M2A2: Microscale Modular Assembled ASICs for High-Mix, Low-Volume, Heterogeneously Integrated Designs

Aseem Sayal, *Student Member, IEEE*, Paras Ajay, Mark W. McDermott, *Life Member, IEEE*, S. V. Sreenivasan, *Member, IEEE*, and Jaydeep P. Kulkarni, *Senior Member, IEEE*

***Abstract*—With CMOS process technology scaling, the mask cost for fabricating nano-scale transistors, contacts, and interconnects has become prohibitively expensive, especially, for low volume designs. Moreover, higher transistor density has resulted in higher design complexity and large-sized die, which has led to an increase in the design cycle time and degradation in the process yield. These challenges are forcing low-volume application-specific integrated circuits (ASICs) toward highly suboptimal field-programmable gate arrays (FPGAs). In this article, we propose a new approach for designing and fabricating high-mix, low-volume heterogeneously integrated ASICs, referred to as Microscale Modular Assembled ASIC (M2A2), consisting of: 1) pick-and-place assembly of prefabricated blocks (PFBs) which utilizes the nano-precision placement capabilities developed in jet-and-flash imprint lithography (J-FIL) and 2) EDA design methodology utilizing unsupervised learning and graph-matching techniques. The EDA methodology leverages existing CAD tool infrastructure for easy adoption into the current EDA ecosystem. The proposed fabrication technology makes use of pick-and-place assembly technique to allow nano-precise assembly of PFBs. The PFBs can be fabricated in advanced process nodes and then knitted together on a wafer substrate. Custom-designed low-cost back-end metal layers can then be created/placed on top of the PFB knitted layer to realize a variety of high-mix, low-volume ASIC designs. M2A2 would allow more flexibility in front-end design by optimal PFB selection and knitting compared to the earlier proposed approaches such as structured ASICs (sASICs). In this article, the performance of M2A2-based designs are compared with different design technologies, such as baseline ASICs, FPGAs, and sASICs at 16 nm, 40 nm, and 130 nm CMOS process nodes. The post-PNR simulation results achieved over 15 IWLS benchmarks show that the proposed M2A2 designs achieve 27.11 × −34.89× reduced power-delay-product (PDP) compared to FPGAs, and incur 1.69 × −2.36× larger area compared to the baseline ASICs. The M2A2 designs achieve 15%–68.5% smaller area and 8.5%–52% higher performance compared to the sASIC methodologies. Moreover, the key fabrication steps in the proposed M2A2 technology are presented. The experimental fab results along with the proposed EDA flow simulations show promising results for the proposed M2A2 technology. Design tradeoffs and process challenges for large scale deployment of the M2A2 technology are discussed along with their mitigation strategies.**

***Index Terms*—Application-specific integrated circuit (ASIC), field-programmable gate array (FPGA), jet-and-flash imprint lithography (J-FIL), microscale modular assembled ASIC (M2A2), prefabricated block (PFB).**

I. Introduction

THE AGGRESSIVE scaling of critical dimensions in scaled CMOS technology has resulted in an enormous increase in the cost of the mask-sets for advanced application-specific integrated circuits (ASICs). Furthermore, higher transistor density has resulted in increased design complexity, which has led to an increase in the verification effort, more design respins, increased cycle times and eventually a longer time-to-market (TTM). The combined effects of these factors have made advanced ASICs [Fig. 1(a)] prohibitively expensive for low/mid volume applications [1].

Multiple solutions have been proposed earlier for cost effective low-volume designs. For example: 1) field-programmable gate array (FPGA) [Fig. 1(b)] provides cost feasible solution due to the configurable logic blocks (CLBs) to realize a variety of functions. However, due to the high number of redundant/unused CLBs and input–output blocks (IOBs), and programmable interconnects with long wirelength, the FPGA-based approach incurs higher area, higher power, and lower performance when compared to the baseline ASICs. These design overheads are particularly challenging to overcome in power constrained integrated circuits [2] and 2) Earlier proposed structured ASIC (sASIC) [Fig. 1(c)] methodology addresses the cost *versus* performance tradeoff in FPGA by giving more flexibility in look-up table (LUT) design and reducing the routing overheads [2]. However, sASIC approach still results in placement and routing congestion, and necessitates significant change in the existing commercial CAD tools [3]. This complicates the design implementation process and consequently its widespread adoption.

To achieve a cost-effective design, along with design flexibility, low redundancy, shorter wire-lengths, and low routing

Manuscript received September 22, 2019; revised January 15, 2020; accepted March 14, 2020. Date of publication March 23, 2020; date of current version November 20, 2020. This work was supported by the Cockrell School of Engineering and by the NASCENT Center's Nanoengineering Online Education Program at the University of Texas at Austin. This article was recommended by Associate Editor R. Drechsler. *(Corresponding author: Aseem Sayal.)*

Aseem Sayal, Mark W. McDermott, and Jaydeep P. Kulkarni are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: aseem.sayal@utexas.edu; mcdermot@ece.utexas.edu; jaydeep@austin.utexas.edu).

Paras Ajay is with the Department of Mechanical Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: paras.ajay@utexas.edu).

S. V. Sreenivasan is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA, and also with the Department of Mechanical Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: svs@austin.utexas.edu).

Digital Object Identifier 10.1109/TCAD.2020.2982621

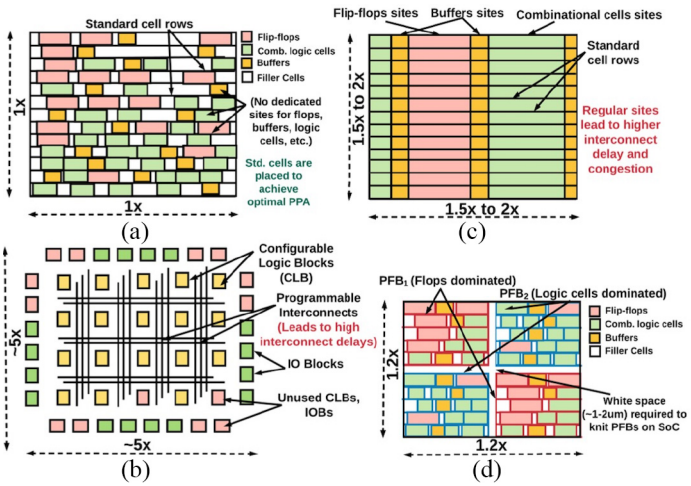

Fig. 1. Different design technologies. (a) ASIC. (b) FPGA. (c) sASIC. (d) M2A2 (proposed in this article).

congestion, we propose a microscale modular assembled ASIC (M2A2) technology. The proposed M2A2 technology could enable the sharing of the critical mask-set cost across many ASICs using prefabricated blocks (PFBs). In the proposed M2A2 approach, an SoC is fabricated by physically assembling PFBs, which contain front-end critical mask layers, i.e., transistors, and front-end interconnects, onto a product wafer, and subsequently connecting the PFBs using custom back-end metal layers (for which the mask cost is low). Fig. 1(d) shows the floorplan of M2A2 design with two types of PFBs knitted on the SoC. This approach would enable amortization of the mask costs, as well as design effort across a large number of designs, while approaching baseline ASIC power-performance-area (PPA) metric. The fabrication of the proposed modular ASICs could be realized by assembling PFBs on a wafer substrate using either existing pick-and-place approaches [4]–[6], or using the preferred approach of Jet-and-Flash-Imprint-Lithography (J-FIL)-based pick-and-place assembly. The preferred assembly technique, which is described in more detail in Section III, leverages sub-5nm precision large-area placement capabilities developed in jet-and-flash imprint lithography (J-FIL) [7], [8].

The main contributions of this article are as follows.

1) The concept of designing a variety of low-volume ASICs using PFBs is proposed.
2) An overview on M2A2 fabrication technology and pick-and-place assembly is presented.
3) An entire M2A2 EDA methodology comprising of front-end and back-end design solutions is presented.
4) The proposed CAD solutions based on unsupervised learning and graph matching techniques are discussed in detail for optimal PFB design, knitting of PFBs on SoC and post-Mask clock tree synthesis (CTS).
5) The M2A2 synthesis, preCTS and routing methodology based on commercial EDA tools is discussed.
6) The experimental results of some of the key steps involved in the M2A2 fabrication and process technology are presented.
7) The detailed performance comparison (post-PNR simulation) of M2A2-based designs over set of various international workshop on logic synthesis (IWLS) benchmarks, and over different CMOS process nodes (130 nm, 40 nm, and 16 nm) is performed with earlier proposed sASICs, FPGAs and baseline ASICs design configurations.

TABLE I
QUALITATIVE COMPARISON WITH OTHER DESIGN TECHNOLOGIES

Metric	FPGA	sASIC	Baseline ASIC	M2A2 (This work)
NRE	Lowest	Low	Highest	Low
TTM	Lowest	High	Highest	Low
PPA	Highest	High	Lowest	Low
Re-configurability to implement a given design	Yes	Limited	No	Yes (same functional category)
Heterogeneous Integration	No	No	No	Yes
Commercial CAD flows	Yes	No	Yes	Yes
Ability to fabricate secured chips at cutting edge nodes	No	No	No	Yes

The key benefits of the proposed M2A2 technology (summarized qualitatively in Table I) are as follows.

1) Enables sharing of the mask-set cost across many ASIC designs, thus reducing the NRE costs for individual designs.
2) Greater flexibility in the front-end design compared to sASICs and metal configurable ASICs, thereby improving PPA.
3) Leverages existing commercial EDA tools to perform tool-based design optimizations with reduced number of ECOs (engineering change order), thus potentially reducing TTM in comparison to ASICs and sASICs.
4) Enables design of domain-specific SoCs using the limited number of PFBs.
5) Enables heterogeneous integration through the assembly of PFBs manufactured using different materials and/or different technology nodes and/or memory technologies.
6) Enables manufacturing of secure ASICs for low-volume security-critical applications. This could be achieved by manufacturing generic PFBs at a commercial foundry using advanced CMOS nodes, manufacturing custom metal die (CMD) comprising of higher metal layers (relaxed pitch; low NRE cost) at a trusted foundry, and then knitting PFBs and CMD using a pick-and-place fabrication technique at a trusted foundry (Fig. 2).

This article is organized as follows. In Section II, we present the concept of a PFB-based SoC. Section III presents the overview of the proposed M2A2 fabrication technology. Section IV describes the proposed M2A2 EDA methodology. Various M2A2 design tradeoffs and guidelines are discussed in Section V. The experimental results of the M2A2 fabrication and process technology are presented in Section VI. In Section VII, M2A2 EDA post-PNR simulation results comparing PPA of M2A2-based designs with baseline ASICs, sASICs, and FPGAs are presented. Various design and process challenges along with their mitigation strategies are discussed in Section VIII. The conclusions are presented in Section IX.

II. M2A2 BIG PICTURE

A PFB is a circuit element which is 10–100 s of microns in lateral dimension, and consists of transistors and front-end interconnects. In the default configuration, it comprises of a base layer made up of transistors and front-end metal

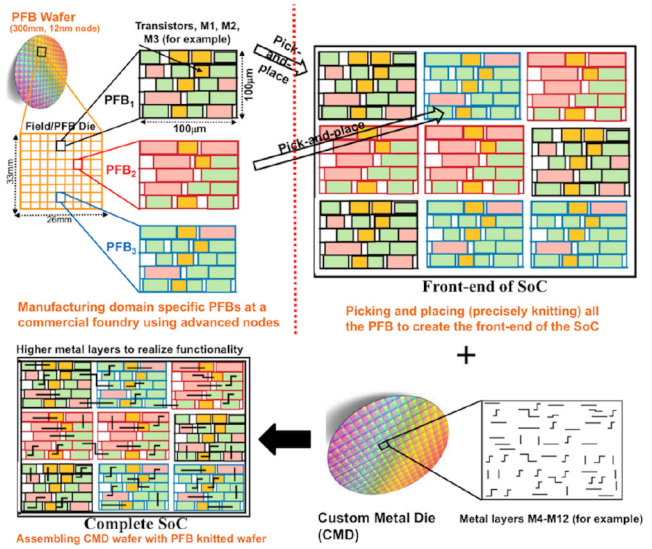

Fig. 2. Fabrication methodology of PFB-based SoC.

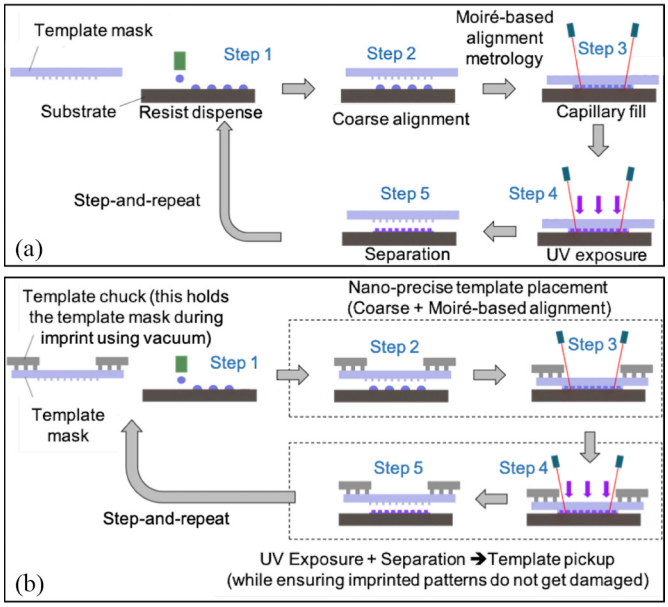

Fig. 3. (a) Standard J-FIL process consists of five primary substeps—step 1: resist dispense onto the to-be-patterned region(s) of the substrate, step 2: coarse ($\approx$1μm precision) alignment of the template mask on top of the inkjet-dispensed region(s), step 3: gradually bringing the template in contact with substrate, and nano-precision alignment of the template in-fluid (i.e., as the template is in contact with the inkjetted resist), step 4: UV exposure to cure the resist, step 5: separation of the template from the cured resist while ensuring that neither the template features nor the imprinted features get damaged. (b) J-FIL viewed as a vacuum-based pick-and-place technique—in every J-FIL step, a template chuck holds the template mask using vacuum. As the template is urged in contact with the substrate in step 3, it is essentially the template chuck precision-placing the template mask onto the substrate. In step 5, the template chuck picks up the template mask securely and repeats the sequence.

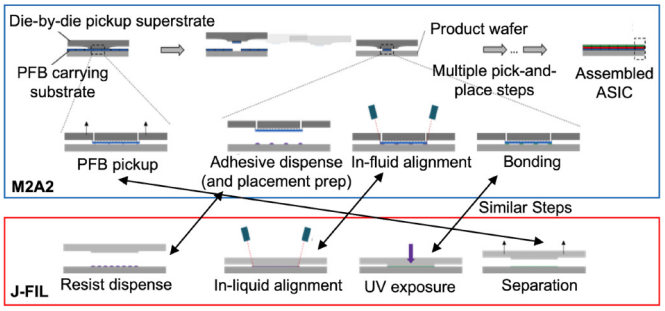

Fig. 4. Proposed pick-and-place process for M2A2 modeled along the lines of J-FIL. Here, the die-by-die pickup superstrate performs the same function as the template chuck during mask pickup in J-FIL. Once a PFB is picked, it is precisely placed and bonded to the product substrate in a manner similar to the in-liquid alignment step in J-FIL.

layers with vias which form front-end interconnects. It may also include power and ground rails at lower metal layer(s) to power the transistors. PFBs can be of multiple types, such as logic PFBs, memory PFBs, IO PFBs, macro cell PFBs to perform logic, memory, IO, and macro cell block operations, respectively. A typical PFB-based SoC can instantiate different types of PFBs multiple number of times (Fig. 2). The PFB instances are interconnected with higher metal layers (requiring low mask-set cost) to implement the desired functionality. In this article, we implement an SoC using logic PFBs. The logic PFB is comprised of optimally placed standard cells with its input and output pins not connected to another logic cell (spare gates).

III. M2A2 Fabrication Overview

The M2A2 fabrication process involves picking PFBs from source substrates and placing them onto product substrates with nanometer-scale placement precision. A placement precision of 1/6th of the PFB top-metal pitch would typically be required [9]. For instance, a top-metal pitch of 100 nm would require a placement precision of 16 nm (3σ). Nano-precise pick-and-place assembly required for the proposed M2A2 technology is based on J-FIL. Sub-20-nm (3σ) pattern placement accuracies have been demonstrated previously for J-FIL [7]–[8], [10]. J-FIL is a form of nanoimprint lithography (NIL), which uses low-viscosity ultra-violet (UV)-curable resists, along with room temperature and atmospheric pressure operation, to enable improved overlay control over other forms of NIL. J-FIL is currently being explored as a next-generation lithography technique for advanced memory [7], and is being deployed in production at the Toshiba Fab by Canon Nanotechnologies Inc.

In connection with M2A2 fabrication, J-FIL can be viewed as a pick-and-place technique for the template mask. Fig. 3(a) shows the standard J-FIL process, whereas Fig. 3(b) shows the J-FIL process viewed as a vacuum-based pick-and-place technique. Viewed from this perspective, J-FIL has already demonstrated sub-2.5-nm overlay precision in the pick-and-place of template masks [11]. The proposed pick-and-place process for M2A2 could utilize these commercially validated nano-precision capabilities of J-FIL for enabling the nano-precise assembly of PFBs. Fig. 4 shows a J-FIL-based pick-and-place process for M2A2. The following are some of the important components of the proposed J-FIL-based pick-and-place process.

A. PFB Fabrication on Source Wafers

PFBs are fabricated on source wafers using conventional semiconductor fabrication methods. These source wafers contain a buried sacrificial layer, such as a buried oxide layer (BOX), which can be partially etched off to leave behind

tethers. The tethers facilitate high-throughput pickup of the PFBs. Wafers with buried insulating layers, such as silicon-on-insulator (SOI), are one possible option for the source wafer. The experimental results for tether formation on SOI source wafers are presented in Section VI-A.

B. Preprocessing of Source Wafers

The source wafers go through the following preprocessing steps to demarcate the PFBs within the source wafer and form tethers which are critical for reliable (without damaging the circuit/functional elements) PFB pickup. Preprocessing is also essential for good bonding performance of the interconnects being formed between the PFBs and the CMD (during the CMD placement step).

1) Chemical Mechanical Polishing: Microscopic roughness of the PFBs can prevent bonding during the placement step. Chemical mechanical polishing (CMP) enables good bonding yield by ensuring that PFB surfaces are mirror polished [12].

2) Encapsulation Layer Coating: PFBs are exposed to corrosive etchants during the tether formation process. To protect the functional elements of PFBs from damage during the tether etch, an encapsulation layers is coated on the PFBs. The encapsulation is composed of chemically inert components, such as parylene, carbon, etc.

3) Access Hole Etch: Prior to the tether etch, access holes are etched through the PFBs, down to the buried sacrificial layer, for the tether etchant to be able to physically reach and etch the sacrificial layer. These access holes are generally relatively sparse (for instance, a 1-μm diameter hole per 10 μm $\times$ 10 μm PFB region).

4) Tether Etch: Tethers are formed in the buried sacrificial layer by partially etching it off. The etch is performed using vapor-phase etchants, to prevent PFB collapse due to stiction.

Some of the above preprocessing unit steps have been demonstrated on bare SOI wafers, and the results are presented in Section VI-B.

C. PFB Pickup

Once the source wafers are adequately preprocessed, a vacuum-based pick-and-place substrate is brought into contact with the source wafer. The vacuum is turned on at specific pickup locations and the PFBs are lifted away, in bulk, from the source wafer. Subsequently, individual PFBs can be picked from the PFB carrying substrate using the die-by-die pickup superstrate, as shown in Fig. 5. It should be noted that during all pickup steps (bulk and individual), proper attachment of PFBs to the carrying substrates needs to be ensured (otherwise, there is risk of large deformations, and subsequent PFB circuit damage or even destruction). As PFBs are lifted away from the source wafer (for instance), the following two competing effects take place: 1) as the gap expands between the PFBs and the source wafer, the air in the gap becomes rarer and 2) to try to increase the pressure of the rarefying air in the gap, air rushes in from the edges of the PFB, where large holes in the carrying substrates (10s of microns in width) maintain the pressure at 1 atm. These two effects have significantly different timescales. Thus, if a PFB is picked up faster than pressure equalizes in the gap, it could either lose suction entirely or in parts, and potentially be subjected to large deformations and circuit damage. To alleviate this risk, and ensure pickup with minimal PFB deformation, we have modeled the fluid flow conditions during the PFB pickup steps. The model and results are presented in Section VI-C.

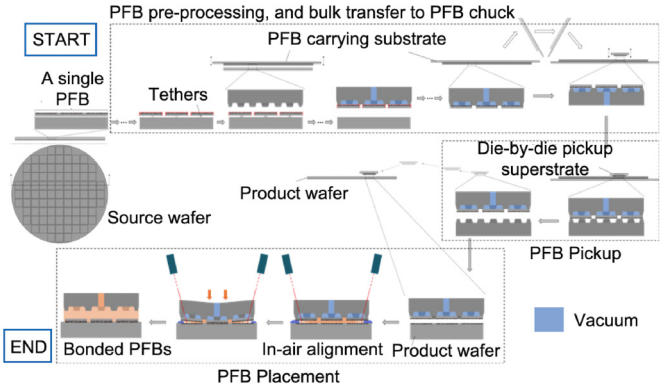

Fig. 5. Illustration of M2A2 pick-and-place assembly sequence in greater detail—PFBs fabricated on source wafers, are first processed to form tethers in the buried sacrificial layer and then transferred, in bulk, to a PFB carrying substrate. Subsequently, a die-by-die pickup superstrate picks up individual PFBs from the carrying substrate and precisely places them onto the product substrate. During the placement, an in-air alignment step can be used in place of in-liquid alignment, where the lubricating/bonding fluid is dispensed only on the periphery of the PFB and not in the center.

D. PFB Placement

Once picked up and attached to the (die-by-die) superstrate, the PFBs need to be placed onto a precise grid on the product wafer. This is necessary to ensure that the subsequent CMD placement (on the PFB knitted layer) can happen with the required alignment precision for formation of well-aligned interconnects (correctly functional) between the PFBs and CMD. During this step, the interfacial fluid between the product substrate and the PFB (liquid or air or a combination thereof) helps in maintaining lubrication between the two as the placement is taking place. This step is similar to the imprint step in J-FIL. Many of the solutions developed in J-FIL, such as the magnification/shape control system [13] (for correcting in-plane nano-scale distortions during PFB pickup), thermal and hybrid actuation systems [14] (for improved overlay control over and above the magnification/shape control system), and moiré-based overlay metrology [15] (for monitoring alignment of PFB with the substrate in the real time to ensure reliable, nano-precise placement) could be directly utilized during the placement of the PFBs. The alignment results obtained using the hybrid actuation approach are presented in Section VI-D.

E. PFB Bonding

To ensure that the PFBs do not lose their nano-precise registration (with respect to the product substrate), proper bonding of the PFBs to the substrate is necessary. During this step, PFBs are first temporarily and then permanently bonded to the product substrate. This step has been demonstrated before by a number of groups, and is commonly employed in wafer bonders [12], [16], [17] for packaging applications.

F. CMD Placement and Bonding on Top of PFB Assembled Layer

Once all the PFBs are assembled, a CMD, comprising of only higher metal interconnects, is placed on top of the prior PFB assembled layer. The CMD assembly is performed using a similar process sequence as used for the PFB assembly. In this way, a full SoC can be fabricated by assembling PFBs and CMD in the proposed M2A2 technology.

It should be noted that a thin oxide layer covers the surface of the copper contacts on the PFBs. Thus, once bonded and without any further processing, the Cu-Cu interface would have an oxide layer sandwiched between the copper surfaces. If left untreated, this could compromise the conductivity of the bonded interface. A bake step is performed to alleviate this problem—during the bake, copper oxide diffuses along the interface and forms clumps of copper oxide, which helps the rest of the interface form better quality Cu-Cu bonds [18].

IV. M2A2 EDA Methodology

In this section, we first present an overview of the proposed EDA design flow. Next, we discuss the PFB design algorithm to design logic PFBs. Then, M2A2 front-end design implementation flow comprising of the PFB knitting algorithm and post-Mask ECO synthesis is presented. Finally, back-end design implementation comprising of pre-CTS optimizations, post-Mask CTS, route, and buffer insertion solutions is discussed.

A. M2A2 EDA Flow Overview

Fig. 6 describes the EDA methodology for design implementation of logic PFBs-based SoC. In the logic PFB design generation, a limited number of PFBs are generated based on timing and placement data from multiple baseline ASICs (Section IV-B). Once the PFBs are generated, it serves as the design library. In the M2A2 design implementation phase, PFBs are first knitted together to realize a PFB-based SoC (Section IV-C). The PFBs are chosen and placed such that they meet the functional requirements of a given design. Next, PFB design and PFB knitted SoC placement data are processed to generate the netlist, and design exchange format (DEF) files in the data preparation phase (Section IV-C). Then, the commercial ECO tool (Cadence Conformal) is used to perform post-Mask ECO synthesis (Section IV-D). In the post-mask design, the base layer of transistors/standard cells remains frozen/fixed. This allows synthesizing of the design using spare cells preplaced in PFB knitted SoC. For best timing results, physical layout estimation (PLE) ECO synthesis flow is enabled [19]. Furthermore, for bigger benchmarks, min-cut partitioning is performed to segment benchmark into multiple smaller modules. Each module is first synthesized separately using PLE ECO synthesis flow, and then all modules are stitched back to realize a complete synthesized design for a given benchmark. These steps form the front-end design phase.

In the back-end design phase, preclock tree synthesis (pre-CTS) optimizations are first performed using a commercial EDA tool (Cadence Innovus) (Section IV-E). Next, post-mask CTS is performed. In order to implement post-Mask CTS, first a commercial EDA tool (Cadence Innovus) is used to build a clock tree by inserting cells in the desired regions of SoC. The desired regions are those where spare clock buffers, inverters

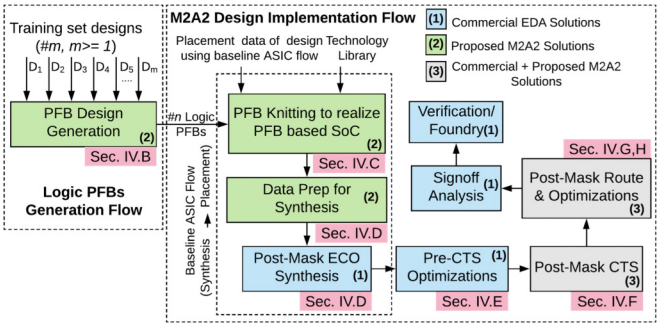

Fig. 6. Proposed M2A2 EDA flow for logic PFB-based SoC.

and clock gating cells are placed. Then, the newly added clock tree cells are mapped to the existing spare cells using min-cost bipartite graph matching algorithm (Section IV-F). Once the clock tree is built optimally using the spare cells, post-Mask routing and signoff analysis is performed using commercial EDA tools (Section IV-G). For post-route timing closure, post-Mask buffer insertion solution based on greedy matching technique is used to insert preplaced spare buffers in the paths to resolve the timing violations (Section IV-H). Finally, design is functionally verified and the GDSII file is generated.

B. PFB Design Algorithm

1) Overview: The PFB design problem can be formulated as designing a limited number of optimal PFBs which can be used to implement multiple ASICs. The random selection and adhoc placement of standard cells in PFBs may cause congestion, high interconnect delay, and timing closure issues. Further, PFB knitted design may end up using high number of PFBs, thus, degrading PPA metric. On the other hand, the greedy mapping-based clustering [20] techniques can be used for PFB design. However, these methods do not guarantee global optimal solution since decisions are made iteratively based on the information available in each iteration, rather than optimizing the overall objective function [20]. In order to design optimal PFBs, we propose a PFB design algorithm (Fig. 7) based on graph matching and unsupervised learning techniques.

2) Key Idea: The key idea is to design PFBs by learning from the placement of standard cells in multiple baseline ASICs (training set designs). We first identify regions in the training set designs which have similarity in the placement of standard cells using the min-cost bipartite graph matching technique [21]. Next, regions with similar standard cell placement are grouped together to generate PFBs using *k-means* clustering algorithm. The random initialization of centroids in *k-means* clustering usually results in a suboptimal solution [22]. To address this issue, we have determined initial centroid positions such that each centroid lies within unique cluster.

3) Rationale: PFBs are generated considering only the physical placement of standard cells, and not the logical connectivity among the cells. This is due to the fact that PFBs in this case comprise only of standard cells with floating input and output pins. In case, PFBs comprise of standard cells which are interconnected using metal layers (M2–M5), logical connectivity of cells should also be considered.

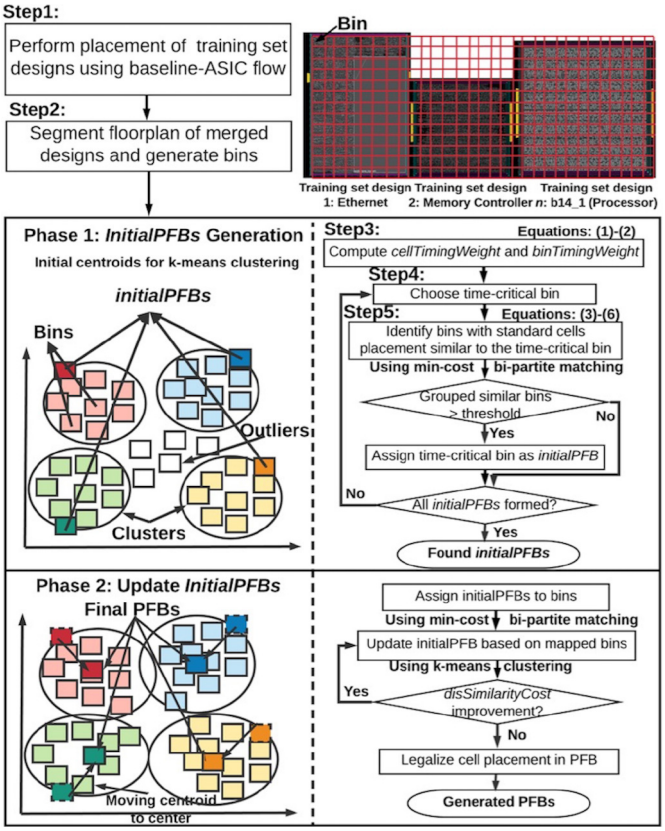

Fig. 7. Overview of the proposed PFB design algorithm.

4) Algorithm Details: The PFB design algorithm (Fig. 7) can be divided into two phases: 1) generating *initialPFBs* which serve as initial centroids for *k-means* clustering algorithm and 2) improving *initialPFBs* by performing *k-means* clustering and generating a final set of PFBs.

4.1) Phase 1 (InitialPFBs Generation): In step 1, the training set designs are synthesized and placed using baseline ASIC flow. Next, these training set designs are placed (merged) next to each other, and cells placement and timing (slack, period, etc.) information is processed.

In step 2, the floorplan of the merged designs is segmented into multiple small regions, named as bins [Fig. 7 and Fig. 8(a)]. The dimensions of bins are kept the same as that of a PFB (user input). Next, the relative location of each cell in a bin is calculated by assuming lower left corner of the bin as the origin [Fig. 8(b)].

The steps 3–5 determine the similarity in the placement of standard cells with the same functionality across different bins. It should be noted that only functionality (ignoring drive strength and VT class attributes) of the standard cell is considered in the similarity analysis. For example, cells *AND2X1_LVT* and *AND2X2_HVT* are considered the same, since both implement the functionality of a 2-input AND gate.

In step 3, *cellTimingWeight* metric for each cell instance is evaluated, as given by (1). The product of clock frequency and number of stages for each timing path passing through a given cell is computed, and the maximum product value is taken as a *cellTimingWeight*

$$\text{cellTimingWeight} = \max_{\forall\ \text{path}_i \in \text{all Paths}} \left(\#\text{stages}_{\text{path}_i} * \text{freq}_{\text{path}_i} \right). \tag{1}$$

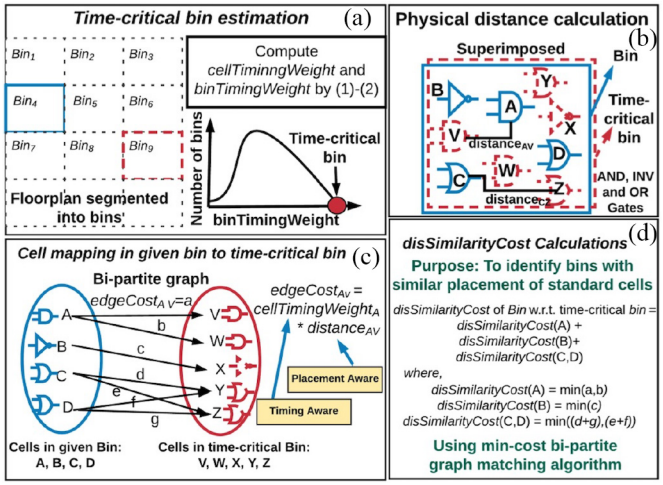

Fig. 8. Illustration of *disSimilarityCost* analysis in the PFB design algorithm.

Typically, higher number of stages, and/or faster clock speed leads to lower timing slack margin in each stage of a timing path. Thus, *cellTimingWeight* signifies the timing critical factor for a given cell. Then, *binTimingWeight* metric for each bin is calculated by accumulating the *cellTimingWeight* values of all the cells placed in a given bin, as given by (2). The bins where critical timing path cells are placed, and/or bins with higher number of cells have higher values of *binTimingWeight* metric [Fig. 8(a)]

$$\text{binTimingWeight} = \sum_{i=1}^{m} \text{cellTimingWeight}_i \tag{2}$$

where m is the total number of cells present in bin.

In step 4, the most timing critical bin (bin with the highest *binTimingWeight*) is assigned as the time-critical bin. All the other bins are compared with the time-critical bin. The similarity analysis is performed using a bipartite graph [Fig. 8(c)]. A bipartite graph is a set of graph vertices decomposed into two disjoint sets, say A and B such that every edge connects vertex in A to one in B [21]. The cells of a given bin (set A) are matched to the cells of the time-critical bin (set B). The edge cost in a bipartite graph for each pair of cells is then evaluated which represents the timing critical factor weighted manhattan distance between the cell in a given bin and time-critical bin. The manhattan distance between the relative locations of the cells in a given and time-critical bin is calculated as shown in Fig. 8(b). The *edgeCost* metric is then computed by taking the product of *cellTimingWeight* and its distance with the mapped cell the in the time-critical bin, as given by

$$\text{edgeCost}_{xy} = \text{cellTimingWeight}_x * \text{relative\_distance}(x, y) \tag{3}$$

where x is the cell in bin and y is the cell in time-critical bin.

In step 5, *matchingCost* for all the cells in a given bin is evaluated, as given by (4). The cells in a given bin are mapped to the logically equivalent cells in the time-critical bin such that the total cost of matching (*edgeCost*) in a bipartite graph is minimized [Fig. 8(d)]. In order to optimize for the run time, min-cost bipartite graph matching algorithm is implemented in

$O(n\log(n))$ time complexity, where n is the number of vertices (cells) to be matched. For the cells which are not matched to the cells of the time-critical bin, *penaltyCost* is determined, as given by (5). For each unmatched cell, the manhattan distance between a given cell and the farthest bin edge is multiplied by its *cellTimingWeight* to calculate its *edgeCost*. Then, *penaltyCost* is calculated by adding the *edgeCost* for all unmatched cells, and multiplying it with a penalty factor (p, set by user based on number of PFB types). The *disSimilarityCost* of each bin, given by (6), is then calculated by adding the *matchingCost* and *penaltyCost* for all logical types of cells placed in the bin. It qualitatively represents the dis-similarity in the standard cell placement between the given bin and the time-critical bin (Fig. 8)

$$\text{matchingCost} = \min\left(\sum_{j=1}^{m}\text{edgeCost}_j\right) \tag{4}$$

$$\text{penaltyCost} = p * \left(\sum_{k=1}^{q}\text{cellTimingWeight}_k * \text{edgeCost}_k\right) \tag{5}$$

$$\text{disSimilarityCost} = \sum_{i=1}^{n}\left(\text{matchingCost}_i + \text{penaltyCost}_i\right) \tag{6}$$

where m is the number of matched cells for a given logic type, q is the number of unmatched cells for a given logic type, n is the number of different logic cell types placed in the bin, and p is the penalty factor.

The bins with *disSimilarityCost* less than the threshold value are grouped together. This threshold value is set by the user based on target performance specifications. It signifies the maximum displacement allowed in the location of the same type of standard cells for a given bin, when compared to the cell locations in the time-critical bin. Next, *binTimingWeight* metric of the grouped bins is averaged out to determine *groupedBinsWeight* metric. This metric qualitatively represents the relative size of the cluster being formed by the grouped bins. Higher value signifies that a substantial number of bins are grouped together, and the cluster formed is not an outlier. To determine if the value of this metric is high or low, we compare it with threshold value, *groupingThreshold* metric, which is determined dynamically based on the required number of PFBs, earlier matched PFBs, etc. If the *groupedBinsWeight* metric value exceeds the *groupingThreshold* value, time-critical bin is assigned as an *initialPFB*. All the grouped bins are assigned as the matched bins. Otherwise, *initialPFB* is not formed. This process is repeated till we get the required number of *initialPFBs*. In the subsequent iterations, only the unmatched bins are considered.

The time complexity of this phase of the algorithm (*initialPFB* generation) is of the order of $O(\alpha * j * M * n\log(n))$ where M is the total number of bins, n is the average number of cells per bin, and α is the average of the fractions of unmatched bins over all iterations ($0 < \alpha < 1$), and j is the number of iterations to form the required number of *initialPFBs*. Here, the number of bins M depends on the floorplan dimensions of the training set designs and PFB dimensions. Typically, the number of iterations (j) value is much smaller than the total number of bins (M).

4.2) Phase 2 (Update InitialPFBs): The *k-means* clustering algorithm is performed next to improve the *initialPFB* design

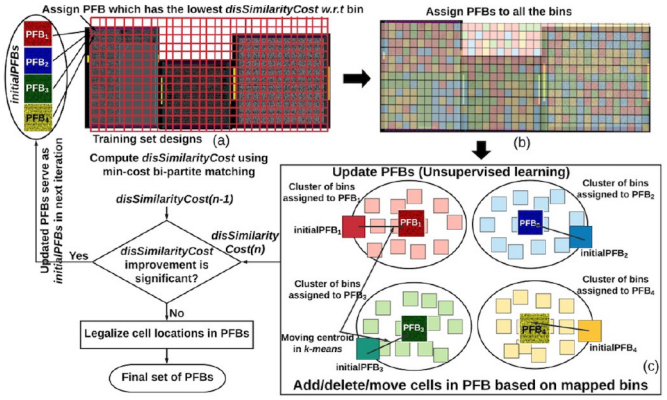

Fig. 9. *K-means* clustering to generate final PFBs in PFB design algorithm.

(Fig. 9). First, each bin is matched to one of the *initialPFBs* which has the lowest *disSimilarityCost* [Fig. 9(a) and (b)]. Once all the bins are assigned to one of the *initialPFBs*, *initialPFB* design is updated based on the matched bins [Fig. 9(c)]. The standard cells are added, and/or deleted and/or moved in *initialPFB* to reduce its *disSimilarityCost* with respect to the matched bins. This can be viewed as moving each centroid to the center of its cluster. The *k-means* clustering algorithm is used iteratively until no or minimal improvement in *disSimilarityCost* is observed. Finally, drive strength and VT class to each standard cell in PFB is assigned based on the matching bins, and standard cell placement legalization is performed in each PFB such that total cell displacement is minimized. The time complexity of this phase of the algorithm (updating *initialPFB*) is of the order of $O(i * k * M * n\log(n))$ where M is the total number of bins, n is the average number of cells per bin, k is the number of desired PFBs (same as the number of centroids in *k-means* clustering algorithm), and i is the number of iterations of *k-means* clustering algorithm. Here, i and k values are much smaller than the total number of bins (M). Hence, the overall time complexity of the PFB design algorithm is $O(\alpha * j * M * n\log(n)) + O(i * k * M * n\log(n)) \approx O(M * n\log(n))$.

C. PFB Knitting Algorithm

The goal of a PFB knitting algorithm (Fig. 10) is to choose and place PFBs on a substrate such that the PFB knitted SoC can realize the functionality of a given design at optimal PPA. The synthesis and placement of a given design is first performed using baseline ASIC flow to get the placement distribution of the standard cells. Then, PFBs are knitted onto SoC such that it resembles the standard cell placement of ASIC design. In order to do so, the entire ASIC design floorplan is first segmented into multiple small regions, named as bins [Fig. 10(a)]. The dimensions of the bin are kept same as that of a PFB. Next, *cellTimingWeight* and *binTimingWeight* for all the cells and bins are determined using (1) and (2), respectively. In the next step, valid PFB sites are defined which have standard cell utilization greater than the threshold value (depends on the area constraints).

Next, all the bins with valid PFB sites are matched to the PFBs using the min-cost bipartite graph matching technique. A given bin is compared with each PFB, and is mapped to the one which has the lowest *disSimilarityCost* [computed

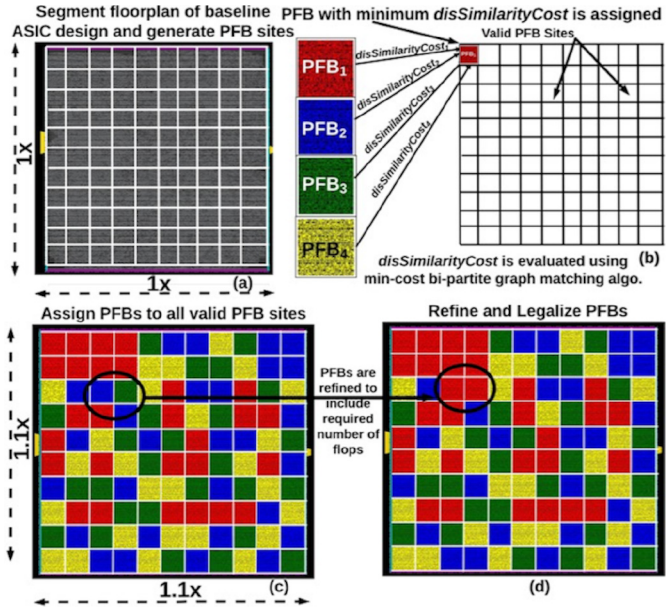

Fig. 10. Overview of the proposed PFB knitting algorithm on data encryption standard IP at 40 nm node.

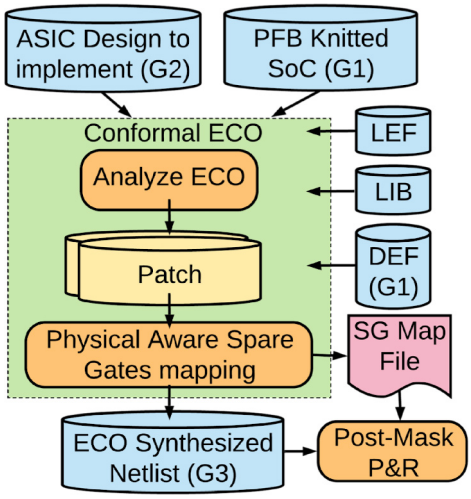

Fig. 11. Post-mask ECO synthesis flow using Cadence conformal ECO.

using (4)–(6), Fig. 10(b)]. Thus, a PFB whose standard cell placement is most similar to the standard cell placement of a PFB site is assigned to it. This step is repeated until one of the PFBs is assigned to each valid PFB site [Fig. 10(c)]. In this assignment process, it is ensured that the count of sequential elements (flip flops, latches, etc.) in the PFB knitted design is not less than the required count (used in baseline ASIC). This can be achieved by: 1) setting higher *cellTimingWeight* for the sequential elements so that PFBs dominated by sequential elements will be assigned to the PFB sites with higher sequential elements and 2) adding extra PFBs in the design such that the required number of sequential elements is placed in PFB knitted SoC.

Once the selection of PFBs is refined, the placement legalization of PFBs in the design is performed. In this step, the PFBs are aligned to the standard cell rows in such a way that total PFB displacement is minimized [Fig. 10(d)]. In this step, it is also ensured that spaces between PFBs is an integral multiple of filler PFB dimensions, so that filler cells PFB can be inserted in empty places to meet the density requirements. Finally, the PFB placement data reports are generated which are fed to the ECO tool to perform post-Mask ECO synthesis. The overall time complexity of the PFB knitting algorithm is of the order of $O(k * M * n \log(n))$ where M is the total number of bins in the design, n is the average number of cells per bin, and k is the total number of PFB types. Typically, the number of PFB types (k) is much smaller than total number of bins in design (M); thereby resulting in time complexity $\approx O(M * n \log(n))$.

D. Post-Mask ECO Synthesis

The placement data of a PFB knitted SoC is fed to the Cadence Conformal ECO tool to perform physical-aware post-Mask ECO synthesis [23]. According to Fig. 11, G1 represents the PFB knitted SoC. The netlist and DEF files of the PFB knitted SoC are generated based on the placement and design information of the standard cells preplaced in PFBs. G2 in Fig. 11 represents the ASIC design which needs to be implemented using M2A2 technology. The Conformal ECO tool reads in both the G1 and G2 design collaterals, maps the primary input and output ports, and analyzes the design to generate a patch file based on the G2 design functionality requirements. The patch file generated by ECO tool and library information (LEF, LIB) is then fed to a commercial synthesis tool (Cadence Genus) to perform post-Mask ECO synthesis. The ECO netlist generated (G3) synthesizes the functionality of the ASIC design to be implemented (G2) using spare cells placed in the PFB knitted design (G1).

The traditional synthesis tool uses wire-load models based on fan-out to estimate the interconnect delay. However, these models do not provide accurate wire delay information. For deep submicrometer designs, a significant portion of the delay is contributed by the interconnect delay. Thus, it becomes critical to estimate and optimize for interconnect delay during synthesis. Thus, Cadence physical location estimation (PLE) synthesis flow is used to perform timing (physical) aware post-Mask ECO synthesis [19]. This flow uses technology information from the LEF libraries and parasitic resistance and capacitance (RC) values from the capacitance tables to estimate the interconnect delays throughout the optimization process. Thus, for an M2A2 design with preplaced standard cells in a PFB knitted SoC, PLE flow performs better modeling of local interconnects and thereby improves the design performance. Besides enabling PLE synthesis flow, partitioning approach [24]–[26] is adopted for synthesizing large-sized designs to further improve the timing results. The design (G2) is segmented into n partitions using Fiduccia-Mattheyses (FM) min-cut partitioning algorithm [27], where each partition dimension is small enough to perform PLE post-Mask ECO synthesis without any long nets. The ECO netlist files for each partition are then stitched back to realize the synthesized design (G1').

E. Pre-CTS Optimizations

The pre-CTS step commences the back-end design phase of the M2A2 design implementation flow. The standard pre-CTS optimizations like pin swapping, cell swapping, and pre-CTS

useful skew optimizations, etc. are performed using a commercial EDA tool (Cadence Innovus).

F. Post-Mask CTS

Once the pre-CTS optimizations are performed, a clock tree is built [28]. To perform post-Mask CTS, first a commercial EDA tool is used to build the clock tree by inserting buffers (newly added CTS cells) in the desired regions of SoC. The desired regions are those where spare buffers, inverters, and clock gating cells are preplaced. The desired regions in the design are created using preferred cell stripes tool command in the Cadence Innovus P&R tool. Then, the newly added clock tree cells are mapped to the existing spare cells using min-cost bipartite graph matching technique, as discussed in Section IV-B. The newly added CTS cells form the one set, and spare buffers and inverter pairs form the other set of a bipartite graph. The edge cost of each newly added CTS cell is calculated with respect to all the spare cells placed in design. The edge cost is a function of the CTS cell fan-out and load capacitance, manhattan distance between the CTS cell and spare cell, and their drive strengths. If the edge cost exceeds a certain threshold value, the edge is removed from the bipartite graph. This typically happens when the manhattan distance between CTS cell and spare cell is large. The mapping of newly added CTS cells to the spare cells is performed such that total *edgeCost* is minimized. If the number of CTS added cells are more than the number of spare cells, extra (number of CTS cells - number of spare cells) CTS added cells are not mapped to any spare cell; and are removed from the design. The CTS cells which get mapped to spare cells are replaced with respective spare cells, such that clock tree is built using preplaced spare cells. Thus, post-Mask CTS is performed while optimizing for skew and insertion delay.

G. Post-Mask Route

Once the clock tree is built optimally without changing the base layer, post-Mask routing is performed using the commercial EDA tool (Cadence Innovus). All the tool optimizations which do not make changes to the base layer such as wire cutting, wire rerouting, post-Route useful skew, etc. are enabled during this step to fix setup and hold timing violations.

H. Post-Mask Buffer Insertion

The preplaced spare buffers are used to resolve the setup and hold timing violations in the routed design. It is challenging to insert buffers in the post-mask design (all the spare buffers are frozen/fixed) using the existing commercial EDA tools [29]. The proposed buffer insertion solution makes use of greedy-mapping-based heuristic technique to insert spare buffers or inverters in the timing violated nets. The details of this algorithm are as follows. First, all the violating timing paths are arranged in the descending order of negative slack. In step 2, for each violating timing path, negative slack and the maximum stage delay values are analyzed. If the maximum stage delay (cell with maximum propagation delay in a given timing path) value is high enough, the load capacitance, fan-out, and transition time values are analyzed to find the target net where the buffer should be inserted. Then, the spare buffer or inverter pair in the neighborhood is searched. This heuristic-based search considers the distance of the spare buffer from the target net and drive strength of the spare buffer to decide if the buffer should be inserted or not. If the buffer is inserted, the next timing path is analyzed. Otherwise, the cell with next to maximum stage delay value is taken and step 2 is repeated. Thus, preplaced buffers are inserted in violating timing paths to fix setup violations. It should be noted that the list of spare buffers and target nets is maintained to ensure that multiple buffers are not inserted on the same target net which exists in multiple timing paths. To fix the hold violation, a similar approach to insert the spare buffer in the target net is adopted.

V. Design Tradeoffs and Guidelines

In this section, various design guidelines and tradeoffs for optimal PFB design and knitting of PFBs are presented.

A. PFB Dimensions

The sizing of a PFB depends on various design and cost tradeoffs. The smaller PFB size results in less number of standard cells within each PFB. This makes it less generic, and more types of PFBs are needed to minimize the total *disSimilarityCost*. Increasing the required number of PFB types leads to increased NRE cost since it increases the assembly tool time and cost. On the other hand, smaller sized PFB generally leads to lesser area overhead of M2A2 designs when compared to ASICs, since additional PFBs add lesser area due to reduced dimensions. Moreover, *disSimilarityCost* in PFB knitting algorithm depends on the relative placement of standard cells in the bin with the matched PFB (discussed in Section IV-C). Thus, smaller sized PFBs usually results in smaller *disSimilarityCost* due to smaller PFB dimensions. This leads to lower interconnect delay and power dissipation of M2A2 enabled designs with smaller-sized more types of PFBs, when compared to less types of large sized PFBs, thereby improving PPA metric. From a fabrication perspective, smaller sized PFBs would result in higher overall time-of-assembly, and lower throughput for the assembly process. Smaller sized PFBs could also present challenges related to inline inspection of defects. Therefore, PFB dimensions need to be carefully chosen by assessing the tradeoffs between the NRE cost, the PPA impact on M2A2 designs, and the fabrication constraints.

B. Optimal/Limited Number of PFB Types

The *k-means* clustering (phase 2 of the PFB design algorithm) is performed for different values of k. The minimum value of k for which *disSimilarityCost* is not reduced further by increasing k is chosen as the optimal number of PFB types. This ensures that the least number of PFBs are designed which achieve reasonably good similarity in standard cell placement across regions/bins of the training set designs.

C. PFB Knitting Considerations

For successful synthesis of a given design, it is essential to have required number of standard cells in a PFB knitted SoC. The proposed PFB knitting algorithm ensures that sufficient number of sequential cells are placed in the PFB knitted SoC. For insufficient combinational standard cells, post-mask ECO synthesis (using an existing EDA tool) is performed to realize logic functionality using spare combinational logic gates.

If the synthesis still fails even after applying logic restructuring techniques, additional PFBs are knitted ensuring the rectilinear floorplan is maintained. In the worst case scenario, the addition of a single PFB to the floorplan may result in increasing the floorplan area by an entire row/column of the PFB (comprehended in M2A2 area calculations). In our analysis, single iteration of PFB inclusion followed by post-mask ECO synthesis enabled successful synthesis of a design.

D. Effectiveness of PFB-Based M2A2 for Variety of Designs

PFBs are generated by applying PFB design algorithm on the training set designs belonging to a certain set of functional categories. Hence, a given design having a similar functional composition can be optimally realized, thus making M2A2 well suited for ASICs requiring multiple variants of similar functionality designs (domain specific SoCs). However, limited PFBs may not optimally realize any arbitrary design, having different functional composition than the training set designs. This limitation is generic to training on a labeled dataset in machine learning (not specific to the M2A2). This limitation can be mitigated by expanding the PFB library to comprehend the functional composition of new designs at the expense of increased NRE cost due to additional PFBs.

E. Metal Layers Support in PFB

In advanced CMOS nodes, the intermediate metal layers (M2-M4) may require critical mask-set incurring high NRE costs; thus requiring PFBs to include intermediate metal layers. To include the intermediate metal layers, there can be two possibilities. First, PFBs comprise of spare cells with M1-Via1-M2-Via2-M3-Via3 (super-via) for all input/output pins of standard cells. This will reduce the placement density of spare cells to meet the DRCs for super-vias, thereby resulting in PPA degradation. Second, the intermediate layers can be used to interconnect cells in PFB, thereby resulting in no spare cells. This can limit the PFB design flexibility, and may result in a significant increase in the number of PFB types and/or high PFB instances are required to knit a design. Thus, tradeoff analysis among the number of PFB types, number of metal layers and PPA of M2A2 design is necessary.

VI. Experimental Results—M2A2 Fabrication

In this section, we present the experimental results related to the process steps involved in the M2A2 fabrication.

A. PFB Fabrication on Source Wafers

The source wafer consisted of 200 mm Soitec Unibond wafers [30], with 1.5-μm thick device (Si) layer and 1-μm thick sacrificial oxide layer was used to fabricate PFBs. These wafers were procured from Nova Electronic Materials [31].

B. Preprocessing of Source Wafers

The following process steps were performed to demonstrate PFB preprocessing up till the point of tether formation.

1) Lithography and Etching of Access Holes: The access holes consisted of 10 μm diameter holes in a square grid, with a pitch of 40 μm [Fig. 12(a)]. The lithography to form the access holes was performed using a Carl SUSS MA6 aligner [32], and the etch using a PlasmaTherm Versaline Deep

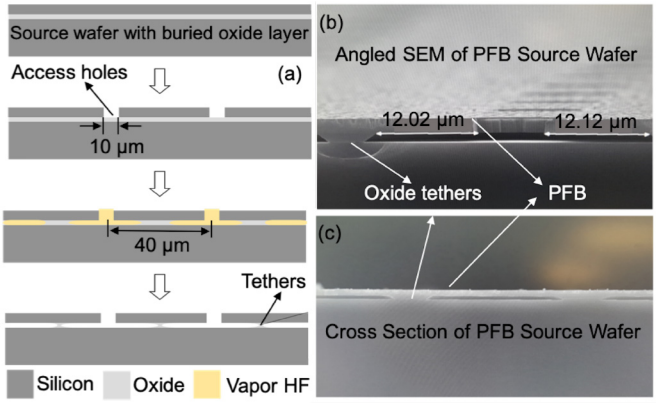

Fig. 12. (a) Process flow for demonstration of tether formation in SOI wafers. PFBs are supported on the silicon wafer using oxide tethers. (b) Angled SEM image. (c) Cross-section SEM image.

Silicon etcher [33]. Both process steps were conducted at the Microelectronics Research Center [34] at UT Austin.

2) Sacrificial Layer Etching: The sacrificial layer etch to form the tethers was performed using hydrofluoric acid (HF) [Fig. 12(a)]. HF in vapor form (vHF) is used as an etchant of silicon dioxide, since it has excellent selectivity between silicon and the oxide (vHF attacks Si at vanishingly small rates). A custom 8 In vHF etcher was developed for the sacrificial layer etch. The etcher can accommodate both silicon and glass substrates, up to 8 In $\times$ 8 In in size. It can mask parts of the substrate using custom teflon (PTFE) masks. An externally attached heater can control the temperature of the substrates from 40 °C to 60 °C. Fig. 12(b) and (c) shows SEM images of a vHF-etched wafer with oxide tethers. These particular tethers were etched for 36 min with an etch rate of 335 nm/min at an etchant temperature of 45 °C.

C. PFB Pickup

As described earlier in Section III-C, if PFBs are picked up faster than the pressure equalizes in the gap (between the PFB and the carrying substrate), they risk losing suction and potentially getting damaged or destroyed. To alleviate this risk, we have derived a suction ensuring motion plan for the superstrate and the PFB carrying substrates.

To derive suction-ensuring superstrate motion plans, time estimates are derived using Monte Carlo simulations. In the molecular and transitional flow regimes, which occur during PFB pickup and placement, the assumptions of continuum and thermodynamic equilibrium break down. Thus, conventional Navier-Stokes-based tools (for example, ANSYS Fluent CFD [35]) can no longer be used to predict the air flow. One needs to solve the Boltzmann transport equation (BTE), which offers a more fundamental description of gas flow, to predict the air flow in these regimes. We have chosen the direct simulation Monte Carlo (DSMC) method [36] for the air flow simulations of pick-and-place assembly. DSMC scales well for two and 3-D problems, is readily parallelizable, and mature open-source simulation tools are available. We have used the open source toolbox dsmcFoam, which is part of the OpenFOAM project [37], for their simulations. The code is primarily run on the Stampede2 supercomputer at the Texas Advanced Computing Center (TACC) [38].

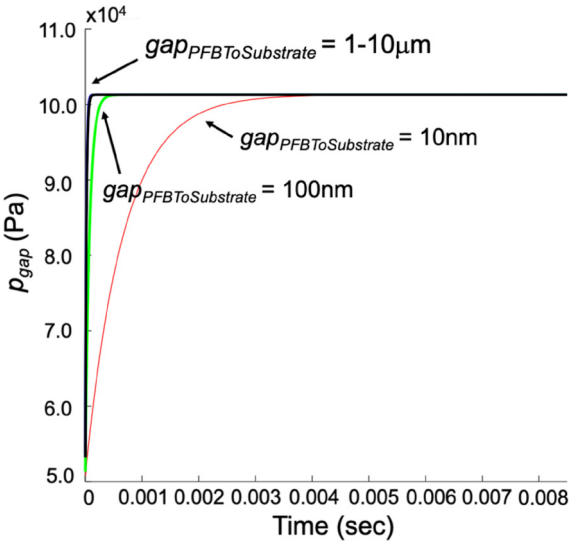

Fig. 13. Evolution of pressure in the gap between the PFB and the carrying substrate, *pgap versus* time at various values of the starting gap value, $gap_{PFBToSubstrate}$, for PFB width of 5 mm.

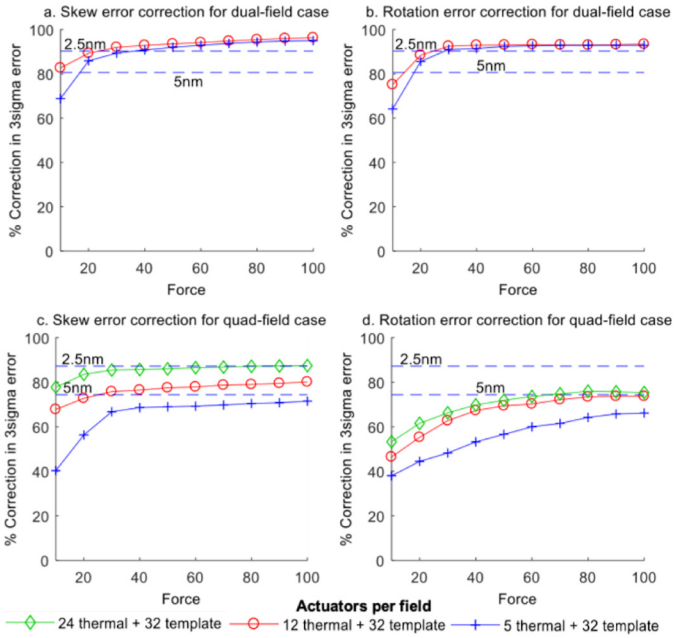

Fig. 14. Percent reduction in overlay error vs limiting (template) actuator force for dual (a), (b) and quad-field (c), (d) overlay cases using hybrid actuation [14]. Here "3sigma error" refers to the mean + 3 × standard deviation metric for overlay error. The dual (or quad) field error case consists of the case where one of two (or four) lithographic fields has the specified overlay error, with the goal being to simultaneously correct the error in all fields.

Fig. 13 shows the evolution of the pressure in the gap between PFBs and the source wafer as they are being picked up for PFBs of width 5 mm, at various values of the starting gap. Note that the pressure equalization rate starts slowing down significantly around a gap of ≈100 nm, which is likely when the molecular flow starts dominating. At these gaps, to ensure suction, the pickup would have to be done at a correspondingly slower rate. For instance, at an initial gap of 10 nm, the pickup would have to be done at a rate of about 3.3 μm/s (≈10 nm/0.003 s). Thus, PFB pick-up rates, which ensure proper suction without damaging PFB circuit can be determined using this model.

D. PFB Placement

We have explored a hybrid actuation scheme, which optimally combines template and wafer thermal actuation, for sub-5 nm overlay control in J-FIL, with potential applicability to nano-precise placement of PFBs as well. To obtain sub-5 nm large-area residual overlay using template actuation alone, the number of actuators per side has to be prohibitively high from a design standpoint. On the other hand, thermal actuation provides better correction for magnification and translation errors, but almost none for theta errors. A combination of the above two schemes can reduce the deficiencies with each individual scheme. Fig. 14 shows our simulation results of the hybrid actuation scheme for various J-FIL field configurations [14]. It can be seen that sub-5 nm overlay can be obtained using this scheme even with the challenging rotation and skew-type error cases. This is promising for PFB placement, where the multiple modes of overlay correction, provided by hybrid actuation, could help in achieving nano-precise placement.

VII. Simulation Results—M2A2 EDA Methodology

In this section, we first present the criteria of choosing training and test set benchmarks, and design parameters used to evaluate the performance of M2A2, sASIC, baseline ASIC, and FPGA-based designs. Next, performance of M2A2-based designs is compared with baseline ASICs, FPGAs, and sASICs (LUT, sASIC, and AOI22) at 130 nm node. Then, performance of M2A2, ASIC, and FPGA designs is analyzed for same and different functional categories benchmarks using commercial 40nm process node libraries. Finally, M2A2-based designs performance is compared with ASICs and FPGA designs at advanced CMOS node using open source 15 nm process node libraries.

A. Criteria of Choosing Training Set Designs

The first step in M2A2 analysis is to generate PFBs using representative training set designs. Nine IWLS'05 benchmarks [39] (*des_area*, *des_perf*, *b14_1*, *b15_1*, *s35932*, *s38584*, *s38584*, *s38417*, *vga_lcd*, and *Ethernet*) with larger gate count (3 k–124 k) from different functional categories, such as encryption standards, processors, controllers, and communication IPs are chosen as the training set designs. The rationale behind choosing these benchmarks is: 1) to compare M2A2 results with sASIC-based designs on same IPs/benchmarks [40], [41] and 2) open source benchmarks with multiple functional categories. These benchmarks vary significantly in terms of standard cell placement. This diverse placement distribution avoids over-fitting in the training set data to design PFBs. In a real SoC design, variants of different IPs can be chosen as the training set to design domain-specific PFBs.

B. Criteria of Choosing Testing Set Designs

The testing is performed for designs belonging to the same functional categories (same as the training set design categories) as well as different functional categories which are

TABLE II
M2A2 DESIGN METRICS

Metrics	130nm	40nm	16nm	16nm*
PFB Width	110.00um	55.00um	28.00um	31.00um
PFB Length	110.16um	55.44um	28.45 um	30.72 um
#PFB Types	5	4	4	4
PFB metal layers	M1	M1	M0, M1	M0-M3
Backend CMD metal layers	M2-M8	M2-M8	M2-M8	M4-M8

**PFB comprises of super-via (M1-M3), CMD comprises of M4-M8*

TABLE III
FPGA DESIGN METRICS

Metrics	130nm	40nm	16nm
Device Family	Virtex-II	Virtex-6	Virtex UltraScale+
Device	xc2v250	xc6vlx75t	xcvu3p
Package	fg456	ff484	ffvc1517
Speed Grade	-6	-2	-3

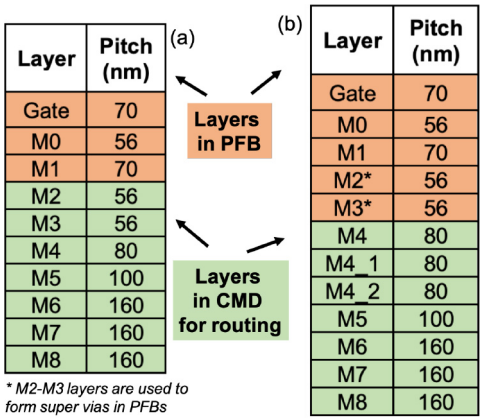

Fig. 15. 16-nm layer stack for M2A2 designs with CMD comprising of: (a) M2–M8 and (b) M4–M8 metal layers.

not used in training (*b12*, *b21_1*, *s5378*, *aes_cipher*, *spi*, and *wb_dma*). Thus, the effectiveness of the M2A2 methodology for realizing domain-specific designs as well as random (out of domain) designs can be evaluated. Out of nine IWLS'05 [39] testing set designs, six designs belong to the same functional categories as that of the training set designs (same domain) while three designs belong to different functional categories (randomly chosen).

C. M2A2 Design Parameters

Table II lists down the important M2A2 design parameters (PFB dimensions, number of PFB types, number of metal layers in PFB) for 130 nm, 40 nm, and 16 nm CMOS nodes, which are used in our performance analysis. To compare the PPA metric of M2A2 designs with sASICs, ASICs, and FPGAs at iso-technology CMOS node (130 nm and 40 nm), typical corner design libraries provided by a commercial foundry at 130 nm and 40 nm are, respectively, used. To compare the performance of M2A2 designs at an advanced CMOS node, 15-nm FinFET-based open cell library (OCL) is used [42].

The OCL technology LEF library uses metal pitch of 64 nm for layers M1–M6. To reflect the typical back-end metal layer spacing and pitch, Intel 14-nm CMOS process metal pitch information is used [43]. Four types of PFBs, (each sized 28.00 μm $\times$ 28.45 μm) comprising of M0–M1 metal layers are used. This design approach may result in two issues: 1) the NRE cost associated with intermediate metal layers (M2–M4) for 16-nm node is high; thereby minimizing the NRE savings of the M2A2 technology and 2) the overlay alignment required to precisely knit PFBs and CMDs is sub-10-nm (challenging), since M2 metal pitch is 56nm, and overlay alignment is typically 1/6th to 1/10th of the metal pitch [9]. To address the above mentioned issues, intermediate metal layers (M2, M3) are added to PFBs such that the back-end metal stack comprises of metal layers M4–M8 with metal pitch $\geq$ 80 nm. PFBs comprise of M1-Via1-M2-Via2-M3-Via3 super-via at all the floating input and output pins of all the spare standard cells. To meet the design rule checks (DRCs) for super-vias, placement density of the standard cells is relaxed, resulting in larger sized PFBs (31.00 μm $\times$ 30.72 μm). Since M2–M3 metal layers are not used for routing, two additional M4 layers (M4_1, M4_2) with metal pitch same as M4 layer are added to route the design, as shown in Fig. 15.

The PFB dimension in each technology node is chosen such that each PFB comprises of $\approx$1000 standard cells. However, PFB dimension should be determined optimally considering various design and cost tradeoffs. The number of PFB types is obtained by executing PFB design algorithm for different values of k, and the minimum value of k for which *disSimilarityCost* is not reduced further by increasing k is chosen as the number of PFB types, as discussed in Section V. The PFB design generation algorithm has taken around 6–8 h to generate desired PFBs for each technology node. Comparing the run times for IWLS designs implemented using baseline ASIC and M2A2 implementation, the M2A2 enabled designs have 25%–30% less tool run-times (compared to baseline ASIC flow) in PNR implementation, since only post-Mask back-end design optimizations are enabled. The synthesis process takes almost the same run-time in both the design implementations.

D. FPGA Design Metrics

In order to compare M2A2 designs performance with FPGAs at iso-technology node, Xilinx Virtex-II [44], Xilinx Virtex-6 [45], and Xilinx Virtex UltraScale+ [46] device family FPGAs are used for 130 nm, 40 nm, and 16 nm node comparisons, respectively. Table III summarizes the FPGA device configurations used for each technology node. The maximum speed-grade, smallest size devices, and packages are used for each family. However, for larger IWLS benchmarks (*des_perf* and *aes_cipher*), a larger package size device with sufficient input output blocks (IOBs) is chosen.

E. EDA Tools for Baseline ASIC, M2A2, and FPGA Designs

For baseline ASIC design implementation, synthesis is performed using Cadence Genus [19], back-end physical design implementation and parasitic extraction using Cadence Innovus [47], and timing/power analysis is performed using Synopsys PrimeTime tool [48]. For M2A2-based designs implementation, Cadence ECO Conformal tool [23] is used for post-Mask ECO synthesis, Cadence Innovus tool is used for

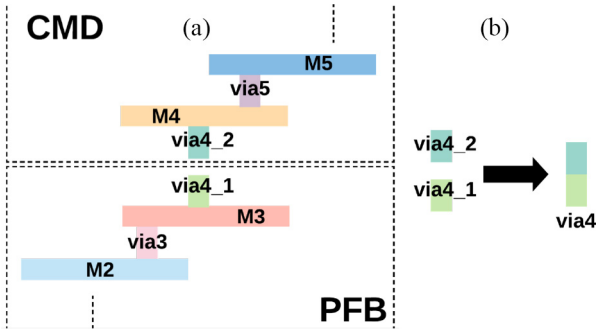

Fig. 16. (a) Illustration of metal and via layers in PFB and CMD. (b) via4 definition by stacking vias in the M2A2 design flow.

back-end design implementation. For FPGA designs implementation, synthesis, back-end design implementation, and timing analysis is performed using the Xilinx Vivado HLS tool [49] for Virtex-6 and Virtex UltraScale+ FPGA families, whereas Xilinx ISE tool [50] is used for Virtex-II family. The power analysis is performed using the Xilinx XPower Analyzer.

F. Timing Constraints for Baseline ASIC, M2A2, and FPGAs

We have adopted the method of Hoe *et al.* [40], [41] and Kuon and Rose [51] to compare M2A2, FPGA, sASIC, and ASIC design performance. The desired clock rate was set to an unattainable higher frequency during a first round of physical synthesis (physical aware by providing post-placement DEF), and the resulting frequency obtained was used during a second round of synthesis and PNR. At iso-technology node, the same constraints file (used for baseline ASIC implementation) was used for the first round of synthesis of M2A2 and FPGA-based designs. The changes to clock period were made based on resulting frequency to perform synthesis and PNR.

G. Parasitic Extraction for M2A2 Designs

For M2A2 design implementation, the parasitics are extracted using the same design libraries (Technology LEF and capacitance table files) provided by TSMC and OCL, which were used for ASIC implementations, since the PFBs and CMD are manufactured at the commercial foundry using the same process flow (discussed in Section II, Fig. 2). The only change in parasitics would occur in the via which connects the PFB to the CMD. For example, if PFB comprises up to M3 metal layers, and CMD comprises of M4–M8 metal stack [Fig. 16(a)]. Then, via4 is formed by bonding via4_1 (part of PFB) and via4_2 (part of CMD). The *RC* values of via4 depend on the foundry process technology (determines via4_1 and via4_2 dimensions/RC values), pick and-place alignment precision and the bonding characteristics (contact RC values). We have created a via4 definition in the TECH LEF file which stacks two conventional vias (via4_1 and via4_2), as shown in Fig. 16(b). The stacked via is the sole "via4" definition used in M2A2 design implementation. Since, the alignment precision would be sub 5-nm, the vias can be assumed to be well aligned, and any changes in *RC* values due to alignment distortion can be ignored. We have assumed an ideal bond between the two vias (via4_1 and via4_2) with very little contact resistance [18] (discussed in Section III-F), which is ignored in determining the *RC* values of via4.

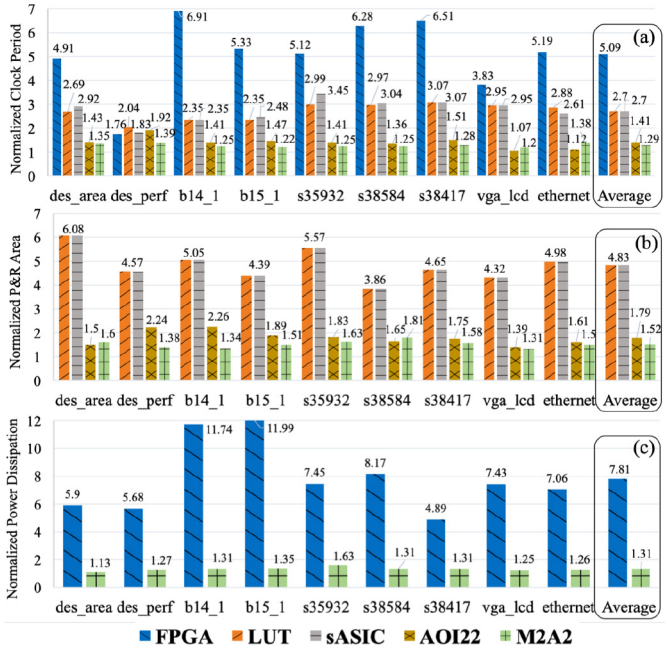

Fig. 17. (a) Normalized clock period. (b) Normalized P&R area. (c) Normalized power dissipation comparison for different design technologies at 130 nm node.

H. Power Analysis for Baseline ASIC, M2A2, and FPGAs

For conservative and fair power comparison of FPGA designs with M2A2-based designs, total FPGA power includes clock power, CLB power, and signals power. The leakage and IO power in FPGAs has been ignored since design implementation on the chosen FPGA device might result in redundant IOBs and CLB logic slices with high leakage power. Similarly, leakage power is ignored in power analysis of ASIC and M2A2 designs. The static probability and toggle probability are kept the same (50% and at 25%) in baseline ASIC, M2A2, and FPGA-based design implementations for fair power comparison.

I. Comparison With Baseline ASICs, Structured ASICs, and FPGAs at 130 nm Node

In 130-nm process node, nine IWLS benchmarks are used in the training and testing set to compare the area and the clock period of M2A2-based designs with existing sASIC design technologies, such as LUT [41], sASIC [40], and AOI22 [41], as well as baseline ASIC and FPGA design technologies. Fig. 17 shows the clock period, area, and power comparison of different design technologies (normalized to baseline ASICs) at 130 nm node. The area comparison is not made for FPGAs since CLB area for Virtex-II FPGA is unknown. The power dissipation for prior sASIC design technologies (sASIC, LUT, AOI22) is not reported. Hence, clock period and area comparison is made for sASICs [40], [41].

Over set of these benchmarks, M2A2 designs achieve 15%–68.5% smaller area and 8.5%–52% lower clock period when compared to sASICs [Fig. 17(a) and (b)]. M2A2 designs when compared with the FPGA design implementations result in 74.66% lower clock period, and 83.23% lower power dissipation, leading to power delay product benefit of 27.25× for M2A2-based designs. On the other hand, M2A2 designs consume 31% more power, occupy 52% more area, and operate

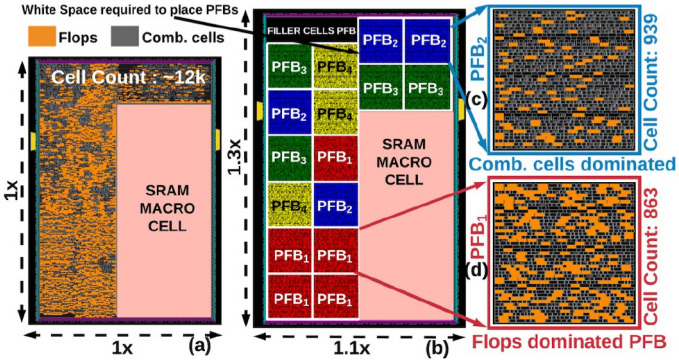

Fig. 18. 40-nm Ethernet comparison. (a) Baseline ASIC flow. (b) Proposed M2A2 flow. (c) Comb. logic dominated PFB. (d) Flops dominated PFB.

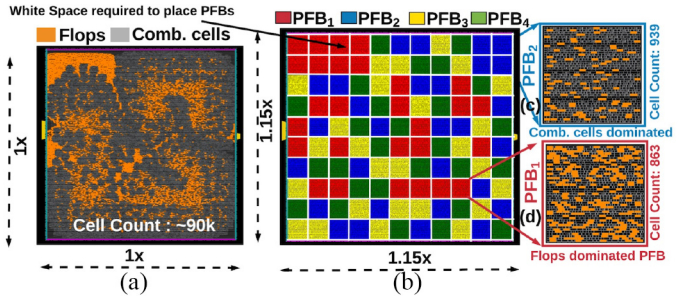

Fig. 19. 40-nm Data Encryption Standard (des_perf) floorplan comparison. (a) Baseline ASIC flow. (b) Proposed M2A2 flow. (c) Combinational cells dominated PFB. (d) Flops dominated PFB.

29% slower in 130 nm node when compared to baseline (standard cell) ASICs (Fig. 17). The routing congestion analysis show that M2A2 designs result in 8.5% and 31.7% decrease in wirelength when compared to AOI22/LUT designs, respectively; and significant improvement in interconnect delay compared to FPGAs.

J. Comparison With Baseline ASICs and FPGAs at 40 nm (Same Functional Categories of Training and Test Set Designs)

To evaluate and compare the PPA metric (Power × Clock period × Area) of M2A2-based designs with ASICs and FPGAs, nine designs for training set and six designs of the same functional category for testing set are used. Figs. 18 and 19 show the ASIC and M2A2-based design implementation of *Ethernet* IP and *des_perf* benchmarks, respectively, in 40 nm node. The area comparison of FPGAs is not made with baseline ASICs since the area of Virtex-6 CLB is unknown. Over set of 15 IWLS benchmarks, FPGA-based designs operate 4.80× slower and dissipate 10.42× more power when compared to baseline ASIC designs. However, M2A2-based designs occupy 49% more area, operate 29% slower and dissipate 45% more power when compared to baseline ASIC designs. Comparing M2A2 and FPGA designs performance, M2A2 designs result in 27.11× power delay product benefit when compared with FPGA designs. It should be noted that the training and testing set designs have shown consistent value of PPA proving the scalability of M2A2 designs for domain-specific applications, i.e., designs with similar functional categories.

K. Comparison With Baseline ASICs at 40 nm Node (Different Functional Categories of Training and Test Set Designs)

To evaluate the effectiveness of the M2A2 methodology to design a random ASIC from the PFBs (generated using training set designs on different functional categories), three IWLS benchmarks are randomly chosen as test designs. Over set of three benchmarks, M2A2 designs occupy 52% more area, operate 38% slower, and dissipate 53% more power when compared to baseline ASIC designs. The designs were able to be implemented using a higher number of PFBs resulting in 21% PPA degradation compared to designs belonging to the same functional categories. This indicates the importance of having similar functional categories of the training and test set designs, suggesting suitability in realizing domain-specific design better than any randomly selected design. However, it should be noted that M2A2 methodology is still effective to realize arbitrary designs/benchmarks, when compared with FPGAs. Over set of these three benchmarks, FPGA designs operate 3.52× slower and dissipate 6.72× more power when compared to M2A2-based designs.

L. Comparison With Baseline ASICs and FPGAs at 16 nm Node (PFBs Upto M1 Layer)

To evaluate and compare the PPA metric of M2A2-based designs with ASICs and FPGAs at advanced 16 nm CMOS node, nine designs in the training set and six designs of the same functional category in testing set are used. Over set of 15 IWLS benchmarks, FPGA-based designs operate 8.93× slower and dissipate 8.74× more power when compared to baseline ASIC designs. The area comparison of FPGAs is not made with baseline ASICs since the area of the CLB in Virtex UltraScale+ is unknown. M2A2-based designs occupy 52% more area, operate 31% slower, and dissipate 59% more power when compared to baseline ASIC designs. The PPA metric of M2A2 designs when compared with baseline ASICs increased from 2.81× in 40-nm node to 3.21× in 16-nm node; leading to 14.23% PPA degradation. This degradation can be attributed to higher interconnect parasitics resulting in more power and slower clock period. Similarly, FPGA design implementations also result in power delay product degradation due to increased sensitivity of parasitics. Thus, M2A2 designs result in 34.89× power delay product benefit when compared with FPGAs.

M. Comparison With Baseline ASICs and FPGAs at 16 nm Node (PFBs Upto M3 Layer)

As discussed above in Section VII-C, PFBs comprising of M1–M3 super-via are used to lower the NRE cost and relax the overlay alignment requirements. This approach results in a slight increase in the area due to reduced placement density of cells, higher latency and power due to longer interconnects. Over set of 15 IWLS benchmarks, M2A2-based designs occupy 81% more area, operate 39% slower, and dissipate 70% more power when compared to baseline ASIC designs. When compared with FPGAs, M2A2-based designs with super-via PFBs still result in 30.66× power delay product benefit, thus making M2A2 technology preferable at advanced CMOS nodes.

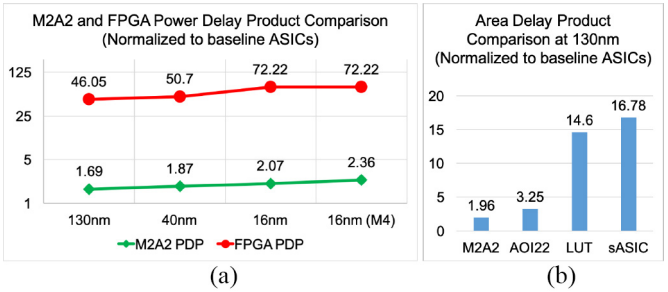

Fig. 20. Post-PNR the simulation results summary. (a) Normalized power delay product comparison of M2A2 and FPGA designs at 130 nm, 40 nm, and 16 nm nodes). (b) Normalized area delay product comparison of M2A2, AOI22, LUT, and sASIC designs at 130 nm.

N. Post-PNR Simulation Results Summary

Fig. 20 summarizes the performance of M2A2-based designs over 15 IWLS benchmarks at 130 nm, 40 nm, and 16 nm CMOS nodes as presented in Sections VII-I–VII-L. The proposed M2A2 technology-enabled designs achieve power-delay-product (PDP) benefit of $27.11\times - 34.89\times$ when compared with FPGAs, and are $1.69\times - 2.36\times$ worse compared to baseline ASICs. The M2A2 designs achieve 15%–68.5% smaller area and 8.5%–52% higher performance ($1.66\times - 8.56\times$ area-delay product benefit) compared to earlier sASIC methodologies.

VIII. Design Limitations and Improvements

In this section, design improvements and mitigation strategies are proposed to minimize some of the limitations.

A. M2A2 Fabrication: Challenges and Future Work

The proposed M2A2 pick-and-place technology has some differences from J-FIL which would require further development of M2A2-specific nano-precision pick-and-place methods. For instance, during the step of PFB pickup, which corresponds to the template separation in J-FIL, the amount of the overlay control required is higher in pick-and-place assembly than in J-FIL, which would require greater focus on the pickup trajectory of the PFB carrying substrate and the die-by-die pickup superstrate. Future work to fabricate test-chip using M2A2 technology would also include the fabrication of the PFB carrying substrates, fabrication of source wafers with real devices, retrofitting an actual J-FIL tool to perform pick-and-place assembly, development of in-situ vHF etching, and finally testing of assembly overlay performance.

B. M2A2 EDA: Limitations and Improvements

The PFB design flow can be improved by incorporating logic restructuring techniques in PFB design algorithm. This optimization can result in use of lesser number of PFBs, or PPA metric improvement using an existing set of PFBs. Further, spare gates in PFB occupying till M1 metal layer will increase the NRE cost for designs implemented in aggressive process nodes, whereas PFBs occupying till M3 super-via degrades the PPA of M2A2-based designs. In order to overcome these limitations, critical metal layers till M3–M4 need to be used to make interconnections in PFB. This makes the PFB design harder, since PFB no longer contains spare cells. The spare cells are interconnected in PFB to form functional elements using these intermediate metal layers. This limits the design flexibility of PFB, which may result in a significant increase in the number of PFBs required in the design library. It requires better understanding of cost models to address the tradeoffs between the number of PFBs, number of metal layers in PFB and M2A2-based designs PPA. The current greedy mapping-based buffer insertion solution also needs to be improved.

TABLE IV
Abbreviations

Abbr.	Full Form	Abbr.	Full Form
AOI22	2-2 AND-OR-Invert (AOI) gate	LIB	Liberty Timing File
ASIC	Application Specific Integrated Circuit	M2A2	Microscale Modular Assembled ASIC
BOX	Buried Oxide Layer	NIL	Nanoimprint Lithography
CLB	Configurable Logic Block	NRE	Non-Recurring Engineering
CMD	Custom Metal Die	OCL	Open Cell Library
CMP	Chemical Mechanical Polishing	PFB	Pre-fabricated Block
CTS	Clock Tree Synthesis	PDP	Power Delay Product
DEF	Design Exchange Format	PNR	Placement and Route
ECO	Engineering Change Order	PPA	Power-Performance-Area
FPGA	Field Programmable Gate Array	PLE	Physical Layout Estimation
IOB	Input Output Block	SoC	System on Chip
IWLS	International Workshop on Logic Synthesis	SOI	Silicon-on-Insulator
J-FIL	Jet-and-Flash Imprint Lithography	sASIC	Structured ASIC
LEF	Library Exchange Format	TTM	Time-to-Market
LUT	Look-up Table	vHF	Vapor Hydrofluoric Acid

IX. Conclusion

In this article, we proposed M2A2 technology as a cost-effective solution for high-mix, low-volume heterogeneously integrated ASIC designs. High NRE mask-set cost is shared across many ASIC designs using the limited number of PFB types. An entire EDA design methodology implementing unsupervised learning and graph matching algorithms, as well as leveraging existing commercial EDA tools infrastructure is discussed in detail. The post-PNR simulation results achieved over 15 IWLS benchmarks show that the proposed M2A2 technology-enabled designs achieve PDP benefit of $27.11\times - 34.89\times$ when compared with FPGAs, and are $1.69\times - 2.36\times$ worse compared to baseline ASICs. The M2A2 designs achieve 15%–68.5% smaller area and 8.5%–52% higher performance compared to earlier proposed sASIC methodologies. M2A2 enabled designs power-delay product benefit over FPGAs increases at advanced nodes due to more sensitivity of interconnect delay/parasitics, thereby making M2A2 technology preferable to realize cost-effective high-performance domain specific low-volume high-mix ASICs. Moreover, the key fabrication steps in the proposed M2A2 technology are presented. The experimental fab results along with the proposed EDA flow simulations show promising results for the proposed M2A2 technology. Design tradeoffs and process challenges for large scale deployment of M2A2 technology are discussed along with their mitigation strategies.

Appendix A
Abbreviations

Various abbreviations used in this article are listed in Table IV.

ACKNOWLEDGMENT

The authors would like to thank the engineers of Advanced Node SoC Implementation team at Cadence Design Systems, Austin, Texas for their helpful discussions and suggestions to leverage commercial solutions in M2A2 EDA design flow. The authors would also like to thank the engineers of Molecular Imprints Inc. (MII) and Canon Nanotechnologies Inc. who have worked on developing the J-FIL technology, which is leveraged for M2A2 nano-precise pick-and-place assembly.

REFERENCES

[1] D. M. Tennant, "Limits of conventional lithography," *Nanotechnology*. New York, NY, USA: Springer, 1999, pp. 161–205.

[2] R. Mosher, *Structured ASIC Based SoC Design*, AMI Semiconductor, Dallas, TX, USA, Jan. 30, 2004. [Online]. Available: https://www.design-reuse.com/articles/search/?q=Structured+ASIC+Based+SoC+Design

[3] M. Santarini, *Are structured ASICs a dead end for EDA?* San Francisco, CA, USA: EE Times, 2003. [Online]. Available: https://www.eetimes.com/are-structured-asics-a-dead-end-for-eda/

[4] T. Westphalen, S. Hengesbach, C. Holly, M. Traub, and D. Hoffmann, "Automated alignment of fast- axis collimator lenses for high-power diode laser bars," *Proc. SPIE*, vol. 8965, Mar. 2014, Art. no. 89650V.

[5] V. Liimatainen, M. Kharboutly, D. Rostoucher, M. Gauthier, and Q. Zhou, "Capillary self-alignment assisted hybrid robotic handling for ultra-thin die stacking," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 1403–1408.

[6] J. N. Burghartz, W. Appel, C. Harendt, H. Rempp, H. Richter, and M. Zimmermann, "Ultra-thin chip technology and applications, a new paradigm in silicon technology," *Solid-State Electron.*, vol. 54, no. 9, pp. 818–829, 2010.

[7] S. V. Sreenivasan, "Nanoimprint lithography steppers for volume fabrication of leading-edge semiconductor ICs," *Microsyst. Nanoeng.*, vol. 3, Sep. 2017, Art. no. 17075.

[8] H. Takeishi *et al.*, "Nanoimprint system development and status for high volume semiconductor manufacturing," in *Proc. Altern. Lithograph. Technol. VII*, vol. 9423. San Jose, CA, USA, Mar. 2015, Art. no. 94230C. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/9423/94230C/Nanoimprint-system-development-and-status-for-high-volume-semiconductor-manufacturing/10?SSO=1

[9] H. J. Levinson, *Principles of Lithography*. Bellingham, WA, USA: SPIE, 2005.

[10] M. Hiura, T. Hayashi, A. Kimura, and Y. Suzaki, "Overlay improvements using a novel high-order distortion correction system for NIL high-volume manufacturing," in *Proc. Emerg. Pattern. Technol.*, vol. 10584. San Jose, CA, USA, 2018, Art. no. 105840U.

[11] Y. Takabayashi, T. Iwanaga, M. Hiura, H. Morohoshi, T. Hayashi, and T. Komaki, "Nanoimprint system alignment and overlay improvement for high volume semiconductor manufacturing," in *Proc. Novel Pattern. Technol. Semicond.*, vol. 10958. San Jose, CA, USA, 2019, Art. no. 109580B.

[12] L. Di Cioccio *et al.*, "An overview of patterned metal/dielectric surface bonding: Mechanism, alignment and characterization," *J. Electrochem. Soc.*, vol. 158, no. 6, pp. 81–86, 2011.

[13] C. Anshuman *et al.*, "Nanoscale magnification and shape control system for precision overlay in jet and flash imprint lithography," *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 1, pp. 122–132, Feb. 2015.

[14] P. Ajay, A. Cherala, B. A. Yin, E. E. Moon, R. F. Pease, and S. V. Sreenivasan, "Multifield sub-5 nm overlay in imprint lithography," *J. Vacuum Sci. Technol. B*, vol. 34, no. 6, 2016, Art. no. 061605.

[15] E. Moon, "Interferometric-spatial-phase imaging for sub-nanometer three-dimensional positioning," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci. MIT, Cambridge, MA, USA, 2004.

[16] Y. Beilliard *et al.*, "Chip to wafer copper direct bonding electrical characterization and thermal cycling," in *Proc. IEEE Int. 3D Syst. Integr. Conf. (3DIC)*, San Francisco, CA, USA, 2013, pp. 1–7.

[17] P. Gueguen *et al.*, "Copper direct-bonding characterization and its interests for 3D integration," *J. Electrochem. Soc.*, vol. 156, no. 10, pp. H772–H776, 2009.

[18] Y. Beilliard *et al.*, "Advances toward reliable high density Cu-Cu interconnects by Cu-SiO2 direct hybrid bonding," in *Proc. Int. 3D Syst. Integr. Conf. (3DIC)*, Kinsdale, Ireland, 852014, pp. 1–8.

[19] *RTL Compiler-Physical Application Note Product Version RTL Compiler 11.2*, Cadence, San Jose, CA, USA, Jun. 2012.

[20] R. A. Whitaker, "A fast algorithm for greedy interchange for large-scale clustering and median location problems," *Inf. Syst. Oper. Res.(INFOR)*, vol. 21, no. 2, pp. 95–108, 1983.

[21] K. Fukuda and T. Matsui, "Finding all minimum-Ř cost perfect matchings in bipartite graphs," *Networks*, vol. 22, pp. 461–468, Aug. 1992.

[22] P. S. Bradley and U. M. Fayyad, "Refining initial points for K-Means clustering," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 91–99.

[23] *Post-Mask ECO Using Conformal ECO Conformal-ECO (CECO) Version 15.10-p100*, Cadence, San Jose, CA, USA, Sep. 2015.

[24] J. Li, L. Behjat, and A. Kennings, "Net cluster: A net-reduction-based clustering preprocessing algorithm for partitioning and placement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 4, pp. 669–679, Apr. 2007.

[25] J. Li and L. Behjat, "A connectivity based clustering algorithm with application to VLSI circuit partitioning," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 5, pp. 384–388, May 2006.

[26] K. Blutman, H. Fatemi, A. Kapoor, A. B. Kahng, J. Li, and J. Pineda de Gyvez, "Logic design partitioning for stacked power domains," *IEEE Trans. Very Large Scale Integr. Syst. (VLSI)*, vol. 25, no. 11, pp. 3045–3056, Nov. 2017.

[27] C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. 19th Design Autom. Conf.*, Las Vegas, NV, USA, 1982, pp. 175–181.

[28] A. Rajaram and D. Z. Pan, "Robust chip-level clock tree synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 6, pp. 877–890, Jun. 2011.

[29] K. Ho, Y. Chen, J. Fang, and Y. Chang, "ECO timing optimization using spare cells and technology remapping," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 5, pp. 697–710, May 2010.

[30] *UNIBOND Wafers: Smart Cut Technology*. Accessed: April 8, 2020. [Online]. Available: https://www.semiconductoronline.com/doc/unibond-wafers-smart-cut-technology-0001

[31] *Nova Electronic Materials LLC*. Accessed: April 8, 2020. [Online]. Available: http://www.novawafers.com

[32] *Photolithography i-g Line Mask Aligner SussMicrotec—MA6/BA6*. Accessed: April 8, 2020. [Online]. Available: https://www.mrc.utexas.edu/facilities/equipment/photolithography-i-g-line-mask-aligner-sussmicrotec-ma6ba6

[33] *Etcher ICP Deep Silicon PlasmaTherm Versaline*. Accessed: April 8, 2020. [Online]. Available: https://www.mrc.utexas.edu/facilities/equipment/etcher-icp-deep-silicon-plasmatherm-versaline

[34] *Microelectronics Research Center (MRC)*. Accessed: April 8, 2020. [Online]. Available: https://www.mrc. utexas.edu

[35] *ANSYS Fluent-CFD Software*. Accessed: April 8, 2020. [Online]. Available: https://www.ansys.com/products/fluids/ansys-fluent

[36] R. C. Palharini and T. Scanlon, "Atmospheric reentry modelling using an open source DSMC code," Ph.D. dissertation, Dept. Mech. Aerosp. Eng., Univ. Strathclyde, Glasgow, U.K., 2014. [Online]. Available: https://pureportal.strath.ac.uk/en/publications/atmospheric-reentry-modelling-using-an-open-source-dsmc-code

[37] *The OpenFOAM Foundation*. Accessed: April 8, 2020. [Online]. Available: https://openfoam.org

[38] *Texas Advanced Computing Center (TACC)*. Accessed: April 8, 2020. [Online]. Available: https://www. tacc.utexas.edu

[39] (2005). *International Workshop on Logic Synthesis (IWLS) Benchmarks 2005*. [Online]. Available: http://www.iwls.org/iwls2005/benchmarks.html

[40] M. H. Hoe *et al.*, "Structured ASIC: Methodology and comparison," in *Proc. Int. Conf. Field Program. Technol.*, Beijing, China, 2010, pp. 377–380.

[41] M.-H. Hoe *et al.*, "Architecture and design flow for a highly efficient structured ASIC," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 3, pp. 424–433, Mar. 2013.

[42] M. Martins *et al.*, "Open cell library in 15nm FreePDK technology," in *Proc. Int. Symp. Phys. Design (ISPD)*, 2015, pp. 171–178.

[43] C.-H. Jan *et al.*, "A 14 nm SoC platform technology featuring 2nd generation Tri-Gate transistors, 70 nm gate pitch, 52 nm metal pitch, and 0.0499 um2 SRAM cells, optimized for low power, high performance and high density SoC products," in *Proc. Symp. VLSI Circuits (VLSI Circuits)*, Kyoto, Japan, 2015, pp. T12–T13.

[44] *Virtex-II Platform FPGAs: Complete Data Sheet Product Specification DS031 (v4.0)*, Xilinx, San Jose, CA, Xilinx, Apr. 2014.

[45] *Virtex-6 Family Overview Product Specification DS150 (v2.5)*, Xilinx, San Jose, CA, USA, Aug. 2015.

[46] *UltraScale Architecture and Product Data Sheet: Overview Product Specification DS890 (v3.7)*, Xilinx, San Jose, CA, USA, Feb. 2019.
[47] *Innovus User Guide Product Version 16.22*, Cadence, San Jose, CA, USA, Mar. 2017.
[48] *PrimeTime: Golden Timing Signoff Solution and Environment Datasheet*, Synopsys, Mountain View, CA, USA, 2017.
[49] *Vivado Design Suite Properties Reference Guide Product Specification UG912 (v2016.3)*, Xilinx, San Jose, CA, USA, Dec. 2016.
[50] *ISE Design Suite Software Manuals Product Specification UG681 (v 11.4)*, Xilinx, San Jose, CA, USA, Dec. 2009.
[51] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 203–215, Feb. 2007.

Aseem Sayal (Student Member, IEEE) received the bachelor's degree in electrical and electronics engineering from Delhi Technological University (formerly, Delhi College of Engineering), New Delhi, India, in 2013, and the M.S. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering.

From 2013 to 2015, he worked as a Physical Design CAD Engineer with Qualcomm India Pvt. Ltd., Bengaluru, India. He was also an Intern with Apple Inc., Cupertino, CA, USA, and Google LLC, Sunnyvale, CA, USA, where he was involved in developing clock tree simulation methodology and average power estimation solutions for long running arbitrary workloads, respectively. His research is focused on designing energy-efficient hardware accelerators for machine learning applications, and developing EDA design methodologies for heterogeneously integrated secured ASICs.

Mr. Sayal was a recipient of the Chancellor Gold Medal and Meritorious Student Award from Delhi Technological University in 2013.

Paras Ajay received the Bachelor of Technology (B.Tech.) degree in mechanical engineering from the Indian Institute of Technology, New Delhi, India, in 2012, and the Ph.D. degree in mechanical engineering from the University of Texas at Austin, Austin, TX, USA, in 2019.

He is currently a Postdoctoral Researcher with the NASCENT Center, University of Texas at Austin. He has published one journal paper on overlay control in nano-imprint lithography, one book chapter on overlay control in advanced lithography, one conference paper which was awarded the best poster award at the 30th American Society for Precision Engineering Conference in 2015. His graduate research has also resulted in five patent applications primarily in the area of nano-precise overlay and alignment control, two of which have been granted, and one licensed by Canon Nanotechnologies, Inc. His research interests lie in the design, fabrication, integration and control of nanofabrication systems, specifically in the subdomains of nano-precise motion control, thermo-mechanical distortion control, thermal management and metrology for advanced nanopatterning and nanofabrication applications.

Mark W. McDermott (Life Member, IEEE) received the B.S. degree in electrical engineering from the University of New Mexico, Albuquerque, NM, USA, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Austin, Austin, TX, USA, in 1988 and 2014, respectively.

He has 40 years of industry experience in the product development of silicon systems. This includes the Senior Director with Apple Inc., Austin, the Vice President/General Manager with Intrinsity, Inc., Austin, and the Director/ General Manager with Intel Corporation, Austin. He is currently a Professor of practice with the Electrical and Computer Engineering Department, University of Texas at Austin, and a Technical Advisor with Insilxa Inc., Sunnyvale, CA, USA. He holds 19 patents and has 15 publications in the areas of VLSI design, engineering education, and engineering management.

Prof. McDermott is a member of the Association for Computing Machinery and the Texas Society of Professional Engineers.

S. V. Sreenivasan (Member, IEEE) received the B.Tech. degree in mechanical engineering from the National Institute of Technology Tiruchirappalli, Tiruchirappalli, India, in 1987, and the Ph.D. degree in mechanical engineering from Ohio State University, Columbus, OH, USA, in 1994.

He is the Joe C. Walter Endowed Chair of engineering, a Professor of mechanical engineering, and a Professor of electrical and computer engineering with the University of Texas at Austin (UT-Austin), Austin, TX, USA. He is a Nanotechnologist with an interest in creating high throughput nanofabrication systems that enable applications in electronics and healthcare sectors. He has published over 130 technical articles and holds over 100 U.S. patents in the area of nanoscale manufacturing. He is the Director of the NASCENT Center, a National Science Foundation funded Nanosystems Engineering Research Center in the area of nanoscale manufacturing. NASCENT is composed of an interdisciplinary team of 100+ students, researchers, and professors. He co-founded Molecular Imprints Inc. (MII), Austin, a nanotech spin out from UT-Austin. He currently serves as the Chief Technologist of Canon Nanotechnologies, Inc., Austin, a company formed as a result of the acquisition of the semiconductor business of MII by Canon Corporation in 2014. Additionally, the display division of MII was acquired in 2015 by Magic Leap, Inc., Plantation, FL, USA, a leader in augmented/mixed reality displays.

Dr. Sreenivasan has received several awards for his work, including the Technology Pioneer Award by the World Economic Forum in 2005, the University of Texas Chancellors' Award for Entrepreneurship in 2007, the ASME Leonardo da Vinci Award in 2009, the TAMEST O'Donnell Award for Technology Innovation in 2010, the ASME William T. Ennor Manufacturing Technology Award in 2011, the UT-Austin Inventor of the Year Award in 2012, and the ASME Machine Design Award in 2017. He was named as a fellow of the National Academy of Inventors in April 2017.

Jaydeep P. Kulkarni (Senior Member, IEEE) received the B.E. degree in electronics engineering from the University of Pune, Pune, India, in 2002, the M.Tech. degree in electronics engineering from the Indian Institute of Science (IISc), Bengaluru, India, in 2004, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2009.

From 2009 to 2017, he was with Intel Circuit Research Lab, Hillsboro, OR, USA, where he worked on energy-efficient integrated circuit technologies. He is currently an Assistant Professor of electrical and computer engineering with the University of Texas at Austin, Austin, TX, USA, and currently holds the AMD endowed chair position in computer Engineering. He has filed 35 patents and published 75 papers in referred journals and conferences. His research is focused on machine learning hardware accelerators, in-memory computing, emerging nano-devices, hardware security, and design methodologies for heterogenous integration.

Dr. Kulkarni received the Best M.Tech. Student Award from IISc, the Intel foundation Ph.D. Fellowship Award, the SRC Best Paper and Inventor Recognition Awards, the Purdue Outstanding Doctoral Dissertation Award, seven Intel Divisional Recognition Awards, the 2015 IEEE Transactions on VLSI Systems Best Paper Award, the SRC Outstanding Industrial Liaison Award, and Micron Faculty Awards. He is also serving as an Associate Editor for IEEE SOLID STATE CIRCUIT LETTERS and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He has served as the Conference General Co-Chair for 2018 ISLPED, and is currently participating in the technical program committees of CICC, ICCAD, and AICAS conferences. He is currently serving as the Chair of IEEE Central Texas SSCS/CAS joint chapter. He is a member of ACM.