictive behavior for the guaranteed lifetime can only be considered.

Performances of electronic circuit usually experience deterioration over time i.e. aging, due to certain physical processes which are: Negative-Bias-Temperature-Instability (NBTI) (it has a dominant influence to the speed of the circuit), Channel-Hot-Carrier (CHC) i.e. Hot-Carrier-Injection (HCI), Time-Dependent-Dielectric-Breakdown (TDDB), and the interactions between them [1-4] etc. They all induce significant parameters shifts that can cause parametric or catastrophic faults in the system, which makes a prediction of VLSI delay degradation under aging an unavoidable task.

Due to the complexity of this task, analysis of timings' deterioration caused by aging of the digital circuit is still not a part of standard circuit design procedures [5]. Early estimation of the influence of components' aging to the timings of the events along the signal flows in the digital system can bring benefit to the designers, since it helps in the identification of potential future critical delay degradations, occurring later in the exploitation. Such an analysis could be helpful for development of design techniques that effectively diminish aging's negative impact on performances, improve design predictability and robustness, and guarantee the circuit long-term performances. The precise changes of the delays along paths in a circuit caused by the components' aging could be acquired only in the last phases of system design process. If the speed of the aged-circuit does not satisfy the initial timing specifications, the circuit must be redesigned [2], [4], [6]. This strongly encourages performing an aging-aware delay estimation analysis in the initial phases of system design process, i.e. during the HDL description and functional verification. This kind of design improvement is recognized as the Design for reliability (DFR) [7].

The difficulty of the aging effects analysis for higher design levels increases if one considers that the degree of degradation is defined by the circuit lifetime activity and environment conditions, which are often unknown during the circuit design [8]. In order to establish the suggested methodology for large digital circuits, two separate problems need to be solved. First, one needs to adopt aging-aware delay models for gates, and the second is to adopt a timing engine capable of dealing with such gate delay models implemented in larger digital systems.

The first problem assumes conduction of accelerated aging experiments that relate to the particular IC manufacturing technology, and corresponding delay degradation measurements. Acquired data from digital gates' reliability and aging analysis are usually statistical in nature and represented by statistical figures of merit. Obtaining required statistical parameters for gate models are subject of many studies in the field of semiconductor physics. Modeling of aging behavior at the transistor level could be classified as the most accurate, but then too complex to be considered for large digital circuits [9]. Many studies deal with the problem of speeding up this analysis, by development of gate level aging model that can be applied into timing analysis engines [5], [6], [10], [11]. Most of them analyze the single or combined influence of NBTI, HCI, TDDB aging effects on the circuit parameter degradation in order to derive the equational form of this

This research is partially funded by The Ministry of Education and Science of Republic of Serbia under a contract no. TR32004.

Digital Object Identifier 10.4316/AECE.2019.03009

dependence. Nevertheless, applying complex equations for gate models in timing analysis machines might not give acceptable analysis speed up for large digital circuits [6], [12], [13], and should be avoided.

Second problem in the DFR strategy is the choice of digital circuit delay analysis methodology. Key tasks of any delay or timing analysers are to check: whether the signals timely arrive at specified outputs, are they stabile long enough to be correctly and timely processed, is their propagation slope correct, and finally, can the circuit run properly for a specified speed. The simplest methodology that can answer all above questions is the simulation. Pspice-based simulations are widely accepted and the most accurate tool. Nevertheless, when dealing with large digital circuits, they are very slow and are mostly avoided. Correct conclusions about all mentioned tasks require simulations for all possible combinations of input vector i.e. exhaustive simulations. This approach can again be very time consuming and still impractical and inapplicable. Alternative approaches are: Static Timing Analysis (STA), Statistical Static Timing Analysis (SSTA), or some kind of accelerated Monte-Carlo analysis (MCA).

A design and manufacturing of a single integrated circuit is a very complex, and time and money consuming procedure. In this industry, it simply does not pay off to iteratively repeat procedures of design, testing and modifying until the final – correct product is achieved. If any of the performances such as consumption, area, speed is not predicted well, the resulting product might not meet the required quality criteria. If a digital system ages, delays in its logic blocks increase. After a long exploitation time, at some point the digital system that was manufactured as a fault-free and fulfilling all specification, might become unacceptably slow and violate its speed limits

The digital circuit aging degradation assessment is already provided in the form of reliability simulator tools within some commercial design environments such as Cadence, Synopsys or ELDO [12]. They offer high accuracy, but the fact that they are based on Pspice simulators makes them time consuming and inefficient for analysis of complex digital circuits. Because of their cost, they are often not available to the designer. On the other hand, an event-based logic simulators based on Verilog or VHDL hardware description languages are very popular and affordable in many forms. Every digital circuit design begins with a logic simulation.

The methodology described here uses a standard VHDL language and a VHDL simulator that initially do not support timing or aging or power analysis. For implementing these features, it is not necessary to develop a new, or buy an expensive reliability simulator. Instead, with the modifications that are suggested in this study, some of these extremely useful features, could be supported within the existing logic simulators. The aim in this study is to establish a novel method that extends the usage of a standard logic i.e. VHDL simulator. It enables quantifying the circuit aging robustness and helps in deciding between different realizations of some digital circuit. Results of the proposed timing analysis are delay degradations along all paths in a digital circuit caused by aging of the gates. Introduction of a new step in the digital circuit design process, that performs delay degradation analysis and assesses the aging robustness of a digital circuit in early design phases is illustrated in Fig. 1.

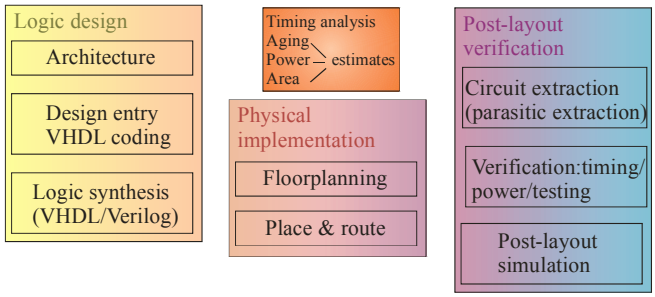

Figure 1. Digital circuits design procedures and aging aware timing analysis step

This paper is structured as follows. The next section gives details about the influence of different aging mechanisms to the device's timing reliability. Third section explains the basic concept of HDL modelling and evaluation of the timing performance degradations in one combinational digital system with a particular attention to the influence of components' aging. In Section V a VHDL implementation of the suggested methodology will be given, followed by the results of our aging analysis applied on a set of ISCAS'89 benchmark circuits with different size and complexity. Concluding remarks are given at the end.

II. Physical Mechanisms of Component Delay Degradation Caused by Aging

As mentioned earlier Negative-Bias-Temperature-Instability NBTI, Hot-Carrier-Injection (HCI), Time-Dependent-Dielectric-Breakdown (TDDB), and the interactions between them are dominant aging mechanisms that affect circuit performances over its lifetime. Aging also varies from place to place on a chip [14].

NBTI represents the most severe aging effect in nowadays CMOS technology. It causes the threshold voltage drift in MOS transistors [15]. NBTI particularly degrades inverted transistors [16]. Transistor's stress rate and signal probabilities define this degradation, but on the other hand, it can also recover totally or partially. Its dependence over time is logarithmic.

Hot-Carrier-Injection (HCI) is a phenomenon of permanent change of the transistor switching characteristics. Hot carriers (electrons or holes) increase their speed and cause degradations in transistors when its conducting condition frequently changes. It results in the increase of the voltage needed to turn the transistor on [15]. As this threshold voltage rises, the transistor switches slower. By measuring the transition density (TD) it is possible to evaluate transistor's stress level. HCI linearly depends on temperature, and exponentially on supply voltage, while its time dependence is logarithmic.

TDDB is another MOSFETs aging mechanism [14] that appears when a gate voltage creates defects in the dielectric i.e. traps that are electrically active. If their number becomes critical, they can create a short circuit between the gate and the channel. This can cause an oxide breakdown, i.e. time-dependent dielectric breakdown. Defects like this appear suddenly, causing catastrophic failure of the transistor.

The aging caused delay degradation analysis begins with the specification of the delay degradation model for each gate type from the gates' library. Data necessary for performing this step should be provided by the IC manufacturer. Nevertheless, creation of realistic technology-based delay degradation model is not a part of this study since it requires time consuming and complex physical and mathematical experiments. Models used in this study are hypothetical and sublime the aging aware delay degradation trends of different logic gates that can be found in the latest researches [7], [10]. Their accuracy can be improved using a proper technology and implementation data.

Another source of delays in modern digital circuits are interconnections i.e. wires. With the advances of IC technologies, they even became dominant cause of delays. Wires can also become aged when being affected by the electromigration phenomenon. Since exploitation of a circuit implies a current flow through metal wires, electrons that create that flow also carry metal atoms along the wire [17]. In particular, electron winds created by high density currents force atoms of metal to slowly migrate over time. When atoms of metal are removed from the metal interconnects, their resistance rises. Supply wires are particularly sensitive to this aging effect. The resulting effect of electromigration on metal wires are open or short circuits that create catastrophic faults [18], [19].

On the other hand, to deal with a wires' delay in digital circuit analysis, a Block-based timing analysis could be applied. It is based on the idea of progressive computation [20], where gates and wires are modelled as a timing block. The disadvantages of this approach during the delay and delay degradation analysis is the fact that the number and sizes of all wires in a digital circuit are available only after the implementation of the entire digital circuit. This makes the methodology hardly applicable in the first design stages of the digital circuit design flow.

III. Timing Degradation Analysis with a VHDL Simulator

Timing analysers are programs that can calculate signal timings i.e. delays of a circuit that consist of primitive gates. These tools can also compute changes in delays caused by the finite ranges of parameter tolerances [21], [22]. STA methods do not need simulations to evaluate timings in digital circuit. Nevertheless, a common logic simulator such as VHDL can also be utilized for rough but helpful early assessment of delays in a digital system in a very efficient way, for small amount of CPU time. In some previous works [21], [23] it was shown that, a logic VHDL-based simulator can be adjusted to perform a timing analysis (TA). This requires some modifications of the simulation mechanism and enables estimation of all delays that propagate through the paths of the circuit structure for rising and falling signal transitions by invoking only one run of the VHDL simulator. The same mechanism facilitates fast Monte-Carlo or SSTA analysis of a combinational circuit [21], [24]. Besides, a VHDL-based simulator can also perform power estimation for a particular stimuli sequence of a digital circuit [25].

In this methodology the only information that signals carry is the delay. Logic values of the signals are mostly irrelevant for this analysis. The simulation-based delay estimation implies one simulation run that concurrently propagate possible input combinations through the circuit. It evaluates all transitions possibilities through that one run, and does not require any kind of stimuli. To encounter a timing degradation caused by aging within the logic gate, the same mechanisms and signal descriptions can be used since the delay degradation represents the same type of data as the delay.

A. Modelling of signals and basic logic blocks

The signal (S) in a digital circuit, carries two delay degradation values that correspond to two transitions of S: d1 (S) – the longest path delay degradation for a rising transition at S, and d0 (S) – the longest path delay degradation for a falling transition at S. A composite VHDL signal type can consist of several signal attributes, and can satisfy this requirement.

To perform the delay degradation estimation and process composite signals, the gate has to be modelled in two levels: level that deals with delay degradation, and level that triggers composite signal propagation. Also, the gate description needs to contain process that calculates the maximal degradation of delay for rising and falling transitions of signal. Triggering the signal propagation in a gate is a process that is initiated by any change of the corresponding signal attribute. When the calculations in the gate activate, the output signal delay degradations are estimated using the gate's input signal delay degradations and the delay degradations introduced by the gate itself. When the final gate output delay degradation is determined (the delay degradation attribute), the triggering attribute of the gate output signal switches in order to initiate same activities in the following gates.

To accumulate the timing delay degradation from inputs of a particular gate, delay degradation variables can be combined using the following operators [22], [26]:

addition: When a delay degradation x reaches the gate's input and propagates through it with a delay degradation y, the resulting output delay degradation variable z can be calculated $z = x+y$.

maximation: If two delay degradations x and y reach at inputs of one gate, a resulting maximal input delay degradation $z = \max(x, y)$ should be calculated first, before the delay degradation of that particular gate is added.

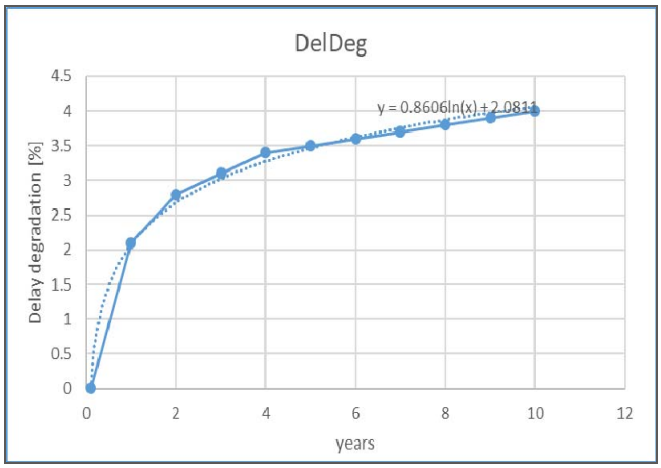

Figure 2. Percentual approximation of the gate delay degradation during years of exploitation

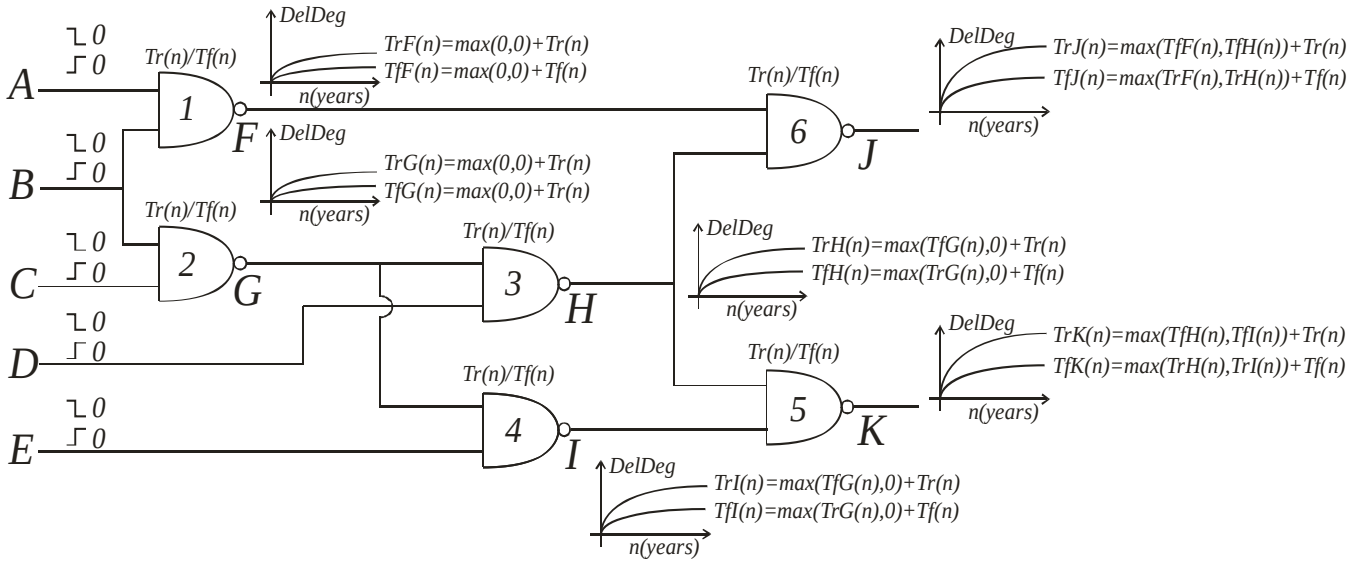

Figure 3. Illustration of the delay degradation evaluation using logic simulator in C17 ISCAS'89 benchmark circuit

The gate delay degradation affected by aging, should be described in the form of a trend that depends on the gate time of exploitation, and should incorporate the real parameters' changes of a particular fabrication technology [22], obtained through the accelerated aging experiments. Also, this trend should be able to adjust according to: the amount of the activity i.e. switching of the particular gate, type of transitions, gate loads and other relevant factors [22], [23]. The logic values that are present at the gate's terminals as well as its logic function are considered during the delay degradation calculation only when propagation of delay degradations depends on them. For instance, when low-to-high transition appears at any of input ports of a two-input NOR gate, this event initiates the high-to-low transition at its output. On the other hand, high-to-low transition at one gate input terminal can initiate a low-to-high transition at the output terminal only if another high-to-low transition had already reached the second input terminal of that gate.

Using delay degradation data given in [3], [10], a simple but satisfying trend approximation for the aging caused delay degradation in logic gates (DelDeg) is constructed and shown in Fig. 2. Delay degradation here is represented in percentages of the initial gate delay, that changes over years of exploitation.

B. The algorithm of fast aging caused path delay degradation analysis

The illustration of delay degradation estimation is shown in Fig. 3. It demonstrates the calculation of maximal delay degradation along each circuit path. The circuit needs to be described at the gate level. It is assumed that maximum delay degradations of all circuit gates for both types of signal transitions in the particular year and the conditions of exploitation are known. In the initialization phase both types of signal transitions are simultaneously applied to all input terminal of the combinational digital circuit in order to trigger delay degradation calculation processes in gates that form the first topological level. This event changes triggering attributes of all primary inputs' signals to "true". Also, both types of transition delay degradations are renewed (altered from the initial zero value) with the propagation through each gate. When all these processes are completed for the first topological level, the same activities in the second topological level is initiated. The gates' delay degradations are dynamically accumulated along paths and dynamically updated just like the propagation of the signal transitions from primary inputs towards primary outputs. The delay degradation estimation of the circuit for a particular year of exploitation is completed when all activations reach circuit's primary outputs [22]. The

```
SET      time of exploitation (in years)
READ     maximal delays for rising and falling
         signal edges for all gates
READ     circuit description
READ     package of functions, technology,
         exploitation and implementation data
SET      composite signal type:
         two delay degradation attributes –
         representing both signal transitions
         two triggering flags – for the
         initialization of gate delay degradation
         calculation
SET      all triggering flags to "false"
SET      all signal delay degradation attributes
         to '0'
SET      triggering flags of input signals in
         order begin the delay degradation
         calculation process
UNTIL    all output signal triggering flags
         become "true"
EXECUTE  following calculations by passing
         through circuit's topological levels
         CHOOSE  the gate from the netlist that is
                 unprocessed
         WAIT FOR        gates input signal
                         triggering flags to be
                         "true"
                 BEGIN   delay degradation
                         calculation process
                  SET    gates output signal delay
                           degradation attributes
         END FOR
         SET     gates output signal triggering
                 flags to "true"
PRINT TO FILE    aging estimation results for
                 output signals of the circuit
```

Figure 4. Performing aging aware delay degradation analysis for particular year and conditions of circuit exploitation

information about the maximal delay degradations for every signal paths of a particular combinational digital system then becomes available, written in a text file.

All phases of the delay degradation estimation process are described as a pseudo code in Fig. 4.

IV. VHDL Implementation

There are two important reasons that support the choice of VHDL as the language and the simulator for implementation of the proposed methodology. First, VHDL can work with composite type of signals. Second, it is widely exploited in digital systems design communities and also supported by many commercial and non-commercial design tools. For processing the data obtained in the analysis, Matlab or Excel could be used and they perform equally well.

```
library ieee;
use ieee.std_logic_1164.all;
use work.aged_logic_pkg.all;
use ieee.std_logic_arith.all;
use IEEE.math_real.all;
entity norg_aged is
        generic (
        age: integer:= 5;
        activity: real:= 1.0;
        tpdhl: real := 0.95;    -- [nanoseconds]
        tpdlh: real := 1.05);   -- [nanoseconds]
        port (o1: out aged_logic := (0.0, 0.0,
        false, false);
        i1, i2: in aged_logic := (0.0, 0.0,
        false, false));
end entity norg_aged;
architecture norg_aged of norg_aged is
begin
        p1: process (i1.d0mx, i1.d1mx, i2.d0mx,
        i2.d1mx, i1.arr0mx, i1.arr1mx,
        i2.arr0mx, i2.arr1mx)
        variable s, t, ys: real;
        variable act_rate : real;
        begin
        act_rate := activity;
        ys := real(age);
             s:=act_rate*(tpdhl*0.01*(0.8606*log
             (ys)/0.4343+2.0811));
             t:=act_rate*(tpdlh*0.01*(0.8606*log
             (ys)/0.4343+ 2.0811));
        if (i1.arr1mx or i2.arr1mx) then
             o1.d0mx   <= max (i1.d1mx, i2.d1mx)
             + s;
             o1.arr0mx <= true;
        end if;
        if (i1.arr0mx and i2.arr0mx) then
             o1.d1mx   <= max (i1.d0mx, i2.d0mx)
             + t;
             o1.arr1mx <= true;
        end if;
        end process p1;
end norg_aged;
```

Figure 5. VHDL implementation of aging aware delay degradation model for NOR logic gate

The VHDL implementation of the aging aware delay degradation trend approximation, required IEEE.math_real package. VHDL description for NOR logic gate that supports delay degradation analysis is given in Fig. 5. The delay degradation analysis process is implemented inside each gate for both signal transitions. The entity description of the NOR gate contains several generics. They are: age, activity, delay degradation of the rising (tpdlh) or falling (tpdhl) transitions. They all have some default value. This value can be later modified in the circuit netlist during the instantiation of the particular gate. Other logic gates are also described in this way. An appropriate library containing all

```
library ieee;
use ieee.std_logic_1164.all;
use work.aged_logic_pkg.all;
use ieee.std_logic_arith.all;
use IEEE.math_real.all;
 entity c17 is
   port(inp:in aged_logic_vector (0 to 4);
   outp:out aged_logic_vector(0 to 1));
 end entity c17;
 architecture gate_level of c17 is
 component nandg_aged
 generic (
       age: integer:= 5;
       activity: real:= 1.0;
       tpdhl: real := 0.95;    --
[nanoseconds]
       tpdlh: real := 1.05);   --
[nanoseconds]
       port (out1: out aged_logic := (0.0,
       0.0, false, false);
       in1, in2: in aged_logic := (0.0, 0.0,
       false, false));
 end component;
 -- Internal signal declarations:
  signal x1: aged_logic := (0.0, 0.0, false,
  false);
  signal x2: aged_logic := (0.0, 0.0, false,
  false);
  signal x3: aged_logic := (0.0, 0.0, false,
  false);
  signal x6: aged_logic := (0.0, 0.0, false,
  false);
  signal x7: aged_logic := (0.0, 0.0, false,
  false);
  signal x10: aged_logic := (0.0, 0.0, false,
  false);
  signal x11: aged_logic := (0.0, 0.0, false,
  false);
  signal x16: aged_logic := (0.0, 0.0, false,
  false);
  signal x19: aged_logic := (0.0, 0.0, false,
  false);
  signal x22: aged_logic := (0.0, 0.0, false,
  false);
  signal x23: aged_logic := (0.0, 0.0, false,
  false);
 begin
 -- netlist:
 g1: nandg_aged
 generic map ( 5, 0.5, 0.95, 1.05)
 port map (x10, x1, x3);
 g2: nandg_aged
 generic map ( 5, 0.5, 0.95, 1.05)
 port map (x11, x3, x6);
 g3: nandg_aged
 generic map ( 5, 0.5, 0.95, 1.05)
 port map (x16, x2, x11);
 g4: nandg_aged
 generic map ( 5, 0.5, 0.95, 1.05)
 port map (x19, x11, x7);
 g5: nandg_aged
 generic map ( 5, 1.0, 0.95, 1.05)
 port map (x22, x10, x16); -- primary out
 g6: nandg_aged
 generic map ( 5, 1.0, 0.95, 1.05)
 port map (x23, x16, x19); -- primary out
 -- Connecting vector inp to netlist inputs:
 s1 <= inp(0);
 s2 <= inp(1);
 s3 <= inp(2);
 s6 <= inp(3);
 s7 <= inp(4);
 -- Connecting vector outp to netlist outputs:
 outp(0) <= x22;
 outp(1) <= s23;
```

Figure 6. VHDL netlist of the ISCAS C17 benchmark circuit, accommodated for the aging aware delay degradation analysis

of them is created. It can be invoked by the netlist of the digital circuit to be analyzed and also enables additional modifications that increase the accuracy of calculations.

The default generic number can be updated with the value that matches the actual conditions in the circuit, depending on the particular circuit activity or activity trend, temperature or temperature trend, delay degradation trend and other long-term exploitation conditions. For this purpose, a special program that manipulates with text (search and replace) have been developed. In this particular study, this program was executed on a group of ISCAS'89 benchmark circuits that have a uniform description and enable easier modifications of the instance's generics.

The simulation and delay degradations analysis are not possible without a circuit netlist. Those used for logic simulations are a good choice. A standard VHDL netlist of a certain circuit needs only a few adjustments in order to be used in delay degradation analysis. The netlist describing ISCAS C17 benchmark circuit is shown in Fig. 6. The circuit is small enough and contains only NAND logic gates which makes it easy to understand. The entity of this gate matches description of the NAND component in the C17 netlist. Signals connecting all gates need to have a specific type - **`aged_logic`**, while generic mapping and port mapping of all six logic gates must correspond to the gates' entity as well as to real implementation in the circuit. Also, a VHDL testbench must also be prepared. Here, each year of the circuit exploitation requires a new instance of the circuit netlist. All aging aware timing analysis results are written in the text file, that can later be processed and studied.

V. Simulation Results and Discussion

As already mentioned, the proposed methodology was verified on ISCAS'89 benchmark combinational circuit. For this experiment all gates have same maximal initial delays: t_F = 0.95ns for the delay of the high-to-low signal transition, and t_R = 1.05ns for the delay of the low-to-high signal transition. The first circuit that was analysed was C17. Results of delay degradation analysis are shown in Fig. 7. This is the most illustrative form of aging analysis representation, but it is inadequate for circuits with large number of output terminals. These graphs represent four types of delay increase in ns: rising transition delay degradation at the output J (circuit in Fig. 3) – first raw of bars, falling transition delay degradation at the output J – second raw of bars, rising transition delay degradation at the output K – third raw of bars, and falling transition delay degradation at the output K – fourth raw of bars.

Using the obtained results, and considering particular delay tolerances specification, the circuit wear-out phase can be estimated. Similar analyses were performed for other ISCAS'89 benchmark circuits. Results of these analysis are organized in Table I. Columns in this table represent the following data: circuit name, number of signals in the circuit, number of circuit inputs/outputs, number of gates in the circuit, maximal topological level, initial delays for the falling (D_{fmx}) and the rising (D_{rmx}) transition of the longest paths for each circuit in ns, delay degradation Δ of the longest paths falling and rising transitions in ns after 5, and after 10 years of circuit exploitation, respectively.

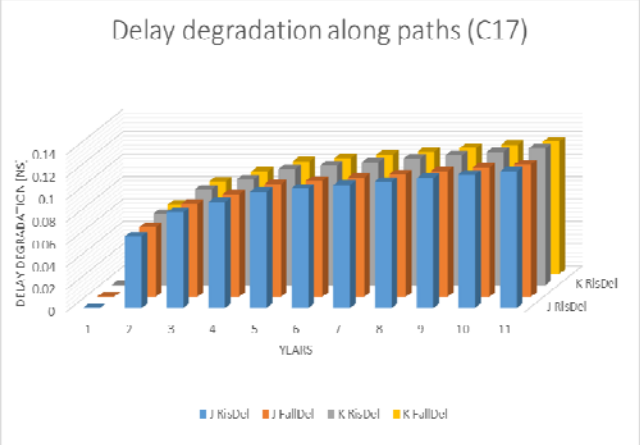

Figure 7. Delay degradations along two paths for two signal transitions for 10 years of C17 exploitation

The switching rate of the gate has a large influence on the delay degradation results. This is analysed for the C432 circuit, and these comparative results are illustrated in Fig. 8. Here, the gates' activity rate α is set to 1 (active 100%), and 0.8 (active 80%), respectively. Delay degradations for α=1 activity are denoted with a prime (’), while those for α=0.8, are denoted with a secundum (‘‘). Results are available for 5 and 10 years of circuit exploitation. The introduced simplifications that enable exploitation of the gates' activities bring a certain inaccuracy in the gate delay degradation model [3], but the aim in this study was to find the way to utilize this information within the proposed procedure.

Table II gives the simulation run times for ISCAS'89 benchmark set. These results show aging aware delay degradation analysis durations, achieved on Intel Xeon processor at 3.5GHz, with 32 GB RAM.

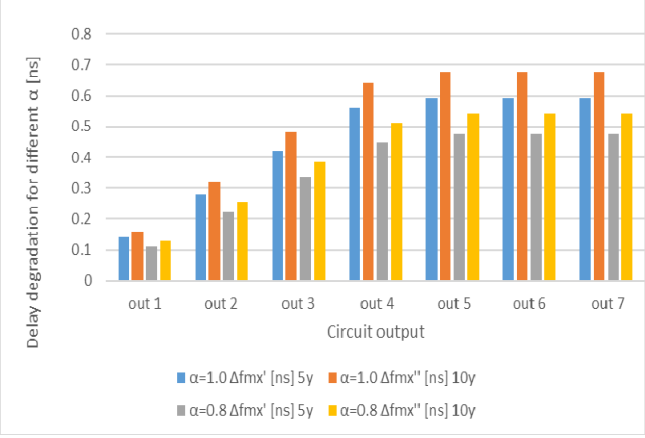

Figure 8. The influence of gates activity to delay degradation along all paths in C432 benchmark circuit

The efficiency of the proposed methodology has been compared with the Monte-Carlo analysis accommodated for the VHDL simulator. The simulation run times for both approaches were selected as the figure of merit for comparison. It should be emphasized that in order to determine largest delay degradation with a standard Monte-Carlo loop, one needs to perform repeated simulations for all possible input combinations, representing every year of circuit exploitation (only 100 Monte-Carlo loop runs were performed in this case, which is far from all possible input

TABLE I. DELAY DEGRADATIONS OF LONGEST PATHS FOR ISCAS'89 BENCHMARK CIRCUITS

Circ. name	Nr. of signals	Nr. of inputs/ outputs	Nr. of gates	Topol. level	D_{fmx} [ns] 0 y's	D_{rmx} [ns] 0 y's	Δ_{fmx} [ns] 5 y's	Δ_{rmx} [ns] 5 y's	Δ_{fmx} [ns] 10 y's	Δ_{rmx} [ns] 10 y's
c17	11	5/2	6	3	2.95	3.05	0.103	0.107	0.103	0.118
c432	196	36/7	160	17	16.95	17.05	0.593	0.597	0.678	0.682
c499	243	41/32	202	11	11.35	11.45	0.397	0.401	0.454	0.458
c880	443	60/26	383	24	24.00	24.20	0.84	0.847	0.96	0.968
c1355	587	41/32	546	24	24.10	23.90	0.843	0.836	0.964	0.956
c2670	1426	233/140	1193	32	32.30	32.40	1.130	1.134	1.292	1.296
c3540	1719	50/22	1669	47	47.55	47.75	1.664	1.671	1.902	1.910
c5315	2485	178/123	2307	49	49.35	48.65	1.727	1.703	1.974	1.946
c6288	2448	32/32	2416	124	124.00	124.00	4.340	4.340	4.960	4.960
c7552	3719	207/108	3512	43	42.95	43.05	1.503	1.507	1.718	1.722

vector combinations). Instead, one run of the proposed methodology covers all possible delay degradation transitions.

Obtained delay degradations for each path in the circuit can be used for a circuit lifetime assessment. When aged circuit exceeds the lowest speed limitation, one can conclude that the wear-out region of the circuit is reached. The result of the methodology presented here is the year when this situation appears. It should be emphasized that only the speed of the circuits i.e. timing variations are analyzed here as the cause of circuit aging. The probability of their occurrence is very high, but they are not the only cause of circuit ware-out. During their manufacturing and exploitation, other catastrophic faults appear in a great extent (stuck-at faults, short or open circuits), but they were not the subject of this research.

The suggested methodology enables the evaluation of circuit behavior during ages of exploitation in the early design stages by avoiding the use of expensive and time-consuming software, instrumentation, sensors, devices and methods for this purpose, as well as early detection of design solution that do not satisfy the required aging robustness.

The research conducted and explained in this paper is limited to large combinational circuits. Nevertheless, they represent only one class of logic circuits. It is possible and very useful for the designers to extend this methodology to other classes of logic gates. Asynchronous and synchronous circuits cover a wide range of digital circuits' applications. Lower power consumption is the key advantage of the asynchronous circuits over the synchronous ones. It would not take much effort to model the delay degradation caused by aging of asynchronous elements in the way similar to the one exploited in combinational gates delay modeling. The same conclusions could be drawn for basic sequential circuits such as flip-flops. When instantiating such elements in a larger sequential circuit, it does not make sense to estimate delay or delay degradation of the entire circuit, since the events at the output of flip-flops are determined only by the appropriate edge of the clock signal. Alternatively, embedded sequential elements allow the application of the suggested methodology, since they must satisfy the expected timing limitations. The signal at the output of the sequential element must be ready and stable before the appearance of the next clock impulse. When flip-flops become aged, this may not be the case.

Another problem that further needs to be studied is the existence of the false paths in digital circuits. The proposed methodology does not take into consideration the possibility of the false path presence in a circuit. Because of that the methodology could produce some pessimistic result of aging caused delay degradation analysis of combinational digital circuit. This is particularly problematic when long false paths are present in the circuit. As an idea for further research and improvement of the described methodology, one can see the development of some procedure that uses a standard logic simulator and is able to efficiently eliminate false paths in the combinational circuit.

TABLE II. SIMULATION RUN TIMES FOR DIFFERENT APPROACHES (PERIOD OF 10 YEARS)

Circuit	Aging aware delay degradation analysis [s]/number of input combinations	100 runs of Monte-Carlo simulation [s]
c17	$0.1 / 2^5$	0.8
c432	$0.1 / 2^{36}$	0.8
c499	$0.1 / 2^{41}$	0.8
c880	$0.1 / 2^{60}$	0.8
c1355	$0.3 / 2^{41}$	2.7
c2670	$0.6 / 2^{233}$	3.1
c3540	$1.2 / 2^{50}$	3.4
c5315	$1.6 / 2^{178}$	4.7
c6288	$6.9 / 2^{32}$	7.8
c7552	$4.1 / 2^{207}$	6.2

VI. CONCLUSION

A VHDL-based aging-aware static timing analysis engine has been proposed in this paper. The presented methodology extracts delay variations of all paths in a digital circuit induced by aging effects in a very efficient way with a single run of a logic simulator, and without any stimuli. The specific aging-aware delay modelling of a single gate is incorporated using approximations from different studies. This gate delay information is exploited by a VHDL-based static timing analyzer in order to estimate the timing degradation of a particular digital system described at a gate level. The obtained delay degradations for each path in the circuit can be used for a circuit lifetime assessment, similarly to a yield calculation. This methodology is applicable in the early design stage, using a standard logic simulator. It also enables an early detection of design solutions that do not satisfy the required aging robustness and avoids the use of slow Pspice simulator, expensive and time-consuming commercial software, instrumentation, sensors, devices and methods for this purpose. Further improvement of the methodology implies the utilization of some statistical techniques in the analysis of the delay degradations along paths of a combinational digital circuit.

REFERENCES

[1] W. Wenping, V. Reddy, B. Yang, V. Balakrishnan, S. Krishnan, and Y. Cao, "Statistical prediction of circuit aging under process variations," in Proc. IEEE Custom Int. Circ. Conf. 2008. pp. 13-16. doi:10.1109/CICC.2008.4672007

[2] S. Duan, B. Halak, and M. Zwolinski, "An ageing-aware digital synthesis approach," In 14th IEEE International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2017, pp. 1-4. doi:10.1109/SMACD.2017.7981556

[3] H. M. Abbas, M. Zwolinski, and B. Halak, "Static aging analysis using 3-dimensional delay library," In ERMAVSS@DATE, 2016, pp. 34-37. https://eprints.soton.ac.uk/id/eprint/385876

[4] P. F. Butzen, V. D. Bem, A. I. Reis, and R. P. Ribas, "Design of CMOS logic gates with enhanced robustness against aging degradation," Microelectronics Reliability, vol. 52, no. 9-10, 2012, pp. 1822-1826. doi:10.1016/j.microrel.2012.06.092

[5] S. Karapetyan, and U. Schlichtmann, "Integrating aging aware timing analysis into a commercial STA tool," In VLSI Design, Automation and Test (VLSI-DAT), April 2015. pp. 1-4. IEEE. doi:10.1109/VLSI-DAT.2015.7114528

[6] A. Hussam, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Reliability-aware design to suppress aging," In 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2016, pp. 1-6. doi:10.1145/2897937.2898082

[7] C. Bestory, F. Marcand, and H. Levi, "Statistical analysis during the reliability simulation," Microelectronics Reliability, vol. 47, no. 9-11, 2007, pp.1353-1357, doi:10.1016/j.microrel.2007.07.079

[8] S. Shaheen, G. Golan, M. Azoulay, and J. Bernstein, "A comparative study of reliability for FINFET," Facta Unversitatis Series: Electronics and Energetics, vol. 31, no 3, September 2018, pp. 343-366. doi:10.2298/FUEE1803343S

[9] B. Li, M. Hashimoto, and U. Schlichtmann, "From process variations to reliability: A survey of timing of digital circuits in the nanometer era," IPSJ Transactions on System LSI Design Methodology, vol. 11, 2018, pp.2-15. doi:10.2197/ipsjtsldm.11.2

[10] N. Koppaetzky, M. Metzdorf, R. Eilers, D. Helms, and W. Nebel, "RT level timing modeling for aging prediction," In Proc. of the Conf. on Design, Automation & Test in Europe, March 2016, pp. 297-300, EDA Consortium.

[11] M. Milić, M. Ljubenović, S. Đošić, and D. Lukač, "Aging aware HDL modelling of delays in logic gates," In Proc. of IcETRAN, Palić, Serbia, June 2018, pp. 896-900.

[12] D. Lorenz, M. Barke, and U Schlichtmann, "Efficiently analyzing the impact of aging effects on large integrated circuits," Microelectronics Reliability, vol. *52*, no. 8, 2012, pp.1546-1552. doi:10.1016/j.microrel.2011.12.029

[13] J. Chen, S. Wang, N. Bidokhti, and M. Tehranipoor, "A framework for fast and accurate critical-reliability paths identification," In IEEE North Atlantic test workshop (NATW), May 2011, pp. 1-4.

[14] J. Keane, and H. K. Chris, "An odomoeter for CPUs," IEEE Spectrum, vol. 48, no. 5, 2011, pp. 28-33. doi:10.1109/MSPEC.2011.5753241

[15] D. Lorenz, M. Barke, and U. Schlichtmann, "Aging analysis at gate and macro cell level," In Proc. of the Int. Conf. on Computer-Aided Design, November 2010, pp. 77-84.

[16] J. H. Stathis, S. Mahapatra, and T. Grasser, "Controversial issues in negative bias temperature instability," Microelectronics Reliability, vol. 81, 2018, pp.244-251. doi:10.1016/j.microrel.2017.12.035

[17] A. W. Strong, E. Y. Wu, R.-P. Vollertsen, J. Sune, G. La Rosa, S. E. Rauch, and T. D. Sullivan, Reliability Wearout Mechanisms in Advanced CMOS Technologies. Series on Microelectronic Systems. IEEE Press, 2009, pp.566.

[18] E. Mizan, Efficient fault tolerance for pipelined structures and its application to superscalar and dataflow machines. Ph.D. thesis, Electrical and Computer Engineering Dept., University of Texas, Austin, 2008.

[19] K. Naghmeh, J.-L. Danger, F. Lozac'h. and S. Guilley, "Predictive aging of reliability of two delay PUFs." In International Conference on Security, Privacy, and Applied Cryptography Engineering Cham, 2016, pp. 213-232. doi:10.1007/978-3-319-49445-6_12

[20] L. Zhang, W. Chen, Y. Hu, C.C. Chen, "Statistical static timing analysis with conditional linear MAX/MIN approximation and extended canonical timing model," IEEE Trans. on CAD Integr. Circuits Syst. vol. 25, no. 6, 2006, pp:1183–1191. doi: 10.1109/TCAD.2005.855979

[21] M. Sokolović, V. Litovski, and M. Zwolinski, "New concepts of worst-case delay and yield estimation in asynchronous VLSI circuits," Microelectronics Reliability, vol. 49, no. 2, 2009, pp.186-198. doi:10.1016/j.microrel.2008.11.007

[22] R. Spence, and R. Soin R, Tolerance design of electronic circuits, Addison-Wesley, 1988.

[23] D. M. Maksimovic, and V. B. Litovski, "Tuning logic simulators for timing analysis," Electronics Letters, vol. 35, *no.* 10, 1999, pp.800-802, doi: 10.1049/el:19990545

[24] M. Sokolović, V. Litovski, and M. Zwolinski, "Efficient and realistic statistical worst-case delay computation using VHDL," Electrical Engineering, vol. 91, no. 4-5, 2009, pp.197. doi:10.1007/s00202-009-0133-y

[25] B. S. Kirei, V. J. M. Chereja, S. Hintea, and M. D. Topa, "PAELib: A VHDL library for area and power dissipation estimation of CMOS logic circuits," Advances in Electrical and Computer Engineering, vol. 19. no. 1, 2019, pp. 9-16. doi:10.4316/AECE.2019.01002

[26] A. Agarwal, V. Zolotov, and D. Blaauw, "Statistical timing analysis using bounds and selective enumeration," IEEE Trans. CAD Integr. Circuits Syst., vol. 22, no. 9, 2003, pp. 1243–1260. doi:10.1109/TCAD.2003.816217