Revisiting FPGA Routing under Varying Operating Conditions

Behnam Khaleghi, Sahand Salamat, and Tajana Šimunić Rosing
CSE Department, UC San Diego, La Jolla, CA 92093, USA
Email: {bkhaleghi, sasalama, tajana}@ucsd.edu

Abstract—FPGA devices are continually integrating more and more resources to satisfy emerging applications’ performance requirements, which has increased the power of cutting-edge devices beyond CPUs. Consequently, more aggressive power reduction techniques have been explored recently, including voltage scaling, which is also adopted by several commercial FPGA families. In this paper, we investigate the FPGA routing (both the interconnection network and the routing algorithm) under variable voltage, temperature, as well as degradation. We first examine routing switch boxes (SBs) and point out that SBs with different wire segment lengths have different sensitivity/tolerance of resources to varying operating conditions. Accordingly, we show how architectures with similar overall efficiency in the nominal condition can have a different performance at the scaled voltage or temperature. Finally, we reveal that unlike current FPGA flow that first accomplishes the placement and routing and ex-posts multi-condition timing analysis, bringing the timing information of the actual operating condition in the placement and routing steps helps the underlying algorithms to utilize the resources that have better relative efficiency in the target condition, leading to higher performance.

I. INTRODUCTION

FPGA devices are perpetually evolving to satisfy the performance demand of emerging applications with affordable power consumption [1]–[3]. Adopting leading-edge process technologies unleashes smaller nodes’ performance and power benefits and enables integration of more transistors hence resources [4]. Improvements on the logic and circuit side incorporate tuning the FPGA architectural parameters in tandem with efficient transistor sizing. These enhancements are urged by migration to newer technology node, the advance of CAD tools, and requirements of emerging applications, e.g., machine learning accelerators, which have accentuated a new dimension of hard-macro calibration [5].

In recent years, there has been a shift towards more radical techniques such as voltage over-scaling [6], mitigating voltage transients enabling aggressive voltage scaling [7], and temperature-aware voltage calibration [8] to mention a few. The urge for such enhancements primarily stems from the end of Dennard scaling, whereby scaling the supply voltage (\(V_{dd}\)) in recent technology nodes has slowed down. However, the number of FGPA’s resources is continually growing, making FGPA’s power consumption exceed CPUs [7]. As a result, FPGA vendors have adopted similar techniques to those used in CPUs to lower the power density. E.g., Intel has introduced SmartVID solution that enables FGPA’s to operate at a lower voltage as far as the timing requirement is met [9].

Running digital circuits, particularly FPGA devices due to their customized cell layouts, in unconventional conditions (e.g., scaled voltage) involves various circuit-level challenges. In a recent work [10], the authors have investigated the performance of FPGA logic and interconnect resources under voltage scaling and have observed an excessive increase in delay when the voltage scales down, i.e., more than 6\(\times\) for look-up tables (LUTs) and 3\(\times\) for routing switch box (SB) multiplexers for just 200 mV voltage decrement. Another recent study [11] has pointed out that various FPGA resources (viz., logic, routing, memory, and DSP) exhibit different behavior under varying voltage and temperature, and leveraged it for maximal voltage reduction and consequently power minimization while keeping the performance intact. A similar observation is also found for degradation impact (which aggravates in deep-nano processes), showing a different rate in various FPGA resources [12].

To the best of our knowledge, the architecture-level impacts of such observations (i.e., operating at non-nominal conditions such as scaled voltages) have not been explored. There are a few studies on architectural evaluation of logic cluster [13], input crossbar [14], and routing topology [15], however, these studies consider a specific operating condition (e.g., 25 °C, constant temperature and voltage as in COFFE [16]), hence the resultant insights may not hold in the new generation of devices capable of operating at non-nominal condition. That is, an FPGA architecture optimized for a nominal operating condition may no longer be efficient under varying conditions.

In this paper, we investigate FPGA interconnect resources and place and route (P&R) flow under variable voltage and temperature, as well as degradation. We first analyze the FPGA resources at the circuit level and show how routing networks with different interconnection (wire/segment lengths) are affected under varying conditions. It is a key observation for our study as previous works have merely considered same-length (homogeneous) interconnect wires [10], [11], which overlooks the subsequent architectural and algorithmic implications. Afterward, we examine the architectural implications of our observations, i.e., how different architectures’ performance is affected under variable conditions. Finally, we show how neglecting our achieved insights leads to non-optimal place and route (P&R) flow. The current FPGA flow [19] performs the P&R under a specific operating condition and post-process the timing analysis for the desired condition. We show that in contemporary architectures with heterogeneous (i.e., varied-length) routing networks and varying operating conditions, ignoring the target condition in the primary P&R stage leads to a non-optimal implementation due to the different sensitivity/tolerance of resources to varying factors.
Through extensive experiments, our key findings revealed that when operating at lower voltages, short-length routing segments experience a delay increase of up to 1.68× more than longer wires, suggesting favoring longer wires at the architecture exploration step. Consequently, we observed that augmenting an exemplary routing architecture with longer segments increases its relative efficiency by 5.54% at low voltages. Note that this is valuable in the concept of architecture exploration that struggles to achieve small improvements (e.g., 1% improvement in Stratix V over Stratix IV by replacing H4 wires with a combination of H3/H6 wires [20], or 4.5% improvement in Stratix 10 by adding more variety to the segment lengths [21]). Moreover, we found out that in contemporary heterogeneous routing networks, a 'condition-aware' P&R achieves an average frequency boost of 11.0% compared to conventional flow that first accomplishes the P&R and ex-posts the timing analysis.

II. BACKGROUND

A. FPGA Architecture

Fig. 1(a) illustrates an overview of the contemporary architecture of FPGAs [3], known as island-style. Each CLB comprises \(N\) LUTs, where a \(K\)-input LUT is capable of implementing Boolean functions of up to \(K\) variables. LUTs bind together to implement more complex functions. Switch Boxes (SBs) and Connection Blocks (CBs) are in charge of global routings. Outputs of CLBs are directly connected to SB multiplexers (denoted by 1 in Fig. 1(b) and (c)), and output of each SB multiplexer composes one of channels’ wire segments. Each of \(J\) CB multiplexer (3) is connected to a subset of adjacent channel wires and passes the selected one by appropriate configuration of its select SRAM cells. Besides the CLB outputs, other inputs of SB multiplexers are connected to the branches of intersecting channel wires: as denoted by 2 in Fig. 1(b), each wire segment entering the channels intersection splits to \(F_s\) branches (\(F_s=3\) here).

Letting \(L\) define the length of wire segments, a length-\(L\) wire existing an SB multiplexer heads to the next direct (i.e., at the same track) SB multiplexer after traversing \(L\) hops (switch matrices). However, depending on the network parameters, the segment might have \(F_s\) branches at each of the intermediate hops (denoted by 2 in Fig. 1(b) and (d)). Long segments are beneficial for distant connections as they face only one multiplexer delay (and a long wire delay), whereas utilizing a combination of short segments would bear the same aggregate wire delay but with several SB multiplexers delay. Analogously, shorter wires are advantageous for neighboring connections. The routing algorithm is responsible for the optimal use of interconnect resources to yield high performance.

B. Experimental Setup

We need to model both the FPGA fabric to estimate the delay at different conditions and the FPGA CAD flow to enable architecture exploration by implementing different benchmarks and evaluating the P&R under varying operating conditions.

To model the FPGA fabric, we use COFFE 2.0 [16]. COFFE generates the SPICE netlist for a given (customizable) FPGA architecture and optimizes transistors’ size to achieve a pseudo-globally optimal architecture. We use the baseline area-delay product cost with four rounds of optimization iterations. COFFE requires transistor model to perform SPICE simulations, for which we use 22 nm Predictive Technology Model (PTM) process [22] with \(V_{dd}=0.8\) V and \(V_{ssram}=1.0\) V. We try homogeneous (i.e., same-length) and heterogeneous routing networks consisting of \(L=1\) up to \(L=6\) wires.

Table I summarizes the parameters of considered FPGA architectures, which are used as inputs for COFFE. We vary the routing parameters (\(L\)) to evaluate different routing networks, but the logic tiles are fixed among all networks and follow the Intel devices [17], [18], [20]. \(K\), \(N\), and \(I\) are also defined in Section II-A. \(F_{cin}\) denotes the fraction of adjacent channel tracks a CB multiplexer is connected to, therefore, number of inputs of a CB multiplexer is equal to \(F_{cin} \times W_{max}\). Analogously \(F_{cin}\) determines the number of connections from a cluster output to neighboring channels. The length of routing segment \(L\) affects the routability hence the required channel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(K) (LUT size)</td>
<td>6</td>
<td>(F_{cin})</td>
<td>0.15</td>
</tr>
<tr>
<td>(N) (cluster size)</td>
<td>10</td>
<td>(F_{cont})</td>
<td>0.10</td>
</tr>
<tr>
<td>channel tracks (width)</td>
<td>(W_{max})</td>
<td>(F_s)</td>
<td>3</td>
</tr>
<tr>
<td>(L) (segment length)</td>
<td>1-6</td>
<td>(MUX_{local}) inputs</td>
<td>25</td>
</tr>
<tr>
<td>(I) (cluster inputs)</td>
<td>4</td>
<td>BRAM</td>
<td>1024×32 bit</td>
</tr>
</tbody>
</table>
The delay of each resource is normalized to its delay in the nominal condition.

**TABLE II**

<table>
<thead>
<tr>
<th>Segment length</th>
<th>W_{max}</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>150</td>
</tr>
<tr>
<td>L2</td>
<td>144</td>
</tr>
<tr>
<td>L3</td>
<td>160</td>
</tr>
<tr>
<td>L4</td>
<td>170</td>
</tr>
<tr>
<td>L5</td>
<td>178</td>
</tr>
<tr>
<td>L6</td>
<td>192</td>
</tr>
</tbody>
</table>

tracks (W_{max}) that guarantees all benchmarks are successfully routable on the defined architecture. Therefore, in Table II we report the minimum channel tracks for routing networks with different segments, which we obtained by placing and routing the VTR benchmarks [23] using VPR 8.0 and finding the maximum required channels benchmarks. In practice, we use 1.2× of the reported value to give the router enough flexibility for efficient routing.

**III. CIRCUIT-LEVEL IMPACT OF OPERATING CONDITIONS**

Fig. 2 shows the delay of routing SB multiplexers under varying Voltage, Temperature, and Degradation. We dub these combined effects as VTD. To obtain these results, we used the generated SPICE netlist of COFFE at nominal condition (25 °C, 0.8 V) and changed the voltage or temperature. Each curve represents a certain L and is normalized to its value at the nominal condition. To model degradation, we extracted threshold voltage (V_{th}) shift and mobility (µ) degradation for 22 nm technology of various years from [24] and replaced in the PTM model parameters.

As we can see in the figure, as the length of segment increases, the SB multiplexer’s sensitivity to variations reduces, i.e., L1 has the highest delay increase under VTD while L6 takes less impact. Clearly, the reduction of voltage has a more significant impact on the delay than temperature and degradation, where we see up to 1.9× delay increase for L1 when the voltage reduces from 0.8 V to 0.6 V. It is noteworthy that our results concur with the study in [10], where the authors report ~1.6× delay increase for L=4 at 0.6 V, for which we obtained 1.63×. Note that, as we mentioned earlier, [10] only considers a homogeneous routing network (i.e., only consists of L4), and its purpose is to enhance the SBs and LUTs netlists for low voltage regimes (we followed their suggestion and used boosted SRAM voltages for SBs; otherwise, the delay increase would exceed 3×).

Although the size of transistors, especially the output buffer of multiplexers, plays a role in how a change of voltage or temperature affects the delay of a resource [25], the main reason for the observed different sensitivity of resources is the output load wire (routing tracks) of these multiplexers. The delay of routing tracks amortizes the change in the delay of the multiplexer and buffer structures. Therefore, since the SBs in routing tracks with longer segments drive longer wires (which means larger load), the ratio of overall delay change (i.e., the total delay of SB multiplexer and accompanied track) is smaller compared to short-length routing networks. For the same reason, as we report in the next subsection, we observe a narrow range of delay increase for other resources, e.g., [1.45× to 1.61×] for connection block (CB) multiplexers in different networks. It is because in all L1−6 networks, each CB multiplexer drives the same type and number of resources. In these cases, the difference in the sensitivities is because of transistor/buffer sizing. As mentioned in Section II-B, COFFE tries to achieve a globally optimal fabric; thus, although the logic parameters are the same for all networks, the size of building transistors are not necessarily similar.

As a summary and a key observation, we noticed that by decrementing the voltage by 200 mV, the delay of L1 SBs increase by 1.90×, while it consistently reduces for larger lengths the delay of L6 SBs increase by 1.47×. Similarly, the impact of temperature and degradation on L1 (L6) is 1.45× (1.27×) and 1.17× (1.09×). Taking all phenomena into consideration, according to our measurements, the combined impact of low voltage, high temperature, and degradation increase the delay of L1 multiplexers by 3.20×, whereas this is 2.25× for L6 (and even smaller for longer segments, e.g., ~1.68× for L8). Therefore, L1 SBs observe 42% higher delay increase under the corner VTD condition.

**IV. ARCHITECTURAL IMPACT OF OPERATING CONDITIONS**

Commercial devices use heterogeneous-length routing networks with a combination of short, medium, and long segments. Although the distribution of the lengths and bends of routed nets in benchmark designs, as well as physical attributes of logic clusters (e.g., width and height), have significant roles in deciding the architecture of routing network, eventually the predefined cost of architecture (e.g., performance, area, and power) determines which architecture is optimal. As we showed in the previous section, the difference in routing resources’ sensitivity to variable operating conditions affects the relative performance; therefore, the architectures’ efficiency in a varying operating regime. In this section, we aim to quantify the impact of VTD on architectural efficiency.
In Table III, we show the nominal delay (at 25 °C, 0.8 V) of the resources of each routing network along with the factor of delay increase of that resource under reduced voltage of 0.6 V (under the columns labeled VTD), and under combined VTD impact (under the columns labeled VTD). For instance, for the architecture with L1 segments, the SB delay is 68 ps, which increases to 68×1.90=129 ps at 0.6 V, and to 68×3.19=217 ps under combined 0.6 V, 85 °C, and degradation. We obtained these results in the same manner explained in Section III. Note that since COFFE does not model the DSP block, we first synthesized a divisible 36x36 multiplier with NanGate standard cell library using Synopsys Design Compiler. Using Synopsys SiliconSmart, we created two modified cell libraries, one for reduced voltage and the other for combined VTD using the post-layout SPICE netlist of the gates provided by NanGate. We used the generated libraries to obtain the DSP delay rate at different conditions, as listed in Table III. For BRAMs, we used the same BRAM netlist for all the architectures. Also, since BRAM is implemented using a low power (high $V_{th}$) transistor technology and uses a separate voltage rail (with $V_{bram}=0.95$ V), we assumed the BRAM voltage is not changed. However, under VTD, we consider the impact of temperature for BRAMs. We should also remind that, for LUTs, following [10], we assumed gate-boosted LUTs, in which the local (intra-cluster) multiplexers that drive LUT inputs are augmented by level shifter to boost the voltage of LUTs input buffers, which precludes excessive delay increase of LUTs in reduced voltages. Thus, the delay of LUTs virtually remains the same at 0.6 V; however, combined VTD affects LUT delay (by 1.28–1.37× depending on which of six inputs). As alluded before, although the logic parameters of all architectures are the same (see Table I), there are slight differences in both the nominal delay and sensitivity of some logic resources among different architectures since optimizing the routing network affects the logic sizing (e.g., the driving strength of SB multiplexer impacts the sizing of CB multiplexer and so on).

To examine the architectures efficiency, using the data of Table III, we placed and routed 19 VTR benchmarks [23] with the architecture described in Table I. As the initial placement affects routing quality, we placed and routed each design three times using different placement seeds for the VPR tool to average out the variance. In addition to six homogeneous routing networks (L1–L6), we considered three heterogeneous routing networks, namely HET1 with 75% L2 and 25% L4 tracks, HET2 with 25% L2 and 75% L4, and HET3 with equal number of L2, L4, and L6 routing tracks. Similar to homogeneous architectures, we also first obtained the minimum required tracks ($W_{max}$) to route all VTR benchmarks and used 1.2× of that. We obtained $W_{max}=148, 152, and 156$ for HET1, HET2, and HET3, respectively.

Fig. 3 shows the geometric mean of the area (in terms of $10^6$ minimum-width transistor area) and the delay of VTR benchmarks implemented on the architectures mentioned above. As mentioned earlier, for each benchmark, we use the average area and delay of three seeds. Accordingly, in Fig. 4, we show each architecture’s area-delay product. The blue curve (with triangle markers) shows the area×delay of all architectures in the nominal operating condition. Of independent interest is that, among homogeneous architectures, L2 achieves the minimum area-delay product while the majority of previous work in academia have adopted L4 routing network. We believe the choice of L4 as the base homogeneous routing network has arisen from simplifying the architecture of older Intel Stratix IV devices; however, the routing network has undergone significant changes. From Fig. 3 we can observe that L2 and L4 architectures have a similar area while L2 has a smaller delay, making it more efficient than L4 overall.

Finally, the yellow curve (with circle markers, correspond-
ing to the right-hand axis) of Fig. 4 shows the area-delay product of the same architectures under VTD condition. We have adjusted the scales of the left (nominal) and right (VTD) vertical axes such that the two curves are easily comparable with respect to L2 switches. Note that each architecture has the same area, and only their delay at VTD condition affects the overall area-delay product. Clearly, all architectures except L1 are shifted downwards, meaning that, relative to L2, they face a smaller ratio of delay increase in non-nominal condition. Especially, as L grows, the gap is larger, where L6 has the largest gap (i.e., relative improvement). Consequently, although L2, HET1 and HET2 have very close values of area×delay at nominal condition, at VTD condition, HET1 and HET2 surpass the L2 routing network as both of them have introduced longer L4 segments that perform better when the voltage or temperature is changed, or the transistors degraded.

To elaborate further, in Fig. 5, we show the area-delay product of all benchmarks implemented on HET1 architecture, normalized to the area-delay of the same benchmark implemented on L2, at both nominal condition denoted by HET1/L2(nom), and when the condition is changed to VTD corner, denoted by HET1/L2(VTD). The relative delay (hence, the area-delay product) of all except three benchmarks (viz., boundtop, diffeq1, and or1200) improves under VTD condition. Operating at nominal condition, the area-delay product of HET1 over L2 is 1.019× (i.e., ~1.9% less efficient), which reduces to 0.969×, i.e., becomes 3.23% more efficient at VTD condition (5.13% improvement) due to better performance of L4. Analogously, Fig. 6 demonstrates a similar comparison between HET2 and L2 routing architectures. Again, except for three benchmarks, the efficiency of HET2 increases at the VTD condition. At nominal condition, HET2 has 1.52% better area-delay efficiency than L2, which further improves to 7.06% at VTD (5.54% improvement) compared to HET1. HET2 sees slightly more improvement as it comprises a higher percentage of longer L4 segments.

In summary, we observed that although the homogeneous L4 network is less efficient than L2 network (in both nominal and VTD condition), adding L4 segments to the homogeneous L2 network improved its efficiency in VTD condition, pointing out the necessity of bringing the condition-awareness into architecture exploration. Particularly the optimal architecture at nominal condition has not a global minimum (e.g., in [20], choosing different segment lengths for horizontal channels shows small impact), therefore we can prioritize the solution with better performance at non-nominal conditions supported in recent devices. While we observed 5.5% increase in the relative efficiency of HET1 at VTD condition, we believe this efficiency will be even higher for heterogeneous architectures utilizing L8 and L12 long global segments [3].

V. IMPACT OF OPERATING CONDITIONS ON P&R

Some FPGA CAD tools such as Intel Quartus enables static timing analysis of a placed and routed design at several operating corners [19]. The primary P&R, however, is performed at a specific unchangeable condition. Nonetheless, we showed that certain resources (particularly long segments routing tracks) perform relatively better at lower voltages and higher temperatures, while the others (e.g., the global output multiplexers of BLEs) provide very poor performance. In this section, we study whether a VTD-aware P&R helps optimize the underlying algorithms for non-nominal conditions.

To examine the aforementioned concept, we first place and route the VTR benchmarks using delay information of nominal condition, and then obtain the delay of the implemented benchmarks by replacing the delays with VTD point delays we obtained in previous sections. This procedure is the same as conventional routing flow [19]. For ‘operating condition-awareness’, we directly use the VTD delay, so the P&R algorithms are aware of the delay of resources in the target operating condition. To average out the impact of initial
Fig. 7. Average performance improvement of VTD-aware P&R on HET1 routing network.

Fig. 8. Average performance improvement of VTD-aware P&R on HET2 routing network.

placement, we repeat the experiments using six different seed values.

Fig. 7 and Fig. 8 compare the performance improvement of VTD-aware P&R versus the conventional flow, on the HET1 and HET2 architectures, respectively. For each benchmark, we show the average improvement of six different placement seeds. As a reminder, HET1 comprises 75% L2 and 25% L4 segments, while HET2 consists of 25% L2 and 75% L4 segments. We obtain an average of 10.8% and 11.0% performance improvement for the HET1 and HET2 networks, respectively.

To better understand the efficacy of VTD-aware flow, we obtained the utilization rate of L2 and L4 segments in the critical path of the conventional and the proposed flow. Note that the critical paths (source-sink nodes) of the benchmarks are not necessarily the same in these flows. As shown in Fig. 9, unanimously in all benchmarks, the VTD-aware flow uses more L4 segments. In the conventional flow, the average L2 – L4 utilization rate is 76%–24%, which is close to the percentage of these segments in the architecture (i.e., 75%–25%). The VTD-aware flow, however, increases the average L4 utilization of the critical paths to 45% because of the better efficiency of these segments in the VTD point.

Besides the utilization rate, we further look over the performance details of the two flows. Remember that the conventional P&R flow first uses the delay information of the nominal condition and obtains the performance of the target VTD condition using subsequent static timing analysis. In contrast, VTD-aware P&R directly uses the target delay information in the placement and routing. Therefore, to inspect how these two flows compare at nominal condition, we replace the resource delays of the VTD-aware implemented design with the original nominal delays (the conventional P&R has already obtained the performance using the original delays).

Fig. 10 shows the logic and routing delay of the VTD-aware P&R flow, normalized to the conventional flow, both at the nominal condition. We can see a notable difference in these two flows. At nominal condition, the critical paths of VTD-aware flow have an overall smaller logic delay (0.86$x$) compared to the conventional flow, while it has a larger routing delay (1.17$x$ of the conventional flow). When we use the delays of the target VTD condition delays (in which the designs will be operating), according to Fig. 11, the logic delay ratio further improves to 0.77$x$ (from 0.86$x$), and the routing ratio improves to 1.05$x$ (from 1.17$x$). It reveals that the VTD-aware P&R flow appropriately selects and uses the resources that are more efficient in the target VTD point. Also, the VTD-aware P&R has targeted a ‘holistic improvement’. That is, the average routing delay of the VTD-aware flow is higher than conventional flow (although it is reduced from the 1.17$x$ at the nominal condition to 1.05$x$ at the desired VTD point due to using more L4 segments as discussed), but since the logic delay is reduced to 0.77$x$, the overall performance is better. Note that in the VPR flow, the routing delay includes SBs and CBs delay, whereas the logic delay, in addition to LUTs, DSPs, and BRAMs, also includes the intra-cluster routing consisting of all local input/output multiplexers as well as the global output multiplexers of the BLEs, which on average reach up to ~50% of the total delay. Thus, the improvement in the logic delay overshadows the slight increase of routing delay.

Finally, in Fig. 12 we show the average number of routing and logic resources at ten longest paths of all VTR benchmarks implemented on HET1 architecture. As already discussed, the number of utilized L2 switch boxes reduces while L4 increases. On the logic side, the number of BLEs (see Fig. 1) local and global output multiplexers are roughly halved, while the number of local multiplexers are increased. This trend follows the tolerance of these resources to delay increase at VTD condition, which is shown by ▲ (data is taken from Table III). The local and specially global output multiplexers have a large delay increase, so the VTD-aware flow has moved toward using local multiplexers to realize routing through intra-cluster connections.

VI. RELATED WORK

FPGA architecture exploration has a background of over two decades, starting with the seminal works by Betz and Rose [26], [27]. Among the most recent studies, Chiasson and Betz [17], by introducing the COFFE automated transistor sizing tool [18], have explored and evaluated different circuit-level implementation of multiplexers and LUTs. In [15], Petelin and Betz have explored complex interconnect patterns and interconnection topologies. Zghieb and Ienne use the so-called FPRESSO [28] tool (which is a faster but less-accurate alternative of COFFE) to investigate the crossbar density of
FPGAs [13]. None of these studies consider the impact of voltage, temperature, or degradation.

In [29], Khaleghi and Rosing leverage the temperature-delay dependency of FPGA resources to boost the performance at lower temperatures. Alternatively, in [11], the authors use the temperature-delay dependency to scale the voltage at lower temperatures to reduce the power consumption. Similarly, Zhao et al. [8] find out the performance-voltage correlation of FPGA designs at different temperatures (through implementing and measuring the critical paths before implementing the actual design) in order to scale the voltage or frequency in the runtime. None of these studies investigate the routing network architecture, nor do they analyze the sensitivity of various routing resources with different segment lengths to achieve optimal architecture or P&R flow. Eventually, in [10], Ahmed et al. investigate the impact of voltage scaling on LUTs and length-4 routing switch boxes’ performance to design more-tolerant resources for voltage scaling (i.e., to reduce delay increase at low voltages). They found out SB multiplexer with voltage-boosted SRAM voltage is significantly more tolerant for voltage scaling. They leveraged a similar approach for LUTs by gate boosting the pass-transistors LUTs by augmenting the intra-cluster local multiplexers (that drive the LUTs) with voltage level shifters. Our experiments leverage the same
techniques in all architectures to make them more tolerant for voltage scaling, though evidently, our work pursues different goals.

VII. CONCLUSION

This paper investigated the impact of voltage reduction, temperature increase, and transistor degradation on FPGA resources. We observed that routing networks with longer segment lengths have less sensitivity (higher tolerance) to the varying impacts, making them potentially advantageous for such operating conditions. We showed that introducing long-length segments makes the architecture more efficient when operating at reduced voltage, implying that awareness of operating conditions should be considered during architecture optimization. Finally, as certain logic and routing resources such as long-length segments show more tolerance to the change of operating condition, we showed rather than accomplishing the placement and routing under a fixed condition and then post-processing the timing analysis for the target condition, making the P&R flow optimize for the target condition helps it boost the performance by using optimal resources.

REFERENCES


