Cell Transformations and Physical Design Techniques for 3D Monolithic Integrated Circuits

SHASHIKANTH BOBBA, École Polytechnique Fédérale de Lausanne (EPFL)
ASHUTOSH CHAKRABORTY, Oracle Microelectronics
OLIVIER THOMAS and PERRINE BATUDE, CEA-LETI
GIOVANNI DE MICHELI, École Polytechnique Fédérale de Lausanne (EPFL)

3D Monolithic Integration (3DMI), also termed as sequential integration, is a potential technology for future gigascale circuits. In 3DMI technology the 3D contacts, connecting different active layers, are in the order of few 100nm. Given the advantage of such small contacts, 3DMI enables fine-grain (gate-level) partitioning of circuits. In this work we present three cell transformation techniques for standard cell-based ICs with 3DMI technology. As a major contribution of this work, we propose a design flow comprising of a cell transformation technique, *cell-on-cell stacking*, and a physical design technique (CELONCEL$_{\mathrm{PD}}$) aimed at placing cells transformed with *cell-on-cell* stacking. We analyze and compare various cell transformation techniques for 3DMI technology without disrupting the regularity of the IC design flow. Our experiments demonstrate the effectiveness of CELONCEL design technique, yielding us an area reduction of 37.5%, 16.2% average reduction in wirelength, and 6.2% average improvement in overall delay, compared with a 2D case when benchmarked across various designs in 45nm technology node.

Categories and Subject Descriptors: B.7.2 [**Design Aids**]: Layout

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: 3D integration, 3D monolithic, cell transformation techniques, layout

ACM Reference Format:
Bobba, S., Chakraborty, A., Thomas, O., Batude, P., and De Micheli, G. 2013. Cell transformations and physical design techniques for 3D monolithic integrated circuits. ACM J. Emerg. Technol. Comput. Syst. 9, 3, Article 19 (September 2013), 28 pages.
DOI: http://dx.doi.org/10.1145/2491675

1. INTRODUCTION

3D integration provides an effective platform for realizing future gigascale circuits by integrating multiple layers of active devices vertically [Saraswat 2010; Pavlidis and Friedman 2009]. 3D fabrication technologies can be broadly classified into two groups according to the used integration scheme: (a) 3D parallel integration (or *Through-Silicon Via,* TSV, -based technology) in which each active layer, along with its respective interconnect metal layers, is fabricated separately and is subsequently stacked via TSVs [Koester et al. 2008; Sillon et al. 2008], and (b) 3D Monolithic Integration (3DMI),

This work is primarily funded by the European grant: ERC-2009-AdG-246810, and partly supported by the ST-IBM-LETI alliance program.

Authors' addresses: S. Bobba (corresponding author), Integrated Systems Laboratory (LSI), Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne 1015, Switzerland; email: Shashikanth.bobba@epfl.ch; A. Chakraborty, Oracle Microelectronics; O. Thomas and P. Batude, CEA-LETI, Grenoble, 38054 France; G. De Micheli, Integrated Systems Laboratory (LSI), Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne 1015, Switzerland.

DOI: http://dx.doi.org/10.1145/2491675

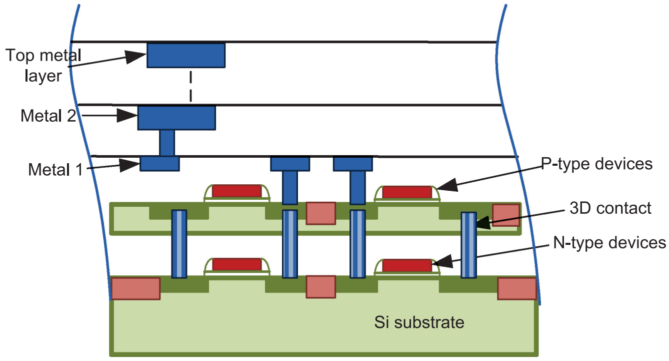


Fig. 1. Cross-section of a 3D monolithic die with two active layers.

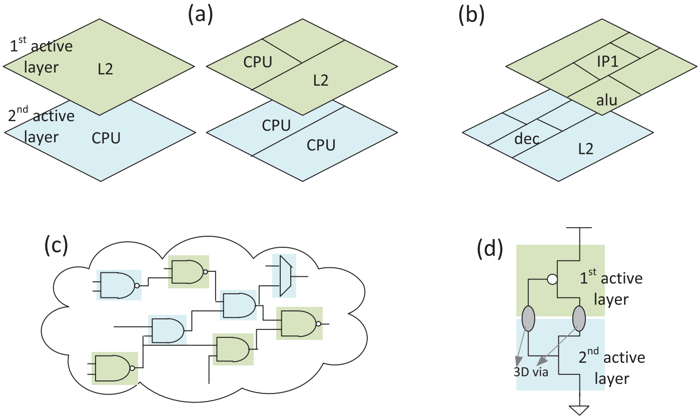


Fig. 2. Coarse-grain to fine-grain circuit partitioning for 3D circuits [Loh et al. 07] (a) memory/core on a core; (b) functional unit blocks on top of each other; (c) logic gates distributed across different layers; and (d) transistor-scale partitioning.

in which the stacked active layers are processed sequentially on the same wafer from bottom to top layer. Figure 1 shows the cross-section of a wafer manufactured by 3D monolithic process having N-type (NMOS) devices in the bottom active layer and P-type (PMOS) devices on the top active layer [Batude et al. 2009b]. The two active layers are connected using a 3D contact which is similar to the conventional inter-layer vias. With the latest advancement in 3DMI technology (low-temperature top FETs, intermediate metal layer between the active layers, and high-quality bonding) [Batude et al. 2009a], we can build 3DMI circuits in the near future.

The performance of ICs in advanced technology nodes is dominated by the interconnect delay [Havemann and Hutchby 2001]. Migrating to 3D ICs, we can envisage reduced interconnect delay and chip area which is achieved by placing the logic gates, on the critical path, very close to each other using multiple active layers. Loh et al. have shown the benefits of 3D ICs in terms of wirelength, latency, and power depending on the granularity level at which various processing elements are partitioned across multiple active layers [Loh et al. 2007]. Figure 2 illustrates the circuit partitioning of a processor at various granularities. For example, at a coarse-grain level, we can have cache on top of cores, or cores on top of cores, as presented in Figure 2(a). At a finer level of granularity we can realize functional blocks on top of each other (Figure 2(b). Going at even finer level, we can perform 3D stacking at the gate and standard cell level, as illustrated in Figure 2(c) and (d). Care should be taken while realizing fine-grain partitioning for routing intense designs, as the routing complexity is further increased.

In the case of TSV technology, due to low precision of the alignment capability of the equipment and the relatively large size of TSVs, ~1000nm [Tezzaron], circuit integration at the transistor/gate level cannot be done. Kim et al. have shown that the gate-level integration does not gain in wirelength reduction with the current TSV sizes [Kim et al. 2009]. Consequently, 3DMI is an ideal choice for ultra-high-density 3D circuits. In this work, we analyze all the possible cell transformation techniques for realizing fine stacking at the transistor/gate level. One of the simplest techniques is *intra-cell* transformation, where standard cells can be partitioned across multiple layers (Figure 2(d), in a 3D assembly and employ existing physical synthesis tools to complete the IC design flow [Ieong et al. 2003; Bobba et al. 2010]. However, with this technique, gates on the critical path cannot be placed close to each other in the third dimension. In this work we propose a novel *cell-on-cell* transformation technique, where the cells are placed on top of each other (Figure 2(c)). At this level, redesign effort is high and hence a new physical synthesis tool, CELONCEL$_{\text{PD}}$, for logic-to-layout synthesis is presented in this article.

In recent years, there has been extensive work in developing new physical design tools for 3D IC design [Das et al. 2003; Cong et al. 2007; Zhou et al. 2006]. However, all these tools are mainly linked to the 3D TSV technology, as the main objective is to minimize the number of TSVs while reducing the wirelength of the routed circuit. 3D monolithic integration has seen substantially less research effort at the CAD level. Many authors have solved the placement problem for 3DMI by reducing the weight of the TSV in their wirelength optimization formulation [Deng and Maly 2001]. However, practical details of the technology are not considered. For instance, in 3DMI technology the intermediate metal layer between the active layers is tungsten, as it has a high thermal coefficient when compared to copper. However, tungsten is three times more resistive than copper, thereby making it more suitable for local routing (for example, routing within the standard cell) than for intermediate routing. With this work we take the first step towards providing a complete design flow for 3D monolithic technology. CELONCEL design flow, comprised of CELONCEL$_{\text{PD}}$ and CELONCEL$_{\text{LIB}}$, can be integrated into the traditional 2D design flow. To the best of our knowledge we are the first to address the cell and physical design issues at a fine granularity for application-specific integrated circuit design for 3DMI technology.

To summarize, the main contributions of this article are as follows.

(1) We present three cell transformation techniques, *intra-cell stacking*, *cell-on-cell stacking*, and *intra-cell folding*. For practical reasons we consider 3DMI technology with only two active layers separated by an intermediate metal layer. A 2D standard cell library at 45nm is mapped to 3DMI technology with the various cell transformation techniques.
(2) All the three cell transformation techniques are analyzed to study the performance gains coming from each different technique. The regularity of the standard cell design flow is ensured for all the transformations.
(3) In order to study the benefits of cell-on-cell stacking, we present a new 3D physical design tool, CELONCEL$_{\text{PD}}$, which places cells in two active layers for improved area, wirelength, and delay. CELONCEL$_{\text{PD}}$ is a pre-/postprocessor for existing 2D placement engines which focuses on partitioning across two active layers and the detailed placement for each active layer. The runtime of the CELONCEL$_{\text{PD}}$ is improved by clustering of cells within a standard cell row.
(4) The ability of CELONCEL design flow (comprising a standard cell library CELONCEL$_{\text{LIB}}$ and CELONCEL$_{\text{PD}}$) is demonstrated through the 3D physical

synthesis flow on a set of open-source benchmarks [Opencores 2013] as well as several large (2M gates) synthetic benchmarks. We also show how CELONCEL fits into a conventional 2D tool-chain while building the cores in 3D.

The preliminary work presented in Bobba et al. [2011] is here extended to include an extended set of transformations from 2D to 3D mapping of standard cells for 3DMI technology. In this work, in addition to Bobba et al. [2011], we also consider a novel intra-cell folding transformation. We also provide a detailed explanation of the $\text{CELONCEL}_{\text{PD}}$. The new version of the $\text{CELONCEL}_{\text{PD}}$ is further optimized by efficiently clustering the cells in each row and solving them independently. Bigger benchmarks with over 2M gates are considered for the runtime analysis of the $\text{CELONCEL}_{\text{PD}}$.

The remainder of this article is organized as follows. In Section 2, we give necessary technology background and survey the state-of-the-art in circuit design for 3DMI technology. In Section 3, we explain the various 2D to 3D standard cell transformations specific to 3D monolithic integration. Our design flow is presented in Section 4. Section 5 deals with the description of our new 3D placement tool for *cell-on-cell stacking* transformation. The experimental setup is detailed in Section 6. Experimental results are presented in Section 7 to demonstrate the effectiveness of our proposed placement tool. In Section 8, we address crucial technology and design challenges for 3DMI circuits. Section 9 concludes the article by shedding some light on future work for further assessment of design methodologies for 3DMI circuits.

2. TECHNOLOGY BACKGROUND AND PREVIOUS WORK

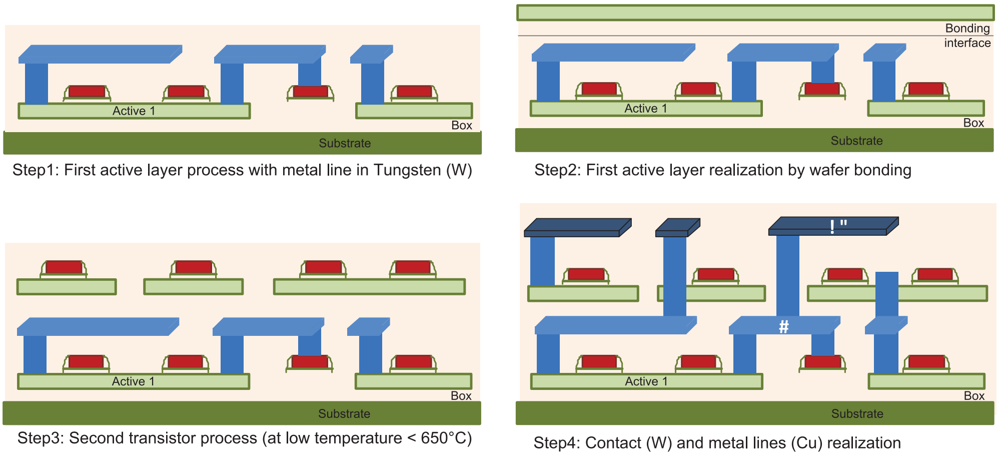
This section surveys previous work related to technology and design of 3D monolithic integrated circuits, which illustrates the 3DMI technology underlying our proposed design methodology. It also summarizes previous approaches to leverage 3DMI technology for high-density SRAM cell and FPGA design.

2.1. 3D Monolithic Technology

In 3D monolithic integration, multiple active layers are processed sequentially on the same die. As the alignment of top transistor lithography levels occurs after the bonding of a new top active layer, its precision is limited only by the performance of the stepper (for example, $3 = 10$nm for the 45nm node equipment [ITRS 2013]). Currently, 3D contact dimensions of $\sim$100nm have been demonstrated [Jung et al. 2007]. On the other hand, in 3D parallel integration (or 3D-TSV integration), two wafers (or dies) are stacked after they are individually processed, thereby leading to poor alignment precision. For example, at 45nm the 3σ of the stepper is 1μm [MIT 2013], which is one order of magnitude higher than 3DMI technology.

Figure 3 describes a general monolithic integration process flow. In step 1, the bottom transistor is fabricated with metal interconnections in tungsten. In step 2, the top active layer is obtained thanks to the use of wafer bonding. In step 3, the top transistor is fabricated with a limited thermal budget of 650°C. In step 4, the contact (W) and metal lines (Cu) are fabricated. The main challenge of this integration lies in the fabrication of a high-quality top FET with a low thermal budget in order to preserve bottom FET and metal interconnections from any degradation.

The minimal *Thermal Budget* (TB) to obtain a top FET with equivalent performance to a standard FET is around 600°C. This minimum TB value is dictated by the dopant activation step [Batude et al. 2008a]. Currently, *Solid Phase Epitaxy* (SPE) at 600°C has been demonstrated to be a viable option for dopant activation [Batude et al. 2009a]. The bottom transistor can endure such low thermal budget (600°C) at the exception of classical NiSi silicide that needs to be stabilized [Batude et al. 2008a]. However, classical copper lines cannot sustain such thermal budget and need to be replaced by a more



Step1: First active layer process with metal line in Tungsten (W)

Step2: First active layer realization by wafer bonding

Step3: Second transistor process (at low temperature < 650°C)

Step4: Contact (W) and metal lines (Cu) realization

Fig. 3. Process steps for the fabrication of 3DMI technology with two active layers seperated by an intermediate metal layer.

stable metal. Tungsten appears to be a good choice for its microelectronic compatibility and its thermal stability [Kim et al. 2002]. Though tungsten is thermally stable at high temperatures when compared to copper, it is three times more resistive than copper. In the current generation of integrated circuits, interconnect delays are attributable to a dominant part of the overall circuit delay. Hence the number of intermediate metal lines (tungsten metal lines in-between the active layers) should be low such that the circuit delay does not considerably increase. In this work we consider only one *Intermediate Metal* (IM) layer. The IM layer is employed for realizing intra-cell routing of the bottom standard cells. From our analysis, we see that employing tungsten instead of copper for intra-cell routing has negligible impact on the delay of the cell. This can be explained by the high resistance contribution from the transistors, contact resistance, and the high resistance of the active region, when compared to the resistance from the short interconnects within the cell.

3DMI technology described earlier is not limited to only two active layers. Devices and circuits have been fabricated and tested successfully [Batude et al. 2011]. Due to the novelty of the approach, and due to the advantage with respect to TSV technology, the first set of experiments has been done with two active layers. Multiple active layers will be effective only if the number of intermediate metal layers between the active layers is increased. With the current 3DMI technology, intermediate metal (tungsten) is more resistive than conventional metal layers (copper). Hence, we took efforts to minimize the usage of intermediate metal layers by limiting to only one metal layer between the active layers.

2.2. Previous Work in 3D Monolithic Integrated Circuits

3D monolithic integration is less known when compared to 3D-TSV technology. The latest advancement in 3DMI technology, with active layers being processed at low temperatures, has created substantial interest among various researchers [Batude et al. 2009a]. 3DMI technology promises very small 3D contacts in the order of a few 100nm [Jung et al. 2007], thereby enabling circuit partitioning at a fine granularity. Few publications proposed 3D FPGA architecture employing 3DMI technology [Wong et al. 2007; Lin et al. 2007]. The authors proposed to partition the blocks with the SRAM cells (used for configuration) in one layer and the logic part on the other layer. Batude et al. have proposed a compact and robust 4T SRAM bit cell which leverages the dynamic coupling offered by 3DMI technology with a thin inter-layer dielectric [Batude et al. 2008b]. Most advanced demonstrations were shown by Samsung through the use of their *Single-crystal Si layer Stacking* (S^3) technology [Jung et al. 2004]. They have demonstrated Flash memories and SRAM stacked up to three layers [Jung et al. 2007]. It has to be noted that the stability of the 6-T SRAM cell is sustained from the mechanical stress asserted by the 3D contacts. Hence for ASIC design, we assume that mechanical stress caused by 3D contacts will not have a huge impact on the behavior of the random logic circuits.

In this work we bring 3DMI technology to ASIC design. Additional technology features add to the design complexity, hence we need new CAD tools, especially physical synthesis tools, to bridge the time gap for designers. Previous research on 3D physical design adopts existing 2D placement tools for placing cells across multiple active layers [Deng and Maly 2001; Roy et al. 2005; Chan et al. 2006; Cong et al. 2007]. However, researchers have mainly focused on placement for 3D-TSV technology with an objective of reducing the estimated wirelength of the placed netlist with an additional constraint on minimizing the number of TSVs. In the work by Deng et al., the authors have adopted CAPO [Roy et al. 2005] to partition the circuit across multiple layers [Deng et al. 2001]. By reducing the weight of the TSVs in their problem formulation the authors briefly cover the placement problem for 3DMI technology. Our work differentiates from the existing work in many ways. First, the design technique proposed is closely linked to the current technology. Second, the $\text{CELONCEL}_{\text{PD}}$ presented in this work does not modify the 2D placement engine; however, it acts as a wrapper around the 2D placement engine to place standard cells in 3D. For instance, the state-of-the-art physical design tools have been developed and tuned over a decade [Chan et al. 2006; Jiang et al. 2006; Roy et al. 2005; Encounter 2013; IC Compiler] and separate customization of the tools for different technologies can be very expensive, if not adapted carefully. Compared to academic placers, industrial placers (e.g., Cadence Encounter, IC Compiler, etc.) offer complete physical synthesis flow (with steps such as buffer insertion, gate sizing, fanout optimization, repeater insertion, etc.) for advanced timing closure. Hence, in our work we build $\text{CELONCEL}_{\text{PD}}$ as a wrapper around the industrial placement engine [Encounter 2013] to study timing benefits of various cell transformation techniques related to 3DMI technology. However, our placement technique is also compatible with other existing 2D placement engines.

3. CELL TRANSFORMATIONS AND LIBRARY DESIGN

In this section, we discuss three methods to modify a traditional 2D standard cell library for 3D monolithic integrated circuits. Using the first method (*intra-cell stacking* transformation), a 2D cell is mapped into a 3D cell by realizing the *Pull-Up Network* (PUN) of the cell on the top active layer and *Pull-Down Network* (PDN) on the bottom active layer [Ieong et al. 2003; Batude et al. 2009b; Bobba et al. 2010]. In the second method (*cell-on-cell stacking* transformation), 2D cells can be placed on top of each other without any pin conflicts. In the third method (*intra-cell folding* transformation), a 2D cell is mapped into a 3D cell by folding the width across multiple active layers. Throughout this article we call cell width and height the dimension on x-and y-axes, respectively. In this work, we abide to 3DMI technology with two active layers separated by an intermediate metal layer. For the technology backend (metal lines) we considered design rules of a 45nm process node, where only one metal layer is employed for intra-cell routing [Nangate 2013]. However, for the future technology nodes with 1D routing (i.e., with lower metal layers having either vertical or horizontal orientation) we require at least two metal layers for intra-cell routing. The cell transformation techniques presented in this work can be easily extended to the future nodes by considering two intermediate metal layers for designing the bottom cells and two metal layers for the top cells.

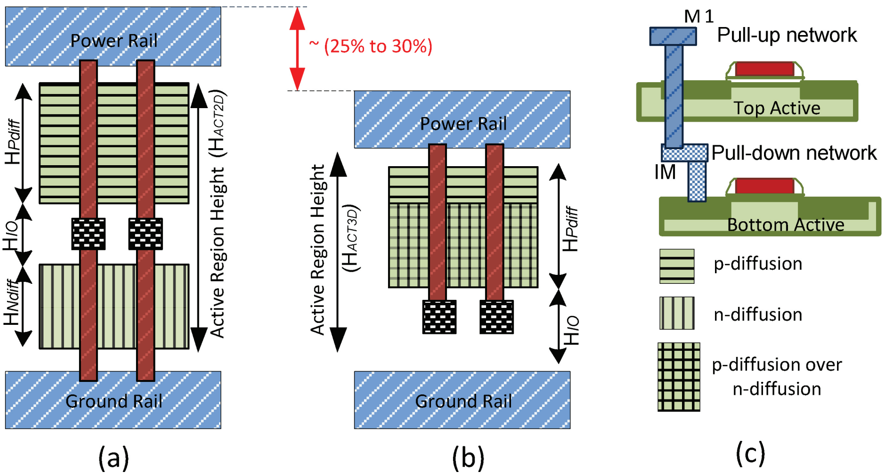


Fig. 4. (a) Typical cell in 2D configuration; (b) intra-cell transformation, in two active layers, by realizing pull-up network on the top layer and pull-down network at the bottom layer; (c) cross-sectional view of the two active layers with the metals (IM and M1) for realizing PUN and PDN of the cell.

3.1. Intra-Cell Stacking Transformation

Standard cells implement predefined logic functions (for example, NAND gates, NOR gates, and flip-flops) and have fixed height but varying widths. The structure of a typical 2D standard cell layout is shown in Figure 4(a). The power and ground rails are located at the top and bottom end of the cell. Active region height (H_{ACT}) of the cell is where the transistors are fabricated. The distance between two diffusion regions is called the diffusion gap region, where the input pins are placed. Since 3DMI technology offers multiple active layers adjacent to each other, the layout of the standard cell can be folded in multiple layers [Ieong et al. 2003; Batude et al. 2009b]. For instance, as illustrated in Figure 4(b), *p-type* devices are realized on the top active layer and *n-type* devices on the bottom active layer. Since the *p-diffusion* is typically wider than the *n-diffusion*, the active region height for a 3D cell (H_{ACT3D}) is limited by the height of the *p-diffusion* (*HPdiff*). The active region height of a 3D library is given by the following equation, when mapped directly from a 2D library.

$$H_{ACT3D} = H_{ACT2D} - H_{Ndiff} = H_{IO} + H_{Pdiff}$$

In the preceding transformation, the reduction in height of a 3D cell is due to the *n-diffusion* region. Moreover, there can be a slight increase in the space needed for *Input-Output* (IO) pins in the 3D layout, as the design rules should be followed, considering the close proximity of wide power rails. The active region (in green) with horizontal stripes represents a p-active region, whereas the green vertical stripes represent the n-active region. The overlap between these two active regions, realized in two different layers, has a gridded pattern.

3.2. Cell-on-Cell Stacking Transformation

To achieve truly stacked cells, we propose the method of *cell-on-cell stacking*. In *cell-on-cell* stacking, instead of distributing the diffusion regions of the cell in two active layers, the cells are realized with one active layer and one metal layer, but such cells can be placed on top of each other. One of the main challenges for this approach is to access the IO pins of the bottom cell from the top metal layers (for instance, metal 2) without short-circuiting the IO pins of the cell placed on the top active layer. Though in many cases it may be possible to shift the cell in the top active layer laterally to

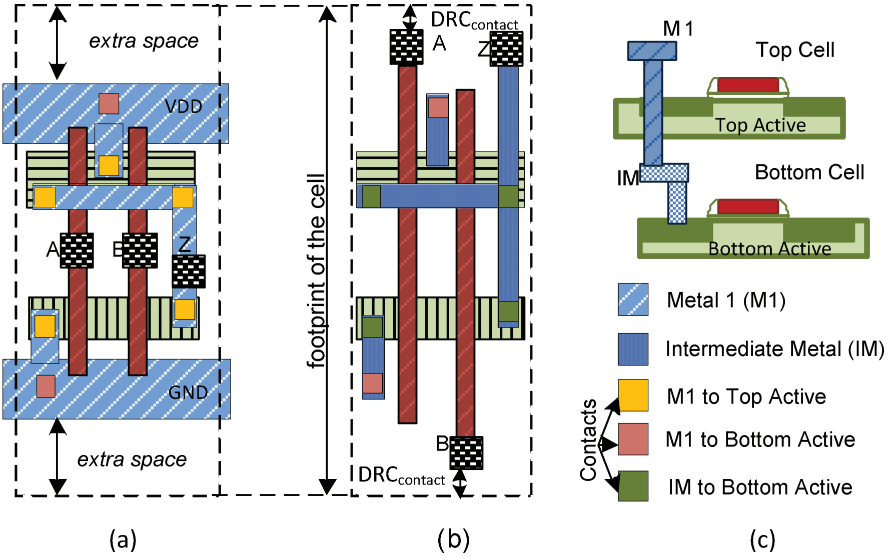


Fig. 5. CELONCEL NAND2 layout: (a) cell realized in the top active layer; (b) corresponding cell in the bottom active layer; and (c) cross-sectional view of the two active layers with the metals (IM and M1) to realize the cells.

access pins of the cell lying at the bottom layer, this technique is not generic since many conflicting cell pairs could exist for which there is no way to access the pins of both the cells. Figure 5 illustrates an example of *cell-on-cell* stacking of two cells on top of each other. Figure 5(a) and Figure 5(b) show a 2-input NAND gate, realized in the top active layer and the bottom active layer such that pin access can be maintained. The *Intra-Cell Routing* (ICR) of the bottom cell is realized with the intermediate metal layer in between the active layers. Tungsten is used for ICR of the bottom cell, whereas copper is used for the top cell. We did not observe any considerable delay degradation of the bottom cell when compared to the similar cell realized in the top layer.

In order to access the IO pins of the bottom cell to the top metal layers, extra space is allocated in the top active layer. For instance, the IO pins of the top cell are placed in between the power and ground rails (VDD and GND rails in Figure 5). By contrast, the IO pins of the bottom cell are placed beyond the rails. Hence the cell height (or footprint) has to consider the additional space for the IO pins coming from the bottom cell and also the respective design rule for avoiding conflicts (DRC$_{contact}$) with the IO pins of the neighboring cell. This leads to an increase in the standard cell height.

3.3. Intra-Cell Folding Transformation

In the previous transformations we have seen that the height of the standard cell was altered in both the cases. However, we can also envisage a 3D cell built across multiple active layers by folding the width of the standard cell. Unlike realizing the *p-type* and *n-type* devices in two different layers, in this transformation we fold the gates and fingers of the cells across two different layers. For example, consider a two-input (inputs A, B) NAND gate. The width of the cell can be folded by realizing the gate A in the bottom layer and gate B in the top layer. The benefits of this transformation can be maximized when applied to cells with high driving strength, where the large transistors are implemented by multiple fingers.

Figure 6(a) shows a 2-input NAND with high driving strength. With intra-cell folding transformation, we realize the fingers in the top active layer thereby resulting in a

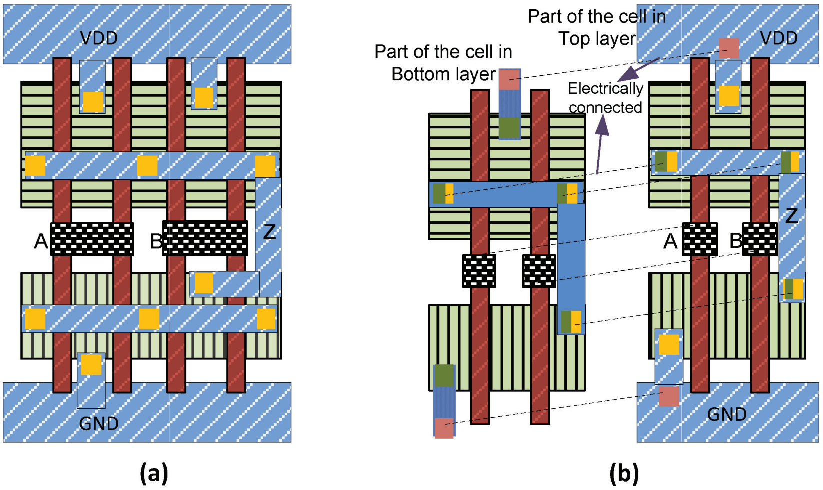

VDD
A
B
Z
GND
(a)
Part of the cell in Top layer
Part of the cell in Bottom layer
Electrically connected
VDD
A
B
Z
GND
(b)

Fig. 6. (a) Two-input NAND cell with a high drive strength having finger transistors; (b) corresponding cell built in 3D with intra-cell folding transformation where the fingers are realized in the bottom/top active layer.

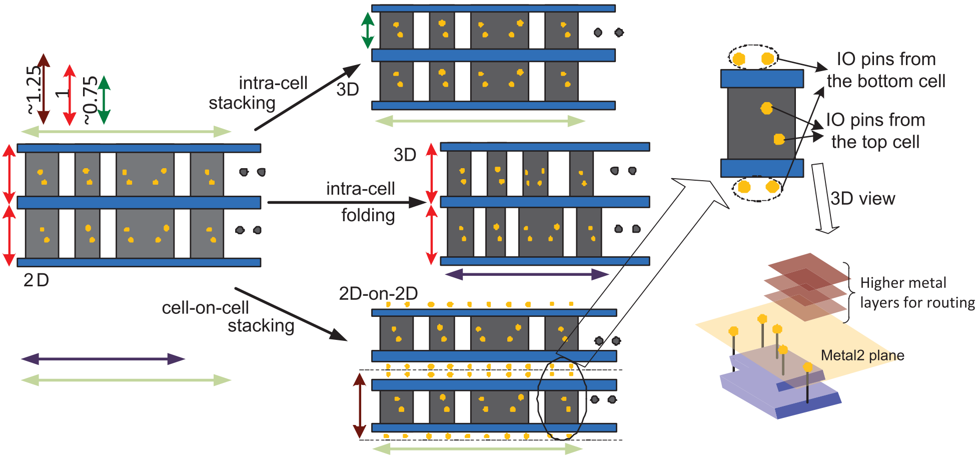


Fig. 7. 2D to 3D cell transformation.

compact cell as shown in Figure 6(b). The layout to the left of Figure 6(b) is a part of the NAND gate placed at the bottom layer. The dotted line represents the electrical connections between the top cell part and the bottom cell part. Both of the parts are connected to form a 3D cell. Unlike the intra-cell stacking transformation, where the gain in height is constant throughout the library, the gain in width with intra-cell folding transformation varies depending on the type of the cell, number of inputs (*fanin*) to the cell, and the driving strength (*fanout*) of the cell.

3.4. Quantifying 2D-to-3D Library Mapping

Until now we have explained various cell transformation techniques. In this section, we focus on the implementation details of these transformations. Figure 7 shows the three approaches to realize a 3D standard cell library from a 2D library. Corresponding

Table I. Normalized Height of Existing Standard Cell Libraries Before and After Cell Transformtation

Cell Height	45nm Nangate library	45nm commercial library	65nm commercial library
Planar (2D)	100 %	100 %	100 %
Intra-cell stacking (3D)	71.43 %	71.61 %	69.05 %
Cell-on-cell (2D-on-2D)	125.71 %	125.93 %	125.0%
Intra-cell folding (3D)	100 %	100 %	100 %

Table II. Percentage Improvement in Width of the Standard Cells Before and After the Folding Transformtation

	Width gain (%)	
Standard Cells	*Low drive* (2*X*, 4*X*)	*High drive* (≥4*X*)
inv	0	33%
nand2/nor2	33%	40.62%
nand3/nor3	25%	42.86%
and3/or3	40%	40%
aoi21/oai21	25%	42.86%
aoi22/oai22	40%	44.44%
SD-flipflop	50%	50%

vertical arrows in green, red, and brown quantify the average standard cell height in all cases with respect to the 2D implementation.

Table I compares the standard cell height of existing 2D standard cell libraries before and after the cell transformation. We have benchmarked across three important cell libraries at 45nm and 65nm technology node. Intra-cell folding transformation does not have any impact on the height of the standard cell, however, only affects the width of the cells. Table II presents the percentage improvement in width of the folded cells when compared to the 2D cells, while mapping the 45nm Nangate 2D cell library [Nangate 2013].

There are a few key observations from 2D to 3D cell transformation.

(1) By intra-cell stacking, all the cells are spread across two active layers, thereby making a 3D cell library. On average, we observe 29% reduction in the standard cell height with intra-cell stacking. The height of the standard cell is directly related to the footprint of the circuit. Hence a ~30% reduction of the cell height leads to almost 30% reduction in the overall area. However, in current technologies the performance of a circuit is more important than the area. With 3D cells, we envisage significant decrease in the interconnect length as an outcome of the reduced footprint. One of the primary advantages of this transformation is the ease of integration with existing design flows, since the only design effort required is building the 3D cell library. The CAD part for realizing the logic-to-layout (RTL-to-GDSII) flow does not need any alteration, as the physical design tool when solving the placement problem models the cells as rectangular boxes with the IO pins located at the center of the box. In other words, the placement tool does not differentiate a 2D cell from a 3D cell.

(2) Cell-on-cell stacking leads to 25% increase in the cell height. However, in this case all the cells occupy one active layer and, therefore, one cell can be placed on top of the other. Hence, with 25% increase in the footprint of the cell, we can accommodate 2× the number of cells in the two active layers. Moreover, the number of the neighboring cells is doubled as compared to the 2D or intra-cell implementations. Figure 8 shows the cells with their immediate and next neighboring cells for all the

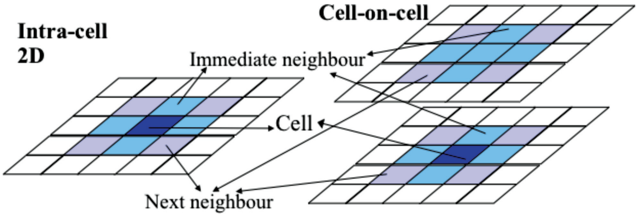


Fig. 8. Neighboring cells in the case of planar, intra-cell, and cell-on-cell.

cases. The design effort for cell-on-cell stacking is higher as the number of cells is doubled, for the top and bottom active layers. Moreover, a new physical design tool is needed to place the cells in multiple active layers.

(3) With intra-cell folding, the cells are built in a 3D manner by folding the width of the cell while keeping the height constant. From Table II we can see that the gain in width depends on the type of cell, fanin, and fanout of the cell. Consequently this transformation cannot be justified for small cells (e.g., inv, nand2, nor2, etc.). However, maximum area gain can be achieved for complex gates (e.g., flip-flops) and cells with high driving strength. Hence, the total area gain (the sum of the device area and the metal routing area for a benchmark circuit) is not uniform unlike with intra-cell stacking transformation. The physical design flow for handling these cells is similar to the intra-cell stacking case, where traditional 2D placement tools can be employed.

(4) For all the aforesaid cell transformation techniques we can observe that for the Input-Output (IO) pin density is increased per unit area. Hence designs with low to moderate routing needs can benefit from these techniques. On the other hand, for design requiring high routing resources, coarse-grain (block-level) partitioning is advisable.

3.5. Cell Delay Characterization

In this work we assume similar device characteristics for all the active layers. For example, consider a 45nm thin-box *silicon-on-insulator* technology for the top and bottom active layers. This assumption facilitates us to evaluate the impact of parasitic interconnect on the cell delay.

In the case of *intra-cell* transformations, the intermediate metal is not employed for routing the cell. Hence the delay of the 3D cell is similar to the 2D cell. However, the physical attributes of the 3D cell differ from the respective 2D cell depending on the applied intra-cell transformation technique (either intra-cell stacking or intra-cell folding).

In the case of *cell-on-cell* transformation, every cell (X) in the standard cell library has two versions, X_{top} (cell X placed in the top layer) and X_{bottom} (cell X placed in the bottom layer). Hence the number of cells in the standard cell library is doubled. All the bottom version of the cells employ highly resistive intermediate metal (tungsten) layer for cell routing, whereas the top version of cells employ regular metal (copper) for intra-cell routing. We characterized a few complex cells by taking into account the layout parasitics (using Calibre xRC [Mentor 2013]). We observe negligible delay degradation (less than 0.1%) for the bottom cell compared to the top version of the cell because of the high resistive intermediate metal layer. This agrees with the fact that the impact of local interconnect on the delay of the circuit is minimal. Hence for the rest of the work, we assume similar delay characteristics for the top and bottom cells. It has to be noted that the footprint of the top and bottom cell is kept the same (see Section 3.2).

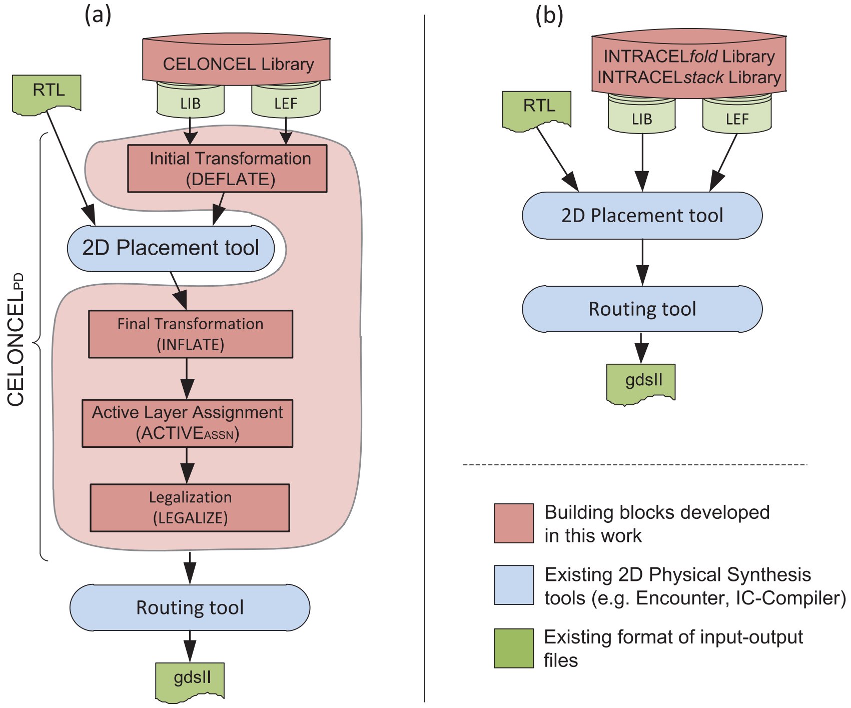

(a)
CELONCEL Library
LIB
LEF
RTL
Initial Transformation (DEFLATE)
2D Placement tool
Final Transformation (INFLATE)
Active Layer Assignment (ACTIVEASSN)
Legalization (LEGALIZE)
CELONCELPD
Routing tool
gdsII
(b)
INTRACELfold Library
INTRACELstack Library
RTL
LIB
LEF
2D Placement tool
Routing tool
gdsII
Building blocks developed in this work
Existing 2D Physical Synthesis tools (e.g. Encounter, IC-Compiler)
Existing format of input-output files

Fig. 9. Logic-to-layout design flow for (a) *cell-on-cell*; (b) *intra-cell* transformations.

4. DESIGN FLOW FOR CELL-ON-CELL AND INTRA-CELL TRANSFORMATIONS

Figure 9 presents the IC design flow, from logic to layout, for *cell-on-cell* and *intra-cell* transformations. Both *intra-cell stacking* and *intra-cell folding* transformation map a 2D cell to a 3D cell. "INTRACEL*stack* Library" and "INTRACEL*fold* Library" shown in the figure correspond to the 3D libraries designed by *intra-cell stacking* and *intra-cell folding* transformations. One of the key advantages of these techniques is the usability of the existing physical synthesis tools. INTRACEL design flow presented in Figure 9(b) is similar to the conventional 2D design flow, with an extra design effort in building the 3D cell libraries. In the rest of the article we refer to INTRACEL*stack* and INTRACEL*fold* as design flows for the two *intra-cell* techniques.

On the other hand, for *cell-on-cell stacking* transformation, new blocks are incorporated into the existing physical synthesis design flow. The CELONCEL design flow is presented in Figure 9(a). CELONCEL$_{\text{LIB}}$ is a novel standard library with the cells designed by *cell-on-cell stacking* (Section 3.2). CELONCEL$_{\text{PD}}$ has four main steps in the flow. The details of each step are described in the following section. The first two steps, DEFLATE and INFLATE transformations, help in employing existing 2D placement engines as a core placement tool. The physical information of the standard cell library (e.g., LEF file for Cadence tool flow) is altered with the DEFLATE transform. The width of the cells is reduced by half. At this stage most commercial/academic placement tools can be used to generate a virtual seed placement without any overlap among the transformed cells. With the INFLATE transform, the width of the cells is doubled in the seed placement result. This generates overlaps among the neighboring cells. The next step is ACTIVE$_{ASSN}$ that performs the active layer assignment of the cells. This step reduces the overlap among cells by an order of magnitude. Finally, minimum

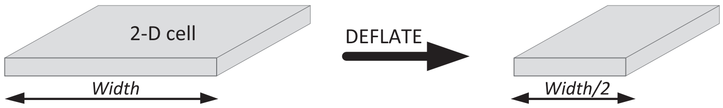


Fig. 10. DEFLATE transformation applied to all the library cells.

perturbation legalization is done to remove rest of the overlaps in the step LEGALIZE, thus completing the placement.

5. PHYSICAL DESIGN TOOL: CELONCEL$_{PD}$

From the previous section, we observe that existing 2D physical synthesis tools are sufficient for both the *intra-cell* transformations. In this section we explain the various steps of CELONCEL$_{PD}$, a novel physical synthesis tool for *cell-on-cell* transformation. The key observations we take forward with the *cell-on-cell* stacking is that the footprint and the delay of the cell, when placed in the top or bottom active layer, does not alter. Based on this, we conjecture that during physical synthesis the choice of active layer for each cell can be abstracted as a purely overlap issue without any impact on timing of the design. Once the active-layer-oblivious layout is obtained, the choice of active layer is made by a dedicated step. One of the critical benefits of isolating layer assignment and placement is that several physical synthesis steps that run during in-place timing optimization within placement can be performed transparently. These steps include aggressive buffer insertion, gate sizing, cell replication, clock tree generation, clock buffer placement, latch resizing, etc.

5.1. Initial Transformation (DEFLATE)

The DEFLATE transformation generates a virtual cell library from a given real cell library such that cell dimension and pin location are modified. Since we consider two active layers in our work, we shrink the width of each cell by half. Note that, to avoid placement errors, we also need to scale down the x-coordinates of the pin geometry defined for such a cell. Figure 10 shows an example of a 2D cell undergoing this initial transformation. At this stage, we can run any 2D placement engine to generate legalized placement consisting of transformed cells. Previous works such as Yang et al. [2003] and Chakraborty et al. [2009] have used the concept of cell expansion/deflation for congestion alleviation and transforming placement with blockages to contiguous placement, respectively.

ALGORITHM Sketch 1: DEFLATE

```
Input: Celoncel.lib, Celoncel.lef
Output: Virtual.lib, Virtual.lef
for each cell SC in Celoncel.lib and Celoncel.lef
    Scale down the width of SC by 50%
    Scale down the pin coordinates of SC by 50%
end
Write modified cells as Virtual.lef and Virtual.lib
Update verilog to use modified cell variant
/* Virtual.lib and Virtual.lef are employed by the 2D placement engines to do the initial
placement of the benchmark circuits*/
```

5.2. Final Transformation (INFLATE)

The INFLATE transform takes the placement information from the solution of a commercial placer on the virtual library and applies an inverse transform such that the width of the cells is expanded back to their original size. While doing this expansion,

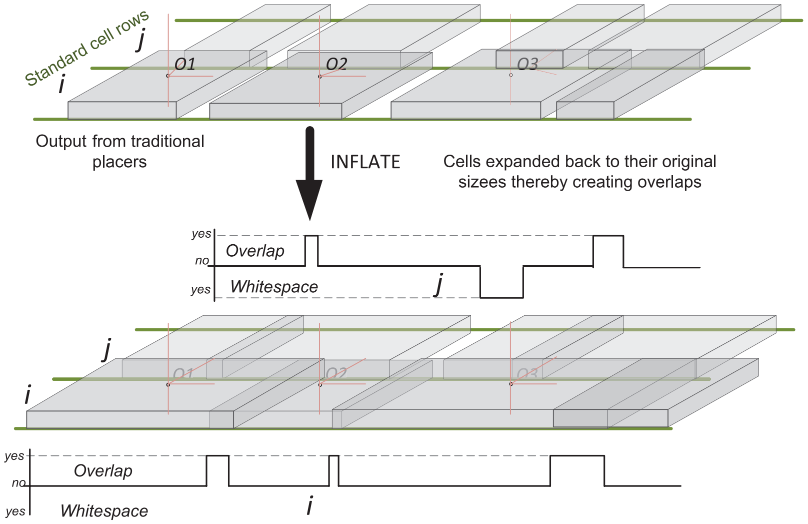

Standard cell rows
j
i
O1
O2
O3
Output from traditional placers
INFLATE
Cells expanded back to their original sizees thereby creating overlaps
yes
no
yes
Overlap
Whitespace
j
j
i
O1
O2
O3
yes
no
yes
Overlap
Whitespace
i

Fig. 11. INFLATE transformation shown for neighboring standard cell rows (i and j). The width of the cells is doubled, while keeping their centers (e.g., $O1$, $O2$, and $O3$) fixed. Morphing the cell width leads to overlaps and white spaces between the cells.

we assume that the *center* of the cell remains fixed. Due to expansion of the width of the cells, it is possible that part of some cells may lie outside the floorplan area. INFLATE also snaps such cells inside the placement area. Once the width of all the 2D cells is doubled, the placement has a huge number of overlaps. All the cells are now placed in only one active layer oblivious of the availability of another active layer. Figure 11 shows an example of a few cells, placed in two neighboring standard cell rows (i and j), undergoing INFLATE transform. The center of all the cells (for example, O1, O2, and O3 in the figure) remain fixed while undergoing the INFLATE transformation. The corresponding overlap and white space for both the rows are shown in Figure 11.

ALGORITHM Sketch 2: INFLATE

```
Input: initialPlacement /* initialPlacement is the layout from 2D Placement tool for a chosen
benchmark */
Output: inflatedPlacement
for each cell C in initialPlacement
    Scale up the width by 50% keeping its center of gravity fixed
    if (C not totally in the die area) /* Fix expanded cells protruding the die area */
        snap C to be inside the die
    end
end
Write modified cells as inflatedPlacement
```

5.3. Active Layer Assignment (ACTIVE$_{\text{ASSIGN}}$)

This step assigns the active layer of each cell with the objective of minimizing the overlap with the neighboring cells. During this stage, we assume that all cells are fixed in their active area plane at locations determined by the placer and only their z-dimension (i.e., active layer) can be modified. This problem can be formulated as a *Zero-One Linear Program* (ZOLP). Solving one large ZOLP for the entire chip is impossible due to runtime issues. However, owing to the structure of the placement and the type of overlaps resulting due to INFLATE transform, we can decompose the active layer assignment of all the cells as sequence of active layer assignment of each circuit row independently without sacrificing the optimality of the solution.

The objective function to minimize is the remaining overlap after active layer assignment is performed. A small remaining overlap directly means less movement of cells from their optimal location, determined by the placer, during the legalization step. Consider a floorplan with N standard cell rows of width W_{row}. Let us denote the set of cells lying in a circuit row i by C_i. Further, let *OV(a,b)* denote the 2D overlap between two cells a and b in the row. For each cell a, let X_a be the binary variable whose value determines the active layer in which the cell a will reside in the 3D layout, and W_a be the width of the cell a. With this terminology, the ZOLP can be formulated as

$$\begin{aligned}
\text{Minimize :}\quad & \sum_{i=1}^{N}\left(\sum \mathrm{OV(c1, c2)}\,(\mathrm{X_{c1}\ XNOR\ X_{c2}})\right) \quad \forall\, \mathrm{c1, c2} \in C_i \\
\text{Subject To :}\quad & \sum X_c \times W_c \leq W_{row} \qquad\quad \forall\, c \in C_i \\
& \sum (1 - X_c) \times W_c \leq W_{row} \quad \forall\, c \in C_i \\
& \mathrm{X_c} \in (0, 1).
\end{aligned}$$

ALGORITHM Sketch 3: ACTIVE$_{\text{ASSIGN}}$

```
Input: inflatedPlacement /* initialPlacement is the layout from the INFLATE transform */
Output: Place_layer0, Placer_layer1
for each row R in inflatedPlacement
    let CELLS be the cells in R
    Scan CELLS from left to right creating non-overlapping clusters C
end
for each cluster C of independent cells
    solve ZOLP-minimize_Overlap(C) to get active layer coordinates for the CELLS in C
end
for each active layer L /*2 in our example */
    Write the (x, y) coordinates of the Cells assigned to L
end
```

The possible overlap between two cells is multiplied by the XNOR of the binary variables associated with their layer assignment. Thus, only when the two cells are assigned to the same active layer, the corresponding overlap value adds to the cost function. The two set of constraints of the preceding formulation are to bound the cells within the footprint of the standard cell row in which they are placed. Figure 12 shows the active layer assignment of the inflated placement from Figure 11. The overlap between the neighboring cells is removed by spreading the cells across the bottom and top active layers. The cells in the row $i0$ and $j0$ are assigned to the bottom active layer and cells in the row $i1$ and $j1$ are placed in the top active layer.

Note that XNOR implies multiplication of two variables thus the formulation is no longer linear but quadratic in nature. However, by virtue of the variables being binary, each quadratic term can be decomposed into linear terms by adding an auxiliary binary variable as follows. Let X_A and X_B be the two binary variables whose product (i.e., X_AX_B) appears in the cost function expression. Introduce a new binary variable X_{AB} such that

$$\begin{aligned}
&\mathrm{C1\colon X}_A + \mathrm{X}_B \leq 1 + \mathrm{X}_{AB} \\
&\mathrm{C2\colon (1 - X}_A) + (1 - \mathrm{X}_B) \leq 2 - 2 * \mathrm{X}_{AB} \quad (\text{i.e., } \mathrm{X}_A + \mathrm{X}_B \geq 2 * \mathrm{X}_{AB}).
\end{aligned}$$

By replacing X_AX_B by X_{AB} and adding the aforesaid constraints to the ILP, the new problem formulation avoids multiplication of binary variables, for example, when $X_A = 0$

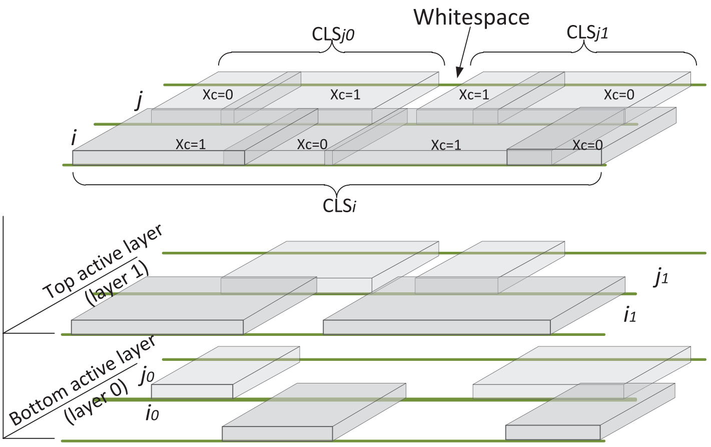


Fig. 12. Active layer assignment shown for neighboring standard cell rows (i and j). Overlap between the cells is removed by assigning the cells to different active layers with the help of the ZOLP formulation. White space between the cells helps in forming small clusters to speed up the ILP.

and $X_B = 0; X_A X_B = 0$. Constraint C1 leads to $0 \leq 1 + X_{AB}$, that is, $-1 \leq X_{AB}$. This does not force X_{AB} to a unique value, both $X_{AB} = 0$ and $X_{AB} = 1$ satisfy the equation $-1 \leq X_{AB}$. With the constraint C2, when $X_A = 0$ and $X_B = 0$, we have $0 + 0 \geq 2 * X_{AB}$. This forces X_{AB} to be 0. Hence with the two constraints, C1 and C2, binary variable X_{AB} is similar to $X_A X_B$. A truth table with the various combinations of X_A and X_B is shown next.

X_A	X_B	$X_A X_B$	X_{AB}		
			C1	C2	C1 ∩ C2
0	0	0	0, 1	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	1	1	1, 0	1

ZOLP Speedup. The number of binary variables in the ZOLP given before is equal to the number of cells in a circuit row. For big benchmarks and real-world designs, this number can be in the order of several thousands. To alleviate this problem, we can decompose the ZOLP problem by finding independent clusters as follows. We scan the layout of a row from left to right. Any time a white space is encountered, the ZOLP problem of the cells on left of the white space is solved independently to the ZOLP problem of the cells on the right. This is because during the active layer assignment cells cannot move in the 2D plane thus the cells on both sides of a white space cannot generate new overlaps between them and can be treated independently. For example, in Figure 12, two independent clusters (*CLSj0* and *CLSj1*) can be identified in the row j (*CLSj*) formed by the white space separating both the clusters.

5.4. LEGALIZATION: Removing Overlaps in Each Layer

Major overlaps are minimized in the layer assignment phase. However, some overlap may still remain, mainly due to different sizes of the cells. We perform legalization to remove these overlaps minimizing the cost function that is the total displacement of all cells in their own active layer from the optimal location determined by the placement tool (note that ACTIVE$_{ASSN}$ maintains the location of the cell). For this objective, the problem can be decomposed into solving each row *independently without loss of optimality of the overall solution*. For each row, legalization can be cast as a linear program as described next. Let us denote the set of cells lying in a circuit row on active layer 0 by CLS_0 and active layer 1 as CLS_1. Further, let the original and postlegalization x-location of cell a be denoted by XO(a) and X(a), respectively. Thus, the magnitude of movement of the cell is $|X(a) - XO(a)|$ due to legalization. Note that during legalization no cell changes its circuit row or active layer, therefore, the y- and z-coordinate of each cell do not change due to legalization. We also denote the width of cell a by W(a) and the cell on its right side on the same active layer by RT(a). The leftmost and the rightmost cell in the row are denoted by L0 and R0 for the bottom active layer, L1 and R1 for the top active layer. The x-coordinate of the left and right extreme of the span of the row is represented by START and END. With this terminology, the LP for legalization can be written as follows.

$$\text{Minimize: } \sum |X(a) - XO(a)| \quad \forall \quad a \text{ in } \{CLS_0\} \cup \{CLS_1\}$$

Subject To:

$$\begin{aligned} X(a) + W(a) &\leq X(RT(a)) \, \forall \text{ a in } \{CLS_0\} \\ X(L_0) &\geq START \\ X(R_0) + W(R_0) &\leq END \\ X(a) + W(a) &\leq X(RT(a)) \, \forall \text{ a in } \{CLS_1\} \\ X(L_1) &\geq START \\ X(R_1) + W(R_1) &\leq END \end{aligned}$$

The cost function is the sum of the displacement of all cells. The formulation can be easily changed to minimizing the largest displacement (instead of current form to minimize total displacement). There are two sets of constraints for the LP, one for each active layer. Though the function $|X(a) - X0(a)|$ is nonlinear, the previous LP can still be solved by replacing the function by a variable $MOVE_a$ and the following constraints can be added.

$$\begin{aligned} X(a) - XO(a) &\leq MOVE_a \\ X(a) - XO(a) &\geq -MOVE_a \end{aligned}$$

Adding the aforesaid constraints forces the variable $MOVE_a$ to behave like the absolute distance between X(a) and X0(a) when the objective is to minimize $|X(a) - X0(a)|$.

ALGORITHM Sketch 4: Legalization

```
Input: Place_layer0, Place_layer1
Output: legalPlace_bot, legalPlace_top
for each active layer L
    Write LP for Legalization of L
    Solve LP to get new coordinates
end
Run legality checker to ascertain legal layout
Write the def file for bottom and top active layer
```

6. EXPERIMENTAL SETUP AND RESULTS

The core components shown in Figure 9 were written in C++ and compiled with g++ 4.4.4. We used open-source MILP solver Gurobi [2013] as our ZOLP and LP solver engine. Synopsys Design Compiler (A-2007.12-SP4) [SDC 2013] was used for

Table III. Characteristics of Benchmarks Used in Our Experiments

Benchmark		*#Nets*	*#Cell*	*#Pins*	*Dmin* (*ns*)
Circuit	function				
LDPC	Low Density Parity Check decoder	48K	44K	4100	6.904
WbC	Wishbone Interconnect Matrix IP core	29K	27K	2546	2.382
B19	Synthetic design	99K	87K	77	4.305
Ethernet	Ethernet	43K	42K	210	14.738
Des	Data Encryption Standard	59K	56K	298	2.532

The coloumns denote the number of nets, cells and pins. *DMIN* gives the delay of the circuit under ideal interconnect conditions (with resistance and capacitance set to zero).

mapping the RTL of the benchmarks onto the target standard cell library. Cadence SOC Encounter (v8.1) [Encounter 2013] was used as the physical synthesis engine to generate the virtual seed placement. Timing analysis was performed with Synopsys PrimeTime (D-2009.12-SP2) using the capacitance table of the standard cell library.

In this study we have mapped the open-source 45nm Nangate [Nangate 2013] (v1.3) library to different 3D libraries by changing only the physical attributes of the cell. For a fair comparison to study the interconnect delay for all the four cases (2D and all the three 3D variants) we assume similar delay characteristics for all the cells while the physical attributes vary depending on the layout style. INTRACEL*stack* has cells, built in 3D by intra-cell stacking transformation, with 30% less height. INTRACEL*fold* has cells built in 3D where the width of the cells is altered as per the discussion presented in Section 3.3. CELONCEL$_{\text{LIB}}$ has cells, which are 2D cells with a capability of either accommodating a cell on the top or below, that span 25% more in height.

To evaluate the performance of the various cell transformations for 3DMI technology, we used a broad range of designs, from interconnect-dominated circuits, such as *Low-Density Parity Check* (LDPC) decoder, to the complex synthetic design b19, comprising of almost 100K nets. The majority of the designs are obtained from opencores [Opencores 2013], while the big synthetic design b19 is taken from the ITC99 suite [Itc99 1999]. The design parameters are given in Table III, which reports the number of nets, cells, and pins in the input RTL of the benchmarks.

The last column (*Dmin*) indicates the minimum possible delay achievable if no changes in the circuit netlist are allowed during placement. This value was obtained by performing timing analysis of the benchmark with the value of interconnect resistance and capacitance set to zero. In the absence of any netlist change (i.e., resizing, buffering, logic duplication, etc.), the virtue of a placement can be gauged by observing how closely the postplacement timing tracks *Dmin* for the circuit. Note that if netlist changes are allowed, the physical synthesis engine can achieve delays lower than *Dmin*. However, in that case the number of nets, cells, and pins can change.

CELONCEL$_{\text{PD}}$ is configured in three modes: in the first mode, *wirelength-driven placement* is run, in the second mode, *timing-driven placement* is run, and in the third mode, timing-driven optimization along with *in-place optimization* is run which performs various optimizations such as buffer insertion, gate sizing, cell replication, etc. Note that *Dmin* sets the starting seed value for timing optimization. In order to check the runtime of the complete CELONCEL$_{\text{PD}}$ design flow, we have created bigger benchmarks (∼2M gates) with multiple instances of the existing benchmarks (Table VI).

7. RESULTS AND DISCUSSION

In this section we present the performance improvement when mapping a 2D circuit to 3DMI technology with various cell transformation techniques, as explained in Section 3.

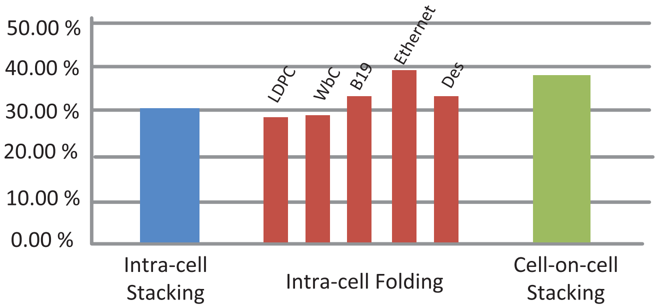


Fig. 13. Percentage improvement in the total area for all the cases.

First, we study the performance gain in terms average wirelength after placing the circuit. Similar to the existing 3D placement tools, based on TSV technology, we run the CELONCEL$_{PD}$ in *wirelength-driven* placement. Second, we study the improvement in the timing of the circuits, when driven in *timing-driven placement* mode. With the help of *in-place optimization* mode, we also study the timing improvement after physical synthesis techniques like buffer insertion, gate sizing, repeater insertion, and cell replication.

7.1. Area Comparison Employing Various 2D to 3D Transformations

In the first instance, we compared the area gain of the various 3D transformations with the 2D case. Figure 13 shows the percentage improvement in total chip area. Note that the area analysis presented here is carried under wirelength-driven optimization mode. Intra-cell stacking decreases the cell height by ~30%. The reduced cell height reflects in the increase of the number of standard cell rows for a given floorplan. Hence we can observe a 30% area gain. In the case of cell-on-cell stacking, the cell area is increased by 25%, however, we have twice the floorplan area to stack the cells on top of each other. Hence, an overall chip area improvement of 37.5% is achieved with cell-on-cell stacking. On the other hand, for intra-cell folding transformation, the decrease in area depends on the type of the cell and its respective fanin and fanout (see Section 3.3). Hence the area gain of the example can place the cells within 60% of the planar area. This reduced area of the circuit depends on the number of area-efficient cells in the synthesized netlist. For example, the best transformation technique for Ethernet circuit is intra-cell folding as reflected in the circuit depends on the number of area-efficient cells in the synthesized netlist for reduced wirelength as well as improved timing. From Figure 13, we can observe that with the CELONCEL flow we achieve better area gain when compared to the INTRACEL flow for most of the designs, with the only exception of Ethernet benchmark. Among the two intra-cell techniques, intra-cell folding seems to be a better choice.

7.2. Wirelength-Driven Placement Optimization

In the *wirelength-driven placement* mode, the physical design tool places the cells of the given netlist in such a way that the average interconnect length is minimized. Table IV reports the average wirelength for the various benchmark circuits. When comparing all the benchmark circuits, it can be inferred from the table that interconnect plays a dominant role in *Low-Density Parity Check* (LDPC) decoder. Though the number of cells and nets are similar for LDPC and *Ethernet* circuits (see Table III), the average wirelength for the LDPC circuit after the placement phase is 3.5× higher than *Ethernet*. Percentage improvement in wirelength, for all the benchmarks, when employing *intra-cell* and *cell-on-cell* design techniques is plotted in Figure 14. In general, the results

Table IV. Improvement in Wirelength of the Benchmark Circuits Subjected to Wirelength-Driven Optimization

Circuits	Planar (2D)	Intra-Cell Stacking	Intra-Cell Folding	Cell-on-Cell
LDPC	15.4E+05um	13.8E+05um	15.0E+05um	13.7E+05um
WbC	3.70E+05um	3.53E+05um	3.31E+05um	3.24E+05um
B19	8.29E+05um	7.24 E+05um	6.99E+05um	6.89 E+05um
Ethernet	4.21 E+05um	3.72 E+05um	3.30E+05um	3.43 E+05um
Des	5.84 E+05um	5.08 E+05um	4.74+05um	4.54 E+05um

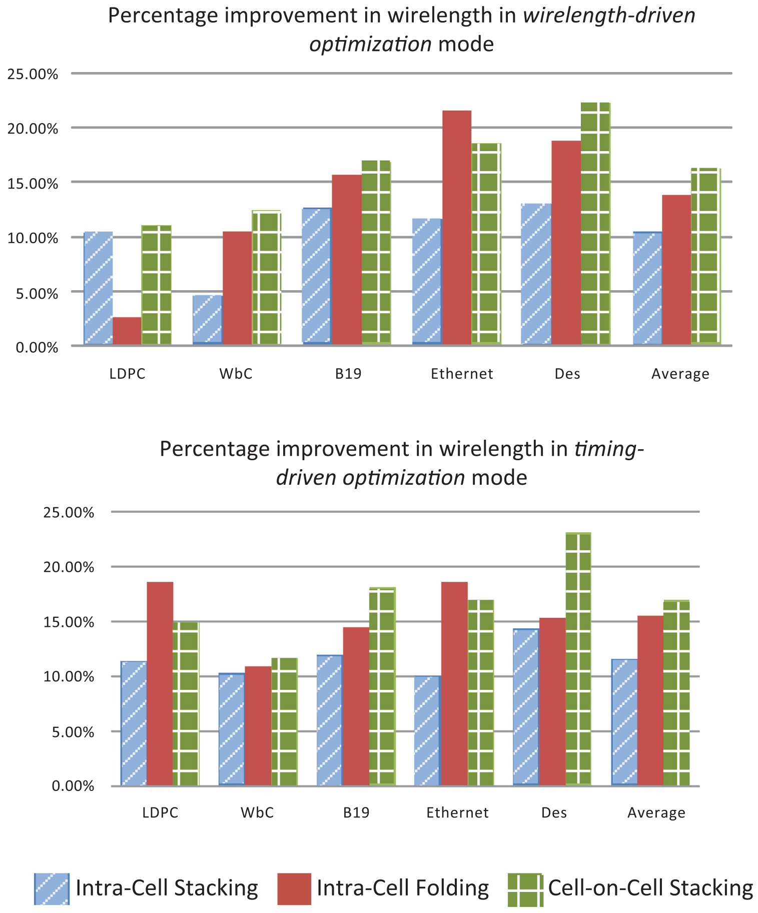


Fig. 14. Performance improvement in wirelength of various benchmark circuits when subjected to wirelength-driven and timing-driving placement.

for *cell-on-cell* are better than *intra-cell* techniques. The average improvement in wirelength over a 2D case employing CELONCEL, INTRACEL*stack*, and INTRACEL*fold* are 16.2%, 10.5%, and 13.9%, respectively.

7.3. Timing-Driven Placement Optimization

In the timing-driven placement mode, the placer is allowed to move the cells to reduce timing without changing the netlist in any manner. Simulation results for *timing-driven optimization* are summarized in Table V. In this table we report total wirelength,

Table V. Wirelength, Delay and Power Information of Benchmark Circuits for Various Cases when Subjected to Timing-Driven and In-Place Optimization

		Timing Driven Optimization				*Timing Driven In-Place Optimization*			
Circuits	Metrics	Planar (2D)	IntraCell Stacking	IntraCell Folding	Cell-on-Cell	Planar (2D)	IntraCell Stacking	Intra-Cell Folding	Cell-on-Cell
LDPC	*Wirelength*(*um*)	1.67E+06	1.48E+06	1.36e+06	1.42E+06	1.83E+06	1.60E+06	3.02E+06*	1.54E+06
	Delay (*ns*)	6.877	6.904	6.866	6.904	2.461	2.421	4.692*	2.129
	Power(*mW*)	1147	1064	1092	1018	1554	1461	2058*	1470
WbC	*Wirelength*(*um*)	3.77 E+05	3.38E+05	3.36e+05	3.33E+05	3.76E+05	3.64E+05	3.34E+05	3.33E+05
	Delay (*ns*)	4.661	4.628	4.342	4.449	1.039	1.041	1.096	1.083
	Power(*mW*)	100.4	99.93	99.45	99.64	70.63	71.14	70.69	72.05
B19	*Wirelength*(*um*)	8.60E+05	7.57E+05	7.36E+05	7.04E+05	7.93E+05	7.00E+05	6.77E+05	6.49E+05
	Delay (*ns*)	4.723	4.691	4.694	4.691	4.224	4.219	4.184	4.185
	Power(*mW*)	434.5	429.7	429.1	425.2	337	312.7	325.3	314.7
Ethernet	*Wirelength*(*um*)	4.3E+05	3.87E+05	3.50E+05	3.57E+05	4.94E+05	4.38E+05	3.91E+05	3.97E+05
	Delay (*ns*)	30.598	29.063	26.194	26.427	1.252	1.281	1.223	1.336
	Power(*mW*)	175.3	170.6	166	165.6	133.2	132.7	137.3	130.7
Des	*Wirelength*(*um*)	6.06E+05	5.19E+05	5.13E+05	4.66E+05	6.71E+05	5.81E+05	22.9E+5*	5.45E+05
	Delay (*ns*)	3.854	3.944	3.731	3.39	1.132	0.971	4.021*	1.016
	Power(*mW*)	535.8	525.3	524.8	517.5	620.2	608.2	920.2*	580.5

*For LDPC decoder and Des circuits, few convergence issues were observed while carrying timing-driven in-place optimization for *intra-cell folding* transformation.

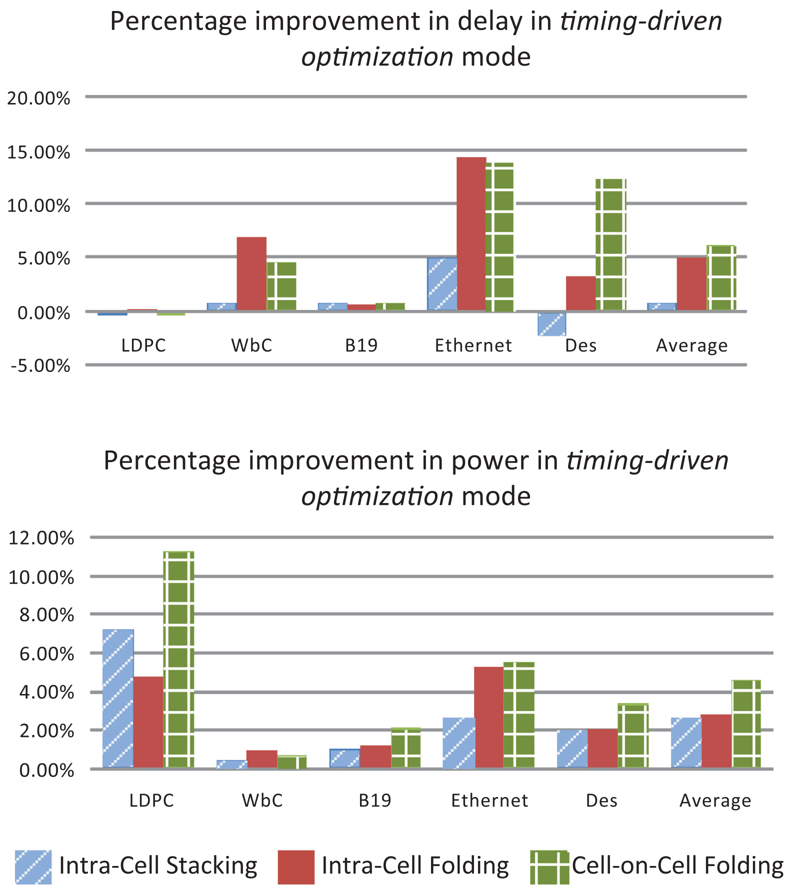


Fig. 15. Performance improvement of various benchmark circuits when subjected to timing-driving placement.

total power, and critical-path delay of different benchmarks. All numbers are reported using Cadence Encounter (EDI) v9.1 (2010 release). The power numbers include all components of the power dissipation, namely leakage power, switching power, and internal power. Due to smaller die sizes when CELONCEL or INTRACEL flows are used, we conjecture that critical-path delay should also decrease accordingly. Averaged over all benchmarks, the critical-path delay of the circuit using the CELONCEL flow is 6.1% smaller than the 2D circuit. However, the INTRACEL*stack* does not exhibit any consistent trend compared to the 2D case with the average improvement in the critical-path delay less than 1%. On the other hand, INTRACEL*fold* shows consistent gains similar to the CELONCEL case with an average improvement of 5%. For this set of experiments, the timing constraint for each benchmark was set to be equal to the theoretical maximum performance that can be achieved. The maximum performance is obtained by setting interconnect resistance and capacitance equal to zero and running the timing analysis.

The percentage improvement in performance (wirelength, delay, and power) for all the benchmarks is plotted in Figure 14 and Figure 15. From Figure 15, we can observe that the timing optimization has almost no impact in the case of the LDPC decoder. The

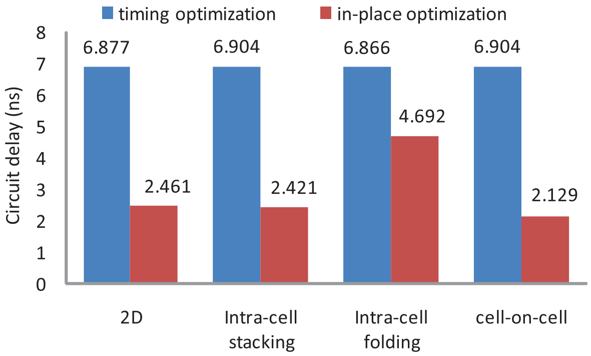


Fig. 16. Delay reduction of an LDPC decoder with in-place optimization.

2D and 3D cases achieve very similar delays. This could be attributed to the dominance of interconnect for the LDPC circuit. Though the delays are similar, the overall power consumption is reduced for all the 3D cases when compared to the 2D case (shown in Figure 15). For instance, with *cell-on-cell* transformation, LDPC decoder consumes 10.5% less power compared to the 2D case.

7.4. Timing-Driven Placement with *In-Place* Optimization

For completeness, we have also looked into the timing-driven placement with in-place optimization. This case study showcases the adaptability of the CELONCEL design flow with the existing 2D placement engines. With in-place optimization, the placer has the flexibility to apply any synthesis or timing optimization transforms to the netlist on-the-fly to improve the timing. For these sets of experiments, we set the timing constraint corresponding to an unachievable speed (10 GHz). In this manner, we can test the best performance that each of the techniques can produce. Compared to the 2D case, employing CELONCEL can reduce the critical-path delay even further by 2.75%. Similarly, by using INTRACEL, the critical-path delay can be reduced by approximately 2.7%. Note that this improvement in critical-path delay is additional to the best solution obtained using the 2D case, thus hard to obtain.

Figure 16 shows the reduced delay of the LDPC circuit with in-place optimization in all the cases (2D and 3D cases). With cell-on-cell transformation the minimum possible delay of 2.129ns is realized. Though the 2D case has a slightly better delay over cell-on-cell in the timing-driven mode, we see better delay characteristics with in-place optimization. The reason could be related to the double amount of neighboring slots with cell-on-cell stacking. Since more neighboring cells can be accommodated next to each other (see Figure 8) with cell-on-cell, we could speedup the circuit by 13.49% when compared to the 2D case.

7.5. Runtime of the CELONCEL$_{PD}$

All benchmarks are run on an Intel Xeon CPU X5650 Linux workstation running at 2.67 GHz. The runtime of the CELONCEL$_{PD}$, running on a single thread, for various benchmarks in timing-driven mode is shown in Table VI. The total time taken by the CELONCEL$_{PD}$ is the sum of the time taken by the 2D engine (Encounter in our experiment) as well as the time taken for solving our ILP formulation for active layer assignment and legalization steps.

On an average, active layer assignment and legalizer takes 11.4% of the total time taken for 3D placement. Our flow has also been tested for bigger benchmarks which

Table VI. Total Runtime with Celoncel Placer, which Includes the Time Taken by the 2D Engine and the Time Taken by Active-Layer Assignment and Legalizer Step

Benchmarks	# of Gates	*CELONCELPD*		Δt_{3D} %
		2D Engine	*ACTIVE$_{Assign}$ + Legalize*	
WbC	27K	101.295s	12.96s	11.35
Ethernet	42K	175.623s	23.434s	11.77
LDPC	44K	170.219s	19.892s	10.46
DES	56K	198.331s	24.314s	10.97
B19	87K	285.581s	48.015s	14.39
*B19_10X**	*870K*	3786.416s	515.864s	11.99
*LDPC_20X**	*880K*	6020.149s	445.53s	14.51
*LDPC_40X**	*1.76M*	12506.88s	1189.49s	8.69
*DES_LDPC_B19_10X**	*~2M*	5274.994s	491.493s	8.52

Avg. 11.4%

*These benchmarks are made up by instantiating more modules. For example B19_10X has 10 instances of B19. DES_LDPC_B19_10X has 10 instances of DES, LDPC and B19.

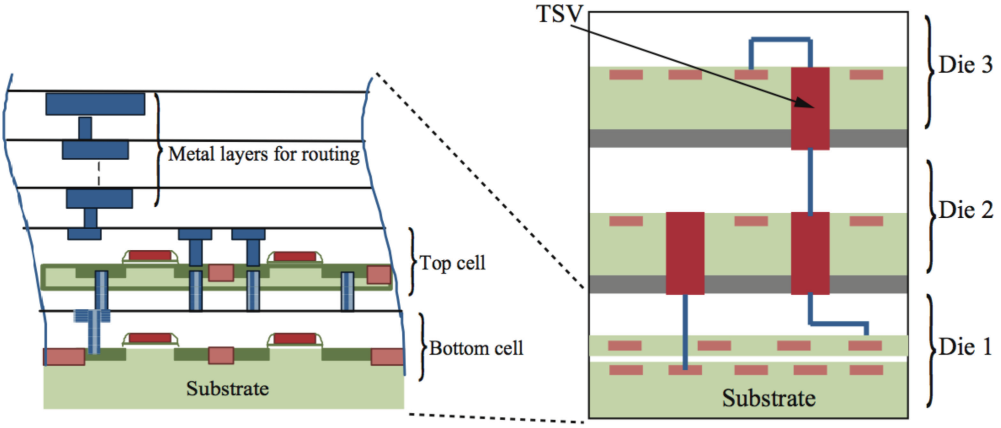


Fig. 17. Cointegration of 3DMI and 3D-TSV technologies.

were created by instantiating many modules of the basic blocks. For the largest benchmark, DES_LDPC_B19_10X (~2M), our ILPs were solved in ~8 minutes. The runtime benefit comes from clustering only the overlapping cells in each row and solving their respective ILP formulation for active layer assignment and legalization.

8. TECHNOLOGY AND DESIGN CHALLENGES

In this section we highlight the future technology and design challenges for 3D monolithic integrated circuits.

8.1. Cointegration of 3D-TSV and 3DMI Technologies

In the near future we envisage cointegration of both 3DMI and 3D-TSV technologies. The design methodology proposed in this work studies the physical design technique for fine-grain partitioning of circuits, which is feasible with 3DMI technology and cannot be extended to 3D-TSV technology as the size of the TSVs is large (~1 um). Hence it is beneficial to apply 3DMI technology to leverage the benefits from fine-grain partitioning of the circuit and 3D-TSV technology for benefitting from coarse-grain partitioning. Figure 17 illustrates the cointegration of 3DMI and 3D-TSV technologies,

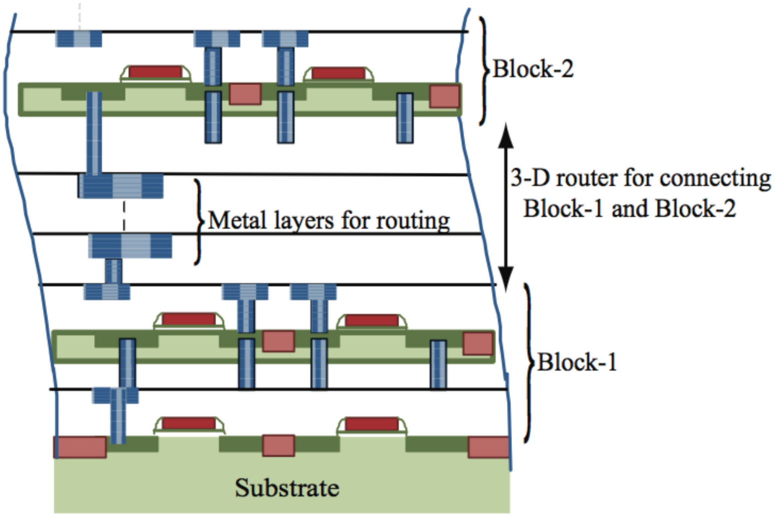


Fig. 18. 3DMI with multiple active layers.

with a die fabricated with 3DMI technology being a part a of 3D system realized with 3D-TSV technology.

In order for 3DMI to replace 3D-TSV technology, further advancement in 3DMI technology is required. State-of-the-art 3DMI technology employs tungsten, which is more resistive than copper (~3X times). In our design technique we have minimized the use of tungsten by employing it only for local interconnections. Hence 3DMI technology with low resistive intermediate metal is required for building complex 3D systems.

8.2. Scaling of 3DMI Technology

In this work, we have presented fine-grain cell transformation techniques for 3DMI technology with two active layers. Though the presented techniques are scalable, in theory, fine-grain stacking of more than two active layers does not lead to further benefits in area. Moreover, the problem of routing congestion is further escalated. More than two active layers can be considered when an intermediate metal layer is employed for routing. Figure 18 illustrates possible design techniques for 3 active layers. Fine-grain partitioning of a circuit block_(*Block-1*) can be carried with the bottom two active layers, by applying either the INTRACEL or CELONCEL design technique. Intermediate metal layers between the 2nd active layer and 3rd active layer are employed to connect both the blocks (*Block*-1 and *Block*-2). Existing 3D routing tools, proposed for 3D-TSV technology, can be used to connect the blocks.

8.3. Routing Congestion

Fine-grain partitioning of the circuit with 3DMI technology results in a smaller footprint of the active area when compared to a planar implementation. Since the number of signal routes feeding the IO pins of the cells remains the same, the effective density of IO pins per unit area is increased. For example, with 3DMI technology with two active layers, all the IO pins should be accessible in half the active area (of Metal-2) as compared to the 2D case. For designs with low to moderate routing needs, 3DMI technology is a plausible contender. However, for designs requiring high routing resources, this increased pin density will likely cause routing congestion. We envisage further research into 3DMI technology for such routing-intense circuits.

9. CONCLUSIONS AND FUTURE WORK

3DMI technology, offering 3D contacts with sizes in the order of ∼100nm, is an effective vehicle for future gigascale circuits. In this work, we focused on various standard cell transformation techniques and their corresponding physical synthesis flow for ASICs. Intra-cell stacking realizes a standard cell in 3D by stacking the p-diffusion area over n-diffusion area (or vice versa), thereby reducing the cell height and thus the die area. On an average, the wirelength, critical-path delay, and the die area are improved by 10.45%, 1%, and 29%, respectively. Intra-cell folding realizes a standard cell in 3D by folding the active area within the cells, thereby reducing the width of the cell. On an average, the wirelength, critical-path delay, and the die area are improved by 13.9%, 5%, and 32%, respectively. Cell-on-cell stacking, on the other hand, allows cells to be placed on top of each other considering the pin access issues. Since the current placement tools cannot be applied for cell-on-cell stacking, unlike for intra-cell transformations, a physical design tool ($\text{CELONCEL}_{\text{PD}}$) was proposed that transforms the monolithic 3D placement problem into a virtual 2D problem solved using existing 2D placers. A highly parallelizable zero-one linear program formulation is used for layer assignment followed by linear-program-based minimum perturbation for high-quality 3D layout. As compared to traditional 2D physical synthesis flow, with CELONCEL we can reduce the wirelength, critical-path delay, and the die area by 15%, 6.1%, and 37.5%, respectively.

In this work we dealt with *Silicon-On-Insulator* (SOI) active layers separated by an intermediate metal layer, henceforth assuming similar delay characteristics for a cell when placed in either the top or bottom layer. However, a cost-effective 3DMI technology in the near future would be Si-bulk at the bottom layer and a thin SOI active layer at the top. This leads to difference in the delay characteristics of a cell when placed in different layers. For example, the cell when placed at the bottom layer will be faster (high performance) and when placed at the top layer will consume low power (SOI technology). Future extension of the $\text{CELONCEL}_{\text{PD}}$ will be adopted such that the cells on the critical path will be assigned to the bottom layer during the layer assignment step.

REFERENCES

Batude, P., Vinet, M., Pouydebasque, A., Clavelier, L., Previtali, B., et al. 2008a. Enabling 3D monolithic integration. In *Proceedings of the Electro-Chemical Society Spring Meeting (ECS'08).*Vol. 16, 47.

Batude, P., Jaud, M.-A., Thomas, O., Clavelier, L., Pouydebasque, A., et al. 2008b. 3d cmos integration: Introduction of dynamic coupling and application to compact and robust 4t sram. In *Proceedings of the IEEE International Conference on Integrated Circuit Design and Technology and Tutorial (ICICDT'08).* 281–284.

Batude, P., Vinet, M., Pouydebasque, A., Le Royer, C., Previtali, B., et al. 2009a. Advances in 3d cmos sequential integration. In *Proceedings of the IEEE International Electronic Devices Meeting (IEDM'09).* 1–4.

Batude, P., Vinet, M., Pouydebasque, A., Le Royer, C., Previtali, B., et al. 2009b. GeOI and soi 3d monolithic cell integrations for high density applications. In *Proceedings of the Symposium on VLSI Technology.* 166–167.

Batude, P., Vinet, M., Xu, C., Previtali, B., Tabone, C., et al. 2011. Demonstration of low temperature 3d sequential fdsoi integration down to 50nm gate length. In *Proceedings of the IEEE Symposium on VLSI Technology.* 158–159.

Bobba, S., Chakraborty, A., Thomas, O., Batude, P., Pavlidis, V. F., and de Micheli, G. 2010. Performance analysis of 3D monolithic integrated circuits. In *Proceedings of the IEEE International 3D Systems Integration Conference (3DIC'10).* 1–4.

Bobba, S., Chakraborty, A., Thomas, O., Batude, P., Ernst, T., et al. 2011. CELONCEL: Effective design technique for 3d monolithic integration targeting high performance integrated circuits. In *Proceedings of the 16th Asia and South Pacific Design Automation Conference (ASP-DAC'11).* 336–343.

Chakraborty, A., Kumar, A., and Pan, D. Z. 2009. Regplace: A high quality open-source placement framework for structured asics. In *Proceedings of the 46th ACM/IEEE Design Automation Conference (DAC'09).* 442–447.

Chan, T. F., Cong, J., Shinnerl, J. R., Sze, K., and Xie, M. 2006. MPL6: Enhanced multilevel mixed-size placement. In *Proceedings of the International Symposium on Physical Design (ISPD'06).* ACM Press, New York, 212–214.

Cong, J., Luo, G., Wei, J., and Zhang, Y. 2007. Thermal-aware 3d ic placement via transformation. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC'07).* IEEE Computer Society, Washington, DC, 780–785.

Das, S., Chandrakasan, A., and Reif, R. 2003. Design tools for 3d integrated circuits. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC'03).* 53–56.

Deng, Y. and Maly, W. P. 2001. Interconnect characteristics of 2.5-D system integration scheme. In *Proceedings of the International Symposium on Physical Design (ISPD'01).* ACM Press, New York, 171–175.

Encounter. 2013. SOC encounter tool. http://www.cadence.com/products/di/soc_encounter/pages/default.aspx.

Gurobi. 2013. Gurobi optimization. http://www.gurobi.com/.

Havemann, R. H. and Hutchby, J. A. 2001. High-performance interconnects: An integration overview. *Proc. IEEE 89*, 5, 586–601.

Ieong, M., Guarini, K. W., Chan, V., Bernstein, K., Joshi, R., Kedzierski, J., and Haensch, W. 2003. Three dimensional cmos devices and integrated circuits. In *Proceedings of the IEEE Custom Integrated Circuits Conference*. 207–213.

ITC99. 1999. http://www.cerc.utexas.edu/itc99-benchmarks/bendoc1.html.

Itrs. 2009. www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables.

Jiang, Z.-W., Cheny, T.-C., Hsuy, T.-C., Chenz, H.-C., and Changyz, Y.-W. 2006. Ntuplace2: A hybrid placer using partitioning and analytical techniques. In *Proceedings of the International Symposium on Physical Design (ISPD'06).* 215–217.

Jung, S.-M, Jang, J., Cho, W., Moon, J., Kwak, K., et al. 2004. The revolutionary and truly 3-dimensional 25f2 sram technology with the smallest s3 (stacked single-crystal si) cell, 0.16um2, and sstft (stacked single-crystal thin film transistor) for ultra high density sram. In *Proceedings of the Symposium on VLSI Technology, Digest of Technical Papers.* 228–229.

Jung, S.-M., Lim, H., Yeo, C., Kwak, K., Son, B., et al. 2007. High speed and highly cost effective 72m bit density s3 sram technology with doubly stacked si layers, peripheral only cosix layers and tungsten shunt w/l scheme for standalone and embedded memory. In *Proceedings of the Symposium on VLSI Technology*. 68–69. http://toc.proceedings.com/02217webtoc.pdf.

Kim, H. S., Xue, L., Kumar, A., and Tiwari, S. 2002. Fabrication and electrical properties of buried tungsten structure for direct three dimensional integration. In *Proceedings of the International Conference on Solid State Device and Materials (SSDM'02).*

Kim, D. H., Athikulwongse, K., and Lim, S. K. 2009. A study of through-silicon-via impact on the 3d stacked ic layout. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers (ICCAD'09).* 674–680.

Koester, S. J., Young, A. M., Yu, R. R., Purushothaman, S., Chen, K.-N., La Tulipe, D. C., Rana, N., Shi, L., Wordeman, M. R., and Sprogis, E. J. 2008. Wafer-level 3d integration technology. *IBM J. Res. Devel. 52*, 6, 583–597.

Lin, M., El-Gamal, A., Lu, Y.-C., and Wong, S. 2007. Performance benefits of monolithically stacked 3d fpga. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst. 26*, 2, 216–229.

Loh, G. H., Xie, Y., and Black, B. 2007. Processor design in 3d die-stacking technologies. *IEEE Micro 27*, 3, 31–48.

Mentor. 2013. Calibre xrc. http://www.mentor.com/products/ic_nanometer_design/verification-signoff/circuit-verification/calibre-xrc/.

Mit. 2013. 3D Design Kits, version 3DEM.

Nangate. 2013. 45nm library. http://www.nangate.com/.

Opencores. 2013. www.opencores.org.

Pavlidis, V. and Friedman, E. 2009. *Three-Dimensional Integrated Circuit Design*. Morgan Kaufmann.

Roy, J. A., Papa, D. A., Adya, S. N., Chan, H. H., Ng, A. N., Lu, J. F., and Markov, I. L. 2005. Capo: Robust and scalable open-source min-cut floorplacer. In *Proceedings of the International Symposium on Physical Design (ISPD'05).*

Saraswat, K. C. 2010. 3-D ics: Motivation, performance analysis, technology and applications. In *Proceedings of the 17th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA'10)*. 1–6.

SDC. 2013. Synopsys design compiler. http://www.synopsys.com/home.aspx.

SILLON, N., ASTIER, A., BOUTRY, H., DI CIOCCIO, L., HENRY, D., AND LEDUC, P. 2008. Enabling technologies for 3D integration: From packaging miniaturization to advanced stacked ics. In *Proceedings of the IEEE International Electron Devices Meeting (IEDM'08)*. 1–4.

SON, Y.-H., LEE, J.-W., KANG, P., KANG, M.-G., KIM, J.- B., ET AL. 2007. Laser induced epitaxial growth (leg) technology for high density 3d stacked memory with high productivity. In *Proceedings of the IEEE Symposium on VLSI Technology, Digest of Technical Papers*. 80–81.

TEZZARON. 2013. Wafer stack with super contacts. http://www.tezzaron.com/about/PhotoAlbum/Products/Wafer_Pair_Super-Contacts.html.

WONG, S., EL-GAMAL, A., GRIFFIN, P., NISHI, Y., PEASE, F., AND PLUMMER, J. 2007. Monolithic 3d integrated circuits. In *Proceedings of the International Symposium on VLSI Technology, Systems and Applications*. 1–4.

YANG, X., WANG, M., KASTNER, R., GHIASI, S., AND SARRAFZADEH, M. 2003. Congestion reduction during placement with provably good approximation bound. *ACM Trans. Des. Autom. Electron. Syst. 8*, 3, 316–333.

ZHOU, L., WAKAYAMA, C., AND SHI, C.-J. R 2006. CASCADE: A standard super-cell design methodology with congestion-driven placement for three-dimensional interconnect-heavy very large scale integrated circuits. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst. 26*, 7, 1270–1282.

Received October 2011; revised May 2012; accepted June 2012