Layout Considerations of Basic Arithmetic Logic Units Using an N-layer 3D Nanofabric Process Flow

Edouard Giacomin[1], Juergen Boemmels[2], Julien Ryckaert[2], Francky Catthoor[2,3] and Pierre-Emmanuel Gaillardon[1]
[1]University of Utah, Salt Lake City, UT, USA
[2]IMEC, Leuven, Belgium
[3]KU Leuven, Leuven, Belgium
edouard.giacomin@utah.edu

***Abstract*—In the past few years, novel fabrication schemes such as parallel and monolithic 3D integration have been proposed to keep sustaining the need for more powerful integrated circuits. By stacking several devices, wafers, or dies, the footprint, delay, and power can be decreased when compared to traditional 2D implementations. While parallel 3D does not enable very fine-grained vertical connections, monolithic 3D currently only offers a limited number of transistor tiers due to the high cost of the additional masks and processing steps, limiting the benefits of using the third dimension. In this paper, we introduce an innovative planar circuit netlist and layout approach, which enables a new 3D integration flow called *3D Nanofabric*. The flow, consisting of N identical vertical tiers, is aimed at single instruction multiple data processor *Arithmetic Logic Units* (ALUs). By using a single metal routing layer for each vertical tier, the process flow is significantly simplified since multiple vertical layers can potentially be patterned at once, similar to the 3D NAND flash process. In our study, we thoroughly investigate the layout constraints arising from the Nanofabric flow and the unique metal layer rule and propose several ways to overcome them. We then show that by stacking 32 layers to build a 32-bit ALU, the footprint is reduced by $8.7\times$ when compared to a conventional 7nm FinFET implementation.**

I. Introduction

For many years, the semiconductor industry has continued to scale down the *Metal-Oxide-Semiconductor Field-Effect Transistor* (MOSFET) to increase the number of devices per area unit, thus enhancing the performances of *Integrated Circuits* (ICs). Novel transistor topologies have emerged in the past few years as an alternative to planar transistors, such as FinFETs [1]. They allow better electrostatic control, decreased leakage, and reduced short-channel effects, improving electrical performances. However, FinFETs still suffer from the short-channel effect, as well as other physical limitations, such as quantum effects [2], and can not be scaled indefinitely. Therefore, alternative routes are being investigated to sustain the continuous need for more performant ICs for a given footprint.

In particular, in recent years, three-dimensional integrated circuits (3D ICs) have been proposed [3]–[10], [14]. A 3D IC is an integrated circuit manufactured by stacking silicon wafers, dies, or transistors. They are then interconnected vertically to achieve performance improvements at reduced power, thanks to the shorter interconnects when compared to conventional 2D approaches. Furthermore, stacked device layers increase the number of transistors per unit footprint without requiring costly feature size reduction. In the past few years, two 3D integration schemes have emerged: parallel 3D [3]–[5] where wafers or dies are stacked and interconnected using *Through Silicon Vias* (TSVs) and bonding techniques, and monolithic 3D [6]–[10], [14], where multiple layers of transistors and/or memory are deposited sequentially on top of one another on the same starting substrate.

While the interconnection density of parallel 3D integration is limited by the large size of the TSVs, monolithic 3D allows a finer integration granularity. However, state-of-the-art monolithic 3D works [6]–[8], [10], [14] are currently constrained by the numbers of active tiers (2-4), limiting the potential offered by 3D integration. In this paper, we introduce a new 3D integration scheme, called *3D Nanofabric*. The *Nanofabric* consists of N identical vertical tiers, each realizing the same logic function. As such, it is aimed at *Single Instruction Multiple Data* (SIMD) processor *Arithmetic Logic Units* (ALUs), where each vertical tier is one ALU bit. We propose here to use a single metal routing layer at each vertical tier, to greatly simplify the process flow, as multiple vertical layers can potentially be patterned at once. While we are aware of the challenges 3D technologies bring, such as thermal aspects including cooling, power distribution, yield, and reliability, those are out of the scope of the paper and are part of ongoing and future work. Instead, the goal of this paper is to first focus on the layout constraints and prove that conventional designs can be integrated into the *3D Nanofabric* flow, given the constraints of a planar graph without crossing wires within a vertical tier. The contributions of this paper are:

- We introduce a novel 3D design style using a very simplified set of masks and describe a possible process flow that could enable a sufficiently high yield across all layers.
- We investigate the physical design constraints arising from the single metal layer rule and propose solutions to planarize standard cells so they can be used in the proposed *Nanofabric*.
- We show that, at the circuit level, by stacking up to

978-1-7281-3915-9/19/$31.00 ©2019 IEEE

32 layers to build a larger 32-bit ALU, the footprint is reduced by $8.7\times$ when compared to a 2D planar 7nm FinFET implementation.

The rest of this paper is organized as follows: Section II, presents related work. Section III briefly presents the proposed 3D Nanofabric concept and describes a possible technology process flow. Section IV discusses the different physical design constraints and proposes several solutions. Section V shows experimental footprint results. Section VII concludes this paper.

II. Related Work

Several works on parallel 3D integration have been proposed [3]–[5], where devices on separate dies or wafers are fabricated in parallel and followed by a bonding and interconnection step. The stacking can be done with μ-bumps and TSVs [4], which are vertical connections that pass entirely through the wafer or the die. While TSVs allow a fine-grained integration of several dies into a single 3D stack, they also consume a significant area ($\sim\mu m$ pitch), which does not allow them to be used to realize very fine grain interconnects.

Other works focused on monolithic 3D (also called sequential 3D), where multiple transistor tiers and/or memory cells are vertically stacked sequentially on the same starting substrate [6]–[10], [14]. Monolithic 3D opens several opportunities, such as stacking 2 nodes $N-1$ instead of a node N [9], in a Logic-on-Logic or Memory-on-Logic way [6], or more disruptive approaches where emerging technologies can be stacked on top of CMOS [10], [14]. However, only four active tiers have been demonstrated up to this date [14], limiting the benefits of using the third dimension.

On the other hand, 3D NAND flash, consisting of a highly repetitive mask set, has also been introduced [11], [12] for memory applications. Recently, up to 128 vertical layers have been demonstrated for the 3D NAND [12], resulting in a minimal footprint per stored bit.

Our proposed *3D Nanofabric* aims at a similar objective as the 3D NAND, namely, to exploit repetitive vertical layers to decrease the footprint, but is targeted at logic applications. However, this can only be achieved by proposing a circuit netlist topology and layout that relies solely on a single layer where the device channel, poly, and metal wires are all embedded without any other crossing than the gate on top of the device channel. To the best of our knowledge, that is a crucial challenge that has not been enabled by any other proposed netlist approach.

III. Proposed *3D Nanofabric* Concept

In this section, we briefly summarize the proposed *3D Nanofabric* concept and then present a possible fabrication flow.

A. General Overview

The proposed *3D Nanofabric* consists of N identical stacked vertical tiers, depicted in Fig. 1 (a). In other words, the *3D Nanofabric* is a 3D ALU where each tier is an ALU bit. Hence, it is aimed at realizing SIMD processor datapaths, where the datapath is composed of an array of 3D ALUs. The way the *Nanofabric* communicates with the other parts of the processor (control, memory, *etc.*) is out of the scope of this paper and is one of our current studies. To be able to stack many layers, we propose here to use a very restricted set of masks (i.e., only a single metal routing track), which allows multiple layers to be patterned at once during fabrication, as it will be explained in Section III-B. As shown in Fig. 1 (a), the global signals which are shared among all the vertical layers, such as the select signals $sel[0:M]$ (M depending on the number of operations the ALU can realize) or V_{dd} and V_{ss}, are provided through vertical pillars. The other signals (inputs and outputs of each ALU slice) are fed independently to each vertical layer from the side, using staircase-like structures similar to 3D NAND [11] chips.

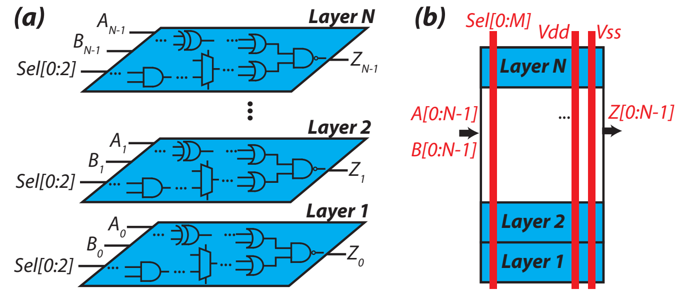

Fig. 1. *3D Nanofabric* concept: (a) Identical transistor tiers; (b) Cross-section general organization.

B. Possible Nanofabric Process Flow

In this Section, we briefly describe a possible technological solution for manufacturing the proposed *3D Nanofabric*. The flow, based on the Coventor® modeling software, has been used to derive the design and layout rules which are presented in this section and which have been employed to obtain the results of Section V. Note that a more complete and thorough process flow study is out of the scope of this paper. While a simple solution would be to create the structure sequentially layer-by-layer, this would not be cost-effective at all as most steps would have to be repeated for each layer. Instead, we propose a solution that only uses a single metal routing layer and patterns multiple vertical layers at once. When patterning multiple layers, special care must be taken about the interaction of the different layers, *e.g.*, we should not destroy any active area by patterning the gate. This means that many of the operations need to happen from the side, as it will be explained later. Furthermore, we need to make sure that the structure always stays mechanically connected to the bulk, and that we never completely undercut a structure.

The processing starts by depositing the layer-stack: for each vertical layer, we deposit an active layer, a sacrificial layer which will become the gate (dummy-Gate), and an interlayer-dielectric. While there are multiple possible options for creating active layers, we propose here to use layer transfer of crystalline silicon, as it is done for *Silicon On Insulator* (SOI) processes. Those SOI-like silicon devices are

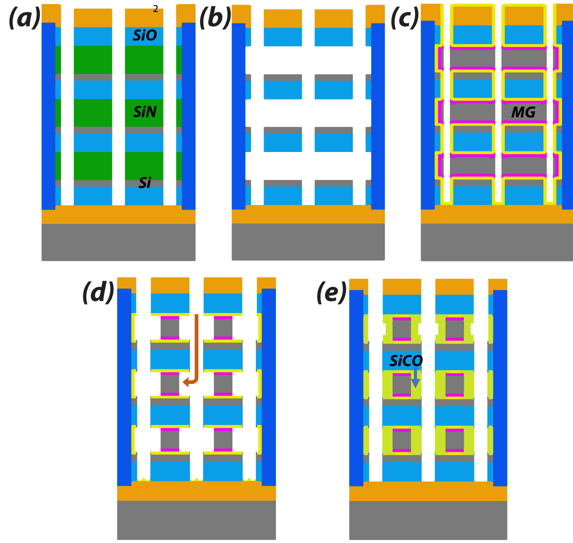

Fig. 2. Cross-section of the gate patterning: a) Gap cut where source and drain will be formed; b) Dummy-gate removal; c) Gate-oxide and metal-gate filling; d) Metal recess; e) Spacer fill and etch-back.

well understood and have good electrical characteristics. The gate patterning process, which happens from the sides, is shown in Fig. 2. In fact, there is no gate layer in the design as it is formed in a purely collateral fashion. Instead, the gate is defined by the extended Source-Drain region and is formed indirectly. The layer *ANTIGATE* is surrounding every gate at a fixed distance (in classical terms, the spacer-width). As it is also needed to "repair" the interlayer-dielectric, the *ANTIGATE* layer has a fixed width. As gates will form on both sides of the *ANTIGATE*, a dielectric layer *OXWALL* is used in order to prevent the formation of an unwanted gate but also gives mechanical stability to the structure. First, (Fig. 2 (a)), a gap is cut where the source and drain regions will be formed, by employing a high-aspect-ratio etch. By using a selective isotropic etch process, the dummy-gate material is then removed (Fig. 2 (b)). As we are removing a lot of material between the layers, we need to make sure that every layer is always mechanically supported. This is achieved by the design rule that every gate-island is touching an *OXWALL*. A high-k gate-oxide and then a metal-gate are deposited in the space left by the dummy-Gate (Fig. 2 (c)). To form the spacers, we first recess the metal (Fig. 2 (d)) by an isotropic metal etch and fill the formed cavity with the spacer material. The excess material in the *ANTIGATE*-trenches is removed by an anisotropic high-aspect-ratio etch using the hardmask (Fig. 2 (e)). The active patterning process is not explained here, as it is similar to the gate patterning. The final step is the formation of the metal. For the vertical connection, holes are etched through the layer-stack where *CONT_VERT* layout requests them. Also, for *METALCUT*, holes are formed, which are used as filling ports for the metal lines. The metal lines are filled over the whole length of the line through these filling ports. Therefore, a very conformal deposition is needed to avoid pinch-off. In the last step, the metal is removed from the *METALCUT* plugs and refilled with a dielectric, to cut the metal line at this location.

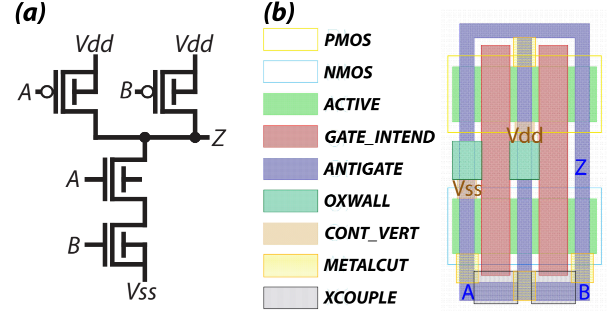

Fig. 3. NAND2: (a) Schematic; (b) Layout with layer legend.

To illustrate the proposed flow, the layout of a conventional NAND2 gate is depicted in Fig. 3 (b). For this paper , we consider FDSOI devices using a gate length $L = 24nm$ and a gate pitch $C_{pp} = 48nm$, similar to current 7nm technologies. As discussed, each gate (*GATE_INTEND*) is surrounded by an *ANTIGATE* layer. As such, some metal breakers are required (METALCUT) to achieve all the different connections. The *XCOUPLE* layer is used to provide a connection between the gate and the routing layer (*ANTIGATE*). Two *OXWALL* squares can be observed, which are in direct contact with the gates to mechanically support the vertical structure, and also act as metal routing breakersBesideses, the V_{dd} and V_{ss} supply lines are fed through vertical pillars (brown *CONT_VERT* squares) to the logic gate. Note that, as explained earlier, the *GATE_INTEND* layer is not a physical mask as the gates are formed indirectly throughout the flow. This layer is only shown here for layout purposes to ease the design step.

IV. Layout Constraints and Solutions

In this section, we first describe the different layout constraints arising from the *3D Nanofabric* process flow. We then present solutions to overcome those and produce logic designs using a single metal layer.

A. Layout Constraints

The main layout limitation is that only a single metal routing track can be used within the *Nanofabric*, which considerably restricts the physical design. This means that when designing, no upper metal level layers can be used in case of metal crossing in high congestion areas. While any crossing possibility means that complex gates, such as XOR2 or the FA are challenging to design, some solutions are proposed in the next section. Besides, it is not possible to have the metal routing layer spanning across unrelated gates or active layers, as it is the case in conventional 2D technologies. As explained in Section III-B, the $GATE_INTEND$ layer is directly derived from the *ANTIGATE* layer, so they are not distinct from a processing point of view. As such, it is strictly impossible to have the *ANTIGATE* layer spanning on the $GATE_INTEND$ or $ACTIVE$ layers, which adds some restrictions. Finally, the $GATE_INTEND$ layer has to be surrounded on every four sides by the *ANTIGATE* layer. As a

result, for complex cells, some breakers have to be employed to achieve distinct connections on the different source and drain sides.

B. Layout Solutions to Avoid Metal Crossing

Here, we present the algorithm used to overcome the single metal layer and other layout constraints of the *Nanofabric*. We first describe each step with examples and then provide the complete algorithm.

1) Step 1: Resolving Loops at the Cell Level: The first step to resolve metal crossing is to make sure that no metal loop is present within a single logic cell. To do so, several techniques are employed:

Due to the non-conventional way of designing logic cells, there is more freedom to move the transistors vertically and horizontally, instead of having fixed top *p-well* and bottom *n-well* zones as in traditional 2D designs. For complex logic gates like the XOR2, the transistor sharing the same gate signals (mainly A and B) can be stacked on top of each other to relieve congestion within the cell, as shown in Fig. 4 (a). Note that unlike conventional design styles, there is no fixed height for the different logic cells, as complex gates such as the XOR2 will require a larger height due to the transistor stacking. Therefore, more different design styles are possible for a given cell, depending on the desired shape and the internal cell structure. Global signals, including V_{dd}, V_{ss}, or the ALU control signals, which are shared among all the vertical layers to perform the same logic function, are provided to the *Nanofabric* through vertical pillars. In particular, unlike conventional 2D designs, the standard cell power supply grid lines are removed. This relieves metal routability since those signals will not block the metal routing layer. Also, the primary inputs and outputs of the *Nanofabric* are also fed through vertical pillars. However, those are not shared among all the vertical tiers, as each layer requires separate inputs and outputs. As a result, similarly to the 3D NAND process [11], staircase-like structures are employed to convey all the signals to the appropriate tiers independently.

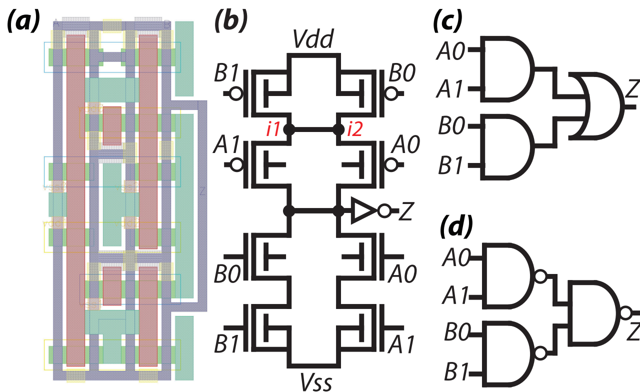

Fig. 4. (a) XOR2 logic gate layout using the proposed *Nanofabric* rules; AO22 gate schematic: (b) Transistor-level based design; (c) Gate-level based design using AND/OR gates; (d) Gate-level based design using NAND gates.

A solution to design complex gates is to use gate-level based designs instead of transistor-level based designs. For the AO22 gate, which transistor level-based design is depicted in Fig. 4 (b), the different connections, notably *i1* and *i2*, make it impossible to be designed using the proposed *Nanofabric* flow. Since each gate has to be surrounded by the metal layer, and there is only a single metal layer, these kinds of connections where 4 transistors share the same drain or source are particularly challenging. However, using the gate-level based design shown in Fig. 4 (c) greatly simplifies the routing and makes it possible, by merely cascading basic gates (NAND2, NOR2, *etc.*). While the gate-level based design uses more transistors (18 instead of 10), it can be rearranged using De Morgan's equation, as shown in Fig. 4 (d), and only uses 2 more transistors than the transistor-level based implementation. Also, due to the boolean commutativity rules, the gate inputs can be re-ordered to facilitate routing.

2) Step 2: Resolving Loops at the Netlist Level: Once all logic gates do not contain any internal metal loop, they are used to build a complete ALU. To resolve any additional metal loop in the netlist when connecting the different gates, duplicated gates can be used. As illustrated in Fig. 5 (a), the input arrangement of the AO22 gate is causing a metal crossing, and there is no way to simply move the gates to overcome this issue. This metal crossing can be resolved by duplicating the OR2 gate (in blue) on the side. As depicted in Fig. 5 (b), its output is now able to be connected to the AO22 gate without being confined, as it was the case before. Note that while it brings an area overhead, duplicating logic gates will always resolve any crossing issue as the gates can be duplicated up to the netlist primary inputs.

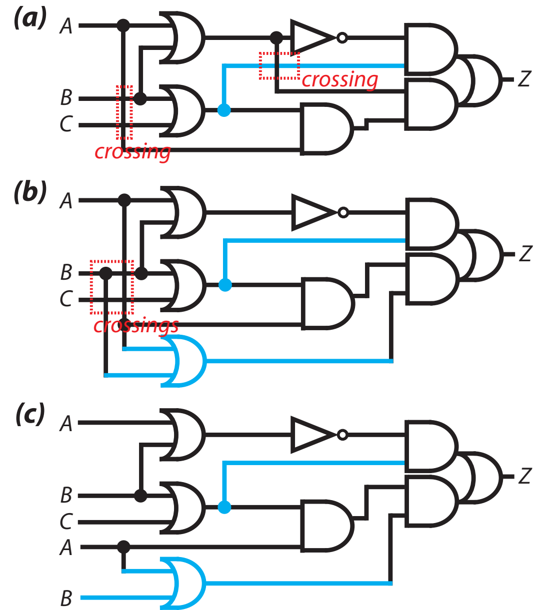

Fig. 5. Logic circuit schematic: (a) Containing 2 metal crossings; (b) Alleviating 1 metal crossing through duplicated inputs from the staircase; (c) Alleviating both metal crossings by using a duplicate gate and duplicated inputs from the staircase.

3) Step 3: Duplicating Signals Through Staircases and Vertical Signals: As explained in Section III-A, each 2D layer will receive its primary inputs from its sides. However, the first logic level of the ALU may require some inputs to be fed to several parallel gates, implying possible metal crossing, as shown in Fig. 5 (b). In this example, input B is driving three parallel gates. However, since there is no way to place them next to each other, the B metal wire has to cross inputs A and C. Since the primary inputs of each 2D layer are provided through a vertical staircase, they can be duplicated to be fed to more gates in the ALU. As depicted in Fig. 5 (c), by duplicating the primary inputs A and B, both metal crossings can be resolved. Besides, as using step 2 might also result in several duplicated primary inputs, the staircase will be able to feed them to the ALU while avoiding metal crossing. As the control signals are provided through vertical pillars, those can also be easily duplicated if they need to control several logic gates.

C. Single Metal Layer Layout Algorithm

In this section, we present the complete algorithm to produce the layout for an ALU netlist while only using a single metal layer. It consists of all the previous layout solutions combined. The algorithm starts from one of the last logic gate (producing an output) and propagates backward through the netlist. For each gate, it first solves the internal gate crossings, before solving the metal loops at the netlist level (between several gates). Once all the gates of a given logic level have been treated, it moves to the previous logic level until it reaches the primary inputs. If necessary, those primary inputs are duplicated through the staircases or the vertical signals. Here, we assume that the netlist does not contain feedback loops. While feedback loops are generally present in sequential circuits, the goal here is to design combinational ALUs for SIMD processors, so it is unlikely to happen. Besides, a proper synthesis of the ALU function would also get rid of the feedback loops within the netlist.

Algorithm 1: *3D Nanofabric* gate placement.

```
Starts at the output node (last level of logic depth);
Logic_level = Get_Total_Nb_Logic_Levels();
while (Logic_level != 1) do
    Number_gates =
      Get_Current_Logic_Level_Nb_Gates();
    while (Number_gates != 1) do
        if Current_gate has internal crossings then
            Duplicate_Primary_Input();
            Use_Gate_Based_Logic_Cell();
        else
            Use_Transistor_Based_Logic_Cell();
        end
        Number_gates = Number_gates - 1;
    end
    if Crossing between gates then
        Use_Duplicate_Gate();
    end
    Logic_level = Logic_level - 1;
end
Duplicate_Signals();
```

V. Experimental Results

A. Experimental Methodology

For the footprint evaluations, we developed an in-house PDK for the *3D Nanofabric* flow, following the technological assumptions presented in Section III-B. For the 2D baseline, we considered 2 cases: (a) the ASAP 7nm FinFET design kit from ASU [13] and (b) an in-house FinFET IN7 node. For a fair area comparison, transistors are minimum sized in all cases. For all cases, the ALU area values were obtained after synthesis by using the complete available logic libraries. For the 3D case, an extra step is performed to draw the layout by hand following the novel approach described above.

B. Logic Gate Area Comparison

Table I shows the area of a few conventional logic gates, using the proposed *3D Nanofabric* flow when compared to other technologies. As expected, when compared to a highly and aggressively optimized IN7 library, using the *3D Nanofabric* process brings an area overhead (1.8× on average) due to the non-crossing rule, which requires extra transistors or spacing for complex gates. In particular, the area overhead is even more important for gate-level based cell such as the AO22 gate due to the additional transistors. Note that the logic gate area is reduced (17% in average) when compared to ASAP7 since the proposed *Nanofabric* allows us to design compact gates, as the $nmos$ and $pmos$ transistors can be placed closer to each other. In addition, the significant difference between ASAP7 and IN7 is due to the fact that IN7 is equivalent to a commercial foundry 5nm technology node, due to its aggressive dimensions and multiple design boosters enabling a 6-track library, while ASAP7 can only achieve a 7.5 track instance.

TABLE I
Logic gates area (in μm^2) using ASAP7, IN7 and the proposed *3D Nanofabric* process.

Gate	ASAP7	IN7	*3D Nanofabric*
INVD1	0.044	0.016	0.029
NOR2D1	0.058	0.024	0.041
AO22D1	0.092	0.040	0.127
XOR2D1	0.117	0.072	0.083
NOR3D1	0.073	0.032	0.052
Average	0.077 (-17%)*	0.037 (+1.8×)*	0.066

* *3D Nanofabric* area overhead/reduction, when compared to ASAP7 and IN7 respectively.

C. ALU Footprint Comparison

In this section, we first consider a basic 1-bit ALU aimed at SIMD processor applications, capable of performing the following operations: $A + B + C_{in}$, $A\&B$, $A|B$, $A\hat{}B$. Its layout using the proposed *3D Nanofabric* is shown in Fig. 6. Note the presence of several *OXWALL* regions, which fill the extra empty spaces required to route the single metal level layer. Here, there is no need for dummy-poly as in a

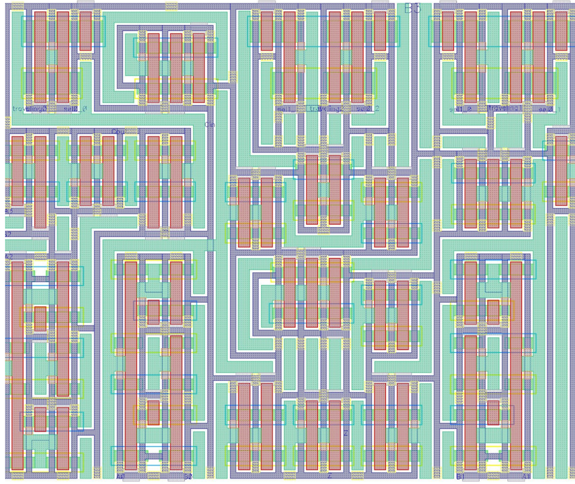

Fig. 6. 1-bit basic ALU layout view using the proposed *3D Nanofabric* rules and process flow.

FinFET technology where the gate is needed to define the Source-Drain. Instead, the empty spaces are filled with the *OXWALL* dielectric layer. Also, the gate to gate distance is always enforced (36nm) to ensure that all the gate are aligned, so the layout is fully regular. As shown in Table I, ASAP7 and IN7 have a $1.6\times$ and $3.7\times$ smaller area than the proposed *3D Nanofabric*, respectively for the 1-bit ALU as some gates have to be duplicated to avoid crossing. Besides, some extra space is required for routing where 2D processes simply use higher metal layers. However, by going to 3D and stacking several transistor tiers to build larger ALUs, we can observe considerable footprint gains. In particular, when going to 2 and 4 layers, we can already remark some footprint reduction when using the proposed *Nanofabric* flow when compared to ASAP7 (45%) and IN7 (20%), respectively. More importantly, using 32 vertical layers to build a 32-bit ALU reduces the footprint even further by a factor of $16.9\times$ and $8.7\times$ when compared to ASAP7 and IN7, respectively. We believe that stacking 32 vertical layers is a fair assumption, as current 3D NAND processes have demonstrated up to 128 stacked layers [12]. Also, a higher number of vertical layers could be considered once the technology is more mature. Note that while the results presented in this section are for the specific ALU depicted in Fig. 6 (a), similar results are expected when considering different ALU designs.

VI. Discussion and Ongoing Work

In ongoing work, we are currently assessing the delay and power benefits of the proposed *3D Nanofabric* netlist and layout approach. Delay and power improvement are expected as each vertical layer is thin (single transistor tier and routing layer). Hence the vertical connections will be short when compared to a 2D implementation. In the context of an adder or multiplier function, the carry propagation path would therefore be shorter. Besides, the 3D staircase area is also being studied. As the idea is to have an array of tiles (each tile being a 3D Nanofabric ALU) with an inter-tile communication, staircases

TABLE II
3D Nanofabric ALU footprint compared to ASAP7 and IN7 for an N-bit ALU.

Number of bits N	**Footprint (in μm^2)***		
	ASAP7	IN7	3D
1	0.787 (+1.6×)	0.338 (+3.7×)	1.257
2	1.822 (-1.4×)	0.758 (+1.7×)	1.257
3	2.186 (-1.7×)	1.193 (+1.05×)	1.257
4	2.668 (-2.1×)	1.516 (-1.2×)	1.257
8	4.765 (-3.8×)	2.991 (-2.4×)	1.257
16	11.033 (-8.8×)	5.539 (-4.4×)	1.257
24	16.169 (-12.9×)	8.265 (-6.6×)	1.257
32	21.257 (-16.9×)	10.999 (-8.7×)	1.257

* Also shows the *3D Nanofabric* footprint overhead/reduction, when compared to ASAP7 and IN7 respectively.

are only needed for the primary I/Os of the array, absorbing the area overhead of such structure.

VII. Conclusion

In this paper, we introduced a novel 3D design flow called *3D Nanofabric*. The flow consists of several identical stacked logic layers, making it well suited for SIMD processor applications where many basic regular ALUs are repeated. We thoroughly investigated the layout constraints of the *Nanofabric* flow and proposed solutions to overcome them so that basic SIMD ALUs can be designed. We showed that by using 32 vertical layers, the 32-bit ALU footprint is reduced by a factor of $8.7\times$ when compared to a traditional 2D approach using a 7nm FinFET technology. We believe that this novel 3D approach enables cost-effective 3D scaling for SIMD processors, to propose more performant circuits at a smaller footprint.

References

[1] S. Natarajan *et al.*, *A 14nm logic technology featuring 2nd-generation FinFET, air-gapped interconnects, self-aligned double patterning and a 0.0588 μm^2 SRAM cell size*, IEDM, 2014.

[2] J.P. Colinge, *FinFET and other multigate transistors*, Springer, 2007.

[3] T. T. Chua *et al.*, *3D interconnection process development and integration with low stress TSV*, ECTC, 2010.

[4] E. Beyne *et al.*, *Through-silicon via and die stacking technologies for microsystems-integration*, IEDM 2008.

[5] W. Ruythooren *et al.*, *Cu-Cu Bonding Alternative to Solder based Micro-Bumping*, EPTC 2007.

[6] P. Batude *et al.*, *Advances, Challenges and Opportunities in 3D CMOS Sequential Integration*, IEDM 2011.

[7] L. Brunet *et al.*, *First demonstration of a CMOS over CMOS 3D VLSI CoolCube™ integration on 300mm wafers*, VLSI 2016.

[8] A. Mallik *et al.*, *The impact of Sequential-3D integration on semiconductor scaling roadmap*, IEDM 2017.

[9] D. Gitlin *et al.*, *Cost model for monolithic 3D integrated circuits,* S3S, 2016.

[10] M. M. Sabry Aly *et al.*, *Energy-Efficient Abundant-Data Computing: The N3XT 1,000x,* in Computer, 48(12): 24-33, 2015.

[11] D. Kang *et al.*, *A 512Gb 3-bit/Cell 3D 6th-Generation V-NAND Flash Memory with 82MB/s Write Throughput and 1.2Gb/s Interface*, ISSCC 2019.

[12] C. Siau *et al.*, *A 512Gb 3-bit/Cell 3D Flash Memory on 128-Wordline-Layer with 132MB/s Write Performance Featuring Circuit-Under-Array Technology*, ISSCC 2019.

[13] L.T. Clark *et al.*, *ASAP7: A 7-nm FinFET Predictive Process Design Kit*, Microelectronics Journal, 53: 105-115, 2016.

[14] M. Shulaker *et al.*, *Three-dimensional integration of nanotechnologies for computing and data storage on a single chip*, Nature 547, 74–78, 2017.