An IP-Agnostic Foundational Cell Array Offering Supply Chain Security

CHRISTOPHER TALBOT, DEEPALI GARG, LAWRENCE PILEGGI, and KENNETH MAI, Carnegie Mellon University, USA

Growing IC manufacturing complexity and reliance on third-party fabrication create supply chain fragility, contributing to chip shortages and IP security risks. General-purpose ICs can mitigate manufacturing security risks but rely on software-based configurations, which is not optimal for high-consequence applications. Our work proposes a novel IP-agnostic Foundational Cell Array (FC-Array) platform to overcome these challenges. Built on verified standard cells and industry-standard EDA tools, this platform preserves many advantages of an ASIC. By incorporating 3D split manufacturing, we provide semantically secure IP protection and a base wafer that can be stockpiled. Our tests demonstrate both power-efficient (100 MHz) and high-performance (1 GHz) options. In a post-place-and-route simulated 28nm design, our FC-Array shows a worst-case 1.85x increase in power consumption and a 2.61x increase in area compared to standard cell ASICs for equivalent timing performance.

ACM Reference Format:
Christopher Talbot, Deepali Garg, Lawrence Pileggi, and Kenneth Mai. 2024. An IP-Agnostic Foundational Cell Array Offering Supply Chain Security. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3657364

1 INTRODUCTION

The current inter-dependencies in our globalized IC ecosystem create a fragility where local or global disasters can disrupt everything from hand-held consumer goods to the automobile industry [3, 19]. The recent chip shortage has highlighted the importance of supply chain resilience [11]. One popular way to reduce IC supply risk is to design software-programmed IPs to target general-purpose ICs, such as field programmable gate arrays (FPGAs). Since FPGA fabric can implement a wide variety of soft IP without hardware revisions, it is an attractive solution to be stockpiled for supply chain resilience.

However, FPGAs incur significant area & performance costs. In a recent study [2], the performance difference between an ASIC & FPGA for intensive arithmetic operations can range from 2.8x to 6.3x, with the area ratio varying from 13x to 31x. Additionally, while active, FPGAs store their IP configuration in memory, typically using BRAMs in most commercial FPGAs, raising concerns about mutability. Various factors such as aging, temperature, noise, supply fluctuation, radiations, light exposure, and physical shock could alter the device configuration. Unless additional protections are implemented, which come at the expense of area and performance, FPGAs are not suitable for high-consequence applications such as industrial controllers, automobiles, and aerospace/defense, where utmost reliability and robustness are imperative.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

A programmable fabric that embeds the IP into the hardware rather than in memory could mitigate mutability concerns and offer better performance. To this end, we can build a mask-programmable, design-agnostic IC base wafer. Mask programmable gate arrays (MGPAs), proposed in [8, 13, 14] meet these requirements but require custom cells and tooling. Design and validation of custom cells add considerable cost and development time.

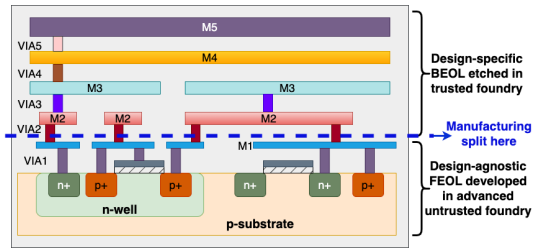

Fig. 1. Layer construction of split manufacturing with our FC Array. The fixed FEOL is manufactured in an advanced untrusted foundry. The higher-pitch upper mask layers (BEOL) can be manufactured in a trusted foundry.

This work proposes an IP-agnostic front-end-of-line (FEOL) cell array comprising only of foundry-based standard cells. We propose a platform to use these cell arrays as the underlying foundation for any digital design. We refer to this work as a **Foundational Cell Array (FC-Array)**. The proposed platform is compatible with industry-standard commercial tool flows and eliminates custom tool requirements. Although our current FC-Array only contains digital fabric, the platform could easily integrate memories and/or Hardened IPs within the fabric, through 3-D stacking, or through advanced packaging [15].

Combined with existing split manufacturing techniques [12, 18], we can produce a base wafer up to Metal 1 that can be stockpiled for later manufacture. The back-end-of-line (BEOL) can be built to order later per the design specifications. This design-agnostic FEOL additionally avoids information leakage to an untrusted foundry and discourages tampering. The split manufacturing proposed using our FC-Array is illustrated in Figure 1. The key contributions of this paper are as follows:

(1) We show a method of modifying a commercial ASIC flow compatible with our FC-Array.
(2) We present a design exploration methodology for optimal Foundational Cell Block (FCB) patterns, which forms the basis of our FC-Array design.
(3) We create and evaluate a generic and targeted pattern against various design benchmarks.
(4) We show that our design offers semantically secure IP protection with current split manufacturing methods.

Table 1 summarizes various trade-offs across existing IC manufacturing options, highlighting the scope of this work. We analyze general trade-offs a designer is sensitive to, such as design cost, the need for custom/vendor-specific tools, mutability in design, and Power, Area, and Performance (PAP). We additionally compare how

	FPGA	MPGA	Redaction	**FC-Array**	Split Fab	ASIC
Design Cost	Low	Medium	High	**Medium**	High	High
Custom Tools	No	Yes	Yes	**No**	No	No
Design Mutability	Yes	No	Yes	**No**	No	No
Power, Area & Performance	Poor	Moderate	Poor	**Moderate to Best**	Moderate to Best	Best
Obfuscation	High	High	High	**High**	Medium	None
Stockpile	Yes	Yes	No	**Yes**	No	No

Table 1. Comparison of manufacturing trade-offs in advanced semiconductor supply chain security.

well the manufacturing obfuscates the design from an untrusted foundry to protect the confidentiality and integrity of the IP.

2 FOUNDATIONAL CELL ARRAY (FC-ARRAY)

An IP-agnostic FEOL requires that the underlying physical cells, floor plan, and power plan are generic across designs and would freeze the layout in the transistor definition layers (i.e. polysilicon, diffusion, oxide). Since lower metal layers are in finer pitches, freezing these metal layers can be desirable to additionally lower manufacturing costs. The design is completed in the BEOL, potentially manufactured in a less advanced foundry, since the signals are routed in the upper metal layers with a higher pitch. The separation between the FEOL and BEOL creates an inherent boundary compatible with the existing split manufacturing techniques [12], illustrated in Figure 1.

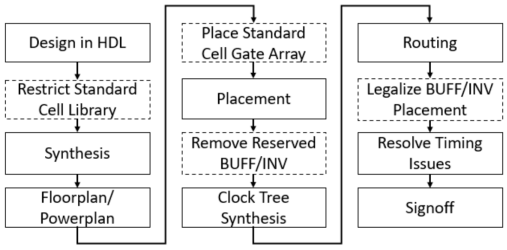

Fig. 2. The modified ASIC Design Flow to produce our FC-Array. Steps encased in a solid line are traditional flow steps, and steps encased in a dotted line are the modifications necessary.

As custom cell design and custom design tools add complexity to the design flow, our FC-Array only relies on process node standard cells and standard commercial tools. These standard cell libraries are released with a process node, are verified by the foundry, and are compatible with widely used commercial tools. These allow us to abstract the underlying process node from the design and target different process nodes more efficiently.

Figure 2 shows how to modify an industrial digital design flow to incorporate our FC-Array. The modifications required are in the synthesis, placement, and routing stages, described in detail below.

2.1 Foundation Cell Block design

The basis of the FC-Array is a regular pattern of standard cells, the layout of which forms our Foundation Cell Block (FCB). This pattern allows us to flexibly scale the floorplan based on manufacturing or design needs. In a commercial setting, the floorplan size is fixed, but a foundry could offer different floorplan sizes of an FC-Array depending on a consumer's demand.

2.2 Synthesis design flow

The FCB requires a restricted set of standard cells that are universal enough to realize any possible HDL design. To complete logic synthesis, commercial tools require, at the very least, an inverter, a 2-input gate with complete functionality (e.g. NAND2 or NOR2), and a flip flop. In addition, TIE-Hi/TIE-Lo cells are required for tools to synthesize constants, and eventually to tie off unused cells to avoid unnecessary power consumption. Many designs additionally require a flip-flop with set-and-clear capabilities. While not strictly necessary for functional synthesis, buffers and clock-NANDs are essential for the synthesis and the placement and routing (PnR) tools to address issues related to design setup and hold time violations.

2.3 Place and Route (PnR) flow

As the traditional ASIC design flow normally assumes a constraint-free environment, we must modify the flow for awareness of our FCB. Many design flows feature an Engineering Change Order (ECO) flow to support post-chip designs, which allows us to place our FCB and modify the flow for FCB-aware PnR.

2.3.1 Placement flow. In this step, we instruct the tool to place our FCB and map the synthesized design to the FCBs. Two considerations are necessary at this step: a) the optimal mix of standard cells within the FCB and b) the physical placement of the cells. Once the FCB is placed, we instruct the EDA tool to fit the synthesized design into the FCBs. We then tie off all unused FCB cells to logic 1 or 0.

The tool cannot complete clock tree synthesis (CTS) or routing if the floor plan is full after placement. EDA tools at these steps usually attempt to place certain buffers and inverters for timing specifications but must be made aware of pre-placed buffers/inverters from FC-Array. We must, therefore, leave reserved spaces for these buffers and inverters for the tool to complete the following steps.

2.3.2 Routing Flow. ASIC tools will place buffers and inverters in CTS and routing without regard to regularity. Therefore, to re-establish the regularity required for our intended FEOL, we must go back and "legalize" buffer and inverter placement after the routing stage. Since they will be in reserved locations, it is ensured that the cell movements are minimal not to disrupt the timing solution EDA tools have optimized for thus far.

2.4 Split manufacturing support

Previous work [12] has shown that it is possible to split fabrication with a 28nm process, with the FEOL produced in an advanced foundry and the BEOL produced in another foundry. We illustrate the split manufacturing proposed for our FC-Array construction in Figure 1. In typical standard cells, intra-cell connections were

at Metal 1 and 2, and as such, these metal layers are candidates to be the topmost layer of our FEOL. Metal 1 is filled with intra-cell connections, allowing very few inter-cell routing connections. Metal 2 is heavily used in routing, so freezing this layer significantly degraded routing. As such, we chose Metal 2 and above to form the BEOL for our work. This choice is compatible with existing split manufacturing shown in previous work [12].

3 EVALUATION METHODOLOGY

While it is possible to optimize FC-Arrays towards specific design domains, each FC-Array would require manufacturing and stockpiling. As such, we desire to explore a more generalized FC-Array that offers a balanced PAP across several design domains. However, this platform can produce specialized FC-Arrays to suit specific application domains with ease. The generalized FC-Arrays proposed in this work have been tested across a range of circuit types, including cryptography-based, arithmetic-based, and processor-based designs, ranging from 7,000 to 70,000 gates.

Our process begins with an extensive of the FCB design space, which forms the basis of the FC-Array. We aim to achieve a balanced PAP by mirroring standard cell compositions and patterns (Section 4). Optimal FCB candidates are identified and placed across the floorplan, allowing seamless mapping of any design (Section 2).

In Section 5, we compare FC-Array designs to their standard cell ASIC counterparts for area and power at performance efficiency. Area overheads stem from core PnR floorplan design, while power overheads are simulated post-PnR under normal operation. We use an industrial 28nm process with only standard industry tools.

4 FCB DESIGN

FCB is the regular core pattern of our FC-Array and is built entirely of standard cells. Through years of FPGA design and tool development, it has become evident that crafting a general-purpose, efficient fabric lacks a straightforward closed-form solution, necessitating thorough empirical analysis [10]. In contrast, the FCB is not bounded by a static routing structure and does not have a homogeneous LUT structure, further complicating the solution space. Our FCB design tackles this complexity through empirical and experimental insights, ensuring real-world effectiveness. Creating an FCB involves navigating three design spaces: a) types of cells, b) proportions of cells, and c) physical cell placement within the FCB.

4.1 FCB cell type compositions

Modern standard cell libraries consist of thousands of cell types per threshold voltage. To achieve regular patterns for IP-agnostic FEOL stockpiling, our first task is to determine an optimal subset of cells. Previous research on restricted libraries has shown that a smaller set can sometimes yield optimal results [5].

We synthesized all designs under test in ASIC to observe the most common cell types to be used as a basis for the FCB. As observed from these ASIC synthesis results and as pointed out in previous work related to restricted cell libraries, [4, 6, 17] basic logic cells such as NAND, NOR, AOI, OAI, INV were the most commonly used cells. We also saw that XOR, XNOR, MUX, and Full Adder were commonly used depending on the type of design. We thus choose these as viable candidates for cell types.

4.1.1 Logic Cell Combinations. To evaluate the impact of candidate cells from a restricted library on synthesized cell count compared to ASIC designs, we incrementally introduced cells and tested their effect on area and power. All FCB designs additionally included an inverter, buffer, flip-flop with set and clear functionality, clock NAND2, and TIE-Hi/TIE-Lo cells, and we minimized cell sizes for optimal power and area efficiency. Starting with NAND2 resulted in the highest gate count overhead (average 3.64x), which lowered to 3.16x with the inclusion of NOR2. Incorporating AOI cells, and their complementary OAI22 cells, further reduced the average overhead to 2.27x. Notably, when a cell was included, its complement cell led to a reduction in gate count.

Some previous work on restricted libraries [6] featured more complex cells, notably XOR and a Full Adder. Some designs benefited more from complex cell types like XOR and a Full Adder. For instance, arithmetic circuits benefited from XOR2/XNOR2 and even more from a Full Adder (FA) cell. Other circuits, such as AES, DES3, and SHA3, didn't use any FA cells when provided. Including these larger cells in a general-purpose FCB could result in wasted area for some applications, suggesting their suitability for application domain-specific FC-Arrays.

4.1.2 Performance Considerations. To evaluate the PAP overheads of our restricted cell libraries, we conducted benchmark placements and routings covering various application requirements. All benchmark circuits achieved a maximum frequency of 1 GHz with standard cell Place-and-Route. We systematically reduced the clock frequency from 1 GHz, observing consistent timing slack until approximately 130 MHz when slack increased, indicating that timing specifications were already met. This pattern persisted across all designs, leading us to examine frequencies from 100 MHz to 1 GHz in our experiments.

When working with restricted library sets, we found that designs with minimal cell size restrictions performed well up to 500 MHz in the 28 nm process node we targeted. To handle higher clock frequencies, we introduced higher drive-strength buffers along with low-strength buffers to allow the tools to reach an optimal timing solution satisfying both setup and hold constraints. With our analysis, we found that we could keep the logic cells minimally sized and could introduce higher strength buffers for frequencies above 500 MHz. This enabled the restricted libraries to match the performance of our designs with their standard cell ASIC counterparts. In the 28nm node, size-1, size-4, and size-12 buffers were the minimal combination that optimally fulfilled the timing requirements.

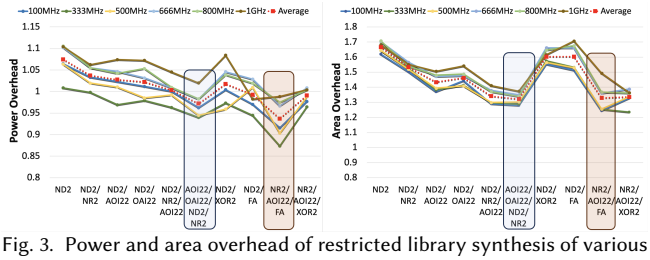

Fig. 3. Power and area overhead of restricted library synthesis of various cell compositions, across a range of operational frequency requirements from 100MHz to 1GHz. The two most suitable candidates are highlighted - in blue(*Vanilla-FCB*) and brown(*Crafted-FCB*).

4.1.3 Flavors. Following logic gate count overheads, we then evaluated the power and area overheads for various cell compositions with minimally sized buffers up to 500 MHz and a combination of size-2, size-4, and size-12 buffers for frequencies above 500 MHz, as shown in Figure 3.

We find that NAND/AOI/FA offers minimal overheads, and constitutes our ***Crafted-FCB***. However, as highlighted in 4.1.2, the use of complex cells like FA can be tricky for a general-purpose FC-Array design. The FA only benefits certain designs and contributes on average <5% of the total cell composition across our benchmarks.

As a result, we keep the second most optimal candidate under consideration to compare the overheads of the two FCBs after place-and-route. The NAND/NOR/AOI/OAI composition worked well across all designs and frequencies and had overheads just above the NAND/AOI/FA composition, this forms our ***Vanilla-FCB***.

4.2 Relative proportions of cells

With the cell types chosen, we must now determine the ratio of each gate to include in our FCB. We analyze the synthesized designs and find the ratio of the gates used in each design. As a representative, these ratios are shown for low-power Vanilla-FCB composition synthesized at 100 MHz in Figure 4. Our low-power Vanilla-FCB is designed to maintain cell proportions which approximately meets the mean of each cell-type ratio in these bars.

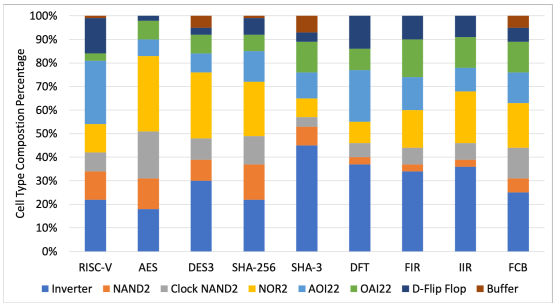

Fig. 4. Relative proportions of cells that compose the Vanilla Foundation Cell Block (FCB), optimized for a lower power consumption targeting a 100 MHz clock.

As described in Section 2.3.2, Clock Tree Synthesis (CTS) and routing stages introduce buffers and inverters, which are not accounted for by the synthesis stage. We thus increase the ratio of buffers to account for these additional cell placements in the later stages of the PnR tool. However, it is important to note that the ratio of buffers is not representative of how many were actually used, but representative of the required reserved space required by the PnR tool to produce satisfactory timing closure on our designs.

The higher performance vanilla and crafted FCBs proportions are determined in a very similar way. With the higher-performance FCBs, we observed through place-and-route designs with the restricted library that about an equal proportion of BUFF1, BUFF4, and BUFF12 were required.

4.3 Physical placement of cells within the FCB

In our initial attempts, we aimed to design the FCB placement in a style similar to FPGAs, where we paired a small amount of functionally complete logic (NAND2, buffer, inverter) with a D Flip Flop (DFF), as shown in Figure 5-*Version 1*. However, area constraints approached 4-5x due to routing congestion, and power overhead dominated by switching power over long routes. We experimented with reducing FCB cell density, which increased area overhead with minimal power savings. Given routing congestion issues even with this basic cell composition, we didn't pursue more complex cell types with such a placement pattern.

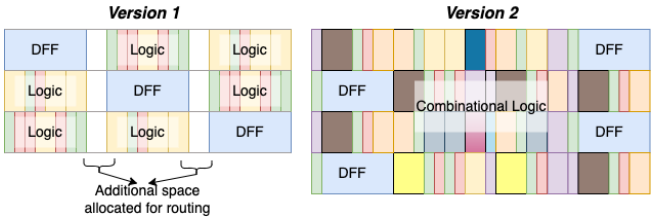

Fig. 5. Initial attempts at FCB design. *Version 1* shows FCB using the minimal cell composition. *Version 2* shows FCB with interleaved flip-flop placement, which proved sub-optimal for timing.

We then attempted a different placement pattern to align with the desired cell composition and proportions outlined in Sections 4.1 and 4.2. Since clocked designs switch between sequential and combinational logic, we tried clustering combinational logic between vertical rows of DFFs, assuming this would better organize the logic to ease routing congestion. This design is illustrated in Figure 5-*Version 2*. However, we found that the tool placed flip-flops sparsely without considering the combinational logic, making it challenging to meet timing specifications. we observed that, as expected, the placement of DFFs relative to the combinational logic significantly affected the FCB design's ability to meet setup and hold constraints.

In summary, we noted that when the tools couldn't find an ideal placement solution, our area limitations were mainly influenced by routing congestion rather than a lack of necessary gates. Our findings revealed that mirroring the placement patterns of a standard cell design provides the most substantial benefits in power, area, and performance. This strategy is effective because it fosters the inherent collaboration between the routing and placement tools to achieve an optimal solution.

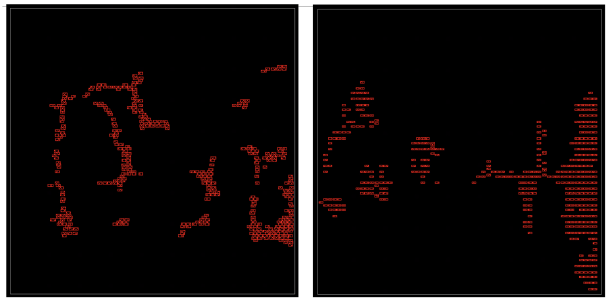

Fig. 6. An AES design in ASIC versus FC-Array that highlights the mimicking of flip-flop clumping patterns, which the relative placement of flops within our FCB enables.

4.3.1 Final Version. We examined how flip-flops are placed in standard cell designs to reduce routing congestion and enhance timing, as seen in Figure 6. Typically, flip-flops naturally form clusters during placement. To replicate this behavior, we designed our FCB with

vertical and horizontal stacks of flip-flops. Given that flip-flops are relatively large, this arrangement promotes efficient clustering and facilitates the interleaving of combinational logic between them.

We also interspersed combinational logic to maintain diversity in neighboring cell types and evenly distributed reserved buffer space to help tools meet setup/hold constraints. To prevent routing congestion near the DFFs, we placed inverters on either side of them and included TIE cells in close proximity. Clock NAND gates were positioned closer to the flip-flops, and well-taps were integrated in compliance with DRC rules without disrupting FCB regularity. The current FCB measures $36\mu m \times 12\mu m$, serving as the basic unit of our FC-Array floorplan granularity. This completes our FCB design, which is replicated all across the fabric to construct the FC-Array as shown in Figure 7.

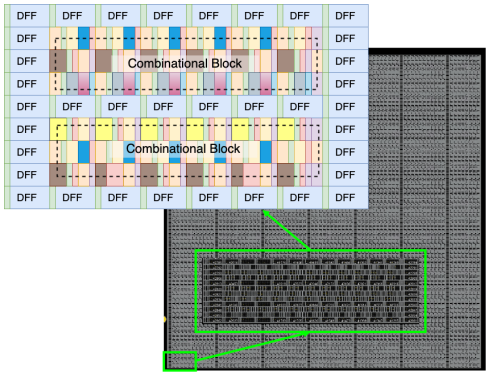

Fig. 7. A representation of our FC-Array constructed out of designed FCB.

5 EVALUATION

Using the FCB design, we create FC-Array designs following the platform in Section 2 and evaluate them per Section 3. Focusing on optimal candidates, namely *Vanilla FCB* and *Crafted FCB* from Section 4.1.3, we assess their low-power and high-performance configurations as detailed in Section 4.1.2. The high-performance evaluation involves selecting the highest frequency (1 GHz), while the low-power FC-Array evaluation uses a frequency of 100 MHz. We then compare the area and power penalties of various FC-Array designs against their standard cell ASIC counterparts at these frequency points.

5.1 Low-power FC-Array

	Crafted FCB		**Vanilla FCB**	
Penalty (Ratio)	Power	Area	Power	Area
RISC-V	1.15	1.78	1.10	2.01
AES	1.31	3.70	1.31	2.74
DES3	1.09	2.89	1.12	2.25
SHA-256	1.29	1.40	1.86	1.62
SHA-3	1.72	2.17	1.42	1.36
DFT	1.11	1.92	1.17	2.47
FIR	1.04	2.47	1.31	1.65
IIR	1.07	2.25	1.45	1.41
Mean	**1.22**	**2.32**	**1.24**	**1.88**
Std. Deviation	0.22	0.72	0.38	0.51

Table 2. Area and power penalty ratios of the FC-Array at 100 MHz (Contains only BUFF1). Vanilla-FCB uses NAND/NOR/AOI/OAI, and Crafted FCB uses NAND/AOI/FA logic cells.

Table 2 shows area and power penalties for our designs compared to standard cell counterparts at 100 MHz. In low-power applications, our IP-agnostic FEOL increases power by *24%* and area by *88%* on average versus the ASIC. FC-Array power penalties, using vanilla and crafted FCBs, are generally similar. The highest vanilla FCB power penalty (1.86x for SHA-256) is significantly reduced by the crafted FCB to 1.29. However, the crafted FCB negatively impacts non-arithmetic designs like SHA-3, resulting in a higher power penalty (1.72x vs. 1.42x in the vanilla case).

Regarding area, the crafted FCB is generally more expensive than the vanilla FCB. With the vanilla FCB, AES incurs the highest penalty of 2.74x, while FIR has the lowest penalty of 1.31x. For the crafted FCB, area penalties are more pronounced in designs like AES, DES3, and SHA-3 (which don't use the FA) and reduced in SHA-256 and DFT compared to the vanilla FCB. The crafted FCB trades off lower power for more area in designs such as RISCV, IIR, and FIR.

5.2 High-performance FC-Array

Table 3 displays power and area overheads for all designs at a 1 GHz frequency specification, considering both vanilla and crafted FCB compositions. It's important to highlight that each design meets specified timing constraints in both configurations. In high-performance scenarios, we observe, on average, higher area overheads compared to lower power situations. This is attributed to the need for a stronger drive strength buffer and increased buffering to meet timing requirements, particularly noticeable in IIR, FIR, and SHA-3 designs, which need additional buffers for timing closure.

	Crafted FCB		**Vanilla FCB**	
Penalty (Ratio)	Power	Area	Power	Area
RISC-V	1.20	1.96	1.37	1.96
AES	2.22	2.25	1.63	2.64
DES3	1.20	1.65	1.27	2.25
SHA-256	1.79	1.96	2.02	2.05
SHA-3	2.69	3.78	2.46	3.56
DFT	2.25	1.91	2.22	1.27
FIR	2.09	3.16	1.89	3.16
IIR	2.16	3.56	1.96	3.56
Mean	**1.95**	**2.53**	**1.85**	**2.61**
Std. Deviation	0.52	0.84	0.41	0.88

Table 3. Area and power penalty ratios of the FC-Array at 1GHz(Contains BUFF1/BUFF4/BUFF12). Vanilla-FCB uses NAND/NOR/AOI/OAI, and Crafted FCB uses NAND/AOI/FA logic cells.

Comparing overheads between Crafted-FCB and Vanilla-FCB, differences are relatively lower than in low-power counterparts. Buffers serve as the bottleneck for timing solutions in PnR, with overheads only mildly influenced by the choice of logic cells. Notably, there are slight advantages in using crafted FCB in RISC-V and SHA-256 designs. However, in most other designs, the introduction of more complex cells, such as FA, exacerbates overheads.

5.3 IP protection with FC-Array

As outlined in Section 1, the FC-Array aims to safeguard IP through split manufacturing, illustrated in Figure 1. In Section 2.4, we freeze the front-end of line (FEOL) layers, enabling an untrusted foundry to create the base wafer without revealing the core design. The sensitive parts, situated in the back-end of line (BEOL) routing

structure, will be manufactured by a trusted foundry. Unlike previous split manufacturing attempts vulnerable to security breaches, we enhance security by evaluating the FC-Array-based split manufacturing under the Simulation Security Definition [1, 7]. This definition is designed to offer superior security compared to empirical attack evaluations, addressing potential leaks from heuristics exploited by Place & Route tools [16].

Proof. The adversary $\mathcal{A}^{Oq,c_o}(1^\lambda, c_e, L(c_o))$ differs from Simulator $\mathcal{S}^{Oq,c_o,\mathcal{A}}(1^\lambda, L(c_o))$ by having access to the locked netlist c_e. In our FC-Array split manufacturing, c_e extends to the top FEOL metal layer (Metal 1). Metal 1 contains all intra-cell connections for combinational cell gates and some intra-cell connections of flip-flops; inter-cell connections in Metal 1 are prohibited as outlined in Section 2.4. Thus, c_e includes the FCB composition, floorplan size, and potential input/output pin count. The only information leaked is limited to the maximum number of gates and input/output pins, aligning with the leakage function $L(c_o)$ of the locking mechanism and the leakage function of the universal circuit [7]. Thereby, here access to c_e is equivalent to the adversary A having access solely to the leakage function $L(c_o)$. In this scenario, $\mathcal{A}$ and $\mathcal{S}$ are identical, establishing simulation security for FC-Array construction under split manufacturing. Note that the proof is agnostic of the FCB design and holds true for any FC-Array design.

5.4 Comparison with eFPGA

Compared to the closest alternative, FPGAs/eFPGAs, our proposed FC-Array serves both stockpiling and IP protection purposes. We conduct a meaningful comparison by evaluating our low-power Vanilla FC-Array against an eFPGA [9], specifically for AES, Mul18, and Sobel designs. Aligning with the reference eFPGA clocked at 200 MHz, we set our FC-Array's low-power design clock to 200 MHz to match performance. The eFPGA was fabricated using a 16nm FinFET process, comprising 256 CLBs with eight 6-input LUTs each, totaling an area of $1.74mm^2$ [9]. For a fair assessment, we constrain the FC-Array floorplan to barely contain the AES design, resulting in a total floorplan area of approximately $0.06mm^2$. Despite the eFPGA being in a more advanced process node, the FC-Array consumes only *5.6%* of the eFPGA's power for identical designs. In the best-case scenario, the soft eFPGA occupies **29x** more area and consumes **17.8x** more power than our FC-Array fabric.

6 CONCLUSION

Our work introduces a versatile IP-agnostic FEOL FC-Array compatible with existing processes and ASIC workflows. We establish a flexible FC-Array design platform, demonstrating its performance through two example designs with modest power and area overheads and no degradation in performance. The crafted FCB examples reveal the benefits of complex gates in specific applications. Machine learning optimizes composition and placement based on prevalent IC patterns. The foundry can offer tailored FC-Array variations, including different processing versions (e.g. HVT, LVT). While our current FC-Array focuses on digital logic designs, integration of memories, hardened IPs, and analog/RF components is possible through direct integration, 3-D stacking, and/or advanced packaging like chiplets [15]. Our work marks the first step towards seamlessly integrating FC-Array into commercial workflows, providing a flexible solution for diverse IC design requirements. This innovation offers supply chain resilience by maintaining a fixed-hardware stockpile, addressing mutability and efficiency issues often seen in Field-Programmable Gate Arrays (FPGAs). This FC-Array additionally safeguards IP for critical military and aerospace applications through 3D split manufacturing.

REFERENCES

[1] Peter Beerel, Marios Georgiou, Ben Hamlin, Alex J Malozemoff, and Pierluigi Nuzzo. 2022. Towards a formal treatment of logic locking. *Cryptology ePrint Archive* (2022).

[2] Andrew Boutros, Sadegh Yazdanshenas, and Vaughn Betz. 2018. You Cannot Improve What You Do Not Measure: FPGA vs. ASIC Efficiency Gaps for Convolutional Neural Network Inference. *ACM Trans. Reconfigurable Technol. Syst.* 11, 3, Article 20 (dec 2018), 23 pages. https://doi.org/10.1145/3242898

[3] Tom Coughlin. 2020. Impact of COVID-19 on the consumer electronics market. *IEEE Consumer Electronics Magazine* 10, 1 (2020), 58–59.

[4] Man-Ho Ho, Yan-Qing Ai, Thomas Chun-Pong Chau, Steve C. L. Yuen, Chiu-Sing Choy, Philip H. W. Leong, and Kong-Pang Pun. 2013. Architecture and Design Flow for a Highly Efficient Structured ASIC. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 21, 3 (2013), 424–433. https://doi.org/10.1109/TVLSI.2012.2190478

[5] Mehmet Meric Isgenc, Mayler GA Martins, V Mohammed Zackriya, Samuel N Pagliarini, and Larry Pileggi. 2019. Logic IP for Low-Cost IC Design in Advanced CMOS Nodes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 2 (2019), 585–595.

[6] Mehmet Meric Isgenc, Mayler G. A. Martins, V. Mohammed Zackriya, Samuel N. Pagliarini, and Larry Pileggi. 2020. Logic IP for Low-Cost IC Design in Advanced CMOS Nodes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 2 (2020), 585–595. https://doi.org/10.1109/TVLSI.2019.2942825

[7] Elisaweta Masserova, Deepali Garg, Ken Mai, Lawrence Pileggi, Vipul Goyal, and Bryan Parno. 2022. Logic Locking-Connecting Theory and Practice. *Cryptology ePrint Archive* (2022).

[8] Petra Michel and Martin Geiger. 1989. Basic cell for a gate array arrangement in CMOS Technology. US Patent 4,884,115.

[9] Prashanth Mohan, Oguz Atli, Onur Kibar, Mohammed Zackriya, Larry Pileggi, and Ken Mai. 2021. Top-down physical design of soft embedded fpga fabrics. In *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 1–10.

[10] Kevin E. Murray, Oleg Petelin, Sheng Zhong, Jai Min Wang, Mohamed ElDafrawy, Jean-Philippe Legault, Eugene Sha, Aaron G. Graham, Jean Wu, Matthew J. P. Walker, Hanqing Zeng, Panagiotis Patros, Jason Luu, Kenneth B. Kent, and Vaughn Betz. 2020. VTR 8: High Performance CAD and Customizable FPGA Architecture Modelling. *ACM Trans. Reconfigurable Technol. Syst.* (2020).

[11] NASDAQ. [n. d.]. Putting the Chip Shortage into the Context of Long-Term Trends. https://www.nasdaq.com/articles/putting-the-chip-shortage-into-the-context-of-long-term-trends. ([n. d.]).

[12] Office of the director of National Intelligence. [n. d.]. IARPA Trusted Integrated Chips (TIC) Program. https://www.ndia.org/-/media/sites/ndia/meetings-and-events/divisions/systems-engineering/past-events/trusted-micro/2016-august/mccants-carl.ashx. *IARPA* ([n. d.]).

[13] Larry Pileggi, Herman Schmit, Andrzej J Strojwas, Padmini Gopalakrishnan, Veerbhan Kheterpal, Aneesh Koorapaty, Chetan Patel, Vyacheslav Rovner, and K Yaw Tong. 2003. Exploring regular fabrics to optimize the performance-cost trade-off. In *Proceedings of the 40th annual Design Automation Conference*. 782–787.

[14] Brian D Possley. 2005. Gate array architecture. US Patent 6,974,978.

[15] Gang Qu, Serge Leef, Chip-Hong Chang, David Kehlet, and Murthi Sadhasivan. 2023. CAD for Assurance: Hardware Security 2.0: What Are The New Frontiers? https://ieee-ceda.org/presentation/webinar/cad-assurance-hardware-security-20-what-are-new-frontiers.

[16] Jeyavijayan Rajendran, Ozgur Sinanoglu, and Ramesh Karri. 2013. Is split manufacturing secure?. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1259–1264. https://doi.org/10.7873/DATE.2013.261

[17] Hui-Hsiang Tung, Rung-Bin Lin, Mei-Chen Li, and Tsung-Han Heish. 2012. Standard Cell Like Via-Configurable Logic Blocks for Structured ASIC in an Industrial Design Flow. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 20, 12 (2012), 2184–2197. https://doi.org/10.1109/TVLSI.2011.2170712

[18] Kaushik Vaidyanathan, Bishnu P Das, Ekin Sumbul, Renzhi Liu, and Larry Pileggi. 2014. Building trusted ICs using split fabrication. In *2014 IEEE international symposium on hardware-oriented security and trust (HOST)*. IEEE, 1–6.

[19] Xiling Wu, Caihua Zhang, and Wei Du. 2021. An analysis on the crisis of "chips shortage" in automobile industry——Based on the double influence of COVID-19 and trade Friction. In *Journal of Physics: Conference Series*, Vol. 1971. IOP Publishing, 012100.