

# Hayat: Harnessing Dark Silicon and Variability for Aging Deceleration and Balancing

Dennis Gnad, Muhammad Shafique, Florian Kriebel, Semeen Rehman, Duo Sun, Jörg Henkel  
Chair for Embedded Systems, Karlsruhe Institute of Technology, Germany  
Corresponding Author: muhammad.shafique@kit.edu

**Abstract**—Elevated power densities result in the so-called Dark Silicon constraint that prohibits simultaneous activation of all the cores in an on-chip system (in the full performance mode) to respect the safe thermal limits, thus enforcing a significant amount of on-chip resources to stay ‘dark’ (i.e., power-gated). In this paper, we show that how Dark Silicon together with the manufacturing process induced variability can be harnessed to mitigate reliability threats in the nano-era. In particular, we propose a run-time system Hayat\* that harnesses Dark Silicon to decelerate and/or balance temperature-dependent aging, while also considering variability in order to improve the overall system performance for a given lifetime. Experimental evaluation across a range of chips to account for process variations illustrates that our Hayat system can provide a significant aging/performance improvement and decelerates the chip aging by 6 months – 5 years (depending upon the required lifetime constraint) compared to state-of-the-art techniques.

**Keywords:** Dark Silicon, Reliability, Aging, Soft Error, Temperature, Optimization, Multi-Core, Process Variations.  
\*Hayat means *Life* <http://en.wikipedia.org/wiki/Hayat>. In our case, it is the prolonged life of a chip through decelerated aging.

## I. INTRODUCTION AND RELATED WORK

The breakdown of the *Dennard Scaling model* in the nano-era have resulted in elevated power-densities that can no longer be fully dissipated through cost-effective cooling techniques. This leads to the so-called *Dark Silicon* problem that restricts the maximum number of cores to be simultaneously switched-on at the *nominal voltage* (or at maximum performance level) under a Thermal Design Power (*TDP*) budget, thus leaving a significant chip fraction to be ‘dark’ (i.e., power-gated) [1, 2]. This is important to ensure a thermal-safe operation, i.e., the peak temperature ( $T_{peak}$ ) does not exceed the safe-operating temperature ( $T_{safe}$ ), otherwise the dynamic thermal management (DTM) is triggered.

Recent studies in [3] have presented the revised prediction trends for Dark Silicon considering the technology scaling data from ITRS [4] and various advanced processor features like DVFS. According to [3], on average, 13%, 16%, and > 40% of the chip area will stay dark in the 16, 11, and 8nm technology nodes, respectively. Although dark silicon can be seen as a problem, recent research trends have leveraged dark silicon to improve performance and soft error resilience [5, 6]. A survey of such techniques can be found in [1, 2].

In this paper, we target the following key research challenge: *if and how can the ‘potentially dark’ cores still be harnessed to improve the aging of on-chip systems, that has emerged as one of the most critical reliability threats, within the  $T_{peak}$  constraint while also accounting for the manufacturing process induced variability?*

In the following, we briefly discuss these problems and state-of-the-art.

**Negative Biased Temperature Instability (NBTI)-induced Aging** is one of the most critical aging threats and it is caused by the stress in the PMOS transistors ( $V_{gs} = -V_{dd}$ ) that leads to threshold voltage

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
DAC’15, June 07–11, 2015, San Francisco, CA, USA  
Copyright 2015 ACM 978-1-4503-3520-1/15/06...\$15.00.  
<http://dx.doi.org/10.1145/2744769.2744849>

shift (by an amount  $\Delta V_{th} \geq 50mV$  [7]) and slows-down the transistors (and consequently the critical path) [8]. This is called *short-term aging*, which is partially reversed in case the stress is released (at  $V_{gs} = 0$ ). Since 100% recovery is not possible, the circuit’s delay continuously increases over years which is called *long-term aging*; see Fig. 1(a). A significant work has been done in developing so-called aging/wearout sensors/monitors to detect delay degradation at run-time [9, 10].

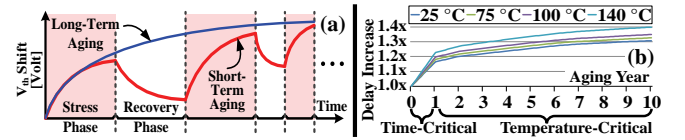


Fig. 1: (a) An Abstract Depiction of Short- and Long-Term Aging. (b) Temperature-Dependent Increase in Aging of a LEON3 processor (synthesized for 45 nm).

To ensure timing-safe operations throughout the chip lifetime, designers provide significant timing guardbands<sup>1</sup> that lead to a loss in the maximum achievable frequency by a factor  $\Delta f$  ( $\geq 20\%$  over its lifetime) [11, 14, 15]. These guardbands are further aggravated in the presence of *process variations* that results in chip-to-chip and core-to-core frequency and leakage power variations. Fig 2(o) shows cores’ frequency variations for two example process variation maps (see model in Section III). The guardbanding can either be (1) at the chip level, i.e., all cores execute at the same reduced safe-operating frequency, but this leads to a significant performance drop; or (2) at the core level, i.e., each core may execute at its reduced safe-operating frequency, but it requires core-level dynamic frequency scaling support [11, 15]. This paper considers the later case due to long-term aging and the cost/power/performance constraints. Chip-level performance degradation can no longer be addressed by simply employing design-time guardbanding [12, 16]. A class of related work targets aging-aware application scheduling [17], but such techniques do not account for temperature and process variations. The FaceLift approach in [11] reduces aging through chip-wide changes to  $V_{dd}$  and performs workload-aware task scheduling considering process variations. However, this approach does not account for dark silicon and its influence on the chip thermal profile. Especially, the spatial temperature influence from the neighboring cores may significantly affect the aging rate of other cores (see aging analysis in Section II). In the following, we discuss how temperature influences the aging rates.

**Influence of Temperature on the NBTI-induced Aging:** In manycore systems, simultaneous powering-on of several adjacent cores for executing massively parallel workloads and/or the performance-boosting techniques like Intel’s Turbo Boost [21] lead to elevated temperatures that further aggravate the NBTI-induced aging, as also shown in the Fig 1(b) (see details on aging estimation in Section IV-B), and directly affect the reliability/lifetime. For instance, a difference between  $10^\circ\text{C}$ – $15^\circ\text{C}$  can result in a 2x difference in the mean-time-to-failure of the devices [22]. Improving thermal profiles leads to an increased chip lifetime.

**In Summary,** *state-of-the-art have not yet exploited the run-time optimization and interplay of dark silicon, on-chip temperature variations, and manufacturing variations to jointly decelerate and balance NBTI-induced aging in manycore systems.* In this paper, we show how dark silicon, temperature and process variations impact the aging (see analysis in Section II) and how these factors can be synergistically leveraged to

<sup>1</sup>Current processors include guardbands for 7-10 years [11–13].

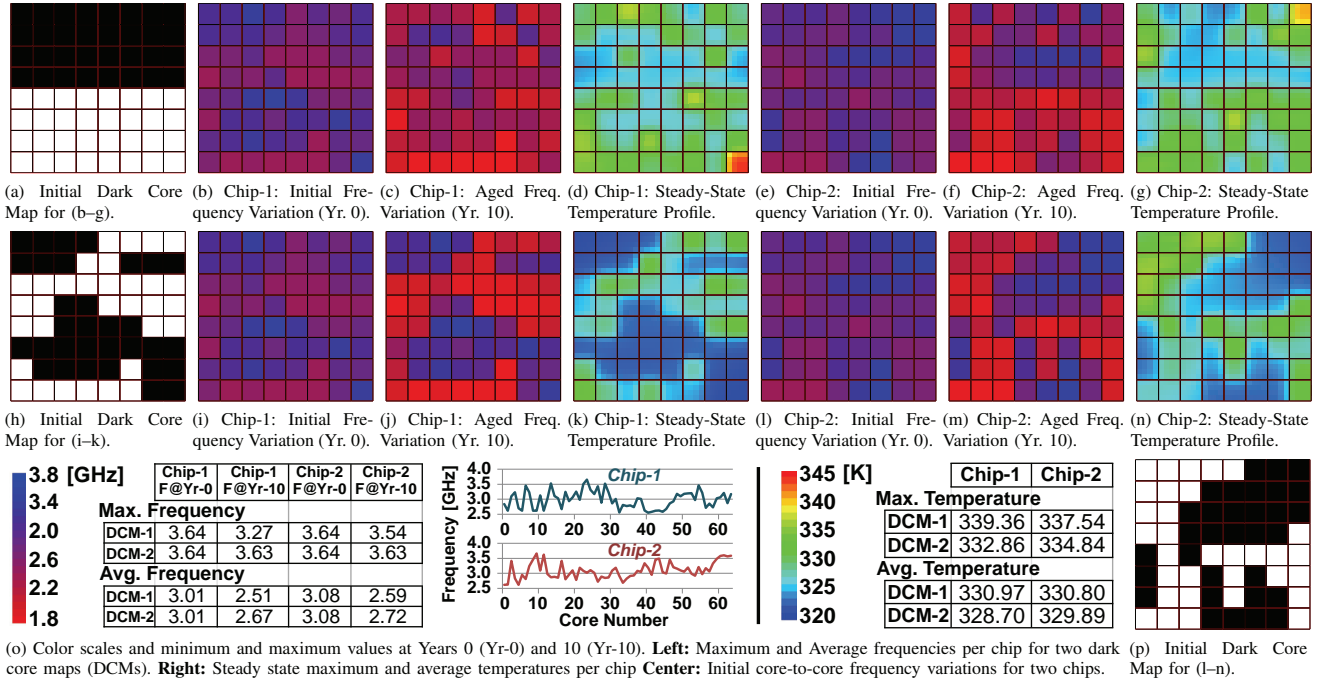


Fig. 2: Aging and Thermal Analysis for different *Dark Core Maps* for two chips with process variations and 50% Dark Silicon. **Setup:** 8x8 Alpha 21264 with 2MB L2 cache, 3GHz Nominal Freq., 1.13V, 22nm data scaled to 11nm as per ITRS-factors [4] to reflect dark silicon, McPAT v1.1 [18], Size of single core:  $1.70 \times 1.75mm^2$ , Gem5 [19], Hotspot [20], Multi-threaded applications of Parsec (“bodytrackhigh”, “x264” with 5 HD-sequences). See further details in Section IV-B and V.

decelerate the chip aging and to achieve a high performance during the desired system lifetime (see optimization in Section IV).

#### A. Definitions

**Dark Core Map (DCM)** is defined as the core power state map with a sub-set of cores being kept ‘dark’ such that  $T_{peak} < T_{safe}$ .

**Health of a Core**  $i$  at time  $t > 0$  is defined as its maximum safe-operating frequency ( $f_{max,i,t}$ ) normalized to the initial variation-dependent maximum frequency ( $f_{max,i,init}$ ). It is measured by health monitors and is given as  $f_{max,i,t}/f_{max,i,init}$ . High aging rates lead to health degradation, i.e., high  $f_{max,i}$  degradation.

**Health Map** is defined as the map of health of all cores in a chip.

#### B. Our Novel Contributions and Concept Overview

To address the above-discussed challenges, in this paper, we introduce a **novel run-time aging management system Hayat** (Section IV) that harnesses dark silicon and variability for aging optimization in on-chip manycore systems. It selects a subset of cores to meet the throughput requirements of concurrently executing multi-threaded applications, while minimizing the overall chip aging rate. To proactively achieve this, our *Hayat* system determines (1) an appropriate *Dark Core Map (DCM)* that decelerates the chip aging through improved heat dissipation due to dark cores; and (2) performs *variation-aware thread-to-core mapping* to achieve balanced aging by scheduling high temperature-generating threads to the *healthy cores* and vice versa while meeting application’s throughput constraints. At different time instances, it continuously optimizes DCMs to conceal faster aging rates under heavy workload scenarios while keeping  $T_{peak} < T_{safe}$  and accounting for frequency/leakage power variations. Our experimental evaluation shows that our *Hayat* system outperforms state-of-the-art aging optimization techniques.

To realize proactive aging optimization, *Hayat* requires a **lightweight technique for online estimation of the chip’s Health Map** (Section IV-B). For a candidate aging optimization solution, *Hayat* estimates temperature-dependent health degradation by exploiting (1) online predicted chip thermal profile and duty cycles, and (2) 3D-aging tables

generated offline using precise SPICE simulations and critical path analysis for various temperature and duty cycle settings.

For an efficient design, it is important to understand *how dark cores and process variations may affect the aging profile*. Towards this end, we perform a comprehensive **aging analysis for different DCMs under process variations** (Section II).

## II. AGING ANALYSIS OF DARK SILICON CHIPS

For our aging analysis, we simulated two chips with different process variation maps and DCMs for a set of multithreaded workloads at 50% dark silicon with different throughput constraints. Each core can run at its safe-operating frequency. In case of thermal hot spots at run time, the corresponding tasks are migrated to the coldest cores.

Figs. 2 (a-g) illustrate *contiguous* DCM and the corresponding aging maps at year 0 and 10, and the steady-state temperature for the two chips. Figs. 2 (h-n, p) illustrate *variation-dependent temperature-optimizing* DCMs and the corresponding aging maps at year 0 and 10, and the steady-state temperature for the two chips. Due to frequency and leakage power variations, the DCMs are different for two chips; see Figs. 2 (h and p). We estimate the NBTI-induced aging as a function of workload-dependent duty cycle and temperature profiles (see model details in Section IV-B). Fig. 2 shows that a better DCM is determined by the variation and temperature-aware policy that leads to an improved chip aging profile. Detailed frequency and temperature values are shown in Fig. 2(o). Note, running a dense *contiguous* DCM leads to high  $T_{peak}$  and consequently an increased number of task migrations, which can be inferred by steady state temperature in Fig. 2(d,g). In case of *variation-dependent temperature-optimizing DCMs*, less migrations and reduced temperatures were observed leading to improved aging profiles.

Additionally, as a secondary effect, migrating to cores selected only by temperature can lead to frequency degradation of cores that should better be *saved for later*. That is, it may be beneficial to not to age some of the high-frequency cores (if possible considering tasks’ deadline) as they should only be used to fulfill the deadline constraints of a critical

(single-threaded) application. This essentially makes “core selection” a more *heterogenous* choice, which is dependent on the tasks’ demand for different scheduling policies.

Our analysis shows that aging management in Dark Silicon chips need to synergistically account for DCMs, chip thermal profile, and workload management, while accounting for process variation and temperature-dependent leakage increase.

### III. SYSTEM MODELS

**Processor Model:** The manycore processor  $\mathcal{C} = \{C_1, C_2, \dots, C_N\}$  consists of  $N$  homogeneous cores. Each core  $C_i \in \mathcal{C}$  has its private L1-instruction and data caches, and shared L2 cache. In this paper we focus only on the aging management of cores and assume fixed area/power budgets for the uncore components. Each core  $C_i$  has at least one (soft) thermal sensor  $T_i$  and aging sensor  $D_i$  (like [9, 10]) to monitor its current temperature and health level (i.e., age in terms of delay), respectively. If  $C_i$  is ‘dark’, its power state  $ps_i$  is set to 0, else 1. The total number of powered-on and dark cores is given as:  $N_{on} = \sum_{i=1}^N ps_i$  and  $N_{off} = N - N_{on}$ , respectively. Considering core-level guardbanding against timing errors, each core  $C_i$  has a maximum safe frequency  $f_{i,max}$  under a chip-level voltage  $V_{dd}$  constraint, thus leading to  $p_{i,max}$  as its maximum total power consumption. Due to process variations, frequency and leakage power values vary for different cores while NBTI-induced aging degrades the core’s frequency depending upon its duty cycle and temperature; these factors will be refined below.

**Application Program Model:** A set of  $M$  multi-threaded applications  $\mathcal{A} = \{A_1, A_2, \dots, A_M\}$  executes on the processor  $\mathcal{C}$ , such that  $A_j = \{\tau_{(j,1)}, \tau_{(j,2)}, \dots, \tau_{(j,K_j)}\}$ , where  $K_j$  is the number of threads of  $A_j$ . Each thread  $\tau_{(i,j,k)}$  executes on its dedicated core  $C_i$  and requires a minimum frequency  $f_{\tau,min}$  to meet its throughput or deadline requirements. Considering the malleable application model [23, 24], the value of  $K_j$  can vary depending upon the value of  $N_{on}$ , thus providing a varying degree of parallelism. The duty cycle, current throughput (measured in Instructions-Per-Second: IPS), and power consumption of a thread  $\tau_{(i,j,k)}$  are given as  $d_{(i,j,k)}$ ,  $IPS_{(i,j,k)}$ , and  $p_{(i,j,k)}$ , respectively. The thread-to-core mapping function is given as:

$$m_{(i,j,k)} = \begin{cases} 1, & \text{if thread } \tau_{(i,j,k)} \text{ is executing on core } C_i \\ 0, & \text{otherwise} \end{cases}$$

**Process Variation Model:** We consider both core-to-core leakage power and frequency variations. In this paper, we deploy the existing experimentally validated variation model of [25, 26] that partitions the chip area into  $N_{chip} \times N_{chip}$  grid points overlaid over cores. A process parameter  $\vartheta_{u,v}$  (modeled as a Gaussian random variable) is associated to each grid point  $(u, v) \in [1, N_{chip}]^2$ . The mean, standard deviation, and spatial correlation of  $\vartheta_{u,v}$  are given as  $\mu_\vartheta$ ,  $\sigma_\vartheta$ , and  $\rho$ , respectively (see further details in [25, 26]). The process variations affect the threshold voltage ( $V_{th}$ ), thus leading to a linear impact on the frequency and an exponential impact on the leakage power. The maximum frequency of a core  $C_i$  (with critical path  $CP(C_i)$ ) is determined using Eq. 1.

$$f_i = \alpha \times \min_{x,y \in S_{CP(C_i)}} (1/\vartheta_{x,y}) \quad (1)$$

where:  $\alpha$  is the technology dependent constant and  $S_{CP(C_i)}$  is the set of grid points overlaid on the critical path. The total power consumption of a core  $C_i$  executing thread  $\tau_{(i,j,k)}$  is computed using Eq. 2.

$$P_{(i,j,k)} = p_{(i,j,k)}^{dyn} + \sum_{(u,v) \in C_i} p_{u,v}^{leak} \times e^{V_{th} \vartheta_{u,v} / V_T} \quad (2)$$

$p_{(i,j,k,f)}^{dyn}$  is the dynamic power consumption when executing thread  $\tau_{(i,j,k)}$  of application  $A_j$  on core  $C_i$  at frequency  $f$ .  $p_{(u,v)}^{leak}$  is the nominal leakage power consumption of grid point  $(u, v)$ .  $V_T = KT_i/q$  is the thermal voltage ( $T_i$  is the temperature of core  $C_i$ ) that also captures the temperature dependence of leakage power.

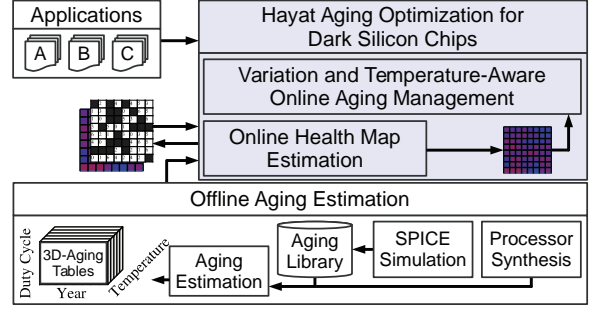


Fig. 3: Overview of our Hayat System.

### IV. HAYAT: AGING MANAGEMENT FOR DARK SILICON CHIPS

Fig. 3 presents an overview of our Hayat system. It performs two key operations (discussed in detail in the subsequent sections). First, it performs *variation and temperature-aware online aging management* that determines appropriate Dark Core Maps (DCMs) and thread-to-core mapping such that overall chip aging is decelerated. It leverages dark cores to achieve lower  $T_{peak}$  and current chip’s health map for core allocation. To find a chip’s health map optimizing solution, health degradation needs to be evaluated for every candidate solution at run time in our optimization algorithm (Section IV-C). Therefore, Hayat employs a *light-weight online health estimation* technique that avoids online aging simulations through offline generated aging-induced frequency degradation tables for various possible temperature and duty cycle combinations. At run time, it estimates health degradation of different cores based on offline generated temperature and duty cycle dependent aging-tables, current health state of the cores, and online predicted chip’s thermal profile for a candidate solution.

Since chip aging is a long-term phenomenon (i.e., over years), we introduce the notion of “aging epoch” (see Fig. 4) in order to take different runtime effects into consideration. This is also important for accelerated aging evaluation. A manycore system experiences different workloads with varying duty cycles and frequency levels, which lead to continuously changing stress levels and thermal variations on different cores. This can further lead to different DTM events triggering task migration and frequency/voltage throttling to avoid thermal emergencies. These variations would introduce randomness in the system state, which makes it difficult to evaluate long-duration cycle-accurate simulations with precise capturing of these effects. Furthermore, since aging happens over several years, such long-term simulations are not possible in a realistic amount of time.

Hence, we define coarse-grained *aging epochs* that determine the granularity of our health monitoring and aging evaluation. Further, we use fine-grained transient simulations during each epoch (see Fig. 4). For each epoch, we consider a certain measured health and variation from the health monitors, i.e.,  $f_{i,max} \forall C_i$ . For experiments, we use our aging-models. We then estimate the next health together with fine-grained transient simulations for thermal and duty-cycle characteristics within an epoch. This data is leveraged to take different DCM and task mapping decisions during an epoch. After an epoch is finished, two steps are performed: (1) the data from the fine-grained simulation is upscaled to the time range of the epoch, and (2) the next epoch starts considering the same set of workloads (or potentially a different one, given multiple sets of workloads).

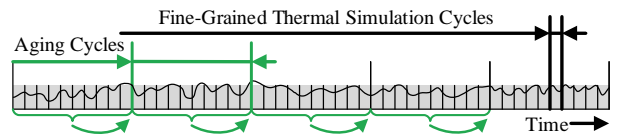


Fig. 4: Accelerated Aging Evaluation: Temperature and Duty Cycle across one aging epoch will affect the next one.

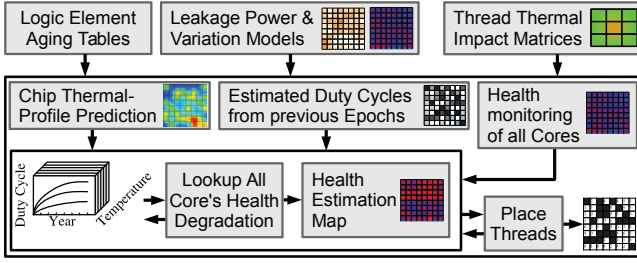


Fig. 5: Detailed flow of our online health estimation technique.

### A. Problem Formulation

**Find** an aging-minimizing *joint* patterning and mapping (Eq. 3):

$$m_{i,j,k} \forall P_j \in \mathcal{P}, \tau_{j,k} \in P_j, s.t., C_i \in \mathcal{C} \text{ executes } \tau_{j,k} \quad (3)$$

That fulfills the **constraints** to stay below thermally safe runtime temperature (Eq. 4):

$$T_i < T_{safe} \forall C_i \in \mathcal{C} \quad (4)$$

and each core  $C_i \in \mathcal{C}$  executes only one thread  $\tau_{i,j,k}$  (Eq. 5):

$$\sum_{j=1}^M \sum_{k=1}^{K_j} m_{i,j,k}; \forall i \in [1, N] \leq 1 \quad (5)$$

While at the same time *maximizing* for the **goal** to increase the sum of future healths'  $H_{i,next}$  average over all cores in  $\mathcal{C}$  (Eq. 6):

$$MAX \left( \sum_{j=1}^M H_{i,next} \right) \forall C_i \in \mathcal{C} \quad (6)$$

The problem can be formulated as an Integer Linear Programming (ILP) problem, but it is not feasible to be evaluated at run time in polynomial time complexity.

### B. Online Estimation of Temperature-Dependent Health Degradation

Fig. 5 illustrates the procedure of estimation of health degradation of all cores experiencing different temperature and duty cycles due to varying thread-to-core mapping, thermal conductance from the neighboring cores, thread's workloads, leakage power variations, and leakage-dependent temperature increase, and thread migration effects. The key steps of our estimation technique are discussed below.

**(1) Generating Aging Tables:** First, we build a library of aging estimates for different logic element (like *NOR*, *NOT*, memory elements, etc.) using an accurate in-house ngspice-23 based aging estimator, which is *developed together with our industrial partner*. It models aging using Equation 7 and is based on the measured sample data for various technology nodes (e.g., 65nm–22nm) following the reaction-diffusion theory [8]. It requires device parameters from cell library data sheets, load capacitance, temperature, etc.

$$\Delta V_{th} = 0.05 \cdot e^{-1500/T} \cdot V_{dd}^4 \cdot y^{1/6} \cdot d^{1/6} \quad (7)$$

$\Delta V_{th}$  is the mean  $V_{th}$  shift in Volts,  $T$  is the temperature in Kelvin,  $V_{dd}$  is the supply voltage in Volts,  $y$  is the transistor's age in years, and  $d$  is the duty cycle denoting stress on the transistor.

Afterwards, a set  $\mathcal{P}(C_i) = \{cp_{(i,1)}, cp_{(i,2)}, \dots, cp_{(i,X_i)}\}$ , with  $X_i$  being the number of top- $x\%$  critical paths of  $C_i$ , is obtained after the hardware synthesis (e.g., using *Synopsis Design Compiler*). The signal probabilities through gate-level simulations (e.g., using *ModelSim*) are obtained to get the duty cycles for different logic elements of the critical paths  $\mathcal{P}(C_i)$ .

For a given core  $C_i$ , our core-level aging estimator then estimates the degradation of critical paths  $\mathcal{P}(C_i)$  over 10 years using Equation 8.

<sup>2</sup>The parameter  $x$  trades off between coverage and analysis time.

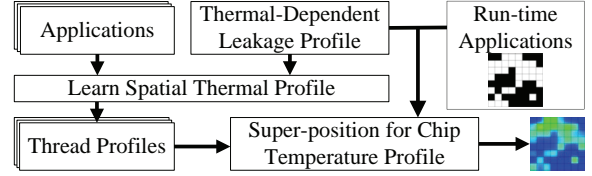


Fig. 6: Flow of our online thermal profiling and estimation.

$$\Delta D(cp_{(i,j)}) = \sum_{\forall le \in cp_{(i,j)}} (D(le) + \Delta D(le, d, T, y)) \quad (8)$$

$D(le)$  is the un-aged delay of the logic element,  $\Delta D(le)$  is the delay degradation which is proportional to its  $\Delta V_{th}$ . We generate 3D-aging tables using different temperature and duty cycle values for all cores. Note that this is only a start-up time effort for a given chip. Some aging results are already shown in Section II.

**(2) Predict Chip's Thermal Profile Considering Spatial Thermal Influence and Leakage-Dependent Temperature Increase:** For this, we employ our online chip thermal predictor [27]. Here, we present a short overview. Our technique operates in two main steps: (1) *Offline learning of spatial thermal profiles* for different application threads, and (2) *Online prediction of chip thermal profile* by super-positioning offline-generated thermal profiles for different applications along with a correction for temperature-dependent leakage. Fig. 6 shows the overview of this scheme. We take thermal-dependent leakage data as an input, as well as applications that are executed to obtain spatial thermal profiles. During run time, our technique performs super-positioning of thermal profiles of the concurrently executing applications' threads in order to generate a chip-level thermal profile prediction. At the same time, we also account for a factor that incorporates temperature-dependent leakage increase of a core due to neighboring cores' temperature influence. Note, for estimating the health degradation, temperature is an input and any temperature prediction technique can be employed here.

**(3) Estimating Health Degradation and New Health Map:** We record the worst-case temperature over time and the duty cycle (0%...100%) for each core. Depending on application-analysis, the core-level duty cycle is multiplied with the worst- or average-case duty cycle of a typical application mix with respect to the PMOS transistors in our processor pipeline model. Together with per-core delay monitoring, we can then find the current estimated position/index in the 3D-aging tables from Step-(1) for each core in order to obtain the current *chip health map*.

In order to obtain the estimated *degraded health map*, we perform the following steps. First, from the positions of each core, we can follow a new 3D-path inside the table depending upon our temperature-prediction from Step-(2) and re-estimate the duty cycle using the core-level duty cycle and applications duty cycles. Afterwards, together with the length of our epoch we can then obtain the estimated *degraded health map*.

### C. Variation and Dark Silicon Aware Aging Management

Algorithm 1 presents the heuristic for decelerating NBTI-induced aging under thermal and process variations. It operates in two main steps. First as set of potential solutions is prepared in form of a list  $S$  (line 2), which is then sorted w.r.t. different weighting. The algorithm iterates through this list of all runnable threads of all programs (starting line 4). For each candidate cores  $C_{cand.} \in \mathcal{C}$  (line 5) this thread can use (i.e., fulfills the speed requirements), it evaluates the possible outcomes in terms of temperature and health (line 7–19).

First per-core short-term temperature prediction is performed (line 8) to get the cores future predicted temperature  $T_{i,next}$ , which might only be required for cores that are affected by choosing  $C_{cand.}$ , and might therefore not need to be evaluated for all cores. If the temperature is not within the safe limits, the algorithm discards this candidate (line 12–13), otherwise it estimates the future (e.g., 1 year) health of each cores  $H_{i,next}$

based on  $T_{i,next}$  and the runtime of one *aging epoch*; as discussed in Section IV-B. The duty cycle can be set with either a *generic* (i.e., 50%), *known* (estimated from offline data by an available netlist), or *worst-case* (85 – 100%) at our predicted temperature. Afterwards, the chip’s health profile is estimated using the lookup process discussed in Section IV-B (line 15). Average values for health ( $H_{avg,next}$ ) and temperature ( $T_{avg,next}$ ) are determined and pushed as a candidate solution to our list  $S$  (line 17–19). Finally, the algorithm sorts  $S$  according to a *weighting function* (will be shortly explained below) and assign the currently processed thread the best candidate (line 22–23). The algorithm then continues with the next thread in the loop.

---

**Algorithm 1** Hayat

---

```

1:  $\mathcal{M}' \leftarrow \mathcal{M}$ 
2:  $S \leftarrow$  new list of candidate structs with different attributes
3:  $L \leftarrow$  list of all cores, sorted by health( $H$ )
4: for all threads  $\tau_j$  of all runnable applications  $A_k$  do
5:   for all  $C_{cand.} \in \mathcal{C}$  do
6:      $\mathbf{m}'_{h,j,k} \leftarrow 1$ 
7:     for all  $C_i \in \mathcal{C}$  do
8:        $T_{i,next} \leftarrow \text{predictTemperature}(C_i, \tau_j)$ 
9:       if  $T_{max,next} < T_{i,next}$  then
10:         $T_{max,next} \leftarrow T_{i,next}$ 
11:       end if
12:       if  $T_{i,next} > T_{safe}$  then
13:        skip this  $C_{cand.}$  and continue with next  $C_{cand.}$ 
14:       end if
15:        $H_{i,next} \leftarrow \text{estimateNextHealth}(C_i, T_{i,next}, \tau_j)$ 
16:     end for
17:      $T_{avg,next} \leftarrow (\sum_{i=0}^N T_{i,next}) / N$ 
18:      $H_{avg,next} \leftarrow (\sum_{i=0}^N H_{i,next}) / N$ 
19:      $S.\text{push}(H_{avg,next}, H_{candidate,next}, T_{avg}, T_{max}, \text{Position})$ 
20:      $\mathbf{m}'_{i,j,k} \leftarrow 0$ 
21:   end for
22:    $S.\text{sort-by}(\text{weight-function } w)$ 
23:    $\mathbf{m}_{S.\text{front} \rightarrow Pos,j,k} \leftarrow 1$ 
24: end for

```

---

*Discussion on the Weighting Function:* As shown in Fig. 1, there is a time-/duty cycle-critical early-aging part and a temperature-critical late-aging. We empirically formulated Eq. 9 as the total aging weight considering two weighting coefficients, that are biased by  $\alpha$  and  $\beta$ , depending on late- or early-aging.

$$w = \max \left( w_{max}, \frac{\alpha}{f_{max,i,t} - f_{req}} \right) + \beta \frac{H_{candidate,next}}{H_{candidate,t}} \quad (9)$$

A higher weight means a higher chance to get the selected choice  $S.\text{front}$ . In early-aging phase, Hayat will try to balance more cores with a frequency  $f_{max,i,t}$  (at a time instance  $t$ ) only as required by the application threads’ minimum required frequency  $f_{req}$  that gets asymptotically close to  $f_{max,i,t}$ , but is limited to a certain maximum weight  $w_{max}$ . The second part is the candidates next estimated health versus the current one. The coefficients are discussed in Section V.

### V. EXPERIMENTAL SETUP

Our setup is based on power and performance traces obtained through cycle-accurate simulations from integrated closed-loop Gem5 [19] and McPAT [18] simulations. We generated several mixes using the multithreaded applications from the *Parsec* benchmark suite. We additionally derived throughput constraints for these tasks as a function of the minimum required frequency they need to run on. With these traces a large-scale chip can be simulated in a reasonable time, while preserving the accuracy of cycle-accurate simulations. To obtain both average leakage power and worst-case delay for frequency variation, numerous  $V_{th}$  process variation maps are generated based on the models from Section III and overlaid on our chip’s floorplan. For instance, we reach

a frequency variation of about 30%-35% at 1.13V, 3-4GHz. To enable closed-loop thermal simulations, this simulator is integrated with the Hotspot [20] tool as a library. As with this transient thermal simulation, a maximum safe temperature  $T_{safe}$  (here we use  $95^\circ\text{C}$  as adopted in Intel mobile *i5*) might be reached, DTM will migrate threads from the hottest cores  $\geq T_{safe}$  to the coldest cores, if they are within  $T_{safe} - 10^\circ\text{C}$ , or throttle them if this is not possible. Detailed simulator parameters and settings are already provided in Fig. 2. Additional parameters are: the nominal subthreshold leakage of 1.18W per core and remaining leakage of 0.019W in power-gated mode. Besides the hotspot model, we apply a temperature dependent leakage as implemented in the *McPAT* simulator [18], as an estimate of temperature dependent leakage after a given time-period (6.6ms in our experiments). This is applied on the variation-dependent leakage power to obtain the total leakage power. Our NBTI models are based on a 45nm TSMC library (obtained together with our industrial partners) and are scaled to 11nm by extrapolation for  $\Delta V_{th}$  using the scaling factors provided by Intel. For the weighting function we explained before, we experimentally found  $\alpha \leftarrow 0.6$  ( $> 1.0$  weight at 600 MHz) and  $\beta \leftarrow 1$  good for early-aging and  $\beta \leftarrow 0.3$  and  $\alpha \leftarrow 4$  good for late-aging. Our weight limit for the required-frequency matching is at  $w_{max} = 10$ .

### VI. AGING AND TEMPERATURE COMPARISON TO STATE-OF-THE-ART

We compare our approach to state-of-the-art mapping approach as used in [28]. For fairness of comparison, we extended the approach of [28] towards being variability- and aging-aware for maximum throughput mapping, to support epoch knowledge, DTM, core-level frequency scaling support, temperature dependent leakage increase, etc. For brevity, we call it VAA. In both comparison partners, threads get assigned to cores that fulfill frequency requirements *at their current age*, no chip-level guardbanding is considered. Threads only run at their required frequency and not faster. Additionally, threads get migrated from cores running too hot to the coldest current core, to fulfill their throughput requirements. The comparison is done for aging rates of maximum frequency per-core and per-chip, peak temperature, and DTM events; see Figs. 7 – 10.

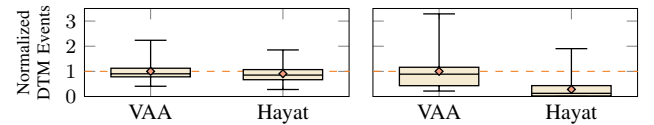


Fig. 7: DTM migration across 25 different chips normalized to VAA, **Left:** min. 25 % dark silicon **Right:** min. 50 %

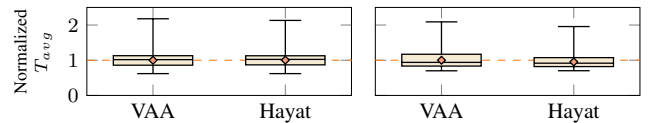


Fig. 8: Temperature over  $T_{ambient}$  across all Cores and Chips across 25 different chips normalized to VAA, **Left:** min. 25% dark silicon **Right:** min. 50 %

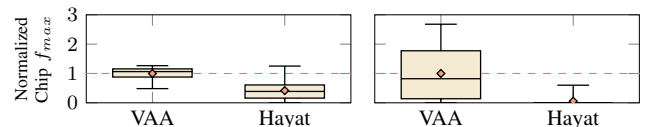


Fig. 9: Aging-rate of Maximum Frequency per Chip, over 25 different chips normalized to VAA, **Left:** min. 25 % dark silicon **Right:** min. 50 %

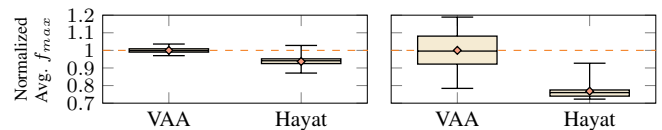


Fig. 10: Aging-rate of Per-Core Maximum Frequencies across 25 different chips normalized to VAA, **Left:** min. 25 % dark silicon **Right:** min. 50 %

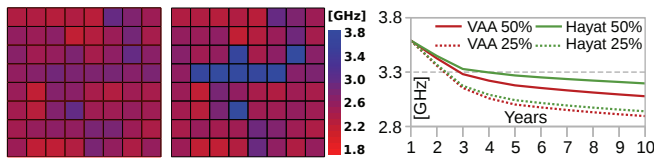


Fig. 11: **Left:** Aged Frequency of VAA vs. Hayat for an example 8x8 chip after 10 years. **Right:** Average aging across all chips over 10 years

Considering this, our *Hayat* system reduces DTM migrations by 10% for a minimum of 25% dark silicon (as Fig. 7). This increases to a significant reduction by 72% less events with 50% dark silicon, as more thermal headroom is left due to optimized DCM. This also indicates towards reduced performance overhead. The average temperature is reduced by 5% in case of 50% dark silicon as more spatial headroom is available there, and with no change at 25%. But these temperatures from Fig. 8 are only system-wide average temperatures, and therefore aging can still be decreased by reducing single-core temperatures. This can be seen in Fig. 10 and 9. In the first figure, the aging rate for the maximum available system frequency of a single core can be seen to be better by 95% after 10 years in 50% dark silicon, as the *Hayat* system preserves these high frequency cores for later lifetime years or for short-deadline applications, if other single-threaded workloads with high-ILP are considered. In Fig. 9, the aging rate of the average frequencies are shown. For 25% dark silicon, the dark silicon can be exploited to decelerate the aging rate by 6.3%. With more headroom at 50% dark silicon, the average frequency aging rate is decreased by 23%.

Fig. 11 shows a comparison between VAA and *Hayat*. The left side shows two example maps of the VAA and *Hayat* approach after 10 years, which supports the above observations. The right side of Fig. 11 shows the average frequencies for VAA mapping and *Hayat* system over a lifetime of 10 years. In case of 50% dark silicon, *Hayat* improves the lifetime by 3 months if the required lifetime is 3 years. The lifetime savings are improved significantly to 2 $\times$  if the required lifetime is 10 years. This shows that our *Hayat* system is, in particular, beneficial for systems with longer lifetime constraints.

**Overhead Discussion:** The cost of *Hayat* is two-fold: (1) During each aging epoch (i.e., 3 or 6 months), an estimation about the chip’s health map is made for the next aging epoch and saved in tables. This takes about 1 – 10 seconds each 3 or 6 months, which is negligible. (2) In case a new application starts within an aging epoch (typically in intervals of several minutes after the previous decision), we estimate the age by table lookup, and estimate the temperature-dependency (in detail: “estimateNextHealth”: 10 $\mu$ s, “predictTemperature”: 25 $\mu$ s). In the worst case, 1.6ms can be required in total.

## VII. CONCLUSIONS

In this paper, we propose a novel aging deceleration and balancing technique called *Hayat* that leverages dark silicon and process variations to optimized the NBTI-induced aging while also considering the impact of temperature on aging and meeting threads minimum throughput requirements. To enable this, we also proposed a light-weight online aging estimation technique and performed a comprehensive aging analysis for dark silicon chips. We evaluated our system considering several realistic system aspects, like temperature-dependent leakage increase, DTM triggers even in case of a naive optimization, accelerated aging simulations, workload variations over a single simulation run, etc. *Hayat* improves the aging rates in terms of decelerated frequency degradation and improved thermal profiles. Experimental results demonstrate that, compared to state-of-the-art, *Hayat* decelerates the chip aging significantly. *Hayat* makes a case, where the dark silicon problem can be turned to an opportunity to improve the lifetime of the system.

**Acknowledgments:** This work is supported in parts by the German Research Foundation (DFG) as part of the priority program *Dependable*

*Embedded Systems* (SPP 1500 – <http://spp1500.itec.kit.edu>) and as part of the Transregional Collaborative Research Centre *Invasive Computing* (SFB/TR 89 – <http://invasic.de>).

## REFERENCES

- [1] M. Shafique, S. Garg, J. Henkel, and D. Marculescu. The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives. In *Design Automation Conference (DAC)*, 2014.
- [2] M. Shafique, S. Garg, T. Mitra, S. Parameswaran, and J. Henkel. Dark silicon as a challenge for hardware/software co-design. In *International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS*, 2014.
- [3] J. Henkel, H. Khdr, S. Pagani, and M. Shafique. New trends in dark silicon. In *IEEE Design Automation Conference, DAC*, 2015.
- [4] International technology roadmap for semiconductors, <http://public.itrs.net/reports.html>.
- [5] Y. Turakhia et al. Hades: Architectural synthesis for heterogeneous dark silicon chip multi-processors. In *Design Automation Conference (DAC)*, 2013.
- [6] F. Kriebel et al. Aser: Adaptive soft error resilience for reliability-heterogeneous processors in the dark silicon era. In *(DAC)*, 2014.
- [7] Dieter K. Schroder and Jeff A. Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Applied Physics*, 94(1):1–18, 2003.
- [8] M.A. Alam and S. Mahapatra. A comprehensive model of PMOS NBTI degradation. *Microelectronics Reliability*, 45(1):71 – 81, 2005.
- [9] J. Keane et al. An all-in-one silicon odometer for separately monitoring hci, bti, and tddb. *Journal of Solid-State Circuits*, 45(4):817–829, 2010.
- [10] E. Karl et al. Compact in-situ sensors for monitoring negative-bias-temperature-instability effect and oxide degradation. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 410–623, 2008.
- [11] A. Tiwari and J. Torrellas. Facelift: Hiding and slowing down aging in multicores. In *41st IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 129–140, 2008.
- [12] Charles R. Lefurgy et al. Active guardband management in power+ to save energy and maintain reliability. *IEEE Micro*, 33(4):35–45, 2013.
- [13] J. Abella et al. Penelope: The nbt-aware processor. In *International Symposium on Microarchitecture (MICRO)*, pages 85–96, 2007.
- [14] J. Henkel et al. Reliable on-chip systems in the nano-era: Lessons learnt and future trends. In *DAC*, 2013.
- [15] K. Kang et al. Nbti induced performance degradation in logic and memory circuits: how effectively can we approach a reliability solution. In *ASP-DAC*, pages 726–731, 2008.
- [16] J. Shin et al. A proactive wearout recovery approach for exploiting microarchitectural redundancy to extend cache sram lifetime. In *ISCA*, pages 353–362, 2008.
- [17] A. Masrur et al. Schedulability analysis for processors with aging-aware automatic frequency scaling. In *RTCSA*, 2012.
- [18] Li et al. Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Symposium on Microarchitecture*, pages 469–480, 2009.
- [19] N. Binkert et al. The gem5 simulator. *SIGARCH Comput. Archit. News*, pages 1–7, August 2011.
- [20] K. Skadron et al. Temperature-aware microarchitecture. In *ACM SIGARCH Computer Architecture News*, volume 31, pages 2–13. ACM, 2003.
- [21] E. Rotem et al. Power-management architecture of the intel microarchitecture code-named sandy bridge. *IEEE Micro*, 32(2):20–27, 2012.
- [22] R. Viswanath et al. Thermal performance challenges from silicon to systems. *Intel Technol. J.*, 23(Q3):16, 2000.
- [23] H. Shojaei et al. A parameterized compositional multi-dimensional multiple-choice knapsack heuristic for cmp run-time management. In *Design Automation Conference (DAC)*, pages 917–922, 2009.
- [24] Gerald S. and Matthew L. Moldable parallel job scheduling using job efficiency: An iterative approach. In *(JSSPP), ACM SIGMETRICS*, 2006.
- [25] J. Xiong, V. Zolotov, and L. He. Robust extraction of spatial correlation. *Computer-Aided Design of Integrated Circuits and Systems, Transactions on*, 26(4):619–631, April 2007.
- [26] Raghunathan et al. Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors. In *Conference on Design, Automation and Test in Europe*, pages 39–44, 2013.
- [27] M. Shafique, D. Gnad, S. Garg, and J. Henkel. Variability-aware dark silicon management in on-chip many-core systems. In *IEEE Design, Automation and Test in Europe Conference, DATE*, 2015.
- [28] M. Fattah et al. Smart hill climbing for agile dynamic mapping in many-core systems. In *Design Automation Conference (DAC)*, 2013.