Global Is the New Local: FPGA Architecture at 5nm and Beyond

Stefan Nikolić
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
stefan.nikolic@epfl.ch

Francky Catthoor
IMEC and KU Leuven
Leuven, Belgium
francky.catthoor@imec.be

Zsolt Tőkei
IMEC
Leuven, Belgium
zsolt.tokei@imec.be

Paolo Ienne
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
paolo.ienne@epfl.ch

ABSTRACT

It takes only high-school physics to appreciate that the resistance of a wire grows with a diminishing cross section, and a quick look at any plot about Moore's law immediately suggests that such cross section must decrease over time. Clearly, everyone can easily imagine that this trend must have a deep influence on FPGA architectures. What is difficult to predict is whether and when well-established architectural ideas will break—and what can replace them. Unfortunately, in architectural research, we often use fairly simplistic models of the underlying technology nodes which limit our ability to visualize the detailed impact of technology evolution. In this paper, we develop, from the available industrial disclosures, a consistent electrical model of the metal stacks of recent and current technologies, as well as future trends. We combine it to a plausible layout strategy to have an accurate idea of how wire characteristics play nowadays into architectural decisions. To demonstrate our models, necessarily speculative due to the paucity of reliable industrial information, we use them to explore the evolution of a typical architectural family across technology nodes and to reevaluate one of the most basic design parameters—namely, cluster size. We notice effects which may in fact explain some recent changes in commercial architectures. We also observe how conventional architectures may fail to take advantage of the performance improvements of future nodes. Although conceptually straightforward, this study signals how profoundly our understanding of FPGAs will be affected by technology while moving towards the 3 nm node.

ACM Reference Format:
Stefan Nikolić, Francky Catthoor, Zsolt Tőkei, and Paolo Ienne. 2021. Global Is the New Local:, FPGA Architecture at 5nm and Beyond. In *Proceedings of the 2021 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '21), February 28-March 2, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3431920.3439300

1 GLOBAL IS THE NEW LOCAL

Clusters exist in FPGAs to provide connectivity between adjacent *Look-Up Tables* (LUTs) that is either cheaper or faster than using the global (intercluster) routing—or, most likely, both. But, can they keep justifying their existence for much longer? To illustrate why one might have concerns about that, let us take a look at a floorplan sketch of a cluster of ten LUTs in two different technologies, shown in Figure 1. Annotated are the resistance and the intrinsic delay of a local (intracluster) interconnect line routing the signals between LUTs within the cluster and a global wire spanning the height of the cluster. The former is naturally drawn in one of the fine-pitch metal layers, closer to the active devices, whereas the latter, as customary for longer interconnects, uses a thicker metal layer from higher up in the stack. The numbers, accounting for area, metal wires, and logic delay scaling, are shown for two technology nodes, approximately representative of commercial 16-nm and 7-nm nodes. It seems rather plausible that, with such typical layout assumptions, reaching the top LUT from the bottom one in the same cluster will at some point take more time than reaching the bottom LUT in the cluster above it. Such a situation would make for an interesting challenge to a packing algorithm: it should try to spread timing critical connections across clusters instead of keeping them internal!

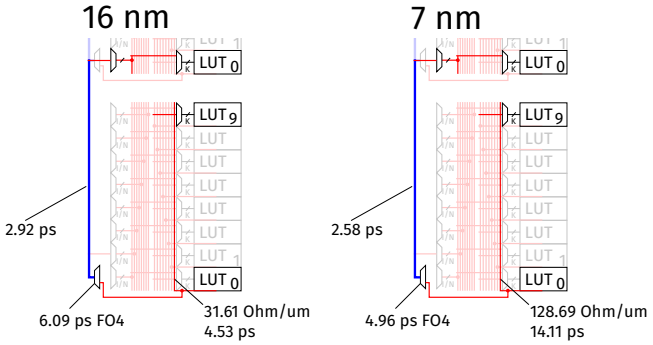


Figure 1: Evolution of local (intra-) and global (intercluster) interconnect. The local lines in red use thin lower metal layers while the global blue ones employ thick higher layers. The two technologies are approximately representative of commercial 16-nm and 7-nm technologies (full details in Section 2).

Of course, things are not this simple: One should also consider the cost of the additional multiplexing, horizontal offset and the via stack required to reach the thicker wire, etc. In practice, it is not necessarily clear whether the described hypothetical situation will actually happen, and, if so, when. The purpose of this paper is to develop a framework for modeling FPGA fabrics at advanced technology nodes, so as to answer quantitatively similar questions.

FPGA '21, February 28-March 2, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8218-2/21/02…$15.00
https://doi.org/10.1145/3431920.3439300

1.1 Technology Assumptions and Trends

The first three sections of the paper contain our assumptions on technology and FPGA architecture. We first present the models of resistance and capacitance of modern interconnect stacks, including vias, in Section 2. They are based on slightly simplified state-of-the-art models derived from recent literature on silicon device technology. We complement them with physical parameters inspired by data published by foundries and leading research institutions in the field. Our intention is not necessarily to be faithful to the characteristics of technology nodes by any particular foundry (which would simply be impossible to us), but rather to develop a credible and consistent sequence of data points useful to expose the trends imposed on FPGAs by the so-called *back-end of line* (BEOL) of silicon manufacturing. In Section 3 we develop a set of scalable area and wirelength models based on a plausible layout organization and consistent with information made available by some FPGA vendors. This lets us build realistic models not only of the interconnect delay, but also of the wiring bandwidth available above a certain active area—that is, in our case, of how many wires of each sort and in each direction can be drawn above a particular FPGA tile before its area becomes metal bound. This crucial aspect, often neglected in the literature on FPGA architecture, has a significant impact on the achievable features of reconfigurable arrays in modern technology nodes. Most definitely, its importance will be only amplified by the increased heterogeneity of the BEOL stack in future nodes. Finally, Section 4 briefly discusses our assumptions concerning the evolution of the active devices across technologies.

From these technological hypotheses, we extrapolate some trends. After some details in Section 5 on how we obtain delays from the electrical parameters, Section 6 shows the first results on interconnect delays, essentially supporting the hypothesis of Figure 1. We then move on to show how these results influence architectures. Since a thorough sweep of many design parameters is not possible, we discuss in Section 7 our exploration methodology. Section 8 contains our main results, experimentally showing how the performance of different cluster sizes evolves across technology nodes. These conclusions support some recent changes in commercial FPGA architectures and help us develop conjectures for the future.

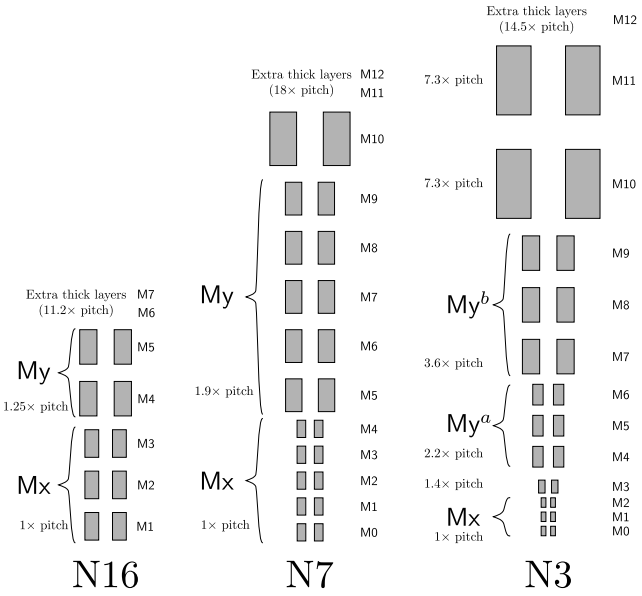
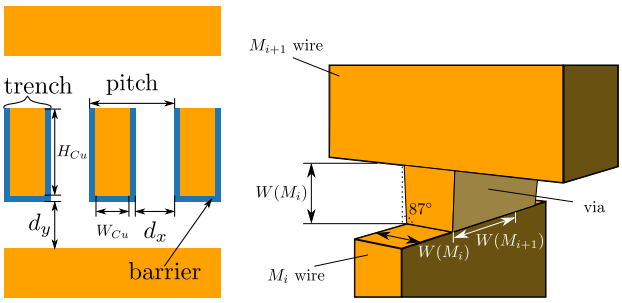


Figure 2: Representative metal stacks for the 16-, 7-, and 3-nm nodes [1–4]. All wires are drawn to scale.



(a) Wire cross section. **(b) Via.**

Figure 3: Relevant dimensions of a wire and a via.

2 INTERCONNECT MODELING

In this section, we present models used to derive resistance and capacitance of all the connections that have a significant impact on performance of architectures explored in this work. We then apply them on wire and via geometries representative of all the considered technology nodes. These geometries are later also used to assess feasibility of tracing the desired number of tracks over a given tile area.

2.1 Layers

Representative metal stacks for several of the technology nodes of interest are shown in Figure 2. We assume that two pitch options are used to route all wires, referring to the tighter as Mx and to the more relaxed as My. In most cases these correspond to the tightest and the second tightest pitch in the interconnect stack. The exception is the 3-nm node, where we also explore a possibility of promoting My one step further, as illustrated in the figure. In all cases, however, the layer group labeled as My is considered to be immediately above the layer group labeled as Mx, as the intermediate layers can be omitted. We assume that all connections within the individual blocks (LUTs, multiplexers, etc.) are routed at M2 or below, as customary for basic cells, whereas the intracluster connections connecting different LUTs together and passing external inputs to the crossbar multiplexers (*LAB lines* in the Stratix architectures [5]), as well as all the intercluster wires are routed at M3 and above.

2.2 Cross-sectional Wire Dimensions

All dimensions relevant to computing per-unit-length resistance and capacitance of wires, using the models of this section, are shown in Figure 3a. Typically, the only one that is readily available in public data from the foundries is the pitch and therefore we use it to derive all the other dimensions. Representative pitches for all the technology nodes of interest are shown in Table 1. As their names and the references in the footnotes suggest, the considered technology nodes are represented through parameters strongly inspired by corresponding commercial technologies (e.g., F16 resembles TSMC 16 nm). Speculative nodes (e.g., F3a/b representing hypothetical 3-nm nodes) are derived following the progression suggested in relevant literature and some manufacturability considerations. The F4 node is a speculative node that we suppose foundries might

Table 1: Metal pitches for all the technology nodes of interest. F7, for instance, is our hypothetical 7-nm node.

	F16	F7	F5	F4	F3a	F3b
Mx [nm]	64[1]	40[2]	38[3]	26[4]	22[5]	22
My [nm]	80[6]	76[7]	72[8]	50[9]	48[10]	80[11]

want to introduce as an intermediate step if moving directly to a 3-nm node would prove too dramatic a move.

We assume that the trench width equals 1.1× half pitch, to mitigate the resistance increase and ease via contacting. Representative barrier thicknesses that are used to obtain the final copper dimensions of the Mx layers are 3 nm until F5 inclusive and 2 nm from F4 onwards [6]. For the My layers, we assume a constant barrier thickness of 4 nm across all nodes. Spacing between the layers (d_y in Figure 3a) is set to equal the trench width, reflecting the typical via aspect ratio (height over width) close to 1. Finally, the height of the wire (H_{Cu} in Figure 3a) is determined through a sweep that seeks to minimize the $R'C'$ product (see below). The maximum allowed aspect ratio is set to 2 for all layers apart from Mx of F3b, for which we assumed possible an aspect ratio of 3 to mitigate the resistance surge at the expense of increased capacitance. With F3b we explore a different trade-off that could be made in a future advanced node. Taller wires are considered difficult to manufacture, so it is unlikely that it will be possible to further reduce resistance through aspect ratio optimization.

2.3 Resistance

Resistance suffers the greatest impact from the aggressive scaling of the wire pitch, due to the quadratic reduction of the cross-sectional area. Here, we adopt a slightly simplified version of the resistivity model introduced by Ciofi et al. [6], that is valid for all the technology nodes of interest to us. By assuming no tapering (i.e., wire sides are completely vertical, as in Figure 3a), integration of equation (1) of Ciofi et al. simplifies substantially and we obtain the following expression determining the resistance per unit length of a wire:

$$R' = \frac{1}{H_{Cu}W_{Cu}}\left(32.05 + 615\left(\frac{\tanh(0.133W_{Cu})}{W_{Cu}} + \frac{\tanh(0.133H_{Cu})}{H_{Cu}}\right)\right) \quad (1)$$

Variables W_{Cu} and H_{Cu} correspond to the definition of Figure 3a, while the constants have been empirically determined for a 7-nm technology node [6], which is in the middle of the range that we intend to explore.

2.4 Capacitance

Capacitance is less impacted by the pitch scaling than resistance. Pitch reduction does decrease the distance to neighboring wires (d_x and d_y in Figure 3a), thus increasing the coupling capacitance; yet, line width and height (W and H) decrease as well, balancing this out. Hence, for modeling capacitance, we use a less recent model due to Wong et al. [7], available at the PTM website [8], without any modification. For all technology nodes, we assume a relative permittivity of 2.8 for the lower metal layers and 3.0 for the intermediate ones, which is representative of the current trends in industry. The obtained resistance and capacitance per unit length are reported in Table 2. In all cases, the $R'C'$ optimization resulted in the maximum allowed aspect ratio. As predicted, we can see a substantial rise in R' between consecutive nodes, due to the shrinking of cross-sectional area, whereas C' remains constant since the dimensions with opposing influence scale uniformly.

[1]TSMC 16 nm [1].
[2]TSMC 7 nm [2].
[3]Fit to match the Mx RC increase from F7, amounting to about 16% [3].
[4]Close to the limits of a single-patterned EUV and a reasonable intermediate point between F5 and the predicted F3.
[5]IMEC prediction [4].
[6]TSMC 16 nm [1].
[7]TSMC 7 nm [2].
[8]Assuming 1.9× as in F7 [2].
[9]Assuming 1.9× as in F7 [2].
[10]IMEC prediction for the M4–M6 layers [4].
[11]IMEC prediction for the M7–M9 layers [4].

Table 2: Wire resistance and capacitance per micrometer length. Maximum allowed aspect ratio for Mx of N3b was increased to 3, to reduce the resistance at the expense of increased capacitance.

	F16	F7	F5	F4	F3a	F3b
Mx						
W_{Cu} [nm]	29.2	16.0	14.9	10.3	8.1	8.1
H_{Cu} [nm]	67.4	41.0	38.8	26.6	22.2	34.3
R' [Ω/μm]	31.6	128.7	151.6	392.9	666.4	396.7
C' [fF/μm]	0.22	0.22	0.22	0.22	0.22	0.28
My						
W_{Cu} [nm]	36.0	33.8	31.6	19.5	18.4	36.0
H_{Cu} [nm]	84.0	79.6	75.2	51.0	48.8	84.0
R' [Ω/μm]	18.7	21.6	25.1	75.7	86.4	18.7
C' [fF/μm]	0.24	0.24	0.24	0.24	0.24	0.24

Table 3: Resistance of vias. The reported values correspond to the resistance of a single via connecting two neighboring layers in the Mx group, or the buffer output at an Mx layer and a wire at an My layer, in case of stacked vias.

	F16	F7	F5	F4	F3a	F3b
Mx-Mx [Ω]	10.9	34.8	39.9	58.9	92.9	92.9
Stacked Vias (M2-M5)						
H [nm]	246.4	154.0	146.3	100.1	84.7	108.9
R [Ω]	19.2	30.5	34.7	69.8	88.0	44.9

2.5 Vias

To mitigate the effects of high resistance increase at lower metal layers, more and more signals are routed at higher ones. This means traversing long vertical distances, so it is important to accurately model via resistance, which is itself affected by technology scaling.

A via connecting layers M_i and M_{i+1} is shown in Figure 3b. We assume a classical 87°-tapered via [9]. We compute the width of the via at half the height and use it in place of H_{Cu} in Equation (1). $W(M_{i+1})$ is used in place of W_{Cu}, to obtain the via resistance per unit length. Here we note that for connecting layers of different pitch, the shape of the via is typically different and cannot be accurately modelled with this approach. However, as this is a reasonably small penalty that needs to be paid only twice per connection, we chose to prioritize modeling simplicity over accuracy. As stated before, we assume a unit aspect ratio for vias, so the final resistance requires multiplication by $W(M_i)$. To account for the resistance of the top and the bottom barrier, we assume a constant resistivity of 1,200 Ωnm [9], while the barrier thicknesses correspond to those of the layers that the via connects. Values of single via resistances

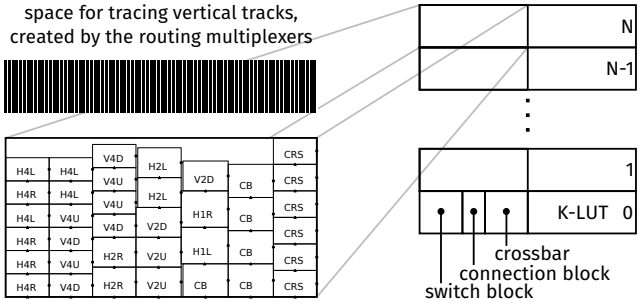


Figure 4: A fully stacked floorplan of the tile showing the position of the LUTs and various routing multiplexers, similar to a Stratix architecture [5]. A concrete example of a 3-nm architecture based on a cluster of four 6-LUTs is shown on the left. It illustrates a simple greedy multiplexer positioning algorithm, which stacks multiplexers sorted by decreasing input count, starting from those of the crossbar, right next to the LUT, then proceeding with those of the connection-block, appending them to the left, and finally, finishing with those of the switch-block. Each time the height within one column exceeds the height of the adjacent LUT, a new column is appended to the left. This creates additional space for tracing vertical wires above the tile.

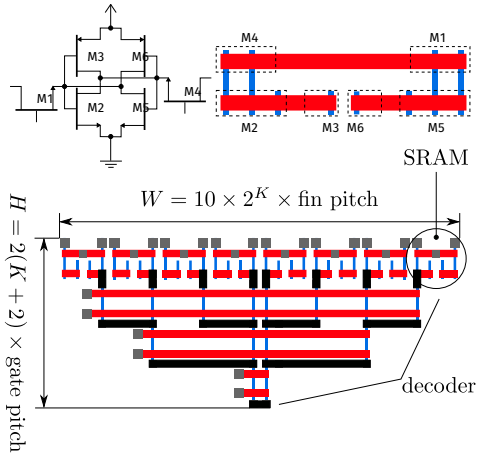


Figure 5: A sketch of the assumed LUT layout, based on the one due to Abusultan and Khatri [14]. The SRAM design is adopted from Young et al. [15].

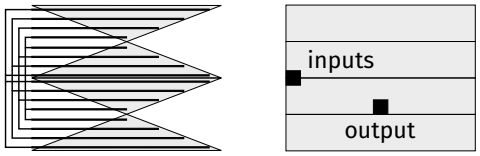


Figure 6: A 6-LUT composed of four stacked 4-LUTs (represented by triangles), with their input connections. In a monolithic 6-LUT, these lengths would amount to 4× the width of the depicted 4-LUT. Assumed pin locations are shown on the right.

obtained for all technology nodes of interest, using the pitch information from Table 1, are reported in Table 3. The table also shows the corresponding resistances of stacked vias connecting buffer outputs to the My wires. We assume that the buffer output pin is at M2 and that the target My layer is M5, meaning that the via needs to traverse the height of two Mx trenches and three Mx vias.

3 AREA AND WIRELENGTH MODELING

Scaling influences delay due to interconnect in two opposing ways: (1) reduction of cross-sectional area and wire separation impacts the resistance and capacitance per unit length (R' and C', see the previous section) and (2) the increase in density reduces the physical lengths of the connections. We have quantified the first phenomenon in the previous section, but without quantifying the second, we cannot assess the influence of technology scaling on the architectural decisions. For instance, the intrinsic delay of a wire depends quadratically on its length, through both R and C, so if length reduces quickly enough with scaling, that could possibly mitigate the negative effects of the smaller pitch. It is hence imperative to have a reasonable model of wire lengths. Unfortunately, commonly used models rely on metrics such as transistor counting [10], assume very loosely defined floorplans [11], and have already been shown to suffer from serious inaccuracies [12], even in older technologies. Thus, they are no longer adequate for scaled nodes, where seemingly minor variation of the length of local connections can result in large impact on their delay. While we cannot hope to match actual layouts, in this section, we attempt area and wirelength models which we believe are suitable to the goal of this work.

3.1 Tile Floorplan

This work explores impact of technology on fundamental decisions that enter the design of the logic fabric. Hence, the architectures considered contain only a single tile type, composed of LUTs, FFs, and routing multiplexers. We adopt a floorplan similar to that used in the Stratix-series architectures, where the LUTs are stacked on top of each other and the routing multiplexers are arranged in columns to their left [5]. When designing the detailed routing architectures, we pay attention that the multiplexers can be evenly divided between the LUTs, similarly to Agilex [13], and match the height of the multiplexers adjacent to one LUT to the height of the LUT itself [5]. A general sketch of the floorplan is shown on the right of Figure 4, while positions of individual multiplexers for one concrete example architecture are shown on the left.

3.2 LUT Dimensions

LUTs play a dominant role in determining the layout of the tile. We based our layout assumptions on a layout due to Abusultan and Khatri [14], shown in Figure 5. It consists of a decoder tree with inputs coming from the left and the output produced on the bottom, horizontally centered. We assume that two-gate-pitch SRAM cells [15] are placed next to each other, above the decoder. To provide the necessary stability [16], the SRAM cells are assumed to be sized as *1:2:3*—i.e., that the NMOS transistors of the two inverters have 3 fins, the PMOS have 1 fin, and that the access transistors have 2 fins [17]. We also assume that there is a one-fin spacing between the NMOS and the PMOS transistors, as allowed by the ASAP7 design rules [17]. This means that the width of the LUT mask amounts to $10 \cdot 2^K$ fin pitches, where K is the LUT input count. The LUT mask is considered to fully determine the width of the entire LUT, as it leaves ample space for increasing the size of the decoder transistors. The height of the LUT is determined as $2K$ gate pitches for the decoder, two gate pitches for the mask, and two more gate pitches for the mask buffers (not shown in the layout sketch).

Because the width of this layout increases exponentially with the increase in the number of LUT inputs, the distance that the input signals need to travel before reaching the most distant decoder transistor quickly becomes intolerable. A similar situation

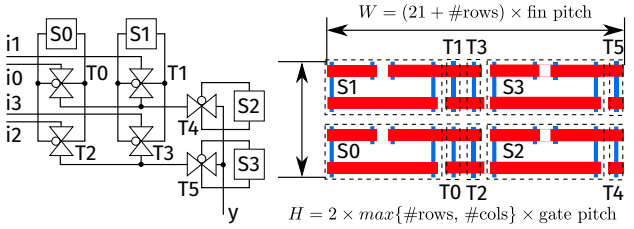


Figure 7: A two-level multiplexer and its layout.

occurs for the output that must reach the routing multiplexers. For this reason, we consider a 4-LUT—size explored by Abusultan and Khatri—to be the largest for which the resistance of the horizontal wires connecting the LUT to the routing multiplexers is acceptable. To create larger LUTs, we stack the required number of 4-LUTs on top of each other. This greatly reduces the distances that the LUT input and output signals need to cross, as can be seen in Figure 6. It also creates more vertical space for the routing multiplexers.

We assume that the flip-flops and the register multiplexing circuitry can fit inside the empty space created by the triangular shape of the (4-)LUT. As an illustration backing this assumption, a flip-flop of the ASAP7 [17] 7-nm standard cell library takes up approximately 116 square gate pitches, whereas the empty space left by a minimally sized 4-LUT equals approximately 600 square gate pitches according to the ASAP7 design rules.

3.3 Routing Multiplexers

We assume that all routing multiplexers are built of transmission gates, which follows the trends visible in Agilex [13]. We also adopt the previous results of Chiasson [18] which showed that already in the planar technologies, it is sufficient that all transistors in the multiplexer transmission gates are minimally sized. A sketch of the layout of a 4:1 multiplexer is shown in Figure 7. We assume that the 2-gate-pitch SRAM cells of Young et al. [15] are stacked on top of each other, one for each column of the multiplexer. Immediately to the right, first-level transmission gates are aligned with the appropriate SRAM cells, followed by the SRAMs of the rows and the transmission gates of the second level, still to the right.

This results in a multiplexer height of twice the maximum between the number of rows and columns of gate pitches, and a width equal to 20 fin pitches for the SRAMs, one for the second-level transmission gates, and one for each multiplexer row. We note that the proposed layout sketch may be slightly optimistic for transmission gates as it may be underestimating the required well spacing.

The connection-block and the switch-block multiplexers require buffers at the output whereas the crossbar multiplexers drive only one LUT input pin which has a buffer of its own, removing the need for additional buffering. We assume that where required, the buffers are placed below the appropriate multiplexer. The increase in the total multiplexer height in the number of gate pitches is determined from the buffer's drive strength, after folding it to pack it into the the horizontal space used by the multiplexer itself.

In general, it is possible to optimize the aspect ratio of the individual multiplexers by adjusting their row and column counts, while optimizing multiplexer placement, so that their combined area is minimized (see Figure 4). This goes beyond the scope of the present work and for the moment we rely instead on optimizing each multiplexer type individually, to minimize the number of used SRAM control bits. Then we populate the columns from the LUT to the left, starting from placing all crossbar multiplexers, then all connection-block multiplexers, and finally all switch-block multiplexers. Each time the LUT height is exceeded, a new column is appended to the left. An example result of application of this simple algorithm is shown on the left of Figure 4.

Table 4: Representative device geometry and nominal supply voltages. All values are taken from Wu et al. [19], apart from F4 which is an interpolation between F5 and F3, and F16 which comes from FreePDK15 [20] and PTM [8].

	F16	F7	F5	F4	F3
gate pitch [nm]	64	56	48	44	41
fin pitch [nm]	40	30	28	24	22
gate length [nm]	20	18	16	15	14
fin height [nm]	26	35	45	50	55
fin width [nm]	12	6.5	6	5.5	5.5
Vdd [V]	0.85	0.75	0.7	0.65	0.65

Table 5: FO4 delays at nominal voltages and at 0.7 V. The delays at 0.7 V of supply voltage are useful to validate the relative speedup, also shown. For this, note that F16 and F7 are two generations apart and that F4 models a possible half-node between F5 and F3. The values indicate that a reasonable speedup roughly around 10% between adjacent nodes is maintained.

	F16	F7	F5	F4	F3
At nominal Vdd [ps]	6.09	4.96	4.69	4.69	4.48
At 0.7 V [ps]	7.02	5.09	4.69	4.52	4.30
Δ		−27%	−8%	−4%	−5%

Table 6: Average input to output delays of a 6-LUT. The values are scaled from the *K6_N10_mem32K_40nm* VTR architecture file [21].

	F16	F7	F5	F4	F3
Average Delay [ps]	94	68	64	64	61

4 DEVICE MODELING

For device modeling, we rely on the PTM [8] and ASAP7 [17] predictive models. We leave the 16 nm PTM models for F16 completely unchanged. For the nodes scaled further down, we update the fin dimensions and the gate lengths of ASAP7 as indicated in Table 4. We leave the remaining parameters which have a less pronounced effect on the drive current unchanged. The same fin and gate pitches are used to convert the wire lengths computed by the scalable model of the previous section to metric units.

5 DELAY EXTRACTION METHODOLOGY

Many aspects related to delay modeling have been described in the previous sections. Here we present the final steps that we use to obtain all the necessary component delays.

5.1 Look-up Tables

As our focus in this work is interconnect, we take the LUT delays reported for Stratix IV in the *K6_N10_mem32K_40nm* architecture distributed with VTR 8.0 [21] and scale them using the equations of Stillmaker and Baas [22], from and to the closest nodes, until F7. From F7 onwards, we assume the scaling of *fanout-of-4 inverter* (FO4) delays at nominal voltage values, reported in Table 5. In doing so, we somewhat underestimate the importance of wires inside the LUTs themselves. We leave addressing this issue for future work. The resulting delays are reported in Table 6.

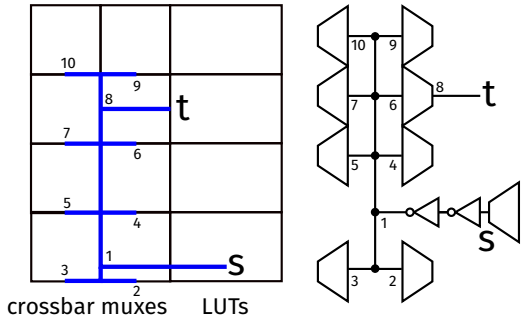


Figure 8: Setup to measure local wire delay.

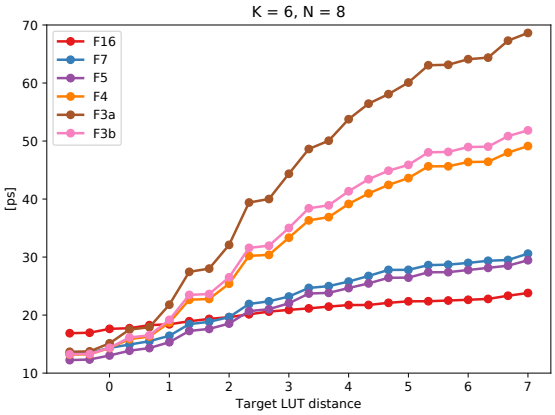


Figure 9: Delays from the output of the bottom LUT in a K6N8 cluster to all LUT inputs reachable through a 50% sparse crossbar.

5.2 Local Wires

Long local connections traced at the highly resistive Mx layers are particularly sensitive to capacitive load of the various multiplexers that connect to them. For this reason, we fairly precisely model all the wires that participate in local signal distribution: We assume that one long vertical line distributes the signal and that shorter horizontal wires bring it to individual multiplexer inputs (Figure 8). We also assume that two loading multiplexers are fully switched on, which corresponds to the typical fanout of a net in a real circuit [23]; one of them is assumed to be in the middle of the line, while the other one is that at which the delay is being measured. The fraction of the loading multiplexers that have only the first level transmission gates turned on is determined as an inverse of the average number of columns in these multiplexers, which is the probability that the one column SRAM which is high is controlling the transmission gate connected to the wire [10]. The horizontal position of the long vertical line is assumed to be at the center between the output of the LUT and the most distant driven multiplexer. We sweep the buffer sizes to minimize delay, assuming that the maximum strength of the first inverter is 5, to avoid overloading the minimally sized transmission gates of the driving multiplexer, and that the second inverter can be at most 5× larger than the fist one. A similar setup is used for measuring the global wire to LUT-input delay, with the only difference being the position of the signal source, the connection-block delay being counted besides that of the crossbar, and the long vertical line positioned at half the distance between the connection-block output and the furthest driven crossbar multiplexer input.

5.3 Global Wires

For measuring global wire delay, we again determine the exact position of the loading multiplexers (see Figure 4). We assume that the My wire brings the signal to the average of the loading multiplexer input coordinates, from which point it is further distributed using a simple rectilinear Mx tree, similar to the one modeling the routing of signals within the cluster (Figure 8).

We also assume that the vertical (horizontal) global wires are positioned at the center of the tile (LUT), horizontally (vertically), and account for the horizontal (vertical) spans needed to access them as well as to take the signal back to the target multiplexers. Because the driver sizes influence the multiplexer stacking, we predetermine them by simulations of a simplified load model, with the same overall buffering approach as the one used for local wires.

6 EXTRACTED LOCAL WIRE DELAYS

In this section we present the resulting delays of local connections, as extracted using the methodology of the last section. The values support our interest to explore anew different cluster sizes.

6.1 The Low Performance of Low Metal Layers

Figure 9 shows the delays from the output of the bottom LUT of an eight 6-LUT cluster (*K6N8*) to each of the other LUT inputs that it can access through a 50% sparse crossbar, for all the considered technology nodes. We can see that for F16 it is business as usual: the delay increase with distance is reasonably modest. As the resistance rises in more advanced nodes, however, the delay increase rate grows rapidly—which is intuitive and somehow predictable. Yet, it is the magnitude of this increase which is interesting: it eventually leads to connections to the other end of the cluster being comparable with and even surpassing that of a 6-LUT. Also, connections between immediately adjacent LUTs, dominated by the logic delay, are faster in newer technologies until F5, when device performance increase decelerates. At the other end, between far-away LUTs, F16 is the technology node which achieves the fastest connectivity—often by far. Finally, it is interesting to note that using an average delay as a single number representing local connection delays (which is often done for architectural research) could be justified for older technologies, as supported by the relatively flat curve of F16. For scaled technologies, however, it is imperative that the CAD tools are aware of the delay disparity.

6.2 Can You Repeat, Please?

One way of mitigating the effect of delay increase due to higher resistance is repeater insertion. To see what an effect this could make, we consider two situations: (1) an optimistic setting where the repeaters can be inserted in the long vertical local wire itself and (2) a more realistic setting where the repeaters are located close to the LUT output, in the cavity created by two constituent 4-LUTs. In both cases, we vary the repeater number and size, assuming that they are located at equal space, aligned with LUT output heights, and that their size equals the size of the second inverter of the main driver. The results for F4 are plotted in Figure 10. We can see that in-line buffer insertion does mitigate the delay to an extent but that it still remains substantial. Yet, as mentioned, it is not realistic

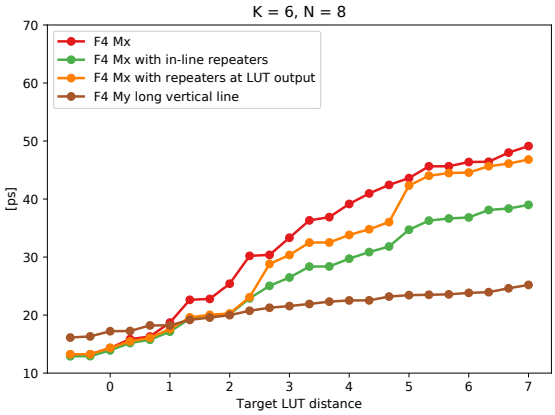


Figure 10: Delays from the output of the bottom LUT in a K6N8 cluster to all LUT inputs reachable through a 50% sparse crossbar, when buffering and when moving the local line to My. The plot shows values for F4 with optimistic (in line) and realistic (at LUT output) repeater insertion; it also shows the effect of raising the long vertical line to My, resulting in a much more dramatic delay reduction.

to assume that each local wire can be buffered in-line, because the repeaters would make it hard to maintain the dense spacing between the lines, even if there were sufficient space left by the crossbar multiplexers. The more realistic buffering scenario shows some benefit but delays to faraway LUTs remain almost unchanged.

6.3 The Rise of Thick Metal Wires

Together with different buffering options, Figure 10 shows the delay resulting from raising the long vertical line to the My layer, with access wires remaining at Mx. The significant delay benefit is immediately apparent: the slope of the curve reduced dramatically and the most distant LUT input can be reached within about a third of the representative LUT delay. We may note, however, that connections to about two LUTs away are still faster or roughly equal when performed on Mx, due to the lower wire capacitance. Similar situation is obtained across the considered scaled technology nodes.

6.4 Thick Metal Wires Are Scarce

The previous section may suggest that lifting the long local connections to My would provide a solution for the delay increase with technology scaling. This may not be such a wise idea, however.

Until now, we have focused on connections between LUTs in the same cluster to provide motivation. Yet, the increase of the delay penalty that intercluster connections need to pay upon entering the cluster, while being dispatched from the connection-block output to the appropriate crossbar multiplexer over increasingly resistive local wires is likely even more important, for intercluster connections occur more often on a typical critical path [16]. Hence, to really see the benefit of raising local wires to the My layers, all of them would need to be raised, and not only those routing the LUT outputs. For a K6N8 cluster, that would mean occupying 40 tracks, while the tile width of such a cluster in F4 (including the routing multiplexers) can typically accommodate about 180 My tracks. This means that about 20% of the available routing space would be locked inside the cluster and unavailable to intercluster signals, potentially inducing a large impact on routability. This impact may be somewhat reduced by the increased number of metal layers in newer technologies, but the recent trends have shown that already in the latest existing technologies it may be desirable to keep as many connections at the lower layers as possible [24].

Table 7: Maximum wire spans for F16–F5 and F3b as a function of the cluster size N. For F4 and F3a, all entries are halved, because the tighter My pitch lowers the distance that can be optimally traversed before buffering.

N	2	4	8	16
V	16	8	4	2
H	8	8	8	8

An alternative solution is again suggested by Figure 10. By observing that communication within a two LUT range is faster at Mx, we may suspect that a smaller cluster (e.g., $N = 2$) could be efficient in satisfying the local communication requirements, while communication with more distant LUTs can be achieved through global routing. This way the performance gain from moving to an upper layer is reduced, but the tracks are not locked within the cluster. This motivates our next set of experiments.

7 ARCHITECTURAL SPACE EXPLORATION METHODOLOGY

We explore cluster sizes of 2, 4, 8, and 16 LUTs. In this section we present the last remaining details of the routing architecture specification and how they are explored. Although we have no way to be comprehensive in the exploration, the aim is to find a reasonable setting to expose interesting trade-offs.

7.1 Crossbar

We compute the number of cluster inputs from the Rent's rule, for the exponent set to 0.8:

$$I = \left\lceil \frac{K \times N^{0.8}}{N} \right\rceil \times N \tag{2}$$

This corresponds closely to Stratix architectures [5]. The division and multiplication by N guarantees that the connection block multiplexers can be evenly divided between LUTs, which eases layout modeling. Unlike the latest Intel architectures [25], we assume the classical setting in which the feedback connections directly enter the crossbar, without passing through the connection-block [5]. The crossbar is assumed to be 50% sparse in delay measurements, to be representative of commercial architectures [5], but is modelled as fully populated in the final VPR experiments, to remove one more possible source of routability impacting the results.

7.2 Routing Channels: General Approach

Similarly to Agilex, we assume that an equal number of wires of each length and direction begin and end at the height of each LUT and that they drive only the switch-blocks at their end [13]. We consider exclusively unidirectional wires occurring in pairs of opposing direction.

7.3 Routing Channels: Maximum Wire Spans

Before exploring exact channel compositions, we determine the maximum lengths of wires for each cluster size in each technology. We do this by finding the longest wire that is still faster than two wires half its length connected in a sequence through a switch block multiplexer. We only consider lengths that are powers of two. The resulting maximum spans are shown in Table 7. We can observe

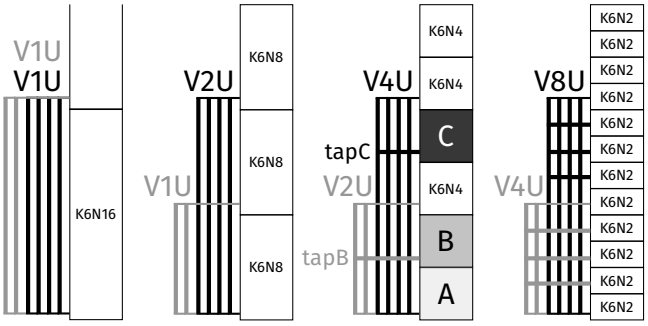


Figure 11: Wire length scaling and taps. Wire lengths are explored for the eight 6-LUT cluster ($K6N8$) to minimize delay. Combinations of wires longer than 1 (black in the figure) are enumerated in a brute-force manner, while to each particular combination, length-1 wires (grey in the figure) are added until the tile width becomes metal bound, including the active area extension due to wire addition (Figure 4). Smaller clusters inherit the solution adapted such as to maintain the physical length of the wires. Taps are added to offset less capable local interconnects. For instance, without *tapB*, the $K6N4$ cluster B would not be reachable from cluster A, while without *tapC*, C would not be reachable from it in one hop. The $K6N16$ cluster also inherits the solution, with length-1 wires replacing those of length < 1, after scaling.

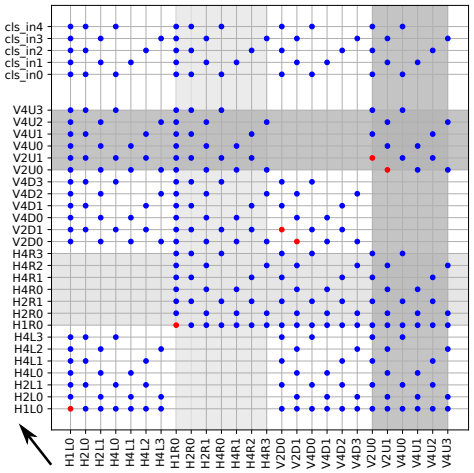


Figure 12: Adjacency matrix of the switch-block pattern of the 3-nm $K6N4$ architecture whose floorplan was shown in Figure 4. All points correspond to connections between wires ending and starting at the same LUT height, apart from those marked in red, which are replicated also to the wires starting at the height of the adjacent LUTs. *V* and *H* stand for vertical and horizontal wires, respectively, while *U, D, L,* and *R* designate the up, down, left, and right direction, respectively. The first number in the name stands for the wire's logical length, while the second is its index within its type.

that with increasing physical height of the tile, the maximum logical distance that makes sense with respect to delay decreases. Similarly, for technology nodes with higher My resistance (F4 and F3a), the maximum spans that can be efficiently realized further reduce.

7.4 Routing Channels: Reference Composition

We determine the exact combination of wires of different lengths by enumerating all possibilities to (partially) fill the channel with wires longer than 1. We limit the exhaustive exploration by forcing the tile to be active-area bound. Then, we pad the remaining space in each combination by length-1 wires until it is the metal which determines the tile dimensions. After each additional wire set is inserted (two wires per LUT, in opposing directions), the multiplexer positions are recomputed (Figure 4), to account for a possible increase of the tile width, due to the increased size of the multiplexers driven by the newly added wires, as well as due to the addition of the new multiplexers driving the inserted wires themselves. Since this process does not influence the capacity of the horizontal channel, as it is fully determined by the height of the stacked LUTs, but it may increase the capacity of the vertical channel, due to the tile width increase, horizontal channel is padded first.

In all cases, vertical wires are assumed to be traced in one My layer and horizontal wires in another. An example of a vertical channel composition, corresponding to the floorplan of Figure 4, is shown in Figure 11, with the padded length-1 wires drawn in grey.

7.5 Routing Channels: Taps and Scaling

To make a fair comparison of different cluster sizes, we do not vary the routing track combination from one size to another. Rather, in each technology, we optimize the channel composition on the architecture with eight 6-LUT clusters (representative of dominant cluster sizes until recently) by placing and routing a subset of benchmarks (see Section 7.7). Then, we scale the logical length of wires for other clusters so that the physical length is maintained. Because the short logical wires may disappear from architectures with smaller clusters, we introduce taps to maintain routability, as suggested in Figure 11. This is conceptually consistent with Agilex wires maintaining the logical length of the large cluster, but allowing multiple entry points into it [13]. This method may be suboptimal for smaller clusters, but is also conservative: if a smaller cluster demonstrates an advantage, this advantage could only increase if its routing channels were individually optimized.

7.6 Switch-Block Patterns

In the most advanced technologies, the switch-pattern must also minimize the distances crossed at the lower metal layers. To address this, we allow connections only between wires ending and starting at the same LUT height (see Figure 4). The exact pattern is formed in such a way that each wire drives at least one wire of each type (length and direction) and is in turn driven by at least one wire of each type, unless it is coming from the direction in which the driving wire is going. To further increase routability, length-1 wires are allowed to also drive length-1 wires in their own direction, starting at the height of the immediately adjacent LUTs. This again draws on concepts reported for Agilex [13].

We allow the LUTs to drive all wires that start at their height. Similarly, all wires ending at a particular LUT height are evenly distributed among the connection-block multiplexers of the LUT. The pattern generation algorithm itself is available in the source code repository accompanying the paper (see Section 9), while Figure 12 gives one concrete example, corresponding to the floorplan of Figure 4, as an illustration.

7.7 Experimental Setup

We rely on the 20 largest MCNC [26] benchmarks in this study. Although there have been concerns about their fitness for architectural exploration, it has been shown that, with respect to basic architectural parameters, they result in the same decisions as larger and newer benchmarks [27]. Besides, we are now essentially reassessing the same decisions that were made on these very benchmarks in older technologies [28]. We exclude *bigkey*, *dsip*, and *des*

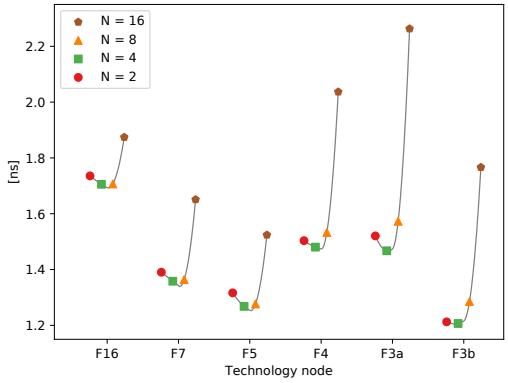


Figure 13: Geometric mean delays of the MCNC benchmarks for different cluster sizes over all considered technology nodes.

which underutilize the minimum-sized FPGA. The 10 smallest of the remaining 17 circuits are used for ranking routing channel compositions, while all 17 are used for the main experiments.

To suppress the experimental noise possibly induced by the choice of the particular channel composition, we perform final experiments on the three best-ranked compositions for which all circuits are routable for all considered cluster sizes. The FPGA grid dimensions in the number of tiles are computed so that the physical width and height of the grid are both approximately equal and minimum, given the requirements of the particular circuit.

We implement all circuits with VTR 8.0 [21], taking the median routed delay of three different placements. Then, for each circuit, the median delay obtained on the three chosen channel compositions is taken as representative, and finally, a geometric mean of such representative delays is computed over all circuits, to represent the particular cluster size in the given technology.

8 A NEW BALLGAME FOR ARCHITECTURE?

Results of the architectural study are reported here. We also discuss how they may relate to the recent trends visible in commercial architectures, as well as what could be their influence on the future outlook.

8.1 A Future of Small Clusters

Figure 13 shows the performance of all cluster sizes at all technology nodes. We can see that until F5 the cluster size ranking is largely maintained as in older technology nodes, with $N = 8$ being the best, or very nearly the best option for optimizing delay [28]. Yet, as interconnect resistance grows, we visualize the trend we were suspecting: smaller clusters emerge as the best solution. The turning point is, in our technology sequence, between F5 and F4, when $N = 4$ takes a clearer lead against $N = 8$, and $N = 2$ surpasses it as well; it is perhaps worth reiterating here that we do not claim this to be an absolute point, since different foundries evolve technology nodes differently and in ways that are impossible for us to know.

Needless to say, we are also painfully aware that these results are comparable to the noise margin typical of such experiments [29], possibly still worsened by the somewhat unconventional and not thoroughly explored routing channel and switch-pattern designs used in the study. Yet, we believe the observed trend, which follows the theoretical expectations, is clear and inescapable.

8.2 What Can the Large Clusters Tell Us?

It is worth observing the behavior of the largest, $N = 16$ cluster as well. Its delay is more sensitive to resistance increase than that of the others, both due to the larger load of the more numerous crossbar multiplexers, and because its height is approaching the range where the capacitive load of the long local wires themselves starts to become an important factor in determining their delay (Figure 11 of Ciofi et al. [6]). We must note here that while the performance trend of $K6N16$ follows the theoretical expectations, it may be slightly disadvantaged compared to other clusters for two reasons: first, due to wire length saturation (see Figure 11), it may receive more vertical wires than other cluster sizes, which increases its tile width and thus negatively impacts the delay as well; second, for the smallest benchmarks, the grid height in the number of clusters that would make the physical grid square drops below that for which VPR can compute the router lookahead maps [30] without adverse edge effects. Hence, some grids retain a higher aspect ratio than desired. Nevertheless, neither of these effects has influence on the delay of the local connections and should thus merely impact the speed of $K6N16$ relative to other clusters, but not its decelerating trend with respect to resistance increase.

8.3 Has the Future Already Happened?

The reader may have noticed that the general design of the parametric architectures used in this work is heavily inspired by the Agilex architecture [13]. Indeed, one does observe there the splitting of the local interconnect into multiple small pieces and the displacement of the connectivity to the global routing instead. This essentially represents a decomposition of the cluster as it is typically understood. In fact, Figure 9 of Chromczack et al. [13] seems very reminiscent of our Figure 10. The difference in the absolute values of the delays might be induced, among other causes, by a higher capacitive load and by larger physical dimensions of their much more complex architecture. Such higher absolute values might in turn have augmented the effects of the rising resistance per unit length, similarly to the case of $K6N16$ in the previous section, and thus accelerated the trends compared to our predictions.

8.4 The Other Side of the Coin

Versal—also a 7-nm architecture—has, however, increased the cluster size from 8 to 32 [24]. A quick extrapolation of Figure 13 to $N = 32$ would most likely raise some concerns. Among the many differences between the simple clusters used in our study and those of Versal, one that stands out is that Versal's $K6N32$-equivalent is not a single tall column of LUTs, but most likely four $K6N8$ columns placed side by side (Figure 4 of Gaide et al. [24]). This greatly reduces the maximum length of the local connections and thus increases their speed. While basing the column on $K6N8$ could be the right choice at 7 nm, even according to the simplified model of our study, with technology scaling further, this may fail to optimize delay, as suggested by Figure 13. Of course, it is not clear whether the delay is the first metric to optimize, given that smaller clusters almost inevitably impact routability—something that falls beyond the scope of the present work, but that Gaide et al. state as an important factor in moving to larger clusters.

8.5 So, What is the Answer?

Let us return to the question that started this paper: Does it ever become faster to route to the next cluster than to the furthest LUT in the same one? According to our measurements, yes, it does, but after all the cost of exiting and entering again the cluster, this happens only for F3a, and by an extremely small margin, contrary to the expectations of Figure 10.[12] A human designer could reduce this additional cost, e.g., by positioning the global wires replacing the intracluster interconnect of the larger clusters closer to the LUTs, but on the other hand, the stark rise in the delay of the local wires distributing intercluster signals to LUT inputs is obvious already with the present modeling[13] and demonstrates itself in the results of Figure 13. Hence, likely the greatest benefit of the aforementioned decomposition of the large cluster in Agilex is in fact the reduction of the penalty that intercluster signals need to pay upon entering the target cluster. The fact that Xilinx architectures do not rely explicitly on connection-blocks and crossbars [31] may have also made the transition to a much larger cluster in case of Versal a lot less costly than the results reported here would suggest.

It looks like one can either cast this problem into its arguably simplest form—reassessing the optimal cluster size ranges, as we did here—or into the much more complex and yet more promising task of determining the optimal local interconnect architecture. Either way, both industrial examples suggest that well-established solutions have to be revisited.

8.6 The End of a Free Lunch

There is perhaps a much more worrying effect to be noticed in Figure 13: not only newer technology nodes bring disarray to well-established architectural tenets, but, even with corrections to past habits, they do not appear to bring any speed advantages. On the contrary. Of course, our architectural exploration is very limited, but we believe that its merit is at least to show how essential a more thorough exploration has become. And the grim image of Figure 13 should fuel innovation because radical upheaval may be, here as in other fields, the only way to profit from the few new technologies still in sight. The days of straightforward evolution are over.

8.7 Custom Technology Nodes for FPGAs

Another solution, somewhat complementary to architecture enhancements, could be to customize the interconnect stack to the very needs of FPGAs. This is addressed to an extent by our speculative F3b node. As evidenced by Table 8, the delay improvement of F3b over F4, due to a more relaxed My pitch and its lower resistance (see Tables 1 and 2), comes at the expense of almost no density increase and reducing the available track count above the tile by more than a third. This means that while F3b offers some tangible speed benefits, perhaps insufficient to justify moving from a 5-nm to a 3-nm node, its utility cannot be properly assessed without taking the adverse impact on routability into consideration.

Customizing the back-end of line to FPGAs remains an important avenue of potential improvement, and, alas, one well beyond the reach of the present work.

[12] It takes 21.4 ps for the output of the LUT to reach a $V4$ intercluster wire in $K6N2$, the delay of the $V4$ wire is 27.8 ps, and a further 18.7 ps are needed for the signal to reach the LUT input, starting from the global wire. In total, 67.9 ps, compared to the maximum delay of 68.6 ps inside $K6N8$.

[13] For instance, a typical value for $K6N8$ at F3a is 39.5 ps, compared to the above 18.7 ps in case of $K6N2$. At F16, the difference is 27.3 ps to 20.7. In the worst case, the impact is even higher, following the trends similar to those of Figure 9.

Table 8: Areas and channels in the various technologies. Area A_m is used by the channels and area A_a is the active area. Each column corresponds to the median-area architecture of the three that were chosen for $K6N8$ for the particular technology. All architectures are slightly metal-area bound.

	F16	F7	F5	F4	F3a	F3b
A_m [μm^2]	393	239	203	154	136	149
Δ		−39%	−15%	−24%	−12%	−3%
A_a [μm^2]	374	230	186	144	124	123
Δ		−38%	−19%	−23%	−14%	−15%
H-tracks	320	288	272	352	336	208
V-tracks	192	144	144	176	176	112

8.8 And What about Density?

Table 8 shows the evolution of tile area and cumulative channel sizes over the technology nodes. Compared to performance improvement, density scaling seems a lot more promising. In the most advanced nodes, observed area reduction even meets the typical expectations (about 40% per node; remember that F4 is an intermediate node). We must note, however, that the reported area is influenced by the employed methodology intended to maximize the available wiring bandwidth above the tile (see Section 7.4). In reality, routability requirements of more complex circuits and the availability of multiple My layers must be taken into account as well. We leave this for future work.

Table 8 reports only the data for $N = 8$. For $N = 4$ and $N = 2$, the metal area and horizontal channel track width are two and four times smaller, respectively, while the vertical channels maintain the width, due to the way in which they are composed (see Figure 11). The only slight variation exists in the active area, due to the different crossbar and connection-block multiplexer size and count. The aforementioned wire length saturation effect causes the vertical channels to be somewhat wider for $N = 16$, and its area a bit more than twice that reported in Table 8.

9 CONCLUSIONS

Technology scaling is predicted to continue for at least a couple more nodes. Yet, the scaling of some parts of the technology stack are having dramatic effects on manufacturable circuits and these results are likely to worsen. In this paper we take a stab at the potential impact on FPGAs of the back-end of line evolution in contemporary and future technology nodes. The landscape we discover is fairly bleak, with well-established architectural beliefs shaken and future performance improvements uncertain. Certainly, our exploration is quite imperfect in a number of ways: we limit ourselves to the reconfigurable logic of FPGAs, we consider a single layout floorplan for our FPGA tile, we restrict ourselves to some fairly simple ways of employing the metal stacks for interconnect channels, and we explore limited strategies to exploit the available channels. Yet, we believe our results are still representative of actual and future trends; a feeling supported by recognizing some of the effects we see in recent commercial products. We hope that this work will at least revive the interest in FPGA architecture research and will ultimately lead our community to the revolutionary architectural innovations which will be needed by FPGAs to stay competitive against other computational platforms.

The source code used to produce the results of this study is available at `https://github.com/EPFL-LAP/fpga21-scaled-tech`.

REFERENCES

[1] S.-Y. Wu, C. Y. Lin, M. Chiang, J. Liaw, J. Cheng, S. Yang, M. Liang, T. Miyashita, C. Tsai, B. Hsu *et al.*, "A 16nm FinFET CMOS technology for mobile SoC and computing applications," in *Proceedings of the 2013 IEEE International Electron Devices Meeting*, Washington, DC, USA, Dec. 2013, pp. 9.1.1–4.

[2] S. Wu, C. Y. Lin, M. C. Chiang, J. J. Liaw, J. Y. Cheng, S. H. Yang, C. H. Tsai, P. N. Chen, T. Miyashita, C. H. Chang, V. S. Chang, K. H. Pan, J. H. Chen, Y. S. Mor, K. T. Lai, C. S. Liang, H. F. Chen, S. Y. Chang, C. J. Lin, C. H. Hsieh, R. F. Tsui, C. H. Yao, C. C. Chen, R. Chen, C. H. Lee, H. J. Lin, C. W. Chang, K. W. Chen, M. H. Tsai, K. S. Chen, Y. Ku, and S. M. Jang, "A 7nm CMOS platform technology featuring 4th generation FinFET transistors with a 0.027 um2 high density 6-T SRAM cell for mobile SoC applications," in *Proceedings of the 2016 IEEE International Electron Devices Meeting*, San Francisco, CA, USA, Dec. 2016, pp. 2.6.1–4.

[3] G. Yeap, S. S. Lin, Y. M. Chen, H. L. Shang, P. W. Wang, H. C. Lin, Y. C. Peng, J. Y. Sheu, M. Wang, X. Chen, B. R. Yang, C. P. Lin, F. C. Yang, Y. K. Leung, D. W. Lin, C. P. Chen, K. F. Yu, D. H. Chen, C. Y. Chang, H. K. Chen, P. Hung, C. S. Hou, Y. K. Cheng, J. Chang, L. Yuan, C. K. Lin, C. C. Chen, Y. C. Yeo, M. H. Tsai, H. T. Lin, C. O. Chui, K. B. Huang, W. Chang, H. J. Lin, K. W. Chen, R. Chen, S. H. Sun, Q. Fu, H. T. Yang, H. T. Chiang, C. C. Yeh, T. L. Lee, C. H. Wang, S. L. Shue, C. W. Wu, R. Lu, W. R. Lin, J. Wu, F. Lai, Y. H. Wu, B. Z. Tien, Y. C. Huang, L. C. Lu, J. He, Y. Ku, J. Lin, M. Cao, T. S. Chang, and S. M. Jang, "5nm CMOS production technology platform featuring full-fledged EUV, and high mobility channel FinFETs with densest 0.021μm2 SRAM cells for mobile SoC and high performance computing applications," in *Proceedings of the 2019 IEEE International Electron Devices Meeting*, San Francisco, CA, USA, Dec. 2019, pp. 36.7.1–4.

[4] D. Prasad, S. T. Nibhanupudi, S. Das, O. Zografos, B. Chehab, S. Sarkar, R. Baert, A. Robinson, A. Gupta, A. Spessot *et al.*, "Buried power rails and back-side power grids: Arm® CPU power delivery network design beyond 5nm," in *Proceedings of the 2019 IEEE International Electron Devices Meeting*, San Francisco, CA, USA, Dec. 2019, pp. 19.1.1–4.

[5] D. Lewis, D. Cashman, M. Chan, J. Chromczak, G. Lai, A. Lee, T. Vanderhoek, and H. Yu, "Architectural enhancements in Stratix V™," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, USA, Feb. 2013, pp. 147–56.

[6] I. Ciofi, A. Contino, P. J. Roussel, R. Baert, V.-H. Vega-Gonzalez, K. Croes, M. Badaroglu, C. J. Wilson, P. Raghavan, A. Mercha, D. Verkest, G. Groeseneken, D. Mocuta, and A. Thean, "Impact of wire geometry on interconnect RC and circuit delay," *IEEE Transactions on Electron Devices*, vol. 63, no. 6, pp. 2488–96, May 2016.

[7] S.-C. Wong, G.-Y. Lee, and D.-J. Ma, "Modeling of interconnect capacitance, delay, and crosstalk in VLSI," *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, no. 1, pp. 108–11, Feb. 2000.

[8] "Predictive technology model," http://ptm.asu.edu/, accessed: 26.08.2020.

[9] I. Ciofi, P. J. Roussel, Y. Saad, V. Moroz, C.-Y. Hu, R. Baert, K. Croes, A. Contino, K. Vandersmissen, W. Gao, P. Matagne, M. Badaroglu, C. J. Wilson, D. Mocuta, and Z. Tőkei, "Modeling of via resistance for advanced technology nodes," *IEEE Transactions on Electron Devices*, vol. 64, no. 5, pp. 2306–13, Apr. 2017.

[10] C. Chiasson and V. Betz, "COFFE: Fully-automated transistor sizing for FPGAs," in *Proceedings of the 2013 International Conference on Field-Programmable Technology*, Kyoto, Japan, Dec. 2013, pp. 34–41.

[11] G. Zgheib and P. Ienne, "Automatic wire modeling to explore novel FPGA architectures," in *Proceedings of the 2016 International Conference on Field-Programmable Technology*, Xi'an, China, Dec. 2016, pp. 181–84.

[12] F. F. Khan, "Towards accurate FPGA area models for FPGA architecture evaluation," Ph.D. dissertation, Ryerson University, 2017.

[13] J. Chromczak, M. Wheeler, C. Chiasson, D. How, M. Langhammer, T. Vanderhoek, G. Zgheib, and I. Ganusov, "Architectural enhancements in Intel® Agilex™ FPGAs," in *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Monterey, CA, USA, Feb. 2020, p. 140–49.

[14] M. Abusultan and S. P. Khatri, "A comparison of FinFET-based FPGA LUT designs," in *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, Houston, TX, USA, May 2014, pp. 353–58.

[15] S. P. Young, Y. Song, and N. Chong, "Two gate pitch FPGA memory cell," US Patent 9 177 634 B1, 2015.

[16] D. Lewis and J. Chromczak, "Process technology implications for FPGAs," in *Proceedings of the 2012 International Electron Devices Meeting*, San Francisco, CA, USA, Dec. 2012, pp. 25.2.1–4.

[17] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "ASAP7: a 7-nm FinFET predictive process design kit," *Microelectronics Journal*, vol. 53, pp. 105–15, Jul. 2016.

[18] C. Chiasson, "Optimization and modeling of FPGA circuitry in advanced process technology," Master's thesis, University of Toronto, 2013.

[19] T. Wu, H. Luo, X. Wang, A. Asenov, and X. Miao, "A predictive 3-D source/drain resistance compact model and the impact on 7 nm and scaled FinFETs," *IEEE Transactions on Electron Devices*, vol. 67, no. 6, pp. 2255–62, May 2020.

[20] K. N. Bhanushali and W. R. Davis, "FreePDK15: An open-source predictive process design kit for 15nm FinFET technology," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, Monterey, CA, USA, Mar. 2015, pp. 165–70.

[21] K. E. Murray, O. Petelin, S. Zhong, J. M. Wang, M. Eldafrawy, J.-P. Legault, E. Sha, A. G. Graham, J. Wu, M. J. P. Walker, H. Zeng, P. Patros, J. Luu, K. B. Kent, and V. Betz, "VTR 8: High-performance CAD and customizible FPGA architecture modelling," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 13, no. 2, pp. 9:1–60, May 2020.

[22] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180nm to 7nm," *Integration*, vol. 58, pp. 74–81, Jun. 2017.

[23] M. Hutton, "Characterization and parameterized generation of digital circuits," Ph.D. dissertation, University of Toronto, 1997.

[24] B. Gaide, D. Gaitonde, C. Ravishankar, and T. Bauer, "Xilinx adaptive compute acceleration platform: Versal architecture," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Monterey, CA, USA, Feb. 2019, pp. 84–93.

[25] D. Lewis, G. Chiu, J. Chromczak, D. Galloway, B. Gamsa, V. Manohararajah, I. Milton, T. Vanderhoek, and J. Van Dyken, "The Stratix™ 10 highly pipelined FPGA architecture," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Monterey, CA, USA, Feb. 2016, pp. 159–68.

[26] S. Yang, "Logic synthesis and optimization benchmarks user guide, version 3.0," Microelectronics Center of North Carolina, Technical Report, Jan. 1991.

[27] G. Zgheib and P. Ienne, "Evaluating FPGA clusters under wide ranges of design parameters," in *Proceedings of the 27th International Conference on Field Programmable Logic and Applications*, Ghent, Belgium, Sep. 2017, pp. 1–8.

[28] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deep-submicron FPGA performance and density," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 3, pp. 288–98, Mar. 2004.

[29] R. Rubin and A. DeHon, "Timing-driven pathfinder pathology and remediation: Quantifying and reducing delay noise in VPR-pathfinder," in *Proceedings of the ACM/SIGDA 19th International Symposium on Field Programmable Gate Arrays*, Monterey, CA, USA, Feb. 2011, pp. 173–76.

[30] O. Petelin and V. Betz, "The speed of diversity: Exploring complex FPGA routing topologies for the global metal layer," in *Proceedings of the 26th International Conference on Field Programmable Logic and Applications*, Lausanne, Switzerland, Aug. 2016, pp. 1–10.

[31] M. B. Petersen, S. Nikolić, and M. Stojilović, "NetCracker: A peek into the routing architecture of Xilinx 7-Series FPGAs," in *Proceedings of the 2021 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Feb. 2021.